Prevođenje programskih jezika

Na Desktopu napraviti direktorijum sa nazivom **PPJ.sep1.ime.prezime.brojIndeksa.godinaUpisa** i u njemu sačuvati svoj rad. Dakle, ako **Pera Perić** sa indeksom **123/2015** polaže ispit potrebno je da na Desktopu napravi direktorijum sa nazivom **PPJ.sep1.Pera.Peric.123.2015** i u njemu da sačuva svoj rad. **Sintaksno neispravni zadaci se ne pregledaju. Vreme za izradu ispita je 3h. Uslov za polaganje ispita je minimum 22p.**

Sintaksna analiza naviše

Potrebno je napisati interpreter za zamišljeni programski jezik nad skupovima celih brojeva.

1. Interpreter treba da omogući jednostavno definisanje skupova nalik na C i njihovo štampanje. Dodatno, treba omogućiti definisanje univerzalnog skupa koji predstavlja nadskup svih skupova u programu.

```
def x;                  // definiše prazan skup x
def y = {1,2,3};        // definiše skup y sa tri elemente
print(x);               // na stdout štampa {}
print(y);               // na stdout štampa {1,2,3}
univ = [0, 10];         // definiše univerzalni skup
```

2. Interpreter treba da omogući sve uobičajene skupovne operacije nad skupovima i konstantama (/\ - presek, \/ - unija, \ - razlika, ~ - komplement koji se računa u odnosu na univerzalni skup). U slučaju da nije definisan univerzalni skup, komplement skupa je prazan skup.

```
x = x  \/ {3,4,5,6};
print (x /\  y);        // na stdout štampa {3}
print(~(y \ x));        // na stdout štampa {0,3,4,5,6,7,8,9,10}
```

3. Interpreter treba da omogući proveru pripadnosti elementa skupu (in), proveru da li je jedan skup podskup drugog (sub) i određivanje broja elemenata, tj. kardinalnosti skupa (size).

```
5 in x \ {4,5,6};       // na stdout štampa False
x sub ~y;               // na stdout štampa False
size({7,8}) in y;       // na stdout štampa True
```

4. Interpreter treba da omogući upoređivanje skupova (==, !=). Skupovi su jednaki ako sadrže iste elemente.

```
x \/ y == univ;         // na stdout štampa True/False u zavinosti od uslova
```

5. Modifikovati interpreter tako da rezultat interpretacije bude reprezentacija programa pomoću sintaksnog stabla. Sintaksno stablo treba da omogući interpretaciju, tj. izvršavanje programa i štampanje stabla na standardni izlaz.

Napomena: Makefile je obavazen deo rešenja.

Napomena 2: Zadaci 1-4 i zadatak 5 se odvojeno pregledaju. Zbog toga, rešenja zadataka 1-4 i zadatka 5 sačuvajte u odvojenim direktorijumima unutar svog direktorijuma. Na primer, rešenja zadataka 1-4 sačuvajte u direktorijumu „Interaktivni parser“, a rešenje zadatka 5 sačuvajte u direktorijumu „Sintaksno stablo“.