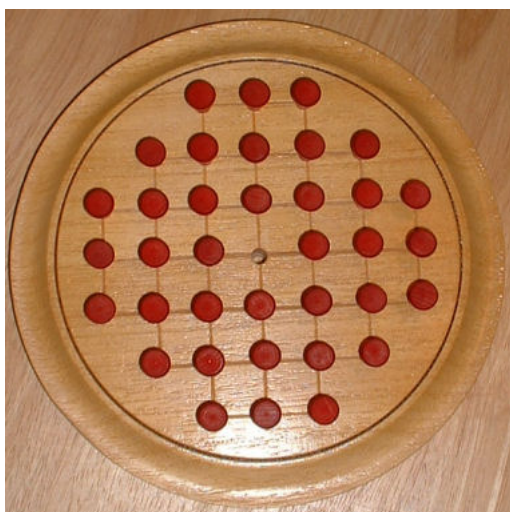


Hersenkrakers: De computer lost het voor je op

(Profielwerkstukthema gebaseerd op graaftransformaties)

Hoe zet je acht koninginnen op een schaakbord, zodat ze elkaar niet kunnen slaan? Of hoe zorg je dat je maar één pion in het midden overhoudt op het solitaire bord (Figuur 1, bij solitaire spring je met een pion over één andere heen naar een leeg vakje, en verwijder je de pion waar je overheen sprong).

Heb je altijd al het antwoord op dit soort puzzels willen weten, maar heb je nooit zin om er over na te denken, dan is er maar één oplossing: laat de computer het voor je doen.



Figuur 1: Het speelbord van Solitaire

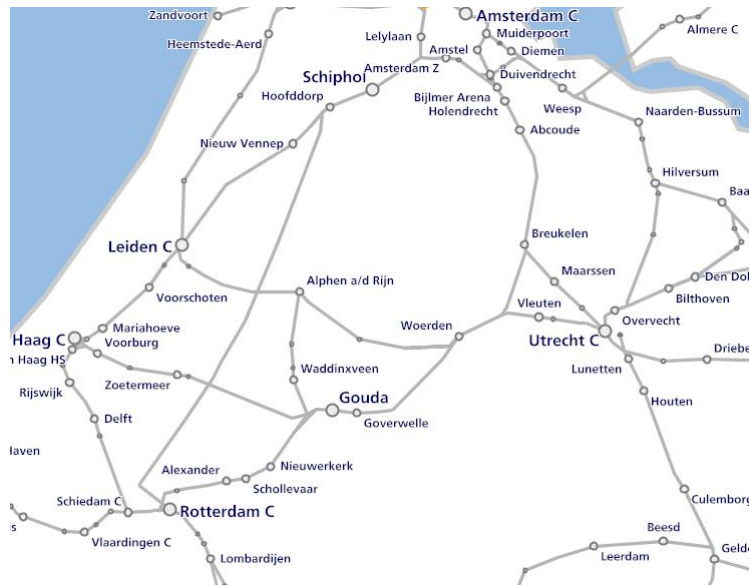
1 Wat ga je doen?

Voor dit profielwerkstuk ga je je verdiepen in graaftransformatie. Dit is een techniek waarbij systemen, zoals de hierboven genoemde puzzels maar ook bijvoorbeeld ingewikkelde computerprogramma's, worden beschreven door een *graaf*. Met behulp van regels die we *transformaties* noemen kunnen we dan interessante eigenschappen van de graaf afleiden. Deze eigenschappen zeggen dan ook iets of het systeem of computerprogramma waar de graaf voor gemaakt is, en we kunnen die eigenschappen gebruiken om te kijken of dat systeem goed werkt, of om naar een bepaalde toestand van dat systeem te zoeken.

Omdat graaftransformaties van computerprogramma's erg ingewikkeld kunnen zijn, gaan we die niet voor dit profielwerkstuk gebruiken. In plaats daarvan gebruiken we simpele puzzels, en laten we in §4 zien hoe je met graaftransformaties zo'n puzzel kan oplossen.

In §2 vind je een korte introductie over grafen en graaftransformaties. Vervolgens staan in §3 een aantal suggesties voor onderzoeksvragen die je zou kunnen gebruiken voor je profielwerkstuk. Tot slot staat in §4 een klein voorbeeld uitgewerkt.

Aan het eind van de opdracht kan je een woordenlijst vinden met belangrijke en moeilijke begrippen en een korte uitleg daarvan vinden. Als je toch ergens niet uitkomt, kan je altijd hulp vragen bij de universiteit. Stuur daarvoor een mailtje naar Arend Rensink (rensink@cs.utwente.nl).



Figuur 2: Een voorbeeldgraaf: Het spoorwegennet

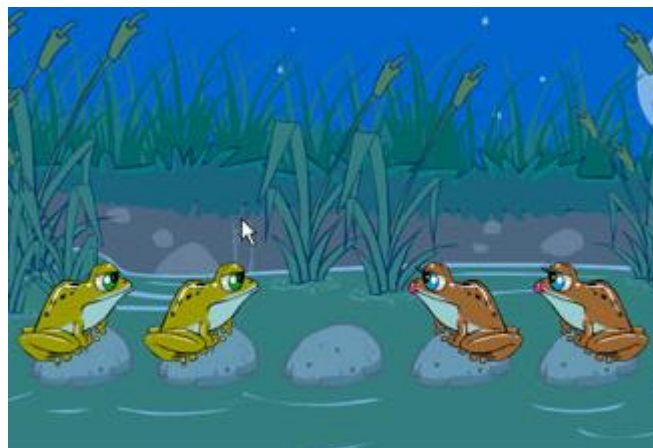
2 Wat zijn Graaftransformaties

2.1 Grafen

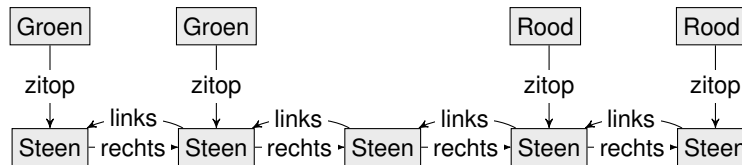
Een graaf in de wiskunde is een verzameling knopen (Engels: nodes). Om aan te geven dat er een bepaald verband is tussen twee knopen kunnen we een pijl (Engels: edge) trekken tussen deze knopen. De knopen kunnen nul, één of meer labels hebben; pijlen hebben altijd precies één label.

Als dit lastig klinkt, kijk dan bijvoorbeeld naar Figuur 2: dit is een spoorwegaanpak van de NS-website. De kaart is eigenlijk een graaf waarbij de knopen de treinstations en de pijlen de rails tussen de treinstations voorstellen. Behalve spoorwegen kunnen we met grafen nog veel meer situaties uit de praktijk op een simpele manier beschrijven.

Ter illustratie gebruiken we in de rest van deze toelichting de *kikkerpuzzel*. De beginsituatie van de kikkerpuzzel is zoals in Figuur 3. Er zijn vijf stenen: op de meest linker twee staat een groene kikker, en op de meest rechter twee staat een rode kikker. Het doel van de puzzel is om alle kikkers met alleen de volgende zetten naar de overkant te krijgen:



Figuur 3: Voorbeeldpuzzel: de kikkers



Figuur 4: Voorbeeldpuzzel: de graaf

- Als de steen rechts van een groene kikker leeg is, mag de groene kikker een stap naar rechts doen.
- Als op de steen rechts van een groene kikker een rode kikker zit, dan kan de groene kikker daar overheen springen. Dit mag alleen als de steen waarop de groene kikker land leeg is, en de groene kikker mag maar over één rode kikker tegelijk heen springen.
- Als de steen links van een rode kikker leeg is, mag de rode kikker een stap naar links doen.
- Als op de steen links van een rode kikker een groene kikker zit, dan kan de rode kikker daar overheen springen. Dit mag alleen als de steen waarop de rode kikker land leeg is, en de rode kikker mag maar over één groene kikker tegelijk heen springen.

Als we deze puzzel in een graaf willen beschrijven zullen we negen knopen nodig hebben; vijf knopen die de stenen voorstellen en vier knopen die de kikkers voorstellen. Om aan te geven op welke volgorde de stenen liggen, tekenen we links- en rechtspijlen tussen twee stenen die naast elkaar liggen. Tot slot tekenen we een aantal 'zitop' pijlen tussen de kikkers en de stenen om aan te geven op welke steen de kikkers zitten. De graaf ziet er dan uit zoals in Figuur 4

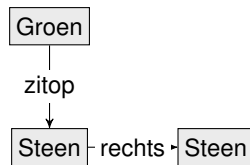
2.2 Transformaties

Als we eenmaal een graaf hebben en we bepaalde eigenschappen van die graaf willen onderzoeken, kunnen we daarvoor graaftransformaties gebruiken. Een graaftransformatie is niets anders dan een aanpassing van de graaf naar een nieuwe graaf. Dit kan bijvoorbeeld door het toevoegen van nieuwe knopen aan de graaf, of door het verplaatsen van pijlen. Door het na elkaar uitvoeren van verschillende transformaties kunnen we de graaf zo veranderen dat we gemakkelijk belangrijke eigenschappen kunnen aflezen.

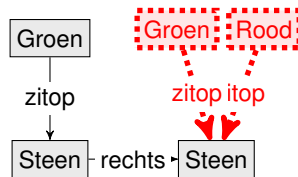
Denk bijvoorbeeld weer aan de voorbeeldgraaf van de kikkerpuzzel in Figuur 4. De graaftransformaties die we voor deze graaf willen uitvoeren zijn alle geldige zetten die we voor de kikkerpuzzel mogen doen. Door deze transformaties achter elkaar uit te voeren, kunnen we het maken van deze puzzel simuleren en zo naar de oplossing zoeken.

De zetten in de puzzel kunnen in verschillende volgordes worden uitgevoerd. In het begin kan je bijvoorbeeld de groene kikker een stapje naar rechts zetten, of de rode kikker een stapje naar links zetten. Uiteraard zullen we meer dan één zet nodig hebben om de puzzel op te lossen. Om een oplossing voor de puzzel te vinden moeten we dus naar alle mogelijk combinaties van zetten kijken, en kijken of er een combinatie tussen zit die tot de oplossing leidt. Omdat dit veel werk is gebruiken we hier een computerprogramma voor, namelijk GROOVE. In §2.3 vertellen we wat meer over GROOVE, en in §4 laten we zien hoe GROOVE gebruikt kan worden om de kikkerpuzzel op te lossen.

Voordat we verder gaan met uitleg over GROOVE, laten we eerst nog even zien hoe je graaftransformaties kan specificeren (specificeren betekent op een precieze en wiskundige manier opschrijven). Graaftransformaties worden uitgedrukt in *regels*: elke regel beschrijft één soort transformatie. Een (graaf)transformatieregel werkt met behulp van patroon herkenning. Dit betekent dat elke regel een patroon bevat, en dat een regel toegepast kan worden als de graaf aan het patroon dat in de regel staat voldoet.



Figuur 5: Transformatiepatroon: Groene kikker stap vooruit 1



Figuur 6: Transformatiepatroon: Groene kikker stap vooruit 2

Een voorbeeld van zo'n patroon kan je vinden in Figuur 5. Dit patroon is voor de zet waarbij de groene kikker één stap vooruit wordt gezet. Voor deze zet is niet de hele graaf interessant, dus beschrijft het patroon maar een klein deel van de graaf: de groene kikker die we willen verplaatsen, de steen waar de kikker op zit, en de steen waar we de kikker naar toe willen verplaatsen.

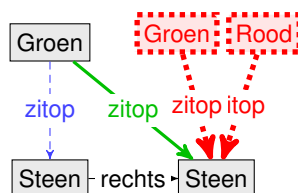
Denk eraan dat als we de groene kikker een stap vooruit willen zetten, dat de nieuwe steen waar de kikker op zal komen te staan wel leeg moet zijn. Dit moeten we expliciet in de transformatieregel zeggen, en dit doen we door rode knopen aan de graaf toe te voegen. Dit betekent dat deze regel alleen mag worden gebruikt als de zwarte knopen *wel* en de rode knopen *niet* in de graaf voorkomen. In ons geval betekent dat, dat er geen kikkers op de volgende steen mogen zitten, zie ook Figuur 6.

In Figuur 6 hebben we nu een patroon dat bepaalt wanneer we een transformatie mogen gebruiken, maar we weten nog niet wat we moeten aanpassen. Voor deze transformatie willen wij dat de groene kikker een stap naar voren zet, ofwel dat de 'zitop'-pijl naar de volgende steen wordt verplaatst. Dit doen we in de regel door aan te geven dat de oude 'zitop'-pijl moet worden verwijderd, en dat er een nieuwe 'zitop'-pijl moet worden toegevoegd. De regel ziet er dan uit zoals in Figuur 7. De oude 'zitop'-pijl is nu blauw gestippeld, wat betekent dat de pijl wel in de graaf voor de transformatie, maar niet in de graaf na de transformatie zit. De nieuwe 'zitop'-pijl is aangegeven met groen en zit in de nieuwe graaf, maar niet in de graaf vóór de transformatie.

Als we transformatieregels wat algemener bekijken zien we dat deze regels zelf ook weer een graaf zijn! De onderdelen in de regels kunnen vier functies hebben, die we allemaal met een eigen kleur aangeven:

Zwarte delen (smal dooregetrokken) Knopen en pijlen die voor én na de transformatie in de graaf voorkomen.

Blauwe delen (smal onderbroken) Knopen en pijlen die wel vóór de transformatie, maar niet erna voorkomen.



Figuur 7: Transformatiepatroon: Groene kikker stap vooruit 3

Groene delen (breed doorgetrokken) Knopen en pijlen die aan de graaf worden toegevoegd.

Rode delen (breed onderbroken) Knopen en pijlen die niet vóór de transformatie mogen voorkomen.

2.3 GROOVE

Voor deze opdracht zal je gebruik gaan maken van het computerprogramma GROOVE. GROOVE is speciaal gemaakt om grafen mee te tekenen, en om automatisch graaftransformaties uit te kunnen voeren. Van <http://sf.net/projects/groove/> kan je GROOVE downloaden. De download bevat ook een korte Engelse handleiding van het programma. In §4 zullen we ook laten zien hoe je GROOVE kan gebruiken om het kikkerprobleem van het voorbeeld uit §2 op te lossen.

Voordat je GROOVE kan gebruiken heb je ook Java nodig. Als je nog geen Java hebt geïnstalleerd, dan kan je de nieuwste versie downloaden via <http://www.java.com/en/download/installed.jsp>.

3 Onderzoeksvragen

In §4 laten we al zien hoe je met graaftransformaties een oplossing kunt vinden voor de kikkerpuzzel. Maar behalve de kikkerpuzzel zijn er nog veel meer puzzels waar je met behulp van graaftransformaties de oplossing voor kan vinden. Hier laten we een paar interessante zien.

De wolf, de geit en de kool

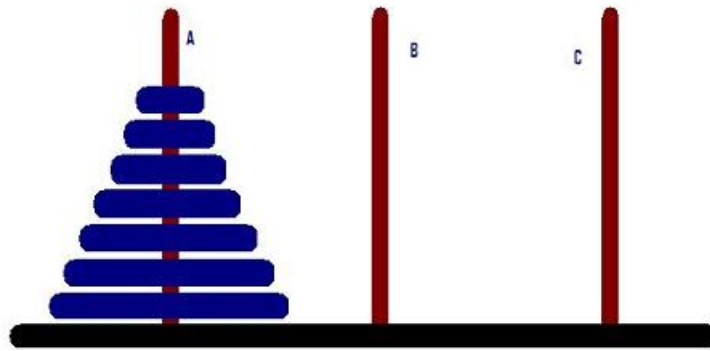
Een boer heeft een wolf, een geit en een kool. Aan het begin staan ze alle drie aan de kant van een rivier, en de boer wil ze met zijn boot naar de overkant brengen. De boot is alleen niet zo groot, dus hij kan maar één ding tegelijk meenemen. Maar als de boer de wolf en de geit alleen laat, dan zal de wolf de geit opeten. En net zo, als de boer de geit alleen laat met de kool, dan eet de geit de kool op. De vraag is nu: hoe kan de boer de wolf, de geit en de kool naar de overkant krijgen, zonder dat er eentje wordt opgegeten? Let op dat de boer ook met een lege boot naar de overkant mag, en dat hij dingen die al aan de overkant staan, ook weer mee terug mag nemen.

Om dit op te lossen met graaftransformaties, moet je eerst een startgraaf maken die beschrijft dat de wolf, de geit, de kool en de boer met zijn boot aan één kant van een rivier staan. Daarna moet je regels maken die beschrijven dat de boer iets naar de overkant van de rivier verplaatst en dat dingen elkaar kunnen opeten.

Drie missionarissen en drie kannibalen

In deze puzzel staan er drie missionarissen en drie kannibalen aan de kant van een rivier. Ze willen naar de overkant en hebben daarvoor een bootje waar twee mensen in kunnen. Het bootje kan niet leeg naar de overkant, er moet altijd minimaal één persoon in het bootje zitten. Als er aan één kant van de rivier meer kannibalen zijn dan missionarissen (de mensen die in het bootje zitten tellen ook mee, dus niet alleen de mensen die op de kant staan), dan worden de missionarissen opgegeten door de kannibalen. Hoe krijg je nu iedereen aan de overkant zonder dat er iemand wordt opgegeten?

Om dit op te lossen met graaftransformaties, moet je eerst een startgraaf maken die beschrijft dat de missionarissen en kannibalen aan één kant van een rivier staan. Daarna moet je regels maken die beschrijven dat het bootje met mensen naar de overkant gaat en dat de missionarissen kunnen worden opgegeten.



Figuur 8: De torens van Hanoi

De torens van Hanoi

In het Boeddhistische Hanoi zijn monniken al eeuwen bezig een toren van 64 schijven van oplopende grootte van de ene berg naar de andere te verhuizen, door de zware schijven één voor één te verplaatsen. Naast de beginplaats en de gewenste eindplaats is er een tussenstop waarop schijven tijdelijk kunnen worden neergelegd. Een schijf mag nooit op een kleinere schijf komen te liggen. Hoe moeten de monniken die doen?

Een kleinere toren, van 7 schijven, is schematisch afgebeeld in Figuur 8: A is de beginpositie, B de tussenpositie en C de gewenste eindpositie.

Het aantal stenen in de puzzel staat niet vast. Dit betekent dat je kan beginnen met een simpele versie van de puzzel (bijvoorbeeld met drie stenen), en daarna de puzzel kan uitbreiden met meer stenen.


4 Een simpel voorbeeld

In dit §gaan we het voorbeeld van de kikkerpuzzel uitwerken. De begin situatie van de kikkerpuzzel is zoals in Figuur 3 en het doel van de puzzel is om alle kikkers met alleen de volgende zetten naar de overkant te krijgen:

- Als de steen rechts van een groene kikker leeg is, mag de groene kikker een stap naar rechts doen.
- Als op de steen rechts van een groene kikker een rode kikker zit, dan kan de groene kikker daar overheen springen. Let op, dit mag alleen als de steen waarop de groene kikker landt leeg is, en de groene kikker mag maar over één rode kikker tegelijk heen springen.
- Als de steen links van een rode kikker leeg is, mag de rode kikker een stap naar links doen.
- Als op de steen links van een rode kikker een groene kikker zit, dan kan de rode kikker daar overheen springen. Let op, dit mag alleen als de steen waarop de rode kikker land leeg is, en de rode kikker mag maar over één groene kikker tegelijk heen springen.

De uitwerking van dit voorbeeld bestaat uit drie stappen: het maken van een graaf van de begin situatie; het maken van de transformatieregels; de transformatieregels toepassen om de oplossing te vinden. Voor het uitwerken zullen we gebruik maken van het programma GROOVE; in §2.3 staat waar je dit programma kan downloaden.

4.1 De startgraaf


Open GROOVE en begin met een nieuwe grammatica door bij 'File' op 'new Grammar' te klikken. Het onderste vak van de linker kolom is nu nog leeg, maar bevat straks de lijst met grafen die hebben gemaakt. Klik in dit blok op de nieuwe graaf knop (, let op de kleine G), en een nieuw scherm wordt geopend waarin we grafen kunnen tekenen.

Bovenin het scherm staan nu een aantal knoppen. Door te klikken en te slepen kan je nieuwe knopen en pijlen tekenen, en bestaande knopen verslepen. Als je een knoop of pijl een (nieuw) label wilt geven, moet je dit doen door erop te dubbelklikken.

Nu zou je zelf de startgraaf moeten kunnen tekenen, hij moet er ongeveer uitzien zoals in Figuur 4. Denk eraan dat als knopen of pijlen in de graaf hetzelfde label hebben, dit ook precies hetzelfde is gespeld, inclusief hoofdletters. Als je dit niet goed doet, zullen later de transformatieregels niet goed werken.

Denk er aan de graaf op te slaan als hij klaar is.

4.2 Transformatieregels

Nu we onze startgraaf hebben gemaakt, kunnen we beginnen aan de transformatieregels. De transformatieregels komen bovenaan in de linker kolom te staan. Als je daar op de nieuwe regel knop drukt (, let op de kleine 'R'), opent er weer een scherm waar je grafen kan tekenen. Dit scherm werkt hetzelfde als het scherm dat we gebruikte voor de startgraaf, je kunt nu dus zelf de knopen en de pijlen voor de regel in Figuur 7 tekenen.

Om de knopen en pijlen hun speciale betekenis (de kleuren) te geven, moeten we zogenaamde keywords toevoegen aan hun labels. De keywords die je nodig zou kunnen hebben kan je vinden in §6. Er zijn drie belangrijke keywords:

new: Betekent dat een knoop of pijl nieuw moet worden aangemaakt, een groene knoop of groene pijl dus.

del: Betekent dat een knoop of pijl verwijderd moet worden, een blauwe knoop of blauwe pijl dus.

not: Betekent dat een knoop of pijl helemaal niet mag voorkomen, een rode knoop of rode pijl dus.

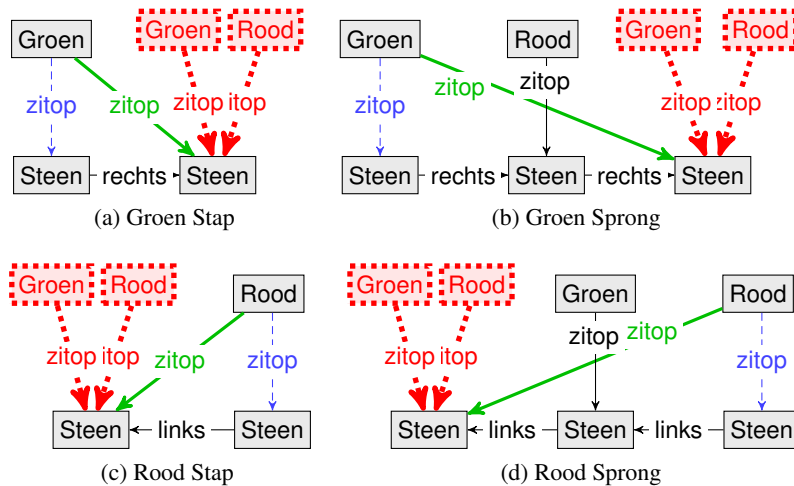
Let op! Als je aan een pijl een keyword wil toevoegen moet dit op één regel direct voor het label. Als je bij een knoop een keyword wil toevoegen moet dit op een aparte regel. Gebruik 'Shift+Enter' om in de editor een nieuwe regel in te voeren.

Omdat we vier soorten zetten mogen doen in deze puzzel, hebben we ook vier transformatieregels nodig. In Figuur 9 kan je vinden hoe deze regels eruit komen te zien.

4.3 Toestandsruimte

Nu we ook de transformatieregels hebben, kunnen we GROOVE laten zoeken naar een oplossing van de puzzel. Dit doen we door GROOVE alle toestanden van het systeem te laten uitrekenen, de zogenaamde *toestandsruimte*. De toestandsruimte bevat alle grafen die we kunnen maken met de startgraaf en de transformatieregels die we hebben gemaakt. In ons geval zal de toestandsruimte dus alle mogelijk combinaties van zetten bevatten, en bij elke combinatie de plaats waar de kikkers staan. Om de toestandsruimte te berekenen moet je (in GROOVE) op 'Explore' en dan op 'Explore State Space' klikken. In het tabblad 'LTS' zal nu de toestandsruimte verschijnen.

Als je alles goed hebt gedaan, zal de toestandsruimte 23 grafen bevatten. Als je één van de toestanden selecteert, kan je in het meest linker tabblad zien hoe deze graaf er uit ziet. De toestandsruimte bevat een paar speciale toestanden: de groene 's0' graaf is onze startgraaf, de oranje grafen zijn eindtoestanden, vanuit deze situaties kunnen we dus geen nieuwe transformaties (en dus zetten) meer doen. Als je wil



Figuur 9: Transformatieregels voor de kikkerpuzzel

weten welke transformaties (en dus zetten) nodig zijn om een bepaalde graaf te bereiken, moet je de pijlen vanuit 's0' naar die graaf volgen.

Nu gaan we in de toestandsruimte op zoek naar de graaf die de oplossing beschrijft. In deze graaf staan de rode kikkers helemaal links en de groene kikkers helemaal rechts, en kunnen we geen zetten meer doen. De oplossing zou dus een van de oranje grafen moeten zijn, en inderdaad graaf 's22' is onze eindsituatie. Om de oplossing te vinden hoeven we nu alleen maar de pijlen van 's0' naar 's22' te volgen en te kijken welke zet er bij elke pijl wordt uitgevoerd.

In plaats van zelf naar de gewenste eindtoestand te zoeken kan je dit ook weer aan GROOVE overlaten. Maak een nieuwe transformatieregel die test op het gewenste eindpatroon; in dit geval de graaf waarin de groene kikkers rechts en de rode links op het veld staan. De regel hoeft niets aan deze graaf te veranderen. Als je nu de toestandsruimte uitrekent kan je zien waar (d.w.z., in welke toestand) deze nieuwe regel toepasbaar is. Het pad in te toestandsruimte van de begintoestand naar deze eindtoestand is de oplossing van de puzzel.

5 Woordenlijst

Graaf Verzameling knopen en pijlen, die samen een toestand (situatie) beschrijven. Denk bijvoorbeeld aan een spoorwegaanpak; de knopen zijn de treinstations en de pijlen de rails tussen de treinstations.

Graaftransformatie Aanpassen van een graaf naar een nieuwe graaf. Geldige aanpassingen zijn: het toevoegen van nieuwe knopen, het verwijderen van knopen, het toevoegen van nieuwe pijlen, het verwijderen van pijlen en het verplaatsen van pijlen. Een graaftransformatie wordt beschreven met een (graaf)transformatieregel.

Grammatica Verzameling transformatieregels.

GROOVE GROOVE is een computer programma dat gebruikt kan worden voor het ontwerpen en uitvoeren van graaftransformaties.

Knoop (Engels: node): Element in een graaf waartussen pijlen getrokken kunnen worden. Een knoop heeft nul, één of meer labels.

Label Tekst op een knoop of pijl.

Pijl (Engels: edge): Gerichtte verbinding tussen twee knopen, voorzien van een label.

Simuleren Bij een (computer)simulatie laten we de computer uitrekenen wat er allemaal kan gebeuren. In het geval van de puzzels in deze opdracht laten we de computer dus berekenen welke zetten er allemaal mogelijk zijn.

Specificeren Als we iets specificeren (bijvoorbeeld een transformatieregel), bedoelen we dat we het op een wiskundige en precieze manier opschrijven.

Toestandsruimte Als we een startgraaf hebben en een verzameling transformatieregels voor deze graaf, dan bevat de toestandsruimte alle nieuwe grafen die we kunnen maken met deze startgraaf en transformatieregels. Als de startgraaf een puzzel voorstelt en de transformatieregels de mogelijke zetten zijn, dan zal de toestandsruimte alle mogelijke tussenstappen van de puzzel bevatten.

Transformatieregel Wiskundige notatie voor het opschrijven van een bepaalde graaftransformatie.

6 GROOVE aanwijzingen

6.1 Keywords

In graaftransformaties is het mogelijk knopen en pijlen toe te voegen en te verwijderen. Met de volgende keywords kan je aangeven welke knopen of pijlen moeten worden aangepast. Als je een keyword aan een knoop toe wil voegen, moet dit op een nieuwe regel. In GROOVE ga je niet met ‘enter’ naar een nieuwe regel, maar met ‘shift + enter’.

Een nieuwe pijl Zet ‘new:’ vóór het label van de pijl.

Een nieuwe knoop Zet ‘new:’ in een apart label van de knoop.

Een pijl verwijderen Zet ‘del:’ vóór het label van de pijl.

Een knoop verwijderen Zet ‘del:’ in een apart label van de knoop.

Een verboden pijl Zet ‘not:’ vóór het label van de pijl.

Een verboden knoop Zet ‘not:’ in een apart label van de knoop.

6.2 Prioriteiten

Normaal gesproken worden in alle grafen van de toestandsruimte alle regels uitgeprobeerd. Soms is dat niet de bedoeling en is een regel alleen geldig als een andere regel niet toepasbaar is. Bijvoorbeeld zouden we in de kikkerpuzzel als extra beperking kunnen invoeren dat een kikker niet mag stappen als er een (andere) kikker kan springen, of vice versa; of dat rood niets mag doen als er nog een groene kikker kan bewegen.

Dit kan in GROOVE worden aangegeven door sommige regels een hogere *prioriteit* te geven. Klik met de rechtermuisknop op een regel en kies “Rule Properties”; vul dan in de tabel bij “Priority” een positief getal in. Deze regel wordt nu als eerste uitgeprobeerd; andere regels komen alleen aan de beurt als geen van de regels met hogere prioriteit niet uitgevoerd kunnen worden.

Bijvoorbeeld kan je de kikker-sprong-regels prioriteit 10 geven, en de andere prioriteiten op 0 laten; of juist de stap-regels. (Ga na dat in het laatste geval de puzzel niet meer oplosbaar is.)

6.3 Discriminatie

Some is het nodig in een regel aan te geven dat twee knopen *niet* hetzelfde kunnen zijn. Dit kan je doen door een speciale pijl tussen die twee knopen te trekken met als label ‘!=’. Figuur 10 illustreert dit aan de hand van een regel die zegt dat een missionaris wordt opgegeten als er twee verschillende kannibalen en geen andere missionaris op dezelfde oever zijn.

