

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

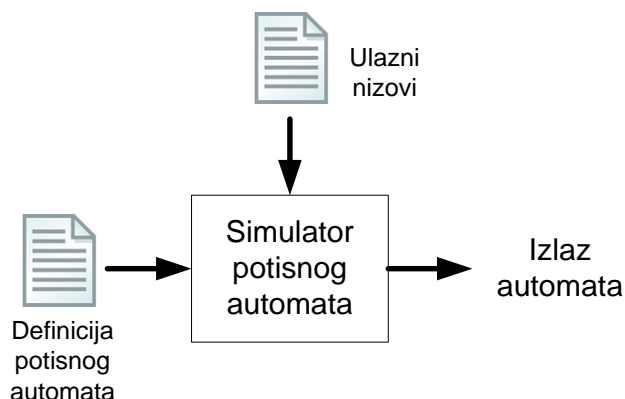
UVOD U TEORIJU RAČUNARSTVA

Ak. God. 2010/2011

2. Laboratorijska vježba



Programski ostvariti simulator nedeterminističkog potisnog automata (PA) koji nizove prihvaća prihvatljivim stanjem. Ulaz u simulator su dvije tekstualne datoteke koje sadrže definiciju PA i ulazne nizove (Slika 1).



Slika 1 Simulator potisnog automata

Definicija automata je u datoteci zapisana u sljedećem formatu:

1. red: skup stanja automata
2. red: skup ulaznih znakova
3. red: skup znakova stoga
4. red: početno stanje
5. red: znak dna stoga
6. red: skup prihvatljivih stanja
7. do zadnjeg reda: funkcija prijelaza oblika

`trenutnoStanje, ulazniZnak, znakStoga->novoStanje, nizZnakovaStoga`

U skladu s definicijom u udžbeniku, znakovi stoga koji se u prijelazu stavljaju na stog stavljaju se na stog zdesna nalijevo, tj. znak na vrhu stoga nakon prijelaza je krajnje lijevi znak u nizu `nizZnakovaStoga`. Umjesto znaka ε koristi se znak \$ (vidi primjer). Ako iz istog stanja, za isti ulazni znak i znak vrha stoga automat može preći u više parova (novo stanje, niz znakova stoga), onda su ti parovi odvojeni znakom # (vidi primjer). Kao u prvom labosu, stanja, ulazni znakovi, znakovi stoga i prihvatljiva stanja bit će odvojeni zarezom (ako ih ima više od jedan).

Automat opisan ulaznom datotekom zadovoljavat će sljedeća ograničenja (**Napomena:** ova ograničenja ne treba provjeravati, nego program može pretpostaviti da su ona zadovoljena):

1. Imena svih stanja su nizovi malih i velikih slova engleske abecede i dekadskih znamenaka. Duljina imena stanja neće biti veća od 20 znakova. Broj stanja automata neće biti veći od 20. Svako stanje automata će se u prvom retku datoteke navesti točno jednom.
2. Skup ulaznih znakova bit će podskup skupa malih slova engleske abecede i dekadskih znamenaka. Svaki ulazni znak će se u drugom retku datoteke pojaviti točno jednom.

3. Skup znakova stoga bit će podskup skupa velikih slova engleske abecede. Svaki znak stoga će se u trećem retku datoteke pojaviti točno jednom.
4. Skup prihvatljivih stanja će sadržavati nula ili više stanja iz skupa stanja navedenih u prvom retku. Svako prihvatljivo stanje bit će navedeno točno jednom.
5. Svaki redak funkcije prijelaza će s lijeve strane znakova \rightarrow imati jedinstvenu trojku `trenutnoStanje, ulazniZnak, znakStoga`.
6. U niti jednom prijelazu `nizZnakovaStoga` neće biti dulji od 5 znakova.
7. **VAŽNO:** Zadana funkcija prijelaza neće sadržavati **epsilon prijelaze koji povećavaju stog**. Drugim riječima, ako je `ulazniZnak` u nekom prijelazu `$`, `nizZnakovaStoga` s desne strane bit će ili točno jedan znak stoga ili `$`. Ovo ograničenje modela potisnog automata ne smanjuje klasu jezika koju model može prepoznati (kontekstno neovisni jezici).

Za automat koji zadovoljava gore navedena ograničenja, ostvareni simulator treba završiti izvođenje u **maksimalno 10 sekundi po ulaznom nizu**.

Primjer ulazne datoteke s definicijom PA koji prihvaća jezik $\{ww^R \mid w \in (0+1)^*\}$ prikazan je na slici 2.

```
q0,q1,qp
0,1
J,N,K
q0
K
qp
q0,$,K->qp,K
q0,0,K->q0,NK
q0,1,K->q0,JK
q0,0,N->q0,NN#q1,$
q0,0,J->q0,NJ
q0,1,N->q0,JN
q0,1,J->q0,JJ#q1,$
q1,0,N->q1,$
q1,1,J->q1,$
q1,$,K->qp,K
```

Slika 2 Primjer definicije potisnog automata

Datoteka s ulaznim nizovima može sadržavati više nizova pri čemu se svaki niz nalazi u zasebnom redu (Slika 3). Ulazni nizovi neće biti dulji od 20 znakova svaki.

Za svaki zadani ulazni niz simulator treba ispisati svaki prijelaz automata i pripadna stanja i sadržaje stoga te navesti prihvaća li se niz ili ne.

```
0
01
00

110011
111110
```

Slika 3 Primjer datoteke s ulaznim nizovima