02-02-2019

Universiteit Utrecht

Killersudoku's oplossen met Prolog

EEN ZOEKTOCHT NAAR DE OVEREENKOMST
TUSSEN DE MENS EN EEN PROGRAMMA

Door: Steven van Blijderveen (5553083)
Begeleider: Tomas Klos
Tweede beoordelaar: Bjørn Jespersen
7,5 ECT

Inhoudsopgave

Samenvatting

Om een toevoeging te doen aan de zoektocht naar kunstmatige intelligentie wordt in deze scriptie geprobeerd een Prolog programma te maken dat Killersudoku's op een menselijke manier oplost. Prolog wordt gebruikt omdat dit een declaratieve taal is, in tegenstelling tot eerdere andere onderzoeken waarbij imperatieve talen werden gebruikt. Ook maakt Prolog gebruik van backwards chaining, waarvan bewezen is dat mensen dit ook gebruiken bij het oplossen van logische problemen. Om antwoord te kunnen geven op de onderzoeksvraag moet worden onderzocht wat precies die menselijke manier is. Dit gebeurt aan de hand van menselijke data: proefpersonen worden gevraagd twee Killersudoku's, een makkelijke en een moeilijke, op te lossen en dit te filmen terwijl ze hardop denken. Aan de hand van deze data kan worden bekeken of mensen deze puzzels überhaupt op dezelfde manier oplossen en vervolgens of deze manier dan hetzelfde is als het gemaakte programma. Bij de makkelijke Killersudoku blijkt er geen duidelijke overeenkomst te zijn in de manier van oplossen tussen verschillende proefpersonen. Het programma vertoont ook geen significante overeenkomst met de data. Bij de moeilijke Killersudoku is er reden om aan te nemen dat er wel een overeenkomst is tussen de menselijke data. Het programma komt bij deze Killersudoku niet tot een oplossing waardoor de data maar deels kan worden vergeleken, deze data komt echter goed overeen met de data van de proefpersonen.

Hoofdstuk 1: Inleiding

In de afgelopen decennia is de mens steeds meer opzoek naar kunstmatige intelligentie: het intelligent laten lijken van een levenloos voorwerp. Volgens het boek *Artificial Intelligence, a Modern Approach*[1] zijn er vier manieren om kunstmatige intelligentie te bereiken. In deze scriptie wil ik een poging doen één van deze vier manieren, namelijk 'thinking humanly', te benaderen, om op die manier iets toe te voegen aan de zoektocht. Met 'thinking humanly' of 'menselijk denken' wordt bedoeld dat een programma een gegeven probleem op een menselijke manier oplost. Dit wil ik doen aan de hand van de Killersudoku. Normale Sudoku's worden regelmatig gebruikt om intelligente algoritmes te testen[2][3], aangezien dit een NP-moeilijk probleem is.[4] Dit betekent dat er (nog) niet één manier gevonden is om elke willekeurige Sudoku binnen afzienbare tijd op te lossen. Ik maak gebruik van de Killersudoku in plaats van een gewone Sudoku omdat hierbij nog een extra vaardigheid komt kijken: rekenen. Door het toevoegen van rekenen aan de al logische Sudoku puzzel hoop ik te bereiken dat mensen op een nog logischere en meer gestructureerde manier de puzzel oplossen. Aangezien Killersudoku's pas de laatste jaren langzaam hun opkomst maken in buitenlandse kranten[5], is er nog weinig onderzoek naar gedaan. Zo is één van de weinige intelligente algoritmes die hierop is losgelaten een genetisch algoritme van Haynes en Corns.[6] Naar het op een menselijke manier oplossen van gewone Sudoku's is al wel eerder onderzoek gedaan.[7] Dit onderzoek gebruikt echter een imperatieve manier van programmeren. Dat wil zeggen dat ze bepaalde menselijke strategieën omzetten in programmeertaal en vervolgens het programma deze strategieën laten uitvoeren op een Sudoku. In deze scriptie wordt onderzocht of het ook op een andere manier mogelijk is: declaratief programmeren. Bij declaratief programmeren geeft de programmeur geen directe oplossingsstrategieën aan het programma, maar een set van regels waar het programma aan moet voldoen. Vervolgens vindt het programma aan de hand van die regels een oplossing voor het probleem. In deze scriptie zal gebruik gemaakt worden van de declaratieve taal Prolog. Hiervoor is gekozen omdat Prolog een taal is die gebruik maakt van *backward chaining*. Dat houdt simpel gezegd in dat Prolog vanaf het doel terugwerkt naar de vraag. Deze manier van programmeren is uitgevonden door mensen te observeren[8], wat het dus geschikt maakt om menselijk gedrag na te bootsen.

Om te onderzoeken of het programma een Killersudoku daadwerkelijk op een menselijke manier oplost zal er zowel naar menselijke data als naar data gegenereerd door het programma gekeken moeten worden. Dit is als volgt gebeurd: een aantal proefpersonen heeft een Killersudoku opgelost die het programma ook kan oplossen. Deze data is vergeleken met de data die het programma genereert bij het oplossen van dezelfde Killersudoku. Zowel de volgorde van het invullen van getallen als de volgorde van gebruikte strategieën blijkt per persoon te verschillen en daarom niet te vergelijken met het programma. Ook is er door een aantal proefpersonen een Killersudoku opgelost die het programma niet kan oplossen. Dit is gebeurd om te zien of mensen tot hetzelfde knelpunt komen als het programma, of dat ze op een totaal andere volgorde of met andere strategieën de puzzel oplosten. Daarnaast is de moeilijke puzzel ook gebruikt om de hypothese te testen dat moeilijkere puzzels leiden tot meer dezelfde oplossingsstrategieën en volgordes van toewijzingen. Dit zou zo kunnen zijn omdat er bij moeilijkere puzzels minder opties zijn om getallen in te vullen of te elimineren uit een cel, wat kan leiden tot meer vaste patronen in de puzzel. Uiteindelijk waren door de moeilijkheid van de puzzel helaas maar drie proefpersonen in staat de puzzel op te lossen. Hierbij bleek wel dat de overeenkomst tussen deze drie proefpersonen bij de moeilijkere puzzel groter was dan de overeenkomst tussen proefpersonen bij de makkelijke puzzel. Daarnaast kwamen twee proefpersonen tot hetzelfde knelpunt als het programma, waardoor te zien is wat ervoor zorgt dat het programma vastloopt.

In de volgende hoofdstukken zal er antwoord worden gegeven op de vraag: “Kan een Killersudoku op een menselijke manier worden opgelost in een declaratieve programmeertaal zoals Prolog?”. Om goed antwoord te kunnen geven op deze vraag zal in hoofdstuk 2 een uitleg volgen van Killersudoku’s en in hoofdstuk 3 een uitleg van Prolog. Vervolgens zal in hoofdstuk 4 het onderzoek verder worden uitgewerkt. Tot slot worden deze resultaten besproken in de conclusie en discussie in hoofdstuk 5.

Hoofdstuk 2: Killersudoku's

Wat is een Killersudoku?

Om het onderzoek goed te kunnen begrijpen is eerst wat basiskennis van de Killersudoku nodig. Een Killersudoku is één van de vele variaties op de populaire Sudoku puzzels. Op de Killersudoku zelf zijn ook weer variaties mogelijk, de volgende beschrijving is echter van de standaard versie van een Killersudoku die tijdens de rest van dit paper ook gebruikt zal worden.
Een Killersudoku, zoals te zien in afbeelding 1, bestaat uit een raster van 9 bij 9 vierkantjes, hierna *cellen* genoemd, die zich bevinden in 9 grotere vierkanten van 3 bij 3 cellen, hierna *nonets* genoemd. De cellen zijn ingedeeld in *kooien*: groepjes aangrenzende cellen van verschillende groottes en vormen. In de cel het meest linksboven staat een getal dat de totale waarde van de kooi aangeeft. Een Killersudoku heeft altijd één mogelijke oplossing.
Het doel van een Killersudoku is het invullen van alle cellen met de getallen 1 tot en met 9 terwijl de volgende regels in acht worden genomen:
- De getallen 1 tot en met 9 moeten allemaal exact 1 keer voorkomen in elke rij, elke kolom en elke nonet.
- De getallen binnen een kooi moeten allemaal verschillend zijn en optellen tot de totale waarde van de kooi, die linksboven in de kooi staat aangegeven.

Oplossingsstrategieën

Voor het invullen van een Killersudoku zijn veel verschillende oplossingsstrategieën. Aangezien binnen een kooi gebruik wordt gemaakt van het optellen van getallen, zijn dit naast de strategieën voor een standaard Sudoku ook strategieën die gebruik maken van rekenen. Het is bij veel van de strategieën van belang om in cellen bij te houden welke mogelijkheden zij nog hebben, aangezien het vaak niet in één keer mogelijk is om een getal in een cel in te vullen. In de volgende sectie zullen een aantal basistechnieken worden besproken die gebruikt kunnen worden om een Killersudoku op te lossen.
Een techniek die men in het begin van een Killersudoku kan toepassen is gebruik maken van de beperkte mogelijkheden van een kooi. Sommige kooien kunnen door hun grootte en de som van de getallen maar een beperkt aantal mogelijkheden hebben. Zie bijvoorbeeld de kooi bestaande uit de cellen B1 en B2 in afbeelding 1. De kooi heeft een grootte van 2 cellen en een som van 16, dit betekent dat in deze kooi enkel de getallen 7 en 9 passen. Twee keer het getal 8 invullen is geen optie, aangezien een kooi moet bestaan uit verschillende getallen. Er zijn dus maar twee mogelijkheden om deze kooi te vullen. Namelijk een 7 in cel B1 en een 9 in cel B2, of andersom.
Een andere techniek die ook in het begin veel gebruikt kan worden is het zoeken naar nonets, rijen of kolommen die bestaan uit volledige kooien, op één cel na. Deze techniek wordt ook wel het zoeken naar 'innies' genoemd. Zie bijvoorbeeld de nonet helemaal linksboven in afbeelding 1, waarbij cel C3 de enige binnen de nonet is die niet bij een volledige kooi binnen de nonet hoort. Door het verschil van de som van de getallen 1 tot en met 9 (45) en de som van de totale waardes van de overige kooien binnen het blok (9 + 16 + 6 + 6 = 37) te nemen komt men erachter dat in cel C3 het getal 8 (45 – 37) hoort. Dezelfde techniek kan ook tegenovergesteld gebruikt worden. In dit geval noemt men het zoeken naar 'outties'. Als er precies één cel is van alle kooien in een rij, kolom of nonet die niet binnen die rij, die kolom of die nonet ligt, kan men het verschil van de som van alle waardes linksboven in de kooien en 45 nemen.
Deze technieken kunnen beide ook gebruikt worden met meerdere rijen, kolommen of nonets tegelijk. Zie bijvoorbeeld kolom A en B. Alle kooien zitten volledig binnen deze twee kolommen op één cel na, namelijk C6. Dit is dus een outtie. Door alle kooien bij elkaar op te tellen (9 + 16 + 6 + 13 + 7 + 14 + 17 + 15 = 97) en het verschil met tweemaal (want het zijn twee kolommen) alle mogelijke

waardes (90) te nemen komen we tot de conclusie dat cel C6 het getal 7 bevat. Tot slot kunnen deze technieken ook gebruikt worden als twee of drie cellen binnen of buiten een kolom, rij of nonet vallen. In dit geval kunnen ze meestal niet meteen ingevuld worden, maar blijven er wel minder opties over voor deze cellen.
Naast deze twee typische Killersudoku strategieën zijn ook de normale Sudoku strategieën van toepassing met kleine uitbereidingen. Dit betekent dat men rekening houdt met het feit dat een getal maar één keer per kolom, rij, nonet en kooi mag voorkomen. In de praktijk kan deze regel als volgt worden toegepast: Men weet al dat de getallen 7 en 9 in de tweede kolom staan, namelijk in de cellen B1 en B2. Dit betekent dat de getallen 7 en 9 niet meer op een andere plek in de tweede kolom kunnen staan, bijvoorbeeld in cel B9. Hierdoor weet men ook weer dat in cel A9 geen 6 of 8 kan staan, aangezien dan de som die hoort bij de kooi met cellen A9 en B9 niet meer tot 15 kan leiden.

	A	B	C	D	E	F	G	H	I
1	9	16	6	6		18		11	
2				14	25		7		27
3	6								
4	13	7	6	11		15			28
5					8		12		
6	14	17		14		8			6
7			12		13		13		
8				21				13	
9	15						14		

Afbeelding 1: Een voorbeeld van een Killersudoku

Hoofdstuk 3: Prolog

Nu het duidelijk is wat een Killersudoku is en hoe deze opgelost kan worden is het tijd om te bekijken hoe deze strategieën geprogrammeerd kunnen worden. Om dit goed te begrijpen is eerst wat basiskennis van Prolog nodig.

Hoe werkt Prolog?

De taal Prolog bestaat uit drie soorten *clauses*.[9] Ten eerste zijn er *feiten*: dit zijn clauses die altijd en zonder verdere voorwaarden waar zijn. Ten tweede zijn er *regels*: clauses die waar zijn afhankelijk van één of meerdere voorwaarden. Tot slot zijn er vragen of *queries*. Een query is een manier waarop de gebruiker aan Prolog kan vragen of iets waar is. Van feiten en regels volgt later een voorbeeld. Een voorbeeld van een query voor de 4x4 Killersudoku in afbeelding 2 ziet er zo uit:

```
solution(
[kooi([A1,A2],6),
kooi([B1],1),
kooi([C1,C2],5),
kooi([D1],4),
kooi([B2,B3],5),
kooi([D2],1),
kooi([A3,A4],4),
kooi([C3],4),
kooi([D3,D4],5),
kooi([B4,C4],5)]
).
```

In deze query zien we het predicaat *solution*. Dit predicaat slaagt als het programma (een deel van) de oplossing van een Killersudoku kan vinden. Dit predicaat heeft als argument een lijst met daarin allerlei kooien die de Killersudoku beschrijven. De kooien hebben allemaal een bepaalde structuur: eerst een lijst met daarin de cellen die de kooi bevat, gevolgd door een getal wat de totale som van de kooi aangeeft. De cellen zijn met hoofdletters geschreven aangezien alles wat met een hoofdletter begint in Prolog een variabele is. Aan variabelen kan Prolog een waarde toewijzen, dit wordt unificatie genoemd. Unificatie is de manier waarop Prolog queries beantwoordt: het probeert de variabelen die gegeven worden in een query te unificeren met waarden aan de hand van de gegeven feiten en regels. Uit bovenstaande query komt dit antwoord van Prolog:

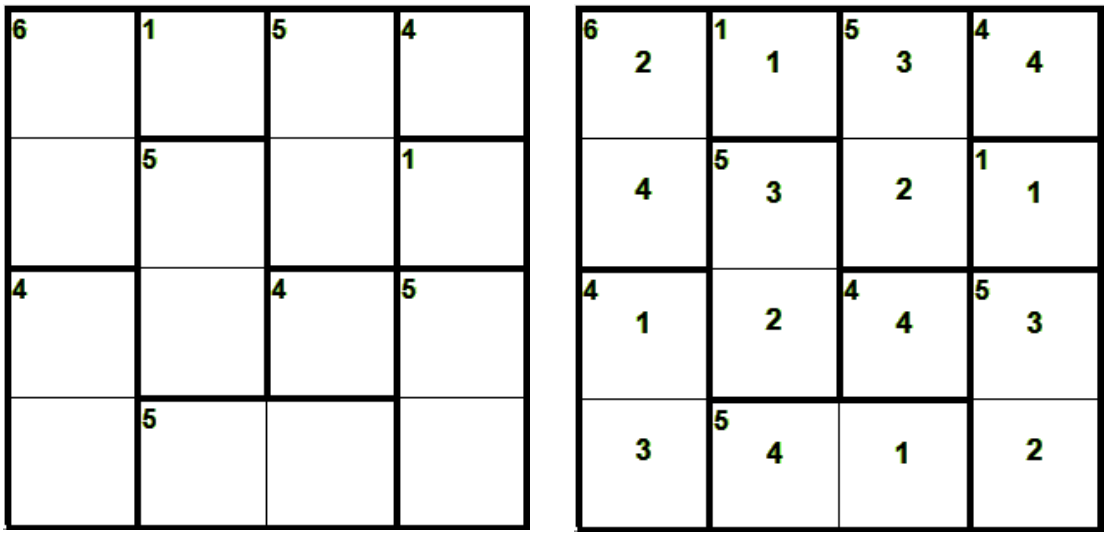
```
A1 = C2, C2 = B3, B3 = D4, D4 = 2,

B1 = D2, D2 = A3, A3 = C4, C4 = 1,

C1 = B2, B2 = D3, D3 = A4, A4 = 3,

D1 = A2, A2 = C3, C3 = B4, B4 = 4.
```

Te zien is dat het antwoord overeenkomt met de ingevulde Killersudoku in afbeelding 2. Maar hoe komt Prolog tot dit antwoord?
Dat werkt als volgt: binnen het predicaat solution staan verschillende voorwaarden waar aan voldaan moet worden. Als aan alle voorwaarden voldaan wordt is het predicaat geslaagd en laat Prolog alle unificaties zien die het programma uitgevoerd heeft. Om een beter begrip van Prolog te krijgen zal ik één van deze voorwaarden iets verder uitwerken: de voorwaarde die ‘zoekt’ naar een innie in de kolommen.

```
1     vindInnieKolom([],_).
2     vindInnieKolom([kolom(L)|T],Kooien) :-
3           inniecheck(L,L,Kooien,0,Innie),
4           innieToewijzing(L,Kooien,Innie,1),
5           vindInnieKolom(T,Kooien), !.
6     vindInnieKolom([_|T],Kooien) :-
7           vindInnieKolom(T,Kooien), !.
```

Dit predicaat vindInnieKolom bevat twee argumenten, gescheiden door een komma. Het predicaat slaagt wanneer alle innies in enkele kolommen gevonden zijn en wanneer de waarde die in die innies hoort geünificeerd is met de variabelen die de innies representeren. Het eerste argument is een lijst met alle kolommen, het tweede argument een lijst met alle kooien. Wat opvalt is dat het predicaat gedefinieerd wordt door drie verschillende gevallen. In regel 1 staat een geval van vindInnieKolom dat in diezelfde regel nog wordt afgesloten door een punt. In regel 2 staat opnieuw een geval van het predicaat vindInnieKolom, die pas in regel 5 wordt afgesloten door een punt. Tot slot wordt het geval dat begint in regel 6 afgesloten in regel 7. Prolog probeert door unificatie één van de gevallen te laten slagen. Dit doet het altijd op een vaste volgorde, namelijk van boven naar beneden. Als het eerste geval niet slaagt kijkt Prolog pas verder naar het volgende geval.
De eerste regel is een voorbeeld van een feit. Dit geval slaagt hoe dan ook zolang het eerste argument een lege lijst is, aangegeven door de vierkante blokhaken. Als er geen kolommen meer zijn om innies in te vinden, zijn alle innies gevonden en is het predicaat geslaagd. Aangezien het predicaat in eerste instantie gecontroleerd wordt met een lijst van kolommen in het eerste argument slaan we deze regel voor nu nog even over. In regel 2 is het eerste argument van vindInnieKolom niet leeg, er staat namelijk `[kolom(L)|T]`. Dit geeft aan dat we de eerste kolom L noemen en de rest van de kolommen T. Vervolgens zien we in regel 3 dat er voldaan moet worden aan de voorwaarde *inniecheck*. De voorwaarde inniecheck slaagt als kolom L daadwerkelijk een innie bevat. Als inniecheck slaagt gaan we door naar regel 4 waar moet worden voldaan aan de voorwaarde *innieToewijzing*, die slaagt als unificatie plaatsgevonden heeft in Prolog waardoor de waarde die de innie moet krijgen ook daadwerkelijk is toegewezen aan de variabele. Als innieToewijzing slaagt komt er in regel 5 een zogenaamde recursieve aanroep. Er wordt opnieuw bekeken of het predicaat vindInnieKolom slaagt, maar nu met de lijst T in het eerste argument. Dit zijn dus alle kolommen zonder de kolom die zojuist bekeken is. Natuurlijk kan het ook zo zijn dat inniecheck niet slaagt, aangezien er geen innie in een bepaalde kolom zit. In dat geval komen we uit bij regel 6, waar direct de recursieve aanroep met de lijst T gedaan wordt en de huidige kolom dus wordt overgeslagen. Als dit een aantal keer gedaan is loopt de lijst met kolommen langzaam leeg, waardoor we vanzelf uitkomen bij het geval met een lege lijst in het eerste argument. Dit geval wordt ook wel het basisgeval genoemd. Het basisgeval zorgt ervoor dat een methode uit zijn recursie kan komen. Het basisgeval in deze methode slaagt altijd, waardoor de laatste aanroep van vindInnieKolom dus ook slaagt, waardoor de aanroep daarvoor ook slaagt, etc. Op deze manier slaagt de methode vindInnieKolom dus altijd en unificeert het waar mogelijk innies met de bijbehorende waardes.
Een belangrijk deel van de code die nog niet besproken is, is het gebruik van de uitroeptekens in regel 5 en 7. Deze uitroeptekens worden ‘cut-operatoren’ genoemd en zijn er om backtracking tegen te gaan. Prolog maakt gebruik van *Depth First Search*, wat inhoudt dat het unificaties doet tot het bij

een oplossing komt. Als er een conflict is gaat Prolog terug naar het laatste keuzemoment waar nog een alternatieve unificatie mogelijk is, en unificeert het deze mogelijkheid. Dit terug gaan naar een vorig keuzemoment heet *backtracking*. In dit geval zou een voorwaarde in het predicaat solution die na de voorwaarde vindInnieKolom kunnen falen, waardoor het programma teruggaat naar vindInnieKolom en daar bijvoorbeeld niet meer voor het geval in regel 2 kiest, maar voor het geval in regel 6. Dit kan alleen maar leiden tot een slechtere manier van oplossen, aangezien dan de unificatie van een eerder gevonden innie teruggedraaid wordt en de kolom waar die innie in staat niet aan de voorwaarde inniecheck en innietoewijzing hoeft te voldoen. Voor die kolom wordt direct voor het geval in regel 6 gekozen, waardoor de innie niet meer wordt gevonden. Daarom staan er aan het eind van de twee gevallen in regel 5 en 7 uitroeptekens, die ervoor zorgen dat deze keuze niet opnieuw gemaakt kan worden.



Afbeelding 2: Een voorbeeld van een 4x4 Killersudoku en de ingevulde versie

Hoofdstuk 4: Het onderzoek

Relevantie

Uit de uitleg over Killersudoku's blijkt dat dit een logische puzzel is: de puzzel is op te lossen door na te denken, niet door te gokken. De manier om deze puzzel op te lossen is simpelweg het gebruiken van verschillende technieken en deze op de juiste plaats in de puzzel en op het juiste moment in te zetten. Op deze manier kan Prolog ook worden gebruikt. Hoewel Prolog bij het gebruik van Depth First Search in combinatie met backtracking juist altijd gokt, het kiest namelijk altijd het eerste antwoord wat het tegenkomt, kan dit gemakkelijk worden omzeild. Door Prolog de juiste voorwaarden bij problemen te geven en door middel van de 'cut-operator' backtracking te voorkomen kan in Prolog zo geprogrammeerd worden dat het programma in één keer tot een goede oplossing komt.
We weten nu dus dat er een programma in Prolog te schrijven valt dat een Killersudoku oplost op een logische en gestructureerde manier, namelijk door het gebruiken van bepaalde technieken op het juiste moment en op de juiste plek. Daarnaast is backward chaining, waar Prolog gebruik van maakt, een manier waarop mensen logische problemen aanpakken zoals in de inleiding is uitgelegd. Maar lost dit programma dan ook werkelijk Killersudoku's op zoals mensen dat in de praktijk ook doen? Hier zijn drie mogelijke antwoorden op. Ten eerste kan dit kloppen, het programma lost inderdaad de puzzel op dezelfde manier op als een mens. Ten tweede kan het niet kloppen, hierin vallen twee gevallen te onderscheiden: Mensen lossen de puzzel significant verschillend op in vergelijking tot het programma, of mensen lossen de puzzel allemaal op verschillende manieren op waardoor er geen vergelijking te trekken is met het programma.
Het eerste antwoord is het meest interessant. Als het programma dit logische probleem op dezelfde manier aanpakt als de mens, houdt dit in dat het programma een stukje van de menselijke probleemoplossing nabootst. Dit nabootsen is iets waar veel wetenschappers in de kunstmatige intelligentie naar op zoek zijn. Vervolgens zou het programma kunnen worden aangepast en uitgebreid naar andere, grotere logische problemen. In dit geval zou het een probleem veel sneller maar toch op dezelfde manier als een mens kunnen oplossen. Dit is iets waar bijvoorbeeld in de wereld van de robotica naar gezocht wordt.
Een groot deel van de wetenschap is het falsificeren van kennis. Dit houdt in dat men mogelijkheden wegstreept. Als het programma de puzzels totaal niet oplost zoals een mens dat zou doen, streept het in ieder geval de mogelijkheid weg dat in Prolog een adequate nabootsing van de menselijke probleemoplossing gemaakt kan worden.
Tot slot kan het ook zijn dat mensen de puzzel allemaal op verschillende manieren oplossen. In dit geval is dit op zichzelf staande feit een interessante ontdekking, namelijk dat mensen zelfs bij een logisch probleem met een duidelijke structuur verschillende mogelijkheden vinden om dit probleem op te lossen.

Het programma Solver.pl

Om te bekijken of het programma inderdaad Killersudoku's op dezelfde manier oplost als een mens is het eerst belangrijk te weten hoe het programma in elkaar zit. Het programma Solver.pl is een Prolog programma dat Killersudoku's kan oplossen. In de volgende paragraaf zal een korte uitleg volgen over het programma, een meer diepgaande uitleg is te vinden in de appendix.

Om het programma te laten starten is, zoals in hoofdstuk 3 uitgelegd, een query nodig. De query voor het programma Solver.pl bevat een aantal specifieke eisen. De query bestaat uit het aanroepen van het predicaat *solution*. Dit predicaat bevat, zowel direct als indirect, alle verdere regels en feiten om de Killersudoku waardes toe te schrijven en er waardes uit te elimineren. Solution slaagt als de volgende argumenten voldoen aan de set van regels en feiten: een lijst met alle kolommen, een lijst met alle rijen, een lijst met alle nonets en een lijst met alle kooien. Binnen deze lijsten staat aangegeven welke cellen zich bevinden in de kolom, rij, nonet of kooi. Bij kooien staat ook de totale waarde van de kooi aangegeven.
Het eerste wat plaats vindt binnen het predicaat solution is een voorwaarde die slaagt als binnen elke kooi per cel een lijst met mogelijke waarden wordt toegevoegd. Deze lijst houdt direct rekening met de minimale en de maximale waarde die in een kooi kan zitten en sluit dubbele cijfers uit bij een kooi die twee cellen bevat. Door deze opties per cel toe te voegen aan de kooien hebben we nu de mogelijkheid cijfers te elimineren uit cellen.
Wat het programma hierna doet is het controleren van een set van voorwaarden en aan de hand daarvan mogelijkheden elimineren uit cellen en getallen toewijzen aan cellen. Deze voorwaarden zijn het zoeken naar innies en outties op verschillende manieren, het verwijderen van mogelijkheden als het getal al in de kolom, rij , nonet of kooi zit en het verwijderen van mogelijkheden als het al zeker is dat een getal ergens anders binnen die kolom, rij, nonet of kooi moet komen te staan. Als deze set met voorwaarden volledig is doorgewerkt print het programma de huidige puzzel (met opties per cel en ingevulde cellen) en gaat het programma opnieuw langs de set van voorwaarden, om te zien of met de nieuwe informatie in de puzzel de voorwaarden weer nieuwe eliminaties of toewijzingen kunnen doen. Na het gebruiken van de set van voorwaarden controleert het programma telkens of de waardes in de Killersudoku verschillen van de waardes voordat de set voorwaarden was gebruikt. Op het moment dat dit niet meer zo is slaagt het programma, aangezien er geen vooruitgang meer wordt geboekt. De grootst mogelijke oplossing, die met de meeste toegewezen variabelen, die het programma kan vinden is op dit moment gevonden. Dit kan dus het punt zijn dat de Killersudoku volledig is ingevuld, maar dit kan ook een punt zijn waarop het programma met de huidige set voorwaarden niet meer verder komt.

Onderzoeksplan

Nu we weten hoe het programma in elkaar zit zijn we weer een stapje dichterbij het antwoord op de vraag: "Kan een Killersudoku op een menselijke manier worden opgelost in een declaratieve programmeertaal zoals Prolog?". Nu is het van belang te onderzoeken wat precies die menselijke manier is. Dit is vanzelfsprekend te onderzoeken door mensen daadwerkelijk Killersudoku's te laten maken. Vervolgens kunnen de resultaten hiervan op verschillende manieren worden vergeleken met de resultaten van het programma: Vulden ze op dezelfde volgorde getallen in? Werden op dezelfde volgorde getallen geëlimineerd? Werden dezelfde strategieën toegepast?
Ten eerste moet dus een aantal mensen de Killersudoku's oplossen. Om bovenstaande vragen te beantwoorden is het van belang dat we weten op welke volgorde ze getallen toevoegen en elimineren en welke strategieën ze daarvoor gebruiken. Om al deze informatie te verkrijgen is het van belang dat de proefpersonen de Killersudoku filmen terwijl ze deze oplossen. Tijdens het filmen moeten ze hardop denken zodat op film te volgen is waarom ze bepaalde getallen invullen of

elimineren. De proefpersonen krijgen allemaal twee Killersudoku's, een makkelijke die het programma ook op kan lossen en een iets moeilijkere waarbij het programma vast komt te zitten. De Killersudoku's zijn te zien in appendix C en D. In *A SAT Attack on Killer Sudoku Problems* [10] wordt beschreven dat een Killersudoku met maximale kooigrootte 4 (zoals de moeilijkere Killersudoku) ook voor mensen moeilijker is dan een Killersudoku met maximale kooigrootte 2 (zoals de makkelijke Killersudoku). Dit komt doordat het aantal mogelijkheden voor een kooi met grootte 2 kleiner is dan voor een kooi met grootte 4. De moeilijkere Killersudoku wordt meegestuurd om verschillende redenen. Het kan bijvoorbeeld zijn dat mensen bij een moeilijkere Killersudoku meer gestructureerd te werk gaan, omdat ze er niet uit komen. Daarnaast is het interessant om te zien of mensen ook tot het knelpunt komen waar het programma vastloopt en hoe ze hier dan verdergaan, of dat ze het begin van de puzzel al totaal anders aanpakken. Vervolgens kan van alle proefpersonen worden bekeken of ze op ongeveer dezelfde manier de Killersudoku's oplossen. Dit kan dus zijn op basis van ingevulde getallen, geëlimineerde getallen of gebruikte strategieën. Als de proefpersonen duidelijke overeenkomsten laten zien in het oplossen van de Killersudoku's, kan dit worden vergeleken met het programma. Mocht er een duidelijk verschil te zien zijn tussen de proefpersonen met ervaring in het maken van Killersudoku's en die zonder ervaring, dan kan het ook nog interessant zijn deze los te vergelijken met het programma. Er kan ook nog onderscheid worden gevonden in het oplossen van de makkelijke Killersudoku tegenover het oplossen van de moeilijkere Killersudoku. Vervolgens kan op basis van de bevindingen een conclusie worden getrokken uit het onderzoek.

Onderzoeksmethode

Aan het onderzoek deden negen proefpersonen mee. De proefpersonen werkten mee met het onderzoek omdat ze positief reageerden op de vraag of ze hier tijd voor en zin in hadden. De proefpersonen kregen zoals in het onderzoeksplan beschreven twee Killersudoku's opgestuurd per mail. De exacte opdracht die in de mail is meegegeven is te lezen in appendix B. De makkelijkere Killersudoku is te zien in appendix C en de moeilijkere in appendix D. De proefpersonen kregen de keuze om één van de twee Killersudoku's te maken, of beide wanneer ze daar voldoende tijd en motivatie voor hadden. Aangeraden werd dat proefpersonen die nog nooit een Killersudoku gemaakt hadden begonnen met de makkelijke en daarna eventueel de moeilijke nog zouden maken. Aan deze proefpersonen werd ook aangeraden voor het maken van de puzzel zelf even op te zoeken op het internet hoe de puzzel precies werkt. Hier werden geen verdere instructies over gegeven, om de manier van oplossen niet te beïnvloeden. Proefpersonen die al vaker Killersudoku's gemaakt hadden werd juist aangeraden te beginnen met de moeilijkere, zodat daar ook voldoende resultaten van zijn. De proefpersonen werd in de mail uitgelegd dat het belangrijk is dat ze tijdens het maken van de Killersudoku's de puzzel filmen en hardop nadenken. Zo kon duidelijk gevolgd worden welke stappen ze zetten wanneer ze een getal opschreven of elimineerden. Tot slot werd aan de proefpersonen gevraagd of ze bij het opsturen van het filmpje wilden laten weten of ze al eerder een Killersudoku gemaakt hadden.

Onderzoeksresultaten

Helaas kregen twee van de negen proefpersonen het niet voor elkaar de makkelijke Killersudoku op te lossen en kon hun data niet gebruikt worden door een onbruikbaar filmpje en een foutieve oplossing. Van de overige zeven proefpersonen losten zes proefpersonen de makkelijke Killersudoku succesvol op. Vijf proefpersonen deden een poging de moeilijke Killersudoku op te lossen, waarvan er drie slaagden. Een overzicht hiervan is te zien in afbeelding 3.
Van alle geslaagde oplossingen is in Appendix E een overzicht te zien van de volgorde van de toegewezen waarden en de strategieën die hierbij gebruikt zijn. Er is voor gekozen niet alle eliminaties weer te geven aangezien dit ruim 600 extra regels toe zou voegen. Aan de toewijzingen valt de volgorde van bekeken cellen al te zien en ook tot op zekere hoogte de gebruikte strategieën. Van het programma is eenzelfde overzicht te zien in Appendix F.
De strategieën zijn kort genoteerd om het overzichtelijk te houden, hier volgt voor de onduidelijke strategieën een uitleg.
Totale waarde kooi: Het toegewezen getal maakt de totale waarde van de kooi compleet. Dit kan zijn doordat alle getallen binnen een kooi al zijn ingevuld of doordat er zekere kennis bestond over de andere getallen binnen de kooi.
Enige positie binnen (…): Dit getal werd op deze plek ingevuld omdat het de enige positie binnen een bepaalde structuur (rij, kolom of nonet) was waar dit getal nog paste.
Enige overige optie: Er was nog één optie beschikbaar voor deze cel. Dit kan door meerdere factoren komen. Bijvoorbeeld dat alle andere getallen al in de rij, kolom, nonet of kooi stonden of dat de minimale/maximale waarde van een kooi niets anders toeliet.
Wat opvalt aan de resultaten is dat een aantal proefpersonen de Killersudoku op wist te lossen zonder gebruik te maken van veel verschillende strategieën. Zo wist proefpersoon 1 de moeilijkere Killersudoku op te lossen puur door het gebruik van minimale en maximale waardes binnen kooien en het wegstrepen van opties aan de hand van die waarden.
De reden dat het programma deze Killersudoku niet kan oplossen terwijl het wel over deze strategieën (en meer) beschikt is als volgt. Wat het programma niet kan is nieuwe kooien maken van kooien die al deels zijn ingevuld of kooien die ontstaan door een (drie)dubbele innie of outtie. Stel dat een kooi grootte 3 en totale waarde 16 heeft en er is al een 8 ingevuld. Wat de proefpersonen in dit geval doen is de twee nog niet ingevulde cellen behandelen als een kooi van 8. Dit doet het programma niet. Hierdoor weet het programma bijvoorbeeld niet dat in de overige twee cellen niet twee keer het getal 4 kan voorkomen. Hetzelfde geldt dus voor een dubbele innie die samen 8 is.

	Ervaren proefpersoon: Makkelijke Killersudoku	Ervaren proefpersoon: Moeilijke Killersudoku	Onervaren proefpersoon: Makkelijke Killersudoku	Onervaren proefpersoon: Moeilijke Killersudoku
Opgelost	2	2	4	1
Niet opgelost	0	0	2	2
Geen poging	1	1	0	3

Afbeelding 3: een overzicht van de proefpersonen

Analyse van de resultaten

Uit de resultaten blijkt dat er bijna geen samenhang te vinden is tussen de oplossingsmethoden van de verschillende proefpersonen bij de makkelijke puzzel. In appendix G is een overzicht van de gemiddelde afwijking per cel te zien. Hoe deze afwijking is berekend valt het best uit te leggen met een voorbeeld. Te zien in de ruwe data in appendix E is dat proefpersoon 1 aan cel G9 als 6^{e} een getal toewees. Proefpersoon 2 wees G9 echter pas als 42^{e} een getal toe. Hiervan kan het verschil genomen worden: 42 – 6 = 36. Vervolgens nemen we ook het verschil tussen proefpersoon 1 en proefpersoon 3, 4, 5 en 6. En het verschil tussen proefpersoon 2 en proefpersoon 3, 4, 5 en 6 etc. Het gemiddelde van al deze getallen laat zien hoe groot de afwijking tussen alle proefpersonen gemiddeld is, waaruit op te maken valt of de getallen op ongeveer hetzelfde moment aan een cel wordt toegewezen. Bij de makkelijke puzzel zijn dit relatief hoge getallen. Het gemiddelde van al deze gemiddeldes, wat dus aangeeft hoeveel de afwijking gemiddeld over de hele puzzel is, is voor de makkelijke puzzel 23,99. Bij de moeilijke puzzels is dezelfde berekening gemaakt. Deze berekening is iets minder betrouwbaar dan die bij makkelijke puzzels, door het kleinere aantal proefpersonen. Desalniettemin is het een opvallend resultaat dat het gemiddelde van alle gemiddeldes hierbij maar 9,52 is. Dit pleit voor de hypothese dat moeilijkere puzzels meer op dezelfde manier gemaakt worden door mensen.
Deze gemiddelde afwijking geeft echter alleen antwoord op een deelvraag: “Lossen mensen Killersudoku’s op gelijkwaardige manieren op?”. Daarom is er ook gekeken naar het verschil tussen het gemiddelde toewijzingsmoment van de proefpersonen per cel en het toewijzingsmoment van het programma per cel. Dit is te zien in appendix H. Het gemiddelde verschil tussen het programma en de proefpersonen is 14,46 bij de makkelijke puzzel. Wat dit aangeeft is dat er gemiddeld 13 stappen zitten tussen het toewijzen van een getal aan een cel door het programma en het toewijzen van een getal aan een cel door de proefpersonen. Doordat het programma van de moeilijkere Killersudoku maar een aantal getallen invult is deze data veel minder relevant. Toch is het opvallend om te zien dat het gemiddelde verschil hier 4,19 is. Dit betekent dat de proefpersonen de Killersudoku op een vergelijkbare manier starten als het programma. Twee proefpersonen werkten zelfs tot exact hetzelfde knelpunt als het programma. Hierna vullen ze echter beide verschillende cellen in, allebei gebruik makend van ‘nieuwe kooien’: het construeren van nieuwe kooien door zeker te weten dat, bijvoorbeeld bij proefpersoon 3, C1 en C2 een 6 en een 7 bevatten hoewel dit geen kooi is.
Wat verder uit het filmmateriaal blijkt is dat de proefpersonen vaak graag een bepaald gebied uitwerken, maar niet altijd in dezelfde volgorde. Zo kan eerst een tijd in de bovenste 3 nonets gewerkt worden, waarna hier niks meer te vinden is en men doorgaat naar bijvoorbeeld de linker 3 nonets of de 3 nonets onder de bovenste 3. In dit zoekgebied is geen direct vast patroon te vinden, behalve dat een genoemde reden vaak, maar niet altijd, is dat er veel getallen in het gebied staan.
Wat ook een reden is dat het programma niet te vergelijken is met de menselijke data is dat mensen vaak dingen over het hoofd zien. Zo was er in veel gevallen een structuur te zien van het invullen van één getal door eliminatie, waarna het andere getal binnen de kooi ingevuld kon worden doordat men de totale waarde van de kooi wist. Zoals bijna elk proefpersoon bij de makkelijke Killersudoku begint met het invullen van een 9 in cel D1 door eliminatie, waarna een 8 is in te vullen in cel C1 door de totale waarde van de kooi min 9 te nemen. Echter zijn er meerdere voorbeelden waar men na deze structuur lange tijd te volgen plotseling vergeet dit triviale tweede getal in te vullen.

Hoofdstuk 5: Conclusie en discussie

Conclusie

Wat duidelijk blijkt uit de analyse van de resultaten is dat mensen zelfs deze logische puzzels die een redelijke structuur lijken te hebben op veel verschillende manieren oplossen, ook al zijn de verschillen bij moeilijkere puzzels kleiner. Het antwoord op de vraag “Kan een Killersudoku op een menselijke manier worden opgelost in een declaratieve programmeertaal zoals Prolog?” kan in dit geval dus met nee worden beantwoord. Hier zitten echter wel wat haken en ogen aan, die in de discussie besproken worden. Maar aangezien in een declaratieve taal zoals Prolog altijd een vast systeem zal zitten die een aantal regels in een vooraf bepaalde volgorde volgt om de puzzel op te lossen kan er geen perfecte imitatie worden gemaakt van menselijk oplossingsgedrag. Mensen blijken te vaak ogenschijnlijk willekeurige beslissingen te maken, zoals in welk gebied ze op zoek gaan naar mogelijke eliminaties of toewijzingen en zoals het loslaten van de structuur die ze eerder wel hadden.

Discussie

De conclusie dat mensen de puzzel op verschillende manieren oplossen is echter niet volledig waterdicht. Om dit te onderzoeken is namelijk veel meer data nodig dan de data gebruikt in dit onderzoek, wat in de korte tijd en met de beperkte capaciteit nu niet mogelijk was. Dat er bij zo weinig proefpersonen meteen grote afwijkingen te zien zijn geeft wel de mogelijkheid een redelijk sterke hypothese op te stellen dat mensen de puzzel verschillend maken. Een andere optie tot vervolgonderzoek is het gebruiken van nog moeilijkere puzzels. De hypothese dat moeilijkere puzzels leiden tot meer gelijke oplossingen is nog niet onderuit gehaald, er is zelfs enige overeenkomst te ontdekken in de drie oplossingen van de moeilijke puzzel uit dit onderzoek. Deze overeenkomst is echter geen bewijs voor de stelling dat moeilijkere puzzels op een meer gelijke manier opgelost worden, aangezien dit ook toeval kan zijn. Ik raad toekomstige onderzoekers hierbij wel aan dat ze alleen ervaren proefpersonen zoeken bij moeilijkere puzzels, aangezien maar één van de drie proefpersonen zonder ervaring de moeilijke puzzel daadwerkelijk heeft opgelost.
Naast het gebruik van meer proefpersonen kan ook het veranderen van het programma invloed hebben op de uitkomst van de onderzoeksvraag. Zo zou een andere programmeur mogelijk een Prolog programma kunnen maken dat dichterbij de menselijke manier van het oplossen van Killersudoku’s komt. Daarnaast zijn er meer declaratieve talen dan Prolog, waardoor er nog meer andere mogelijkheden tot programma’s zijn. Prolog lijkt echter door het gebruik van backward chaining wel de ideale kandidaat om menselijk gedrag na te bootsen.

Literatuurlijst

[1] Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. (3e editie) Malaysia; Pearson Education Limited,.

[2] Lynce, I., & Ouaknine, J. (2006). Sudoku as a SAT Problem. In *ISAIM*.

[3] Geem, Z. W. (2007). Harmony search algorithm for solving sudoku. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 371-378). Springer, Berlin, Heidelberg.

[4] Yato, T., & Seta, T. (2003). Complexity and completeness of finding another solution and its application to puzzles. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, *86*(5), 1052-1060.

[5] M. Rizwan (2008). Solving Killer Sudoku. Geraadpleegd op 2 februari 2019, van http://carriehall.co.uk/mohammedrizwan/res/Killer_Sudoku_SETA.pdf

[6] Haynes, D., & Corns, S. (2013). EA-EMA Optimization Applied to Killer Sudoku Puzzles. *Procedia Computer Science*, *20*, 58-64.

[7] Pillay, N. (2012). Finding solutions to Sudoku puzzles using human intuitive heuristics. *South African Computer Journal*, *49*(1), 25-34.

[8] Langley, P. (2006). Intelligent behavior in humans and machines. In *American Association for Artificial Intelligence*.

[9] Bratko, I. (2001). *Prolog programming for artificial intelligence*. (3e editie) Pearson education.

[10] S. Wang, A. Venkatesh (2016). A SAT Attack on Killer Sudoku Problems. Geraadpleegd op 2 februari 2019, van https://uva-kr16.github.io/KilerSudoku/report.pdf

Appendix A: Uitgebreide uitleg Solver.pl

Query

Het programma vereist een redelijk specifieke query waarin alle gegevens over de Killersudoku door worden gegeven aan het programma. Deze query ziet er als volgt uit:

```
solution([rij([A1,B1,C1,D1,E1,F1,G1,H1,I1]),rij([A2,B2,C2,D2,E2,F2,G2
,H2,I2]),…],[kolom([A1,A2,A3,A4,A5,A6,A7,A8,A9]),kolom([B1,B2,B3,B4,B
5,B6,B7,B8,B9]),…],[nonet([A1,B1,C1,A2,B2,C2,A3,B3,C3]),nonet([D1,E1,
F1,D2,E2,F2,D3,E3,F3]),…],[kooi([A1,B1],3),kooi([C1,D1],17),kooi([E1,
F1],11),kooi([G1,H1],8),…]).
```

Ten eerste is er de aanroep van het predicaat *solution(…)*. Hierin staan alle voorwaarden waaraan de Killersudoku moet voldoen. Als dit predicaat slaagt laat Prolog alle getallen zien de geünificeerd zijn met de variabele cellen.
Vervolgens is er een lijst (te zien aan de vierkante haak), die bestaat uit structuren die het gemakkelijk maken om een rij te weergeven. Elke rij bevat een lijst met 9 variabelen, die namen krijgen corresponderend met de cellen binnen deze rij. Let op dat de drie puntjes aan het einde van de lijst niet werkelijk in de code staan, maar ervoor zijn om het overzichtelijk te houden. In werkelijkheid staat in het eerste argument van *solution* een lijst met daarin 9 keer een rij met bijbehorende variabelen. Na de lijst van rijen is er eenzelfde soort lijst met 9 kolommen met bijbehorende variabelen en vervolgens nog een lijst met 9 nonets met bijbehorende variabelen. Tot slot is er een lijst met alle kooien. Een kooi heeft de volgende structuur: Ten eerste een lijst met alle variabelen die zich in de kooi bevinden, en vervolgens een getal wat de som van de getallen binnen de kooi aangeeft. Al deze informatie is direct te zien wanneer men een Killersudoku begint en kan daarom dus ook direct worden ingevoerd in het programma.

Beginfase

Wanneer Prolog begint aan bovenstaande query is de eerste voorwaarde waar het aan moet voldoen er één die slaagt als de structuur van kooien is veranderd. In de query bestond een kooi uit een lijst met variabelen die zich in de kooi bevinden en vervolgens de waarde die gelijk moet zijn aan de som van de cellen binnen de kooi. Hier wordt tussen de lijst en de waarde nog een nieuwe lijst aan toegevoegd: een lijst met mogelijke waardes per cel. Deze lijst bevat voor elke cel binnen de kooi een lijst met daarin de getallen die mogelijk in deze cel passen. Dit predicaat houdt direct rekening met de minimale en de maximale waarde die in een kooi passen. Als een kooi bijvoorbeeld grootte 2 heeft en totale waarde 17, dan is het voor een mens duidelijk dat hier geen 7 in kan, de tweede cel zou dan waarde 10 moeten krijgen en dit kan niet. Voor deze kooi zou het programma dus alleen de opties 8 en 9 voor beide cellen geven. Daarnaast houdt dit predicaat ook rekening met dubbele getallen binnen een kooi met grootte twee. Zo zal een kooi met grootte 2 en totale waarde 10 dus geen 5 krijgen als optie, aangezien er niet twee dezelfde getallen binnen één kooi mogen.
Als dit predicaat geslaagd is gaat het programma verder met een voorwaarde voor de makkelijkste kooien, namelijk die met maar één cel waarvan de waarde nog variabel is. Voor elke kooi wordt gecontroleerd of deze maar één vrije variabele in zich heeft. Als dit zo is neemt het predicaat de totale waarde van de kooi en haalt eventueel al ingevulde waarden daar vanaf. Vervolgens wijst het de gevonden waarde toe aan de vrije variabele. Op deze manier worden de eerste waardes dus aan de variabelen toegekend.

Innies

Nadat dit is gebeurd gaat het programma opzoek naar innies zoals uitgelegd bij de oplossingsstrategieën van de Killersudoku. Dit gaat in verschillende stappen. Er zijn verschillende checks voor innies binnen rijen, innies binnen kolommen en innies binnen nonets. Daarnaast zijn er nog checks voor innies binnen 2 rijen, 2 kolommen, 3 rijen en 3 kolommen. Om wat beter te laten zien hoe dit werkt volgt hier een voorbeeld hoe het programma naar innies binnen rijen zoekt. De andere aanroepen werken ongeveer hetzelfde.
Het predicaat *innieRij* heeft twee argumenten. Als eerste een lijst met alle rijen, als tweede een lijst met alle kooien. Het predicaat checkt voor elke afzonderlijke rij ten eerste of er een innie in zit, als dit het geval is wijst hij een waarde toe aan de innie en gaat door met de volgende rij. Als dit niet het geval is gaat hij zonder verder iets te doen door met de volgende rij. Het predicaat slaagt bij het basisgeval: wanneer de lijst met rijen leeg is.

Het predicaat *inniecheck* wat checkt of een rij een innie bevat heeft 5 argumenten. De eerste twee zijn gelijk aan elkaar, namelijk de lijst van in dit geval alle variabelen van de rij die we bekijken. Het derde argument is een lijst van alle kooien, het vierde argument is een zogenaamde accumulator, wat dit precies is wordt later uitgelegd. Ten slotte is het laatste argument de variabele die de cel vertegenwoordigd die een innie is. Deze is bij de aanroep nog een willekeurige variabele, aangezien we nog niet weten welke cel een innie is. Mocht het predicaat slagen, dan zal deze variabele gelijk worden gemaakt aan de variabele die een innie is.
Om te beginnen wordt het eerste element van de rij bekeken. Dit kan twee dingen zijn, namelijk een nog niet toegekende variabele (zoals bijvoorbeeld A3) of een waarde die al wel is toegekend (zoals bijvoorbeeld 5). Als dit element een al toegekende waarde is gaat het predicaat verder met het volgende element in de rij. Als dit element een variabele is wordt er bekeken of de kooi waar deze zich in bevindt zich in het geheel in de rij bevindt. Als dit zo is, gebeurt er verder niks en gaat het predicaat weer verder met het volgende element in de rij. Als dit niet zo is, komt de accumulator in beeld. Een accumulator is een handige manier om iets bij te houden. In dit geval houdt de accumulator bij hoeveel cellen binnen de rij in een kooi zitten die ook deels buiten de rij valt. Als de kooi deels buiten de rij valt wordt het getal één bij de accumulator opgeteld. Bij de aanroep van dit predicaat zetten we de accumulator op nul. Op deze manier kunnen we precies kijken hoeveel cellen binnen de rij in een kooi zitten die deels buiten de rij valt. Naast het feit dat de accumulator met één opgehoogd wordt, verandert het vijfde argument, de innie, van waarde. Deze is nu gelijk aan de cel die gecheckt werd, dat was namelijk een mogelijke innie. Het enige wat het predicaat nu nog doet is de hele rij doorlopen op bovenstaande manier. Tot slot zijn er twee mogelijke uitkomsten. De ene mogelijkheid is dat de accumulator nul of groter dan één is. In dit geval zit er geen innie in de rij en zal het predicaat niet slagen. De andere mogelijkheid is dat de accumulator precies gelijk is aan één. In dit geval slaagt het predicaat en is het vijfde argument dus gelijk aan de variabele die een innie is.

Als het predicaat *inniecheck* slaagt en er dus een innie in de rij zit, moeten we aan deze innie nog een waarde toewijzen. Dit gebeurt in het predicaat *innieToewijzing*. Dit predicaat heeft 4 argumenten. Ten eerste de lijst met variabelen waar de innie in zit, in dit geval dus de lijst variabelen die in de rij zitten. Het predicaat heeft ook de lijst met kooien nodig, de innie die we bij *inniecheck* hebben gevonden en een bepaald cijfer. Dit cijfer is er om aan te geven hoeveel rijen, kolommen of nonets we bekijken. In ons geval zou het dus een 1 zijn, aangezien we in één rij kijken. Het predicaat *innieToewijzing* roept meteen een predicaat *innieToewijzing* met 5 argumenten aan. Het toegevoegde argument is een accumulator die begint op de vaste totale waarde van een rij, kolom of nonet. In het geval van een 9x9 Killersudoku is dat dus 45.
Net zoals bij *inniecheck* bekijkt het predicaat telkens het eerste element van de lijst. Er zijn een aantal

mogelijkheden wat dit eerste element is. Als het eerste element een al toegewezen waarde is, haalt het predicaat die waarde van de accumulator (die dus begonnen is op 45) af, en gaat het verder met het volgende element. Als het eerste element gelijk is aan de innie doet het predicaat niks en gaat het direct door met het volgende element. Als het eerste element een nog niet toegewezen variabele is, bekijkt het predicaat wat de totale waarde is van de kooi waar dat element in zit en trekt die waarde van de accumulator af. Vervolgens haalt het alle variabele elementen die in die gevonden kooi zitten uit de lijst waar we doorheen aan het gaan zijn, zodat een kooi niet tweemaal van de accumulator gehaald wordt. Als er ook toegewezen variabelen in de kooi zitten worden deze weer bij de accumulator opgeteld. Dit laatste gebeurt omdat deze toegewezen variabelen eventueel ook buiten de rij kunnen liggen, in dit geval moeten deze vanzelfsprekend niet van de accumulator afgehaald worden. Als de lijst waar doorheen gegaan werd leeg is, is de innie gelijk aan de accumulator en wordt deze waarde aan de innie toegewezen. In het geval dat er twee rijen of kolommen worden bekeken moet er nog eens 45 bij de accumulator opgeteld worden, bij drie rijen of kolommen 90.

Nu een innie is toegewezen gaat het predicaat *innieRij* dus door met de volgende rij, om die weer eerst te checken en daarna eventueel een variabele een waarde toe te wijzen.

Outties

In essentie werkt het predicaat *outtieRij* vrijwel hetzelfde als het predicaat voor innies. Ook dit predicaat checkt eerst of er een outtie in een rij zit, waarna het deze een waarde geeft. In het checken zelf zit echter een klein verschil. Het predicaat moet namelijk niet alleen kijken of er precies één cel is die in een kooi zit die buiten de rij valt, maar het moet dan ook nog precies één cel zijn die buiten de rij valt, niet meer. Dus als *outtiecheck* een variabele vindt die zich in een kooi bevindt die niet volledig binnen de rij valt, moet er iets meer gebeuren dan bij *inniecheck*. Dit gaat als volgt: het predicaat zoekt de lijst van variabelen van de kooi waarin de gevonden variabele zich bevindt op. Uit die lijst wordt de gevonden variabele weggehaald. Vervolgens wordt er gekeken of er precies één variabele in die lijst overblijft, die zich niet in de lijst van variabelen bevindt waar het predicaat doorheen gaat. In het geval van een outtie bij een rij zou het predicaat dus checken of er precies één variabele in de kooi zit die zich niet in de rij bevindt. In dat geval is er een outtie gevonden en onthoudt het predicaat deze. Ook wordt net als bij *inniecheck* nu de accumulator met één opgehoogd. Als de accumulator precies op één eindigt slaagt *outtiecheck* en wordt, net als bij *inniecheck*, de variabele in het laatste argument aangepast naar de variabele waarde van de outtie.

Het toewijzen van een outtie werkt exact hetzelfde als het toewijzen van een innie, op één belangrijk punt na. Bij het toewijzen van outties begint de accumulator op nul in plaats van 45 en worden de gevonden waardes van al ingevulde cellen en kooi opgeteld bij de accumulator. Vervolgens wordt, als de rij volledig doorgelopen is, 45 van de accumulator afgehaald. Natuurlijk kan dit ook weer met twee of drie rijen of kolommen, in dit geval moet er respectievelijk 90 of 135 van de accumulator worden afgehaald.

Opties verwijderen

Na deze eerste predicaten zullen de eerste getallen ingevuld zijn. Dit betekent dat er een aantal opties uit de cellen kan worden verwijderd. Een getal kan namelijk maar één keer voorkomen in elke rij, kolom en nonet. Het predicaat *opties* doet dit in het programma. Dit doet het als volgt: het gaat door de lijst van kooien heen en bekijkt voor elke cel in welke rij, kolom en nonet deze zich bevindt. Vervolgens verwijdert het alle waarden die al geünificeerd zijn die in deze rij, kolom of nonet voorkomen uit de lijst van opties.
Doordat deze opties nu zijn verwijderd blijven er binnen de kooien mogelijk getallen over die niet

langer een optie zijn. Stel dat uit een kooi met grootte twee en totale waarde acht uit één cel de 5 wordt verwijderd, dan is de 3 in de andere cel uit die kooi ook niet meer mogelijk. Om dit probleem op te lossen is er het predicaat *kooiSomElim* wat getallen dus elimineert op basis van de som van de kooi. Dit predicaat gaat door de lijst van kooien heen en bekijkt zo elke kooi apart. Het bekijkt voor de eerste cel in de kooi of het tot de totale waarde kan komen als het de eerste optie uit die cel optelt bij een willekeurige optie uit de andere cellen. Mocht dit slagen dan gaat het door met de tweede optie uit die cel totdat er geen opties meer zijn. Als dit niet slaagt dan verwijdert het de optie uit de cel. Als vervolgens een cel helemaal bekeken is, gaat het predicaat door met de volgende cel binnen de kooi, tot alle cellen zijn geweest. Vervolgens gaat het door met de volgende kooi, tot alle kooien zijn geweest. Hierna zet het predicaat de kooien met de bijgewerkte opties in het laatste argument, waarmee het programma weer verder kan werken.
Nu een aantal opties verwijderd zijn is het mogelijk dat er cellen zijn met nog maar één optie. Hier is een apart klein predicaat voor, *enigeOptieToewijzing*, wat ervoor zorgt dat deze enige optie ook echt word toegewezen aan de cel.
Tot slot is er nog één predicaat wat opties verwijdert, namelijk het predicaat *verwijderMogelijkheden.* Dit predicaat gaat ten eerste opzoek naar kooien waarvoor alle cellen geldt dat hun opties gelijk zijn aan die van de rest van de cellen binnen de kooi. Als er dan ook nog net zoveel opties per cel zijn als cellen in de kooi, is het dus duidelijk welke getallen in die kooi moeten, alleen de volgorde is nog onbekend. Bijvoorbeeld een kooi grootte twee en totale waarde 17 heeft voor beide cellen de opties 8 en 9. In dit geval is het dus duidelijk dat in die kooi de getallen 8 en 9 komen, alleen nog niet in welke cel welk getal. Dit feit kan uitgebuit worden door op zoek te gaan naar een rij, kolom of nonet waar de kooi volledig in ligt. Als deze rij, kolom of nonet gevonden wordt kunnen uit alle cellen binnen die rij, kolom of nonet de opties 8 en 9 weg worden gehaald. Dit is precies wat *verwijderMogelijkheden* doet. Als het eerste deel van het predicaat slaagt en het inderdaad een kooi vindt waarvan het duidelijk is welke getallen erin horen, gaat het predicaat kijken of de kooi volledig binnen een rij, kolom of nonet ligt. Mocht dit zo zijn dan verwijdert het de gevonden opties uit die rij, kolom of nonet en gaat het door met de volgende kooi. Dit predicaat geeft ook weer in zijn laatste argument de kooien met bijgewerkte opties terug.

Recursie

Nu al deze predicaten geslaagd zijn is de Killersudoku hoogstwaarschijnlijk nog niet helemaal ingevuld. Het predicaat *solution* roept nu een predicaat *solution2* aan, wat precies hetzelfde doet als *solution* op de begin- en eindfase na. Het slaat namelijk de beginfase van *solution* over: de kooien hoeven geen lijst met opties per cel te krijgen, die hebben ze al. In de beginfase van *solution2* gebeurt wel iets anders: de lijst van kooien (met daarin dus ook de opties) wordt opgeslagen in een variabele. Nadat dit gebeurd is gaat *solution2* op dezelfde volgorde als *solution* weer proberen het aantal opties te verkleinen en waarden aan variabelen toe te wijzen. Er wordt wederom gekeken of er nog kooien zijn met één cel die nog niet is ingevuld, er wordt op verschillende manieren naar innies en outties gezocht en er worden weer op verschillende manieren opties verwijderd. Tot slot bekijkt het predicaat of de lijst van kooien zoals die er op dit moment uitziet verschilt van de kooien zoals deze aan het begin zijn opgeslagen. Als dit zo is roept het predicaat zichzelf opnieuw aan en probeert het dus nogmaals opties te verminderen en cellen in te vullen. Als dit niet zo is gaat het predicaat naar een grensgeval wat ervoor zorgt dat het slaagt en stopt het programma. Aan het begin van elke aanroep van *solution2* wordt de Killersudoku zoals hij er op dat moment uitziet geprint op het scherm, met niet alleen de ingevulde cellen, maar ook de opties voor nog niet ingevulde cellen. Op deze manier kan de gebruiker bijhouden wat er per aanroep van *solution2* verandert aan de Killersudoku en waar het programma eventueel vastloopt.

Appendix B: Opdracht proefpersonen

Hoi!

Bedankt dat je mee wilt werken aan mijn scriptieonderzoek!

Zoals al een beetje was uitgelegd heb ik een filmpje nodig waarin jij een Killer Sudoku oplost.

In dit filmpje is het belangrijk dat je hardop denkt (het is voor mij van belang dat ik van elk cijfer wat je toevoegt weet waarom je dat doet en ook van elke mogelijkheid die je wegwerkt) en dat de puzzel goed te zien is.

Bijgevoegd zijn twee puzzels: een makkelijke (Ongeveer 15 minuten puzzelen) en een iets moeilijkere (Ongeveer 30 minuten puzzelen).

Het liefst zou ik een filmpje terugkrijgen van beide puzzels, maar ik snap dat niet iedereen daar tijd voor heeft.

Voor de mensen die nog nooit een Killer Sudoku hebben gemaakt raad ik aan eerst even snel op het internet op te zoeken hoe je zo'n puzzel oplost, daarna te beginnen met de makkelijke puzzel en als je daarna nog zin/tijd hebt de moeilijkere.

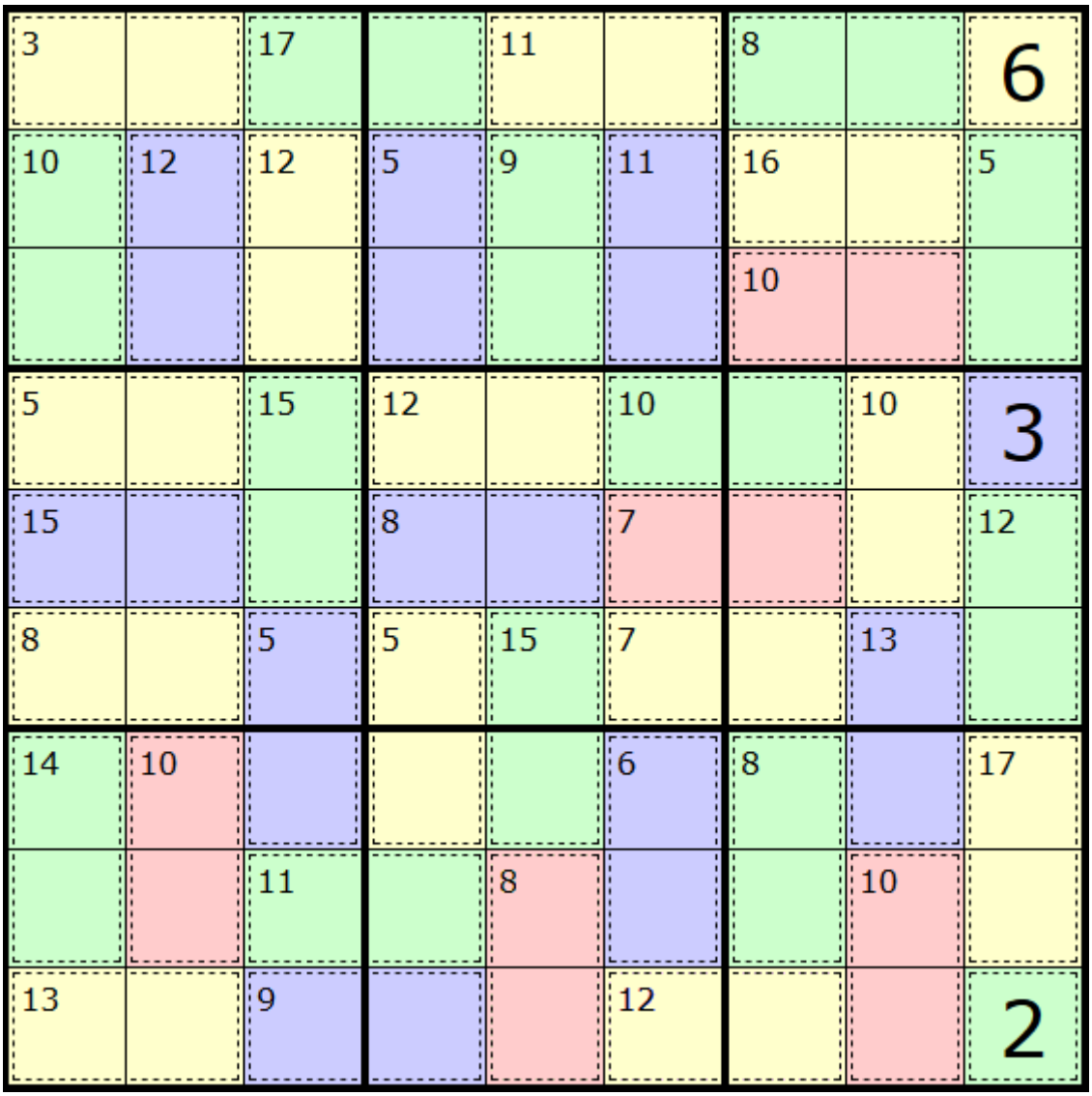
Als je al wel vaker een Killer Sudoku hebt gemaakt heb ik het liefst dat je eerst de moeilijke maakt, zodat ik daar ook een aantal filmpjes van heb. Als je daarna nog zin hebt kan je natuurlijk altijd nog de makkelijke opsturen.

Graag in je mail terug even laten weten of je al eerder een Killer Sudoku hebt gemaakt.

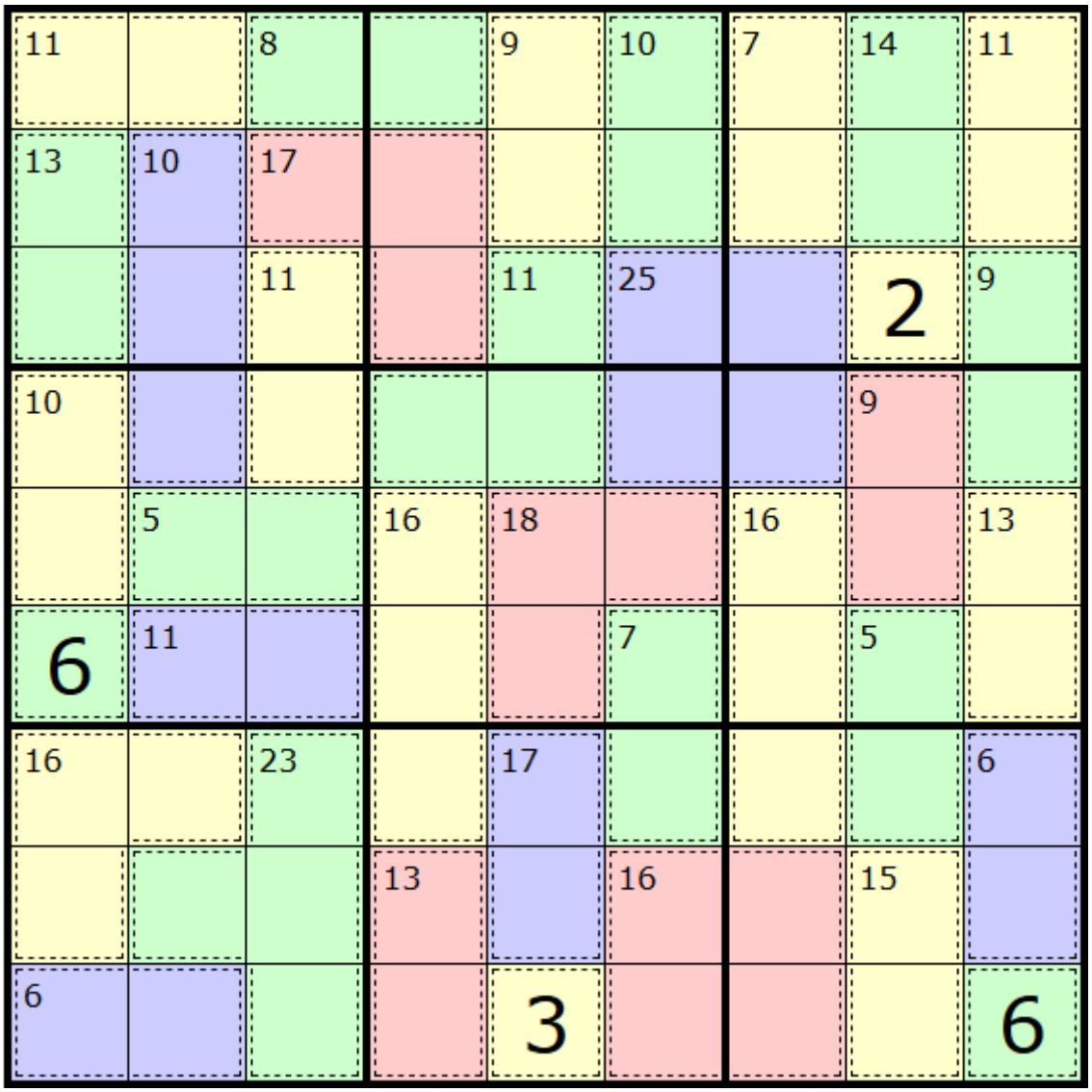
Alvast heel erg bedankt voor het helpen!

Groetjes Steven

Appendix C: Makkelijke Killersudoku



Appendix D: Moeilijke Killersudoku



Appendix E: Menselijke resultaten Killersudoku's

Makkelijke Killersudoku

Proefpersoon 1

Ervaring: Nee.

Toegewezen Cel	Getal	Strategie
D1	9	Innie (nonet)
C1	8	Totale waarde kooi
C6	2	Innie (nonet)
C7	3	Totale waarde kooi
H7	5	Enige overige optie
G9	3	Enige overige optie
H6	8	Totale waarde kooi
F9	9	Totale waarde kooi
E6	9	Enige positie binnen nonet
E7	6	Totale waarde kooi
C8	4	Enige overige optie
C9	1	Enige overige optie
D9	8	Totale waarde kooi
D8	7	Totale waarde kooi
E8	3	Enige positie binnen nonet
E9	5	Totale waarde kooi
D7	1	Enige positie binnen nonet
F7	4	Enige positie binnen nonet
F8	2	Totale waarde kooi
D6	4	Totale waarde kooi
H9	4	Innie (rij)
H8	6	Totale waarde kooi
G7	7	Enige overige optie
G8	1	Totale waarde kooi
A8	5	Enige positie binnen rij
A7	9	Totale waarde kooi
B8	8	Enige overige optie
I8	9	Enige overige optie
I7	8	Totale waarde kooi
B7	2	Totale waarde kooi
B1	1	Enige overige optie
A1	2	Totale waarde kooi
A5	8	Enige positie binnen nonet
B5	7	Totale waarde kooi
B4	4	Enige overige optie
A4	1	Totale waarde kooi
B6	5	Enige positie binnen nonet
A6	3	Totale waarde kooi

A9	7	Enige positie binnen kolom
B9	6	Totale waarde kooi
F6	1	Enige positie binnen rij
G6	6	Totale waarde kooi
I6	7	Enige overige optie
I5	5	Totale waarde kooi
H5	1	Enige positie binnen nonet
H4	9	Totale waarde kooi
H2	7	Enige positie binnen nonet
G2	9	Totale waarde kooi
H3	2	Enige positie binnen kolom
G3	8	Totale waarde kooi
H1	3	Enige overige optie
G1	5	Totale waarde kooi
E1	4	Enige positie binnen rij
F1	7	Totale waarde kooi
B3	9	Enige positie binnen kolom
B2	3	Totale waarde kooi
C3	7	Enige positie binnen nonet
C2	5	Totale waarde kooi
D2	2	Enige overige optie
D3	3	Totale waarde kooi
E3	1	Enige overige optie
E2	8	Totale waarde kooi
F2	6	Enige overige optie
F3	5	Totale waarde kooi
I3	4	Enige overige optie
I2	1	Totale waarde kooi
A3	6	Enige overige optie
A2	4	Enige overige optie
C4	6	Enige overige optie
C5	9	Totale waarde kooi
E5	2	Enige overige optie
D5	6	Totale waarde kooi
F5	3	Enige overige optie
G5	4	Totale waarde kooi
D4	5	Enige overige optie
E4	7	Totale waarde kooi
F4	8	Enige overige optie
G4	2	Totale waarde kooi

Proefpersoon 2

Ervaring: Nee.

Toegewezen Cel	Getal	Strategie
C1	8	Innie (nonet)
D1	9	Totale waarde kooi
C6	2	Innie (nonet)
C7	3	Totale waarde kooi
A9	7	Enige overige optie
B9	6	Totale waarde kooi
B4	4	Enige positie kolom
A4	1	Totale waarde kooi
B5	7	Enige overige optie
A5	8	Totale waarde kooi
A6	3	Enige overige optie
B6	5	Totale waarde kooi
B1	1	Enige overige optie
A1	2	Totale waarde kooi
D6	4	Enige overige optie
D7	1	Totale waarde kooi
C8	4	Enige overige optie
D8	7	Totale waarde kooi
C9	1	Enige overige optie
D9	8	Totale waarde kooi
D4	5	Enige positie kolom
E4	7	Totale waarde kooi
D5	6	Enige positie kolom
E5	2	Totale waarde kooi
C5	9	Enige overige optie
C4	6	Totale waarde kooi
G5	4	Enige overige optie
F5	3	Totale waarde kooi
H4	9	Enige overige optie
H5	1	Totale waarde kooi
F6	1	Enige overige optie
G6	6	Totale waarde kooi
G4	2	Enige overige optie
F4	8	Totale waarde kooi
E6	9	Enige overige optie
E7	6	Totale waarde kooi
I6	7	Enige overige optie
I5	5	Totale waarde kooi
H6	8	Enige overige optie
H7	5	Totale waarde kooi
F9	9	Enige positie nonet

G9	3	Totale waarde kooi
H9	4	Enige positie rij
E9	5	Enige overige optie
E8	3	Totale waarde kooi
H8	6	Totale waarde kooi
F8	2	Enige overige optie
F7	4	Totale waarde kooi
B8	8	Enige overige optie
B7	2	Totale waarde kooi
G7	7	Enige overige optie
G8	1	Totale waarde kooi
I8	9	Enige overige optie
I7	8	Totale waarde kooi
A8	5	Enige overige optie
A7	9	Totale waarde kooi
G3	8	Enige overige optie
H3	2	Totale waarde kooi
H2	7	Enige overige optie
G2	9	Totale waarde kooi
G1	5	Enige overige optie
H1	3	Totale waarde kooi
F1	7	Enige overige optie
E1	4	Totale waarde kooi
D2	2	Enige overige optie
D3	3	Totale waarde kooi
E3	1	Enige overige optie
E2	8	Totale waarde kooi
C2	5	Enige overige optie
C3	7	Totale waarde kooi
I3	4	Enige overige optie
I2	1	Totale waarde kooi
F2	6	Enige overige optie
F3	5	Totale waarde kooi
B2	3	Enige overige optie
B3	9	Totale waarde kooi
A2	4	Enige overige optie
A3	6	Totale waarde kooi

Proefpersoon 3

Ervaring: Ja.

Toegewezen Cel	Getal	Strategie
D1	9	Innie (nonet)
C1	8	Totale waarde kooi
E3	1	Enige overige optie
E2	8	Totale waarde kooi
D3	3	Enige overige optie
D2	2	Totale waarde kooi
A4	1	Enige overige optie
B4	4	Totale waarde kooi
C2	5	Enige overige optie
C3	7	Totale waarde kooi
B3	9	Enige overige optie
B2	3	Totale waarde kooi
I3	4	Enige overige optie
I2	1	Totale waarde kooi
A3	6	Enige overige optie
A2	4	Totale waarde kooi
F3	5	Enige overige optie
F2	6	Totale waarde kooi
B6	5	Enige overige optie
A6	3	Totale waarde kooi
A1	2	Enige overige optie
B1	1	Totale waarde kooi
C6	2	Enige overige optie
C7	3	Totale waarde kooi
F6	1	Enige overige optie
G6	6	Totale waarde kooi
D6	4	Enige overige optie
D7	1	Totale waarde kooi
E6	9	Enige overige optie
E7	6	Totale waarde kooi
H7	5	Enige overige optie
H6	8	Totale waarde kooi
I6	7	Enige overige optie
I5	5	Totale waarde kooi
G4	2	Enige overige optie
F4	8	Totale waarde kooi
G3	8	Enige overige optie
H3	2	Totale waarde kooi
H1	3	Enige overige optie
G1	5	Totale waarde kooi
H4	9	Enige overige optie

H5	1	Totale waarde kooi
C4	6	Enige overige optie
C5	9	Totale waarde kooi
F5	3	Enige overige optie
G5	4	Totale waarde kooi
E5	2	Enige overige optie
D5	6	Totale waarde kooi
E4	7	Enige overige optie
D4	5	Totale waarde kooi
E1	4	Enige overige optie
F1	7	Totale waarde kooi
F9	9	Enige overige optie
G9	3	Totale waarde kooi
G7	7	Enige overige optie
G8	1	Totale waarde kooi
G2	9	Enige overige optie
H2	7	Totale waarde kooi
A7	9	Enige overige optie
A8	5	Totale waarde kooi
I7	8	Enige overige optie
I8	9	Totale waarde kooi
B7	2	Enige overige optie
B8	8	Totale waarde kooi
F7	4	Enige overige optie
F8	2	Totale waarde kooi
E8	3	Enige overige optie
E9	5	Totale waarde kooi
C8	4	Enige overige optie
D8	7	Totale waarde kooi
C9	1	Enige overige optie
D9	8	Totale waarde kooi
H8	6	Enige overige optie
H9	4	Totale waarde kooi
A9	7	Enige overige optie
B9	6	Totale waarde kooi
A5	8	Enige overige optie
B5	7	Totale waarde kooi

Proefpersoon 4

Ervaring: Ja.

Toegewezen Cel	Getal	Strategie
E2	8	Enige overige optie
E3	1	Totale waarde kooi
I3	4	Enige overige optie
I2	1	Totale waarde kooi
D1	9	Enige overige optie
C1	8	Totale waarde kooi
E5	2	Enige positie kolom
D5	6	Totale waarde kooi
G5	4	Enige overige optie
F5	3	Totale waarde kooi
F6	1	Enige overige optie
G6	6	Totale waarde kooi
E6	9	Enige overige optie
E7	6	Totale waarde kooi
D6	4	Enige overige optie
D7	1	Totale waarde kooi
I5	5	Enige overige optie
I6	7	Totale waarde kooi
C5	9	Enige positie rij
H5	1	Enige positie rij
H4	9	Totale waarde kooi
H2	7	Enige overige optie
G2	9	Totale waarde kooi
H6	8	Enige overige optie
H7	5	Totale waarde kooi
G4	2	Enige overige optie
F4	8	Totale waarde kooi
E4	7	Enige overige optie
D4	5	Totale waarde kooi
E1	4	Enige overige optie
F1	7	Totale waarde kooi
G7	7	Enige overige optie
G8	1	Totale waarde kooi
H1	3	Enige overige optie
G1	5	Totale waarde kooi
H3	2	Enige overige optie
G3	8	Totale waarde kooi
D3	3	Enige overige optie
D2	2	Totale waarde kooi
C4	6	Totale waarde kooi
C6	2	Enige positie rij

C7	3	Totale waarde kooi
F9	9	Enige positie nonet
G9	3	Totale waarde kooi
C9	1	Enige overige optie
D9	8	Totale waarde kooi
C8	4	Enige overige optie
D8	7	Totale waarde kooi
H8	6	Enige overige optie
H9	4	Totale waarde kooi
E9	5	Enige overige optie
E8	3	Totale waarde kooi
F8	2	Enige overige optie
F7	4	Totale waarde kooi
C2	5	Enige overige optie
C3	7	Totale waarde kooi
F2	6	Enige overige optie
F3	5	Totale waarde kooi
B2	3	Enige overige optie
B3	9	Totale waarde kooi
A3	6	Enige overige optie
A2	4	Totale waarde kooi
A4	1	Enige overige optie
B4	4	Totale waarde kooi
A1	2	Enige overige optie
B1	1	Totale waarde kooi
B6	5	Enige overige optie
A6	3	Totale waarde kooi
B7	2	Enige overige optie
B8	8	Totale waarde kooi
I8	9	Enige overige optie
I7	8	Totale waarde kooi
B5	7	Enige overige optie
A5	8	Totale waarde kooi
A7	9	Enige overige optie
A8	5	Totale waarde kooi
A9	7	Enige overige optie
B9	6	Totale waarde kooi

Proefpersoon 5

Ervaring: Nee.

Toegewezen Cel	Getal	Strategie
D1	9	Enige overige optie
C1	8	Totale waarde kooi
E3	1	Enige overige optie
E2	8	Totale waarde kooi
I3	4	Enige overige optie
I2	1	Totale waarde kooi
D3	3	Enige overige optie
D2	2	Totale waarde kooi
A2	4	Enige overige optie
A3	6	Totale waarde kooi
F3	5	Enige overige optie
F2	6	Totale waarde kooi
A4	1	Enige overige optie
B4	4	Totale waarde kooi
B1	1	Enige overige optie
A1	2	Totale waarde kooi
I5	5	Enige overige optie
I6	7	Totale waarde kooi
C6	2	Enige positie nonet
C7	3	Totale waarde kooi
C2	5	Enige overige optie
B2	3	Enige overige optie
A9	7	Enige overige optie
B9	6	Totale waarde kooi
A5	8	Enige overige optie
B5	7	Totale waarde kooi
B3	9	Totale waarde kooi
C3	7	Totale waarde kooi
C8	4	Enige overige optie
D8	7	Totale waarde kooi
C9	1	Enige overige optie
D9	8	Totale waarde kooi
D5	6	Enige overige optie
D4	5	Enige positie kolom
E4	7	Totale waarde kooi
E5	2	Totale waarde kooi
E1	4	Enige overige optie
F1	7	Totale waarde kooi
C4	6	Enige overige optie
C5	9	Totale waarde kooi
H4	9	Enige positie rij

G2	9	Enige overige optie
H2	7	Totale waarde kooi
F4	8	Enige overige optie
G4	2	Totale waarde kooi
D6	4	Enige overige optie
D7	1	Totale waarde kooi
F7	4	Enige overige optie
F8	2	Totale waarde kooi
F9	9	Enige overige optie
G9	3	Totale waarde kooi
G1	5	Enige overige optie
H1	3	Totale waarde kooi
G3	8	Enige overige optie
H3	2	Totale waarde kooi
H9	4	Enige overige optie
G5	4	Enige positie nonet
F5	3	Totale waarde kooi
E6	9	Enige overige optie
E7	6	Totale waarde kooi
E8	3	Enige overige optie
E9	5	Totale waarde kooi
F6	1	Enige overige optie
G6	6	Totale waarde kooi
G7	7	Enige overige optie
G8	1	Totale waarde kooi
B6	5	Enige overige optie
A6	3	Totale waarde kooi
B7	2	Enige overige optie
B8	8	Totale waarde kooi
H8	6	Totale waarde kooi
H6	8	Enige overige optie
H7	5	Totale waarde kooi
A7	9	Enige overige optie
A8	5	Totale waarde kooi
I8	9	Enige overige optie
I7	8	Totale waarde kooi
H5	1	Enige overige optie

Proefpersoon 6

Ervaring: Nee.

Toegewezen Cel	Getal	Strategie
D1	9	Enige overige optie
C1	8	Totale waarde kooi
D2	2	Enige overige optie
D3	3	Totale waarde kooi
E2	8	Enige overige optie
E3	1	Totale waarde kooi
I2	1	Enige overige optie
I3	4	Totale waarde kooi
A2	4	Enige overige optie
A3	6	Totale waarde kooi
F2	6	Enige overige optie
F3	5	Totale waarde kooi
A5	8	Enige overige optie
B5	7	Totale waarde kooi
A9	7	Enige overige optie
B9	6	Totale waarde kooi
A4	1	Enige overige optie
B4	4	Totale waarde kooi
C6	2	Enige overige optie
C7	3	Totale waarde kooi
B1	1	Enige overige optie
A1	2	Totale waarde kooi
B6	5	Enige overige optie
A6	3	Totale waarde kooi
C2	5	Enige overige optie
C3	7	Totale waarde kooi
B2	3	Enige overige optie
B3	9	Totale waarde kooi
D6	4	Enige overige optie
D7	1	Totale waarde kooi
G4	2	Enige overige optie
F4	8	Totale waarde kooi
C4	6	Enige overige optie
C5	9	Totale waarde kooi
H4	9	Enige positie rij
H5	1	Totale waarde kooi
G2	9	Enige overige optie
H2	7	Totale waarde kooi
H3	2	Enige overige optie
G3	8	Totale waarde kooi
G5	4	Enige positie nonet

H6	8	Enige positie nonet
G6	6	Enige positie nonet
I5	5	Enige overige optie
I6	7	Totale waarde kooi
H7	5	Totale waarde kooi
A8	5	Enige overige optie
A7	9	Totale waarde kooi
H8	6	Enige overige optie
H9	4	Totale waarde kooi
C8	4	Enige positie nonet
C9	1	Enige positie nonet
F7	4	Enige overige optie
F8	2	Totale waarde kooi
B7	2	Enige overige optie
B8	8	Totale waarde kooi
I7	8	Enige overige optie
I8	9	Totale waarde kooi
G8	1	Enige positie nonet
G7	7	Totale waarde kooi
G9	3	Enige positie nonet
E8	3	Enige positie nonet
E9	5	Totale waarde kooi
D4	5	Enige positie nonet
E4	7	Totale waarde kooi
F1	7	Enige overige optie
E1	4	Totale waarde kooi
F5	3	Totale waarde kooi
F6	1	Totale waarde kooi
E5	2	Enige overige optie
D5	6	Totale waarde kooi
E6	9	Enige positie nonet
E7	6	Totale waarde kooi
F9	9	Totale waarde kooi
D9	8	Totale waarde kooi
D8	7	Totale waarde kooi
G1	5	Enige overige optie
H1	3	Totale waarde kooi

Moeilijke Killersudoku

Proefpersoon 1

Ervaring: Nee.

Toegewezen Cel	Getal	Strategie
D8	6	Enige overige optie
D9	7	Totale waarde kooi
H8	7	Enige overige optie
H9	8	Totale waarde kooi
G7	3	Enige positie nonet
H7	4	Enige overige optie
H6	1	Totale waarde kooi
G3	4	Enige positie nonet
I3	7	Enige overige optie
I4	2	Totale waarde kooi
G4	7	Enige positie nonet
F5	9	Enige positie kolom
I6	9	Enige overige optie
I5	4	Totale waarde kooi
A4	9	Enige overige optie
A5	1	Totale waarde kooi
D7	5	Enige overige optie
F7	2	Enige overige optie
F6	5	Totale waarde kooi
F4	6	Enige overige optie
F3	8	Totale waarde kooi
E3	6	Totale waarde kooi
D4	4	Enige overige optie
E4	1	Totale waarde kooi
D1	1	Enige overige optie
C1	7	Totale waarde kooi
C2	6	Enige overige optie
G1	6	Enige overige optie
G2	1	Totale waarde kooi
B6	7	Enige overige optie
C6	4	Totale waarde kooi
B4	5	Enige overige optie
C4	8	Enige overige optie
C3	3	Totale waarde kooi
B5	3	Enige overige optie
C5	2	Totale waarde kooi
E6	2	Enige overige optie
E5	7	Totale waarde kooi
D6	3	Enige overige optie
D5	8	Totale waarde kooi

G6	8	Enige overige optie
G5	5	Totale waarde kooi
H5	6	Enige overige optie
H4	3	Totale waarde kooi
B2	4	Enige overige optie
B3	1	Totale waarde kooi
E1	4	Enige overige optie
E2	5	Totale waarde kooi
H1	5	Enige overige optie
H2	9	Totale waarde kooi
F2	7	Enige overige optie
F1	3	Totale waarde kooi
I2	3	Enige overige optie
I1	8	Totale waarde kooi
A2	8	Enige overige optie
A3	5	Totale waarde kooi
B1	9	Enige overige optie
A1	2	Totale waarde kooi
B9	2	Enige overige optie
A9	4	Totale waarde kooi
D2	2	Enige overige optie
D3	9	Totale waarde kooi
F8	4	Enige overige optie
F9	1	Enige overige optie
G8	2	Enige overige optie
G9	9	Totale waarde kooi
A7	7	Enige positie nonet
B7	6	Enige overige optie
A8	3	Enige overige optie
B8	8	Enige positie kolom
E7	8	Enige overige optie
E8	9	Totale waarde kooi
I8	5	Enige overige optie
I7	1	Totale waarde kooi
C8	1	Enige overige optie
C9	5	Enige overige optie
C7	9	Enige overige optie

Proefpersoon 2

Ervaring: Ja.

Toegewezen Cel	Getal	Strategie
D9	7	Enige overige optie
D8	6	Totale waarde kooi
H8	7	Enige overige optie
H9	8	Totale waarde kooi
F5	9	Innie (4 kolommen)
D4	4	Outie (kolom)
D7	5	Enige overige optie
F3	8	Enige overige optie
E3	6	Enige overige optie
E4	1	Totale waarde kooi
H7	4	Enige overige optie
H6	1	Totale waarde kooi
I5	4	Enige overige optie
I6	9	Totale waarde kooi
I4	2	Enige overige optie
I3	7	Totale waarde kooi
G3	4	Enige overige optie
G4	7	Enige overige optie
F4	6	Totale waarde kooi
H4	3	Enige overige optie
H5	6	Totale waarde kooi
G7	3	Totale waarde kooi
F7	2	Enige overige optie
F6	5	Totale waarde kooi
G6	8	Enige overige optie
G5	5	Totale waarde kooi
D6	3	Enige overige optie
D5	8	Totale waarde kooi
C6	4	Enige overige optie
B6	7	Totale waarde kooi
E5	7	Enige overige optie
E6	2	Totale waarde kooi
A5	1	Enige positie rij
A4	9	Totale waarde kooi
B4	5	Enige overige optie
C4	8	Enige overige optie
C3	3	Totale waarde kooi
B3	1	Enige overige optie
B2	4	Totale waarde kooi
A3	5	Enige overige optie
A2	8	Totale waarde kooi

E2	5	Enige overige optie
E1	4	Totale waarde kooi
A1	2	Enige overige optie
B1	9	Totale waarde kooi
D1	1	Enige overige optie
C1	7	Totale waarde kooi
C2	6	Enige overige optie
D3	9	Enige overige optie
D2	2	Totale waarde kooi
G2	1	Enige overige optie
G1	6	Totale waarde kooi
F1	3	Enige overige optie
F2	7	Totale waarde kooi
H2	9	Enige overige optie
H1	5	Totale waarde kooi
I1	8	Enige overige optie
I2	3	Totale waarde kooi
C5	2	Enige overige optie
B5	3	Totale waarde kooi
B8	8	Totale waarde kooi
I7	1	Enige overige optie
I8	5	Totale waarde kooi
C7	9	Enige overige optie
C8	1	Enige overige optie
C9	5	Totale waarde kooi
A9	2	Enige overige optie
B9	4	Totale waarde kooi
F9	1	Enige overige optie
F8	4	Enige overige optie
G9	9	Enige overige optie
G8	2	Totale waarde kooi
E8	9	Enige overige optie
E7	8	Totale waarde kooi
A7	7	Enige overige optie
B7	6	Enige overige optie
A8	3	Totale waarde kooi

Proefpersoon 3

Ervaring: Ja.

Toegewezen Cel	Getal	Strategie
D9	7	Enige overige optie
D8	6	Totale waarde kooi
H9	8	Enige overige optie
H8	7	Totale waarde kooi
D4	4	Outtie (4 kolommen)
F5	9	Outtie (kolom)
D7	5	Enige overige optie
B7	6	Enige positie nonet
F7	2	Enige overige optie
F6	5	Totale waarde kooi
F4	6	Enige overige optie
F3	8	Enige positie kolom
H4	3	Enige overige optie
H5	6	Totale waarde kooi
E3	6	Enige positie nonet
E4	1	Totale waarde kooi
G3	4	Enige overige optie
G4	7	Totale waarde kooi
I3	7	Enige positie nonet
I4	2	Totale waarde kooi
I7	1	Enige overige optie
I8	5	Totale waarde kooi
H7	4	Enige overige optie
H6	1	Totale waarde kooi
G7	3	Enige overige optie
I5	4	Enige overige optie
I6	9	Totale waarde kooi
G6	8	Enige overige optie
G5	5	Totale waarde kooi
D6	3	Enige overige optie
D5	8	Totale waarde kooi
E5	7	Enige overige optie
E6	2	Totale waarde kooi
A5	1	Enige positie rij
A4	9	Totale waarde kooi
B4	5	Enige overige optie
C4	8	Enige overige optie
C3	3	Totale waarde kooi
A3	5	Enige positie rij
A2	8	Totale waarde kooi
B3	1	Enige positie rij

B2	4	Totale waarde kooi
D3	9	Enige overige optie
E2	5	Enige overige optie
E1	4	Totale waarde kooi
A1	2	Enige overige optie
B1	9	Totale waarde kooi
D1	1	Enige overige optie
D2	2	Totale waarde kooi
G1	6	Enige overige optie
G2	1	Totale waarde kooi
H2	9	Enige overige optie
H1	5	Totale waarde kooi
I2	3	Enige overige optie
I1	8	Totale waarde kooi
F2	7	Enige overige optie
F1	3	Totale waarde kooi
C1	7	Totale waarde kooi
C2	6	Totale waarde kooi
C5	2	Enige overige optie
B5	3	Totale waarde kooi
B6	7	Enige overige optie
C6	4	Totale waarde kooi
B9	2	Enige overige optie
B8	8	Enige overige optie
A9	4	Totale waarde kooi
A8	3	Enige overige optie
A7	7	Enige overige optie
C7	9	Enige overige optie
C8	1	Enige overige optie
C9	5	Totale waarde kooi
E7	8	Enige overige optie
E8	9	Totale waarde kooi
F8	4	Enige overige optie
F9	1	Enige overige optie
G8	2	Enige overige optie
G9	9	Totale waarde kooi

Appendix F: Resultaten programma

Makkelijke Killersudoku

Toegewezen Cel	Getal	Strategie
C1	8	Innie (nonet)
C6	2	Innie (nonet)
I5	5	Innie (2 rijen)
E1	4	Innie (3 kolommen)
D1	9	Totale waarde kooi
F1	7	Totale waarde kooi
I6	7	Totale waarde kooi
C7	3	Totale waarde kooi
C2	5	Enige overige optie
C3	7	Totale waarde kooi
B2	3	Enige overige optie
F2	6	Enige overige optie
B3	9	Totale waarde kooi
C8	4	Enige overige optie
F3	5	Totale waarde kooi
D8	7	Totale waarde kooi
D3	3	Enige overige optie
E3	1	Enige overige optie
A4	1	Enige overige optie
D5	6	Enige overige optie
E5	2	Totale waarde kooi
D2	2	Totale waarde kooi
E2	8	Totale waarde kooi
B4	4	Totale waarde kooi
D4	5	Innie (kolom)
C9	1	Innie (kolom)
A1	2	Enige overige optie
A2	4	Enige overige optie
I3	4	Enige overige optie
E4	7	Totale waarde kooi
F4	8	Enige overige optie
C5	9	Enige overige optie
B6	5	Enige overige optie
E6	9	Enige overige optie
F6	1	Enige overige optie
D9	8	Totale waarde kooi
B1	1	Totale waarde kooi
I2	1	Totale waarde kooi
A3	6	Totale waarde kooi
C4	6	Totale waarde kooi
D4	6	Totale waarde kooi

G4	2	Totale waarde kooi
F5	3	Totale waarde kooi
G5	4	Totale waarde kooi
A6	3	Totale waarde kooi
G6	6	Totale waarde kooi
E7	6	Totale waarde kooi
F7	4	Totale waarde kooi
F8	2	Totale waarde kooi
B5	7	Enige overige optie
A9	7	Enige overige optie
B9	6	Enige overige optie
A5	8	Totale waarde kooi
H4	9	Innie (rij)
F9	9	Innie (kolom)
D6	4	Innie (nonet)
H6	8	Innie (nonet)
G3	8	Enige overige optie
H3	2	Totale waarde kooi
H5	1	Totale waarde kooi
D7	1	Totale waarde kooi
B8	8	Totale waarde kooi
G9	3	Totale waarde kooi
H9	4	Enige overige optie
G2	9	Totale waarde kooi
B7	2	Totale waarde kooi
H7	5	Totale waarde kooi
H8	6	Totale waarde kooi
G1	5	Innie (kolom)
E9	5	Innie (rij)
H1	3	Totale waarde kooi
A7	9	Enige overige optie
G7	7	Enige overige optie
E8	3	Totale waarde kooi
G8	1	Totale waarde kooi
I8	9	Enige overige optie
I7	8	Totale waarde kooi
A8	5	Totale waarde kooi

Moeilijke Killersudoku

Toegewezen Cel	Getal	Strategie
F5	9	Innie (2 kolommen)
D4	4	Outtie (kolom)
D8	6	Enige overige optie
D9	7	Enige overige optie
H9	8	Enige overige optie
D7	5	Totale waarde kooi
H8	7	Totale waarde kooi

Appendix G: Gemiddelde afwijking proefpersonen

Makkelijke Killersudoku

Cel	Gemiddelde Afwijking
A1	20,27
A2	37,53
A3	37,13
A4	24,53
A5	35,07
A6	29,53
A7	21,73
A8	22,93
A9	37,07
B1	20,93
B2	30,33
B3	30,07
B4	24,67
B5	35,33
B6	28,87
B7	17,33
B8	19,07
B9	37,07
C1	1,67
C2	29,40
C3	28,07
C4	16,40
C5	21,20
C6	16,67
C7	16,67
C8	27,33
C9	27,20
D1	1,53
D2	33,33
D3	33,73
D4	26,07
D5	31,93
D6	13,60
D7	13,87
D8	32,27
D9	32,00
E1	17,87
E2	34,00
E3	33,47

Cel	Gemiddelde Afwijking
E4	26,33
E5	31,27
E6	30,60
E7	30,60
E8	21,33
E9	21,87
F1	16,80
F2	33,47
F3	34,13
F4	19,20
F5	29,87
F6	27,60
F7	17,20
F8	17,33
F9	24,87
G1	18,20
G2	16,73
G3	10,73
G4	20,27
G5	28,00
G6	21,40
G7	19,87
G8	19,47
G9	21,20
H1	19,40
H2	16,60
H3	11,40
H4	11,13
H5	22,73
H6	25,73
H7	27,47
H8	22,00
H9	20,27
I2	35,13
I3	35,00
I5	14,67
I6	14,27
I7	19,87
I8	19,87

Moeilijke Killersudoku

Cel	Gemiddelde Afwijking
A1	9,33
A2	10,00
A3	11,33
A4	13,33
A5	12,00
A7	5,33
A8	6,67
A9	4,67
B1	8,00
B2	4,00
B3	5,33
B4	2,67
B5	17,33
B6	21,33
B7	45,33
B8	6,00
B9	6,00
C1	21,33
C2	21,33
C3	2,67
C4	2,67
C5	16,00
C6	22,67
C7	8,67
C8	6,67
C9	6,67
D1	15,33
D2	8,00
D3	12,67
D4	12,00
D5	8,00
D6	8,00
D7	6,67
D8	0,67
D9	0,67
E1	2,67
E2	4,00
E3	8,67
E4	9,33

Cel	Gemiddelde Afwijking
E4	9,33
E5	4,67
E6	3,33
E7	2,00
E8	0,67
F1	3,33
F2	3,33
F3	8,67
F4	6,00
F5	4,67
F6	9,33
F7	9,33
F8	7,33
F9	7,33
G1	16,00
G2	14,67
G3	6,00
G4	4,67
G5	10,67
G6	10,67
G7	13,33
G8	7,33
G9	7,33
H1	4,67
H2	3,33
H4	20,67
H5	19,33
H6	11,33
H7	11,33
H8	0,67
H9	0,67
I1	2,00
I2	3,33
I3	6,67
I4	6,67
I5	8,67
I6	9,33
I7	35,33
I8	34,00

Appendix H: Toewijzingsmomenten

Makkelijke Killersudoku

Cel	Gemiddeld Toewijzingsmoment proefpersonen	Toewijzingsmoment Programma	Verschil
A1	28,33	27	1,33
A2	40,17	28	12,17
A3	40,17	39	1,17
A4	24,00	19	5,00
A5	38,67	53	14,33
A6	38,17	45	6,83
A7	56,33	72	15,67
A8	56,33	78	21,67
A9	39,00	51	12,00
B1	28,00	37	9,00
B2	41,83	11	30,83
B3	42,83	13	29,83
B4	24,33	24	0,33
B5	39,00	50	11,00
B6	37,50	33	4,50
B7	56,00	66	10,00
B8	56,00	62	6,00
B9	40,00	52	12,00
C1	2,50	1	1,50
C2	39,50	9	30,50
C3	41,17	10	31,17
C4	41,67	40	1,67
C5	38,67	32	6,67
C6	18,00	2	16,00
C7	19,00	8	11,00
C8	37,33	14	23,33
C9	38,33	26	12,33
D1	1,83	5	3,17
D2	30,00	22	8,00
D3	30,00	17	13,00
D4	45,50	25	20,50
D5	42,50	41	1,50
D6	25,33	20	5,33
D7	25,67	56	30,33
D8	42,67	61	18,33
D9	43,00	16	27,00
E1	50,33	36	14,33
E2	24,00	4	20,00
E3	23,67	23	0,67
E4	45,83	18	27,83

E5	42,50	30	12,50
E6	36,17	21	15,17
E7	37,17	34	3,17
E8	50,33	47	3,33
E9	50,67	74	23,33
F1	50,67	70	19,33
F2	39,00	6	33,00
F3	39,33	12	27,33
F4	41,67	15	26,67
F5	47,00	31	16,00
F6	40,00	43	3,00
F7	47,67	35	12,67
F8	48,00	48	0,00
F9	44,83	49	4,17
G1	52,83	55	2,17
G2	44,50	69	24,50
G3	45,83	65	19,17
G4	41,33	58	16,67
G5	42,33	42	0,33
G6	36,50	44	7,50
G7	47,67	46	1,67
G8	48,33	73	24,67
G9	43,00	75	32,00
H1	52,83	63	10,17
H2	44,50	71	26,50
H3	45,83	59	13,17
H4	35,50	54	18,50
H5	41,83	60	18,17
H6	36,00	57	21,00
H7	36,67	67	30,33
H8	51,67	68	16,33
H9	49,00	64	15,00
I2	28,17	38	9,83
I3	27,50	29	1,50
I5	32,33	3	29,33
I6	32,33	7	25,33
I7	58,33	77	18,67
I8	58,00	76	18,00

Moeilijke Killersudoku

Cel	Gemiddeld Toewijzingsmoment Proefpersonen	Toewijzingsmoment programma	Verschil
A1	49,33	nvt	nvt
A2	45,33	nvt	nvt
A3	45,00	nvt	nvt
A4	28,00	nvt	nvt
A5	27,67	nvt	nvt
A7	70,00	nvt	nvt
A8	71,00	nvt	nvt
A9	64,33	nvt	nvt
B1	49,67	nvt	nvt
B2	42,00	nvt	nvt
B3	41,67	nvt	nvt
B4	34,33	nvt	nvt
B5	52,00	nvt	nvt
B6	40,67	nvt	nvt
B7	50,67	nvt	nvt
B8	65,33	nvt	nvt
B9	63,67	nvt	nvt
C1	43,67	nvt	nvt
C2	44,67	nvt	nvt
C3	36,33	nvt	nvt
C4	35,33	nvt	nvt
C5	51,67	nvt	nvt
C6	41,00	nvt	nvt
C7	70,00	nvt	nvt
C8	70,00	nvt	nvt
C9	71,00	nvt	nvt
D1	39,67	nvt	nvt
D2	53,33	nvt	nvt
D3	51,33	nvt	nvt
D4	11,33	2	9,33
D5	33,00	nvt	nvt
D6	32,00	nvt	nvt
D7	10,33	6	4,33
D8	1,67	3	1,33
D9	1,33	4	2,67
E1	45,00	nvt	nvt
E2	44,67	nvt	nvt
E3	15,33	nvt	nvt
E4	16,67	nvt	nvt
E5	33,67	nvt	nvt
E6	34,00	nvt	nvt

E7	72,33	nvt	nvt
E8	72,67	nvt	nvt
F1	54,00	nvt	nvt
F2	53,67	nvt	nvt
F3	13,67	nvt	nvt
F4	16,67	nvt	nvt
F5	7,67	1	6,67
F6	17,67	nvt	nvt
F7	16,67	nvt	nvt
F8	69,00	nvt	nvt
F9	69,33	nvt	nvt
G1	43,33	nvt	nvt
G2	43,67	nvt	nvt
G3	14,00	nvt	nvt
G4	15,67	nvt	nvt
G5	32,33	nvt	nvt
G6	31,33	nvt	nvt
G7	17,33	nvt	nvt
G8	71,00	nvt	nvt
G9	71,33	nvt	nvt
H1	52,67	nvt	nvt
H2	52,33	nvt	nvt
H4	25,67	nvt	nvt
H5	26,00	nvt	nvt
H6	14,33	nvt	nvt
H7	13,33	nvt	nvt
H8	3,33	7	3,67
H9	3,67	5	1,33
I1	55,33	nvt	nvt
I2	55,00	nvt	nvt
I3	14,67	nvt	nvt
I4	15,00	nvt	nvt
I5	17,67	nvt	nvt
I6	18,00	nvt	nvt
I7	52,33	nvt	nvt
I8	52,67	nvt	nvt