$i\,\mathcal{R}\,\mathbb{Z}^+$

Zbirka rešenih nalog za predmet
Izračunljivost in računska zahtevnost

Predgovor

Pred vami je zbirka rešenih nalog za utrjevanje snovi pri predmetu Izračunljivost in računska zahtevnost. Poglavja v zbirki si sledijo v enakem vrstnem redu kot na predavanjih in vajah. Rešitve smo, upamo, dovolj dobro preverili, nikakor pa ne trdimo, da je vsaka od njih »optimalna« (npr. avtomat z najmanjšim številom stanj, gramatika z najmanjšim številom nekončnih simbolov ipd.). Včasih smo načrtno izbrali »suboptimalno« rešitev, če se nam je zdelo, da je enostavnejša ali razumljivejša od »optimalne«.

Zbirka se bo bržkone še dopolnjevala in nadgrajevala, za zdaj pa vam želimo le še veliko užitkov pri reševanju nalog.

Ekipa IRZ,
septembra 2023

Kazalo

Naloge

1 Izreki in dokazi

1. Dokažite, da za vsako celo število n velja, da je število n^3 deljivo z 9 ali pa se od devetkratnika razlikuje za 1.
2. Dokažite, da za množico $S = \{1, 2, 3, \ldots, 9\}$ velja, da vsaka njena podmnožica velikosti 6 vsebuje vsaj en par števil z vsoto 10.
3. Dokažite, da za vsak algoritem za stiskanje brez izgub obstajajo vhodi, ki jih algoritem ne stisne. Dokažite še naslednje: če algoritem strogo stisne vsaj en vhod, potem obstaja tudi vhod, ki ga razširi. Lahko predpostavite, da stiskamo dvojiške nize.
4. Naj bosta m, $n \in \mathbb{Z}$ in $m + n \geqslant 11$. Dokažite, da velja $m \geqslant 5$ ali $n \geqslant 6$.
5. Dokažite, da je število $n \in \mathbb{Z}$ liho natanko takrat, ko je število $n^2 + 4n + 6$ liho.
6. Dokažite, da ne obstaja najmanjše pozitivno racionalno število.
7. Dokažite, da je praštevil neskončno mnogo.
8. Dokažite, da je $\sqrt[3]{2}$ iracionalno število.
9. Dokažite, da za vsako naravno število $n \geqslant 1$ velja $1 + 2 + \ldots + n = \frac{n(n+1)}{2}$.
10. Dokažite, da za vsako naravno število $n \geqslant 2$ in realno število $x > 0$ velja $(1 + x)^n > 1 + nx$.
11. Naj bo h višina uravnoteženega dvojiškega drevesa z n vozlišči. Dokažite, da velja $\log_2(n + 1) - 1 \leqslant h \leqslant \log_2 n$.
12. Dokažite, da za vsako končno množico A velja $|2^A| = 2^{|A|}$.
13. Dokažite, da je
    - sodih števil toliko kot naravnih;
    - celih števil toliko kot naravnih;
    - pozitivnih racionalnih števil toliko kot naravnih;
    - realnih števil z intervala $[0, 1)$ več kot naravnih.

2 Končni avtomati

1. Zapišite DKA (deterministični končni avtomat), ki sprejema besede $w \in \{0, 1\}^*$, ki se začnejo z nizom 0011.
2. Zapišite DKA, ki sprejema besede $w \in \{0, 1\}^*$, ki vsebujejo liho število ničel.

3. Zapišite DKA, ki sprejema besede $w \in \{0,1\}^*$, ki se končajo z nizom 01 ali 10.

4. Zapišite DKA, ki sprejema besede $w \in \{0,1\}^*$, v katerih vsaki ničli sledi natanko ena enica ali natanko tri enice.

5. Zapišite DKA, ki sprejema presek jezikov iz nalog 1 in 2.

6. Zapišite DKA, ki sprejema unijo jezikov iz nalog 1 in 2.

7. Zapišite DKA, ki sprejema besede $w \in \{0,1\}^*$, ki se ne pričnejo niti z 000 niti z 111.

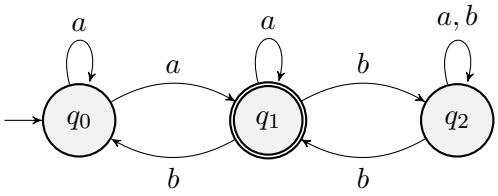
8. Zapišite DKA, ki sprejema besede $w \in \{0,1\}^*$, ki predstavljajo števila v dvojiškem zapisu, ki so deljiva s 5.

9. Zapišite DKA, ki sprejema besede $w \in \{0,1\}^*$, v katerih je tretji simbol s konca besede enak 1.

10. Zapišite NKA (nedeterministični končni avtomat), ki sprejema jezik iz naloge 9.

11. Za NKA iz naloge 10 narišite drevo izvajanja za besedi 1011 in 10110.

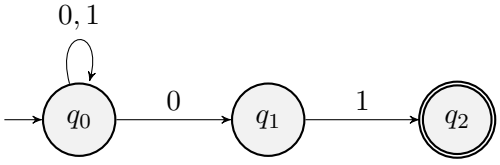
12. Narišite drevo izvajanja, ki ga dobimo, če sledečemu avtomatu podtaknemo besedo *aabbba*:



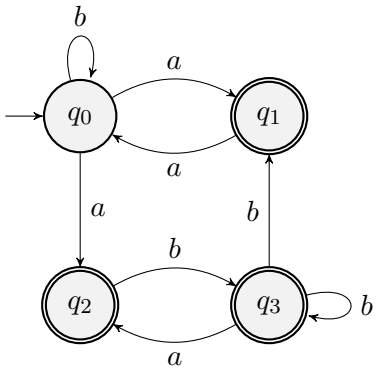
13. Zapišite NKA, ki sprejema besede $w \in \{a,b\}^*$, ki se začnejo z 0 ali 1 ponovitvijo simbola a, nadaljujejo z 0 ali 1 ponovitvijo simbola b, končajo pa z 0 ali 1 ponovitvijo niza ab.

14. Zapišite NKA, ki sprejema besede $w \in \{a,b\}^*$, ki vsebujejo vsaj po eno pojavitev niza aa in niza bb ali pa nobene pojavitve nizov aa in bb.

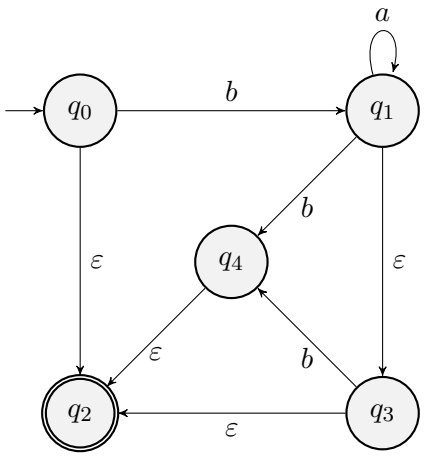
15. Za spodaj podani NKA zapišite pretvorbo v DKA. Najprej jo zapišite po definiciji — podajte vsa možna stanja in pripadajoče prehode po vseh simbolih abecede. Nato narišite končni avtomat, pri katerem podate samo tista stanja, ki so dosegljiva iz začetnega stanja.



16. Spodaj podani NKA pretvorite v DKA. Rišite samo dosegljiva stanja.



17. Spodaj podani ε-NKA pretvorite najprej v NKA, nato pa še v DKA.



3 Regularni izrazi

1. Zapišite nekaj primerov besed v jeziku za regularne izraze nad $\Sigma = \{0, 1\}$:

   (a) $L_1 = L(0)$.

   (b) $L_2 = L(1 + 0^*)$.

   (c) $L_3 = L((101)^* + 01(10)^*)$.

   (d) $L_4 = L(01 + (0 + 01)^*)$.

2. Zapišite regularni izraz nad $\Sigma = \{0, 1\}$, ki opisuje vse besede, ki ne vsebujejo podniza 01.

3. Nad abecedo $\Sigma = \{a, b\}$ sestavite regularne izraze za sledeče jezike:

   (a) besede, ki se ne končajo z aa;

   (b) besede, ki vsebujejo sodo mnogo znakov a;

   (c) besede, ki vsebujejo liho mnogo znakov a;

   (d) besede, ki vsebujejo sodo mnogo znakov a in sodo mnogo znakov b.

4. Regularni izraz $(01)^*(10)^*$ po pravilih pretvorite v ε-NKA. Nato ga poskusite še čimbolj poenostaviti.

5. Regularni izraz $(0 + 1)^*00$ po pravilih pretvorite v ε-NKA. Nato ga poskusite še čimbolj poenostaviti.

6. Podan je regularni izraz $0(1+0)^*11 + (101)^*$.

   (a) Zapišite DKA, ki sprejema isti jezik, kot ga opisuje zgornji regularni izraz.

   (b) Koliko besed dolžine 15 sprejema ta avtomat?

   (c) Koliko besed dolžine 15 je v komplementu jezika, podanega z zgornjim regularnim izrazom?

   (d) Za poljuben n zapišite, koliko besed dolžine n je v komplementu jezika, podanega z zgornjim regularnim izrazom.

4 Lema o napihovanju za regularne jezike

1. Dokažite, da jezik $L = \{0^k 1^k \mid k \in \mathbb{N}\}$ ni regularen.
2. Dokažite, da jezik $L = \{a^i b^j \mid i < j\}$ ni regularen.
3. Dokažite, da jezik $L = \{a^i b^j \mid i > j\}$ ni regularen.
4. Dokažite, da jezik $L = \{xx^R \mid x \in (0+1)^*\}$, pri čemer je x^R zrcalna slika niza x (npr. $10010^R = 01001$), ni regularen.
5. Dokažite, da jezik $L = \{1^p \mid p \text{ je praštevilo}\}$ ni regularen.
6. Dokažite, da jezik $L = \{a^{k^3} \mid k \in \mathbb{N}\}$ ni regularen.
7. Dokažite, da jezik $L = \{a^i b^j \mid i \neq j\}$ ni regularen.
8. Dokažite, da je jezik $L = \{a^{3k} \mid k \in \mathbb{N}\}$ regularen. Na besedi $z = a^{3n}$ uporabite lemo o napihovanju. Kje se »zalomi«?
9. Za jezik $a^*b(ba)^*a$ poiščite n iz leme o napihovanju. Za dve dovolj dolgi besedi poiščite delitev, ki napihnjeno besedo ohranja v jeziku.
10. Naj $\#_a(w)$ označuje število pojavitev simbola a v besedi w. Ali je jezik

$$L = \{w \in (a+b)^* \mid \#_a(w) \bmod 3 = 0 \wedge \#_b(w) \bmod 4 = 0\}$$

regularen? Odgovor utemeljite!

5 Kontekstno neodvisne gramatike

1. Sestavite kontekstno neodvisno gramatiko za jezik $(0 + 1^*)(01 + \varepsilon)$.
2. Sestavite kontekstno neodvisno gramatiko za jezik $(1(01)^* + 00)^*$.
3. Sestavite kontekstno neodvisno gramatiko za jezik $\{a^n b c^n \mid n \in \mathbb{N}\}$.
4. Sestavite kontekstno neodvisno gramatiko za jezik palindromov nad abecedo $\Sigma = \{a, b\}$.
5. Sestavite kontekstno neodvisno gramatiko za jezik pravilnih oklepajnih izrazov. Nekaj primerov besed iz jezika: ε, (), ((())), (()())((())), (()())(()(())).
6. Sestavite kontekstno neodvisno gramatiko za jezik $L = \{a^i b^j c^k \mid i \neq j \vee j \neq k\}$.
7. Sestavite kontekstno neodvisno gramatiko za jezik $L = \{0^i 1^j 0^k \mid i + j \geqslant 2k\}$.

8. Za gramatiko $S \to SS \mid (S) \mid \varepsilon$

   (a) poiščite skrajno levo izpeljavo niza $w = ((()))()$;

   (b) poiščite skrajno desno izpeljavo niza w;

   (c) narišite drevo izpeljav za niz w.

9. Za gramatiko

$$\begin{aligned} S &\to aB \mid bA \\ A &\to a \mid aS \mid bAA \\ B &\to b \mid bS \mid aBB \end{aligned}$$

   pokažite, da je dvoumna.

10. Pretvorite dvoumno gramatiko

$$E \to E + E \mid E \cdot E \mid (E) \mid 0 \mid \ldots \mid 9$$

    v nedvoumno.

11. Dokažite, da so regularni jeziki vsebovani v kontekstno neodvisnih jezikih (tj. da za vsak regularni izraz E obstaja kontekstno neodvisna gramatika G, da je $L(G) = L(E)$).

12. Dokažite, da je jezik regularnih izrazov (npr. nad abecedo $\{0, 1\}$) kontekstno neodvisen, ne pa tudi regularen, jezik regularnih izrazov brez oklepajev pa regularen.

6 Skladovni avtomati

1. Podan je skladovni avtomat $(Q, \{a, b\}, \{Z, S\}, \delta, q_0, Z, \{q_F\})$ s sledečimi prehodi:

$$\begin{aligned} \delta(q_0, a, Z) &= \{(q_0, SZ), (q_0, SSZ)\} \\ \delta(q_0, a, S) &= \{(q_0, SS), (q_0, SSS)\} \\ \delta(q_0, b, S) &= \{(q_1, \varepsilon)\} \\ \delta(q_1, b, S) &= \{(q_1, \varepsilon)\} \\ \delta(q_1, \varepsilon, Z) &= \{(q_F, \varepsilon)\} \end{aligned}$$

   (1) Narišite drevo (ali graf) trenutnih opisov za besede *aab*, *aabbb* in *aabbbbb*.

   (2) Kateri jezik sprejema ta skladovni avtomat?

   (3) Zapišite kontekstno neodvisno gramatiko, ki tvori isti jezik.

2. Zapišite skladovni avtomat za jezik

$$L = \{w \in \{a, b\}^* \mid \#_a(w) = 3\}.$$

3. Zapišite skladovni avtomat za jezik

$$L = \{a^n b^n \mid n \geqslant 0\}.$$

4. Zapišite *deterministični* skladovni avtomat za jezik

$$L = \{a^n b^n \mid n \geqslant 0\}.$$

5. Zapišite skladovni avtomat za jezik

$$L = \{a^{2n}b^n \mid n \geqslant 0\}.$$

6. Zapišite skladovni avtomat za jezik

$$L = \{w \in (a+b)^* \mid \#_a(w) = \#_b(w)\}.$$

7. Zapišite skladovni avtomat za jezik

$$L = \{a^i b^j c^k \mid i + j = k,\ i \geqslant 0,\ j \geqslant 0\}.$$

8. Zapišite skladovni avtomat za jezik

$$L = \{a^i b^j c^k \mid i + k = j,\ i \geqslant 0,\ k \geqslant 0\}.$$

9. Zapišite skladovni avtomat za jezik

$$L = \{w \in \{a,b\}^* \mid w = w^R\}.$$

10. Zapišite skladovni avtomat, ki sprejema isti jezik, kot ga tvori sledeča kontekstno neodvisna gramatika:

$$\begin{aligned} S &\to ASA \mid T \\ A &\to Aa \mid Ab \mid \varepsilon \\ T &\to cTd \mid \varepsilon \end{aligned}$$

7 Lema o napihovanju za kontekstno neodvisne jezike

Z uporabo leme o napihovanju pokažite, da sledeči jeziki niso kontekstno neodvisni:

1. $L = \{a^n b^n c^n \mid n \geqslant 0\}$.
2. $L = \{a^i b^j c^k \mid i < j \wedge i < k\}$.
3. $L = \{a^i b^j a^i b^j \mid i, j \geqslant 0\}$.
4. $L = \{a^p \mid p \text{ je praštevilo}\}$.
5. $L = \{ww \mid w \in (a+b)^*\}$.
6. $L = \{ww^R w \mid w \in (a+b)^*\}$.

Za vsakega od sledečih jezikov določite, ali je regularen, kontekstno neodvisen ali nič od tega.

7. $L = \{a^i b^j a^j b^i \mid i, j \geqslant 0\}$.
8. $L = \{w \in (a+b)^* \mid \#_a(w) \bmod 3 = 0 \wedge \#_b(w) \bmod 4 = 0\}$.
9. $L =$ jezik regularnih izrazov nad abecedo $\{0, 1\}$.

8 Turingovi stroji

1. Zapišite Turingov stroj, ki razpoznava soda dvojiška števila. Z uporabo trenutnih opisov simulirajte izvajanje stroja nad besedo 10010.

2. Zapišite Turingov stroj, ki razpoznava jezik $L = \{a^i b^i \mid i \geqslant 0\}$. Simulirajte izvajanje stroja nad besedo *aabb*.

3. Zapišite Turingov stroj za jezik $L = \{a^i b^i c^i \mid i \geqslant 0\}$.

4. Zapišite Turingov stroj za jezik $L = \{w\#w \mid w \in (0+1)^*\}$.

5. Zapišite Turingov stroj za jezik $L = \{a^i b^j c^k \mid k = ij,\ i > 0,\ j > 0\}$.

6. Zapišite Turingov stroj za jezik $L = \{a^{2^k} \mid k \geqslant 0\}$. Kako bi stroj prilagodili, da bi izračunal vrednost $\lceil \log_2 n \rceil$, kjer je n dolžina vhodne besede?

7. Zapišite Turingov stroj, ki podano dvojiško število pomnoži z 2.

8. Zapišite Turingov stroj, ki podanemu dvojiškemu številu prišteje 1.

9. Zapišite Turingov stroj, ki podvoji podano eniško število. (Število n je v eniškem zapisu predstavljeno kot 1^n.)

9 Razširitve Turingovih strojev

1. Zapišite Turingov stroj, ki razpoznava jezik $L = \{a^i b^i c^i \mid i \geqslant 1\}$. Uporabite štiri trakove.

2. Zapišite Turingov stroj s tremi trakovi, ki sešteje števili, podani v dvojiškem zapisu. Rezultat naj bo zapisan na tretjem traku. Vhodni števili sta zapisani na prvem traku, ločeni pa sta z znakom +.

3. Za nedeterministični Turingov stroj, podan s prehodi

$$\begin{aligned}
\delta(q_0, 0) &= \{(q_0, 1, R)\} \\
\delta(q_0, 1) &= \{(q_1, 0, R)\} \\
\delta(q_1, 0) &= \{(q_1, 0, R), (q_0, 0, L)\} \\
\delta(q_1, 1) &= \{(q_1, 1, R), (q_0, 1, L)\} \\
\delta(q_1, B) &= \{(q_2, B, R)\}
\end{aligned}$$

   kjer je B prazni tračni simbol in $F = \{q_2\}$, narišite drevo izvajanj za vhodni niz 011.

4. Zapišite nedeterministični Turingov stroj za jezik $L = \{ww \mid w \in (0+1)^*\}$. Uporabite lahko poljubno število trakov.

5. Zapišite nedeterministični enotračni Turingov stroj za jezik $L = \{ww \mid w \in (0+1)^*\}$.

6. Zapišite nedeterministični Turingov stroj, ki preveri, ali je podano eniško število sestavljeno. Uporabite lahko poljubno število trakov.

7. Zapišite deterministični Turingov stroj, ki preveri, ali je podano eniško število sestavljeno. Uporabite lahko poljubno število trakov.

8. Pokažite, da je razred odločljivih jezikov zaprt za unijo in presek.

10 Odločitveni problemi in univerzalni Turingov stroj

1. Odločitveni problem razbitja množice je definiran takole:

   > Naj bo $A \subset \mathbb{Z}^+$ neka končna množica. Ali obstajata množici $A_1 \subseteq A$ in $A_2 \subseteq A$, tako da je $A_1 \cap A_2 = \varnothing$, $A_1 \cup A_2 = A$ in $\sum_{x \in A_1} x = \sum_{x \in A_2} x$?

   Zamislite si kodiranje primerkov tega problema in zapišite Turingov stroj, ki sprejema jezik pozitivnih primerkov problema (torej kode primerkov, za katere obstaja razbitje).

2. Definirajte problem iskanja maksimuma v zaporedju pozitivnih celih števil kot odločitveni problem. Zapišite Turingov stroj za ta problem.

3. Optimizacijski problem 0-1 nahrbtnika je podan takole:

   > Podanih je $n > 0$ predmetov s prostorninami $v_1, v_2, \ldots, v_n \in \mathbb{Z}^+$ in cenami $c_1, c_2, \ldots, c_n \in \mathbb{Z}^+$. Poišči največjo možno skupno ceno predmetov, ki jih je mogoče spraviti v nahrbtnik prostornine $V \in \mathbb{Z}^+$.

   Definirajte táko odločitveno različico tega problema, da bo optimizacijsko različico mogoče rešiti s karseda malo primerki odločitvene različice. Zapišite Turingov stroj, ki rešuje odločitveno različico.

4. Zapišite Turingov stroj, ki sprejme opise takih Turingovih strojev, ki imajo vsaj en prehod iz nekončnega v končno stanje.

5. Dokažite, da obstaja jezik, za katerega ne obstaja Turingov stroj.

6. Dokažite naslednje lastnosti odločljivih in polodločljivih jezikov:

   (a) Komplement odločljivega jezika je odločljiv jezik.
   (b) Unija dveh odločljivih jezikov je odločljiv jezik.
   (c) Unija dveh polodločljivih jezikov je polodločljiv jezik.
   (d) Če sta jezika L in $\overline{L}$ (komplement jezika L) oba polodločljiva, potem sta oba odločljiva.

7. Dokažite, da obstaja polodločljiv jezik, ki ni odločljiv.

11 Prevedbe

1. Dokažite: če je jezik L odločljiv, je odločljiv tudi jezik $\overline{L}$.

2. Dokažite: če sta jezika L in $\overline{L}$ polodločljiva, sta oba odločljiva.

3. Naj bo jezik L_1 prevedljiv na jezik L_2. Dokažite:

   (a) če je L_1 neodločljiv, potem je L_2 neodločljiv;
   (b) če L_1 ni polodločljiv, potem L_2 prav tako ni polodločljiv.

4. Dokažite, da obstaja jezik, za katerega ni Turingovega stroja.

5. Dokažite da jezik $\overline{L_u} = \{\langle M, w\rangle \mid w \notin L(M)\}$ ni polodločljiv.

6. Dokažite da je jezik $L_u = \{\langle M, w\rangle \mid w \in L(M)\}$ polodločljiv, ne pa tudi odločljiv.

7. Dokažite, da je jezik $L_{ne} = \{\langle M \rangle \mid L(M) \neq \varnothing\}$ polodločljiv, ne pa tudi odločljiv.
8. Dokažite, da jezik $L_e = \{\langle M \rangle \mid L(M) = \varnothing\}$ ni polodločljiv.
9. Dokažite, da jezik $L_{42} = \{\langle M \rangle \mid |L(M)| = 42\}$ ni polodločljiv.
10. Dokažite, da jezik $L_* = \{\langle M \rangle \mid L(M) = \Sigma^*\}$ ni polodločljiv.
11. Je jezik $L_{re} = \{\langle M \rangle \mid L(M)$ je polodločljiv$\}$ odločljiv? Je polodločljiv? Dokažite!
12. Dokažite, da jezik $L_r = \{\langle M \rangle \mid L(M)$ je odločljiv$\}$ ni polodločljiv.
13. Dokažite, da jezik $L_{nr} = \{\langle M \rangle \mid L(M)$ ni odločljiv$\}$ ni polodločljiv.
14. Dokažite, da jezik $L = \{\langle M_1, M_2 \rangle \mid L(M_1) \cap L(M_2) = \varnothing\}$ ni polodločljiv.
15. Ali je jezik $L = \{\langle M_1, M_2 \rangle \mid L(M_1) \backslash L(M_2) \neq \varnothing\}$ odločljiv? Je polodločljiv? Dokažite!
16. Podan je jezik $L = \{\langle M, w_1, G, w_2 \rangle \mid w_1 \in L(M) \wedge w_2 \notin L(G)\}$, kjer je G kontekstno neodvisna gramatika. Je L odločljiv? Je L polodločljiv? Dokažite!
17. Naj bo $f(M, w)$ število korakov, ki jih izvede Turingov stroj M, če mu na vhodu podamo besedo $w \in \{0,1\}^*$.
    (a) Zapišite primer stroja M, za katerega velja $f(M, w) = 2\,|w|$ za vsak $w \in \{0,1\}^*$.
    (b) Je jezik $L = \{\langle M, w \rangle \mid f(M, w) = 2\,|w|\}$ odločljiv? Je polodločljiv? Odgovora utemeljite!
    (c) Je jezik $L = \{\langle M_1, M_2 \rangle \mid \exists w \in \{0,1\}^* \colon f(M_1, w) < f(M_2, w)\}$ odločljiv? Je polodločljiv? Odgovora utemeljite!

Rešitve

1 Izreki in dokazi

1. Dokaz s konstrukcijo. Vsako število n lahko zapišemo na enega od sledečih načinov: (1) $n = 3k$; (2) $n = 3k + 1$; (3) $n = 3k - 1$. Torej:

   (1) $(3k)^3 = 27k^3 = 9l$

   (2) $(3k + 1)^3 = 27k^3 + 27k^2 + 9k + 1 = 9l' + 1$

   (3) $(3k - 1)^3 = 27k^3 - 27k^2 + 9k - 1 = 9l'' - 1$

2. Načelo golobnjaka. Množico S lahko razbijemo na množice $A_1 = \{1, 9\}$, $A_2 = \{2, 8\}$, $A_3 = \{3, 7\}$, $A_4 = \{4, 6\}$ in $A_5 = \{5\}$. Vsako število iz množice S pripada natanko eni od množic $A_1, \ldots, A_5$. Ker moramo izbrati šest števil, množic pa je 5, obstaja množica, v katero padeta dve od izbranih števil. Vsota takih števil je 10.

3. Načelo golobnjaka (in dokaz s protislovjem). Obstajata dva vhoda dolžine 1: beseda 0 in beseda 1. Obe besedi bi lahko stisnili kvečjemu v ε, a ju potem med seboj ne bi mogli razlikovati. To pomeni, da lahko stisnemo kvečjemu eno od besed 0 in 1.

   Naj bo w najkrajša beseda, ki jo algoritem strogo stisne, in naj bo $n = |w|$. Vse krajše besede pri stiskanju ohranijo svojo dolžino. Recimo, da algoritem besedo w stisne na dolžino $k < n$. To pomeni, da množico $2^k + 1$ besed (vse besede dolžine k in besedo w) preslika v množico besed dolžine k. Ker je moč te množice enaka 2^k, obstajata dve različni besedi, ki ju algoritem preslika v isto besedo. To pa je v nasprotju s predpostavko o brezizgubnem stiskanju.

4. Dokaz s protislovjem. Če je $m < 5$ in $n < 6$, potem je $m + n < 11$.

5. ($\Longrightarrow$) Če je n lih, ga lahko zapišemo kot $n = 2k + 1$. Od tod sledi $n^2 + 4n + 6 = (2k + 1)^2 + 4(2k + 1) + 6 = 4k^2 + 12k + 11 = 2(2k^2 + 6k + 5) + 1$. To število je očitno liho.

   ($\Longleftarrow$) Če je $n^2 + 4n + 6$ liho, je liho tudi $n^2 + 4n = n(n + 4)$. Torej je tudi n lih; če bi bil sod, bi bil produkt $n(n+4)$ tudi sod. Možen je tudi dokaz s kontrapozicijo: trditev $n^2 + 4n + 6$ liho $\Longrightarrow$ n liho lahko prepišemo v n sodo $\Longrightarrow$ $n^2 + 4n + 6$ sodo.

6. Dokaz s protislovjem. Recimo, da je x najmanjše pozitivno racionalno število. Če ga delimo z 2, dobimo še manjše pozitivno racionalno število, kar pomeni, da x ni najmanjše tako število.

7. Dokaz s protislovjem. Recimo, da obstajajo samo praštevila $p_1, p_2, \ldots, p_k$. Število $n = p_1 p_2 \ldots p_k + 1$ ni deljivo z nobenim od števil $p_1, p_2, \ldots, p_k$, kar pomeni, da je najmanjše praštevilo, s katerim je deljivo, večje od p_k. Ta ugotovitev pa je v nasprotju s predpostavko, da je p_k največje praštevilo.

8. Dokaz s protislovjem. Če je $\sqrt[3]{2}$ racionalen, ga lahko zapišemo kot $\sqrt[3]{2} = a \,/\, b$, kjer je $\gcd(a, b) = 1$. Torej velja $a^3 = 2b^3$. Število a^3 je torej sodo, kar pomeni, da je tudi

število a sodo. Torej je $a = 2k$ in $(2k)^3 = 2b^3$. Od tod sledi $b^3 = 4k^3$. Zato sta tudi števili b^3 in (potemtakem) b sodi. Torej je $\gcd(a, b) > 1$ — protislovje.

9. Dokaz s popolno indukcijo. Za $n = 1$ trditev očitno drži ($1 = \frac{1\cdot 2}{2}$), indukcijski korak $T(n) \implies T(n+1)$ pa pokažemo tako, da iz indukcijske predpostavke $\sum_{i=1}^{n} i = n(n+1)\,/\,2$ izpeljemo $\sum_{i=1}^{n+1} i = (n+1)(n+2)\,/\,2$:

$$1 + 2 + \ldots + n + (n+1) = \frac{n(n+1)}{2} + n + 1 = \frac{n^2 + n + 2n + 2}{2} = \frac{(n+1)(n+2)}{2}.$$

V prvem koraku smo uporabili indukcijsko predpostavko.

10. Dokaz s popolno indukcijo. Pri $n = 2$ imamo $(1+x)^2 = 1 + 2x + x^2 > 1 + 2x$. Predpostavimo, da velja $(1+x)^n > 1 + nx$, in računajmo:

$$(1+x)^{n+1} = (1+x)^n(1+x) > (1+nx)(1+x) = 1+x+nx+nx^2 = 1+(n+1)x+nx^2 > 1 + (n+1)x.$$

Tako smo iz predpostavke, da trditev velja za število n, izpeljali, da trditev velja tudi za število $n + 1$.

11. Najprej dokažimo, da ima polno uravnoteženo drevo višine h natanko $2^{h+1}-1$ vozlišč. Tole bo popolna indukcija: za $h = 0$ trditev očitno drži, polno uravnoteženo drevo višine $h + 1$ pa je sestavljeno iz korena in dveh polno uravnoteženih dreves višine h, kar po induktivni predpostavki pomeni, da ima $1 + 2(2^{h+1} - 1) = 2^{h+2} - 1$ vozlišč. Uravnoteženo drevo višine h ima torej najmanj 2^h in največ $2^{h+1} - 1$ vozlišč, višina drevesa z n vozlišči pa je potemtakem celo število z intervala $[\log_2(n+1) - 1, \log_2 n]$.

12. Popolna indukcija. Ker je $|2^{\varnothing}| = |\{\varnothing\}| = 1 = 2^{|\varnothing|}$, trditev za $n = 0$ očitno drži. Recimo, da trditev $|2^A| = 2^{|A|}$ drži za poljubno množico A z n elementi, torej $A = \{a_1, a_2, \ldots, a_n\}$. Pokažimo, da velja tudi za množico $A' = \{a_1, a_2, \ldots, a_{n+1}\}$. Potenčna množica množice A' je sestavljena iz

- množic iz množice 2^A (teh je po induktivni predpostavki 2^n);
- množic iz množice 2^A, ki jim je dodan element a_{n+1} (teh je prav tako 2^n).

Ker velja $2^n + 2^n = 2^{n+1}$, velja tudi $|2^{A'}| = 2^{|A'|}$.

13. Če med neskončnima množicama A in B najdemo bijekcijo, imata enako moč.

- Naj bo S množica vseh sodih števil. Preslikava $f\colon \mathbb{N} \to S$ s predpisom $f(n) = 2n$ je bijekcija, zato imata množici enako moč.
- Preslikava $f\colon \mathbb{N} \to \mathbb{Z}$ s predpisom

$$f(n) = \begin{cases} \frac{n}{2}, & \text{če je } n \text{ sod;} \\ \frac{-n-1}{2}, & \text{če je } n \text{ lih,} \end{cases}$$

  je bijekcija.
- Pozitivna racionalna števila so natanko ulomki $\frac{a}{b}$, kjer je $a, b \in \mathbb{Z}^+$. Ulomke uredimo po naraščajoči vsoti $a + b$, v okviru iste vsote pa po naraščajočem številu: $\frac{1}{1}$, $\frac{1}{2}$, $\frac{2}{1}$, $\frac{1}{3}$, $\frac{2}{2}$, $\frac{3}{1}$, $\frac{1}{4}$, $\frac{2}{3}$, $\frac{3}{2}$, $\frac{4}{1}$ … Ulomke zaporedno oštevilčimo in tako definiramo bijekcijo $\mathbb{N} \to \mathbb{Q}^+$.
- Če bi obstajala bijekcija $f\colon \mathbb{N} \to [0, 1)$, bi jo lahko zapisali v obliki tabele:

n	$f(n)$
1	$0{,}a_{11}a_{12}a_{13}a_{14}\ldots$
2	$0{,}a_{21}a_{22}a_{23}a_{24}\ldots$
3	$0{,}a_{31}a_{32}a_{33}a_{34}\ldots$
4	$0{,}a_{41}a_{42}a_{43}a_{44}\ldots$
$\vdots$	$\vdots$

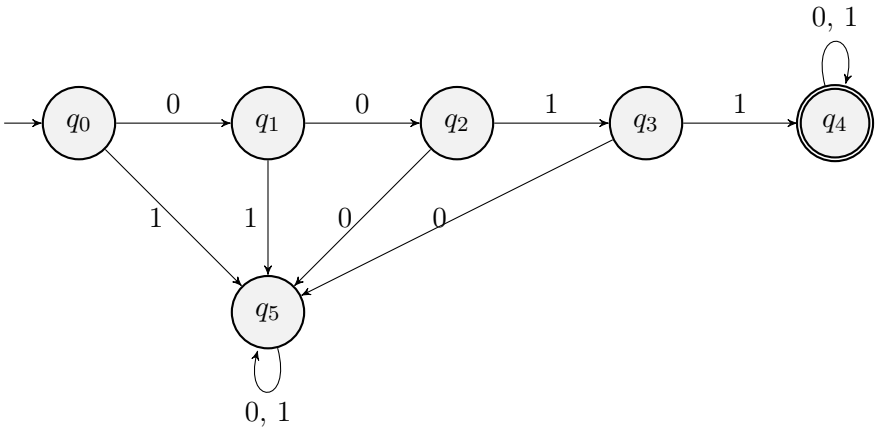
Če je f bijekcija, mora tabela vsebovati vsa realna števila z intervala $[0, 1)$. Definirajmo

$$b_i = \begin{cases} 1 & \text{če je } a_{ii} \neq 1; \\ 2 & \text{če je } a_{ii} = 1. \end{cases}$$

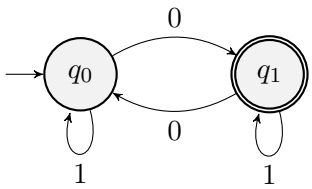
Število $0{,}b_1b_2b_3b_4\ldots$ je realno število v intervalu $[0, 1)$, a se v vsaj eni decimalki razlikuje od vseh števil v tabeli. Tabela potemtakem ne vsebuje vseh realnih števil z intervala $[0, 1)$, preslikava f pa zato ni bijekcija.

2 Končni avtomati

1. Do končnega stanja prispemo natanko v primeru, če se niz prične z 0011. Če kjerkoli skrenemo z glavne poti, obtičimo v slepem stanju. (Pri DKA mora biti ciljno stanje definirano za vsak par stanja in vhodnega simbola.)



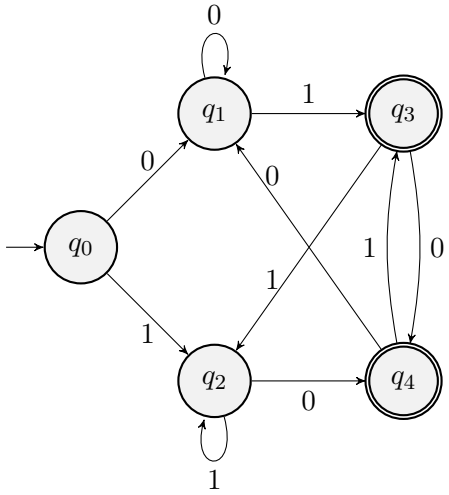
2. Do stanja q_0 vodijo besede s sodim številom ničel, do stanja q_1 pa besede z lihim številom ničel.



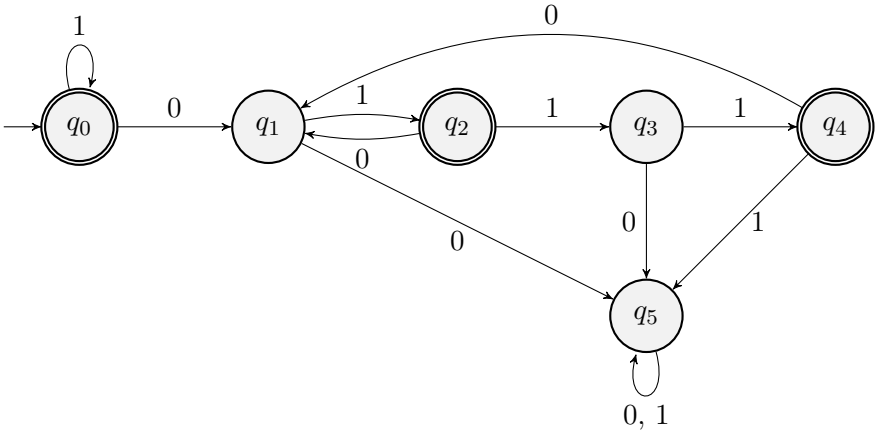
3. Stanje q_0 je začetno, ostala štiri stanja pa imajo tak pomen:

   - q_1: pravkar smo prebrali 0, pred tem pa nismo prebrali 1;
   - q_2: pravkar smo prebrali 1, pred tem pa nismo prebrali 0;
   - q_3: pravkar smo prebrali 01;

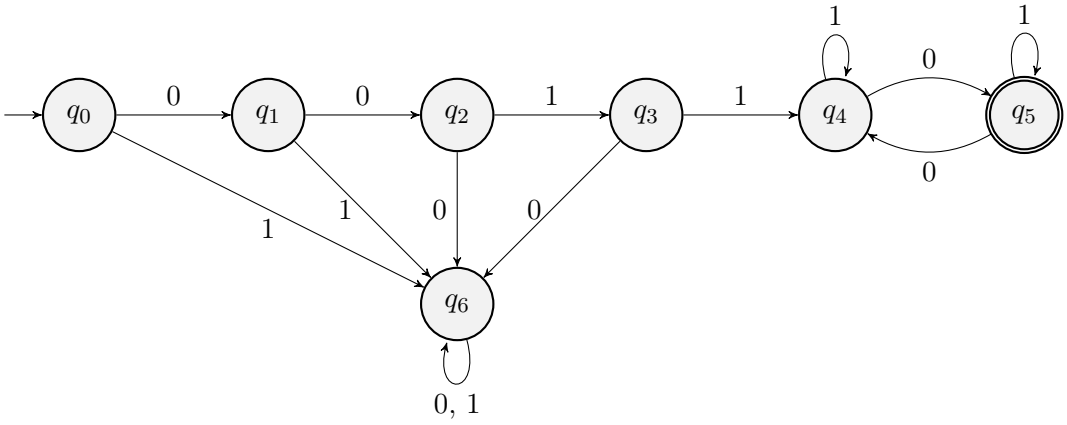
- q_4: pravkar smo prebrali 10.



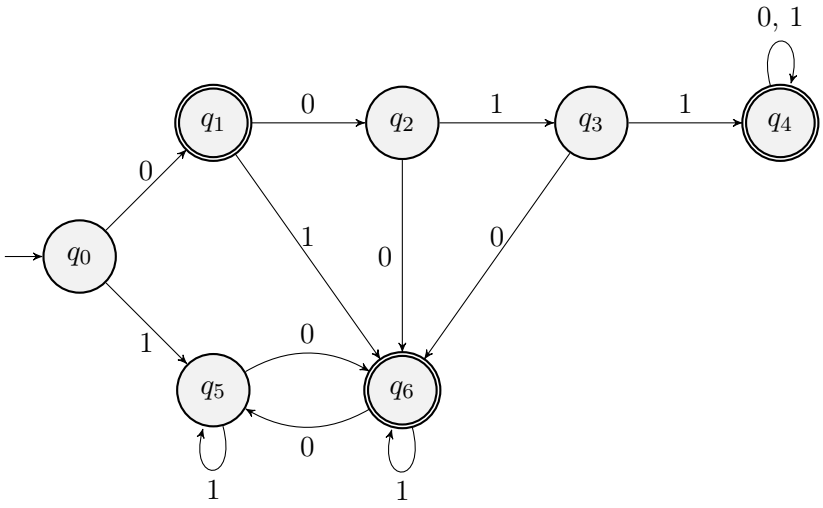
4. Če ničli sledi 0, 110 ali 1111, obtičimo v slepem stanju. Stanje q_1 si lahko predstavljamo kot vstopno točko v podprogram za preverjanje števila enic po pravkar prebrani ničli.



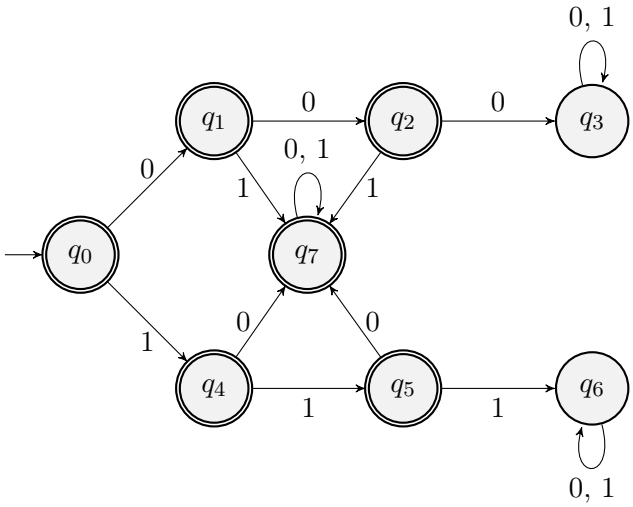
5. Lahko zgradimo produktni avtomat, lahko pa do cilja pridemo tudi hitreje: na začetku moramo prebrati 0011 (karkoli drugega nas privede v slepo stanje), od tam naprej pa moramo zagotoviti, da bo število ničel liho.



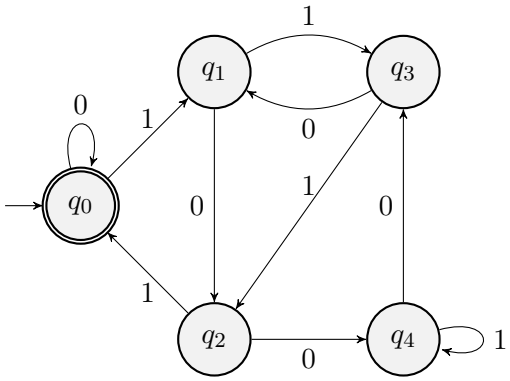
6. Lahko zgradimo ε-NKA in ga pretvorimo v DKA, lahko pa si pomagamo s sledečim razmislekom: če na začetku preberemo 0011, smo »zmagali«, če kjerkoli skrenemo s te poti, pa vstopimo v podavtomat za preverjanje lihosti števila ničel, in sicer v stanje, ki pripada pariteti doslej prebranega števila ničel.



7. Ker je jezik komplement jezika $L = \{w \in \{0,1\}^* \mid w \text{ se prične z } 000 \text{ ali } 111\}$, lahko nalogo rešimo tako, da zapišemo DKA za jezik L, nato pa končna stanja pretvorimo v nekončna, nekončna pa v končna. Toda pozor: to pravilo velja le za DKA, ne pa tudi za NKA!



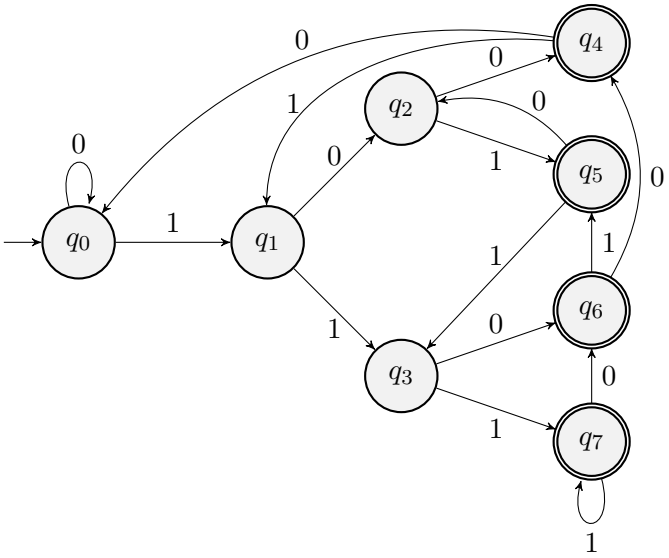
8. Avtomat vsebuje stanja $q_0, \ldots, q_4$. Stanje q_i nam pove, da je doslej prebrani del števila enak i po modulu 5. Če smo nekoliko formalnejši: $\delta(q_0, w) = q_{\langle w \rangle \bmod 5}$, kjer $\langle w \rangle$ predstavlja številsko vrednost besede w. Pri določanju prehodov upoštevamo, da velja $\langle w0 \rangle \equiv 2\langle w \rangle \pmod 5$ in $\langle w1 \rangle \equiv 2\langle w \rangle + 1 \pmod 5$.



9. Potrebujemo $2^3 = 8$ stanj. Pri besedi dolžine 3 ali več je njihov pomen tak:

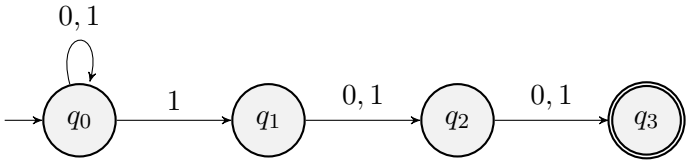
   - q_0: pravkar smo prebrali 000;
   - q_1: pravkar smo prebrali 001;
   - q_2: pravkar smo prebrali 010;
   - q_3: pravkar smo prebrali 011;
   - q_4: pravkar smo prebrali 100;
   - q_5: pravkar smo prebrali 101;
   - q_6: pravkar smo prebrali 110;
   - q_7: pravkar smo prebrali 111.

   Pri prehodih se ravnamo po sledečem primeru. Če, denimo, v stanju q_5 preberemo ničlo, preidemo v stanje q_2 ($101 \mapsto [1]010$), enica pa nas pripelje v stanje q_3 ($101 \mapsto [1]011$). V splošnem: v stanju q_i nas ničla privede do stanja $q_{2i \bmod 8}$, enica pa do stanja $q_{(2i+1) \bmod 8}$.

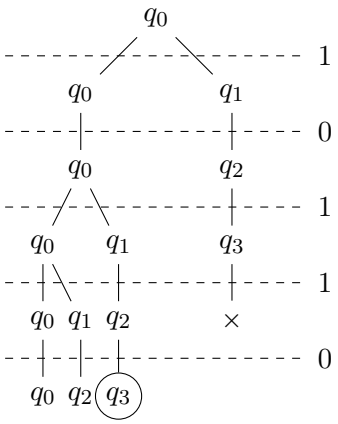


   Avtomat je pravzaprav podoben tistemu iz naloge 8. V neko končno stanje nas privedejo vsa števila v dvojiškem zapisu, ki so po modulu 8 enaka 4, 5, 6 ali 7.

10. NKA je bistveno manjši od DKA. Ko nam nekdo prišepne, da nam do konca besede ostanejo še trije znaki, moramo zgolj preveriti, ali je naslednji znak enica.

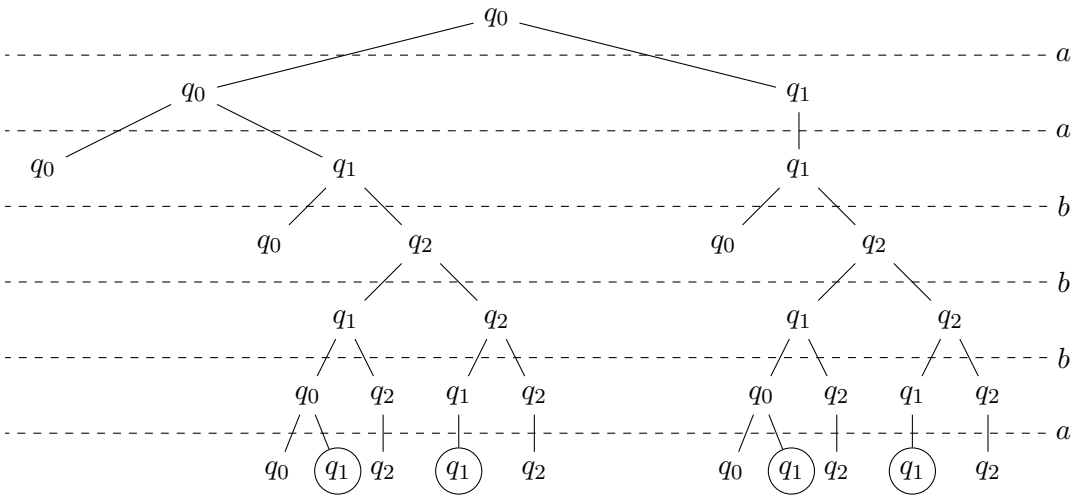


11. Drevo lahko narišemo takole:

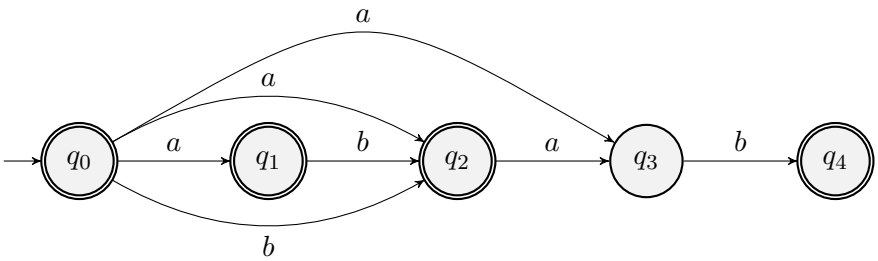


Pri besedi 1011 nas nobena pot po drevesu ne privede do končnega stanja, pri besedi 10110 pa taka pot obstaja. Beseda 1011 potemtakem ne pripada jeziku avtomata, beseda 10110 pa mu pripada.

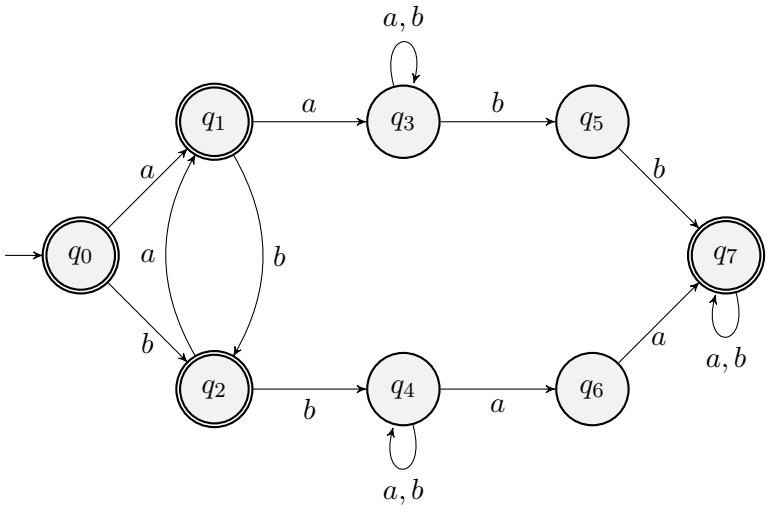
12.



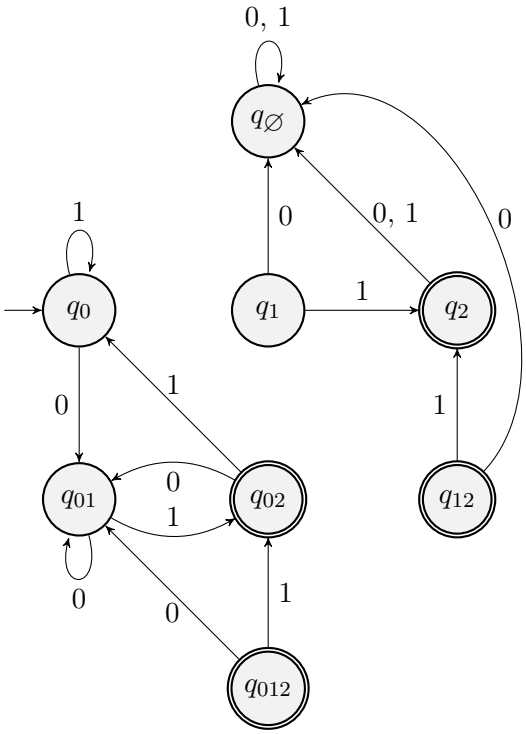
13. Jezik vsebuje zgolj 7 besed: ε, a, b, ab, aab, bab in $abab$.



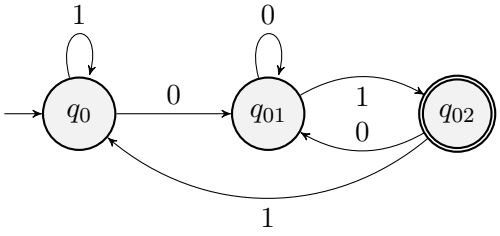
14. Čim preberemo aa (oz. bb), moramo zagotoviti, da beseda vsebuje tudi bb (oz. aa). Med pojavitvama nizov aa in bb je lahko karkoli, prav tako pa se lahko beseda zaključi s čimerkoli.



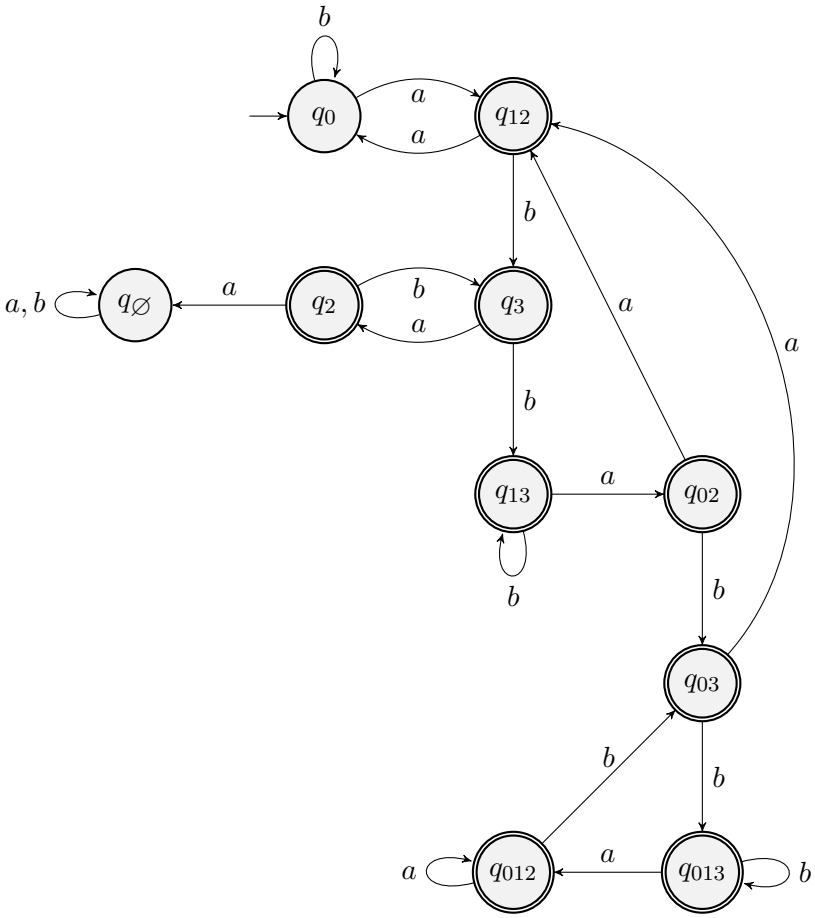
15. DKA po definiciji:



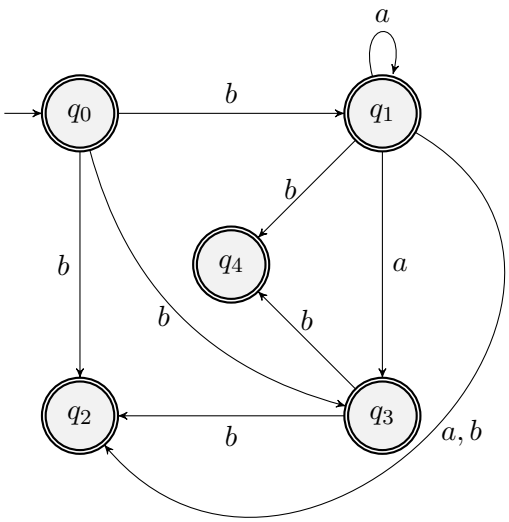
DKA brez nedosegljivih stanj:



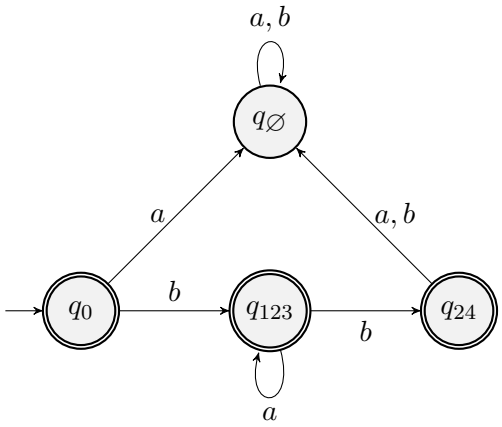
16.


q_0
q_{12}
$q_\varnothing$
q_2
q_3
q_{13}
q_{02}
q_{03}
q_{012}
q_{013}
a, b
a
b

17. NKA:


q_0
q_1
q_2
q_3
q_4
a
b
a, b

DKA:



3 Regularni izrazi

1. (a) 0

   (b) 1,
   ε, 0, 00, 000, ...

   (c) ε, 101, 101101, 101101101, ...,
   01, 0110, 011010, 01101010, ...

   (d) 01,
   ε,
   0, 01,
   00, 001, 010, 0101,
   000, 0001, 0010, 0100, 00101, 01001, 01010, 010101, ...

2. $0^* + 1^*0^* = 1^*0^*$

3. (a) $(a + b)^*b(\varepsilon + a) + \varepsilon + a$

   (b) Besedo razbijemo na odseke ab^*a. Odseki so med seboj ločeni z zaporedji b^*. Poljubno dolgo zaporedje b-jev se nahaja tudi pred prvim in za zadnjim odsekom. Opisani razmislek nas pripelje do sledečega izraza:

   $$b^*(ab^*ab^*)^*$$

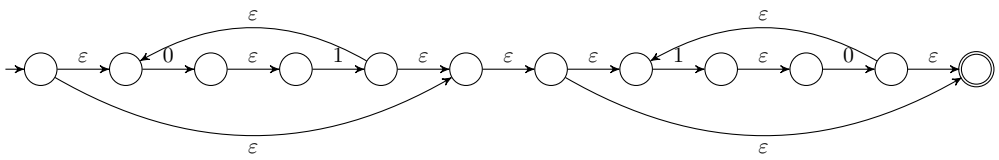
   (c) Izhajamo iz rešitve prejšnje podnaloge:

   $$b^*ab^*(ab^*ab^*)^*$$

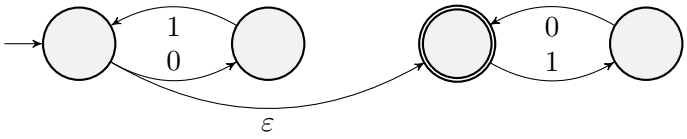
   (d) Potujemo po besedi in brišemo podnize aa in bb, dokler ne naletimo na podniz ab ali ba. Tega ohranimo, nato pa ponavljamo opisani postopek do konca besede. Na koncu nam ostanejo kvečjemu podnizi ab in ba. Teh mora biti sodo mnogo. To pomeni, da si lahko besedo predstavljamo kot zaporedje parov nizov ab in ba, ki so med seboj ločeni s poljubno mnogo nizi aa in bb. Poljubno mnogo takih nizov je lahko tudi na začetku in na koncu zaporedja. Opisani razmislek nas pripelje do sledečega izraza:

   $$((ab + ba)(aa + bb)^*(ab + ba) + aa + bb)^*$$

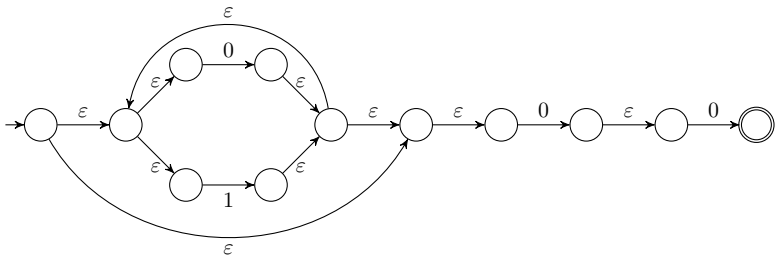
4. Po pravilih:



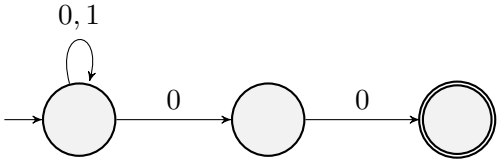
Poenostavljen:



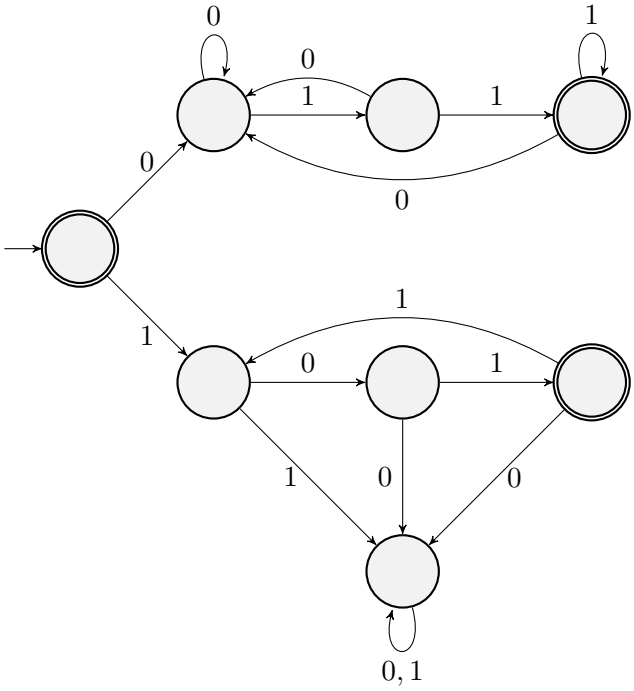
5. Po pravilih:



Poenostavljen:



6. (a)



(b) 2^{12} možnosti pri podizrazu $0(0+1)^*11$ in beseda $(101)^5 \implies 2^{12}+1 = 4097$.

(c) Vse besede dolžine 15 minus tiste v jeziku $= 2^{15} - (2^{12}+1) = 28671$.

Lahko izračunamo tudi takole: komplement vsebuje vse besede dolžine 15, ki se začnejo z 1, razen $(101)^5$ ($2^{14}-1$ besed) in vse besede dolžine 15, ki se začnejo

z 0 in se ne končajo z 00 ($3 \cdot 2^{12}$ besed). Skupno število besed torej znaša $2^{14} - 1 + 3 \cdot 2^{12} = 28671$.

(d)

$$2^n - \begin{cases} 1 & \text{pri } n = 0; \\ 0 & \text{pri } n \in \{1, 2\}; \\ 2^{n-3} + 1 & \text{pri } n \geqslant 3 \text{ in } n \bmod 3 = 0; \\ 2^{n-3} & \text{pri } n \geqslant 3 \text{ in } n \bmod 3 \neq 0. \end{cases}$$

4 Lema o napihovanju za regularne jezike

1. Obravnavamo besedo $z = 0^n 1^n \in L$, kjer je n domnevna konstanta iz leme o napihovanju. Naj bo $z = uvw$ veljavna delitev besede z. Ker mora veljati $|uv| \leqslant n$ in $|v| \geqslant 1$, se morata tako u kot v nahajati znotraj zaporedja ničel. Vsaka veljavna delitev besede je torej oblike $u = 0^p$, $v = 0^q$, $w = 0^{n-p-q}1^n$, pri čemer je $q \geqslant 1$ in $p + q \leqslant n$. Naj bo $z'(i) = uv^i w$. Izračunajmo $z'(0)$:

   $$z'(0) = uv^0 w = uw = 0^p 0^{n-p-q} 1^n = 0^{n-q} 1^n.$$

   Ker je $q \geqslant 1$, velja $z'(0) \notin L$. Ker lahko za *vsako* veljavno delitev besede z najdemo tak i, da bo $z'(i) \notin L$ (dejansko lahko v vseh primerih vzamemo katerikoli $i \neq 1$), zaključimo, da jezik L ni regularen.

2. Vzamemo besedo $z = a^n b^{n+1}$. Vse veljavne delitve so oblike $u = a^p$, $v = a^q$, $w = a^{n-p-q} b^{n+1}$, kjer je $q \geqslant 1$ in $p + q \leqslant n$. Izračunajmo $z'(2)$:

   $$z'(2) = uv^2 w = a^{p+2q+n-p-q} b^{n+1} = a^{n+q} b^{n+1}.$$

   Iz $q \geqslant 1$ sledi $z'(2) \notin L$.

3. Vzamemo besedo $z = a^{n+1} b^n$. Vse veljavne delitve so oblike $u = a^p$, $v = a^q$, $w = a^{n+1-p-q} b^n$, kjer je $q \geqslant 1$ in $p + q \leqslant n$. Izračunajmo $z'(0)$:

   $$z'(0) = uw = a^{p+n+1-p-q} b^n = a^{n+1-q} b^n.$$

   Iz $q \geqslant 1$ sledi $z'(0) \notin L$.

4. Vzamemo $z = 0^n 1^n 1^n 0^n = 0^n 1^{2n} 0^n$ in $u = 0^p$, $v = 0^q$, $w = 0^{n-p-q} 1^{2n} 0^n$ (pri $q \geqslant 1$ in $p + q \leqslant n$). Izračunajmo $z'(0)$:

   $$z'(0) = 0^p 0^{n-p-q} 1^{2n} 0^n = 0^{n-q} 1^{2n} 0^n.$$

   Ker je $q \geqslant 1$, je $z'(0) \notin L$.

5. Obravnavamo besedo $z = 1^p$, kjer je p praštevilo z lastnostjo $p \geqslant n$. Vse veljavne delitve so oblike $u = 1^a$, $v = 1^b$, $w = 1^{p-a-b}$, pri čemer je $b \geqslant 1$ in $a + b \leqslant n$. Izračunajmo $z'(p + 1)$:

   $$z'(p+1) = uv^{p+1} w = 1^a 1^{b(p+1)} 1^{p-a-b} = 1^{p+bp} = 1^{p(1+b)}.$$

   Ker je p praštevilo, je $p \geqslant 2$. Ker je $1 \leqslant b \leqslant n$, je $1 + b \geqslant 2$. Od tod sledi, da je število $p(1 + b)$ sestavljeno. Za vsako delitev besede z je torej $z'(p + 1) \notin L$.

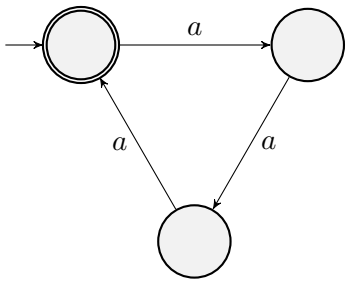
6. Vzamemo $z = a^{n^3}$ in $v = a^q$, pri čemer je $1 \leqslant q \leqslant n$. Velja $z'(2) = a^{n^3+q}$. Vrednost izraza $n^3 + q$ je potemtakem med $n^3 + 1$ in $n^3 + n$, torej strogo med n^3 in $(n + 1)^3 = n^3 + 3n^2 + 3n + 1$. Zatorej $z'(2) \notin L$.

7. Tale je nekoliko težja. Na primer, ne moremo vzeti $z = a^n b^{n+1}$ ali $z = a^n b^{2n}$, saj nam ne bo pri *vsaki* veljavni delitvi besede z uspelo najti eksponenta, ki bi število a-jev izenačil s številom b-jem. Problem pa uženemo, če vzamemo $z = a^n b^{n+n!}$. Besedo lahko razdelimo le kot $u = a^p$, $v = a^q$, $w = a^{n-p-q} b^{n+n!}$, pri čemer je $q \geqslant 1$ in $p + q \leqslant n$. Sedaj izračunamo $z'(\frac{n!}{q} + 1)$ (ker je $1 \leqslant q \leqslant n$, velja $q \mid n!$):

$$z'(\frac{n!}{q} + 1) = a^{p+n!+q+n-p-q} b^{n+n!} = a^{n+n!} b^{n+n!} \notin L.$$

Če se želimo naloge lotiti bolj sistematično, pričnemo z besedo $z = a^n b^{f(n)}$ in poiščemo neznano funkcijo f. Na podlagi delitve $u = a^p$, $v = a^q$, $w = a^{n-p-q} b^{f(n)}$ ($q \geqslant 1$, $p + q \leqslant n$) dobimo $z'(i) = a^{n+q(i-1)} b^{f(n)}$. Poiskati moramo tak i, da bo $n + q(i-1) = f(n)$. Od tod sledi $i = 1 + (f(n) - n) / q$. Ker mora biti i celo število in ker je q lahko karkoli med 1 in n, mora biti izraz $f(n) - n$ deljiv z vsemi števili med 1 in n. Torej je $f(n) - n = n!$ oziroma $f(n) = n! + n$.

8. Jezik je regularen, ker lahko zanj zapišemo regularni izraz $((aaa)^*)$ ali končni avtomat:



Uporabimo lemo o napihovanju na besedi $z = a^{3n}$. Pri delitvi $u = \varepsilon$, $v = 3$, $w = 3n-3$ za *vsak* $i \geqslant 0$ velja

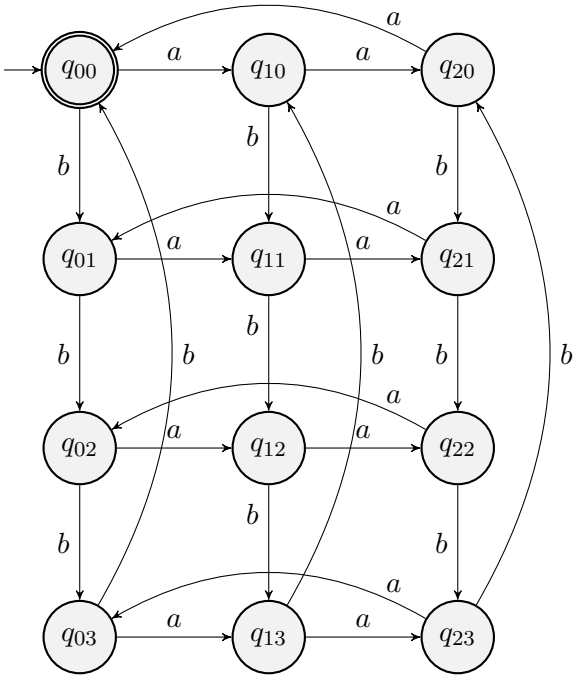
$$z'(i) = uv^i w = a^{3i+3n-3} = a^{3(n+i-1)} \in L.$$

Torej smo za vsako dovolj dolgo besedo našli delitev, ki nas pri vseh »napihovalnih« eksponentih ohranja v jeziku. Seveda pa to *ni* dokaz, da je jezik regularen!

9. Iskani n je enak 3. Pri $n = 2$ imamo namreč besedo ba, ki je ni mogoče »napihniti«. Pri $n = 3$ pa za vsako besedo dolžine n ali več velja, da med prvimi n znaki vsebuje podniz v a^* ali v $(ba)^*$, ki ga lahko »napihnemo«.

Na primer, besedo aba razdelimo kot $u = \varepsilon$, $v = a$, $w = ba$ (velja $|uv| \leqslant n$ in $|v| \geqslant 1$), besedo $bbaa$ pa kot $u = b$, $v = ba$, $w = a$ (zopet velja $|uv| \leqslant n$ in $|v| \geqslant 1$). V obeh primerih lahko podniz v poljubno »napihujemo«.

10. Jezik je regularen, saj ga sprejema končni avtomat. V sledečem DKA se v stanju q_{ij} nahajamo natanko v primeru, če smo do tistega trenutka prebrali $(3k + i)$ simbolov a in $(4\ell + j)$ simbolov b (za nek $k \geqslant 0$ in $\ell \geqslant 0$).



Poleg tega je jezik presek dveh enostavnejših regularnih jezikov nad $\Sigma = \{a, b\}$:

$$L = \{w \in \Sigma^* \mid \#_a(w) \bmod 3 = 0\} \cap \{w \in \Sigma^* \mid \#_b(w) \bmod 4 = 0\}$$

Gornji avtomat je dejansko produktni avtomat, zgrajen na podlagi avtomata za prvi jezik in avtomata za drugi jezik.

5 Kontekstno neodvisne gramatike

1. Jezik je stik dveh podjezikov:

$$S \to AB$$

Jezik, ki ga tvori spremenljivka A, je unija dveh podjezikov:

$$A \to C \mid D$$
$$C \to 0$$
$$D \to D1 \mid \varepsilon$$

Jezik, ki ga tvori spremenljivka B, je prav tako unija:

$$B \to 01 \mid \varepsilon$$

2. Jezik je Kleenova ovojnica:

$$S \to SA \mid \varepsilon$$

Jezik, ki ga tvori spremenljivka A, je unija:

$$A \to 1B \mid 00$$

Jezik, ki ga tvori spremenljivka B, je Kleenova ovojnica:

$$B \to B01 \mid \varepsilon$$

3. Tale ni težka:
$$S \to aSc \mid b$$

4. Palindrom nad abecedo $\Sigma = \{a, b\}$ lahko zgradimo tako, da pričnemo z besedo ε, a ali b, nato pa v vsakem koraku na obeh straneh dodamo bodisi a bodisi b:
$$S \to aSa \mid bSb \mid \varepsilon \mid a \mid b$$
Ni težko preveriti, da ta gramatika ne generira nič drugega kot palindrome: besede ε, a in b so palindromi, produkciji $S \to aSa$ in $S \to bSb$ pa palindromskost ohranjata.
Sedaj pa pokažimo, da gramatika generira *vse* palindrome. Za začetek se omejimo na palidrome sode dolžine. Z indukcijo bomo pokazali, da lahko palindrom dolžine $2n$ pridobimo z izpeljavo dolžine $n + 1$. Za $n = 0$ trditev očitno velja: palindrom ε pridobimo z izpeljavo $S \Rightarrow \varepsilon$. Sedaj pa predpostavimo, da za poljuben palindrom w dolžine $2n$ obstaja izpeljava $S \Rightarrow^* w$ dolžine $n + 1$, in obravnavajmo palindrom w' dolžine $2(n + 1)$. Ta palindrom je oblike $aw''a$ ali $bw''b$, kjer je w'' palindrom dolžine $2n$. V prvem primeru lahko palindrom w' z uporabo induktivne predpostavke pridobimo kot $S \Rightarrow aSa \Rightarrow^* aw''a$, v drugem pa kot $S \Rightarrow bSb \Rightarrow^* bw''b$. V obeh primerih smo palidrom dolžine $2(n + 1)$ pridobili z izpeljavo dolžine $n + 2$.
Na podoben način dokažemo, da gramatika generira tudi vse palindrome lihe dolžine.

5. Oklepajni izraz je sestavljen iz zaporedja poljubno mnogo zaključenih oklepajnih izrazov. Na primer, oklepajni izraz (()())(())((())()) je sestavljen iz zaključenih oklepajnih izrazov (()()), (()) in ((())()). Denimo, da vsak zaključeni oklepajni izraz generiramo s simbolom A. Ker bi radi tvorili poljubno mnogo takih izrazov, pričnemo takole:
$$S \to SA \mid \varepsilon$$
Zaključen oklepajni izraz se prične s predklepajem in konča z zaklepajem, vmes pa je poljuben oklepajni izraz:
$$A \to (S)$$
Simbola A se lahko tudi znebimo in gramatiko zapišemo takole:
$$S \to S(S) \mid \varepsilon$$

6. Jezik je unija štirih enostavnih jezikov:
   - $i < j$, k je poljuben
   - $i > j$, k je poljuben
   - i je poljuben, $j < k$
   - i je poljuben, $j > k$

   Ko se tega zavemo, gre vse kot po maslu:
$$\begin{aligned} S &\to XC \mid YC \mid AZ \mid AW \\ X &\to b \mid Xb \mid aXb \\ Y &\to a \mid aY \mid aYb \\ Z &\to c \mid Zc \mid bZc \\ W &\to b \mid bW \mid bWc \\ A &\to Aa \mid \varepsilon \\ C &\to Cc \mid \varepsilon \end{aligned}$$

7. Niz sestavimo kot zaporedje $AB \ldots$

$$S \to AB$$

$\ldots$ pri čemer se simbol A razvije v podniz 0^i, simbol B pa v podniz 1^j0^k. V podniz A (torej v podniz, ki se razvije iz simbola A) lahko dodamo poljubno mnogo ničel, v podniz B pa poljubno mnogo enic:

$$A \to 0A \mid \varepsilon$$
$$B \to 1B \mid \varepsilon$$

Da zagotovimo $i + j \geqslant 2k$, v podniz B dodamo ničlo samo takrat, ko vanj obenem dodamo še dve enici ali pa ko v podniz A (oziroma na začetek celotnega niza) dodamo še dve ničli:

$$B \to 11B0$$
$$S \to 00S0$$

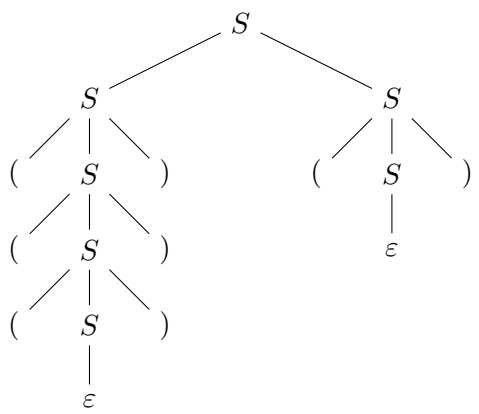
Gramatika generira samo pravilne besede, saj vsa pravila ohranjajo lastnosti jezika. Toda — ali gramatika generira *vse* veljavne besede? Poskusimo takole: če imamo besedo oblike $0^i1^j0^k$, pri čemer je $i + j \geqslant 2k$, potem najprej med podniza 0^i in 1^j vstavimo simbol A (z obratno uporabo produkcije $A \to \varepsilon$), med podniza 1^j in 0^k pa simbol B. Sedaj niz z obratnimi uporabami produkcij $A \to 0A$ in $B \to 1B$ pretvorimo v obliko $0^pA1^qB0^r$, pri čemer je $p + q = 2r$. Če je število q (in s tem tudi p) sodo, z obratnimi uporabami produkcij $B \to 11B0$ odstranimo vse enice, nato pa niz AB zamenjamo s simbolom S in s pomočjo produkcije $S \to 00S0$ odstranimo še vse ničle. Če je število q liho, pa nam bo po odstranitvi vseh parov enic ostal podniz $0^pA1B0^{(p+1)/2}$, v katerem bi — da bi se lahko znebili končnih ničel — morali podniz $0A1B0$ zamenjati s simbolom S. Ker tega ne moremo storiti, v gramatiko dodamo pripadajočo produkcijo:

$$S \to 0A1B0$$

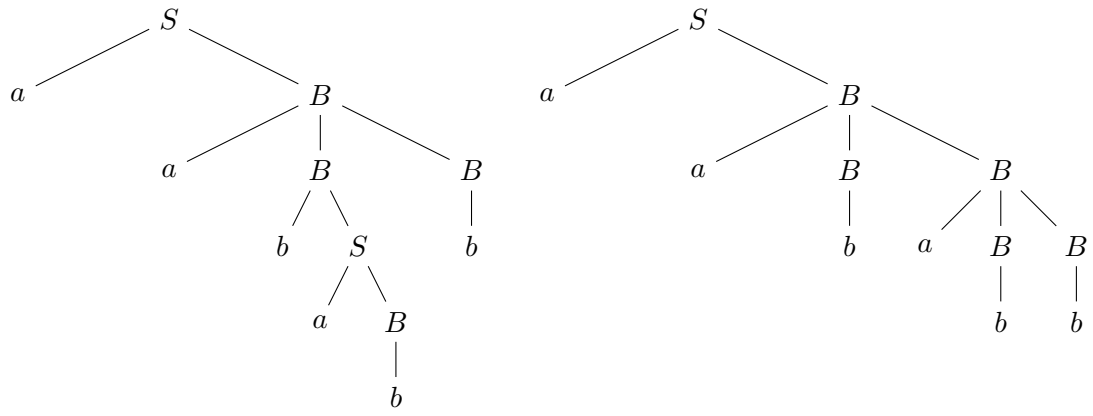
Rešitev naloge je torej sledeča gramatika:

$$S \to AB \mid 0A1B0 \mid 00S0$$
$$A \to 0A \mid \varepsilon$$
$$B \to 1B \mid 11B0 \mid \varepsilon$$

8. (a) $S \Rightarrow SS \Rightarrow (S)S \Rightarrow ((S))S \Rightarrow (((S)))S \Rightarrow ((()))S \Rightarrow ((()))(S) \Rightarrow ((()))()$
   (b) $S \Rightarrow SS \Rightarrow S(S) \Rightarrow S() \Rightarrow (S)() \Rightarrow ((S))() \Rightarrow (((S)))() \Rightarrow ((()))()$
   (c)



9. Eden od nizov z dvema različnima drevesoma izpeljav je *aababb*:



Mimogrede, gramatika tvori nize z enakim številom a-jev in b-jev. Simbol A se razvije v nize, pri katerih je število a-jev za 1 večje od števila B-jev, simbol B pa v nize, pri katerih je število b-jev za 1 večje od števila a-jev.

10. Sledeča gramatika je enakovredna, a nedvoumna:

$$E \to E + T \mid T$$
$$T \to T \cdot F \mid F$$
$$F \to 0 \mid \ldots \mid 9 \mid (E)$$

Naj besedna zveza *prosti simbol* označuje simbol, ki nastopa izven oklepajev. Pokažimo sledeče trditve:

(1) Vsaka beseda w z lastnostjo $E \Rightarrow^* w$ ima natanko eno drevo izpeljav.

(2) Vsaka beseda w z lastnostjo $T \Rightarrow^* w$ ima natanko eno drevo izpeljav. Poleg tega velja, da beseda w ne vsebuje prostega simbola $+$.

(3) Vsaka beseda w z lastnostjo $F \Rightarrow^* w$ ima natanko eno drevo izpeljav. Poleg tega velja, da beseda w ne vsebuje niti prostega simbola $+$ niti prostega simbola $\cdot$.

Očitno je, da iz simbola F ne moremo pridelati besede s prostim simbolom $+$ ali $\cdot$, iz simbola T pa ne moremo pridelati besede s prostim simbolom $+$. Trditev, da ima beseda, ki jo tvori spremenljivka $X \in \{E, T, F\}$, natanko eno drevo izpeljav, pa dokažemo z indukcijo po dolžini besede w. Če ima beseda $w \in L$ dolžino 1, velja $w \in \{0, 1, \ldots, 9\}$, zato jo lahko pridelamo le po poti $E \Rightarrow T \Rightarrow F \Rightarrow w$. Sedaj pa predpostavimo, da vse tri trditve veljajo za vse veljavne besede, krajše od n. Naj bo $w \in L$ beseda dolžine n. Ločimo lahko zgolj sledeče primere:

- Če velja $w = (w')$, kjer je $w' \in L$, lahko w pridobimo le po poti $E \Rightarrow T \Rightarrow F \Rightarrow (E) \Rightarrow^* (w')$. Ker je pot $E \Rightarrow^* w'$ po induktivni predpostavki enolično določena, je takšna tudi pot do besede w.
- Če beseda w vsebuje prosti simbol $+$, jo zapišemo kot $w = w_1 + w_2$, kjer w_2 ne vsebuje prostega simbola $+$. Besedo w lahko potem pridelamo kot $E \Rightarrow E + T \Rightarrow^* w_1 + w_2$. Če bi beseda w_2 vsebovala prosti simbol $+$, je ne bi mogli pridelati iz simbola T, zato je skrajno desni prosti simbol $+$ edina možna delilna točka takšne besede w.

- Če beseda w vsebuje prosti simbol $\cdot$, prostega simbola $+$ pa ne vsebuje, jo zapišemo kot $w = w_1 \cdot w_2$, kjer w_2 ne vsebuje prostega simbola $\cdot$. V tem primeru lahko beseda w nastane le po poti $E \Rightarrow T \Rightarrow T \cdot F \Rightarrow^* w_1 \cdot w_2$. Podobno kot v prejšnji alineji lahko besedo w izpeljemo le v primeru, če kot delilno točko vzamemo zadnji prosti simbol $\cdot$.
- Če beseda w ne ustreza pogojem iz nobene od gornjih treh alinej, je lahko le števka. V tem primeru jo lahko izpeljemo le po poti $E \Rightarrow T \Rightarrow F \Rightarrow w$.

11. Za vsako pravilo za gradnjo regularnih izrazov moramo poiskati pripadajoče pravilo za gradnjo enakovrednih gramatik:

    - Regularnemu izrazu $\varnothing$ ustreza gramatika brez produkcij ali gramatika s produkcijo $S \to S$.
    - Regularnemu izrazu ε ustreza gramatika s produkcijo $S \to \varepsilon$.
    - Regularnemu izrazu a za nek $a \in \Sigma$ ustreza gramatika s produkcijo $S \to a$.
    - Naj regularnima izrazoma R_1 in R_2 ustrezata gramatiki z začetnima simboloma S_1 in S_2. Potem
        - regularnemu izrazu $R_1 + R_2$ ustreza gramatika $S \to S_1 \mid S_2$;
        - regularnemu izrazu $R_1 R_2$ ustreza gramatika $S \to S_1 S_2$;
        - regularnemu izrazu R_1^* ustreza gramatika $S \to S S_1 \mid \varepsilon$.

12. Naj bo $\Sigma = \{[,], 0, 1, e, n, p, k, z\}$, pri čemer smo simbole, ki lahko nastopajo v regularnih izrazih, prekrstili takole: $(\mapsto [$, $) \mapsto]$, $\varepsilon \mapsto e$, $\varnothing \mapsto n$, $+ \mapsto p$, $\cdot \mapsto k$, $* \mapsto z$. Sledeča gramatika tvori jezik regularnih izrazov:

    $$E \to EpE \mid EkE \mid Ez \mid [E] \mid 0 \mid 1 \mid e \mid n$$

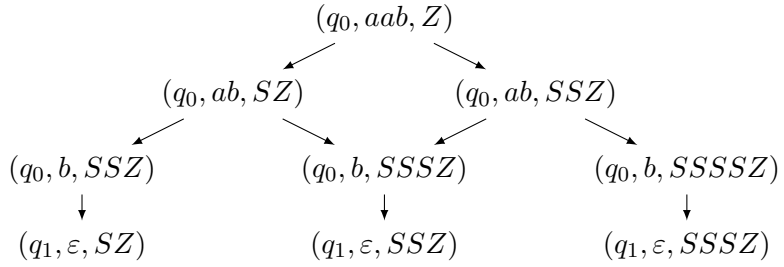
    Jezik regularnih izrazov ni regularen: vzamemo $z = [^n 0]^n$ in uporabimo lemo o napihovanju.

    Jezik regularnih izrazov brez oklepajev je regularen. Regularni izraz zanj je

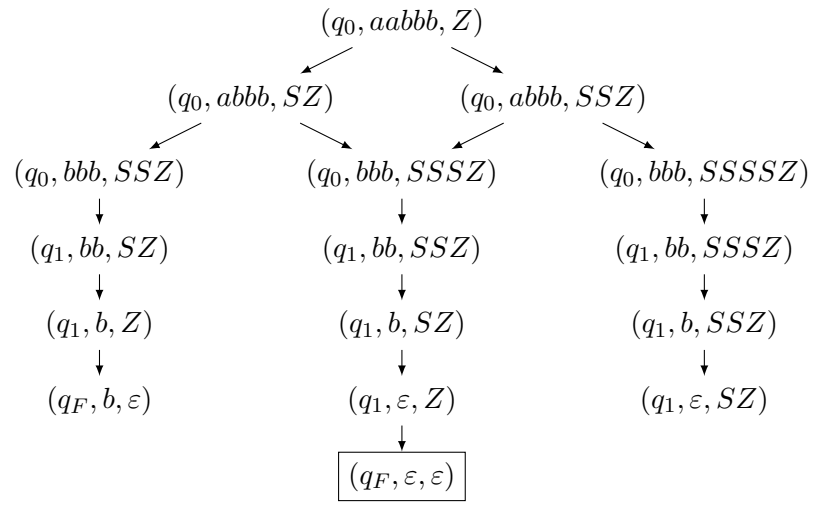
    $$(0 + 1 + e + n)z^*((p + k)(0 + 1 + e + n)z^*)^*$$

6 Skladovni avtomati

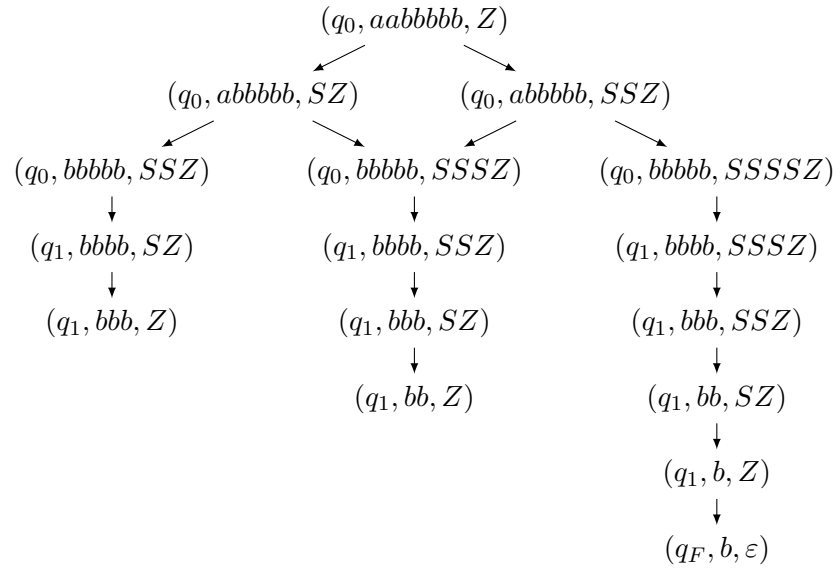
1. (1) $\underline{w = aab}$



$\underline{w = aabbb}$



$\underline{w = aabbbbb}$



(2) Skladovni avtomat sprejema jezik $\{a^m b^n \mid 1 \leqslant m \leqslant n \leqslant 2m\}$.

(3) Enakovredno kontekstno neodvisno gramatiko lahko zapišemo takole:

$$S \to aSb \mid aSbb \mid ab \mid abb$$

2. Ker bi ta jezik lahko razpoznali tudi s končnim avtomatom, sklada sploh ne potrebujemo. Zadoščajo štiri stanja:

   - v začetnem stanju (q_0) vztrajamo, dokler ne preberemo prvega simbola a;
   - ko preberemo prvi simbol a, preidemo v stanje q_1;
   - po drugem simbolu a preidemo v stanje q_2;
   - po tretjem simbolu a preidemo v stanje q_F (končno stanje).

   Naš avtomat definiramo kot $(\{q_0, q_1, q_2, q_F\}, \{a, b\}, \{Z\}, \delta, q_0, Z, \{q_F\})$, pri čemer je funkcija δ določena takole:

$$\delta(q_0, a, Z) = \{(q_1, Z)\}$$
$$\delta(q_1, a, Z) = \{(q_2, Z)\}$$

$$\delta(q_2, a, Z) = \{(q_F, Z)\}$$
$$\delta(q_0, b, Z) = \{(q_0, Z)\}$$
$$\delta(q_1, b, Z) = \{(q_1, Z)\}$$
$$\delta(q_2, b, Z) = \{(q_2, Z)\}$$
$$\delta(q_F, b, Z) = \{(q_F, Z)\}$$

3. Avtomat si pomaga s skladom in skladovnima simboloma Z (začetni simbol) in S. Dokler bere simbole a, vztraja v stanju q_0 in hkrati na sklad dodaja simbole S (po en S za vsak prebrani a). Ko prebere prvi b, se prestavi v stanje q_1 in s sklada pobere vrhnji S. Nato ostaja v stanju q_1 in s sklada pobira vrhnje simbole S, dokler bere b-je. Če po branju celotnega vhoda na skladu ostane samo simbol Z, vemo, da je vhod vseboval natanko toliko b-jev kot a-jev.

   Stanje q_0 je začetno, stanje q_F pa končno. Funkcija prehodov je definirana takole:

$$\delta(q_0, a, Z) = \{(q_0, SZ)\}$$
$$\delta(q_0, a, S) = \{(q_0, SS)\}$$
$$\delta(q_0, b, S) = \{(q_1, \varepsilon)\}$$
$$\delta(q_1, b, S) = \{(q_1, \varepsilon)\}$$
$$\delta(q_1, \varepsilon, Z) = \{(q_F, \varepsilon)\}$$

   Ker moramo sprejeti tudi besedo ε, dodamo še naslednji prehod:

$$\delta(q_0, \varepsilon, Z) = \{(q_F, \varepsilon)\}$$

   Prepričajmo se, da avtomat sprejema besede oblike $a^n b^n$ in nobenih drugih. Če kakšnemu b-ju sledi a, bomo obstali v stanju q_1, saj v tem stanju nimamo prehoda za simbol a. (Če smo s sklada pravkar pobrali vse simbole S, bi se lahko po ε-prehodu prestavili v stanje q_F, vendar pa bi del vhoda ostal neprebran.) To pomeni, da mora biti vhod oblike $a^p b^q$. Če je $p > q$, nam bo po branju celotnega vhoda na skladu ostal vsaj en S (to je razlog, zakaj poleg simbola Z potrebujemo še en skladovni simbol), v primeru $p < q$ pa bomo sklad sicer izpraznili, vendar pa bo del vhoda ostal neprebran. Avtomat bo torej sprejemal le besede, za katere velja $p = q$.

4. Avtomat iz prejšnje naloge ni deterministično, ker sta množici $\delta(q_0, a, Z)$ in $\delta(q_0, \varepsilon, Z)$ obe neprazni. To pomeni, da lahko v stanju q_0, simbolu a na vhodu in simbolu Z na vrhu sklada izberemo dva različna prehoda. Pomagamo si z dodatnim stanjem:

$$\delta(q_0, \varepsilon, Z) = (q_F, Z)$$
$$\delta(q_F, a, Z) = (q_1, SZ)$$
$$\delta(q_1, a, S) = (q_1, SS)$$
$$\delta(q_1, b, S) = (q_2, \varepsilon)$$
$$\delta(q_2, b, S) = (q_2, \varepsilon)$$
$$\delta(q_2, \varepsilon, Z) = (q_F, \varepsilon)$$

5. Ko beremo a-je, smo v stanju q_0 in na sklad vsakokrat dodamo po en simbol S. Ko beremo b-je, smo v stanju q_1 ali q_2 in s sklada vsakokrat odvzamemo po dva simbola S. Ker ne moremo v istem koraku s sklada pobrati več kot enega simbola, si pomagamo z dodatnim stanjem (q_2). Če je beseda res oblike $a^{2n} b^n$, bo na skladu na koncu ostal samo simbol Z.

$$\delta(q_0, a, Z) = \{(q_0, SZ)\}$$

$$\delta(q_0, a, S) = \{(q_0, SS)\}$$
$$\delta(q_0, b, S) = \{(q_1, \varepsilon)\}$$
$$\delta(q_1, \varepsilon, S) = \{(q_2, \varepsilon)\}$$
$$\delta(q_2, b, S) = \{(q_1, \varepsilon)\}$$
$$\delta(q_2, \varepsilon, Z) = \{(q_F, \varepsilon)\}$$
$$\delta(q_0, \varepsilon, Z) = \{(q_F, \varepsilon)\}$$

Zadnji prehod potrebujemo zato, ker moramo sprejeti tudi prazen niz.

6. Ideja »ko vidiš a, dodaj simbol na sklad, ko vidiš b, pa ga odstrani« deluje samo za besede, pri katerih za vsako predpono velja, da je število a-jev večje ali enako številu b-jev. Lahko pa si pomagamo z dvema skladovnima simboloma (poleg začetnega Z-ja):

   - na skladu so vsakem trenutku poleg začetnega Z-ja bodisi zgolj A-ji bodisi zgolj B-ji;
   - vsak simbol A predstavlja presežek enega a-ja v doslej prebranem vhodu (če imamo na skladu A^kZ, je število doslej prebranih a-jev za k večje od števila doslej prebranih b-jev);
   - vsak simbol B predstavlja presežek enega b-ja v doslej prebranem vhodu (niz B^kZ na skladu nam pove, da smo doslej prebrali k b-jev več kot a-jev).

   Če je na skladu samo simbol Z, je število doslej prebranih a-jev torej enako številu doslej prebranih b-jev.

   Z upoštevanjem gornjih pravil lahko avtomat zapišemo takole:

$$\delta(q_0, a, Z) = \{(q_0, AZ)\}$$
$$\delta(q_0, b, Z) = \{(q_0, BZ)\}$$
$$\delta(q_0, a, A) = \{(q_0, AA)\}$$
$$\delta(q_0, a, B) = \{(q_0, \varepsilon)\}$$
$$\delta(q_0, b, A) = \{(q_0, \varepsilon)\}$$
$$\delta(q_0, b, B) = \{(q_0, BB)\}$$
$$\delta(q_0, \varepsilon, Z) = \{(q_F, \varepsilon)\}$$

7. V stanju q_0 beremo a-je, v stanju q_1 b-je, v stanju q_2 pa c-je. Za vsak prebrani a in za vsak prebrani b dodamo na sklad po en simbol, za vsak prebrani c pa enega poberemo. Če je beseda pravilne oblike, bo na skladu na koncu ostal samo simbol Z.

$$\delta(q_0, a, Z) = \{(q_0, SZ)\}$$
$$\delta(q_0, a, S) = \{(q_0, SS)\}$$
$$\delta(q_0, b, S) = \{(q_1, SS)\}$$
$$\delta(q_1, b, S) = \{(q_1, SS)\}$$
$$\delta(q_1, c, S) = \{(q_2, \varepsilon)\}$$
$$\delta(q_2, c, S) = \{(q_2, \varepsilon)\}$$
$$\delta(q_2, \varepsilon, Z) = \{(q_F, \varepsilon)\}$$

   Upoštevati moramo tudi možnost, da katera od skupin simbolov manjka. Pri pravilnem vhodu lahko tako na začetku namesto simbola a vidimo simbol b (ne pa simbola

c), zaporedju a-jev pa lahko neposredno sledi simbol c. Jeziku pripada tudi beseda ε.

$$\begin{aligned}\delta(q_0, b, Z) &= \{(q_1, SZ)\}\\ \delta(q_0, c, S) &= \{(q_2, \varepsilon)\}\\ \delta(q_0, \varepsilon, Z) &= \{(q_F, \varepsilon)\}\end{aligned}$$

8. V stanju q_0 beremo a-je in vsakokrat dodamo po en simbol na sklad. Ko preberemo prvi b, preidemo v stanje q_1. V tem stanju najprej med branjem b-jev pobiramo simbole s sklada, ko nam na skladu ostane samo Z, pa preidemo v stanje q_2 in simbole na sklad vnovič pričnemo dodajati. Ko preberemo prvi c, preidemo v stanje q_3. V tem stanju simbole s sklada zopet pobiramo. Sklad se na koncu izprazni samo v primeru, če je $i + k = j$.

$$\begin{aligned}\delta(q_0, a, Z) &= \{(q_0, SZ)\}\\ \delta(q_0, a, S) &= \{(q_0, SS)\}\\ \delta(q_0, b, S) &= \{(q_1, \varepsilon)\}\\ \delta(q_1, b, S) &= \{(q_1, \varepsilon)\}\\ \delta(q_1, b, Z) &= \{(q_2, SZ)\}\\ \delta(q_2, b, S) &= \{(q_2, SS)\}\\ \delta(q_2, c, S) &= \{(q_3, \varepsilon)\}\\ \delta(q_3, c, S) &= \{(q_3, \varepsilon)\}\\ \delta(q_3, \varepsilon, Z) &= \{(q_F, \varepsilon)\}\end{aligned}$$

Ali morda sprejemamo besede oblike $a^i b^j c^k$ pri $i = j + k$? Ne, ker se sklad v stanju q_1 ne izprazni, zato ne preidemo v stanje q_2 in tudi ne v q_3.

Upoštevati moramo tudi možnost, da c-jev sploh nimamo. Poleg tega se beseda lahko začne z b-jem, lahko pa je tudi prazna:

$$\begin{aligned}\delta(q_1, \varepsilon, Z) &= \{(q_F, \varepsilon)\}\\ \delta(q_0, b, Z) &= \{(q_2, SZ)\}\\ \delta(q_0, \varepsilon, Z) &= \{(q_F, \varepsilon)\}\end{aligned}$$

9. Ključna ideja je, da »uganemo« sredino besede, pri čemer moramo upoštevati, da je beseda lahko lihe ali sode dolžine. Do prebranih $\lfloor n / 2 \rfloor$ znakov (kjer je n dolžina besede) se nahajamo v stanju q_0 in za vsak prebrani a dodamo na sklad simbol A, za vsak prebrani b pa simbol B. Če je beseda sode dolžine, takoj zatem preidemo v stanje q_1 po ε-prehodu, v primeru besede lihe dolžine pa v stanje q_1 preidemo po branju naslednjega simbola (sklad se v nobenem primeru ne spremeni). V obeh primerih smo pravkar prebrali $\lceil n / 2 \rceil$ znakov, zato pričnemo s preverjanjem palidromskosti. Simbol, ki ga vidimo na vhodu, se mora namreč ujemati s simbolom na vrhu sklada: če vidimo a, mora na vrhu skladu biti A, če vidimo b, pa B.

$$\begin{aligned}\delta(q_0, a, Z) &= \{(q_0, AZ)\}\\ \delta(q_0, b, Z) &= \{(q_0, BZ)\}\\ \delta(q_0, a, A) &= \{(q_0, AA), (q_1, A)\}\\ \delta(q_0, a, B) &= \{(q_0, AB), (q_1, B)\}\\ \delta(q_0, b, A) &= \{(q_0, BA), (q_1, A)\}\\ \delta(q_0, b, B) &= \{(q_0, BB), (q_1, B)\}\end{aligned}$$

$$\begin{aligned}
\delta(q_0, \varepsilon, A) &= \{(q_1, A)\} \\
\delta(q_0, \varepsilon, B) &= \{(q_1, B)\} \\
\delta(q_1, a, A) &= \{(q_1, \varepsilon)\} \\
\delta(q_1, b, B) &= \{(q_1, \varepsilon)\} \\
\delta(q_1, \varepsilon, Z) &= \{(q_F, \varepsilon)\}
\end{aligned}$$

Na primer, pri besedi *abba* bomo po prvem b-ju uporabili prehod $\delta(q_0, \varepsilon, B) \mapsto (q_1, B)$, pri besedi *ababa* pa bomo po prvem b-ju uporabili prehod $\delta(q_0, a, B) \mapsto (q_1, B)$. V obeh primerih pristanemo v stanju q_1, sklad pa vsebuje niz BAZ, kar se ujema s preostankom vhoda (ba).

Palindromi so tudi ε, a in b:

$$\begin{aligned}
\delta(q_0, \varepsilon, Z) &= \{(q_F, \varepsilon)\} \\
\delta(q_0, a, Z) &:= \delta(q_0, a, Z) \cup \{(q_F, \varepsilon)\} \\
\delta(q_0, b, Z) &:= \delta(q_0, b, Z) \cup \{(q_F, \varepsilon)\}
\end{aligned}$$

10. Avtomat ima eno samo stanje in sprejema po kriteriju izpraznitve sklada. Za gramatiko (V, T, P, S) zgradimo avtomat ($\{q\}$, T, $V \cup T$, δ, q, S). Za vsako produkcijo $A \to \alpha$ dodamo prehod $(q, \varepsilon, A) \mapsto (q, \alpha)$, za vsak simbol $a \in \Sigma$ pa dodamo prehod $(q, a, a) \mapsto (q, \varepsilon)$. Funkcija δ je v našem primeru torej definirana takole:

$$\begin{aligned}
\delta(q, \varepsilon, S) &= \{(q, ASA), (q, T)\} \\
\delta(q, \varepsilon, A) &= \{(q, Aa), (q, Ab), (q, \varepsilon)\} \\
\delta(q, \varepsilon, T) &= \{(q, cTd), (q, \varepsilon)\} \\
\delta(q, a, a) &= \{(q, \varepsilon)\} \\
\delta(q, b, b) &= \{(q, \varepsilon)\} \\
\delta(q, c, c) &= \{(q, \varepsilon)\} \\
\delta(q, d, d) &= \{(q, \varepsilon)\}
\end{aligned}$$

7 Lema o napihovanju za kontekstno neodvisne jezike

1. Za vsak $n > 0$ moramo najti tako besedo $z \in L$, ki je dolga vsaj n, da bo za vsako delitev $z = uvwxy$ z lastnostma $|vx| \geqslant 1$ in $|vwx| \leqslant n$ obstajal tak $i \geqslant 0$, da bo $z'(i) = uv^i wx^i y \notin L$.

   Naj bo n konstanta iz leme. Poskusimo z besedo $z = a^n b^n c^n$. Ta beseda je pri vseh n dolga vsaj n in pri vseh n pripada jeziku. Če nam za *vsako* veljavno delitev besede z uspe najti eksponent $i \geqslant 0$, tako da bo $z'(i) \notin L$, bomo dokazali, da jezik L ni kontekstno neodvisen.

   Ker mora veljati $|vwx| \leqslant n$, imamo glede v in x (podnizov, ki ju »napihujemo«) zgolj sledeče možnosti:

   - Če je podniz v oblike $a^p b^q$ pri $p > 0$ in $q > 0$, se bodo pri njegovem napihovanju z eksponentom $i > 1$ simboli a in b med seboj prepletli. Enako velja v vseh drugih primerih, kjer je podniz v ali x sestavljen iz več kot ene vrste simbolov. To pomeni, da se lahko osredotočimo le na primere, ko sta tako v kot x oblike d^i, kjer je $d \in \{a, b, c\}$ in $i \geqslant 0$.

- Podniza v in x se oba nahajata znotraj podniza a^n. V tem primeru lahko vzamemo $i = 0$ in dobimo niz $z'(0) = uv^0wx^0y = a^{n-|vx|}b^nc^n$. Ker je $|vx| \geqslant 1$, ta niz vsebuje kvečjemu $(n-1)$ a-jev, a še vedno po n b-jev in c-jev, zato ne pripada jeziku.
- Podniza v in x sta oba znotraj podniza b^n ali oba znotraj podniza c^n. Ta primera sta analogna prejšnjemu.
- Podniz v je znotraj podniza a^n, podniz x je znotraj podniza b^n, poleg tega pa velja $|v| \geqslant 1$ in $|x| \geqslant 1$ (možnosti $|v| = 0 \wedge |x| \geqslant 1$ in $|v| \geqslant 1 \wedge |x| = 0$ smo že upoštevali). V tem primeru lahko vnovič vzamemo $i = 0$. Niz $z'(0) = a^{n-|v|}b^{n-|x|}c^n$ namreč še vedno vsebuje n c-jev, a kvečjemu po $(n-1)$ a-jev in b-jev, zato ne pripada jeziku.
- Podniz v je znotraj podniza b, podniz x pa znotraj podniza c. Ta primer je analogen prejšnjemu.

Drugih možnosti ni. Zaradi lastnosti $|vwx| \leqslant n$ se ne more zgoditi, da bi bil podniz v znotraj podniza a^n, podniz x pa znotraj podniza c^n.

2. Poskusimo z besedo $z = a^nb^{n+1}c^{n+1}$. Analizirajmo vse možne položaje podnizov v in x:

   - Če sta tako v kot x znotraj podniza a^n, lahko zapišemo $vx = a^p$ pri $p > 0$. Vzamemo $i = 2$ in dobimo $z'(2) = a^{n+p}b^{n+1}c^{n+1}$. Ker je $p > 0$, ta beseda ne pripada jeziku.
   - V primeru $vx = b^p$ $(p > 0)$ vzamemo $i = 0$ in dobimo $z'(0) = a^nb^{n+1-p}c^{n+1}$. Ker je $p > 0$, velja $z'(0) \notin L$ (število a-jev ni več strogo manjše od števila b-jev).
   - Na enak način obravnavamo primer $vx = c^p$ $(p > 0)$.
   - V primeru $v = a^p$ in $x = b^q$ $(p > 0 \wedge q > 0)$ velja $z'(2) = a^{n+p}b^{n+1+p}c^{n+1} \notin L$.
   - V primeru $v = b^p$ in $x = c^q$ $(p > 0 \wedge q > 0)$ velja $z'(0) = a^nb^{n+1-p}c^{n+1-q} \notin L$.

   Izčrpali smo vse možnosti, saj se s primeri, ko podniz v ali x vsebuje več kot eno vrsto simbolov, tudi tokrat nima smisla ukvarjati.

3. Do rešitve nas pripelje beseda $z = a^nb^na^nb^n$. Zopet se lahko omejimo na primere, ko podniza v in x obsegata le po eno vrsto simbolov, saj ostali primeri pri $i > 1$ povzročijo prepletanje. Če je $vx = a^p$ $(p > 0)$, vzamemo $i = 0$ in dobimo $z'(0) = a^{n-p}b^na^nb^n \notin L$. Do enakih zaključkov vodijo tudi primeri $vx = b^p$, $vx = a_2^p$ (zapis a_2 predstavlja simbol a iz druge polovice besede) in $vx = b_2^p$. Če je $v = a^p$ in $x = b^q$ $(p > 0 \wedge q > 0)$, imamo $z'(0) = a^{n-p}b^{n-q}a^nb^n \notin L$. Na enak način obravnavamo tudi primer $v = b^p$ in $x = a_2^q$ ter primer $v = a_2^p$ in $x = b_2^q$.

4. Pričnemo z besedo $z = a^p$, kjer je p najmanjše praštevilo, ki je enako najmanj n. V vseh primerih velja $|vx| = a^k$, kjer je $k > 0$. Besedo lahko torej zapišemo kot $z = a^{p-k}a^k$ (v prvem delu smo združili podnize u, w in y, v drugem pa v in x). Pri napihovanju dobimo $z'(i) = a^{p-k}a^{ki} = a^{p+k(i-1)}$. Če vzamemo $i = p+1$, nastane niz $z'(p+1) = a^{p+kp} = a^{p(1+k)}$. Ker je $p \geqslant 2$ in $k \geqslant 1$, je število $p(1+k)$ sestavljeno, zato velja $z'(p+1) \notin L$.

5. Pričnemo z besedo $z = a^nb^na^nb^n$. Če sta tako v kot x del iste skupine n simbolov (npr. če sta oba znotraj prve skupine b-jev), vzamemo $i = 0$ in porušimo enakost med prvo in drugo polovico. Enako se zgodi v primerih, ko v in x pripadata sosednjima skupinama. Vsakokrat lahko vzamemo $i = 0$ in ugotovimo, da se zgodi vsaj eno od sledečega: (1) število a-jev iz prve skupine postane različno od števila a-jev iz druge

skupine; (2) število b-jev iz prve skupine postane različno od števila b-jev iz druge skupine.

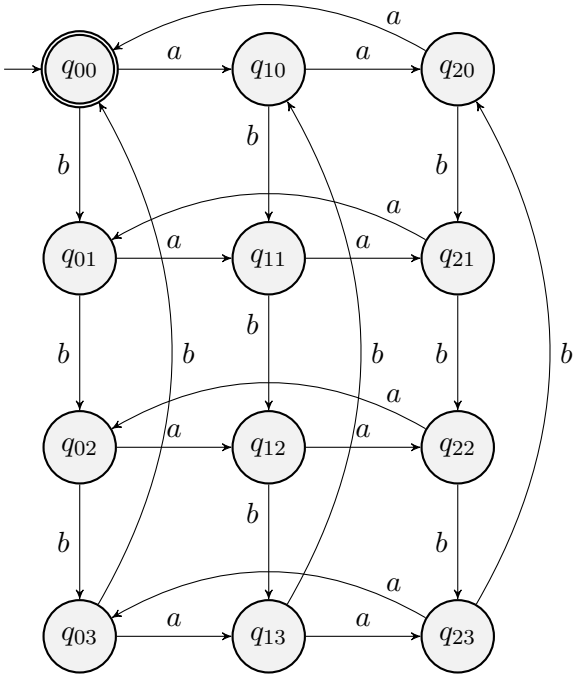
6. Pričnemo z besedo $z = a^n b^n b^n a^n a^n b^n$ in postopamo enako kot v prejšnjem primeru, le da imamo še več možnosti (6 za primere, ko sta v in x oba znotraj iste skupine n simbolov, in 5 za primere, ko v in x pripadata sosednjima skupinama simbolov).

7. Jezik je kontekstno neodvisen, ker lahko zanj zapišemo kontekstno neodvisno gramatiko (ali zgradimo skladovni avtomat):

$$S \to aSb \mid A$$
$$A \to bAa \mid \varepsilon$$

Jezik *ni* regularen. Trditev lahko dokažemo z lemo o napihovanju za regularne jezike. Pričnemo z besedo $z = a^n b^n a^n b^n$ in obravnavamo vse možne delitve $z = uvw$, kjer je $|uv| \leqslant n$ in $|v| \geqslant 1$. Pri vseh takih delitvah velja $v = a^p$ $(1 \leqslant p \leqslant n)$. Če besedo »napihnemo« z eksponentom 0, dobimo $z'(0) = a^{n-p} b^n a^n b^n \notin L$.

8. Jezik je regularen, saj ga sprejema končni avtomat. V sledečem DKA se v stanju q_{ij} nahajamo natanko v primeru, če smo do tistega trenutka prebrali $(3k + i)$ simbolov a in $(4\ell + j)$ simbolov b (za nek $k \geqslant 0$ in $\ell \geqslant 0$).



Poleg tega je jezik presek dveh enostavnejših regularnih jezikov nad $\Sigma = \{a, b\}$:

$$L = \{w \in \Sigma^* \mid \#_a(w) \bmod 3 = 0\} \cap \{w \in \Sigma^* \mid \#_b(w) \bmod 4 = 0\}$$

Gornji avtomat je dejansko produktni avtomat, zgrajen na podlagi avtomata za prvi jezik in avtomata za drugi jezik.

Ker je jezik regularen, je seveda tudi kontekstno neodvisen.

9. Jezik je kontekstno neodvisen, ker ga tvori sledeča gramatika (simbol e predstavlja regularni izraz ε):

$$E \to T \mid T + E$$

$$T \to F \mid FT$$
$$F \to \varnothing \mid e \mid 0 \mid 1 \mid F^* \mid (E)$$

Da jezik *ni* regularen, pokažemo z lemo o napihovanju za regularne jezike. Če vzamemo besedo $z = ({}^n0)^n$ (n predklepajev, ničla, n zaklepajev), vidimo, da je pri vseh veljavnih delitvah $z = uvw$ niz v oblike $({}^p$ pri $p > 0$. Torej je $z'(0) = ({}^{n-p}0)^n \notin L$.

8 Turingovi stroji

Turingov stroj pri vseh nalogah prične v stanju q_0, njegova bralno-pisalna glava pa se na začetku nahaja nad prvim simbolom vhodne besede. Pri vseh nalogah je končno stanje eno samo, in sicer q_F.

1. Preveriti moramo, ali se vhodna beseda konča z ničlo. V ta namen se v stanju q_0 pomikamo desno, dokler ne naletimo na simbol B (presledek). Nato preidemo v stanje q_1 in se premaknemo za eno celico v levo. Če ta celica vsebuje ničlo, preidemo v končno stanje in se ustavimo. Turingov stroj lahko torej formalno predstavimo kot $(\{q_0, q_F\},\ \{0,1\},\ \{0,1,B\},\ \delta,\ q_0,\ B,\ \{q_F\})$, pri čemer je funkcija δ definirana s sledečimi pravili:

$$\delta(q_0, 0) = (q_0, 0, R)$$
$$\delta(q_0, 1) = (q_0, 1, R)$$
$$\delta(q_0, B) = (q_1, B, L)$$
$$\delta(q_1, 0) = (q_F, 0, R)$$

   Če se beseda ne konča z ničlo, v stanju q_1 nimamo prehoda, zato ne dosežemo končnega stanja.

   Na besedi 10010 se Turingov stroj obnaša takole:

$$q_0 10010 \vdash 1q_0 0010 \vdash 10q_0 010 \vdash 100q_0 10 \vdash$$
$$1001q_0 0 \vdash 10010q_0 B \vdash 1001q_1 0 \vdash 10010q_F B$$

2. Če je vhodna beseda oblike $a^n b^n$, potem Turingov stroj izvrši n ciklov, pri čemer v vsakem ciklu spremeni prvi simbol a v simbol X, prvi simbol b pa v simbol Y. Vsak cikel se prične in konča v stanju q_0, bralno-pisalna glava pa bere prvi simbol a. Po izvedbi i-tega cikla je trenutni opis Turingovega stroja potemtakem enak $X^i q_0 a^{n-i} Y^i b^{n-i}$.

   Vsak cikel izvedemo s sledečimi koraki:

   (1) V stanju q_0 simbol a spremenimo v X, preidemo v stanje q_1 in se premaknemo desno.

   (2) Sprehodimo se desno do prvega simbola b, pri čemer potujemo čez simbole a in Y.

   (3) Simbol b spremenimo v Y, preidemo v stanje q_2 in se premaknemo levo.

   (4) Sprehodimo se levo do zadnjega simbola X, pri čemer potujemo čez simbole Y in a.

   (5) Preidemo v stanje q_0 in se premaknemo desno. Bralno-pisalna glava po vseh ciklih razen po zadnjem v tem trenutku bere prvi simbol a, po zadnjem ciklu pa bere prvi simbol Y.

$$\begin{aligned}
\delta(q_0, a) &= (q_1, X, R) \text{ (korak 1)}\\
\delta(q_1, a) &= (q_1, a, R) \text{ (korak 2)}\\
\delta(q_1, Y) &= (q_1, Y, R)\\
\delta(q_1, b) &= (q_2, Y, L) \text{ (korak 3)}\\
\delta(q_2, Y) &= (q_2, Y, L) \text{ (korak 4)}\\
\delta(q_2, a) &= (q_2, a, L)\\
\delta(q_2, X) &= (q_0, X, R) \text{ (korak 5)}
\end{aligned}$$

Pri vhodni besedi $a^n b^n$ bo po n ciklih trenutni opis Turingovega stroja enak $X^n q_0 Y^n$. Sedaj se moramo le še sprehoditi desno do prvega presledka in se prepričati, da je na naša pot tlakovana zgolj s simboli Y. Če naletimo na kak b, vemo, da je bilo število b-jev v vhodni besedi večje od števila a-jev, zato ne smemo preiti v končno stanje.

$$\begin{aligned}
\delta(q_0, Y) &= (q_3, Y, R)\\
\delta(q_3, Y) &= (q_3, Y, R)\\
\delta(q_3, B) &= (q_F, B, L)
\end{aligned}$$

Če bi bilo v vhodni besedi več a-jev kot b-jev, bi v stanju q_2 prispeli do presledka in ne bi imeli ustreznega prehoda. Če simboli a in b ne bi nastopali v pravilnem vrstnem redu, bi prav tako prej ali slej ostali brez prehoda.

Upoštevati moramo še dejstvo, da prazna beseda tudi pripada jeziku:

$$\delta(q_0, B) = (q_F, B, R)$$

Stroj se na besedi $aabb$ obnaša takole:

$$\begin{aligned}
q_0aabb \vdash Xq_1abb \vdash Xaq_1bb \vdash Xq_2aYb \vdash q_2XaYb \vdash\\
Xq_0aYb \vdash XXq_1Yb \vdash XXYq_1b \vdash XXq_2YY \vdash Xq_2XYY \vdash\\
XXq_0YY \vdash XXYq_3Y \vdash XXYYq_3B \vdash XXYq_FY
\end{aligned}$$

3. Postopamo na enak način kot pri prejšnji nalogi. Turingov stroj pri vhodni besedi $a^n b^n c^n$ izvrši n ciklov, pri čemer je trenutni opis po i-tem ciklu enak $X^i q_0 a^{n-i} Y^i b^{n-i} Z^i c^{n-i}$.

$$\begin{aligned}
\delta(q_0, a) &= (q_1, X, R) && \text{(prvi } a \text{ v } X)\\
\delta(q_1, a) &= (q_1, a, R) && \text{(desno do prvega } b)\\
\delta(q_1, Y) &= (q_1, Y, R)\\
\delta(q_1, b) &= (q_2, Y, R) && \text{(prvi } b \text{ v } Y)\\
\delta(q_2, b) &= (q_2, b, R) && \text{(desno do prvega } c)\\
\delta(q_2, Z) &= (q_2, Z, R)\\
\delta(q_2, c) &= (q_3, Z, L) && \text{(prvi } c \text{ v } Z)\\
\delta(q_3, Z) &= (q_3, Z, L) && \text{(levo do zadnjega } X)\\
\delta(q_3, b) &= (q_3, b, L)\\
\delta(q_3, Y) &= (q_3, Y, L)\\
\delta(q_3, a) &= (q_3, a, L)\\
\delta(q_3, X) &= (q_0, X, R) && \text{(naslednji cikel)}
\end{aligned}$$

Po n ciklih se trenutni opis glasi $X^n q_0 Y^n Z^n$. Sedaj le še preverimo, ali do konca besede sledijo zgolj simboli Y in Z.

$$\delta(q_0, Y) = (q_4, Y, R)$$

$$\delta(q_4, Y) = (q_4, Y, R)$$
$$\delta(q_4, Z) = (q_4, Z, R)$$
$$\delta(q_4, B) = (q_F, B, L)$$

Jeziku pripada tudi beseda ε:

$$\delta(q_0, B) = (q_F, B, R)$$

4. Turingov stroj pri vhodni besedi oblike $a_1a_2 \dots a_n\#a_1a_2 \dots a_n$ ($a_i \in \{0, 1\}$ za vsak i) izvrši n ciklov. Po i-tem ciklu je trenutni opis enak $X^i q_0 a_{i+1} \dots a_n \# Y^i a_{i+1} \dots a_n$. Stroj izvrši i-ti cikel na sledeči način:

   - Simbol a_{i+1} spremeni v X in preide v stanje q_{10} (če je veljalo $a_{i+1} = 0$) oziroma q_{11} (če je veljalo $a_{i+1} = 1$). Simbol, ki smo ga prebrali, bomo morali primerjati z istoležnim simbolom v drugi polovici besede, zato si ga moramo zapomniti. To storimo z drugo komponento indeksa stanja.
   - Potuje desno. Ko prispe do simbola #, preide v stanje q_{20} (če je bil prej v stanju q_{10}) oziroma q_{21} (če je bil prej v stanju q_{11}).
   - Potuje desno preko simbolov Y, takoj zatem pa se ustavi. Če se trenutni simbol pod bralno-pisalno glavo ujema z drugim indeksom stanja, spremeni simbol v Y, zamenja stanje v q_3 in se premakne levo.
   - Potuje levo in se ustavi pri zadnjem simbolu X. Stanje spremeni v q_0 in se premakne desno.

$$\delta(q_0, 0) = (q_{10}, X, R)$$
$$\delta(q_0, 1) = (q_{11}, X, R)$$
$$\delta(q_{10}, 0) = (q_{10}, 0, R)$$
$$\delta(q_{10}, 1) = (q_{10}, 1, R)$$
$$\delta(q_{11}, 0) = (q_{11}, 0, R)$$
$$\delta(q_{11}, 1) = (q_{11}, 1, R)$$
$$\delta(q_{10}, \#) = (q_{20}, \#, R)$$
$$\delta(q_{11}, \#) = (q_{21}, \#, R)$$
$$\delta(q_{20}, Y) = (q_{20}, Y, R)$$
$$\delta(q_{21}, Y) = (q_{21}, Y, R)$$
$$\delta(q_{20}, 0) = (q_3, Y, L)$$
$$\delta(q_{21}, 1) = (q_3, Y, L)$$
$$\delta(q_3, Y) = (q_3, Y, L)$$
$$\delta(q_3, \#) = (q_3, \#, L)$$
$$\delta(q_3, 0) = (q_3, 0, L)$$
$$\delta(q_3, 1) = (q_3, 1, L)$$
$$\delta(q_3, X) = (q_0, X, R)$$

Po zaključku n-tega cikla je trenutni opis enak $X^n q_0 \# Y^n$. Sedaj še enkrat potujemo desno in preverimo, ali je preostanek vhoda enak $\#Y^*$.

$$\delta(q_0, \#) = (q_4, \#, R)$$
$$\delta(q_4, Y) = (q_4, Y, R)$$
$$\delta(q_4, B) = (q_F, B, L)$$

5. Turingov stroj pri besedi $a^p b^q c^{pq}$ izvede p ciklov, pri čemer je trenutni opis po i-tem ciklu enak $X^i q_0 a^{p-i} b^q Z^{qi} c^{(p-i)q}$. V i-tem ciklu spremeni prvi simbol a v X, q simbolov desno od zadnjega simbola Z pa spremeni v simbole Z. To operacijo izvrši v q iteracijah, pri čemer v vsaki iteraciji zamenja prvi simbol b z Y in prvi simbol c z Z. Po zaključku tega postopka simbole Y spet zamenja s simboli b.

$$\begin{aligned}
\delta(q_0, a) &= (q_1, X, R) && \text{(začetek zunanje zanke; prvi } a \text{ v } X) \\
\delta(q_1, a) &= (q_1, a, R) \\
\delta(q_1, b) &= (q_2, Y, R) && \text{(začetek notranje zanke; prvi } b \text{ v } Y) \\
\delta(q_2, b) &= (q_2, b, R) \\
\delta(q_2, Z) &= (q_2, Z, R) \\
\delta(q_2, c) &= (q_3, Z, L) && \text{(prvi } c \text{ v } Z) \\
\delta(q_3, Z) &= (q_3, Z, L) \\
\delta(q_3, b) &= (q_3, b, L) \\
\delta(q_3, Y) &= (q_1, Y, R) && \text{(naslednji obhod notranje zanke)} \\
\delta(q_1, Z) &= (q_4, Z, L) \\
\delta(q_4, Y) &= (q_4, b, L) && (Y^q \to b^q) \\
\delta(q_4, a) &= (q_4, a, L) \\
\delta(q_4, X) &= (q_0, X, R) && \text{(naslednji obhod zunanje zanke)}
\end{aligned}$$

Po p ciklih bo trenutni opis enak $X^p q_0 b^q Z^{pi}$. Sedaj se le še sprehodimo do konca besede.

$$\begin{aligned}
\delta(q_0, b) &= (q_5, b, R) \\
\delta(q_5, b) &= (q_5, b, R) \\
\delta(q_5, Z) &= (q_6, Z, R) \\
\delta(q_6, Z) &= (q_6, Z, R) \\
\delta(q_6, B) &= (q_F, B, L)
\end{aligned}$$

Zakaj spremenimo stanje, ko preberemo prvi Z? Če tega ne bi storili, bi sprejemali tudi besede oblike $a^i b^j c^{ij} bb^*$.

6. Turingov stroj pri vhodni besedi a^{2^n} najprej spremeni prvi simbol a v X in se vrne na začetek besede, nato pa izvrši n ciklov. Po i-tem ciklu (za $1 \leqslant i \leqslant n-1$) je trenutni opis enak $q_2 X^{2^i} a^{2^n - 2^i}$. V vsakem ciklu torej podvoji število začetnih simbolov X. Pri tem si pomaga s simboli Y in Z: v vsaki iteraciji zamenja prvi simbol X s simbolom Y, prvi simbol a pa s simbolom Z. Ko se ta postopek izteče, stroj simbole Z med potovanjem v desno zamenja s simboli X in preveri, ali takoj zatem sledi presledek. Če to drži, besedo sprejme, sicer pa med potovanjem v levo prepiše vse simbole Y s simboli X in prične z naslednjim ciklom.

$$\begin{aligned}
\delta(q_0, a) &= (q_1, X, R) && \text{(prvi } a \text{ v } X) \\
\delta(q_1, a) &= (q_2, a, L) \\
\delta(q_2, X) &= (q_3, Y, R) && \text{(prvi } X \text{ v } Y) \\
\delta(q_3, X) &= (q_3, X, R) \\
\delta(q_3, Z) &= (q_3, Z, R) \\
\delta(q_3, a) &= (q_4, Z, L) && \text{(prvi } a \text{ v } Z) \\
\delta(q_4, Z) &= (q_4, Z, L)
\end{aligned}$$

$$\begin{aligned}
\delta(q_4, X) &= (q_4, X, L) \\
\delta(q_4, Y) &= (q_2, Y, R) \\
\delta(q_2, Z) &= (q_5, X, R) \quad (Z^* \to X^*) \\
\delta(q_5, Z) &= (q_5, X, R) \\
\delta(q_5, B) &= (q_F, B, L) \quad \text{(konec!)} \\
\delta(q_5, a) &= (q_6, a, L) \quad \text{(ni še konec)} \\
\delta(q_6, X) &= (q_6, X, L) \\
\delta(q_6, Y) &= (q_6, X, L) \quad (Y^* \to X^*) \\
\delta(q_6, B) &= (q_2, B, R) \quad \text{(naslednji cikel)}
\end{aligned}$$

Beseda a je tudi v jeziku:

$$\delta(q_1, B) = (q_F, B, L)$$

Vrednost $\lceil \log_2 n \rceil$ bi izračunali s štetjem ciklov: števec inicializiramo na 0, po vsakem prehodu v stanje q_2 pa ga povečamo za 1.

7. Na konec besede dodamo ničlo.

$$\begin{aligned}
\delta(q_0, 0) &= (q_0, 0, R) \\
\delta(q_0, 1) &= (q_0, 1, R) \\
\delta(q_0, B) &= (q_F, 0, L)
\end{aligned}$$

8. Premaknemo se do zadnjega simbola besede. Če je simbol enak 0, ga spremenimo v 1 in postopek zaključimo. Če je enak 1, ga spremenimo v 0, se pomaknemo za eno celico v levo in postopek ponovimo. Pri vhodni besedi 11^* bomo spremenili tudi presledek tik pred začetkom besede.

$$\begin{aligned}
\delta(q_0, 0) &= (q_0, 0, R) \\
\delta(q_0, 1) &= (q_0, 1, R) \\
\delta(q_0, B) &= (q_1, B, L) \\
\delta(q_1, 0) &= (q_F, 1, L) \\
\delta(q_1, 1) &= (q_1, 0, L) \\
\delta(q_1, B) &= (q_F, 1, L)
\end{aligned}$$

9. V vsakem ciklu zamenjamo prvo enico s simbolom X, prvi presledek desno od besede pa s simbolom Y. Vhodno besedo 1^n tako pretvorimo v $X^n Y^n$, nato pa le še zamenjamo vse simbole X in Y z enicami.

$$\begin{aligned}
\delta(q_0, 1) &= (q_1, X, R) \\
\delta(q_1, 1) &= (q_1, 1, R) \\
\delta(q_1, Y) &= (q_1, Y, R) \\
\delta(q_1, B) &= (q_2, Y, L) \\
\delta(q_2, Y) &= (q_2, Y, L) \\
\delta(q_2, 1) &= (q_2, 1, L) \\
\delta(q_2, X) &= (q_0, X, R) \\
\delta(q_0, Y) &= (q_3, Y, L) \\
\delta(q_3, X) &= (q_3, X, L)
\end{aligned}$$

$$\delta(q_3, B) = (q_4, B, R)$$
$$\delta(q_4, X) = (q_4, 1, R)$$
$$\delta(q_4, Y) = (q_4, 1, R)$$

Lahko dodamo še prehod $\delta(q_4, B) = (q_F, B, L)$, ni pa treba, saj v primerih, ko Turingov stroj deluje kot računalnik, ni važno, v katerem stanju se ustavimo. Pomembno je le to, da se ustavimo.

9 Razširitve Turingovih strojev

1. Simbole a med potovanjem v desno prepišemo na drugi trak, simbole b na tretji trak, simbole c pa na četrti trak. Pomembno je, da stroj vsako skupino simbolov obravnava v ločenem stanju, sicer bi lahko sprejel besedo, v kateri se simboli a, b in c med seboj prepletajo. Ko prekopiramo celotno vhodno besedo, pričnemo na drugem, tretjem in četrtem traku sočasno potovati v levo. Če na vseh treh trakovih hkrati naletimo na presledek, preidemo v končno stanje in se ustavimo.

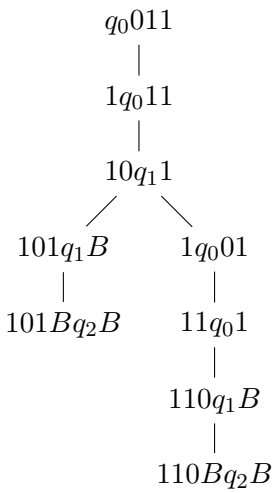
$$\delta(q_0, a, B, B, B) = (q_0, (a, R), (a, R), (B, S), (B, S))$$
$$\delta(q_0, b, B, B, B) = (q_1, (b, R), (B, S), (b, R), (B, S))$$
$$\delta(q_1, b, B, B, B) = (q_1, (b, R), (B, S), (b, R), (B, S))$$
$$\delta(q_1, c, B, B, B) = (q_2, (c, R), (B, S), (B, S), (c, R))$$
$$\delta(q_2, c, B, B, B) = (q_2, (c, R), (B, S), (B, S), (c, R))$$
$$\delta(q_2, B, B, B, B) = (q_3, (B, S), (B, L), (B, L), (B, L))$$
$$\delta(q_3, B, a, b, c) = (q_3, (B, S), (a, L), (b, L), (c, L))$$
$$\delta(q_3, B, B, B, B) = (q_F, (B, S), (B, S), (B, S), (B, S))$$

2. Prvi sumand pustimo na prvem traku. Drugi sumand prepišemo na drugi trak, na prvem pa ga prepišemo s presledki. S presledkom prepišemo tudi znak +. Na obeh trakovih se nato premaknemo levo do zadnje števke. Zatem po obeh trakovih usklajeno potujemo v levo, seštevamo istoležne števke in rezultate zapisujemo na tretji trak. S seštevanjem pričnemo v stanju q_{30}. Ko pri računanju vsote istoležnih števk pridelamo prenos, preidemo v stanje q_{31}, ko nimamo prenosa, pa se vrnemo v stanje q_{30}. Presledek obravnavamo na enak način kot ničlo. Dva presledka seštejemo samo v primeru, če smo v stanju q_{31}. Če smo v takšni situaciji v stanju q_{30}, se ustavimo.

$$\delta(q_0, 0, B, B) = (q_0, (0, R), (B, S), (B, S))$$
$$\delta(q_0, 1, B, B) = (q_0, (1, R), (B, S), (B, S))$$
$$\delta(q_0, +, B, B) = (q_1, (B, R), (B, S), (B, S))$$
$$\delta(q_1, 0, B, B) = (q_1, (B, R), (0, R), (B, S))$$
$$\delta(q_1, 1, B, B) = (q_1, (B, R), (1, R), (B, S))$$
$$\delta(q_1, B, B, B) = (q_2, (B, L), (B, S), (B, S))$$
$$\delta(q_2, B, B, B) = (q_2, (B, L), (B, S), (B, S))$$
$$\delta(q_2, 0, B, B) = (q_{30}, (0, S), (B, L), (B, S))$$
$$\delta(q_2, 1, B, B) = (q_{30}, (1, S), (B, L), (B, S))$$

$$\begin{aligned}
\delta(q_{30}, 0, 0, B) &= (q_{30}, (0, L), (0, L), (0, L)) \\
\delta(q_{30}, 0, 1, B) &= (q_{30}, (0, L), (1, L), (1, L)) \\
\delta(q_{30}, 1, 0, B) &= (q_{30}, (1, L), (0, L), (1, L)) \\
\delta(q_{30}, 1, 1, B) &= (q_{31}, (1, L), (1, L), (0, L)) \\
\delta(q_{30}, B, 0, B) &= (q_{30}, (B, S), (0, L), (0, L)) \\
\delta(q_{30}, 0, B, B) &= (q_{30}, (0, L), (B, S), (0, L)) \\
\delta(q_{30}, B, 1, B) &= (q_{30}, (B, S), (1, L), (1, L)) \\
\delta(q_{30}, 1, B, B) &= (q_{30}, (1, L), (B, S), (1, L)) \\
\delta(q_{31}, 0, 0, B) &= (q_{30}, (0, L), (0, L), (1, L)) \\
\delta(q_{31}, 0, 1, B) &= (q_{31}, (0, L), (0, L), (0, L)) \\
\delta(q_{31}, 1, 0, B) &= (q_{31}, (0, L), (0, L), (0, L)) \\
\delta(q_{31}, 1, 1, B) &= (q_{31}, (0, L), (0, L), (1, L)) \\
\delta(q_{31}, B, 0, B) &= (q_{30}, (B, S), (0, L), (1, L)) \\
\delta(q_{31}, 0, B, B) &= (q_{30}, (0, L), (B, S), (1, L)) \\
\delta(q_{31}, B, 1, B) &= (q_{31}, (B, S), (1, L), (0, L)) \\
\delta(q_{31}, 1, B, B) &= (q_{31}, (1, L), (B, S), (0, L)) \\
\delta(q_{31}, B, B, B) &= (q_{30}, (B, S), (B, S), (1, L))
\end{aligned}$$

3.



4. Uporabimo dva trakova. V stanju q_0 zgolj preverimo, ali je vhodna beseda prazna. V stanju q_1 po prvem traku potujemo desno, dokler ne preberemo domnevne prve polovice vhodne besede. Nato preidemo v stanje q_2 in vse simbole do konca besede prestavimo na drugi trak, na prvem traku pa jih zamenjamo s presledki. V drugi fazi (stanje q_3) se na obeh trakovih premaknemo levo do desnega roba besede. V tretji fazi (stanje q_4) se po obeh trakovih sočasno pomikamo levo in primerjamo besedi po simbolih. Če na obeh trakovih sočasno prispemo do presledka, preidemo v končno stanje in se ustavimo.

$$\begin{aligned}
\delta(q_0, B, B) &= \{(q_F, (B, S), (B, S))\} \\
\delta(q_0, 0, B) &= \{(q_1, (0, S), (B, S))\} \\
\delta(q_0, 1, B) &= \{(q_1, (1, S), (B, S))\} \\
\delta(q_1, 0, B) &= \{(q_1, (0, R), (B, S)), (q_2, (B, R), (0, R))\} \\
\delta(q_1, 1, B) &= \{(q_1, (1, R), (B, S)), (q_2, (B, R), (1, R))\}
\end{aligned}$$

$$\begin{aligned}
\delta(q_2, 0, B) &= \{(q_2, (B, R), (0, R)\} \\
\delta(q_2, 1, B) &= \{(q_2, (B, R), (1, R)\} \\
\delta(q_2, B, B) &= \{(q_3, (B, L), (B, S)\} \\
\delta(q_3, B, B) &= \{(q_3, (B, L), (B, S)\} \\
\delta(q_3, 0, B) &= \{(q_4, (0, S), (B, L)\} \\
\delta(q_3, 1, B) &= \{(q_4, (1, S), (B, L)\} \\
\delta(q_4, 0, 0) &= \{(q_4, (0, L), (0, L)\} \\
\delta(q_4, 1, 1) &= \{(q_4, (1, L), (1, L)\} \\
\delta(q_4, B, B) &= \{(q_F, (B, S), (B, S)\}
\end{aligned}$$

5. V stanju q_0 spremenimo prvi simbol v X, preidemo v stanje q_{10} (če smo prebrali ničlo) oziroma q_{11} (če smo prebrali enico) in se premaknemo desno. V enem oziroma drugem stanju se nato pomikamo desno, dokler ne uganemo, da smo se znašli na prvem simbolu druge polovice besede. Če se ta simbol ujema z drugo komponento indeksa stanja, spremenimo simbol v Y in se premaknemo v levo, nato pa sledi že znan postopek primerjave prve in druge polovice besede. Simbole v prvi polovici zamenjujemo s simboli X, simbole v drugi polovici pa s simboli Y.

$$\begin{aligned}
\delta(q0, 0) &= \{(q10, X, R)\} \\
\delta(q0, 1) &= \{(q11, X, R)\} \\
\delta(q10, 0) &= \{(q10, 0, R), (q2, Y, L)\} \\
\delta(q10, 1) &= \{(q10, 1, R)\} \\
\delta(q11, 0) &= \{(q11, 0, R)\} \\
\delta(q11, 1) &= \{(q11, 1, R), (q2, Y, L)\} \\
\delta(q2, 0) &= \{(q2, 0, L)\} \\
\delta(q2, 1) &= \{(q2, 1, L)\} \\
\delta(q2, X) &= \{(q3, X, R)\} \\
\delta(q3, 0) &= \{(q40, X, R)\} \\
\delta(q3, 1) &= \{(q41, X, R)\} \\
\delta(q40, 0) &= \{(q40, 0, R)\} \\
\delta(q40, 1) &= \{(q40, 1, R)\} \\
\delta(q41, 0) &= \{(q41, 0, R)\} \\
\delta(q41, 1) &= \{(q41, 1, R)\} \\
\delta(q40, Y) &= \{(q50, Y, R)\} \\
\delta(q41, Y) &= \{(q51, Y, R)\} \\
\delta(q50, Y) &= \{(q50, Y, R)\} \\
\delta(q51, Y) &= \{(q51, Y, R)\} \\
\delta(q50, 0) &= \{(q2, Y, L)\} \\
\delta(q51, 1) &= \{(q2, Y, L)\} \\
\delta(q2, Y) &= \{(q2, Y, L)\} \\
\delta(q3, Y) &= \{(q6, Y, R)\} \\
\delta(q6, Y) &= \{(q6, Y, R)\} \\
\delta(q6, B) &= \{(qF, B, L)\} \\
\delta(q0, B) &= \{(qF, B, R)\}
\end{aligned}$$

6. Definirali bomo dvotračni nedeterministični Turingov stroj. Denimo, da imamo na prvem traku zapisano eniško število n. Na drugi trak zapišemo število, ki je enako najmanj 2, nato pa preverimo, ali je to število manjše od n in obenem deli n.

   Potek izvajanja Turingovega stroja lahko razdelimo v tri faze:

   - Najprej na drugi trak zapišemo najmanj dve enici. Ko zaključimo, se na drugem traku pomaknemo na začetek. Na prvem traku zaenkrat ne počnemo ničesar.

$$\begin{aligned}
\delta(q_0, 1, B) &= \{(q_1, (1, S), (1, R))\} \\
\delta(q_1, 1, B) &= \{(q_2, (1, S), (1, R))\} \\
\delta(q_2, 1, B) &= \{(q_2, (1, S), (1, R)), (q_3, (1, S), (B, L))\} \\
\delta(q_3, 1, 1) &= \{(q_3, (1, S), (1, L))\} \\
\delta(q_3, 1, B) &= \{(q_4, (1, S), (B, R))\}
\end{aligned}$$

   - Preverimo, ali je število na prvem traku (deljenec) strogo večje od števila na drugem traku (delitelj). Nato se na obeh trakovih premaknemo na začetek.

$$\begin{aligned}
\delta(q_4, 1, 1) &= \{(q_4, (1, R), (1, R))\} \\
\delta(q_4, 1, B) &= \{(q_5, (1, L), (B, L))\} \\
\delta(q_5, 1, 1) &= \{(q_5, (1, L), (1, L))\} \\
\delta(q_5, B, B) &= \{(q_6, (B, R), (B, R))\}
\end{aligned}$$

   - Preverimo, ali je število na prvem traku večkratnik števila na drugem traku. To storimo tako, da na prvem traku potujemo od levega do desnega roba besede, na drugem pa (usklajeno s premikanjem po prvem traku) izmenično od levega do desnega in od desnega do levega roba. Če na obeh trakovih sočasno prispemo do presledka, vemo, da je število na prvem traku večkratnik števila na drugem traku.

$$\begin{aligned}
\delta(q_6, 1, 1) &= \{(q_6, (1, R), (1, R))\} \\
\delta(q_6, B, B) &= \{(q_F, (B, S), (B, S))\} \\
\delta(q_6, 1, B) &= \{(q_7, (1, S), (B, L))\} \\
\delta(q_7, 1, 1) &= \{(q_7, (1, R), (1, L))\} \\
\delta(q_7, B, B) &= \{(q_F, (B, S), (B, S))\} \\
\delta(q_7, 1, B) &= \{(q_6, (1, S), (B, R))\}
\end{aligned}$$

7. Naloge se lotimo podobno kot prejšnje, le da delitelja ne uganemo, ampak sistematično preizkušamo kandidatne delitelje 2, 3, 4 itd. Zaključimo, ko kandidatni delitelj postane večji ali enak vhodnemu številu.

$$\begin{aligned}
\delta(q0, 1, B) &= (q1, (1, S), (1, R)) \\
\delta(q1, 1, B) &= (q2, (1, S), (1, L)) \\
\delta(q2, 1, 1) &= (q2, (1, R), (1, R)) \\
\delta(q2, 1, B) &= (q3, (1, L), (B, L)) \\
\delta(q3, 1, 1) &= (q3, (1, L), (1, L))
\end{aligned}$$

$$\begin{aligned}
\delta(q3, B, B) &= (q4, (B, R), (B, R)) \\
\delta(q4, 1, 1) &= (q4, (1, R), (1, R)) \\
\delta(q4, 1, B) &= (q5, (1, S), (B, L)) \\
\delta(q5, 1, 1) &= (q5, (1, R), (1, L)) \\
\delta(q5, 1, B) &= (q4, (1, S), (B, R)) \\
\delta(q4, B, B) &= (qF, (B, S), (B, S)) \\
\delta(q5, B, B) &= (qF, (B, S), (B, S)) \\
\delta(q4, B, 1) &= (q6, (B, L), (1, S)) \\
\delta(q5, B, 1) &= (q6, (B, L), (1, S)) \\
\delta(q6, 1, 1) &= (q6, (1, L), (1, S)) \\
\delta(q6, B, 1) &= (q6, (B, S), (1, L)) \\
\delta(q6, B, B) &= (q2, (B, R), (1, S))
\end{aligned}$$

8. Naj bosta jezika L_1 in L_2 odločljiva in naj bosta M_1 in M_2 enotračna Turingova stroja, ki ju sprejemata in se ustavita pri vsakem vhodu. Naj bosta Q_1 in Q_2 množici njunih stanj, F_1 in F_2 pa množici njunih končnih stanj. Stroj M za jezik $L_1 \cup L_2$ sestavimo kot dvotračni Turingov stroj z množico stanj $Q_1 \times Q_2$ in množico končnih stanj $F_1 \times Q_2 \cup Q_1 \times F_2$. Stroj M najprej prekopira svoj vhod na drugi trak, nato pa na prvem traku simulira stroj M_1, na drugem pa sočasno stroj M_2 (v vsakem koraku na prvem traku izvede en korak stroja M_1, na drugem pa en korak stroja M_2). Ker se stroja M_1 in M_2 ustavita na vseh vhodih, bo to veljalo tudi za stroj M. Stroj M sprejme svoj vhod, če se vsaj eden od strojev M_1 in M_2 ustavi v končnem stanju. Stroj M torej sprejema jezik $L_1 \cup L_2$, obenem pa se ustavi na vsakem vhodu, zato je jezik $L_1 \cup L_2$ odločljiv.

   Stroj za jezik $M_1 \cap M_2$ sestavimo na enak način, le da je njegova množica končnih stanj enaka $F_1 \times F_2$. Vhod sprejmemo natanko v primeru, če ga sprejmeta oba stroja.

10 Odločitveni problemi in univerzalni Turingov stroj

1. Množico $A = \{a_1, a_2, \ldots, a_n\}$ zakodiramo kot $01^{a_1}01^{a_2}\ldots01^{a_n}0$. Jezik, ki je enakovreden problemu, je torej sestavljen iz vseh besed oblike $01^{a_1}01^{a_2}\ldots01^{a_n}0$, pri katerih obstajajo indeksi $i_1, \ldots, i_k$, tako da je $\sum_{j=1}^{k} a_{i_j} = \frac{1}{2}\sum_{i=1}^{n} a_i$.

   Stroj za reševanje tega problema sestavimo kot 3-tračni nedeterministični Turingov stroj. Vhodni niz je kot običajno zapisan na prvem traku. Drugi trak bo med delovanjem stroja hranil vsoto elementov (v eniškem zapisu), ki smo jih do danega trenutka postavili v prvo podmnožico, tretji trak pa vsoto elementov, ki smo jih do danega trenutka postavili v drugo podmnožico. Ob vsakem pričetku zaporedja enic se nedeterministično odločimo, ali bomo zaporedje prepisali na drugi ali na tretji trak. Na koncu zgolj preverimo, ali sta zaporedji enic na drugem in tretjem traku enako dolgi.

   Na primer, pri množici $A = \{3, 6, 4, 1, 2\}$ bo prvi trak na začetku vseboval niz 01110-11111101111010110, drugi in tretji trak pa bosta prazna. Ko bomo prvo, tretje in četrto zaporedje enic prepisali na drugi trak, drugo in peto pa na tretji trak, bosta tako drugi kot tretji trak vsebovala po 8 enic, kar pomeni, da se odgovor na vprašanje, ali je množico A mogoče razbiti na dve disjunktni podmnožici z enako vsoto, glasi »da«.

$$\delta(q_0, 0, B, B) = \{(q_1, (0, R), (B, S), (B, S))\}$$

$$\begin{aligned}
\delta(q_1, 1, B, B) &= \{(q_2, (1, R), (1, R), (B, S)), (q_3, (1, R), (B, S), (1, R))\} \\
\delta(q_2, 1, B, B) &= \{(q_2, (1, R), (1, R), (B, S))\} \\
\delta(q_3, 1, B, B) &= \{(q_3, (1, R), (B, S), (1, R))\} \\
\delta(q_2, 0, B, B) &= \{(q_1, (0, R), (B, S), (B, S))\} \\
\delta(q_3, 0, B, B) &= \{(q_1, (0, R), (B, S), (B, S))\} \\
\delta(q_1, B, B, B) &= \{(q_4, (B, S), (B, L), (B, L))\} \\
\delta(q_4, B, 1, 1) &= \{(q_4, (B, S), (1, L), (1, L))\} \\
\delta(q_4, B, B, B) &= \{(q_F, (B, S), (B, S), (B, S))\}
\end{aligned}$$

2. Problem iskanja maksimuma v zaporedju (a_1, a_2, …, a_n) lahko prevedemo v odločitveni problem takole:

   Ali je število a_k (pri nekem $k \in \{1, \ldots, n\}$) maksimum zaporedja?

   Sestavimo 2-tračni deterministični Turingov stroj. Na prvi trak zapišemo kodirano zaporedje in element k, pri čemer ju med seboj ločimo z dvema ničlama:

   $$01^{a_1}01^{a_2}\ldots01^{a_n}001^{a_k}0$$

   Dokler na prvem traku ne prispemo do dveh zaporednih ničel, prepisujemo zaporedja enic na drugi trak, pri čemer vsakokrat pričnemo na začetku traku. Po prebranih i elementih (zaporedjih enic) bo vsebina drugega traku torej enaka $1^{\max\{a_1,a_2,\ldots,a_i\}}$.

   $$\begin{aligned}
   \delta(q_0, 0, B) &= (q_1, (0, R), (B, S)) \\
   \delta(q_1, 1, B) &= (q_1, (1, R), (1, R)) \\
   \delta(q_1, 1, 1) &= (q_1, (1, R), (1, R)) \\
   \delta(q_1, 0, B) &= (q_2, (0, S), (B, L)) \\
   \delta(q_1, 0, 1) &= (q_2, (0, S), (1, L)) \\
   \delta(q_2, 0, 1) &= (q_2, (0, S), (1, L)) \\
   \delta(q_2, 0, B) &= (q_3, (0, R), (B, R)) \\
   \delta(q_3, 1, 1) &= (q_1, (1, S), (1, S)) \\
   \delta(q_3, 0, 1) &= (q_4, (0, R), (1, S)) \\
   \delta(q_4, 1, 1) &= (q_4, (1, R), (1, R)) \\
   \delta(q_4, 0, B) &= (q_5, (0, R), (B, S)) \\
   \delta(q_5, B, B) &= (q_F, (B, S), (B, S))
   \end{aligned}$$

3. Problem nahrbtnika lahko v odločitveni problem prevedemo takole:

   Ali pri podani n-terki prostornin predmetov $(v_1, \ldots, v_n)$, n-terki cen predmetov $(c_1, \ldots, c_n)$, prostornini nahrbtnika (V) in številu $c^* \geqslant 0$ obstaja množica $I \subseteq \{1, \ldots, n\}$, tako da velja $\sum_{i \in I} v_i \leqslant V$ in $\sum_{i \in I} c_i > c^*$?

   Takšna definicija nam omogoča učinkovito reševanje optimizacijske različice problema. Tvorimo števila $c^* = 1$, $c^* = 2$, $c^* = 4$, $c^* = 8$, $c^* = 16$, … in vsakokrat rešimo odločitveni problem. Ko najdemo tak k, da bo odgovor za $c^* = 2^k$ enak »da«, za $c^* = 2^{k+1}$ pa »ne«, vemo, da se optimalna skupna cena nahaja med $2^k + 1$ in 2^{k+1}, zato jo lahko poiščemo z bisekcijo.

   Primerek $((v_1, \ldots, v_n), (c_1, \ldots, c_n), V, c^*)$ zakodiramo takole:

   $$01^{v_1}01^{v_2}\ldots01^{v_n}001^{c_1}01^{c_2}\ldots01^{c_n}001^{V}01^{c^*}0$$

Sestavimo 5-tračni nedeterministični Turingov stroj. Prostornine predmetov pustimo na prvem traku, cene prenesemo na drugi trak, prostornino nahrbtnika in število c^* (niz $1^V 01^{c^*}$) pa prestavimo na tretji trak. Nato se na vsakem od teh treh trakov premaknemo na začetek niza.

$$\begin{aligned}
\delta(q_0, 0, B, B, B, B) &= \{(q_1, (B, R), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_1, 1, B, B, B, B) &= \{(q_1, (1, R), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_1, 0, B, B, B, B) &= \{(q_2, (0, R), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_2, 1, B, B, B, B) &= \{(q_1, (1, R), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_2, 0, B, B, B, B) &= \{(q_3, (B, R), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_3, 1, B, B, B, B) &= \{(q_3, (B, R), (1, R), (B, S), (B, S), (B, S))\} \\
\delta(q_3, 0, B, B, B, B) &= \{(q_4, (B, R), (0, R), (B, S), (B, S), (B, S))\} \\
\delta(q_4, 1, B, B, B, B) &= \{(q_3, (B, R), (1, R), (B, S), (B, S), (B, S))\} \\
\delta(q_4, 0, B, B, B, B) &= \{(q_5, (B, R), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_5, 1, B, B, B, B) &= \{(q_5, (B, R), (B, S), (1, R), (B, S), (B, S))\} \\
\delta(q_5, 0, B, B, B, B) &= \{(q_6, (B, R), (B, S), (0, R), (B, S), (B, S))\} \\
\delta(q_6, 1, B, B, B, B) &= \{(q_6, (B, R), (B, S), (1, R), (B, S), (B, S))\} \\
\delta(q_6, 0, B, B, B, B) &= \{(q_7, (B, R), (B, S), (0, R), (B, S), (B, S))\} \\
\delta(q_7, B, B, B, B, B) &= \{(q_8, (B, L), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_8, B, B, B, B, B) &= \{(q_8, (B, L), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_8, 0, B, B, B, B) &= \{(q_9, (0, L), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_8, 1, B, B, B, B) &= \{(q_9, (1, L), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_9, 0, B, B, B, B) &= \{(q_9, (0, L), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_9, 1, B, B, B, B) &= \{(q_9, (1, L), (B, S), (B, S), (B, S), (B, S))\} \\
\delta(q_9, B, B, B, B, B) &= \{(q_{10}, (B, S), (B, L), (B, S), (B, S), (B, S))\} \\
\delta(q_{10}, B, 0, B, B, B) &= \{(q_{10}, (B, S), (0, L), (B, S), (B, S), (B, S))\} \\
\delta(q_{10}, B, 1, B, B, B) &= \{(q_{10}, (B, S), (1, L), (B, S), (B, S), (B, S))\} \\
\delta(q_{10}, B, B, B, B, B) &= \{(q_{11}, (B, S), (B, S), (B, L), (B, S), (B, S))\} \\
\delta(q_{11}, B, B, 0, B, B) &= \{(q_{11}, (B, S), (B, S), (0, L), (B, S), (B, S))\} \\
\delta(q_{11}, B, B, 1, B, B) &= \{(q_{11}, (B, S), (B, S), (1, L), (B, S), (B, S))\} \\
\delta(q_{11}, B, B, B, B, B) &= \{(q_{12}, (B, R), (B, R), (B, R), (B, S), (B, S))\}
\end{aligned}$$

V naslednji fazi se za vsak $i \in \{1, \ldots, n\}$ nedeterministično odločimo, ali bomo i-ti predmet postavili v nahrbtnik. Če se odločimo za to možnost, potem zaporedje 1^{v_i} prepišemo na četrti, zaporedje 1^{c_i} pa na peti trak, sicer pa zaporedji enostavno preskočimo.

$$\begin{aligned}
\delta(q_{12}, 1, 1, 1, B, B) &= \{(q_{13}, (1, R), (1, R), (1, S), (1, R), (1, R)), \\
&\qquad (q_{14}, (1, R), (1, R), (1, S), (B, S), (B, S))\} \\
\delta(q_{13}, 1, 1, 1, B, B) &= \{(q_{13}, (1, R), (1, R), (1, S), (1, R), (1, R))\} \\
\delta(q_{13}, 1, 0, 1, B, B) &= \{(q_{13}, (1, R), (0, S), (1, S), (1, R), (B, S))\} \\
\delta(q_{13}, 0, 1, 1, B, B) &= \{(q_{13}, (0, S), (1, R), (1, S), (B, S), (1, R))\} \\
\delta(q_{13}, 0, 0, 1, B, B) &= \{(q_{12}, (0, R), (0, R), (1, S), (B, S), (B, S))\} \\
\delta(q_{14}, 1, 1, 1, B, B) &= \{(q_{14}, (1, R), (1, R), (1, S), (B, S), (B, S))\} \\
\delta(q_{14}, 1, 0, 1, B, B) &= \{(q_{14}, (1, R), (0, S), (1, S), (B, S), (B, S))\}
\end{aligned}$$

$$\delta(q_{14}, 0, 1, 1, B, B) = \{(q_{14}, (0, S), (1, R), (1, S), (B, S), (B, S))\}$$
$$\delta(q_{14}, 0, 0, 1, B, B) = \{(q_{12}, (0, R), (0, R), (1, S), (B, S), (B, S))\}$$

Na četrtem traku je sedaj zapisana vsota prostornin, na petem pa vsota cen izbranih predmetov. To pomeni, da moramo le še preveriti, ali je število enic na četrtem traku kvečjemu enako številu enic, ki se na tretjem traku nahajajo pred ničlo, število enic na petem traku pa strogo večje od števila enic, ki se na tretjem traku nahajajo za ničlo.

$$\delta(q_{12}, B, B, 1, B, B) = \{(q_{15}, (B, S), (B, S), (1, S), (B, L), (B, S))\}$$
$$\delta(q_{15}, B, B, 1, 1, B) = \{(q_{15}, (B, S), (B, S), (1, R), (1, L), (B, S))\}$$
$$\delta(q_{15}, B, B, 1, B, B) = \{(q_{16}, (B, S), (B, S), (1, R), (B, S), (B, S))\}$$
$$\delta(q_{16}, B, B, 1, B, B) = \{(q_{16}, (B, S), (B, S), (1, R), (B, S), (B, S))\}$$
$$\delta(q_{16}, B, B, 0, B, B) = \{(q_{17}, (B, S), (B, S), (0, R), (B, S), (B, L))\}$$
$$\delta(q_{15}, B, B, 0, B, B) = \{(q_{17}, (B, S), (B, S), (0, R), (B, S), (B, L))\}$$
$$\delta(q_{17}, B, B, 1, B, 1) = \{(q_{17}, (B, S), (B, S), (1, R), (B, S), (1, L))\}$$
$$\delta(q_{17}, B, B, 0, B, 1) = \{(q_F, (B, S), (B, S), (0, S), (B, S), (1, S))\}$$

4. (Enotračni deterministični) Turingov stroj kodiramo kot $C_1 11 C_2 11 \ldots C_{n-1} 11 C_n$, kjer so C_1, C_2, $\ldots$, C_n kode posameznih prehodov. Prehod

$$\delta(q_i, X_j) = (q_k, X_l, D_m)$$

kodiramo kot $0^i 10^j 10^k 10^l 10^m$, pri čemer je q_1 začetno stanje, q_2 edino končno stanje ter $X_1 = 0$, $X_2 = 1$, $X_3 = B$, $D_1 = L$ in $D_2 = R$. Naloga torej zahteva, da definiramo Turingov stroj, ki preveri, ali v podani kodi Turingovega stroja vsaj enkrat velja $i \neq 2$ in $k = 2$. Jezik tega problema je celo regularen, saj bi lahko zapisali regularni izraz, Turingov stroj pa je seveda prav tako mogoče definirati. Enotračni deterministični stroj bo povsem dovolj, zaradi enostavnosti pa bomo predpostavili, da je koda na vhodu pravilne oblike.

Naj stanja q_{sA}, q_{sB} oziroma q_{sC} (pri poljubnem s) predstavljajo situacije, ko velja $i = 1$, $i = 2$ oziroma $i \geqslant 3$, stanja q_{3tA}, q_{3tB} oziroma q_{3tC} (pri poljubnem t) pa situacije, ko velja $k = 1$, $k = 2$ oziroma $k \geqslant 3$. V končno stanje torej preidemo iz stanj q_{3AB} in q_{3CB}, ki ustrezajo situacijam, ko velja $i \neq 2$ in $k = 2$.

$$\delta(q_0, 0) = (q_{1A}, 0, R)$$
$$\delta(q_{1A}, 0) = (q_{1B}, 0, R)$$
$$\delta(q_{1B}, 0) = (q_{1C}, 0, R)$$
$$\delta(q_{1C}, 0) = (q_{1C}, 0, R)$$
$$\delta(q_{1A}, 1) = (q_{2A}, 1, R)$$
$$\delta(q_{1B}, 1) = (q_{2B}, 1, R)$$
$$\delta(q_{1C}, 1) = (q_{2C}, 1, R)$$
$$\delta(q_{2A}, 0) = (q_{2A}, 0, R)$$
$$\delta(q_{2B}, 0) = (q_{2B}, 0, R)$$
$$\delta(q_{2C}, 0) = (q_{2C}, 0, R)$$
$$\delta(q_{2A}, 1) = (q_{3A}, 1, R)$$
$$\delta(q_{2B}, 1) = (q_{3B}, 1, R)$$
$$\delta(q_{2C}, 1) = (q_{3C}, 1, R)$$

$$\begin{aligned}
\delta(q_{3A}, 0) &= (q_{3AA}, 0, R)\\
\delta(q_{3C}, 0) &= (q_{3CA}, 0, R)\\
\delta(q_{3AA}, 0) &= (q_{3AB}, 0, R)\\
\delta(q_{3AB}, 0) &= (q_{3AC}, 0, R)\\
\delta(q_{3AC}, 0) &= (q_{3AC}, 0, R)\\
\delta(q_{3CA}, 0) &= (q_{3CB}, 0, R)\\
\delta(q_{3CB}, 0) &= (q_{3CC}, 0, R)\\
\delta(q_{3CC}, 0) &= (q_{3CC}, 0, R)\\
\delta(q_{3B}, 0) &= (q_{3B}, 0, R)\\
\delta(q_{3AB}, 1) &= (q_F, 1, R)\\
\delta(q_{3CB}, 1) &= (q_F, 1, R)\\
\delta(q_{3AA}, 1) &= (q_4, 1, R)\\
\delta(q_{3CA}, 1) &= (q_4, 1, R)\\
\delta(q_{3AC}, 1) &= (q_4, 1, R)\\
\delta(q_{3CC}, 1) &= (q_4, 1, R)\\
\delta(q_{3B}, 1) &= (q_4, 1, R)\\
\delta(q_4, 0) &= (q_4, 0, R)\\
\delta(q_4, 1) &= (q_5, 1, R)\\
\delta(q_5, 0) &= (q_5, 0, R)\\
\delta(q_5, 1) &= (q_6, 1, R)\\
\delta(q_6, 1) &= (q_0, 1, R)
\end{aligned}$$

5. Ker lahko vsakemu Turingovemu stroju s pomočjo kodiranja priredimo naravno število, vsakemu naravnemu številu pa Turingov stroj (števila, ki pripadajo neveljavnim kodam, predstavljajo Turingove stroje brez prehodov), je vseh Turingovih strojev številno neskončno mnogo. Po drugi strani pa je vsak jezik neka podmnožica množice $\mathbb{N}$, zato je množica vseh možnih jezikov enakomočna množici $2^{\mathbb{N}}$. To pomeni, da je jezikov neštevno neskončno mnogo. Zato mora obstajati jezik, ki nima Turingovega stroja, ki bi ga sprejemal.

   Če želimo poiskati tak jezik, si pomagamo z diagonalizacijo. Naj bo $\{M_1, M_2, M_3, \ldots\}$ množica vseh Turingovih strojev (stroj M_i pripada številu i), $\{w_1, w_2, w_3, \ldots\}$ pa množica vseh besed (beseda w_j pripada številu j). Pripravimo si tabelo, v kateri vrstice predstavljajo stroje $M_1, M_2, \ldots$, stolpci pa besede $w_1, w_2, \ldots$ Naj bo element tabele v i-ti vrstici in j-tem stolpcu (a_{ij}) določen takole:

$$a_{ij} = \begin{cases} 1, & \text{če } w_j \in L(M_i);\\ 0, & \text{če } w_j \notin L(M_i). \end{cases}$$

   Definirajmo jezik $L_d = \{w_k \mid w_k \notin L(M_k)\}$ (*diagonalni* jezik). Če za jezik L_d obstaja Turingov stroj, potem se mora ta stroj nahajati v neki vrstici tabele (recimo v D-ti), saj so v tabeli našteti *vsi* Turingovi stroji. Sedaj se vprašajmo, ali w_D (beseda v D-tem stolpcu) pripada jeziku tega stroja (torej stroja M_D):

   - Če $w_D \in L(M_D)$, potem $w_D \notin L_d$ (po definiciji jezika L_d) in zato $w_D \notin L(M_D)$ (ker je po predpostavki $L(M_D) = L_d$). Kot vidimo, dobimo protislovje.
   - Če $w_D \notin L(M_D)$, potem $w_D \in L_d$ in zato $w_D \in L(M_D)$. Tudi v tem primeru nastane protislovje.

Ker obe možnosti vodita do protislovja, zaključimo, da ne obstaja Turingov stroj, ki bi sprejemal diagonalni jezik.

6. (a) Če je jezik L odločljiv, obstaja Turingov stroj, ki ga sprejema in se ustavi pri vseh mogočih vhodnih besedah — bodisi v končnem bodisi v nekončnem stanju. Če končna stanja spremenimo v nekončna in obratno, dobimo Turingov stroj, ki se še vedno ustavi pri vseh mogočih vhodnih besedah, sprejema pa komplement jezika L.

   (b) Naj bosta jezika L_1 in L_2 odločljiva in naj bo M_1 stroj za jezik L_1, M_2 pa stroj za jezik L_2. Stroj M za jezik $L_1 \cup L_2$ deluje tako, da na podani vhodni besedi najprej požene stroj M_1. Če stroj M_1 besedo sprejme, jo sprejme tudi stroj M (se ustavi v končnem stanju), če je ne, pa stroj M požene še stroj M_2 in besedo sprejme natanko v primeru, če jo sprejme M_2. Stroj M torej sprejema jezik $L_1 \cup L_2$. Ker se stroja M_1 in M_2 ustavita na vseh vhodnih besedah, velja to tudi za stroj M. Jezik $L_1 \cup L_2$ je torej odločljiv.

   (c) Ideja iz prejšnje alineje v primeru polodločljivih jezikov ne deluje, saj se lahko zgodi, da se stroj M_1 na podani besedi sploh ne ustavi. Lahko pa stroja M_1 in M_2 na vhodni besedi poganjamo vzporedno (npr. vsakega na svojem traku) in besedo sprejmemo, brž ko jo sprejme eden od njiju.

   (d) Turingov stroj M' naj deluje tako, da na podani vhodni besedi w vzporedno požene stroj M za jezik L in stroj $\overline{M}$ za jezik $\overline{L}$. Če beseda w pripada jeziku L, jo bo prej ali slej sprejel stroj M (tedaj jo sprejme tudi stroj M'), v nasprotnem primeru pa jo bo prej ali slej sprejel stroj $\overline{M}$ (tedaj jo stroj M' zavrne). Stroj M' torej sprejema jezik L, obenem pa se ustavi na vseh mogočih vhodnih besedah.

7. Primer takega jezika je $\overline{L_d} = \{w_k \mid w_k \in L(M_k)\}$. Jezik je polodločljiv, saj lahko zgradimo Turingov stroj M, ki na podani vhodni besedi w_k simulira stroj M_k. Če beseda w_k pripada jeziku stroja M_k, jo bo M_k prej ali slej sprejel. Takrat jo sprejme tudi stroj M.

   Če bi bil jezik $\overline{L_d}$ odločljiv, bi bil odločljiv tudi njegov komplement (L_d), za katerega pa smo dokazali, da ni niti polodločljiv (če bi bil L_d polodločljiv, bi iz polodločljivosti $\overline{L_d}$ sledilo, da sta oba odločljiva).

11 Prevedbe

1. Če je jezik L odločljiv, obstaja Turingov stroj M, ki sprejme jezik L in se vedno ustavi. Stroj za jezik $\overline{L}$ pa lahko deluje tako, da na svoji vhodni besedi w požene stroj M in reče »da«, če stroj M reče »ne«, in »ne«, če stroj M reče »da«. Opisani stroj sprejme jezik $\overline{L}$ se ustavi na vseh vhodnih besedah, zato je jezik $\overline{L}$ odločljiv.

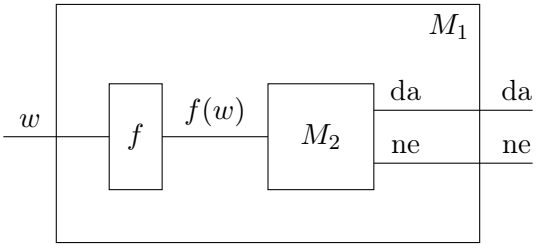
2. Stroj M za jezik L, ki se vedno ustavi, lahko deluje tako, da na vhodni besedi w vzporedno požene stroj za jezik L in stroj za jezik $\overline{L}$. Če stroj za jezik L reče »da«, potem stroj M reče »da«, če pa »da« reče stroj za jezik $\overline{L}$, potem stroj M reče »ne«. Ker vsaka beseda pripada bodisi jeziku L bodisi jeziku $\overline{L}$, se stroj M vedno ustavi.

   Ker je jezik L odločljiv, je po trditvi iz prejšnje naloge odločljiv tudi jezik $\overline{L}$.

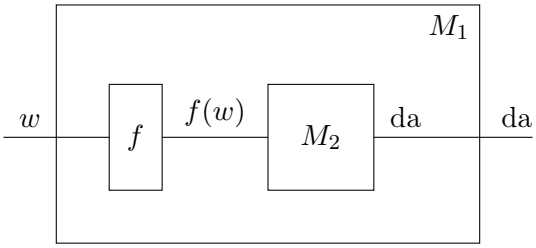
3. Če je jezik L_1 prevedljiv na jezik L_2, potem obstaja funkcija $f\colon \Sigma^* \to \Sigma^*$, tako da velja

   - $w \in L_1 \implies f(w) \in L_2$;
   - $w \notin L_1 \implies f(w) \notin L_2$.

Naj bo jezik L_1 neodločljiv. Za potrebe dokaza s protislovjem predpostavimo, da je jezik L_2 odločljiv. Če to drži, obstaja stroj M_2, ki sprejme jezik L_2 in se vedno ustavi. Če pa tak stroj obstaja, se izkaže, da obstaja tudi stroj M_1, ki sprejme jezik L_1 in se vedno ustavi. Res je: stroj M_1 deluje tako, da preslika svojo vhodno besedo w v besedo $f(w)$, nato pa na njej požene stroj M_2. Če M_2 reče »da«, potem M_1 reče »da«, če M_2 reče »ne«, pa to stori tudi M_1. Stroj M_1 potemtakem sprejme jezik L_1 in se vedno ustavi, zato je jezik L_1 odločljiv. Ta zaključek pa je v protislovju z izhodiščno predpostavko, da je jezik L_1 neodločljiv.



Na enak način dokažemo tudi drugo trditev: s pomočjo hipotetičnega stroja za jezik L_2 zgradimo stroj za jezik L_1, le da pri tem ne zahtevamo, da se stroj ustavi tudi pri vhodih, ki ne pripadajo jeziku.



4. Naj bo w_i ($i \in \mathbb{N}$) i-ta beseda nad abecedo $\{0,1\}$ (npr. $w_0 = \varepsilon$, $w_1 = 0$, $w_2 = 1$, $w_3 = 00$, $w_4 = 01$, $w_5 = 10$, $w_6 = 11$, $w_7 = 000$ itd.), M_i pa Turingov stroj s kodo w_i. Definirajmo *diagonalni jezik*:

$$L_d = \{w_i \mid w_i \notin L(M_i)\}.$$

Trdimo, da diagonalni jezik nima svojega Turingovega stroja. Za potrebe dokaza predpostavimo, da ga ima in da gre za stroj M_k s kodo w_k. Sedaj pa se vprašajmo, ali beseda w_k pripada jeziku stroja M_k. Če je $w_k \in L(M_k)$, potem je po definiciji $w_k \notin L_d$, zato velja $w_k \notin L(M_k)$, saj je M_k stroj za jezik L_d. Iz $w_k \notin L(M_k)$ pa sledi $w_k \in L_d$ in od tod $w_k \in L(M_k)$. Ker v obeh primerih pridelamo protislovje, je edini možni zaključek ta, da stroj za diagonalni jezik ne obstaja.

5. Trditev dokažemo tako, da jezik L_d prevedemo na jezik $\overline{L_u}$. Poiskati moramo funkcijo $f\colon w \mapsto \langle M', w' \rangle$, tako da velja

   - $w \in L_d \implies \langle M', w' \rangle \in \overline{L_u}$;
   - $w \notin L_d \implies \langle M', w' \rangle \notin \overline{L_u}$

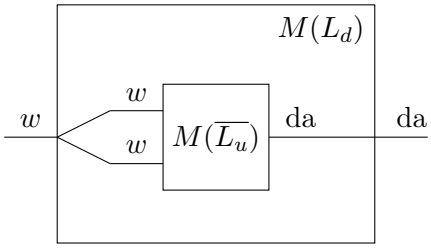
   oziroma (po definiciji jezikov L_d in $\overline{L_u}$)

   - $w \notin L(w) \implies w' \notin L(M')$; [*sic*[1]]

[1] w je tako beseda kot opis stroja.

- $w \in L(w) \implies w' \in L(M')$.

Takšno funkcijo zlahka zgradimo: vzamemo $w' = w$ in $M' = w$ oziroma $f(w) = (w, w)$. Ker smo našli prevedbo $L_d \to \overline{L_u}$, bi lahko s pomočjo stroja za $\overline{L_u}$ zgradili stroj za L_d:



Ker stroj za L_d ne obstaja, lahko zaključimo, da stroj za $\overline{L_u}$ prav tako ne obstaja. Od tod sledi, da jezik $\overline{L_u}$ ni polodločljiv.

6. Jezik L_u je polodločljiv, ker obstaja Turingov stroj M_u, ki ga sprejme. Stroj M_u (tj. univerzalni Turingov stroj) požene vhodni stroj M na vhodni besedi w in reče »da«, če to stori stroj M. Ker $\overline{L_u}$ ni polodločljiv, jezik L_u ni odločljiv.

7. Jezik L_{ne} je polodločljiv. Turingov stroj zanj (M_{ne}) deluje tako, da sistematično tvori pare (i, j) za i, $j \in \mathbb{N}$ (npr. po naraščajoči vsoti $i + j$, v okviru iste vsote pa po naraščajočih vrednostih i — torej $(0,0)$, $(0,1)$, $(1,0)$, $(0,2)$, $(1,1)$, $(2,0)$, $(0,3)$, $(1,2)$, $(2,1)$, $(3,0)$ ...) in za vsak par (i, j) požene vhodni stroj M na besedi w_i, pri čemer ga omeji na j korakov. Če stroj M reče »da«, potem to stori tudi stroj M_{ne}. Če jezik $L(M)$ vsebuje vsaj eno besedo, jo bo stroj M_{ne} na ta način gotovo našel.

   Neodločljivost jezika L_{ne} dokažemo s prevedbo $L_u \to L_{ne}$. Poiskati moramo funkcijo $f : \langle M, w\rangle \mapsto \langle M'\rangle$, tako da velja

   - $\langle M, w\rangle \in L_u \implies \langle M'\rangle \in L_{ne}$;
   - $\langle M, w\rangle \notin L_u \implies \langle M'\rangle \notin L_{ne}$

   oziroma

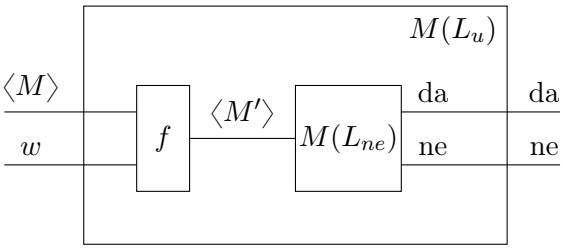
   - $w \in L(M) \implies L(M') \neq \varnothing$;
   - $w \notin L(M) \implies L(M') = \varnothing$.

   Opišimo, kako zgradimo stroj M' za podani par $\langle M, w\rangle$. Naj bo x vhod stroja M'. Stroj M' požene stroj M na besedi w. Če M sprejme w, potem M' sprejme svoj vhod (x). Jezik stroja M' je torej

$$L(M') = \begin{cases} \Sigma^*, & \text{če } w \in L(M); \\ \varnothing, & \text{če } w \notin L(M). \end{cases}$$

   Stroj M' zadošča pogojem, ki jih določa prevedba: v primeru $w \in L(M)$ velja $L(M') \neq \varnothing$, v primeru $w \notin L(M)$ pa $L(M') = \varnothing$.

   Če bi za jezik L_{ne} obstajal stroj, ki se vedno ustavi, bi lahko zgradili tudi stroj za jezik L_u, ki se vedno ustavi:



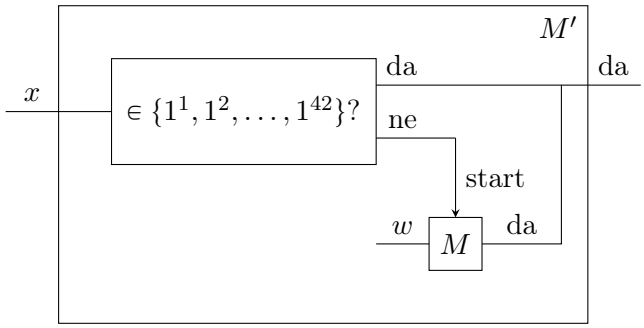
Ker za jezik L_u ne obstaja stroj, ki se vedno ustavi, to velja tudi za jezik L_{ne}. Jezik L_{ne} potemtakem ni odločljiv.

8. Ker je jezik L_{ne} podlodločljiv, a neodločljiv, jezik L_e (njegov komplement) ni podlodločljiv. Trditev lahko dokažemo tudi s prevedbo $\overline{L_u} \to L_e$. Postopek je podoben prevedbi $L_u \to L_{ne}$.

9. Poiščimo prevedbo $\overline{L_u} \to L_{42}$. Prevedbena funkcija f mora par $\langle M, w \rangle$ preslikati v opis stroja M', tako da bo veljalo

   - $w \notin L(M) \implies |L(M')| = 42$;
   - $w \in L(M) \implies |L(M')| \neq 42$.

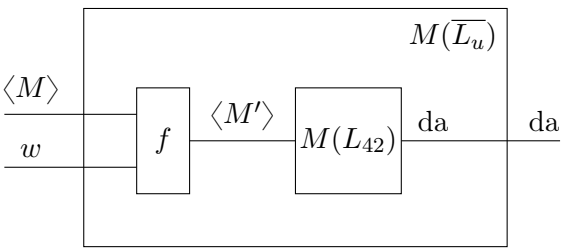
   Stroj M' lahko sestavimo tako, kot prikazuje sledeča slika:



   Stroj M' torej najprej preveri, ali njegov vhod x pripada množici $A = \{1, 1^2, \ldots, 1^{42}\}$. Če $x \in A$, potem M' reče »da«, sicer pa M' požene stroj M na besedi w; če M reče »da«, to stori tudi M'. Jezik stroja M' je potemtakem

$$L(M') = \begin{cases} \Sigma^*, & \text{če } w \in L(M); \\ A, & \text{če } w \notin L(M). \end{cases}$$

   Torej je $|L(M')| = 42$ natanko tedaj, ko $w \notin L(M)$. Funkcija f je torej res prevedba jezika $\overline{L_u}$ na jezik L_{42}. Če bi obstajal stroj za jezik L_{42}, bi z njegovo pomočjo lahko zgradili tudi stroj za jezik $\overline{L_u}$:

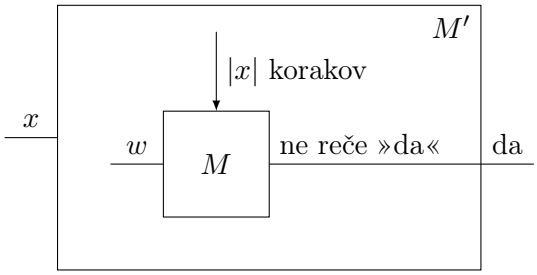


Ker stroj za jezik $\overline{L_u}$ ne obstaja, stroj za jezik L_{42} prav tako ne obstaja. Jezik L_{42} torej ni polodločljiv.

10. Poiščemo prevedbo $\overline{L_u} \to L_*$. Prevedbena funkcija f mora torej preslikati par $\langle M, w\rangle$ v opis stroja M', tako da bo veljalo

    - $w \notin L(M) \implies L(M') = \Sigma^*$;
    - $w \in L(M) \implies L(M') \neq \Sigma^*$.

    Stroj M' požene stroj M na besedi w in ga omeji na $|x|$ korakov, kjer je x vhod v stroj M'. Če stroj M v $|x|$ korakih *ne* reče »da«, potem M' reče »da«:



    Če $w \notin L(M)$, potem M nikoli (ne glede na $|x|$) ne reče »da«, zato bo stroj M' pri vseh vhodih x rekel »da«. V primeru $w \in L(M)$ pa stroj M sprejme besedo w; recimo, da za to potrebuje k korakov. To pomeni, da bo stroj M' rekel »da« samo pri besedah x, krajših od k. Potemtakem:

$$L(M') = \begin{cases} \Sigma^*, & \text{če } w \notin L(M); \\ \{x \in \Sigma^* \mid |x| < k\}, & \text{če } w \in L(M). \end{cases}$$

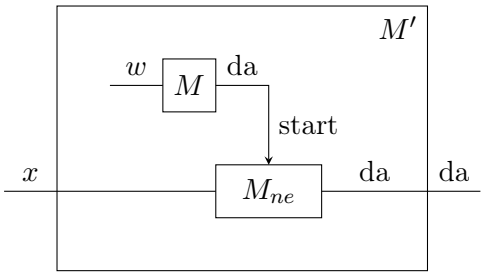
    Jezik $\overline{L_u}$ smo uspešno prevedli na jezik L_*, zato lahko zaključimo, da jezik L_* ni polodločljiv.

11. Ker je jezik vsakega Turingovega stroja polodločljiv, jezik L_{re} vsebuje vse opise Turingovih strojev. Ker vsak dvojiški niz predstavlja nek Turingov stroj, je $L_{re} = \Sigma^*$. Jezik L_{re} je torej odločljiv.

12. Poiščemo prevedbo $\overline{L_u} \to L_r$, torej funkcijo $f\colon \langle M, w\rangle \mapsto \langle M'\rangle$, tako da velja

    - $w \notin L(M) \implies L(M')$ je odločljiv;
    - $w \in L(M) \implies L(M')$ ni odločljiv.

    Stroj M' požene stroj M na besedi w. Če stroj M reče »da«, potem stroj M' preveri, ali njegov vhod (x) pripada jeziku L_{ne}:



Velja

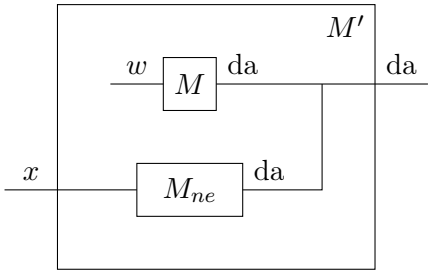
$$L(M') = \begin{cases} L_{ne}, & \text{če } w \in L(M); \\ \varnothing, & \text{če } w \notin L(M). \end{cases}$$

Ker je jezik L_{ne} neodločljiv, jezik $\varnothing$ pa odločljiv, je funkcija f res prevedba jezika $\overline{L_u}$ na jezik L_r.

13. Poiščemo prevedbo $\overline{L_u} \to L_{nr}$, torej funkcijo $f\colon \langle M, w\rangle \mapsto \langle M'\rangle$, tako da velja

- $w \notin L(M) \implies L(M')$ ni odločljiv;
- $w \in L(M) \implies L(M')$ je odločljiv.

Stroj M' vzporedno požene stroj M na besedi w in stroj za jezik L_{ne} na svojem vhodu (x). Če eden ali drugi stroj reče »da«, to stori tudi stroj M':



Velja

$$L(M') = \begin{cases} \Sigma^*, & \text{če } w \in L(M); \\ L_{ne}, & \text{če } w \notin L(M). \end{cases}$$

Ker je jezik Σ^* odločljiv, jezik L_{ne} pa neodločljiv, je funkcija f res prevedba jezika $\overline{L_u}$ na jezik L_{nr}.

14. Lahko poiščemo prevedbo $\overline{L_u} \to L$, še enostavneje pa bo najti prevedbo $L_e \to L$. Iščemo torej funkcijo $f\colon \langle M\rangle \mapsto \langle M_1, M_2\rangle$, tako da bo

- $L(M) = \varnothing \implies L(M_1) \cap L(M_2) = \varnothing$;
- $L(M) \neq \varnothing \implies L(M_1) \cap L(M_2) \neq \varnothing$.

Vzamemo $M_1 = M_2 = M$ (torej $f(M) = \langle M, M\rangle$) in smo na konju!

15. Jezik L ni niti odločljiv niti polodločljiv. Neodločljivost dokažemo s prevedbo $L_{ne} \to L$: poiščemo funkcijo $f\colon \langle M\rangle \mapsto \langle M_1, M_2\rangle$, tako da bo

- $L(M) \neq \varnothing \implies L(M_1)\backslash L(M_2) \neq \varnothing$;
- $L(M) = \varnothing \implies L(M_1)\backslash L(M_2) = \varnothing$.

Nastavimo $M_1 = M$, za M_2 pa vzamemo stroj, ki ne sprejme ničesar. Pri takšni izbiri bo $L(M_1)\backslash L(M_2) = L(M)\backslash\varnothing = L(M)$, ta jezik pa bo seveda prazen natanko tedaj, ko bo prazen jezik $L(M)$.

Da jezik L ni niti polodločljiv, pa dokažemo s prevedbo $\overline{L_u} \to L$. Iščemo funkcijo $f\colon \langle M, w\rangle \mapsto \langle M_1, M_2\rangle$, tako da bo

- $w \notin L(M) \implies L(M_1)\backslash L(M_2) \neq \varnothing$;
- $w \in L(M) \implies L(M_1)\backslash L(M_2) = \varnothing$.

Stroj M_1 sestavimo tako, da reče »da« pri poljubnem vhodu, stroj M_2 pa tako, da požene stroj M na besedi w in reče »da«, če to stori stroj M. Velja torej $L(M_1) = \Sigma^*$ in

$$L(M_2) = \begin{cases} \Sigma^*, & \text{če } w \in L(M); \\ \varnothing, & \text{če } w \notin L(M). \end{cases}$$

Zlahka preverimo, da jezik $L(M_1) \backslash L(M_2)$ izpolnjuje pogoje, ki jih narekuje prevedba.

16. Jezik L je polodločljiv, ker lahko zgradimo Turingov stroj M_L, ki se ustavi na vseh njegovih besedah. Stroj M_L na enem od trakov simulira stroj M na besedi w_1, na drugem pa vzporedno izvaja algoritem CYK nad gramatiko G in besedo w_2. Če M sprejme w, algoritem CYK pa dá nikalen odgovor, M_L sprejme svoj vhod.

    Jezik L ni odločljiv, ker obstaja prevedba $L_u \to L$. Prevedbena funkcija preslika par $\langle M, w \rangle$ v četverko $\langle M, w, G, \varepsilon \rangle$, kjer je gramatika G sestavljena iz produkcije $S \to 0$. Ker velja $\varepsilon \notin L(G)$, velja tudi $\langle M, w \rangle \in L_u \iff \langle M, w, G, \varepsilon \rangle \in L$.

17. (a) Stroj se najprej sprehodi do konca vhodne besede (do znaka B, ki sledi besedi), nato pa do začetka besede. Začetek besede si označimo, da se bomo lahko na njem ustavili; če bi se ustavili na znaku B pred besedo, bi izvedli en korak preveč.

$$\begin{aligned} \delta(q_0, 0) &= (q_1, X, R) \\ \delta(q_0, 1) &= (q_1, X, R) \\ \delta(q_1, 0) &= (q_1, 0, R) \\ \delta(q_1, 1) &= (q_1, 1, R) \\ \delta(q_1, B) &= (q_2, B, L) \\ \delta(q_2, 0) &= (q_2, 0, L) \\ \delta(q_2, 1) &= (q_2, 1, L) \end{aligned}$$

    (b) Jezik L je odločljiv. Stroj M poženemo na besedi w, pri tem pa ga omejimo na $2|w|$ korakov. Če se stroj natanko po $2|w|$ korakih ustavi, potem vemo, da velja $\langle M, w \rangle \in L$. Če se ustavi že prej ali če takrat še teče, pa zaključimo, da velja $\langle M, w \rangle \notin L$. Ker je jezik L odločljiv, je tudi polodločljiv.

    (c) Jezik L je polodločljiv. Sistematično tvorimo pare (w, k) in za vsak par obema vhodnima strojema (M_1 in M_2) podtaknemo besedo w. Oba stroja omejimo na k korakov. Če velja $\langle M_1, M_2 \rangle \in L$, potem bomo na ta način gotovo našli besedo w^*, tako da se stroj M_1 na njej ustavi po k^* korakih, stroj M_2 pa zanjo potrebuje več korakov (po k^* korakih se še ne ustavi).

    Jezik L ni odločljiv. Neodločljivost lahko dokažemo s prevedbo znanega neodločljivega polodločljivega jezika na jezik L. Tak jezik je

$$L_u = \{\langle M, w \rangle \mid w \in L(M)\}.$$

    Ker je

$$L = \{\langle M_1, M_2 \rangle \mid \exists w \colon f(M_1, w) < f(M_2, w)\},$$

    lahko prevedbo zastavimo takole: preslikaj par $\langle M, w \rangle$ v par $\langle M_1, M_2 \rangle$, tako da

$$w \in L(M) \implies \exists w' \colon f(M_1, w') < f(M_2, w')$$

    in

$$w \notin L(M) \implies \forall w' \colon f(M_1, w') \geqslant f(M_2, w').$$

    Pri podanem paru $\langle M, w \rangle$ definiramo stroj M_1 tako, da požene stroj M na besedi w (ne glede na svoj vhod) in

- se ustavi v končnem stanju (reče »da«), če to stori stroj M;
- se zacikla, če se stroj M ustavi v nekončnem stanju (reče »ne«).

(Če se stroj M ne ustavi, se stroj M_1 seveda prav tako ne bo.) Stroj M_2 pa definiramo preprosto tako, da se zacikla ne glede na njegov vhod.

Če $w \in L(M)$, se bo stroj M_1 vsaj pri eni vhodni besedi (v resnici celo pri vseh, ker je od njih neodvisen) prej ali slej ustavil, stroj M_2 pa bo veselo tekel naprej. Če $w \notin L(M)$, pa bo M_1 na vseh vhodih izvršil vsaj toliko korakov kot M_2 (v resnici se bosta oba zaciklala pri vseh vhodih).