

Teme Proiect Prolog

1. Problema canibalilor

- 3 misionari și 3 canibali se află la marginea unui râu, cu scopul de a trece pe celălalt mal.
- Ei au la dispoziție o barcă de două persoane.
- Dacă la un moment dat, pe un mal sau pe celălalt numărul canibalilor întrece pe cel al misionarilor, misionarii sînt în pericol de a fi mâncați de canibali.
- Problema constă în a afla cum pot trece râul cele 6 persoane în deplină siguranță.

Indiciu implementare:

- puteți folosi un obiect `stare(nr_canib_mal_stg, nr_canib_mal_dr, nr_misionari_mal_stg, nr_misionari_mal_dr)` pentru a reprezenta domeniul problemei.
 - un predicat `in_siguranta(stare(CL1,CD1,ML1,MD1), stare(CL2,CD2,ML2,MD2))` care verifică dacă `nr_canibal < nr_misionar` pe cele doua maluri, la un anumit moment
 - un predicat `traverseaza(stare(CL1,CD1,ML1,MD1), stare(CL2,CD2,ML2,MD2))` care actualizeaza `nr_canibal` si `nr_misionari` dupa o traversare (la dreapta sau la stînga) cu verificarea condiției `in_siguranta/2`
 - un predicat `traseu(lista_traversari)` cu ajutorul căruia să va rețineți traversările (fiecare traversare va fi adăugată la lista rezultat).
 - veți apela `traseu(stare(3,3,stanga,0,0),stare(0,0,dreapta,3,3),[stare(3,3,left,0,0)])`

2. Lupul Capra și Varza

Un țăran la târg plecase

Și de vânzare luase

Un lup, un ied și o varză.

Nvrând nici una să piarză

Și nefiind nici călare,

Vrea să treacă un râu mare,

Care era să-l înoate

Și să le treacă pe toate.

Stând în loc, se socotește

Și întru sine șoptește,

Cum și în ce chip să facă

Câte una să le treacă,

Că fiind apa prea lată,
Nu putea două dodată.
"Să trec întâi lupul, zice,
Capra varza o să-mi strice,
Să trec varza, ș-așa încă,
Lupul capra îmi mănâncă."
Deci dacă-i veni în minte
Și trecu capra nainte,
Stătu iar să să gândească
În ce chip s-o nemerească.
Gândind, zicea întru sine:
"Trecui una, merse bine,
Pân-aci toate scăpară;
Acum care să trec dară?
Trecând varza și lăsând-o,
O strică iedul rozând-o,
Precum lupul și el iară
Îmi face iedul papară.
O, ce vită neunită
Și marfă nepotrivită!"
Dar mai gândind: "Ha! el zise,
Nevoia minte-mi trimise."
Trecu lupul, clătind capul,
Ș-întoarse înapoi țapul.
Trecu și varza îndată,
Mereu făcând judecată.
Și mergând a doua oară,
Trecu țapul subsuoară.
Omul dacă să gândește,
Orce i să înlesnește,
Prejudecând cele grele,
Le găsește ușurele,
Că pe cât el să gândește,
P-atât mintea i s-ascuțește.
Și orce, cu judecată,

3. Generarea unui limbaj de mici dimensiuni - limba română

Pornind de la o gramatică (un set de simboluri numiți terminali, un set de simboluri numiți neterminali, o colecție de reguli, fiecare arătând cum poate fi expandată o categorie compusă în subcompuși și un simbol de start). Se dorește generarea unei exprimări corecte gramatical, în limba română.

4. Generarea unui limbaj de mici dimensiuni - limba engleză

Pornind de la o gramatică (un set de simboluri numiți terminali, un set de simboluri numiți neterminali, o colecție de reguli, fiecare arătând cum poate fi expandată o categorie compusă în subcompuși și un simbol de start). Se dorește generarea unei exprimări corecte gramatical, în limba engleză.

5. Enigma lui Einstein

Există cinci case.

1. Englezul locuiește în casa roșie.
 2. Spaniolul are un câine.
 3. Se bea cafea în casa verde.
 4. Ucraineanul bea ceai.
 5. Casa verde este imediat la dreapta casei de fildeș (casa albă).
 6. Fumătorul de Old Gold are melci.
 7. Se fumează Kools în casa galbenă.
 8. Laptele se bea în casa din mijloc.
 9. Norvegianul locuiește în prima casă.
 10. Omul care fumează Chesterfields locuiește în casa de lângă cea cu vulpea.
 11. Kools se fumează în casa de lângă cea în care se ține calul.
 12. Fumătorul de Lucky Strike bea suc de portocale.
 13. Japonezul fumează Parliaments.
 14. Norvegianul locuiește lângă casa albastră.
- În acest puzzle, trebuie să deduceți cine bea apă și cine deține zebra

6. Sistem Expert - Recomandare carieră

Un sistem expert în Prolog pentru recomandări de carieră poate fi dezvoltat pentru a ajuta persoanele să exploreze și să aleagă cariere care se potrivesc cu abilitățile, interesele și obiectivele lor profesionale. Un exemplu de predicate ce pot fi folosite:

- Definirea bazelor de cunoștințe despre cariere: `cariere(programator, dezvoltare_software, it).`
- Abilitățile necesare pentru fiecare carieră: `abilitati(programator, [c++, java, dezvoltare_web]).`
- Interesele asociate cu fiecare carieră: `interese(programator, [tehnologie, rezolvare_probleme, inovatie]).`
- Recomandare de carieră bazată pe abilități și interese Exemplu interogare:
`recomandare_cariera([dezvoltare_web, inovatie], [tehnologie, rezolvare_probleme],
Recomandare).`

7. Sistem expert - Recomandare rețetă culinară

Un sistem expert pentru recomandări de rețete culinare poate ajuta utilizatorii să descopere rețete delicioase și să aleagă ce să gătească în funcție de preferințele lor, ingredientele disponibile și restricțiile dietetice. Un exemplu de predicate ce pot fi folosite:

- Baza de cunoștințe cu rețete și ingrediente `ingredient(oua).`
- Baza de cunoștințe cu rețete culinare `reteta(omleta, [oua, lapte, ceapa, unt]).`
- Baza de cunoștințe cu rețete culinare `reteta(omleta, [oua, lapte, ceapa, unt]).`
- Regulă pentru recomandarea de rețete bazată pe ingredientele disponibile
`recomanda_reteta(IngredienteDisponibile, RetetaRecomandata) :-`
- Regulă pentru recomandarea de rețete bazată pe preferințele alimentare
`recomanda_reteta_cu_preferinte(IngredienteDisponibile, Preferinte, RetetaRecomandata) :-`
- Regulă pentru a afișa lista de ingrediente disponibile
`afiseaza_ingredient_disponibile(IngredienteDisponibile) :- write('Ingrediente disponibile:
'), write(IngredienteDisponibile), nl.`
- Exemplu de utilizare `IngredienteDisponibile = [oua, ceapa, rosii],`
`recomanda_reteta(IngredienteDisponibile, RetetaRecomandata),`
`afiseaza_ingredient_disponibile(IngredienteDisponibile),`
`write('Puteți găti: '), write(RetetaRecomandata), nl.`

8 Sistem Expert - Recomandare Ținute vestimentare

Un sistem expert pentru recomandări de ținute vestimentare poate ajuta utilizatorii să aleagă haine potrivite în funcție de diferite factori, cum ar fi vremea, evenimentul, stilul personal și preferințele lor. Un exemplu de structură de baze de cunoștințe este prezentat mai jos:

- Baza de cunoștințe cu tipuri de haine `haina(pulover).`
- Baza de cunoștințe cu stiluri vestimentare `stil(casual).`
- Baza de cunoștințe cu recomandări de ținute pentru diverse situații `recomandare_ținuta(vreme_rece, casual) :- haina(pulover), haina(pantaloni).`
- Regulă pentru recomandarea de ținute bazată pe situația specifică `recomanda_ținuta(Situatie, StilRecomandat) :-`

- Exemplu de utilizare `recomanda_ținuta(vreme_rece, StilRecomandat),`
`recomanda_ținuta(vreme_cald, StilRecomandat).`

Empty markdown cell. Double click to edit