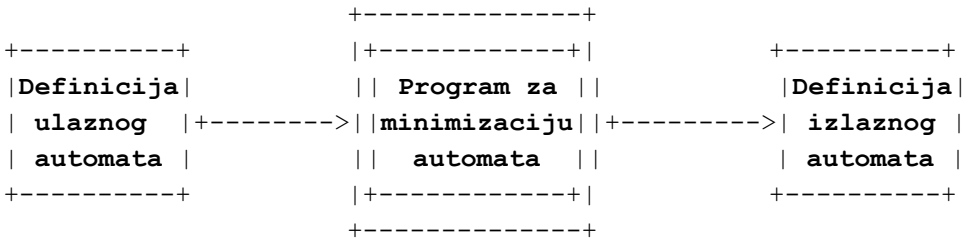
Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva
Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

UVOD U TEORIJU RAČUNARSTVA

Ak. God. 2018/2019

2. Laboratorijska vježba

Zadatak 2. laboratorijske vježbe je programsko ostvarenje minimizacije determinističkog konačnog automata (DKA). Programsko ostvarenje treba ostvariti uklanjanje nedohvatljivih stanja i uklanjanje istovjetnih stanja. Ulaz i izlaz programa za minimizaciju jesu tekstualne definicije DKA. Na slici 1 prikazan je načelan rad programa kojeg je potrebno ostvariti.



Slika 1 - Načelni rad programa za minimizaciju automata

Format za zapis definicije ulaznog i izlaznog automata jest sljedeći:

- 1. redak: Skup stanja odvojenih zarezom, leksikografski poredana.
- 2. redak: Skup simbola abecede odvojenih zarezom, leksikografski poredana.
- 3. redak: Skup prihvatljivih stanja odvojenih zarezom, leksikografski poredana.
- 4. redak: Početno stanje.
- 5. redak i svi ostali retci: Funkcija prijelaza u formatu trenutnoStanje,simbolAbecede->idućeStanje, pri čemu su prijelazi 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.

Definicija ulaznog 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". Ovaj redoslijed može se dobiti u svim podrzanim jezicima ugrađenim operatorima/funkcijama za uspoređivanje nizova znakova.

U definicij automata, svaki redak završava znakom za kraj retka (\n). Zadnji znak u svakoj definiciji automata također je znak kraja retka. Drugim riječima, kada ispisujete očekivani izlaz, nakon sadržaja svakog retka (pa tako i posljednjeg retka) treba staviti znak kraja retka.

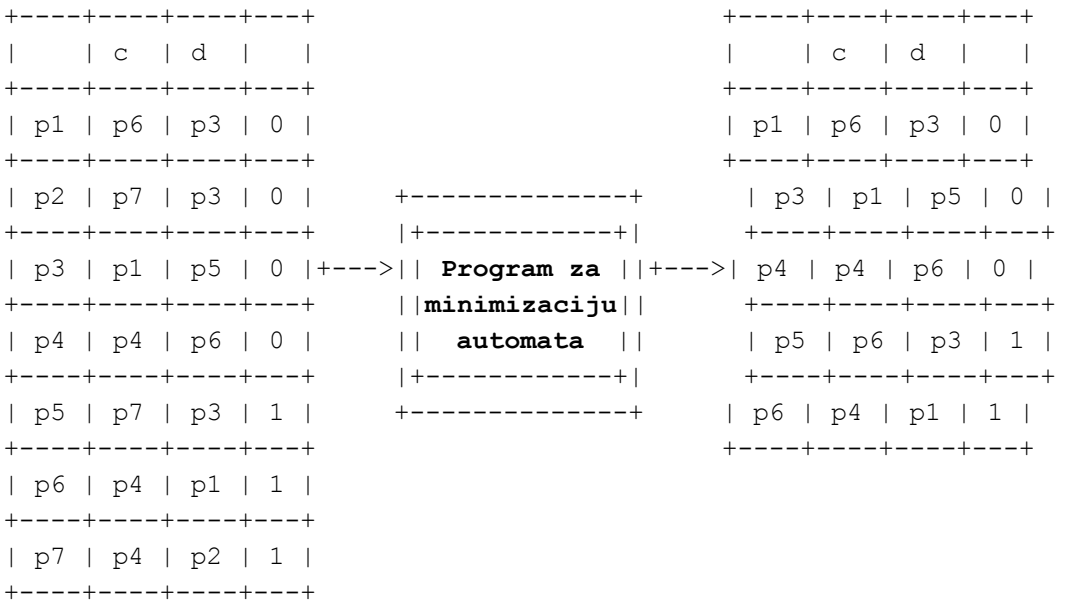
Ilustrativni, ali nepotpuni primjer definicije DKA prikazan je na slici 2. Brojevi s lijeve strane slike označavaju

retke i nisu dio definicije.

```
01 stanje1,stanje2,stanje3,stanje4,stanje5,q3,z
02 a,befg,cce,ddd
03 stanje2,stanje5,q3
04 stanje1
05 stanje1,a->stanje2
06 stanje3,befg->q3
...
N stanje5,ddd->stanje5
```

Slika 2 - Primjer definicije konačnog automata

Ostvareni program treba iz definicije automata ukloniti nedohvatljiva i istovjetna stanja te prijelaze vezane uz ta stanja. Pri tome, ako je skup stanja istovjetan, potrebno je u sačuvati leksikografski prvo stanje, dok ostala stanja treba ukloniti. Primjerice, ako su stanja stanje1 i stanje2 i q3 istovjetna, potrebno je ukloniti stanja stanje2 i q3, a sačuvati stanje stanje1. Na slici 3 prikazan je načelni rad programa za DKA zadan slikama 2.13 i 2.14 u udžbeniku, dok slike 4 i 5 prikazuju ulaz i očekivani izlaz programa.



Slika 3 - Načelni rad programa za minimizaciju automata za primjer zadan slikama 2.13 i 2.14 u udžbeniku

```
p1,p2,p3,p4,p5,p6,p7
c,d
p5,p6,p7
p1
p1,c->p6
p1,d->p3
p2,c->p7
p2,d->p3
p3,c->p1
p3,d->p5
p4,c->p4
p4,d->p6
p5,c->p7
p5,d->p3
p6,c->p4
p6,d->p1
```

```
p7,c->p4
p7,d->p2
```

Slika 4 - Definicija ulaznog automata zadanog slikom 2.13 u udžbeniku

```
p1,p3,p4,p5,p6
c,d
p5,p6
p1
p1,c->p6
p1,d->p3
p3,c->p1
p3,d->p5
p4,c->p4
p4,d->p6
p5,c->p6
p5,d->p3
p6,c->p4
p6,d->p1
```

Slika 5 - Očekivani izlaz programa za minimizaciju za definiciju automata sa slike 4

Napomene:

1) U definiciji izlaznog automata treba očuvati leksikografski poredak stanja, simbola abecede i prijelaza.

2) Nije potrebno provjeravati ispravnost formatiranja ulazne datoteke ili ispravnost rada automata. Drugim rijecima, 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), svi prijelazi za sva stanja bit će definirani, neće biti prijelaza za nedefinirana stanja ili nedefinirane simbole abecede i definicija će sigurno biti valjani DKA. Nije nužno da će svaki automat imati prihvatljivih stanja. Neće biti preklapanja između skupa stanja i skupa simbola abecede.

3) Nije nužno da postupci doslovno implementiraju neki od algoritama opisanih u udžbeniku, tj. rad algoritma se ne provjerava vec samo krajnji rezultat. Svejedno, nužno je da se prvo provede postupak uklanjanja nedohvatljivih stanja, a tek onda postupak uklanjanja istovjetnih stanja.

4) Vremensko ograničenje na izvođenje programa za bilo koju ulaznu definiciju automata jest 10 sekundi. Budući da će se tipične implementacije postupka minimizacije izvoditi manje od 1 sekunde, 10 sekundi je i više nego dovoljno.

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