

AI Kaleidoscoop, deeltentamen 1

20 Oktober 1999

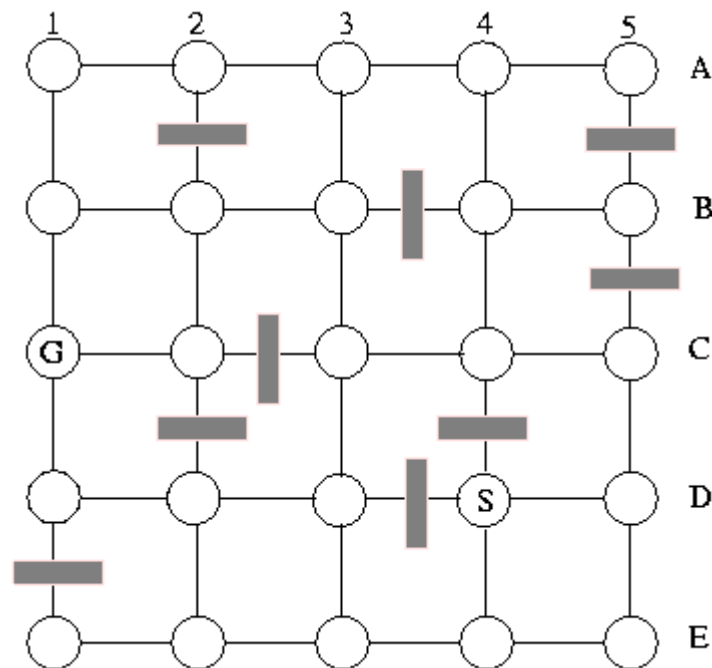
Beoordeling: De beoordeling van de vragen is als volgt:

Vraag:	1a	1b	2a	2b	3a	3b	4	5a	5b
Punten:	15	15	10	10	10	10	10	10	10

De eindscore is het totaal aantal punten (max 100) gedeeld door 10.

Vraag 1: Gebruik van zoekmethoden

Gegeven is het volgende doolhof:



Doel van het doolhof is een weg te zoeken van de startpositie D4 (gemarkeerd met S) naar de doelpositie C1 (gemarkeerd met G).

(a) Teken voor dit doolhof de zoekboom van een achterwaarts zoekend depth-first zoekalgorithme. Ga hierbij als volgt te werk:

- Nummer de knopen van de boom in de volgorde waarin het algorithme ze bezoekt
- Knopen die al eerder bezocht zijn door het algorithme kunnen worden weggelaten
- Als een knoop meerdere kinderen heeft, dan worden deze in alfabetische volgorde bezocht (A2 voor B1, A1 voor A2)

(b) Teken de zoekboom voor een voorwaarts zoeken A* algorithme dat gebruikt maakt van de Manhattan afstand als heuristiek $h(n)$, d.w.z. $h(n)$ = de "hemelsbrede" afstand tussen vakjes door horizontaal en verticaal te lopen, zonder rekening te houden met de muren in het doolhof. Welk pad wordt uiteindelijk door A* gevonden?

Nummer weer de knopen in de volgorde waarin ze door het algorithme worden bezocht. Eeder bezochte knopen kunnen weer worden weggelaten. Geef bij elke knoop in de boom ook aan wat de door A* gebruikte waarde van die knoop is. Bij gelijke waarden wordt weer de alfabetisch eerste mogelijkheid gekozen.

Vraag 2: kiezen tussen zoekmethoden

(a) Geef twee redenen die bepalen wanneer je voorwaarts of achterwaarts wil zoeken. Geef bij elke

reden aan wanneer je voorwaarts en wanneer je achterwaarts zoeken zou kiezen.

(b) Bespreek de voor- en nadelen van breadth-first search en depth-first search. Gebruik in de bespreking minstens twee verschillende eigenschappen, en geef aan welke van de twee methoden op elk van deze eigenschappen in het voordeel is.

Vraag 3: Eigenschappen van zoekmethoden

(a) Wanneer is een zoek-algorithme "*admissible*"? Geef een voorbeeld van een zoekmethode die niet *admissible* is, en geef een zoekboom waaruit dat blijkt.

(b) Een alternatieve heuristische functie voor het doolhof uit vraag 1 zou zijn $h'(n)$ = "het aantal bochten dat je hemelsbreed minimaal moet maken tot aan de doelpunt" (dus weer zonder rekening te houden met muren). Als je deze heuristiek $h'(n)$ vergelijkt met de Manhattan afstand $h(n)$ uit vraag 1b, welk van de twee is dan meer geïnformeerd? Motiveer je antwoord.

Vraag 4: Zekerheidsfactoren

De volgende twee regels zijn onderdeel van een kennissysteem dat moet besluiten of een waterkering gesloten moet worden bij bepaalde omstandigheden. Omdat het hier om onzekere kennis gaat zijn de regels voorzien van zekerheidsfactoren:

1. **ALS** tij=springvloed **DAN** sluiten (CF=0.5)
2. **ALS** weer=storm **EN** windrichting=zuidwest **DAN** sluiten (CF=0.9)

Vorige week maandag was de windrichting zuidwest (met zekerheid 1), er werd er met zekerheid 0.6 een storm voorspeld. Verder wisten we dat er geen springvloed zou zijn.

Bepaal op grond van bovenstaande gegevens met welke zekerheidsfactor het kennissysteem vorige week maandag besloten zou hebben of de waterkering gesloten moest worden of niet. Laat duidelijk de stappen in je berekening zien.

Vraag 5: Productieregel systemen

Gegeven zijn de volgende vier regels in een productieregel systeem:

- | | |
|---------------------------------|---------------------------------------|
| 1. ALS (0) | DAN (ADD 1) EN (ADD 2) |
| 2. ALS (1) | DAN (ADD 10) EN (DEL 1) |
| 3. ALS (1) EN (2) | DAN (ADD 10) |
| 4. ALS (10) | DAN (DEL 10) |

De onderstaande tabel heeft drie kolommen: de eerste kolom beschrijft de inhoud van het werkgeheugen, de tweede beschrijft de conflict-verzameling, en de derde kolom bevat de geselecteerde regel. Aangegeven is dat aanvankelijk "0" het enige element in het werkgeheugen is:

Werkgeheugen	Conflict-verzameling	Geselecteerde regel
0		

(a) Neem de tabel over en vul hem in voor de eerste 4 cycli van het gegeven productiesysteem (één regel van de tabel per cyclus), als we alleen gebruik maken van *specificity* als conflict-resolutie

strategie

(b) Als vraag (5a), maar nu bij het gebruik van de volgende conflict-resolutie strategie:

- eerst *recency*
- zo nodig *specificity* op de door *recency* geselecteerde regels
- er moet altijd aan *refractoriness* voldaan zijn.