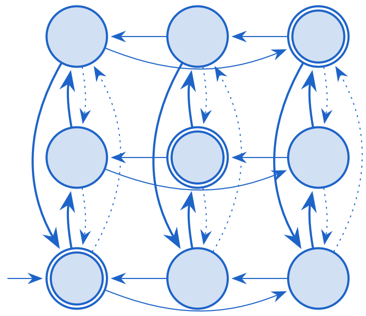
1. Zijn de volgende talen (a) regulier of niet, (b) contextvrij of niet, en (c) beslisbaar of niet?

1. $\{w \in \{a,b,c\}^* : 2\,\#_a(w) \bmod 3 = \big(\#_b(w) - \#_c(w)\big) \bmod 3\}$,

   Deze taal is regulier. Een eindigetoestandsautomaat met negen toestanden volstaat: stel $K = \{0,1,2\} \times \{0,1,2\}$, $s = (0,0)$, $A = \{(0,0),(1,1),(2,2)\}$, en voor alle $(x,y) \in K$

   $$\begin{aligned} \delta((x,y),a) &= (x+2 \bmod 3, y), \\ \delta((x,y),b) &= (x, y+1 \bmod 3), \\ \delta((x,y),c) &= (x, y-1 \bmod 3). \end{aligned}$$

   Zo houden we de waarde van $2\,\#_a \bmod 3$ in de eerste component bij en van $\#_b - \#_c \bmod 3$ in de tweede component.

   Aangezien de taal regulier is, is ze ook contextvrij en beslisbaar.

   Opmerking: zoals opgemerkt door enkele studenten, lukt het ook met slechts drie toestanden, door de waarde $2\,\#_a(w) - \#_b(w) + \#_c(w) \bmod 3$ *bij te houden.*

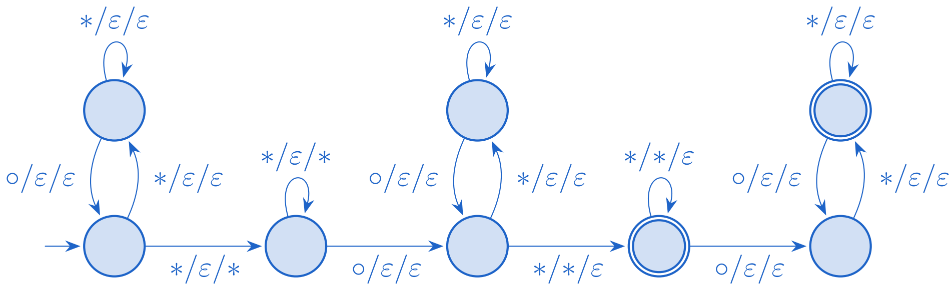
2. $\left\{ w \in \{a,b,\circ\}^* : \begin{array}{l} w = w_1 \circ w_2 \circ \cdots \circ w_n \text{ met elke } w_i \in \{a,b\}^+ \\ \text{en er bestaan } 1 \le i \ne j \le n \text{ waarbij } w_i = w_j^{\mathsf{R}} \end{array} \right\}$,

   Deze taal is niet regulier. Veronderstel van wel, dan is ook de doorsnede met de reguliere taal $\mathcal{L}(a^* \circ a^*)$ regulier, en deze doorsnede is precies de taal $\{a^n \circ a^n : n \ge 1\}$. Pas het pumping lemma toe op het woord $w = a^k \circ a^k$ om een ontbinding $w = xyz$ te krijgen zoals in het pumping lemma. De eigenschap dat $|xy| \le k$ betekent dat $y = a^{|y|}$, waaruit $w_2 = a^{k+|y|} \circ a^k$ niet langer in de taal zit, een strijdigheid.

   De taal is wel contextvrij, zoals volgende grammatica aantoont. Q genereert een enkel deelwoord in $\{a,b\}^+$, P genereert een reeks van dergelijke deelwoorden gescheiden door $\circ$'s, en C genereert de vereiste twee woorden w_i en w_j met $w_i = w_j^{\mathsf{R}}$.

   $$\begin{cases} S \to P \circ C \circ P \mid C \circ P \mid P \circ C \mid C \\ C \to a\,D\,a \mid b\,D\,b \\ D \to a\,D\,a \mid b\,D\,b \mid \circ\,P\,\circ \mid \circ \\ P \to Q \mid P \circ Q \\ Q \to a\,R \mid b\,R \\ R \to a\,R \mid b\,R \mid \varepsilon \end{cases}$$

   Of een pushdownautomaat (waarbij bijvoorbeeld $*/*/\varepsilon$ de transities $a/a/\varepsilon$ en $b/b/\varepsilon$ afkort):



Aangezien de taal contextvrij is, is ze ook beslisbaar.

3. $\{\langle M\rangle : M$ stelt een Turingmachine voor met $|\mathcal{L}(M)| \leq 2018$ en $|\langle M\rangle| \leq 2018\}$.

   Ieder woord in de taal is absoluut begrensd in lengte; de taal is dus eindig en triviaal regulier, contextvrij en beslisbaar.

Bewijs telkens je antwoord.

2. Beschouw de volgende operator `Behead`, die van elk woord in de taal de eerste letter afhaalt:

$$\texttt{Behead}(\mathcal{L}) = \{w \in \Sigma^* : (\exists x \in \Sigma)(xw \in \mathcal{L})\}.$$

Bewijs dat het beeld $\texttt{Behead}(\mathcal{L})$ van een reguliere taal $\mathcal{L}$ nog steeds regulier is.

We geven een constructie voor een automaat M' vanuit een eindigetoestandsautomaat M voor $\mathcal{L}$.

Merk op dat het niet volstaat elke transitie vanuit de starttoestand te vervangen door een ε-transitie, want mogelijks zijn er paden die terugkeren langs deze toestand. We voegen daarom een nieuwe starttoestand s' toe en definiëren de ε-transities $\delta(s', \varepsilon) = \{\delta(s, x) : x \in \Sigma\}$ vanuit s' (de originele transities ongewijzigd latend). Noteer $\mathcal{L}'$ voor de taal aanvaard door deze gemodificeerde automaat, die niet-deterministisch is; we tonen nu aan dat $\mathcal{L}' = \texttt{Behead}(\mathcal{L})$.

Beschouw een woord $w \in \mathcal{L}'$. Het gegeven dat de nieuwe automaat dit woord aanvaardt, betekent dat er een aanvaardend pad bestaat vanuit s'. De eerste stap in dit pad moet zo'n nieuw toegevoegde ε-transitie naar een zekere toestand $\delta(s, x)$ zijn (voor zekere $x \in \Sigma$). Er zijn geen toestanden terug naar s', dus vervolgens wordt w volledig verwerkt binnen de originele automaat M in M':

$$(s', w) \Rightarrow_{M'} (\delta(s, x), w) \Rightarrow^*_{M'} (q, \varepsilon),$$

met $q \in A$ een aanvaardende toestand. Omdat ook $(s, xw) \Rightarrow_M (\delta(s, x), w)$ binnen M volgt dat

$$(s, xw) \Rightarrow_M (\delta(s, x), w) \Rightarrow^*_M (q, \varepsilon),$$

zodat $xw \in \mathcal{L}(M) = \mathcal{L}$. Per definitie betekent dit dat $w \in \texttt{Behead}(\mathcal{L})$.

Beschouw nu een woord $w \in \texttt{Behead}(\mathcal{L})$. Dan bestaat er een letter $x \in \Sigma$ met $xw \in \mathcal{L}$, zodat

$$(s, xw) \Rightarrow_M (\delta(s, x), w) \Rightarrow^*_M (q, \varepsilon),$$

voor zekere aanvaardende toestand $q \in A$. Per constructie bestaat er vanuit de starttoestand s' van M' een ε-transitie naar $\delta(s, x)$, en zijn er vervolgens geen paden meer die passeren langs s', dus

$$(s', w) \Rightarrow_{M'} (\delta(s, x), w) \Rightarrow^*_{M'} (q, \varepsilon).$$

We zien dat $w \in \mathcal{L}'$, en het gestelde is bewezen.

3. Toon aan dat de volgende taal (a) wel semibeslisbaar maar (b) niet beslisbaar is:

$$\left\{ \langle M_1, M_2\rangle : \begin{array}{l} M_1 \text{ en } M_2 \text{ zijn twee Turingmachines waarvoor een input } w \text{ bestaat} \\ \text{die door } M_1 \text{ aanvaard en door } M_2 \text{ geweigerd wordt, of omgekeerd} \end{array} \right\}.$$

Maak voor (b) gebruik van een expliciete reductie, zonder te steunen op de stelling van Rice.

We tonen eerst aan dat deze taal semibeslisbaar is. Input die niet van de vorm $\langle M_1, M_2\rangle$ is, kunnen we meteen verwerpen. Anders sommen we alle woorden in Σ^* lexicografisch op, en verifiëren we of M_1 een woord aanvaardt dat M_2 weigert of omgekeerd. Omdat het niet uitgesloten is dat een machine M_1 of M_2 in een oneindige lus belandt, gebruiken we het dovetailing-principe: som dus alle woorden $w_1, w_2, w_3, \ldots \in \Sigma^*$ lexicografisch op, en voer volgende lus uit.

- Laat M_1 en M_2 hoogstens één instructie draaien op het eerste woord w_1.

- Laat M_1 en M_2 hoogstens één instructie draaien op w_2 en hoogstens twee instructies op w_1.
- Laat M_1 en M_2 hoogstens één instructie draaien op w_3, dan hoogstens twee instructies op w_2 en hoogstens drie instructies op w_1.
- ...

We kunnen aanvaarden zodra we een woord w_i gevonden hebben waarop beide machines stoppen, terwijl de ene aanvaardt en de andere weigert.

Vervolgens tonen we aan dat de taal niet beslisbaar is, via een reductie vanuit het halting problem. Beschouw de volgende reductiemachine R. Voor input niet van de vorm $x = \langle M, w \rangle$ geeft R een vaste output $\langle M^*, M^* \rangle$ terug, met M^* een Turingmachine die alle input aanvaardt. Voor input wél van de vorm $x = \langle M, w \rangle$ stelt R coderingen $\langle M_1, M_2 \rangle$ op van machines met het volgende gedrag:

- M_1 weigert alle input;
- M_2 negeert de input, simuleert M op w, en aanvaardt (ongeacht de originele input).

Merk op dat $\langle M_1, M_2 \rangle$ tot de taal in de opgave behoort precies als M stopt op w. Voor deze machine geldt dat $R(x) = \langle M_1, M_2 \rangle \in \mathcal{L} \Leftrightarrow x \in \mathcal{H}$, dus dit geeft de gezochte reductie die aantoont dat $\mathcal{L}$ niet beslisbaar is—we weten immers dat $\mathcal{H}$ niet beslisbaar is.

4. Bewijs dat de volgende taal tot de complexiteitsklasse P behoort:

$$\left\{ \langle M \rangle : \begin{array}{l} M \text{ is een niet-deterministische eindigetoestandsautomaat} \\ \text{die oneindig veel woorden } a^n \text{ (waarbij } n \in \mathbb{N}\text{) aanvaardt} \end{array} \right\}.$$

Het kan verleidelijk zijn om M deterministisch te maken (om eenvoudiger sluitingseigenschappen zoals complementering te kunnen toepassen), maar dit kost reeds een tijd exponentieel in het aantal toestanden. We blijven dus werken met de niet-deterministische automaat M zelf.

Controleren of de input van de vorm $\langle M \rangle$ is, lukt zeker in polynomiale tijd. Nagaan of we oneindig veel woorden van de vorm a^n aanvaarden kunnen we doen door M op te vatten als gerichte graaf met de toestanden als toppen en de a-transities als bogen. Met andere woorden, bouw de graaf G op met K als toppenverzameling en $\{(p, q) : (p, a, q) \in \Delta\} \subseteq K \times K$ als (gerichte) bogen. We hoeven daartoe in feite enkel de bogen op te stellen uit Δ. Dit lukt eenvoudig in polynomiale tijd: voor elke transitie $(p, a, q) \in \Delta$ die we lezen, schrijven we een boog (p, q).

Merk nu op dat M oneindig veel woorden a^n aanvaardt als en slechts als er een pad bestaat vanuit de starttop naar een zekere tussentop, een lus rond deze tussentop, en een vervolgend pad naar een aanvaardende top. We zoeken in de graaf G dus een (gerichte) cykel met een top die bereikbaar is vanuit de starttop én met een top die naar een aanvaardende top gaat.

Overloop voor elke top v in G via een *depth-first traversal* de paden vanuit v om te bepalen of deze top v in een cykel ligt—dit kost tijd $\mathcal{O}(|K| \cdot |\Delta|) = \mathcal{O}(|\langle M \rangle|^2)$. Zo'n *depth-first traversal* vanuit de starttoestand leert ook welke toestanden allemaal bereikbaar zijn via a-transities, en toegepast op de aanvaardende toestanden maar "in de omgekeerde richting" vinden we de "productieve" toestanden die naar een aanvaardende toestand kunnen leiden (eveneens kwadratische tijd per aanvaardende toestand, dus hoogstens kubische tijd in zijn geheel). Uiteindelijk moeten we enkel nog controleren of een top zowel in een cykel ligt, bereikbaar is, en productief is.

De totale tijd voor deze aanpak is in essentie $\mathcal{O}(|\langle M \rangle|^3)$ en optimalisaties zijn zeker nog mogelijk. Een implementatie op een éénbands-Turingmachine moet dan ook lukken in polynomiale tijd.