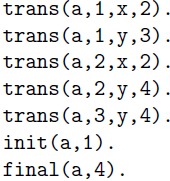
**Prolog, zadaci za vježbu**

1. Konačni automat može da se predstavi sa sljedećim skupom pravila:
   1. **trans(a, s1, alpha, s2)** predstavlja prelazak iz stanja **s1** u stanje **s2** za ulaz **alpha** u automatu **a**
   2. **init(a, s)** predstavlja početno stanje **s** u automatu **a**
   3. **final(a, s)** predstavlja završno stanje **s** u automatu **a**

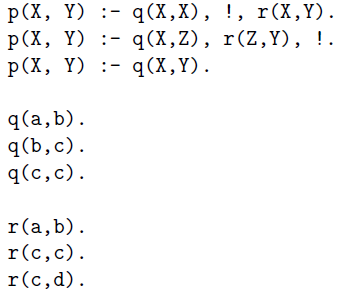
Sljedeći skup pravila pedstavlja jedan konačni automat.



Uvodi se pretpostavka da konačni automat ima samo jedno početno i završno stanje i da nema ε prelazaka.

* Napisati predikat **accepts(A, L)** koji za dati konačni automat **A** i listu simbola **L** uspijeva akko automat prihvata datu listu simbola. Na primjer za automat sa prethodne slike **accepts(a, [x, x, y])** uspijeva a **accepts(a, [x, x])** i **accepts(a, [y, y, x])** ne uspijeva.
* Neka je **L** jezik određen konačnim automatom. Kažemo da je automat djelimično palindromski ako postoji neki string s∈L takav da je sR∈L gdje je sR inverzni string za s. Automat sa prethodne slike je djelimično palindromski jer prihvata string **yy**. Napisati predikat **patrial(A)** koji uspijeva akko je automat **A** djelimično palindromski.

1. Napisati PROLOG program strangeSort koji sortira listu pozitivnih cijelih brojeva na sljedeći način: prvi element u rezultujućoj listi je najmanji element iz originalne liste, drugi element u rezultujućoj listi je najveći elemet iz originalne liste, treći element rezultujuće liste je prvi po veličini poslije minimalnog iz originalne liste, četvrti je pretposljednji po veličini iz originalne liste itd. Na primjer, ako je data lista [3, 6, 1, 2, 4, 7, 5] program treba da generiše listu [1, 7, 2, 6, 3, 5, 4].
2. Ternarno stablo je sastavljeno od termova n(a, b, c) koji se nazivaju čvorovi, gdje su a, b, i c ili čvorovi ili cijeli brojevi. Pretpostavite da su dozvoljeni cijeli brojevi 0 i 1. Napisati PROLOG predikat koji će da vrati listu svih 0 i listu svih 1 u datom drvetu. Na primjer, enum(n(n(0, 1, 0), 1, 0), X, Y) treba da postavi X na [0, 0, 0], a Y na [1, 1]. Obavezno koristiti dvije liste.
3. Razmatra se zadatak sa slike. Koliko odgovora generiše PROLOG interpreter za upit  p(b, X)? Objasniti. Koliko odgovora generiše PROLOG interpreter za upit  p(c, X)? Objasniti.



1. Binarnim stablom se naziva stablo u kojem svaki čvor sadrži tačno dva potomka: lijevo i desno podstablo. Čvor koji nije ničiji potomak naziva se korijen. Čvor koji nema potomke naziva se list. Binarno stablo se u PROLOG-u na sljedeći način:

* Prazno stablo je binarno stablo i označava se sa **null**
* Binarno stablo je struktura **t(N, L, R)**, gdje je **N** čvor a **L** i **R** su lijevo i desno podstablo binarnog stabla.

Primjer: Struktura **t(a, t(b, null,null), t(c,null,null))** predstavlja drvo sa sljedeće slike.

a

b

c

Implementirati sljedeće predikate:

1. **isBinaryTree(T)** – **T** je binarno drvo;
2. **count(T, C)** – **C** je broj čvorova u drvetu **T**;
3. **preorder(T, L)** – **L** je lista čvorova drveta **T** u preorder poretku;
4. **leaves(T, L)** – **L** je lista listova iz drveta **T**. Redosled je zadržan, tj. listovi se upisuju u listu sa lijeva u desno; za drvo sa gornje slike **L = [b, c]**;
5. **equal(T1, T2)** **--** **T1** i **T2** su jednaka drveta;
6. **postorder(T, L)** -- **L** je lista čvorova drveta **T** u postorder poretku;
7. **height(T, H)** – **H** je visina binarnog drveta **T**;
8. **in\_tree(X, T)** – **X** je element drveta binarnog pretraživanja **T**;
9. **insert(X, T1, T2)** – **T2** je drvo binarnog pretraživanja koje se dobija dodavanjem elementa **X** u drvo binarnog pretraživanja **T2**;
10. **delete(X, T1, T2)** – **T2** je drvo binarnog pretraživanja koje se dobija brisanjem elementa **X** iz drveta binarnog pretraživanja **T2**;
11. Ternarno stablo je sastavljeno od termova n(a, b, c) koji se nazivaju *čvorovi*, gdje su *a*, *b*, i *c* ili čvorovi ili cijeli brojevi. Pretpostavite da su dozvoljeni cijeli brojevi 0 i 1. Napisati PROLOG predikat koji će da vrati listu svih 0 i listu svih 1 u datom drvetu. Na primjer, enum(n(n(0, 1, 0), 1, 0), X, Y) treba da postavi X na [0, 0, 0], a Y na [1, 1]. Obavezno koristiti dvije liste.
12. Napisati PROLOG program **split(Numbers, Positives, Negatives)**, koji razdvaja datu listu cijelih brojeva **Numbers** na dvije nove liste, listu pozitivnih brojeva (0 je uključena u pozitivne brojeva) **Positives** i listu negativnih brojeva **Negatives**. Na primjer, **split([3, -6, 5, 0, -2], Positives, Negatives)** je tačno ako je **Positives = [3, 5, 0]** i **Negatives = [-6, -2])**.
13. Napisati PROLOG predikat **disjoint(L1, L2)**, koji je tačan ako liste **L1** i **L2** nemaju zajedničih elemenata**.**