

## UVOD U TEORIJU RAČUNARSTVA

Ak. God. 2015/2016  
5. Laboratorijska vježba

U petoj laboratorijskoj vježbi zadatak je programski ostvariti simulator osnovnog modela Turingovog stroja (TS). Ulaz u simulator Turingovog stroja je tekstualni zapis njegove definicije, tekstualni zapis trake Turingovog stroja i početni položaj glave.

Format za zapis ulaznog niza i definicije TS je:

- 1. redak: Skup stanja odvojenih zarezom
- 2. redak: Skup ulaznih znakova odvojenih zarezom
- 3. redak: Skup znakova trake odvojenih zarezom
- 4. redak: Znak kojim se označava prazna ćelija
- 5. redak: Tekstualni zapis trake Turingovog stroja
- 6. redak: Skup prihvatljivih stanja odvojenih zarezom
- 7. redak: Početno stanje
- 8. redak: Početni položaj glave Turingovog stroja
- 9. redak i svi ostali retci: Funkcija prijelaza u formatu:  
*trenutnoStanje, znakNaTraci -> novoStanje, noviZnakNaTraci, PomakGlave*

Turingov stroj opisan ulaznom datotekom zadovoljavat će sljedeća ograničenja:

- 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 TS neće biti veći od 20. Svako stanje TS bit će navedeno točno jednom.
- 2) Skup ulaznih znakova bit će podskup skupa malih slova engleske abecede i dekadskih znamenaka. Svaki ulazni znak će se pojaviti točno jednom. Svaki ulazni znak predstavljen je točno jednim znakom koji pripada malim slovima engleske abecede ili dekadskim znamenkama.
- 3) Skup znakova trake bit će unija podskupa skupa velikih slova engleske abecede i skupa ulaznih znakova. Svaki znak trake će se pojaviti točno jednom. Svaki znak trake predstavljen je točno jednim znakom.
- 4) Traka Turingovog stroja će biti ograničena na 70 znakova. Tekstualni zapis trake će biti prikazan kao niz od 70 znakova. Neka je tekstualni zapis trake TS dan na sljedeći način:

$$Z_0 Z_1 Z_2 \dots Z_{68} Z_{69}$$

gdje svaki  $Z_i$  predstavlja jedno polje trake.

- 5) Znak kojim se označava prazna ćelija jest jedan znak iz skupa znakova trake.
- 6) Skup prihvatljivih stanja će sadržavati nula ili više stanja iz skupa stanja. Svako prihvatljivo stanje bit će navedeno točno jednom.
- 7) Početno stanje jest jedan znak iz skupa znakova kojima se označavaju stanja.
- 8) Početni položaj glave jest jedan cijeli broj od 0 do 69 koji predstavlja indeks polja trake iznad kojeg se nalazi glava TS na početku simulacije.
- 9) Svaki redak funkcije prijelaza će imati sljedeći oblik:  
*trenutnoStanje, znakNaTraci -> novoStanje, noviZnakNaTraci, PomakGlave*,  
gdje *PomakGlave* može imati vrijednost *L* ili *R*, *L* ako se glava pomiče u lijevo, *R* ako se pomiče u desno.  
Svaka dvojka *trenutnoStanje, znakNaTraci* pojavit će se točno jednom.

Primjer ulazne datoteke TS koji prihvaća jezik  $L(M) = \{0^n 1^n \mid n \geq 1\}$

```

01   q0, q1, q2, q3, q4
02   0, 1
03   0, 1, X, Y, B
04   B
05   0011BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
06   q4
07   q0
08   0
09   q0, 0->q1, X, R
10   q1, 0->q1, 0, R
11   q2, 0->q2, 0, L
12   q1, 1->q2, Y, L
13   q2, X->q0, X, R
14   q0, Y->q3, Y, R
15   q1, Y->q1, Y, R
16   q2, Y->q2, Y, L
17   q3, Y->q3, Y, R
18   q3, B->q4, B, R

```

TS staje s radom kada se nađe u stanju u kojem za pročitani znak trake nema definiran prijelaz. Ostvareni simulator TS kao izlaz treba ispisati:

- konačno stanje u kojem se nalazio nakon što više nije bilo definiranih prijelaza,
- položaj glave kao indeks polja trake iznad kojeg se glava TS zaustavila (Indeksi počinju od 0, za polje  $Z_0$  INDEKS je 0 itd..., za polje  $Z_{69}$  INDEKS je 69),
- tekstulani prikaz trake TS,
- te prihvatljivost niza (0 ili 1).

Konačno stanje, položaj glave, tekstualni prikaz trake i prihvatljivost niza odvojeni su u izlaznom nizu znakom |.

Postoje dva posebna slučaja u kojima TS neće moći izvršiti simulaciju zbog ograničenja trake:

1) Primjerice, neka se TS prilikom simulacije nađe u stanju  $q_0$ , neka je glava stroja na polju trake  $Z_0$ , neka je u tom trenutku na traci na polju  $Z_0$  neki znak  $T_0$ . Ukoliko za par  $(q_0, T_0)$  postoji prijelaz oblika  $(q_0, T_0) \rightarrow (q_i, T_i, L)$ , dakle prijelaz koji pomiče glavu u lijevo, tada TS staje zbog ograničenja trake te ignorira prijelaz. TS ispisuje izlaz kao u slučaju da nije bilo definiranih prijelaza.

2) Slično kao i 1.) neka se TS prilikom simulacije nađe u stanju  $q_0$  na polju trake  $Z_{69}$ , neka je u tom trenutku na traci na polju  $Z_{69}$  neki znak  $T_0$ . Ukoliko za par  $(q_0, T_0)$  postoji prijelaz oblika  $(q_0, T_0) \rightarrow (q_i, T_i, R)$ , dakle prijelaz koji pomiče glavu u desno, tada TS staje zbog ograničenja trake te ignorira prijelaz. TS ispisuje izlaz kao u slučaju da nije bilo definiranih prijelaza.

Primjer izlaza simulatora TS za gore navedeni primjer konfiguracije TS:

```
q4 | 5 | XXYYBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB | 1
```

Napomene.

- 1) Nije potrebno provjeravati ispravnost formatiranja ulazne datoteke ili ispravnost rada TS. Neće biti preklapanja između skupa stanja i skupa simbola abecede.
- 2) Vremensko ograničenje na izvođenje programa za bilo koju ulaznu definiciju TS jest 10 sekundi
- 3) Ulazna točka za Java rješenja treba biti u razredu SimTS, a ulazna točka u Python rješenja treba biti u datoteci SimTS.py.