

Prevodioci i interpretatori - mart 2007. - praktični deo ispita

1. Posmatrajmo fragment programskog jezika PASCAL u kome je dopušteno korišćenje naredbi **while**, **if**, dodele brojevnih konstante promenljivoj i poređenje dve promenljive/konstante. Npr.

```
while a > 3 do
begin
  if a > 5 then
    a = 2;
  if a > b then
  begin
    tmp = a;
    a = b;
    b = tmp
  end
end
```

- (a) LL(1) gramatikom opisati opisani fragment jezika i odrediti skupove izbora.
- (b) Napisati PERL skript koji vršeci simulaciju potisnog automata naniže proverava da li program u datoteci čije se ime navodi kao prvi argument komandne linije pripada datom fragmentu jezika.
2. Implementirati sistem koji korisniku daje mogućnost davanja odgovora o svojstvima relacija definisanih na konačnim domenima. Konačni skupovi elemenata se definišu ključnom rečju **type** kao intervali **a..b**, ili navođenjem njihovih elemenata u vitičastim zagradama. Relacije se navode ključnom rečju **relation**, zatim navođenjem jednog ili više domena relacije i nakon toga definisanjem relacije u obliku skupa n-torki. Relacija = je automatski na raspolaganju. Pitanja o svojstvima relacije se postavljaju ključnom rečju **question**, a odgovori na njih se dobijaju ključnom rečju **answer**. Pitaja su rečenice logike prvog reda na datom relacijskom jeziku. Konjunkcija se označava operatorom **&**, disjunkcija operatorom **|**, implikacija operatorom **-->**, negacija operatorom **~**. Uz kvantifikatore se navodi kvantifikatorska promenljiva i njen domen. Univerzalni kvantifikator se označava sa **!(x::domain). formula**, a egzistencijalni sa **?(x::domain). formula**

```
> type int4 = 1..4;
    Type int4 defined as {1, 2, 3, 4}
> type odd4 = {1, 3};
    Type odd4 defined as {1, 3}
> relation R(int4, odd4) = {(1, 3), (2, 1), (3, 3), (4, 1)};
    Relation R defined
> relation S(int4, int4, int4) = {(1, 1, 1), (1, 2, 3), (3, 2, 1)};
    Relation S defined
> relation T(int4, odd4) = {(1, 2)};
    Error: T definition incorrect. Skipping.
> question 1R3 = R(1, 3);
    Question 1R3 defined
> answer 1R3;
    Answer: TRUE
> question questionAboutS = ~(S(1, 1, 1) | S(2, 1, 3));
    Question questionAboutS defined
> question isFunctionR = !(x::int4). !(y1::odd4). !(y2::odd4).
    (R(x, y1) & R(x, y2) --> y1 = y2);
    Question isFunctionR defined
> answer questionAboutS;
    Answer: FALSE
> answer isFunctionR;
    Answer: TRUE
> relation R(int4, odd4) = {(1, 1), (1, 3), (2, 1), (2, 3)};
    Relation R defined
> answer isFunctionR;
    Answer: FALSE
```

Predlog redosleda rešavanja zadatka:

- Odabrati strukturu podataka za reprezentovanje konačnih skupova i omogućiti njihovo definisanje (**type** direktivu).
- Definisati strukturu podataka za reprezentovanje relacije i omogućiti njihovo definisanje (**relation** direktivu). U početku preskočiti proveru korektnosti.
- Definisati strukturu podataka za reprezentovanje formula (pitanja) i omogućiti definisanje i odgovaranje (**question**, **answer**) isključivo na atomična pitanja ($R(x_1, \dots, x_n)$).
- Omogućiti pojavu operatora **&**, **|**, **-->** i **~** u pitanjima.
- Omogućiti pojavu kvantifikatora u pitanjima.