Analiza Algoritmilor

Tema 1 - Mașina Turing

Termen de predare: 8 Noiembrie 2017, 23:55

Responsabili temă: Cătălin-Emil FETOIU
George-Sebastian PÎRTOACĂ

1 Simulare Mașină Turing cu două benzi - 60p

1.1 Introducere

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

- Σ: o mulțime finită de simboluri (alfabetul peste care MT este definită)
- K: o mulțime finită de stări
- F: mulțimea de stări finale ($F \subseteq K$)
- δ: funcția de tranziție
- s_0: starea inițială ($s_0 \in K$)

Până acum Mașina Turing a fost prezentată ca având o singură bandă de intrare, împreună cu un cap de citire asociat acesteia. Această definiție poate fi extinsă la mai multe benzi de intrare, fiecare având capul de citire propriu, diferența fiind dată de către funcția de tranziție δ, care va ține cont de mulțimea caracterelor citite de pe fiecare bandă pentru a decide starea următoare. De asemenea, simbolurile scrise pe fiecare bandă vor fi independente, la fel și direcțiile în care se vor deplasa capetele de citire. Astfel:

- Pentru o MT cu o singură bandă δ va fi $(K \times \Sigma \to K \times \Sigma \times \{L, R, H\})$
- Pentru o MT cu două benzi δ va fi $(K \times (\Sigma \times \Sigma) \to K \times \Sigma \times \{L, R, H\} \times \Sigma \times \{L, R, H\})$

Alfabetul Σ este format din mulțimea cifrelor împreună cu literele mici și mari ale alfabetului englez și caracterul special $\#$ (diez), ce reprezintă zonele nefolosite ale unei benzi.

O stare va fi reprezentată de un șir conținând caractere alfanumerice (e.g. start, end, state0, state1 sunt stări).

Se cere implementarea unui program care primește ca date de intrare codificarea unei MT cu două benzi și cuvintele aflate pe acestea și produce rezultatul execuției mașinii pentru inputul dat. Rezultatul execuției unei MT este dat de cuvintele scrise pe cele două benzi atunci când aceasta ajunge într-o stare finală. În cazul în care nu mai poate fi efectuată nicio tranziție din starea curentă va fi scris un mesaj special in fișierul de output, conform secțiunii 1.3).

1.2 Date de intrare

Codificarea MT, cât și conținutul celor două benzi va fi citit din fișierul `"task1.in"`. Structura fișierului este următoarea:

- Pe prima linie se va găsi un număr natural N ($1 \leq N \leq 1000$) ce reprezintă numărul de stări ale MT. Pe a doua linie se vor afla N șiruri de caractere separate prin spațiu, fiecare șir reprezentând o stare a mașinii.
- Pe a treia linie se va găsi un număr natural M ($1 \leq M \leq N$) ce reprezintă numărul de stări finale ale MT. Pe a patra linie se vor afla M șiruri de caractere separate prin spațiu, fiecare șir reprezentând o stare finală a mașinii.
- Pe următoarea linie se va găsi un șir de caractere ce reprezintă starea inițială.
- Pe următoarea linie se va găsi un număr natural P ($1 \leq P \leq 10000$) ce reprezintă numărul tranzițiilor. Fiecare din următoarele P linii vor avea următorul format:

  `current_state tape1_read_symbol tape2_read_symbol next_state tape1_write_symbol tape1_direction tape2_write_symbol tape2_direction`

  Fiecare linie din cele P definește o tranziție astfel: dacă MT se află în starea `current_state` și citește caracterele `tape1_read_symbol, tape2_read_symbol` de pe prima, respectiv a doua bandă, trece în starea `next_state`, scrie caracterele `tape1_write_symbol, tape2_write_symbol` pe prima, respectiv a doua bandă și deplasează cele două capete de citire conform `tape1_direction, tape2_direction` (`L` - stânga, `R` - dreapta, `H` - stă pe loc).
- Pe ultimele două linii se vor gasi 2 șiruri de caractere, reprezentând conținuturile celor două benzi, fiecare pe o linie separată. Fiecare șir va incepe cu un singur $\#$.

1.3 Date de ieșire

Conținutul final al celor două benzi va fi scris în fișierul `"task1.out"`, prima bandă pe prima linie, respectiv a doua bandă pe a doua linie. În cazul în care din starea curentă nu se poate efectua nicio tranziție, iar această stare nu este finală, în fișierul de output se va afișa doar mesajul `"The machine has blocked!"`, pe prima linie.

Caracterele $\#$ de la început și sfârșit vor fi ignorate și nu vor afecta evaluarea corectitudinii output-ului (i.e. `"####abc#######"` este echivalent cu `"#abc#"`).

1.4 Exemplu

"task1.in"

```
3
S0 S1 S2
1
S2
S0
9
S0 a a S0 a H a R
S0 a b S0 a H b R
S0 b a S0 b H a R
S0 b b S0 b H b R
S0 a # S1 a H # R
S0 b # S1 b H # R
S1 a # S1 # R a R
S1 b # S1 # R b R
S1 # # S2 # H # H
#abab#
#bbba#
```

"task1.out"

```
######
#bbba#abab#
```

1.5 Restricții și precizări

- capetele de citire pentru fiecare bandă se vor afla inițial pe poziția 1 (indexând de la zero)
- sunt suficiente 10000 de celule pentru fiecare bandă
- lungimea șirului de caractere corespunzător fiecărei stări va fi cuprinsă între 1 și 100 (inclusiv)
- funcția de tranziție este definită corect (nu există 2 definiții pentru aceeași tranziție, și nu există tranziții care implică stări inexistente)
- toate stările finale sunt valide (se găsesc în mulțimea stărilor)
- starea inițială este validă (se găsește în mulțimea stărilor)
- cele două benzi se consideră infinite doar la dreapta (nu se va citi/scrie vreun simbol la stânga primului caracter al unei benzi)

- implementarea temei se va face în C/C++ sau Java

2 Găsire caracter dominant - 40p

2.1 Introducere

Se cere implementarea unei MT cu o singură bandă care să determine caracterul dominant dintr-un șir. Prin caracter dominant se înțelege caracterul care are numărul de apariții strict mai mare decat jumătate din lungimea șirului. Pentru simplitate, șirul va fi format doar din caracterele `a, b, c, d`. Mașina va fi descrisă într-un fișier denumit **task2**, conform formatului prezentat în secțiunea 1.2, însă, pentru că mașina are o singură bandă, tranzițiile vor avea următorul format:

`current_state tape_read_symbol next_state tape_write_symbol tape_direction`
Astfel fiecare tranziție va fi definită de o stare curentă, caracterul citit, starea următoare, caracterul scris și direcția de deplasare a capului de citire.

Un exemplu de codificare a unei mașini Turing cu o singură bandă este următorul:

```
3
SO S1 S2
1
S2
S0
5
S0 0 S0 1 R
S0 1 S1 1 R
S1 0 S1 0 R
S1 1 S1 1 R
S1 # S2 # H
```

2.2 Date de intrare

Banda mașinii va conține un șir format doar din caracterele `a, b, c, d`, acesta fiind precedat de un singur #.

2.3 Date de ieșire

După ce mașina ajunge într-o stare finală, pe bandă trebuie să rămână doar caracterul dominant, iar șirul inițial trebuie supracris de caracterul #. Caracterele # de la început și sfârșit vor fi ignorate și nu vor afecta evaluarea corectitudinii output-ului (i.e. `"####a#######"` este echivalent cu `"#a#"`).

2.4 Exemplu

`"input"`

```
#abbaaccaabaad#
```

`"output"`

```
#####a########
```

Explicatie: Caracterul `'a'` este dominant deoarece are 7 apariții, lungimea șirului fiind egală cu 13.

2.5 Restricții și precizări

- capul de citire al benzii se va afla inițial pe poziția 1 (indexând de la zero)
- se garantează că va exista mereu un caracter dominant.
- prin jumătatea lungimii șirului se înțelege partea întreaga a împărțirii lungime / 2.
- **Nu este necesară** implementarea părții I pentru implementarea părții a II-a.

3 Trimitere

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

- Fișierele sursă necesare pentru partea I a temei.
- Fișierul **task2**, ce conține rezolvarea pentru partea a II-a a temei.
- Un fișier **Makefile** având următoarele reguli: **build** (compilează fișierele sursă), **run** (rulează simulatorul de MT) și **clean** (șterge toate fișierele rezultate în urma compilării).

- Un fișier **README** care să conțină descrierea soluției.

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.