TKT20005 Laskennan mallit Viikko5

Tehtävä 1. Turingin kone.

Tässä tehtävässä harjoitellaan lukemaan Turingin koneen tilakaaviota.
Esitä allaolevan Turingin koneen laskenta syötteellä 0111 luettelemalla laskennan aikana syntyvät tilanteet, kuten luentomateriaalin sivulla 181 (tai Sipserin sivun 172 alaosassa).

$q_0 0111$	$\vdash$	q0	0		1		1		1		
	$\vdash$		a	q1	1		1		1		
	$\vdash$		a		1		1		1	q1	
	$\vdash$		a		1		1	q2	1		
	$\vdash$		a	q3	1		1		B		
	$\vdash$	q4	a		1		1		B		
	$\vdash$		a	q0	1		1		B		
	$\vdash$		a		b		1	q1	B		
	$\vdash$		a		b	q2	1		B		
	$\vdash$		a	q3	b		B		B		
	$\vdash$		a		b	q5	B		B		
	$\vdash$		a		b		B		B	q5	
	$\vdash$		a		b		B	q6	B		
	$\vdash$		a		b	q9	B		X		
	$\vdash$		a	q9	b		B		X		
	$\vdash$		a		X	q10	B		X		
	$\vdash$		a		X		B		X	q10	$\rightarrow q_{acc}$

Tehtävä 2. Laskentaongelman ratkaisu Turingin koneella.

Turingin konetta voi luonnollisella tavalla käyttää paitsi kielen tunnistamiseen myös funktion laskemiseen. Määritellään, että Turingin kone laskee funktion $f\colon \Sigma^* \to \Sigma^*$, jos millä tahansa syötemerkkijonolla w se pysähtyy tilanteeseen, jossa nauhalla on alussa merkkijono $f(w)$ ja sen jälkeen pelkkiä tyhjämerkkejä.
Tarkastelemme seuraavassa binäärikoodattujen luonnollisten lukujen käsittelemistä Turingin koneella. Kun $n \in \mathbb{N}$ on luonnollinen luku, käytämme merkintää $\langle\, n \,\rangle$ tarkoittamaan luvun n binääriesitystä (aakkoston $\Sigma = \{\, 0, 1 \,\}$ merkkijonona) ilman turhia etunollia. Esim. $\langle\, 12 \,\rangle = 1100$.
Määritellään lukujen binääriesityksille edeltäjäfunktio p niin, että jos $w = \langle\, n+1 \,\rangle$ jollain $n \in \mathbb{N}$, niin $p(w) = \langle\, n \,\rangle$. Jos w ei ole muotoa $\langle\, n+1 \,\rangle$ (eli $w = \varepsilon$ tai $w = \langle\, 0 \,\rangle$ tai w:ssä on ylimääräisiä etunollia), määritellään $p(w) = \varepsilon$.
Kuvaile yksinauhainen Turingin kone, joka laskee funktion p. Sopiva tarkkuus esitykselle on suunnilleen vastaava kuin luentomateriaalin sivulla 186. Toisin sanoen selitä, miten nauhaa käytetään ja muut mahdolliset korkean tason periaatteet, mutta älä mene yksittäisten tilojen ja tilasiirtymien tasolle.

Kone toimii kolmessa vaiheessa:

(1) **Kelpoisuustesti ja tyhjätapaukset:**

- Jos ensimmäinen solu on tyhjä (eli $w = \varepsilon$): pysähdy koska nauha on valmiiksi tyhjä
- Jos ensimmäinen merkki on 0:
  - Jos seuraava solu on tyhjä (eli $w = 0$): pyyhi tuo 0 ja pysähdy (tulos ε)
  - Muuten (pituus ≥ 2 ja johtava 0 eli ylimääräisiä etunollia): pyyhi koko syöte (korvaa kaikki 0/1:t tyhjällä) ja pysähdy, tulos ε
- Muuten ensimmäinen merkki on 1 (kelvollinen koodi bin$(n{+}1)$): jatka vaiheeseen 2

(2) **Vähennä yksi (binääri -1 oikealta):**
Siirry syötteen loppuun (kulje oikealle kunnes tyhjö, sitten yksi vasemmalle)

- Jos solu on 1: kirjoita 0 ja siirry vaiheeseen 3
- Jos solu on 0: toista, kunnes löydät 1:n vasemmalta:
  - jokaiselle peräkkäiselle 0:lle: kirjoita 1 ja siirry yhden askeleen vasemmalle;
  - ensimmäisen 1:n kohdalla: kirjoita 0 ja siirry vaiheeseen 3

Tässä vaiheessa nauhalla on bin(n), mutta tapauksessa $w = 10^k$ vasemmalle jäi yksi johtava nolla

(3) Poista johtava nolla tarvittaessa:
Palaa vasempaan reunaan (kulje vasemmalle tyhjään, sitten yksi oikealle)

- Jos vasemmanpuoleisin merkki on 1: pysähdy.
- Jos vasemmanpuoleisin merkki on 0:
  - jos seuraava solu on tyhjä (jäännös ”0”, tapaus $w = 1$): pysähdy ja jätä ”0”paikalleen
  - muuten (pituus ≥ 2 ja johtava 0): poista vasemman laidan 0 siirtämällä sanaa yhden solun verran vasemmalle:
    * kulje oikealle ja jokaisessa solussa kopioi sen sisältö edelliseen soluun (ylikirjoita); jatka kunnes kohtaat tyhjän;
    * mene yksi vasemmalle ja kirjoita tyhjä (tyhjää viimeinen kopio)
    * palaa vasempaan reunaan ja pysähdy

testejä

- $w = \varepsilon \ \Rightarrow \ \varepsilon$ (Vaihe1)
- $w = 0 \ \Rightarrow \ \varepsilon$ (Vaihe 1)
- $w = 1 \ \Rightarrow \ 0$ (Vaihe 2 tuottaa ”0”; Vaihe 3 tunnistaa yksittäisen nollan)
- $w = 10 \ \Rightarrow \ 01 \overset{\text{V3}}{\Rightarrow} 1$
- $w = 101000 \ \Rightarrow \ 100111$ (3 ei poista mitään)

Tehtävä 3. Turing-tunnistettavuus.

Tässä tehtävässä tarkastellaan, mitä Turing-tunnistettavuus tarkoittaa, ja harjoitellaan, kuinka jokin kieli todistetaan Turing-tunnistettavaksi.
Tarkastellaan kieltä

$$A = \{ \langle M, w, q \rangle \mid M \text{ on Turingin kone, joka syötteellä } w \text{ menee ainakin kerran tilaan } q \} .$$

Todista, että kieli A on Turing-tunnistettava.

Todistus. Määritellään Turingin kone R

Syöte: koodattu kolmikko $\langle M, w, q \rangle$

1. Tarkista, että q on M:n tilajoukon koodaus. Jos ei ole hylkää
2. Simuloi M:ää syötteellä w askel askeleelta universaalilla simulaattorilla Pidä kirjaa M:n nykyisestä tilasta.
   - Jos simulaation jossain askeleessa M:n tila on q, hyväksy
   - Jos M pysähtyy hyväksyen tai hyläten ennen kuin tila q esiintyy, hylkää.
   - Muussa tapauksessa M jatkaa ikuisesti eikä koskaan vieraile q:ssa.

Jos $\langle M, w, q \rangle \in A$, niin simuloinnissa on jokin hetki, jolloin M:n tila on q, ja R hyväksyy

Jos $\langle M, w, q \rangle \notin A$, on kaksi tapausta: joko M pysähtyy koskaan vierailematta q:ssa (eli R hylkää), tai M ei pysähdy eikä vieraile q:ssa, jolloin R jatkaa simulaatiota ikuisesti (ei hyväksy). Eli R hyväksyy täsmälleen A:n syötteet, joten A on Turing-tunnistettava. ∎

Tehtävä 4. Turing-ratkeavuus.

Tässä tehtävässä tarkastellaan, mitä Turing-ratkeavuus tarkoittaa, ja harjoitellaan, kuinka jokin kieli todistetaan Turing-ratkeavaksi.

Kieli T koostuu kaikista sellaisista pareista $\langle M, w \rangle$, joissa M on Turingin kone, w jokin sen syöte ja nämä toteuttavat ehdon, että koneen M laskennassa syötteellä w ainakin yksi tila toistuu ainakin kaksi kertaa.

Todista, että kieli on ratkeava.

$$T = \{\langle M, w\rangle \mid M\text{:llä syötteellä } w \text{ jokin ohjaustila esiintyy vähintään kahdesti}\}$$

T on Turing-ratkeava

Todistus. Rakennetaan päättävä kone D syötteelle $\langle M, w\rangle$
Laske $N := |Q_M|$, siis tilojen lukumäärä M:n kuvauksesta
Simuloi M:ää syötteellä w enintään N siirtymää pitäen kirjaa ohjaustiloista missä käyty

- Jos simulaation aikana jokin tila toistuu, hyväksytään
- Jos M pysähtyy ennen toistoa, hylätään
- Simulaatio on rajattu N askeleeseen jolloin D pysähtyy aina

jos jokin tila toistuu jossain vaiheessa, kyyhkyslakkaperiaate takaa, että toisto tapahtuu viimeistään N askeleen kuluessa (koska $N+1$ tilakäyntiä N:ssä tilassa pakottaa kahden käynnin osumaan samaan tilaan).

$\rightarrow$ Siksi D havaitsee toiston ja hyväksyy. Jos toistoa ei ole, silloin M pysähtyy ilman toistoa (hylkäys yllä), tai muuten N askelta myöhemminkään toistoa ei esiinny — mikä on mahdotonta, joten tätä tapausta ei synny. Siis D päättää kielen T ■