

Tentamen Representatie & Zoeken

Antwoorden 13 Januari '97

Beoordeling: Iedereen maakt vraag 1 t/m 4. AI studenten maken ook vraag 5, de rest maakt vraag 6. Dat levert voor iedereen 100 punten op. De eindscore is 1/10 van het behaalde aantal punten.

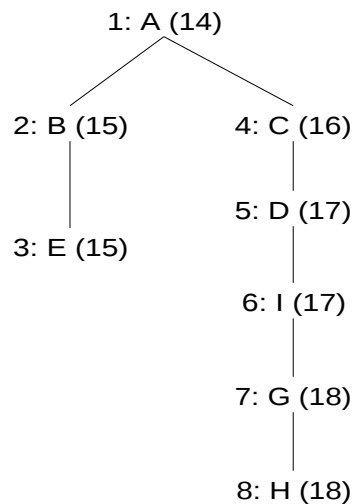
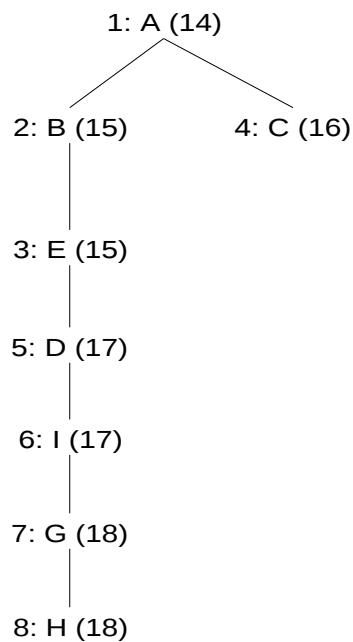
Vraag 1: zoekalgorithmen (25 punten)

Vraag 1a: A* (10 punten)

Beide onderstaande antwoorden zijn goed, en ook de variaties van beide bomen waarbij “8 : $H(18)$ ” staat als kind van E, eventueel met weglating van de andere knopen met waarde 18 (G en H).

Het gevonden pad kun dus een van de volgende waarden zijn:

- A - B - E - D - I - G - H
- A - B - E - H
- A - C - D - I - G - H



Vraag 1b: hill-climbing (10 punten)

De boom bestaat hier alleen uit het direct gevonden pad: A - B - E - H (kies steeds de laagste som van afgelegde plus geschatte afstand, en onthou geen alternatieven).

Omdat bij E de kans bestaat dat de route via D korter zou uitkomen dan direct naar H (immers $5+5+2+5 \nmid 5+5+8$) werd ook A - B - E - D - I - G - H goedgekeurd.

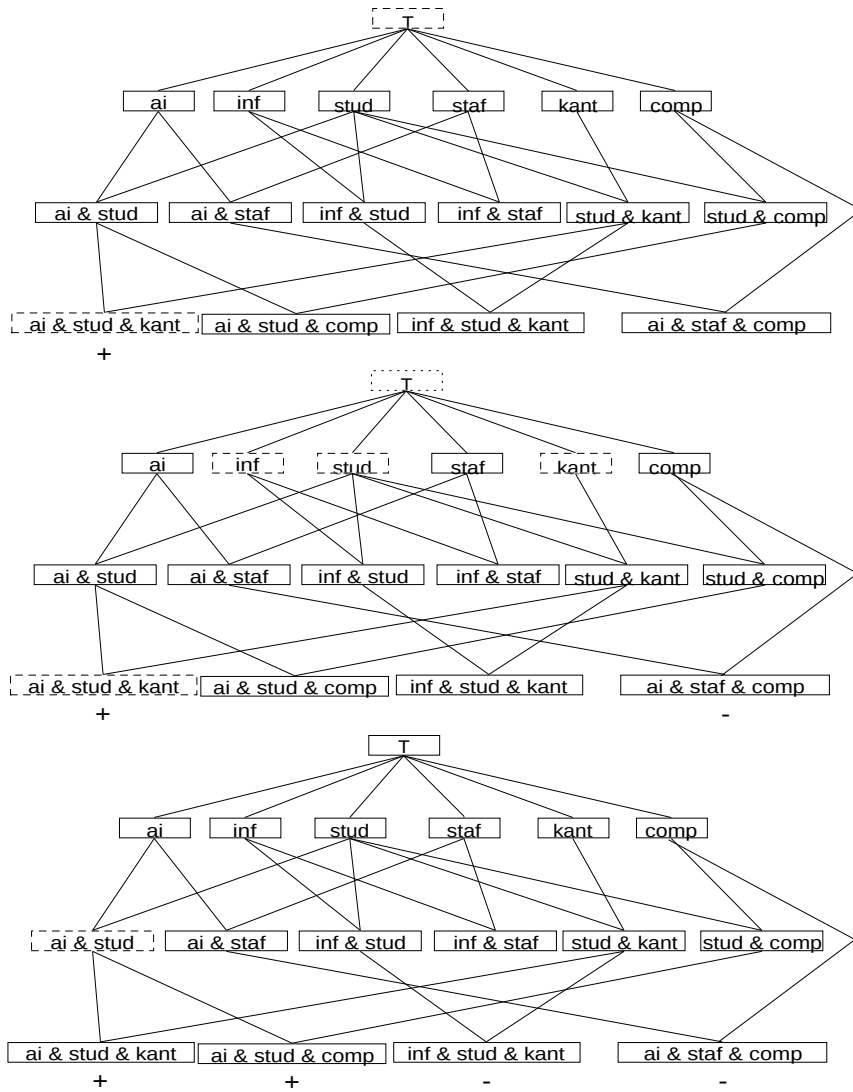
Alleen kijken naar de minimale afgelegde afstand levert A - C - D - I - G - H op, wat ook werd goedgekeurd.

Vraag 1c: vergelijking (5 punten)

In dit geval leveren beide methode paden op met dezelfde lengte (nl. 18), maar dat hoeft niet. A* levert altijd het minimale pad, maar hill-climbing kan zich soms vergissen. Maak bijvoorbeeld de afstand tussen C en D heel hoog (bijv 100), dan zal A* deze route uiteindelijk vermijden, maar hill-climbing zal aich laten verleiden doo de aantrekkelijke stap naar C en vervolgens vastzitten op deze slechte route.

Vraag 2: Version Space Search (20 punten)**Vraag 2a (5 punten)****Vraag 2b (10 punten)**

De het gestippelde concept laag in de hierarchie is de verzameling S , de gestippelde concepten hoog in de hierarchie zijn de verzameling G . Na het laatste voorbeeld vallen de twee verzamelingen samen.



Vraag 2c (5 punten)

- De knopen boven G zijn concepten die te algemeen zijn om de aangeboden voorbeelden te beschrijven, omdat ze sommige negatieve voorbeelden omvatten.
- De knopen onder S zijn concepten die te specifiek zijn om de aangeboden voorbeelden te beschrijven, omdat ze niet alle positieve voorbeelden omvatten.
- De knopen tussen S en G zijn nog kandidaten voor het te leren concept, want ze omvatten alle positieve en geen van de negatieve voorbeelden.
- Het kan voorkomen dat G onder S komt te liggen. In dat geval zijn er geen kandidaten meer om het te leren concept te beschrijven. Dat kan gebeuren als de concept-taal niet rijk genoeg is, of als er foutieve voorbeelden of tegenvoorbeelden zijn aangeboden (ruis).

Vraag 3: Natuurlijke taal (25 punten)

Vraag 3a (10 punten)

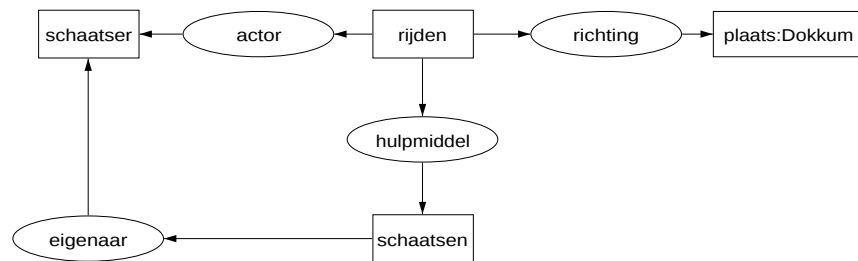
Terminale symbolen zijn **vetgedrukt**. Optionele onderdelen zijn gerealiseerd door regels die geen terminale symbolen opleveren.

zin	=	heen retour kaart viavia tijd
heen	=	ik schaats van plaats naar plaats
retour	=	en terug
retour	=	
kaart	=	met stempelkaart
kaart	=	
viavia	=	via plaatsen
viavia	=	
plaatsen	=	plaats
plaatsen	=	plaats en plaats
tijd	=	met aankomsttijd tijdstip
tijd	=	met vertrektijd tijdstip
tijd	=	
plaats	=	Dokkum Sneek Leeuwarden ...
tijdstip	=	uurgetal uur minuutgetal
uurgetal	=	0 1 2 ... 23
minuutgetal	=	0 1 2 ... 59

Vraag 3b (5 punten)

Een case-frame is een representatie van de semantiek (betekenis) van een zin waarin onderdelen zijn opengelaten die moeten worden ingevuld op basis van de details van de zin die verwerkt moet worden. Omdat de betekenis van de zin moet “passen” op het case-frame (de open onderdelen moeten kunnen worden ingevuld) beperkt het case-frame het aantal interpretaties van de zin. Interpretaties van de zin die wel syntactisch correct zijn (correct volgens de grammatica) maar waarmee het case-frame niet ingevuld kan worden, kunnen worden genegeerd. Op deze manier sturen we de syntactische analyse (het parseren) met behulp van semantische informatie (de case-frame).

Vraag 3c (10 punten)



Voor “*De schaatser*” zou het label **schaatser** veranderen in **schaatser:#nn** waarbij **nn** één of ander nummer of label is om aan te geven dat het om een specifieke maar ongenoemde schaatser gaat.

Voor “*Klasina Seinstra*” zou het label **schaatser** veranderen in **schaatser:Klasina Seinstra** om aan te geven dat het om deze specifieke schaatster gaat.

Vraag 4: Onzekerheid (20 punten)

Vraag 4a (10 punten)

$$\begin{array}{llll}
 CF(A_1) & = & \min(CF(E_1), CF(E_2)) * 0.6 & = 0.6 \\
 CF(A_2) & = & \text{op basis van 2e regel: } CF(E_2) * 0.2 & = 0.2 \\
 & & \text{op basis van 4e regel: } \max(CF(E_1), CF(E_2)) * 0.3 & = 0.3 \\
 & & \text{de combinatie van deze twee: } 0.2 + 0.3 - (0.2 * 0.3) & = 0.44 \\
 CF(A_3) & = & CF(E_3) * 0.4 & = -0.4 \\
 CF(B_1) & = & \text{op basis van 5e regel: } \max(CF(A_1), CF(A_3)) * 0.7 & = 0.42 \\
 & & \text{op basis van 7e regel: } CF(A_2) * 0.4 & = 0.176 \\
 & & \text{de combinatie van deze twee: } 0.42 + 0.176 - 0.42 * 0.176 & = 0.52208 \\
 CF(B_2) & = & \min(CF(A_1), CF(A_3)) * 0.9 & = -0.36
 \end{array}$$

Vraag 4b (10 punten)

Mogelijke redenen voor het redeneren met onzekerheid zijn:

- imperfecte domain *kennis*, zoals
 - een incomplete theorie (zoals in de geneeskunde)
 - een incorrecte theorie (zoals ook soms in de geneeskunde)
 - vage concepten (zoals bijv. in de psychiatrie)
 - een complete theorie zou te inefficiënt zijn
- Imperfecte domain *data*, zoals
 - incorrecte data (data is wel beschikbaar maar niet precies, bijv door slechte metingen)
 - incomplete data (data is niet beschikbaar, bijv omdat het te duur is om te meten)

Vraag 5: Alleen voor AI studenten (10 punten)

Invoer: De invoer van het ID3 algorithm is een verzameling voorbeelden, die elk bestaan uit een aantal attributen met de waarden voor die attributen, plus de klasse waartoe het voorbeeld behoort.

Uitvoer: De uitvoer is een beslis-boom waarin iedere knoop een gegeven instantie naar een van zijn kinderen stuurt op basis van een test op een bepaalde attribuut-waarde. Bij de bladeren staat aangegeven tot welke klasse de gegeven instantie behoort als het bij dat blad is aangekomen.

Stappen: Het ID3 algorithm kiest als top-knoop van de te construeren boom de attribuut-test die de voorbeeld-verzameling zo efficiënt mogelijk in delen opdeelt. Elke resulterende deelverzameling van de voorbeelden wordt nu bij de bijbehorende kind-knoop

recursief weer zo efficiënt mogelijk opgedeeld op basis van een ander attribuut, net zo lang tot alle elementen van een deelverzameling tot nog slechts één klasse behoren. Die klasse is dan de label van het bijbehorende blad in de boom. De keuze van de attribuut-test om een verzameling voorbeelden zo efficiënt mogelijk op te delen werkt als volgt. ID3 splitst de verzameling op basis van het attribuut met de hoogste informatie-winst. Informatie-winst van een attribuut-opsplitsing is gedefinieerd als het verschil tussen (i) de informatie-waarde van dat attribuut in de voorbeeldverzameling (gemeten in aantallen bits volgens de formule van Shannon) en (ii) de informatie die nog nodig is om de resterende deelverzamelingen verder op te splitsen.

Terminatie: Het ID3 algoritme stopt op het moment dat het, bij het recursief opsplitsen van de invoerverzameling in deelverzamelingen is aangekomen bij een deelverzameling die in zijn geheel tot één klasse behoort.

Vraag 6: Alleen voor de niet-AI studenten (10 punten)

Invoer: De invoer voor resolutie is (i) een aantal logische formules die omgezet zijn naar clausal form, en (ii) de negatie van het de te bewijzen doel-formule, ook weer omgezet in clausal form.

Uitvoer: De uitvoer is de variabele bindingen waaronder de doel-formule volgt uit de gegeven logische formules, of “nee” als de doel-formule niet volgt uit de gegeven formules.

Stappen: In elke stap wordt resolutie gedaan, dwz: er worden twee clauses gezocht zodat een bepaalde expressie positief in de ene clause voorkomt en negatief in de andere. Het resultaat van die stap is dan de combinatie van de twee clauses, met weglating van de positieve en negatieve expressies die met elkaar overeenkwamen. Mogelijkerwijs komen die positieve en negatieve expressies alleen maar overeen als bepaalde variabele bepaalde waarden krijgen (unificatie), en dan moeten die variabele-waarde worden toegepast op het resultaat van de stap. **Terminatie:** Bovenstaande stap wordt net zolang herhaald totdat de lege clause wordt afgeleid (in dat geval volgt de doel-formule blijkbaar uit de gegeven formules), of totdat er geen resolutie meer mogelijk is (in dat geval is het antwoord “nee”).