Opgaven 1

Verwijzingen in deze opgaven betreffen het boek van Peter Linz.

1. Toon de volgende eigenschappen uit de verzamelingenleer aan: *Exercises 1, 3, 5, 6, 7, 10 blz. 12-13* (neem aan dat er een universele verzameling U is gegeven)

   (a) $|2^S| = 2^{|S|}$ voor eindige S.

   (b) $|S_1 \times S_2| = |S_1||S_2|$ voor eindige S_1, S_2

   (c) $\overline{S_1 \cup S_2} = \overline{S_1} \cap \overline{S_2}$

   (d) $\overline{S_1 \cap S_2} = \overline{S_1} \cup \overline{S_2}$

   (e)
   $$\overline{\bigcup_{i=0,\ldots,n} S_i} = \bigcap_{i=0,\ldots,n} \overline{S_i}$$
   $$\overline{\bigcap_{i=0,\ldots,n} S_i} = \bigcup_{i=0,\ldots,n} \overline{S_i}$$

   (f) $S_1 \cup S_2 = \overline{\overline{S_1} \cap \overline{S_2}}$

   (g) $S_1 \times (S_2 \cup S_3) = (S_1 \times S_2) \cup (S_1 \times S_3)$

2. Zij (V, E) een willekeurige graaf waarvoor er een wandeling bestaat tussen v_i en v_j ($v_i, v_j \in V$). Bewijs dat er voor deze graaf een pad van lengte ten hoogste $|V| - 1$ tussen deze twee punten bestaat. *Exercise 21, blz. 14*

3. Bewijs voor ieder zin u en iedere $n \in \mathbb{N}$ dat $|u^n| = n|u|$. *Exercise 1 blz 26*

4. We definiëren de omgekeerde van een zin recursief:
   $$\lambda^R = \lambda$$
   Als w^R gedefiniëerd is voor zekere $w \in \Sigma^*$ en $a \in \Sigma$, dan:
   $$(wa)^R = a(w^R).$$
   Bewijs voor $u, v \in \Sigma^*$ dat $(uv)^R = v^R u^R$. *Ex 2, blz 26*

5. Bewijs: $\forall w \in \Sigma^*.(w^R)^R = R$. *Ex 3, blz 26*

6. Bestaan er talen waarvoor $\overline{L}^* = \overline{L^*}$? *Ex 6, blz 26*

7. Vind grammatica's voor $\Sigma = \{a, b\}$ die de volgende talen genereren:

   (a) alle zinnen met precies één a

   (b) alle zinnen met op zijn minst één a

   (c) alle zinnen met ten hoogste drie a's

   (d) alle zinnen met op zijn minst drie a's

   Ex. 10, blz 27

8. Zij $|w|_a$ het aantal keren dat a in w voorkomt (voor willekeurige $w \in \Sigma^*, a \in \Sigma$). Vind grammatica's voor $\Sigma = \{a, b\}$ die de volgende talen genereren:

   (a) $L = \{w \in \Sigma^* : |w|_a = |w|_b\}$

   (b) $L = \{w \in \Sigma^* : |w|_a = |w|_b + 1\}$

9. Welke taal wordt door de grammatica met de volgende productieregels gegenereerd?

$$\begin{array}{ccc} S & \rightarrow & Aa \\ A & \rightarrow & bS \\ S & \rightarrow & \lambda \end{array}$$

   Ex. 11, blz 27

10. Welke taal wordt door de grammatica met de volgende productieregels gegenereerd?

$$\begin{array}{ccc} S & \rightarrow & Aa \\ A & \rightarrow & B \\ B & \rightarrow & Aa \end{array}$$

   Ex. 12, blz 27

11. Voor $\Sigma = \{a, b\}$, construeer edh's die de volgende verzamelingen aanvaarden:

   (a) alle zinnen met precies één a

   (b) alle zinnen met op zijn minst één a

   (c) alle zinnen met op zijn hoogst drie a's

   (d) alle zinnen met op zijn minst één a en precies twee b's

   (e) alle zinnen met precies twee a's en meer dan twee b's

   Ex. 2 blz 45

12. Laat zien dat de taal $L = \{a^n : n \geq 0, n \neq 4\}$ regulier is. *Ex 13, blz 46*

Opgaven 2

1. Bewijs: $\delta^*(q, uv) = \delta^*(\delta^*(q, u), v)$.
2. Maak een equivalente edh van de ndh uit *Ex. 3, blz 61.*
3. Bewijs dat alle eindige talen regulier zijn. *Ex. 11, blz 61*
4. Minimalizeer het aantal toestanden in de edh uit *Ex 1, blz 68*
5. Geef een reguliere uitdrukking voor de verzameling $\{a^n b^m : (m+n) \ \mathit{even}\}$ *Ex. 4, blz 76.*
6. Geef reguliere uitdrukkingen voor de volgende talen: *Ex 5, blz 76*
   (a) $L_1 = \{a^n b^m : n \geq 4, m \leq 3\}$
   (b) $L_2 = \{a^n b^m : n < 4, m \leq 3\}$
   (c) $\overline{L_1}$
   (d) $\overline{L_2}$
7. Geef een reguliere uitdrukking voor: *Ex. 13, blz 76*
   $$L = \{w \in \{0,1\}^* : w \text{ heeft precies een paar opeenvolgende nullen}\}$$
8. Geef een ndh die de taal $L(ab^*aa + bba^*ab)$ herkent. *Ex 1, blz 86*
9. Geef een ndh die het complement van de taal uit de vorige opgave herkent.
10. Beschouw de veralgemeniseerde transitiegraaf uit *Ex. 8, blz 87*
    Zoek een equivalente veralgemeniseerde transitiegraaf die enkel twee toestanden bevat en bepaal de taal die wordt herkend door deze graaf.

Opgaven 3

1. Bepaal een edh die de taal herkent die voortgebracht wordt door de volgende grammatica:

$$\begin{array}{rcl} S & \rightarrow & abA \\ A & \rightarrow & baB \\ B & \rightarrow & aA|bb \end{array}$$

   Ex 1, blz 97

2. Construeer een links-lineaire grammatica voor de taal uit de vorige oefening. *Ex 3, blz 97*

3. Zoek een reguliere grammatica die de taal $L(aa^*(ab+a)^*)$ genereert. *Ex 2, blz97*

4. Construeer een ndh die de taal $L(ab^*a^*) \cap L(a^*b^*a)$ herkent. Gebruik de constructie van doorsnedes van talen. *Ex 2b, blz 109*

5. Definiëer het symmetrische verschil van twee verzamelingen S_1, S_2:

$$S_1 \ominus S_2 = \{x : x \in S_1 \text{ of } x \in S_2 \text{ maar niet } x \in S_1 \text{ en } x \in S_2\}$$

   Bewijs dat de collectie reguliere talen gesloten is onder symmetrische verschillen. *Ex 6, blz 109*

6. We noemen een taal L palindroom als $L = L^R$. Beschrijf een algoritme om na te gaan of een gegeven reguliere taal palindroom is. *Ex 5, blz 113*

7. Beschrijf een algortime dat na gaat of een gegeven reguliere taal woorden w bevat waarvoor $w \in L^R$. *Ex 6, blz 113*

Opgaven 4

1. * Bewijs de volgende versie van het pompend lemma:
   Zij L een oneindige reguliere taal, dan bestaat er een m zodanig dat iedere $w \in L$ met $|w| \geq m$ geschreven can worden als:

   $$w = xyz$$
   met
   $$|yz| \leq m$$
   en
   $$|y| \geq 1$$

   zó dat
   $$w_i = xy^iz \in L$$
   voor alle $i \in \mathbb{N}$. *Ex 1, blz 121*

2. Laat zien dat $L = \{w : |w|_a = |w|_b\}$ niet regulier is. *Ex 3, blz 123*

3. Bewijs dat de volgende talen niet regulier zijn: *Ex 4,5*

   (a) $L = \{a^n b^l : n \leq l\}$
   (b) $L = \{a^n b^l a^k : n = l \text{ of } l \neq k\}$
   (c) $L = \{w : |w|_a \neq |w|_b\}$
   (d) $L = \{ww : w \in \{a,b\}^*\}$
   (e) $L = \{www w^R : w \in \{a,b\}^*\}$
   (f) $L = \{a^p : p \text{ priem }\}$
   (g) $L = \{a^p : p \text{ niet priem }\}$
   (h) * $L = \{a^{(n^2)} : n \in \mathbb{N}\}$
   (i) * $L = \{a^{(2^n)} : n \in \mathbb{N}\}$

4. Bepaal of de volgende talen regulier zijn en bewijs uw bewering.

   (a) * $L = \{a^{F_n} : n \in \mathbb{N}\}$, waarbij F_n het n-de Fibonacci-getal is.
   (b) $L = \{a^n b^l : n + l \text{ is priem }\}$
   (c) $L = \{a^n : n \text{ priem of het product van twee of meer priemgetallen }\}$
   (d) * $L = \{a^n b^l : n \leq l \leq 2n\}$

Opgaven 5

1. Geef een contextvrije grammatica voor de verzameling van alle reguliere uitdrukkingen op willekeurig (eindig) alfabet Σ. *gemodificeerde Ex 22, blz 135*

2. Geef een s-grammatica voor $L(aaa^*b + b)$. *Ex 1, blz 145*

3. $\star$ Laat zien dat iedere s-grammatica ondubbelzinnig is. *Ex 4*

4. Verwijder alle nutteloze producties van de grammatica:

$$\begin{array}{lcl} S & \rightarrow & aS|AB \\ A & \rightarrow & bA \\ B & \rightarrow & AA \end{array}$$

   Welke taal genereert deze grammatica? *Ex 5, blz 162*

5. $\star$ Verwijder alle λ, één en nutteloze producties van:

$$\begin{array}{lcl} S & \rightarrow & a|aA|B|C \\ A & \rightarrow & aB|\lambda \\ B & \rightarrow & Aa \\ C & \rightarrow & cCD \\ D & \rightarrow & ddd \end{array}$$

   Ex 6

6. Verwijder alle λ-producties uit:

$$\begin{array}{lcl} S & \rightarrow & AaB|aaB \\ A & \rightarrow & \lambda \\ B & \rightarrow & bbA|\lambda \end{array}$$

   Ex 7

7. $\star$ Is de volgende grammatica in Chomsky normaalvorm te brengen? Indien dat het geval is geef deze.

$$\begin{array}{lcl} S & \rightarrow & abAB \\ A & \rightarrow & bAB|\lambda \\ B & \rightarrow & BAa|A|\lambda \end{array}$$

8. Zet de volgende grammatica om in Greibach normaal vorm:

$$S \to ab|aS|aaS$$

Ex 12, blz 171

9. $\star$ Zet de volgende grammatica om in Greibach normaal vorm:

$$\begin{array}{lcl} S & \to & ABb|a \\ A & \to & aaA|B \\ B & \to & bAb \end{array}$$

Ex 13

10. $\star$ Kan iedere lineaire grammatica omgezet worden in een equivalente grammatica waarin alle producties van de vrom $A \to ax$ zijn, waarbij $a \in T$ en $x \in V \cup \{\lambda\}$? *14*

11. Construeer npda's voor de volgende talen: *Ex 3,4, blz 182*

    (a) $L = L(aaa^*b)$

    (b) $L = \{a^n b^{2n} : n \in \mathbb{N}\}$

    (c) $\star$ $L = \{wcw^R : w \in \{a,b\}^*\}$

    (d) $\star$ $L = \{w_1cw_2 : w_1, w_2 \in \{a,b\}^*, w_1 \neq w_2^R\}$

    (e) $\star$ $l = \{ab(ab)^n b(ba)n : n \in \mathbb{N}\}$

Opgaven 6

1 Opgaven JFLAP

Bij deze cursus maken we gebruik van JFLAP. Dit programma is verkrijgbaar via minerva, bij de documenten, of via de internet pagina www.jflap.org . Om JFLAP te kunnen starten is het nodig dat java 1.4 of hoger is geïnstalleerd op de computer.
Start JFLAP op door onder windows te dubbelklikken op het bestand JFLAP.jar, onder linux is het op te starten met het commando "java -jar JFLAP.jar".

1. De taal $L = \Sigma^*$ is evident regulier. Implementeer in JFLAP de edh voor deze taal. Doe het zelfde voor $\overline{L}$.
2. We hebben al bewezen dat de taal $L = \{a^n : n \neq 4\}$ regulier is. Implementeer in JFLAP de bijbehorende edh en test deze voor enkele inputs. Gebruikt hiervoor input $->$ Multiple Run.
3. Beschouw de volgende vijf talen met alfabet $\Sigma = \{a, b\}$:
   (a) Woorden die geen deelwoord *aa* bevatten.
   (b) Woorden die geen deelwoorden *aa* of *aaa* bevatten.
   (c) Woorden die geen deelwoorden *aa* of *aba* bevatten.
   (d) Woorden die geen deelwoorden *aa* of *abaa* bevatten.
   (e) Woorden die geen deelwoorden *aaa* of *aab* bevatten.

   Zijn deze talen regulier? Welke van deze talen zijn aan elkaar gelijk? Gebruik JFLAP om je vermoeden te testen. Gebruik test $->$ compare equivalence.

Optioneel: Implementeer en test met JFLAP enkele edh's uit project1.

2 $\star$-opgaven

1. Bewijs dat $L = \{a^n b^j : n \geq (j-1)^3\}$ niet context-vrij is.
2. Bewijs dat $L = \{a^n b^j c^k : k = jn\}$ niet context-vrij is.

3. Bewijs dat $L = \{a^n b^j c^k : k > n, k > j\}$ niet context-vrij is.
4. Bewijs dat $L = \{w \in \{a, b, c\}^* : |w|_a = |w|_b = 2|w|_c\}$ niet context-vrij is.
5. Bewijs dat $L = \{a^n b^n a^m b^m : n, m \geq 0\}$ context-vrij is, maar niet lineair is.
6. Bewijs dat $L = \{a^{n^2} : n \in \mathbb{N}\}$ niet context-vrij is.
7. Bewijs dat $L = \{w \in \{a, b, c\}^* : |w|_a + |w|_b = |w|_c\}$ context-vrij maar niet lineair is.

Opgaven 7

1. Zij $W \subset \Sigma^*$ (met $\Sigma = \{a, b\}$) eindig. Is $L = \{w = a^n b^n : n \in \mathbb{N}, w \notin W\}$ context-vrij? Bewijs uw vermoeden.
2. $\star$ Bewijs dat de collectie van contex-vrije talen gesloten is onder homomorfismen. *Ex. 4, blz. 220*
3. Is de collectie van niet-context-vrije talen gesloten onder homomorfismen? Bewijs uw bewering.
4. Bewijs dat $L = \{a^n b^n c^n d^n e^n f^n : n \in \mathbb{N}\}$ niet context-vrij is.
5. Bewijs dat de collectie van context-vrije talen gesloten is onder omkering. *Ex 5*
6. $\star$ Bewijs dat $L = \{a^n b^n : n \in \mathbb{N}, n \text{ is geen veelvoud van } 5\}$ context-vrij is.
7. Omschrijf voor gegeven $n \in \mathbb{N}$ een algoritme om te bepalen of een context-vrije taal woorden bevat van lengte kleiner dan n.
8. $\star$ (met zijn tweeen) Bewijs dat iedere $LL(k)$ grammatica ondubbelzinnig is.
9. $\star$ (met zijn tweeen) Bewijs: als G een $LL(k)$ grammatica is dan is $L(G)$ een deterministische context-vrije taal.
10. Beschrijf een algoritme dat voor gegeven k vast stelt of een grammatica in Greibach normaalvorm $LL(k)$ is.
11. $\star$ Bewijs dat iedere s-grammatica ondubbelzinnig is. *opgave 3, reeks 5*
12. $\star$ Bewijs dat $L = \{w \in \{a, b, c\}^* : |w|_a + |w|_b = |w|_c\}$ context-vrij, maar niet lineair is.

Opgaven 8

1. Is de volgende grammmatica $LL(1)$?

$$\begin{array}{ll} S \to ABC, & D \to dd, \\ A \to BCDE, & D \to \lambda, \\ B \to CDEF, & E \to e, \\ C \to cc, & F \to \lambda, \\ C \to \lambda, & F \to faeb. \end{array}$$

   Bepaal voor het antwoord de EERSTE-, VOLGENDE- en LOOKAHEAD-verzamelingen.

2. Construeer Turing machines voor:
   (a) $L = L(aba^*b)$
   (b) $L = \{w : |w| \text{ is even }\}$
   (c) $L = \{a^n b^{2n} : n \geq 1\}$
   (d) het berekenen van $x + y$ (voor $x, y \in \mathbb{N}$)voor alle even x, maar de machine stopt niet voor oneven x.

3. $\star$ Construeer een Turing machine die $L = \{w \in \{a, b\}^+ : n_a(w) < n_b(w)\}$ aanvaardt.

4. Stel L is regulier en bevat geen lege woorden, construeer een Turing machine die L aanvaardt.

5. $\star$ Construeer een Turing machine die de functie $f : w \mapsto w^R$ implementeert voor $w \in \Sigma^*$ met Σ eindig.

6. $\star$ Zij n gegeven. Construeer een Turing machine die de functie $f : \mathbb{N} \to \mathbb{N} : x \mapsto x \bmod n$ implementeert.

7. Construeer een Turing machine die de volgende functie implementeert: $f : \mathbb{N}^2 \to \mathbb{N} : (x, y) \mapsto x - y$ als $x \geq y$, $(x, y) \mapsto 0$ als $x < y$.

8. $\star$ (met zijn tweeen) Construeer een Turing machine die $L = \{w \in \{a, b, c\}^+ : n_a(w) = n_b(w) + n_c(w)\}$ aanvaardt. Zij $m > 3$ gegeven. Construeer een machine die $L = \{w \in \{a_1, \ldots, a_m\}^+ : n_{a_1} = n_{a_2} + \cdots + n_{a_m}\}$ aanvaardt.

9. $\star$ (met zijn tweeen) Construeer een Turing machine die $f : \mathbb{N}^2 \to \mathbb{N} : (x, y) \mapsto xy$ implementeert.

Opgaven 9

1 JFLAP opgaven

In deze opgaven gebruiken we alfabet $\Sigma = \{a, b\}$ of $\Sigma = \{1\}$.

1. Implementeer een Turing machine in JFLAP die $\{a, b\}^* - \{\lambda\}$ accepteert.
2. Implementeer (in JFLAP) de machine $M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, q_0, \Box, \{q_3\})$ met:
$$\begin{aligned} \delta(q_0, a) &= (q_1, a, R) \\ \delta(q_0, b) &= (q_2, b, R) \\ \delta(q_1, b) &= (q_1, b, R) \\ \delta(q_1, \Box) &= (q_3, \Box, R) \\ \delta(q_2, b) &= (q_2, b, R) \\ \delta(q_2, a) &= (q_3, a, R) \end{aligned}$$
Welke taal aanvaardt deze machine?
3. Implementeer een machine (in JFLAP) die enkel woorden van even lengte accepteert.
4. Idem voor woorden w met $n(w)_a = n(w)_b$.
5. Implementeer in JFLAP de functie $f : x \mapsto 1 + x$.
6. Idem voor $f : x \mapsto 2x$.
7. Idem voor de functie f met waardes $f : x \mapsto x/2$ als x even is, $f : x \mapsto (x + 1)/2$ als x oneven is.

2 $\star$-opgaven

1. JFLAP implementeert Turing machines niet geheel volgens onze definitie van een standaard Turing-machine. Zo is de input-alfabet gelijk aan de tape-alfabet en kan het dus voorkomen dat de machine woorden uit $(\Sigma \cup \Gamma)^*$ accepteert. Geef voor willekeurige Turing machine M een equivalent machine $\hat{M}$ die dit probleem in JFLAP niet heeft.

2. JFLAP heeft behalve L en R ook nog S (blijf op de plaats, Stationary) als mogelijke 'verplaatsing' van de lees-kop. Bewijs dat voor iedere Turing-machine met zulke verplaatsingen er een equivalente Turing-machine bestaat zonder zulke verplaatsingen.

3. Geef een Turing machine die enkel stopt als er een nul op de tape zit (ook als deze geïsoleerd staat, dwz omringd door $\square$).

4. (met zijn tweeen) Construeer een Turing machine die $L = \{w \in \{a, b, c\}^+ : n_a(w) = n_b(w) + n_c(w)\}$ aanvaardt. Zij $m > 3$ gegeven. Construeer een machine die $L = \{w \in \{a_1, \ldots, a_m\}^+ : n_{a_1} = n_{a_2} + \cdots + n_{a_m}\}$ aanvaardt.

5. (met zijn tweeen) Construeer een Turing machine die $f : \mathbb{N}^2 \to \mathbb{N} : (x, y) \mapsto xy$ implementeert.

6. Bewijs dat iedere s-grammatica ondubbelzinnig is.

7. (met zijn tweeen) Bewijs dat iedere $LL(k)$ grammatica ondubbelzinnig is.

8. (met zijn tweeen) Bewijs: als G een $LL(k)$ grammatica is dan is $L(G)$ een deterministische context-vrije taal.

Opgaven 10

1. Implementeer in JFLAP turing machines die de volgende basis primitief-recursieve functies berekenen:
   (a) $z(x)$
   (b) $s(x)$
   (c) p_1^1, p_1^2, p_2^3
2. $\star$ Zij i, n gegeven. Construeer een Turing-machine die de functie p_i^n berekent.
3. Gegeven twee Turing-berekenbare functies $f, g : \mathbb{N} \to \mathbb{N}$, omschrijf een Turing machine die de compositie berekent.
4. Gegeven Turing-berekenbare functies $f, g : \mathbb{N} \to \mathbb{N}$. Omschrijf een Turing machine die de volgende h berekent:
$$h(x, 0) = f(x)$$
$$h(x, s(y)) = g(h(x, y))$$
5. Construeer een turing machine die de functie $x \mapsto c$ voor zekere constante c berekent. Is deze primitief recursief? Bewijs je bewering.
6. De Ackermann functie was als volgt gedefiniëerd:
   $F_0(n) = n + 1$
   $F_{k+1}(n) = F_k^{(n)}(n)$
   $F(n) = F_n(n)$
   (a) Wat is F_1? En F_2, F_3, F_4?
   (b) Bewijs dat iedere F_k primitief recursief is.
   (c) Bewijs dat iedere F_k Turing berekenbaar is.
   (d) $\star$ Bewijs dat de functies F_k strikt stijgend zijn.
   (e) $\star$ Bewijs dat voor iedere n de functies $x \mapsto F_x(n)$ strikt stijgend zijn.
   (f) Bewijs voor de basis primitef-recursieve functies g dat er een k bestaat met $g(\vec{x}) \leq F_k(\max(\vec{x}))$.
   (g) Bewijs dit nu voor alle primitief-recursieve functies g.

7. Construeer een enumeratie procedure voor $L = \{a^n b^n : n \geq 1\}$

8. Bestaat er een beslissingsalgoritme voor recursief opsombare talen die beslist of een r.o. taal woorden bevat van lengte kleiner of gelijk aan 6? Bewijs uw bewering.

Opgaven 11

1. Bewijs: $f(n) = O(a_m n^m + \cdots + a_0) \Rightarrow f(n) = O(n^m)$.
2. Bewijs $L_1, L_2 \in TIME(t(n)), t(n) \geq n \Rightarrow L_1 \cup L_2 \in TIME(t(n))$.
3. Is $L \in P$ beslisbaar? Bewijs uw bewering.
4. Bewijs: L is regulier $\Rightarrow L \in P$.
5. Bewijs: $\{1^n : n \geq 1\} \in P$.
6. Omschrijf een verifyer voor $L = \{n : n$ is de som van twee kwadraten $\}$. Is $L \in NP$?
7. Is $\{< a_1, \ldots, a_n >$: de verzameling a_i-tjes is op te splitsen in twee disjuncte verzamelingen welker som der elementen aan elkaar gelijk zijn $\} \in NP$?

Opmerkingen en uitwerkingen Opgaven 1

1. (a) Inductie naar $|S|$: Het geval $|S| = 0$ is triviaal. Neem $|S| = n+1$ en willekeurige, doch vaste, $x \in S$. Dan:

$$2^S = 2^{(S-\{x\})} \cup \{X \cup \{x\} : X \in 2^{(S-\{x\})}\}.$$

De twee verzamelingen aan de rechterkant zijn disjunct, dus:

$$|2^S| = |2^{(S-\{x\})}| + |\{X \cup \{x\} : X \in 2^{(S-\{x\})}\}| =$$

$$|2^{(S-\{x\})}| + |2^{(S-\{x\})}| \overset{I.H.}{=} 2 \cdot 2^n = 2^{n+1}.$$

$\square$

(b) $|S_1 \times S_2| = |\{(x,y) : x \in S_1, y \in S_2\}| = |\bigcup_{y \in S_2} \{(x,y) : x \in S_1\}| =$ (disjunctie) $\sum_{y \in S_2} |\{(x,y) : x \in S_1\}| = \sum_{y \in S_2} |S_1| = |S_2| \cdot |S_1|$.

$\square$

(c) $x \in \overline{S_1 \cup S_2} \Leftrightarrow x \notin S_1 \cup S_2 \Leftrightarrow x \notin S_1$ en $x \notin S_2 \Leftrightarrow x \in \overline{S_1}$ en $x \in \overline{S_2} \Leftrightarrow x \in \overline{S_1} \cap \overline{S_2}$

$\square$

(d) Analoog aan de vorige opgave.

(e) Gebruik inductie en c.

(f) Gebruik inductie en d.

(g) Schrijf de definitie van de productverzameling uit.

2. Inductie naar $|V|$: Het geval $|V| = 1$ is triviaal, neem aan $|V| = n+1$. Zij $(v_i, x_0)(x_0, x_1) \ldots, x_m)(x_m, v_j)$ een wandeling waarin $x_l \neq v_j$ voor alle $l \leq m$. (deze bestaat, immers: indien er een l is met $x_l = v_j$ nemen we voor zulke l minimaal: $(v_i, x_0) \ldots (x_{l-1}, x_l)$ als nieuwe wandeling).
Dan ligt de wandeling $(v_i, x_0)(x_0, x_1) \ldots (x_{m-1}, x_m)$ geheel in de graaf $(V - \{v_j\}, E - \{(x, v_j), (v_j, x) : x \in V\})$. $|V - \{v_j\}| = n$, dus de I.H. levert een pad $(v_i, y_0) \ldots (y_r, x_m)$ van lengte $n-1$. Dus: $(v_i, y_0) \ldots (y_r, x_m)(x_m, v_j)$ is een pad van lengte n.

$\square$

3. Gebruik inductie naar n en dat $|vu| = |v| + |u|$.

4. Gebruik inductie naar $|v|$ en de definitie.
5. Gebruik inductie naar $|w|$.
6. Ja, neem bijvoorbeeld $L = \emptyset$.
7. (a)

$$\begin{aligned} S \to &\quad BaB \\ B \to &\quad bB|\lambda \end{aligned}$$

   (b)

$$\begin{aligned} S \to &\quad BaB \\ B \to &\quad bB|aB|\lambda \end{aligned}$$

   (c)

$$\begin{aligned} S \to &\quad BABABAB \\ A \to &\quad a|\lambda \\ B \to &\quad bB|\lambda \end{aligned}$$

   (d)

$$\begin{aligned} S \to &\quad BaBaBaB \\ B \to &\quad aB|bB|\lambda \end{aligned}$$

8. Boek, pagina 23. Voor b: Neem de oplossing van onderdeel a, maar vervang daar het startsymbool voor variabele S_1, voeg de volgende productieregels toe:

$$\begin{aligned} S \to &\quad aS_1 \\ S \to &\quad S_1S \end{aligned}$$

9. $L = \{b^n a^n : n > 0\}$
10. $L = \emptyset$

Opmerkingen en uitwerkingen Opgaven 2

1. We gebruiken inductie naar $|v|$:
   - $|v|$=0: $\delta^*(q, u\lambda) = \delta^*(q, u) = \delta^*(\delta^*(q, u), \lambda)$
   - $|v|$=n+1, dwz. $v = wa$ met $|w| = n, a \in \Sigma^*$.
     $\delta^*(q, uwa) \overset{def}{=} \delta(\delta^*(q, uw), a) \overset{I.H.}{=} \delta(\delta^*(\delta^*(q, u), w), a) \overset{def}{=} \delta^*(\delta^*(q, u), wa)$
2. Gebruik procedure van bladzijde 58.
3. $\sum_{w \in L} w$ is een reguliere uitdrukking als L eindig is of construeer mbv λ-transities een ndh die alle $w \in L$ herkent.
4. Gebruik procedure van bladzijde 65.
5. $n + m$ even betekent dat n en m beiden even of beiden oneven zijn, dus:
   $$(aa)^*(bb)^* + a(aa)^*b(bb)^*$$
6. (a)
   $$aaaaa^* + aaaaa^*b + aaaaa^*bb + aaaaa^*bbb$$
   (b)
   $$\sum_{n<4, m\leq 3} a^n b^m$$
   (c) Werk met $\overline{L_1} = \{a^n b^m : n < 4, m > 3\} \cup L((a + b)^*ba(a + b)^*)$
   (d) Werk met $\overline{L_2} = \{a^n b^m : n \geq 4, m > 3\} \cup L((a + b)^*ba(a + b)^*)$
7. $$(1 + 01)^*00 + (1 + 01)^*001(1 + 01)^*$$
8.
9. Zet ndh uit de vorige opgave om in een edh, gebruik de procedure om de edh van het complement van een taal te bepalen.

Opmerkingen en uitwerkingen Opgaven 3

1. Gebruik de procedure van pagina's 91-92.
2. Maak eerst een ndh van de omgekeerde van de taal, zoek de bijbehorende rechts-lineaire grammatica, gebruik dan de procedure op pagina 96:

$$\begin{array}{ccc} S & \rightarrow & V_3b \\ V_3 & \rightarrow & Bb \\ B & \rightarrow & V_2a \\ V_2 & \rightarrow & Ab \\ A & \rightarrow & Ba|V_1b \\ V_1 & \rightarrow & a \end{array}$$

3. Construeer ndh, zoek bijbehorende links dan wel rechts lineaire grammatica:

$$\begin{array}{ccc} S & \rightarrow & aq_1|a \\ q_1 & \rightarrow & aq_1|a_q2|a \\ q_2 & \rightarrow & bq_1 \end{array}$$

4.
5. $S_1 \ominus S_2 = S_1 \cup S_2 - (S_1 \cap S_2)$
6. Zij M een edh met $L(M) = L$, construeer edh M^R waarvoor $L(M^R) = L^R$. Gebruik theorem 4.7 bladzijde 112 om te bepalen of $L = L^R$.
7. Construeer een edh voor $L \cap L^R$. Gebruikt theorem 4.6 om te bepalen of $L \cap L^R$ niet-leeg is.

Opmerkingen en uitwerkingen Opgaven 4

1.

2. We gebruiken het pompend lemma:
Zij m gegeven. Neem $w = a^m b^m \in L$ en zij x, y, z zodanig dat:

$$|xy| \leq m$$
$$|y| \geq 1$$

Dan $x = a^{|x|}, y = a^{|y|}$ en $z = a^{m-|x|-|y|}b^m$. $xy^0z = a^{m-|y|}b^m \overset{|y|\geq 1}{\notin} L$, we hebben dus een tegenvoorbeeld van de conclusie van het pompend lemma. L is oneindig, dus L is niet regulier.

$\square$

3. (a) Gebruik weer het pompend lemma met voor gegeven m de w uit opgave 2, en bijvoorbeeld $i = 2$.

   (b) Pompend lemma is weer direct mogelijk, maar ook geruik van de afsluitingseigenschappen van de collectie van reguliere talen is mogelijk: Stel voor een tegenspraak dat L regulier is, dan is ook $L \cap L(a^*b^*) = \{a^n b^n : n \in \mathbb{N}\}$ regulier (doorsnedes van reguliere talen zijn regulier). Van deze taal weten we dat deze niet regulier is, dus was L ook niet regulier.

   $\square$

   (c) Gebruik dat het complement van een reguliere taal weer regulier is en opgave 2.

   (d) Gebruik het pompend lemma met voor gegeven m: $w = a^m b$.

   (e) Gebruik het pompend lemma met w als hierboven.

   (f) We geven weer een tegenvoorbeeld van het pompend lemma:
   Zij m gegeven, neem $w = a^{p_m}$, waarbij p_m het m-de priemgetal is en zij x, y, z met $w = xyz$ zodanig dat $|xy| \leq m, |y| \geq 1$. Dan $xy^{p_m+1}z = a^{p_m+|y|p_m} \notin L$, wat een tegenvoorbeeld van de conclusie van het pompend lemma is, dus L is niet regulier.

   $\square$

(g)

(h)

4. (a)

   (b) Gebruik dat de reguliere talen afgesloten zijn onder homomorfismen en het homomorfisme met $f(a) = a, f(b) = a$.

   (c) $L = \Sigma^* - \{\lambda, a\}$

   (d)

Opmerkingen en uitwerkingen Opgaven 5

1. Om verwarring te voorkomen gebruiken we $\bar{\lambda}$ en $\bar{\emptyset}$. $\Sigma = \{a_1, \ldots, a_n\}$

$$\begin{array}{ll} S \to & S + S|S \cdot S|S^*|(S) \\ S \to & \bar{\emptyset}|\bar{\lambda} \\ S \to & a_1|\ldots|a_n \end{array}$$

   Deze grammatica heeft als alfabet $\Sigma \cup \{\bar{\emptyset}, \bar{\lambda}, +, \cdot, *, (,)\}$.

2.
$$\begin{array}{ll} S \to & aA|b \\ A \to & aB \\ B \to & aB|b \end{array}$$

3. Zij G een s-grammatica, $w \in L(w)$ en stel we hebben twee meest-linkse afleidingen:

$$\begin{array}{l} S \Rightarrow \cdots \Rightarrow w \\ S \Rightarrow \cdots \Rightarrow w \end{array}$$

   We bewijzen met inductie naar de lengte van de afleiding dat deze gelijk zijn.

   - $S \Rightarrow x$, G is een s-grammatica, dus er bestaat maar één productieregel $S \to aA$ waarbij $w = aw'$. Dus moet in beide afleidingen de eerste productie $S \Rightarrow aA$ zijn.
   - Stel $S \Rightarrow x_1 \Rightarrow \cdots \Rightarrow x_n$ is het beginstuk in de twee afleidingen. Dan (want G is een s-grammatica) $x_n = w$ of $x_n = a_1 \ldots a_n B$ met $w = a_1 \ldots a_n w'$. Neem het laatste aan. (in het eerste geval zijn we klaar)
     G is een s-grammatica, er is dus maar één productie regel $B \to bC$ waarbij $w' = bw''$. De volgende stap in beide afleidingen moet dus zijn $x_n \Rightarrow a_1 \ldots a_n bC$.

$\square$

4. We houden de lege grammatica over. De grammatica genereert de lege taal.

5.

$$\begin{aligned} S \rightarrow &\quad a|aA \\ A \rightarrow &\quad aB \\ B \rightarrow &\quad aA|a \end{aligned}$$

6.

$$\begin{aligned} S \rightarrow &\quad aB|aaB|aa \\ B \rightarrow &\quad bb \end{aligned}$$

7.

$$\begin{aligned} S \rightarrow &\quad V_aV_{bAB}|V_aV_{bB}|V_aV_{bA}|V_aV_b \\ A \rightarrow &\quad V_bV_{AB}|V_bB|V_bV_a|b \\ B \rightarrow &\quad BV_{Aa}|BV_a|AV_a|a \\ V_a \rightarrow &\quad a \\ V_b \rightarrow &\quad b \\ V_{bAB} \rightarrow &\quad V_bV_{AB} \\ V_{bB} \rightarrow &\quad V_bB \\ V_{bA} \rightarrow &\quad V_bA \\ V_{AB} \rightarrow &\quad AB \\ V_{Aa} \rightarrow &\quad AV_a \end{aligned}$$

8.

$$\begin{aligned} S \rightarrow &\quad aB|aS|aC \\ B \rightarrow &\quad b \\ C \rightarrow &\quad aS \end{aligned}$$

9.

$$\begin{aligned} S \rightarrow &\quad aV_aABv_b|bAv_bBV_b|a \\ A \rightarrow &\quad aV_aA|bAV_b \\ B \rightarrow &\quad bAV_b \\ V_a \rightarrow &\quad a \\ V_b \rightarrow &\quad b \end{aligned}$$

10. Nee, neem bijvoorbeeld de grammatica met producties:

$$S \rightarrow aAb, A \rightarrow aAb|\lambda$$

Opmerkingen en uitwerkingen Opgaven 5

1. Neem aan (voor een tegenspraak) dat L context vrij is. Dan bestaat er volgens het pompend lemma voor context-vrije talen een m waarvoor voor $w = a^{(m-1)^3} b^m$ er u, v, x, y, z bestaan met
$$w = uvxyz,$$
$$|vxy| \leq m, |vy| \geq 1$$
en
$$w_i = uv^i xy^i z \in L$$
voor alle $i \in \mathbb{N}$.
We ondescheiden de verschillende mogelijkheden:
1 - vxy 'zit geheel in $a^{(m-1)^3}$', dus $vxy = a^{|vxy|}$. Dan
$$w_0 = a^{(m-1)^3 - |xy|} b^m \notin L.$$
2 - vxy 'zit geheel in b^m', dus $vxy = b^{|vxy|}$. Dan
$$w_2 = a^{(m-1)^3} b^{m+|xy|} \notin L.$$
3- v 'zit geheel in $a^{(m-1)^3}$' en y 'zit geheel in b^m', waarbij $|v|, |y| \geq 1$ (anders hebben we weer geval 2 of 1). Dan
$$w_i = a^{(m-1)^3 + i|v|} b^{m+i|y|}.$$
neem $i = max\{|v|, |y|\} + 1$, dan
$$(m + i|y|)^3 > m^3 + i^4 > (m-1)^3 + i|v|,$$
ofwel $w_i \notin L$.
De andere gevallen zijn triviaal, dus bestaat er een i waarvoor $w_i \notin L$ wat een tegenspraak is, dus L is niet context-vrij.

2. Neem $w = a^{m^2} b^b c^m$ en ga als hierboven de gevallen na.

3. Idem, met $w = a^m b^m c^{m+1}$.

4. Idem, met $w = a^m b^m c^{2m}$.

5. Context vrije grammatica:

$$\begin{aligned} S \rightarrow &\quad AB \\ A \rightarrow &\quad aAb|\lambda \\ B \rightarrow &\quad aBb|\lambda \end{aligned}$$

Neem (voor een tegenspraak) aan dat L lineair is en zij m het pompend getal uit het pompend lemma voor lineaire talen. Kies $w = a^m b^m a^m b^m$. Zij $w = uvxyz$ met $|uvyz| \leq m$ en $|vy| \geq 1$. Dan $w_0 = a^{m-|v|} b^m a^m b^{m-|x|} \notin L$. Dus L is niet lineair.

6. Volg het bewijs voor het niet regulier zijn van L.

7. Construeer ndpda voor L. Gebruik voor het bewijs van niet lineair zijn $w = a^m b^m c^{2m}$.

Opmerkingen en uitwerkingen Opgaven 7

1. Ja, $L = \{a^n b^n, n\mathbb{N}\} \cap \bar{W}$, waarbij W en dus ook $\bar{W}$ regulier is. De doorsnede van een context-vrije met een reguliere taal is context-vrij, dus L is context vrij.
2. Zij L context-vrij met grammatica G en h een homomorfisme. Construeer $\hat{G}$ door de productieregels van G te veranderen: $A \to x_1 \dots x_n$ wordt $A \to y_1 \dots y_n$, waarbij $y_i = x_i$ als x_i een niet-terminaal is en $y_i = h(x_i)$ als x_i een terminaal is. Dan $h(L) = L(\hat{G})$ en $\hat{G}$ is context-vrij.
3. Neen, stel dat deze wel gesloten is onder homomorfismen, neem niet-context vrije taal $\{a^n b^n c^n\}$ met homomorfisme $h : c \mapsto b$, dan zou $h(L)$ niet context-vrij moeten zijn, terwijl deze dat wel is (bewijs zelf dat deze taal context-vrij is), dus is de collectie niet context-vrije talen niet gesloten onder homomorfismen.
4. Gebruik homomorfisme $h : d, e, f \mapsto c$.
5. Maak grammatica $\hat{G}$ met productieregels $A \to x_n \dots x_1$ voor iedere productieregel $A \to x_1 \dots x_n$ uit G. Bewijs met behulp van inductie naar de lengte van de afleiding dat $L(g)^R = L\hat{G})$.
6. $L = \{a^n b^n\} \cup \{w : |w|$ is geen veelvoud van 5 $\}$. Dus L is de doorsnede van een context-vrije en een reguliere taal, ofwel L is context-vrij.
7. Gebruik de ndpda va de taal en de edh die woorden van lengte kleiner of gelijk aan n herkent.
8. Analoog aan het bewijs voor simpele grammaticas.
9.
10.
11. Gebruik $w = a^m b^m c^{2m}$.

Opmerkingen en uitwerkingen Opgaven 8

1. Ja.
2.
3.
4. Zorg ervoor dat de eindtoestanden van de edh geen eindtoestanden in de Turing machine zijn, neem $\delta(q_i, a) = (q_j, a, R)$ waarbij $q_i \xrightarrow{a} q_j$ in de edh. Neem verder $\delta(q_i, \Box) = (q_f, \Box, R)$ waarbij q_f eindtoestand is.
5. Plak een hulpsymbool op het eind van het woord (bijv #). Zorg er voor dat de machine iteratief de eerste letter van het woord op het eind plaatst totdat # berijkt wordt, wis deze en zet de kop aan het begin van de uitkomst.
6. Tel de invoor af met behulp van n toestanden, Gebruik hekjes om 0 mod n te markeren. als het woord afgelopen is verwijder alles voor de laatste markering en plaats leeskop op het begin van de uitkomst.
7. Verwijder iteratief voor iedere 1 in y deze 1 en een 1 uit x indien die er nog zijn, anders verwijder enkel die in y.
8. Zelfde als voor $n_a(w) = n_b(w)$, waarbij de δ voor de a_i-tjes reageert alsof het b-tjes waren.
9. Tel voor iedere 1 in x y een keer op in de uitkomst.

Opmerkingen en uitwerkingen Opgaven 9

1. Construeer een herkenner voor Σ^*, met δ'. Zij δ de originele transitiefunctie en zorg ervoor dat de toestanden van de twee Turing machines disjunct zijn (waaarom?). Neem nu $\delta''(q_i, a) = \delta$ voor q_i in de originele Turing machine en $\delta''(q_i, a) = \delta'$ voor q_i in de herkenner, verder: $\delta''(q_i, a) = (q_j, a, S)$ voor eindtoestanden q_i van de herkenner en begintoestand q_j van de originele Turing machine. Neem als de eindtoestanden van de nieuwe machine die van de originele Turing machine.

2. Voeg voor iedere q_i, a met $\delta(q_i, a) = (q_j, b, S)$ een toestand $\hat{q}_i$ toe aan de machine. Verander δ als volgt:
   Voor iedere q_i, a met $\delta(q_i, a) = (q_j, b, S)$: $\delta(q_i, a) = (\hat{q}_i, b, R)$ en $\delta(\hat{q}_i, c) = \delta(q_j, c, L)$ voor alle $c \in \Sigma \cup \Gamma$.

3. Slechts naar rechts zoeken is niet voldoende, omdat een 0 ook in de 'linkerhelft' van de oneindige tape zou kunnen zitten. Los dit op door heen en weer te zoeken.
   $\delta(q_0, \square) = (q_1, 1, L)$
   $\delta(q_0, 1) = (q_0, 1, R)$
   $\delta(q_1, \square) = (q_0, 1, R)$
   $\delta(q_1, 1) = (q_1, 1, L)$
   Deze machine stopt als het een 0 tegen komt omdat δ dan niet gedefiniëerd is.

4. Vorige serie

5. Vorige serie

6. Serie 5

7. Volg het bewijs voor s-grammaticas.

Opmerkingen en uitwerkingen Opgaven 10

1. z is de nulfunctie, dus neem een machine die de invoer wist. s is de opvolger functie, zie vorige serie. $p_i^n(x_1, dots, x_n)$ wist de invoer, behalve x_i.

2. Neem:
   $\delta(q_{i-1}, 1) = (q_{i-1}, 1, R)$
   $\delta(q_{i-1}, 0) = (q_i, 0, R)$
   Voor $j = 0, \dots, i-2, i, \dots, n-1$:
   $\delta(q_j, 1) = (q_j, \square, R)$
   $\delta(q_j, 0) = (q_j, \square, R)$
   en:
   $\delta(q_{n-1}, \square) = (q_n, \square, L)$
   $\delta(q_n, \square) = (q_n, \square, L)$
   $\delta(q_n, 1) = (q_{n+1}, 1, L)$
   $\delta(q_{n+1}, \square) = (q_{n+2}, \square, R)$

3. Machine berekent eerst f, dan g.

4. Neem voor het gemak een twee-bands Turing machine. Zet op eerste band x, de tweede y. Bereken eerst $f(x)$ op de eerste band. Herhaal de volgende procedure totdat de leeskop van band twee op een $\square$ staat: controleer of de leeskop van de tweede band op een 1 staat, zoja verplaats deze leeskop een plaats naar rechts en bereken g(inhoud eerste band).
   Let op dat bij het berekenen van f, en g de tweede leesband ongemoeid wordt gelaten.

5. De functie $x \mapsto c$ is primitief recursief, immers $c = s^{(c)}(z(x))$, ofwel een compositie van basis primitief recursieve functies.
   Turing machine: een mogelijkheid is om de compositie van machines uit opgave 1 te nemen. Een andere mogelijkheid is om de invoer te wissen en dan c keer een 1 op de band te zetten.

6. (a) $F_1(n) = F_0^{(n)}(n) = n + n = 2n$
   $F_2(n) = F_1^{(n)}(n) = 2^n n$

$F_2(n) = F_2^{(n)}(n) \geq n^{\cdot^{\cdot^{\cdot^{n}}}}$
$F_3(n) = \dots$

(b) Bewijs met inductie naar k, gebruik in de inductiestap primitieve recursie om $h(x,y) := F_k^{(y)}(x)$ te definiëren. Dan: $F_{k+1}(n) = h(p_1^1(n), p_1^1(n))$.

(c) Bewijs met inductie naar k, de inductiestap als hierboven, met hulp van opgave 4.

(d) Inductie naar k: F_0 is zeker strikt stijgend. Stel F_k is strikt stijgend en $n < m$, dan $F_{k+1}(n) = F_k^{(n)}(n) \leq F_k^{(n+1)}(n) \leq \cdots \leq F_k^{(m)}(n) < F_k^{(m)}(m) = F_{k+1}(m)$.

□

(e) Stel $x < y$, we gebruiken inductie naar y:

- $y = x + 1$, dan $F_y(n) = F_x^{(n)}(n) > F_x(n)$.
- $y = x + z + 1$ waarbij $F_x(n) < F_{x+z}(n)$ voor alle n. Dan $F_y(n) = F_{x+z}^{(n)}(n) > F_x(n)$.

□

(f) $z(x) = 0 < F_0(x)$
$s(x) = x + 1 = F_0(x)$
$P_i^n(x_1, \dots, x_n) = x_i \leq \max(\vec{x}) < F_0(\max(\vec{x}))$

(g) We gebuiken inductie over de constructie van de primitief recursieve functies:

- De basisfuncties, zie hierboven.
- Stel er zijn $l, l_1, \dots, l_n$ met $f(\vec{x}) \leq F_l(\max(\vec{x}))$ en $g_i(\vec{y}) \leq F_{l_i}(\max(\vec{y}))$. Dan
$$h(\vec{y}) := f(g_1(\vec{y}), \dots, g_n(\vec{y})) \leq F_l(\max_i(g_i(\vec{y})))$$
F_l is stijgend dus:
$$F_l(\max_i(g_i(\vec{y}))) = \max_i F_l(g_i(\vec{y})) \leq \max_i F_l(F_{l_i}(\max(\vec{y})))$$
(e) levert:
$$\max_i F_l(F_{l_i}(\max(\vec{y}))) \leq F_{l+l_1+\dots l_n+1}(\max(\vec{y}))$$
Dus voor h voldoet $k = l + l_1 + \dots l_n + 1$.
- stel $f(x) \leq F_{k_1}, g(x,y,z) \leq F_{k_2}(\max(x,y,z))$ en
$$h(x, 0) = f(x)$$
$$h(x, s(y)) = g(x, y, h(x, y))$$

Dan

$$h(x,y) \leq F_{k_2}(\max(x,y,h(x,y-1)))$$
$$\vdots$$
$$h(x,1) \leq F_{k_2}(\max(x,y,h(x,0))) = F_{k_2}(\max(x,y,f(x))) \leq F_{k_2}(F_{k_1}(x))$$

Dus:
$h(x,y) \leq F_{k_2}^{(y)}(F_{k_1}(x)) \leq F_{k_1+k_2}^{(\max(y,x))}(\max(x,y)) = F_{k_1+k_2+1}(\max(x,y))$.

$\square$

7. Plaats eerste $a, b, \#$:
   $\delta(q_0, \square) = (q_1, b, L)$
   $\delta(q_1, \square) = (q_2, a, L)$
   $\delta(q_2, \square) = (q_s, \#, S)$
   Voeg telkens een a, b toe:
   $\delta(q_s, \#) = (q_3, a, R)$
   $\delta(q_3, a) = (q_3, a, R)$
   $\delta(q_3, b) = (q_3, b, R)$
   $\delta(q_3, \square) = (q_4, b, L)$
   $\delta(q_4, a) = (q_4, a, L)$
   $\delta(q_4, b) = (q_4, b, L)$
   $\delta(q_4, \square) = (q_s, \#, S)$

8. Nee, er bestaan immers r.o. talen die woorden van lengte ≤ 6 bevatten en er bestaan r.o. talen die niet zulke woorden bevatten (zoek zelf voorbeelden daarvan). Dus de stelling van Rice uit de slides vertelt ons dat er geen beslissingsalgoritme voor deze eigenschap bestaat.

Opmerkingen en uitwerkingen 11

1. Volgens de definitie bestaan er c, n_0 waarvoor voor alle $n > n_0$ geldt: $f(n) < c \cdot (a_m n^m + \cdots + a_0) \leq c \cdot (a_m n^m + \cdots + a_0 n^m) = c(a_m \ldots a_0) \cdot n^m$

   $\square$

2. Construeer een Turing machine die eerst invoer w kopiëert, waarna eerst wordt gekeken of het origineel in L_1 zit, daarna of de kopie in L_2 zit. Om te voorkomen dat de twee processen met elkaar in conflict komen op de band verschuif je de kopie eerst nog $ct(n)$ plaatsen op die volgens het gegeven dient te bestaan. De nieuwe herkenner werkt in tijd $O(nt(n))$.

3. Nee, er bestaan talen die in P zitten, en er bestaan talen die niet in P zitten (zoek zelf zo een taal). Dus volgens de stelling van Rice uit de slides is dit probleem niet beslisbaar.

   $\square$

4. We hebben eerder een Turing machine geconstrueerd uit een edh, deze loopt binnen n+1 stappen af.

   $\square$

5. Deze taal is regulier $\Rightarrow$ zit in P.

6. We gebruiken als certificaat voor de verifyer een codering van twee getallen. De invoer wordt dus $< n, < a, b >>$. De machine werkt nu als volgt: op de eerste tape staat n, de tweede tape a, de derde tape b, bereken eerst a^2 en b^2 op de bijbehorende tape, tel het resultaat b^2 op bij a^2 op de tweede tape, vergelijk tenslotte de inhoud van de tweede tape met de inhoud van de eerste tape. Alle stappen kunnen binnen polynomiale tijd uitgevoerd worden, en de driebands machine kan (stelling uit de slides) binnen polynomiale tijd nagebootst worden op een eenbands machine, dus deze verifyer werkt binnen polynomiale tijd, ofwel het gegeven probleem zit in NP.

   $\square$

7. Ja, gebruik als certificaat voor de verifyer een codering van twee verzamelingen. De verifyer gaat dan na of de twee verzamelingen deelverzamelingen zijn, disjunct zijn en of de sommen van de elementen van de twee verzamelingen aan elkaar gelijk zijn.