

Examen - LFA

Exercise 1.

1.1. O mașină Turing cu două capete și o bandă este un \times tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ - unde Q, Σ, Γ sunt mulțimi finite și:

- Q reprezintă mulțimea de stări
- Σ este alfabetul dat ca input (nu conține simbolul blank "-")
- Γ este alfabetul de pe bandă, "-" $\in \Gamma$ și $\Sigma \subseteq \Gamma$
- δ este funcția de tranziție

$$\delta : Q \times \Gamma \times \Gamma \rightarrow Q \times \Gamma \times \Gamma \times \{L, R, P\} \times \{L, R, P\}$$

- $q_0 \in Q$ este starea de start
 - $q_{\text{accept}} \in Q$ este starea de accept
 - $q_{\text{reject}} \in Q$ este starea de respingere
- } $q_{\text{accept}} \neq q_{\text{reject}}$

specificatii: $L \rightarrow$ vine de la "left" și presupune mutarea cu o poziție mai la stânga în inputul dat

$R \rightarrow$ vine de la "right" și presupune mutarea cu o poziție mai la dreapta în inputul dat

$P \rightarrow$ presupun că se rămâne pe loc. Nu se face nicio mutare.

Exercise 1.2

TM - 2 heads and one tape computation.

Mașina Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ funcționează astfel:

- Se primește un input $w \in \Sigma^*$ în partea din stânga a benzii și restul benzii rămâne goală (de aceea după input se pune simbolul "-").

→ Cele două capete pornesc din partea stângă a benzii (de pe poziția 0 din stringul inputat).

→ La pornirea mașinii Turing, se respectă regulile date de funcția de tranziție.

↳ funcția de tranziție arată astfel:

~~stare curentă caracterul găsit la capăt 1~~

~~caracterul găsit la capăt 2 stare~~

stare curentă stare în care se ajunge

caracter găsit la capăt 1 caracter găsit la capăt 2

caracter cu care se înlocuiește cel de pe poz. capătului 1

caracter cu care se înlocuiește cel de pe poz. capătului 2

direcția pentru capăt 1 direcția pentru capăt 2

→ Dacă M vrea să mute un capăt înafara benzii (la stânga), capătul va sta pe loc chiar dacă funcția de tranziție dorește mutarea.

→ Procesul continuă până când se ajunge în starea de acceptare sau în starea de respingere. În caz contrar, M continuă la mersărit.

→ De-a lungul procesului, se fac modificări cu privire la starea curentă, la conținutul inputului (de pe bandă) și la poziția celor 2 capete.

→ Nu apare nicio problemă în cazul în care capetele se găsesc pe aceeași poziție, pentru că funcția de tranziție prezintă modificări în cadrul fiecărui capăt și în funcție de conținutul benzii, nu a pozițiilor capetelor.

Exemplu

Exercițiul 2

Cele 2 capete sunt setate la începutul stringului, iar starea de început este precizată în fișier. Se verifică dacă fișierul este valid:

1. există toate secțiunile,
2. tranzițiile sunt valide

3. sunt valide simbolurile pt left, right, pe loc,

Cele 2 capete fiind puse în cod, de mine, sigur există și sunt 2 la număr.

Dacă configurarea este validă, se dă un input și cu ajutorul fișierului corespunzător se verifică dacă limbajul este valid și se afișează varianta modificată.

Partea de calcul se termină doar în cazul în care se ajunge într-una din cele 2 stări: stare de acceptare și stare de reject. Astfel se verifică dacă există o tranziție potrivită pentru starea și capetele curente și se produc modificări în starea curentă, capete și inputul de pe bandă.

Exercise 3

TM creată este simplă întrucât cazurile care par mai complicate sunt rezolvate de tranziții.

Așa cum am specificat la exercitiul 1, cele două capete pornesc de la capătul din stânga al benzii (adică de la poziția 0 în input).

În mare, mașina funcționează astfel (pe baza fișierului dat).

I Dacă primul caracter din input este despărțitorul #, atunci înseamnă că nu există cuvânt \Rightarrow se trece în starea de respingere \Rightarrow limbajul nu este acceptat

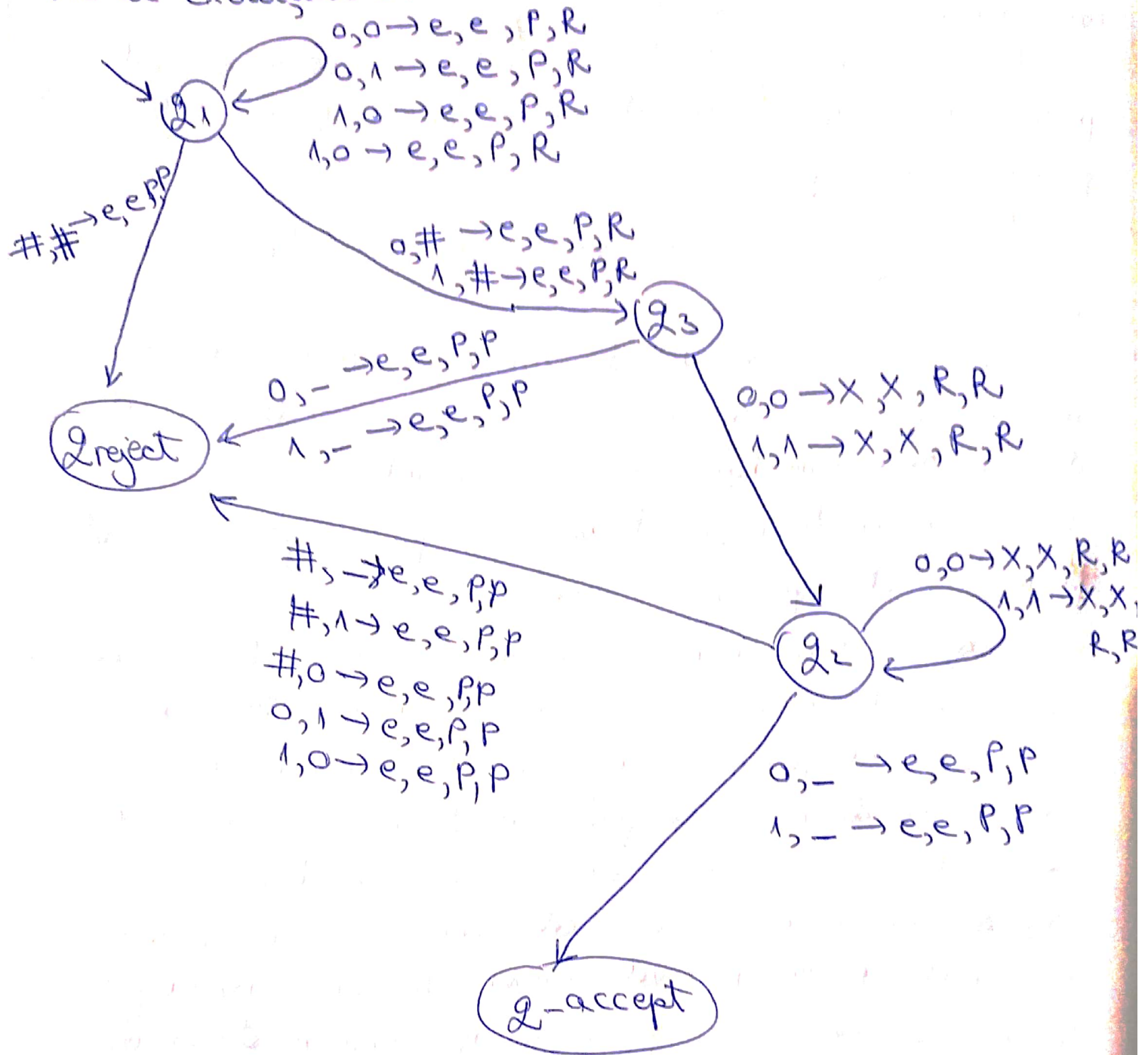
ii În caz contrar, mișcă un capăt până ajung la caracterul # (celălalt capăt nu se mișcă).

Subcaz 1: Dacă după # urmează simbolul blank înseamnă că nu există prefix \Rightarrow se trece în starea de respingere.

Subcaz 2: Verifică (cu ajutorul celor 2 capete) dacă caracterele de pe cele 2 poziții sunt egale). În caz afirmativ, se continuă verificarea un pas mai la dreapta. În caz contrar, înseamnă că acel cuvânt de după # nu este prefix pentru primul cuvânt. Procesul se repetă până la un caz negativ sau până la ajungerea în starea de acceptare.

Limbajul este acceptat doar dacă primul capăt nu a ajuns la # și al doilea a ajuns la finalul inputului, adică la simbolul blank).

Diagrama pentru mașina Turing creată în fișierul de la exercitiul 3.



Exercise 4

Mașina doar mută cuvântul de după caracterul $+$ (specific adunării) lângă cuvântul de pe bandă ce se află înaintea caracterului $+$.

Precizare: nu se poate face suma dacă există un singur operand. Exemplu: $+a-$; $+a- \rightarrow ok=0$
 \hookrightarrow va spune că accept-status este 0

Diagrama pt mașina Turing creată în fișierul de la exercitiul 4.

