

Automaten en Berekenbaarheid: opgave 1

Prof. B. Demoen (Bart.Demoen@cs.kuleuven.be)
W. Van Onsem (Willem.VanOnsem@cs.kuleuven.be)

5 november 2013

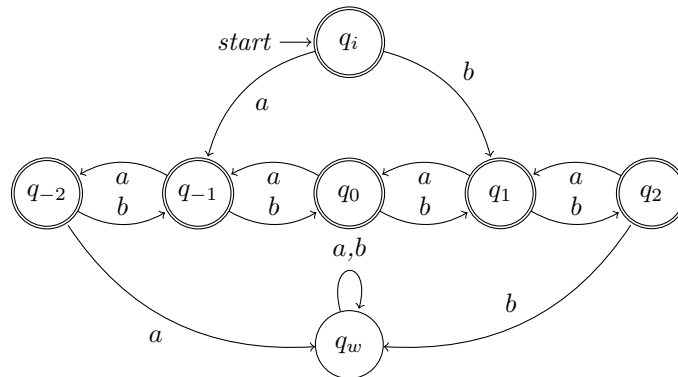
Richtlijnen Je schrijft enkel op de gekleurde bladen die je gekregen hebt: op je werkplek liggen geen andere bladen dan die gekleurde bladen en de bladen van de cursustekst. De bladen van je cursustekst waarop oplossingen van oefeningen of aanwijzingen tot oplossingen van oefeningen staan, steek je ook weg. Je jas, je boekentas, je gsm ... leg je vooraan in het auditorium, zoals gebruikelijk bij een examen. Bij het afgeven zorg je dat je naam op je bladen staat, en dat je alles aan elkaar niet. Er zijn 4 vragen.

Vraag 1 (Pompend lemma voor reguliere talen). *Indien de voorgestelde taal niet regulier is: bewijs dit aan de hand van het pompend lemma voor reguliere talen. Indien de voorgestelde taal dit wel is: geeft een reguliere expressie of een eindige toestandsautomaat die de taal beschrijft/beslist.*

1. De verzameling strings uit $\Sigma = \{a, b\}$ zodat voor elke string $s \in \Sigma^*$, $s \in L$ indien voor elke prefix¹ van s het aantal a 's en b 's in de string nooit meer dan 2 verschilt.
2. $L = \{a^m c^n b^m | m, n \in \mathbb{N} \text{ met } m > n\}$
3. $L = \{a^m b^n | m, n \in \mathbb{N} \text{ met } m < n \text{ en } m < 981\}$.
4. $L = \{wv^*w | v \in \Sigma, w \in \Sigma^*\}$ voor een willekeurige eindige Σ die uit minstens twee elementen bestaat.

Antwoord 1.1 (Pompend lemma voor reguliere talen). *We beschouwen de verschillende talen.*

1. *Regulier. Vermits de eigenschap voor iedere prefix geldt, kunnen we een transitie-eigenschap uitbuiten: een string maakt nog kans om geaccepteerd te worden indien in het verleden het verschil nooit meer dan twee bedroeg. We beschouwen dus 5 accepterende toestanden die het verschil van -2 tot 2 opslaan. Telkens wanneer een a of b wordt uitgelezen, gaan we naar een toestand die ééntje hoger of lager is. Wanneer we een a lezen bij -2 of een b bij 2 komen we in een ongeldige toestand terecht. Deze zal de overige karakters inlezen zonder in een accepterende eindtoestand te eindigen.*



¹Een prefix van s is een substring van s die begint bij het eerste karakter van s en op een willekeurige plaats k eindigt.

2. We beschouwen we de string $s = a^d b^{d-1} c^d$. Voor deze string geldt dus $|s| \geq d$ met d de pomplengte. Voor elke indeling $s = xyz$ geldt dat y uitsluitend uit a 's kan bestaan (vermits $|xy| \leq d$ en $|y| > 0$). We kunnen dus besluiten dat $y = a^k$ met k onbekend maar $0 < k \leq d$. De taal is regulier wanneer voor elke i , $xy^i z \in L$, maar $xy^2 z = a^{d+k} b^{d-1} c^d$, dit behoort duidelijk niet tot L , bijgevolg is de taal niet regulier.
3. Regulier. We construeren eerst een reguliere expressie aan de hand van een parameter i :

$$a^i b^{i+1} b^* \quad (1)$$

Vervolgens is L gelijk aan een eindige unie:

$$\bigcup_{i=0}^{980} a^i b^{i+1} b^* \quad (2)$$

Concreet ziet de reguliere expressie er dus als volgt uit:

$$bb^* \mid abb^* \mid aabbbb^* \mid aaabbbb^* \mid \dots \mid \underbrace{aaaaa \dots a}_{980 \text{ keer}} \underbrace{bbbbbb \dots b}_{981 \text{ keer}} b^* \quad (3)$$

4. Niet-regulier. Stel dat a en b twee elementen uit Σ zijn². In dat geval beschouwen we de string $s = a^d b a^d$. Voor deze string geldt dus $|s| \geq d$ met d de pomplengte. Voor elke indeling $s = xyz$ geldt dat y uitsluitend uit a 's kan bestaan (vermits $|xy| \leq d$ en $|y| > 0$). We kunnen dus besluiten dat $y = a^k$ met k onbekend maar $0 < k \leq d$. De taal is regulier wanneer voor elke i , $xy^i z \in L$, maar $xy^2 z = a^{d+k} b c^d$, dit behoort duidelijk niet tot L , bijgevolg is de taal niet regulier.

Vraag 2 (Algebra van talen). Niet alle talen behoren tot RegLan , denk bijvoorbeeld aan $\text{AGELIJKB} = \{a^n b^n \mid n \in \mathbb{N}\}$. Stel dat L_1 en L_2 verschillende niet-reguliere talen zijn; en L_3 en L_4 verschillende oneindige reguliere talen niet gelijk aan Σ^* met Σ het alfabet van de talen in kwestie. Zijn volgende talen dan regulier, niet-regulier of hangt dit af van de talen in kwestie. Bewijs of geef tegenvoorbeelden.

1. $L_1 \cap L_3$.
2. $L_1 \cup L_3$.
3. $L_1 \cup L_2$.
4. $\overline{L_1} \cup L_3$.

Antwoord 2.1 (Pompend lemma voor reguliere talen). We beschouwen de verschillende talen.

1. Hangt van de talen af:

- De doorsnede kan regulier zijn, bijvoorbeeld $L_1 = \{a^m b^n \mid m \leq n\}$ en $L_2 = \{a b^m a^n \mid m \geq n\}$, dan is $L_1 \cap L_2 = \{a a b b^*\}$ wat regulier is.
- De doorsnede kan niet-regulier zijn, bijvoorbeeld $L_1 = \{a^m b^n \mid m \leq n\}$ en $L_2 = \{a^m b^n \mid m \geq n\}$, dan is $L_1 \cap L_2 = \{a^m b^m \mid m \in \mathbb{N}\}$ wat niet-regulier is.

2. Hangt van de talen af:

- De unie kan regulier zijn, bijvoorbeeld: stel $L_1 = \{a^m b^n \mid m \geq n\}$ en $L_3 = \{a^* b^*\}$. De unie is $L_1 \cup L_3 = \{a^* b^*\}$, deze taal is regulier.
- De unie kan niet-regulier zijn, bijvoorbeeld: stel $L_1 = \{a^m b^n \mid m \geq n\}$ en $L_3 = \{a a^* b\}$. De unie is $L_1 \cup L_3 = \{a^m b^n \mid m \geq n\}$, deze taal is niet-regulier.

²Hoe de symbolen eruit zien maakt niet uit, evenmin dat er nog andere symbolen betrokken zijn.

3. Hangt van de talen af:

- De unie kan regulier zijn, bijvoorbeeld stel $L_1 = \{a^m b^n | m \geq n\}$ en $L_2 = \{a^m b^n | m \leq n\}$. De unie is $L_1 \cup L_2 = \{a^* b^*\}$, deze taal is regulier.
- De unie kan niet-regulier zijn, bijvoorbeeld: stel $L_1 = \{a^m b^n | m \geq n\}$ en $L_2 = \{a^m b^n | m > n\}$. De unie is $L_1 \cup L_2 = \{a^m b^n | m \geq n\}$, deze taal is niet-regulier.

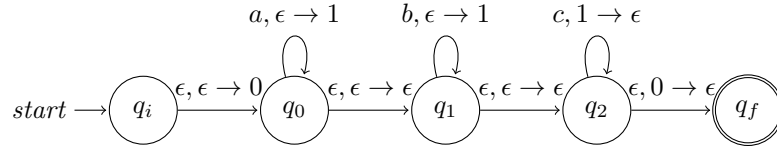
4. Hangt van de talen af: vermits het complement van een reguliere taal, regulier is, weten we dat $\overline{L_1}$ ook niet-regulier is. Bijgevolg grijpen we terug naar de voorbeelden uit 1 om aan te tonen dat het van de talen afhangt.

Vraag 3 (Pompend lemma voor contextvrije talen). Indien de voorgestelde taal niet context-vrij is: bewijs dit aan de hand van het pompend lemma voor contextvrije talen. Indien de voorgestelde taal dit wel is: geeft een reguliere expressie, een eindige toestandsautomaat, een contextvrije grammatica of een push-down automaat die de taal beschrijft/beslist.

1. $L = \{a^m b^n c^{n+m} | m, n \in \mathbb{N}\}$
2. $L = \{a^m b^n c^m d^n | m, n \in \mathbb{N}\}$
3. $L_R = \{wv\hat{w} | w \in \Sigma^* \wedge v \in R\}$ voor een willekeurige eindige Σ die uit minstens twee elementen bestaat en R een reguliere taal over Σ .

Antwoord 3.1 (Pompend lemma voor contextvrije talen). We beschouwen de verschillende talen.

1. Context-vrij. Volgend push-down automaat beslist de taal:



2. Niet context-vrij. We beschouwen de string $s = a^p b^p c^p d^p$. Voor deze string geldt dus $|s| \geq p$ met p de pomplengte. Voor elke indeling $s = wxyz$ geldt dat v en y uitsluitend uit één soort karakters kunnen bestaan, indien niet zal men bij het pompen bijvoorbeeld na een sequentie b 's terug een sequentie a 's kunnen tegenkomen. Iets wat zeker niet tot de taal behoort. Bijgevolg kunnen we stellen dat $v = q^k$ en $y = r^l$ met $q, r \in \Sigma$. Omdat $|vxy| \leq p$ weten we ook dat v en y minder dan p plaatsen van elkaar gelegen moeten zijn. Dit betekent dus dat ofwel $q = r$ en beide delen dus eenzelfde sequentie van karakters bepalen. In dat geval kan de string niet gepompt worden: wanneer we immers i verhogen ontstaat er een onbalans: stel dat we $q = r = a$ nemen dan zullen we met $i = 2$, de string $a^{p+k+l} b^p c^p d^p$ genereren, wat duidelijk niet tot L behoort. In een ander geval zijn de karakters verschillend: $q \neq r$. Stel $q = a$ en $r = b$, dan genereren we bij $i = 2$ de string $a^{p+k} b^{p+l} c^p d^p$ dit behoort opnieuw niet tot L , dit laatste geldt ook voor andere segmenten zoals bijvoorbeeld $q = b$ en $r = c$.

3. Context-vrij. R is een reguliere taal, dus kan R geschreven worden als een reguliere expressie E . We kunnen deze reguliere expressie inductief omzetten naar een contextvrije grammatica met startsymbool R_E als volgt:

- Indien $E = \epsilon$, dan $R_E \rightarrow \epsilon$;
- Indien $E = a$ met $a \in \Sigma$, dan $R_E \rightarrow a$;
- Indien $E = E_1 | E_2$ met E_1 en E_2 reguliere expressies, dan $R_E \rightarrow R_{E_1}$ en $R_E \rightarrow R_{E_2}$;
- Indien $E = E_1 E_2$ met E_1 en E_2 reguliere expressies, dan $R_E \rightarrow R_{E_1} R_{E_2}$;
- Indien $E = (E_1)^*$ met E_1 een reguliere expressie, dan $R_E \rightarrow R_{E_1} R_E$ en $R_E \rightarrow \epsilon$;

- Indien $E = \phi$, dan definiëren we geen overgangsregel voor R_E .

We bekommen dus een symbool R_E die de taal van R beschrijft. Vervolgens bouwen we volgende context-vrije grammatica:

- Voor elke $a \in \Sigma$ definiëren we de regel: $S \rightarrow aSa$.
- We voorzien ook een regel $S \rightarrow R_E$.

De finale grammatica heeft als startsymbool S .

Vraag 4 (Myhill-Nerode relaties). Gegeven de volgende twee partities:

$$P_1 = \{\{a, b\}, \{c\}, \{d\}, \{e, f, g\}, \{h\}\} \quad (4)$$

$$P_2 = \{\{a, e\}, \{b, c\}, \{d, h\}, \{f, g\}\} \quad (5)$$

Bepaal het supremum van de twee partities volgens de theorie in de cursus (sectie 13.3).

Antwoord 4.1 (Myhill-Nerode relaties). Het supremum is gedefinieerd als de fijnste relatie die grover is dan de originele relaties. De resulterende relatie is dan:

$$P = \{\{a, b, c, e, f, g\}, \{d, h\}\} \quad (6)$$