HOGER INSTITUUT VOOR WIJSBEGEERTE

KATHOLIEKE
UNIVERSITEIT
LEUVEN

COMPUTERBEWIJZEN IN DE WISKUNDIGE PRAKTIJK

Promotor: prof. dr. Leon Horsten

verhandeling aangeboden tot het verkrijgen van de graad van Licentiaat in de Wijsbegeerte door:

Koen Vervloesem

Leuven, 2007

HOGER INSTITUUT VOOR WIJSBEGEERTE

KATHOLIEKE
UNIVERSITEIT
LEUVEN

COMPUTERBEWIJZEN IN DE WISKUNDIGE PRAKTIJK

Promotor: prof. dr. Leon Horsten

verhandeling aangeboden tot het verkrijgen van de graad van Licentiaat in de Wijsbegeerte door:

Koen Vervloesem

Leuven, 2007

Dankwoord

Deze eindverhandeling is niet alleen het resultaat van mijn eigen inspanningen, maar ook van heel wat hulp die ik gekregen heb. Ik wil dan ook graag enkele mensen bedanken die mij tijdens het schrijven van mijn eindverhandeling hebben gesteund. Ik bedank in het bijzonder mijn promotor Leon Horsten, die me voor het kiezen van dit onderwerp geïnspireerd heeft door me de boeken *New Directions in the Philosophy of Mathematics* en *Philosophy of Mathematics: an Anthology* in mijn handen te drukken. Ik bedank hem ook voor de vruchtbare discussies en de waardevolle commentaar op mijn ideeën en kladversies. Hij spoorde me ook aan om mijn idee voor Hoofdstuk 9, *Wiskundige concepten in computerbewijzen* uit te werken voor een presentatie op de *Perspectives on Mathematical Practices 2007* conferentie.

Dank ook aan de organisatoren van de *Perspectives on Mathematical Practices 2007* conferentie die mijn voorstel voor een presentatie aanvaardden. Dank ook aan verschillende deelnemers van de conferentie, waarmee ik interessante discussies had over mijn presentatie. Mijn bijzondere dank gaat uit naar Paolo Mancosu, die me aanraadde om mijn bevindingen meer te situeren binnen de bestaande filosofische literatuur van *mathematical explanation* en die me daarover een uitgebreide bibliografie bezorgde. De mogelijkheid om een hoofdstuk uit mijn eindverhandeling te presenteren op deze conferentie heeft me verplicht om het duidelijker uit te werken en dat is de kwaliteit ervan zeker ten goede gekomen.

Dank ook aan Stijn Janssens, Marijke Van Bogaert, Lorenz Demey en Jo Allemeersch voor onze boeiende en gezellige leesgroep. Tot slot wil ik mijn vriendin Liesbeth bedanken die me een jaar lang heel wat avonden heeft zien doorbrengen achter de computer, al zoekend, lezend en typend voor deze eindverhandeling, en ook op andere momenten vaak met mijn gedachten verkerend in een platonische wereld. Ik bedank haar ook voor het minutieus nalezen van mijn verhandeling en de hulp bij het opstellen van de bijlagen.

Inhoudsopgave

Lijst van figuren

1. Inleiding

Filosofie van de wiskunde is een breed onderwerp en bespreekt vragen uit ontologie[1], epistemologie[2], methodologie, … Ook computerbewijzen zijn in de filosofie van de wiskunde vanuit verschillende standpunten onderzocht: vanuit epistemologie (geven computerbewijzen ons kennis en indien ja, is dat a priori of a posteriori kennis?)[3], vanuit ontologie, vanuit de wetenschapsfilosofie (hebben computers de wiskunde veranderd?)[4]. We raken deze aspecten wel aan in deze eindverhandeling, sommige bespreken we zelfs uitgebreid, maar het gezichtspunt van waaruit we in deze eindverhandeling computerbewijzen bekijken is vooral dat van de *wiskundige praktijk*. Een duidelijke definitie van de wiskundige praktijk is die van Ursula Martin:[5]

> producing conjectural mathematical knowledge by means of speculation, heuristic arguments, examples and experiments, which may then be confirmed as theorems by producing proofs in accordance with a community standard of rigour, which may be read by the community in a variety of ways.

Computerbewijzen passen in het tweede deel van de definitie van Martin. We kunnen ons dus afvragen of de bewijzen van computers overeenkomen met de ‘community standard of rigour’ van wiskundigen, met andere woorden of ze betrouwbaar zijn. We zien inderdaad dat heel wat discussie over computerbewijzen zich toespitst op de betrouwbaarheid ervan. Een tegenreactie vinden we in een gemeenschap van onderzoekers die wiskunde door computers volledig wil laten formaliseren, zodat er zo weinig mogelijk discussie is over de betrouwbaarheid. Maar dit geeft dan weer problemen met de inzichtelijkheid van de bewijzen. Mensen schrijven hun bewijzen namelijk ook niet helemaal geformaliseerd tot in alle details op. Dit bespreken we in het tweede deel van onze eindverhandeling. Er zijn ook sociologische aspecten van computerbewijzen, maar dat laten we hier grotendeels terzijde.[6]

[1] [Benacerraf1965] is belangrijk, in mindere mate ook [Resnik1981]

[2] [Benacerraf1973] is belangrijk, in mindere mate ook [Resnik1982], [Pagin1994] en [Katz1995]

[3] [Burge1993] en [Burge1998] zijn invloedrijk.

[4] [Tymoczko1979]

[5] [Martin1999]

1.1. Wat is een bewijs?

Het idee ‘wiskundig bewijs’ is een concept dat zich doorheen de geschiedenis ontwikkeld heeft tot wat het nu is. De Grieken kwamen in de oudheid voor het eerst op het idee om kennis over wiskunde samen te vatten in een aantal principes. Zo werkten zij de *axiomatische methode* uit: je baseert je op een beperkt aantal axioma's en bouwt daarop een wiskundig systeem. Euclides vatte zo de wiskunde van zijn tijd samen in zijn *Elementen*. Naast een samenvatting heeft een bewijs ook als doel om een verantwoording of rechtvaardiging te zijn voor een stelling.

Descartes werkte het idee van bewijs verder uit. In zijn *Règles pour la direction de l'esprit* legt hij uit wat een wiskundige moet doen om een bewijs te *kennen*. Een wiskundige moet volgens hem alle details van het bewijs stap voor stap nagaan, dat wil zeggen verifiëren of elke uitspraak in het bewijs volgt uit voorgaande uitspraken door gebruik van een waarheidsbehoudende inferentieregel.[7] Hij legt dus aan een bewijs de beperking op dat elke stap moet te rechtvaardigen zijn, maar in de praktijk deden wiskundigen dit niet en zelfs Descartes liet in zijn bewijzen wel eens gaten.

In de negentiende eeuw begonnen verschillende wiskundigen met een project om de *wiskundige intuïtie* te elimineren uit de wiskunde. In wezen wilden zij het Cartesiaanse verhaal van een bewijs in de praktijk brengen: elke stap moest door een logische inferentieregel verantwoord kunnen worden en niet door een beroep op intuïtie. Ze wilden de intuïtie in de wiskundige basisprincipes leggen, zodat een bewijs enkel nog van logica gebruik maakt. Frege bijvoorbeeld definieerde als onderdeel van zijn formeel systeem een bewijs als een eindige lijst van uitspraken zodat elke uitspraak ofwel een axioma is ofwel kan afgeleid worden uit voorgaande uitspraken door middel van een geldige inferentieregel. Eén van de grootste pleitbezorgers van dit idee van bewijs was de wiskundige David Hil-

[6] Bijvoorbeeld in [Martin1999]: ‘Even if doubts remain, the “social effect” of a widely accepted computer proof (particularly one that is difficult or costly to replicate) inside a field is, generally, that the attention of the community shifts to other problems.’ De wiskundige William Thurston geeft een voorbeeld hiervan in [Thurston1994]. Hij beschrijft dat hij als jonge wiskundige verschillende resultaten bewees in foliatietheorie, maar zijn bewijzen waren heel technisch en hij legde niet uit hoe anderen zijn nieuwe technieken zouden kunnen gebruiken. Als gevolg hiervan verlieten andere wiskundigen het domein, omdat ze schrik hadden dat Thurston alle belangrijke resultaten zou bewezen hebben tegen de tijd dat ze zijn technieken onder de knie zouden hebben. Later deed Thurston belangrijk werk in het domein van Haken-variëteiten, maar deze keer hechtte hij veel belang aan het uitleggen van zijn technieken en het bewijzen van resultaten die andere wiskundigen als opstapje konden gebruiken. Het resultaat was dat een hele gemeenschap van wiskundigen resultaten behaalde in het domein. Alhoewel dit sociale effect zeker niet te onderschatten is, denk ik dat het zeker niet altijd het gevolg zal hebben dat wiskundigen het domein verlaten. Een computerbewijs zal door zijn beperkte inzichtelijkheid altijd een onbevredigd gevoel nalaten bij wiskundigen en hen net aansporen om een ‘beter’ bewijs te vinden dat hen meer inzicht geeft. Wiskundigen hebben overigens verschillende redenen om een stelling die al bewezen is op een andere manier te bewijzen, zoals John Dawson in [Dawson2006] uitlegt. Atle Selberg ontving in 1950 zelfs de prestigieuze Fieldsmedaille gedeeltelijk voor zijn elementair bewijs (samen met Erdös) van de al bewezen priemgetalstelling.

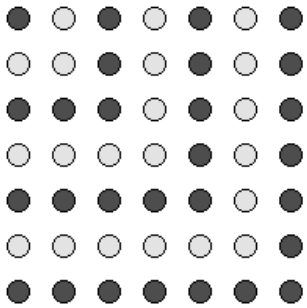
[7] [Fallis2003] bespreekt dit ‘Cartesiaanse verhaal’ van wat een bewijs is.

bert. Hij kon dit idee van bewijs doordrukken tot onze standaarddefinitie van een wiskundig bewijs, door Gian-Carlo Rota in 'The phenomenology of mathematical proof' geparafraseerd als:[8]

> A proof of a mathematical theorem is a sequence of steps which leads to the desired conclusion. The rules to be followed by such a sequence of steps were made explicit when logic was formalized early in this century, and they have not changed since.

Een bewijs is in deze Fregeaanse/Hilbertiaanse definitie dus een eindige opeenvolging van formules die elk ofwel een axioma zijn ofwel uit voorgaande formules volgen door middel van een inferentieregel. Dit formele idee van een bewijs is grotendeels een twintigste-eeuwse uitvinding: voor de uitvinding van formele logica en de bijbehorende Hilbertiaanse bewijzen was een bewijs voor de meeste wiskundigen gewoon een overtuigend argument.[9] Dit argument kan ook bestaan uit diagrammen[10], zoals het volgende overtuigende 'bewijs' dat de Grieken in de oudheid gebruikten voor de stelling dat de som van de eerste n oneven getallen gelijk is aan n^2:

Figuur 1.1. Een bewijs zonder woorden



$$1 + 3 + 5 + \cdots + (2n - 1) = n^2$$

In de praktijk werken wiskundigen niet met Hilbertiaanse (of Cartesiaanse) bewijzen, maar met informele bewijzen. Wiskundigen zijn zich daar zelf ook goed bewust van.[11] Bewijzen bevatten gaten en wiskundigen accepteren dat. Vaak staan er in een bewijs zinnen als 'De uitwerking hiervan laten we over aan de lezer', 'Dit resultaat is eenvoudig in te zien'[12], enzovoort.[13] Er zijn ook redenen waarom wiskundigen dit doen: als alle stap-

[8][Rota1997b] p. 218

[9][Bundy2005] p. 2377. Een goed historisch overzicht van de ontwikkeling van ons idee van formeel bewijs is te vinden in [Galda1981] en [Kleiner1991]

[10]Een goede bespreking van het gebruik van diagrammen in wiskundige bewijzen is te vinden in [Casselman2000].

[11]De wiskundige G.H. Hardy beschrijft dit in [Hardy1929] p. 16-17 als een tegenstelling tussen de formele, wiskundige, officiële bewijzen en de informele, onofficiële en *betekenisvolle* bewijzen.

[12]De wiskundige Pierre-Simon Laplace was berucht om zijn rechtvaardiging 'Il est aisé à voir' in bewijzen.

pen tot in de kleinste details worden uitgewerkt, heeft de wiskundige geen overzicht meer en ziet hij de *ideeën* in het bewijs niet meer.[14] Yehuda Rav noemt het ideale type van bewijzen (de Hilbertiaanse bewijzen) 'afleidingen' (*derivations*) en de bewijzen met gaten krijgen dan de naam 'bewijzen'. Hij suggereert om de relatie tussen bewijzen en afleidingen analoog te beschouwen met de relatie tussen de informele notie van een effectief berekenbare functie en de formele notie van een partieel-recursieve functie in de *these van Church*.[15]*Hilberts these*[16] zegt dan dat we elk informeel bewijs in principe kunnen omzetten naar een volledig formeel bewijs. Elk bewijs (in Ravs betekenis) kunnen we dus omzetten naar een afleiding door de gaten op te vullen. Deze analogie gaat echter niet helemaal op, zoals Rav ook zegt: terwijl de these van Church in twee richtingen werkt, is er geen manier om van een geformaliseerde versie van een bewijs het originele informele bewijs te reconstrueren. Jody Azzouni heeft Ravs aanpak meer uitgewerkt tot de zogenaamde *derivation-indicator view of ordinary mathematical proof*: een bewijs van een stelling wijst op een afleiding van de stelling.[17]

Bij de filosofische studie van computerbewijzen wordt het informele type wiskundige bewijs, wat wiskundigen in de praktijk gebruiken, ook belangrijk. Een vaak gehoorde kritiek is immers dat men het bewijs van een computer niet stap voor stap kan nakijken omdat het veel te lang is. In de praktijk doen wiskundigen dit echter ook niet bij menselijke bewijzen. Naast de formele logica gebruiken wiskundigen in bewijzen dus ook informele logica.[18]

[13][Fallis2003] geeft een goede bespreking van deze praktijk. Fallis bespreekt de verschillende soorten gaten die wiskundigen laten in bewijzen.

[14]Peter Renz verwijst naar een voorbeeld in [Renz1981] p. 85: een student van Hugo Steinhaus creëerde een volledig formeel bewijs van de stelling van Pythagoras uitgaande van Hilberts axioma's van de Euclidische meetkunde, met als resultaat een werk van 80 pagina's lang, niet het soort bewijs dat je aan schoolkinderen presenteert om de stelling *uit te leggen*.

[15][Rav1999] p. 11: 'The relation between proofs and derivations is in a limited sense analogous to the relation between the non-technical term of effectively computable function and the technical term of partially recursive function. *Church's Thesis* serves as a bridge between the intuitive and the technical notion of computability.' Hoofdstuk 5 van [Horsten2004] bespreekt effectieve berekenbaarheid en de these van Church.

[16]Hilbert heeft zelf deze these nooit uitgesproken, ze is achteraf aan hem toegeschreven. De eerste die de these zo noemde, lijkt Martin Davis geweest te zijn, zoals hij zelf schrijft op de *Foundations of Mathematics* mailing list: http://cs.nyu.edu/pipermail/fom/1999-February/002622.html

[17][Azzouni2004] p. 95: 'Since, however, the day-to-day practice of mathematicians isn't to actually *execute* such derivations, but only to *indicate*, to themselves or to others in their profession, such derivations, it's clear why *proof* and not *derivation* must occupy centerstage in mathematical practice; and this is despite the fact that, in a very clear sense, its's *derivation* which provides the skeleton for (the flesh of) *proof*.'

[18][Aberdein2006] past inzichten uit de argumentatietheorie toe op wiskundige bewijzen om de informele logica ervan te bestuderen. Hij past onder andere het argumentenpatroon van Stephen Toulmin en de dialectiek van Douglas Walton op wiskundige bewijzen, onder andere het computerbewijs van het vierkleurenprobleem.

1.2. De status van bepaalde bewijsmethodes

De geschiedenis van de wiskunde kent verschillende stromingen die bepaalde bewijsmethodes niet aanvaardden of als minder 'hoogstaand' beschouwden. Zo is er in het begin van de twintigste eeuw het intuïtionisme van Brouwer dat niet-constructieve bewijzen niet aanvaardde omdat ze onze menselijke beperkingen te boven gaan. De *reductio ad absurdum* bewijstechniek (ook wel 'bewijs door contradictie' genoemd: beginnen van een veronderstelling en daaruit een contradicitie afleiden, waaruit je dan kan afleiden dat de negatie van de veronderstelling geldt) was voor het intuïtionisme dan ook uit den boze, omdat ze niet constructief is.[19] Niet-constructieve existentiebewijzen geven bijvoorbeeld aan dat een bepaald wiskundig object bestaat, maar niet hoe je het kan berekenen.

Fundamentele bezwaren tegen bepaalde bewijsmethodes zijn overigens van alle tijden. Het intuïtionisme kende bijvoorbeeld een voorloper in het werk van Cavalieri, Guldin, Wallis en Arnauld die in de 17de eeuw de wiskunde wilden vrijwaren van bewijzen door contradictie. Zulke bewijzen werden in vraag gesteld door verschillende filosofen en wiskundigen.[20] Zo gebruikte Cavalieri in de meer dan vijfhonderd pagina's bewijzen van de eerste zes boeken van zijn *Geometria* bewust slechts één keer een bewijs door contradicitie en hij werkte nog jaren aan een direct bewijs van de stelling, dat hij in zijn *Exercitationes* presenteerde.[21]

Volgens Paolo Mancosu was de tegenstelling tussen *bewijsmethodes* en *ontdekkingsmethodes* voor de zeventiende-eeuwse wiskundigen een belangrijke factor om na te denken over de status van bewijzen door contradictie. Een bewijs door contradictie bewijst een stelling wel, maar geeft niet aan hoe de wiskundige het resultaat gevonden heeft en geeft dus minder inzicht in de stelling.[22] Zowel Wallis als Arnauld geven aan bewijzen door contradictie een lagere epistemologische status dan aan directe bewijzen. Ze wijten bovendien beiden de inferioriteit van bewijzen door contradictie aan het feit dat ze weinig geven, maar ze twijfelen niet aan de correctheid van de bewijsmethode.[23]

Arnauld realiseerde zich dat niet elke stelling kon bewezen worden door een direct bewijs en liet bewijzen door contradictie toe in gevallen wanneer hij de stelling niet direct kon bewijzen. We zien hier een parallel met de houding van veel hedendaagse wiskundigen tegenover computerbewijzen: ze beschouwen computerbewijzen minderwaardig aan menselijke bewijzen, wijten dit aan het gebrek aan inzicht dat computerbewijzen geven, maar ze twijfelen tegenwoordig niet meer aan de correctheid van (goed uitgevoerde)

[19][Horsten2004] geeft een overzicht van het intuïtionisme.

[20][Mancosu1991] geeft een grondig overzicht van deze stroming en vergelijkt ze met het twintigste-eeuwse intuïtionisme.

[21][Mancosu1991] p. 21

[22][Mancosu1991] p. 24: 'The attempt to overcome the opposition between *methods of proof* and *methods of discovery* was probably one of the main factors that led many mathematicians to think seriously about the status of *reductio ad absurdum* in mathematics.'

[23][Mancosu1991] p. 32

computerbewijzen. Bovendien accepteren ze computerbewijzen in gevallen dat ze geen menselijk bewijs kunnen vinden. We kunnen ons dan ook afvragen of het in vraag stellen van computerbewijzen te vergelijken is met een stroming zoals het intuïtionisme of de zeventiende-eeuwse stroming tegen bewijzen door contradictie. Zo eenvoudig ligt het echter niet, omdat er niet één bewijstechniek is die door alle computerbewijzen gedeeld wordt. Vandaar dat we in deze eindverhandeling een groot aantal computerbewijzen bestuderen en daarna een classificatie maken van de gebruikte bewijstechnieken in computerbewijzen. We kunnen de reacties op computerbewijzen dus niet helemaal vergelijken met de reacties op bewijzen door contradictie, maar er zijn zeker wel gelijkenissen. De belangrijkste factor in beide gevallen is namelijk dat de bewijzen geen inzicht geven. In het tweede deel van deze eindverhandeling bespreken we welke gebruikte bewijstechnieken in computerbewijzen voor dit gebrek aan inzicht zorgen.

1.3. Wat is een computerbewijs?

Wiskundigen hebben computers op verschillende manieren toegepast op wiskundige bewijzen. De term ‘computerbewijs’ kan dan ook voor verschillende zaken staan. We onderscheiden drie mogelijke betekenissen en de naam die we er in deze eindverhandeling aan geven:

1. **Computergeassisteerd bewijs:** Een bewijs waarvan een deel geverifieerd is door middel van een computerberekening die door een wiskundige geprogrammeerd is.
2. **Automatisch bewijsprogramma:** Een computerprogramma dat een bewijs vindt door automatisch te redeneren welke stappen het moet uitvoeren om de stelling te bewijzen uitgaande van de gegevens.
3. **Formele verificatie:** Een bewijs waarvan elke stap formeel geverifieerd is door een computerprogramma.

In deze eindverhandeling beperken we ons in het algemeen tot de eerste twee betekenissen. Wanneer we het over een ‘computerbewijs’ hebben, hebben we het dus in het algemeen over de twee eerste betekenissen samen. Wanneer we het specifiek hebben over programma's die zelf het bewijs opstellen, zullen we het over automatische bewijsprogramma's hebben, in de gevallen dat de computer slechts een deel van het bewijs verifieert door middel van een berekening zullen we spreken over een computergeassisteerd bewijs. Op formele verificatie zullen we ook ingaan, maar als we het over ‘computerbewijs’ hebben, bedoelen we niet formele verificatie.

1.4. De mogelijkheid van computerbewijzen na Gödel en Turing

Iemand die de geschiedenis van de wiskunde van de 20ste eeuw enigszins kent, vraagt

zich natuurlijk af waarom wiskundigen zich met computerbewijzen inlaten. Alan Turing heeft in zijn klassieke artikel ‘On computable numbers, with an application to the Entscheidungsproblem’ (1936)[24] immers aangetoond dat we wiskunde nooit volledig kunnen mechaniseren. We zullen nooit alle wiskundige vragen aan een computer kunnen stellen en in elk geval een ja of nee als antwoord krijgen. Op het eerste gezicht is dit een flinke opdoffer voor de doelstellingen van onderzoekers in *automated theorem proving*. Deze onderzoekers proberen computerprogramma's te ontwikkelen die, gegeven enkele axioma's en eigenschappen, een gegeven stelling proberen te bewijzen. Andere negatieve resultaten van Church en Gödel maken het op het eerste gezicht alleen maar erger.[25] Turing, Church en Gödel toonden samen aan dat we wiskunde en logica niet volledig tot berekeningen kunnen reduceren. Hilberts formalistische programma werd hierdoor alleszins de grond in geboord. Waarom bleef de mogelijkheid van computerbewijzen nog open?

Er zijn verschillende ‘loopholes’[26] waarmee we deze negatieve resultaten gedeeltelijk kunnen ontsnappen. Michael Beeson geeft drie mogelijke ontsnappingsroutes:[27]

- Misschien bestaat er voor een specifiek axiomatisch systeem A wel een beslissingsprocedure waarmee we kunnen berekenen of een bepaalde uitspraak P uit het axiomatisch systeem A volgt.
- Misschien bestaan er algoritmes f die gegeven een axiomatisch systeem A en een uitspraak P *soms* kunnen bepalen of P uit A volgt.
- Zelfs als zulke f enkel voor een specifiek axiomatisch systeem A zou werken, zou dit al interessant kunnen zijn.

In de eerste ontsnappingsroute zijn er heel wat resultaten. Zo bestaat er een beslissingsprocedure voor de Presburger-rekenkunde, dit is PA zonder de vermenigvuldiging. Martin Davis implementeerde deze beslissingsprocedure in een computerprogramma dat in 1954 bewees dat $1 + 1 = 2$. Dit is waarschijnlijk de eerste stelling die door een computer is bewezen. [28] Er bestaat ook een beslissingsprocedure voor trigonometrische identiteiten met lineaire functies in de argumenten, zoals $\cos(2x) = \cos^2 x - \sin^2 x$. Andere axiomatische systemen met beslissingsprocedures zijn de theorie van reële gesloten velden, de Euclidische meetkunde (door Descartes' reductie tot algebra: analytische meetkunde) en hyperbolische meetkunde.[29] In principe zijn alle problemen in deze systemen mechanisch op te lossen door een computer, maar uiteraard kunnen de beslissingsmethodes in de praktijk inefficiënt zijn, bijvoorbeeld als we met 100 variabelen te maken

[24][Turing1936]

[25]Gödels eerste onvolledigheidsstelling: [Gödel1931]

[26]Ik leen de term in deze context van Michael Beeson, die in [Beeson2003] een uitstekend overzicht geeft van computerbewijzen en de historische context.

[27][Beeson2003]

[28][Beeson2003]

[29]Beeson geeft in [Beeson2003] een goed overzicht van de systemen waarvoor we een beslissingsprocedure kennen.

hebben, of het nadeel hebben dat ze geen inzicht geven.

Overigens is Gödels resultaat over de onvolledigheid van consistente formele systemen die de Peano rekenkunde omvatten niet van toepassing voor de meeste *reële* problemen waarmee wiskundigen te maken krijgen. Over de oplosbaarheid van de meeste reële problemen zijn wiskundigen het eens. Aangezien computerbewijzen ook maar bewijzen zijn van reële problemen waarin wiskundigen geïnteresseerd zijn, komen de onvolledigheidsresultaten van Gödel en Turing in de praktijk niet tot uiting bij computerbewijzen.

1.5. Berekeningen versus redeneringen.

Computerbewijzen zijn eigenlijk *bewijzen* die tot *berekeningen* gereduceerd worden: dit is de mechanisatie van wiskunde. Het grootste deel van de computerbewijzen die we in deze eindverhandeling bespreken, behoren tot de klasse van computergeassisteerde bewijzen en dit zijn voorbeelden van berekeningen. Een vaak gehoorde kritiek die filosofen en wiskundigen op dit soort bewijzen geven, is dan ook dat het ‘louter berekeningen’ zijn. De computergedeeltes van bewijzen als die van de vierkleurenstelling of van het Keplervermoeden zijn eigenlijk uitgebreide berekeningen en dus eerder verificaties dan redeneringen. Een andere categorie van bewijzen begaat meer de weg van redeneringen, dit zijn de bewijzen die automatische bewijsprogramma's zoals OTTER en EQP vinden: deze programma's voeren zelf logische afleidingen uit en hebben zo bijvoorbeeld het Robbinsvermoeden bewezen.

Deze twee vormen van wiskunde, berekeningen en redeneringen, vinden we ook in menselijke wiskunde. Vaak bestaat een bewijs zowel uit berekeningen als uit redeneringen. Je kan berekeningen wel omzetten in redeneringen en andersom, maar dat zorgt dan voor onnatuurlijke en moeilijk te volgen bewijzen. Er is een duidelijk verschil tussen de twee methodes:[30]

> [But] there is an intuitive distinction: a calculation proceeds in a straightforward manner, one step after another, applying obvious rules at each step, until the answer is obtained. While performing a calculation, one needs to be careful, but one does not need to be a genius, once one has figured out what calculation to make. It is ‘merely a calculation.’ When finding a proof, one needs insight, experience, intelligence –even genius– to succeed, because the search space is too large for a systematic search to succeed.

Computers zijn in de afgelopen tientallen jaren heel sterk vooruitgegaan in hun vermogens om ingewikkelde wiskundige uitdrukkingen te berekenen, niet alleen numeriek maar ook symbolisch. Wiskundige programma's als MATLAB, MAPLE, MATHEMATICA, MAXIMA, GAP, MAGMA en PARI-GP laten toe om heel wat berekeningen uit te voeren en zo ook bepaalde stellingen te bewijzen door ‘louter een berekening’. Voor heel wat klas-

[30] [Beeson2003]

sen van problemen bestaan er beslissingsprocedures, algoritmes die gegarandeerd het gegeven probleem kunnen oplossen en die in computerprogramma's geïmplementeerd zijn.[31]

Op vlak van redeneren hebben computerprogramma's heel wat minder vooruitgang gemaakt. Zo komt het dat de bekendste computerbewijzen, bijvoorbeeld dat van de vierkleurenstelling, in zekere zin gewoon complexe berekeningen zijn. Het eerste computerbewijs van een niet-triviale stelling dat is bekomen door gebruik te maken van redeneren, was William McCunes bewijs van het Robbinsprobleem door het automatische bewijsprogramma EQP.

1.6. Computerbewijzen als een nieuwe ontwikkeling in wiskunde

We kunnen computerbewijzen als een relatief nieuwe ontwikkeling in de wiskunde beschouwen. De eerste niet-triviale computerbewijzen dateren uit de jaren 60 van de twintigste eeuw. Sommigen vinden het een belangrijke ontwikkeling die nieuwe elementen in de wiskunde brengt. Zo zouden computerbewijzen volgens de filosoof Thomas Tymoczko voor het eerst empirische elementen in de wiskunde brengen.[32] In deel 1 van deze eindverhandeling bekijken we uitgebreid verschillende computerbewijzen en onderzoeken we wat er zo nieuw aan is en wat het verandert aan de wiskunde.

Meer nog dan de vraag wat computerbewijzen eventueel veranderen aan de wiskunde, is het belangrijk om te onderzoeken wat de *waarde* is van computerbewijzen voor de wiskunde. Wiskundigen staan namelijk niet neutraal tegen nieuwe ontwikkelingen in de wiskunde. Sommige ontwikkelingen, zoals nieuwe bewijstechnieken, vinden ze waardevoller dan andere.[33] Volgens David Corfield gebruiken wiskundigen de volgende criteria bij het bepalen van het belang van een nieuwe ontwikkeling in de wiskunde:[34]

1. De ontwikkeling laat toe om nieuwe berekeningen uit te voeren in een bestaand probleemdomein, mogelijk met de oplossing van een oud vermoeden als gevolg.
2. De nieuwe ontwikkeling verbindt twee al bestaande domeinen, waardoor resultaten en technieken tussen de twee kunnen overgedragen worden.

[31][Beeson2003]: ‘Symbolic mathematics up to and including freshman calculus can thus be regarded as completely mechanized at this point.’

[32][Tymoczko1979]

[33]Denk bijvoorbeeld aan de *forcing*-techniek die de wiskundige Paul Cohen introduceerde om te bewijzen dat noch de continuümhypothese noch het keuzeaxioma kan bewezen worden uit de standaard ZF-axioma's. Hierna pasten wiskundigen zijn techniek met succes toe op heel wat andere problemen. Forcing is dus een heel waardevolle techniek.

[34][Corfield2003] p. 205

3. De ontwikkeling introduceert een nieuwe manier om resultaten in bestaande domeinen te organiseren, waardoor misschien een verduidelijking of verschuiving van de domeingrenzen mogelijk is.
4. De ontwikkeling opent perspectieven op nieuwe conceptueel gemotiveerde domeinen.
5. De ontwikkeling leidt redelijk direct tot succesvolle toepassingen buiten wiskunde.

Sommige ontwikkelingen scoren goed op één criterium en slecht op andere, maar als een ontwikkeling hoog scoort op verschillende of zelfs al deze criteria, dan gaat het duidelijk om iets belangrijks.[35] Uiteraard hebben niet al deze criteria hetzelfde gewicht en sommige (gemeenschappen van) wiskundigen geven aan bepaalde criteria andere gewichten dan aan andere criteria. De lijst met criteria lijkt echter wel overeen te komen met de praktijk van wiskundigen. Als we computerbewijzen als nieuwe ontwikkeling binnen de wiskunde gaan bekijken, moeten we ze evalueren volgens deze vijf criteria. We zullen dit in het besluit van deze eindverhandeling doen op basis van ons onderzoek van computerbewijzen en de filosofische aspecten ervan.

We kunnen al één probleem aanduiden dat computerbewijzen met deze criteria hebben. Veel computerbewijzen scoren wel goed op criterium 1 omdat ze een belangrijk vermoeden kunnen oplossen (bijvoorbeeld het vierkleurenprobleem, het Robbinsprobleem of het Keplervermoeden), maar op de andere criteria scoren ze vaak vrij zwak. Zo zorgen ze meestal niet voor conceptueel nieuwe inzichten en verbinden ze geen verschillende domeinen met elkaar.[36]

1.7. Methodologie

Hoe moet je als filosoof het fenomeen computerbewijzen bestuderen? Velen hebben dit al gedaan, dus je kan hun manier en impliciete vooronderstellingen overnemen. Maar het is ook de moeite om eens stil te staan bij wat een filosofische studie van computerbewijzen moet bereiken. In de vorige paragraaf heb ik al aangegeven dat ik computerbewijzen

[35]Corfield zegt hierover in [Corfield2003] p. 205: ‘My perception is that, very reasonably, if a development is seen either to be doing well or to have the potential to do well according to the majority of the criteria, then interest is guaranteed.’

[36]Corfield geeft de computerbewijzen van het vierkleurenprobleem als voorbeeld in [Corfield2003] p. 205-206: ‘Few now deny that the theorem is true, or that the various computer proofs warrant our belief in it, and yet a widespread feeling persists that unless there is some more conceptual success, for example, by linking the theorem to other branches in illuminating ways, then little has been achieved.’

wil bekijken als een nieuwe ontwikkeling in de wiskunde. Algemener wil ik computerbewijzen in hun bredere wiskundige context bekijken. Computerbewijzen zijn enerzijds wel een nieuwe ontwikkeling in de wiskunde, maar anderzijds moeten we ons daar niet op blindstaren. De filosofische 'problemen' met computerbewijzen hebben op zich niets te maken met computers, maar eerder met de *wiskundige* aspecten van de bewijzen. [37] Het is niet de computer die van belang is, maar de structuur van de bewijzen. Het enige dat al deze bewijzen gemeen hebben, is dat ze toevallig met behulp van een computer zijn gevonden. Dit is geen intrinsieke eigenschap van de bewijzen, die heel wat van elkaar verschillen, zoals we na een classificatie van de besproken computerbewijzen zullen zien.

Mijn bedoeling is om het midden te houden tussen een abstracte filosofische studie van computerbewijzen en een technische studie van specifieke systemen. Beide kanten hebben volgens Natarajan Shankar namelijk het risico om fouten te maken, respectievelijk overgeneralisatie en subjectiviteit:[38]

> The use of computer programs for theorem proving, program verification, and proof checking has been under criticism from various quarters. Most critics have based their comments on misgivings that are grounded in their philosophical attitudes, rather than on any actual experience using such programs. By the same token, proponents of automated theorem proving and program verification have mainly been either users or builders of such systems and can therefore hardly claim objectivity.

Ik heb er dan ook bewust voor gekozen om waar mogelijk de technische en wiskundige artikels te lezen van specifieke computerbewijzen.[39] Ik heb zelf ook een aantal bewijsprogramma's uitgeprobeerd om te kijken hoe ze werken en hoe een wiskundige er mee kan werken.[40] Ik vind het eveneens belangrijk om te weten wat wiskundigen zelf denken over computerbewijzen en heb daarom veel van hun opmerkingen in deze eindverhandeling gebruikt als 'empirisch bewijsmateriaal' om mijn stellingen te onderbouwen. Wanneer het gaat om de plaats van computerbewijzen in de wiskundige praktijk, zijn wiskundigen namelijk de maatstaf. Anderzijds is er ook een filosofische reflectie nodig en daar-

[37]Ook Jeremy Avigad heeft deze mening. Hij besluit zijn artikel 'Computers in mathematical inquiry' (2007) met: 'Issues regarding the use of computers in mathematics are best understood in a broader epistemological context. Although some of the topics explored here have become salient with recent computational developments, none of the core issues are specific to the use of the computer *per se*. Questions having to do with the pragmatic certainty of mathematical results, the role of computation in mathematics, and the nature of mathematical understanding have a much longer provenance, and are fundamental to making sense of mathematical inquiry. What we need now is *not* a philosophy of computers in mathematics; what we need is simply a better philosophy of mathematics.' ([Avigad2007b])

[38][Shankar1988] p. 475

[39]Enkel bij de vierkleurenstelling is het me niet gelukt om het originele bewijs ([Appel1977] en [Appel1977b]) te vinden, maar ik heb wel toegankelijke voorstellingen van de auteurs gevonden over de structuur van het bewijs ([Appel1977c] en [Appel1978]).

[40]Eén van de filosofen die dit ook heeft gedaan is David Corfield, in zijn boek *Towards a philosophy of real mathematics* ([Corfield2003])

om ben ik ook vertrokken van de standpunten van filosofen die bewijzen in het algemeen en computerbewijzen in het bijzonder bestudeerden. Daarbij hield ik wel rekening met de soms te algemene uitspraken die ze maakten omdat ze geen verschillende klassen van computerbewijzen onderscheidden.

Ik volg in deze eindverhandeling een ‘milde vorm van naturalisme’, zoals Leon Horsten die ook in zijn artikel ‘Platonistic formalism’ veronderstelt. Deze vorm van naturalisme zegt dat de beste redenen die we hebben om wiskundige principes te aanvaarden of te weigeren tot de wiskundige praktijk behoren.[41] De wiskundige gemeenschap moet dus op basis van *wiskundige* redenen beslissen of ze computerbewijzen aanvaarden en niet op basis van filosofische redenen.[42] De taak van de filosoof van de wiskunde is hier het identificeren en analyseren van deze wiskundige redenen. Ik volg in deze eindverhandeling de hiervoor vermelde vijf criteria van Corfield en kom in het besluit terug op de vraag hoe computerbewijzen hierop scoren en hoe dat de houding van wiskundigen tegenover computerbewijzen verklaart. Samengevat kan ik zeggen dat ik in deze eindverhandeling een naturalistische methodologie gebruik, analoog aan wat Penelope Maddy in een andere context deed, namelijk in een filosofische studie van de verzamelingentheorie.[43]

1.8. Het computerbewijs van de stelling van Pappus

In 1969 bewezen Elsie Cerutti en Philip Davis de stelling van Pappus[44] met een compu-

[41][Horsten2001] p. 175

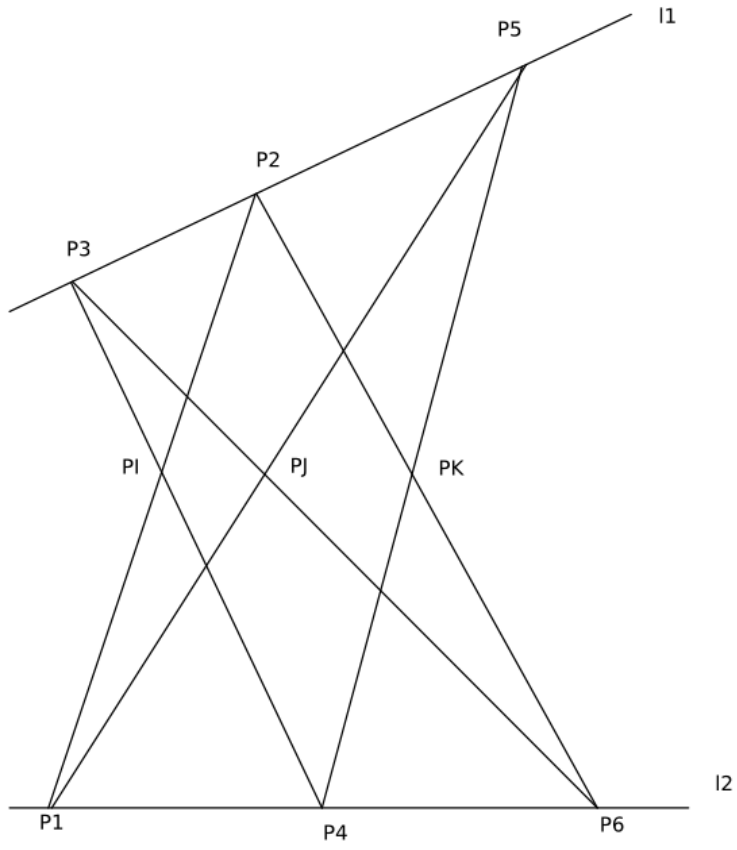
[42][Horsten2001] p. 175: ‘[But] every philosophical attempt to argue that a substantial body of mathematics that is generally accepted by the mathematical community is nevertheless fundamentally flawed has to be regarded with much suspicion. And the same holds for philosophical attempts to show that certain principles, not generally accepted by the contemporary mathematical community, are in fact basic mathematical truth and are henceforth to be taken as axioms.’ Uiteraard is het niet altijd duidelijk of bepaalde redenen nu wiskundige redenen of filosofische redenen zijn. Dat is ook één van de redenen waarom ik zoveel mogelijk meningen van wiskundigen zelf aan bod laat komen. Als wiskundigen bepaalde redenen aanhalen, is het plausibel dat het om redenen gaat die in de wiskundige praktijk geworteld zijn.

[43]Maddy is vooral geïnteresseerd in het begrijpen en uitleggen van de methodes en redenen die verzamelingentheoretici gebruiken om overeen te komen welke axioma's ze aanvaarden. Ze doet dit in [Maddy1997].

terprogramma.[45] Dit is één van de vroegste computerbewijzen, maar wordt vaak niet opgenomen in besprekingen van computerbewijzen omdat het bewijs eigenlijk bestaat uit één grote symbolische berekening van een determinant. Het bewijs is op zich ook niet zo belangrijk omdat er al verschillende elementaire meetkundige bewijzen van de stelling bekend waren. Cerutti en Davis zien hun bewijs echter als een interessant experiment om te onderzoeken welke rol de computer kan hebben in wiskundige bewijzen. Hun opmerkingen over de betrouwbaarheid van computerbewijzen worden door anderen bovendien opnieuw aangehaald in de filosofische discussie naar aanleiding van het computerbewijs van het vierkleurenprobleem. Het is dus een interessante inleiding op de bespreking van een aantal computerbewijzen in deel 1 van deze eindverhandeling.

De stelling van Pappus. Neem l_1 en l_2 als twee rechte lijnen in het vlak. Neem op l_1 drie willekeurige punten P_1, P_4 en P_6 en op l_2 drie willekeurige punten P_3, P_2 en P_5. Verbind nu de punten kriskras zoals in de figuur. Noem de aangegeven snijpunten P_I, P_J en P_K. Deze punten liggen op één lijn.

Figuur 1.2. De stelling van Pappus



[44]Cerutti en Davis noemen het de stelling van Pappus, maar er zijn meerdere stellingen naar Pappus genoemd. De hier besproken stelling staat ook bekend als de *zeshoekstelling van Pappus*, naar de zeshoek die in de duale versie van de stelling verschijnt.

[45][Cerutti1969]

Om het probleem op te lossen, schreven Cerutti en Davis een computerprogramma in de taal FORMAC. Ze voerden de berekening uit op een IBM 360/50 met 256 Kbyte geheugen. Ze vertaalden het meetkundig probleem in een algebraïsch probleem: de punten P_1 tot en met P_6 krijgen symbolische coördinaten, de punten P_I, P_J en P_K worden opgelost in termen van deze coördinaten en de verificatie dat deze punten op één lijn liggen is algebraïsch eenvoudig uit te drukken als een determinant die gelijk aan nul moet zijn.[46] De uitwerking hiervan is echter enorm veel werk om handmatig te doen. Op hun computer kregen ze na een kleine vijf minuten het resultaat:

```
DE = 0
```

Hiermee is de stelling bewezen. Cerutti en Davis merken op dat het niet slechts om dom rekenwerk gaat. Het programma vereenvoudigt expressies waar mogelijk en dit liet hen ook toe om nieuwe stellingen te vinden na inspectie van de uitvoer van de computer, een praktijk die de auteurs 'computer assisted theorem derivation' noemen.[47] Interessanter echter is hun discussie of dit nu wel een bewijs is.[48] De auteurs wijzen erop dat de objecties die je tegen de correctheid van het computerbewijs kan uiten, eveneens gelden voor traditionele menselijke bewijzen. Bij mensen zijn er volgens hen zelfs extra factoren die doen twijfelen aan de correctheid: mensen worden moe, hebben een slecht geheugen en zijn niet altijd eerlijk.[49] Dit zijn allemaal factoren die in latere discussies over computerbewijzen terugkomen.

Cerutti en Davis besluiten de discussie met een quasi-empirische opvatting over wiskunde. Computerbewijzen kunnen gecontroleerd worden door het programma verschillende keren uit te voeren, het programma te inspecteren en andere programma's te schrijven voor hetzelfde probleem. Als de resultaten hetzelfde zijn, geeft dit extra reden om in de correctheid ervan te geloven. Hetzelfde geldt volgens de auteurs voor menselijke bewijzen. Een wiskundig bewijs heeft volgens hen veel gemeen met een fysisch experiment: de geldigheid ervan is niet absoluut, maar hangt af van herhaalde experimenten.[50] Later, in de discussie rond het computerbewijs van het vierkleurenprobleem, nemen Michael Detlefsen en Mark Luker deze opvatting over.[51]

In 1972 werkt Davis zijn quasi-empirische opvatting op wiskunde verder uit en geeft hij meer argumenten waarom wiskunde sommige aspecten van de empirische wetenschappen heeft.[52] Zijn bespreking is duidelijk erg geïnspireerd door zijn experimenten met het

[46]Ulf Grenander noemt deze methode in [Cerutti1969] 'the method of artificial stupidity' (p. 896).
[47][Cerutti1969] p. 901-902
[48][Cerutti1969] p. 903: 'What if the programming was erroneous? What if the initial data were false? What if there was a machine malfunction? What if the programmer, in a moment of pique, simply programmed the computer to type out `DE = 0`, and let it go at that?'
[49][Cerutti1969] p. 902: 'Human processing is subject to such things as fatigue, limited knowledge or memory, and to the psychological desire to force a particular result to "come out".'
[50][Cerutti1969] p. 904
[51][Detlefsen1980]

FORMAC computersysteem, maar hij ziet de computer niet als een factor die empirische elementen in de wiskunde *introduceert*: de wiskunde is volgens Davis altijd al quasi-empirisch geweest en hij illustreert dit met verschillende voorbeelden.

Davis wijst allereerst op het dilemma tussen formaliseerbaarheid en overtuigingskracht van een bewijs.[53] Hij citeert ook Bourbaki die zegt dat het verifiëren van elke stap van een bewijs niet genoeg is om het te *begrijpen*.[54] De kern van de zaak is volgens Davis echter dat we in wiskunde symbolen manipuleren. We creëren verschillende symbolen, herkennen ze, reproduceren ze, schrijven ze aaneen, enzovoort. Om met symbolen te werken hebben we een fysische afdruk van de symbolen nodig: een '1' op een krijtbord of een elektronische puls in een computer. Het communiceren van deze fysische afdrukken van symbolen lukt nooit perfect, maar is volgens de communicatietheorie van Claude Shannon altijd geassocieerd met een bepaalde foutkans.[55] Om deze reden zegt Davis dat een som als 12345 + 54321 niet gelijk is aan het getal 66666, maar aan een waarschijnlijkheidsverdeling met gemiddelde 66666.[56] Davis besluit dan ook dat de correctheid van een wiskundig bewijs niet absoluut is, maar slechts probabilistisch. Als gevolg hiervan zijn lange bewijzen volgens hem quasi zeker incorrect.[57] Het maakt daarbij niet uit of het bewijs gevonden of geverifieerd is door een mens of een computer.[58]

[52][Davis1972]

[53][Davis1972] p. 255: 'For the professional mathematician, proof may be less a matter of convincing oneself psychologically of the truth of a statement than of merely assigning the tags "true" or "false" to the statement.'

[54][Davis1972] p. 255: 'Indeed, every mathematician knows that a proof has not been "understood" if one has done nothing more than verify step by step the correctness of the deductions of which it is composed and has not tried to gain a clear insight into the ideas which have led to the construction of this particular chain of deductions in preference to every other one.'

[55][Davis1972] p. 256

[56][Davis1972] p. 258: 'The sum 12345 + 54321 is not 66666. It is not a number. It is a probability distribution of possible answers in which 66666 is the odds-on favorite.'

[57][Davis1972] p. 260: 'Proofs cannot be too long, else their probabilities go down and they baffle the checking process. To put it in another way: all really deep theorems are false (or at best unproved or unprovable). All true theorems are trivial.'

[58][Davis1972] p. 262

1.9. Inhoud

In het eerste deel van deze eindverhandeling bespreken we een aantal computerbewijzen en de receptie ervan door wiskundigen en filosofen. Bij elk voorbeeld bespreken we in grote lijnen wat het wiskundige probleem is, hoe het bewijs er uitziet en wat het grote belang van de computer in het bewijs is. Daarnaast bespreken we ook reacties van de wiskundige en filosofische wereld op het bewijs in kwestie.

We zullen lang stilstaan bij de vierkleurenstelling omdat op het computerbewijs daarvan door Appel en Haken de meeste reactie gekomen is. Veel filosofische studies van computerbewijzen beperken zich tot het Appel-Haken bewijs van de vierkleurenstelling, maar om te vermijden dat we te veel generaliseren aan de hand van één voorbeeld, bekijken we ook andere, minder bekende computerbewijzen. In deze inleiding illustreren we wat een computerbewijs is met het bewijs van de stelling van Pappus door Davis. In de volgende hoofdstukken bespreken we de verschillende computerbewijzen van de vierkleurenstelling, de stelling dat er geen eindige projectieve vlakken zijn van orde 10, het Robbinsprobleem, Thomas Hales' oplossing van Keplers probleem, probabilistische priembewijzen en DNA-bewijzen. Daarna bespreken we kort enkele minder bekende computerbewijzen.

In het tweede deel van deze eindverhandeling stellen we op basis van de besproken bewijzen een classificatie van computerbewijzen op. Van elke categorie vermelden we wat de eigenschappen van deze bewijzen zijn en wat er gemeenschappelijk is. We bekijken de filosofische relevantie van deze eigenschappen en bespreken kort de receptie van de computerbewijzen in het licht van deze eigenschappen. Daarna gaan we in op de vraag of een wiskundige een computerbewijs kan begrijpen, of hij erdoor inzicht in de bewezen stelling kan krijgen. We focussen ons op twee factoren die bepalen of een wiskundige inzicht kan krijgen in een bewijs: de in het bewijs gebruikte wiskundige concepten en de structuur van het bewijs.

Terwijl we in het eerste deel van deze eindverhandeling hoofdzakelijk concrete computerbewijzen bespreken en de reacties van wiskundigen erop, zal het tweede deel meer *constructief* zijn: we bespreken waarom computerbewijzen nog altijd geen dagelijks onderdeel van de wiskundige praktijk zijn[59] en wat ontwerpers van bewijsprogramma's hieraan kunnen doen.

[59][Bundy2006] p. 481: ‘most mathematicians have shown little or no interest in automated proof.’ en p. 485: ‘Automated theorem provers have been largely ignored.’

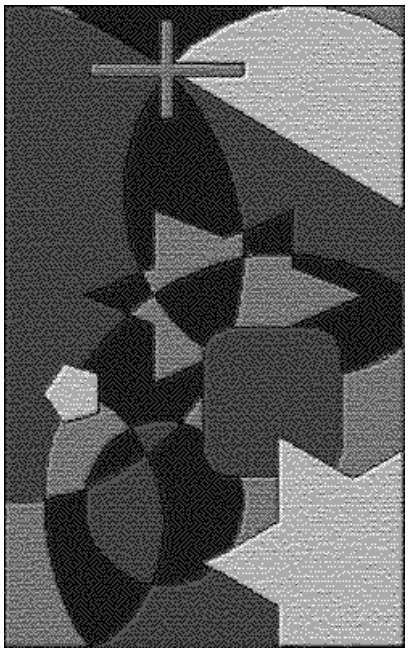
Deel I. Computerbewijzen en hun receptie

2. De vierkleurenstelling

2.1. Het probleem en de eerste pogingen tot bewijs

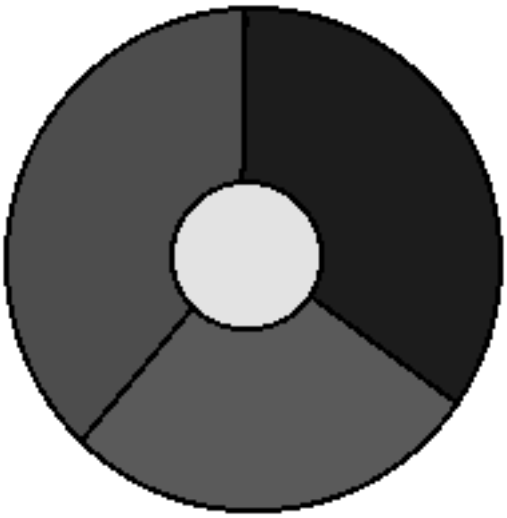
De vierkleurenstelling zegt dat het voor een willekeurige landkaart mogelijk is om de landen met hoogstens vier kleuren zó in te kleuren dat geen twee aangrenzende landen dezelfde kleur hebben. Twee landen zijn aangrenzend als ze een grens delen, dus landen die slechts in een punt aan elkaar liggen (zoals twee overliggende spieën van een gesneden taart) zijn niet aangrenzend. Met een land wordt bovendien een samenhangend deel van de kaart bedoeld.

Figuur 2.1. Voorbeeld vierkleurenstelling



Een voorbeeld van een eenvoudige kaart waarvoor we vier kleuren nodig hebben is de volgende:

Figuur 2.2. Een kaart met vier kleuren



De vierkleurenstelling is eenvoudig te formuleren, maar wiskundigen hebben er 125 jaar over gedaan om ze te bewijzen, en dat lukte dan nog enkel met de hulp van computers.[1] De juiste oorsprong van het probleem is niet helemaal duidelijk.[2] De eerste geschreven meldingen van het probleem stammen uit 1852[3]: in dat jaar schreef de student Francis Guthrie aan zijn jongere broer Frederick in een brief dat het hem leek dat de landen van elke kaart altijd met slechts vier kleuren kunnen ingevuld worden zodat geen twee aangrenzende landen dezelfde kleur hebben. Francis vroeg aan zijn broer of hij misschien wist of hij zijn *four-colour conjecture* wiskundig kon bewijzen. Frederick stelde de vraag aan de prominente wiskundige Augustus De Morgan, bij wie beide broers les volgden in het University College in Londen. De Morgan kon Guthries vermoeden niet bewijzen, maar hij kon wel bewijzen dat vijf landen niet op een kaart kunnen geplaatst worden op een manier zodat ze elk aan de andere vier grenzen. De Morgan kon hieruit afleiden dat je nooit meer dan vijf verschillende kleuren nodig hebt om een kaart te kleuren op de manier van Guthrie.

[1]Een goed historisch overzicht van de ontwikkelingen rond de vierkleurenstelling tot het bewijs van Appel en Haken is te vinden in [MacKenzie1999] en [Appel1978]. Ook [Mayer1982] geeft een goede historische uitleg, vooral van de vroege periode.

[2][Mayer1982] p. 44: ‘Sur l'origine du problème, des légendes se sont transmises durant plusieurs décennies; mais une critique sérieuse ne leur a apporté aucune confirmation. Le problème des quatre couleurs aurait été connu de Möbius, d'Euler, ou même remonterait aux cartographes de la Renaissance.’

[3][May1965]. Voor het vaak gehoorde verhaal dat kaartenmakers al lang wisten dat vier kleuren genoeg zijn, hebben we volgens May geen bewijzen: ‘There is no evidence that mapmakers were or are aware of the dufficiency of four colors. A sampling of atlases in the large collection of the Library of Congress indicates no tendency to minimize the number of colors used. Maps utilizing only four colors are rare, and those that do usualy require only three. Books on cartography and the history of mapmaking do not mention the four-color property, though they often discuss various other problems relating to the coloring of maps.’

De Morgan speelde het probleem door aan William Hamilton,[4] die er geen interesse in leek te hebben. De Morgan bleef aan verschillende collega's vragen of ze het vierkleurenprobleem konden oplossen. Peirce probeerde het in 1860 en nadien ebde de interesse in het probleem even weg. Nadat Arthur Cayley van het vermoeden had gehoord, stelde hij het in 1878 voor aan de London Mathematical Society en vroeg hij of het al bewezen was. In 1979 publiceerde de Royal Geographical Society een artikel van Cayley waarin hij de moeilijkheden beschrijft om het vierkleurenprobleem te bewijzen: 'On the colouring of maps'.[5] Hetzelfde jaar nog publiceerde Alfred Bray Kempe, aangemoedigd door Cayley[6], een bewijs van het vermoeden in de American Journal of Mathematics.[7] De wiskundige gemeenschap was onder de indruk van Kempes bewijs en hij werd hierdoor verkozen tot Fellow of the Royal Society. Later werd hij zelfs vice-president en schatbewaarder.[8] Pas elf jaar later, in 1890, ontdekte Percy John Heawood dat Kempes bewijs fout was.[9]

Vele wiskundigen hebben hun tijd besteed aan het vierkleurenprobleem en er volgden vele verkeerde bewijzen, onder andere van Peter Guthrie Tait en Frederick Temple. Het probleem lijkt dan ook misleidend eenvoudig. Het verhaal gaat dat Minkowski ooit beweerde dat het vierkleurenprobleem nog niet opgelost was omdat enkel derderangs wiskundigen er aandacht aan besteed hadden. Op een bepaald moment begon Minkowski dus zelf het vierkleurenprobleem te bestuderen, overtuigd dat hij het in geen tijd zou oplossen. Enkele weken later liet hij het vierkleurenprobleem stilletjes voor wat het was en wijdde hij zich weer aan zijn ander werk.[10]

[4][Mayer1982] p. 44: 'La première mention écrite du problème, son acte de naissance pour ainsi dire, se trouve dans une lettre d'Augustus de Morgan à Sir William Rowan Hamilton du 23 october 1852.'

[5]De moeilijkheid ligt er volgens Cayley hier in: "Supposing a system of *n* areas coloured according to the theorem with four colours only, if we add an (n+1)th area, it by no means follows that we can *without altering the original colouring* colour this with one of the four colours." ([Cayley1879]) Een eenvoudig bewijs door inductie is hier dus niet mogelijk.

[6][Crilly2005] p. 298-299

[7][Kempe1879]

[8][MacKenzie1999] p. 18

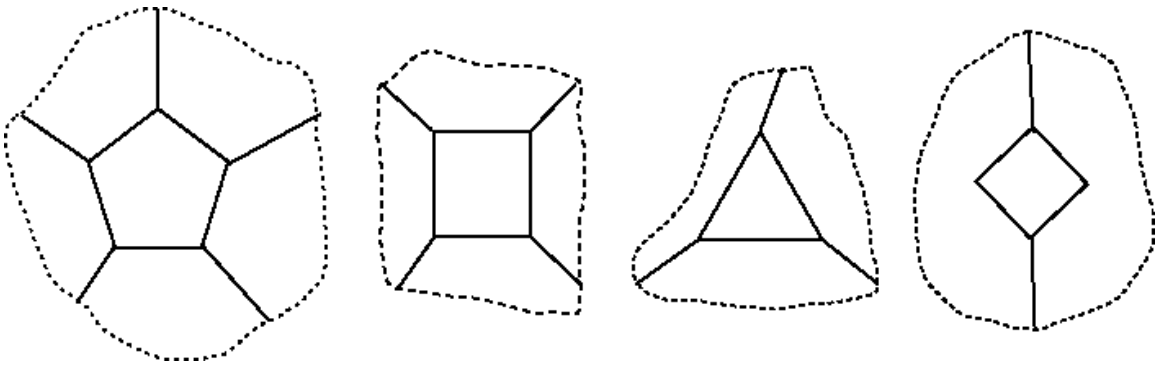
[9][Heawood1890]

[10][Mitchem1981] p. 114

2.2. De basisideeën van het bewijs

Hoewel Kempes bewijs fout was, introduceerde hij er wel twee belangrijke concepten mee waar alle huidige bewijzen van de stelling op gebaseerd zijn: *onvermijdelijke verzamelingen* (unavoidable sets) en *reduceerbare configuraties* (reducible configurations). Kempe had namelijk bewezen dat elke normale kaart tenminste één land bevat dat twee, drie, vier of vijf buren heeft.[11] Kempes configuraties zagen er zo uit:

Figuur 2.3. Kempes reduceerbare configuraties



De verzameling van configuraties die bestaan uit een land en twee tot vijf buren is dus *onvermijdelijk*: elke kaart moet minstens één configuratie uit deze verzameling bevatten. Het idee van *reduceerbaarheid* gaat als volgt: een configuratie is reduceerbaar als kan aangetoond worden dat ze niet kan voorkomen in een minimale kaart met vijf kleuren. Als je nu een onvermijdelijke verzameling van reduceerbare configuraties hebt, is dat genoeg om de vierkleurenstelling te bewijzen: je hebt dan namelijk bewezen dat elke kaart één van de configuraties uit die verzameling moet bevatten. Bovendien weet je dat als de kaart een reduceerbare configuratie bevat, de kaart geen vijf kleuren kan bevatten. Dus zal ze maximum vier kleuren bevatten.

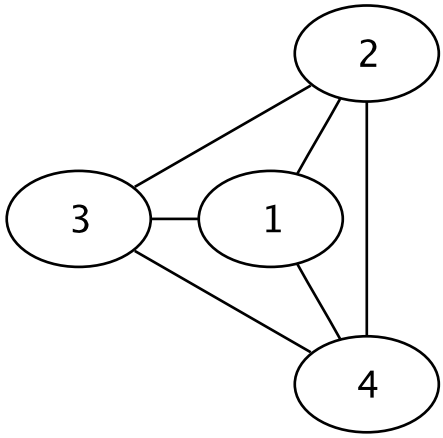
Kempes verkeerde bewijs was een poging om een onvermijdelijke verzameling van reduceerbare configuraties te vinden. Hij had wel een onvermijdelijke verzameling van configuraties gevonden, maar Heawood had aangetoond dat één van de vier configuraties (dit met de vijf buren) niet reduceerbaar was. Met de jaren werd het duidelijk dat een onvermijdelijke verzameling van reduceerbare configuraties heel wat complexer zou moeten zijn dan de vier configuraties van Kempe. Uiteindelijk bleek het zelfs zo complex te zijn dat er een computer voor nodig was.

De vierkleurenstelling is trouwens gemakkelijker formeel te onderzoeken als je ze herformuleert in *grafentheorie*. Grafentheorie bestudeert grafen. Een graaf is een verzameling van punten die door middel van lijnen met elkaar verbonden zijn. De vierkleurenstelling

[11] [Kempe1879] p. 198: “Every map drawn on a simply connected surface must have a district with less than six boundaries.”

voor grafen zegt dat de punten van elke vlakke graaf gekleurd kunnen worden met ten hoogste vier kleuren zodat geen enkele twee aangrenzende punten dezelfde kleur hebben. De stelling voor kaarten is equivalent met de stelling voor grafen als je elk land op de kaart vervangt door een punt van de graaf en twee punten van de graaf verbindt door middel van een zijde als de overeenkomende landen aan elkaar grenzen. De overeenkomende graaf voor de eenvoudige kaart in Figuur 2.2, “Een kaart met vier kleuren” is de volgende:

Figuur 2.4. Een graaf met vier kleuren



De rest van de zoektocht naar een bewijs bestond dus uit een zoektocht naar een onvermijdelijke verzameling van reduceerbare configuraties. Zo bewees George David Birkhoff in 1913 dat een aantal grote configuraties reduceerbaar zijn. Philip Franklin bewees met dezelfde technieken in 1922 dat een vijfchromatische kaart (een kaart die, hypothetisch gezien, vijf kleuren zou nodig hebben) meer dan 25 landen zou moeten bevatten.[12] In 1938 had Franklin dat aantal kunnen optrekken tot 32 en in 1940 bewees Winn dat alle kaarten met maximum 36 landen met vier kleuren in te kleuren zijn.[13] Men had dus nog geen algemeen bewijs, maar kon de stelling wel bewijzen voor steeds grotere grafen. Al deze pogingen om het vierkleurenprobleem te bewijzen waren ondertussen een grote stimulans voor de verdere ontwikkeling van grafentheorie.[14]

In 1936 begon Heinrich Heesch aan de vierkleurenstelling te werken, nadat hij een fout vond in een bewijs van de stelling door zijn vriend Ernst Witt.[15] In 1947 en 1948 presenteerde hij zijn ideeën in seminaries aan de universiteiten van Hamburg en Kiel. Hij uitte

[12][Franklin1922] p. 235: “Every map containing 25 or fewer regions can be colored in four colors.”

[13]Kenneth O. May had er in 1965 echter nog geen vertrouwen in dat een algemeen bewijs zou gevonden worden: “The consensus is that the conjecture is correct but unlikely to be proved in general. It seems destined to retain for some time the distinction of begin both the simplest and most fascinating unsolved problem of mathematics.” ([May1965] p. 25)

[14]Het eerste tekstboek over grafentheorie kwam in 1936 uit: *Theorie der endlichen und unendlichen Graphen* van de Hongaar Denes König.

[15][MacKenzie1999] p. 25

het vermoeden dat het mogelijk was een onvermijdelijke verzameling van reduceerbare configuraties te vinden. Hij schatte dat de configuraties klein genoeg zouden zijn om de reductie ervan te kunnen bewijzen en dat het er ongeveer tienduizend zouden zijn. Wolfgang Haken was toen als student te Kiel aanwezig in het seminarie van Heesch. Heesch was zo de eerste die inzag dat het probleem tot een *eindig* aantal gevallen kon gereduceerd worden. Al de vorige onderzoekers (Birkhoff, Franklin, Winn, …) lieten in het midden of het probleem afhing van een eindig aantal configuraties.[16]

Op dat moment was het nog heel moeilijk om zo'n grote verzameling van configuraties in de praktijk te zoeken, maar met de opkomst van de eerste computers kregen de wiskundigen toch wat meer vertrouwen. Heesch begon daarom de bekende methodes om te bewijzen dat bepaalde configuraties reduceerbaar zijn te formaliseren. Eén van deze formalisaties, *D-reductie*, leek hem heel geschikt om door een computer uitgevoerd te worden en zijn student Karl Dürre schreef een computerprogramma dat deze methode implementeerde.[17] Hoewel het programma tientallen uren nodig had voor complexere configuraties, was het belangrijk als voorbode van het gebruik van computers voor het vierkleurenprobleem.

Heesch bestudeerde aanvankelijk vooral hoe hij reduceerbare configuraties kon vinden. Hierna richtte hij zich op het zoeken van onvermijdelijke verzamelingen en in 1969 introduceerde hij hiervoor een nieuwe procedure: *ontladen* (discharging). Deze methode was analoog aan het verplaatsen van ladingen in een elektrisch netwerk. Heesch had nu alle benodigdheden voor een bewijs van de vierkleurenstelling, behalve één: computerkracht.

2.3. Het bewijs van Appel en Haken

In 1970 waren alle technieken beschikbaar die tot het uiteindelijke bewijs van de vierkleurenstelling zouden leiden, maar schattingen over de nodige berekeningen toonden aan dat het veel te lang zou duren. Toch begon Haken na verbeteringen van de ontladingsprocecures te hopen op een oplossing. Dürres programma was echter niet efficiënt voor grote configuraties. Appel en Haken twijfelden dus of ze wel een bewijs konden vinden voordat er krachtiger computers ontwikkeld zouden worden. In 1972 schreven ze een computerprogramma om een onvermijdelijke verzameling van *waarschijnlijk* reduceerbare configuraties te vinden. In de jaren erna schaafden ze het programma bij en kregen ze meer en meer inzicht in het probleem. Ze probeerden hun software uit op beperkte versies van het vierkleurenprobleem en schatten hoeveel moeilijker het algemene probleem zou zijn.

[16][Mayer1982] p. 47: 'Le mérite d'avoir conçu clairement la finitisation du problème revient à Heinrich Heesch.'
[17][MacKenzie1999] p. 27

In 1975 kregen Appel en Haken er goede hoop in. Ze hadden al een programma om een onvermijdelijke verzameling te vinden, nu moesten ze zich nog richten op het bewijzen van de reduceerbaarheid van de configuraties. Ze schreven een eenvoudig computerprogramma om de reduceerbaarheid te testen. In 1974 vervoegde student John Koch hen voor het programmeerwerk. In de volgende jaren verbeterden ze hun ontladingsprocedure en van januari tot juni 1976 werkten ze hun bewijs af. Dat ontstond in een dialoog met de computer, door van een onvermijdelijke verzameling configuraties één voor één te proberen te berekenen of de configuraties reduceerbaar waren. Lukte dit bij een bepaalde configuratie niet binnen een zekere tijd, dan pasten Appel en Haken hun ontladingsprocedure aan om die moeilijke configuratie te vermijden en probeerden het opnieuw.

Het uiteindelijke bewijs bestond uit twee delen: het eerste deel beschreef de globale bewijsstrategie en de 487 ontladingsregels, met de hand uitgewerkt, om de onvermijdelijke verzameling te construeren. Het tweede deel somde de onvermijdelijke verzameling van 1482 configuraties op en beschreef de computerprogramma's om te bewijzen dat elk van deze configuraties reduceerbaar is.[18] Voor dit tweede deel *moest* men wel een computer gebruiken, aangezien er enorm veel berekeningen voor nodig waren. Voor het bewijs van reduceerbaarheid van sommige configuraties moest het computerprogramma meer dan tweehonderdduizend verschillende mogelijke inkleuringen uitproberen.[19] Op microfiche stelden ze nog meer dan 400 pagina's beschikbaar met diagrammen en gedetailleerde verificaties van lemma's die ze gebruikten in hun artikel.

De aankondiging van Appel en Haken dat ze het vierkleurenprobleem opgelost hadden, kon op heel wat scepsis rekenen. Dat was ook normaal, gezien de lange geschiedenis van verkeerde bewijzen. Enkele jaren tevoren kondigde Yoshio Shimamoto bijvoorbeeld nog een computerbewijs van de vierkleurenstelling aan, dat niet juist bleek te zijn. Nadat experts het bewijs van Appel en Haken echter hadden gecontroleerd, werd het geaccepteerd en gepubliceerd in de *Illinois Journal of Mathematics*.[20] Dit bewijs was het eerste bewijs van een belangrijke wiskundige stelling waarbij de computer een essentieel instrument was.

2.4. Het bewijs van Allaire

Voor de volledigheid vermelden we nog dat Frank Allaire in 1977 een vereenvoudigd bewijs heeft gegeven, gebaseerd op het bewijs van Appel en Haken, waarvan hij één van de

[18] Eerst hadden ze 1936 configuraties, later reduceerden ze het tot 1482 en nog later konden ze het aantal reduceren tot 1405 ([Mayer1982] p. 48)

[19] [Mayer1982] p. 49

[20] [Appel1977] en [Appel1977b]

referees was. Zijn bewijs gebruikte dezelfde globale strategie van het zoeken naar een onvermijdelijke verzameling van reduceerbare configuraties. Hij gebruikte echter een verschillende ontladingsprocedure, een verschillende onvermijdelijke verzameling en andere reduceerbaarheidsbewijzen.[21] De details van zijn bewijs zijn volgens MacKenzie echter nooit gepubliceerd.[22]

2.5. Het bewijs van Robertson

In 1993 wilden Neil Robertson, Daniel Sanders, Paul Seymour en Robin Thomas nagaan of de vierkleurenstelling echt waar was.[23] Ze begonnen het bewijs van Appel en Haken dus na te gaan, maar gaven dit al vlug op.[24] Ze besloten dus om een eigen bewijs te construeren, met gebruik van dezelfde methode (een onvermijdelijke verzameling van reduceerbare configuraties) als Appel en Haken. Het basisidee van hun bewijs is dus hetzelfde, maar het bewijs verschilt in de uitwerking. Robertson en zijn collega's hebben een onvermijdelijke verzameling van 633 configuraties (tegenover 1482 voor Appel en Haken) en slechts 32 ontladingsregels (tegenover 487 voor Appel en Haken). Het theoretische deel van het bewijs is een traditioneel wiskundig bewijs, maar het vertrouwt op twee resultaten die door een computerprogramma geverifieerd zijn.

Het bewijs van Robertson is eenvoudiger dan dat van Appel en Haken. Volgens de auteurs is het ook gemakkelijker te controleren. Het bewijs van de onvermijdelijke verzameling is formeel neergeschreven, zodat het door een computer in een aantal minuten kan geverifieerd worden. Een extra voordeel van het bewijs is dat het kan omgezet worden tot een algoritme dat in kwadratische tijd een kaart kan inkleuren, terwijl de tijdscomplexiteit bij het bewijs van Appel en Haken bikwadratisch is, dus $O(n^4)$. Bovendien heeft het nieuwe bewijs relaties blootgelegd tussen de vierkleurenstelling en andere wiskundige uitspraken. De auteurs hopen dat een aanpassing van hun bewijs kan gebruikt worden om algemenere vermoedens dan de vierkleurenstelling te bewijzen.[25]

[21][Swart1980] p. 698: ‘Allaire's proof also involves a discharging/reducibility approach but only requires some 50 hours of computer time. It is, moreover, based on an entirely different discharging procedure and a completely independently developed reducibility testing program. At the very least Allaire's proof must rank as an independent corroboration of the truth of the four-color conjecture, and there can be little doubt that even if the Haken/Appel proof is flawed the theorem is nevertheless true.’

[22][MacKenzie1999] p. 38

[23][Robertson1996] is een algemeen overzicht van het bewijs en [Robertson1997] bevat het bewijs zelf.

[24][Robertson1997] p. 2: “To check that the members of their ‘unavoidable set’ were all reducible would require a considerable amount of programming, and *also* would require us to input by hand into the computer descriptions of some 1400 graphs; and this was not even the part of their proof that was most controversial.”

2.6. Het formeel bewijs van Gonthier

In 2004 formaliseerden Georges Gonthier en Benjamin Werner het bewijs van Robertson in het bewijssysteem COQ. Dit programma bewijst de correctheid van hun bewijs, waardoor wiskundigen enkel nog moeten vertrouwen op de correctheid van COQ. Hiermee lijkt de weinige achterdocht die er nog was over de correctheid van Robertsons bewijs verleden tijd.[26] De correctheid van Gonthiers bewijs hangt nu immers nog wel af van de correcte werking van de gebruikte computerhardware, maar niet meer van de specifieke software die gebruikt is voor het bewijs van de vierkleurenstelling (COQ heeft daarvan een correctheidsbewijs). Het COQ bewijssysteem genereert bovendien een *bewijsgetuige* (proof witness), een gedetailleerde lijst met de genomen formele logische stappen in het correctheidsbewijs, dat in principe onafhankelijk kan nagekeken worden. Specialisten in computerverificatie bleken in de wolken over Gonthiers bewijs. Freek Wiedijk noemde het bewijs een groots werk.[27] Onder de grafentheoretici was er echter nog altijd scepsis over het bewijs.[28]

Gonthiers bewijs is vooral een formalisering van het bewijs van Robertson en heeft dus dezelfde globale structuur.[29] Gonthier nam bijvoorbeeld de lijst van 633 reduceerbare configuraties die Robertson gevonden had over, evenals de 32 ontladingsregels. Een ander verschil was dat verschillende concepten moesten geherformuleerd worden. Terwijl in het niet-computerdeel van het bewijs van Robertson nog heel wat 'intuïtieve' concepten voorkwamen die vertrouwden op het menselijke geometrische voorstellingsvermogen, kon het COQ bewijssysteem niet overweg met zulke geometrische concepten. Gonthier formaliseerde dus al deze concepten tot combinatorische eigenschappen, waardoor het bewijs minder intuïtief is voor mensen, maar des te gemakkelijker te bewerken voor een computer. Een ander voordeel van deze gedetailleerde formalisatie was dat Gonthier enkele moeilijkheden en schoonheidsfoutjes in het bewijs van Robertson kon vermijden.[30] Gonthier heeft veel tijd besteed aan het zoeken van het juiste formalisme en als gevolg daarvan zijn grote delen van het bewijs volgens hem triviaal geworden.[31]

---

[25][Thomas1998] p. 857

[26][Gonthier2004] p. 2: "Our work can be seen as an ultimate step in this clarification effort, completely removing the two weakest links of the proof: the manual verification of combinatorial arguments, and the manual verification that custom computer programs correctly fill in parts of those arguments. To achieve this, we have written a *formal proof script* that covers both the mathematical and computational parts of the proof."

[27][Mackenzie2005]. Wiedijk geeft ook wat uitleg over het bewijs in [Wiedijk2006]

[28]Mackenzie citeert de Sloveense grafentheoreticus Bojan Mohar in [Mackenzie2005]: 'I have no serious doubts that computers have done their part flawlessly. But I cannot onfirm that Gonthier has made the correct translation of the [human] proof into computer form.'

[29]Dit geldt eigenlijk voor alle tot nu toe bekende bewijzen van de vierkleurenstelling (dat van Appel en Haken, dat van Allaire, dat van Robertson en dat van Gonthier). Al deze bewijzen zijn gebaseerd op het principe van Heesch (dat Kempes techniek expliciteert) van het zoeken naar een onvermijdelijke verzameling van reduceerbare configuraties met behulp van ontladingsregels.

[30]Zoals het gebruik van een "folklore theorem" waarvan het bewijs volgens Robertson nooit gepubliceerd is, maar wel eenvoudig is, hoewel lang. ([Gonthier2004] p. 17)

[31][Gonthier2004] p. 38

Het was in het begin trouwens helemaal niet de bedoeling van Gonthier om een alternatief bewijs van de vierkleurenstelling te leveren.[32] Het begon heel onschuldig toen hij een programmeerproject zocht voor studenten in een inleidende cursus computerwetenschappen. Hij overwoog om hen de opdracht te geven de reduceerbaarheidsberekeningen van Robertson te optimaliseren met modernere programmeertechnieken. Nadat hij dit zelf bestudeerd had, begon Gonthier te vermoeden dat de berekening efficiënt kon gedaan worden in een COQ bewijsprogramma. Hij wou dit vervolgens nagaan, aangezien de berekening hem een interessante *real life* test bleek voor het COQ bewijssysteem. Uit zijn onderzoek bleek dat het voor een bepaalde functie moeilijker was om ze te programmeren dan om de correctheid ervan te bewijzen. Dit bracht Gonthier ertoe om een ambitieuzer project uit te proberen: een volledig getest formeel bewijs van het reduceerbaarheidsgedeelte van de vierkleurenstelling. Toen dit lukte, bleek na verdere studie van Robertsons bewijs dat hij het gedeelte van de onvermijdelijke verzameling sterk kon vereenvoudigen. Toen pas vermoedde Gonthier dat hij een *volledig* formeel bewijs van de vierkleurenstelling zou kunnen leveren. Het uiteindelijke bewijs bestond uit 50000 regels code om het volgende te bewijzen:

```
forall m : map R, simple_map m -> map_colorable 4 m
```

2.7. Het intrinsieke belang van de vierkleurenstelling

Je kan je natuurlijk afvragen wat het belang is van de vierkleurenstelling, waarom wiskundigen zo hard hebben geprobeerd het te bewijzen en of de bewijzen ons wel iets leren over het *waarom* van de stelling. Sommige wiskundigen hebben beweerd dat de vierkleurenstelling maar een geïsoleerd probleem in de wiskunde is,[33] maar dit blijkt niet zo te zijn. Er zijn immers heel wat verbanden te leggen tussen de vierkleurenstelling en andere stellingen, binnen en buiten de grafentheorie. Zo verzamelde Thomas Saaty in 1972, nog vóór het bewijs van de stelling, al 29 equivalente formuleringen van wat toen nog het vierkleurenvermoeden heette.[34] In de decennia erna zijn nog heel wat diepere verbanden gevonden. Robin Thomas bespreekt enkele recente resultaten.[35] Eén van de verrassendste resultaten is een stelling van Louis H. Kauffman over vectorkruisproducten in drie dimensies die equivalent is aan de vierkleurenstelling. Deze twee domeinen lijken niets met elkaar te maken te hebben en toch is er een verband tussen.[36]

[32] [Gonthier2004] p. 51-54

[33] Bijvoorbeeld [Bonsall1982] p. 13: “The problem itself may have very little intrinsic importance.”

[34] [Saaty1972]

[35] [Thomas1998]

2.8. De complexiteit van de vierkleurenstelling

Alle huidige bewijzen van de vierkleurenstelling zijn door hun uitgebreidheid niet in detail door mensen na te kijken. Ze maken allemaal gebruik van dezelfde bewijsmethode: het zoeken van een onvermijdelijke verzameling van reduceerbare configuraties. Wiskundigen begonnen zich af te vragen of er geen eenvoudiger bewijs te vinden is van de vierkleurenstelling, misschien zelfs een bewijs waar de computer niet meer nodig is. Appel en Haken vinden dit erg onwaarschijnlijk voor bewijzen die van de techniek van onvermijdelijke verzameling van reduceerbare configuraties gebruik maken, omdat er een groot aantal configuraties nodig zijn. Ook Mayer geeft dit als reden voor de grote complexiteit van het bewijs van Appel en Haken. Elk bewijs dat deze bewijstechniek gebruikt, zal volgens hem moeten beroep doen op computerberekeningen.[37] Uiteraard sluit dit niet uit dat iemand ooit een bewijs zonder computer kan leveren dat gebruik maakt van een heel andere bewijstechniek. Appel en Haken suggereerden dit en ook Mayer ziet dit als een mogelijkheid. Hij vermoedt echter dat er eerst heel wat wiskundige doorbraken moeten gebeuren vooraleer dit mogelijk wordt.[38]

2.9. De receptie van de bewijzen bij filosofen en wiskundigen

Het bewijs van Appel en Haken was het eerste bewijs van een niet-triviale stelling dat deels door een computer geleverd werd. Volgens MacKenzie kan het als een anomalie gezien worden.[39] Er kwam dan ook heel wat reactie op dit bewijs, zowel van de kant van wiskundigen als van de kant van filosofen. Onder de filosofen werd de discussie vooral losgeweekt door een artikel van Thomas Tymoczko over het bewijs van Appel en Haken.[40] De bewijzen van Robertson en Gonthier hebben door hun recente datum niet

[36][Thomas1998] p. 850. Het vectorkruisproduct is niet-associatief, dus twee associaties, bijvoorbeeld $(v_1 \times v_2) \times (v_3 \times v_4)$ en $((v_1 \times v_2) \times v_3) \times v_4$, geven vaak verschillende uitkomsten. Kauffman geeft de volgende stelling: 'Laat i,j, k de gebruikelijke eenheidsvectorbasis zijn van R^3. Als twee associaties van $v_1 \times v_2 \times \ldots \times v_k$ gegeven zijn, bestaat er een toekenning van i, j, k aan $v_1, v_2, \ldots, v_k$ zodat de evaluaties van de twee associaties gelijk en niet nul zijn.' Kauffman bewees dat deze stelling equivalent is aan de vierkleurenstelling. Hij heeft geen rechtstreeks bewijs kunnen vinden van de stelling, dat wil zeggen zonder ze te reduceren tot de vierkleurenstelling. Hier zien we dus dat een stelling uit de vectormeetkunde kan bewezen worden door gebruik te maken van een resultaat uit de grafentheorie, wat op zijn minst verrassend is.

[37][Mayer1982] p. 54: 'Cette partie de la preuve, la plus lourde de beaucoup au point de vue combinatoire, est à peu près incompressible: l'aide de l'ordinateur ne peut en être éliminée, même a posteriori.'

[38][Mayer1982] p. 57

[39][MacKenzie1999] p. 8: "It was an entity that was hard to fit into the 'boxes' of accepted ways of thinking. Like all anomalies, the Appel-Haken solution raised the question of boundaries —in this case, the boundary between mathematics and the empirical sciences."

[40][Tymoczko1979]

veel reactie meer losgeweekt, maar ze laten wel toe kanttekeningen te plaatsen bij de reacties op het Appel-Haken bewijs.

De reacties kunnen onderverdeeld worden in vier categorieën, die we in deze sectie één voor één uiteenzetten. Het eerste standpunt is dat het bewijs van Appel en Haken helemaal geen bewijs is. Het tweede standpunt is dat het bewijs van Appel en Haken ons verplicht het concept van bewijs te veranderen. Het derde is dat het bewijs van Appel en Haken helemaal niet anders is dan andere bewijzen. Los van deze kwestie waren er ook heel wat mensen die reageerden dat de Appel-Haken oplossing wel een bewijs was, maar geen 'mooi' bewijs.[41]

2.9.1. Een computerbewijs is geen bewijs

Een eerste categorie van reacties kwam van mensen die het Appel-Haken bewijs helemaal geen bewijs vonden. Deze mening kwam vooral voor onder wiskundigen, die vaak als reden opgaven dat ze het bewijs niet konden inspecteren zoals een klassiek wiskundig bewijs en dat ze als gevolg daarvan geen zekerheid hebben over de correctheid.[42] Dit werd al vlug duidelijk na de aankondiging van het bewijs van Appel en Haken. In augustus 1976 kondigde Haken het bewijs van het 125 jaar oude vierkleurenvermoeden aan op de jaarlijkse bijeenkomst van de American Mathematical Society. Hij gaf er een uiteenzetting over het computerbewijs. Donald Albers, die daar aanwezig was, beschrijft dat hij na de uiteenzetting van dit grote resultaat een staande ovatie verwachtte, maar dat de wiskundigen in de plaats slechts uit beleefdheid applaudisseerden.[43]

Albers verbaasde zich over deze koele receptie en begon een aantal van de aanwezigen te ondervragen. Wiskundige na wiskundige bleek zich ongemakkelijk te voelen bij een bewijs waar de computer zo'n grote rol speelde. Ze vertrouwden het niet dat er zo'n 1200 computeruren nodig waren om honderdduizend gevallen te controleren en ze waren ervan overtuigd dat er in de honderden pagina's computeruitvoer wel een fout zou zitten. Ze hoopten ook dat iemand een korter bewijs zou vinden.

Armin Haken, de zoon van Wolfgang Haken, was in die tijd student aan de University of California in Berkeley en gaf een voordracht over het bewijs. Na zijn voordracht kwamen er twee soorten reacties, die leken gecorreleerd te zijn met de leeftijd. De aanwezigen die ouder waren dan 40 konden niet geloven dat een computerbewijs correct kon zijn en de jongere aanwezigen konden niet geloven dat het deel met 700 pagina's berekeningen met de hand correct kon zijn.[44] Het leek dus dat vooral de generatie wiskundigen die niet met de computer als instrument opgegroeid waren, niet in de correctheid van een computerbe-

[41] Een overzicht van de reacties op de bewijzen, maar niet zo strikt in categorieën onderverdeeld, is te vinden in [MacKenzie1999] en [Calude2001].

[42] Zie bijvoorbeeld wat Mayer over het onbehagen van veel wiskundigen hierover zegt, [Mayer1982 p. 49: 'Certains mathématiciens ont manifesté leur gêne devant le long cheminement de la preuve combinatoire, dissimulé pour sa plus grande partie dans les mémoires et les circuits de la machine.']

[43] [Albers1981]

[44] [MacKenzie1999] p. 41

wijs geloofden. Het feit dat het bewijs van Robertson, zo'n 20 jaar later, bijna geen zichtbare negatieve reacties losweekte, lijkt dit te bevestigen. Robertson en zijn collega's leken ook meer te vertrouwen op het computerdeel van het Appel-Haken bewijs dan op het handmatige deel.[45]

De wiskundige F. F. Bonsall ging zelfs verder dan twijfelen aan de correctheid en ontkende botweg dat de Appel-Haken oplossing als een bewijs kan beschouwd worden. Een loutere toepassing van bestaande methodes in een verificatie van verschillende gevallen door een computer, is volgens Bonsall geen bewijs.[46] Hij vergelijkt het accepteren van een computerverificatie met het accepteren van het woord van een andere wiskundige zonder het zelf te verifiëren. Dit is volgens hem een doodzonde. Als gevolg hiervan kunnen we volgens Bonsall niet *begrijpen* waarom de stelling waar is.[47] Computerbewijzen beschrijft hij als 'quasi-bewijzen', die we moeten zien als een uitdaging om een 'echt' bewijs te vinden.[48] Ook Paul Halmos lijkt deze mening te hebben en hij vergelijkt het aannemen van een computerbewijs met het geloven van het antwoord van een orakel.[49]

2.9.2. Computerbewijzen veranderen het concept van bewijs

Onder de filosofen brak de discussie over computerbewijzen vooral los na een artikel van Thomas Tymoczko uit 1979: 'The four-color problem and its philosophical significance'.[50] Tymoczko argumenteert hierin dat een computerbewijs op een essentieel punt verschilt van een traditioneel bewijs: de inspecteerbaarheid. In tegenstelling tot Bonsall neemt Tymoczko het bewijs van de vierkleurenstelling wel aan. Hij zegt echter dat we hierdoor verplicht zijn ons concept van *bewijs* te veranderen.[51] Als we ons traditionele idee van bewijs gebruiken, is de vierkleurenstelling volgens Tymoczko niet bewezen.[52]

---

[45][Robertson1996] p. 1: "There has remained a certain amount of doubt about its validity, basically for two reasons: (i) part of the A&H proof uses a computer, and cannot be verified by hand, and (ii) even the part of the proof that is supposed to be checked by hand is extraordinarily complicated and tedious, and as far as we know, no one has made a complete independent check of it. Reason (i) may be a necessary evil, but reason (ii) is more disturbing, particularly since the 4CT has a history of incorrect 'proofs'. So in 1993, mainly for our own peace of mind, we resolved to convince ourselves somehow that the 4CT really was true."

[46][Bonsall1982] p. 13: "It is worse still if the solution involves computer verification of special cases, and in my view such a solution does not belong to mathematical science at all."

[47][Bonsall1982] p. 13: "We cannot possibly achieve what I regard as the essential element of a proof —our own personal understanding— if part of the argument is hidden away in a box."

[48]Bonsalls sterke uitspraken geven soms echter de indruk dat ze meer uit protectionistische motieven voortkomen, uit schrik dat wiskundigen vervangen zullen worden door computers, zoals de laatste zin van [Bonsall1982]: "So let us avoid wasting those funds on pseudo mathematics with computers and use them instead to support a few real live mathematicians."

[49]Hersh citeert Halmos in [Hersh1997] p. 157: 'The present proof relies in effect on an Oracle, and I say down with Oracles! They are not mathematics.'

[50][Tymoczko1979], paginaverwijzingen zijn uit [Jacquette2001]

[51][Tymoczko1979] p. 246: "If we accept the 4CT as a theorem, we are committed to changing[…] the sense of the underlying concept of 'proof'."

Het bewijs van Appel en Haken is volgens hem een traditioneel bewijs met een gat in, dat ingevuld wordt door een 'goed uitgedacht experiment'. Dit maakt van de vierkleurenstelling voor Tymoczko de eerste wiskundige uitspraak die we *a posteriori* weten.

Volgens Tymoczko heeft een traditioneel wiskundig bewijs drie karakteristieken:[53]

1. Een bewijs is overtuigend.
2. Een bewijs is inspecteerbaar.
3. Een bewijs is formaliseerbaar.

Een traditioneel bewijs moet allereerst overtuigend zijn. Dit is omdat wiskunde een menselijke activiteit is: als wiskundigen niet overtuigd zijn van een bewijs, is het geen bewijs volgens Tymoczko. Uiteraard is overtuigingskracht niet de enige eigenschap. Een bewijs moet traditioneel gezien ook inspecteerbaar zijn. Een bewijs van een stelling dient als garantie voor de waarheid van de stelling, dus we moeten het kunnen nalezen, de stappen verifiëren en zo tot begrip kunnen komen van het bewijs. Ten derde moet een bewijs volgens Tymoczko formaliseerbaar zijn: je kan altijd een formele taal vinden waarin je een informeel bewijs rigoureus kan uitschrijven.

Tymoczko beweert nu dat het bewijs van Appel en Haken van de vierkleurenstelling wel overtuigend en formaliseerbaar is, maar niet inspecteerbaar. Daarom is het volgens hem geen traditioneel bewijs. Het deel wat Appel en Haken gepubliceerd hebben, is een traditioneel bewijs dat inspecteerbaar is (én overtuigend en formaliseerbaar), maar het lemma dat door de computer bewezen werd, is niet inspecteerbaar.[54] Tymoczko geeft daarmee in zekere zin de wiskundigen gelijk die de Appel-Haken oplossing geen bewijs vinden omdat ze het niet kunnen nakijken. Maar hij is toch genoeg overtuigd van de waarde van het bewijs om voor te stellen dat we ons concept van bewijs moeten veranderen. Met dit bewijs laten we volgens Tymoczko empirische factoren in de wiskunde binnen.

Het statuut van het bewijs van het reduceerbaarheidslemma door de computer illustreert Tymoczko met een parabel. Hij stelt zich voor dat er een gemeenschap van marsmannetjes bestaat die een wiskunde zoals bij ons ontwikkeld hebben. Op een bepaald moment krijgen ze het bezoek van een (buiten)aards wiskundig genie, Simon. Deze begon heel wat belangrijke wiskundige stellingen te bewijzen met behulp van de hen bekende traditionele wiskundige technieken en de marsbewoners waren onder de indruk van zijn resultaten. Na een tijdje begon Simon zijn resultaten echter te verantwoorden met uitspraken als 'Het bewijs is veel te lang om te publiceren, maar ik heb het zelf geverifieerd.' Eerst

[52][Tymoczko1979] p. 246: "No mathematician has seen a proof of the 4CT, nor has any seen a proof that it has a proof."
[53][Tymoczko1979] p. 246
[54][Tymoczko1979] p. 255: "Has the 4CT a surveyable proof? Here the answer is no. No mathematician has surveyed the proof in its entirety; no mathematician has surveyed the proof of the critical reducibility lemma. It has not been checked by mathematicians, step by step, as all other proofs have been checked. Indeed, it cannot be checked that way."

deed Simon dit enkel voor combinatorische lemma's, maar na een tijdje deed hij dit zelfs voor belangrijke abstracte stellingen. De wiskundige marsbewoners konden vaak na heel wat moeite wel zelf een bewijs vinden, maar soms ook niet. Omdat Simons wiskundige genialiteit echter zonder voorgaande was, accepteerden ze ook die resultaten. Ze werden gecatalogiseerd onder de rubriek ‘Simon zegt het’. Tymoczko beweert dat ons vertrouwen in computerberekeningen voor een bewijs overeenkomt met de uitspraak ‘Simon zegt het’: we geloven het resultaat omwille van de autoriteit van Simon/de computer op vlak van berekeningen. Computers zijn dus een vorm van autoriteit die ons geen bewijs geven, enkel een resultaat.

Ook als het gaat over de correctheid van het Appel-Haken bewijs, is Tymoczko het eens met de sceptici. Computerprogramma's kunnen fouten bevatten die lange tijd onopgemerkt kunnen blijven. Tymoczko vindt het bewijs van de vierkleurenstelling minder betrouwbaar dan een traditioneel bewijs, omdat het afhangt van een complexe verzameling empirische factoren. Een computerbewijs is als een wetenschappelijk experiment, dat je meerdere keren kan herhalen in verschillende omstandigheden (i.c. op verschillende computers) om meer zekerheid te hebben dat je theorie juist is. We laten hiermee de onzekerheid van de experimentele wetenschappen de wiskunde binnen.[55]

Wiskundigen hebben dus volgens Tymoczko twee manieren om bewijzen te controleren: direct zoals het traditioneel al eeuwen gebeurt of indirect met behulp van een computer. De extra onzekerheid die we hiermee introduceren, moeten we er maar bij nemen.[56] Tymoczko vindt het Appel-Haken bewijs vanuit filosofisch standpunt dan ook belangrijk omdat het toont dat ons geïdealiseerde beeld van wiskundige kennis moet evolueren: het beeld van wiskunde als 100% zekere, a priori kennis, is achterhaald. Hij stelt een nieuwe filosofie van de wiskunde voor, die empirische overwegingen serieus neemt.

Tymoczko heeft zowel argumenten voor als tegen computerbewijzen geleverd en zijn subtiele standpunten zijn niet altijd even duidelijk. Op één plaats geeft hij echter heel duidelijk weer wat hij wil zeggen:[57]

> Sometimes readers assume that because I raise *questions* about computer proofs, I mean to reject them. This is quite wrong. In fact I think that mathematicians should accept some computer proofs but accept them for the right reasons. Obviously mathematicians shouldn't accept all purported computer proofs any more than they should accept all purported hand proofs. In both cases, some purported proofs are simply wrong. In the case of hand proofs mathematicians distinguish the good from the bad by examining the written copy and verifying that the argument presented is valid. This is precisely what they cannot do in the case of computer proofs.

[55][Tymoczko1980] p. 133: “That small possibility of error does preclude mathematicians from knowing the Four-Color Theorem with absolute certainty. The proof is not rigorous.”
[56][Tymoczko1980] p. 136: “The indirect check brings a new kind of fallibility into mathematics, but that is the price of progress.”
[57][Tymoczko1981] p. 124

> What is presented is not a valid argument, but at best a probabilistic argument that a valid proof exists. The right reasons should make it clear that the probability is high enough to justify mathematician's acceptance of the result.

2.9.3. Een computerbewijs is een bewijs

Het verbaast natuurlijk niet dat Appel en Haken zelf er alle moeite voor deden om anderen ervan te overtuigen dat hun computerbewijs een echt bewijs is. Ze lieten hun bewijs dus nakijken door de beste experts van dat moment. Jean Mayer keek het deel met hun ontladingsprocedure na en Frank Allaire het deel voor de reduceerbaarheid. Allaire vergeleek de resultaten van de programma's van Appel, Haken en Koch met de resultaten van zijn eigen programma's. Alle gevallen die hij vergeleek, kwamen exact overeen.

Een belangrijk argument in het kamp van de ‘believers’ is dat je de resultaten van een computerbewijs kan controleren door verschillende programma's de berekeningen te laten uitvoeren op verschillende computers en ze te vergelijken. Volgens Appel en Haken ligt de oorzaak van het wantrouwen van wiskundigen voor computerverificatie van computerbewijzen in het feit dat ze voor de tijd van de computers hun opleiding hebben gehad. Deze wiskundigen zien de computer dan ook niet als een werktuig dat ze kunnen gebruiken.[58] Appel en Haken beweren zelfs dat bij handmatig controleren van een bewijs er meer fouten zullen gebeuren dan bij controle door een computerprogramma. Als de berekeningen elementair genoeg zijn, is de correctheid van de programma's zelfs eenvoudiger te verifiëren dan de correctheid van handmatige berekeningen.

Een hiermee samenhangende vergelijking in dit kamp is dat de computer gewoon een instrument is dat het redeneervermogen van de mens uitbreidt, net als potlood en papier. Als we een bewijs op papier construeren door diagramma's en tekst te schrijven, twijfelt niemand (als het formeel correct is) dat het een bewijs is. Edward Swart bijvoorbeeld, die ook aan het vierkleurenprobleem werkte, vindt het vreemd dat wiskundigen wel bewijzen met potlood en papier aanvaarden, maar geen bewijzen met behulp van een computer.[59]

De bewijzen van vele (grafentheoretische en andere) stellingen vallen uiteen in drie delen:

1. Bewijs dat de stelling waar is indien een bepaalde eindige verzameling van gevallen een zekere eigenschap hebben.
2. Bekom een exhaustieve lijst van deze gevallen.
3. Bewijs dat alle gevallen van deze verzameling de gevraagde eigenschap bezitten.

[58] [Appel1978] p. 207 in [Jacquette2001]

[59] [MacKenzie1999] p. 50, interview met Swart: “For the most part I regard computer-assisted proof as just an extension of pencil and paper. I don't think there is some great divide which says that OK, you are allowed to use pencil and paper but you are not allowed to use a computer because that changes the character of the proof.”

We noemen deze bewijsvorm *case testing*. Als de verzameling klein genoeg is en de gevallen eenvoudig genoeg, kan het testen van de verschillende gevallen uit het hoofd gedaan worden. De vierkleurenstelling ligt (op dit moment) aan het andere eind van het spectrum: de verzameling is zo groot en de gevallen zijn zo complex dat we het bewijs onmogelijk zonder computer kunnen leveren. Swart deelt stellingen waarvan de bewijzen van de *case testing* vorm zijn in vier categorieën in:[60]

1. De stellingen waarvan het testen van de gevallen in ons hoofd kan gedaan worden.
2. De stellingen waarvan het testen van de gevallen onmogelijk kan uitgevoerd worden zonder de hulp van potlood en papier.
3. De stellingen waarvan het testen van de gevallen slechts in duizenden manuren met potlood en papier kan uitgevoerd worden.
4. De stellingen waarvan het testen van de gevallen onmogelijk kan uitgevoerd worden zonder de hulp van een computer.

Een stelling zit slechts in een bepaalde categorie omdat er een bepaald bewijs voor geleverd is. Met de ontwikkeling van nieuwe wiskundige technieken en als gevolg daarvan nieuwe bewijzen kan een stelling naar een andere categorie verplaatsen. De vierkleurenstelling zit nu nog altijd in categorie 4. Als iemand een ingenieus kort bewijs vindt, kan de stelling echter van categorie veranderen, tot categorie 3 of wie weet zelfs categorie 2. Als iemand (Swart reageert hier onder andere tegen Tymoczko) dus zegt dat de vierkleurenstelling geen a priori waarheid is op basis van het feit dat de huidige bewijzen een computer nodig hebben, vindt Swart dat heel vreemd. Je classificeert een stelling dan als a priori of a posteriori kennis op basis van een tijdelijke categorisatie.[61]

Volgens Swart maakt het niet uit voor de waarheid van een wiskundige stelling of ze met potlood en papier of met een computer bekomen is. Hij vergelijkt dit met bewijzen dat een bepaald getal een priemgetal is. Voor bepaalde getallen (bijvoorbeeld 31) kunnen we dit uit ons hoofd, maar voor andere getallen (bijvoorbeeld $2^{30402457} - 1$, een *Mersennepriemgetal*) kunnen we dit enkel met een computer. Swart zegt dat het absurd is om het eerste een ‘a priori priemgetal’ en het tweede een ‘a posteriori priemgetal’ te noemen.

Terwijl Tymoczko beweert dat het bewijs van Appel en Haken niet zo betrouwbaar is als een traditioneel bewijs omdat het afhangt van empirische factoren, beweert Swart het omgekeerde: computerbewijzen zijn betrouwbaarder dan traditionele bewijzen omdat wiskundigen moe of afgeleid worden en computers niet.[62] Een traditioneel bewijs dat

[60][Swart1980] p. 699

[61][Swart1980] p. 700

[62]Swart gaat zelfs verder in [Swart1980] p. 700: “I would go so far as to say that any lack of reliability of the present proofs of the 4CT resides less in the use of a computer for the reducibility testing and more in the fact that a computer was not used to create the unavoidable set of configurations arising from the discharging procedure.” Dit komt overeen met wat Robertson in [Robertson1996] als motivatie gaf voor hun bewijs van de vierkleurenstelling.

door een wiskundige nagekeken wordt hangt volgens Swart bovendien van empirische factoren af. Er zullen in een computerbewijs ook wel eens fouten voorkomen, maar dat zijn dan fouten in de computerimplementatie van het algoritme, te vergelijken met schrijffouten in bewijzen met potlood en papier.

Paul Teller heeft ruwweg dezelfde opvatting als Swart. Voor hem is er ook geen vuiltje aan de lucht met de introductie van computerbewijzen in de wiskunde en hij reageert sterk tegen Tymoczko. Hij spreekt zowel diens bewering dat we ons concept van bewijs moeten veranderen tegen als diens bewering dat computerbewijzen empirische experimenten in de wiskunde introduceren. Volgens Teller maakt Tymoczko geen onderscheid tussen *een bewijs* en *verificatie van een bewijs*. De inspecteerbaarheid van een bewijs is geen noodzakelijke voorwaarde om van een traditioneel bewijs te kunnen spreken, maar slechts een eigenschap die we graag zouden hebben. Dat we met computerprogramma's een nieuwe manier hebben om een bewijs te verifiëren, verandert niets aan ons concept van bewijs.[63] Moeilijke inspecteerbaarheid is trouwens niet enkel een eigenschap van computerbewijzen, zegt Teller. Sommige traditionele bewijzen zijn zo moeilijk of uitgebreid dat maar een klein aantal wiskundigen ze verstaan en de correctheid ervan kunnen nagaan.[64]

De ‘Simon zegt het’ parabel van Tymoczko komt volgens Teller niet overeen met hoe wij met een computerbewijs omgaan. De marsmannetjes van Tymoczko begrijpen niet hoe Simon aan zijn resultaten komt en ze verifieert, terwijl we van computerbewijzen volledig kunnen nagaan hoe de computer zijn resultaten vindt. Het gaat om een nieuwe manier van verifiëren, geen nieuw begrip van wat er geverifieerd wordt. Het feit dat de reduceerbaarheidstesten in het Appel-Haken bewijs door een computer zijn geleverd in plaats van door wiskundigen van vlees en bloed, verandert dus niets aan de status van het bewijs.[65] Op dezelfde wijze is een heel complex bewijs van een goede wiskundige nog geen ander soort bewijs dan een eenvoudig bewijs dat een gemiddelde scholier kan volgen.

Ook het feit dat er fouten kunnen voorkomen in een computerbewijs, verandert volgens Teller niets aan ons concept van bewijs. Tymoczko geeft namelijk geen enkele reden om aan te nemen dat de fouten die computers maken en de fouten die wiskundigen maken verschillend zijn. Teller ziet dan ook niet in waarom we een computerbewijs als een experiment moeten zien en een traditioneel bewijs niet.

Michael Detlefsen en Mark Luker reageren nog sterker dan Swart en Teller op Tymocz-

[63][Teller1980] p. 798: “A shift in the means of surveying actually used means only a shift in the methods of checking proofs, not a shift in our conception of the things checked.” en p. 799: “The use of a computer in the 4CT is an extension of our means of surveying, not a change in our concept of proof. […] The computer is used merely to survey the long list of combinations. The novelty is in the means of surveying, not in the means of proof.”

[64]Denk aan het Wiles-Taylor bewijs van de Laatste Stelling van Fermat. Zie [Singh1998] voor een geschiedenis van het probleem en zijn bewijs.

[65][Teller1980] p. 800: “If a computer is programmed to use the same methods of proof we use, a proof that it produced would be a proof in our old sense.”

ko.[66] Zij zijn het met Teller eens dat het Appel-Haken bewijs even goed een bewijs is als traditionele bewijzen en dat dit niets te maken heeft met inspecteerbaarheid. Zij zijn het echter eens met Tymoczko dat er empirische factoren meespelen in het computerbewijs, maar ze vinden dat Tymoczko dit niet ver genoeg doordenkt. Volgens Detlefsen en Luker steunen we bij *alle* bewijzen op empirische overwegingen.

Ze wijzen er ook op dat Tymoczko te veel belang hecht aan het Appel-Haken bewijs als een resultaat dat ons dwingt om empirische factoren in de wiskunde te brengen. Elsie Cerutti en Philip Davis hebben immers tien jaar eerder al een computerbewijs van een stelling van Pappus beschreven en een empirische opvatting op wiskunde geïntroduceerd.[67] Detlefsen en Luker nemen deze opvatting van Davis over en beweren dus dat empirische bewijzen wijdverspreid zijn in de wiskunde.

Als Tymoczko zegt dat de inspecteerbaarheid van een bewijs ervoor zorgt dat het ons a priori kennis levert, onafhankelijk van empirische experimenten, dan zijn Detlefsen en Luker het daar niet mee eens. De inspecteerbaarheid van een bewijs garandeert niet dat het niet gebaseerd is op empirische overwegingen. Dat wiskundige stellingen a priori gekend zijn, nemen zij dus niet aan. Volgens hen neemt Tymoczko's filosofie van de wiskunde empirische overwegingen niet serieus genoeg.[68]

Ook Israel Krakowski en Margarita Levin bleken tot dezelfde conclusie te komen als Detlefsen en Luker: Tymoczko's argumenten zijn sterker dan hij zelf denkt en computerbewijzen zeggen dus niets nieuws over empirische elementen in de wiskunde.[69] Krakowksi reageert allereerst tegen Tymoczko's bewering dat Appel en Hakens bewijs van de vierkleurenstelling niet inspecteerbaar is: natuurlijk is het bewijs inspecteerbaar, zegt hij, omdat elk eindig formeel bewijs inspecteerbaar is.[70] Bovendien is het ook effectief geïnspecteerd: verschillende computers hebben het combinatorische lemma onafhankelijk van elkaar geverifieerd en het maakt niet uit dat het niet door een mens maar door een computer gebeurd is.[71] De meeste wiskundigen kunnen trouwens complexe bewijzen niet helemaal inspecteren, maar vertrouwen op de autoriteit van andere wiskundigen die het wel gedaan hebben, zoals de *referees*.[72] Dat geen enkele mens het bewijs kan inspec-

[66][Detlefsen1980]

[67][Cerutti1969], dit hebben we in de inleiding besproken.

[68][Detlefsen1980] p. 817: “It is the empirical character of survey and the empirical character of calculation or computation which are responsible for (at least much of) the presence of empirical elements in mathematical proof.”

[69][Krakowski1980] en [Levin1981]. Krakowksi en Levin kwamen blijkbaar onafhankelijk van elkaar en van Detlefsen en Luker tot dezelfde conclusie. Ze verwijzen alleszins niet naar elkaar, noch naar Davis die eigenlijk al tien jaar tevoren hetzelfde standpunt innam.

[70][Krakowski1980] p. 92: ‘That it is surveyable follows from the fact that any finite formal proof os surveyable; mankind need merely learn to live longer to deal with the longer proofs.’

[71][Krakowski1980] p. 92: ‘For the *computer* has, in a step by step fashion, surveyed and proved this lemma. To suggest otherwise is chauvinism.’

[72][Krakowski1980] p. 92: ‘The vast majority could not follow the most complex proofs, they must appeal to authority; the proof of the 4CT merely extends the class of humans who must so appeal.’

teren is volgens Kurakowski bovendien een *empirisch accident*: als we langer zouden leven, zouden we het wel kunnen.[73]

Krakowski verwijt (in dezelfde lijn als Teller) Tymoczko dat hij met zijn 'Simon zegt het' parabel van ons vertrouwen in de resultaten van de computer iets dubieus maakt. Simon *rechtvaardigt* namelijk zijn uitspraken niet en de computer in het bewijs van de vierkleurenstelling wel.[74] Ook volgens Levin gaat de analogie met Simon niet op: we weten wat de computer doet, omdat wij hem gebouwd en geprogrammeerd hebben.[75] Levin beklemtoont dat de introductie van de computer in wiskundige bewijzen geen kwalitatieve verandering teweegbrengt, maar slechts een kwantitatieve. De computer brengt geen nieuw soort berekening in de wiskunde binnen, slechts snellere berekeningen.[76] Ze vergelijkt de stap van traditionele bewijzen naar computerbewijzen met de stap van tellen op je vingers naar tellen op een telraam. Haar besluit: 'The use of computers marks no epistemological revolution in mathematical method.'[77]

Tymoczko bekritiseert de computer ook omdat hij fouten kan maken. Krakowksi reageert daarop dat de computer zeker niet meer fouten maakt dan de mens. Hij vergelijkt zelfs de verschillende types fouten die de computer kan maken en hun equivalent bij mensen. Enerzijds kan er een subtiele fout in het computerprogramma staan, maar de menselijke wiskundige kan iets analoogs meemaken: een subtiele ongeldige stap in een redenering of een subtiele inconsistentie in het formalisme waarin hij aan het werken is.[78] Anderzijds kan de machine zelf onbetrouwbaar zijn in zijn werking, hier gaat het dan om mechanische fouten. Bij de menselijke wiskundige komen zo'n mechanische fouten volgens Krakowski echter ook voor: zijn geheugen laat het afweten of hij zet een komma of punt verkeerd.[79] De nieuwe empirische elementen die de computer in de wiskunde binnenbrengt,

[73][Krakowski1980] p. 92-93: 'There is no principled reason why, when longevity reaches astrronomical proportions, a bright young mathematician could not spend a few millenia going through the entire proof.'

[74][Krakowski1980] p. 93

[75][Levin1981] p. 83. Ook p. 85:'We must bear in mind that computers are not alien objects whose workings we have to discover or oracles that seem to *have* no workings: they are manmade objects whose workings obey our mathematics.'

[76][Levin1981] p.85:'The improvement or advantage introduced by the use of computers in proofs is, so to speak, quantitative, not qualitative: more computations can be done more quickly, but no new sort of computation that is beyond human beings has appeared.'

[77][Levin1981] p.86

[78]Denk bijvoorbeeld aan Kempes foutieve bewijs van de vierkleurenstelling, waar Heawood pas 11 jaar later een subtiele fout in vond, of Freges inconsistente logische systeem in zijn *Grundgesetze der Arithmetik* waaruit Bertrand Russell de zogenaamde 'paradox van Russell' afleidde. Meer hierover is te vinden in [Horsten2004] p. 43-48.

[79]Als tweedejaarsstudent burgerlijk ingenieur heb ik dit aan den lijve ondervonden. In een projectwerk waarbij ik met twee andere studenten de uitdaging aanging om een computerprogramma te schrijven om grote getallen te factoriseren, baseerden wij ons op een thesis van een wiskundige die een efficiënt algoritme presenteerde, de *self initializing quadratic sieve*. Onze implementatie van het algoritme bleek niet te werken en we zochten ons een maand lang suf op alle mogelijke programmeerfouten die we gemaakt konden hebben. Pas toen onze begeleider de thesis waarop we ons baseerden onderzocht en het algoritme zelf afleidde, bleek dat de auteur in één van de formules die we gebruikten een - in plaats van een + had geschreven. Dit was waarschijnlijk een toevallige typfout, maar hier zullen heel wat mensen op gevloekt hebben.

zijn er dus al zolang dat de mens aan wiskunde doet.[80] Krakowski besluit zijn betoog met:[81]

> The 4CT does not, I conclude, raise any new issues of philosophical importance, Yet there is something distinctive about the proof: I think it is that the proof *highlights* the already existing empirical elements of mathematical knowledge. And insofar as it does focus attention on these it has philosophical impact.

2.9.4. Een computerbewijs is geen bevredigend bewijs

Veel wiskundigen waren van mening dat een computerbewijs van de vierkleurenstelling wel een correct bewijs is, maar geen *mooi* of bevredigend bewijs. We zien dit bijvoorbeeld in de reacties op een computerbewijs (waar later een fout in gevonden werd) van de vierkleurenstelling door Yoshio Shimamoto in 1971. William Tutte, één van de beste grafentheoretici van die tijd, was niet tevreden over het bewijs omdat het door 'brute kracht' werd geleverd.[82] Hoewel Tutte geen fout in Shimamoto's redenering vond, begon hij enkel door de ongeloofwaardigheid van de oplossing eraan te twijfelen of het bewijs wel juist was.[83]

Wolfgang Haken werd na zijn bewijs soms zelfs vijandig benaderd. Eén wiskundige probeerde zelfs te vermijden dat Haken contact kreeg met de studenten in zijn departement. Omdat het bewijs van de vierkleurenstelling was geleverd, zouden andere wiskundigen niet meer proberen om een alternatief bewijs te vinden. Alle eer gaat immers naar het eerste bewijs. Maar omdat Haken het probleem op een 'ongepaste' manier had opgelost, aldus die collega, zou een 'beter' bewijs er nooit komen. Zijn studenten moesten dan ook beschermd worden tegen 'zondaars' als Haken.[84]

Ook Ian Stewart was niet tevreden over het bewijs, omdat het geen bevredigende uitleg geeft *waarom* de vierkleurenstelling juist is. Ten dele is dit omdat het bewijs zo lang en zo moeilijk te begrijpen is, maar ook omdat het weinig structuur lijkt te hebben volgens Stewart. *Waarom* er een onvermijdelijke verzameling van reduceerbare configuraties is, komen we niet te weten door het bewijs van Appel en Haken. Het bewijs toont gewoon dat er zo'n verzameling is.[85] Daniel I.A. Cohen noemde het bewijs van Appel en Haken zelfs denigrerend 'computerfoefjes' die intellectueel onbevredigend zijn.[86]

---

[80][Krakowski1980] p. 94

[81][Krakowski1980] p. 95

[82][MacKenzie1999] p. 31 citeert Tutte: "The feeling is that the Four Colour Theorem ought not to have been provable like that, —'by brute force', … I have wavered between belief and disbelief in Shimamoto's proof, but I have never liked it."

[83][MacKenzie1999] p. 31 citeert Tutte en Whitney: "We found no essential flaw in Shimamoto's reasoning… We therefore decided that the computer result must be wrong." Meer informatie over de fout die er in gevonden werd, is te vinden in [Mayer1982] p. 48.

[84][MacKenzie1999] p. 41 citeert Haken: "So we had done something very, very bad, and things like that should not be committed again, and he had to protect the innocent souls of his students against us."

De houding van deze wiskundigen dat ze de Appel-Haken oplossing wel accepteren als bewijs, maar er niet tevreden over zijn, kan verklaard worden door het onderscheid te maken tussen verificatie en bewijs, zoals Gian-Carlo Rota doet.[87] Verificatie is een argument dat de waarheid van een uitspraak garandeert door alle mogelijke gevallen na te gaan, zoals in alle tot nu toe geleverde bewijzen van de vierkleurenstelling gebeurd is. Verificatie is één van de verschillende soorten wiskundige bewijzen en zelfs één van de eenvoudigste. Door de eenvoud van het argument kunnen wiskundigen moeilijk anders dan het aannemen als bewijs, zelfs al gaat het om een enorm groot aantal gevallen. Maar omdat zo'n argument niet de reden geeft waarom de stelling waar is, zijn wiskundigen er niet tevreden over. Rota geeft het Appel-Haken bewijs als voorbeeld van een bewijs door verificatie. De verschillende pogingen om het bewijs te vereenvoudigen ziet hij ook als een aanwijzing dat wiskundigen niet tevreden zijn met een verificatie, maar een bewijs willen dat hen inzicht geeft.[88]

Het is trouwens opvallend dat Rota over ‘the four-color *conjecture*’ spreekt. Enerzijds lijkt hij te zeggen dat het bewijs door Appel en Haken een echt bewijs is, al is het geen bevredigend bewijs. Anderzijds blijft hij systematisch over de vierkleurenstelling als een *vermoeden* spreken. In een interview in *Los Alamos Science* uit 1985 heeft Rota het ook over het ‘four-color conjecture’. Wanneer Sharp hem onderbreekt en opmerkt dat ‘I thought that had been settled by a computer proof,’ antwoordt Rota:[89]

> Not really. What we want is a rational proof. It doesn't help to have a brutally numerical answer spewed out by a computer. A problem is interesting only when it leads to ideas; nobody solves problems for their own sake, not even chess problems. You solve a problem because you know that by solving the problem you may be led to see new ideas that will be of independent interest. A mathematical proof should not only be correct, but insightful. Although, as Erdös says, nobody gets blamed if his first proof is messy.

2.9.5. A priori kennis door computerbewijzen

Twintig jaar nadat de filosofische discussie over het vierkleurenprobleem uitbrak, pikt

[85][MacKenzie1999] p. 41 citeert Stewart: “The answer appears as a kind of monstrous coincidence. Why is there an unavoidable set of reducible configurations? The best answer at the present time is: there just is. The proof: here it is, see for yourself. The mathematician's search for hidden structure, his pattern-binding urge, is frustrated.”

[86][MacKenzie1999] p. 46-47 citeert Cohen: “Admitting the computer shenanigans of Appel and Haken to the ranks of mathematics would only leave us intellectually unfulfilled.”

[87][Rota1997b] in [Jacquette2001]

[88][Rota1997b] p. 220: “No computer verification of the four colour conjecture will be accepted as definitive. Mathematicians are on the lookout for an argument that will make all computer programs obsolete, an argument that will uncover the still hidden reason for the truth of the conjecture. […] Verification is proof, but verification may not give the reason.”

[89][Rota1985] p. 99

Tyler Burge ze terug op in zijn artikel 'Computer proof, apriori knowledge, and other minds'.[90] Hij wil de dominante visie bestrijden dat vertrouwen op een computer voor een bewijs onze kennis van de stelling empirisch maakt. Volgens Burge zijn de uitspraken van Tymoczko, Detlefsen en Luker en anderen verkeerd omdat ze zich baseren op een verkeerd idee van a priori kennis. We kunnen namelijk perfect a priori kennis hebben van een uitspraak die 'afhangt van' empirische ervaringen:[91]

> Rationalism claims rather that sense experience does not contribute to the normative or justificational force carried by some warrants. So arguing that a belief 'depends on' sense experience does nothing in itself to support a empiricist epistemology.

Volgens Burge maken deze filosofen geen onderscheid tussen rechtvaardigende (of epistemische) afhankelijkheid en andere types van afhankelijkheid. Van een rechtvaardiging zegt Burge dat die *a priori* is als we noch naar zintuiglijke indrukken noch naar perceptuele overtuigingen verwijzen of erop vertrouwen om de rechtvaardigende kracht van de rechtvaardiging te bepalen.[92] Kortom, we moeten een onderscheid maken tussen de rol die zintuiglijke indrukken spelen bij het bekomen van een overtuiging en bij het bijdragen aan de normatieve kracht van de rechtvaardiging van de overtuiging.[93]

Als we dus op empirische factoren vertrouwen zoals de fysische werking van onze eigen hersenen tijdens het uitschrijven van een bewijs, vinden we niet dat dit bijdraagt aan de rechtvaardiging van onze overtuiging dat de wiskundige stelling waar is. De factoren spelen wel een rol als we willen bepalen hoe we de overtuiging dat de stelling waar is bekomen zijn. De vraag is volgens Burge dus of vertrouwen op de correcte werking van een computer hetzelfde is als vertrouwen op de correcte werking van onze hersenen. Een duidelijk verschil in het geval van de vierkleurenstelling is dat het computerbewijs van Appel en Haken niet inspecteerbaar is. Volgens Burge volgt hier echter niet uit dat ons geloof in het resultaat van de computer empirisch is.[94]

Samengevat ziet Burge geen enkel verschil tussen het aannemen van resultaten van een computer en het aannemen van resultaten van andere wiskundigen of van zichzelf. De computer wordt een uitbreiding van je eigen rationele vermogens.[95] De kennis dat een

[90][Burge1998]

[91][Burge1998] p. 2

[92][Burge1998] p. 3: 'A warrant (either a justification or an entitlement), is *a priori* if neither sense experiences nor sense-perceptual beliefs are referred to or relied upon to contribute to the justificational force particular to that warrant.'

[93][Burge1998] p. 3: 'The role of sense experience in the psychology and acquisition of belief must be distinguished from its role in contributing to the normative force associated with the belief's warrant.'

[94][Burge1998] p. 7-8: 'Is the unsurveyability of the computer's deduction sufficient ground in itself for taking belief in the computer's result to be empirical? The answer is "no". An unsurveyable deductive argument is no *more* inherently empirical than a non-demonstrative argument in mathematics.'

[95][Burge1998] p. 31: 'As one comes to learn from the computer, to understand and rely on its argument and results, it becomes analogous to one of one's own rational faculties.'

stelling is bewezen door de computer kan ondersteund worden door a priori rechtvaardiging. Andere wiskundigen of computers kunnen bronnen zijn van a priori rechtvaardiging van je kennis. Kennis die gecommuniceerd wordt is volgens Burge daarom niet direct empirisch, we kunnen a priori kennis overdragen door communicatie.[96]

2.9.6. Locale versus globale inspecteerbaarheid

O. Bradley Bassler analyseerde de verschillende categorieën reacties op het computerbewijs van de vierkleurenstelling en plaatste dit in een historisch perspectief in 'The surveyability of mathematical proof: a historical perspective'.[97] Hij onderscheidt twee vormen van inspecteerbaarheid: locale en globale. Een bewijs is lokaal inspecteerbaar wanneer je alle individuele stappen van het bewijs in een bepaalde volgorde kan inspecteren. Een bewijs is globaal inspecteerbaar wanneer je het bewijs als een mentaal te bevatten geheel kan inspecteren. Veel bewijzen zijn niet globaal inspecteerbaar en volgens Bassler heeft de drang van wiskundigen om meer begrijpelijke bewijzen te vinden als gevolg dat de nieuwe bewijzen ook globaal meer inspecteerbaar zijn.[98]

Volgens Bassler probeerden filosofen in de 17de eeuw, zoals Descartes, globale inspecteerbaarheid te reduceren tot lokale inspecteerbaarheid.[99] Zo kon een wiskundige volgens Descartes van een stelling enkel kennis hebben als hij het bewijs ervan volledig stap voor stap had geverifieerd. Het vurige debat rond het computerbewijs van de vierkleurenstelling is volgens Bassler te wijten aan dezelfde fout die in de 17de eeuw gemaakt werd: het onvermogen om het onderscheid te maken tussen lokale en globale inspecteerbaarheid.[100] Volgens Tymoczko drijft het computerbewijs van de vierkleurenstelling een wig tussen de criteria van inspecteerbaarheid en formaliseerbaarheid van een bewijs. Volgens Bassler gaan *alle* antwoorden in het debat op deze stelling ervan uit dat inspecteerbaarheid niet zo belangrijk is voor een bewijs, omdat ze formalisering zien als een idealisatie van de notie van locale inspecteerbaarheid. Globale inspecteerbaarheid negeren ze dus.[101]

Bassler somt de respondenten op:[102] voor Margarita Levin is een bewijs een eindig aantal stappen, waaronder ook berekeningen. Dat die berekeningen door computers of mensen

[96][Burge1998] p. 31: 'Apriori knowledge can be transmitted through communication —even when the recipient cannot alone justify his knowledge, and even when the source must be accored special authority if reliance on it is to be warranted.' De overdraagbaarheid van a priori kennis door communicatie is het onderwerp van [Burge1993].

[97][Bassler2006]

[98][Bassler2006] p. 104: '[…] within the mathematical community considerable time is devoted to improving proofs to make them more comprehensible and, hence, more globaly surveyable.'

[99][Bassler2006] p. 100

[100][Bassler2006] p. 110

[101][Bassler2006] p. 119: 'In all the responses to Tymoczko's article, positions have been crafted in such a way that the purported nonsurveyability of the proposed proof of the 4CT does not present any serious philosophical problem. This, I claim, is in general an immediate consequence of the strongly held commitment to formalism as itself an idealization of the notion of local surveyability, with a consequent exclusion of any concern for global surveyabiliity.'

[102][Bassler2006] p. 117-118

uitgevoerd worden, maakt volgens haar geen kwalitatief verschil. Formalisering is voor haar dus het enige criterium van een bewijs. Krakowski reduceert volgens Bassler inspecteerbaarheid tot de twee andere criteria van een bewijs: overtuigendheid en formaliseerbaarheid. Ook Teller aanvaardt inspecteerbaarheid niet als een criterium voor een bewijs. Volgens hem heeft inspecteerbaarheid te maken met het verifiëren van een bewijs. Ook Detlefsen en Luker zien inspecteerbaarheid niet als een cruciaal element van onze notie van bewijs.

Uiteraard geldt dit niet voor de groep in het debat die het computerbewijs van de vierkleurenstelling geen bewijs noemt. Mensen als Bonsall en Halmos hekelden de niet-inspecteerbaarheid van het bewijs en wilden het daarom niet aannemen. Bassler noemt ook Stuart Shanker die beweert dat de vierkleurenstelling niet bewezen is omdat het 'bewijs' ervan niet inspecteerbaar is.[103]

Als we in retrospect kijken naar deze discussie, zien we volgens Bassler dat de deelnemers inspecteerbaarheid enkel als lokale inspecteerbaarheid beschouwden. Volgens Bassler is het echte dilemma waar het bewijs van de vierkleurenstelling ons op wijst het feit dat het bewijs globaal inspecteerbaar is zonder lokaal inspecteerbaar te zijn.[104] De globale structuur van het bewijs is immers globaal inspecteerbaar: we zien eenvoudig dat het vinden van een onvermijdelijke verzameling van reduceerbare configuraties de stelling bewijst. De niet-inspecteerbaarheid van het bewijs is enkel te vinden in de grote verzameling berekeningen die verifiëren dat alle configuraties in de onvermijdelijke verzameling reduceerbaar zijn.[105] We hebben hier dus een lokaal niet-inspecteerbaar maar globaal wel inspecteerbaar bewijs. Als gevolg hiervan weten we dat globale inspecteerbaarheid niet kan gereduceerd worden tot lokale inspecteerbaarheid.

[103][Bassler2006] p. 118

[104][Bassler2006] p. 123

[105][Bassler2006] p. 126: 'Then, finally, there is the monumental computer calculation which is involved in the verification that what is constructed in the procedure described in Appel, Haken and Koch's article is indeed an unavoidable set.' Wat Bassler hier zegt is overigens niet helemaal juist. De computerberekeningen verifiëren dat alle configuraties in de onvermijdelijke verzameling reduceerbaar zijn; het bewijs dat de verzameling van configuraties onvermijdelijk is, hebben Appel en Haken volledig met de hand uitgewerkt.

3. Er bestaat geen eindig projectief vlak van orde 10

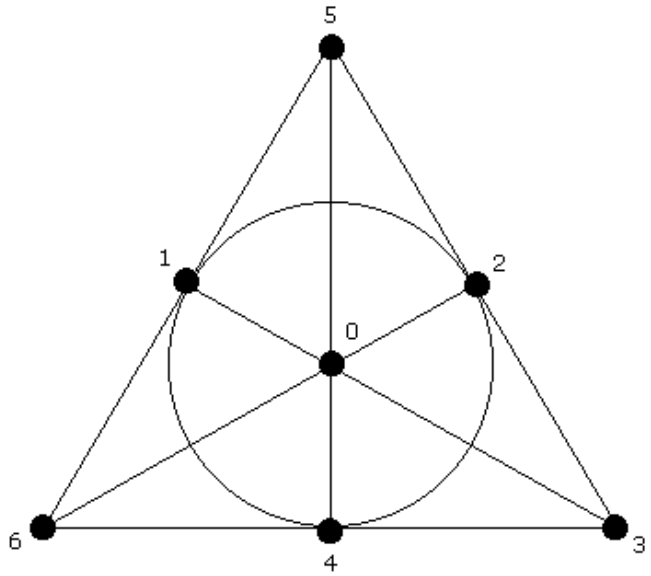
3.1. Het probleem en zijn geschiedenis

Een *eindig projectief vlak van orde n*, met n > 0, is een verzameling van $n^2 + n + 1$ lijnen en $n^2 + n + 1$ punten die aan de volgende voorwaarden voldoen:

1. Elke lijn bevat n + 1 punten.
2. Elk punt ligt op n + 1 lijnen.
3. Twee verschillende lijnen snijden elkaar in exact één punt.
4. Twee verschillende punten liggen op exact één lijn.

Het kleinste voorbeeld van een eindig projectief vlak verkrijg je voor n = 1: het eindig projectief vlak van orde 1, een driehoek. Elke lijn van de driehoek bevat 2 punten en elk punt van de driehoek ligt op twee lijnen. Twee verschillende lijnen van de driehoek snijden elkaar in exact één punt en twee verschillende punten liggen op één lijn. Het kleinste *niet-triviale* voorbeeld is van orde 2: zeven lijnen en zeven punten die aan de vier voorwaarden voldoen. Dit projectief vlak staat bekend als het *Fano-vlak*.

Figuur 3.1. Het Fano-vlak



Eén van de lijnen in het projectieve vlak is hier voorgesteld als een cirkel, die punten 1, 2 en 4 bevat.

Van 1904 tot 1907 bewezen Oswald Veblen, William Henry Bussey en Joseph Henry Maclagen Wedderburn het bestaan van de meeste eindige projectieve vlakken van kleine orde. Het geval n = 6 bleek echter te moeilijk te zijn. In 1938 relateerde de Indiase wiskundige Raj Chandra Bose het bestaan van een eindig projectief vlak van orde n met het bestaan van een *Grieks-Latijns vierkant*. Hiervoor moeten we eerst een ander concept introduceren:

Een *Latijns vierkant* van orde n is een n x n matrix met de volgende eigenschappen:[1]

1. Op elke plaats in de matrix bevindt zich een getal tussen 1 en n.
2. Elk getal komt maar één keer voor in een rij.
3. Elk getal komt maar één keer voor in een kolom.

We noemen twee Latijnse vierkanten *orthogonaal* als alle geordende tweetallen van de getallen op overeenkomende plaatsen van de twee vierkanten verschillend zijn. De twee volgende Latijnse vierkanten van orde 4 zijn bijvoorbeeld orthogonaal:

Figuur 3.2. Latijns vierkant 1

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

en

[1]Merk de gelijkenis op met de sudoku puzzels die tegenwoordig zo populair zijn. Meer uitleg over Latijnse vierkanten en sudoku's is te vinden in het populariserend boek [VanDenEssen2006].

Figuur 3.3. Latijns vierkant 2

1	2	3	4
3	4	1	2
4	3	2	1
2	1	4	3

Het geordend tweetal (1, 1) komt bijvoorbeeld maar één keer voor, net zoals alle andere. Van twee orthogonale Latijnse vierkanten van orde n kunnen we een *Grieks-Latijns vierkant* (ook Eulervierkant genoemd) van orde n maken door de getallen op de overeenkomende plaatsen bij elkaar te plaatsen. Zo kunnen we uit de twee voorgaande orthogonale Latijnse vierkanten van orde 4 het volgende Grieks-Latijns vierkant construeren:

Figuur 3.4. Grieks-Latijns vierkant

11	22	33	44
23	14	41	32
34	43	12	21
42	31	24	13

De oorsprong van de benaming Grieks-Latijns vierkant ligt in een artikel van Euler uit 1782, ‘Recherches sur une nouvelle espèce de quarrés magiques’. Euler stelt hierin het volgende probleem voor: stel 36 officieren van 6 rangen en uit 6 regimenten zo op in een vierkanten opstelling dat elke verticale en horizontale lijn van de opstelling exact één of-

ficier van elke rang bevat en exact één officier van elk regiment. Euler gaf de rangen de Griekse letters α, β, γ, δ, ε, ζ en de regimenten de Latijnse letters a, b, c, d, e, f. Eulers probleem reduceert zich dan tot het construeren van een Grieks-Latijns vierkant van orde 6 door een paar van orthogonale Latijnse vierkanten van orde 6 te zoeken.

Euler wist dat er geen Grieks-Latijns vierkant bestond voor n = 2. Dit is triviaal: er zijn maar twee Latijnse vierkanten van orde 2 en deze zijn niet orthogonaal. Hij kon bovendien geen oplossing vinden voor n = 6, zijn probleem van de 36 officieren. Hij uitte daarop het vermoeden dat er geen Grieks-Latijns vierkant bestaat voor alle n met rest 2 bij deling door 4, dus voor ordes 2, 6, 10, 14, …[2]

Bose bewees dat er een beperking ligt op het aantal elementen in een verzameling van onderling orthogonale Latijnse vierkanten: voor t onderling orthogonale Latijnse vierkanten van orde n geldt $t \leq n - 1$. Geldt de gelijkheid, dan spreken we van een *volledige* verzameling orthogonale Latijnse vierkanten. Bose bewees nu ook:

Stelling. Voor alle n ≥ 3 kunnen we een projectief vlak van orde n construeren als en slechts als we een volledige verzameling van n - 1 onderling orthogonale Latijnse vierkanten van orde n kunnen construeren.

Als we Eulers vermoeden met de stelling van Bose combineren, betekent dit dat er geen projectieve vlakken van orde $n \equiv 2 \pmod 4$ zouden bestaan. Indien er namelijk geen Grieks-Latijns vierkant van orde n bestaat, bestaat er al zeker geen verzameling van onderling orthogonale Latijnse vierkanten. De Franse amateurwiskundige Gaston Tarry verifieerde in 1900 Eulers vermoeden voor n = 6 door een systematische opsomming van alle mogelijkheden.[3] De stelling van Bose samen met het resultaat van Tarry bewezen dat er geen eindig projectief vlak van orde 6 bestaat. Om het geval n = 10 aan te pakken, was Tarry's methode echter ongeschikt: het aantal paren van Latijnse vierkanten van orde 10 is gewoon te groot.

Bruck en Ryser maakten de volgende belangrijke stap. Zij onderzochten de eigenschappen van de *incidentiematrix* van een projectief vlak. De incidentiematrix $A = [a_{ij}]$ van een projectief vlak van orde n is een $n^2 + n + 1$ bij $n^2 + n + 1$ matrix waarvan de kolommen de punten en de rijen de lijnen van het vlak voorstellen. Het element a_{ij} is 1 als punt j op lijn i ligt; anders is het 0. Het projectief vlak van orde 2 (zie de figuur in het begin van dit hoofdstuk) heeft de volgende incidentiematrix:

[2] Genoteerd als $n \equiv 2 \pmod 4$

[3] Lam schrijft over Tarry's oplossing: ‘Yet, there is something unpleasant about a systematic hand enumeration: it is messy and it is error prone.’ ([Lam1991] p. 308) Dezelfde verwijten krijgen computerbewijzen door exhaustieve verificatie 70 jaar later.

	1	**2**	**3**	**4**	**5**	**6**	**7**
L1	1	1	0	1	0	0	0
L2	0	1	1	0	1	0	0
L3	0	0	1	1	0	1	0
L4	0	0	0	1	1	0	1
L5	1	0	0	0	1	1	0
L6	0	1	0	0	0	1	1
L7	1	0	1	0	0	0	1

De hierboven gegeven eigenschappen van een projectief vlak van orde n vertalen zich in de incidentiematrix als volgt:

1. Elke rij van A heeft als som van zijn elementen n + 1
2. Elke kolom van A heeft als som van zijn elementen n + 1
3. Het inwendig product van elke twee verschillende rijen van A is 1.
4. Het inwendig product van elke twee verschillende kolommen van A is 1.

Het inwendig product van een vector (een rij van getallen) en een andere vector is de som van de producten van de overeenkomende elementen. Dat het inwendig product van elke twee verschillende rijen van A 1 is, betekent dus dat er voor elke twee rijen exact één element is dat in beide rijen 1 is, of (vertaald in termen van het projectief vlak): elke twee lijnen hebben exact één punt gemeenschappelijk. De vier eigenschappen van de incidentiematrix A kunnen trouwens samengevat worden in één matrixvergelijking: $AA^T = nI + J$, waarbij A^T de transpose van matrix A voorstelt (dus rijen en kolommen omgewisseld), I de eenheidsmatrix (de matrix met alle elementen 0 behalve de diagonaalelementen die 1 zijn) en J de matrix met alle elementen 1.

Bruck en Ryser bewezen ook dat de incidentiematrix A van een projectief vlak *normaal* is: $AA^T = A^TA$. Zij leidden hier uit af dat dit de volgende stelling impliceert:

Stelling van Bruck-Ryser: Als $n \equiv 1, 2 \pmod 4$, dan is een noodzakelijke voorwaarde voor het bestaan van een projectief vlak van orde n dat er gehele getallen x en y bestaan waarvoor geldt dat $n = x^2 + y^2$.[4]

De stelling van Bruck-Ryser geeft hiermee nu ook een eleganter bewijs dat er geen projectief vlak van orde 6 bestaat, aangezien 6 congruent is met 2 modulo 4, maar niet kan geschreven worden als de som van twee gehele kwadraten. Daarna gingen wiskundigen

[4]Uit de getaltheorie weten we bovendien dat een natuurlijk getal n als de som van twee gehele kwadraten kan geschreven worden als en slechts als elke priemfactor van n van de vorm 4k + 3 met een even macht voorkomt in de factorisatie van n. Zie hoofdstuk 4 van [Aigner2004] voor een elegant bewijs hiervan.

verder naar het volgende nog niet opgeloste geval: n = 10. $10 = 1^2 + 3^2$, dus volgens de stelling van Bruck-Ryser is er aan de noodzakelijke voorwaarde voor het bestaan van een projectief vlak voldaan. Het is echter geen *voldoende* voorwaarde, dus de stelling vertelt ons niet of er een projectief vlak van orde 10 bestaat. Als Eulers vermoeden juist zou zijn, zou er geen projectief vlak van orde 10 bestaan, omdat $10 \equiv 2 \pmod 4$.

In 1959 construeerden Bose en Shrikhande op basis van een resultaat van Ernest Parker een paar van orthogonale Latijnse vierkanten van orde 22.[5] Dit was het eerste tegenvoorbeeld voor Eulers vermoeden ($22 \equiv 2 \pmod 4$). Bose en Shrikhande onderzochten of ze voor andere ordes ook tegenvoorbeelden konden vinden. Parker hoorde van de resultaten van zijn collega's en was er nu van overtuigd dat Eulers vermoeden ook voor andere ordes onjuist was. Hij verfijnde de resultaten van Bose en Shrikhande en vond voor het eerst een paar van orthogonale Latijnse vierkanten van orde 10.[6] Bose, Shrikhande en Parker begonnen te corresponderen en konden samen bewijzen dat Eulers vermoeden fout is voor ordes groter dan 6.[7] Hierdoor schatte men terug de kansen hoger in dat er een projectief vlak van orde 10 bestaat.

3.2. Het computerbewijs

Na deze theoretische doorbraken was de algemene opinie dat er een projectief vlak van orde 10 moest bestaan en er werden dan ook pogingen ondernomen om met een computer één te proberen construeren. Het geval n = 10 bleek al vlug een harde noot om te kraken.[8] In 1970 gaf Edward F. Assmus echter een lezing getiteld *The projective plane of order ten?* Hierin legde hij een nieuwe aanpak uit. Met een incidentiematrix A van een eindig projectief vlak van orde 10 liet hij een vectorruimte gegenereerd door de rijen van A over het eindige veld met elementen {0, 1} overeenkomen. Elke vector in V is een *codewoord*. Het *gewicht* van een codewoord is het aantal 1'en in het codewoord. Definiëren we nu w_i als het aantal codewoorden met gewicht i. De *gewichtsenumerator* van V wordt dan ge-

[5] [Bose1959], later gegeneraliseerd in [Bose1960]

[6] [Parker1959], [Parker1959b]

[7] Hun bewijs werd op 26 april 1959 voorpaginanieuws in de *Sunday New York Times*, die titelde: 'Major mathematical conjecture propounded 177 years ago is disproved' ([VanDenEssen2006] p. 59)

[8] [Cairns1954] p. 30: 'Even the existence problem for a pair of mutually orthogonal ones of order 10 has resisted all efforts to date: programming the method of exhaustive search, which Tarry carried out by hand in the case m = 6, on a modern electronic computer shows it to be beyond the scope of current machines in spite of all improvements thus far devised.', [Parker1959] p. 860: 'Considerable effort has been expended in searching by digital computer for a pair of orthogonal latin squares of order 10, but machines have proved too slow to cope with a search of such magnitude.', [Hall1955] p. 21: 'For n = 10 extensive searches on the SWAC machine at UCLA have failed to produce any orthogonal squares. But even 100 machine hours will not cover more than a microscopic part of the complete search.'

definieerd als $\Sigma w_i x^i$ voor alle i van 0 tot en met 111 (= $10^2 + 10 + 1$). Assmus en Mattson bewezen dat de gewichtsenumerator uniek afhangt van w_{12}, w_{15} en w_{16}. De hoop was nu dat men hieruit een contradictie kon afleiden, zoals een negatief of niet-geheel gewicht.

MacWilliams, Sloane en Thompson bewezen dat $w_{15} = 0$.[9] Hun bewijs bestaat uit een theoretisch argument, maar maakt gebruik van een computer om de onmogelijkheid van een bepaalde configuratie te verifiëren. Dit gebeurde in minder dan 3 uur op een GENERAL ELECTRIC 635 computer. Hiermee werd ook duidelijk dat de vraag of er een projectief vlak van orde 10 bestaat heel geschikt is om met de computer te proberen oplossen. In zijn doctoraatsthesis nam Carter de taak op zich om w_{16} te berekenen met gelijkaardige methodes als die van MacWilliams en haar collega's. Hij bewees dat er zes verschillende startconfiguraties zijn om een codewoord van gewicht 16 te construeren. In 100 uren computertijd kon hij verifiëren dat vier van deze gevallen geen oplossing gaven en een deel van het vijfde geval ook niet. Op dat moment pikten Clemens Lam, Larry Thiel en Stanley Swiercz de draad op en zetten zich aan de taak om w_{12} te berekenen. In 1982 gaf hun computerprogramma na 183 dagen rekenen het resultaat: $w_{12} = 0$.[10] Daarna deden ze Carters werk over en na 80 dagen rekenen op een VAX-11/780 was het resultaat $w_{16} = 0$.

Wat wisten Lam en zijn collega's nu? Er zijn geen codewoorden met gewicht 12, 15 of 16 in de vectorruimte gegenereerd door de rijen van het projectief vlak van orde 10. Ze konden nu ook de gewichtsenumerator berekenen[11] en daaruit vonden ze $w_{19} = 24675$. Als er dus een projectief vlak van orde 10 bestaat, moet het 24675 codewoorden van gewicht 19 bevatten. Dit was de sleutel tot de uiteindelijke oplossing: met dezelfde methode die ze gebruikt hadden voor w_{12}, w_{15} en w_{16} konden ze alle mogelijke configuraties uitproberen. Op die manier konden ze ofwel een projectief vlak van orde 10 construeren ofwel aantonen dat er geen bestaat. In januari 1989 gaf hun computerprogramma dit laatste als resultaat.[12] De zoektocht naar een eindig projectief vlak van orde 10 was dus over: er bestaat geen.[13]

[9] [MacWilliams1973]

[10] Volgens Lam werd Marshall Hall pas op dat moment pessimistisch over het probleem: 'For the first time, I doubt that a plane of order 10 exists.' ([Lam1991] p. 313)

[11] door een resultaat van Mallows en Sloane, [Mallows1974]

[12] [Lam1989]

[13] Hiermee kwam de hoop van Ryser uit die in 1955 schreef: 'For n = 10, there are something like $2^{10,000}$ possible candidates for an incidence matrix. Machines cannot cope with numbers of this magnitude. But the right combination of computation and theory will continue to produce worthwhile results.' ([Ryser1955] p. 30).

3.3. De receptie van het bewijs

Lam vindt zelf dat zijn bewijs geen echt bewijs in de traditionele zin is. De reden daarvoor is volgens hem dat een mens de berekeningen van de computer onmogelijk kan verifiëren. Hij vergelijkt zijn bewijs met het computerbewijs door Appel en Haken van de vierkleurenstelling.[14] Hij noemt zijn bewijs dat er geen eindig projectief vlak van orde 10 bestaat dan ook een 'experimenteel resultaat' dat schreeuwt om een theoretische uitleg.[15] Voor Lam is het dus duidelijk: een computerbewijs is geen bewijs maar een experimenteel resultaat. Ook Marcel Gorissen, die onlangs een methode ontwierp om alle eindige projectieve vlakken van een bepaalde orde te genereren, noemt Lams bewijs geen bewijs.[16]

Als experimenteel resultaat noemt Lam zijn bewijs echter heel betrouwbaar. Hij noemt twee mogelijke fouten: programmeerfouten en fouten in de computerhardware.[17] Om programmeerfouten te ontdekken, gebruiken Lam en zijn collega's twee methodes. Ten eerste gebruiken ze waar mogelijk verschillende programma's en vergelijken de resultaten.[18] Indien mogelijk controleren ze de resultaten ook manueel. Een tweede manier om programmeerfouten te ontdekken die Lam en zijn collega's gebruiken zijn controles op de interne consistentie. Van bepaalde submatrixen kunnen ze bijvoorbeeld schatten hoeveel isomorfe matrixen daarvan het programma genereert. Door deze te tellen, kan je nagaan of het aantal niet te veel afwijkt van het geschatte aantal. Een tweede mogelijke fout is een niet-ontdekte fout in de computerhardware. Op de CRAY-1A computer die ze gebruikten, gebeurt er zo één fout om de duizend uren rekenwerk. Een voorbeeld van zo'n fout is het willekeurig omschakelen van een bit van 0 naar 1 of andersom. Lam en zijn collega's ontdekten zelfs zo'n fout. De kans dat zo'n fout ervoor zorgt dat het programma een projectief vlak van orde 10 over het hoofd ziet is echter infinitesimaal klein. Toch is het niet zo vanzelfsprekend dat het bewijs geen (belangrijke) fouten bevat. Volgens Gorissen is de beschrijving van de computermethodes van Lam en zijn collega's in hun artikel niet voldoende om het bewijs te reconstrueren.[19]

Het bewijs van Lam en zijn collega's heeft binnen de filosofische gemeenschap helemaal

[14][Lam1991] p. 316: 'These are not proofs in the traditional mathematical sense. It is impossible for any human being to check through all the calculations.'

[15][Lam1991] p. 316: 'I want to emphasize that this is only an experimental result and it desperately needs an independent verification, or better still, a theoretical explanation.' en [Lam1989]: 'Because of the use of a computer, one should not consider these results as a "proof", in the traditional sense, that a plane of order 10 does not exist. They are experimental results and there is always a possibility of mistakes.' en 'We hope that someone else will do an independent verification of the results.'

[16][Gorissen2007] p. 14, voetnoot 3: 'This is not a proof in the traditional mathematical sense. It is impossible for any human to check all the calculations. Furthermore, programming mistakes are easily made and untraceable random computing errors are likely to occur during such long computations. '

[17][Lam1989]

[18]Dat past perfect in hun experimentele opvatting over een computerbewijs. De betrouwbaarheid van een experiment kan namelijk verhoogd worden door het onafhankelijk te verifiëren met een ander experiment: 'We find no discrepancy, which gives us faith in the programs.' ([Lam1989])

niet de stormvloed aan reacties losgeweekt die het bewijs door Appel en Haken van de vierkleurenstelling bereikte. Het is zelfs grotendeels genegeerd. Enerzijds kan dit wel te maken hebben met het feit dat het meer dan 10 jaar na het bewijs van de vierkleurenstelling kwam. Anderzijds is de stelling gewoon minder bekend. MacKenzie bespreekt wel kort de kritiek die Lam op zijn eigen bewijs gaf.[20] Voor de rest lijkt het bewijs in de filosofische discussie over het statuut van computerbewijzen genegeerd.

[19][Gorissen2007] p. 14: 'Despite the excellent exposition, the method used in the article gives no clear leads on how to reproduce his findings. The main reason for this is that many computations have been done by different persons spread over many years.'

[20][MacKenzie1999] p. 47

4. Het Robbinsprobleem

4.1. Het probleem en zijn geschiedenis

In 1933 stelde E.V. Huntington de drie volgende axioma's voor als een basis voor een Booleaanse algebra:[1]

1. **Commutativiteit:** `x + y = y + x`
2. **Associativiteit:** `(x + y) + z = x + (y + z)`
3. **Huntington axioma:** `~(~x + y) + ~(~x + ~y) = x`

Hierin staat + voor de of-functie en ~ voor het complement of negatie. Een Booleaanse algebra wordt normaal gedefinieerd in termen van *of*, *en*, *complement*, *0* en *1*. Als we Huntingtons axioma's nemen, kunnen we daaruit bewijzen dat er constanten 0 en 1 bestaan met de bekende eigenschappen.[2] Elke Huntington algebra is dus een Booleaanse algebra. Hierna vroeg Herbert Robbins zich af of we het Huntington axioma kunnen vervangen door het volgende axioma:

Robbinsaxioma: `~(~(x + y) + ~(x + ~y)) = x`

We kunnen eenvoudig nagaan dat de Robbinsvergelijking in elke Booleaanse algebra geldig is. Elke Booleaanse algebra is dus een Robbinsalgebra. Een moeilijkere vraag is nu: vormen de commutativiteits- en associativiteitsaxioma's samen met het Robbinsaxioma een Booleaanse algebra? Of korter: is elke Robbinsalgebra een Booleaanse algebra? Zowel Robbins als Huntington konden hiervoor geen bewijs vinden, noch een tegenvoorbeeld. Robbins speelde het probleem door aan Alfred Tarski en die legde het probleem op zijn beurt aan veel van zijn studenten en collega's voor.[3]

In 1979 hoorde Larry Wos van het Argonne National Laboratory in Illinois van het Robbinsprobleem en ondernam samen met zijn collega's verschillende vruchteloze pogingen om het Robbinsprobleem met behulp van computers op te lossen. Wos suggereerde aan zijn student Steve Winker om naar voldoende voorwaarden te zoeken opdat een Robbinsalgebra een Booleaanse algebra is. Als ze een voldoende voorwaarde zouden vinden en konden aantonen dat elke Robbinsalgebra aan deze voorwaarde voldoet, zouden ze immers het Robbinsprobleem bewezen hebben. Zo is het bijvoorbeeld eenvoudig aan te tonen dat een Robbinsalgebra die aan `~~x = x` voldoet een Booleaanse algebra is.[4] Win-

[1] [Huntington1933] en de correctie [Huntington1933b]

[2] `0 + x = x` voor alle x in de Booleaanse algebra B en `~(~x + x) = 0` voor alle x in B. De eigenschappen van 1 zijn analoog.

[3] [McCune1997]. Stanley Burris, die later het computerbewijs omzette tot een menselijk leesbaar bewijs, zegt dat Tarski hem het probleem voorlegde in de vroege jaren 1970. ([Kolata1996])

[4] Substitueer x in het Robbinsaxioma door ~x, neem het complement van beide kanten van de vergelijking en vereenvoudig ~~x door x. Het resultaat is het Huntington axioma.

ker bekwam door de bewijsprogramma's van Argonne de volgende voorwaarden die elk voldoende zijn om van een Robbinsalgebra een Booleaanse algebra te maken:

1. ∀x (x + x = x)
2. ∃c∀x (c + x = x)
3. ∃c∀x (c + x = c)

Al deze voorwaarden waren echter te sterk: het bleek heel moeilijk om te bewijzen dat elke Robbinsalgebra hieraan voldoet. Winker bewees daarna met de hand verschillende zwakkere voorwaarden. De twee die in het uiteindelijk gevonden computerbewijs belangrijk zijn, zijn de volgende:

Lemma 1: Een Robbinsalgebra die voldoet aan ∃c∃d (c + d = c) is een Booleaanse algebra.

Lemma 2: Een Robbinsalgebra die voldoet aan ∃c∃d (~(c + d) = ~c) is een Booleaanse algebra.

4.2. Het computerbewijs van McCune

In 1984 kwam William McCune bij de groep onderzoekers van Wos. Hij ontwikkelde het computerprogramma EQP ('equational prover') om wiskundige vergelijkingen te bewijzen. Op 10 oktober 1996 vond EQP na acht dagen rekenwerk op een RS/6000 computer een bewijs van het Robbinsprobleem. McCune maakte hierbij dankbaar gebruik van de voldoende voorwaarden die Winker had gevonden, in het bijzonder het hierboven vermelde lemma 1. Het programma EQP vond namelijk een bewijs voor het volgende:

Lemma 3: Alle Robbinsalgebra's voldoen aan ∃c∃d (c + d = c).

Uit lemma 1, lemma 3 en het feit dat de Robbinsvergelijking geldig is in alle Booleaanse algebra's volgt dat alle Robbinsalgebra's Booleaanse algebra's zijn. Het Robbinsprobleem was dus opgelost. [5] Het eerste dat opvalt aan EQP's bewijs van lemma 3 is dat het enorm kort is: 12 regels. Het begint met de Robbinsvergelijking en 12 stappen later heeft het ~(~(3x) + x) + 2x = 2x afgeleid, wat van de vorm is die lemma 1 vraagt: er bestaat een object c, namelijk 2x, en een object d, namelijk ~(~(3x) + x), zodat c + d = c.[6] Het bewijs is zo kort dat we het hier volledig kunnen reproduceren in de vorm waarin EQP het geeft:[7]

[5] [McCune1997] legt uit hoe het Robbinsprobleem werd opgelost.

[6] Dat de uitkomst van de vorm d + c = c is en niet c + d = c, maakt niet uit: we werken namelijk in een commutatieve algebra.

[7] [McCune1997], n(x) staat voor ~(x). Bewijzen door EQP en OTTER van lemma's 1, 2 en 3 en andere gerelateerde stellingen zijn te vinden op de webpagina http://www.cs.unm.edu/~mccune/papers/robbins/jar.html

```
7    n(n(n(x)+y)+n(x+y)) = y           [Robbinsvergelijking]
10   n(n(n(x+y)+n(x)+y)+y) = n(x+y)                  [7->7]
11   n(n(n(n(x)+y)+x+y)+y) = n(n(x)+y)               [7->7]
29   n(n(n(n(x)+y)+x+2y)+n(n(x)+y)) = y             [11->7]
54   n(n(n(n(n(x)+y)+x+2y)+n(n(x)+y)+z)+n(y+z)) = z  [29->7]
217  n(n(n(n(n(x)+y)+x+2y)+n(n(x)+y)+n(y+z)+z)+z) =
                                          n(y+z)    [54->7]
674  n(n(n(n(n(n(x)+y)+x+2y)+n(n(x)+y)+n(y+z)+z)+z+u)+
                                  n(n(y+z)+u)) = u  [217->7]
6736 n(n(n(n(3x)+x)+n(3x))+n(n(n(3x)+x)+5x)) =
                                  n(n(3x)+x)       [10->674]
8855 n(n(n(3x)+x)+5x) = n(3x)         [6736->7,simp:54,flip]
8865 n(n(n(n(3x)+x)+n(3x)+2x)+n(3x)) =
                       n(n(3x)+x)+2x              [8855->7]
8866 n(n(n(3x)+x)+n(3x)) = x               [8855->7,simp:11]
8870 n(n(n(n(3x)+x)+n(3x)+y)+n(x+y)) = y          [8866->7]
8871 n(n(3x)+x)+2x = 2x                [8865,simp:8870,flip]
```

De interactie tussen EQP en de gebruiker is slechts minimaal. De gebruiker geeft zijn vermoeden aan, stelt enkele zoekparameters in (bijvoorbeeld de maximale lengte van de te gebruiken formules in de tussenstappen) en start het programma. Tijdens het zoeken toont EQP welke vergelijkingen het gevonden heeft en een aantal statistieken. Als het programma het vermoeden niet kan bewijzen of als de gebruiker de zoektocht niet in de juiste richting ziet gaan, past de gebruiker de parameters aan en start het programma opnieuw. Op deze manier werkte McCune vijf weken lang aan het sturen van EQP in de juiste richting.[8]

Na het bewijs liet McCune EQP verder zoeken. Ten eerste wou hij zien hoe hij het bewijs met andere parameters sneller kon vinden. Ten tweede wou hij een eenvoudiger bewijs vinden. En ten derde wou hij bewijzen van de andere voldoende voorwaarden vinden. Het belangrijkste resultaat hiervan was dat hij de Huntington vergelijking rechtstreeks uit de Robbinsvergelijking kon bewijzen, zonder zich op één van de lemma's van Winker te baseren. Het kleinste bewijs dat een Robbinsalgebra aan de lemma 2 voorwaarde voldoet bestaat uit 8 stappen, het kleinste bewijs dat een Robbinsalgebra aan de lemma 1 voorwaarde voldoet (dus een bewijs van lemma 3) bestaat uit 12 stappen en het kleinste bewijs dat een Robbinsalgebra aan de Huntington vergelijking voldoet heeft lengte 86. McCune vermeldt dat de stappen in de bewijzen van lengte 8 en 12 echter heel ingewikkeld zijn, terwijl het langere bewijs van de Huntington vergelijking uit relatief eenvoudige stappen bestaat. EQP vond trouwens ook bewijzen van Winkers lemma 1 en 2.

[8] McCune legt dit proces, dat hij ‘well-behaved search’ noemt, uit in [McCune1997b].

4.3. Logisch redeneren

Hoe vindt het programma EQP welke stappen het moet uitvoeren? Het programma maakt zoals bijna alle automatische bewijsprogramma's gebruik van *resolutie* en *unificatie*. Resolutie is een veralgemening van de *modus ponens* inferentieregel. Met de modus ponens regel kunnen we uit p en `~p | q` het volgende afleiden: q. In zijn eenvoudigste vorm zegt de resolutieregel: heb je `p | r` en `~p | q`, leid dan `r | q` af. Algemener kunnen we in plaats van r en q verschillende proposities hebben. Zo kunnen we uit `p | r | s` en `~p | q | t` de uitdrukking `r | s | q |t` afleiden. We kunnen resolutie beschouwen als het ‘annuleren’ van p tegenover ~p. Als we p en ~p afgeleid hebben, leidt resolutie tot een ‘lege zin’, die een contradictie voorstelt.[9]

Naast resolutie is er nog een inferentieregel nodig, namelijk *unificatie*. Alan Robinson publiceerde het unificatie-algoritme in zijn ondertussen klassieke artikel ‘A machine-oriented logic based on the resolution principle’ (1965)[10], maar de techniek was al ‘in the air’ vanaf 1962.[11] Met het unificatiealgoritme kan je waardes voor variabelen vinden om twee termen overeen te laten komen. Een voorbeeld: als we `f(x, g(x))` en `f(g(c), z)` unificeren, krijgen we `x = g(c)` en `z = g(g(c))`. De overeenkomende variabelen worden dus als het ware aan elkaar gelijkgesteld: het unificatiealgoritme geeft als uitvoer de substitutie die de variabelen aan elkaar gelijkstelt.

De volledige kracht van beide inferentieregels wordt pas duidelijk als je ze combineert. Robinson bewees in zijn artikel dat de combinatie van resolutie en unificatie *refutation complete* is: als we een lijst van zinnen geven die samen een contradictie vormen, dan bestaat er een bewijs door middel van resolutie en unificatie van de lege zin beginnend van de originele lijst. Resolutie zonder unificatie is eigenlijk niet nuttig en daarom wordt de combinatie van resolutie en unificatie vaak kortweg ‘resolutie’ genoemd. Dit was de start van een nieuw paradigma van automatische bewijsvoering, waarop nog bijna alle huidige automatische bewijsprogramma's gebaseerd zijn. We starten met de axioma's van de theorie waarin we werken, samen met de negatie van wat we willen bewijzen. We voeren resolutie uit tot het programma een contradictie vindt of moet opgeven na lang zoeken. Wanneer het een contradictie gevonden heeft, betekent het dat de te bewijzen stelling uit de axioma's volgt, aangezien de negatie van de stelling voor de contradictie zorgde. De verschillende bewijsprogramma's hebben boven deze inferentieregel methodes ontwikkeld om de verschillende manieren om de inferentieregel toe te passen te leiden naar de ‘juiste’ paden. Vaak is dit heel moeilijk algemeen te leiden en programma's als EQP bieden dan ook een heleboel parameters aan die de gebruiker kan instellen om de zoektocht op bepaalde manieren te leiden.

[9] [Beeson2003] p. 16
[10] [Robinson1965]
[11] [Beeson2003] p. 16-17

4.4. De receptie van het bewijs

Het computerbewijs van McCune is geen *computergeassisteerd bewijs* zoals de bewijzen van de vierkleurenstelling en het niet-bestaan van een projectief vlak van orde 10. Het computergeassisteerd bewijs van de vierkleurenstelling ging slechts exhaustief allerlei mogelijkheden in een beperkt domein na, binnen het bewijsschema dat Appel en Haken hadden opgesteld. Zij hadden het computerprogramma daarvoor specifiek geschreven. EQP daarentegen is een algemeen 'redeneerprogramma', een *automatisch* bewijsprogramma dat zelf zoekt naar een bewijs. Het bewijs van het Robbinsvermoeden werd in 1997 in *AI Magazine* dan ook samen met het schaakprogramma Deep Blue en autonoom rijdende voertuigen als één van de vijf belangrijkste verwezenlijkingen in kunstmatige intelligentie genoemd.[12]

McCune gebruikte een algemeen redeneerprogramma, EQP. Het enige dat hij moest ingeven specifiek voor dit probleem was de vorm van de vergelijking die EQP moest bewijzen, de axioma's die het daarvoor kon gebruiken en een aantal parameters om de zoektocht te richten. De kracht van bewijsprogramma's als EQP werd nog aangetoond doordat McCunes gelijkaardige programma OTTER andere (minder bekende) resultaten kon bewijzen: zo vond het bijvoorbeeld een kort axioma dat een basis voor Booleaanse algebra is[13] of het kortste formele bewijs van bepaalde stellingen.[14] Volgens David Corfield zijn dit soort bewijzen de grootste successen van OTTER en andere automatische bewijsprogramma's, omdat dit soort problemen heel moeilijk zijn voor mensen.[15] Computerprogramma's moeten gewoon lang genoeg zoeken en kunnen dat ook.

De kritiek dat dit zoeken 'maar wat rekenwerk is' is, spreekt Louis Kauffman tegen. Hij noemt het bewijs van EQP een intelligente toepassing van patroonherkenning, wat dicht ligt bij de activiteit van een menselijke wiskundige.[16] Dat was ook de mening van Larry Wos, het hoofd van het computer reasoning project in Argonne National Laboratory. Volgens Wos was McCunes computerbewijs een grote stap voorwaarts en zou het het begin van een toekomst zijn waarin wiskundigen slechts nieuwe vermoedens ontdekken en de bewijzen aan de computers overlaten.[17] Wiskundige Stanley Burris noemde het bewijs

[12][Martin1999]

[13][McCune2002], zie ook [Mann2003] voor een reconstructie van het bewijs.

[14][Thiele2002]. De vraag wat het kortste formele bewijs van een bepaalde stelling is, staat bekend als 'Hilberts vierentwintigste probleem'. Dit is pas in het midden van de jaren 1990 gevonden in een notaboekje van Hilbert, bijna een eeuw na de formulering van zijn 23 problemen.

[15][Corfield2003] p. 50: 'A considerable portion of the automated theorem prover's successes have been in establishing that a smaller number of axioms can form the basis of an algebra than had been thought.'

[16][Kauffman2001]: 'EQP's proof is much more than a calculation. The proof depends upon a successful search among a realm of possibilities and the skillful application of pattern recognition and the application of axioms. This is very close to the work of a human mathematician.'

[17][Kolata1996]: Kolata citeert Wos die McCunes bewijs een 'quantum leap forward' noemt en schrijft verder: 'Wos predicts that the result may mark the beginning of the end for mathematics research as it is now practiced, eventually freeing mathematicians to focus on discovering new conjectures, and leaving the proof to computers.'

een doorbraak in computerbewijzen en stelde dat dit toonde dat de grens tussen rekenwerk en creativiteit dun en misschien zelfs artificieel is.[18] Computerwetenschapper Robert Boyer ziet de computer voor dit soort bewijzen dan weer als een collega, ook al is hij niet altijd even behulpzaam.[19] Ook de kritiek dat het computerbewijs niet het inzicht geeft dat een wiskundige van een ander bewijs krijgt, weerlegt Kauffman: hij genoot even goed van EQP's bewijs als van 'menselijke' bewijzen.[20]

4.4.1. Vertalingen van het bewijs

McCune lijkt zelf helemaal geen interesse te hebben in de filosofische implicaties van zijn werk.[21] Hij deed echter wel heel wat moeite om aan te tonen dat het bewijs van EQP correct is. Hij heeft het bewijs met de hand nagekeken en het bewijs is ook door een ander programma, OTTER, geverifieerd. Op de website van McCune staat de uitvoer van EQP en OTTER, zodat iedereen het bewijs kan controleren. Zo heeft Mark Stickel van het Stanford Research Institute in Palo Alto het bewijs ook met de computer geverifieerd en de wiskundige Stanley Burris met de hand.[22] Nadien hebben verschillende personen het bewijs nog vereenvoudigd.[23]

Elke stap van het EQP-bewijs bestaat uit formules met een groot aantal haakjes, wat het heel moeilijk maakt voor mensen om na te kijken. De volgende gelijkheid is bijvoorbeeld één regel van het bewijs:

```
~(~(~(~(~(x)+x)+~(~(x)+x)+x+x+x+x)+~(~(~(x)+x)+x+x+x)+x)+x)
=~(~(~(x)+x)+~(~(x)+x)+x+x+x+x)
```

Volgens Burris was de computeruitvoer vrij onleesbaar. EQP gaf wel aan bij elke stap uit welke andere stappen dit volgde, maar de details moest Burris zelf invullen om het bewijs te begrijpen.[24] Hij zette het bewijs om naar eenvoudigere stappen die gemakkelijker te begrijpen zijn voor een mens.[25] Ook Branden Fitelson noemt het bewijs ingewikkeld om te volgen en hij suggereert dat dit komt omdat het bewijs niet *conceptueel* is.[26] Verschillende stappen van het bewijs bestaan uit moeilijke substituties.[27] Het EQP-bewijs

[18][Kolata1996]: '[It shows that] it's a very thin line between the mechanical and the creative and it may disappear.'

[19][Kolata1996]: 'It's best to think of a computer as just another colleague, one that is sometimes helpful, but often not.'

[20][Kauffman2001]: 'I understood EQP's proof with an enjoyment that was very much the same as the enjoyment that I get from a proof produced by a human being.'

[21][Kolata1996]: 'I just work on the problems and try to solve them.'

[22][Kolata1996]

[23]Bijvoorbeeld [Dahn1997]. Allen Mann heeft EQP's bewijs ook vertaald en gecontroleerd in zijn thesis [Mann2003].

[24][Kolata1996]: 'It was pretty unreadable. The machine says "I got that step from two steps before", but it doesn't fill in all the details.'

[25][Burris1996]. Burris zegt over zijn bewijs in [Peterson1997]: 'I ended up with a lot of little equations. You could easily sit on a bus and go through the hundred or so steps of the proof.'

zou mensen zelfs niet helpen om het bewijs te *begrijpen*.[28]

Kauffman stelde zich de vraag of mensen het bewijs wel kunnen volgen.[29] Hij beantwoordde deze vraag bevestigend, al is er volgens hem een vertaling van de notatie nodig. Zo heeft hij het bewijs van EQP vertaald in een notatie met vakjes, die door zijn tweedimensionale karakter veel eenvoudiger te volgen is.[30] Dat lijkt ook de mening van Fitelson, die aantoont hoe je met behulp van het computerprogramma MATHEMATICA het bewijs van EQP kan begrijpen. Eén van de redenen die hij opsomt om daarvoor MATHEMATICA te gebruiken, is namelijk dat dat programma de complement operator voorstelt als een lijn boven een expressie, wat wij mensen gemakkelijker kunnen volgen dan geneste haakjes.[31] Ook Corfield is het met Kauffman en Fitelson eens dat we vertalingen nodig hebben om bewijzen zoals dat van het Robbinsprobleem begrijpelijker te maken voor wiskundigen.[32]

Volgens Corfield wordt het aannemen van computerbewijzen zoals dat van het Robbinsprobleem tegengehouden doordat mens en computer een verschillende 'taal' spreken. De computer spuit symbolen uit, die de mens niet begrijpt.[33] Wat we nodig hebben, zegt Corfield, is een gemeenschappelijke taal voor mens en machine om wiskunde voor te stellen. Hij verwijst hier naar Peter Galison die in zijn boek *Image and Logic* (1997) praat over *pidgins* om communicatie tussen verschillende gemeenschappen van onderzoekers te vergemakkelijken.[34] Volgens Galison hebben in de wetenschappelijke praktijk bijvoorbeeld de experimentele fysici, theoretische fysici en de ingenieurs die instrumenten ontwikkelen manieren gevonden om met elkaar te communiceren zonder elkaars taal volledig te moeten spreken. Volgens Corfield hebben we ook een pidgin nodig zodat wiskundigen en automatische bewijsprogramma's met elkaar kunnen communiceren en hij zegt

[26][Peterson1997]: 'It's a very complicated proof, not at all elegant or conceptual. It's a perfect example of the type of proof that a machine can find but we can't because of its complexity.'

[27][Fitelson1998]: 'The proof found by EQP is quite complex and difficult to follow. Some of the steps of the EQP proof require highly complex and unintuitive substitution strategies. As a result, it is nearly impossible to reconstruct or verify the computer proof of the Robbins conjecture entirely by hand.'

[28][Fitelson1998]: 'The EQP proof object does little to help human beings *understand how to prove* the Robbins conjecture. In fact, the EQP proof object is quite difficult to follow (or even parse!).'

[29][Kauffman2001]

[30][Kauffman2001]: 'It is very difficult for a human being to keep track of nested parentheses, but not too hard to look at patterns of nested boxes. Accordingly I translated the steps in the proof into a nested box notation.', 'The eye is immediately met with the regions delineated by the boxes and hence by the corresponding enclosures of parentheses.' Deze notatie had Kauffman al voorgesteld in [Kauffman1990].

[31][Fitelson1998]: 'The keys to my success were (1) MATHEMATICA's two-dimensional Boolean notation for the complementation operator which allowed me to see many complex syntactical patterns that were all but invisible in EQP's one-dimensional notation, and (2) MATHEMATICA's powerful, interactive symbolic engine, which allowed me to experiment with many complicated substitution strategies in "real time".'

[32][Corfield2003] p. 51: 'To facilitate this process will probably require translation procedures to more humanly comprehensible forms.'

[33][Corfield2003] p. 55: 'The desire on the part of mathematicians for an improved understanding of a field of research discourages interest in devices generating lines of incomprehensible symbols.'

[34][Corfield2003] p. 56

dat Kauffmans werk hier een begin van is.[35]

4.4.2. De waarde van het bewijs

Corfield vindt dat het bewijs van EQP ons wel degelijk meer inzicht geeft in Booleaanse algebra, we weten namelijk iets meer erover. Bovendien kunnen we OTTER en ander bewijsprogramma's gebruiken als we iets willen weten over een bepaalde algebra en er echt geen intuïties over hebben.[36] Ook al begrijpen we het bewijs niet helemaal omdat het niet conceptueel is, het resultaat van het bewijs is op zich al belangrijk: we weten dat de stelling waar is.[37]

Als een computeralgebrasysteem een bepaalde complexe berekening uitvoert die mensen niet kunnen volgen, zijn we volgens Corfield enkel geïnteresseerd in het resultaat. Als we de feitelijke berekeningen zouden bestuderen, welke stappen het systeem achtereenvolgens doet, zouden we hier niets nieuws uit leren.[38] De stappen in EQP's bewijs betekenen volgens Corfield echter wel degelijk iets. Het feit dat het programma juist met die stappen succes bereikt, is iets waar we volgens hem heel wat uit kunnen leren over het probleem.[39] Ik heb echter tijdens mijn literatuurstudie nog geen concreet niet-triviaal inzicht gevonden dat een studie van EQP's bewijs van het Robbinsprobleem opleverde.

Kauffman stelt zichzelf naar aanleiding van het Robbinsprobleem de vraag of een computer wel een wiskundig bewijs kan leveren. Een bewijs moet volgens hem allereerst overtuigend zijn *voor een wiskundige*. Hij neemt aan dat een argument dat overtuigend is voor een wiskundige per definitie een wiskundig bewijs is.[40] Een computer kan volgens Kauffman enkel de stappen geven die nodig zijn voor het resultaat, maar is zelf niet overtuigd van het bewijs.[41] Het is me niet heel duidelijk wat Kauffman hiermee bedoelt en waarom dat zijn stelling verdedigt dat een bewijs pas een bewijs is als het overtuigend is voor een wiskundige. Eerder zei Kauffman dat een bewijs overtuigend moet zijn voor *een wiskundige*, maar voor een computerbewijs lijkt hij te eisen dat de computer *zelf* moet

[35][Corfield2003] p. 56: ‘Kauffman is encouraging us to encode our concepts in a form acceptable to computers, and then to learn to translate from their languages to ones accessible to us.’

[36][Corfield2003] p. 51: ‘We can anticipate that mathematicians and scientists arriving at a situation where they would like to know more about a particular algebra but have little intuition about it will turn to automated provers such as OTTER and EQP for assistance.’

[37][Corfield2003] p. 51-52: ‘Certainly, when, as has happened, a computer's proof or disproof of a proposition prompts a mathematician to produce their own proof or counter-example just because she knows which is the correct thing to do, then we can say that the machine has played an important role.’

[38][Corfield2003] p. 52: ‘No selection has occurred in the application of its algorithms.’

[39][Corfield2003] p. 52: ‘EQP's proof does involve the selection of a combination of inference steps from an enormous space of such combinations. Without needing to credit the machine with intelligence, there is still something to learn from devices which manage to find one or more successful paths from a very large space of possibilities.’

[40][Kauffman2001]: ‘A proof is not a proof until a person is convinced by it. In fact a mathematical proof is exactly an argument that is completely convincing to a mathematician!’

[41][Kauffman2001]: ‘In this sense, a computer does not, can not produce a proof. The computer is not convinced of anything. […] It does not know the proof. It only finds the steps.’

overtuigd zijn van een bewijs. Dat lijkt mij niet correct. Na zijn onduidelijk intermezzo komt Kauffman echter terug to-the-point met zijn conclusie: enkel wanneer mensen het resultaat van een computer positief beoordelen, is het een bewijs.[42]

Het is opvallend dat veel van deze mensen die commentaar geven op het bewijs van het Robbinsprobleem wiskundigen en computerwetenschappers zijn. In de filosofische discussie over het statuut van computerbewijzen lijkt EQP's bewijs van het Robbinsprobleem grotendeels genegeerd. Fitelsons artikel is vooral een wiskundig artikel, maar hij vermeldt wel de filosofische relevantie van het probleem. Pas in 2003 zien we een uitgebreide filosofische bespreking van het bewijs in David Corfields *Towards a philosophy of real mathematics*.[43] Anderen is de discussie, bijvoorbeeld over de inzichtelijkheid van het resultaat en de rol van de notatie hierin, dan weer volledig ontgaan.[44]

[42][Kauffman2001]: 'It is a human judgment that propels the result of the computer's search into a statement that the computer has "found a proof". Indeed the computer has helped us and it has found something that we can examine. If we judge that to be a proof, then it is a proof (for us).'

[43][Corfield2003] hoofdstuk 2: 'Communicating with automated theorem provers', vooral p. 48-55

[44]Zo schrijft Michael Beeson in zijn voor de rest excellente overzichtsartikel 'The mechanization of mathematics': 'Since the proof was easily checkable by humans, there was no flurry of discussion about the acceptability of the proof, as there had been about the four-color problem.' ([Beeson2003]). Hij is blijkbaar niet op de hoogte van de problemen die Fitelson, Kauffman en anderen hebben aangeduid om het bewijs te begrijpen of zelfs maar te verifiëren. Het verifiëren was niet onoverkomelijk, maar zeker niet gemakkelijk.

5. Het Keplervermoeden

5.1. Het probleem en zijn geschiedenis

In de jaren 1590 was Sir Walter Raleigh zijn schip aan het klaarmaken voor een expeditie en hij vroeg zich af hoeveel kanonskogels er in een bolstapeling van een bepaalde hoogte konden. Hij gebruikte de normale piramidevormige stapeling van kanonskogels. Zijn assistent Thomas Harriot vond de vergelijking om deze hoogte te berekenen. Jaren later vermeldde Harriot het probleem aan de astronoom Johannes Kepler. Die uitte in 1611 in zijn werk *Strena sue de nive sexangula* (Over de zeshoekige sneeuwvlok) het vermoeden dat de piramidevormige opstapeling van kanonskogels de efficiëntste manier was om de ruimte te gebruiken, maar hij had er geen bewijs voor. Zijn vermoeden werd bekend als het *Keplervermoeden*.[1]

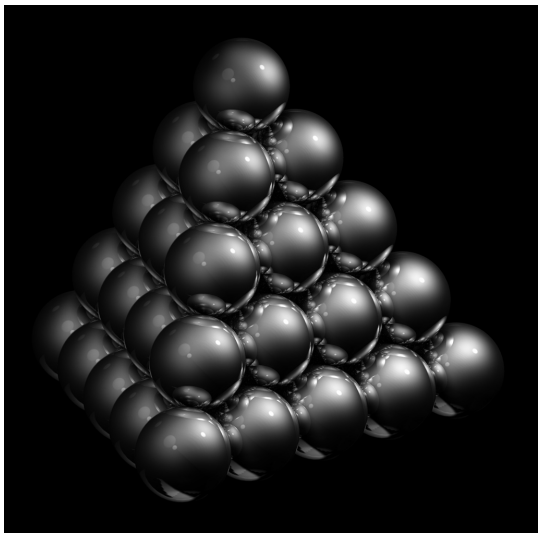
De piramidevormige opstapeling van kanonskogels wordt in wiskundige termen het *vlak gecentreerde kubische rooster* ('face-centered cubic packing') genoemd. Je hebt een aantal bollen met straal 1 in een Euclidische ruimte die elkaar niet overlappen. De afstanden tussen de middelpunten van de bollen zijn dus groter dan of gelijk aan 2. Het Keplervermoeden zegt dat er geen stapeling van identieke bollen bestaat waarvan de densiteit groter is dan het vlak gecentreerde kubische rooster.[2] De densiteit wordt gedefinieerd als het percentage van het volledige volume dat ingenomen wordt door de bollen. Bij de piramidevormige stapeling is dit ongeveer 74%.[3] Hoe groter de densiteit, hoe efficiënter je de beschikbare ruimte gebruikt. Daar gaat het natuurlijk om bij het stapelen van kanonskogels of appelsienen.

[1] Een korte geschiedenis van het probleem en de pogingen tot bewijs is te vinden in [Hales2006].

[2] Er zijn wel andere stapelingen van bollen die *dezelfde* densiteit hebben, zoals de *honingraatstructuur*.

[3] $\pi/\sqrt{18}$ of ongeveer 0,74048.

Figuur 5.1. Het vlak gecentreerde kubische rooster



Carl Friedrich Gauss bewees in 1831 dat het Keplervermoeden juist is voor bollen in een regelmatig rooster. Zijn bewijs is slechts enkele lijnen lang en heeft geen berekeningen nodig.[4] Dit betekende dus dat een tegenvoorbeeld voor Keplers vermoeden in een onregelmatige stapeling moet liggen. Hiermee was duidelijk hoe een bewijs er in grote lijnen kon uitzien: bewijs voor alle mogelijke onregelmatige stapelingen dat hun densiteit groter is dan de densiteit van de piramidevormige stapeling. Helaas leek dit onbegonnen werk, aangezien het aantal mogelijke stapelingen oneindig is. Toen er in 1900 nog geen vooruitgang was gemaakt, nam Hilbert het vermoeden op in zijn bekende lijst van 23 onopgeloste problemen: het Keplervermoeden vormde een deel van zijn achtiende probleem.

In het midden van de twintigste eeuw kwam er terug schot in de zaak. De Hongaarse wiskundige László Fejes Tóth bewees in 1953 dat het bepalen van de maximum densiteit van alle stapelingen kon gereduceerd worden tot een eindig aantal berekeningen, ook al is dat aantal heel groot en zijn de berekeningen complex. Nu was men dus zeker dat een bewijs door een case-by-case analyse in principe mogelijk was en Fejes Tóth stelde een strategie voor om het vermoeden te bewijzen. De volgende belangrijke stap werd eveneens door de Hongaar gezet: in 1964 suggereerde hij als eerste dat het probleem met een computer kon opgelost worden. Hij had namelijk een aanpak gevonden die het probleem reduceert tot een optimalisatieprobleem van een functie van een eindig aantal variabelen. Het minimum van dit optimalisatieprobleem kon volgens hem in principe benaderd worden door computerberekeningen.[5]

[4] [Hales2000] p. 442. Dit staat in schril contrast met het bewijs van Hales voor het algemene geval.

[5] Hales citeert Fejes Tóth in [Hales2002b] p. 798: ‘Thus it seems that the problem can be reduced to the determination of the minimum of a function of a finite number of variables, providing a programme realizable in principle. In view of the intricacy of this function we are far from attempting to determine the exact minimum. But, mindful of the rapid development of our computers, it is imaginable that the minimum may be approximated with great exactitude.’

Anderen probeerden het bewijs met louter theoretische wiskunde op te lossen. De strategie bestond erin om een bovengrens te vinden voor de maximumdensiteit van elke mogelijke stapeling van bollen. De bovengrens die men kon bewijzen werd altijd kleiner, van 0,884 tot 0,7731. Men kwam echter niet aan de densiteit van 0,74 van de piramidevormige stapeling.[6]

Net als bij de vierkleurenstelling is de geschiedenis van het Keplervermoeden ook geplaagd door incorrecte bewijzen. In 1990 beweerde Wu-Yi Hsiang van de University of California in Berkeley dat hij een bewijs had van het Keplervermoeden. In 1993 werd zijn bewijs gepubliceerd in de *International Journal of Mathematics*[7], maar al vlug na de publicatie werden er fouten en belangrijke gaten gevonden in het bewijs. Volgens George Szpiro twijfelen een aantal mensen aan de manier waarop het nakijken van het bewijs gebeurd is, aangezien de editors van het *International Journal of Mathematics* de Berkeley-collega's van Hsiang zijn.[8] Sindsdien sprak Hsiang over het artikel als slechts een ‘outline’, geen volledig bewijs. Pas in 2002 vulde hij de gaten op, maar hij publiceerde het bewijs niet in een peer-reviewed tijdschrift, maar in een boek. Dat gaf heel wat wiskundigen twijfels over de geldigheid van het bewijs. Er zijn geen fouten gevonden in dit nieuwe bewijs van Hsiang, maar de algemene opinie is dat zijn bewijs onvolledig is.[9]

[6][Hales2006] p. 9

[7][Hsiang1993]

[8][Szpiro2003b] p. 150. Simon Singh citeert bovendien in [Singh1998] p. 321 de wiskundige Doug Muder: ‘Het stuk van Hsiang was onvoldoende getoetst, als het al getoetst was. Het feit dat het *Journal* wordt geredigeerd door Hsiangs collega's van Berkeley geeft het verhaal een tintje van vriendjespolitiek. Het *Journal* had tot aan dit stuk geen belangstelling voor bolstapelen. Het lijkt wel duidelijk dat Hsiang voor het *International Journal* koos omdat het door zijn vrienden werd geredigeerd, niet omdat het de aangewezen plaats was voor zijn stuk.’ Singh bespreekt de onfrisse affaire met Hsiang uitgebreid op pp. 319-322.

[9][Chang2004] citeert Frank Quinn: ‘Hsiang has not such a good track record. I don't want to spend time proving it's wrong.’. Ook Hales, die uiteraard geen neutrale partij is, citeert uitspraken van collega's die het bewijs van Hsiang afwijzen: [Hales2006] p. 12. Zo noemen Conway en Sloane het een bewijs met heel wat onvolkomenheden.

5.2. Het computerbewijs van Hales

Thomas Hales vond de suggestie van Fejes Tóth om er computers op los te laten veelbelovend. Hij bewees dat de maximumdensiteit van alle mogelijke stapelingen van bollen kon gevonden worden door een functie van 150 variabelen te minimaliseren. Op 9 augustus 1998 kondigde Hales aan dat hij samen met zijn doctoraatsstudent Samuel Ferguson het Keplervermoeden had opgelost.[10] Het volledige bewijs was verspreid over een reeks artikels met in totaal 250 pagina's. In tegenstelling tot vroegere bewijspogingen, maakte hij gebruik van een computer in een groot deel van het bewijs. De computerbestanden met de programma's en de gegevens hadden in totaal 3 Gbyte schijfruimte nodig.

Hales legt de structuur van zijn bewijs uit in [Hales2006b], pp. 23-30. Ook de introductie van [Hales2006e] is een goede uitleg van de structuur van het bewijs en wat de rol van de computer daarin is. Hales bewijst in [Hales2006e] eerst dat elke graaf die geassocieerd is met een tegenvoorbeeld van het Keplervermoeden een aantal eigenschappen heeft. Hij noemt een graaf met deze eigenschappen een *tamme graaf*.[11] Hales heeft met behulp van een computer ook een classificatie uitgevoerd van alle tamme grafen, dit zijn er 5128.[12] In het tweede deel van het artikel bewijst Hales dat een tamme graaf, met uitzondering van drie exemplaren, geen tegenvoorbeeld van het Keplervermoeden kan zijn. Enkel een graaf die isomorf is met één van deze drie grafen kan dus op een tegenvoorbeeld van het Keplervermoeden wijzen. Dat dit geen tegenvoorbeelden kunnen zijn, werd bewezen in [Ferguson2006] en sectie 8 van [Hales2006c]. Dit maakt het bewijs van het Keplervermoeden volledig.

De computer was een essentieel onderdeel van het bewijs van Hales en niet alleen in het verifiëren van een case-by-case analyse.[13] Heel wat bewijzen verwijzen enkel naar een computerberekening.[14] In [Hales2006c] wordt er in totaal 42 keer verwezen naar een computerberekening, in [Hales2006d] in totaal 123 keer, in [Hales2006e] totaal 71 keer en zelfs in het *Annals*-artikel [Hales2005] verwijst hij 40 keer naar een computerberekening.[15] Hales vermeldt de volgende manieren waarop de computer gebruikt werd voor zijn bewijs:[16]

1. **Bewijzen van ongelijkheden met intervalrekenkunde.** Een methode om met de

[10] http://www.math.pitt.edu/~thales/kepler98/announce

[11] De definitie van een tamme graaf zou ons te ver leiden en is te vinden in [Hales2006e] p. 207.

[12] Op de webpagina http://annals.math.princeton.edu/keplerconjecture/ is het programma te vinden om al deze grafen te genereren.

[13] [Hales2006] p. 16: 'As this project has progressed, the computer has replaced conventional mathematical arguments more and more, until now nearly every aspect of the proof relies on computer verifications. Many assertions in these papers are results of computer calculations.'

[14] Bijvoorbeeld het bewijs van lemma 6.10 in [Hales2005] p. 1119 gaat als volgt: 'Proof. This is CALC-586468779.'

[15] Deze tellingen gebeurden door mij door een zoekfunctie op de PDF-bestanden van de artikels.

[16] [Hales2006] p. 17

computer verschillende ongelijkheden in een klein aantal variabelen te bewijzen, gebruik makend van intervalrekenkunde.

2. **Combinatorics.** Een computerprogramma classificeert alle vlakke grafen die relevant zijn voor het Keplervermoeden.
3. **Grenzen voor lineair programmeren.** Vele niet-lineaire optimalisatieproblemen die voor het bewijs nodig zijn worden vervangen door linaire problemen die bovengrenzen vormen. Deze worden opgelost door computermethodes voor lineair programmeren. Elk probleem heeft tussen de 100 en 200 variabelen en tussen de 1000 en 2000 beperkingen. Het hele bewijs bestaat uit zo'n 100 000 van deze problemen.
4. **Branch-and-bound methodes.** De methodes voor lineair programmeren worden gecombineerd met branch-and-bound methodes wanneer de gevonden grenzen niet voldoende zijn.
5. **Numerieke optimalisatie.** Hales heeft tijdens de zoektocht naar het bewijs heel wat gebruik gemaakt van software voor symbolische wiskunde en niet-lineaire optimalisatie.

Het bewijs van Hales werd met gemengde gevoelens onthaald. De problemen begonnen al bij het verifiëren. Robert MacPherson, editor van het prestigieuze tijdschrift *The Annals of Mathematics*, spoorde Hales aan om zijn bewijs in te dienen, maar MacPherson had zijn twijfels bij het computergedeelte van het bewijs. De referees wilden het bewijs uiteraard volledig nakijken, maar dit bleek al vlug onmogelijk. In september 1998 werd een eerste groep van twaalf referees aangesteld.[17] Ze hielden een conferentie in Princeton om de strategie te bespreken. De referees hadden verschillende jaren nodig voor hun werk. Ze vonden geen fouten, maar hadden zichzelf uitgeput door het verifiëren van de vele berekeningen. Het volstond namelijk niet om de berekeningen over te doen, ze moesten nakijken of de programma's van Hales wel berekenden wat hij in zijn artikel beweerde. De referees voerden dus consistentiecontroles uit, reconstrueerden de gedachten achter de stappen van het bewijs en onderzochten de vooronderstellingen van de computerprogramma's.[18] In 2003 zei het hoofd van het team referees, Gábor Fejes Tóth (zoon van László Fejes Tóth), dat ze 99% zeker waren van de correctheid van het bewijs, maar dat ze de correctheid van alle computerberekeningen niet konden garanderen omdat ze niet elke lijn computercode konden nakijken. Robert MacPherson, één van de editors van het tijdschrift, vergeleek het nakijken van het bewijs met het proeflezen van de nummers in een telefoonboek.[19]

De referees waren uitgeput van het controleren van het bewijs, waarna MacPherson Hales een brief stuurde met het slechte nieuws.[20] MacPherson liet ook zijn irritatie door-

[17]Dit aantal is uitzonderlijk hoog. Voor de meeste wiskundige artikels worden er één tot drie referees aangesteld.
[18][Szpiro2003] p. 12-13
[19][Chang2004]

schijnen om het feit dat Hales zijn bewijs in vijf verschillende artikels had afgeleverd die elk lichtjes verschillende notaties gebruikten en in een ‘ongewone’ stijl geschreven waren, alsof het om verslagen van laboratoriumexperimenten ging.[21]

Ook de publicatie van Hales' bewijs zorgde voor de nodige problemen. *The Annals of Mathematics* wilde het bewijs eerst enkel publiceren met een waarschuwing dat het niet volledig was nagekeken. Hierop kwam echter kritiek van verschillende wiskundigen. Conway vond bijvoorbeeld dat het niet eerlijk was tegenover Hales. De editors lieten het bewijs daarop nog eens nakijken door een andere referee, die het theoretische gedeelte als correct beoordeelde. Daarop besloten de editors van *The Annals of Mathematics* om Hales' bewijs in twee te splitsen: ze publiceerden enkel het theoretische gedeelte van het bewijs[22], dat op de traditionele manier door referees was nagekeken. Dit deel beschouwden ze als een correct en zelfs hoogstaand bewijs.[23] Voor de publicatie van het computergedeelte moest Hales zijn toevlucht zoeken tot het gespecialiseerde tijdschrift *Discrete and Computational Geometry*. Het julinummer van 2006 werd er volledig aan gewijd.[24]

5.3. Naar een formeel bewijs van het Keplervermoeden

De uitspraak dat zijn bewijs slechts voor 99% zeker was, viel niet in goede aarde bij Hales. Hierop besloot hij in het begin van 2003 om elke stap van zijn bewijs automatisch te laten verifiëren door computers.[25] Het Flyspeck-project was geboren: een volledige formele verificatie van het Keplervermoeden.[26] De naam *Flyspeck* is afgeleid van FPK, de afkorting voor ‘Formal Proof of Kepler’.[27] Volgens Hales zal het 20 werkjaren duren

[20]Szpiro citeert uit de brief in [Szpiro2003] p. 13: ‘The news from the referees is bad, from my perspective. They have not been able to certify the correctness of the proof, and will not be able to certify it in the future, because they have run out of energy to devote to the problem. This is not what I had hoped for.’

[21][Szpiro2003] p. 13, uit dezelfde brief: ‘One can speculate whether their process would have converged to a definitive answer had they had a more clear manuscript from the beginning, but this does not matter now.’

[22][Hales2005]

[23][Chang2004] citeert MacPherson: ‘The part that's going in *The Annals of Mathematics* is a proof. We feel he made a serious contribution to mathematics.’

[24][Hales2006], [Hales2006b], [Hales2006c], [Hales2006d], [Ferguson2006] (herziene versie van Fergusons doctoraatsthesis) en [Hales2006e]

[25]Te vergelijken met wat Georges Gonthier in [Gonthier2004] gedaan heeft voor het vierkleurenprobleem.

[26][Hales2006f]

[27]De stand van zaken van het project is te vinden op *The Flyspeck Project Fact Sheet* op het internetadres http://www.math.pitt.edu/~thales/flyspeck/

voordat het volledige bewijs geformaliseerd en geverifieerd is, een schatting die hij baseert op Freek Wiedijks schatting dat het één week duurt om één pagina uit een tekstboek wiskunde te formaliseren.[28] Aangezien het om zo'n groot werk gaat, nodigde hij specialisten in bewijsprogramma's uit om deel te nemen aan het project. Volgens Hales heeft het Flyspeck-project één groot voordeel: het bewijs van het Keplervermoeden bestaat in essentie uit een lange lijst van relatief eenvoudige resultaten in discrete meetkunde.[29] Dit suggereert dat men bij het formaliseren van het bewijs geen grote conceptuele problemen zal tegenkomen.[30]

De eerste grote stap in het Flyspeck-project was een formeel bewijs van de Jordankrommestelling. Deze stelling uit de topologie wordt gebruikt in het bewijs van het Keplervermoeden, waar het gaat om vlakke grafen. De Jordankrommestelling zegt in eenvoudige woorden het volgende: elke eenvoudige gesloten kromme in het vlak verdeelt het vlak in een binnen- en buitenkant.[31] In januari 2005 formaliseerde Hales het bewijs van de Jordankrommestelling in het bewijssysteem HOL-LIGHT. Het bewijs bestond uit 40000 regels code.

In 2006 beëindigde Gertrud Bauer de formalisering van een belangrijk deel van het bewijs van Hales. Zij bewees in haar doctoraatsthesis met de hulp van het programma ISABELLE/HOL dat de verzameling van tamme grafen die Hales berekend had, volledig is.[32] Hiervoor formaliseerde zij de noties van *vlakke graaf* en *tamme graaf* in ISABELLE/HOL. Daarna liet zij in het Isabelle bewijsprogramma alle tamme grafen genereren (van dit deel heeft ze ook een correctheidsbewijs) en vergeleek ze het resultaat met het resultaat van het JAVA-programma van Hales. Beide programma's geven hetzelfde resultaat. Omdat het ISABELLE-programma correct bewezen is, is hiermee bewezen dat het programma van Hales alle tamme grafen genereert. Het bewijs van de volledigheid van de verzameling tamme grafen bevatte slechts een klein aantal *gaten* die nog formeel moesten geverifieerd worden, maar waar Bauer wel informele bewijzen voor geleverd heeft.[33]

Bauer ontdekte echter wel dat Hales' definitie van tamme grafen te ruim was en een aantal grafen bevat die niet werden gegenereerd door zijn algoritme.[34] Hales kon echter bewijzen dat het bewijs blijft gelden als men de definitie van tamme grafen uitbreidt met een extra voorwaarde zodat ze overeenkomt met de gegenereerde grafen. Alle tegenvoor-

[28][Hales2006f] p. 2

[29][Hales2006f] p. 7

[30]En dit is tegelijk ook een aanduiding dat het huidige bewijs *conceptueel arm* is. Op de invloed van de gebruikte concepten in een bewijs op de inzichtelijkheid, gaan we in deel 2 van deze eindverhandeling in.

[31]De Jordankrommestelling lijkt triviaal (zelfs zo triviaal dat wiskundigen ooit niet de nood zagen om de stelling te bewijzen), maar werd pas in 1905 bewezen door Oswald Veblen.

[32][Bauer2006], samengevat in [Bauer2006b]. In Hales' JAVA-programma uit 1998 om grafen te genereren, bleek achteraf een fout in te zitten, waardoor het programma een aantal grafen miste. In 2002 verbeterde Hales de fout, maar door de fout uit 1998 waren er natuurlijk nog twijfels dat de 2002-versie wel correct was. Pas door Bauers thesis weten we dat we de versie uit 2002 wel kunnen vertrouwen.

[33][Bauer2006] p. 135

[34][Bauer2006] hoofdstuk 4, vooral p. 71-72

beelden van het Keplervermoeden voldoen immers ook aan de strengere definitie van een tamme graaf. In de gepubliceerde versies van Hales' bewijs werd de extra definitie toegevoegd.

Bauer werkte samen met Tobias Nipkow en Paula Schultz verder op de resultaten van haar thesis en ze kwamen zo tot een volledig geformaliseerd bewijs van de volledigheid van de verzameling tamme grafen.[35] Bovendien ontdekte Nipkow dat bijna de helft van de grafen in Hales' verzameling *redundant* waren. Van een aantal grafen waren er namelijk isomorfe kopieën in de verzameling aanwezig en andere grafen in de verzameling waren *niet* tam. Deze redundantie heeft geen enkel negatief effect op de correctheid van Hales' bewijs, het duidt er slechts op dat Hales' verzameling van tamme grafen niet zo groot hoefde te zijn. Nipkow reduceerde Hales' verzameling van 5128 grafen tot 2771.[36] De volledige formalisering door Nipkow en zijn collega's bestaat uit 17000 lijnen definities en bewijzen. De verificatie duurde 165 minuten op een computer met een INTEL XEON-processor.

Aan de andere delen van het formele bewijs van het Keplervermoeden is nog veel werk. Roland Zumkeller bestudeert een formalisering van de niet-lineaire ongelijkheden die in het bewijs van Hales voorkomen. Steven Obua heeft de methode van lineair programmeren in ISABELLE/HOL geïntegreerd en schetst hoe een formele verificatie van de onderdelen lineair programmeren in het bewijs kan gebeuren.[37] Een andere moeilijkheid zal nog zijn om alle gedeeltelijke formaliseringen samen te brengen, aangezien ze in verschillende bewijsprogramma's uitgevoerd zijn. Dit is een niet te onderschatten werk.[38]

5.4. De receptie van het bewijs

In 2000 en 2003 waren er al computerbewijzen gepubliceerd in *The Annals of Mathematics*.[39] Na de moeilijke bevalling van het bewijs van Hales (dat in 1998 was ingediend en in 2005-2006 pas volledig gepubliceerd was) besloten de editors van *The Annals* echter hun (tot dan toe ongepubliceerde) beleid aan te passen en ze vermeldden dat computerbewijzen wel hun verdienste hebben, maar voor het tijdschrift toch een lagere status hebben

[35] [Nipkow2006]

[36] [Nipkow2006] p. 33

[37] [Obua2006]

[38] [Bundy2006] p. 483-484: 'Different parts of the proof have been automated in different theorem provers, including HOL-LIGHT, COQ and ISABELLE. This diversity is unfortunate, as it serves to undermine the assurance of correctness and to make the proof more difficult to understand.'

[39] [Hass2000] en [Gabai2003]

dan traditionele bewijzen. Computerbewijzen werden vergeleken met laboratoriumexperimenten om een stelling te ondersteunen. Sindsdien staat er op de website van *The Annals* de volgende tekst:

> ## Statement by the Editors on Computer-Assisted Proofs
>
> Computer-assisted proofs of exceptionally important mathematical theorems will be considered by the *Annals*.
>
> The human part of the proof, which reduces the original mathematical problem to one tractable by the computer, will be refereed for correctness in the traditional manner. The computer part may not be checked line-by-line, but will be examined for the methods by which the authors have eliminated or minimized possible sources of error: (e.g., round-off error eliminated by interval artihmetic, programming error minimized by transparent surveyable code and consistency checks, computer error minimized by redundant calculations, etc. [Surveyable means that an interested person can readily check that the code is essentially operating as claimed]).
>
> We will print the human part of the paper in an issue of the *Annals*. The authors will provide the computer code, documentation necessary to understand it, and the computer output, all of which will be maintained on the *Annals of Mathematics* website online.

Hieruit blijkt al dat niet elk computerbewijs aanvaard wordt, de stelling moet namelijk ‘exceptioneel belangrijk’ zijn.

De reacties van de wiskundige gemeenschap op het computerbewijs van het Keplervermoeden leken zo uit de jaren zeventig van Appel en Haken te komen. Princeton-professor John Conway uitte zijn ongenoegen omdat het bewijs je niet het idee gaf dat je begreep waarom de stelling geldt.[40] Eenzelfde kritiek was te horen van Princeton-wiskundige Pierre Deligne.[41] MacPherson, onder de indruk van het debacle van het nakijken van het computerbewijs, vond het zelfs geen goed idee om wiskundigen een computerbewijs te laten verifiëren, omdat het zo lang duurt en de wiskundigen hun tijd beter aan ‘nuttiger’ zaken zouden besteden.[42]

Conway, Goodman-Strauss en Sloane wijzen op het experimentele karakter van het bewijs van Hales, zowel in het computergedeelte als in het analytische gedeelte.[43] Ze vinden het bewijs als gevolg hiervan niet gemakkelijk te lezen, maar stippen wel aan dat Hales en Ferguson er alles aan gedaan hebben om het nakijken mogelijk te maken: ze heb-

[40][Chang2004] citeert Conway: ‘I don't like them, because you sort of don't feel you understand what's going on.’

[41][Szpiro2003] p. 13 citeert Deligne: ‘I believe in a proof if I understand it.’

[42][Chang2004] citeert MacPherson: ‘In some cases, it's not a good idea to verify computer proofs. It took the effort of many mathematicians for many years, and nothing came out of it.’

[43][Conway1999] p. 39

ben gedetailleerde logboeken bijgehouden van wanneer ze bepaalde gevallen met de computer behandelden, wanneer ze een geval in enkele subgevallen splitsten omdat het te moeilijk was voor het programma, enzovoort. Conway en zijn collega's noemen de manier waarop Hales en Ferguson dit aangepakt hebben een model voor toekomstige computerbewijzen. Ze prezen ook de nauwkeurigheid waarmee de berekeningen werden uitgevoerd en gecontroleerd.

Hales zelf lijkt ook niet helemaal tevreden over de manier waarop zijn bewijs verliep.[44] De reden is voor hem echter niet dat hij de computer nodig had om het vermoeden te bewijzen, integendeel.[45] Volgens Hales heeft hij in het bewijs heel wat manuele procedures uitgewerkt die geautomatiseerd hadden kunnen worden.[46] Als hij nog meer zou automatiseren, zou hij uiteindelijk zelfs een bewijs van het Keplervermoeden kunnen vinden dat kort en elegant is, gelooft hij:[47]

> Ultimately, a properly automated proof of the Kepler conjecture might be short and elegant. The hope is that the Kepler conjecture might eventually become an instance of a general family of optimization problems for which general optimization techniques exist. Just as today linear programming problems of a moderate size can be solved without fanfare, we might hope that problems of a moderate size in this family might be routinely solved by general algorithms. The proof of the Kepler conjecture would then consist of demonstrating that the Kepler conjecture can be structured as a problem in this family, and then invoking the general algorithm to solve the problem.

Hales ziet de computer als een essentieel instrument voor het verbeteren van zijn bewijs van het Keplervermoeden.[48] Dezelfde strategie die hij in het bewijs van het Keplervermoeden gebruikt had, bleek bovendien ook voor andere problemen toepasbaar. Samen

[44][Hales2002] p. 1: 'This is not a *proof from the book*, in the sense of Erdös.' De wiskundige Paul Erdös beweerde dat God een boek had met alle stellingen uit de wiskunde, samen met de 'mooiste' bewijzen ervan. Wiskundigen moesten volgens Erdös deze 'proofs from the book' vinden. Wanneer hij zelf een elegant bewijs zag, riep Erdös uit: 'This is one from the book!' ([Peterson1997]). De wiskundigen Martin Aigner en Günter Ziegler publiceerden onder invloed van Erdös het boek *Proofs from THE BOOK*, dat een aantal van de mooist bekende bewijzen bevat. De *preface* begint als volgt: 'Paul Erdös liked to talk about The Book, in which God maintains the perfect proofs for mathematical theorems, following the dictum of G.H. Hardy that there is no permanent place for ugly mathematics. Erdös also said that you need not believe in God but, as a mathematician, you should believe in The Book.' ([Aigner2004])

[45][Hales2003] p. 489: 'The thesis underlying this article is that the proof is complex because it is highly underautomated.'

[46][Hales2006f] p. 5: 'In the original 1998 proof, I tried to avoid computers, except when paper calculations would have been out of the question. As a result, many results are proved by hand that could have been done in a simpler manner by computer.'

[47][Hales2003] p. 489

[48][Hales2000] p. 446: 'The eventual proof was shaped by the capabilities of computers. If computers had been more powerful, the proof might be drastically shorter. If computers had been less powerful, I would still be working toward a solution. As a result of the development of computers, the proof of the Kepler Conjecture fifty years from now will likely be entirely different from what it is today.'

met zijn doctoraatsstudent Sean McLaughlin bewees Hales in 1998 het *dodecaëdervermoeden*. Het bewijs steunt eveneens uitgebreid op computerberekeningen en heeft grotendeels dezelfde structuur.

Toch kwam er niet enkel kritiek op Hales' bewijs. Brian Davies is optimistisch over een verbetering van het bewijs van Hales waardoor we zekerder zullen zijn van de correctheid ervan.[49] De kritiek dat we niet zeker kunnen zijn dat er geen fouten in het computerprogramma zitten, weerlegt hij als irrelevant, ook al is de kritiek waar. Van mensen weten we immers ook niet zeker dat ze geen fouten maken. We mogen volgens Davies dan ook niets van een computerbewijs verwachten wat we van een menselijk bewijs niet vragen. Hij wijst op het belang van een goede computerverificatie die meer fouten vindt dan menselijke referees.[50] Het werk van Bauer (ten tijde van Davies' artikel nog niet bekend) toont aan dat Davies op dit vlak juist is: formele computerverificatie heeft in Hales' bewijs al fouten gevonden die aan de referees in die vijf jaar tijd ontsnapt waren.

[49][Davies2005] p. 1355: 'It seems to the author that the prospects for a complete proof of the Kepler problem are better than they are for the classification of finite simple groups.'

[50][Davies2005] p. 1355: 'All one can ask of the formal computer verification of proofs is that they perform better than human beings, in the sense that they find mistakes in proofs that humans have missed and that humans recognize once they are pointed out.'

6. Probabilistische computerbewijzen

Alle voorgaande computerbewijzen waren vormen van deductieve bewijzen. In dit hoofdstuk bespreken we *probabilistische bewijzen*. Deze methodes bewijzen een bepaalde uitspraak met een bepaalde waarschijnlijkheid, die in de praktijk dicht genoeg bij zekerheid ligt. We bekijken hier twee probabilistische bewijsmethodes en bespreken de filosofische relevantie hiervan voor de filosofie van de wiskunde.

6.1. Probabilistische priembewijzen

6.1.1. De bewijzen

De conceptueel eenvoudigste manier om te bewijzen dat een bepaald getal n een priemgetal is of niet, is n proberen te delen door alle gehele getallen kleiner dan of gelijk aan √n. Als geen enkel van deze getallen een deler is van n, dan is n een priemgetal. Dit wordt de *trial division test* genoemd. Het is een deductieve procedure: als we de test correct uitvoeren, dan is n een priemgetal.

Er bestaan een aantal probabilistische methodes om met een aan zekerheid grenzende waarschijnlijkheid te bewijzen dat een bepaald getal een priemgetal is: de Fermat priemtest, de Solovay-Strassen priemtest en de Miller-Rabin priemtest. Alledrie zijn op hetzelfde algemene principe gebaseerd, maar ze verschillen in de uitwerking ervan. De wiskundige details doen hier niet toe, daarvoor verwijzen we naar de oorspronkelijke artikels.[1] We beschrijven hier de algemene opzet en maken dus abstractie van de verschillende tests. We zullen het daarom in dit hoofdstuk hebben over een *probabilistische priemtest*.

Een probabilistische priemtest kan van een getal zeggen of het samengesteld is of *waarschijnlijk* een priemgetal. Als het geteste getal samengesteld is, geeft de test dus een perfect deductief bewijs hiervan. Is het getal echter een priemgetal, dan kan de test dit slechts met een bepaalde waarschijnlijkheid zeggen. Dit werkt als volgt. We nemen het te testen getal *n* en kiezen een willekeurig getal *k* kleiner dan *n*. Als er een bepaalde wiskundige relatie geldt tussen *k* en *n* waaruit volgt dat *n* een samengesteld getal is, dan zeggen we dat *k* een 'getuige' is van de samengesteldheid van *n*. Een getuige genereren is niet eenvoudig, maar berekenen of een willekeurige *k* een getuige is, gaat heel snel.[2] Als we dus willen weten of *n* samengesteld is, kunnen we van een groot aantal willekeurige

[1] [Solovay1977], [Rabin1976] en [Rabin1980]. Rabin baseerde zijn probabilistische priemtest op een deterministische methode van Miller ([Miller1975]), die echter uitging van de (nog altijd) onbewezen gegeneraliseerde Riemann-hypothese. [Chaitin2004] geeft een interessante herformulering van probabilistische priemtesten in algoritmische informatietheorie.

[2] De complexiteit van de berekening is *polylogaritmisch* in n, genoteerd als $O((\log n)^c)$. Dit wil zeggen dat de tijd dat de berekening duurt evenredig is met een veelterm van de logaritme van het getal n, ofwel (aangezien de logaritme van het getal het aantal cijfers in het getal geeft) evenredig met een veelterm van het aantal cijfers van het getal n.

's kleiner dan *n* berekenen of ze een getuige zijn. Van zodra we een getuige gevonden hebben, weten we zeker dat *n* samengesteld is. Maar als we na verschillende berekeningen nog geen getuige gevonden hebben, weten we nog niet zeker dat *n* een priemgetal is.

Het probabilistische aspect zit hierin: als *n* samengesteld is, is een bepaald percentage van de getallen kleiner dan *n* een getuige. In de Miller-Rabin test is dit 3/4 van de getallen.[3] Als we nu 100 willekeurige getallen kleiner dan *n* kiezen, is de kans dat geen van deze getallen een getuige is $(1/4)^{100}$. Deze kans is enorm klein, dus als de Miller-Rabin test geen getuige vindt voor een getal na 100 testen, dan kunnen we er vrij zeker van zijn dat het om een priemgetal gaat. Het zou namelijk heel erg toeval zijn als we bij het kiezen van willekeurige getallen telkens naast de getuigen grepen, die toch 75% van de getallen kleiner dan *n* uitmaken. We kunnen bovendien zelf onze graad van waarschijnlijkheid kiezen door het aantal tests dat we uitvoeren te variëren.

6.1.2. De receptie van de bewijzen

De probabilistische methodes om te bewijzen dat een getal een priemgetal is, zijn veel sneller dan de bestaande deductieve bewijsmethodes, maar omdat ze niet deductief zijn, wordt hun resultaat niet als bewezen aanvaard. Er bestaan bijvoorbeeld lijsten van de grootste bekende priemgetallen, waarop een getal slechts wordt geplaatst als er een deductief bewijs van het priemgetal gevonden is.[4] Als een probabilistische priemtest geen getuige voor het samengesteld zijn van een getal gevonden heeft, noemt men het getal een ‘waarschijnlijk’ priemgetal. In cryptografische toepassingen, bijvoorbeeld coderingen bij elektronische banktransacties, worden deze getallen wel als priemgetallen aanvaard, maar in de wiskunde niet. De tests kunnen wel gebruikt worden om kandidaatpriemgetallen te selecteren waarvan men dan met deductieve methodes probeert te bewijzen dat het om een priemgetal gaat.

Michael Detlefsen en Mark Luker waren bij de eersten die het over de filosofische aspecten van probabilistische priembewijzen hadden. Ze besluiten hun artikel over het computerbewijs van het vierkleurenprobleem namelijk met een bespreking van de Miller-Rabin test. Na het bespreken van de foutkans van het algoritme merken zij op dat deze probabilistische algoritmes op vlak van betrouwbaarheid toch zeker niet moeten onderdoen voor deductieve bewijzen en dat de betrouwbaarheid dan ook geen reden kan zijn om probabilistische bewijzen uit te sluiten.[5] De auteurs citeren ook de wiskundige Ronald Graham die toegaf dat hij meer vertrouwen had in resultaten van Rabins probabilistische priemtest dan in resultaten van lange en ingewikkelde traditionele bewijzen. Detlefsen en Luker wijzen erop dat ons vertrouwen in bewijzen afhangt van hun complexiteit, omdat de fout-

[3] Bij de Solovay-Strassen test is dit 1/2.

[4] [Fallis2002] p. 377

[5] [Detlefsen1980] p. 818: ‘From this it would seem to be clear that degree of certainty or reliability is no reason for not accepting probabilistic methods into the canon of accepted methods of mathematical proof. For surely there are many results in traditional mathematics whose degree of certitude is exceeded by that obtainable using Rabin's techniques.’

kans verhoogt met de complexiteit. Dat probabilistische algoritmes geen absolute zekerheid bieden, kan dus geen reden zijn om ze niet toe te laten, want traditionele bewijzen bieden ook geen absolute zekerheid.[6]

Volgens Detlefsen en Luker zou een acceptatie van probabilistische bewijzen in de canon van aanvaarde bewijsmethodes het karakter van wiskundige bewijzen volledig veranderen. Dan zouden namelijk voor de eerste keer in de geschiedenis van de wiskunde niet-deductieve argumenten als bewijs gezien worden. Zij beschouwen Rabins werk om die reden als veel belangrijker dan de voorgaande deductieve computerbewijzen, inclusief dat van de vierkleurenstelling.[7] Ze merken nog op dat er uiteraard andere redenen kunnen zijn buiten het verlangen naar betrouwbare methodes om toch de aanvaarde bewijsmethodes te beperken tot deductieve methodes en dat Rabins algoritme dan nooit aanvaard zal worden als bewijs. Een kwart eeuw later blijkt dat dit nog altijd zo is.

Bij vrijwel alle wiskundigen heerst nog de overtuiging dat enkel *deductieve bewijzen* de waarheid van een wiskundige bewering kunnen ondersteunen. Zij keuren dus probabilistische (computer)bewijzen af. Don Fallis argumenteert echter dat wiskundigen geen enkele *epistemische* redenen hebben voor deze afwijzing.[8] Fallis bestudeert de doelstellingen van wiskundigen in het raamwerk van *means/end reasoning*. Om te begrijpen waarom wiskundigen doen wat ze doen, moeten we namelijk hun doelen identificeren. In dit raamwerk bestudeert Fallis een belangrijke methodologische keuze die wiskundigen maken, namelijk dat ze enkel deductieve bewijzen aanvaarden om de waarheid van een wiskundige bewering te bevestigen. Volgens Fallis ligt deze methodologische keuze in de lijn van de voorkeur voor deductieve argumenten over inductieve argumenten bij filosofen. Een verklaring waarom wiskundigen probabilistische bewijzen afwijzen, kan volgens Fallis dus ook iets leren over waarom deductieve argumenten als superieur beschouwd worden ten opzichte van inductieve argumenten.

Een deductief bewijs van een wiskundige bewering wordt beschouwd als een procedure die, als ze correct wordt uitgevoerd, verzekert dat de bewering waar is. Zelfs bij de lange computerbewijzen van het vierkleurenprobleem is nooit twijfel gerezen over de geldigheid van de procedure op zich, maar wel over de correctheid van de uitvoering ervan of ons vermogen om de correctheid ervan te verifiëren. Een probabilistisch bewijs van een wiskundige bewering is daarentegen een procedure die, zelfs als ze correct uitgevoerd wordt, niet verzekert dat de bewering waar is. Ze kan wel een hele goede reden geven om te denken dat de bewering waar is en heel wat probabilistische bewijstechnieken zijn heel betrouwbaar.

Ondanks de vaak hoge betrouwbaarheid van probabilistische bewijzen, nemen wiskundi-

[6][Detlefsen1980] p. 819: ‘Because of this, the limited but extremely high certitude provided by Rabin's techniques is no reason for not allowing his methods entrance into the methods of mathematical proof.’

[7][Detlefsen1980] p. 819

[8][Fallis2002] p. 373: ‘In this paper, I argue that none of the epistemic objectives of mathematicians that are currently on the table provide a satisfactory explanation of this rejection of probabilistic proofs.’

gen ze niet aan als bewijzen van de waarheid van een wiskundige bewering.[9] Probabilistische bewijzen horen volgens wiskundigen in de categorie van 'heuristische redeneringen', naast inductieve overwegingen en bewijzen door tekeningen. Deductieve bewijzen vallen onder de categorie van 'demonstratieve redeneringen' en wiskundigen spreken slechts over een 'finished piece of mathematics'[10] als het resultaat enkel door demonstratief redeneren is bereikt. Dit is een methodologische keuze die al eeuwen gemaakt wordt door wiskundigen.[11]

Volgens Fallis zijn er in het raamwerk van means/end reasoning drie mogelijke redenen waarom iemand (i.c. de wiskundige gemeenschap) een bepaalde activiteit (i.c. probabilistische bewijzen) afwijst:

1. De activiteit is geen middel om de doelstellingen te bereiken.
2. De activiteit wordt niet herkend als een middel om de doelstellingen te bereiken.
3. De persoon is niet rationeel.

Volgens Fallis nemen de meeste wiskundigen en filosofen de eerste optie aan waar het gaat om probabilistische bewijzen: een probabilistisch bewijs is geen middel om de doelstellingen van wiskundigen te bereiken. Zo citeert Fallis de wiskundige Carl Pomerance die zegt: '[There is] a qualitative difference between probabilistic verification and mathematical proof that is important to mathematicians.'[12] Volgens Fallis is er echter niet zo'n kwalitatief verschil en hij ziet dan ook geen reden om aan te nemen dat deductieve bewijzen wel een middel zijn om de doelstellingen van wiskundigen te bereiken en probabilistische bewijzen niet.[13] Uiteraard zijn er bezwaren te geven tegen probabilistische bewijzen, maar volgens Fallis kunnen die bezwaren niet epistemisch zijn. Zo kan een wiskundige een probabilistisch bewijs weigeren omdat hij een deductief bewijs esthetisch mooier vindt.

Volgens David Corfield kan je Fallis' uitspraak dat er geen kwalitatief verschil is tussen een probabilistisch bewijs en een deductief bewijs in Bayesiaanse termen uitdrukken:[14]

> The reliability of a mathematical statement is dependent solely on your rational degree of belief in that statement conditioned on all the relevant evidence. Whatever level you set yourself (0.99 or 0.99999), the type of evi-

[9]Fallis citeert de wiskundige David Harel: 'As long as we use probabilistic algorithms only for petty, down-to-earth matters such as wealth, health, and survival, we can easily make do with very-likely-to-be-correct answers to our questions. The same, it seems, cannot be said for our quest for absolute mathematical truth.' en Mark Steiner: 'Journals of mathematics will not publish anything less than a proof of a scholarly result.' ([Fallis2002] p. 375)

[10][Fallis2002] p. 375, citaat van David Sherry

[11][Fallis2002] p. 375 citeert Euler: 'We should take great care not to accept as true such properties of the numbers which we have discovered by observation and which are supported by induction alone.'

[12][Fallis2002] p. 376

[13][Fallis1997]

[14][Corfield2003] p. 110

> dence which has led you there is irrelevant.

Corfield vergelijkt de betrouwbaarheid die we toekennen aan een bepaald resultaat met de situatie waarin je als niet-specialist aan een specialist om zijn raad vraagt. Als de specialist je zegt dat hij heel zeker is van een bepaald resultaat, maakt het je niet uit hoe hij tot deze zekerheid gekomen is.

Een aantal wiskundigen aanvaarden wel probabilistische bewijzen en baseren zich daarbij op het feit dat we heel wat wiskundige beweringen niet kunnen bewijzen zonder probabilistische bewijzen, omdat een deductief bewijs, hetzij door een een mens hetzij door een computer, te lang zou zijn om te produceren. We kunnen bewijzen dat er heel wat wiskundige stellingen bestaan die niet efficiënt bewijsbaar zijn op de klassieke deductieve manier.[15] De menselijke geest is eindig en we leven eindig lang, dus er bestaan stellingen die nooit door een mens kunnen bewezen worden, laat staan begrepen.[16] Er bestaan zelfs stellingen waarvan het kortste bewijs te lang is voor computers.[17] Als we deze problemen wel kunnen oplossen met probabilistische methodes, waarom zouden we dat dan niet doen? Fallis vermeldt als wiskundigen met dit standpunt Doron Zeilberger, Reuben Hersh en Gregory Chaitin.[18]

Fallis vermeldt twee belangrijke epistemische doelstellingen die wetenschappers en wiskundigen hebben: meer ware overtuigingen bekomen en fouten vermijden. Het verschil tussen wetenschappers en wiskundigen lijkt volgens hem in de rangorde van deze doelstellingen te liggen: terwijl wetenschappers gemakkelijker inductieve argumenten gebruiken, zijn wiskundigen daar afkeriger van: zij vinden het belangrijker om fouten te vermijden dan om nieuwe kennis op te doen. Wiskundigen vermijden dus epistemische risico's.[19]

---

[15]Zie [Norwood1982] en [Spencer1983] voor enkele opmerkingen over lange bewijzen. Spencer bewijst de volgende stelling: ‘For all recursive functions F there exist theorems T whose shortest proofs have length at least F(n) where n is the length of T.’

[16][Norwood1982] p. 112: ‘Since human beings can read only a finite number of words a minute, and since human lifetimes are finite, it follows that there are true theorems whose proof the human mind can never comprehend, even though valid proofs exist.’

[17][Norwood1982] p. 112: ‘There has been some grumbling about the use of a computer in Appel and Haken's proof of the four color theorem. But given the existence of true theorems with proofs too long for human comprehension, the use of computers to prove theorems seems unavoidable. It is cold comfort that there are also true theorems whose shortest proof is too long for any computer to handle.’

[18][Fallis2002] p. 377. Zeilbergers provocatieve artikel [Zeilberger1993] schetst een toekomt waarin mensen en computers samenwerken aan bewijzen die quasi-zekerheid geven. Chaitin argumenteert in [Chaitin2004] zelfs op basis van een herinterpretatie van Gödels en Turings resultaten in algoritmische informatietheorie dat er wiskundige uitspraken zijn die *per toeval* waar zijn en waar geen bewijs van te vinden is die de reden geeft. Willekeur neemt volgens hem een belangrijke plaats in de wiskunde in en hij heeft dit standpunt in de loop der jaren in verschillende artikels en boeken uiteengezet. Zie in [Chaitin1975], [Chaitin1982] en [Chaitin2002] voor een toegankelijk overzicht en [Chaitin1974] voor een goede inleiding in algoritmische informatietheorie.

[19][Fallis2002] p. 379. In het voorwoord van [Singh1998] vinden we ook een opmerking in deze zin: John Lynch, redacteur van de BBC-serie Horizon, schrijft hierin dat de wiskundige Peter Sarnak hem vertelde dat wiskundigen ‘domweg een hekel hebben aan het doen van een onware bewering’.

Volgens Fallis kan dit vermijden van epistemische risico's echter niet verklaren waarom wiskundigen geen probabilistische bewijzen aanvaarden, een observatie die Detlefsen en Luker eerder al maakten. Zowel in de Miller-Rabin priemtest als de deductieve trial division test kan men fouten maken. Een wiskundige die de trial division test uitvoert op een getal n, kan namelijk een fout maken bij het testen of n deelbaar is door een bepaald getal. Hierdoor kan hij tot de foute conclusie komen dat het getal een priemgetal is. Als hij de Miller-Rabin test met de hand zou uitvoeren op een getal n, kan hij een fout maken in de test of een bepaald getal een getuige is van de samengesteldheid van dit getal. Ook hier weer kan hij zo tot de foute conclusie komen dat het getal een priemgetal is. Fallis noemt de Miller-Rabin priemtest in de praktijk betrouwbaarder dan de deductieve trial division test. De Miller-Rabin test is namelijk eenvoudiger en vereist in het algemeen veel minder berekeningen dan de trial division test. De waarschijnlijkheid dat je een fout maakt in een berekening is dus kleiner bij de Miller-Rabin test. Bovendien kan je de intrinsieke foutkans van de probabilistische Miller-Rabin test willekeurig klein maken door van meer getallen kleiner dan n te testen of ze een getuige zijn.[20] De redenering blijft natuurlijk gelden als je beide methodes op een computer uitvoert in plaats van met de hand.

Catherine Womack en Martin Farach expliciteren deze intuïties van Fallis: probabilistische bewijzen (die zij ‘randomized algorithms’ of RA's noemen) zijn volgens hen in het algemeen korter dan klassieke bewijzen én ze formaliseren het idee van inherente fouten die altijd in de praktijk van wiskundige processen voorkomen.[21] Hoe langer een bewijs is, hoe moeilijker het is om te verifiëren of het correct is. Wiskundigen hebben dan ook graag dat een bewijs duidelijk en kort is. Voor bepaalde wiskundige problemen waarvoor we geen deductief bewijs van een redelijke lengte kunnen vinden, bestaan er probabilistische oplossingen die veel korter zijn. In combinatie met het feit dat we de foutkans willekeurig klein kunnen maken door het aantal stappen in de procedure te verhogen, zien Womack en Farach probabilistische bewijzen als geldige bewijsprocedures.[22] Het standaard bezwaar tegen probabilistische bewijzen om hun probabilistische karakter is volgens de auteurs, net zoals Fallis zegt, niet epistemologisch gegrond.

Dat wiskundigen toch enkel deductieve bewijzen aanvaarden, is volgens Womack en Farach te verklaren doordat zij willen dat hun methodes *in principe* correct zijn. Deductieve bewijzen zijn in principe correct, maar door menselijke of hardwarefouten kan er in de praktijk toch een fout in het bewijs sluipen. Probabilistische bewijzen zijn in principe niet foutloos, maar in de praktijk is de kans op fouten willekeurig klein te maken. Volgens de auteurs is er geen enkele reden om de zekerheid in principe belangrijker te vinden dan de quasi-zekerheid in de praktijk. De eisen van wiskundigen zijn te streng, te idealistisch en houden geen rekening met de praktijk.[23]

[20][Fallis2002] p. 379-380

[21][Womack2003]. Het artikel in *Synthese* schrijft de namen van beide auteurs verkeerd als Catherine Womach en Matrin Farach.

[22][Womack2003] p. 72: ‘These procedures relax the constraints on standard mathematical proof techniques while optimizing on length of procedure and certainty about the method used.’

Fallis legt dit ‘in principe correct zijn van deductieve bewijzen’ anders uit. Het probleem dat wiskundigen met probabilistische bewijzen hebben is volgens hem niet dat ze niet genoeg zekerheid bieden, maar dat ze niet de *juiste soort* zekerheid bieden.[24] Zowel deductieve als probabilistische bewijzen geven geen absolute zekerheid, maar deductieve bewijzen geven een *voorwaardelijke garantie* dat de bewering waar is. Onder de voorwaarde dat de procedure van een deductief bewijs correct wordt uitgevoerd (dus *in principe*), is de stelling waar. Een probabilistische bewijsprocedure kan echter zelfs geen voorwaardelijke garantie bieden. Wiskundigen geven volgens Fallis duidelijk de voorkeur aan een voorwaardelijke garantie, maar volgens hem is dit meer om esthetische dan om epistemische redenen.[25]

Fallis vraagt zich ook af of probabilistische bewijzen misschien zelfs beter zijn dan deductieve bewijzen als we fouten willen vermijden. Als we willen vermijden dat er een fout zit in een deductief bewijs, moeten we het bewijs minitieus nagaan, maar dat garandeert nog niet dat we een fout vinden. Zo duurde het 11 jaar voordat de fout in Kempes ‘bewijs’ van de vierkleurenstelling werd gevonden. Als we van een probabilistische test zoals de Miller-Rabin test willen vermijden dat er een fout in zit, moeten we gewoon van meer willekeurige getallen onderzoeken of het getuigen zijn.[26]

De kritiek dat probabilistische priemtesten geen inzicht geven in waarom het geteste getal een (waarschijnlijk) priemgetal is, kan ook de afwijzing van deze bewijzen niet verklaren. Er zijn namelijk genoeg voorbeelden bekend van deductieve bewijzen die ook geen inzicht geven in de te bewijzen stelling. Er wordt in het algemeen wel neergekeken op bewijzen die geen inzicht geven, maar ze worden wel aanvaard als correct bewijs. Op probabilistische bewijzen reageren wiskundigen sterker: ze aanvaarden deze bewijzen gewoonweg niet.

6.2. DNA-berekeningen

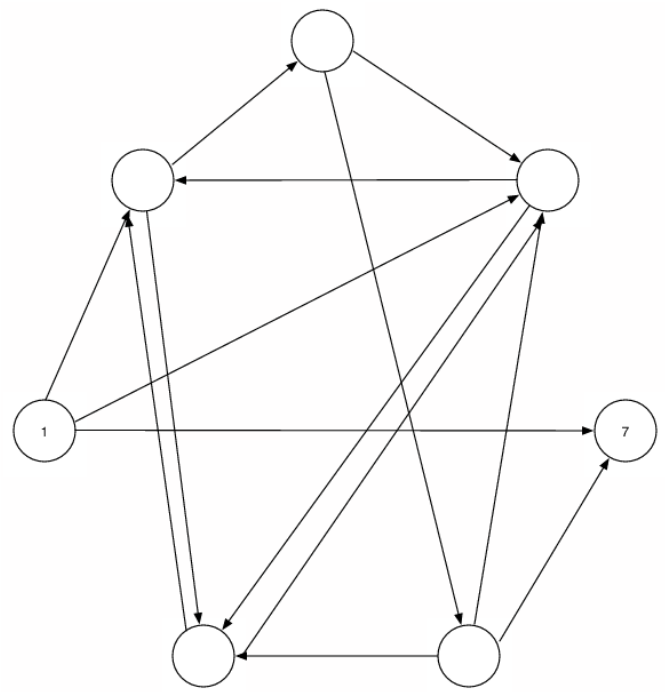
6.2.1. Het probleem en zijn DNA-bewijs

Een ander voorbeeld van een probabilistisch bewijs is een methode die de computerwetenschapper Leonard Adleman ontwikkelde om een *Hamiltoniaans pad* in een gerichte graaf te vinden. Adleman maakte hiervoor niet gebruik van conventionele computers,

[23][Womack2003] p. 79: ‘The certainty conveyed by classical proof trades on the rigor of formal proof, abstracting away from human capabilities of comprehending it. Probabilistic procedures have built-in ways of taking into account the effects our physical and cognitive limitations impose upon us, and provide formal structure for adjusting the tension between the degree of rigor and the length of procedure.’
[24][Fallis1997] p. 174
[25][Fallis1997] p. 175
[26][Fallis2002] p. 381

maar van DNA-moleculen.[27] Een gerichte graaf kan gezien worden als een verzameling steden (punten) die verbonden zijn door eenrichtingswegen (zijden). Je start in een bepaalde stad en je stopt in een bepaalde stad. Een pad is dan een rij van steden die elk door een eenrichtingsweg met hun opvolger verbonden zijn. Een Hamiltoniaans pad is een pad van de beginstad naar de eindstad zodat elke stad in de verzameling exact één keer bezocht wordt.[28] Een voorbeeld van een gerichte graaf met de beginstad en eindstad opgegeven is deze:

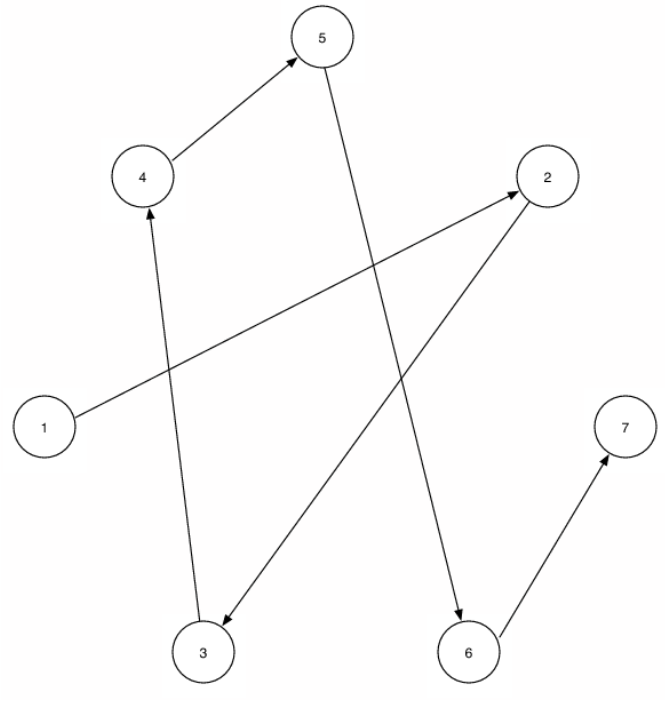
Figuur 6.1. Voorbeeld DHPP-opgave



Het ‘directed Hamiltonian path problem’ (DHPP) bestaat nu uit de vraag of een gegeven gerichte graaf een Hamiltoniaans pad bevat en indien ja welk. Ons voorbeeld heeft deze oplossing:

[27] [Adleman1994]
[28] [Fallis1996]

Figuur 6.2. Voorbeeld DHPP-oplossing



Het DHPP is een voorbeeld van een *NP-volledig* probleem: het is gemakkelijk om een voorgestelde oplossing van een probleem op zijn correctheid te verifiëren, maar het is moeilijk om een oplossing te vinden. Voor een gegeven pad kan je dus eenvoudig verifiëren of het een oplossing is van het DHPP van een bepaalde graaf, maar een Hamiltoniaans pad van die graaf vinden is moeilijk. Alle bekende algoritmes om het probleem op te lossen, hebben een exponentiële complexiteit: als het aantal punten in een graaf toeneemt, neemt het aantal paden die exact dat aantal grafen bezoekt exponentieel toe.

Adleman loste dit probleem van de complexiteit in essentie op door alle mogelijke paden tegelijk te testen. Hij maakte hiervoor gebruik van DNA-moleculen. De punten en zijden in een pad van een graaf stelde hij voor door een DNA-streng. Een DNA-streng bestaat uit een bepaalde soort moleculen, *basen* genaamd. De vier types basen in DNA worden aangeduid met de letters A, T, G en C. CTTGAG stelt bijvoorbeeld een DNA-streng van 6 molecules lang voor. Onder de juiste omstandigheden binden twee strengen DNA zich aan elkaar vast en vormen dan de karakteristieke *dubbele helix*. De base A bindt zich enkel aan de base T en G enkel C. Adleman stelt nu een stad X bijvoorbeeld voor door de DNA-streng GCTATTCGAGCTTAAAGCTA en een stad Y door de DNA-streng GGCTAGGTACCAGCATGCTT. Een weg van stad X naar stad Y wordt dan voorgesteld door de basen die zich binden aan een deel van stad X en een deel van stad Y:[29]

[29][Fallis1997] p. 169

Voorbeeld 6.1. DNA-streng van een weg

```
{      Stad X      }{      Stad Y      }
GCTATTCGAGCTTAAAGCTAGGCTAGGTACCAGCATGCTT
          ||||||||||||||||||||
          GAATTTCGATCCGATCCATG
          { Weg van X naar Y }
```

Adleman voerde dan in de proefbuisjes door middel van chemische reacties het volgende algoritme uit:

1. Genereer een groot aantal paden door de graaf.
2. Verwijder alle paden behalve degene die beginnen met de startstad en eindigen met de eindstad.
3. Verwijder alle paden behalve degene die exact N steden bezoeken.
4. Verwijder alle paden behalve degene die elke stad bezoeken.

Elke streng DNA die na deze vier stappen overblijft, stelt een Hamiltoniaans pad van de graaf voor. Als je hieraan twijfelt, kan je dit altijd eenvoudig controleren. Uiteraard moet dit hierboven vermelde algoritme in de praktijk uitgevoerd worden met DNA-moleculen: elke stap wordt door een bepaalde chemische reactie uitgevoerd.[30]

6.2.2. De receptie van de methode

Je kan deze DNA-methode op twee manieren gebruiken. Als de methode een Hamiltoniaans pad vindt, kan je concluderen dat het probleem een oplossing heeft. Je kan dit ook heel eenvoudig controleren. Als de methode *geen* Hamiltoniaans pad geeft, kan je hier nog geen deductief besluit uit trekken. Je kan bijvoorbeeld heel wat pech hebben dat de oplossing niet tussen de paden in de eerste stap van het algoritme gegenereerd is. Deze kans is echter heel klein, zodat je in dat geval wel een ‘probabilistisch bewijs’ hebt van het niet bestaan van een Hamiltoniaans pad van de graaf.

In het eerste geval, met andere woorden als de methode een oplossing vindt, beschouwt Fallis de DNA-methode epistemisch volledig analoog aan deductieve computerbewijzen zoals van de vierkleurenstelling.[31] We voeren namelijk in beide gevallen een berekening uit die in principe een correcte oplossing geeft. Het vierstappenalgoritme selecteert enkel correcte oplossingen: de paden die na stap 4 overblijven, moeten een Hamiltoniaans pad zijn. De berekeningen worden natuurlijk in de praktijk door een fysisch apparaat uitge-

[30] Een goede uitleg is te vinden in [Fallis1996] p. 492-493.
[31] [Fallis1996] p. 493

voerd. Dit apparaat kan fouten maken, maar dit kunnen we nooit helemaal uitsluiten. Door een goed ontwerp hebben we echter goede inductieve aanwijzingen dat het apparaat correct werkt. Als de biochemische processen die Adleman uitvoerde het algoritme correct uitvoeren, geven ze de juiste oplossing.

Het enige epistemisch relevante verschil tussen deze DNA-berekening en een deductief computerbewijs kan volgens Fallis dus in de betrouwbaarheid van het fysische apparaat liggen. Dit is echter een puur contingente factor, die afhangt van de stand van de techniek. Epistemisch gezien zijn een DNA-berekening en een deductief computerbewijs hier equivalent. Er is hier geen sprake van een expliciet probabilistisch karakter.[32]

In het tweede geval, met andere woorden als de methode *geen* oplossing vindt, kunnen we niet deductief concluderen dat de graaf geen Hamiltoniaans pad heeft en heeft het bewijs een probabilistisch karakter. Zelfs als de biochemische processen het algoritme correct uitvoeren en er na stap 4 geen DNA-strengen meer overblijven, kan de graaf toch nog een Hamiltoniaans pad hebben. Stap 1, waarin een groot aantal DNA-strengen wordt gegenereerd die elk paden in de graaf voorstellen, geeft ons namelijk niet de garantie dat *alle* mogelijke paden gegenereerd worden. Welke paden er worden gegenereerd, hangt af van toeval. Als er geen oplossing overblijft na stap 4, is er echter wel een heel grote waarschijnlijkheid dat er effectief geen oplossing is.[33] Fallis besluit met de volgende redenen waarom een probabilistische verificatie zoals in Adlemans experiment epistemisch niet verschilt van traditionele methodes die door wiskundigen worden aangenomen:

1. Een probabilistisch DNA-bewijs is niet foutloos, maar een traditioneel bewijs is dat evenmin.
2. Een probabilistisch DNA-bewijs construeert geen rigoureus wiskundig bewijs, maar heel wat bewijzen en bewijstechnieken die wiskundigen aanvaarden doen dat evenmin.
3. Een probabilistisch DNA-bewijs biedt goede aanwijzingen dat we een rigoureus wiskundig bewijs van het resultaat kunnen geven, net zoals veel traditionele bewijzen doen.
4. Een probabilistisch DNA-bewijs steunt niet meer op empirische factoren dan een traditioneel bewijs en kan in principe zelfs in het hoofd van een wiskundige uitgevoerd worden.

Desondanks nemen wiskundigen DNA-bewijzen niet aan.

[32][Fallis1996] p. 494

[33]Fallis citeert Keith Devlin die zegt dat ‘for all practical purposes this result could be taken as certainty, since most “definite” conclusions in everyday life are based on far lower probabilities.’ ([Fallis1996] p. 495)

7. Andere computerbewijzen

Uiteraard zijn de voorbeelden die ik in de voorgaande secties gaf niet de enige computerbewijzen die we kennen. In de laatste tien à twintig jaren is de computer meer en meer doorgebroken als instrument bij wiskundigen, ook om bewijzen te leveren. Case-by-case verificaties zoals de bewijzen van het vierkleurenprobleem en die van het niet bestaan van een projectief vlak van orde 10 zijn heel geschikt om door een computer te laten uitvoeren. In deze categorie bestaan er heel wat computerbewijzen die minder bekend zijn, vaak omdat het niet zo'n beruchte open problemen waren of veel specifiekere stellingen. We beschrijven hier kort enkele andere computerbewijzen, opgedeeld in vier categorieën: symbolische berekeningen, combinatorische problemen, numerieke benaderingsproblemen en axiomatische bewijsproblemen.

7.1. Symbolische berekeningen

Computers worden al tientallen jaren ingezet voor symbolische berekeningen. Het bewijs van Philip Davis en Elsie Cerutti van de stelling van Pappus is een voorbeeld hiervan: hun FORMAC-programma evalueerde een uitdrukking in symbolen en verifieerde dat de uitkomst 0 was.[1] Tegenwoordig gebruiken heel wat wiskundigen symbolische wiskundesoftware, zoals MATHEMATICA, MAPLE en MATLAB. Uiteindelijk zijn dit veredelde rekenmachines, maar ze kunnen wel met symbolen rekenen en vaak ook uitdrukkingen vereenvoudigen. Hierdoor zijn deze programma's te beschouwen als een rechtstreekse uitbreiding van de menselijke vermogens om te rekenen. Wiskundigen kunnen natuurlijk gebruik maken van de resultaten van zo'n berekeningen in bewijzen, wat ook veelvuldig voorkomt.

Een belangrijk type van bewijzen in deze categorie is het bewijzen van identiteiten van bepaalde veeltermen. Doron Zeilberger heeft hier een krachtig algoritme voor uitgevonden en heel wat software om identiteiten te bewijzen is hierop gebaseerd.[2] Met het oog hierop heeft Axel Riese het computerpakket QMULTISUM ontwikkeld. Dit pakket kan in de MATHEMATICA software ingeplugd worden om recursieve betrekkingen tussen q-hypergeometrische multisommen te berekenen.[3] Alexander Berkovich en Axel Riese bewezen met behulp van QMULTISUM een aantal nieuwe gelijkheiden van veeltermen.[4] Een andere recente succesvolle toepassing vinden we in de computerbewijzen van vijf identiteiten tussen harmonische getallen die men al vermoedde, maar waar men geen bewijs voor kon vinden. Peter Paule en Carsten Schneider vonden hier in 2003 met behulp van twee verschillende computermethodes bewijzen van.[5]

[1] [Cerutti1969]
[2] [Petkovsek1996] geeft een goed overzicht van de algoritmes.
[3] [Riese2003]
[4] [Berkovich2002]
[5] [Paule2003]

De algoritmes van Zeilberger en anderen die hypergeometrische identiteiten kunnen bewijzen, kunnen volledig mechanisch en zonder menselijke inbreng moeilijke sommen met binomiaalcoëfficiënten, faculteiten, breuken en machten vereenvoudigen tot verrassende resultaten.[6] Hoewel de berekeningen lang kunnen duren, kunnen de resultaten eenvoudig geverifieerd worden.[7] De WZ-methode die de auteurs bespreken in hun boek is een gestandaardiseerde bewijsprocedure die je op elke hypergeometrische identiteit kan toepassen. De berekeningen zijn uiteraard vaak heel moeilijk te volgen voor mensen, maar de procedure garandeert dat, als het programma correct werkt, het een bewijscertificaat aflevert, dat een rationele functie R(n, k) is. Met deze rationele functie kan je heel eenvoudig het bewijs verifiëren.[8]

7.2. Combinatorische oplossingen

Meestal wordt het bewijs door Appel en Haken van de vierkleurenstelling als het eerste computerbewijs voorgesteld. Dit klopt echter niet. Het is wel het eerste *grote* bewijs van een stelling waarbij men zijn toevlucht tot computers moest nemen. De computer werd echter al veel vroeger in de bewijsvoering binnengebracht. In de jaren zestig en zeventig werden al verschillende (relatief kleine vergeleken met het vierkleurenprobleem) combinatorische problemen opgelost met behulp van computers. De getaltheoreticus Derrick Lehmer, die meegewerkt had aan de Eniac-computer, was één van de eerste.[9] Als p een priemgetal is van de vorm 6m + 1, bestaan de getallen $1^3, 2^3, \ldots, (p - 1)^3 \pmod p$ uit 2m verschillende getallen tussen 1 en p - 1. Deze getallen heten de *kubische residu's* van p. Er waren al kleine priemgetallen van de vorm 6m + 1 bekend die geen drie opeenvolgende kubische residu's hebben en deze werden *uitzonderlijke priemgetallen* genoemd.

[6][Petkovsek1996] p. 17: ‘They can find very pretty proofs of very difficult theorems in the field of combinatorial identities. The computers do that by themselves, unassisted by hints or nudges from humans.’

[7][Petkovsek1996] p. 17: ‘Although the computers will have to blink their lights for quite a long time, when they are finished they will give to us people a short certificate from which it will be easy to check the truth of what they are claiming.’

[8]Zie [Petkovsek1996] p. 124 voor een samenvatting van hoe je een identiteit kan verifiëren op basis van zijn WZ-certificaat en hoe je zo'n WZ-certificaat vindt van een identiteit.

[9][Lehmer1962]

Voorbeeld 7.1. Het priemgetal 13 is uitzonderlijk

De kubische residu's van 13 zijn 1, 5, 8 en 12.

Voorbeeld 7.2. Het priemgetal 97 is niet uitzonderlijk

De kubische residu's van 97 zijn 1, 8, 12, *18*, *19*, *20*, 22, 27, 28, 30, …

Sinds 1928 was al bekend dat alle 'voldoende grote' priemgetallen van de vorm 6m + 1 drie opeenvolgende kubische residu's hebben. Dit betekende dus dat er slechts een eindig aantal uitzonderlijke priemgetallen bestaan. Lehmer vond met een computer alle uitzonderlijke priemgetallen[10] en bewees (eveneens met een computerprogramma) dat bij elk niet-uitzonderlijk priemgetal het triplet van opeenvolgende kubische residu's maximum 23532, 23533, 23534 is. Het uitvoeren van het bewijs duurde 40 minuten op een IBM 7090 computer. Lehmers bewijs deelt al een heel aantal eigenschappen met de latere grote computerbewijzen. Zo zijn de resultaten van het bewijs te uitgebreid om met de hand na te gaan.[11] Vreemd genoeg leek daar in die tijd niemand van wakker te liggen: de computerberekening werd gewoon aanvaard als deel van het bewijs. De referee vond Lehmers bewijs zelfs geen 'echt' *computer*bewijs, aangezien het computerprogramma niet zelf zocht naar het bewijs, maar slechts de case-by-case analyse van Lehmer verifieerde.[12] Hij zag het computergedeelte blijkbaar als een zuivere uitbreiding van de menselijke rekenvermogens, wat ook de mening van Swart was in de discussie over het bewijs van Appel en Haken.

In 2002 bewees Petr Hlinený een vermoeden dat Dharmatilake in 1994 in zijn doctoraatsthesis uitsprak over binaire matroides. Hlinený's bewijs maakte gebruik van resultaten die hij met zijn programma MACEK berekende.[13] Om twijfels over de correctheid van de berekeningen af te zwakken, wijst Hlinený er op dat het MACEK programma niet speciaal voor dit bewijs geprogrammeerd is, maar dat het een algemene toolkit voor berekeningen

[10] De enige uitzonderlijke priemgetallen zijn 2, 3, 7, 13, 19, 31, 37, 43, 61, 67, 79, 127 en 283.

[11] [Lehmer1962] p. 408: '[Even] the verification of these results using the data supplied by the machine would be far too long and hazardous a calculation to do by hand.'

[12] [Lehmer1962] p. 407: 'The referee comments that the proof of Theorem 1, described below, is "not a machine proof in the sense of the theorem-proving programs now being developed." This is true.' Lehmer gaat verder en heeft het over de stand van zaken in het toen prille automated theorem proving onderzoeksdomein: 'The aim of most writers on this subject is to consider a very general program enabling a digital computer to prove a wide class of theorems at a vary low level, beginning with the axioms, setting its own goals, and trying to achieve them without human intervention. This is, in a way, a simulation problem. Speculations about such programs involve (significantly) such notions as decidability. Meanwhile, no really new theorems seem to emerge. Perhaps too much is expected of a single program.'

[13] [Hlinený2002] p. 6, bewijs van lemma 4.4: 'Verifying this lemma is clearly a matter of a finite case check. We have done the case analysis with help of the computer program MACEK.'

met matroides is.[14] De berekening duurde ongeveer een dag op een AMD DURON 800 MHz computer (in 2002 een normale thuiscomputer) onder GNU/LINUX. Het MACEK-programma werd gecompileerd met de GCC 2.96-compiler. Hlinený verifieerde de resultaten door de berekening over te doen op verschillende combinaties van hardware- en softwareplatforms: een SOLARIS-systeem op een SPARC-computer, een OSF-systeem op een ALPHA-computer en een NETBSD-systeem op een INTEL-computer. Bovendien werd het programma gecompileerd met andere versies van de compiler GCC: 2.7, 2.8 en 2.95.[15] Al deze verschillende combinaties moesten ervoor zorgen dat eventuele fouten aan het licht kwamen door verschillende resultaten.

Kris Coolsaet en Jan Degraer bewezen in 2005 door een exhaustieve zoektocht dat de Perkelgraaf uniek, is.[16] De Perkelgraaf is gedefinieerd als de graaf die aan een bepaalde verzameling van voorwaarden voldoet. Het computerbewijs van Coolsaet en Degraer genereert alle mogelijke grafen die aan die voorwaarden voldoen (zogenaamde pseudo-Perkelgrafen) en bewijst voor elk van deze grafen dat ze isomorf zijn met de Perkelgraaf.[17] Opvallend is dat Coolsaet en Degraer twee verschillende methodes beschrijven om de uniciteit van de Perkelgraaf te bewijzen en beide methodes ook geprogrammeerd en uitgevoerd hebben om de resultaten te vergelijken en zeker te zijn dat ze geen programmeerfouten zouden maken. Bovendien werden beide programma's grotendeels door verschillende personen geschreven.[18] Hoewel de auteurs helemaal niet lijken te twijfelen aan de geldigheid van een computerbewijs, nemen ze op deze manier toch meer maatregelen om de kans op fouten te verminderen dan ze waarschijnlijk bij een traditioneel bewijs zouden doen, een voorzichtigheid die we bij al de computerbewijzen uit deze categorie lijken te zien.

[14][Hlinený2002] p. 11: ‘We want to emphasize that the MACEK program we use is a general toolkit for matroid computations, and not a specialized closed program prepared only for one task.’

[15][Hlinený2002] p. 11

[16][Coolsaet2005]

[17][Coolsaet2005] p. 156: ‘The proof of the uniqueness of the Perkel graph depends for a large part on an extensive computer search for all possible distance matrices of pseudo Perkel graphs.’

[18][Coolsaet2005] p. 155: ‘To minimize the risk of computer errors we have used two different methods to establish the same theorem and as an added precaution large parts of the corresponding programs were written by different authors.’

7.3. Numerieke benaderingen van ongelijkheden

Numerieke benaderingsmethodes worden in wiskundige natuurkunde zonder enige filosofische reflectie gebruikt, maar daar is men ook niet zozeer geïnteresseerd in rigoureuze bewijzen. Sommige ‘computerbewijzen’ in de wiskunde die gebruik maken van numerieke benaderingsmethodes zijn tevens sterk geïnspireerd door natuurkundige problemen en ook daar blijkt dat de auteurs vaak niet stilstaan bij vragen over de status van hun bewijs.[19] Deze klasse van bewijzen laten we terzijde liggen, aangezien de grens tussen *bewijs* en *berekening* hier vervaagt. Deze problemen liggen sowieso al op de rand van wiskunde en natuurkunde en kunnen we dus niet aanhalen in de discussie of computerbewijzen empirische elementen in de wiskunde introduceren.

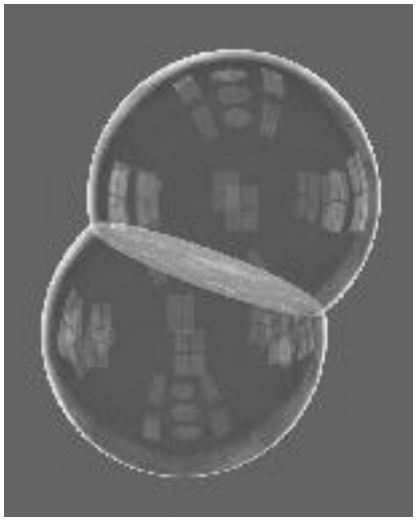
In 1985 bewezen MacKay en Percival met intervalrekenkunde dat er geen invariante torussen bestaan.[20]Een berekening in intervalrekenkunde geeft als resultaat een *interval* waarin de oplossing zich gegarandeerd bevindt. Ook al heb je geen precieze oplossing, de onder- en bovengrens van het interval garanderen wel grenzen waarbinnen de oplossing ligt. Intervalrekenkunde zou hierna een belangrijke techniek worden in computerbewijzen van heel wat belangrijke stellingen, onder andere Hales' bewijs van het Keplervermoeden. In 1995 kondigden Joel Hass en Roger Schlafly aan dat ze een bewijs hadden voor het dubbele-zeepbelvermoeden.[21] Dit vermoeden ging als volgt: van alle oppervlakken in de driedimensionale ruimte die twee gelijke volumes omvatten, heeft de dubbele zeepbel de kleinste oppervlakte. De dubbele zeepbel bestaat uit twee identieke bollen die aan elkaar ‘plakken’, waarbij hun raakvlak een vlakke schijf is. De Belgische fysicus J. Plateu verzamelde in het midden van de negentiende eeuw empirisch bewijsmateriaal voor dit vermoeden. Een wiskundig bewijs voor het vermoeden was er echter niet. Hass en zijn collega's probeerden het met een directe computationele aanpak. Zij konden met analytische en meetkundige methodes de ruimte van oppervlakken die twee gelijke volumes omvatten, die uit oneindig veel dimensies bestaat, reduceren tot een unie van verzamelingen met een eindig aantal dimensies en verder tot een compacte tweedimensionale verzameling. De ene dimensie correspondeert met een hoek en de andere met een gemiddelde kromming. Deze rechthoek van oplossingen wordt dan onderverdeeld in 15016 kleinere rechthoekjes waarop een eindig aantal berekeningen wordt uitgevoerd, in totaal 51256 numerieke integralen. Alle berekeningen gebeuren met intervalrekenkunde om afrondingsfouten tegen te gaan.[22]

Hass en Schlafly geven in het begin van hun bewijs een aantal opmerkingen over de status van hun bewijs en verwijzen naar andere computerbewijzen.[23] Hun verantwoording

[19]Enkele voorbeelden van deze bewijzen zijn het bewijs door Feffermann en De La Llave dat de materie stabiel is en het bewijs door Hassard en collega's dat de Lorenzvergelijkingen chaotische oplossingen hebben.
[20][MacKay1985]
[21][Hass1995]
[22][Hass2000] p. 462

van de keuze voor een computer voor een deel van het bewijs is pragmatisch: de berekeningen zijn te talrijk om met de hand te doen. Volgens de auteurs zullen zulke numerieke technieken in de toekomst nog een grotere rol gaan spelen in meetkundige bewijzen.[24] De auteurs stippen nog aan dat ze het programma op verschillende machines hebben uitgevoerd, met identieke resultaten en dat de broncode van het programma op hun website beschikbaar is.[25] De berekeningen duurden in 1999 op een snelle pc slechts 10 seconden.

Figuur 7.1. Een dubbele zeepbel



In 2002 gebruikte Uri Zwick een computer om een aantal ongelijkheden in sferische volumes te bewijzen.[26] Hij ontwikkelde daar zelf software voor: het programma REALSEARCH. Hij raadt de lezer van zijn bewijs aan dit programma te onderzoeken, aangezien het onderdeel uitmaakt van het bewijs.[27] Met het oog hierop heeft Zwick ervoor gekozen REALSEARCH met heel eenvoudige, naïeve technieken te programmeren, zodat het programma eenvoudig na te kijken is.[28] Zwicks programma REALSEARCH maakt gebruik van intervalrekenkunde om afrondingsfouten uit te sluiten. Zwick besluit zijn artikel met enkele opmerkingen over het statuut van zijn bewijs.[29] Hij vermeldt dat je er niet zeker van kan zijn dat het bewijs door de computer correct is uitgevoerd omdat er fouten in de hardware of software kunnen aanwezig zijn. Hij vermeldt ook dat computerbewij-

[23][Hass2000] p. 461: 'The arguments presented in this paper are a mixture of geometrical analysis and of estimates of geometric quantities obtained by the use of numerical computation. We perform these calculations with strict estimates on the accuracy of the computations. Since it is still somewhat unusual in a mathematical proof to use digital computers to do calculations involving real numbers, we will say a few words about the nature of this part of the argument.'

[24][Hass2000] p. 462

[25][Hass2000] p. 512

[26][Zwick2002]

[27][Zwick2002] p. 503: 'The source code of REALSEARCH, and of the specific procedures used to verify Lemma 4.5 and the other required lemmas, should be considered as part of our proofs.'

[28][Zwick2002] p. 498: 'The main design criterion of REALSEARCH was simplicity. REALSEARCH uses, therefore, very naive techniques that are easily verified and is very concise, only several hundred lines of C++ code.'

zen meestal te lang zijn en minder inzicht in het probleem geven dan een traditioneel bewijs. Hij besluit echter dat het toch om een bewijs gaat.[30]

In 2003 publiceerde de *Annals of Mathematics* een bewijs over hyperbolische 3-variëteiten. Het bewijs van Gabai, Meyerhoff en Thurston maakte uitgebreid gebruik van computermethodes.[31] Een deel van het bewijs gaat ruwweg als volgt: mogelijke oplossingen zitten in een parameterruimte. De parameterruimte is echter niet-compact, waardoor computerverificatie van de oplossingen onmogelijk is. Daarom bewijzen de auteurs eerst dat de niet-compacte parameterruimte kan gereduceerd worden tot een compacte parameterruimte. Deze wordt onderverdeeld in subruimtes en alle subruimtes, uitgezonderd zeven specifieke ruimtes, worden geëlimineerd. Een subruimte wordt geëlimineerd als een bepaalde functie over de hele subruimte begrensd is. Dit wordt voor al deze subruimtes met een computerprogramma benaderend berekend. De auteurs gebruiken geen intervalrekenkunde[32], maar een rigoureuze analyse van de afrondingsfouten bewijst dat de benadering gegarandeerd goed genoeg is. Bewijzen van verschillende stellingen in het artikel komen neer op het verwijzen naar resultaten van twee computerprogramma's, VERIFY en CORONOA.[33] De auteurs gaan niet licht over het gebruik van computerprogramma's in hun bewijs en verdedigen dit gebruik uitgebreid.[34] Zij geven de volgende redenen om in hun bewijs te geloven:

1. Het niet-computergedeelte van het bewijs is op de traditionele manier nagekeken door de auteurs, referees en in seminaries.
2. De computerprogramma's kunnen gecontroleerd worden net zoals wiskundige bewijzen kunnen gecontroleerd worden, wat ook gebeurd is. De computerprogramma's zijn bovendien conceptueel eenvoudig.
3. De enige wiskundige operatoren die het computerprogramma uitvoert zijn +, -, ×, / en √.
4. De computerprogramma's werden op verschillende machines uitgevoerd met verschillende compilers en verschillende computerarchitecturen en de resultaten werden vergeleken.
5. De resultaten van de programma's werden vergeleken met de resultaten van verschillende *common-sense* methodes. Zo werden een groot aantal gegevenspunten geanaly-

[29]Zwick vermeldt enkele andere voorbeelden van computerbewijzen, maar hij vermeldt ook verkeerdelijk Hales' bewijs van het honingraatvermoeden, dat (tot grote verbazing van Hales zelf) relatief kort is en *geen* computerberekeningen nodig heeft. Zie [Hales2001] voor het bewijs en [Hales2000] voor een goede uitleg.

[30][Zwick2002] p. 504: 'Nevertheless, we feel that the use of the word "proof", as made in this paper, is justified.'

[31][Gabai2003], de computerdelen worden vooral besproken in secties 5 tot 8, pagina's 387 tot 427

[32]Dit zou te traag zijn door het grote aantal subruimtes in hun probleem, aldus [Gabai2003] p. 406

[33][Gabai2003] p. 338: 'The proofs of Propositions 1.28, 2.8, and 3.2 amount to having VERIFY and CORONOA analyze several computer files. These computer files are also available at the *Annals* web site.'

[34][Gabai2003] p. 339-341, te beginnen met 'We pose the simple question: why should one have confidence in our proof?'

seerd met een meetkundige aanpak en werden moeilijke gevallen gecontroleerd door een ander programma dat op een ander platform (MACINTOSH) draaide. De referees controleerden ook de robuustheid van de gegevens door het programma VERIFY met een slechtere precisie te laten rekenen en dit tastte de resultaten niet aan.

6. Het bewijs bevat heel wat interne consistentie dat extra redenen geeft om in de correctheid ervan te geloven.[35]

De auteurs merken op dat de referees de programma's nauwkeurig hebben nagekeken en op deze manier wilden *begrijpen* wat er in het bewijs omgaat.[36]

7.4. Axiomatische bewijsproblemen

Ook in de categorie van bewijzen door automatische bewijsprogramma's, zoals McCunes bewijs van het Robbinsprobleem, bestaan er heel wat bewijzen die in de schaduw van dat ene bewijs staan. Het gaat hier meestal om bewijzen waar gevraagd wordt of een bepaalde verzameling van axioma's voldoende is om een bepaalde vergelijking uit af te leiden. Het bewijsprogramma zoekt dan zelf naar een afleiding van de vergelijking. We verwezen al naar OTTERs bewijs dat er een kort axioma bestaat dat in zijn eentje de Booleaanse algebra axiomatiseert.[37] Op dat soort problemen zijn OTTER en andere bewijsprogramma's de laatste tien jaar met succes heel wat losgelaten. Wos beschrijft in zijn boek *Automated reasoning and the discovery of missing and elegant proofs* (2003) hoe hij met automatische bewijsprogramma's bewijzen vond van nog niet-bewezen stellingen en bewijzen die zelfs eleganter zijn dan de tot dan toe bekende bewijzen van stellingen.

Een ander voorbeeld van dit soort bewijzen vinden we bij Johan Belinfante, die automatische bewijsprogramma's inzet om stellingen over ordinaalgetallen te bewijzen.[38] Belinfante gebruikte hiervoor McCunes bewijsprogramma OTTER. Hij gaf het programma een variant van Gödels eindige axiomatisatie van de Neumann-Bernays klassentheorie en in deze formalisatie bewees het programma enkele fundamentele stellingen in ordinaalgetaltheorie, zoals de welordeningsstelling, transfiniete inductie, de classificatie van ordinaalgetallen en de Burali-Forti stelling. Belinfante besloot om noch het regulariteitsaxioma, noch het keuzeaxioma toe te laten in de bewijzen. Beide axioma's zijn niet nodig, maar de bewijzen die Belinfante manueel vond zonder van deze twee axioma's gebruik te maken, waren naar eigen zeggen ‘fairly tricky’. Eén van de motivaties om een bewijsprogramma op de stellingen los te laten, was te onderzoeken of zo een ‘mooier’ bewijs kon

[35]Dit is te vergelijken met de *extrinstieke evidentie* die Kurt Gödel beschreef voor het geloven in de waarheid van een bepaald axioma. Zie [Horsten2004] p. 92

[36][Gabai2003] p. 342: ‘They also checked the programs in great detail, and approached this task with a desire to understand what was really going on behind the scenes.’

[37][McCune2002]

[38][Belinfante1999]

gevonden worden. Belinfante meldt dat zijn bewijsprogramma op dit vlak zeker succes boekte.[39]

Volgens Belinfante was het gebruik van een bewijsprogramma om deze stellingen in ordinaalgetaltheorie te bewijzen zeer vruchtbaar. OTTER vond niet alleen een aantal mooiere bewijzen van al bekende stellingen, het ontdekte ook een aantal eenvoudige nog niet bekende stellingen. Het gebruik ervan gaf Belinfante ook een beter begrip van de wiskunde.[40] Vaak gaven de door de computer geleverde bewijzen zelfs meer inzicht dan die door mensen. Belinfante bespreekt een situatie waarin een wiskundige een bepaald wiskundig object uit zijn hoed tovert, er dan een aantal eigenschappen van bewijst en dan bewijst dat deze eigenschappen de stelling impliceren. De lezer blijft dan vaak achter met de vraag hoe je aan dit object komt. OTTER bleek problemen te hebben om zo'n bewijzen te leveren, aangezien de complexiteit van dit soort 'ad hoc' objecten vaak zo groot is dat het programma het uitproberen van dit object uitstelt.[41] OTTER probeert namelijk eerst eenvoudige formules uit. Als het bewijs in geschikte lemma's werd onderverdeeld, kwam de computer echter vaak wel met hetzelfde complexe object voor de dag, maar dan van de andere kant. De computer redeneerde van de te bewijzen stelling terug en kwam zo op het complexe object. Een lezer van dit bewijs ziet *waarom* dit object nodig is.[42] Misschien is dit ook vaak de manier waarop de wiskundige zijn magisch object gevonden heeft, maar die hij verbergt door het bewijs in zijn uiteindelijk gepubliceerde vorm van de andere kant te laten beginnen, met het uit zijn hoed getoverde object.

Eén van de redenen waarom je met een bewijsprogramma mooiere of kortere bewijzen van bekende stellingen kan vinden of het aantal vooronderstellingen waaronder de stelling geldt kan verminderen, is dat je al de vooronderstellingen in het bewijsprogramma moet ingeven. Je kan niet, zoals zo vaak gebeurt in menselijke bewijzen, stilzwijgende vooronderstellingen maken over eigenschappen van bepaalde getallen. Deze bewijsprogramma's kunnen op deze manier ontdekken dat een bepaalde stelling onder bredere

[39][Belinfante1999] p. 341: 'We wondered whether an automated reasoning program could meet the challenge of finding such proofs without undue assistance, and if so, would perhaps even cleaner proofs be produced than were found by hand? This was indeed the case; several proofs found by OTTER were shorter and appear to be better organized than those produced by hand.'

[40][Belinfante1999] p. 342: 'In the course of proving Isbell's theorems, a number of other simple theorems were discovered that were previously unknown to the author. Using an automated reasoning system definitely enhances one's understanding of the mathematics being studied if only because one is continually being led to ask many questions that one might otherwise not dwell upon.'

[41][Belinfante1999] p. 345: 'This is a strategy typical of many proofs produced by humans, namely, some object is pulled out of a hat, and then it is shown that this object has various desirable properties. For a long time this proof remained a stumbling block for OTTER because OTTER would assign a high weight to the set y, and therefore would postpone considering it.'

[42][Belinfante1999] p. 345: 'When we were done, what we found was typical of the computer proofs of such theorems, namely, that they proceed in the opposite direction from the hand-produced proofs; the computer starts at the end, and proceeds to go backwards, eventually being led to consider the special object that was introduced without apparent motivation in the hand proof. We believe this is actually a desirable feature for pedagogical purposes; people also may have trouble seeing why one considered some particular object, and would much prefer to be shown how one is inexorably led to consider the object in question.'

voorwaarden geldt dan al bekend is. Belinfante geeft hier enkele voorbeelden van.[43] Wanneer een wiskundige op deze manier met een automatisch bewijsprogramma werkt, kan het programma hem *helpen* om nieuwe bewijzen te ontdekken.

[43][Belinfante1999] p. 374: 'When one is using an automated theorem prover, one is naturally led to ask many questions that one might otherwise not dwell upon. In addition to one's constant preoccupation with finding useful demodulators, the desire to reduce the number of literals in a clause frequently prods one to formulate theorems in a more spartan manner than is commonly found in the literature. For example, mathematicians will often make blanket assumptions that all variables in some collection of theorems should refer to ordinals, and not mention such hypotheses further in the statement of theorems. When presented to a theorem proving program, however, these missing hypotheses do have to be supplied, resulting in unwieldy clauses. So one is immediately led to ask just exactly how many of these extra literals can be deleted from the clauses and still preserve the essence of the theorem.'

Deel II. Inzicht in computerbewijzen

Traditioneel gaat filosofie van de wiskunde vooral over 'foundational issues', maar recent is er ook heel wat interesse voor de filosofie van de wiskundige praktijk. Eén van de centrale onderzoeksdomeinen daarin is *mathematical explanation*: welke wiskundige bewijzen of redeneringen geven inzicht, verklaren waarom een stelling geldt en waarom? In het tweede deel van deze eindverhandeling bekijk ik computerbewijzen vanuit dit perspectief. Een veel gehoorde kritiek is immers, zoals we bij de voorbeelden in het vorige deel gezien hebben, dat computerbewijzen geen inzicht geven in waarom de desbetreffende stelling geldt.

Vooraleer we de inzichtelijkheid van computerbewijzen bespreken, situeer ik dit kort binnen de filosofische theorieën van wiskundige verklaringen, toegepast op bewijzen in het algemeen.[44] Er zijn twee grote filosofische theorieën over wiskundige verklaringen: die van Mark Steiner en die van Philip Kitcher. In Steiners theorie kunnen bepaalde bewijzen tout court heel verklarend genoemd worden.Volgens Steiner is een wiskundig bewijs verklarend wanneer het verwijst naar een 'karakteriserende eigenschap' van een entiteit of structuur die in de stelling vermeld wordt.[45] Uit het bewijs moet dus evident blijken dat het resultaat van die eigenschap afhangt. Een karakteriserende eigenschap is een eigenschap die uniek is aan de entiteit binnen de familie waartoe de entiteit behoort.

In hun kritiek op Steiner volgen Michael Resnik en David Kushner[46] Bas van Fraassen die (als het over *wetenschappelijke* verklaringen gaat) zegt dat een verklaring altijd contextafhankelijk is. Of het evident is dat een te bewijzen resultaat afhangt van een bepaalde karakteriserende eigenschap, is afhankelijk van de subgroep waartoe de wiskundige behoort.[47] Bovendien zijn er bewijzen die in het algemeen als verklarend beschouwd worden, maar waar niet zo gemakkelijk een karakteriserende eigenschap in te identifice-

[44]Een goed recent overzicht van de filosofische studie van wiskundige verklaringen is te vinden in [Mancosu2001], met een aantal case studies.

[45][Steiner1978]: 'Generalizability through varying a characterizing property is what makes a proof explanatory.' Door de karakteriserende eigenschap te veranderen, verkrijg je andere stellingen: 'An *explanatory proof* depends on a characterizing property of something mentioned in the theorem: if we "deform" the proof, substituting the characterizing property of a related entity, we get a related theorem.'

[46][Resnik1987]

[47][Resnik1987] p. 146: 'Whether or not something is evident from a proof is relative to subgroups of the mathematical community, at best. A proof that explains to a mathematical logician may be anything but evident to a topologist.'

ren is.[48] Volgens Resnik en Kushner bestaan er dan ook geen verklarende bewijzen tout court.[49]

Volgens van Fraassen komt een (wetenschappelijke) verklaring vragen overeen met het stellen van een *waarom-vraag*.[50] Resnik en Kushner passen dit toe op wiskundige verklaringen. Of een bewijs als verklarend geldt of niet, hangt af van welke waarom-vraag we stellen.[51] Zogenaamde 'verklarende bewijzen' zijn volgens Resnik en Kushner dan gewoon bewijzen die meer informatie tonen en die dus het antwoord tonen op meer waarom-vragen.[52] Erik Weber en Liza Verhoeven zijn het met Resnik en Kushner eens dat Steiners aanpak niet algemeen genoeg is en geven enkele voorbeelden van bewijzen.[53] David Sandborg beweert echter dat een theorie van wiskundige verklaringen die gebruik maakt van waarom-vragen niet voldoende is.[54]

[48][Resnik1987] p. 149 geeft als voorbeeld Henkins bewijs van de volledigheid van de eerste-orde logica: 'The proof is generally regarded as really showing what goes on in the completeness theorem and the proof-idea has been used ag ain and again in obtaining results about other logical systems. Yet again it is not easy to identify the characterizing properties on which it depends.'

[49][Resnik1987] p. 152: 'In view of the difficulties we have found with this account, we should ask whether there are explanatory proofs.'

[50][VanFraassen1980] p. 134: 'An explanation is an answer to a why-question. So, a theory of explanation must be a theory of why-questions.'

[51][Resnik1987] p. 153: 'Whether or not a given proof counts as an explanation depends upon the why-question with which it is approached. If you simply wanted to know why a result is true (rather than false) and were prepared to accept any proof as an answer then you would count all its proofs as explanatory. But you might want to know more. For instance, in addition to wanting to know why the Pythagorean theorem holds you might want to know why it holds *only* for right triangles. Then not every proof of the theorem will contain an answer for you.'

[52][Resnik1987] p. 154: 'Now the so-called explanatory proofs, the ones which "reveal the heart of the matter", present more information and do so more persicuously than do "nonexplanatory" proofs of the same results. Thus they provide the ingredients for answering more why-questions than other proofs. But they are not explanatory in and of themselves.'

[53][Weber2002]

[54][Sandborg1998] p. 604: 'I will grant that explanations *do* answer why-questions and explanatory evaluations *are* context dependent, but claim that the why-question approach nonetheless misses crucial aspects of certain explanatory evaluations.'

De tweede theorie van wiskundige verklaringen is die van Philip Kitcher, die wetenschappelijke verklaring in het algemeen ziet als de theoretische unificatie van verschillende verschijnselen. Volgens Kitcher geldt hetzelfde voor wiskunde: als we verschillende wiskundige disciplines kunnen zien als concrete instantiaties van algemenere algebraïsche structuren, bekomen we wiskundige verklaringen.[55] Volgens Gregory Chaitin leren we uit de algoritmische informatietheorie echter dat er wiskundige feiten zijn die niet te verklaren zijn. Zij zijn als het ware 'per ongeluk' waar.[56] De niet-inzichtelijkheid van de computerbewijzen van de vierkleurenstelling heeft sommigen zelfs ertoe gebracht om te vermoeden dat de vierkleurenstelling zo'n wiskundig feit is dat per ongeluk waar is.[57]

Er is niet altijd overeenstemming over welke bewijzen nu inzichtelijk zijn en welke niet. In Kitchers behandeling van wiskundige verklaringen is bijvoorbeeld de meest algemene verklaring de beste. Anderen vinden dan weer 'elementaire' verklaringen de beste: als je een stelling in de getaltheorie kan bewijzen zonder te verwijzen naar algemenere concepten, heb je een zogenaamd 'elementair bewijs', dat je zou laten zien waarom de stelling geldt. Zo bespreekt Mancosu de elementaire bewijzen van de wiskundige Alfred Pringsheim in de theorie van analytische functies. Pringsheim zelf ziet zijn elementaire methodes als de 'natuurlijke manier' om bepaalde resultaten uit te leggen die in Cauchy's niet-elementaire theorie verschijnen als 'sensational results of a mysterious mechanism'.[58]

Overigens zijn niet alleen filosofen bezorgd om de inzichtelijkheid van bewijzen en het zijn niet enkel computerbewijzen die de vraag naar inzichtelijkheid opwerpen. Zo beschouwen wiskundigen Kleenes bewijs van de recursiestelling in recursietheorie in het algemeen als een hoogst onintuïtief en niet-inzichtelijk bewijs.[59]

[55]Mancosu bespreekt deze twee theorieën kort in [Mancosu2001] p. 102.

[56]Referentie

[57]Referentie

[58][Mancosu2001] p. 111-112: 'Pringsheim claims that the use of elementary methods corresponds to a gain in perspicuity and captures the essence of the matter.' Mancosu merkt echter op dat Pringsheims aanpak niet echt veel gevolg kreeg: de huidige handboeken van complexe analyse volgen zijn aanpak niet. Waarom dit zo is zou volgens Mancosu interessant zijn om te onderzoeken.

[59][Owings1973] p. 95: 'Since that time other fixed-point theorems have been found with similar proofs. All of these theorems tend to strain one's intuition; in fact, many people find them almost paradoxical. The most popular proofs of these theorems only serve to aggravate the situation because they are completely unmotivated, seem to depend upon a low combinatorial trick, and are so barbarically short as to be nearly incapable of rational analysis.' Owings legt in zijn artikel uit hoe we Kleenes bewijs kunnen beschouwen als een aanpassing van een klassiek diagonalisatie-argument, waarin we meer inzicht hebben.

Net als in wetenschappen is er ook in wiskunde geen overeenstemming of het belangrijkste nu resultaten afleiden of deze resultaten begrijpen is.[60] We zullen verder zien dat dit meningsverschil in de praktijk tot uiting komt in het aanvaarden of verwerpen van (een bepaald type van) computerbewijzen. Voor wie enkel of vooral resultaten tellen, is het redelijk om computerbewijzen aan te nemen als dit toelaat om nieuwe resultaten af te leiden, ook al zijn de resulterende bewijzen niet inzichtelijk. Voor wie veel belang hecht aan de inzichtelijkheid van bewijzen, zullen computerbewijzen nog altijd niet even goed aanvaard worden in de wiskundige praktijk als traditionele bewijzen.[61]

In de rest van deze eindverhandeling geef ik enkele observaties over de inzichtelijkheid van bewijzen en pas deze toe op computerbewijzen. Ik focus mij daar op twee eigenschappen van de desbetreffende bewijzen: de gebruikte concepten in de bewijzen en de structuur van de bewijzen.[62] Beide eigenschappen hebben gevolgen voor de inzichtelijkheid van bewijzen. Ik illustreer dit met heel wat voorbeelden van (computer)bewijzen. Een goede algemene bespreking van de twee doelen van een bewijs, namelijk overtuigen en inzicht geven, is te vinden in het artikel ‘Proving is convincing and explaining’ van Reuben Hersh.[63]

[60]Mancosu citeert de Franse wetenschapsfilosoof Pierre Duhem die zegt dat sommige wetenschappers het doel van een natuurkundige theorie zien als het uitleggen van experimenteel bekomen wetten, terwijl andere wetenschappers het doel van een natuurkundige theorie slechts als een ‘samenvatting’ van kennis zien. ([Mancosu2001] p. 103)

[61]Een alternatieve verklaring is dat deze twee partijen verschillende waarom-vragen stellen.

[62]Steiners theorie van verklaringen in functie van karakteriserende eigenschappen is een voorbeeld van een studie van het probleem in functie van de gebruikte concepten in het bewijs.

[63][Hersh1993]

8. Een classificatie van bewijstechnieken in computerbewijzen

In al de voorgaande voorbeelden van computerbewijzen kunnen we duidelijk verschillende categorieën van bewijzen herkennen. We moeten hierbij wijzen op het verschil tussen bewijs en *bewijstechniek*.[1] Heel wat 'computerbewijzen' bestaan namelijk uit een traditioneel bewijs dat door een menselijke wiskundige is opgesteld en uit een computergedeelte. In het computergedeelte wordt een bepaalde bewijstechniek toegepast en dat is wat ons hier interesseert. Zo kan het ook voorkomen dat een bepaald 'computerbewijs' verschillende bewijstechnieken die van een computer afhangen toepast. Hales' bewijs van het Keplervermoeden is hier een voorbeeld van. Een deel van het computergedeelte bestaat uit een combinatorisch probleem en een ander deel uit een numeriek benaderingsprobleem.[2] We bespreken hier kort de verschillende bewijstechnieken die in computerbewijzen voorkomen, hoe we hen kunnen karakteriseren, welke voorbeelden van computerbewijzen er van gebruikmaken en welke niet-wiskundige (en eventueel filosofisch relevante) eigenschappen van de computerbewijzen we eruit kunnen afleiden. Aangezien het om niet-wiskundige eigenschappen gaat, zullen deze eigenschappen nooit voor elk concreet lid van deze categorie gelden. Het gaat hier om een leidraad.

Wat is het belang van deze classificatie? Zoals ik al in Hoofdstuk 2, *De vierkleurenstelling* schreef, is er heel wat reactie gekomen op het bewijs door Appel en Haken. Hun bewijs heeft de vraag wat een bewijs nu eigenlijk is en of een computerbewijs wel een bewijs is, sterk op de filosofische agenda geplaatst. Het is echter jammer dat de andere bewijzen niet diezelfde filosofische reacties hebben losgeweekt en dat een groot deel van de filosofische literatuur over computerbewijzen nog altijd het bewijs van Appel en Haken als prototypisch computerbewijs ziet. De bespreking hierna maakt echter duidelijk dat dit bewijs slechts het prototype is van een bepaalde klasse van bewijzen, namelijk de combinatorische problemen. EQP's bewijs van het Robbinsprobleem toont bijvoorbeeld aan dat een computerbewijs niet onoverzichtelijk lang moet zijn, een veronderstelling die onder invloed van het vierkleurenprobleem nochtans in heel wat filosofische analyses voorkomt. In recentere filosofische discussies over computerbewijzen krijgt Hales's bewijs

[1]Uiteraard zijn de hier besproken bewijstechnieken niet specifiek voor computerbewijzen, ook in bewijzen door wiskundigen komen ze voor.

[2]Dit is misschien één van de redenen dat de referees zo lang nodig hadden om het bewijs na te kijken.

van het Keplervermoeden ook aandacht[3], maar ook dit bewijs is niet representatief voor alle computerbewijzen.

8.1. Symbolische berekeningen

Karakterisering: Een computerprogramma dat een symbolische berekening uitvoert, voert berekeningen uit op expressies met variabelen. Het programma kan uitdrukkingen vereenvoudigen en uitwerken, identiteiten tussen uitdrukkingen bewijzen of andere taken die een menselijke rekenaar ook kan uitvoeren.

Voorbeelden van computerbewijzen: De grens tussen ‘bewijs’ en ‘berekening’ is hier vaak niet zo duidelijk. Computerbewijzen waar de symbolische berekeningen een erg belangrijke rol speelden, zijn onder andere het bewijs van de stelling van Pappus door Cerutti en Davis en recenter de bewijzen van Berkovich en Riese en Paule en Schneider.

Eigenschappen: Voor veel filosofische problemen zorgt deze bewijstechniek niet. Symbolische berekeningen zijn formele bewijzen en zijn overtuigend voor een wiskundige met kennis van zaken die ze nakijkt. De kans op fouten is meestal verwaarloosbaar, inzoverre de correctheid van de software voldoende gecontroleerd is. Het enige probleem dat er kan zijn is dat het soms om lange berekeningen kan gaan die dus in de praktijk niet zo gemakkelijk kunnen nagekeken worden, wat een probleem voor de inspecteerbaarheid kan zijn.[4] Dit is echter een eigenschap die ook menselijke bewijzen hebben die dezelfde berekeningen zouden uitvoeren. Tot slot geven berekeningen niet vaak *inzicht* in waarom het resultaat geldt.

8.2. Combinatorische oplossingen

Karakterisering: Een combinatorisch probleem gaat over een eindig aantal wiskundige objecten. De structuur van deze bewijstechniek om een stelling te bewijzen gaat als volgt: bewijs dat de stelling waar is indien een eindige verzameling van gevallen een bepaalde eigenschap heeft. Bepaal een exhaustieve lijst van al deze gevallen. Bewijs dat alle gevallen van deze lijst de gevraagde eigenschap bezitten.[5]

Voorbeelden van computerbewijzen: De meeste computerbewijzen die in de filosofische literatuur besproken worden, maken van dit type bewijstechniek gebruik. Alle bekende bewijzen van het vierkleurenprobleem zijn van dit type: de computer test of alle

[3]Bijvoorbeeld [Davies2005].

[4]De berekeningen van het FORMAC-programma van Cerutti en Davis zijn bijvoorbeeld te complex voor een mens.

[5]We zagen deze bewijstechniek al bij de bespreking van het vierkleurenprobleem. Edwart Swart besprak de techniek onder de naam *case testing*.

grafen in een onvermijdelijke verzameling reduceerbaar zijn. Ook Lams bewijs van het niet-bestaan van een projectief vlak van orde 10 valt hieronder: het computerprogramma probeerde alle mogelijke configuraties uit. Ook een deel van Hales' bewijs van het Keplervermoeden bestaat uit een combinatorisch probleem: bewijs voor een verzameling van alle tamme grafen dat ze geen tegenvoorbeeld zijn voor het Keplervermoeden. Gabai, Meyerhoff en Thurston delen de parameterruimte van oplossingen op in een eindig aantal subruimtes en voeren hier een case testing techniek op uit. MacKay en Percival doen hetzelfde in hun bewijs dat er geen invariante torussen bestaan. Ook Lehmers bewijs was hier een voorbeeld van, evenals dat van Hlinený. Ook het bewijs van Coolsaet en Degraer dat de Perkelgraaf uniek is, valt hieronder: hun computerprogramma genereert alle mogelijke grafen die aan de eigenschappen van de Perkelgraaf kunnen voldoen en bewijst dan voor elk van deze gevallen dat de desbetreffende graaf isomorf is met de Perkelgraaf.

Eigenschappen: Aangezien de structuur van deze bewijstechniek heel eenvoudig is, zijn deze bewijzen overtuigend. De meerderheid van de filosofen die dit soort bewijzen besprak is het daarover eens. Zelfs Tymoczko, die het bewijs door Appel en Haken als een nieuwigheid in de wiskunde ziet, beschouwt het bewijs als overtuigend. Deze techniek is uiteraard ook volledig formaliseerbaar. De meeste discussie ging over de inspecteerbaarheid van dit soort bewijzen. Vaak ging het erom dat het aantal gevallen dat geverifieerd moet worden, vrij groot is en dat de berekeningen te lang zijn, zodat dit in principe niet is te inspecteren door een menselijke wiskundige. Een andere kritiek die bij al deze bewijzen terugkomt, is dat zo'n bewijs niet de *reden* geeft waarom de stelling waar is, het geeft je geen inzicht. Om die reden wordt dit soort bewijzen ook al eens ‘lelijk’ genoemd. Deze reacties zijn echter niet beperkt tot computerbewijzen. Toen Gaston Tarry Eulers vermoeden voor n = 6 verifieerde door alle mogelijkheden na te gaan, kreeg hij de kritiek dat het geen bevredigend bewijs was en dat er fouten in konden zitten. De correctheid van deze bewijstechniek hangt natuurlijk af van de correctheid van de verschillende delen ervan: het bewijs dat de stelling waar is indien een eindige verzameling van gevallen een bepaalde eigenschap heeft (dit bewijs is vaak een menselijk bewijs), het bepalen van een exhaustieve lijst van al deze gevallen (dit gebeurde bij het Keplervermoeden met een computerprogramma, dat fouten bleek te bevatten maar waarvan de correctheid nu door Bauer bewezen is) en het bewijs dat elk van deze gevallen de desbetreffende eigenschap bevat (dit is computerwerk en ook hier kunnen fouten voorkomen). Door formele correctheidsbewijzen kan de kans op fouten van dit soort bewijzen vrijwel uitgesloten worden. Bij de vierkleurenstelling is dit gebeurd door Gonthier en bij het Keplervermoeden is dit deel van het bewijs door Bauer geformaliseerd en als correct bewezen.

8.3. Numerieke benaderingen van ongelijkheden

Karakterisering: Voor het bewijzen van een bepaalde ongelijkheid kunnen we gebruik maken van benaderingsmethodes. Hiermee hebben we nog geen rigoureus bewijs dat de ongelijkheid geldt. Als we echter kunnen bewijzen dat de afrondingsfouten binnen bepaalde grenzen gelden die geen invloed hebben op de ongelijkheid, dan bekomen we een

rigoureus bewijs van de ongelijkheid.

Voorbeelden van computerbewijzen: Hales' bewijs van het Keplervermoeden maakt veelvuldig gebruik van deze bewijstechniek. Een hele hoop ongelijkheden die hij nodig heeft in zijn bewijs, bekwam hij door computerberekeningen in intervalrekenkunde. Ook MacKay en Percival gebruikten intervalrekenkunde in hun bewijs dat er geen invariante torussen bestaan, evenals Hass en Schlafly in hun bewijs van het dubbele-zeepbelvermoeden en Zwick in zijn bewijzen van een aantal ongelijkheden in sferische volumes. Gabai, Meyerhoff en Thurston daarentegen gebruiken in hun bewijs over hyperbolische 3-variëteiten andere technieken om te bewijzen dat de afrondingsfouten de benadering dicht genoeg bij de oplossing houden.

Eigenschappen: Deze bewijstechniek is overtuigend en formaliseerbaar, maar kan toch op heel wat tegenstand stuiten. Bij veel mensen leeft nog de gedachte dat de computer gemakkelijk benaderingsfouten kan maken en ze schatten de kans op fouten in dit soort bewijzen hoog in. In de filosofische discussie over computerbewijzen wordt zelden verwezen naar dit soort technieken waarmee men kan bewijzen dat de benaderingsfouten begrensd blijven. De benaderingsfouten zijn dus niet de grote boeman hier. Er zijn wel twee andere mogelijke bronnen van fouten. De ene is menselijk: het bewijs dat de afrondingsfouten begrensd blijven, kan een fout bevatten. De tweede foutenbron ligt in de computer: de software die de benadering berekent, moet zijn werk correct doen. Om zeker te zijn van de correctheid hiervan, hebben veel auteurs van dit soort bewijzen heel wat moeite gedaan: verschillende programma's geschreven en de resultaten vergeleken, de programma's zo eenvoudig mogelijk gehouden zodat ze gemakkelijk inspecteerbaar zijn, enzovoort. De nadruk die vele auteurs van bewijzen hierop leggen, suggereert dat zij de software als de belangrijkste foutenbron beschouwen. Over ‘lelijkheid’ of lengte van dit soort bewijzen hoor je geen klachten, hoewel sommige van deze bewijzen vrij lang zijn.

8.4. Logische bewijstechnieken

Karakterisering: Een computerprogramma dat axioma's en een te bewijzen formule krijgt, zoekt zelf de manier om uit de gegeven axioma's de gevraagde formule te bewijzen. Dit is een vorm van symbolisch redeneren en maakt gebruik van een aantal heuristische principes om in de juiste richting te zoeken.

Voorbeelden van computerbewijzen: McCunes bewijs van het Robbinsprobleem met behulp van het bewijsprogramma EQP is het bekendste voorbeeld dat gebruik maakt van deze bewijstechniek. McCune heeft hier eveneens van gebruik gemaakt in een aantal andere bewijzen en ook Belinfantes bewijzen in het domein van ordinaalgetallen vallen onder deze categorie.

Eigenschappen: De bewijzen die geleverd zijn door dit soort bewijsprogramma's zijn niet altijd even overtuigend voor mensen, omdat de bewijsstappen vaak niet erg intuïtief zijn. De bewijzen zijn wel volledig formeel en daardoor is de correctheid ervan gemakkelijker te verifiëren. Aangezien dit soort bewijzen vaak zelfs niet overdreven lang is

(alleszins vergeleken met bijvoorbeeld de bewijzen van het vierkleurenprobleem), is de inspecteerbaarheid ervan niet zo'n probleem.[6] De elegante bewijzen uit het boek *Automated reasoning and the discovery of missing and elegant proofs* van Wos uit 2003 en Belinfantes bewijzen tonen bovendien aan dat automatische bewijsprogramma's geen lelijke bewijzen als resultaat hoeven te hebben.

8.5. Probabilistische bewijstechnieken

Karakterisering: Een probabilistisch (computer)bewijs is een berekening die zegt dat een bepaalde wiskundige bewering met een bepaalde waarschijnlijkheid waar is. Het gaat hier dus niet om een deductief bewijs dat in principe geldt, maar om een uitspraak die geldt onder voorbehoud van een willekeurig klein te maken foutkans.

Voorbeelden van computerbewijzen: De probabilistische priemtesten van Solovay-Strassen en Miller-Rabin behoren tot deze categorie, evenals Adlemans DNA-procedure om te berekenen dat er geen Hamiltoniaans pad in een bepaalde gerichte graaf aanwezig is.

Eigenschappen: Probabilistische bewijzen zijn door hun eenvoud overtuigend en inspecteerbaar, maar een grote kritiek erop is dat ze geen inzicht geven in waarom hun resultaat geldt. Het enige wat ze zeggen is: de kans dat deze uitspraak geldt is 99,999… % omdat deze reeks van berekeningen geldt. Een voordeel is dat probabilistische bewijzen redelijk kort zijn. Deze bewijzen zijn bovendien perfect formaliseerbaar.

8.6. Conclusie

Als we de eigenschappen van de verschillende categorieën bewijstechnieken in computerbewijzen in een tabel zetten, krijgen we dit (er mee rekening houdend dat het om een leidraad gaat en niet om eigenschappen die absoluut gelden):

[6] Maar nu ook weer niet triviaal.

Tabel 8.1. Eigenschappen van bewijstechnieken in computerbewijzen

Bewijs-techniek	formeel	overtuigend	inspecteerbaar	lengte	deductief	geeft inzicht
Symbolische berekening	ja	ja	ja (indien niet te lang)	neutraal	ja	neutraal
Combinatorische oplossing	formaliseerbaar	ja	zelden (wegens te lang)	vaak lang	ja	nee
Numerieke benadering	formaliseerbaar	ja	ja	neutraal	ja	neutraal
Logische bewijstechniek	ja	ja	ja	kort	ja	neutraal
Probabilistisch bewijs	formaliseerbaar	ja	ja	kort	nee	nee

De verschillende categorieën hebben een breed scala aan combinaties van eigenschappen: elke twee technieken verschillen wel in twee kolommen. Gezien de combinatorische bewijstechniek nog altijd als het prototypisch voorbeeld van een computerbewijs gezien wordt, leidt dit vaak tot veralgemeningen naar computerbewijzen tout court. Je vindt zo bijvoorbeeld geregeld de kritiek dat computerbewijzen te lang zijn, niet na te kijken zijn door mensen en geen inzicht geven.[7] Deze drie eigenschappen gelden voor de meeste computerbewijzen die van een combinatorische bewijstechniek gebruik maken, maar bijvoorbeeld niet voor de andere vier categorieën. Bepaalde bewijzen die van logische bewijstechnieken gebruik maken, zoals Belinfantes bewijzen in ordinaalgetaltheorie, scoren zelfs heel goed op deze drie eigenschappen.

Als we bekijken in hoeverre de verschillende bewijstechnieken inzicht geven in de bewezen stelling, dan zien we dat vooral de combinatorische en probabilistische bewijzen op dit vlak problematisch zijn. Niet toevallig zijn dit de twee categorieën die op het meeste tegenstand konden en kunnen rekenen. Herinneren we ons de uitspraken van Bonsall dat het bewijs door Appel en Haken van de vierkleurenstelling geen inzicht geeft en dat het maar een ‘quasi-bewijs’ is dat we als eerste stap naar een ‘echt’ bewijs moeten zien. Of de uitspraak van Stewart dat het bewijs geen patroon of verborgen structuur toont aan de wiskundige, maar slechts een ‘monstrous coincidence’. Wiskundigen zijn er niet tevreden over omdat het bewijs niet de reden geeft waarom de stelling waar is, het geeft slechts

[7] Bijvoorbeeld [Davies2005].

een verificatie. Net hetzelfde zien we bij probabilistische bewijzen, maar dan sterker omdat het toevalsaspect expliciet in het niet-deductieve karakter van het bewijs tot uiting komt.[8] Wiskundigen accepteren dit soort bewijzen niet omdat ze enkel een verificatie geven, zonder reden, en dan nog een verificatie die geen absolute zekerheid geeft.

De belangrijkste verwezenlijking van dit hoofdstuk is de gegeven categorisatie van bewijstechnieken in computerbewijzen, die in de literatuur over computerbewijzen niet te vinden is. Het in kaart leggen van de eigenschappen van deze bewijstechnieken is daar een uitloper van, maar mag niet te absoluut opgevat worden. Het is echter een goede leidraad die de filosofisch relevante verschillen tussen de verschillende bewijstechnieken duidelijk samenvat. Dit alles was pas mogelijk door het bestuderen van een groot aantal voorbeelden van computerbewijzen en de reacties erop.

[8]Meer hierover in Paragraaf 10.5, “Het toevalsaspect in combinatorische bewijsmethodes” in Hoofdstuk 10, *Bewijsplannen en bewijsschetsen*.

9. Wiskundige concepten in computerbewijzen

9.1. Inleiding

Uit de bespreking van een groot aantal computerbewijzen in het eerste deel blijkt dat heel wat wiskundigen bepaalde van deze bewijzen niet zo hoog inschatten omdat ze geen inzicht geven. We kunnen ons afvragen waarom bijvoorbeeld het bewijs door Appel en Haken, en zelfs dat door Robertson en zijn collega's, van de vierkleurenstelling niet echt inzicht geeft in waarom die stelling waar is. In dit hoofdstuk beargumenteer ik de stelling dat de inzichtelijkheid van een wiskundig bewijs afhangt van de gebruikte wiskundige concepten in het bewijs. Ik laat zien dat heel wat van de besproken computerbewijzen op dit vlak sterk verschillen van menselijke bewijzen. Door het gebruik van concepten op laag niveau, zoals bijvoorbeeld in McCunes bewijs van het Robbinsprobleem, lijkt dit bewijs meer op een berekening die een stelling verifieert dan een echt bewijs. In andere gevallen krijgen we gewoon een lange formele redenering waar weinig structuur in te zien is.[1] Ik besluit dit hoofdstuk dan ook met een raad aan ontwerpers van bewijsprogramma's: gebruik de juiste concepten.

In het eerste deel werd duidelijk dat de meeste wiskundigen tegenwoordig niet zozeer problemen hebben met de betrouwbaarheid van computerbewijzen. Ze hebben vooral vragen bij het feit dat computerbewijzen zo moeilijk te begrijpen zijn, al is het maar omdat vele zo lang zijn. Maar ook bij korte bewijzen zoals McCunes bewijs van het Robbinsprobleem hoor je de verzuchting dat het bewijs niet echt inzicht geeft. In dit hoofdstuk tonen we aan dat het belangrijkste verschil tussen computerbewijzen en ‘goede’ menselijke bewijzen niet het feit is dat de berekening door een computer gebeurt, maar het feit dat de gebruikte concepten in het bewijs verschillen. Een menselijke wiskundige probeert in een bewijs ‘de juiste’ concepten te gebruiken, terwijl de huidige computerbewijzen dat niet doen omdat ze slechts met een beperkt aantal voorgedefinieerde concepten kunnen werken.

Om deze bewering te staven, tonen we eerst enkele eenvoudige menselijke bewijzen van elementaire wiskundige stellingen. We laten telkens verschillende bewijzen van de stelling zien: één met concepten op laag niveau en andere die abstractere concepten gebruiken. We vergelijken dan de inzichtelijkheid van de verschillende bewijzen. Daarna ne-

[1] John Dawson vat het probleem met zulke bewijzen als volgt samen: ‘The issue here is not logical, but *psycho*logical: At some point, for example, despite our belief in the transitivity of logical implication, we lose track of conceptual threads when we are presented with a sufficiently long chain of formal deductions. In informal humanly generated proofs, *lemmas* are used to break such chains up into manageable pieces. They serve a ssignposts to mark important conceptual steps in the proof.’ ([Dawson2006])

men we de stap naar computerbewijzen en bespreken we nog eens een aantal van de in de vorige hoofdstukken besproken computerbewijzen en enkele andere, maar nu met de focus op de gebruikte concepten en de inzichtelijkheid van het bewijs.[2] We besluiten het hoofdstuk met resultaten van de onderzoeksgroep van Bundy over programma's die zelf hoogniveau concepten kunnen 'ontdekken'.

9.2. Bewijzen en concepten

We bespreken hier enkele eenvoudige stellingen en van elke stelling een aantal bewijzen. De bewijzen verschillen in de concepten die ze gebruiken. We tonen aan dat de bewijzen die concepten op hoger niveau gebruiken vaak inzichtelijker zijn en dus beter tonen *waarom* de stelling waar is.[3] Dat is ook één van de redenen waarom wiskundigen blijven zoeken naar nieuwe bewijzen van een stelling die ze al bewezen hebben, zoals Dawson ook beschrijft in [Dawson2006].[4] Dit dient als opstapje naar de rest van het hoofdstuk, waarin we argumenteren dat de huidige computerbewijzen vaak niet zo inzichtelijk zijn omdat ze concepten op laag niveau gebruiken.

9.2.1. Cassini's identiteit bij Fibonaccigetallen

De Fibonaccigetallen F_n zijn gedefinieerd als de rij natuurlijke getallen waarvoor geldt: $F_0 = 0$, $F_1 = 1$ en $F_n = F_{n-1} + F_{n-2}$ voor $n > 1$. Voor deze reeks getallen zijn heel wat mooie resultaten bekend, onder andere deze stelling:

[2]Overigens merkt Bassler in [Bassler2006] p. 104 al op dat een bewijs met de juiste concepten voor een betere globale inspecteerbaarheid zorgt en dus (volgens zijn definitie van globale inspecteerbaarheid) tot meer inzicht: 'Global surveyability is improved by conducting a proof with objects of the right level.'

[3]Deze aanpak van een aantal case study's van bewijzen om iets aan te tonen over de waarde die we toekennen aan bepaalde bewijzen, vinden we ook terug in [Avigad2006]. Jeremy Avigad illustreert de criteria die we daarvoor gebruiken met drie stellingen die elk verschillende bewijzen krijgen. Twee van deze stellingen en de bijbehorende bewijzen heb ik hier opgenomen omdat de bewijzen het verschil in concepten goed aantonen. Avigad wil er andere zaken mee aantonen, die er wel verband mee houden, onder andere dat bewijzen die algemener gelden interessanter zijn.

[4][Avigad2007]: 'The fact that there is a gap between knowledge and understanding is made pointedly clear by the fact that one often finds dozens of published proofs of a theorem in the literature, all of which are deemed important contributions, even after the first one has been accepted as correct. Later proofs do not add to our knowledge that the resulting theorem is correct, but they somehow augment our understanding.' Bruno Ernst verzamelt in [Ernst2002] de 'interessantste' bewijzen van de stelling van Pythagoras en geeft bij verschillende bewijzen de opmerking dat ze inzichtelijk of net niet inzichtelijk zijn. Zo zegt Ernst bij een bepaald bewijs '[De clou] komt pas op het eind als een duveltje uit een doosje tevoorschijn, terwijl het bij [het bewijs van] Euclides van meet af aan zichtbaar is aan de vierkanten op de drie zijden van de rechthoekige driehoek.' Volgens Ernst bevat de uitgebreidste collectie wel 370 bewijzen van de stelling van Pythagoras.

Vergelijking 9.1. Cassini's identiteit

$$F_{n+1}F_{n-1} - F_n^2 = (-1)^n \text{ voor } n \geq 1$$

Van Cassini's identiteit zijn heel wat verschillende bewijzen bekend. Drie standaardbewijzen die je overal vindt zijn de volgende:

9.2.1.1. Bewijs door inductie

Bewijs: We bewijzen door inductie dat de identiteit geldt voor alle n ≥ 1. Als n = 1, vullen we de waarden van F_0, F_1 en F_2 in en bekomen we $1\times0 - 1^2 = (-1)^1$, wat klopt. Veronderstel nu dat de stelling geldt voor n = m, met m ≥ 1. We bekijken dan of de stelling waar is voor m + 1:

$$F_{m+2}F_m - F_{m+1}^2 = (F_{m+1} + F_m)F_m - (F_m + F_{m-1})^2$$

$$= F_{m+1}F_m + F_m^2 - F_m^2 - 2F_mF_{m-1} - F_{m-1}^2$$

$$= F_{m+1}F_m - 2F_mF_{m-1} - F_{m-1}^2$$

$$= (F_m + F_{m-1})F_m - 2F_mF_{m-1} - F_{m-1}^2$$

$$= F_m^2 + F_{m-1}F_m - 2F_mF_{m-1} - F_{m-1}^2$$

$$= F_m^2 - F_mF_{m-1} - F_{m-1}^2$$

$$= F_m^2 - (F_m + F_{m-1})F_{m-1}$$

$$= F_m^2 - F_{m+1}F_{m-1}$$

$$= -(-1)^m = (-1)^{m+1}$$

Door onze inductiehypothese geldt de vergelijking dus ook voor n = m+1. QED

9.2.1.2. Bewijs door Binets formule

Een conceptueel iets geavanceerder bewijs kunnen we bekomen door Binets formule voor de Fibonaccigetallen te gebruiken. Het blijkt namelijk dat de Fibonaccigetallen, die we hierboven recursief gedefinieerd hebben, ook berekend kunnen worden met deze for-

mule:

$F_n = (\phi^n - (1 - \phi)^n)/\sqrt{5}$

Het getal ϕ in deze formule is de *gulden snede*, gelijk aan $(1 + \sqrt{5})/2$ of ongeveer 1,618. Het bewijs van Binets formule kan eenvoudig geleverd worden door inductie.

Bewijs: Als we in Cassini's identiteit Binets formule substitueren voor de Fibonaccigetallen en de formule vereenvoudigen, bekomen we $(\phi - \phi^2)^n$. De expressie $\phi - \phi^2$ is echter gelijk aan -1, omdat ϕ een oplossing is van de vergelijking $x^2 - x - 1 = 0$. QED.

Petkovsek, Wilf en Zeilberger vermelden dat heel wat identiteiten met Fibonaccigetallen eenvoudig met wat routinewerk kunnen bewezen worden door gebruik te maken van Binets formule. Ze geven als voorbeeld Cassini's identiteit en tonen de commando's om deze in het computerprogramma MAPLE te bewijzen.[5]

9.2.1.3. Bewijs door determinant

We merken eerst op dat de linkerkant van de vergelijking de determinant is van een 2 bij 2 matrix van Fibonaccigetallen. Die twee bij twee matrix kan voorgesteld worden als de n-de macht van een matrix met determinant -1 (dit is een bekende stelling die door inductie kan bewezen worden), wat het bewijs van de stelling oplevert:

$$F_{n-1}F_{n+1}-F_n^2 = \det\begin{bmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{bmatrix} = \det\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n = \left(\det\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\right)^n = (-1)^n.$$

9.2.1.4. De concepten in de bewijzen

Als we de drie bewijzen vergelijken, zien we een duidelijk verschil in de strategie. Het eerste bewijs maakt enkel gebruik van de definitie van Fibonaccigetallen en wat algebraïsch rekenwerk. Inductie doet de rest. Het is het soort bewijs dat een leerling van de middelbare school die geïnteresseerd is in Fibonaccigetallen zou vinden. Er komt niet echt inzicht aan te pas, eerder wat handigheid met formules. Door dit bewijs heb je ook niet echt het idee dat je weet *waarom* Cassini's identiteit nu geldt. Je ziet na dat rekenwerk dat termen tegenover elkaar wegvallen en dat het uiteindelijk mooi uitkomt, maar er zit niet echt een lijn in.

Het tweede bewijs gebruikt al iets meer concepten. Er wordt een andere formule voor de Fibonaccigetallen gebruikt, maar de rest is ook gewoon rekenwerk. Het blijft dus een routinematig bewijs, dat zelfs een computer kan oplossen, zoals het voorbeeld in MAPLE aantoont. Ook hier zie je niet echt waarom de identiteit geldt.

[5] [Petkovsek1996] p. 12

Het derde bewijs is helemaal anders en is gebaseerd op heel wat meer concepten. Naast de definitie van Fibonaccigetallen gebruik je de matrixvoorstelling van Fibonaccigetallen in de vorm van de n-de macht van de matrix `[[1, 1], [1, 0]]`. Door het matrix-concept te gebruiken hebben we ineens de beschikking tot het krachtige determinantconcept en kunnen we zien dat de linkerkant van de identiteit, die uit Fibonaccigetallen bestaat, eigenlijk te beschouwen is als een determinant van een matrix. Na dit inzicht volgt de rest van het bewijs triviaal.[6] Het rekenwerk is vrijwel nihil, je gebruikt gewoon stap voor stap enkele eigenschappen. Uiteindelijk *zie* je in dit bewijs ook waarom de stelling geldt, aangezien je nu weet dat de linkerkant van de identiteit een determinant is. De -1 en de n-de macht aan de rechterkant van de identiteit betekenen nu iets, je ziet waarom die in de identiteit voorkomen. Dit soort bewijs zal een leerling van de middelbare school niet vlug maken.[7]

Hier zien we al een punt geïllustreerd dat Michael Beeson maakt wanneer hij het heeft over de symbolische rekenenvaardigheden van computerprogamma's. Er zijn natuurlijk takken van de wiskunde die vooral uit berekeningen bestaan. De theorie van Fibonaccigetallen is daar een duidelijk voorbeeld van. Dit kan ons de indruk geven dat alle wiskunde slechts uit berekeningen bestaat: je hebt geen diepe wiskunde nodig om Cassini's identiteit te bewijzen.[8] Wat meer wiskundige concepten maken de identiteit echter wel duidelijker. Ga je naar complexere takken van de wiskunde, dan heb je nog meer concepten nodig en dan heb je niet genoeg aan eenvoudige berekeningen, maar moet je ook je toevlucht nemen tot redeneringen over abstracte concepten.[9]

9.2.2. Fermatpriemgetallen

Jeremy Avigad geeft dit voorbeeld in zijn ‘Mathematical method and proof’[10]: de Franse wiskundige Pierre de Fermat observeerde dat de getallen $2^1 + 1$, $2^2 + 1$, $2^4 + 1$, $2^8 + 1$ en $2^{16} + 1$ (dus waarbij de exponent telkens een volgende macht van 2 is) allemaal priemgetallen zijn. In 1640 uitte hij het vermoeden dat alle natuurlijke getallen van deze vorm

[6]We zien hier een voorbeeld van een probleem in zijn juiste theoretische context, zoals Horsten dat beschrijft in [Horsten2001] p. 187: ‘Kenneth Manders has rightly emphasized that *mathematical understanding* of a problem can only be generated when the problem is investigated in its proper theoretical setting. The *creation* of this setting is often the main ingredient in the acquisition of mathematical insight into the problem.’

[7]Dit illustreert ook dat inzichtelijkheid contextgebonden is. Een leerling van de middelbare school zal zijn ‘elementair’ bewijs door inductie waarschijnlijk inzichtelijker vinden dan het bewijs via matrixtheorie, een punt dat Dirk Schlimm maakte tijdens de *Perspectives on Mathematical Practices* conferentie.

[8][Beeson2003]: ‘Obviously there are some parts of mathematics that consist mainly of computations. The fact is that this part of mathematics includes high-school mathematics and first-year calculus as it is usually thaught, so that people who do not study mathematics beyond that point have the (mis)-impression that mathematics consists of yet more complicated calculations. That is not true.’

[9][Beeson2003]: ‘Beginning with the course after calculus, mathematics relies heavily on proofs. Some of the proofs contain some steps that can be justified by calculation, but more emphasis is placed on precisely defined, abstract concepts, and the study of what properties follow from more fundamental properties by logical implication.’

[10][Avigad2006]

priemgetallen zijn. In 1738 bewees Euler echter dat het volgende getal in de rij, $2^{32} + 1$, deelbaar is door 641 en dus geen priemgetal is. Euler geeft echter oorspronkelijk geen hints over hoe hij dit gevonden heeft.[11] Ook hier zijn een aantal bewijzen mogelijk:

9.2.2.1. Bewijs door berekening

Bewijs: Een berekening toont aan dat $2^{32} + 1 = 4294967297 = 641 \times 6700417$. QED

9.2.2.2. Bewijs door modulaire congruenties

Gauss introduceerde in zijn boek *Disquisitiones Arithmeticae* van 1801 het concept van *congruentie*. Twee gehele getallen x en y zijn congruent modulo z, genoteerd $x \equiv y \pmod{z}$. als z een deler is van x - y. Met deze notatie, en het achterliggende nieuwe concept, wordt het volgende bewijs mogelijk:

Bewijs: 641 is $5 \times 2^7 + 1$, dus $5 \times 2^7 \equiv -1 \pmod{641}$. Als we beide kanten van de congruentie tot de vierde macht verheffen, krijgen we: $5^4 \times 2^{28} \equiv 1 \pmod{641}$. We hebben echter ook $641 = 5^4 + 2^4$, dus $5^4 \equiv -2^4 \pmod{641}$. Als we beide kanten van de congruentie vermenigvuldigen met 2^{28}, dan krijgen we $5^4 \times 2^{28} \equiv -2^{32} \pmod{641}$. Als we de tweede en de vierde congruentie samennemen, krijgen we $1 \equiv -2^{32} \pmod{641}$, of met andere woorden: 641 is een deler van $2^{32} + 1$. QED

9.2.2.3. Bewijs door eigenschappen van delers

In 'Theoremata circa divisores numerorum' (1750) bewijst Euler dat als x en y geen gemeenschappelijke deler hebben buiten 1 en -1, elke factor van $x^m + y^m$ met $m = 2^n$ ofwel 2 is ofwel van de vorm $2^{n+1}k + 1$. Dit impliceert dat $2^{32} + 1$, als het geen priemgetal is, een factor van de vorm 64k + 1 moet hebben. Als je de opeenvolgende waarden van k afgaat en controleert of het resulterende getal een deler is van 2^{32}, ontdek je zo bij k = 10 de deler: 641. QED[12]

9.2.2.4. De concepten in de bewijzen

Het eerste bewijs (in essentie wat Euler de eerste keer gaf) is heel kort en bijgevolg kunnen we het eenvoudig nakijken: vermenigvuldig gewoon de twee getallen aan de rechterkant en verifieer of het product gelijk is aan 4294967297.[13] We zien hierdoor echter hele-

[11][Euler1738] in de Engelse vertaling: 'I have observed this for a time, and by a long route delivered this number to be able to be divided by 641' Een goed overzicht van de wiskunde van Fermatgetallen is te vinden in [Krízek2001].

[12][Euler1750] in de Engelse vertaling: 'And so as I had wanted to examine the truth of this renowned claim of Fermat for the case of $2^{32} + 1$, I managed a great shortening of this, by not having to try division by any prime numbers except those which the formula 64n+1 produces. And so with the problem reduced to this, I soon discovered that, by setting n = 10, the prime number 641 is a divisor of the number $2^{32} + 1$.'

[13][Avigad2006]: 'Sometimes a proof is nothing more than a calculation. In some contexts, this is optimal: it can provide a straightforward verification, requiring little thought or background knowledge.'

maal niet hoe je aan de twee factoren komt. Het gaat gewoon om een berekening met getallen, die voor de rest geen wiskundige concepten gebruikt.

Het tweede bewijs introduceert het concept congruentie en geeft volgens Avigad een gedeeltelijke verklaring van wat er zo speciaal is aan 641, namelijk het feit dat het getal kan geschreven worden als $5\times2^7 + 1$ en $5^4 + 2^4$. Een ander voordeel is dat het bewijs gebruik maakt van de eigenschappen van machtsverheffing en dus ook gedeeltelijk verklaart waarom die machtsverheffing in de formulering van de stelling belangrijk is. Het eerste bewijs maakte daar helemaal geen gebruik van, maar behandelde het getal $2^{32} + 1$ gewoon als 4294967297.

Het derde bewijs is interessanter dan de eerste twee omdat we hierin zien hoe we kunnen ontdekken dat 641 een deler is van $2^{32} + 1$, in tegenstelling tot in de twee andere bewijzen. Dat dit mogelijk is, is te danken aan het feit dat Euler er algemenere concepten bijhaalt. Daardoor zijn de ideeën uit het bewijs ook breder inzetbaar. Met een kleine aanpassing kan je zo bijvoorbeeld bewijzen dat $2^{16} + 1$ wel een priemgetal is.

Avigad merkt op dat de definities (concepten) van deelbaarheid en congruentie in de bewijzen een belangrijke rol spelen. Zulke definities laten namelijk toe om redeneermethodes uit andere domeinen over te dragen of om het probleem in een algemener kader te zien. Dit laatste is als volgt in te zien: congruentie modulo een getal is een *equivalentierelatie* en heeft dus een aantal eigenschappen van de gewone gelijkheid. Deelbaarheid heeft dan weer een *partiële ordening* en heeft daardoor een aantal eigenschappen van de $\leq$ relatie.

9.2.3. Producten van sommen van kwadraten

Ook dit voorbeeld komt uit Avigads ‘Mathematical method and proof’.[14] Het gaat om de volgende stelling: Als x en y kunnen geschreven worden als een som van twee kwadraten, dan kan hun product xy dat ook. Een voorbeeld: $5 = 2^2 + 1^2$ en $13 = 3^2 + 2^2$. Het product van 5 en 13 is 65, wat gelijk is aan $8^2 + 1^2$ en $7^2 + 4^2$.

9.2.3.1. Bewijs door berekening

Bewijs: als $x = a^2 + b^2$ en $y = c^2 + d^2$, dan is $xy = (ac - bd)^2 + (ad + bc)^2$. QED

$xy = (ac + bd)^2 + (ad - bc)^2$ komt ook uit. We zagen al in het voorbeeld dat 65 op twee manieren kan geschreven worden als een som van twee kwadraten.

9.2.3.2. Bewijs door Gauss gehele getalen

Het tweede bewijs blijft niet in de gehele getallen, maar maakt een omweg langs de *Gauss gehele getallen*, dit zijn complexe getallen van de vorm $a + bi$ waar a en b ge-

[14] [Avigad2006]

hele getallen zijn en i de vierkantswortel van -1. Voor een complex getal $\alpha = u + vi$ definiëren we het complex toegevoegde α' als $u - vi$. We definiëren dan de norm $N(\alpha)$ van een complex getal α als $\alpha\alpha'$. De norm is net als het complex toegevoegde multiplicatief: we hebben $(\alpha\beta)' = \alpha'\beta'$ en we hebben $N(\alpha\beta) = N(\alpha)N(\beta)$. Als $\alpha = a + bi$ een Gauss geheel getal is, dan is $N(\alpha) = a^2 + b^2$ een geheel getal. De getallen die te schrijven zijn als de som van twee kwadraten zijn dus de normen van Gauss gehele getallen. Met deze conceptuele bagage kunnen we onze stelling kort bewijzen:

Bewijs: Als $x = N(\alpha)$ en $y = N(\beta)$ sommen van twee kwadraten zijn, dan is $xy = N(\alpha\beta)$ een som van twee kwadraten. QED

9.2.3.3. Bewijs door complexe getallen

Bewijs: Stel dat $x = a^2 + b^2$ en $y = c^2 + d^2$. Dan is:

$$xy = (a^2 + b^2)(c^2 + d^2)$$

$$= (a + bi)(a - bi)(c + di)(c - di)$$

$$= (a + bi)(c + di)(a - bi)(c - di)$$

$$= ((ac - bd) + (ad + bc)i)((ac - bd) - (ad + bc)i)$$

$$= (ac - bd)^2 + (ad + bc)^2$$

Dit bewijs geeft Euler zelf in 1770 in zijn werk *Algebra*.[15]

9.2.3.4. De concepten in de bewijzen

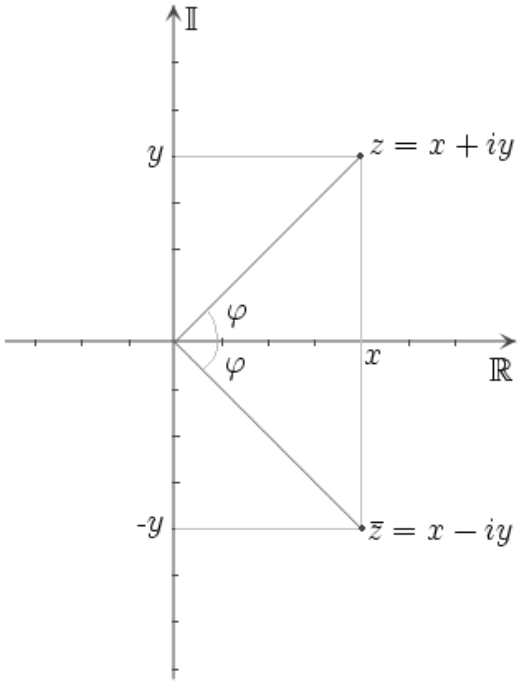
Het eerste bewijs is een eenvoudige berekening, waarbij we enerzijds gebruik maken van de commutativiteit en associativiteit van optellen en vermenigvuldigen en anderzijds de distributiviteit van vermenigvuldigen over optellen en aftrekken.[16]

Het tweede bewijs is conceptueel veel rijker: het maakt gebruik van het concept norm van een complex getal, waardoor de stelling in verband kan gebracht worden met heel wat andere domeinen. Zo correspondeert de positieve vierkantswortel van de norm van een complex getal met de afstand tot de oorsprong van het corresponderende punt in het Euclidische vlak:

[15][Avigad2006]

[16]Avigad merkt op dat de eenvoud van de berekeningen een inzicht met zich meebrengen: dit toont aan dat de stelling in elke *commutatieve ring* geldt, niet enkel voor de natuurlijke getallen. ([Avigad2006])

Figuur 9.1. Het complexe vlak



Door dit concept norm kan het tweede bewijs ook veralgemeend worden.[17] Als we de tweedimensionale algebra van de complexe getallen vervangen door de vierdimensionale algebra van de quaternionen, dan bekomen we een productregel voor sommen van vier kwadraten. We kunnen zelfs nog generaliseren naar de achtdimensionale algebra van de octonionen, met als resultaat een productregel voor sommen van acht kwadraten.[18]

Het derde bewijs komt sterk overeen met het tweede, maar definieert niet het concept norm. In essentie voert het wel dezelfde stappen uit. Een voordeel van dit bewijs ten opzichte van het tweede is dat we zien dat we de termen ook op een andere manier kunnen ordenen, als `xy = (a + bi)(c - di)(a - bi)(c + di)`, met als resultaat een tweede manier om het product als een som van twee kwadraten te schrijven: `(ac + bd)^2 + (ad - bc)^2`. We zien dus dat door een concept ‘te hoog’ te gaan (de norm) we wel een eenvoudiger bewijs krijgen van de stelling, maar door een concept ‘tussenin’ te gebruiken we meer details te weten komen over de manier waarop de stelling geldt. Het is dus niet zo dat meer abstractie absoluut ‘beter’ is: dit hangt af van de specifieke vraag die je stelt.

[17][Avigad2006]

[18]Het is deze mogelijkheid tot generaliseren die Avigad sterk benadrukt in zijn case study's van bewijzen. De mogelijkheidsvoorwaarde tot deze generalisatie zijn echter de gebruikte concepten, waarop ik me in mijn bespreking focus.

9.3. Het optimale niveau van concepten

Met welke concepten we werken in een bewijs heeft een grote invloed op het inzicht dat het bewijs geeft en het gevoel dat we de stelling begrijpen. Als een bewijs algemener is dan een ander, wordt dit meestal als een deugd beschouwd.[19] Het is echter niet zo dat een bewijs dat algemenere concepten gebruikt altijd 'beter' is. Bepaalde feiten worden in een algemenere context niet meer zichtbaar, zoals we in het laatste voorbeeld van de producten van sommen van kwadraten gezien hebben. Generalisatie is goed als het *explicatief* is.[20]

De Joods-Poolse wiskundige Szolem Mandelbrojt[21] schreef in 'Pourquoi je fais des mathématiques' dat elk wiskundig probleem een 'optimaal' niveau van generalisatie heeft. Als we het probleem op dit optimale niveau bewijzen, geeft dit ons het meeste inzicht:[22]

> On sent la nécessité de se placer sur un plan plus élevé, on sait qu'on ne comprendra son vrai caractère que s'il concerne un plus grand nombre d'objets. Il y a un moment où l'ensemble d'objets, auxquels il s'applique, explique le sens même du théorème. Il serait, je crois, just de dire, que c'est après avoir trouvé un théorème, qu'on doit chercher, que d'ordinaire on cherche, le vrai objet, ou le vrai ensemble d'objets, auquel il s'applique. C'est ainsi qu'on obtient la généralisation explicative. Personellement, je sens *qu'il y a un optimum* à cette généralité.

Wanneer het niveau van abstractie correct is, beschrijft ze de wereld van het bestudeerde wiskundige fenomeen op een natuurlijke wijze.[23] Wanneer men te veel abstraheert, komt men volgens Mandelbrojt in de geformaliseerde wereld terecht.[24] De wereld van de formele logica mist volgens hem de 'materie' van de wiskunde.[25] In de hierboven gegeven voorbeelden van bewijzen zien we Mandelbrojts observaties. Het eerste bewijs van Cas-

[19][Mandelbrojt1952] p. 52: 'Le fait d'être général est un grande vertu pour un fait mathématique, et l'on éprouve une sorte de léger mépris pour le fait particulier.'

[20][Mandelbrojt1952] p. 52: 'Je n'arrive pourtant pas à diviniser la généralité en soi. L'affirmation, qu'on pourrait traiter de particulière, que le nombre *e* est transcendant, et surtout sa démonstration, me laisse rêveur, et il y a, d'autre part, tant de faits mathématiques généraux qui m'ennuyent. La généralité est belle lorsqu'elle possède un caractère *explicatif*. Lorsqu'on découvre un théorème avec des hypothèses restrictives qu'on sent peu liées au sujet ou qui paraissant dues uniquement à la méthode employée pour sa démonstration, on cherche, évidemment, à se débarrasser de ces conditions gênantes. On cherche donc à généraliser le résultat pour donner au phénomène son aspect naturel.'

[21]Hij was de oom van de bekendere wiskundige Benoît Mandelbrot, gekend van zijn studie van fractals.

[22][Mandelbrojt1952] p. 53

[23][Mandelbrojt1952] p. 53: 'lorsque cette abstraction indique le monde qui est propre au phénomène découvert.'

[24][Mandelbrojt1952] p. 53: 'Mais si on perd le souci de la recherche de ce monde qui lui est propre et qu'on cherche à généraliser uniquement par goût de généralisation ou le goût d'abstraction, on risque d'entrer dans un monde formalisé'

[25][Mandelbrojt1952] p. 53: 'Je dis, donc, que j'aime à sentir de la matière en mathématiques, ou, pour passer à l'autre extr´mité, je n'aimerais pas vivre dans le monde de la logique formelle telle que l'envisagent quelques-uns de mes collègues.'

sini's identiteit is duidelijk niet de juiste ‘wereld’ van het probleem, terwijl het derde bewijs er al dichter bij zit.

9.4. Definities

Concepten komen ook te voorschijn in *definities*, een essentieel onderdeel van wiskunde.[26] Beeson geeft als voorbeeld de definitie van een continue functie. De standaarddefinitie van het concept continuïteit van een reële functie is de *epsilon-deltadefinitie*:

> De functie f is continu in x ⇔ voor elke ε > 0 bestaat er een δ > 0 zodat voor alle y met $|y - x| < \delta$ geldt dat $|f(x) - f(y)| < \varepsilon$

Als we eenmaal deze definitie hebben en een aantal eigenschappen van het concept continuïteit van een functie bewezen hebben, kunnen we heel eenvoudig andere stellingen bewijzen, zelfs zonder dat we moeten weten wat de epsilon-deltadefinitie inhoudt. Als we bijvoorbeeld willen bewijzen dat $f(x) = (x + 3)^{100}$ een continue functie is, kunnen we eenvoudigweg herkennen dat de functie een samenstelling van twee continue functies is. Daarna verwijzen we naar de stelling dat een samenstelling van twee continue functies een continue functie is[27] en onze stelling is bewezen. Hoewel wij mensen heel intuïtief stellingen kunnen bewijzen met gebruik van zulke definities, kunnen de huidige bewijsprogramma's daar nog niet mee overweg.[28]

Een ander aspect van definities van concepten is dat we ze in een bewijs vaak in beide richtingen nodig hebben. Beeson geeft het voorbeeld van de definitie van een *commutator* in groepentheorie. De definitie van een commutator, genoteerd als $[x,y]$, is $[x,y] = x^{-1}y^{-1}xy$. Zelfs in eenvoudige problemen die enkel de axioma's van groepentheorie gebruiken en als enige extra concept de definitie van commutator vermelden, is redeneren met dit concept voor computers al niet triviaal. Soms zullen we namelijk de definitie $[x,y] = x^{-1}y^{-1}xy$ van links naar rechts moeten toepassen: we vervangen $[x,y]$ door $x^{-1}y^{-1}xy$, waardoor onze uitdrukking tijdelijk ingewikkelder wordt, echter met de mogelijkheid dat we daarna zaken kunnen vereenvoudigen door interactie met de symbolen rond de uitdrukking. Op andere momenten zullen we de definitie van rechts naar links moeten toepassen: als we een uitdrukking in de vorm $x^{-1}y^{-1}xy$ zien staan, vervangen we dit door $[x,y]$. Vaak gaat dit om stellingen waarin moet bewezen worden dat een bepaalde ingewikkelde uitdrukking kan gezien worden als een commutator. Voor bewijs-

[26] Jeremy Avigad maakt overigens een onderscheid tussen definities en concepten: ‘The words “definition” and “concept” seem to have different connotations: someone may know the definition of a group, without having fully understood the group concept.’ ([Avigad2007])

[27] Om deze stelling te bewijzen, moeten we dan weer wel gebruik maken van onze kennis van de epsilon-deltadefinitie.

[28] [Beeson2003]: ‘Merely recognizing $f(x) = (x + 3)^{100}$ as a composition of two functions is beyond the reach of current theorem-provers –it is an application of the author's current research into “second-order unification”.’

programma's vormt dit nog altijd een probleem: ze ‘zien’ niet gemakkelijk wanneer een definitie van links naar rechts of in de andere richting moet toegepast worden.[29]

9.5. Formuleringen met meerdere concepten

In de voorgaande voorbeelden zagen we enkele stellingen waar in de formulering een beperkt aantal concepten voorkwamen, maar waarvan het meest inzichtelijke bewijs wel heel wat extra concepten vereiste. In heel wat *formuleringen* van wiskundige stellingen komen al meer concepten aan te pas en dat heeft als gevolg dat de bewijzen sowieso conceptueel rijker moeten zijn. Beeson geeft als voorbeeld stellingen over regelmatige n-hoeken. Dit is een stelling in de meetkunde, maar omdat het over n-hoeken gaat hebben we ook het concept natuurlijk getal nodig. Omdat we de stelling voor alle n willen bewijzen, zullen we hiervoor waarschijnlijk als bewijsstructuur inductie nodig hebben. Zo verwijzen heel wat meetkundige stellingen naar de theorie van natuurlijke getallen, zodat bewijzen van deze stellingen constant in hun redeneringen constant verspringen naar concepten uit verschillende domeinen. En dit gaat dan nog slechts over eerste-orde meetkunde, een ‘eenvoudige’ theorie.[30]

Een ander domein waarin je verschillende concepten uit andere domeinen moet binnenhalen, is ringtheorie. Ook dit is een ‘eenvoudige’ theorie, die universiteitsstudenten in een inleidende cursus algebra krijgen. Beeson zegt hierover:

> In ring theory, one tries to prove a theorem using only the ring axioms; if one succeeds, the theorem will be true in all rings. However, in books on ring theory one finds many theorems about rings that are not formulated purely in the language of ring theory. These theorems have a larger context: they deal with rings and subrings, with homomorphisms and isomorphisms of rings, and with matrix rings.

Door deze ‘context’ komen er heel wat domeinvreemde concepten in de ringtheorie binnen. Homomorfismes en isomorfismes zijn namelijk *functies* van één ring naar een andere en functies komen niet voor in de axioma's van ringtheorie. Subringen zijn deelverzamelingen van ringen die zelf ring zijn, maar het concept verzameling zit niet in ringtheo-

[29] In [Beeson2003] geeft Beeson nog een eenvoudige ‘exercise in an elementary abstract algebra course’, van de vorm: voor alle a, b en c heeft $c^{-1}[a,b]c$ de vorm $[u,v]$. Het verschil tussen mens en computer komt hier ook tot uiting: ‘OTTER can find several proofs of this theorem, but the u and v in the first few proofs are not the ones a human would find –although it does eventually find the human proof– and OTTER does a fairly large search, while a human does very little searching on this problem.’

[30] [Beeson2003]

rie. Matrixringen zijn dan weer ringen waarvan de elementen matrixen zijn met coëfficiënten uit een gegeven ring. Hiermee brengen we het concept matrix in onze stellingen van ringtheorie binnen. Bovendien drukken bepaalde stellingen beweringen uit voor matrixen met willekeurige grootte, waarmee het concept *natuurlijk getal* ook weer binnensluipt. Kortom, ringtheorie als een geïsoleerde theorie bestuderen zal niet zo veel opleveren.

Beesons student Tony Huang probeerde in zijn *master's thesis* 150 oefeningen uit een tekstboek algebra te formaliseren in eerste-orde ringtheorie.[31] Slechts 14 oefeningen kon hij formuleren in eerste-orde ringtheorie, de rest was complexer omdat de formulering van de oefening al verwees naar domeinvreemde concepten. De 14 eerste-orde oefeningen kon hij echter eenvoudig bewijzen met het bewijsprogramma OTTER. Dit toont al aan dat als computers belangrijke stellingen willen kunnen bewijzen, ze veel meer concepten tegelijk moeten kunnen verwerken.

Beeson schat dat in het algemeen ongeveer 10% van de problemen in een introductiecursus abstracte algebra in de eerste-orde talen van groepen, ringen enzovoort kunnen geformuleerd worden. Van zodra het bijvoorbeeld over subgroepen, homomorfismes, isomorfismes en ordes gaat, breng je er domeinvreemde concepten binnen. Hij noemt als typisch voorbeeld de stelling van Lagrange, een elementaire maar belangrijke stelling in de groepentheorie. Deze stelling zegt dat als H een subgroep van een eindige groep G is, het aantal elementen van H een deler is van het aantal elementen van G. Op een computerbewijs van de stelling van Lagrange is het volgens Beeson nog wachten:

> At present, no theorem-proving program has ever generated a proof of Lagrange's theorem, even though the proof is very short and simple. The obstacle is the mingling of elements, subgroups, mappings, and natural numbers.

Yehuda Rav merkte in zijn artikel ‘Why do we prove theorems?’ al op dat de standaard stellingen in groepentheorie die je in handboeken en artikels vindt meestal niet uit te drukken zijn in een eerste-orde taal.[32] In een voetnoot merkt hij op dat de machinerie van eerste-orde predikatenlogica toepassen op de eerste-orde axioma's van groepentheorie wel heel wat stellingen zal produceren, maar dat bijna geen enkele van die stellingen terug te vinden zijn in de literatuur over groepentheorie. Het werk van Huang over ringtheorie toont wel aan dat Rav in de goede richting zit. Volgens Rav moeten we over wiskundigen dan ook niet denken als ‘deductiemachines’[33] maar uitvinders van methodes en

[31][Huang2002]

[32][Rav1999] p. 17: ‘One just has to think about such fundamental concepts as normal subgroup, torsion group, finite group, composition series;, or such famous theorems such as the Sylow theorems about p-groups, the Jordan-Hölder theorem and the like, to realise that the implicit underlying logic of mainstream group theory is second-order logic.’

[33]Zo merkte de wiskundige Paul Erdös, die bekend stond als een goede ‘problem solver’, ooit op dat een wiskundige een machine is om koffie in stellingen om te zetten.

concepten.[34]

9.6. Wiskundige concepten in de huidige computerbewijzen

David Corfield vermeldt in de inleiding van zijn boek *Towards a philosophy of real mathematics*, in navolging van *Proofs and refutations* van Imre Lakatos, dat conceptualisatie, vermoedens en bewijzen nauw met elkaar verbonden zijn in de activiteit van wiskundigen. Computers kunnen ondertussen wel bewijzen leveren en ze kunnen zelfs vermoedens vormen, bijvoorbeeld door numerieke identiteiten tussen combinaties van constanten te ontdekken, maar Corfield is pessimistisch over de mogelijkheid van computers om de juiste *concepten* te vormen.[35] Wiskundigen doen deze drie activiteiten tegelijkertijd wanneer ze aan een probleem werken. Corfield beschrijft dit proces als volgt:[36]

> Mathematicians perform these activities simultaneously —while clarifying a concept they notice a property which looks like it may hold for all of some class of objects, and while trying to prove that this is so, they find that it pays to introduce conceptual distinctions between elements of that class.

Wiskundigen zoeken ook concepten in de bewijzen van anderen, wat in het geval van computerbewijzen tevergeefs is:[37]

> What mathematicians are largely looking for from each other's proofs are new concepts, techniques and interpretations. Computer proofs certainly give information concerning the truth of a result, but very little beyond this.

We moeten hier een verschil maken tussen computergeassisteerde bewijzen, die slechts een verificatie zijn van berekeningen binnen een vooraf opgegeven bewijsstructuur en automatische bewijsprogramma's die zelf redeneren en de bewijsstructuur zelf uitzoeken.

[34][Rav1999] p. 17: ‘This ought to stir the slumber of those who still think of mathematicians as deduction machines rather than creators of beautiful theories and inventors of methods and concepts to solve humanly meaningful problems.’

[35][Corfield2003] p. 32: ‘One of the lessons we learnt from *Proofs and Refutations* is that when humans do mathematics, three components of their activity —conceptualisation, the formation of conjectures and the construction of proofs— are inextricably linked. Given the very distant prospect of computers being able to supersede mathematicians' capacity to conceptualise, we shall see here whether the latter two components may be disentangled sufficiently to allow machines to augment our capacity to prove and conjecture.’

[36][Corfield2003] p. 35

[37][Corfield2003] p. 56

Deze laatste bewijzen, die vallen onder wat ik in Hoofdstuk 8, *Een classificatie van bewijstechnieken in computerbewijzen logische bewijstechnieken* heb genoemd, kunnen onintuïtieve concepten gebruiken die voor mensen moeilijk te begrijpen zijn. De concepten die in de computergeassisteerde bewijzen voorkomen, zijn echter door mensen uitgevonden en als deze concepten niet zo inzichtelijk zijn, is dat uiteraard de schuld van de wiskundige. Het feit dat een wiskundige in zo'n geval tot zo'n oninzichtelijke concepten komt, kan misschien wel een gevolg zijn van het feit dat hij net een bewijs zoekt met een bepaalde structuur die gemakkelijk door een computerprogramma te verifiëren is door middel van een berekening.

Een algemenere opmerking over de verschillen in de gebruikte concepten tussen computerbewijzen en menselijke bewijzen is dat de kracht van computers heel verschillend is dan die van mensen. Mensen zijn goed in analogisch redeneren, terwijl computers dan weer heel goed zijn in enorm snel rekenwerk.[38] Uiteraard zal een computerbewijs het meeste succes hebben wanneer het van deze kracht gebruik maakt. Dat is ook de mening van Corfield die het geen toeval vindt dat de huidige computerbewijzen de menselijke manier van wiskundige problemen oplossen helemaal niet imiteren.[39] Een deel van de computerbewijzen zal dus altijd wel voor inzichtelijke problemen zorgen voor wiskundigen, omdat computers nu eenmaal uitmunten in bruut rekenwerk en er dus altijd wel computerbewijzen zullen gevonden worden die van deze kracht gebruik maken.[40]Ook Hao Wang zei dit al in 1960: de mensen worden beperkt in wat ze wiskundig kunnen bewijzen omdat ze niet de precisie, snelheid en doorzettingsvermogen van computers hebben.[41]

Volgens Manfred Kerber en Martin Pollet is het ontwerp van logische bewijsprogramma's zoals OTTER volledig verkeerd:[42]

> That foundational systems like first-order logic or set theory can be used to construct large parts of existing mathematics and formal reasoning is one of the deep mathematical insights. Unfortunately it has been used in the field of automated theorem proving as an argument to disregard the need for a diverse variety of representations.

[38] Michael Beeson zegt in [Beeson2003] iets analoogs: 'Machines do mathematics somewhat in the way that submarines swim: ponderously, with more power and duration than a fish, but with less grace and beauty.'

[39][Corfield2003] p. 38: 'I believe it is no accident that the most successful approach to date has been one that has deliberately avoided closely imitating human problem solving techniques. Computers have their own inhuman strengths which need to be harnessed.'

[40]Ook Beeson heeft deze mening: 'We do not expect, however, that all machine-generated proofs will "look human". For example, there exists a machine-generated proof that a certain formula is a single axiom for groups satisfying $x^{19} = 1$ for all x. This proof contains a formula 715 symbols long. No human will find that proof.' ([Beeson2003])

[41][Wang1960] p. 3: 'The human inability to command precisely any great mass of details sets an intrinsic limitation on the kind of thing that is done in mathematics and the manner in which it is done.'

[42][Kerber2002]

Zij beweren dat de resolutiemethode van deze bewijsprogramma's ervoor zorgt dat mensen de bewijzen moeilijk begrijpen én dat de bewijsprogramma's slechts problemen van een beperkte complexiteit kunnen oplossen. Een belangrijk onderdeel van de wiskunde bestaat namelijk uit het relateren van concepten uit verschillende domeinen en niet zozeer uit bewijzen die in eerste-orde logica of verzamelingenleer zijn geformuleerd.[43] De auteurs geven als voorbeeld Cantors diagonalisatie-argument waarmee hij bewees dat de verzameling van reële getallen niet aftelbaar is. Essentieel in dit bewijs is dat je de representatie van een getal in het decimale (of binaire of een ander) talstelsel gebruikt.

Wanneer bewijzen wel volledig op laag niveau geformuleerd zijn, zijn ze moeilijk om te begrijpen. Kerber en Pollet verwijzen naar de *Principia Mathematica* van Russell en Whitehead:

> One of the reasons why the Principia are so rarely read is that the main ideas of the proofs are no longer visible in very long and very detailed proofs.

De auteurs suggereren dat de reden waarom automatische bewijsprogramma's geen ingang vinden bij wiskundigen, gelijkaardig is aan die waarom bijna niemand de *Principia Mathematica* leest: de concepten die beide 'systemen' gebruiken zijn niet aangepast voor het menselijk begrijpen.[44] Als je 362 pagina's moet doorworstelen voordat de auteurs `1 + 1 = 2` kunnen bewijzen, dan is er iets mis.

Ook Jody Azzouni legt in zijn analyse van traditionele informele bewijzen de nadruk op de concepten (objecten) die in de redenering voorkomen. Wiskundig redeneren is volgens hem intrinsiek *semantisch*, niet syntactisch. Ook al kan men wel een lijst van volledig syntactische argumentpatronen vinden die algemeen aanvaard is onder wiskundigen, zoals de axioma's van ZFC, modus ponens, bewijs door inductie en bewijs door gevallen, uiteindelijk redeneren wiskundigen 'by the exploitation of the recognition of properties of the objects the inferences are *about*.'[45] Volgens Azzouni zijn de objecten waarover

[43][Kerber2002]: 'Sometimes an appropriate reformulation of a problem into another representation is already the key step to find a proof. Different representations allow to apply knowledge from different sources to a problem.' Een voorbeeld hiervan is het derde bewijs van Cassini's identiteit dat we hierboven gaven. De belangrijkste stap in het bewijs is inzien dat de linkerkant van de vergelijking een determinant van een matrix is. Dit is te vergelijken met bepaalde visuele bewijzen. Het juiste diagramma bij een bewijs kan soms de rest van het bewijs tot een triviale uitwerking maken. Zo zegt Hao Wang bij een diagramma in een bewijs van de stelling van Pythagoras: 'Here, we would say that for the purpose of proving the desired theorem, finding the above diagram is a much bigger step than the rest.' ([Wang1998])

[44][Kerber2002]: 'It is typically more work even for an experienced human mathematician to formulate the problems in the first place so that an automated theorem prover can prove them than to do the job directly him/herself. We think, when a theorem proving system wants to have a real application as either a proof assistant or a proof tutor, it has to take care about the representation that is used to mathematicians.' Alan Bundy lijkt dezelfde opvatting te hebben. Zie [Bundy2006] p. 481: 'The proofs produced by automated theorem provers consist of a series of low-level logical steps. They tend to be very long and complicated, which obscures the underlying ideas and makes them difficult to understand. Human mathematical proofs, on the other hand, use formalization sparingly, so are much shorter and easier to understand.'

wiskundige redeneringen gaan essentieel voor een begrip in het bewijs en dat is volgens hem dan ook de reden dat het formalistische programma van Hilbert, dat in wezen de wiskundige praktijk wou veranderen, gefaald heeft. [46] Het gevolg is dat we zelfs bij heel korte bewijzen soms wel kunnen zien dat het resultaat uit zijn veronderstellingen volgt, maar niet *waarom* dit zo is.[47]

Azzouni legt dit verschil tussen begrijpen *dat* een resultaat volgt uit iets en begrijpen *waarom* dit zo is uit met de notie van een *inferentiepakket* ('inference package'). Een inferentiepakket kan gezien worden als een soort zwarte doos die specifieke vooronderstellingen en redeneervormen in een bepaald domein encapsuleert. Deze vooronderstellingen zijn niet allemaal introspectief toegankelijk voor de wiskundige die (onbewust) van een bepaald inferentiepakket gebruik maakt. Volgens Azzouni legt het postuleren van inferentiepakketten verschillende observaties uit: waarom diagrammen bij een bewijs zo gemakkelijk iets kunnen uitleggen (ze wekken het gebruik van een bepaald inferentiepakket door onze hersenen op), waarom wiskundige objecten zo belangrijk zijn in de wiskundige praktijk (het zijn de objecten waarop inferentiepakketten inwerken), waarom wiskundigen in bewijzen zo vaak stilzwijgende vooronderstellingen maken (het zijn de vooronderstellingen voor een specifiek inferentiepakket) en waarom sommige diagrammen misleidend zijn (omdat de vooronderstellingen van een inferentiepakket niet introspectief toegankelijk zijn, weten we nooit zeker of een diagram aan dezelfde vooronderstellingen voldoet).[48] Waarom geven sommige concepten ons intrinsiek meer inzicht in een stelling dan andere, als ze allebei dezelfde stelling kunnen bewijzen? Volgens Azzouni heeft dit te maken met inferentiepakketten: we voelen aan dat we een bewijs begrijpen wanneer de gebruikte concepten het gebruik van een bepaald inferentiepakket opwekken. Wanneer dat gebeurt, begrijpen we volgens hem waarom de stelling geldt. [49] Azzouni geeft het voorbeeld van bewijzen in modale logica. Je kan wel de axioma's in modale logica gebruiken om stellingen te bewijzen, maar pas wanneer je gebruik maakt van de Kripkesemantiek om te redeneren over een modaal systeem, wordt volgens hem door een ver-

[45][Azzouni2005] p. 18, ook 'That is, mathematical inference seems to depend largely on insights about the purported *referents* of the mathematical terms so appearing in the proofs.'

[46][Azzouni2005] p. 21: 'The "objects" that mathematical reasoning is prima facie about are *essential to* the capacity of the mathematician to create and understand mathematical proofs.'

[47][Azzouni2005] p. 39: 'The interesting fact is that we feel we grasp or understand a proof not when we can survey it, but only when it "makes sense". This is shown by the fact that even very short derivations, where every step is (derivationally) explicit, needn't as a result be particularly informative. We can, of course, see that the result must follow –because we see how each step must follow from the one before it– but we don't see, as we might put it, *why* the result should follow from the assumptions.'

[48][Azzouni2005] p. 23-26. Over de misleidendheid van sommige wiskundige diagrammen vinden we ook een uitgebreide bespreking in [Casselman2000], bijvoorbeeld p. 1260: 'One of the reasons that mathematicians do not seem to deal well with illustrations is that they tend to see what an illustration is trying to say rather than what it actually says.'

[49][Azzouni2005] p. 40: 'I hypothesize that what we need to feel that we "understand" a proof is the comfortable familiarity of an inference package. When the concepts employed in a proof facilitate the use of an inference package, not only do we feel compelled to the conclusion but we also feel we understand *why* the result follows.'

wijzing naar meetkundige intuïties en concepten het gebruik van inferentiepakketten opgewekt, met als bijproduct het bijbehorende inzicht.

Hoe oninzichtelijk de resulterende computerbewijzen misschien ook zijn, ze voegen wel degelijk *iets* toe. Volgens Corfield kunnen geautomatiseerde bewijsprogramma's bovendien helpen bij het ontwikkelen van nieuwe wiskundige domeinen. Hij suggereert dat wiskundigen misschien wel hun studieobjecten beperkt hebben tot degene die door mensen bewijsbaar zijn.[50] Carlos Simpson ziet dit ook als een belangrijke reden voor de studie van automatische bewijsprogramma's. Ze redeneren namelijk anders dan wij en kunnen zo resultaten vinden die wij door onze beperkingen hebben laten liggen.[51] Larry Wos pleit er zelfs voor om deze kracht van computers zo veel mogelijk uit te buiten: hoe moeilijker het probleem, hoe gemakkelijker de computer het kan bewijzen in vergelijking met de menselijke wiskundige.[52]

Het gebrek aan conceptueel denken bij computers zien we ook wanneer we een menselijk bewijs bekijken waarin de wiskundige een bepaalde substitutie uit zijn hoed tovert. Een mens kan ‘stap achteruit’ zetten in een redenering en zien dat een bepaalde substitutie tot het resultaat zal leiden. Corfield geeft een voorbeeld van zo'n eenvoudige stelling waar een wiskundige onmiddellijk zou zien door welke substitutie hij het kan bewijzen:[53]

> Were a mathematician asked to show that any group all of whose elements square to the identity is commutative, it would probably not be long before they thought to substitute yz for x in `f(x, x) = e`.

Na de substitutie yz voor x zal een wiskundige dus het bewijs beginnen van `(yz)(yz) = 1`. Hij vermenigvuldigt dan links met y en rechts met z en komt zo tot: `y(yz)(yz)z = yz`. Met gebruik van de associativiteit van de groepsoperatie krijgen we: `(yy)zy(zz) = yz` of (door de definitie) `zy = yz`. Hiermee hebben we de commutativiteit van de groep bewezen. Een computer, zoals Corfield ook vermeldt, heeft echter geen visie op welke substituties waarschijnlijk zullen werken en hij zou dan domweg alle mogelijke substituties moeten uitproberen.

Er bestaat geen geautomatiseerde methode om ‘slimme’ substituties te vinden. Wos en Fitelson gaan zelfs verder: zij vinden substitutie, omdat het zo moeilijk te implementeren valt in een automatisch bewijsprogramma, niet nodig en zelfs *niet wenselijk*.[54] Hun beweegreden is de opvatting dat we zoveel mogelijk moeten gebruik maken van de sterke

[50][Corfield2003] p. 39: ‘Automated theorem provers might allow mathematicians to develop mathematical domains which they would not otherwise have done. Perhaps, not only have mathematicians been constrained to study subject areas in which they could calculate, expansion occurring through theoretical, algorithmic and technological advances, but also they have been limited to domains in which results are humanly provable.’

[51]Simpson drukt dit in [Simpson2004] p. 290 sloganesk uit: ‘Locally unmotivated reasoning might lead to globally interesting results.’

[52][Wos2001] p. 91: ‘When a question resists answer or a problem resists solution (at least when a reasoning program is part of the team), to sharply increase the likelihood of success, replace the question or problem with one that is far tougher.’ Hoe onintuïtief deze raad ook lijkt, volgens Wos geldt ze echt.

[53][Corfield2003] p. 46

punten van computers en hen het menselijke denken niet moeten laten imiteren. We zien hier duidelijk twee verschillende kampen: mensen uit de onderzoeksgroep van Wos vertrouwen op de brute rekenkracht van computers en zijn er dan ook niet in geïnteresseerd of het bewijs voor een mens begrijpelijk is. Ze hebben dan ook heel wat resultaten bereikt. Mensen als Wang, Beeson, Kerber, Pollet en Bundy willen bewijsprogramma's net het denken van mensen meer laten imiteren, zodat wij de gevonden bewijzen ook meer kunnen begrijpen. Corfield ziet wel de conceptuele problemen van de huidige computerbewijzen, maar hij lijkt zich er bij neer te leggen dat het zo zal blijven.

Overigens vinden bewijsprogramma's zoals OTTER wel zelf hun bewijs, maar hun succes hangt sterk af van de richting waarin mensen hen sturen door middel van allerlei parameters. Uit Corfields ervaring is OTTER heel gevoelig voor de parameters en is het voor een wiskundige zonder ervaring met OTTER heel moeilijk om belangrijke stellingen te bewijzen.[55] Ook Beeson lijkt dezelfde mening te hebben: hij noemt resultaten behalen met OTTER meer een kunst dan een wetenschap.[56]

We gaan nu kort één voor één een aantal van de besproken computerbewijzen en enkele nieuwe af en we bespreken kort welke concepten de bewijzen gebruiken en de band tussen deze concepten en de inzichtelijkheid in het bewijs.

9.6.1. De stelling van Pappus

Cerutti en Davis hebben het in verband met hun computerbewijs van de stelling van Pappus verrassend genoeg niet echt over de vraag of dit bewijs nu inzicht geeft in waarom de stelling geldig is. Laten we het bewijs eens vergelijken met een traditioneel bewijs van de stelling. Het is niet zo dat een traditioneel bewijs van de zeshoekstelling van Pappus een sterk beroep doet op meetkundige inzichten en dat hierin het verschil ligt met het bewijs door Cerutti en Davis. Een aantal traditionele bewijzen van de stelling vertalen namelijk de meetkundige voorstelling naar een vectorvoorstelling en rekenen dan met de rekenregels van vectoren. Andere bewijzen zijn wel meer meetkundig opgesteld en maken bijvoorbeeld gebruik van eigenschappen van congruente driehoeken in de configuratie van de negen punten en de stelling van Menelaus die een verband geeft tussen het op één lijn liggen van punten in een driehoek en de verhoudingen van lengtes van zijden van de driehoek.

De niet-meetkundige bewijzen van de zeshoekstelling van Pappus zijn ook veel gemakkelijker te begrijpen dan het bewijs van Cerutti en Davis. Met vectorrekenen kan je op één à twee pagina's de stelling van Pappus bewijzen en die stappen zijn eenvoudig te vol-

[54][Wos2002] p. 5: ‘Although instantiation serves well logicians and mathematicians, unless an effective strategy is discovered to control its use, instantiation is unneeded and even unwanted in the context of mechanizing inference rule application and proof finding.’

[55][Corfield2003] p. 48: ‘I find it unlikely that difficult problems will be solved by OTTER without considerable input from someone who has worked closely with Wos's team.’

[56][Beeson2003]: ‘There are so many parameters that running OTTER is more of an art than a science.’

gen. Niet alleen de stappen op zich zijn te volgen, maar je ziet met enige goede wil ook een *bewijsstrategie*: waarom bepaalde omzettingen gebeuren. Het bewijs van Cerutti en Davis voert eigenlijk dezelfde reductie tot vectorrekenen uit, maar gaat nog een stap verder. Het primitieve concept in hun computerbewijs is niet de vector maar de coördinaat van een vector. In plaats van rekenregels van vectoren te gebruiken, wordt het probleem dus gereduceerd tot het berekenen van een determinant van een matrix met een ingewikkelde uitdrukking in coördinaten.[57] Het programma vereenvoudigt wel een aantal expressies, maar wanneer het uiteindelijk het resultaat `DE = 0` geeft en je als wiskundige het bewijs bekijkt, zie je een hoop algebraïsch rekenwerk, met andere woorden wat symbolische manipulatie met getallen. Het spreekt voor zich dat dit bewijs niet echt inzicht geeft. Het traditionele bewijs dat nog één niveau hoger werkt, bestaat ook uit algebraïsch rekenwerk, maar doordat het primitieve concept daar een vector is en men daar kan gebruik maken van de meer inzichtelijke rekenregels van vectoren, is het bewijs minder lang en zie je een strategie.

Cerutti en Davis reproduceren in hun artikel één van de termen uit de determinant.[58] Alhoewel ze de coördinaten speciaal geselecteerd hebben en het computerprogramma zelf vereenvoudigingen heeft uitgevoerd, bestaat die ene term al uit een som van 41 veeltermen in 8 variabelen. Een determinant van een matrix waarin zes van die grote termen staan, is uiteraard heel onoverzichtelijk. In het traditioneel bewijs met vectoren als basisconcepten is het aantal variabelen beperkt tot 6 en is de lengte van de verschillende termen beperkt, wat het bewijs veel overzichtelijker maakt. Uiteraard is het bewijs van Cerutti en Davis niet meer representatief voor de huidige computerbewijzen, maar het illustreert wel goed het verschil op conceptueel vlak tussen de meeste computerbewijzen, die goed zijn in symbolenmanipulaties met concepten op laag niveau, en ‘goede’ menselijke bewijzen, die inzicht op een hoger niveau tonen.

9.6.2. De vierkleurenstelling

Eén probleem dat in veel computerbewijzen voorkomt, is dat meetkundige concepten in algebraïsche concepten moeten omgezet worden. Terwijl wiskundigen vaak probleemloos met meetkundige concepten als *punt*, *lijn* enzovoort kunnen redeneren, daarbij geholpen door hun meetkundige intuïties van in het dagelijkse leven, kunnen computers daar niet op terugvallen. De wiskundige die het computerbewijs ontwerpt, moet dus de meetkundige concepten *volledig* omzetten naar algebraïsche concepten, waarmee de computer kan rekenen. De hele ‘redenering’ van de computer zal dan ook algebraïsch zijn en dat maakt het voor de wiskundige moeilijker om het bewijs te begrijpen, aangezien er nog weinig verwijzingen instaan naar de meetkundige concepten waarover de stelling eigenlijk gaat. Hij kan (een gedeelte van de) algebraïsche afleidingen wel verifiëren op hun correctheid, maar hij kan hiervoor niet terugvallen op zijn meetkundige intuï-

[57] Wat Ulf Grenander in [Cerutti1969] p. 896 ‘the method of artificial stupidity’ noemde.
[58] [Cerutti1969] p. 899

ties.

Een voorbeeld waarin dit duidelijk wordt is Gonthiers formalisering van Robertsons bewijs van de vierkleurenstelling. De intuïtieve concepten die in Robertsons bewijs voorkwamen, zijn niet bruikbaar voor een volledig geformaliseerd computerbewijs.[59] Daarom formaliseerde Gonthier deze concepten tot combinatorische eigenschappen, die perfect door een computer te verwerken zijn. Gonthier bespreekt zelf de intuïtiviteit van verschillende aanpakken om het concept van een vlakke graaf formeel te definiëren. Hij vermeldt bijvoorbeeld een aanpak die gebruik maakt van de Jordankrommestelling[60], maar hij zegt daarover:[61]

> However, this approach results in proofs that, while convincing for humans, are difficult to formalize because they contain an informal mix of combinatorics and topology.

Gonthier koos voor een andere aanpak, meer geoptimaliseerd voor de formaliseerbaarheid dan voor de inzichtelijkheid. Hij gebruikte hiervoor het concept *hyperkaart* ('hypermap'). Dit liet hem toe om de vierkleurenstelling in zuiver combinatorische termen uiteen te zetten, zonder gebruik te maken van de Jordankrommestelling waarvan de bekende bewijzen gebaseerd zijn op complexe analyse en/of homologie.[62] Hij kon een combinatorisch equivalent van de Jordankrommestelling voor hyperkaarten bewijzen. De aanpak met hyperkaarten vereenvoudigde volgens Gonthier heel wat delen van het bewijs. Waar het bewijs van Robertson bijvoorbeeld nog moest verwijzen naar een nooit gepubliceerd bewijs van een 'folklorestelling' dat eenvoudig maar lang is, kon Gonthier deze stelling vermijden. De hyperkaarten maakten het bewijs dus misschien conceptueel minder inzichtelijk, maar de structuur van het bewijs werd er wel eenvoudiger op.[63]

Gonthier verwijst in zijn bespreking van zijn bewijs verschillende keren naar de 'gap between the intuitive, picture-rich proof outline, and the very precise logical statement that had to be fed to the COQ proof assisant.'[64] De concepten die wij mensen gebruiken in een bewijs van bijvoorbeeld de vierkleurenstelling, zijn vaak heel moeilijk om te formaliseren. Gonthier verwijst bijvoorbeeld naar stellingen zoals 'de hele kaart is vlak als en slechts als de twee delen vlak zijn', iets dat wij intuïtief onmiddellijk inzien.[65] De geformaliseerde versie van deze bewijzen verliest dan ook heel wat van de inzichtelijkheid van

[59][Gonthier2004] p. 17: 'Because most graph theory proofs rely on the visual analysis faculties of the reader, and a proof assistant like COQ is fully devoid of such faculties, we mostly had to come up with our own proofs (sometimes using visual intuition as guidance, but not always).'

[60]Elke eenvoudige gesloten kromme in het vlak verdeelt het vlak in een binnen- en buitenkant

[61][Gonthier2004] p. 5. De 'informal mix of combinatorics and topology' is belangrijk.

[62]Bovendien had Thomas Hales de Jordankrommestelling al formeel bewezen in HOL-LIGHT en anderen in MIZAR. Gonthier zag dus niet veel zin in het dupliceren van dit werk.

[63]In [Gonthier2004] p. 17 zegt Gonthier over de hyperkaarten: 'While this notion was generally more cumbersome than the more familiar topological one, on several occasions its additional precision provided a clear argument where the "intuitive" situation seemed muddled at best.'

[64][Gonthier2004] p. 52

het origineel.[66]

Als je het gebruik van hyperkaarten en de discretisatie van het probleem wegdenkt, ligt het bewijs van Gonthier eigenlijk volledig in de lijn van alle andere bewijzen van de vierkleurenstelling.[67] Zijn bewijs is direct gebaseerd op dat van Robertson, dat op zijn beurt dezelfde structuur heeft als dat van Appel en Haken, die dezelfde basisprincipes gebruiken als Kempe in zijn incorrect bewijs. Gonthier ziet zijn bewijs als een logische evolutie van deze bewijzen, naar meer precisie en daardoor meer vertrouwen in de correctheid.[68] Bij het preciseren van het bewijs is echter wel heel wat inzichtelijkheid van het bewijs verloren. Gonthier zegt het niet met zo veel woorden, maar de verschillende keren dat hij het heeft over de kloof tussen intuïtie en formalisering maken duidelijk dat hij het probleem wel opgemerkt heeft. Gonthier heeft met zijn geformaliseerd bewijs misschien wel het vertrouwen in de correctheid van de vierkleurenstelling verhoogd, maar zeker de inzichtelijkheid niet. En net over de inzichtelijkheid was er zo veel discussie bij wiskundigen. Volgens Ursula Martin staan dit soort formaliseringen dan ook haaks op de wiskundige praktijk.[69] William Thurston vindt formaliseringen van wiskunde door computers wel een ‘worthwhile project’, maar hij waarschuwt voor het te veel belang hechten hieraan:
[70]

> We should recognize that the humanly understandable and humanly checkable proofs that we actually do are what is more important to us, and that they are quite different from formal proofs.

Het probleem van inzichtelijkheid ligt niet enkel bij de formalisering door Gonthier van het computerbewijs van Robertson. Volgens sommigen ligt het probleem in zekere zin al bij de combinatorische aanpak waarop al de bewijzen van de vierkleurenstelling zich baseren. Zo denkt Paul Halmos dat dit niet de juiste aanpak is en dat de bewijzen gebruik maken van de verkeerde concepten. Binnen 100 jaar zal de vierkleurenstelling met be-

[65][Gonthier2004] p. 53: ‘However, the proof of the Birkhoff lemma required the graphically obvious operations of cutting up a map along a cycle and conversely of gluing back two map pieces. Defining these operations formally, and proving such intuitively obvious properties as “the whole map is planar if and only if the two pieces are” turned out to be much harder than expected, again because of the gap between pictorial intuition and computerized formalism.’

[66]Gonthier geeft een voorbeeld in [Gonthier2004] p. 53: ‘Indeed, the asymmetric “patch” relation described in section 5.1 emerged directly from the formal statement of the Jordan property —it seems completely counterintuitive when one only looks at plain maps or graphs.’

[67][Gonthier2004] p. 6: ‘The basic proof outline of the Four Colour Theorem, oddly, has not changed since the initial, incorrect proof attempt by Kempe in 1879.’

[68][Gonthier2004] p. 5: ‘Our actual work was to find a systematic way of getting the details right —*all* of them.’

[69][Martin1999]: ‘The latter *[full formal development of large bodies of mathematics]* is at odds with mathematical practice: many mathematicians view formal proof development unsympathetically and it is hard to see it being incorporated except in expensive reworkings of well-understood material.’

[70][Thurston1994] p. 171. Jones heeft in [Jones1998] een gelijkaardige opvatting over bewijzen in topologie: ‘If one “sees” the pictures, then one understands, but otherwise one cannot follow. In principle one could formalize the whole argument, but that would add nothing.’ (p. 213)

hulp van de juiste concepten in een paar pagina's kunnen bewezen worden in een cursus voor universiteitsstudenten, zo voorspelde Halmos.[71]

9.6.3. Het Robbinsprobleem

Het Robbinsprobleem verschilt vanuit inzichtelijk standpunt nogal van de twee vorige bewijzen die we bespraken. Bij de zeshoekstelling van Pappus en bij het vierkleurenprobleem verminderden de computerbewijzen de inzichtelijkheid door intuïtieve meetkundige of topologische concepten om te zetten in meer formale algebraïsche of combinatorische concepten. Het Robbinsprobleem is verschillend omdat het sowieso al in formele, algebraïsche concepten is uitgedrukt. Dat het bewijs van het probleem weinig inzichtelijk is, is hier dus niet enkel te wijten aan de computer, maar ligt deels aan het probleem zelf. Het is nu eenmaal een probleem in de algebra dat weinig beroep doet op intuïtieve inzichten. Hao Wang voorspelde in 1960 al dat computerprogramma's zouden uitmunten in wiskundige domeinen waar mensen weinig intuïties hebben.[72]

Toch ligt het niet enkel aan het probleem zelf dat het bewijs niet zo inzichtelijk is. Het bewijs door McCunes programma EQP bestaat uit lange formules met een groot aantal haakjes, dat door wiskundigen die het bewijs bestudeerd hebben onleesbaar en moeilijk te volgen genoemd wordt.[73] Stanley Burris vond de computeruitvoer onleesbaar, aangezien elke stap in EQP's bewijs uit een aantal andere stappen volgde, maar de details zoals welke substituties er moesten uitgevoerd worden, moest hij zelf invullen om het bewijs te begrijpen. Burris herwerkte het korte maar condense bewijs daarom tot een langer bewijs waarin de opeenvolgende stappen eenvoudiger zijn en hij introduceerde hulpvariabelen zoals `T := D + E + y + y`. Bij elke stap legt Burris uit waaruit deze stap volgt. Zijn bewijs van 4 pagina's (1 pagina definities van hulpvariabelen en 3 pagina's bewijs) is een opeenvolging van 105 korte vergelijkingen die je eenvoudig kan volgen.[74] Dat het

[71][Halmos1990] p. 577: ‘We may still be far from finding a “good” proof of the four-color theorem. We need a simple insight into a new and complicated kind of geometry or intricate algebra, and the distance from there to a purely conceptual, existential proof of the four-color theorem is probably just as great. […] I believe that the computer (and, for another example, the 10,000 pages of published proof solving the simple groups problem) missed the right concept and the right approach. Their time will come. A hundred years from now both theorems (maps and groups) will be exercises in first-year graduate courses, provable in a couple of pages by means of the appropriate concepts, which will be completely familiar by then. Down with oracles, I say —they are of no use in mathematics.’

[72][Wang1960b] p. 231: ‘On the whole, it seems reasonable to think that machines will more quickly excel in areas where man's intuition is not so strong.’ Ook Alan Bundy heeft deze mening in zijn overzichtsartikel over automatische bewijsprogramma's [Bundy1999c]: ‘Conjectures best suited to this approach are combinatorial problems in new areas of mathematics, where human intuitions are less well developed. The Robbins Algebra conjecture is a good example.’

[73][Kolata1996], [Fitelson1998], [Kauffman2001]. Corfield noemt het ‘a proof which seems to lie just a little beyond the grasp of humans’ ([Corfield2003]). Ursula Martin is blijkbaar niet op de hoogte van de niet-inzichtelijkheid en de omslachtigheid van het bewijs en noemt het ‘easily checked by hand’ ([Martin1999]).

[74]Burris zegt over zijn bewijs in [Peterson1997]: ‘I ended up with a lot of little equations. You could easily sit on a bus and go through the hundred or so steps of the proof.’

hem om een bewijs gaat dat voor mensen gemakkelijker te volgen is, maakt hij heel duidelijk in de titel van zijn manuscript: *An anthropomorphized version of McCune's machine proof that Robbins algebras are Boolean algebras*.[75]

Bernd Dahn analyseerde het bewijs van EQP in het computerprogramma ILF, dat de formules in het bewijs kon vereenvoudigen door afkortingen te introduceren voor veel gebruikte deelformules. Uit een statistische analyse van het bewijs bleek namelijk dat de formule `~(~(~x + y) + z)` vaak voorkwam. Deze formule als definitie introduceren maakte het bewijs als geheel niet eenvoudiger, omdat de formule geen bruikbare eigenschappen had. De eenvoudigere formule `δ(x, y) = ~(~x + y)` maakte het bewijs echter wel eenvoudiger.[76]

Branden Fitelson suggereert dat EQP's bewijs moeilijk te volgen is voor mensen omdat het niet *conceptueel* is. Een mens zou nooit zo'n bewijs vinden omdat het te complex is. Er zit geen visie achter, het is toevallig het bewijs dat met de zoekparameters van het programma in redelijke tijd kon gevonden worden door een computer. Het programma gebruikt eenvoudig deductief logisch redeneren met vergelijkingen, kent slechts een beperkt aantal symbolische concepten en vindt zelf geen nieuwe concepten uit. Het kan dan ook nooit een conceptueel, gestructureerd, abstract of elegant bewijs vinden.[77] De substituties die gebeuren zijn bovendien dan ook helemaal niet intuïtief.[78] Fitelson noemt het feit dat computerbewijzen zoals dat van EQP niet intuïtief zijn overigens een groot probleem:[79]

> One of the most pressing current problems facing researchers in automated theorem-proving (according to Professor Kenneth Kunen, personal communication) is the translation and reconstruction of complex, unintuitive computer proofs into forms that are more readily understood by human beings.

EQP en zijn varianten redeneren enkel met de vergelijkingen en het beperkt aantal symbolische concepten die ze gekregen hebben. Ze vinden geen nieuwe concepten uit.[80] Dat maakt het voor wiskundigen echter moeilijk om het bewijs te volgen. Burris' 'antropomorfisering' van EQP's bewijs toont al aan dat wij mensen hulpvariabelen en hulpconcepten nodig hebben om inzicht in een bewijs te krijgen.[81]

---

[75][Burris1996]

[76][Dahn1997]: 'Some of the complex formulas in OTTER's proof could be presented as describing simple properties of the δ-function.'

[77][Peterson1997]: Wos zegt hierover 'Our programs do not learn, do not make judgments, and they do not invent concepts.'

[78][Fitelson1998]: 'The proof found by EQP is quite complex and difficult to follow. Some of the steps of the EQP proof require highly complex and unintuitive substitution strategies.'

[79][Fitelson1998]

[80][Peterson1997]: Wos zegt hierover 'Our programs do not learn, do not make judgments, and they do not invent concepts.'

9.6.4. Het Keplervermoeden

In Hales' bewijs van het Keplervermoeden lijken de problemen niet echt bij de concepten te liggen. Het case-testing aspect komt daar op de voorgrond en het hele bewijs is gewoon uiterst complex, zodat daar het probleem van de betrouwbaarheid meer op de voorgrond komt. Hales zelf denkt dat het Keplervermoeden een kort en elegant bewijs kan krijgen door het te beschouwen als een speciaal geval van een familie optimalisatieproblemen en het bewijs ervan nog meer te automatiseren:[82]

> Ultimately, a properly automated proof of the Kepler conjecture might be short and elegant. The hope is that the Kepler conjecture might eventually become an instance of a general family of optimization problems for which general optimization techniques exist. Just as today linear programming problems of a moderate size can be solved without fanfare, we might hope that problems of a moderate size in this family might be routinely solved by general algorithms. The proof of the Kepler conjecture would then consist of demonstrating that the Kepler conjecture can be structured as a problem in this family, and then invoking the general algorithm to solve the problem.

Hales gelooft dus dat het feit dat zijn bewijs een computerbewijs is, niet echt zorgt voor weinig inzichtelijke concepten. Hij denkt zelfs dat hoe meer hij in het bewijs kan automatiseren, hoe inzichtelijker (‘short and elegant’) het bewijs zal worden. Als je echter kijkt naar Gonthiers formalisering van de vierkleurenstelling, zie je dat dit bewijs minder inzichtelijk is dan dat van Robertson of Appel en Haken. Hales' verwachting dat het bewijs door automatisering inzichtelijker zal worden, strookt dus niet helemaal met de kloof tussen formalisering en intuïtie die Gonthier heeft ervaren en die vrij algemeen lijkt te gelden. In de veronderstelling dat deze kloof een algemeen geldend fenomeen is, is de enige manier waarop Hales' verwachting wel zou kunnen uitkomen, dat hij zou gebruik maken van Bundy's bewijsplannen en Beesons integratie van berekeningen en redeneringen. Hales lijkt echter meer geïnteresseerd in formaliseringen in de school van Wiedijk.

9.6.5. Het XCB-probleem

Een gelijkaardig probleem als bij EQP's bewijs van het Robbinsprobleem, namelijk dat het wel een kort bewijs is, maar de stappen die uit elkaar volgen voor een mens moeilijk te verifiëren zijn, vinden we in een bewijs door OTTER dat de formule XCB een kort enig axioma van de klassieke equivalente calculus is.[83] In 1933 ontdekte Lukasiewicz de eerste formules van lengte 11 die in hun eentje deze theorie konden axiomatiseren. Hij be-

[81] Belinfantes onderzoek naar automatische bewijsprogramma's om stellingen over ordinaalgetallen te bewijzen ([Belinfante1999], en eerder in deze eindverhandeling al besproken), toont echter aan dat bewijsprogramma's als OTTER toch wel ‘conceptueler’ bewijzen kunnen vinden.

[82] [Hales2003] p. 489

[83] Tot voor dit bewijs bevatten alle axiomatiseringen van de klassieke equivalente calculus meerdere axioma's.

wees dat kortere formules niet als enige axioma's konden dienen. Later werden nog verschillende axioma's van lengte 11 gevonden en van veel andere werd bewezen dat ze de theorie niet axiomatiseerden. Sinds 1977 was er nog maar één formule waarvan men niet kon bewijzen of ze de theorie axiomatiseerde of niet: XCB. In 2003 bewezen Wos, Ulrich en Fitelson met OTTER dat uit XCB twee formules volgen waarvan men al wist dat ze samen de klassieke equivalente calculus axiomatiseerden en zo vonden ze dus het laatste korte axioma. OTTER was onmisbaar om dit bewijs te vinden, zo merken ze op in een voetnoot:[84]

> That such assistance was invaluable, and perhaps indispensable, will occur to the reader who attempts to carry out by hand the condensed detachment of line 16 from line 12 to obtain line 17 in the proof given in the following section. The substitution instances of 12 and of 16 required for that condensed detachment are, respectively, 2,939 and 2,919 symbols in length, a consideration that may explain in part why these two questions about XCB remained unanswered for so long.

De auteurs lijken te suggereren dat het probleem intrinstiek zo complex is dat een bewijs ervan sowieso moeilijk door mensen zal te volgen zijn.

9.6.6. Het HBCK-probleem

A. Wronski sprak het vermoeden uit dat een natuurlijk geordende BCK-algebra (HBCK) een variëteit is. Blok en Ferreirim bewezen het vermoeden met modeltheoretische argumenten en Tomasz Kowalski bewees het voor het eerst syntactisch. Robert Veroff bewees de stelling met behulp van OTTER.[85] HBCK wordt gedefinieerd door de volgende axioma's:

- **M3:** `x * 1 = x`
- **M4:** `1 * x = x`
- **M5:** `(x * y) * ((z * x) * (z * y)) = 1`
- **M7:** `x * y != 1 | y * x !=1 | x = y`
- **M8:** `x * x = 1`
- **M9:** `x * (y * z) = y * (x * z)`
- **H:** `(x * y) * (x * z) = (y * x) * (y * z)`

Het probleem is dan om de volgende gelijkheid af te leiden uit de HBCK-axioma's:

```
(((x * y) * y) * x) * x = (((y * x) * x) * y) * y
```

[84] [Wos2003]

[85] http://www.cs.unm.edu/~veroff/HBCK/. Een eerdere (en iets langere) versie van het bewijs is gepubliceerd in [Veroff2002]

OTTER vond een bewijs (van lengte 47), maar opvallend hierbij is dit: OTTER kon het bewijs slechts vinden nadat Veroff de volgende definitie bij de axioma's plaatste:

```
g(x, y) = (((x * y) * y) * x) * x
```

OTTER introduceert in stap 8 de definitie g(x, y) en in stap 46 komt het programma tot `g(x, y) * g(y, x) = 1`. Door gebruik te maken van axioma M8 (`x * x = 1`) komt het in stap 47 tot de te bewijzen stelling: `g(x, y) = g(y, x)`. Door de commutativiteit van de gedefinieerde functie te bewijzen, bewijst OTTER dus het gegeven probleem. We zien hier dus een concreet voordeel van definities en dus concepten. Zonder de definitie vond OTTER geen bewijs, maar met de definitie wel.

9.6.7. Isomorfie tussen groepen

Het project TPTP (*Thousands of Problems for Theorem Provers*) is een verzameling van testproblemen voor automatische bewijsprogramma's.[86] Versie 3.2.0 (augustus 2006) bevat zo'n 9000 testproblemen. De problemen worden formeel voorgesteld in een vorm die door een automatisch bewijsprogramma kan opgelost worden. Er wordt ook verwezen naar oplossingen door bekende bewijsprogramma's, zodat je kan vergelijken hoe de verschillende programma's het probleem aanpakken. Zo'n 800 problemen gaan over groepentheorie. Eén van de problemen is het volgende:[87]

> If G1 has exactly two elements and G2 has exactly two elements, then there exists an isomorphism [a one-to-one and onto homomorphism] between them.

Corfield bespreekt dit als een ‘type of problem that is a very obvious result for a human and yet it is difficult for a machine.’[88] Een wiskundige zal deze stelling vrij informeel bewijzen. Bijvoorbeeld: de twee elementen van de groep zijn *e* (het identiteitselement) en *a*. Omdat *e* het identiteitselement is, kunnen we drie van de waarden in de vermenigvuldigingstabel eenvoudig afleiden: e.e = e, e.a = a en a.e = a. Omdat *a* een inverse nodig heeft en a.e verschillend is van e, is er geen andere oplossing dan a.a = e. Deze redenering geldt voor de vermenigvuldigingstabel van elke andere groep van twee elementen, waardoor de isomorfie bewezen is.

De huidige automatische bewijsprogramma's kunnen zo'n redenering echter niet uitvoeren, ze redeneren volledig expliciet. We zien dit aan de manier waarop dit probleem in computerleesbare vorm gezet wordt. In de commentaar van TPTP probleem GRP025-1 staat het volgende:

> In order to prove the theorem, the group tables and a particular homo-

[86] Te vinden op http://www.cs.miami.edu/~tptp/
[87] TPTP-probleem GRP025-1
[88] [Corfield2003] p. 41

> morphism are specified, and the contradiction comes from the fact that this is the actual isomorphism. Not only is this formulation cheating, but also it does not prove the theorem in full generality.

In de formele definitie van het probleem zien we dat een groep g1 met als enige elementen a en b wordt gedefinieerd en een groep g2 met als enige elementen c en d. Dan worden de vermenigvuldigingstabellen expliciet gegeven: a.a = a, a.b = b, b.a = b, b.b = a en c.c = c, c.d = d, d.c = d, d.d = c. Daarna wordt het isomorfisme gedefinieerd dat a op c afbeeldt en b op d. Het bewijsprogramma krijgt dan de opdracht om te bewijzen dat, gegeven elementen d1, d2 en d3 van g1, waarvoor d1.d2 = d3 geldt, het product van de afbeelding van d1 en de afbeelding van d2 onder het isomorfisme *niet* gelijk is aan de afbeelding van d3. Als het programma een contradictie vindt, hebben we dus een bewijs dat het isomorfisme tussen de twee groepen echt een isomorfisme is. Zoals in de commentaar bij het probleem al staat, hebben de opstellers van het probleem op deze manier de computer heel wat op gang geholpen door de vermenigvuldigingstafels en het isomorfisme al expliciet te geven. De stelling die bewezen is, zegt dus eigenlijk gewoon dat twee specifieke groepen isomorf zijn onder een gegeven afbeelding, terwijl de stelling die zou moeten bewezen worden veel algemener is: tussen alle groepen met exact twee elementen bestaat een isomorfisme.[89]

Een tweede formalisering van het probleem heeft de volgende commentaar:[90]

> In order to prove the theorem, we specify one element of each group as the identity element and take as a previously-proven lemma (obvious) that maps from G1 -> G2 which are not one-to-one or which are not onto need not be considered for isomorphisms between the groups. Thus we consider only the two one-to-one and onto maps between the groups, and show that assuming neither of them are homomorphisms gives a contradiction.

Dit is al beter, maar ook hier nog hebben de bewijsprogramma's hulp nodig. In de formele definitie van het probleem worden de twee mogelijke isomorfismes tussen de groepen gedefinieerd. Dan wordt opgedragen om te bewijzen dat geen van de twee een isomorfisme is. Vindt het programma een contradictie, dan hebben we dus bewezen dat er een isomorfisme bestaat tussen de twee groepen. Overigens zien we bij probleem GRP026-1 en GRP026-2 hetzelfde. Hier gaat het om de stelling dat er een isomorfisme bestaat tussen twee groepen met elk exact drie elementen. GRP026-1 lost dit ook op door de vermenigvuldigingstabellen en een specifiek isomorfisme te definiëren en GRP026-2 lost dit op door de zes mogelijke isomorfismes te definiëren.

Uiteraard zijn dit maar twee problemen in de hele collectie (de enige twee die het woord

[89]Corfield zegt hierover: ‘That a serious attempt to prove such a simple result should fail to establish it in full generality, and should involve so generous a hint to the machine as to be described as “cheating” is surely revealing.’ ([Corfield2003] p. 42)

[90]TPTP-probleem GRP025-2

'cheating' in de commentaar vermelden), maar het laat toch zien dat mensen op een 'hoger niveau' redeneren dan deze bewijsprogramma's. Corfield concludeert:[91]

> What this shows is the difficulty of higher-order reasoning for present-day machines. Problems more naturally expressed in higher-order terms are either represented unnaturally to be fed to a first-order logic theorem prover, and in the process helped considerably, or else they are fed to a higher-order logic theorem prover which almost always needs much assistance throughout the proof.

9.6.8. Hypergeometrische identiteiten

In hun boek *A = B* leggen Petkovsek, Wilf en Zeilberger de methodes uit die zij en anderen hebben ontwikkeld om ingewikkelde identiteiten door een computer te laten bewijzen.[92] De auteurs geven in het begin van hun boek het volgende eenvoudige voorbeeld:[93]

$$\sum_k \binom{n}{k}^2 = \binom{2n}{n}$$

Daarna illustreren ze de methode waarmee het WZ-algoritme de identiteit automatisch bewijst. De WZ-methode kan deze sommen volledig mechanisch oplossen, maar je krijgt hiermee niet echt inzicht in waarom de stelling geldt. Dat inzicht krijg je wel met een menselijk bewijs, omdat die met verschillende concepten werkt, verschillende interpretaties van de identiteit gebruikt, enzovoort. De auteurs geven bijvoorbeeld in vijf regels tekst een eenvoudige bewijs van de identiteit en merken daarna op:[94]

> We must pause to remark that that one is a really nice proof. So as we go through this book whose main theme is that computers can prove all of these identities, please note that we will never claim that computerized proofs are *better* than human ones, in any sense. When an elegant proof exists, as in the above example, the computer will be hard put to top it. On the other hand, the contest will be close even here, because the computerized proof that's coming up is rather elegant too, in a different way.

De auteur van het bewijs geeft dan deze veelzeggende opmerking:

> To continue, the pre-computer proof of (2.3.1) that we just gave was combinatorial, or bijective. It found the combinatorial interpretations of both

[91] [Corfield2003] p. 42

[92] Eigenlijk gaat dit meer om berekeningen dan om bewijzen. Het WZ-algoritme is een volledig geautomatiseerde beslissingsprocedure voor problemen van een bepaalde vorm.

[93] [Petkovsek1996] p. 24

[94] [Petkovsek1996] p. 24

sides of the identity, and showed that they both count the same thing.

Hij heeft het hier over een *interpretatie* van beide kanten van de identiteit. Terwijl het WZ-algoritme niet aan interpretaties denkt en gewoon zijn voorgeprogrammeerde methode die voor alle identiteiten werkt toepast, probeert een wiskundige voor een ‘really nice proof’ meer te doen dan wat bruut rekenwerk, maar probeert hij de getallen in de identiteit in verband te brengen met andere concepten die hij kent en daarmee te redeneren. Een beslissingsprocedure zoals het WZ-algoritme kan dit uiteraard niet.[95]

9.6.9. WEIERSTRASS' bewijs van de irrationaliteit van e

Beeson ontwikkelde het programma WEIERSTRASS om automatisch continuïteitsbewijzen met epsilon-delta argumenten te vinden. Hoewel het programma nog geen nieuwe stellingen heeft kunnen bewijzen, vermelden we het hier om het contrast te laten zien tussen de werking van de huidige krachtige bewijsprogramma's en bewijsprogramma's zoals WEIERSTRASS die meer als menselijke wiskundigen redeneren. Het programma gebruikt hiervoor zowel berekeningen als redeneringen. De bedoeling is dat een bewijs dat gevonden is door WEIERSTRASS door een menselijke wiskundige kan gelezen en nagekeken worden.[96] Daarin verschilt het programma van bijvoorbeeld OTTER waarvan de bedoeling is om een volledig formeel bewijs te produceren, dat daardoor niet altijd even gemakkelijk door mensen is te begrijpen.

Het programma WEIERSTRASS kan berekeningen uitvoeren in algebra, trigonometrie en analyse. Het programma kan redeneren over ongelijkheden en over onder- en bovengrenzen. De resultaten zijn korte bewijzen die heel leesbaar zijn. Zo bestaan bepaalde stappen in de bewijzen uit berekeningen zoals het factoriseren of het vereenvoudigen van veeltermen, waarbij de details worden weggelaten. Dit soort ‘gaten’ komen voor in menselijke bewijzen, maar logische bewijsprogramma's zoals OTTER willen alle logische stappen verifiëren om een volledig formeel bewijs te bekomen. Het bewijs van de continuïteit van $f(x) = x^3$ door WEIERSTRASS past op een halve pagina en is goed te volgen voor ie-

[95]De wiskundige William Thurston ziet verschillende manieren om een wiskundig concept te interpreteren ook als een belangrijk onderdeel van inzicht in wiskunde. In [Thurston1994] p. 163 geeft hij het voorbeeld van verschillende manieren om over afgeleiden te denken en hij zegt hierover: ‘I can remember absorbing each of these concepts as something new and interesting, and spending a good deal of mental time and effort digesting and practicing with each, reconciling it with the others. I also remember coming back to revisit these different concepts later with added meaning and understanding.’ Ook Benzmüller en zijn collega's zien het belang van verschillende interpretaties in en zien het gebrek aan deze mogelijkheid in het bewijsplanprogramma Ωmega als een minpunt: ‘Important mathematical information that is closely related to a concept is concerned with the different representations for the same concept. Mathematicians are able to switch between these representations whenever this seems to be useful. That is, *the unique* formalization of a problem does not always exists. Rather, mathematical problem solving may swith between equivalent representations. Common examples for these switches are the shift from solving equations to merely structural investigations in algebra, or the shift to arithmetic representation of geometry (due to Descartes).’ ([Benzmüller2001])

[96][Beeson1998]: ‘The intention is, to produce a proof that can be read and checked for correctness by a human mathematician; the standard to be met is “peer review”, just as for journal publication.’

mand met een basiskennis wiskunde. Bovendien vindt het programma dit bewijs geheel automatisch zonder enige inbreng of sturing van de gebruiker.

Beeson heeft het programma WEIERSTRASS daarna uitgebreid met verschillende nieuwe mogelijkheden. Het belangrijkste wapenfeit van de verbeterde versie is een volledig automatisch bewijs van de irrationaliteit van het getal e, voor het eerst bewezen door Euler in 1737.[97] Dit is een niet-triviale stelling, die vaak pas in een gevorderde cursus getaltheorie bewezen wordt. De irrationaliteit van e kan als volgt uitgedrukt worden:

```
∀ p, q ∈ N: q > 0 → |p/q - e| > 0
```

In deze vorm kon WEIERSTRASS geen bewijs vinden van de stelling, maar wel toen Beeson het programma het volgende doel gaf:

```
∀ p, q ∈ N: q > 0 → ∃C ∈ N (|p/q - e| ≥ C/q! > 0
```

Zonder de existentiële variabele C en zonder de noemer q! geraakt het programma niet gestart. De noemer q! geeft het programma een ‘hint’ om de ongelijkheid te vermenigvuldigen met q!, waarna het kan beginnen te vereenvoudigen. Het gevonden bewijs is gelijkaardig aan het klassieke bewijs dat je in de tekstboeken vindt, maar gebruikt een verschillende schatting van een grens aan een een bepaalde oneindige reeks.

Een bepaalde ongelijkheid in het bewijs kan WEIERSTRASS bewijzen door inductie. Het inductiebewijs vergt moeilijke berekeningen en redeneringen die het programma ook volledig automatisch vindt.[98] Het programma bevat overigens berekeningen die algemeen zijn en algemene wiskundige ideeën en technieken implementeren die op heel wat domeinen toepasbaar zijn.d

Een voorbeeld van een berekeningsregel die het programma kent is het ontwikkelen van e in een oneindige reeks:

$$e = \sum_{k=0}^{\infty} \frac{1}{k!}$$

Uiteraard mag het programma niet elke keer dat het e tegenkomt het getal substitueren door de oneindige reeks. Daarom zal WEIERSTRASS dit enkel doen als het op een andere manier geen bewijs vindt. In het bewijs van de irrationaliteit van e is deze subsitutie wel essentieel.

[97] [Beeson2001]

[98] [Beeson2001]: ‘A certain inequality involving factorials is needed in the course of the proof; the program finds a proof of this inequality by mathematical induction. The inductive proof requires some not-quite-straightforward algebraic manipulations to make use of the induction hypothesis; the program also finds these steps automatically.’

9.7. Concepten uitvinden

Om bewijzen te vinden die gebruik maken van concepten die te volgen zijn door mensen, moet een computer weten wat interessante concepten zijn voor mensen. Daarom heeft de onderzoeksgroep van Bundy een programma ontwikkeld dat zelf nieuwe concepten kan uitvinden. Op dit vlak is er slechts beperkt succes. Het programma HR[99] van Colton kan wel nieuwe concepten uitvinden door delen van andere concepten te combineren.[100] Zo heeft het programma bijvoorbeeld het concept *herfactoriseerbaar getal* uitgevonden, een natuurlijk getal waarvan het aantal delers zelf een deler van het getal is.[101] Een voorbeeld is het getal 12, waarvan de delers 1, 2, 3, 4, 6 en 12 zijn. Het aantal delers van 12 is dus 6, wat zelf een deler is van 12. De lijst van deze getallen is zelfs opgenomen in de *On-Line Encyclopedia of Integer Sequences*[102] van Neil Sloane, wat wel betekent dat het geen triviaal concept is, maar het neigt toch meer naar recreatieve wiskunde dan dat het een echt 'diep' wiskundig concept is.[103] Naast de herfactoriseerbare getallen werden nog verschillende andere door HR ontdekte getallenreeksen in de encyclopedie opgenomen.

HR vormt automatisch concepten en vormt vermoedens in een bepaald domein van de wiskunde. Daarna probeert het die vermoedens te bewijzen met OTTER of een tegenvoorbeeld te vinden door de modelgenerator Mace. HR werkt in eindige algebra's zoals groepentheorie en ringtheorie, evenals in grafentheorie en getaltheorie. Het programma verenigt dus de drie factoren die volgens Lakatos in de activiteit van wiskundigen voorkomen: conceptualisatie, vermoedens en bewijzen. Aangezien deze drie aspecten in HR nauw met elkaar verbonden zijn en elkaar sterk beïnvloeden, kan het programma misschien in de toekomst Corfield tegenspreken die denkt dat computers het heel moeilijk zullen hebben om de juiste concepten te vormen.[104]

HR beschikt over een aantal *productieregels* die een al gekend concept kunnen omzetten in andere concepten. Voorbeelden van productieregels zijn de conjunctie van twee concepten, de negatie van een concept, existentiële of universele kwantificatie of splitsingen en complexe samenstellingen van concepten. Van elk gevormd concept schat het programma hoe 'interessant' het is. De interessantste concepten komen het eerst in aanmer-

[99]De afkorting staat voor Hardy-Ramanujan, volgens Colton 'to emphasize both a theory-driven and a data-driven approach to concept formation' ([Colton1999b]). Hardy was een goede wiskundige die Srinivasa Ramanujan, een wonderkind, onder zijn hoede nam. Ramanujan vond dikwijls wonderlijke formules, soms zelfs in zijn dromen, die hij niet kon bewijzen maar die later vaak (met heel wat moeite) bewezen werden door anderen.

[100][Colton1999] en [Bundy1999]

[101]Zie [Colton1999b] voor hoe HR deze getallenreeks ontdekte en welke eigenschappen Colton ervan bewees.

[102]http://www.research.att.com/~njas/sequences/A033950

[103]Colton zegt over de ontdekking: 'The first time HR was tried in number theory, it invented the refactorable numbers. When we first saw this sequence, we did not know how it was found, but it looked interesting - it had a mix of odd and even numbers, sufficiently many terms between one and a hundred, and no obvious pattern. Therefore we looked it up in the Online Encyclopedia, and were surprised to find that it was not listed. Only then did we look at the output from HR to see its definition (expecting an unintuitive, complicated explanation), and were then even more surprised that this sequence was missing from the Encyclopedia.' ([Colton1999b]).

[104][Corfield2003] p. 35

king om een rol te spelen in de afleiding van nieuwe concepten. HR gebruikt de vijf volgende criteria om na te gaan of een concept interessant is:

1. **Parsimonie:** Deze grootheid is omgekeerd evenredig met de grootte van de gegevenstabel van het concept. Hoe eenvoudiger een concept, des te interessanter het is.
2. **Complexiteit:** Deze is omgekeerd evenredig met het aantal productieregels dat gebruikt is in de constructie van het concept. Hoe gemakkelijker het concept te begrijpen, hoe beter.
3. **Nieuwheid:** Deze grootheid is omgekeerd evenredig met het aantal andere concepten dat de gekende groepen in de database hetzelfde categoriseert. Een concept dat een nieuwe categorisatie invoert, is interessanter.
4. **Invariantie:** Het aantal paren van groepen dat correct als dezelfde gecategoriseerd wordt.
5. **Discriminatie:** Het aantal paren van groepen dat correct als verschillende gecategoriseerd wordt.

De verschillende criteria krijgen een bepaald gewicht tussen 0 en 1 en bij elk nieuw concept wordt een gewogen som genomen. De gebruiker kan zelf de gewichten instellen naargelang wat voor concepten hij zoekt.[105] Wanneer HR een nieuw concept uitvindt, zoekt het uit of een al bekend concept dezelfde gegevenstabel heeft. Indien ja, dan genereert het programma het vermoeden dat de concepten equivalent zijn en probeert het dit te bewijzen of een tegenvoorbeeld te vinden.

Colton heeft HR ook toegang gegeven tot de Online Encyclopedia of Integer Sequences en gaf het programma de opdracht om te zoeken naar bekende reeksen waarvan de herfactoriseerbare getallen een subreeks vormen. Zo ontdekte het programma dat deze getallen een subreeks vormen van de getallen congruent met 0, 1, 2 of 5 (mod 8). Dit was op dat moment voor Colton een nog onbekend resultaat en hij bewees het na de ontdekking van HR eenvoudig.[106]

Naast de herfactoriseerbare getallen vond HR in getaltheorie tevens het concept priemgetallen uit en het aantal delers van een getal.[107] In groepentheorie vond het ook een interessant nieuw concept uit: de functie $f(G) = |\{(a, b, c) \in G \times G \times G: a * b = c \wedge b * c = a\}|$ classificeert alle groepen tot orde 6, dat wil zeggen met deze functie kan je bepalen of twee gegeven groepen tot orde 6 isomorf zijn.[108]

[105] Zie [Colton2000] voor een studie van verschillende criteria om te bepalen of een concept of een vermoeden interessant is en een vergelijking van hoe verschillende programma's dit bepalen.

[106] [Colton1999b]

[107] Een uitgebreide lijst van getallenreeksen die HR vond staat in [Colton1999b]. We vinden er ook bijvoorbeeld de kwadraatvrije gehele getallen en de machten van 2 en exotische reeksen zoals getallen waarvan de cijfers allemaal priemgetallen zijn (wat volgens mij meer bij recreatieve wiskunde thuishoort).

[108] [Colton1999]: 'The function […] classifies groups up to order 6 and was genuinely surprising as we hadn't thought such a simple function could perform the task.'

Colton heeft ook 20 definities uit een tekstboek over groepentheorie gekozen en gekeken hoeveel van deze concepten door HR werden ontdekt. Het programma vond de volgende 9 concepten: abelse groep, cyclische groep, groep met exponent 2, elementen, identiteiten, inverse van een element, orde van een element, orde van een groep en centrum van een groep. Voor de volgende 7 concepten had het programma kennis nodig over subgroepen: normale subgroep, quotiëntgroep, coset, indes van een subgroep, eenvoudige groep, centrale serie, afgeleide subgroep. En voor de volgende 4 concepten heeft het programma concepten nodig uit andere domeinen: elementaire abelse groep, dihedralgroep, quaternion en p-groep. Het is interessant om te zien dat het programma problemen heeft met het vinden van concepten die in hun definitie concepten uit verschillende domeinen gebruiken, aangezien ook een aantal hier beschreven bewijsprogramma's problemen hebben met concepten uit verschillende domeinen. Graham Steel heeft daarom in zijn *master's* thesis HR uitgebreid met de mogelijkheid om interessante concepten te vinden waarvan de definitie concepten uit verschillende domeinen gebruikt.[109] Overigens heeft Bundy's onderzoeksgroep recent ook een computerprogramma ontwikkeld dat zelf stellingen kan ontdekken en kan oordelen over het belang van de resultaten, analoog aan wat Coltons programma doet met concepten.[110]

9.8. Conclusie

Als we kijken naar wat eerstejaarsstudenten wiskunde leren, dan gaat het in abstracte algebra om eerste-orde axioma's van groepen en ringen, de concepten subgroep, homomorfisme, isomorfisme en een deel van de theorie van natuurlijke getallen om te kunnen spreken over eindige groepen en de orde van een groep. Van getaltheorie leren de meesten slechts elementaire noties: het begrip deler en de unieke factorisatie van natuurlijke getallen. De stellingen die dan in deze beginnerscursus bewezen worden, zijn volgens Beeson al te moeilijk voor bewijsprogramma's:[111]

> These theorems are presently beyond the reach of automated deduction in any honest sense, although of course one could prepare a sequence of lemmas in such a way that the proof could ultimately be found.

[109][Steel2000]: ‘Cross-domain concepts, while not dense in the mathematical literature, often provide the inspirational step leading to results of real importance. These ideas represent what we think of as creative stages in the development of a theory. An effective automated mathematical discovery package should have the ability to form cross-domain concepts, and therefore be able to provide the inspiration for such creative steps.’

[110][McCasland2006]

[111][Beeson2003]. Hij heeft het onder andere over de structuurstelling van een eindige abelse groep en de stelling dat de multiplicatieve groep van een eindig veld cyclisch is.

Verder denken dan deze eenvoudige theorieën, bijvoorbeeld over de complexere Galois-theorie of algebraïsche topologie, wijst Beeson voorlopig af:

> We propose to not even think about automated deduction in these areas of mathematics. Dealing with the challenges of second-order variables (without quantification), definitions, calculations, incorporating natural numbers, sequences, and induction, should keep researchers busy for at least a generation. At that point computers should have more or less the capabilities of an entering Ph. D. student in mathematics. Now, in 2003, they are at approximately freshman level.

In dit hoofdstuk hebben we aangetoond welke problemen de huidige bewijsprogramma's hebben met verschillende concepten in de wiskunde. Deze problemen hebben als gevolg dat deze programma's die concepten vermijden en op andere manieren een gegeven stelling willen bewijzen. Vaak lukt dit ook, maar slechts na heel wat rekenwerk. Het hoeft dan ook niet te verbazen dat mensen problemen hebben met dit soort bewijzen: ze zien een hoop rekenwerk, geen inzicht in de stelling. Bij computerverificaties van bewijzen, zoals de bewijzen van de vierkleurenstelling, komt dit probleem nog meer tot uiting omdat de wiskundigen het bewijs zo hebben opgesteld dat het computergedeelte uit een grote berekening bestaat. Totdat bewijsprogramma's conceptueler denken, zullen wiskundigen altijd problemen blijven hebben met het begrijpen en aanvaarden van computerbewijzen.

Desalniettemin zien we in de school van Wos een streven naar meer ‘menselijke’ bewijzen. Volgens Veroff, die zelf tot de school van Wos behoort, komen hun automatische bewijsprogramma's heel wat moeilijkheden tegen bij de steeds complexere problemen die men aan hen voorschotelt. Zolang deze bewijsprogramma's niet op meerdere niveaus van abstractie kunnen redeneren (dus met concepten in verschillende domeinen), kunnen deze programma's volgens hem slechts een beperkt type van problemen oplossen.[112]

[112][Veroff2000] p 1: ‘Most of our experience with automated reasoning programs has been with relatively small axiom systems having few distinct predicate and function symbols. As we challenge our systems to work on more complex problems, we will necessarily have to develop more sophisticated techniques and features than those currently available. For example, as axiom systems get larger and more complex, there often will be a natural hierarchy of defined terms. Many proofs will contain distinct subproofs at the various levels of abstraction defined by this hierarchy. Automated reasoning programs typically have not been good at pursuing multiple lines of reasoning simultaneously; mixing arguments at different levels of abstraction –similar to mixing cases of a case analysis– is not conducive to effective reasoning. Until our automated reasoning programs can reason effectively at multiple levels of abstraction, we will be severely limited in the kinds of problems we can seriously consider as applications.’

10. Bewijsplannen en bewijsschetsen

10.1. Inleiding

Naast het gebrek aan concepten in computerbewijzen valt ook op dat bewijzen van OTTER en gelijkaardige bewijsprogramma's lange opeenvolgingen van stappen zijn. Er zit in feite geen structuur in. Je kan wel verifiëren dat alle stappen logisch gelden, maar daarmee begrijp je het bewijs nog niet.[1] Net zomin als de huidige bewijsprogramma's zelf concepten uitvinden, vinden ze vaak ook geen lemma's uit voor het bewijs dat ze aan het zoeken zijn. Een menselijk bewijs zal daarentegen vaak zijn opgedeeld in een aantal lemma's, hulpstellingen. Het bewijs van het hoofdresultaat zal dan gebruik maken van deze lemma's en daardoor relatief kort en begrijpelijk zijn.[2] Een ander voordeel is dat de lemma's in andere contexten ook kunnen gebruikt worden. Het zijn elk op zich bewezen stellingen die hun nut kunnen hebben. Zo werken veel bewijsprogramma's echter nog niet.[3]

Omdat bewijsprogramma's zelf een bewijs niet gemakkelijk kunnen opdelen in lemma's, houden heel wat onderzoekers zich bezig met *bewijsplannen* of *bewijsschetsen*: ze delen een bewijs op in hapklare brokken die elk door een bewijsprogramma kunnen bewezen worden. Alan Bundy stelde in 1988 de notie van *bewijsplan* voor om de zoekruimte van automatische bewijsprogramma's kleiner te maken. [4] Volgens Bundy maken bewijsplannen de bewijzen voor een mens ook veel inzichtelijker. Een bepaalde reeks van stappen in een door de computer gevonden inductiebewijs kan dan 'verklaard' worden door het feit dat de computer de uitdrukking in de vorm van de inductiehypothese wil omzetten.[5] Een programma als OTTER geeft als verklaring slechts dat het een bepaald axioma gebruikt. [6] Automatische bewijsprogramma's die gebruik maken van bewijsplannen genereren bewijzen op een hoger abstractieniveau dan resolutieprogramma's zoals OTTER.[7]

---

[1][Rota1997] p. 181: 'Logical verification alone does not enable us to see the role that a statement plays within the theory. It does not explain how such a statement relates to other results, nor make us aware of the relevance of the statement in various contexts. In short, the mere logical truth of a statement does not enlighten us to the sense of the statement.'

[2]We kwamen dit al tegen in ons 'one-liner' bewijs van Cassini's identiteit, waarin we gebruik maakten van een deelresultaat over een matrix van Fibonaccigetallen.

[3][Beeson2003]: 'The most powerful present-day theorem provers never find, organize, or present their proofs in this way (unless led to do so by a human after a failure to find the proof in one go).'

[4][Bundy1988]: 'We believe that human mathematicians can draw on an armoury of such proof plans when trying to prove theorems. It is our intuition that we do this when proving theorems, and the same intuition is reported by other experienced mathematicians. One can identify such proof plans by collecting similar proofs into families having a similar structure, e.g. those proved by *diagonalization* arguments. Many inductive proofs seem to have such a similar structure.'

[5]Een uitgebreide studie (70 pagina's) van de *rippling*-heuristiek om inductieve bewijzen te berekenen is te vinden in [Bundy1993] en een algemenere uitleg van automatisatie van inductieve bewijzen is te vinden in het hoofdstuk [Bundy2001]. Resultaten van de methode uitgeprobeerd op een aantal standaard stellingen uit de wiskundige literatuur zijn te vinden in [Bundy1991].

Bewijsplannen laten bovendien ook toe dat de gebruiker gemakkelijker met het programma samenwerkt om gecombineerd naar een oplossing te zoeken. De computer kan het bewijsplan uitvoeren en van bepaalde delen van het bewijs laten weten dat het ze niet kan bewijzen. Als de gebruiker ziet waarom het programma de stap wil bewijzen en hoe ze in de rest van het bewijs past, kan de wiskundige het misschien oplossen. Het concept van bewijsplannen is trouwens een speciaal geval van het probleem van plannen in kunstmatige intelligentie: gegeven een toestand en methodes die op de toestand kunnen inwerken, welke methodes moet het programma in welke volgorde uitvoeren om tot een specifieke eindtoestand te komen?[8] In het ideale geval maakt het gebruik van AI-planningstechnieken het zelfs mogelijk dat de computer zelf een bewijsplan opstelt dat onderverdeeld is in mini-bewijsplannen (*methodes*) die het tot zijn beschikking heeft.[9] De onderzoeksgroep van Bundy heeft hiervoor de bewijsplanner CLAM ontwikkeld.[10]

Volgens Bundy is een bewijsplan niet alleen een handigheidje om bewijzen te mechaniseren, het is volgens hem een essentieel onderdeel van een bewijs. Logica alleen is niet voldoende om wiskundig redeneren en redeneren in het algemeen te begrijpen. Logica zorgt voor ons begrip op laag niveau van de verschillende stappen en een bewijsplan zorgt voor het begrip op hoog niveau.[11] De rol van logica is volgens Bundy als volgt:

> Logic provides a low-level explanation of a mathematical proof. It esplains the proof as a sequence of steps and shows how each step follows from previous ones by a set of rules. Its concerns are limited to the soundness of the proof, and to the truth of proposed conjectures in models of logical theories.

Bundy gaat uit van de observatie dat wiskundigen een onderscheid maken tussen het begrijpen van elke stap van een bewijs en het begrijpen van het hele bewijs. Je kan een bewijs op laag niveau begrijpen en niet op hoog niveau of omgekeerd. Begrijpen op laag ni-

[6][Bundy1988]: 'Such meta-level explanations are often more intelligible to the human user, e.g. "I am trying to transform the induction conclusion to make it contain the induction hypothesis," rather than "I am applying axiom 42".'

[7][Richardson1999]: 'Extensive experiments with proof planning reveal that a schema-based approach to automating proof construction works, and has useful properties. As opposed to more uniform proof construction techniques such as rewriting, or resolution, proof planning can generate proofs at a level of abstraction which facilitates human understanding, and can exploit failure productively.'

[8]Melis en Bundy leggen proof planning binnen de brede context van kunstmatige intelligentie uit in [Melis1996]. Zie ook [Bundy1998]. De standaardtechniek voor planning in AI is STRIPS (Stanford Research Institute Problem Solver), voorgesteld in [Fikes1971].

[9]Zo beschrijft [Walsh1992] methodes om sommen van getallenrijen te berekenen en [Bundy1999b] methodes om stellingen over lijsten met de informele notatie $a_1 + \ldots + a_n$ te bewijzen. De onderzoeksgroep van Bundy heeft in [Monroy1994] zelfs een methode ontwikkeld om niet-correcte vermoedens te corrigeren en te bewijzen.

[10][Benzmüller2001] geeft kritiek op de implementatie van bewijsplannen door de onderzoeksgroep van Bundy en geeft enkele suggesties om dit te verbeteren.

[11][Bundy1991b]: 'Logic is not enough to understand reasoning. It provides only a low-level, step by step understanding, whereas a high-level, strategic understanding is also required.'

veau gaat door logica en voor begrip op hoog niveau stelt Bundy het bewijsplan voor. In zijn aanpak is het zelfs mogelijk om van begrijpen op verschillende niveaus te spreken, aangezien een bewijsplan hiërarchisch opgebouwd wordt vanuit deelmethodes.[12] Bundy ijvert ervoor dat we een meta-theorie over bewijsplannen moeten hebben, net zoals we metatheorieën over formele logica's hebben.[13]

We zien hier een parallel met O. Bradley Bassler die in zijn studie van de receptie van het computerbewijs van het vierkleurenprobleem lokale en globale inspecteerbaarheid besprak: een bewijs is lokaal inspecteerbaar wanneer je het stap voor stap kan nagaan op zijn correctheid, terwijl een bewijs globaal inspecteerbaar is wanneer je de structuur op hoog niveau kan overzien. Bundy stelt hier hetzelfde voor, maar dan met de nadruk op het *begrijpen* van een bewijs.

Volgens Robert Veroff kunnen automatische bewijsprogramma's zoals OTTER moeilijkere problemen oplossen als een wiskundige ‘hints’ geeft in de vorm van lemma's die het programma zou kunnen bewijzen.[14] Hij noemt dit *bewijsschetsen*. De zoektocht van OTTER kan namelijk met heel wat parameters geleid worden. Als we denken een bewijsschets te hebben van het te zoeken bewijs, bijvoorbeeld enkele lemma's waarvan we vermoeden dat ze gelden en dat ze kunnen helpen om de stelling te bewijzen, dan geven we redeneerstappen van OTTER die in de richting van de bewijsschets gaan een hogere prioriteit. Veroff heeft met behulp van bewijsschetsen in OTTER verschillende systemen met twee axioma's voor Booleaanse algebra gevonden en in al deze resultaten was de bewijsschets volgens hem essentieel. Het resultaat is echter altijd een bewijs waarin alle stappen aanwezig zijn. Voor Veroff is een bewijsschets dus slechts een pragmatisch middel om het bewijsprogramma op weg te helpen. Voor hem gaat het niet zozeer over de inzichtelijkheid voor mensen, maar om de kracht van het bewijsprogramma.

Mensen als Beeson dromen verder over bewijsprogramma's die hun eigen deductieproces kunnen monitoren en op basis daarvan hun redenering kunnen aanpassen (een soort ‘zelfbewustzijn’ dus).[15] Zo zouden de programma's zelf in staat moeten kunnen zijn om lemma's te bepalen: als het programma merkt dat het een bepaalde formule meerdere keren gebruikt, kan het daar een lemma van maken. Volgens Beeson zal een programma met deze mogelijkheid bewijzen kunnen vinden die te moeilijk zijn voor de huidige bewijsprogramma's.[16] Beeson heeft een eenvoudige vorm van lemmavorming geïmplemen-

[12][Bundy1991b]: ‘In fact, this association provides a multi-level explanation. The proof plan associated with the whole proof provides the top-level explanation. The immediate sub-tactics and sub-methods of this proof plan provide a medium-level explanation of the major sub-proofs. The tactics and methods associated with individual rules of inference provide a bottom-level explanation, which is similar to that already rpovided by Logic.’

[13][Bundy1991b]: ‘Just as Logic also has meta-theories about the properties of and relations between logical theories, we may also be able to develop such meta-theories about proof plans.’

[14][Veroff2001]

[15][Beeson2003]: ‘As it is now, humans using a theorem prover monitor the output, and then change parameters and restart the job. In the future, this kind of feedback should be automated and dynamic, so that the parameters of a run can be altered (by the program itself) while the run is in progress.’

teerd in zijn bewijsprogramma WEIERSTRASS.[17] Het bewijs van de irrationaliteit van e kon door enkele lemma's tot de helft teruggebracht worden en de leesbare uitvoer bestaat nu uit iets meer dan 5 pagina's.[18]

Ook in andere onderzoeksdisciplines die over computers en bewijzen gaan, zien we een ontwikkeling naar meer gestructureerde bewijzen. Heel wat onderzoekers zijn bezig met het minutieus formaliseren van bewijzen, die dan volledig door een computerprogramma kunnen geverifieerd worden. We zagen al voorbeelden hiervan: het geformaliseerd bewijs van de vierkleurenstelling door Georges Gonthier in COQ en het Flyspeck-project van Thomas Hales om hetzelfde te doen met het bewijs van het Keplervermoeden. Dit soort bewijzen zijn wel heel goed te vertrouwen omdat alle stappen gecontroleerd worden, maar ze zijn vaak moeilijk te begrijpen door mensen (zie wat we hiervoor gezegd hebben over de gebruikte concepten in Gonthiers bewijs van de vierkleurenstelling), ze lijken meer op computerprogramma's dan wiskundige bewijzen.[19]

De Nijmeegse wiskundige Freek Wiedijk introduceerde hiervoor de notie *formele bewijsschets*.[20] Dit is een bewijs dat het midden houdt tussen een volledig geformaliseerd, verifieerbaar bewijs en helemaal geen bewijs. Eigenlijk komt dit overeen met ons concept van bewijs. Een wiskundig bewijs door mensen gegeven laat ook altijd gaten open om niet te verdrinken in de details en om de structuur duidelijker te laten zien. Wiedijk heeft de notie van formele bewijsschets ontwikkeld als antwoord op de verzuchting dat wiskundigen zich niet inlaten met de formalisering van wiskunde omdat het niet lijkt op de wiskunde waar zij zich mee bezig houden.[21] Volledig formele bewijzen zijn namelijk veel te lang en wie ze wil bestuderen, verliest al vlug het overzicht.[22] Hij beweert dat de

[16][Beeson2003]: 'Giving a program the ability to formulate its own lemmas dynamically might, in conjunction with the ability to modify the criteria for keeping or using deduced formulas, enable the program to find proofs that might otherwise be beyond reach.'

[17][Beeson2001]: 'The prover keeps a list of lemmas; it records sequents in this list on command, and the first thing it does when trying to prove a goal is to compare the goal to the list of lemmas.'

[18]Het bewijs is wat gestructureerder dan een menselijke wiskundige het zou maken, maar toch heel leesbaar. De bewijzen van WEIERSTRASS lijken wat op de 'gestructureerde bewijzen' die Leslie Lamport voorstelde in zijn artikel 'How to write a proof' ([Lamport1995]): het bewijsplan wordt duidelijk uit de structuur op hoog niveau en elk deelresultaat bestaat zelf weer uit een bewijsplan, totdat de details op het laagste niveau zijn uitgewerkt. Je kan het volledige bewijs met al zijn details bekijken wanneer je het bewijs wil verifiëren, maar je beperken tot het bewijsplan op een hoger niveau wanneer je het wil begrijpen.

[19]Henk Barendregt en Freek Wiedijk geven het voorbeeld van een bewijs van de lineariteit van de limiet van een functie in COQ: 'Not even a COQ specialist will be able to understand what is going on in this proof without studying it closely with the aid of a computer.' ([Barendregt2005]). Het bewijs in COQ is nochtans niet zo lang: 22 regels.

[20][Wiedijk2003], [Wiedijk2004]

[21][Wiedijk2004]: 'At the TYPES workshop of 2002 in Nijmegen, Peter Aczel claimed that in order to get mathematicians involved with the formalization of mathematics, technology is needed for *reasoning with gaps*, where one can leave out the details of a formalization that one considers to be obvious or well-known, and one only needs to formalize the interesting parts.'

[22][Wiedijk2003]: 'Generally, the problem with a formal proof is that it is big (because the steps are small), and that one looses grip on the whole because of this. […] we believe that to make a formalization accessible, one has to make it smaller.'

informele bewijzen van wiskundigen goed kunnen uitgedrukt worden als formele bewijsschetsen in een programma als MIZAR.

10.2. Redeneringen en berekeningen

We hebben hiervoor in verschillende voorbeelden gezien dat complexere bewijzen vaak uit een samenspel van redeneringen en berekeningen bestaan. Dit onderscheid gaat in wezen terug tot de Babyloniërs en de Grieken in de oudheid: Babylonische wiskundigen waren goed in berekeningen en algoritmes, maar kenden niet ons idee van bewijs. De Griekse wiskundigen hadden een concept van bewijs door redeneringen, maar als het om berekeningen ging waren zij minder goed.[23]

Een eenvoudige stelling kan je nog met enkel berekeningen of enkel redeneringen bewijzen, maar als het wat te complex wordt, ben je vaak beter af met een samenspel van berekening en redenering. De redenering zorgt voor het 'bewijsplan' op hoog niveau en het verbinden van verschillende concepten, wat het bewijs inzichtelijk en interessant voor wiskundigen maakt. De berekeningen vormen dan het gedeelte van het bewijs dat meer voor een verificatie van de verschillende stappen in de redenering zorgt. Een voorbeeld hiervan vinden we in een artikel van de wiskundige David Mumford, die bij de stelling dat een bepaalde functie correct gedefinieerd is schrijft:[24]

> The proof of this Proposition is a ghastly but wholly straightforward set of computations. It took me several hours to do every bit and as I was no wiser at the end –except that I knew the definition was correct– I shall omit details here.

Volgens Yuri Manin ervaren vele wiskundigen zoiets bij 'mechanical proofs, even ones done by hand'.[25] Zijn moraal is: een goed bewijs maakt ons wijzer. In bepaalde gevallen aanvaarden wiskundigen wel een berekening als bewijs. Als het om een klein tussenresultaat gaat dat geen verdere uitleg nodig heeft bijvoorbeeld. Zo schrijft Dominique Pastre, die het MUSCADET bewijsprogramma ontwierp:[26]

> If a mathematician uses an automated theorem prover as an assistant and only wants to verify a minor point, a black box which answers a yes or no is sufficient, but if he wants to study a delicate point, it is necessary that the prover explains itself by giving important reasons and not tedious argumentation.

[23] De reden dat Euclides bijvoorbeeld zoveel problemen had met eenvoudige algebraïsche vergelijkingen is dat hij ze eerst omzette naar meetkundige proposities en ze zo al redenerend probeerde te bewijzen. Algebra kende hij niet. ([Barendregt2002] p. 322

[24] [Mumford1967] p. 230

[25] [Manin1981] p. 107

[26] [Pastre1999] p. 6. Op p. 7 geeft Pastre het voorbeeld van het factoriseren van veeltermen: daar hoeft een wiskundige vaak niet de redenering achter de berekening te weten en is het resultaat voldoende.

10.3. Redeneringen en berekeningen in computerbewijzen

Volgens Beeson behandelen de huidige computerprogramma's de twee gedeeltes van een bewijs (berekeningen en redeneringen) volledig apart:[27]

> What is interesting, and surprising to people outside the field, is that the mechanization of logic and the mechanization of computation have proceeded somewhat independently. We now have computer programs that can carry out very elaborate computations, and these programs are used by mathematicians 'as required'. We also have 'theorem-provers', but for the most part, these two capabilities do not occur in the same program, and these programs do not even communicate usefully.

Elders beschrijft Beeson het samenspel van berekeningen en redeneringen als volgt:[28]

> Typically computational steps move 'forwards' (from the known facts further facts are derived) and logical steps move 'backwards' (from the goal towards the hypothesis, as in *it would suffice to prove*). The mixture of logic and computation gives mathematics a rich structure that has not yet been captured, either in the formal systems of logic, or in computer programs.

Programma's als OTTER, die gebaseerd zijn op de resolutiemethode, zijn gespecialiseerd in redeneringen, maar voeren bijna geen berekeningen uit.[29] Ze hebben geen 'wiskundige kennis' en kunnen daardoor berekeningen of substituties die voor ons eenvoudig zijn niet uitvoeren.[30] Wiskundigen kunnen dit wel omdat zij door ervaring een 'bag-of-tricks' hebben opgebouwd met technieken om stellingen van een bepaalde vorm te bewijzen. Als een wiskundige een nieuwe stelling wil bewijzen, kan hij terugvallen op zijn redeneervermogens, zijn rekenvermogens en trucendoos van standaardtechnieken.[31] Beeson heeft met WEIERSTRASS een bewijsprogramma gemaakt dat ook op wiskundige kennis kan terugvallen en dus meer de menselijke manier van bewijzen overneemt.[32]

Manfred Kerber, Michael Kohlhase en Volker Sorge stellen voor om de berekeningen

[27][Beeson2003]. Ook Kerber wijst op deze tweedeling in [Kerber1998] p. 327-328: 'This has lead to two rather disjoint academic fields; mechanised reasoning and computer algebra, which each have their own methods, interests and traditions, even though they share common roots.'

[28][Beeson1998]

[29]Beeson noemt deze bewijsprogramma' 'quite limited in their computational abilities' ([Beeson1998]). Zie [Davis2001] voor een historisch overzicht van het begin van deze bewijsprogramma's.

[30][Beeson1998]: 'The program contains no mathematical knowledge except that supplied in the axioms; it only "knows" the laws of logic.'

[31][Beeson2001]: 'The driving idea of this research program is that finding mathematical proofs requires expert knowledge of hundreds of special inference rules. An inference rule typically encapsulates knowledge of how to prove theorems of a certain form or in a certain context.'

van computeralgebrapakketten te integreren in automatische bewijsprogramma's op het niveau van bewijsplannen.[33] Anderen proberen volledig formele bewijzen te leveren van berekeningen in een bewijs, zoals IsPrime(61) of $(x+1)^2 = x^2 + 2x + 1$, maar dit doet de integratie tussen berekeningen en redeneringen dan eigenlijk teniet, omdat het het ene reduceert tot het andere.[34]

Natarajan Shankar wijst erop dat het onderzoeksdomein van geautomatiseerde bewijsprogramma's al van in het begin in twee scholen is opgesplitst. De resolutiemethode van Alan Robinson[35] is gebaseerd op een eenvoudige uniforme methode om bewijzen te zoeken, aangevuld met algemene heuristieken om de zoektocht efficiënter te laten verlopen.[36] Shankar vindt de populariteit van uniforme bewijsmethodes zoals resolutie onterecht. Volgens hem hangt de populariteit af van het 'dogma' dat, omdat eerste-orde logica een algemene taal is om wiskundige uitdrukkingen in voor te stellen, algemene eerste-orde bewijszoekmethodes goed genoeg zijn om bewijzen te vinden. Deze redenering verwart volgens Shankar het uitdrukken van een probleem en het oplossen van een probleem.[37]

De andere school, geleid door Hao Wang, ijvert voor probleemspecifieke combinaties van beslissingsmethodes en semi-beslisbare procedures. Volgens Shankar heeft deze laatste school veel meer perspectieven, hoewel de school van Robinson lange tijd dominant is geweest.[38] Er zijn heel wat specifieke 'inference engines' ontwikkeld voor specifieke domeinen en door die te combineren kunnen bewijsprogramma's volgens Shankar veel krachtiger worden. Dit is in essentie hetzelfde als wat Beeson zegt: de domeinspecifieke beslissingsprocedures injecteren wiskundige kennis in het bewijsprogramma door het programma toe te laten met hoogniveau concepten te redeneren en te rekenen. Bewijsprogramma's met toegang tot domeinspecifieke procedures kunnen in Wangs aanpak een bewijs reduceren tot een combinatie van problemen die met behulp van de procedures opgelost kunnen worden.[39]

Alan Bundy, die zelf tot de tweede school behoort, is zich ook heel bewust van de voor-

[32][Beeson2001]: 'This strategy of embodying mathematical knowledge in inference rules is the key to our success: the inference rules can use mathematical knowledge to instantiate metavariables, rather than relying on unification. Unification (even the more sophisticated versions of it) is a very primitive way to instantiate a metavariable.'

[33][Kerber1998] De algoritmes van het computeralgebrapakket kunnen op die manier gedefinieerd worden als methodes die domeinspecifieke kennis implementeren. Een bewijsplan kan dan van deze domeinspecifieke methodes gebruik maken. De algoritmes moeten dan wel extra informatie genereren zodat het bewijsprogramma op basis daarvan een bewijsplan kan creëren. Bestaande computeralgebrapakketten moeten daarvoor grote veranderingen ondergaan. Kerber en zijn collega's stellen in hun artikel een softwarearchitectuur voor deze integratie voor.

[34][Barendregt2002]

[35][Robinson1965] p. 1

[36][Shankar2002] p. 1

[37][Shankar2002] p. 3: 'This central dogma seems absurd on the face of it. Stating a problem and solving it are quite separate matters.'

[38][Shankar2002] p. 1

onderstellingen van de twee scholen. Zo is volgens hem de eerste school vooral gemotiveerd door het vinden van nieuwe resultaten, zonder de nood om de resultaten te verstaan. Vandaar dat zij dankbaar gebruik maken van empirisch succesvolle bewijsprogramma's die een grote zoekruimte afschuimen.[40] Op lange termijn zullen deze programma's volgens Bundy echter tegen hun grenzen aanlopen en steeds moeilijker nieuwe resultaten behalen.[41]

Onderzoekers zoals Bundy zijn volgens hemzelf meer gemotiveerd door het begrijpen van bewijzen en daarvoor maken zij gebruik van bewijsplannen. Initieel (en we zitten nu nog altijd in de initiële fase) zijn de bewijsplannen niet zo goed en kunnen de bewijsprogramma's zelfs niet zo veel stellingen bewijzen, maar volgens Bundy heeft deze aanpak geen 'deadlock' op lange termijn.[42] Op korte termijn ziet hij echter geen betere bewijsprogramma's die gebruik maken van bewijsplannen.[43]

Een deel van de communicatieproblemen tussen computeralgebrapaketten en bewijsprogramma's is dat de eerste meestal logisch incorrect zijn. Computeralgebrasoftware voert namelijk vaak bewerkingen uit zonder vooronderstellingen te testen. Zo zal deze software uitdrukkingen vereenvoudigen zonder te testen of hierdoor door nul gedeeld wordt.[44] Het verbinden van computeralgebrapaketten en bewijsprogramma's om conceptueel en structureel rijkere bewijzen te maken, zal in de praktijk dus nog heel moeilijk worden.[45] De relatie tussen bewijsprogramma en computeralgebrapakket kan drie vormen hebben: het

[39][Shankar2002] p. 2: 'Central to his approach was the use of domain-specific decision and semi-decision procedures, so that proofs could be constructed by means of reductions to some combination of problems that could each be easily solved.', p. 4: 'Few problems are stated in a form that is readily decidable, but proof search strategies, heuristics, and human guidance can be used to decompose these problems into decidable subproblems.'

[40][Bundy1991b]: 'I take the conventional motivation of automated theorem proving to be the building of theorem provers which are empirically successful, without any necessity to understand why. The methodology is implied by this motivation. The theorem prover is applied to a random selection of theorems. Unsuccessful search spaces are studied in a shallow way and crude heuristics are added which will prune losing branches and prefer winning ones.'

[41][Bundy1991b]: 'This process is repeated until the law of diminishing returns makes further repetitions not worth pursuing. The result is fast progress in the short term, but eventual deadlock as different proofs pull the heuristics in different directions.'

[42][Bundy1991b]: 'Automatic theorem provers based on proof plans make slower initial progress. Initial proof plans have poor generality, and so few theorems can be proved. The motivation of understanding proofs mitigates against crude, general heuristics with low prescriptiveness and no expectancy. […] However, there is no eventual deadlock to block the indefinite improvlement of the theorem prover's performance.'

[43][Bundy1991b]: 'Thus, we expect a science of reasoning will help us build better automatic theorem proving programs in the long term, although probably not in the short term.'

[44]Beeson geeft een voorbeeld: 'Start with the equation a = 0. Divide both sides by a. In all the three systems mentioned, you can get 1 = 0 since the system thinks a/a = 1 and 0/a = 0.' ([Beeson1998]). De drie systemen waarover hij het heeft zijn MATHEMATICA, MAPLE en MACSYMA. Barendregt en Cohen geven een ander voorbeeld: vraag aan een computeralgebrapakket de integraal $\int x^a dx$ en je krijgt als antwoord $x^{a+1}/(a+1)$, terwijl dit niet geldt voor a = -1 ([Barendregt2001]).

[45][Shankar2002] p. 2: 'Such software libraries have not taken root in automated deduction because the scientific and engineering challenges involved are quite significant.'

bewijsprogramma kan alle resultaten van het computeralgebrapakket geloven, het kan sceptisch zijn of 'autarkisch' (zelfvoorzienend). In het eerste geval is de integratie enkel een kwestie van de resultaten doorgeven aan het bewijsprogramma, dat de resultaten dan als axioma's aanneemt, in het tweede geval moet het computeralgebrapakket naast het resultaat ook zogenaamde 'getuigen' van het resultaat meeleveren, waarmee het bewijsprogramma kan verifiëren of het resultaat klopt.[46] In het derde geval zet het bewijsprogramma zelf de berekening om in een formeel bewijs bestaande uit een opeenvolging van redeneerstappen, waardoor het geen computeralgebrapakket nodig heeft.[47]

Beeson heeft wel een experiment gedaan met zijn zelf geschreven computerprogramma WEIERSTRASS dat redeneren en rekenen combineerde. Het programma vond automatisch epsilon-deltabewijzen van de continuïteit van een aantal functies, zoals x^n, $\sqrt{x}$, log x, sin x en cos x. De bewijzen bevatten algebraïsche berekeningen en redeneringen over ongelijkheden. Na een hoop verbeteringen slaagde WEIERSTRASS er zelfs in om automatisch een bewijs van de irrationaliteit van het getal e te vinden. Het bewijs bevat een breed gamma aan redeneringen en berekeningen, zoals ongelijkheden, boven- en ondergrenzen van oneindige reeksen, een deelbewijs door inductie, het kent het verschil tussen natuurlijke en reële getallen en het kan uitdrukkingen met faculteiten en sommen van oneindige meetkundige reeksen vereenvoudigen. Beeson beschrijft dat hij in lezingen werd aangesproken door mensen die dachten dat WEIERSTRASS nu wel vlug de irrationaliteit van Eulers constante γ zou kunnen bewijzen, een nog altijd onopgelost vermoeden, maar zo ver is het niet gekomen. Het volledig geautomatiseerde bewijs van de irrationaliteit van e is echter al een grote prestatie die aantoont waartoe bewijsprogramma's in staat zijn als ze kunnen putten uit een breed gamma aan concepten en berekeningen.

We zien dus dat als bewijsprogramma's hun eis van volledig formele redeneringen laten vallen en willen gebruik maken van berekeningen op hoger niveau en een mengeling van concepten op verschillende niveaus, ze minder moeite hebben met moeilijkere stellingen. WEIERSTRASS' bewijs van de irrationaliteit van e is zo'n 'moeilijke' stelling.[48] Bovendien hebben mensen helemaal geen problemen om berekeningen als onderdeel van een bewijs aan te nemen, zolang de berekeningen maar niet overheersen. In hun eigen bewijzen verwijzen wiskundigen ook vaak naar berekeningen, bijvoorbeeld met zinnen als 'Deze berekening laten we over aan de lezer als oefening', dus verwijzingen als 'Deze berekening is door de computer uitgevoerd' zouden, tenminste voor niet al te ingewikkelde bereke-

[46]Een voorbeeld wordt gegeven in [Barendregt2001] p. 19. Wanneer het bewijsprogramma aan het computeralgebrapakket vraagt gcd(22, 30), dan kan het algebrapakket naast het antwoord 2 (de grootste gemene deler van 22 en 30) ook de getuigen x = -4 en y = 3 leveren. Het 'uitgebreide Euclidische algoritme' voor het berekenen van de grootste gemene deler van twee getallen a en b geeft deze twee getallen terug die moeten voldoen aan de vergelijking `ax + by = gcd(a, b)`. Het bewijsprogramma kan met deze vergelijking het resultaat verifiëren (al is dat ook een berekening). In [Caprotti2001] wordt een sceptisch bewijsprogramma voorgesteld dat met behulp van een computeralgebrapakket een formeel bewijs van het priem zijn van grote getallen kan leveren.

[47][Barendregt2002]

[48][Beeson2001]: 'This is part of the reason that no previous computer program has produced such proofs: most such programs insist on producing formal proofs.'

ningen, zeker mogelijk moeten zijn.[49]

Naast de effectiviteit van het gebruik van hoogniveau concepten en de bijbehorende berekeningen in bewijsprogramma's geeft Beeson nog een reden om verder te gaan met dit onderzoek: als we willen dat de computer voor ons leesbare bewijzen produceert, moeten we zulke berekeningen toelaten en mogen we niet blijven eisen dat het geproduceerde bewijs volledig formeel is. Beeson ziet dan drie mogelijkheden om hierop te reageren:

1. Neem genoegen met computerbewijzen zoals die van WEIERSTRASS die niet volledig formeel zijn en door mensen worden nagekeken.
2. Geef een logisch bewijs voor alle berekeningen en vereenvoudigingen in het bewijs van WEIERSTRASS en substitueer in het bewijs de vereenvoudigingsstappen telkens door hun bewijs, waardoor we een volledig formeel bewijs bekomen.[50]
3. Vraag bovendien dat de correctheid van alle gebruikte algoritmes bewezen is.

Volgens Beeson voldoen menselijke bewijzen niet eens aan eisen 2 en 3, dus volgens hem is het maar redelijk om van een computerbewijs hetzelfde te vragen.[51]

Overigens profiteren niet alleen bewijsprogramma's van een koppeling met computeralgebrasystemen. Ook computeralgebrasystemen kunnen profiteren van de hulp van bewijsprogramma's. Zij kunnen gebruik maken van de redeneervermogens van een bewijsprogramma om de randvoorwaarden van berekeningen te bewijzen en zo resultaten te kunnen geven die gegarandeerd juist zijn. De hiervoor al geciteerde problemen met computeralgebrasystemen die 0 = 1 als uitkomst geven kunnen zo tegengegaan worden. Een voorbeeld van deze richting van integreren vinden we in een project van Andrew Adams en anderen die het computeralgabrasysteem MAPLE geïntegreerd hebben met het bewijsprogramma PVS.[52] Vooral in symbolische integratie is er nood aan het bewijzen van randvoorwaarden.[53]

[49][Martin1999]: ‘Thus for a less significant result, like many of the computations done by computer algebra systems as part of a larger endeavour for example, the community seems to find no problem with admitting a computation as part of a proof provided the authors are trusted to have used the system properly.’

[50]Het gaat hier om stappen zoals $x^2 + 2xy + y^2 = (x + y)^2$. In WEIERSTRASS' bewijs van de irrationaliteit van e zitten volgens Beeson zo'n 1800 vereenvoudigingsstappen. Een volledige formalisering van dit bewijs zou dus heel complex zijn.

[51][Beeson2001]: ‘Human mathematicians do not meet staandards (2) and (3) either, so it seems reasonable to demand of a machine intended as a prototype “mathematician's assistant” that it should meet the standards required for journal publication, instead of a higher standard.’

[52][Adams2001]: ‘Our objective is to provide the CAS user with an interface to an ATP, so that certain side conditions which are implicit in many analytical symbolic computations can be highlighted, checked and either verified or flagged as an error.’ en ‘Our approach takes the view point of a user of a CAS who simply wishes it to be more robust.’

[53][Adams1999]

10.4. Goede bewijsplannen

Als wiskundigen verschillende bewijzen van dezelfde stelling evalueren en met elkaar vergelijken, is één van de criteria de gebruikte concepten, zoals we in het vorige hoofdstuk gezien hebben. Een ander, veel vaker aangehaald criterium is de structuur van het bewijs. In computerbewijzen kunnen bewijsplannen en bewijsschetsen de structuur van het bewijs blootleggen. In menselijke bewijzen wordt dit soort 'hoogniveau' presentatie van een bewijs vaak een *proof outline* genoemd. We zullen deze drie termen alledrie met de naam 'bewijsplan' benoemen. Of een bepaald bewijs inzichtelijk is, hangt uiteraard af van het bewijsplan (of het gebrek eraan). Jean-Paul Van Bendegem beweert dat wiskundigen de 'kwaliteit' van een bewijs als hoog beschouwen wanneer het bewijsplan eenvoudig is. Wanneer het bewijs een complex bewijsplan heeft, is de kwaliteit van het bewijs laag. [54]

Van Bendegem geeft twee redenen voor wiskundigen om de voorkeur aan bewijzen van hoge kwaliteit te geven. Ten eerste is een bewijs met een kort bewijsplan vaak gemakkelijk te generaliseren. Van Bendegem geeft als voorbeeld een bekend bewijs van de irrationaliteit van de vierkantswortel van 2. Dit bewijs (en het bewijsplan) is zo eenvoudig dat het gemakkelijk aan te passen is zodat het werkt voor alle priemgetallen in plaats van enkel voor 2.[55]

Een andere reden om bewijzen van hoge kwaliteit te kiezen is dat zij volgens Van Bendegem meer inzicht geven. Van Bendegem identificeert de 'verklaring' van een bewijs met zijn bewijsplan. [56] Een bewijs met een hoge kwaliteit zoals hiervoor gedefinieerd is dus inzichtelijk. Vreemd is dat Van Bendegem dit illustreert met de computerbewijzen van de vierkleurenstelling, aangezien die in het algemeen als niet-inzichtelijk gezien worden:[57]

> Part of the proof consists of the methodical-mechanical checking of a (large) finite set of highly complicated maps. What is required is for the computer to succeed in coloring them all. The two last sentences are a proof-outline for that part of the full proof. And really there is not that much more one could say about it. The details as to how the computer actually does the coloring, are hardly (within this context) interesting. That is all the explaining there is to do.

Van Bendegems voorstel om de lengte van het bewijsplan als aanduiding van de kwaliteit

[54][VanBendegem1988] p. 252: 'If a proof has a simple proof-outline, then the quality of that proof is considered to be high by mathematicians. Conversely, if the proof has a highly complicated proof-outline, then the quality of the resulting proof is low.'

[55][VanBendegem1988] p. 252

[56][VanBendegem1988] p. 253, voetnoot 2: '[But] mathematicians themselves do use the word ["explanation"] and frequently so. It appears often in connection with "providing insight", "clear", "convincing", etc. It is within that cluster of concepts, I want to define the thesis that it makes sense to define the explanation of a proof as its proof-outline.'

[57][VanBendegem1988] p. 252

van het bewijs te zien, zou de computerbewijzen van de vierkleurenstelling een grote kwaliteit toedienen, aangezien het bewijsplan vrij kort is. Toch is dit niet zo, wat Van Bendegem ook opmerkt.[58] Hij kan dit echter niet verklaren met zijn identificatie van de inzichtelijkheid van een bewijs met de lengte van het bewijsplan. Er zijn dus extra criteria nodig. Een lijst van criteria om bewijsplannen te evalueren vinden we bij Bundy in zijn artikel 'A science of reasoning'. Hij beschouwt zijn meta-theorie van bewijsplannen als een wetenschap en stelt dan ook voor om algemene wetenschappelijke principes te gebruiken om criteria op te stellen.[59] Bij Bundy vinden we de volgende criteria:

- **Intuïtiviteit:** Uiteraard is dit een heel subjectief criterium, wat Bundy ook toegeeft. Met dit criterium kan je bijvoorbeeld geen discussie tussen rivaliserende intuïties beslechten.
- **Psychologische geldigheid:** Je kan experimenten uitvoeren op wiskundigen om te onderzoeken hoe zij bewijzen structureren. Dit kan een wetenschappelijkere manier zijn om de intuïtiviteit te evalueren.
- **Verwachting:** Je moet het succes van een bepaald bewijsplan zo goed mogelijk kunnen voorspellen.
- **Algemeenheid:** Een algemeen bewijsplan dat heel veel stellingen kan bewijzen wordt in het algemeen als goed beschouwd. Maar dit is niet alles: een bewijsplan zoals de resolutiemethode van Robinson[60] dat alle stellingen kan bewijzen als het maar lang genoeg kan zoeken, wordt niet vaak als een goed bewijsplan beschouwd, tenzij je enkel resultaten wil behalen.
- **Gedetailleerdheid:** Een bewijsplan dat in detail uitschrijft welke stappen je moet uitvoeren, zorgt ervoor dat je (= de wiskundige of het computerprogramma) minder lang moet zoeken naar een bewijs. Een nadelig effect is wel dat je overgedetailleerde bewijsplannen bekomt.
- **Eenvoud:** Een goed bewijsplan is eenvoudig. Dit is het criterium van Van Bendegem.
- **Efficiëntie:** Een bewijsplan is (voor de computer) goed wanneer het computationeel efficiënt is.

Bundy voegt daar nog het criterium *parsimonie* aan toe voor het evalueren van een verzameling bewijsplannen: hoe minder algemene bewijsplannen je nodig hebt om een verzameling van bewijzen te leveren, hoe beter.

Een ander bewijs dat vaak wordt vermeld als een 'probleemgeval' voor de inzichtelijkheid is de *classificatie van de eindige eenvoudige groepen*, een bewijs van 15000 pagi-

[58][VanBendegem1988] p. 253: 'For many mathematicians the four-color theorem is not proved yet.'

[59][Bundy1991b]: 'It only remains to propose criteria for associating proof plans with proofs that will enable us to prefer one proof plan to another. This we can do by appealing to general scientific principles.' en 'These criteria are just the sort one would apply to any scientific theory, that is we have merely treated the understanding of mathematical proofs as we would any other object of scientific study.'

[60][Robinson1965] p. 1

na's verspreid over honderden artikels.[61] Volgens Michael Aschbacher zullen wiskundigen in de toekomst meer en meer met dit soort stellingen te maken krijgen die lange bewijzen hebben. Terwijl de bewijzen van de vierkleurenstelling nog een kort bewijsplan hebben, is dit ver te zoeken bij de classificatie van de eindige eenvoudige groepen: hiervan is geen bewijsplan bekend.[62] Volgens Aschbacher moeten we in de toekomst dan ook minder belang hechten aan eenvoud en elegantie en meer aan toepasbaarheid en gevolgen.[63]

Deze criteria bezorgen ons geen absolute oordelen over bewijsplannen. Zo kan een bepaald bewijsplan voor een bewijs goed scoren op algemeenheid, maar slecht op eenvoud. De resolutiemethode van Robinson, geïnterpreteerd als bewijsplan, scoort goed op algemeenheid, maar slecht op gedetailleerdheid. Het bewijsplan van de computerbewijzen van de vierkleurenstelling scoort goed op eenvoud, maar slecht op gedetailleerdheid.[64] Het afwegen van de verschillende criteria is niet voldoende om van een bepaald bewijsplan te zeggen dat het absoluut goed is.[65] Fieldsmedaillewinnaar Terence Tao maakt dezelfde observatie in zijn recent artikel ‘What is good mathematics?’ over wiskunde in het algemeen.[66]

[61]Michael Aschbacher geeft een goed overzicht van het bewijs in [Aschbacher2004].

[62][Aschbacher2005] p. 2401: ‘It would be difficult to establish exactly which papers are actually a necessary part of the proof, and I know of no published outline.’

[63][Aschbacher2005] p. 2404: ‘My guess is that we will begin to encounter many more such problems, theorems, and proofs in the near future. As a result we ill need to re-examine what constitutes a proof, and what constitutes a good proof. Elegance and simplicity should remain important criteria in judging mathematics, but the applicability and consequences of a result are also important, and sometimes these criteria conflict. I believe that some fundamental theorems do not admit simple elegant treatments, and the proofs of such theorems may of necessity be long and complicated. Our standards of rigor and beauty must be sufficiently broad and realistic to allow us to accept and appreciate such results and their proofs.’

[64]Jean Mayer wees al op dit contrast: we moeten ons volgens hem niet laten overdonderen door het grote aantal geteste gevallen in het bewijs van de vierkleurenstelling, want het bewijsplan (*le schéma logique*) is heel eenvoudig. ([Mayer1982] p. 54: ‘Il faut distinguer, dans la démonstration d'Appel et Haken, l'ampleur combinatoire (de l'ordre de 10^{10} bits) et le schéma logique très simple.’)

[65][Bundy1991b]: ‘Nor can we normally expect a precise, mathematical proof that one proof plan is better than another on some dimension. Often the best we can do is appeal to an informal argument that one proof plan is *usually* better than the other on some dimension.’

[66]Tao somt in [Tao2007] p. 1-2 een aantal vormen van ‘goede wiskunde’ op en voegt daaraan toe: ‘As the above list demonstrates, the concept of mathematical quality is a high-dimensional one and lacks an obvious canonical total ordering. I believe this is because mathematics is itself complex and high-dimensional and evolves in unexpected and adaptive ways; each of the above qualities represents a different way in which we as a community improve our understanding and usage of the subject. There does not appear to be universal agreement as to the relative importance or weight of each of the above qualities.’

10.5. Het toevalsaspect in combinatorische bewijsmethodes

Problemen die wiskundigen hebben met computerbewijzen zijn niet altijd terug te voeren tot de gebruikte concepten. In het bewijs dat er geen eindig projectief vlak van orde 10 bestaat, wordt met vectoren en gewichtsenumeratoren gewerkt, waar menselijke wiskundigen toch redelijke intuïties over hebben en waarmee ze goed kunnen redeneren. Het probleem ligt daar meer in het case-testing karakter van het bewijs. Andere bewijzen van deze soort, het type dat we in Hoofdstuk 8, *Een classificatie van bewijstechnieken in computerbewijzen combinatorische oplossingen* genoemd hebben, zijn onder andere de bewijzen van de vierkleurenstelling en een deel van Hales' bewijs van het Keplervermoeden. De algemene reactie tegen dit soort bewijzen wordt door Ursula Martin als volgt beschreven: [67]

> Good theorems should have good proofs. To paraphrase von Neumann a ‘good’ theorem should not just consist of an enumeration of special cases but have some unifying element: a ‘good’ proof will be elegant rather than a rote computation (by hand of machine) and give some sense of why a result is true, and the whole will exhibit some overall architectural structure which reduces complexities to simple guiding notions. In particular a proof with these qualities is more likely to be readily surveyable by the critical reader.

Ook in probabilistische bewijzen ligt het probleem niet zozeer in de gebruikte concepten. Een probabilistische priemtest is conceptueel namelijk perfect te volgen. De gebruikte concepten zoals priemgetallen en machten modulo een getal zitten al honderden jaren in de toolbox van wiskundigen. Het niet-deductieve karakter van deze bewijsvormen zorgt echter voor het probleem.

In dit hoofdstuk vergelijken we de (deductieve) combinatorische bewijsmethode met de (niet-deductieve) probabilistische bewijsmethode. We stippen aan wat de verschillen tussen beide methodes zijn en welke eigenschapen ze gemeenschappelijk hebben. Vervolgens herhalen we de argumenten die wiskundigen en filosofen tegen beide bewijsmethodes hebben gegeven. We laten tot slot zien dat deze argumenten tegen beide bewijsmethodes op hetzelfde neerkomen. In zekere zin ‘straalt’ het toevalsaspect van de probabilistische bewijsmethodes af op combinatorische bewijsmethodes, die wel deductief correct zijn maar door wiskundigen als minderwaardig gezien worden omdat ze geen inzicht geven.

10.5.1. Probabilistische bewijsmethodes

De probabilistische bewijsmethode is eigenlijk geen ‘bewijs’ in de strikte zin, omdat het

[67] [Martin1999]

geen deductieve bewijsgrond geeft. Zowel de probabilistische priembewijzen als de berekeningen op DNA-computers zijn van de volgende vorm:

- Premisse: object x_1 is *geen* getuige voor de wiskundige uitspraak P(y).
- Premisse: object x_2 is *geen* getuige voor de wiskundige uitspraak P(y).
- Premisse: …
- Premisse: object x_n is *geen* getuige voor de wiskundige uitspraak P(y).
- Premisse: de kans dat al deze n objecten x_k geen getuige zijn voor de wiskundige uitspraak P(y) is gelijk aan $(1-p)^n$.
- Conclusie: voor grote getallen n geldt P(y) heel waarschijnlijk niet.

Object x_k is een getuige voor de wiskundige uitspraak P(y) over een object y wanneer er een bepaalde wiskundige relatie bestaat tussen x_k en y waaruit volgt dat P(y) geldt. In probabilistische bewijsmethodes is een bepaald percentage p van de wiskundige objecten een getuige voor P(y). Als we dus n objecten willekeurig gekozen hebben, is de kans dat *geen* van deze objecten een getuige is voor P(y) gelijk aan $(1-p)^n$. Uit de observatie dat $x_1 \ldots x_n$ geen getuigen zijn voor de uitspraak P(y) kunnen we voor een grote n dus afleiden dat het heel onwaarschijnlijk is dat P(y) geldt. Dit nemen we aan als een ‘bewijs’ dat P(y) niet geldt.

Dit algemene schema kunnen we toepassen op de Miller-Rabin priemtest: de wiskundige objecten x_k zijn dan willekeurige getallen k kleiner dan een natuurlijk getal y, de wiskundige uitspraak P(y) zegt dat het natuurlijk getal y een samengesteld getal is en het percentage p is 3/4.

Het bewijs om een Hamiltoniaans pad in een gerichte graaf te vinden met behulp van een DNA-computer past eveneens in dit schema: de wiskundige objecten x_k zijn de gegenereerde paden in de graaf die in de opeenvolgende stappen in het algoritme van Adleman geëlimineerd worden en de wiskundige uitspraak P(y) zegt dat de gerichte graaf y een Hamiltoniaans pad heeft. Het percentage p is hier niet gemakkelijk te kwantificeren, maar het is wel aannemelijk dat $(1-p)^n$ heel klein is.

10.5.2. Combinatorische bewijsmethodes

We hebben eerder al bij de bespreking van het vierkleurenprobleem uitgelegd hoe de combinatorische bewijsmethode werkt. In dit hoofdstuk gaan we de methode iets anders beschrijven, om ze gemakkelijker te kunnen vergelijken met de probabilistische bewijsmethode. De combinatorische bewijsmethode kan als volgt in algemene termen beschreven worden:

- Premisse: object x_1 is *geen* getuige voor de wiskundige uitspraak $P(y_1)$.
- Premisse: object x_2 is *geen* getuige voor de wiskundige uitspraak $P(y_2)$.
- Premisse: …

- Premisse: object x_n is *geen* getuige voor de wiskundige uitspraak $P(y_n)$.
- Premisse: objecten $x_1 \ldots x_n$ vormen exhaustief alle elementen van de eindige verzameling X.
- Premisse: als alle elementen x_k van de verzameling X geen getuige zijn voor de overeenkomstige wiskundige uitspraken $P(y_k)$, dan geldt de wiskundige uitspraak Q.
- Conclusie: Q geldt.

We definiëren ook hier een object x_k als een getuige voor de wiskundige uitspraak $P(y_k)$ over een object y_k wanneer er een bepaalde wiskundige relatie bestaat tussen x_k en y_k waaruit volgt dat $P(y_k)$ geldt. Als alle objecten x_k uit een eindige verzameling X *geen* getuige zijn voor de overeenkomstige wiskundige uitspraken $P(y_k)$ en als we bewezen hebben dat hieruit de uitspraak Q geldt, hebben we Q bewezen.

Dit algemene schema kunnen we bijvoorbeeld toepassen op de bewijzen van de vierkleurenstelling: de wiskundige objecten x_k zijn dan de reduceerbare configuraties en de wiskundige uitspraak $P(x_k)$ zegt dat configuratie x_k kan voorkomen in een minimale kaart met vijf kleuren. Als we aangetoond hebben dat x_k niet kan voorkomen in een minimale kaart met vijf kleuren, is x_k per definitie reduceerbaar en dus geen getuige voor $P(x_k)$. De verzameling X is een onvermijdelijke verzameling: elke kaart moet minstens één configuratie uit deze verzameling bevatten. Verder volgt uit de verificatie van zulke onvermijdelijke verzameling van reduceerbare configuraties dat alle kaarten een configuratie bevatten die niet kan voorkomen in een minimale kaart met vijf kleuren. Hieruit volgt dat de vierkleurenstelling Q geldt: elke kaart heeft maximum vier kleuren nodig om ze correct in te kleuren.

Het combinatorische gedeelte van Hales' bewijs van het Keplervermoeden past eveneens in dit schema: de wiskundige objecten x_k zijn de *tamme grafen* en de wiskundige uitspraak $P(x_k)$ zegt dat x_k een tegenvoorbeeld is van het Keplervermoeden. Van elke tamme graaf wordt dan bewezen dat het geen tegenvoorbeeld van het Keplervermoeden is, dus de graaf x_k is geen getuige van $P(x_k)$. De verzameling X is de volledige verzameling van tegenvoorbeelden. Als alle elementen van deze verzameling geen getuige zijn, geldt het Keplervermoeden.[68]

Een ander voorbeeld van dit soort bewijs vinden we in het bewijs van Coolsaet en Degraer dat de Perkelgraaf uniek is. De wiskundige objecten x_k zijn dan de pseudo-Perkelgrafen, namelijk de grafen die aan dezelfde voorwaarden voldoen als in de definitie van de Perkelgraaf. De wiskundige uitspraak $P(x_k)$ zegt dat graaf x_k isomorf is met de Perkelgraaf. De verzameling X is de verzameling van alle pseudo-Perkelgrafen. Uit het feit dat alle pseudo-Perkelgrafen isomorf zijn met de Perkelgraaf volgt dat de Perkelgraaf

[68]Drie grafen waren hier uitzonderingen: dit waren geen tamme grafen maar konden wel op een tegenvoorbeeld van het Keplervermoeden wijzen. Deze drie gevallen werden door Ferguson en Hales apart uitgewerkt, waarvan we hier abstractie maken.

uniek is. Dit laatste is dan ook stelling Q.

10.5.3. Verschillen en gelijkenissen

Nu we de probabilistische en combinatorische bewijsmethodes in deze vorm hebben voorgesteld, kunnen we ze vergelijken. Een eerste gelijkenis is dat het aantal premissen meestal groot is. Bij de probabilistische bewijsmethode hebben we namelijk veel premissen nodig om de foutkans zo klein mogelijk te maken en bij de combinatorische methode zijn er in veel gevallen veel premissen. Bij het bewijs door Robertson en zijn collega's van de vierkleurenstelling gaat het bijvoorbeeld om 633 configuraties.

Een belangrijk verschil is dat de objecten in de premissen bij de combinatorische bewijsmethode exhaustief alle elementen van een eindige verzameling vormen. Wanneer we kunnen bewijzen dat hieruit de wiskundige uitspraak Q volgt, dan hebben we een volledig deductief bewijs. In de probabilistische bewijsmethode vormen de objecten in de premissen niet exhaustief een bepaalde eindige verzameling die de te bewijzen stelling garandeert. Het verband tussen de premissen en de conclusie is probabilistisch.

10.5.4. Argumenten tegen de methodes

De combinatorische methode is een perfect deductief wiskundig bewijs. Niemand twijfelt dus aan de correctheid van de methode, maar er kunnen wel twijfels rijzen over de correctheid van de implementatie van de methode op een computer. De eerste reacties op het vierkleurenprobleem waren ook van die aard: men discussieerde over de correctheid en het a priori karakter van het bewijs. Gaandeweg, naarmate men meer vertrouwd geraakte met computers en technieken om de correctheid van een bewijs te verifiëren, bleek dit argument niet meer zo sterk te zijn.

Een argument dat nog altijd blijft, is dat zulke bewijzen geen inzicht geven in waarom de stelling waar is. Volgens de wiskundige William Thurston heeft de controverse rond het bewijs door Appel en Haken dan ook meer te maken met ons verlangen om een bewijs te begrijpen.[69] Wiskundigen willen geen grote verzameling met antwoorden, ze willen inzicht.[70]

Ook tegen probabilistische bewijzen was de belangrijkste kritiek eerst dat ze onbetrouwbaar zijn en geen zekerheid bieden. Dat ze geen 100% zekerheid bieden is correct, maar zoals Fallis aangetoond heeft is dit geen reden om probabilistische bewijzen af te wijzen.

[69][Thurston1994] p. 162: ‘The rapid advance of computers has helped dramatize this point, because computers and people are very different. For instance, when Appel and Haken completed a proof of the 4-color map theorem using a massive automatic computation, it evoked much controversy. I interpret the controversy as having little to do with doubt people had as to the veracity of the theorem or the correctness of the proof. Rather, it reflected a continuing desire for *human understanding* of a proof, in addition to knowledge that the theorem is true.’

[70][Thurston1994] p. 162: ‘What they really want is usually not some collection of “answers” –what they want is *understanding*.’

Door de eenvoud van de methode is de betrouwbaarheid ervan namelijk heel groot en ze kan gemakkelijk willekeurig betrouwbaar gemaakt worden door meer getuigen te zoeken.

Een argument dat nog altijd blijft, en dat Fallis ook vermeldt, is dat probabilistische bewijzen niet aantonen waarom de stelling geldt.[71] Fallis citeert Wittgenstein, die in zijn *Remarks on the Foundations of Mathematics* zegt dat 'proof, one might say, does not merely show *that* it is like this, but *how* it is like this'.[72] Uiteraard is dit geen vereiste om van een *geldig* bewijs te kunnen spreken, maar wel een wenselijke eigenschap. Fallis vermeldt dan ook dat een probleem met probabilistische bewijsmethodes is dat ze geen inzicht geven, maar dit is geen epistemisch probleem en daar focust hij zich op. Hij gaat er dan ook niet verder op in. In zijn artikels toont hij aan dat wiskundigen geen epistemische redenen hebben om probabilistische bewijsmethodes te weerleggen.

In beide gevallen geven de bewijsmethodes dus niet echt inzicht in waarom de stelling geldt en dat vinden wiskundigen een groot nadeel van de methodes. Wat betreft de correctheid van de methodes zal men in beide gevallen toch de methodes moeten aanvaarden, aangezien de combinatorische methodes met verificatieprogramma's kunnen gecontroleerd worden en de probabilistische bewijsmethodes door hun eenvoud een heel kleine foutkans hebben.

Ik wil hier een extra reden aanhalen die volgens mij meespeelt bij de manier waarop wiskundigen tegen beide methodes staan. De kern van het probleem met de probabilistische methode ligt bij het *toevalsaspect*. Wat zulke methode antwoordt op de vraag 'Is X een priemgetal?' is samen te vatten als 'Heel waarschijnlijk, want $x_1 \ldots x_n$ gelden en de kans dat deze samen gelden en X geen priemgetal is, is onwaarschijnlijk klein.'

Als we dit vergelijken met de situatie bij een combinatorische bewijsmethode, dan zien we daar iets gelijkaardigs. Het antwoordt van zo'n methode op de vraag 'Zijn alle vlakke kaarten inkleurbaar met maximum vier kleuren?' is samen te vatten als 'Ja, want $x_1 \ldots x_n$ gelden allemaal en als deze allemaal gelden, is de vierkleurenstelling waar.' Omdat je hier met een grote n te maken hebt en wiskundigen bij zo'n grote berekening in hun achterhoofd altijd wel bevreesd zijn dat er een fout in de implementatie van het bewijs of het bewijs zelf kan zitten, krijgt het bewijs schijnbaar een probabilistisch karakter. Als je bekijkt hoe gelijkaardig beide methodes zijn, denk ik dus dat het probabilistische karakter van probabilistische bewijzen in zekere zin 'afstraalt' op de combinatorische bewijsmethode: wiskundigen beschouwen het antwoord als eerder toevallig. De combinatorische bewijsmethode is wat Gian-Carlo Rota 'bewijs door verificatie' noemt. Een verificatie van alle gevallen is wel een bewijs, maar geeft daarom nog niet de reden waarom de stelling geldt. We willen een 'mooi' bewijs dat ons de eigenlijke reden geeft, niet slechts de logische reden.[73] Over de bewijzen van de vierkleurenstelling zegt hij dan ook:[74]

[71] [Fallis1996], [Fallis1997], [Fallis2002]
[72] [Fallis1997] p. 170
[74] [Rota1997b] p. 220

> "No computer verification of the four colour conjecture will be accepted as definitive. Mathematicians are on the lookout for an argument that will make all computer programs obsolete, an argument that will uncover the still hidden reason for the truth of the conjecture. […] The example of the four color conjecture leads to an inescapable conclusion. Not all proofs give satisfying reasons why a conjecture should be true. Verification is proof, but verification may not give the reason."

Rota hecht dan ook weinig waarde aan combinatorische bewijzen. Ze openen geen nieuwe mogelijkheden, terwijl een 'inzichtelijk' bewijs in zijn bewijsvoering verbanden legt die inzicht kunnen geven in andere fenomenen. Zo is het bewijs door Andrew Wiles van de Laatste Stelling van Fermat[75] niet zozeer belangrijk omdat het de waarheid van de stelling bevestigt. Bijna iedere wiskundige was er namelijk al wel van overtuigd dat de stelling gold. De waarde van het bewijs ligt in het feit dat het zo veel concepten uit verschillende domeinen met elkaar verbindt, van getaltheorie tot modulaire vormen en elliptische krommen, en ons inzicht in deze onderwerpen dus verdiept.[76] De computerbewijzen van de vierkleurenstelling geven dit inzicht echter niet omdat ze zo'n verbindingen slechts in beperkte mate bevatten en voor een groot deel bestaan uit een grote verzameling van 'getuigen' die de stelling verifiëren.[77]

Richard De Millo, Richard Lipton en Alan Perlis geven in hun polemische artikel *Social processes and proofs of theorems and programs* een vergelijking tussen het sociale proces van corrigeren van informele bewijzen en de praktijk om bewijzen volledig formeel neer te schrijven. Interessant is dat zij het sociale aspect van beide types bewijzen illustreren. Informele bewijzen inspireren en passioneren de wiskundige, die zijn ontdekkingen zo vlug mogelijk wil delen met de wereld.[78] Formele bewijzen van stellingen of computerprogramma's inspireren echter geen enkele wiskundige: niemand wordt er enthousiast door of gaat het lezen om er ideeën in te vinden.[79]

---

[73][Rota1997] p. 182: 'We say that a proof is beautiful when such a proof finally gives away the secret of the theorem, when it leads us to perceive the actual, not the logical inevitability of the statement that is being proved.'

[75]Eigenlijk bewees Wiles het vermoeden van Taniyama-Shimura, waarvan eerder bewezen was dat het de Laatste Stelling van Fermat impliceert. Voor een gepopulariseerde uitleg van de ontwikkelingen die tot het bewijs leidden, verwijs ik naar het boek [Singh1998].

[76][Rota1997b] p. 222: 'The value of Wiles' proof lies not in what it proves, but in what it opens up, in what it makes possible. Every mathematician silently knows that such an opening up of possibilities is the real value of the proof of Fermat's conjecture.'

[77][Rota1997b] p. 222: 'Every mathematician knows that the computer verification of the four color conjecture is of considerably lesser value than Wiles' proof, because it fails to open up any significant mathematical possibilities.'

[78][DeMillo1979]: 'No mathematician grasps a proof, sits back, and sighs happily at the knowledge that he can now be certain of the truth of his theorem. He runs out into the hall and looks for someone to listen to it. He bursts into a colleague's office and commandeers the blackboard. He throws aside his scheduled topic and regales a seminar with his new idea. He drags his graduate students away from their dissertations to listen. He gets onto the phone and tells his colleagues in Texas and Toronto.'

Yehuda Rav hecht ook veel belang aan de bewijzen, meer dan aan stellingen. Hij ziet bewijzen als de dragers van wiskundige kennis en stellingen als 'samenvattingen' of zelfs namen hiervan.[80] Rav beschrijft dit metaforisch als 'Theorems are the headlines, proofs are the inside story.'[81] Als we in deze metafoor blijven, dan is een combinatorisch bewijs een nieuwsbericht dat bestaat uit een opsomming van feiten en gebeurtenissen, maar ons helemaal niet wijzer maakt over de beweegredenen van de betrokkenen en geen enkele verklaring geeft voor de gebeurtenis. Wie zulk nieuwsbericht leest, zoekt terecht ergens anders naar meer diepgang.

10.6. Conclusie

De laatste 10 jaar is er steeds meer aandacht voor de structuur van computerbewijzen en hoe we die inzichtelijker kunnen maken. De voorstellen van Beeson en de onderzoeksgroep van Bundy om computers meer als mensen te laten denken door het opstellen van bewijsplannen zien er wel interessant uit. De vraag is echter of die technieken krachtig genoeg zijn om ons bewijzen te leveren van vermoedens die menselijke wiskundigen niet kunnen bewijzen. De bewijsprogramma's van de onderzoeksgroep van Wos mogen dan wel oninzichtelijke bewijzen leveren, ze hebben wel tot heel wat nieuwe resultaten geleid, iets dat de programma's van Bundy's onderzoeksgroep nog niet kunnen.

Het is interessant om in het licht van dit hoofdstuk eens te kijken naar wat wiskundigen zelf zeggen over computerbewijzen. In december vorig jaar verscheen in de *Newsletter of the European Mathematical Society* een interview met drie winnaars van de prestigieuze Fieldsmedaille. Twee van hen, Andrei Okounkov en Terence Tao, spraken tijdens het interview over computerbewijzen. Zo vindt Okounkov dat wiskundigen zo veel mogelijk 'routinewerk' in bewijzen moeten afschuiven op computers:[82]

> Computers are no more a threat to mathematicians than food processors

[79][DeMillo1979]: 'Verifications are long and involved but shallow; that's what's wrong with them. The verification of even a puny program can run into dozens of pages, and there's not a light moment or a spark of wit on any of those pages. Nobody is going to run into a friend's office with a program verification. Nobody is going to sketch a verification out on a paper napkin. Nobody is going to buttonhole a colleague into listening to a verification. Nobody is ever going to read it. One can feel one's eyes glaze over at the very thought.'

[80][Rav1999] p. 20: 'Proofs rather than the statement-form of theorems are the bearers of mathematical knowledge. Theorems are in a sense just tags, labels for proofs, summaries of information, headlines of news, editorial devices. The whole arsenal of mathematical methodologies, concepts, strategies and techniques for solving problems, the establishment of interconnections between theories, the systematization of results –the entire mathematical know-how is embedded in proofs. […] Theorems indicate the subject matter, resume major points.'

[81][Rav1999] p. 22

[82][Muñoz2006] p. 33

> are a threat to cooks. As mathematics gets more and more complex while the pace of our lives accelerates, we must delegate as much as we can to machines. And I mean both numeric and symbolic work. Some people can manage without dishwashers, but I think proofs come out a lot cleaner when routine work is automated.

Ook Tao heeft over computerbewijzen een eigen mening. Net als Okounkov vindt hij dat computers een rol in wiskundige bewijzen kunnen hebben, maar hij heeft wel een expliciete voorwaarde:[83]

> I think such proofs can be satisfactory if the computational component of the proof is merely confirming some expected or unsurprising phenomenon (e.g., the absence of sporadic solutions to some equation, or the existence of some parameters that obey a set of mild conditions), as opposed to demonstrating some truly unusual and inexplicable event that cries out for a more human-comprehensible analysis. In particular, if the computer-assisted claims in the proof already have a firm heuristic grounding then I think there is no problem with using computers to establish the claims rigorously. Of course, it is still worthwhile to look for human-readable proofs as well.

Tao vindt inzicht dus heel belangrijk in een bewijs. Als we op een andere manier al een soort verklaring hebben voor een bepaald resultaat, ook al is die verklaring slechts heuristisch en niet deductief, dan ziet Tao dus geen graten in een computerbewijs, aangezien het bewijs dan vooral dient ter *verificatie* van de stelling, om alle saaie details uit te werken. Hij benadrukt echter dat het zelfs in die gevallen nog nuttig is om bewijzen te vinden die door mensen kunnen gelezen worden.

[83] [Muñoz2006] p. 35

11. Besluit

11.1. Algemeen besluit

11.1.1. De veelheid aan computerbewijzen

De bespreking van heel wat computerbewijzen in deze eindverhandeling en hun eigenschappen laat zien dat we niet alle computerbewijzen over één kam kunnen scheren. Dat een bepaalde klasse van computerbewijzen bepaalde eigenschappen hebben, betekent nog niet dat een andere klasse deze eigenschappen ook hebben. Daarom heb ik ervoor gekozen om zo veel mogelijk computerbewijzen te bestuderen, met als resultaat een classificatie van de gebruikte bewijstechnieken. In de literatuur zijn er heel wat artikels te vinden die dit, soms bewust, niet doen en op basis daarvan sterke uitspraken doen die niet algemeen gelden.[1]

Een voorbeeld van een filosofische analyse van computerbewijzen die op dit vlak tekortschiet is te vinden in 'The four-color theorem and its consequences for the philosophy of mathematics' van Izabela Bondecka-Krzykowska.[2] Ten eerste is het anno 2004 al moeilijk om nog iets origineels te zeggen over de invloed van de computerbewijzen van het vierkleurenprobleem op de filosofie van de wiskunde. Daar faalt Bondecka al: ze blijft hangen in een ongestructureerde en vage bespreking van de problemen die het bewijs van Appel en Haken zou stellen aan het onderscheid a priori - a posteriori waarheden in de wiskunde. Ze beperkt zich daarbij tot een bespreking van de standpunten van Tymoczko, Swart en Teller. Ten tweede is het anno 2004 niet meer verantwoord om een filosofische bespreking van computerbewijzen enkel te baseren op het bewijs van het vierkleurenprobleem en uit dat ene voorbeeld te generaliseren. Bondecka is blijkbaar niet op de hoogte van andere computerbewijzen of ziet geen reden om ernaar te verwijzen.[3] Haar artikel heeft dan ook niets origineels te zeggen over het probleem.[4] Doordat ze zich enkel baseert op het bewijs van het vierkleurenprobleem, maakt ze ook incorrecte generalisaties.[5]

[1]Zo schrijft Brian Davies in [Davies2005] p. 1351 over de verschillende voorbeelden van computerbewijzen: 'It would serve no useful purpose to enumerate all such cases, so we turn to the most recent example.', waarna hij de problemen in Hales' bewijs bespreekt. In dit artikel verwijst hij helemaal niet naar bijvoorbeeld EQP's bewijs van het Robbinsvermoeden, dat heel wat andere problemen heeft.

[2][Bondecka2004]

[3][Bondecka2004] p. 6: 'The four-color theorem reveals philosophical problems connected with computer proofs. The situation is so special because in the case of other theorems proved by appealing to computers there was also a traditional proof.'

[4]Zie bijvoorbeeld de conclusie van [Bondecka2004], p. 13: 'Summing up, we can say that incorporating computers to mathematics, especially in proving theorems, reveals many philosophical problems, such as the status of mathematical truths (distinction between *a priori* and *a posteriori*), the status of mathematics as a standard for formal science and finally the problem of acceptable methods of "doing" mathematics. Most of these problems remain unsolved because their solutions very often depend on interpretation of such terms as *a priori* or *a posteriori*.'

Zo suggereert ze dat alle computerbewijzen te lang zijn en dus niet inspecteerbaar zijn. een blik op EQP's bewijs van het Robbinsprobleem leert ons dat dat niet zo is.

11.1.2. Computerbewijzen in de wiskundige praktijk

De vragen die ik mij als filosoof vooral stel zijn: waarom zijn computerbewijzen nog altijd geen onderdeel van de wiskundige praktijk? Wat kunnen ontwerpers van bewijsprogramma's doen om de aanvaarding van computerbewijzen in de wiskundige praktijk te bespoedigen? Op de eerste vraag is een groot deel van het antwoord volgens mij dat veel computerbewijzen tot nu toe niet echt inzicht geven in waarom een stelling geldt, wat samenhangt met de hiervoor besproken armoede aan wiskundige concepten in computerbewijzen. Hiermee spreek ik Jody Azzouni tegen, die beweert dat computerbewijzen vanzelf wel deel zullen uitmaken van de wiskundige praktijk, wanneer wiskundigen zien dat ze tot resultaten leiden die we met traditionele bewijzen niet kunnen vinden.

Dat hij dit zegt, is te begrijpen vanuit een vooronderstelling die hij maakt: waar het uiteindelijk om gaat bij wiskundigen is volgens hem het ontdekken van nieuwe resultaten, onafhankelijk van het feit of we begrijpen waarom ze waar zijn. Het overzicht van computerbewijzen en de reacties erop dat ik in deze eindverhandeling gegeven heb, toont volgens mij aan dat wiskundigen niet alleen kijken naar de resultaten, maar inzicht minstens zo belangrijk vinden. Dit geldt niet alleen in het algemeen, maar komt ook tot uiting in de reacties op en de aanvaarding van computerbewijzen.[6]

In bepaalde domeinen van de wiskunde zien we inderdaad wat Azzouni hier zegt. Denk bijvoorbeeld aan de resultaten van Wos, Veroff en anderen in abstracte algebra. Het feit dat het hier om theorieën gaat waar mensen weinig intuïtieve inzichten in hebben, kan dit echter verklaren. Als je er toch niet gemakkelijk inzicht in krijgt op de traditionele manier, is er niet zo veel inzichtelijk verschil tussen een computerbewijs en een menselijk bewijs. Als computerbewijzen dan tot veel meer resultaten leiden dan menselijke bewijzen, is het te begrijpen dat de wiskundigen die met deze computerbewijzen kunnen werken hiervan dankbaar gebruik maken.[7] In andere domeinen van de wiskunde zien we echter dat wiskundigen heel wat sceptischer staan tegenover het gebruik van computers in bewijzen, aangezien er daar wel een groot verschil is in inzicht.

Op de tweede vraag heb ik ook een antwoord gegeven in Hoofdstuk 9, *Wiskundige concepten in computerbewijzen*: bewijsprogramma's moeten met meer concepten overweg kunnen en vooral concepten uit verschillende domeinen kunnen met elkaar in verband

[5][Bondecka2004] p. 8: ‘There are many traditional proofs which an average mathematician does not understand, but there is no proof which could not be looked over, reviewed and verified by a rational agent. The proof of the four-color theorem, like other computer proofs, does not possess this feature.’

[6][Azzouni2005] p. 44: ‘Mathematical proof practices –as institutionalized in the profession of mathematics– is ultimately a matter of discovering new results; this is regardless of whether we understand “why” they are true or not: Mathematics remains theorem-driven. If mathematicians find they can discover more of these results by means of computers than they can by means of traditional proof, they will desert traditional proof for that reason alone.’

brengen. Aangezien de ‘inzichtelijke’ bewijzen van wiskundigen ook vaak op deze manier werken, zullen zulke computerbewijzen meer op onze traditionele bewijzen lijken en beter aanvaard worden door wiskundigen. De voorspelling van Wos dat wiskundigen het bewijzen van hun vermoedens aan computers kunnen overnemen, zal slechts uitkomen als computers leren redeneren met concepten.

11.1.3. Het belang van computerbewijzen voor de wiskunde

In de inleiding vermeldden we de criteria die wiskundigen volgens Corfield gebruiken om het belang van nieuwe ontwikkelingen in de wiskunde te beoordelen. We komen hier nu samenvattend op terug hoe computerbewijzen scoren op deze criteria:

1. **De ontwikkeling laat toe om nieuwe berekeningen uit te voeren in een bestaand probleemdomein, mogelijk met de oplossing van een oud vermoeden als gevolg.** Het feit dat verschillende stellingen voor het eerst bewezen zijn met behulp van een computer en ondertussen niet op een andere manier bewezen zijn, toont aan dat computers op dit criterium al punten halen. Een bewijs van het vierkleurenprobleem, het Keplervermoeden of het Robbinsprobleem zou nooit door een mens gevonden zijn.
2. **De nieuwe ontwikkeling verbindt twee al bestaande domeinen, waardoor resultaten en technieken tussen de twee kunnen overgedragen worden.** Op dit criterium scoren computerbewijzen heel slecht. De huidige bewijsprogramma's kunnen enkel ‘elementaire bewijzen’ leveren die berekeningen of redeneringen uitvoeren in één do-

[7]Zo bevatte de eerste reactie op een blogtekst van David Corfield over mijn presentatie op de Perspectives on Mathematical Practices 2007-conferentie de volgende opmerking: ‘Well, there are those of us who use automated theorem provers, but don't hold the computer's hand to make them prove known results or to win competitions. Rather we use them in our research to discover new results.’ (http://golem.ph.utexas.edu/category/2007/04/automated_theorem_proving.html#c008782) De uitspraak is van Michael Kinyon, die het programma OTTER gebruikte voor een lemma in een bewijs over lussen en blijkbaar meer geïnteresseerd is in resultaten dan in inzicht. Hij zegt echter in een andere commentaar zelf dat het succes van OTTER in bewijzen over lussen niets te maken heeft met het gebrek aan inzicht dat wiskundigen in het onderwerp hebben: ‘I can't speak for areas in which I don't work, but the success of automated theorem proving in mine is not due to deprivation of intuition and certainly not to connection to other fields. (Most of us get into loop theory because of its intimate links to group theory.) It's rather a function of the age of the subject and the number of people who have worked in it over the years. It's an area in which there are problems that can still be formulated in ways in which a.t.p. [automated theorem proving] can help. Group theory itself was once that way, of course, as a perusal of its history shows, but it matured and the open problems moved in a different direction.’ (http://golem.ph.utexas.edu/category/2007/04/automated_theorem_proving.html#c008793) Een andere commentator bevestigt echter dat het probleem wel ligt bij het gebrek aan intuïties die we hebben in het onderwerp: ‘I have an alternative explanation. Moufang loops and quasigroups are just too foreign to human experience for humans to prove theorems about them unaided. The most developed areas of algebra up to now are those where there is a notion of “representation theory”, a way to represent abstract objects (like groups) to concrete objects (like permutation groups and matrix groups). Now thanks to computers we will find a whole new rich world that was inaccessible to us because of our primitive mammalian brains.’ (http://golem.ph.utexas.edu/category/2007/04/automated_theorem_proving.html#c008850)

mein. Pas wanneer ontwerpers van deze programma's ook de mogelijkheid implementeren om efficiënt in verschillende domeinen tegelijk te werken en verbanden daartussen te onderzoeken, zullen computerbewijzen op dit criterium beter scoren. Het eerste computerbewijs dat een nieuw conceptueel verband introduceert zal een grote doorbraak zijn.

3. **De ontwikkeling introduceert een nieuwe manier om resultaten in bestaande domeinen te organiseren, waardoor misschien een verduidelijking of verschuiving van de domeingrenzen mogelijk is.** Automatische bewijsprogramma's bieden mogelijkheden om op dit criterium goed te scoren, maar dit wordt door weinig onderzoekers echt expliciet gemaakt. Belinfante heeft dit wel gedaan in zijn bewijzen van stellingen in ordinaaltheorie: door interactief met het bewijsprogramma alle veronderstellingen te exploreren, werd het hem duidelijker wanneer een bepaalde stelling geldt. Automatische bewijsprogramma's als OTTER bieden dus heel wat mogelijkheden om begrip in een domein te verduidelijken, mits de gebruiker er op de juiste manier mee omgaat.
4. **De ontwikkeling opent perspectieven op nieuwe conceptueel gemotiveerde domeinen.** Voorlopig scoren computerbewijzen op dit criterium nog slecht, vooral omdat ze nog weinig conceptueel opgebouwd zijn. Wanneer bewijsprogramma's meer met verschillende concepten kunnen omgaan, kunnen we misschien verwachten dat ze ook op dit criterium beter scoren.
5. **De ontwikkeling leidt redelijk direct tot succesvolle toepassingen buiten wiskunde.** Dit is minder aan bod gekomen in deze eindverhandeling, maar bewijsprogramma's worden ook in de industrie breed ingezet. AMD, één van de grootste ontwerpers van computerprocessoren, gebruikt bijvoorbeeld het programma ACL2 om de correcte werking van zijn chips te verifiëren. Ook de programma's die van bewijsplannen gebruik maken zijn buiten wiskunde toegepast. Zo is de bewijsplanner CLAM toegepast in computerprogramma's die Go en Bridge spelen. Automatische bewijsprogramma's scoren dus goed op dit criterium.

Samengevat scoren computerbewijzen voorlopig op drie criteria goed en op twee heel slecht. Als we kijken naar de inhoud van de criteria, dan zien we dat de twee slecht scorende criteria over wiskundige concepten gaan. We kunnen dus besluiten dat een verbetering op de afhandeling van concepten in bewijsprogramma's tot een enorm belangrijke nieuwe ontwikkeling in de wiskunde zou leiden: weinig wiskundige technieken scoren goed op alle vijf criteria. Uit de inhoud van deze twee criteria waar computerbewijzen slecht op scoren wordt ook duidelijk waarom wiskundigen die enkel geïnteresseerd zijn in de waarheid van wiskundige stellingen niet zo'n problemen hebben met de huidige computerbewijzen. Voor deze mensen zijn de gebruikte concepten en conceptuele verbanden van minder belang. Zij vinden het eerste criterium, waarop computerbewijzen goed scoren, namelijk het belangrijkste.

11.1.4. Wat is het doel van wiskunde?

De studie van computerbewijzen en de reacties van wiskundigen hierop maakt duidelijk dat wiskundigen zelf in twee kampen te verdelen zijn wanneer het gaat om het doel van wiskunde. Een eerste groep van wiskundigen is enkel geïnteresseerd in de *waarheid* van wiskundige uitspraken. De belangrijkste verwezenlijking van een bewijs is voor hen dat het de waarheid van een stelling rechtvaardigt. Welke ideeën, concepten en technieken het bewijs gebruikt, is voor hen van minder of geen belang. Deze wiskundigen hebben in het algemeen dan ook weinig problemen met de niet-inzichtelijkheid van een computerbewijs. Volgens de tweede groep van wiskundigen gaat wiskunde om het *begrijpen* van de wiskundige wereld. Zij hechten veel meer belang aan de gebruikte concepten en technieken in een bewijs en stellen zich vragen als ‘wat leren we hieruit?’, ‘welke nieuwe inzichten geeft dit bewijs ons?’, etc. Deze wiskundigen zullen veel meer problemen hebben met de niet-inzichtelijkheid van computerbewijzen. Ian Stewart, Daniel ‘computerfoefjes’ Cohen en Gian-Carlo Rota zijn voorbeelden uit deze groep wiskundigen.

Dezelfde tweedeling vinden we in het onderzoeksdomein van automatische bewijsprogramma's. Enerzijds hebben we mensen als Larry Wos, William McCune en Robert Veroff. Zo zei McCune in een interview ‘I just work on the problems and try to solve them’, waarmee hij aangaf dat hij vooral problemen wil oplossen, wil weten wat de waarheidswaarden van stellingen zijn. De school van Wos is vooral geïnteresseerd in empirisch succesvolle computerprogramma's en tot nu toe zijn hun programma's dat ook: ze behalen resultaten en zijn niet geïnteresseerd in een dieper inzicht daarin. De tweede groep van onderzoekers naar automatische bewijsprogramma's wil niet alleen succesvolle, maar ook begrijpelijke computerprogramma's hebben. Tot deze groep behoren mensen als Alan Bundy en Michael Beeson. Bundy wil dat de automatisch gevonden bewijzen niet alleen een resultaat rechtvaardigen, maar ook inzicht geven in waarom het resultaat geldt. Hij ziet het implementeren van zo'n bewijsprogramma's dan ook als een ‘science of reasoning’, waarin inzicht even veel een rol speelt als empirisch succes.

Deze tweedeling is niet eigen aan de wiskunde: ze komt overeen met het debat tussen *realisten* en *anti-realisten* in de wetenschap(sfilosofie). Dit debat heeft als centraal punt de vraag waar het om gaat in de wetenschap: is het doel van wetenschap het empirisch verklaren van onze wereld of het inzicht geven in de wereld? Anti-realisten hebben er geen probleem mee om abstracte entiteiten uit te vinden louter omdat we er empirisch succesvolle verklaringen voor verschijnselen in de wereld mee kunnen bekomen. Of deze entiteiten met de realiteit overeenkomen, is voor hen niet van belang. Realisten daarentegen geloven dat de entiteiten in onze aanvaarde wetenschappelijke theorieën echt bestaan in de werkelijkheid. Zij eisen dan ook dat onze wetenschappelijke wetten een *inzicht* geven in de structuur van de werkelijkheid.

11.1.5. De fragmentatie van onderzoek naar computerbewijzen

Uit het overzicht van onderzoek naar computerbewijzen wordt duidelijk dat het een onderwerp is dat door verschillende onderzoeksgroepen wordt bestudeerd, die vaak weinig contact met elkaar hebben. Soms is dit door principiële meningsverschillen over het doel van hun onderzoek of de vooronderstelling over het doel van wiskunde, zoals we in de vorige sectie bespraken. Zo zijn de onderzoekers naar automatische bewijsprogramma's duidelijk onderverdeeld in twee groepen: zij citeren elkaars artikels weinig en mengen zich niet met de problemen van de andere groep. Alan Bundy is zich heel bewust van deze tweedeling.

Een andere tweedeling zien we volgens de twee aspecten van een bewijs: berekeningen en redeneringen. De onderzoekers die redeneringen willen automatiseren, hebben we in de vorige sectie al besproken: dat zijn de onderzoekers naar automatische bewijsprogramma's. Daarnaast heb je een grote groep van onderzoekers die berekeningen zoveel mogelijk willen automatiseren. De resultaten hiervan komen terecht in commerciële programma's als MATHEMATICA, MAPLE en MATLAB. De twee aspecten van een bewijs kunnen dus goed geautomatiseerd worden, maar de twee types computerprogramma's verbinden is niet triviaal. Pas de laatste tien jaar is de nood daaraan echt duidelijk geworden bij onderzoekers van beide groepen, maar echte toenadering is nog niet zo zichtbaar. Dat is ook normaal, aangezien ze met verschillende problemen bezig zijn. Op termijn zal het echter voordelig zijn als deze twee onderzoeksgroepen samenwerken, aangezien zowel automatische bewijsprogramma's als computeralgebrapakketten profiteren van de hulp van de ander.

11.2. Verder onderzoek

11.2.1. Een classificatie van wiskundige bewijsmethodes

In het artikel ‘The phenomenology of mathematical proof’ zegt Gian-Carlo Rota dat filosofen van de wiskunde een rigoureuze behandeling moeten kunnen geven van verschillende ‘soorten’ bewijzen.[8] In hetzelfde jaar schrijft Saunders Mac Lane, één van de grondleggers van de categorietheorie in de wiskunde, in zijn artikel ‘Despite physicists, proof is essential in mathematics’ iets gelijkaardigs: we hebben een bewijstheorie nodig die bestudeert hoe we verschillende types van inzichten combineren in bewijzen.[9] Ik heb in mijn eindverhandeling al een eerste aanzet gegeven tot een classificatie van bewijstechnieken in computerbewijzen, maar het is duidelijk dat er veel meer nodig is. Ook van

[8][Rota1997b] p. 225: ‘Mathematical proofs come in different kinds, that need to be classified.’

de bewijsmethodes die in menselijke bewijzen voorkomen, zou er een classificatie moeten komen: bewijzen door inductie, het duivenhokprincipe, *reductio ad absurdum*, diagonalisatieargumenten, … Erik Weber en Liza Verhoeven geven in hun artikel 'Explanatory proofs in mathematics' bij de 'questions for future research' aan dat het interessant zou zijn om te bekijken hoe en wanneer verschillende types van bewijzen inzicht geven, zoals existentiebewijzen, uniciteitsbewijzen en identiteitsbewijzen.[10] Dit is een andere aanpak van de vraag welke bewijzen inzicht geven, namelijk in functie van de vorm van de stelling in plaats van in functie van de gebruikte bewijstechniek. Ik vermoed dat een combinatie van deze twee aanpakken nog de beste filosofische aanpak van *mathematical explanation* zou zijn en zo de meest concrete uitspraken zou kunnen doen.

Wanneer zulke classificatie van bewijstechnieken grondig zou gebeuren, zouden filosofen van de wiskunde veel eenduidiger met elkaar kunnen discussiëren over bijvoorbeeld het inzicht dat een bepaalde bewijsmethode kan geven in tegenstelling tot een andere. Zo'n classificatie zou bovendien toelaten om deze beswijsmethodes ook te mechaniseren en tot de droom van Beeson en Bundy te komen: een bewijsprogramma dat wiskundige bewijzen levert zoals een mens dat zou doen.[11]

11.2.2. De inzichtelijkheid van elementaire bewijzen

Waarom zijn bepaalde elementaire bewijzen inzichtelijk en andere niet? We zagen dat Pringsheim zijn elementaire theorie van analytische functies inzichtelijker vindt dan de niet-elementaire theorie van Cauchy. In andere gevallen, zoals het elementaire bewijs van Selberg en Erdös van de priemgetalstelling, zullen niet veel wiskundigen zeggen dat het elementaire bewijs meer inzicht geeft dan het niet-elementaire.[12] Ook in het geval van Szemerédi's stelling is het elementaire bewijs niet zo inzichtelijk.[13] Van een complex be-

[9][MacLane1997] p. 152: 'Real proof is not simply a formalized document, but a sequence of ideas and insights. The subject of proof theory should be the understanding and the organization of the various types of insights and their astute combinations which do occur in the construction of mathematical proof. I know of little serious work in this philosophical direction beyond the rather naive attempt in my own Ph.d. thesis (1934), republished in 1979.'

[10][Weber2002] p. 307

[11]Bundy stelt zelf in [Bundy1991b] voor om deze bewijsmethodes te identificeren met bewijsplannen in automatische redeneerprogramma's: 'Mathematicians recognise families of proofs which contain common structure. These families are sometimes named, e.g. diagonalization arguments, but, more often, are not. We propose representing such common structures with proof plans.'

[12] Bonsall bespreekt dit in [Bonsall1982] p. 10 vanuit het standpunt van de 'real live mathematician': 'Our real live mathematician has only a limited knowledge of mathematics, so a proof should use economy of force. It should not invoke deep results if a little elementary calculus will do the trick. But on the other hand there is no merit in an elementary argument if it becomes long and boring. Our real live mathematician would prefer to master some difficult tool than to endure prolonged tedium. The Erdös-Selberg elementary proof of the prime number theorem was a most remarkable tour de force, but no real live mathematician would use it in preference to the function theoretic proofs.'

[13]Zie de studie van de verschillende bewijzen van deze stelling in [Tao2007]: 'While Szemerédi's accomplishment is undoubtedly a highlight of this particular story, it was by no means the last word on the matter. Szemerédi's proof of his theorem, while elementary, was remarkably intricate and not easily comprehended.'

wijs als dat van Wiles en Taylor van het Taniyama-Shimuravermoeden kan men zich ook afvragen of een elementaire versie hiervan wel meer inzicht zal geven.[14] Waarom sommige elementaire bewijzen wel inzicht geven en andere niet, is niet zo duidelijk, maar ik vermoed dat dit te maken heeft met de 'natuurlijke context' van de centrale concepten in de stelling, iets wat ik in Hoofdstuk 9, *Wiskundige concepten in computerbewijzen* ook heb besproken. De algemene vraag waarom sommige elementaire bewijzen wel inzicht geven en andere niet, heb ik hier echter niet uitgewerkt. Dit zou een heel goed onderwerp zijn voor verder onderzoek. De studie van computerbewijzen in deze eindverhandeling is daarvoor ook relevant: door de armoede aan concepten in de huidige computerbewijzen zijn de bewijzen die programma's als OTTER leveren elementaire bewijzen. Zolang computerbewijzen niet met meer concepten kunnen redeneren, zullen ze enkel elementaire bewijzen leveren en niet bijvoorbeeld een omweg langs complexe analyse maken om een stelling in de getaltheorie te bewijzen.

11.2.3. Meetkundig redeneren

Over meetkundig redeneren heb ik in deze eindverhandeling weinig gezegd. De laatste jaren hebben filosofen en wiskundigen steeds meer aandacht voor vragen als 'kunnen afbeeldingen bewijzen?'[15], 'wanneer misleiden afbeeldingen in bewijzen ons?'[16], 'kunnen afbeeldingen in bewijzen ons meer inzicht geven?'[17] en 'kan meetkundig redeneren met afbeeldingen gemechaniseerd worden door computers?' Dit domein heb ik bij de studie voor mijn eindverhandeling opengelaten, maar dit kan zeker extra inzichten geven in de praktijk van wiskundige bewijzen.

11.2.4. Formalisering versus intuïtie

In deze eindverhandeling zijn we verschillende keren teruggekomen op de kloof tussen formalisering en intuïtie. Terwijl Hilbert in zijn formalistische programma er nog van overtuigd was dat we wiskundige intuïtie niet konden vertrouwen en onze wiskundige redeneringen dan ook volledig moesten formaliseren, werken de meeste wiskundigen van-

[14]Van Bendegem bespreekt dit in [VanBendegem2003] p. 13-14: 'Takeuti in Takeuti (1978) has shown that, if all definitions used are predicative, then a translation into elementary number theory is always possible. At first sight, it seems that all definitions used in Wiles' and Taylor's proofs are predicative. But, at the same time, it is quite clear that no one seems to be interested to actually write down that proof, as it would probably have a length beyond all comprehension.'

[15][Dove2002] p. 310: 'Diagrams can play a role analogous to (perhaps even equivalent to) the purely symbolic elements of modern mathematics.'

[16][Casselman2000]

[17][VanBendegem2003] p. 23: 'So-called "proofs by looking" have the property that the (formally correct) proof can be read off the drawing and thus explains what is happening.' Ook Michael Resnik geeft in bepaalde situaties de voorkeur aan 'visuele' bewijzen omdat ze meer inzicht geven. Zo zegt hij over het systeem van de Grieken om getallen voor te stellen door een aantal punten in het vlak: 'Using the dot system I have just described one can quickly come to appreciate that addition and multiplication are associative and commutative, that multiplication distributes over addition and that 1 is a mulltiplicative identity. Indeed, this dot system provides a much better understanding of why those principles hold than anything we are usually given.' ([Resnik1992])

daag zo niet. In sommige domeinen van de wiskunde zijn formele bewijzen zelfs helemaal ‘not done’, omdat ze helemaal geen inzicht geven. Knopentheoreticus Vaughan Jones beschrijft zo in zijn artikel ‘A credo of sorts’ een bewijs uit knopentheorie. Het bewijs bestaat uit diagrammen van knopen en operaties op de knopen door het herschikken van delen van de knoop. Het laatste deel van het bewijs gaat als volgt:[18]

> Since we are proceeding one short stretch at a time around the knot, simply isolate that crossing and, if it happens to prevent our throwing over our shoulders, throw it the other way. When we have arrived back at the beginning point, we see a closed braid. This ends the proof.

Let op termen als ‘over onze schouders gooien’ en ‘we zien’. Deze noties zijn volgens Jones niet zo gemakkelijk te formaliseren.[19] Toch is het bewijs volgens Jones gemakkelijk uit te leggen aan een ‘clever high school student’:

> The answer is that Theorem A concerns a very concrete situation, and we are able to bring to bear our full intuition about three-dimensional space on the problem. If we were two-dimensional creatures then proving this theorem would be another story entirely and would require much more *formal* argument.

Wat Jones hier beschrijft zou kunnen te verklaren zijn door Azzouni's inferentiepakketten: wiskundigen (én niet-wiskundigen!) hebben een inferentiepakket om te redeneren over driedimensionale bewegingen en kunnen door deze intuïties die ze bezitten eenvoudig informele bewijzen in knopentheorie volgen. Wanneer de stelling volledig formaal bewezen wordt, kunnen we niet teruggrijpen op ons intuïtieve begrip van onze driedimensionale wereld om het bewijs te doorgronden en moeten we terugvallen op ons intuïtie-arme begrip van symbolenmanipulatie.

Gonthier wees al bij zijn formalisering van het bewijs van het vierkleurenprobleem op de kloof tussen formalisering en intuïtie en ook Szolem Mandelbrojt waarschuwde al voor te veel formalisering. Bij het Robbinsprobleem zagen we dat het bewijs door EQP niet inzichtelijk was, omdat wiskundigen bij het proberen begrijpen ervan slechts beperkt op hun intuïties kunnen terugvallen. Het is opvallend dat zowel Kauffman als Fitelson voorstellen om de formules op een tweedimensionele manier weer te geven: de eerste doet dit met kaders rond formules en de tweede met lijnen boven formules. Mensen lijken zo een gemakkelijker overzicht op lange formules te hebben. Ook dit kan volgens mij verklaard worden door Azzouni's inferentiepakketten. Een verdere studie om dit ‘hard’ te maken, zou heel interessant zijn. Zulk onderzoek hoeft zich helemaal niet te beperken tot computerbewijzen, maar moet zeker ook computerprogramma's voor *formele verificatie* van bewijzen onderzoeken, aangezien daar het formaliseren van een bewijs tot in het extreme

[18] [Jones1998] p. 211

[19] [Jones1998] p. 212: ‘One would have to be precise about the kinds of continuous deformations that are allowed, and constructing the functions required for the “throwing over the shoulder” trick would be a nightmare.’

wordt toegepast.

Bijlage A. Lijst van computerprogramma's

Programma	Soort	Website
ACL2	Automatisch bewijsprogramma	http://www.cs.utexas.edu/users/moore/acl2/
Clam	Bewijsplanner	
Coq	Formeel verificatieprogramma	http://coq.inria.fr/
EQP	Automatisch bewijsprogramma	http://www.cs.unm.edu/~mccune/eqp/
Formac	Programmeertaal, uitbreiding van Fortran	
GAP	Computeralgebrapakket voor discrete algebra	http://www.gap-system.org/
GCC	Compiler	http://gcc.gnu.org/
GNU/Linux	Besturingssysteem	http://www.gnu.org/
HOL-Light	Formeel verificatieprogramma	http://www.cl.cam.ac.uk/~jrh13/hol-light/
HR	Automatische theorievorming	http://www.doc.ic.ac.uk/~sgc/hr/
ILF	Interactief bewijssysteem	
Isabelle/HOL	Formeel verificatieprogramma	http://isabelle.in.tum.de/
Java	Programmeertaal	http://java.sun.com/
Macek	Computeralgebrapakket voor matroïdes	http://www.cs.vsb.cz/hlineny/MACEK/
Macsyma	Computeralgebrapakket	http://www.symbolics.com/Macsyma-1.htm
Magma	Computeralgebrapakket	http://magma.maths.usyd.edu.au/magma/
Maple	Computeralgebrapakket	http://www.maplesoft.com/
Mathematica	Computeralgebrapakket	http://www.wolfram.com/products/mathematica/index.html

Programma	**Soort**	**Website**
Matlab	Pakket voor numerieke berekeningen	http://www.mathworks.com/products/matlab/
Maxima	Computeralgebrapakket	http://maxima.sourceforge.net/
Mizar	Formeel verificatieprogramma	http://www.mizar.org/
NetBSD	Besturingssysteem	http://www.netbsd.org/
Ωmega	Bewijsplanner	http://www.ags.uni-sb.de/~omega/omega/index.php
OSF	Besturingssysteem	
Otter	Automatisch bewijsprogramma	http://www.cs.unm.edu/~mccune/otter/
Pari-GP	Computeralgebrapakket voor getaltheorie	http://pari.math.u-bordeaux.fr/
PVS	Automatisch bewijsprogramma	http://pvs.csl.sri.com/
qMultiSum	MATHEMATICA-pakket om q-hypergeometrische identiteiten te bewijzen	http://www.risc.uni-linz.ac.at/research/combinat/risc/software/qMultiSum/
RealSearch	Programma voor intervalrekenkunde	
Solaris	Besturingssysteem	http://www.sun.com/software/solaris/
Weierstrass	Automatisch bewijsprogramma	http://www.cs.sjsu.edu/faculty/beeson/Papers/weier.html

Bijlage B. Namenlijst

Wegens het groot aantal personen met verschillende achtergronden die ik in deze eindverhandeling aanhaal, geef ik hier een lijst van de meeste besproken personen met hun achtergrond: filosoof, wiskundige, computerwetenschapper, …

Persoon	**Achtergrond**
Andrew Aberdein	filosoof, logicus en wiskundige
Andrew Adams	computerwetenschapper en wiskundige
Leonard M. Adleman	computerwetenschapper en wiskundige
Martin Aigner	wiskundige
Donald J. Albers	wiskundige
Kenneth Appel	wiskundige
Michael Aschbacher	wiskundige
Jeremy Avigad	filosoof, logicus en wiskundige
Jody Azzouni	filosoof en wiskundige
Henk Barendregt	computerwetenschapper en wiskundige
Erik Barendsen	computerwetenschapper en wiskundige
O. Bradley Bassler	filosoof en wiskundige
Gertrud Josefine Bauer	computerwetenschapper
Michael Beeson	computerwetenschapper en wiskundige
Johan G. Belinfante	fysicus
Paul Benacerraf	filosoof
Christophe Benzmüller	computerwetenschapper
Alexander Berkovich	wiskundige
George David Birkhoff	wiskundige
Izabela Bondecka-Krzykowska	wiskundige en logicus
F. F. Bonsall	wiskundige
Raj Chandra Bose	wiskundige
Nicolas Bourbaki	collectief van wiskundigen
Alan Bundy	wiskundige en logicus
Tyler Burge	filosoof
Stanley Burris	wiskundige
Stewart Scott Cairns	wiskundige

Persoon	**Achtergrond**
Andreea S. Calude	wiskundige en taalkundige
Georg Cantor	wiskundige
Olga Caprotti	computerwetenschapper
Bill Casselman	wiskundige
Arthur Cayley	wiskundige
Gregory Chaitin	computerwetenschapper en wiskundige
Kenneth Chang	fysicus, wetenschapsjournalist
Alonzo Church	wiskundige en logicus
Arjeh Cohen	wiskundige
Simon Colton	computerwetenschapper
John Horton Conway	wiskundige
Chris Coolsaet	computerwetenschapper
David Corfield	filosoof en wiskundige
Tony Crilly	wiskundige
Bernd Dahn	computerwetenschapper
Brian Davies	wiskundige
Martin Davis	computerwetenschapper en wiskundige
Philip J. Davis	wiskundige
John Dawson, Jr.	wiskundige
Augustus De Morgan	wiskundige en logicus
Jan Degraer	computerwetenschapper
Pierre Deligne	wiskundige
Michael Detlefsen	filosoof
Ian Dove	filosoof
Martin Dunstan	computerwetenschapper
Bruno Ernst	wiskundige
Leonhard Euler	wiskundige
Don Fallis	filosoof en psycholoog
Martin Farach	computerwetenschapper
Andrew Feist	wiskundige
Gabor Fejes Tóth	wiskundige
László Fejes Tóth	wiskundige

Persoon	**Achtergrond**
Samuel P. Ferguson	wiskundige
Richard Fikes	computerwetenschapper
Branden Fitelson	filosoof en wiskundige
Philip Franklin	wiskundige
David Gabai	wiskundige
Klaus Galda	wiskundige
Peter Louis Galison	filosoof en fysicus
Kurt Gödel	logicus en wiskundige
Georges Gonthier	computerwetenschapper
Chaim Goodman-Strauss	wiskundige
Marcel J.G. Gorissen	wiskundige
Hanne Gottliebsen	computerwetenschapper
Ronald Lewis Graham	wiskundige
Wolfgang Haken	wiskundige
Thomas Callister Hales	wiskundige
Marshall Hall, Jr.	wiskundige
Paul Richard Halmos	wiskundige
Godfrey Harold Hardy	wiskundige
Kenneth Harris	computerwetenschapper
Joel Hass	wiskundige
Percy John Heawood	wiskundige
Heinrich Heesch	wiskundige
Reuben Hersh	wiskundige
Jane Hesketh	wiskundige
David Hilbert	wiskundige
Petr Hlinený	computerwetenschapper
Leon Horsten	filosoof
Wu-Yi Hsiang	wiskundige
Tony Huang	computerwetenschapper
Edward Vermilye Huntington	wiskundige
Michael Hutchings	wiskundige
Andrew Ireland	computerwetenschapper

Persoon	**Achtergrond**
Dale Jacquette	filosoof
Mateja Jamnik	computerwetenschapper en wiskundige
Vaughan Frederick Randal Jones	wiskundige
Jerrold J. Katz	filosoof
Louis H. Kauffman	wiskundige
Tom Kelsey	computerwetenschapper
Alfred Bray Kempe	wiskundige
Manfred Kerber	womputerwetenschapper en wiskundige
Israel Kleiner	wiskundige
Gina Kolata	wiskundige, bioloog en wetenschapsjournalist
Israel Krakowksi	filosoof
Michal Krízek	wiskundige
Clement W.H. Lam	wiskundige
Leslie Lamport	computerwetenschapper en wiskundige
Derrick Henry Lehmer	wiskundige
Emma Lehmer	wiskundige
Steve Linton	computerwetenschapper
Florian Luca	wiskundige
Saunders Mac Lane	wiskundige
Robert Sinclair MacKay	wiskundige
Dana Mackenzie	wiskundige en wetenschapsjournalist
Donald MacKenzie	socioloog
Robert MacPherson	wiskundige
Florence Jessie MacWilliams	wiskundige
Penelope Maddy	filosoof en wiskundige
Colin L. Mallows	wiskundige
Paolo Mancosu	filosoof
Szolem Mandelbrojt	wiskundige
Yuri Ivanovich Manin	wiskundige
Allen Lawrence Mann	wiskundige
Ursula Martin	computerwetenschapper

Persoon	**Achtergrond**
Kenneth O. May	wiskundige
Jean Mayer	letterkundige en amateur-wiskundige
Roy L. McCasland	computerwetenschapper
William McCune	computerwetenschapper en wiskundige
Andreas Meier	wiskundige
Erica Melis	computerwetenschapper
G. Robert Meyerhoff	wiskundige
Gary L. Miller	computerwetenschapper
William H. Mills	wiskundige
John Mitchem	wiskundige
Raul Monroy	computerwetenschapper
David Mumford	wiskundige
Vicente Muñoz	wiskundige
Nils Nilsson	wiskundige
Tobias Nipkow	computerwetenschapper
F.H. Norwood	wiskundige
Alex Nunes	computerwetenschapper
Steven Obua	wiskundige
Martijn Oostdijk	computerwetenschapper
Sam Owre	computerwetenschapper
Peter Pagin	filosoof
E.T. Parker	wiskundige
Dominique Pastre	computerwetenschapper
I.C. Percival	wiskundige
Alan Perlis	wiskundige en scheikundige
Ulf Persson	wiskundige
Ivars Peterson	fysicus, scheikundige en wetenschapsjournalist
Marko Petkovsek	wiskundige
Martin Pollet	wiskundige
Carl Pomerance	wiskundige
Michael Oser Rabin	computerwetenschapper

Persoon	**Achtergrond**
Yehuda Rav	wiskundige
Peter Renz	wiskundige
Michael D. Resnik	filosoof
Julian Richardson	computerwetenschapper
Axel Riese	wiskundige
Neil Robertson	wiskundige
John Alan Robinson	wiskundige
Gian-Carlo Rota	filosoof en wiskundige
Herbert J. Ryser	wiskundige
Thomas L. Saaty	wiskundige en fysicus
Daniel P. Sanders	computerwetenschapper en wiskundige
David Sandborg	filosoof
Roger Schlafly	computerwetenschapper en wiskundige
Jacob T. Schwartz	computerwetenschapper en wiskundige
John L. Selfridge	wiskundige
Paul Seymour	wiskundige
Natarajan Shankar	computerwetenschapper
David H. Sharp	fysicus
Sharadchandra Shankar Shrikhande	wiskundige
Jörg Siekmann	computerwetenschapper en wiskundige
Carlos Simpson	wiskundige
Simon Singh	fysicus en wetenschapsjournalist
Neil James Alexander Sloane	wiskundige
Patrick F. Smith	wiskundige
Robert M. Solovay	wiskundige
Lawrence Somer	wiskundige
Volker Sorge	computerwetenschapper en wiskundige
Joel Spencer	wiskundige
Graham Steel	computerwetenschapper
Mark Steiner	computerwetenschapper
Ian Stewart	wiskundige
Volker Strassen	wiskundige

Persoon	**Achtergrond**
George Szpiro	wiskundige, fysicus en wetenschapsjournalist
Terence Tao	wiskundige
Alfred Tarski	wiskundige en logicus
Paul Teller	filosoof
Ruediger Thiele	wiskundige
Robin Thomas	wiskundige
John Griggs Thompson	wiskundige
Nathaniel Thurston	wiskundige
William Paul Thurston	wiskundige
Alan Mathison Turing	wiskundige
William Thomas Tutte	wiskundige
Thomas Tymoczko	filosoof
Dolph Ulrich	filosoof
Jean-Paul Van Bendegem	filosoof en wiskundige
Arno van den Essen	wiskundige
Bas van Fraassen	filosoof
Frank van Harmelen	computerwetenschapper en wiskundige
Oswald Veblen	wiskundige
Liza Verhoeven	wiskundige en filosoof
Robert Veroff	computerwetenschapper
Toby Walsh	computerwetenschapper, wiskundige en fysicus
Hao Wang	wiskundige, logicus en filosoof
Erik Weber	filosoof
Benjamin Werner	computerwetenschapper
Freek Wiedijk	computerwetenschapper en wiskundige
Catherine Womack	filosoof
Larry Wos	wiskundige
Doron Zeilberger	wiskundige
Günter Ziegler	wiskundige
Uri Zwick	computerwetenschapper

Bijlage C. Verklarende woordenlijst

afleidingsregel	Zie inferentieregel.
algebra, Booleaanse	Een algebraische structuur met de logische operatoren *en*, *of* en *niet*. Deze operatoren zijn direct gerelateerd aan de begrippen doorsnede, vereniging en complement uit de verzamelingenleer. De Brit George Boole ontwikkelde deze algebra om algebraïsche technieken te gebruiken voor logische uitdrukkingen.
algoritme	Een eindige reeks instructies om vanuit een bepaalde begintoestand een gedefinieerd doel te bereiken, zoals de berekening van het product van twee getallen. Het idee van een algoritme is gepreciseerd in het model van een Turing-machine. Zie ook Turing-machine.
associativiteit	Een binaire operatie * is associatief als (a*b)*c gelijk is aan a*(b*c). De volgorde van uitvoering van de operatie maakt dus niet uit.
axioma	Een niet bewezen maar als grondslag aanvaarde stelling.
bewijs, niet-constructief	Een bewijs dat het bestaan van een bepaald wiskundig object aantoont, maar niet onthult hoe dit object geconstrueerd kan worden. Zie ook existentiebewijs.
bewijs door contradictie	Een bewijs dat begint van de veronderstelling dat een bewering onwaar is en daaruit een contradictie afleidt. Dit bewijst dat de stelling waar moet zijn.
binomiaalcoëfficiënt	Een grootheid uit de combinatoriek die aangeeft op hoeveel manieren men uit n verschillende objecten zonder terugleggen k objecten kan kiezen. Zie ook combinatoriek.
bit	De kleinste eenheid van informatie
branch and bound	Een algemene methode om optimale oplossingen voor optimalisatieproblemen te vinden. Ze splitst het domein waarop een functie moet worden geoptimaliseerd op in subdomeinen (‘branch’) om dan op elk van deze domeinen een boven- en ondergrens van de functie te zoeken (‘bound’). Zie ook programmeren, lineair.

combinatoriek	De tak van de wiskunde die discrete (en meestal een eindig aantal) objecten bestudeert. Vragen die in de combinatoriek voorkomen zijn het tellen en construeren van objecten die aan bepaalde criteria voldoen.
commutativiteit	Een binaire operatie is commutatief als de waarde niet verandert wanneer de twee argumenten van plaats verwisselen.
compiler	Een computerprogramma dat een invoer in een bepaalde programmeertaal vertaalt in een door een computer uitvoerbaar programma. Het vertalen of omzetten wordt compilatie of compileren genoemd.
configuratie, reduceerbare	Zie Paragraaf 2.2, “De basisideeën van het bewijs” in Hoofdstuk 2, *De vierkleurenstelling*.
determinant	In de lineaire algebra is de determinant van een vierkante matrix een getal dat berekend wordt uit de elementen van de matrix. De determinant stelt het georiënteerde volume voor van het parallellepipedum gevormd door de als vectoren opgevatte kolommen van de matrix. Zie ook matrix.
distributiviteit	Een binaire operatie * is distributief over een binaire operatie + wanneer a*(b+c) gelijk is aan a*b + a*c. Een voorbeeld is de distributiviteit van vermenigvukldiging over optelling bij de natuurlijke, reële en complexe getallen.
duivenhokprincipe	Het eenvoudige maar krachtige duivenhokprincipe zegt dat, gegeven twee natuurlijke getallen n en m met n > m, als je n objecten in m dozen wil plaatsen, tenminste één doos meer dan één object zal bevatten.
equivalentierelatie	Een equivalentierelatie is reflexief, symmetrisch en transitief. Voor een relatie R wil dat zeggen: aRa, als aRb dan ook bRa, en als aRb en bRc dan ook aRc.
Eulervierkant	Zie vierkant, Grieks-Latijns.
existentiebewijs	Een bewijs van het bestaan van een bepaald wiskundig object. Zie ook bewijs, niet-constructief.
factorisatie, unieke	Elk natuurlijk getal groter dan 1 kan op een unieke manier geschreven worden als het product van priemgetallen. De ordening van de factoren in het product is niet belangrijk. Deze stelling staat bekend als de *fundamentele stelling van*

	de rekenkunde.
faculteit	Het product van alle getallen tot aan een bepaalde waarde n, genoteerd als n!. De faculteit wordt frequent gebruikt in de combinatoriek: ze geeft het antwoord op de vraag op hoeveel manieren je n elementen kunt rangschikken: dat kan op n! manieren. Zie ook combinatoriek.
Fano-vlak	Het projectief vlak van orde 2. Zie ook vlak, projectief.
Fermat, Laatste Stelling van	De stelling in de getaltheorie dat voor gehele getallen n groter dan 2 de vergelijking $a^n + b^n = c^n$ geen oplossing heeft voor gehele getallen a, b en c niet gelijk aan 0.
formalisme	Grondslagenstroming in de wiskunde, met als grootste pleitbezorger de wiskundige David Hilbert. Het formalisme herleidt wiskunde tot niets anders dan afleidingen maken in een formeel systeem. Zie ook systeem, formeel.
functie, lineaire	Een functie van de vorm $f(x) = mx + b$, waarin m en b reële constanten zijn en x een reële variabele.
functie, partieel-recursieve	Een functie die door een Turing-machine berekend wordt die niet op elke invoer stopt. Zie ook Turing-machine.
functie, trigonometrische	Een functie van een hoek, zoals de sinus-, cosinus- en tangensfuncties.
getal, irrationaal	Een reëel getal dat niet rationaal is, dus niet als een breuk kan geschreven worden.
graaf	Een verzameling van punten die verbonden zijn door lijnen.
graaf, gerichte	Een graaf waarvan de lijnen een richting hebben, aangeduid door een pijl. Zie ook graaf.
identiteit, trigonometrische	Een wiskundige gelijkheid tussen trigonometrische functies. Zie ook functie, trigonometrische.
incidentiematrix	Zie Hoofdstuk 3, *Er bestaat geen eindig projectief vlak van orde 10*.

inferentieregel	Een functie van een verzameling van formules naar een formule. De argumenten van een inferentieregel zijn de premissen, de functiewaarde ervan is de conclusie.
infinitesimaal	Een wiskundig object dat min of meer fungeert als een getal en dat in de ordening van de reële getallen kleiner is dan ieder positief reëel getal, maar toch groter is dan nul. Newton en Leibniz gebruikten infinitesimalen in hun uitwerking van de analyse, maar ze werden in de 19de eeuw vervangen door het limietbegrip dat beter kon geformaliseerd worden. In 1960 toonde Abraham Robinson in zijn *niet-standaard analyse* aan dat infinitesimalen ook kunne geformaliseerd worden met behoud van de kracht van de standaard analyse.
informatietheorie, algoritmische	De studie van de relatie tussen informatie en berekeningen, in de late jaren 1960 onafhankelijk van elkaar ontwikkeld door Andrey Kolmogorov, Ray Solomonoff en Gregory Chaitin.
intervalrekenkunde	Zie Paragraaf 7.3, “Numerieke benaderingen van ongelijkheden” in Hoofdstuk 7, *Andere computerbewijzen*.
intuïtionisme	Grondslagenstroming in de wiskunde die rond 1900 opkwam en systematisch is uiteengezet door de Nederlandse wiskundige Brouwer. Volgens het intuïtionisme is wiskunde essentieel een *constructie-activiteit* en de enige geldige bewijzen zijn dan ook *constructieve bewijzen*.
Keplervermoeden	Zie Hoofdstuk 5, *Het Keplervermoeden*.
lemma	Een hulpstelling die moet bewezen worden in de loop van een bewijs van een belangrijker resultaat.
manifold	Zie variëteit.
matrix	Een tabel van getallen of andere wiskundige entiteiten die opgeteld en vermenigvuldigd kunnen worden.
meetkunde, analytische	Het door Descartes ingevoerde systeem om problemen in de Euclidische meetkunde om te zetten in algebraïsche problemen met meetkundige coördinaten. Zie ook meetkunde, Euclidische.
meetkunde, hyperbolische	Een niet-Euclidische meetkunde, door het afwijzen van het parallellenpostulaat, een axioma uit de Euclidische meetkunde.

	Zie ook meetkunde, Euclidische.
meetkunde, Euclidische	De door de Griekse wiskundige voor het eerste geaxiomatiseerde eerste-orde theorie van vlakke meetkunde en ruimtemeetkunde.
modulo	Van twee natuurlijke getallen a en n zegt men dat a mod n gelijk is aan b als en slechts als de rest van a bi deling door n gelijk is aan b. Men noteert dit als $a \equiv b \pmod{n}$.
ordening, lineaire	Zie ordening, totale.
ordening, partiële	Een partiële ordening op een verzameling V is een reflexieve, antisymmetrische en transitieve relatie op V. Voor een relatie R wil dat zeggen: aRa, als aRb en bRa dan geldt a=b, en als aRb en bRc dan ook aRc.
ordening, totale	Als alle elementen in een partiële ordening met elkaar vergelijkbaar zijn, spreken we van een totale of lineaire ordening. Zie ook ordening, partiële.
ordinaalgetal	Geeft de positie van een element in een rij van elementen aan. De ordinaalgetallen zijn een uitbreiding van de natuurlijke getallen. Ze werden ingevoerd door Georg Cantor om oneindige verzamelingen met elkaar te kunnen vergelijken.
pad, Hamiltoniaans	Zie Paragraaf 6.2, “DNA-berekeningen” in Hoofdstuk 6, *Probabilistische computerbewijzen*.
Pappus, stelling van	Zie Paragraaf 1.8, “Het computerbewijs van de stelling van Pappus” in Hoofdstuk 1, *Inleiding*.
Peano-rekenkunde	De eerste-orde theorie van de natuurlijke getallen met optelling, vermenigvuldiging en een oneiding inductie-schema.
Presburger-rekenkunde	De eerste-orde theorie van de natuurlijke getallen met optelling, in 1929 gepubliceerd door Mojzesz Presburger. De Presburger-rekenkunde is niet zo krachtig als de Peano-rekenkunde, aangezien ze vermenigvuldiging mist. Zie ook Peano-rekenkunde.
priemgetalstelling	De priemgetalstelling geeft een schatting voor het aantal priemgetallen kleiner dan een willekeurig reëel getal x.
programmeren, lineair	Een methode voor het oplossen van optimalisatieproblemen waarbij de doelfunctie en de randvoorwaarden alle li-

	neair zijn. Zie ook branch and bound.
reductio ad absurdum	Zie bewijs door contradictie.
residu, kubisch	Zie Paragraaf 7.2, “Combinatorische oplossingen” in Hoofdstuk 7, *Andere computerbewijzen*.
resolutie	Zie Paragraaf 4.3, “Logisch redeneren” in Hoofdstuk 4, *Het Robbinsprobleem*.
Robbinsprobleem	Zie Hoofdstuk 4, *Het Robbinsprobleem*.
rooster, vlak gecentreerd kubisch	Zie Hoofdstuk 5, *Het Keplervermoeden*.
systeem, formeel	Een combinatie van een formele taal en een verzameling afleidings- of transformatieregels die zinnen in de formele taal omzetten in nieuwe zinnen. Voorbeelden zijn propositie- en predikatenlogica.
tijd, exponentiële	Een probleem of algoritme heeft een exponentiële tijdscomplexiteit als het aantal benodigde stappen een exponentiële functie is van de invoer. Zie ook tijdscomplexiteit.
tijd, kwadratische	Een probleem of algoritme heeft een kwadratische tijdscomplexiteit als het aantal benodigde stappen een kwadratische functie is van de invoer. Zie ook tijdscomplexiteit.
tijdscomplexiteit	Het aantal stappen dat een algoritme nodig heeft om een probleem op te lossen in functie van de grootte van de invoer. Zie ook tijd, exponentiële, tijd, kwadratische.
torus	Een driedimensionale meetkundige vorm die ontstaat door een cirkel te wentelen om een lijn die zich in het vlak van de cirkel bevindt maar niet door de cirkel loopt. Het resultaat heeft de vorm van een binnenband van een auto of fiets.
Turing-machine	Een abstract model van een computer dat preciseert wat een algoritme is. Zie ook algoritme, functie, partieel-recursieve.
unificatie	Zie Paragraaf 4.3, “Logisch redeneren” in Hoofdstuk 4, *Het Robbinsprobleem*.
variëteit	Een variëteit is een topologische ruimte die in voldoende

	kleine omgevingen van elk punt lijkt op de n-dimensionale Euclidische ruimte, maar die globaal een heel andere structuur kan hebben Zie ook meetkunde, Euclidische.
vector	Een ‘pijl’ in een vectorruimte, gekarakteriseerd door een grootte en een richting. Zie ook vectorruimte.
vectorkruisproduct	Wiskundige bewerking op twee driedimensionale vectoren, met een vector als resultaat die loodrecht staat op de twee gegeven vectoren. Zie ook vector.
vectorruimte	Een verzameling van vectoren waarop de operaties optelling en vermenigvuldiging gedefinieerd zijn. Voorbeelden van vectorruimtes zijn de twee- en driedimensionale Euclidische ruimte. Zie ook vector.
verzameling, onvermijdelijke	Zie Paragraaf 2.2, “De basisideeën van het bewijs” in Hoofdstuk 2, *De vierkleurenstelling*.
vierkant, Grieks-Latijns	Zie Hoofdstuk 3, *Er bestaat geen eindig projectief vlak van orde 10*.
vierkant, Latijns	Zie Hoofdstuk 3, *Er bestaat geen eindig projectief vlak van orde 10*.
vierkleurenstelling	Zie Hoofdstuk 2, *De vierkleurenstelling*.
vlak, projectief	Zie Hoofdstuk 3, *Er bestaat geen eindig projectief vlak van orde 10*.
ZFC	De standaard axiomatisering van de verzamelingentheorie. ZF staat voor Zermel-Fraenkel verzamelingentheorie en C voor het keuzeaxioma (‘axiom of choice’).

Literatuurlijst

[Aberdein2006] Andrew Aberdein, 'The informal logic of mathematical proof', in Reuben Hersh (ed.), *18 unconventional essays on the nature of mathematics* (2006), Springer, New York, pp. 56-70

[Adams1999] Andrew Adams, Hanne Gottliebsen, Steve Linton en Ursula Martin, 'Automated theorem proving in support of computer algebra: symbolic definite integration as a case study', *Proceedings of the 1999 international symposium on Symbolic and algebraic computation* (1999), pp. 253-260

[Adams2001] Andrew Adams, Martin Dunstan, Hanne Gottliebsen, Tom Kelsey, Ursula Martin en Sam Owre, 'Computer Algebra meets Automated Theorem Proving: Integrating Maple and PVS', *Lecture Notes in Computer Science*, 14th International Conference on Theorem Proving in Higher Order Logics, Vol. 2152 (2001), pp. 27-42

[Adleman1994] Leonard M. Adleman, 'Molecular computation of solutions to combinatorial problems', *Science*, Vol. 266, No. 5187 (1994), pp. 1021-1024

[Aigner2004] Martin Aigner en Günter Ziegler, *Proofs from THE BOOK*, Third Edition (2004), Springer, pp. 239

[Albers1981] Donald J. Albers, 'Polite applause for a proof of one of the great conjectures of mathematics: what is a proof today?', *The Two-Year College Mathematics Journal*, Vol. 12, No. 2 (1981), pp. 82

[Appel1977] Kenneth Appel en Wolfgang Haken, 'Every planar map is four-colorable', Part I. Discharging, *Illinois Journal of Mathematics*, Vol. 21 (1977), pp. 429-490

[Appel1977b] Kenneth Appel, Wolfgang Haken en John Koch, 'Every planar map is four-colorable', Part II. Reducibility, *Illinois Journal of Mathematics*, Vol. 21 (1977), pp. 491-567

[Appel1977c] Kenneth Appel en Wolfgang Haken, 'The solution of the four-color map problem', *Scientific American*, Vol. 237, No. 4 (1977), pp. 108-121

[Appel1978] Kenneth Appel en Wolfgang Haken, 'The four-color problem', in Lynn Arthur Steen (ed.), *Mathematics today: twelve informal essays* (1978), Springer-Verlag, New York, pp. 153-180, in [Jacquette2001], pp. 193-208

[Aschbacher2004] Michael Aschbacher, 'The status of the classification of the finite simple groups', *Notices of the American Mathematical Society*, Vol. 51, No. 7 (2004), pp. 736-740

[Aschbacher2005] Michael Aschbacher, 'Highly complex proofs and implications of such proofs', *Philosophical Transactions of the Royal Society A*, Vol. 363 (2005),

pp. 2401-2406

[Avigad2006] Jeremy Avigad, ‘Mathematical method and proof’, *Synthese*, Vol. 153, No. 1 (2006), pp. 105-159

[Avigad2007] Jeremy Avigad, ‘Understanding proofs’, in [Mancosu2007]

[Avigad2007b] Jeremy Avigad, ‘Computers in mathematical inquiry’, in [Mancosu2007]

[Azzouni2004] Jody Azzouni, ‘The derivation-indicator view of mathematical practice’, *Philosophia Mathematica*, Vol. 12, No. 2 (2004), pp. 81-106

[Azzouni2005] Jody Azzouni, ‘Is there still a sense in which mathematics can have foundations?’, in Giandomenico Sica (ed.), *Essays on the Foundations of Mathematics and Logic.*, Advanced Studies in Mathematics and Logic, 1 (2005), Polimetrica, Monza, pp. 9-47

[Barendregt2001] Henk Barendregt en Arjeh M. Cohen, ‘Electronic communication of mathematics and the interaction of computer algebra systems and proof assistants’, *Journal of Symbolic Computation*, Vol. 32 (2001), pp. 3-22

[Barendregt2002] Henk Barendregt en Erik Barendsen, ‘Autarkic computations in formal proofs’, *Journal of Automated Reasoning*, Vol. 28, No. 3 (2002), pp. 321-336

[Barendregt2005] Henk Barendregt en Freek Wiedijk, ‘The challenge of computer mathematics’, *Philosophical Transactions of the Royal Society*, Vol. 363 (2005), pp. 2351-2375

[Bassler2006] O. Bradley Bassler, ‘The surveyability of mathematical proof: a historical perspective’, *Synthese*, Vol. 148 (2006), pp. 99-133

[Bauer2006] Gertrud Josefine Bauer, *Formalizing plane graph theory - towards a formalized proof of the Kepler conjecture*, Thesis voor de graad van doctor in de natuurwetenschappen, Fakultät für Informatik, Technische Universiteit München (2006), pp. 188

[Bauer2006b] Gertrud Bauer en Tobias Nipkow, ‘Towards a verified enumeration of all tame plane graphs’, in Thierry Coquand, Henri Lombardi, Marie-Françoise Roy, *Mathematics, Algorithms, Proofs* (2006), Internationales Begegnungs- und Forschungszentrum fuer Informatik, Schloss Dagstuhl

[Beeson1998] Michael Beeson, ‘Automatic derivation of epsilon-delta proofs of continuity’, *Lecture Notes in Artificial Intelligence*, Artificial Intelligence and Symbolic Computation, Vol. 1476 (1998), pp. 67-83

[Beeson2001] Michael Beeson, ‘Automatic derivation of the irrationality of e’, *Journal of Symbolic Computation*, Vol. 32, No. 4 (2001), pp. 333-349

[Beeson2003] Michael Beeson, ‘The mechanization of mathematics’, in C. Teuscher (ed.), *Alan Turing: Life and Legacy of a Great Thinker* (2003), Springer-Verlag,

pp. 77-134

[Belinfante1999] Johan G. Belinfante, ‘On computer-assisted proofs in ordinal number theory’, *Journal of Automated Reasoning*, Vol. 22 (1999), pp. 341-378

[Benacerraf1965] Paul Benacerraf, ‘What numbers could not be’, *The Philosophical Review*, Vol. 74, No. 1 (1965), pp. 47-73

[Benacerraf1973] Paul Benacerraf, ‘Mathematical truth’, *The Journal of Philosophy*, Vol. 70, No. 19 (1973), pp. 661-679

[Benzmüller2001] Christoph Benzmüller, Andreas Meier, Erica Melis, Martin Pollet, Jörg Siekmann en Volker Sorge, ‘Proof planning: a fresh start?’, *Workshop on Future Directions in Automated Reasoning (W1) on IJCAR 2001* (2001), pp. 25-37

[Berkovich2002] Alexander Berkovich en Axel Riese, ‘A computer proof of a polynomial identity implying a partition theorem of Göllnitz’, *Advances in Applied Mathematics*, Vol. 28 (2002), pp. 1-16

[Bondecka2004] Izabela Bondecka-Krzykowska, ‘The four-color theorem and its consequences for the philosophy of mathematics’, *Annales UMCS Informatica AI*, Vol. 2 (2004), pp. 5-14

[Bonsall1982] F. F. Bonsall, ‘A down-to-earth view of mathematics’, *The American Mathematical Monthly*, Vol. 89, No. 1 (1982), pp. 8-15

[Bose1959] Raj Chandra Bose en Sharadchandra Shankar Shrikhande, ‘On the falsity of Euler's conjecture about the non-existence of two orthogonal Latin squares of order 4t + 2’, *Proceedings of the National Academy of Sciences*, Vol. 45 (1959), pp. 734-737

[Bose1960] Raj Chandra Bose en Sharadchandra Shankar Shrikhande, ‘On the construction of sets of mutually orthogonal Latin squares and the falsity of a conjecture of Euler’, *Transactions of the American Mathematical Society*, Vol. 95, No. 2 (1960), pp. 191-209

[Bundy1988] Alan Bundy, ‘The use of explicit plans to guide inductive proofs’, *Lecture Notes in Computer Science*, Ninth Conference on Automated Deduction, Vol. 203 (1988), pp. 111-120

[Bundy1991] Alan Bundy, Frank van Harmelen, Jane Hesketh en Alan Smaill, ‘Experiments with proof plans for induction’, *Journal of Automated Reasoning*, Vol. 7, No. 3 (1991), pp. 303-324

[Bundy1991b] Alan Bundy, ‘A science of reasoning’, in Jean-Louis Lassez, Gordon Plotkin, *Computational Logic: Essays in Honor of Alan Robinson* (1991), pp. 178-198, MIT Press

[Bundy1993] Alan Bundy, Andrew Stevens, Frank van Harmelen, Andrew Ireland en Alan Smaill, 'Rippling: A heuristic for guiding inductive proofs', *Artificial Intelligence*, Vol. 62, No. 2 (1993), pp. 185-253

[Bundy1998] Alan Bundy, 'Proof planning', *Proceedings of the 3rd International Conference on AI Planning Systems* (1998), pp. 261-267

[Bundy1999] Alan Bundy, Simon Colton en Toby Walsh, 'HR - A system for machine discovery in finite algebras', *Proceedings of the Machine Discovery Workshop ECAI99* (1999)

[Bundy1999b] Alan Bundy en Julian Richardson, 'Proofs about lists using ellipsis', *Lecture Notes in Artificial Intelligence*, Proceedings of the 6th International Conference, Logic for Programming and Automated Reasoning, Vol. 1705 (1999), pp. 1-12

[Bundy1999c] Alan Bundy, 'A survey of automated deduction', *Lecture Notes in Computer Science*, Artificial Intelligence Today: Recent Trends and Developments, Vol. 1600 (1999), pp. 153-174

[Bundy2001] Alan Bundy, 'The automation of proof by mathematical induction', in [Robinson2001]

[Bundy2005] Alan Bundy, Mateja Jamnik en Andrew Fugard, 'What is a proof?', *Philosophical Transactions of the Royal Society*, Vol. 363 (2005), pp. 2377-2391

[Bundy2006] Alan Bundy, 'The Boole Lecture', A very mathematical dilemma, *The Computer Journal*, Vol. 49, No. 4 (2006), pp. 480-486

[Burge1993] Tyler Burge, 'Content preservation', *The Philosophical Review*, Vol. 102, No. 4 (1993), pp. 457-488

[Burge1998] Tyler Burge, 'Computer proof, apriori knowledge, and other minds', *Philosophical Perspectives*, Vol. 12 (Language, Mind, and Ontology) (1998), pp. 1-37

[Burris1996] Stanley Burris, *An anthropomorphized version of McCune's machine proof that Robbins algebras are Boolean algebras* (1996), Ongepubliceerd manuscript (http://www.cs.unm.edu/~mccune/old-ftp/www-misc/burris-robbins-prf.ps.gz)

[Cairns1954] Stewart Scott Cairns, 'Computational attacks on discrete problems', *The American Mathematical Monthly*, Vol. 61, No. 7 (1954), pp. 29-31

[Calude2001] Andreea S. Calude, 'The journey of the four colour theorem through time', *The New Zealand Mathematics Magazine*, Vol. 38, No. 3 (2001), pp. 27-35

[Caprotti2001] Olga Caprotti en Martijn Oostdijk, 'Formal and efficient primality proofs by use of computer algebra oracles', *Journal of Symbolic Computation*, Vol. 32 (2001), pp. 55-70

[Casselman2000] Bill Casselman, 'Pictures and proofs', *Notices of the American Mathe-*

matical Society, Vol. 47, No. 10 (2000), pp. 1257-1266

[Cayley1879] Arthur Cayley, 'On the colouring of maps', *Proceedings of the Royal Geographical Society and Monthly Record of Geography*, Vol. 1, No. 4 (1879), pp. 259-261

[Cerutti1969] Elsie Cerutti en Philip J. Davis, 'Formac meets Pappus: some observations on elementary analytic geometry by computer', *The American Mathematical Monthly*, Vol. 76, No. 8 (1969), pp. 895-905

[Chaitin1974] Gregory J. Chaitin, 'Information-theoretic computational complexity', *IEEE Transactions on Information Theory*, Vol. IT-20, No. 1 (1974), pp. 10-15, in [Tymoczko1998]

[Chaitin1975] Gregory J. Chaitin, 'Randomness and mathematical proof', *Scientific American*, Vol. 232, No. 5 (1975), pp. 47-52

[Chaitin1978] Gregory J. Chaitin en Jacob T. Schwartz, 'A note on Monte Carlo primality tests and algorithmic information theory', *Communications on Pure and Applied Mathematics*, Vol. 31 (1978), pp. 521-527

[Chaitin1982] Gregory J. Chaitin, 'GÃ¶del's theorem and information', *International Journal of Theoretical Physics*, Vol. 22 (1982), pp. 941-954, in [Tymoczko1998]

[Chaitin2002] Gregory J. Chaitin, 'Computers, paradoxes and the foundations of mathematics', *American Scientist*, Vol. 90 (2002), pp. 164-171

[Chaitin2004] Gregory Chaitin, *Irreducible complexity in pure mathematics* (2004), http://arxiv.org/abs/math.HO/0411091

[Chang2004] Kenneth Chang, 'In math, computers don't lie. Or do they?', *The New York Times* (2004)

[Colton1999] Simon Colton, Alan Bundy en Toby Walsh, 'Automatic concept formation in pure mathematics', *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* (1999), pp. 786-793

[Colton1999b] Simon Colton, 'Refactorable numbers - a machine invention', *Journal of Integer Sequences*, Vol. 2 (1999), article 99.1.2

[Colton2000] Simon Colton, Alan Bundy en Toby Walsh, 'On the notion of interestingness in automated mathematical discovery', *International Journal of Human Computer Studies*, Vol. 53, No. 3 (2000), pp. 351-375

[Conway1999] John Horton Conway, Chaim Goodman-Strauss en Neil James Alexander Sloane, 'Recent progress in sphere packing', in Barry Mazur, W. Schmid, S. T. Yau, D. Jerison, I. Singer, D. Stroock, *Current Developments in Mathematics* (1999), International Press, Somerville, MA, pp. 37-76

[Coolsaet2005] Kris Coolsaet en Jan Degraer, 'A computer-assisted proof of the unique-

ness of the Perkel graph', *Designs, Codes and Cryptography*, Vol. 34 (2005), pp. 155-171

[Corfield2003] David Corfield, *Towards a philosophy of real mathematics* (2003), Cambridge University Press, pp. 287

[Crilly2005] Tony Crilly, 'Arthur Cayley FRS and the four-colour map problem', *Notes & Records of the Royal Society*, Vol. 59 (2005), pp. 285-304

[Dahn1997] Bernd Dahn, *An explanation of EQP/Otter's proof of Winker's second condition for Robbins algebras* (1997), Ongepubliceerd manuscript

[Davies2005] Brian Davies, 'Whither Mathematics?', *Notices of the American Mathematical Society*, Vol. 52, No. 11 (2005), pp. 1350-1356

[Davis1972] Philip J. Davis, 'Fidelity in mathematical discourse: is one and one really two?', *The American Mathematical Monthly*, Vol. 79, No. 3 (1972), pp. 252-263, in [Tymoczko1998], pp. 163-176

[Davis2001] Martin Davis, 'The early history of automated deduction', in [Robinson2001], pp. 3-15

[Dawson2006] John W. Dawson, Jr, 'Why do mathematicians re-prove theorems?', *Philosophia Mathematica*, Vol. 14, No. 3 (2006), pp. 269-286

[DeMillo1979] Richard De Millo, Richard Lipton en Alan Perlis, 'Social processes and proofs of theorems and programs', *Communications of the American Mathematical Society*, Vol. 22, No. 5 (1979), pp. 271-280, in [Tymoczko1998], pp. 267-286

[Detlefsen1980] Michael Detlefsen en Mark Luker, 'The four-color theorem and mathematical proof', *The Journal of Philosophy*, Vol. 77, No. 12 (1980), pp. 803-820

[Dove2002] Ian Dove, 'Can pictures prove?', *Logique & Analyse*, Vol. 179-180 (2002), pp. 309-340

[Ernst2002] Bruno Ernst (ed.), *De interessantste bewijzen voor de stelling van Pythagoras* (2002), Epsilon Uitgaven, Utrecht, pp. 95

[Euler1738] Leonhard Euler, 'Observationes de theoremate quodam Fermatiano aliisque ad numeros primos spectantibus', *Commentarii academiae scientiarum imperialis Petropolitanae*, Vol. 6 (1738), pp. 103-107, Vertaald in het Engels op http://arxiv.org/PS_cache/math/pdf/0501/0501118.pdf

[Euler1750] Leonhard Euler, 'Theoremata circa divisores numerorum', *Novi Commentarii academiae scientiarum Petropolitanae*, Vol. 1 (1750), pp. 20-48, Vertaald in het Engels op http://www.cs.utexas.edu/users/wzhao/e134.pdf

[Fallis1996] Don Fallis, 'Mathematical proof and the reliability of DNA evidence', *The American Mathematical Monthly*, Vol. 103, No. 6 (1996), pp. 491-497

[Fallis1997] Don Fallis, 'The epistemic status of probabilistic proof', *The Journal of Phi-*

losophy, Vol. 94, No. 4 (1997), pp. 165-186

[Fallis2002] Don Fallis, 'What do mathematicians want? Probabilistic proofs and the epistemic goals of mathematicians', *Logique & Analyse*, Vol. 179-180 (2002), pp. 373-388

[Fallis2003] Don Fallis, 'Intentional gaps in mathematical proofs', *Synthese*, Vol. 134 (2003), pp. 45-69

[Ferguson2006] Samuel P. Ferguson, 'Sphere packings', V. Pentahedral prisms, *Discrete and Computational Geometry*, Vol. 36, No. 1 (2006), pp. 167-204

[Fikes1971] Richard Fikes en Nils Nilsson, 'STRIPS: A new approach to the application of theorem proving to problem solving', *Artificial Intelligence*, Vol. 2 (1971), pp. 189-208

[Fitelson1998] Branden Fitelson, 'Using Mathematica to Understand the Computer Proof of the Robbins Conjecture', *Mathematica in Education and Research*, Vol. 7, No. 1 (1998), pp. 17-26

[Franklin1922] Philip Franklin, 'The four color problem', *American Journal of Mathematics*, Vol. 44, No. 3 (1922), pp. 225-236

[Gabai2003] David Gabai, G. Robert Meyerhoff en Nathaniel Thurston, 'Homotopy hyperbolic 3-manifolds are hyperbolic', *Annals of Mathematics*, Vol. 157, No. 2 (2003), pp. 335-431

[Galda1981] Klaus Galda, 'An informal history of formal proofs: from vigor to rigor?', *The Two-Year College Mathematics Journal*, Vol. 12, No. 2 (1981), pp. 126-140

[Gödel1931] Kurt Gödel, 'Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I', *Monatshefte für Mathematik und Physik*, Vol. 38 (1931), pp. 173-198

[Gonthier2004] Georges Gonthier, *A computer-checked proof of the four colour theorem* (2004), Ongepubliceerd manuscript (http://research.microsoft.com/~gonthier/4colproof.pdf)

[Gorissen2007] Marcel J.G. Gorissen, *Generating finite projective planes from non-paratopic Latin squares*, Thesis voor de graad van Master of Science in Mathematics, Radboud Universiteit Nijmegen (2007), http://www.math.kun.nl/~bosma/students/gorissen/gorissenmscthesis.pdf

[Hales2000] Thomas C. Hales, 'Cannonballs and honeycombs', *Notices of the American Mathematical Society*, Vol. 47, No. 4 (2000), pp. 440-449

[Hales2001] Thomas C. Hales, 'The honeycomb conjecture', *Discrete and Computational Geometry*, Vol. 25 (2001), pp. 1-22

[Hales2002] Thomas C. Hales, *Sphere packings in 3 dimensions* (2002), Arbeitstagung

lezing in Bonn, 14 juni 2001 (http://www.math.pitt.edu/~thales/PUBLICATIONS/spherepacking3.pdf)

[Hales2002b] Thomas C. Hales, 'A computer verification of the Kepler conjecture', *Proceedings of the International Congress of Mathematicians*, Vol. 3 (2002), pp. 795-804

[Hales2003] Thomas C. Hales, 'Some algorithms arising in the proof of the Kepler conjecture', *Algorithms and Combinatorics*, Vol. 25 (2003), pp. 489-508

[Hales2005] Thomas C. Hales, 'A proof of the Kepler conjecture', *Annals of Mathematics*, Vol. 162, No. 3 (2005), pp. 1065-1185

[Hales2006] Thomas C. Hales, 'Historical overview of the Kepler conjecture', *Discrete and Computational Geometry*, Vol. 36, No. 1 (2006), pp. 5-20

[Hales2006b] Thomas C. Hales en Samuel P. Ferguson, 'A formulation of the Kepler conjecture', *Discrete and Computational Geometry*, Vol. 36, No. 1 (2006), pp. 21-69

[Hales2006c] Thomas C. Hales, 'Sphere packing', III. Extremal cases, *Discrete and Computational Geometry*, Vol. 36, No. 1 (2006), pp. 71-110

[Hales2006d] Thomas C. Hales, 'Sphere packing', IV. Detailed bounds, *Discrete and Computational Geometry*, Vol. 36, No. 1 (2006), pp. 111-166

[Hales2006e] Thomas C. Hales, 'Sphere packings', VI. Tame graphs and linear programs, *Discrete and Computational Geometry*, Vol. 36, No. 1 (2006), pp. 205-265

[Hales2006f] Thomas C. Hales, 'Introduction to the Flyspeck project', in Thierry Coquand, Henri Lombardi, Marie-FranÃ§oise Roy, *Mathematics, Algorithms, Proofs* (2006), Internationales Begegnungs- und Forschungszentrum fuer Informatik, Schloss Dagstuhl

[Hall1955] Marshall Hall Jr., 'Finite projective planes', *The American Mathematical Monthly*, Vol. 62, No. 7 (1955), pp. 18-24

[Halmos1990] Paul Richard Halmos, 'Has progress in mathematics slowed down?', *The American Mathematical Monthly*, Vol. 97, No. 7 (1990), pp. 561-588

[Hardy1929] Godfrey Harold Hardy, 'Mathematical proof', *Mind*, Vol. 38, No. 149 (1929), pp. 1-25, in [Jacquette2001], pp. 173-186

[Hass1995] Joel Hass, Michael Hutchings en Roger Schlafly, 'The double bubble conjecture', *Electronic Research Announcements of the American Mathematical Society*, Vol. 1, No. 3 (1995), pp. 98-102

[Hass2000] Joel Hass en Roger Schlafly, 'Double bubbles minimize', *Annals of Mathematics*, Vol. 151, No. 2 (2000), pp. 459-515

[Heawood1890] Percy John Heawood, ‘Map colouring theorem’, *Quarterly Journal of Pure and Applied Mathematics*, Vol. 24 (1890), pp. 332-338

[Hersh1993] Reuben Hersh, ‘Proving is convincing and explaining’, *Educational Studies in Mathematics*, Vol. 24, No. 4 (1993), pp. 389-399

[Hersh1997] Reuben Hersh, ‘Prove –once more and again’, *Philosophia Mathematica*, Vol. 5, No. 2 (1997), pp. 153-165

[Hlinený2002] Petr Hlinený, ‘On the excluded minors for matroids of branch-width three’, *The Electronic Journal of Combinatorics*, Vol. 9, No. 1 (2002), http://www.combinatorics.org/Volume_9/Abstracts/v9i1r32.html

[Horsten2001] Leon Horsten, ‘Platonistic formalism’, *Erkenntnis*, Vol. 54, No. 2 (2001), pp. 173-194

[Horsten2004] Leon Horsten (ed.), *Eindig, oneindig, meer dan oneindig*, Grondslagen van de wiskundige wetenschappen (2004), Epsilon Uitgaven, Utrecht, pp. 203

[Hsiang1993] Wu-Yi Hsiang, ‘On the sphere packing problem and the proof of Kepler's conjecture’, *International Journal of Mathematics*, Vol. 4, No. 5 (1993), pp. 739-831

[Huang2002] Tony Huang, *Automated deduction in ring theory*, Master's Writing Project, Department of Computer Science, San Jose State University (2002)

[Huntington1933] Edward Vermilye Huntington, ‘New sets of independent postulates for the algebra of logic, with special reference to Whitehead and Russell's Principia Mathematica’, *Transactions of the American Mathematical Society*, Vol. 35, No. 1 (1933), pp. 274-304

[Huntington1933b] Edward Vermilye Huntington, ‘Boolean algebra. A correction’, *Transactions of the American Mathematical Society*, Vol. 35, No. 2 (1933), pp. 557-558

[Jacquette2001] Dale Jacquette (ed.), *Philosophy of mathematics: an anthology* (2001), Blackwell Publishers, pp. 428

[Jones1998] Vaughan Frederick Randal Jones, ‘A credo of sorts’, in H. G. Dales, Gianluigi Oliveri, *Truth in mathematics* (1998), Oxford University Press, pp. 203-214

[Katz1995] Jerrold J. Katz, ‘What mathematical knowledge could be’, *Mind*, Vol. 104, No. 415 (1995), pp. 491-522

[Kauffman1990] Louis H. Kauffman, ‘Robbins algebra’, *Proceedings of the Twentieth International Symposium on Multiple-Valued Logic* (1990), pp. 54-60

[Kauffman2001] Louis H. Kauffman, ‘The Robbins problem: computer proofs and human proofs’, *Kybernetes*, Vol. 30, No. 5/6 (2001), pp. 726-751

[Kempe1879] Alfred Bray Kempe, ‘On the geographical problem of the four colours’, *American Journal of Mathematics*, Vol. 2, No. 3 (1879), pp. 193-200

[Kerber1998] Manfred Kerber, Michael Kohlhase en Volker Sorge, ‘Integrating computer algebra into proof planning’, *Journal of Automated Reasoning*, Vol. 21, No. 3 (1998), pp. 327-355

[Kerber2002] Manfred Kerber en Martin Pollet, ‘On the design of mathematical concepts’, *Lecture Notes in Computer Science*, Proceedings of the 15th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence, Vol. 2557 (2002)

[Kleiner1991] Israel Kleiner, ‘Rigor and proof in mathematics: a historical perspective’, *Mathematics Magazine*, Vol. 64, No. 5 (1991), pp. 291-314

[Kolata1996] Gina Kolata, ‘Computer math proof shows reasoning power’, *The New York Times* (1996), http://www.nytimes.com/library/cyber/week/1210math.html

[Krakowski1980] Israel Krakowski, ‘The four color problem reconsidered’, *Philosophical Studies*, Vol. 38 (1980), pp. 91-96

[Krízek2001] Michal Krízek, Florian Luca en Lawrence Somer, *17 Lectures on Fermat numbers*, From number theory to geometry (2001), Springer, New York, pp. 257

[Lam1989] Clement W.H. Lam, L. Thiel en S. Swiercz, ‘The non-existence of finite projective planes of order 10’, *Canadian Journal of Mathematics*, Vol. XLI (1989), pp. 1117-1123

[Lam1991] Clement W.H. Lam, ‘The search for a finite projective plane of order 10’, *The American Mathematical Monthly*, Vol. 98, No. 4 (1991), pp. 305-318

[Lamport1995] Leslie Lamport, ‘How to write a proof’, *The American Mathematical Monthly*, Vol. 102, No. 7 (1995), pp. 600-608

[Lehmer1962] Derrick Henry Lehmer, Emma Lehmer, William H. Mills en John L. Selfridge, ‘Machine proof of a theorem on cubic residues’, *Mathematics of computation*, Vol. 16, No. 80 (1962), pp. 407-415

[Levin1981] Margarita Levin, ‘On Tymoczko's argument for mathematical empiricism’, *Philosophical Studies*, Vol. 39 (1981), pp. 79-86

[MacLane1997] Saunders Mac Lane, ‘Despite physicists, proof is essential in mathematics’, *Synthese*, Vol. 111, No. 2 (1997), pp. 147-154

[MacKay1985] Robert Sinclair MacKay en I. C. Percival, ‘Converse KAM: theory and practice’, *Communications in mathematical physics*, Vol. 98, No. 4 (1985), pp. 469-512

[MacKenzie1999] Donald MacKenzie, ‘Slaying the Kraken: the sociohistory of a mathematical proof’, *Social Studies of Science*, Vol. 29, No. 1 (1999), pp. 7-60

[Mackenzie2005] Dana Mackenzie, ‘What in the name of Euclid is going on here?’, *Science*, Vol. 307, No. 5714 (2005), pp. 1402-1403

[MacWilliams1973] Florence Jessie MacWilliams, Neil J.A. Sloane en John Griggs Thompson, ‘On the existence of a projective plane of order 10’, *Journal of Combinatorial Theory*, Vol. 14, No. 1 (1973), pp. 66-78

[Maddy1997] Penelope Maddy (ed.), *Naturalism in Mathematics* (1997), Clarendon Press, Oxford

[Mallows1974] Colin L. Mallows en Neil J.A. Sloane, ‘Weight enumerators of self-orthogonal codes’, *Discrete Mathematics*, Vol. 9 (1974), pp. 391-400

[Mancosu1991] Paolo Mancosu, ‘On the status of proofs by contradiction in the seventeenth century’, *Synthese*, Vol. 88 (1991), pp. 15-41

[Mancosu2001] Paolo Mancosu, ‘Mathematical explanation: problems and prospects’, *Topoi*, Vol. 20 (2001), pp. 97-117

[Mancosu2007] Paolo Mancosu (ed.), *The Philosophy of Mathematical Practice* (2007), Oxford University Press, Oxford

[Mandelbrojt1952] Szolem Mandelbrojt, ‘Pourquoi je fais des mathématiques’, *Revue de métaphysique et de morale*, Vol. 57, No. 4 (1952), pp. 442-429, *Cahiers du séminaire d'histoire des mathématiques*, Vol. 6 (1985), pp. 47-54

[Manin1981] Yuri I. Manin, ‘A digression on proof’, *The Two-Year College Mathematics Journal*, Vol. 12, No. 2 (1981), pp. 104-107

[Mann2003] Allen Lawrence Mann, *A case study in automated theorem proving: Otter and EQP*, Thesis submitted to the Faculty of the Graduate School of the University of Colorado for the degree of Master of Arts in Mathematics (2003), http://math.colorado.edu/~almann/MA/thesis.pdf

[Martin1999] Ursula Martin, ‘Computers, reasoning and mathematical practice’, *Computational Logic*, Vol. 165 (1999), pp. 301-346

[May1965] Kenneth O. May, ‘The origin of the four-color conjecture’, *Isis*, Vol. 56, No. 3 (1965), pp. 346-348

[Mayer1982] Jean Mayer, ‘Le théorème des quatre couleurs: notice historique et aperçu technique’, *Cahiers du séminaire d'histoire des mathématiques*, Vol. 3 (1982), pp. 43-62

[McCasland2006] Roy L. McCasland, Alan Bundy en Patrick F. Smith, ‘Ascertaining mathematical theorems’, *Electronic Notes in Theoretical Computer Science*, Proceedings of the 12th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning (Calculemus 2005), Vol. 151 (2006), pp. 21-38

[McCune1997] William McCune, ‘Solution of the Robbins problem’, *Journal of Auto-*

mated Reasoning, Vol. 19, No. 3 (1997), pp. 263-276

[McCune1997b] William McCune, 'Well-behaved search and the Robbins problem', *Proceedings of the 8th International Conference on Rewriting Techniques and Applications* (1997), pp. 1-7

[McCune2002] William McCune, Robert Veroff, Branden Fitelson, Kenneth Harris, Andrew Feist en Larry Wos, ' Short single axioms for Boolean algebra', *Journal of automated reasoning*, Vol. 29, No. 1 (2002), pp. 1-16

[Melis1996] Erica Melis en Alan Bundy, 'Planning and proof planning', S. Biundo (ed.), *ECAI-96 Workshop on Cross-Fertilization in Planning* (1996), pp. 37-40

[Miller1975] Gary L. Miller, 'Riemann's hypothesis and tests for primality', *Proceedings of seventh annual ACM symposium on Theory of computing* (1975), pp. 234-239

[Mitchem1981] John Mitchem, 'On the history and solution of the four-color map problem', *The Two-Year College Mathematics Journal*, Vol. 12, No. 2 (1981), pp. 108-116

[Monroy1994] Raul Monroy, Alan Bundy en Andrew Ireland, 'Proof plans for the correction of false conjectures', *Lecture Notes in Computer Science*, Proceedings of the Fifth International Conference on Logic Programming and Automated Reasoning, Vol. 822 (1994), pp. 54-68

[Mumford1967] David Mumford, 'On the equations defining abelian varieties. III', *Inventiones Mathematicae*, Vol. 3, No. 3 (1967), pp. 215-244

[Muñoz2006] Vicente Muñoz en Ulf Persson, 'Interviews with three Fields medallists', *Newsletter of the European Mathematical Society*, No. 62 (2006), pp. 32-36

[Nipkow2006] Tobias Nipkow, Gertrud Bauer en Paula Schultz, 'Flyspeck I: tame graphs', in U. Furbach, N. Shankar, *Automated Reasoning* (2006), Springer, pp. 21-35

[Norwood1982] F. H. Norwood, 'Long proofs', *The American Mathematical Monthly*, Vol. 89, No. 2 (1982), pp. 110-112

[Obua2006] Steven Obua, 'Proving bounds for real linear programs in Isabelle/HOL', in Thierry Coquand, Henri Lombardi, Marie-Françoise Roy, *Mathematics, Algorithms, Proofs* (2006), Internationales Begegnungs- und Forschungszentrum fuer Informatik, Schloss Dagstuhl

[Owings1973] James C. Owings, Jr, 'Diagonalization and the recursion theorem', *Notre Dame Journal of Formal Logic*, Vol. 14, No. 1 (1973), pp. 95-99

[Pagin1994] Peter Pagin, 'Knowledge of proofs', *Topoi*, Vol. 13 (1994), pp. 93-100, in [Jacquette2001], pp. 209-217

[Parker1959] E. T. Parker, 'Orthogonal Latin squares', *Proceedings of the National Aca-*

demy of Sciences, Vol. 45 (1959), pp. 859-862

[Parker1959b] E. T. Parker, 'Construction of some sets of mutually orthogonal Latin squares', *Proceedings of the American Mathematical Society*, Vol. 10, No. 6 (1959), pp. 946-949

[Pastre1999] Dominique Pastre, *Can and must a machine prove theorems as humans do ?*, Analysis of some automated proofs by the MUSCADET system (1999), Intern rapport (http://www.math-info.univ-paris5.fr/~pastre/CanAndMust.pdf)

[Paule2003] Peter Paule en Carsten Schneider, 'Computer proofs of a new family of harmonic number identities', *Advances in Applied Mathematics*, Vol. 31 (2003), pp. 359-378

[Peterson1997] Ivars Peterson, 'Computers and proof: applying automated reasoning to prove mathematical theorems', *Science News*, Vol. 151, No. 12 (1997), pp. 178

[Petkovsek1996] Marko Petkovsek, Herbert Wilf en Doron Zeilberger, *A = B* (1996), A K Peters, pp. 212

[Rabin1976] Michael O. Rabin, 'Probabilistic algorithms', in J. F. Traub (ed.), *Algorithms and complexity - New directions and recent results* (1976), Academic Press, New York, pp. 21-39

[Rabin1980] Michael O. Rabin, 'Probabilistic algorithm for testing primality', *Journal of Number Theory*, Vol. 12, No. 1 (1980), pp. 128-138

[Rav1999] Yehuda Rav, 'Why do we prove theorems?', *Philosophia Mathematica*, Vol. 7, No. 1 (1999), pp. 5-41

[Renz1981] Peter Renz, 'Mathematical proof: what it is and what it ought to be', *The Two-Year College Mathematics Journal*, Vol. 12, No. 2 (1981), pp. 83-103

[Resnik1981] Michael D. Resnik, 'Mathematics as a science of patterns: ontology and reference', *Noûs*, Vol. 15, No. 4 (Special Issue on Philosophy of Mathematics) (1981), pp. 529-550

[Resnik1982] Michael D. Resnik, 'Mathematics as a science of patterns: epistemology', *Noûs*, Vol. 16, No. 1 (1982), pp. 95-105

[Resnik1987] Michael D. Resnik en David Kushner, 'Explanation, independence and realism in mathematics', *The British Journal for the Philosophy of Science*, Vol. 38, No. 2 (1987), pp. 141-158

[Resnik1992] Michael D. Resnik, 'Proof as a source of truth', in Michael Detlefsen (ed.), *Proof and Knowledge in Mathematics* (1992), Routledge, pp. 6-32, in [Tymoczko1998], pp. 317-336

[Richardson1999] Julian Richardson en Alan Bundy, 'Proof planning methods as schemas', *Journal of Symbolic Computation*, Vol. 11 (1999)

[Riese2003] Axel Riese, ‘qMultiSum - A package for proving q-hypergeometric multiple summation identities’, *Journal of Symbolic Computation*, Vol. 35 (2003), pp. 349-376

[Robertson1996] Neil Robertson, Daniel P. Sanders, Paul Seymour en Robin Thomas, ‘A new proof of the four-colour theorem’, *Electronic Research Announcements of the American Mathematical Society*, Vol. 2, No. 1 (1996)

[Robertson1997] Neil Robertson, Daniel P. Sanders, Paul Seymour en Robin Thomas, ‘The four-colour theorem’, *Journal of Combinatorial Theory, Series B*, Vol. 70, No. 1 (1997), pp. 2-44

[Robinson1965] John Alan Robinson, ‘A machine-oriented logic based on the resolution principle’, *Journal of the Association for Computing Machinery*, Vol. 12 (1965), pp. 23-41

[Robinson2001] , Alan Robinson, Andrei Voronkov, *Handbook of Automated Reasoning* (2001), Elsevier Science Publishers

[Rota1985] Gian-Carlo Rota en David H. Sharp, ‘Mathematics, Philosophy, and Artificial Intelligence’, *Los Alamos Science*, No. 12 (1985), pp. 92-104

[Rota1997] Gian-Carlo Rota, ‘The phenomenology of mathematical beauty’, *Synthese*, Vol. 111, No. 2 (1997), pp. 171-182

[Rota1997b] Gian-Carlo Rota, ‘The phenomenology of mathematical proof’, *Synthese*, Vol. 111, No. 2 (1997), pp. 183-196, in [Jacquette2001], pp. 218-225

[Ryser1955] Herbert J. Ryser, ‘Geometries and incidence matrices’, *The American Mathematical Monthly*, Vol. 62, No. 7 (1955), pp. 25-31

[Saaty1972] Thomas L. Saaty, ‘Thirteen colorful variations on Guthrie's Four-Color Conjecture’, *The American Mathematical Monthly*, Vol. 79, No. 1 (1972), pp. 2-43

[Sandborg1998] David Sandborg, ‘Mathematical explanation and the theory of why-questions’, *The British Journal for the Philosophy of Science*, Vol. 49, No. 4 (1998), pp. 603-624

[Shankar1988] Natarajan Shankar, ‘A mechanical proof of the Church-Rosser theorem’, *Journal of the Association for Computing Machinery*, Vol. 35, No. 3 (1988), pp. 475-522

[Shankar2002] Natarajan Shankar, ‘Little engines of proof’, *Proceedings of FME 2002: Formal Methods - Getting IT Right* (2002), pp. 1-20

[Simpson2004] Carlos Simpson, ‘Computer theorem proving in mathematics’, *Letters in Mathematical Physics*, Vol. 69 (2004), pp. 287-315

[Singh1998] Simon Singh (ed.), *Het laatste raadsel van Fermat*, Het verhaal van een stelling die de grootste geesten der aarde 358 jaar lang tot wanhoop dreef (1998),

De Arbeiderspers, pp. 367

[Solovay1977] Robert M. Solovay en Volker Strassen, ‘A fast Monte-Carlo test for primality’, *SIAM Journal on Computing*, Vol. 6, No. 1 (1977), pp. 84-85

[Spencer1983] Joel Spencer, ‘Long proofs’, *The American Mathematical Monthly*, Vol. 90, No. 6 (1983), pp. 365-366

[Steel2000] Graham Steel, Simon Colton, Alan Bundy en Toby Walsh, ‘Cross-domain mathematical concept formation’, *Proceedings of the AISB'00 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science* (2000)

[Steiner1978] Mark Steiner, ‘Mathematical explanation’, *Philosophical Studies*, Vol. 34 (1978), pp. 135-151, in [Jacquette2001], pp. 30-39

[Swart1980] Edward R. Swart, ‘The philosophical implications of the four-color theorem’, *The American Mathematical Monthly*, Vol. 87, No. 9 (1980), pp. 697-707

[Szpiro2003] George Szpiro, ‘Does the proof stack up?’, *Nature*, Vol. 424 (2003), pp. 12-13

[Szpiro2003b] George Szpiro, *Kepler's conjecture*, How some of the greatest minds in history helped solve one of the oldest math problems in the world (2003), Wiley, pp. 272

[Tao2007] Terence Tao, ‘What is good mathematics?’, *Bulletin of the American Mathematical Society*, Vol. 44 (2007)

[Teller1980] Paul Teller, ‘Computer proof’, *The Journal of Philosophy*, Vol. 77, No. 12 (1980), pp. 797-803

[Thiele2002] Ruediger Thiele en Larry Wos, ‘Hilbert's twenty-fourth problem’, *Journal of Automated Reasoning*, Vol. 29, No. 1 (2002), pp. 67-89

[Thomas1998] Robin Thomas, ‘An update on the four-color theorem’, *Notices of the American Mathematical Society*, Vol. 45, No. 7 (1998), pp. 848-859

[Thurston1994] William P. Thurston, ‘On proof and progress in mathematics’, *Bulletin of the American Mathematical Society*, Vol. 30, No. 2 (1994), pp. 161-177, in [Tymoczko1998], pp. 337-356

[Turing1936] Alan M. Turing, ‘On computable numbers, with an application to the Entscheidungsproblem’, *Proceedings of the London Mathematical Society, series 2*, Vol. 42 (1936), pp. 230-265

[Tymoczko1979] Thomas Tymoczko, ‘The four-color problem and its philosophical significance’, *The Journal of Philosophy*, Vol. 76, No. 2 (1979), pp. 57-83, in [Tymoczko1998], pp. 243-266

[Tymoczko1980] Thomas Tymoczko, ‘Computers, proofs and mathematicians: a philosophical investigation of the four-color proof’, *Mathematics Magazine*, Vol. 53,

No. 3 (1980), pp. 131-138

[Tymoczko1981] Thomas Tymoczko, 'Computer use to computer proof: a rational reconstruction', *The Two-Year College Mathematics Journal*, Vol. 12, No. 2 (1981), pp. 120-125

[Tymoczko1998] Thomas Tymoczko (ed.), *New Directions in the Philosophy of Mathematics* (1998), Princeton University Press, pp. 448

[VanBendegem1988] Jean-Paul Van Bendegem, 'Non-formal properties of real mathematical proofs', *Proceedings of the 1988 Biennial Meeting of the Philosophy of Science Association* (1988), pp. 249-254

[VanBendegem2003] Jean-Paul Van Bendegem, 'Thought experiments in mathematics: anything but proof', *Philosophica*, Vol. 72 (2003), pp. 9-33

[VanDenEssen2006] Arno van den Essen, *Magische vierkanten*, Van Lo-Shu tot sudoku. De wonderbaarlijke geschiedenis van wiskundige puzzels (2006), Veen Magazines, pp. 238

[VanFraassen1980] Bas van Fraassen, *The Scientific Image* (1980), Clarendon Press, Oxford

[Veroff2000] Robert Veroff, *Reasoning at multiple levels of abstraction*, Technical Report TR-CS-2000-50 (2000), http://www.cs.unm.edu/~treport/tr/00-10/multiplelevels.ps.gz, Department of Computer Science, University of New Mexico

[Veroff2001] Robert Veroff, 'Solving open questions and other challenge problems using proof sketches', *Journal of automated reasoning*, Vol. 27, No. 2 (2001), pp. 157-174

[Veroff2002] Robert Veroff, *Solution to a Challenge Problem in HBCK*, Tech. Report TR-CS-2002-32, Computer Science Department, University of New Mexico (2002)

[Walsh1992] Toby Walsh, Alex Nunes en Alan Bundy, 'The use of proof plans to sum series', *Proceedings of the 11th International Conference on Automated Deduction* (1992), pp. 325-339

[Wang1960] Hao Wang, 'Toward mechanical mathematics', *IBM Journal of Research and Development*, Vol. 4, No. 1 (1960), pp. 2-22

[Wang1960b] Hao Wang, 'Proving theorems by pattern recognition I', *Communications of the ACM*, Vol. 3, No. 4 (1960), pp. 220-234

[Wang1998] Hao Wang, 'Theory and practice in mathematics', in [Tymoczko1998], pp. 129-152

[Weber2002] Erik Weber en Liza Verhoeven, 'Explanatory proofs in mathematics', *Logi-*

que & Analyse, Vol. 179-180 (2002), pp. 299-307

[Wiedijk2003] Freek Wiedijk, 'Formal proof sketches', Wan Fokkink, Jaco van de Pol, *7th Dutch Proof Tools Day* (2003)

[Wiedijk2004] Freek Wiedijk, 'Formal proof sketches', Stefano Berardi, Mario Coppo, Ferruccio Damiani, *Lecture Notes in Computer Science*, Types for Proofs and Programs: Third International Workshop, Vol. 3085 (2004), pp. 378-393

[Wiedijk2006] Freek Wiedijk, 'On the usefulness of formal methods', *Nieuwsbrief van de NVTI* (2006), pp. 14-23

[Womack2003] Catherine Womack en Martin Farach, 'Randomization, persuasiveness and rigor in proofs', *Synthese*, Vol. 134 (2003), pp. 71-84

[Wos2001] Larry Wos, 'A milestone reached and a secret revealed', *Journal of Automated Reasoning*, Vol. 27, No. 2 (2001), pp. 89-95

[Wos2002] Larry Wos en Branden Fitelson, 'The automation of sound reasoning and successful proof finding', in Dale Jacquette (ed.), *Blackwell Companion to Philosophical Logic* (2002), Blackwell Publishers

[Wos2003] Larry Wos, Dolph Ulrich en Branden Fitelson, 'XCB, the last of the shortest single axioms for the classical equivalential calculus', *Bulletin of the Section Logic*, Vol. 32, No. 3 (2003), pp. 129-134

[Zeilberger1993] Doron Zeilberger, 'Theorems for a prize: Tomorrow's semi-rigorous mathematical culture', *Notices of the American Mathematical Society*, Vol. 40 (1993), pp. 978-981

[Zwick2002] Uri Zwick, 'Computer assisted proof of optimal approximability results', *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms* (2002), pp. 496-505