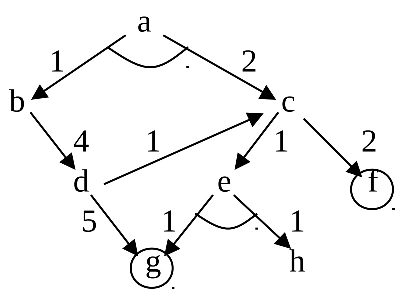
Osnove umetne inteligence, izpit 9. februar 2017

Navodila:
Čas: 80 min. Uporaba literature, zapiskov in elektronskih naprav ni dovoljena.
Točkovanje: vse naloge so enakovredne.
Ustni izpiti: Sreda, 15. februar ob 10h

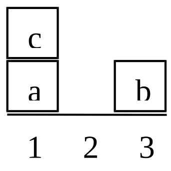
1.
(a) Primerjaj algoritma A* in IDA* glede na njuno časovno in prostorsko zahtevnost. Kaj so v tem pogledu prednosti enega in drugega algoritma? Zapiši tipični velikostni red teh zahtevnosti obeh algoritmov za primer, da je prostor stanj binarno drevo in da algoritma preiskujeta do globine n.
(b) Primerjaj algoritma A* in IDA* glede na njuno sposobnost, da garantirano najdeta optimalno rešitev. Podaj pogoje, ki dajejo tako garancijo.
(c) Ali je možno, da za nek prostor stanj in neko hevristično funkcijo IDA* vrne boljšo rešitev kot A*? Na kratko utemelji odgovor.
(d) Imejmo prostor stanj, ki ima obliko drevesa z uniformnim vejanjem b. Problem rešujemo z algoritmom za preiskovanje v širino. Globino vozlišča štejemo kot oddaljenost od startnega vozlišča. Torej je startno vozlišče na globini 0, vozlišče dva koraka od starta je na globini 2 itd. Ko poženemo naš program, nam ta po 2 sekundah sporoči, da je preiskal vsa vozlišča do globine 4. Po 200 sekundah program sporoči, da je pregledal vse do globine 6 in še vedno ni našel nobene rešitve. Oceni, kako dolgo moramo pustiti teči program, da bo zagotovo našel rešitev, če obstaja rešitev dolžine 9 ali manj.

2.
(a) Predstavitev problemov z AND/OR grafi temelji na dekompoziciji problemov. Kaj je pogoj za primernost uporabe AND/OR grafov? Kako se glasi šibka verzija tega pogoja?
(b) Kako je definirana vzvratna hevristična ocena notranjega vozlišča v drevesu preiskovanja AND/OR grafa odvisno od tega, ali ima vozlišče AND naslednike ali OR naslednike? Napiši ustrezni formuli.
(c) Dan je naslednji AND/OR graf:



Ob povezavah so vpisane cene povezav. Vozlišči f in g sta ciljni vozlišči. Nariši vsa rešitvena drevesa in izračunaj njihove cene.
(d) Izračunaj h*(a) za ta AND/OR graf.

3. Dana je domena planiranja v svetu kock. Domena vsebje tri kocke a, b in c ter tri lokacije 1, 2, 3 na mizi. Možno stanje v tej domeni je:



Stanja so opisana s predikatoma:
c(X), kar pomeni, da je kocka ali lokacija X prosta (nima ničesar na sebi)
on(Y, Z), kar pomeni, da je kocka Y na kocki ali lokaciji Z
Gornje stanje je definirano s seznamom relacij: **[c(c), c(b), c(2), on(a,1), on(b,3), on(c,a)]**

Definirana je akcijska shema **move(Block, From, To)** (prestavi kocko Block s From na To), kot običajno takole:

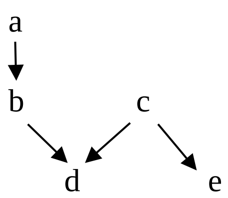
PREDPOGOJ: **[c(Block), c(To), on(Block, From)]**
ob omejitvah, da je **Block** kocka, **From** in **To** pa je kocka ali lokacija, ter **From** ≠ **To** in **To** ≠ **Block**
POZITIVNI UČINKI: **[on(Block,To), c(From)]**
NEGATIVNI UČINKI: **[on(Block,From), c(To)]**

(a) Kakšno je novo stanje, če na gornjem stanju izvedemo akcije **move(c,a,b)**?

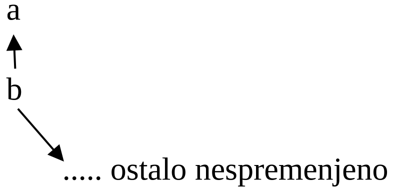
(b) Kaj je rezultat RG regresiranja ciljev **[on(a,b), on(b,c), c(1), c(3)]** skozi akcijo **move(a,1,b)**?

(c) Imejmo program za planiranje, ki uporablja regresiranje ciljev v zgoraj definirani domeni. Podaj množico vseh alternativnih akcij, ki jih bo ta planer upošteval za regresiranje množice ciljev { **on(a,b), on(b,c)**}.

4. Naj bo dana naslednja bayesovska mreža:



(a) Zapiši vse množice E, ki d-ločujejo dogodka a in e.
(b) Naj bodo za gornjo mrežo podane verjetnosti kot običajno: P(a), P(b|a), P(b|~a), ... Zanima nas pogojna verjetnost P(a|bc). Razvij formulo za izračun te verjetnosti, izražene z verjetnostmi, ki so že podane v mreži: P(a|bc) = ...?
(c) Denimo, da v gornji mreži zamenjamo smer puščice med a in b takole:



Drugih strukturnih sprememb v mreži ni. Ali je možno originalno mrežo ekvivalentno nadomestiti z novo mrežo, tako, da ustrezno podamo verjetnosti v novi mreži? Če da, katere verjetnosti bi bile v novi mreži na novo podane? Kako bi bile izražene z originalnimi verjetnostmi?

1. **[25] = [6] + [6] + [6] + [6] + [1]**

(a) A* je tipično hitrejši, IDA* potrebuje veliko manj prostora.

A*: čas in prostor $O(2^n)$

IDA*: čas $O(2^n)$, prostor O(n)

(b) A* je popoln, če uporablja optimistično funkcijo h, prav tako IDA*.

(c) Ja. IDA* lahko najde boljšo rešitev, če h ni optimistična.

(d) 2 sek = kb^4, 200 sek = kb^6; torej $b^2 = 100$, b = 10

Potrebni čas, da preišče vse do vključno globine 9 je več kot $kb^9 = kb^6 * b^3$ = 200.000s ~ 55 ur

2. **[25] = [6] + [6] + [6] + [6] + [1]**

(a) Pogoj: konjunktivno povezani podproblemi so neodvisni.

Šibka verzija pogoja: Obstaja vrstni red konjunktivnih podproblemov,

tako da jih v tem vrstnem redu lahko resujemo neodvisno enega od drugega.

(b) Naj ima N naslednike N1, N2, ...

AND nasledniki: H(N) = SUM[i] (cost(N,Ni) + H(Ni))

OR nasledniki: H(N) = MIN[i] (cost(N,Ni) + H(Ni))

(c) Dve rešitveni drevesi.

(d) h*(a) = 12

3. **[25] = [8] + [8] + [8] + [1]**

(a) **[c(c), c(2), on(a,1), on(b,3), c(a), on(c,b)]**

(b) **[on(b,c), c(3), on(a,1), c(a), c(b)]**

(c) Vse akcije oblike move(a,X,b) (X=1,2,3,c) in

move(b,Y,c) (Y=1,2,3,a), torej vsega 8 akcij.

4. **[25] = [8] + [8] + [8] + [1]**

(a) E je lahko katerakoli od 7 množic: {}, {b}, {c}, {b,c}, {b,d}, {cd}, {b,c,d}

(b) P(a | bc) = P(a|b) (ker sta a in c neodvisna ob danem b) =

= P(a) * P(b|a) / P(b), kjer je P(b) = P(a)P(b|a) + (1-P(a))*P(b|~a)

(c) Da. Nove vrednosti (v novi mreži) so (zapisane kot P'):

P'(b) = P(a)P(b|a) + (1-P(a))*P(b|~a)

P'(a|b) = P(a) * P(b|a)/P'(b)

P'(a|~b) = P(a) * (1-P(b|a))/(1-P'(b))