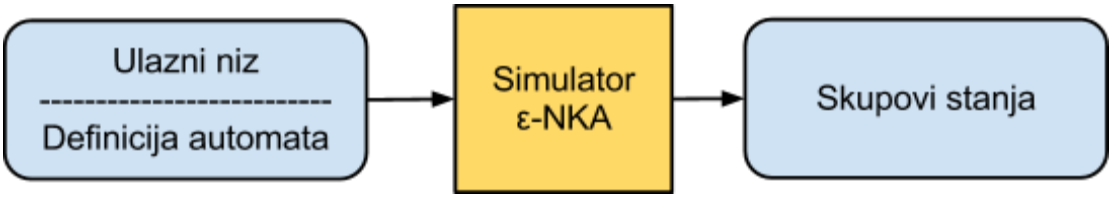
Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva
Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

UVOD U TEORIJU RAČUNARSTVA

Ak. God. 2011/2012

2. Laboratorijska vježba

U drugoj laboratorijskoj vježbi zadatak je programski ostvariti simulator nedeterminističkog konačnog automata s epsilon prijelazima (**ε-NKA**). Ulaz u simulator automata je tekstualni zapis njegove definicije i ulazni niz, a izlaz je tekstualni zapis skupova stanja u kojima se automat nalazio za svaki učitani znak ulaznog niza. Način rada programa koji je potrebno ostvariti u sklopu vježbe prikazan je načelno na slici 1.



Slika 1 - Načelni rad simulatora ε-NKA

Format za zapis ulaznog niza i definicije ε-NKA je:

- 1 redak: Ulazni nizovi odvojeni znakom |. Simboli svakog pojedinog niza odvojeni su zarezom.
- 2. redak: Leksikografski poredan skup stanja odvojenih zarezom.
- 3. redak: Leksikografski poredan skup simbola abecede odvojenih zarezom.
- 4. redak: Leksikografski poredan skup prihvatljivih stanja odvojenih zarezom.
- 5. redak: Početno stanje.
- 6. redak i svi ostali retci: Funkcija prijelaza u formatu trenutnoStanje,simbolAbecede->skupIdućihStanja. U skupu skupIdućihStanja stanja su odvojena zarezom. U definiciji epsilon-prijelaza, umjesto simbola abecede stajat će znak **$** (simbol **ε** nadomještava se simbolom **$**). Prijelazi su leksikografski poredani, promatrajući samo komponente prijelaza. Drugim riječima, prvo su navedeni prijelazi za leksikografski najmanje stanje, od leksikografski najmanjeg simbola do najvećeg, pa prijelazi za leksikografski drugo najmanje stanje, i tako dalje. Skup idućih stanja je prazan ukoliko nisu definirani odgovarajući prijelazi. Prazan skup stanja označava se simbolom **#**.

Definicija automata neće nikada imati više od 100 stanja niti više od 100 simbola abecede. Stanja i simboli abecede zadaju se kao nizovi malih slova engleske abecede i dekadskih znamenaka pri čemu je duljina niza minimalno 1 znak i maksimalno 20 znakova. Leksikografski poredak definiran je na osnovi odnosa ASCII vrijednosti znakova na krajnje lijevom mjestu na kojem se dva niza znakova razlikuju. Nadalje, ako je niz znakova A prefiks niza znakova B, onda je A leksikografski manji od B. Dekadske znamenke su u ASCII kodu manje od malih slova engleske abecede. Na primjer, vrijedi "xy" <"xyz", "xy" < "z", "1" < "a". Simbol $ je leksikografski najmanji, odnosno manji od svih brojki i slova. Ovaj redoslijed može se dobiti u svim podržanim jezicima ugrađenim operatorima/funkcijama za uspoređivanje nizova znakova.

U ulaznom zapisu, svaki redak završava znakom za kraj retka (\n). Ukoliko za par (stanje, ulazni_znak) nije definiran prijelaz u definiciji automata, smatrat će se da postoji prijelaz:

stanje,ulazni_znak->#.

Primjer definicije ε-NKA prikazan je na slici 2. Brojevi s lijeve strane slike označavaju retke i nisu dio

definicije.

```
01 a,pnp,a|pnp,lab2|pnp,a|pnp,lab2,utr,utr
02 p5,s3,s4,st6,stanje1,stanje2
03 a,lab2,pnp,utr
04 p5
05 stanje1
06 s3,a->stanje2
07 s3,lab2->p5,s4
08 s4,$->st6
09 s4,utr->p5,s3
10 stanje1,a->stanje2
11 stanje1,pnp->s3
12 stanje2,$->st6
13 stanje2,a->#
```

Slika 2 - Primjer definicije ε-NKA

Ostvareni simulator ε-NKA treba ispisati u kojem se skupu stanja nalazio automat za svaki ulazni znak pojedinog niza. Skupovi stanja odvojeni su znakom |, a stanja unutar svakog skupa su leksikografski poredana. Rezultati za svaki ulazni niz ispisuju se u zasebnom retku, odnosno svaki redak izlaza odvojen je znakom novog reda (\n). Svaki zapis započinje **skupom** koji sadržava početno stanje automata (**napomena:** skup može sadržavati i više stanja, ovisno o prijelazima automata). Primjer izlaza za automat i ulazne nizove definirane na slici 2 prikazan je na slici 3. Brojevi s lijeve strane slike označavaju retke i nisu dio izlaza.

```
01 stanje1|st6,stanje2|#|#
02 stanje1|s3|p5,s4,st6
03 stanje1|s3|st6,stanje2
04 stanje1|s3|p5,s4,st6|p5,s3|#
```

Slika 3 - Primjer izlaza simulatora automata

Napomene:
1) Prilikom ispisa izlaza simulatora potrebno je očuvati leksikografski poredak unutar svakog skupa stanja.

2) Nije potrebno provjeravati ispravnost formatiranja ulazne datoteke ili ispravnost rada automata. Drugim riječima, uvijek će biti definiran barem jedan ulazni niz, uvijek će postojati barem jedno stanje u skupu stanja i barem jedan simbol u skupu simbola abecede, uvijek će biti definirano točno jedno početno stanje (iako ne nužno prvo navedeno stanje iz skupa stanja). Definicija automata neće nužno biti potpuna, a u slučaju da za par (stanje, ulazni_znak) nije definiran prijelaz u definiciji automata, smatrati će se da postoji prijelaz: stanje,ulazni_znak->#. Nije nužno da će svaki automat imati prihvatljivih stanja. Neće biti preklapanja između skupa stanja i skupa simbola abecede.

3) Vremensko ograničenje na izvođenje programa za bilo koju ulaznu definiciju automata jest 10 sekundi

4) Ulazna točka za Java rješenja treba biti u razredu SimEnka, a ulazna točka u Python rješenja treba biti u datoteci SimEnka.py.