

Subiecte Examen sesiune 2012

IA – Adina Florea

18 ian 2012 - Nr1

Subiectele de teorie au fost cele de anul trecut (IA Nr1 18-01-2011), cel puțin la numărul meu.

La probleme:

- la primul subiect trebuia să aplici rezoluția. -> 15 pct
- la 2 nu-mi aduc aminte -> 15 pct
- la 3 de calculat h și g pentru A^* -> 10 pct
- la 4 o problemă destul de simplă în scheme -> 10 pct

18 ian 2012 – Nr2

Și la celălalt număr au fost la teorie subiectele de anul trecut (Nr2)

Iar la probleme:

1. a) forma clauzala b) rezoluția, graf, să explici pașii ..(ca în ani trecuți)
2. un minimax pentru algoritmul cu jetoane: a) construit grafic + parcurgere minimax iar la b) și c) algoritmul în scheme + ceva modificări/optimizări
3. subiect de învățare automată din exemple - să aplici algoritmul cu disjuncții parca (se dau niște date de intrare -> ce concept se învață + pașii algoritmului)
4. o problemă de implementat în scheme sau prolog (N copii sunt așezați în cerc. la fiecare parcurgere ieșe din cerc al M-lea copil. câștigă cel care rămâne ultimul. Se dau N, M și numele copiilor)

30 ian 2012 – Nr1

Teorie:

1. Hill Climbing First Choice, Hill Climbing Random Restart, Hill Climbing Stochastic

2. Diferențele între realizarea arc-consistenței și d-arc-consistenței.

3. Slot-machine cu 3 sloturi independente. Fiecare slot putea avea valori A, B, C, D cu aceeași probabilitate.

dacă ieșea A|A|A primeai 15 fise

B|B|B 3 fise

C|C|C 2 fise

D|_|_ 2 fise (decă bagi o fisa, câștigi o fisa, ești pe zero)

D|D|_ 5 fise

D|D|D 7 fise

orice alt rezultat înseamnă că pierzi fisa bună

Care era probabilitatea că bagand o fisa sau câștigi ceva?

Care era probabilitatea că bagand o fisa să câștigi fix 2 fise?

4. Să se transforme în formă clauzala:

oricare X, oricare Y (stramos(x,y) -> parinte(x,z) și stramos(z,y))

5. arhitectura generală BDI

6. se dău regulile și faptele DCG pentru îmbogățirea cu caz subiectiv să se transforme în Prolog

7. Problema cadrului, calificării, ramificației. Cum se rezolvă în planificarea liniară și cum în Prolog?

8. Distanța și distanța inferențială. Când e bine să se aplice una, când alta.

Probleme:

1. Se dău mai multe afirmații:

oricare x, oricare y: $\text{munte}(x) \wedge \text{munte}(y) \rightarrow \text{prieten}(x,y)$

oricare x, oricare y: $\sim \text{munte}(x) \wedge \sim \text{munte}(y) \rightarrow \text{prieten}(x,y)$

oricare x, oricare y: prieten(x,y) -> prieten(y,x)

munte(marcu) XOR munte(vlad)

~prieten(vlad, gelu)

munte(gelu)

Se cerea sa se demonstreze teorema prin respingere rezolutiva:

prieten(gelu, marcu)

2. Se dau 12 monezi notate cu litere mari diferite: M I T D A N C E F O L K

Una din monezi e falsa (e mai grea sau mai usoara, nu se stie cum).

Sa se faca un program Prolog care sa afle care este moneda falsa.

Exista o singura solutie?

3. Problema cu lupul si iepurele.

Ei muta alternativ pe un grid NxN.

Sa se genereze in pseudocod si apoi sa se implementeze in Scheme strategia lupului.

Iepurele muta random. Lupul vrea sa-l manance.

4. Se dadeau mai multe exemplu cu + si -

Sa se aplice tehnica clusterizarii, aia cu multimi disjuncte.

Conceptul este unic?

30 ian 2012 – Nr2

Teorie:

1. Ce se invata in algoritmul Real Time A* ? De ce se invata?

2. Comparati strategia Mini-Max cu strategia Alfa-Beta.

3. Transformati in forma cauzala oricare X oricare Y (cumpara(x, y) -> (ceva(y, x) ^ exista z pret(y, z)))

4. Dezavantajele planificarii liniare. Anomalia lui Sussman si cum se poate rezolva.

5. Aceeasi problema cu slot machine, dar se cerea sa se estimeze numarul de jocuri dupa care ramai cu 0 fise, pornind initial cu 10 fise.

6. Cum se construiește o clasa in OWL.

7. Se dadeau niste reguli DCG si sa se scrie in prolog, imbogatita cu semantici asociate.

8. Descrieti arhitectura de subsumare a agentilor reactivi.

Probleme:

1. Se dadeau mai multe axiome si sa se demonstreze o teorema cu respingere rezolutiva

2. Sa se scrie niste predicate in prolog care eliminau negatii dintr-o formula logica data sub forma de lista [Operand1, operator, Operand2] sau [not, Operand].

Formule de tipul [not [not Operand]] sa le transforme in Operand; eliminarea implicatiei; si ceva DeMorgan. Daca isi aminteste cineva mai clar ce si cum aici, share.

3. Un agent porneste de la pozitia (1,1) intr-un grid NxN si cunoaste pozitiile unor celule unde se afla aur. Scopul lui e sa adune tot aurul din grid. Miscarile sunt nedeterministe: 80% se duce in directia in care doreste, 20% nu se intampla nimic. Cunoaste tot timpul pozitia pe care se afla.

Cerinte: scrieti in pseudocod algoritmul prin care agentul aduna tot aurul din grid. Implementare scheme.

4. Se dadeau mai multe seturi de date cu attribute si clasificate + sau -. Se cerea sa se determine inaltimea maxima si respectiv minima a arborelui de decizie ce se construiește pornind se la setul de date si sa se deseneze si arborii respectivi.