

Tentamen Kunstmatige intelligentie

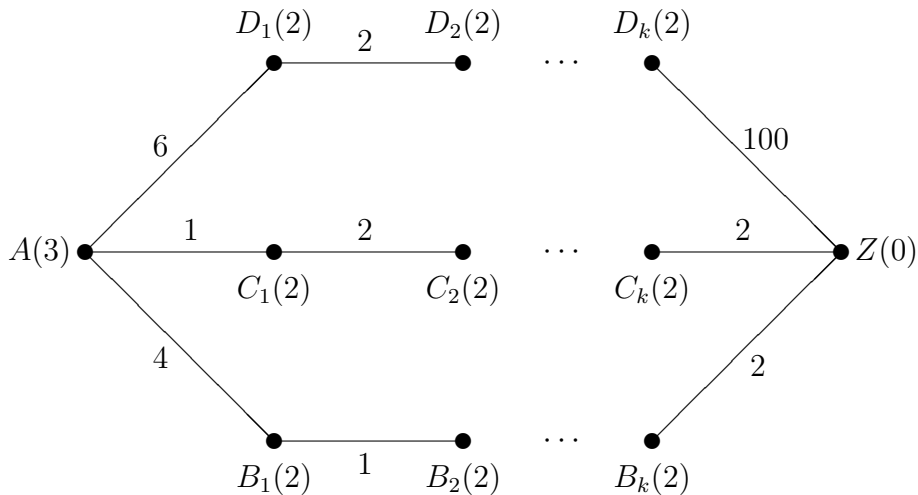
Universiteit Leiden — Informatica

woensdag 4 juni 2003, 13.00–16.00 uur

Geef bij alle antwoorden een korte en duidelijke toelichting. Veel succes!

Opgave 1. A*/IDA* (20 punten)

- a. (6 punten) Leg het A*-algoritme en het IDA*-algoritme uit.
- b. (2 punten) Wanneer is een heuristiek *admissibel*?
- c. (4 punten) Toon aan dat als het A*-algoritme bij een admissibele heuristiek stopt in een doelknoop, het deze optimaal bereikt heeft.
- d. (8 punten) Voer zowel het A*-algoritme als het IDA*-algoritme uit voor onderstaande graaf; hierbij is k een vast positief *even* geheel getal, minstens 2. Gebruik zonodig de pathmax equation. Beginknoop is A , doelknoop is Z . Bij de knopen staat tussen haakjes de (overigens admissibele) heuristische functie. De kostenfunctie staat naast de takken van de graaf; tussen de B_i -knopen zit steeds gewicht 1, tussen de C_i 's en D_i 's steeds 2. Geef duidelijk aan hoe de algoritmes verlopen.



Opgave 2. α - β -algoritme (20 punten)

We bekijken het volgende Nim-achtige spel voor twee personen. Er liggen drie lucifers en drie munten op tafel. De spelers moeten om de beurt een aantal lucifers óf een aantal munten pakken — naar keuze. Ze mogen ook precies evenveel munten als lucifers nemen. Het is verboden alles wat er nog ligt te pakken. Beiden doen één keer een greep. De waarde na afloop is het product van de aantallen munten en lucifers, met 2 opgehoogd als er meer munten dan lucifers liggen, en met 2 afgehoogd als er minder munten dan lucifers liggen; bij gelijke aantallen is het gewoon het product. Speler 1 wil zo *laag* mogelijk eindigen, speler 2 zo *hoog* mogelijk.

- a. (5 punten) Geef de *spelboom* (= *game tree*) die hierbij hoort.
- b. (5 punten) Beschrijf in woorden het *minimax-algoritme*.
- c. (5 punten) Voer dit uit voor de spelboom van a.
- d. (5 punten) Voer het α - β -algoritme uit voor deze spelboom, en geef duidelijk aan waar gesnoeid (= “gepruned”) wordt. Geef ook een korte rechtvaardiging voor dit snoeien. Zorg ervoor dat de ordening van de knopen zo is dat er zoveel mogelijk gesnoeid kan worden!

Opgave 3. Bayesiaanse netwerken (20 punten)

a. (4 punten) We hebben een *Bayesiaans netwerk* voor bioscoopbezoek, waarbij de kwaliteit (K) van een film afhangt van de acteurs (A) en het verhaal (V), terwijl Ceciel (C) een film alleen op (totaal)kwaliteit beoordeelt en Bob (B) ook nog naar de acteurs kijkt. We gebruiken kleine letters als volgt: b betekent dat Bob de film goed vindt, \bar{v} betekent dat het verhaal slecht is, etcetera.

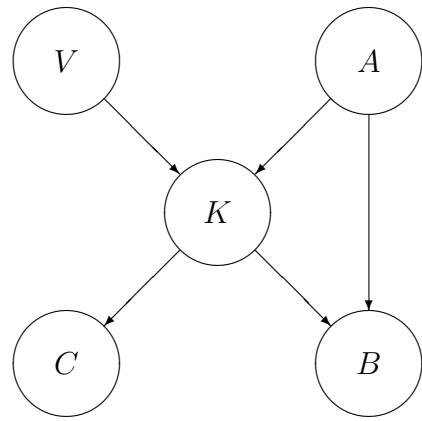
Welke kanstabellen moeten gegeven zijn, en met hoeveel kansen erin? Geef verder een voorbeeld van een voorwaardelijke onafhankelijkheid.

b. (4 punten) Druk de kans $P(c|v, \bar{a})$ dat Ceciel de film goed vindt gegeven dat het verhaal goed is maar de acteurs slecht, uit in uit de tabellen bekende kansen.

c. (4 punten) Idem voor Bob: $P(b|v, \bar{a})$.

d. (4 punten) Druk de kans $P(k|\bar{b}, \bar{c})$ dat de kwaliteit goed is gegeven dat Bob en Ceciel de film slecht vinden, uit in uit de tabellen bekende kansen.

e. (4 punten) Er zijn vier soorten queries; noem deze en leg ze kort uit. Tot welke soort(en) behoren de queries van **b**, **c** en **d**?



Opgave 4. Genetische Algoritmen en Constraint Satisfaction Problemen (20 punten)

We willen een landkaart zodanig kleuren dat aangrenzende landen verschillende kleuren uit R/G/B/Z hebben.

a. (3 punten) Formuleer dit als een *Constraint Satisfaction Problem*.

b. (4 punten) Leg uit hoe de “most constrained variable” (= “minimum remaining values”) heuristiek werkt. Geef een voorbeeld.

c. (4 punten) Leg uit hoe de “most constraining variable” heuristiek werkt, en geef aan waar deze goed gebruikt kan worden. Geef een voorbeeld.

d. (4 punten) Leg uit hoe de “least constraining value” heuristiek werkt. Geef een voorbeeld.

e. (5 punten) Geef globaal aan hoe in dit geval een *Genetisch Algoritme* om dit probleem op te lossen er uit zou kunnen zien.

Opgave 5. Neurale netwerken (20 punten)

We maken een *neuraal netwerk* met één invoerlaag, één verborgen laag en één uitvoerlaag.

a. (5 punten) Leg uit waar “bias-kopen” (extra -1 -inputs) voor nodig zijn.

b. (5 punten) Teken de netwerkarchitectuur. Geef kort in woorden aan hoe het *BackPropagation* algoritme werkt.

c. (4 punten) Leg het principe van testset en trainingsset uit. Gebruik in de uitleg het woord *overfitting*.

d. (6 punten) We willen “hoge” gewichten straffen, en voegen daarom een term $\sum_{i,j} W_{j,i}^2$ toe aan de errorfunctie: $E = 1/2 \sum_i (T_i - O_i)^2 + 1/2 \sum_{i,j} W_{j,i}^2$. Leid met “gradient descent” de leerregel voor gewichten $W_{j,i}$ van hidden laag naar uitvoerlaag af. Hierbij is T_i de i -de target, en O_i de i -de net-uitvoer. De i -de uitvoerknoop levert O_i zoals gebruikelijk door een g -functie los te laten op in_i , de met de $W_{j,i}$'s gewogen som van zijn invoeren. De activatie van de j -de hidden knoop is a_j .