

Analiza Algoritmilor

Tema 1 - Mașina Turing

Termen de predare: 2 Nov 2016 (soft)
9 Nov 2016 (hard)

Responsabili temă: Mihai-Valentin DUMITRU
George-Sebastian PÎRTOACĂ

Data publicării: 16 Oct 2016
Ultima actualizare: 26 Oct 2016

1 Simulare Mașină Turing - 60p

1.1 Introducere

O Mașină Turing Deterministă (MTD) este un tuplu $(\Sigma, K, F, \delta, s_0)$ unde:

- Σ : o mulțime finită de simboluri (alfabetul peste care MTD este definită)
- K : o mulțime finită de stări
- F : mulțimea de stări finale ($F \subseteq K$)
- δ : funcția de tranziție ($K \times \Sigma \rightarrow K \times \Sigma \times \{L, R, H\}$)
- s_0 : starea inițială ($s_0 \in K$)

În contextul acestei teme, alfabetul Σ constă din mulțimea cifrelor împreună cu literele mici și mari ale alfabetului englez plus caracterul $\#$ (diez). Cu alte cuvinte:

$$\Sigma = \{a, b, \dots, z, A, B, \dots, Z, 0, 1, \dots, 9, \#\}.$$

O stare va fi reprezentată astfel: S_n unde n este un număr natural (e.g. S_0, S_{101}, S_{45} sunt stări).

Se cere implementarea unui program care primește ca date de intrare codificarea unei MTD și cuvântul de pe banda mașinii, și produce rezultatul execuției mașinii pentru inputul dat (rezultatul execuției unei MTD este dat de cuvântul scris pe banda mașinii atunci când aceasta ajunge într-o stare finală; pentru cazurile în care mașină se agață, citiți secțiunea [1.5](#)).

1.2 Date de intrare

Codificarea MTD se va citi din fișierul "tm.in", iar conținutul benzii din fișierul "tape.in". Structura fișierului "tm.in" este:

- Pe prima linie se va găsi un număr natural N ce reprezintă numărul de stări ale MTD, urmat de N șiruri de caractere separate printr-un singur spațiu, fiecare șir reprezentând o stare a mașinii (e.g. $4 S_0 S_2 S_{40} S_3$). Stările pot fi date în orice ordine.
- Pe a doua linie se va găsi un număr natural M ($M \leq N$) ce reprezintă numărul de stări finale, urmat de M șiruri de caractere separate printr-un singur spațiu, fiecare șir reprezentând o stare finală a mașinii (e.g. $2 S_2 S_3$).
- Pe a treia linie se va găsi un șir de caractere ce reprezintă starea inițială.

- Pe a patra linie se va găsi un număr natural P reprezentând numărul tranzițiilor. Următoarele P linii vor avea următorul format:

```
current_state read_symbol next_state write_symbol direction
```

- current_state, next_state: șiruri de caractere
- read_symbol, write_symbol: caractere din Σ
- direction: 'L', 'R' sau 'H'

Fiecare linie din cele P definește o tranziție astfel: dacă MTD se află în starea current_state și citește caracterul read_symbol atunci se mută în starea next_state, scrie caracterul write_symbol pe bandă și deplasează cursorul conform direction (L - stânga, R - dreapta, H - stă pe loc).

Fișierul "tape.in" conține o singură linie, pe care e scris un șir de caractere ce reprezintă configurația inițială a benzii. Acesta începe mereu cu un caracter gol ("#").

1.3 Date de ieșire

Conținutul final al benzii va fi scris în fișierul "tape.out" pe prima linie a acestuia.

Caracterele goale (diez) de la început și sfârșit sunt irelevante.

(i.e. "#####100#####100#" este echivalent cu "#100#")

1.4 Exemplu

"tape.in"

```
#1000100110##
```

"tm.in"

```
3 S0 S1 S2
1 S2
S0
6
S0 # S2 # H
S0 0 S1 # R
S0 1 S0 # R
S1 # S2 # H
S1 0 S1 # R
S1 1 S2 1 R
```

"tape.out"

#####100110###

1.5 Restricții și precizări

- $1 \leq N \leq 1000$
- $1 \leq M \leq N$
- $1 \leq P \leq 10000$
- sunt suficiente 1000 de celule de bandă
- capul de citire se află inițial pe poziția 1 (indexând de la zero)
- $\forall S_n \ 0 \leq n \leq 100000$
- funcția de tranziție este definită corect (nu există 2 definiții pentru aceeași tranziție, și nu există tranziții care implică stări inexistente)
- dacă se ajunge într-o stare în care funcția de tranziție nu este definită atunci în fișierul de ieșire se va afișa mesajul "Se agata!"
- toate stările finale sunt valide (se găsesc în mulțimea stărilor)
- starea inițială este validă (se găsește în mulțimea stărilor)
- banda se consideră infinită doar la dreapta (nu se va citi/scrie vreun simbol la stânga primului caracter citit din fișier)
- implementarea temei se va face în C sau C++

2 Adunare binară - 40p

2.1 Introducere

Se cere să implementați o MTD care să realizeze adunarea a două numere în baza 2. Mașina va fi descrisă într-un fișier, conform formatului prezentat în secțiunea [1.2](#).

Notă: recomandăm implementarea acestei mașini într-un format mai lizibil (e.g. cu nume suges-tive de stări), urmând apoi să o convertiți la formatul prezentat.

2.2 Date de intrare

Inițial, banda mașinii va conține două numere, separate între ele printr-un "#". Numerele sunt scrise de la dreapta la stânga (i.e. primul bit este bitul cel mai puțin semnificativ).

2.3 Date de ieșire

După ce mașina se termină, pe bandă se va afla doar rezultatul adunării. Acesta are același format ca numerele date la input (baza 2, scris de la dreapta la stânga). Caracterele goale (diez) de la început și sfârșit sunt irelevante.

(i.e. "#####100#####" este echivalent cu "#100#")

2.4 Exemplu

"tape.in"

```
#011001101#1110101#
```

"tape.out"

```
#101111011#
```

Primul număr este $101100110_b (= 358_{10})$, iar cel de-al doilea este $1010111_b (= 87_{10})$. Suma acestora este $110111101_b (= 445_{10})$.

2.5 Restricții și precizări

- Numerele date nu au zerouri inițiale (e.g. numărul 4 va fi scris mereu ca 001, niciodată ca 001000000).
- Inputul este considerat mereu corect, nu trebuie să tratați în niciun fel vreun caz de eroare.
- **Nu este necesar** să implementați partea I pentru a implementa partea a II-a.

3 Trimitere

Tema se va trimite pe vmchecker, sub forma unei arhive zip. Aceasta trebuie să conțină:

- Fișierele sursă necesare pentru partea I a temei.
- Fișierul "addition.tm", ce conține rezolvarea pentru partea a II-a a temei.
- Makefile cu cel puțin două reguli: "run" (rulează simulatorul de MTD) și "clean" (șterge toate fișierele rezultate în urma compilării).
- Fișierul "README" care să conțină orice informație relevantă despre implementarea aleasă.

4 Punctaj

Tema va fi notată de la 0 la 100 și reprezintă 0.5 puncte din nota finală. Punctajul este împărțit astfel:

- partea I - 60 de puncte
- partea a II-a - 40 de puncte

După deadline-ul soft, se vor scădea 5 puncte pe zi, până la deadline-ul hard.