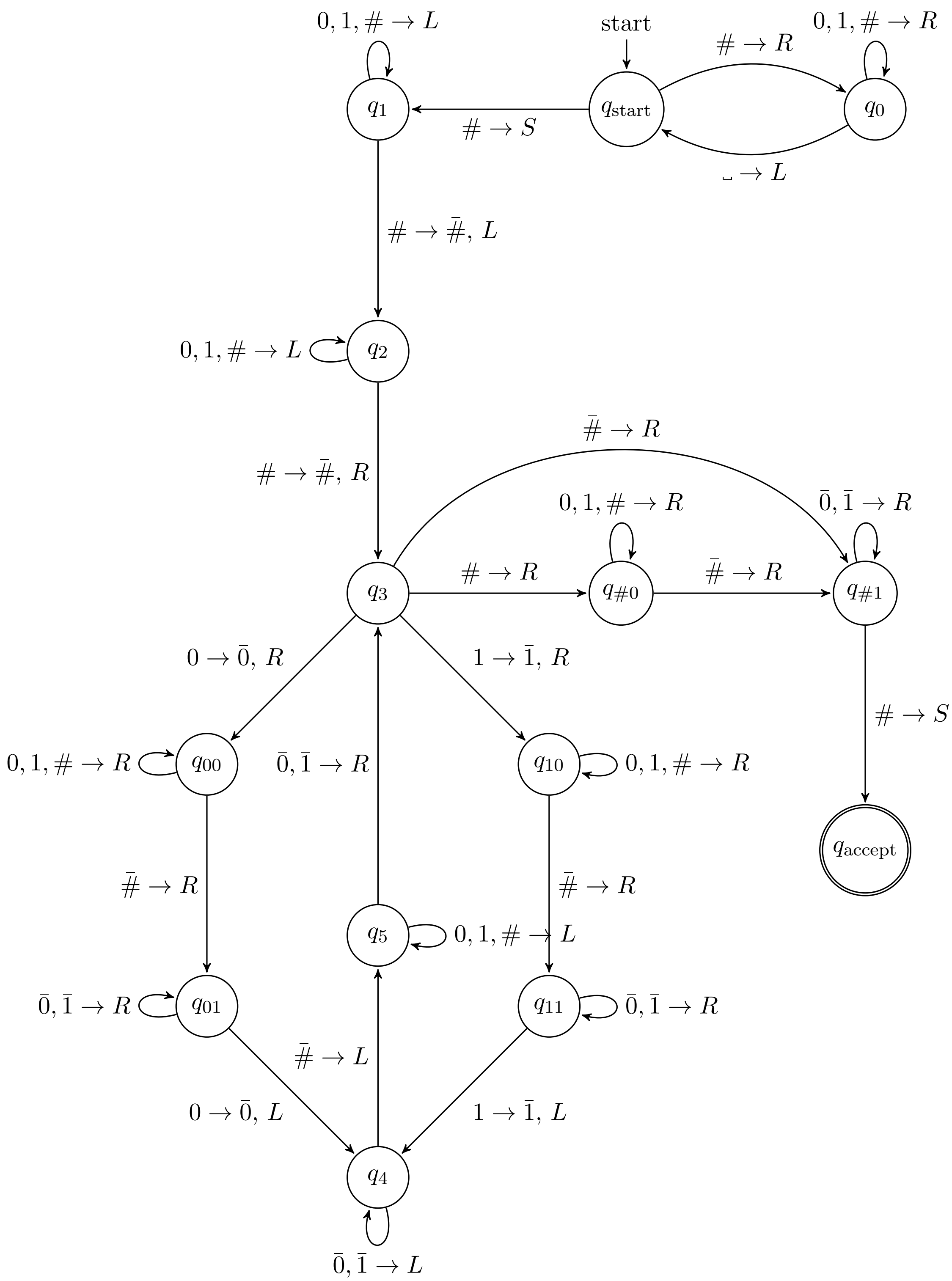
582206 Laskennan mallit, syksy 2012
9. harjoitusten malliratkaisut
Juhana Laurinharju ja Jani Rahkola

1. Esitä tilakaaviona epädeterministinen Turingin kone, joka tunnistaa aakkoston $\{0, 1, \#\}$ kielen

$$\{\#w_1\#w_2\#\dots\#w_n\# \mid w_i \in \{0,1\}^* \text{ kaikilla } i \text{ ja } w_i = w_j \text{ joillakin } i \neq j\}.$$



2. [Sipser Problem 3.9] Merkintä k-PDA tarkoittaa pinoautomaattia, jossa on käytettävänä k pinoa. Siis 0-PDA on NFA ja 1-PDA on tavallinen PDA. Osoita, että

(a)

Väite. *2-PDA pystyy tunnistamaan kieliä, joita 1-PDA ei pysty,*

2-PDA:lla voidaan tunnistaa kieliä, joita pinoautomaatilla ei, sillä 2-PDA on Turing täydellinen. 2-PDA:lla voidaan siis tunnistaa esimerkiksi kieli $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ jota pinoautomaatilla ei voida tunnistaa.

2-PDA:n Turing täydellisyys voidaan osoittaa simuloimalla Turingin konetta 2-PDA:lla:

- Simulaatiossa pinon 2 päälimmäinen alkio kuvaa Turigin koneen lukupään alla olevaa kohtaa. Jos pinossa 2 ei ole alkioita, on lukupään alla tyhjä merkki $\llcorner\!\lrcorner$.
- Merkin korvaaminen tapahtuu ottamalla alkio pinosta 2 ja laittamalla korvaava merkki tilalle.
- Lukupään liikuttaminen vasemmalle tapahtuu ottamalla alkio pinosta 1 ja siirtämällä se pinoon 2. Jos pinossa 1 ei ole alkioita, on lukupää nauhan alussa, eikä tehdä mitään.
- Lukupään liikuttaminen oikealle tapahtuu ottamalla alkio pinosta 2 ja laittamalla se pinoon 1. Jos pinossa 2 ei ole alkioita, on lukupää siirtynyt nauhalla syötteen ohi. Laitetaan tällöin pinoon 1 tyhjä merkki $\llcorner\!\lrcorner$.

Simulaatio alkaa lukemalla koko syöte pinoon 1, ja siirtämällä sitten lukupäätä vasemalle, kunnes se on syötteen alussa.

(b)

Väite. *mutta minkä tahansa 3-PDA:n tunnistama kieli voidaan tunnistaa 2-PDA:lla.*

3-PDA:ta voidaan simuloida nelinauhaisella Turingin koneella. Kolmea nauhaa käytetään pinoautomaatin kolmen pinon simulointiin, ja neljännellä nauhalla suoritetaan pinoautomaation siirtymiä. Tiedämmä myös, että nelinauhaista Turingin konetta voidaan simuloida yksinauhaisella Turigin koneella. Yllä olemme osoittaneet, että Turigin konetta voidaan simuloida 2-PDA:lla. Tämän ketjun kautta voimme siis simuloida 3-PDA:ta 2-PDA:lla.

Vihje: Simuloi Turingin koneen nauhaa kahdella pinolla. Esitä ratkaisun periaate pseudokoodilla tms. menemättä automaattiformalismin yksityiskohtiin.

3. [Sipser Exercise 3.14] *Jonoautomaatti* on muuten kuin pinoautomaatti, mutta pino on korvattu jonolla. Jonoon voidaan kohdistaa kahdenlaisia operaatioita:

- ENQUEUE (a) kirjoittaa merkin a jonon loppuun ja
- DEQUEUE poistaa jonon ensimmäisen merkin ja palauttaa sen arvonaan.

Pinoautomaatin tapaan syöte on luettavissa merkki kerrallaan. Sovitaan, että syötteessä on aina loppumerkkinä (mutta ei muualla) tyhjämerkki $\llcorner\!\lrcorner$. Turingin koneen tapaan pinoautomaatti hyväksyy syötteen siirtymällä erilliseen hyväksyvään tilaan.

Osoita, että mikä tahansa Turing-tunnistettava kieli voidaan tunnistaa deterministisellä jonoautomaatilla. Perusteluksi riittää esittää sopivan tasoisena pseudokoodina, miten Turingin konetta voidaan simuloida jonoa käyttäen.

Simulaatiossa haluaisimme esittää Turgin koneen tilan

$$w_1 \dots w_n q u_1 \dots u_n$$

jonolla

$$u_1 \dots u_n \$ w_1 \dots w_n$$

Jonossa merkki \$ kuvaa nauhan vasenta reunaa. Lisäksi jos \$ merkki on jonon ensimmäisenä, tulkitaan se tyhjäksi merkiksi $\llcorner\!\lrcorner$.

Lukupään siirtämiseksi vasemalle tarvitsemme kahden muuttujan verran apumuistia. Koska nauha-aakkosto on äärellinen, voidaan nämä apumuuttujat koodata jonoautomaatin tiloihin asettamalla tilojen joukoksi $\Gamma \times \Gamma \times Q$. Annetaan muuttujille nimet HEAD ja TAIL ja pidetään näissä muuttujissa jonon ensimmäistä ja viimeistä merkkiä.

Jono

$$u_1 \dots u_n \$ w_1 \dots w_n$$

kuvataan nyt siis jonolla ja muuttujien arvoilla

$$\begin{aligned} \textsc{head} &= u_1 \\ u_2 \dots u_n \$ w_1 \dots w_{n-1} & \\ \textsc{tail} &= w_n \end{aligned}$$

Lukupään alla oleva merkki on siis muuttujassa HEAD.

- Lukupään siirtäminen oikealle toimii nyt seuraavasti:
  1. Jos HEAD $\neq$ \$:
     1. ENQUEUE(TAIL)
     2. TAIL = HEAD
     3. HEAD = DEQUEUE
  2. Muuten:
     1. ENQUEUE (TAIL)
     2. TAIL = $\llcorner\!\lrcorner$

  $$\begin{array}{rcr} \textsc{head} = u_1 & & \textsc{head} = u_2 \\ u_2 \dots u_n \$ w_1 \dots w_{n-1} & \rightarrow & u_3 \dots u_n \$ w_1 \dots w_n \\ \textsc{tail} = w_n & & \textsc{tail} = u_1 \end{array}$$

  Jos lukupää on siirtynyt syötteen ohi, syötämme vain jonoon tyhjän merkin:

  $$\begin{array}{rcr} \textsc{head} = \$ & & \textsc{head} = \$ \\ w_1 \dots w_n u_1 \dots u_{n-1} & \rightarrow & w_1 \dots w_n u_1 \dots u_n \\ \textsc{tail} = u_n & & \textsc{tail} = \llcorner\!\lrcorner \end{array}$$

- Kun lukupäätä siirretään vasemmalle, merkkaamme jonon viimeisenä olevan merkin. Sitten pyöritämme jonoa kunnes merkattu merkki on siirtynyt jonon alkuun. Silloin jono on oikeassa asennossa ja voimme poistaa merkinnän.
  1. Jos TAIL $\neq$ \$:
     1. Merkkaa muuttujassa TAIL oleva merkki.
     2. Kunnes merkattu merkki on muuttujassa HEAD
        1. ENQUEUE(TAIL)
        2. TAIL = HEAD
        3. HEAD = DEQUEUE

3. Poista muuttujassa HEAD olevan merkin merkkaus.

Jos lukupää on nauhan vasemmassa reunassa, on jono tilassa

$$\begin{aligned} \text{HEAD} &= w_1 \\ w_2 \ldots w_n u_1 \ldots u_n \\ \text{TAIL} &= \$ \end{aligned}$$

eikä jonoa tarvitse siis käsitellä mitenkään siirrettäessä lukupäätä vasemmalle. Tämä on myös tila josta simulaatio alkaa, kun koko syöte on ensin luettu jonon.

4. [Sipser Problem 3.15] Näytä että ratkeavien kielten joukko on suljettu seuraavien operaatioiden suhteen. Olkoon A ja B kieliä ja TM_A ja TM_B ne ratkaisevat Turingin koneet.

   (a) Yhdiste

   - Syötteellä w:
     1. Aja TM_A ja TM_B syötteellä w.
     2. Jos jompi kumpi Turingin koneista hyväksyy, hyväksy. Muuten hylkää.

   (b) Ketjutus

   - Syötteellä w:
     1. Jaa merkkijono w epädeterministisesti kahteen osaan $w = w_1 w_2$.
     2. Aja TM_A syötteellä w_1 ja TM_B syötteellä w_2.
     3. Jos molemmat Turingin koneet hyväksyvät hyväksy. Muuten hylkää.

   (c) Tähti

   - Syötteellä w:
     1. Jaa merkkijono w epädeterministisesti n osaan $w = w_1 \cdots w_n$.
     2. Aja TM_A jokaisella osalla $w_i, i \in \{1, \ldots, n\}$.
     3. Jos jokainen ajo hyväksyy, hyväksy. Muuten hylkää.

   (d) Komplementti

   - Syötteellä w:
     1. Aja TM_A syötteellä w.
     2. Jos ajo hyväksyy, hylkää, jos ajo hylkää, hyväksy.

   (e) Leikkaus

   - Syötteellä w:
     1. Aja TM_A ja TM_B syötteellä w.
     2. Jos molemmat Turingin koneista hyväksyvät, hyväksy. Muuten hylkää.

5. [Sipser Problem 3.16] Näytä että Turin-tunnistettavien kielten joukko on suljettu seuraavien operaatioiden suhteen.

   Tunnistaminen eroaa näiden operaatioiden osalta vain yhdisteen tapauksessa.

   (a) Yhdiste

   - Syötteellä w:
     1. Aja TM_A ja TM_B syötteellä w siten, että molempia Turingin koneita suoritetaan yksi tilasiirtymä kerrallaan.
     2. Jos jompi kumpi Turingin koneista hyväksyy, hyväksy.

   Toinen vaihtoehto on käyttää epädeterministisyyttä.

- Syötteellä w:
  1. Valitse epädeterministisesti toinen Turingin koneista TM_A ja TM_B. Aja valittua konetta syötteellä w.
  2. Jos ajettava Turingin kone hyväksyy, hyväksy.

(b) Ketjutus

- Syötteellä w:
  1. Jaa merkkijono w epädeterministisesti kahteen osaan $w = w_1 w_2$.
  2. Aja TM_A syötteellä w_1 ja TM_B syötteellä w_2.
  3. Jos molemmat Turingin koneet hyväksyvät hyväksy.

(c) Tähti

- Syötteellä w:
  1. Jaa merkkijono w epädeterministisesti n osaan $w = w_1 \cdots w_n$.
  2. Aja TM_A jokaisella osalla $w_i, i \in \{1, \ldots, n\}$.
  3. Jos jokainen ajo hyväksyy, hyväksy.

(d) Leikkaus

- Syötteellä w:
  1. Aja TM_A ja TM_B syötteellä w.
  2. Jos molemmat Turingin koneista hyväksyvät, hyväksy.