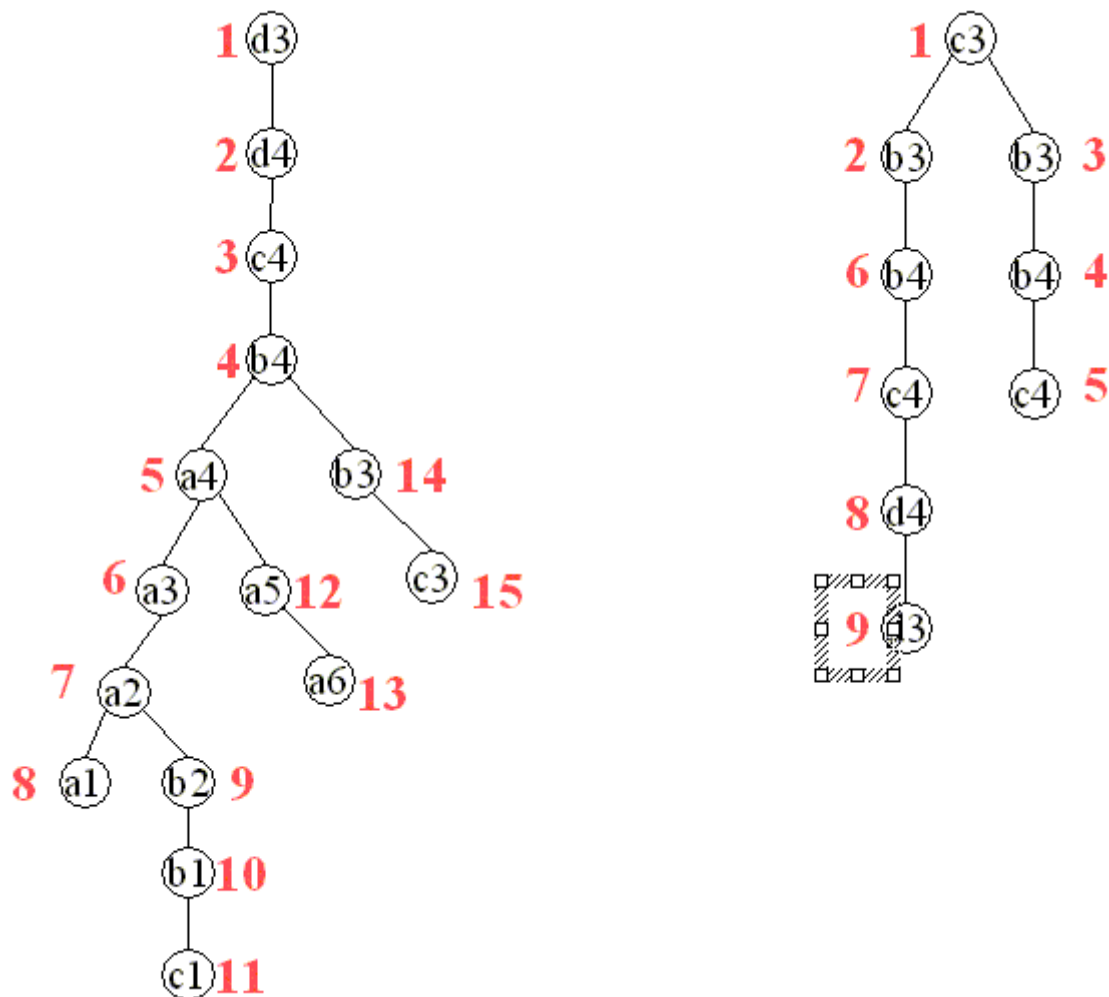


Antwoorden Deeltentamen 1, AI Kaleidoscoop, 24 Oktober 2001

Vraag 1:



Vraag 1a links, vraag 2a rechts; rode getallen geven volgorde aan.

Vraag 2:

(a) Een zoekalgorithme is admissible als het (1) altijd een antwoord vindt (mits er een antwoord bestaat, natuurlijk), en (2) het altijd het antwoord met het kortste zoekpad als eerste vindt.

(b) Depth-first (bijv.) is niet admissible, want (1) het vindt niet altijd een antwoord (bijv. als de zoekboom oneindige takken bevat), en (2) het vindt niet altijd het antwoords met het kortste pad als eerste (als er eerder in de boom diepere antwoorden te vinden zijn).

(c) Voordeel: depth-first iterative deepening heeft veel minder geheugen ruimte nodig (b^d in plaats van b^d).

Nadeel: depth-first iterative deepening doet werk dubbel, en daardoor is de rekentijd (iets) hoger dan die van breadth-first.

(d) We mogen A de naam A^* geven als voor alle knopen geldt $h(n) \leq h^*(n)$, waarbij $h^*(n)$ de echte afstand tot het doel is, en $h(n)$ de geschatte (heuristische) afstand. Met andere woorden, als de heuristisch geschatte afstand altijd een onderschatting is van de echte afstand. Het voordeel is dat A^* admissible is, en A niet.

(e) Beamsearch met oneindige beambreedte komt overeen met best-first search. Beamsearch is zelfs gedefinieerd als best-first search met een max. aantal elementen in de verzameling open knopen. Als die verzameling willekeurig groot mag zijn hebben we precies best-first search.

Vraag 3:

(a) Idealiter zouden we de evaluatie-functie toepassen op de bladeren van de boom, alvorens deze waardes omhoog door te geven via het minimax algoritme. Echter, in het algemeen is het niet mogelijk om de boom geheel tot aan de bladeren af te lopen. Daarom lopen we de boom slechts af tot aan een tussenliggend nivo, en passen we daar de evaluatie functie toe. Die functie is dan een *schatting* van de echte waarde van die knoop. Het horizon effect wordt veroorzaakt doordat we die schatting moeten baseren op de situatie in de tussenliggende knopen, terwijl "net iets voorbij" dat tussenliggende nivo informatie kan liggen die onze schatting sterk zou veranderen (maar die we niet kunnen zien, omdat we de boom niet verder kunnen aflopen). Het gevolg is dat de waardes die door minimax omhoog worden doorgegeven mogelijkterwijs fout zijn.

(b) alpha-beta pruning heeft ook te leiden van het horizon effect, want ook bij alpha-beta pruning kan de boom niet helemaal worden afgelopen. Echter, het effect zal minder zijn, omdat bij alpha-beta pruning de boom verder kan worden afgelopen, en de schattingen dus in het algemeen beter zullen zijn.

Vraag 4:

(a)

```
R1: CF(rain tomorrow)= 0.5 * 1.0 = 0.5
R2: CF(dry today) = -1.0 (want CF(rain today)=1.0)
    CF(dry tomorrow) = -1.0 * 0.5 = -0.5
R3: CF(dry tomorrow) = min(1.0, 0.8) * 0.6 = 0.48
R4: CF(dry tomorrow) = min(1.0, 0.8, 0.9) * 0.7 = 0.56
R5: CF(rain tomorrow) = min(-1.0, -0.9) * 0.65 = -0.65
    (want CF(temp is warm) = -CF(temp is cold))
R6: CF(rain tomorrow) = min(-1.0, -0.9, 0) * 0.55 = -0.55
    (CF(sky is overcast)=0 want niets bekend)
```

Voor CF(rain tomorrow), combineer 0.5, -0.65 en -0.55.

Voor CF(dry tomorrow), combineer -0.5, 0.48 en 0.56.

Voor de combinaties:

1. combinatie van de regels voor twee CF's van gelijke tekens: $X+Y-X*Y$ met de regel voor twee CF's van ongelijk teken
2. twee keer toepassen van de regels voor twee CF's van ongelijke tekens

(b) Bronnen van onzekerheid zijn:

- incomplete domein theorie
- incorrecte domein theorie
- vage begrippen
- complete theorie wel beschikbaar maar te duur of te inefficiënt
- Data zijn niet precies bekend
- Data zijn helemaal niet bekend

Vraag 5:

(a)

- Match conditie deel van alle regels tegen inhoud van het werkgeheugen
Alle matchende regels vormen tezamen de conflict set

- Selecteer een van de regels uit de conflict set (deze stap heet conflict resolutie)
- Executeer het actie deel van de geselecteerde regel (deze stap verandert de inhoud van het werkgeheugen)
- Herhaal totdat er geen regel meer toepasbaar is (= de conflict set is leeg), of totdat een gewenst element aanwezig is in het werkgeheugen.

(b)

- Toestand = inhoud van het werkgeheugen
- begintoestand = inhoud van het initiële werkgeheugen
- eindtoestand = werkgeheugen met gewenst element
- toestandsovergang = elke productie regel definieert een mogelijke toestandsovergang.

(c) De branchingfactor komt overeen met de omvang van de conflict set (= het aantal toepasbare regels in een gegeven toestand)