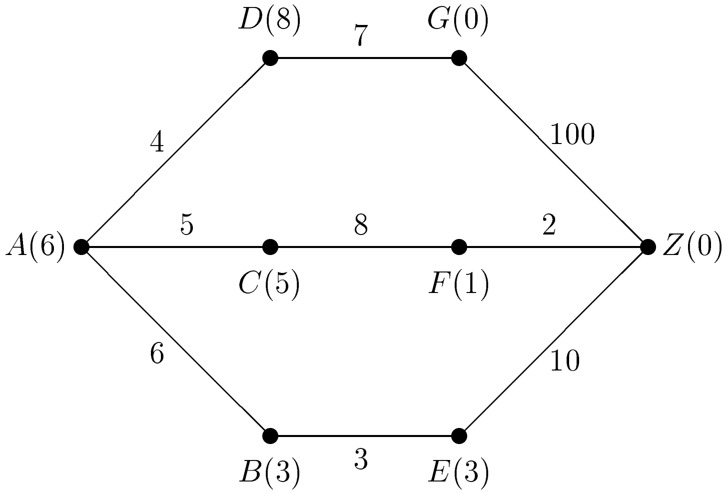
Tentamen Kunstmatige intelligentie
Universiteit Leiden — Informatica
woensdag 5 juni 2002, 13.00–16.00 uur

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) Noem een situatie waarin IDA* bijzonder slecht presteert.
c. (4 punten) Wat is de *pathmax equation* en waarvoor wordt deze gebruikt?
d. (8 punten) Voer zowel het A*-algoritme als het IDA*-algoritme uit voor onderstaande graaf. Gebruik zonodig de pathmax equation. Beginknoop is A, doelknoop is Z. Bij de knopen staat tussen haakjes de (overigens toelaatbare) heuristische functie. De kostenfunctie staat naast de takken van de graaf. Geef duidelijk aan hoe de algoritmes verlopen.



Opgave 2. α-β-algoritme (20 punten)
We bekijken het volgende spel voor twee personen. Het speelt zich af op een drie bij drie bord met daarop de volgende getallen gerangschikt:

1	2	3
4	5	6
6	8	4

Het spel begint bij het middelste vakje met 5. De speler die aan de beurt is mag stoppen, en krijgt dan het getal uit het betreffende vakje. Anders wordt uit de vier directe buren (drie of twee bij randvakjes) eerlijk = random een vervolgvakje gekozen, en is de ander aan de beurt. De eerste speler wil een zo laag mogelijk getal, de tweede zo hoog mogelijk.
a. (5 punten) Geef de *spelboom* (= *game tree*) die hierbij hoort. Bedenk zelf een geschikte notatie voor "kansknopen". Neem aan dat beide spelers maximaal één maal aan de beurt komen.
b. (5 punten) Beschrijf in woorden het *minimax-algoritme.*
c. (5 punten) Voer dit uit voor de spelboom van **a**, inclusief evaluaties van kansknopen.
d. (5 punten) Nu mag de speler *zelf* bepalen met welk buurvakje het spel verder gaat. 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. Belief netwerken (20 punten)

a. (5 punten) Of we gaan TV-kijken hangt af van het weer (mooi/gaat/slecht) en of we het druk hebben (ja/nee). Teken het bijbehorende *belief netwerk*, zowel voor iemand die in een strandtent werkt als voor een ambtenaar. Zorg voor twee verschilende netwerken, en maak de keuzes aannemelijk. Welke kanstabellen moeten gegeven worden?
b. (5 punten) Er zijn vier soorten queries; noem deze en leg ze kort uit.
c. (5 punten) Geef van elke soort query een voorbeeld uit een van de netwerken van **a**.
d. (5 punten) In een verre toekomst hangen voor een ambtenaar zowel het weer als het feit dat we het druk hebben af van het humeur van de leider. Maak allereerst het bijbehorende belief netwerk. Leg uit wat de “cutset conditioning methode” is, hoe deze werkt aan de hand van dit netwerk, en waarvoor deze in het algemeen gebruikt wordt.

Opgave 4. ID3 (20 punten)

We hebben een database waarin een aantal voorbeelden aan de hand van een stel attributen beschreven worden; verder hebben we een Ja/Nee-classificatie voor deze voorbeelden.
a. (10 punten) Leg uit hoe het *ID3-algoritme* werkt. Geef hierbij de definitie van de entropie $I(p/(n+p), n/(n+p))$ (p is het aantal positieve voorbeelden, n het aantal negatieve). Geef met name aan wat er moet (kan) gebeuren in de *vier* verschillende “standaard”-gevallen, bijvoorbeeld als er bij het splitsen geen voorbeelden meer zijn.
b. (5 punten) De volgende database geeft situaties uit het verleden aan. We willen een beslissingsboom maken om te kijken of we al of niet gaan TV-kijken, op grond van het weer en of we het druk hebben. Maak de twee mogelijke beslissingsbomen met hoogte 2.

dag	weer	druk	TV-kijken
1	mooi	ja	nee
2	slecht	nee	ja
3	gaat	nee	ja
4	mooi	ja	nee
5	gaat	ja	nee
6	slecht	ja	ja

c. (5 punten) Reken voor de boom met als wortelvraag de vraag naar het weer, de entropiewinst (“gain”) uit voor deze eerste vraag.

Opgave 5. Neurale netwerken (20 punten)

Stel we maken een *neuraal netwerk* met één invoerlaag en één uitvoerlaag, en geen verborgen lagen.
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. (7 punten) Leid met “gradient descend” de leerregel voor gewichten $W_{j,i}$ van invoerlaag naar uitvoerlaag af uit de errorfunctie $E = 1/2 \sum_i (T_i - O_i)^2$. 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 de g-functie los te laten op in_i, de met de $W_{j,i}$'s gewogen som van zijn invoeren. De j-de invoer is I_j.
d. (3 punten) Wat is *overfitting*?