TIEA241 Automaatit ja kieliopit harjoitustehtäviä

Antti Valmari
Jyväskylän yliopisto, Informaatioteknologian tiedekunta

22. lokakuuta 2018

1 Kertaustehtäviä

1. Olkoot $A = \{2,3,5,7\}$ ja $B = \{1,3,6\}$. Muodosta seuraavat joukot.

(a) $A \cup B$ (b) $A \cap B$ (c) $A \setminus B$ (d) $A \times B$ (e) $\{x \in A \mid x \neq 5\}$

2. Tee verkossa osoitteessa

`http://users.jyu.fi/~ava/t_propositiot1.html`

oleva tehtävä. Yritin tehdä osoitteesta klikattavan linkin. Itselläni se toimi jos ja vain jos veppiselain oli valmiiksi auki (jollei ollut, selain kyllä aukeni, mutta kenttiin ei voinut kirjoittaa). Jollei sinulla toimi, kopioi osoite selaimesi osoiteriville.

Kyseinen ohjelma ei tiedä kuka olet, joten se ei kirjaa pisteitä tilillesi. Siitä huolimatta sinun ei tarvitse vaivautua ottamaan vastauksiasi talteen. Tarkoituksena on, että tehtyäsi tehtävän osaat asian niin hyvin, että samantapaiset tehtävät sujuvat leikiten. Siksi siinä on monta alakohtaa, mutta ne ovat enimmäkseen helppoja. Jos joudut demoissa näyttämään vastauksesi, ei haittaa vaikka kaikki kohdat eivät ihan sujuvasti menisikään.

Jos koet tarvitsevasi apua ohjelman käytössä, kokeile

`http://users.jyu.fi/~ava/yleista.html`

3. Suomenna seuraavat väittämät olettaen, että $A \subseteq \mathbb{Z}$ eli A:n alkiot ovat kokonaislukuja. Koeta löytää suomennokset, jotka viisivuotiaskin ymmärtää. Voit tarvittaessa päätellä. Esimerkiksi älä suomenna $\forall x \in A : \forall y \in A : x = y$ näin: ”kaikilla x, jotka ovat joukossa A pätee, että kaikilla y, jotka ovat joukossa A pätee, että x on yhtäsuuri kuin y”. Suomenna se esim. näin: ”joukon A kaikki alkiot ovat keskenään yhtäsuuret”.

(a) $\forall x \in A : x < 0$
(b) $(\forall x \in A : x < 0) \wedge (\forall x \notin A : x \geq 0)$
(c) $\forall x \in A : x \leq 3$
(d) $3 \in A \wedge \forall x \in A : x \leq 3$
(e) $\forall x \in A : \exists y \in A : x < y$
(f) $\forall x \in A : \forall y \in A : x < y$
(g) $\forall x \in A : \forall y \in A : x \leq y$
(h) $\exists z \in A : \forall x \in A : \forall y \in A : x \leq y$
(i) $\exists x \in A : \exists y \in A : \exists z \in A : x \neq z$
(j) $\forall x \in A : \exists y \in \mathbb{Z} : x = y + y$

2 Äärelliset automaatit ja säännölliset kielet

2.1 Deterministiset äärelliset automaatit

4. Aakkosto on $\{\mathtt{a},\mathtt{b}\}$. Piirrä DFA:t, jotka hyväksyvät seuraavat kielet.

(a) Ne merkkijonot, joiden viimeinen merkki on a.

(b) Ne merkkijonot, joissa on enintään kaksi a:ta.

(c) Ne merkkijonot, joissa a esiintyy jossain kohdassa kahdesti peräkkäin.

5. Aakkosto on $\{\mathtt{a},\mathtt{b},\mathtt{c}\}$. Sovimme, että $\mathtt{a} < \mathtt{b} < \mathtt{c}$. Piirrä DFA:t, jotka hyväksyvät seuraavat kielet.

(a) $\{a_1 \cdots a_n \mid \forall i; 1 \leq i < n : a_i \neq a_{i+1}\}$

(b) $\{a_1 \cdots a_n \mid \forall i; 1 \leq i < n : a_i = a_{i+1}\}$

(c) $\{a_1 \cdots a_n \mid \forall i; 1 \leq i < n : a_i \leq a_{i+1}\}$

6. Aakkosto on $\{\mathtt{a},\mathtt{b},\mathtt{c}\}$. Mitkä seuraavista joukoista ovat äärettömiä?

(a) $\{n \in \mathbb{Z}^+ \mid n < 43\}$

(b) $\{n \in \mathbb{Z} \mid n < 43\}$

(c) $\{n \in \mathbb{Q}^+ \mid n < 43\}$

(d) $\{3{,}14159\ldots\}$ (aaltosulkeiden välissä on luvun π desimaaliesitys)

(e) $\{n \bmod 7 \mid n \in \mathbb{Z}^+\}$

(f) Ne merkkijonot, joiden pituus on vähintään 43.

(g) Ne merkkijonot, joiden pituus on enintään 43.

(h) Ne merkkijonot, joissa ei esiinny muita merkkejä kuin a:ta.

(i) $\{\mathtt{a}, \mathtt{a}\varepsilon, \mathtt{a}\varepsilon\varepsilon, \mathtt{a}\varepsilon\varepsilon\varepsilon, \ldots\}$

(j) Ne DFA:t, jotka hyväksyvät kielen $\emptyset$.

7. Aakkosto on $\{\mathtt{a},\mathtt{b}\}$. Piirrä niin pienet seuraavien ehtojen mukaiset DFA:t kuin keksit. DFA:n koon mittana on tilojen määrän ja kaarten määrän summa.

(a) Se hyväksyy täsmälleen 15 merkkijonoa.

(b) Se hyväksyy mahdollisimman monta merkkijonoa.

(c) Se hyväksyy ne ja vain ne merkkijonot, joiden kolmanneksi viimeinen merkki on a.

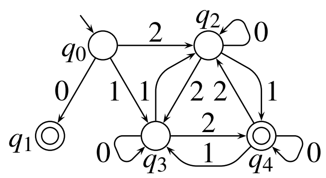
8. Aakkosto on $\{\mathsf{a}\}$.

(a) Luettele kaikki äärelliset kielet, jotka voidaan hyväksyä DFA:lla, jossa on yksi tila.

(b) Luettele kaikki äärelliset kielet, jotka voidaan hyväksyä DFA:lla, jossa on kaksi tilaa.

(c) Ilmoita n:n funktiona niiden äärellisten kielten määrä, jotka voidaan hyväksyä DFA:lla, jossa on n tilaa.

9.



δ	1	2	3	4	5	6
a	–	1	2	2	6	–
b	4	–	3	3	–	3
c	–	–	–	–	–	–

(a) Luettele oheisen DFA:n osat Q, Σ, δ, $\hat{q}$ ja F.

(b) Piirrä DFA, jolle $Q = \{1,2,3,4,5,6\}$, $\Sigma = \{\mathsf{a},\mathsf{b},\mathsf{c}\}$, δ on taulukon mukainen, $\hat{q} = 5$ ja $F = \{1,6\}$.

(c) Piirrä DFA, jolle $Q = \{0,1,2\} \times \{0,1\}$, $\Sigma = \{\mathsf{a},\mathsf{b}\}$, $\delta((i,j),\mathsf{a}) = ((i+1) \bmod 3, j)$, $\delta((i,j),\mathsf{b}) = (i,(j+1) \bmod 2)$, $\hat{q} = (1,0)$ ja $F = \{(i,j) \in Q \mid i+j = 2\}$.

10.

(a) Kuinka monta sellaista DFA:ta on olemassa, joille $Q = \{1,2\}$ ja $\Sigma = \{a,b\}$?

(b) Ilmoita ainakin 15 eri kieltä, jotka ne hyväksyvät, tai perustele, että ne eivät hyväksy niin montaa eri kieltä.

11. Oletetaan, että on olemassa ainakin yksi DFA, joka hyväksyy kielen K. Ilmaise sellainen K:ta koskeva ominaisuus φ, että jos φ pätee, niin jokaisessa K:n hyväksyvässä DFA:ssa joko on turha tila tai sen tilasiirtymäfunktio ei ole täysi, ja jos φ ei päde, niin on olemassa K:n hyväksyvä DFA jossa ei ole turhia tiloja ja jonka tilasiirtymäfunktio on täysi. Ominaisuus pitää ilmaista käyttämättä DFA:n, DFA:n tilan tms. käsitteitä.

12. Määrittelemme, että DFA on *täysi* jos ja vain jos sen tilasiirtymäfunktio on täysi.

(a) Määrittele uusi käsite *täysturha tila* siten, että se muistuttaa turhan tilan käsitettä, mutta jokaiselle kielelle K, joka voidaan hyväksyä DFA:lla, on olemassa täysi DFA, joka hyväksyy K:n ja jossa ei ole täysturhia tiloja.

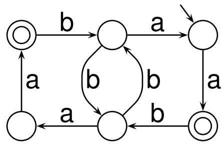
(b) Luonnostele tehokas algoritmi, joka löytää täysturhat tilat.

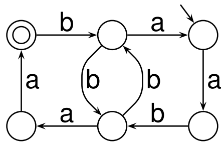
(c) Luonnostele tehokas algoritmi, joka löytää turhat mutta ei täysturhat tilat.

(d) Luonnostele algoritmi, jolla mikä tahansa täysi DFA voidaan muuntaa saman kielen hyväksyväksi täydeksi DFA:ksi jossa on korkeintaan yksi turha tila.

2.2 Determinististen äärellisten automaattien minimointi

13. Minimoi oheiset DFA:t muodostamalla lohkot, halkaisemalla niitä ja muodostamalla lopputulos. Hopcroftin ja uudempia tehostuskikkoja ei tarvitse käyttää.





14. Poistetaan lohkomisalgoritmista vaihe 1 eli turhien tilojen poisto. Mitkä seuraavista väitteistä ovat aina totta? Perustele ne lyhyesti ja anna muille vastaesimerkit.

(a) Jos $q =\sigma\Rightarrow q'$, niin $[q]$ $[=\sigma\Rightarrow]$ $[q']$.

(b) Jos $[q]$ $[=\sigma\Rightarrow]$ $[q']$, niin $q =\sigma\Rightarrow q'$.

(c) Jos kaksi tilaa joutuu eri lohkoihin, niin ne hyväksyvät eri kielet.

(d) Jos kaksi tilaa pysyy loppuun saakka samassa lohkossa, niin ne hyväksyvät saman kielen.

(e) Jos δ on täysi, niin myös $[\delta]$ on täysi.

(f) Jos $[\delta]$ on täysi, niin myös δ on täysi.

(g) Turhat tilat syötteessä aiheuttavat turhia tiloja lopputuloksessa.

(h) Jos lopputuloksessa on turhia tiloja, niin myös syötteessä oli turhia tiloja.

2.3 Epädeterministiset äärelliset automaatit

15.

(a) Piirrä mahdollisimman hyvin tosielämän pankkiautomaattia vastaava DFA tai NFA käyttäen seuraavia aakkosia:

e	pankkiautomaatti ilmoittaa, että tilillä **e**i ole tarpeeksi rahaa nostoon
n	asiakas näppäilee **n**ostopyynnön
r	pankkiautomaatti antaa **r**ahaa
s	asiakas työntää pankkikortin **s**isään
t	asiakas näppäilee **t**unnuslukunsa
u	pankkiautomaatti työntää pankkikortin **u**los
v	pankkiautomaatti ilmoittaa, että tunnusluku oli **v**äärä
y	pankkiautomaatti ilmoittaa, ettei saa **y**hteyttä pankkiin

(b) Aakkosto on $\{a, b\}$. Piirrä mahdollisimman pieni NFA, joka hyväksyy ne ja vain ne merkkijonot, joissa on mielivaltaisessa järjestyksessä täsmälleen neljä a:ta ja mielivaltainen määrä b:tä; tai mielivaltaisessa järjestyksessä täsmälleen neljä b:tä ja mielivaltainen määrä a:ta.

16.

(a) Ilmaise NFA:n suurin mahdollinen kaarten määrä $|Q|$:n ja $|\Sigma|$:n funktiona.

(b) Olkoot $D = (Q, \Sigma, \delta, \hat{q}, F)$ DFA ja N on D tulkittuna NFA:ksi. Ilmoita N:n osat (tilojen joukko jne.)

(c) Olkoon N NFA, jonka aakkosto on $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$, jonka jokaisesta tilasta pääsee jokaiseen tilaan ε-kaaria pitkin, ja jossa on ainakin yksi a-kaari, ainakin yksi b-kaari ja ainakin yksi c-kaari. Mikä on $\mathcal{L}(N)$? Ilmoitetuilla tiedoilla on kaksi vaihtoehtoa.

17. Miten voidaan tehdä NFA, jonka hyväksymä kieli on syötteenä saadun NFA:n hyväksymät merkkijonot takaperin?

18. Tehtävän tavoitteena on osoittaa, että NFA:n komplementointi ei voi olla aina nopea. Olkoon $n > 1$ ja aakkosto on $\{a_1, \ldots, a_{n-1}\}$.

(a) Piirrä NFA N, jossa on n tilaa siten, että $\mathcal{L}(N)$ on niiden merkkijonojen joukko, joissa jokin aakkoston merkki ei esiinny. Esim. kun $n = 4$, niin $a_3a_1a_3 \in \mathcal{L}(N)$ koska a_2 ei esiinny siinä, mutta $a_3a_1a_3a_2 \notin \mathcal{L}(N)$ koska jokainen merkeistä a_1, a_2 ja a_3 esiintyy siinä.

(b) Piirrä NFA $\overline{N}$, jossa on 2^{n-1} tilaa siten, että $\mathcal{L}(\overline{N}) = \Sigma^* \setminus \mathcal{L}(N)$ eli niiden merkkijonojen joukko, joissa jokainen aakkoston merkki esiintyy.

(c) Perustele, että (b)-kohdan kielelle ei ole pienempää NFA:ta. Perustelusi ei tarvitse olla matemaattinen.

(d) Perustele, että ei ole olemassa aina nopeaa algoritmia, joka saatuaan syötteeksi NFA:n muodostaa NFA:n, jonka hyväksymä kieli on syötteenä saadun NFA:n kielen komplementti. Perustelusi ei tarvitse kuulostaa matemaattiselta.

19. Oheinen ohjelma saa syötteeksi luvun n ja kokonaislukutaulukon A, jolle $1, \ldots, n$ ovat laillisia indeksejä.

```
if n > 1 then
  i := 1
  j := 2
  while j ≠ i do
    if A[j] = A[i] + 1 then i := j
    j := j + 1
    if j > n then j := 1
```

(a) Jokaiselle n suunnittele sellainen taulukon A sisältö, että ohjelma pysähtyy mahdollisimman nopeasti.

(b) Mikä on (a)-kohdan suoritusaika n:n funktiona ilmaistuna O-merkinnällä?

(c) Jokaiselle n suunnittele sellainen taulukon A:n sisältö, että ohjelma laskee mahdollisimman kauan.

(d) Mikä on (c)-kohdan suoritusaika n:n funktiona ilmaistuna O-merkinnällä?

20.

(a) Kirjoita ohjelma tai pseudokoodi, joka saa syötteeksi kaksi merkkijonoa ja vastaa onko ensin annettu leksikograafisessa järjestyksessä pienempi kuin jälkimmäinen. Leksikograafinen järjestys vastaa tuttua aakkosjärjestystä. Esimerkiksi `a` $<$ `aabb` $<$ `aba` $<$ `b` $<$ `bab`.

(b) Kirjoita ohjelma tai pseudokoodi, joka saa syötteeksi kaksi merkkijonoa ja vastaa onko ensin annettu shortlex-järjestyksessä pienempi kuin jälkimmäinen. Shortlex-järjestys on muuten kuten leksikograafinen, mutta lyhyempi tulee aina ennen pitempää. Esimerkiksi `a` $<$ `b` $<$ `aba` $<$ `bab` $<$ `aabb`.

(c) Aakkosto on $\{$`a`,`b`$\}$. Kuinka monta merkkijonoa on merkkijonojen `aa` ja `bb` välissä leksikograafisessa järjestyksessä? Entä shortlex-järjestyksessä?

21. Miten NFA:sta voidaan poistaa ε-kaaret ilman, että hyväksytty kieli muuttuu? Ratkaisun pitää toimia polynomiajassa, mikä karkeasti sanottuna tarkoittaa, että se ei saa olla hyvin tehoton. Ratkaisua ei tarvitse esittää pseudokoodina eikä ohjelmana, vaan sanallinen selitys riittää.

22. DFA:ille pätee pumppauslemman kaltainen lause, joka koskee hylättyjä eikä hyväksyttyjä merkkijonoja. Ilmaise ja perustele se. Muista, että DFA:n tilasiirtymäfunktio ei välttämättä ole täysi.

23. Kuinka työlästä on selvittää, kuinka pitkä on lyhin merkkijono, jonka annettu NFA hyväksyy? Perustele vastauksesi kuvailemalla algoritmi, joka toimii lupaamassasi ajassa. Saat erityiskehut, jos lisäksi perustelet, että nopeampaa algoritmia ei ole olemassa.

24. Miten voidaan tehdä DFA, jonka hyväksymä kieli on syötteenä saadun DFA:n hyväksymät merkkijonot takaperin? Voiko mikään algoritmi tälle tehtävälle olla aina nopea?

25. Miten tuloautomaatin käsitettä hieman muuntamalla voidaan muodostaa syöteautomaattien hyväksymien kielten unionin hyväksyvä automaatti siten, että jos syöteautomaatit ovat deterministisiä, niin tuloskin on? Vertaa tätä tapaa siihen, että ensin lasketaan unioniautomaatti siten kuin luennoilla esitettiin ja sitten se muunnetaan deterministiseksi. Vertaa ainakin suoritusaikaa, mielellään myös toimintaperiaatteita.

2.4 Säännölliset lausekkeet

26. Osoita vastaesimerkeillä, että mikään seuraavista ei aina päde. Lisäksi anna jokaiselle esimerkki, jossa $a \in \mathcal{L}(r)$, $b \in \mathcal{L}(s)$, ja yhtäsuuruus pätee.

(a) $\mathcal{L}(rs) = \mathcal{L}(sr)$

(b) $\mathcal{L}((r \mid s)^*) = \mathcal{L}(r^* \mid s^*)$

(c) $\mathcal{L}((rs)^*) = \mathcal{L}(r^*s^*)$

27. Oletamme, että r tuottaa n erilaista merkkijonoa ja s tuottaa m erilaista merkkijonoa. Rajoitumme tapaukseen $n > 0$ ja $m > 0$. Valitse ym. merkkijonot siten, että rs tuottaa (a) mahdollisimman monta (b) mahdollisimman vähän erilaisia merkkijonoja. Valitse aakkosto siten, että saat (a)-kohdassa mahdollisimman monta ja (b)-kohdassa mahdollisimman vähän erilaisia merkkijonoja. Ilmaise tuotettujen merkkijonojen määrä kummassakin tapauksessa n:n ja m:n funktiona.

Ratkaise tämä tehtävä ensin siten, että $n = m = 1$, sitten $n = 1$ ja $m = 2$, sitten $n = 1$ ja $m = 3$ ja niin edelleen, kunnes hoksaat, mikä on lopputulos kun $n = 1$ ja m on mitä tahansa paitsi 0. Sitten valitse $n = m = 2$, $n = 2$ ja $m = 3$ ja niin edelleen. Koeta löytää yleiset kaavat, jotka pätevät kun n ja m ovat mitä tahansa paitsi 0.

28. Tarkastellaan muunnosta NFA $\rightsquigarrow$ säännöllinen lauseke.

(a) Arvioi suurimman mahdollisen välivaiheen tilojen määrää ja kaarten määrää syötteenä annetun NFA:n tilojen määrän ja kaarten määrän funktiona Θ-merkinnällä. Jos et osaa Θ-merkintää, niin arvioi niin hyvin kuin osaat, esimerkiksi sanoilla ”valtava”, ”suuri”, ”melko pieni” tai ”pieni”. Vihje: on helppo löytää esimerkki, jossa tulos on Θ-merkinnän tarkkuudella niin huono kuin se voi olla.

(b) Oletetaan, että tilassa on paikallissilmukat nimillä r ja s sekä kaari toiseen tilaan nimellä t. Vertaa lopputuloksia, jos ensin sovelletaan tuplakaaren poistoa ja sitten paikallissilmukan poistoa siihen, että sovelletaan paikallissilmukan poistoa kahdesti. Perustele, että syntyvät säännölliset lausekkeet tuottavat saman kielen.

3 Yhteysriippumattomat kieliopit

29. Tee verkossa osoitteessa

`http://users.jyu.fi/~ava/t_CFG.html`

oleva tehtävä puoliväliin.

30. Tee verkossa osoitteessa

`http://users.jyu.fi/~ava/t_CFG.html`

oleva tehtävä loppuun.

31. Laadi yhteysriippumattomat kieliopit seuraaville kielille. Aakkosto on $\{\mathtt{a},\mathtt{b}\}$. Perustele, että vastauksesi on oikein. Voit testata vastauksiasi osoitteessa

`http://users.jyu.fi/~ava/t_31.html`

Vihje: olkoon merkkijonon *samuuskohta* sellainen i, että merkkijonon i ensimmäisen merkin joukossa on täsmälleen yhtä monta `a`:ta kuin `b`:tä. Jokaiselle merkkijonolle 0 on samuuskohta. Merkkijonon `abbaaababb` samuuskohdat ovat 0, 2, 4 ja 10. Jos merkkijonon $a_1 \cdots a_n$ ensimmäinen merkki eli a_1 on `a`, ja jos i on toiseksi pienin samuuskohta (pienin on 0), niin mikä merkki on a_i? Jos merkkijono kuuluu (b)-kohdan kieleen, niin mitä voidaan sanoa viimeisen samuuskohdan jälkeisestä osasta?

(a) Ne merkkijonot, joissa on sama määrä `a`:ta kuin `b`:tä.

(b) Ne merkkijonot, joissa on enemmän `a`:ta kuin `b`:tä.

(c) Ne merkkijonot, joissa on eri määrä `a`:ta kuin `b`:tä.

32. Suunnittele tehokas algoritmi, jolle annetaan yhteysriippumaton kielioppi, ja joka kertoo,

(a) tuottaako se yhtään merkkijonoa.

(b) tuottaako se tyhjän merkkijonon.

(c) kuinka pitkä on lyhin sen tuottama merkkijono.

33. Tee verkossa osoitteessa

`http://users.jyu.fi/~ava/t_lausekkeet.html`

oleva tehtävä.

4 Laskettavuus ja Turingin koneet

34. Mitkä funktiot alla olevat aliohjelmat `for1` ja `for2` laskevat? Aliohjelman `for3` laskema funktio voi olla vaikea kuvailla tarkasti, mutta pystytkö sanomaan mitään sen käyttäytymisestä kun `h` kasvaa? Mitä mieltä olet suoritusaikaa koskevasta nyrkkisäännöstä ”yksi sisäkkäinen `for`-silmukka — lineaarinen; kaksi sisäkkäistä `for`-silmukkaa — neliöllinen; kolme sisäkkäistä `for`-silmukkaa — kuutiollinen”?

```
unsigned for1( unsigned n ){
  unsigned tulos = n;
  for( unsigned i = 0; i < n; ++i ){ ++tulos; }
  return tulos;
}

unsigned for2( unsigned m ){
  unsigned tulos = 1;
  for( unsigned j = 0; j < m; ++j ){
    unsigned n = tulos;
    for( unsigned i = 0; i < n; ++i ){ ++tulos; }
  }
  return tulos;
}

unsigned for3( unsigned h ){
  unsigned tulos = 1; ++tulos;
  for( unsigned k = 0; k < h; ++k ){
    unsigned m = tulos;
    tulos = 1;
    for( unsigned j = 0; j < m; ++j ){
      unsigned n = tulos;
```

```
      for( unsigned i = 0; i < n; ++i ){ ++tulos; }
    }
  }
  return tulos;
}
```

35. *Aito* `for`*-silmukka* on muotoa `for( i=a; i<y; ++i ){ ... }`, missä `i`:n arvoa ei vähennetä eikä `y`:n arvoa kasvateta silmukan vartalossa `{ ... }`. Vartalossa saa olla silmukan keskeyttäviä komentoja kuten `break` tai `i=y;`. Ohjelma on *alkeisrekursiivinen* (*primitive recursive*) jos ja vain jos sen kaikki silmukat ovat aitoja `for`-silmukoita eikä siinä ole rekursiota eikä muita naamioituja silmukoita. Funktio on alkeisrekursiivinen jos ja vain jos se voidaan laskea alkeisrekursiivisella ohjelmalla.

(a) Ilmoita yläraja aidon `for`-silmukan kierrosten määrälle.

(b) Voiko alkeisrekursiivinen funktio olla aidosti osittainen funktio?

(c) Perustele, että on olemassa alkeisrekursiivisia ohjelmia, jotka saavat syötteekseen yhden luonnollisen luvun n ja joiden suoritusaika kasvaa rajusti n:n funktiona.

(d) Perustele, että jos silmukan kierrosten määrälle on tiedossa yläraja juuri ennen kuin silmukan kiertäminen alkaa, silmukka voidaan korvata aidolla `for`-silmukalla.

(e) Perustele että jokainen n:n funktio, joka voidaan laskea ajassa $O(n^{1000})$, on alkeisrekursiivinen.

(f) On olemassa rekursiivisia täysiä n:n funktioita, jotka eivät ole alkeisrekursiivisia. Ackermannin funktion yksiargumenttinen versio on sellainen. Perustele, että jokainen sellainen funktio on todella hidas laskea äärettömän monella n.

36. *Diagonalisointi* on päättelytapa, jota on käytetty monessa paikkaa joukko-opissa ja teoreettisessa tietojenkäsittelytieteessä. Se on tuottanut erilaisia tuloksia riippuen tilanteesta, jossa sitä sovelletaan. Tässä tehtävässä kerrataan diagonalisoinnin sovelluksia ja ehkä tavataan uusiakin. Kohdat ovat aikajärjestyksessä sen mukaan milloin varsinainen tulos julkaistiin tai tuli muuten tiedeyhteisön tietoon. Alla olevat kuvaukset eivät vastaa tarkoin alkuperäisjulkaisuja, koska monessa tapauksessa tulosten yksityiskohdat ovat hioutuneet selkeämmiksi alkuperäisjulkaisun jälkeen.

(a) Julkaisussaan vuodelta 1891 Georg Cantor esitti seuraavan idean. Kuvitellaan, että reaalilukujen desimaaliesityksiä on pantu päättymättömään jonoon kuten alla. Niille reaaliluvuille, joilla on kaksi eri desimaaliesitystä (esim. 6,999... = 7,000...), käytetään sitä, jossa on päättymätön jono nollia. Aloitetaan vastauksen muodostaminen kirjoittamalla "0,". Otetaan ensimmäisen luvun ensimmäinen desimaali. Jos se on 0, laitetaan vastauksen perään 1, muutoin 0. Sitten otetaan toisen luvun toinen desimaali ja käsitellään se samalla tavalla. Näin jatketaan loputtomasti. Kuvan esimerkistä syntyy desimaaliluku 0,00010.... Voiko näin syntyvä desimaaliluku olla jonossa? Mikä nykyisin laajasti tunnettu mutta Cantorin aikalaisia yllättänyt ja järkyttänyt seuraus tällä on?

$$\begin{array}{l} 0{,}3333333\ldots \\ 1{,}4142135\ldots \\ 3{,}1415926\ldots \\ 7{,}0000000\ldots \\ 2{,}7182818\ldots \\ \quad\vdots \qquad \ddots \end{array}$$

(b) Tämä idea on samasta julkaisusta, mutta muokattu nykyaikaiseen muotoon. Tarkastellaan mielivaltaista joukkoa A ja mielivaltaista funktiota $f : A \mapsto 2^A$, joka ottaa A:n alkion ja tuottaa A:n osajoukon (siis jos $a \in A$, niin $f(a) \subseteq A$). Sitten määritellään joukko $C = \{a \in A \mid a \notin f(a)\}$. Voiko olla olemassa $c \in A$ siten, että $C = f(c)$? Mitä tästä voidaan päätellä? Jos valitaan $A = \Sigma^*$ missä Σ on äärellinen epätyhjä joukko, niin mitä tulos kertoo kielien eli Σ^*:n osajoukkojen määrästä?

(c) Vuonna 1901 Bertrand Russell määritteli joukon $R = \{x \mid x \notin x\}$ ja kysyi, päteekö $R \in R$ ja päteekö $R \notin R$. Tämä tunnetaan *Russelin paradoksina*. Mitä Russelin paradoksi todisti?

(d) Kurt Gödel suunnitteli hyvin monimutkaisen luonnollisista luvuista puhuvan kaavan φ (julkaisu 1931). Se voidaan suomentaa seuraavasti: ”kaavalle φ ei ole olemassa todistusta”. Jos oletamme, että φ on totta, niin mitä siitä seuraa? Jos oletamme, että φ ei ole totta, niin mitä siitä seuraa? (Gödel epäilemättä tunsi tämän päättelyn, mutta hän eli aikakautena, jolloin totuudesta puhuminen tällä tavalla katsottiin epätieteelliseksi. Siksi Gödelin päättely oli monimutkaisempi. Se johti tulokseen, että joko kumpaakaan kaavoista φ ja $\neg\varphi$ ei voi todistaa, tai matematiikka on eräässä epästandardissa mielessä ristiriitaista. Vuonna 1936 Barkley Rosser muunsi kaavaa ja päättelyä siten, että siirryttiin tavalliseen ristiriitaisuuteen.)

(e) Luennolla esitetyt todistukset pysähtymistesterin olemattomuudelle eivät juurikaan muistuta Turingin todistusta vuodelta 1936. Turingin todistus oli jonkinlainen sekoitus (a)-kohdan ajattelutapaa ja luennoissa esitettyjä ajatuksia. On kuitenkin mahdollista todistaa pysähtymistesterin olemattomuus seuraamalla (a)-kohtaa suoraviivaisesti. Pääohjelma tulostaa `0,`. Sitten se tuottaa kaikki merkkijonot shortlex-järjestyksessä ja toimii kunkin kohdalla seuraavasti. Olkoon i niiden desimaalien määrä, jotka pääohjelma on jo tulostanut desimaalipilkun jälkeen. Pääohjelma testaa, esittääkö merkkijono ohjelmaa. Jos vastaus on kyllä, pääohjelma lisää siihen mekanismin, joka laskee, kuinka monta merkkiä se on tulostanut ja pysäyttää sen, kun $i+1$ on tullut täyteen. Näin saatu ohjelma joko tulostaa enintään $i+1$ merkkiä ja pysähtyy, tai jää ikuiseen silmukkaan tulostaen kaiken kaikkiaan enintään i merkkiä. Täydennä päättely.

(f) Luonnollisten lukujen ominaisuuksia voidaan ilmaista yksipaikkaisilla predikaateilla $P(n)$. Esimerkiksi ”n on parillinen” voidaan ilmaista $\exists i : n = 2i$. Luonnollisten lukujen suhteita voidaan ilmaista kaksipaikkaisilla predikaateilla $Q(n,m)$. Esimerkiksi ”n on suurempi kuin m” voidaan ilmaista ilman muita vertailuoperaattoreita kuin ”=” predikaatilla $\exists i : n = m + i + 1$. Jos on annettu kaksipaikkainen predikaatti $Q(n,m)$ ja luonnollinen luku 3, niin niistä saadaan yksipaikkainen predikaatti $P(n)$ sijoittamalla m:n paikalle Q:ssa 3, siis $P(n) \Leftrightarrow Q(n,3)$. Tämä toimii tietysti myös silloin kun kolmosen paikalla on mikä tahansa muu luonnollinen luku. Voidaanko

$Q(n,m)$ valita siten, että jokainen yksipaikkainen predikaatti saadaan tällä tavalla valitsemalla m:n arvo sopivasti? Vihje: tarkastele predikaattia $R(n) \Leftrightarrow \neg Q(n,n)$.

5 Esseeaiheita

Tentissä tulee olemaan kysymys, jossa sinua pyydetään kirjoittamaan ennalta valmistelemasi essee. Esseen tavoitteena on osoittaa, että olet ymmärtänyt kurssin aihepiiriin kuuluvan tai aihepiiriä lähellä olevan jonkin asian hyvin. Essee kannattaa valmistella etukäteen. Tenttiin se tuodaan omassa päässä. Sen painoarvo on suunnilleen 10 % . . . 15 % tentin pisteistä, esimerkiksi 4 pistettä 30:stä.

Esseen ihannepituus on puolen ja yhden A4-sivun välillä. Jotta esseesi ei paisuisi tätä paljon pitemmäksi, keskity olennaiseen ja jätä toisarvoiset asiat pois. Tarkoituksena ei siis ole kirjoittaa valitusta aiheesta mahdollisimman paljon, vaan jokin yksittäinen asia mahdollisimman hyvin.

Pisteytyksessä kiinnitetään paljon huomiota siihen, että

- esitystapa on täsmällinen,
- väitetyt asiat pitävät paikkansa ja
- niistä muodostuu johdonmukainen kokonaisuus.

Matemaattisten symboleiden käyttö ei periaatteessa ole välttämätöntä, mutta käytännössä se on usein paras keino ilmaista asiat täsmällisesti ja ytimekkäästi. Esimerkkien käyttö on usein hyväksi. Jätä lähdeluettelo pois. Ei ole järkevää vaatia, että opettelisit lähteet ulkoa.

Käsialan tulee olla niin selvä, että lukijan ei tarvitse arvailla mitä sanaa tai symbolia kulloinkin on tarkoitettu. Jos esimerkiksi i ja j ovat yhtäaikaa käytössä matemaattisina symboleina, j:n koukku kannattaa kirjoittaa aina selvästi ja varoa piirtämästä i:lle koukkua.

Alla on ehdotettu joitakin aiheita, mutta voit valita myös oman aiheen. Halutessasi voit hyväksyttää oman aiheesi etukäteen kurssin opettajalla. Voit myös muokata alla olevia aiheita. Useimmissa tapauksissa englanninkielinen Wikipedia on hyvä lähtökohta tietojen etsimiselle, eikä ole pakkoa käyttää muita lähteitä jos asia tuli selväksi Wikipedian perusteella. Tarkoitus on, että ymmärrät asian ja selostat sen omin sanoin. Jos tekstisi näyttää suoralta käännökseltä Wikipediasta tai muusta lähteestä, niin pistemäärä vähenee huomattavasti.

Säännöllisiin kieliin liittyviä

- Mooren kone (Moore machine)
- Mealyn kone (Mealy machine)
- Mooren ja Mealyn koneiden vertailu
- kaksisuuntainen äärellinen automaatti (two-way finite automaton)
- Linuxin `grep`
- `Lex` (eräs leksikaalianalysaattorin automaattinen muodostaja)

Yhteysriippumattomiin kieliin liittyviä

- Chomskyn normaalimuoto (Chomsky normal form)
- Greibachin normaalimuoto (Greibach normal form)
- Cocke–Younger–Kasami-algoritmi eli CYK-algoritmi
- Earleyn jäsennin (Earley parser)
- $LR(1)$-jäsentäminen
- atribuuttikielioppi (attribute grammar)
- `yacc` (eräs jäsentimien automaattinen muodostaja)
- `bison` (eräs jäsentimien automaattinen muodostaja)

Laskettavuuteen liittyviä

- Ricen lause
- Post correspondence problem
- Ackermannin funktio
- ahkera majava (busy beaver)

Muita

- Kolmogorov-kompleksisuus