AB: gekwoteerde oefenzitting 1
30 okt 2018

1 Algebra van talen

Voor een niet-lege eindige taal L_E en een niet-reguliere contextvrije taal L_C, beiden over hetzelfde alfabet, bespreek de volgende uitspraken (altijd juist, altijd fout, soms juist, soms fout) en beargumenteer jouw antwoord.

1. $L_C \cap L_E$ is regulier

   Antwoord Altijd juist.

   L_E is eindig en de doorsnede met een eindige taal is altijd eindig, dus $L_C \cap L_E$ is eindig. Elke eindige taal is regulier (RegEx: $\bigvee_{s \in L_E} s$), dus is $L_C \cap L_E$ regulier.

2. $L_C \cup L_E$ is regulier

   Antwoord Altijd fout.

   De verzameling L van strings met lengte groter dan de langste string in L_E (p_{max}) kan niet uitgedrukt worden met een reguliere taal. Mocht dat wel zo zijn zou het pompemd lemma voor reguliere talen toepasbaar zijn met pomplengte $p \geq p_{max}$ op L_C en zou deze taal regulier zijn.

3. $L_C \cap L_E^C$ is regulier

   Antwoord Altijd fout.

   De verzameling L van strings met lengte groter dan de langste string in L_E (p_{max}) kan niet uitgedrukt worden met een reguliere taal (zie vorige vraag). Al deze strings zitten in L_E^C en omdat we de doorsnede met L_C nemen blijven er geen andere strings over die ook langer zijn dan p_{max}.

4. $L_C \cup L_E^C$ is context-vrij

   Antwoord Altijd juist.

   L_E is een eindige taal, en elke eindige taal is regulier (zie vraag 1). Het complement van een reguliere taal is terug regulier. Omdat elke reguliere taal ook een context-vrije taal is en de context-vrije talen gesloten zijn onder de unie is $L_C \cup L_E^C$ altijd context-vrij.

2 Pompend lemma

Bewijs de volgende uitspraken.

1. De taal $\{0^n10^n | n \in \mathbb{N}\}$ is niet regulier

   Antwoord M.b.v. Pompend lemma voor reguliere talen. Gegeven een pomplengte p, kies $s = 0^p10^p = xyz$. Aangezien $|xy| \leq p$ en $|y| > 0$ zit de string $xy^0z = xz = 0^{p-|y|}10^p$ niet in de taal, dus is de taal niet regulier.

2. De taal $\{w \in \{a,b,c\}^* |$ w bevat een even (=gelijk) aantal a's, b's en c's$\}$ is niet context-vrij

   Antwoord M.b.v. Pompend lemma voor context-vrije talen. Neem de doorsnede met de reguliere taal $\{a^*b^*c^*\}$, dan verkrijg je $\{a^nb^nc^n\}$. We bewijzen dat deze taal niet context-vrij is. Kies de string $s = a^pb^pc^p = xyzvw$. Omdat $|yzv| \leq p$ kan je nooit a, b en c tegelijkertijd pompen.

3 Wat voor taal?

Zijn de volgende talen 1) regulier; 2) niet regulier maar wel context-vrij; of 3) niet context-vrij. Bewijs.

1. $\{a^m b^n c^n d^m | n, m \in \mathbb{N}\}$

   Antwoord 2.

   Niet regulier want: $\{a^m b^n c^n d^m | n, m \in \mathbb{N}\} \cap \{b^* c^*\} = \{b^n c^n\}$ (doorsnede met reguliere taal) en $\{b^n c^n\}$ is niet regulier.

   Context-vrij want we kunnen een grammatica opstellen.

$$S \to aSd \mid M$$
$$M \to bMc \mid \epsilon$$

2. $\{a^n | n \in \mathbb{N}, n \text{ is oneven}, n \text{ is een veelvoud van } 3\}$

   Antwoord 1.

   Regulier want: we kunnen een reguliere expressie opstellen (een DFA kan ook gemakkelijk).

$$aaa(aaaaaa)^*$$

4 Swapped

De operatie $swap(s, i)$ krijgt als input de string $s = a_1a_2...a_n$ en een positie i en geeft als output de string $w = a_1...a_{i+1}a_i...a_n$ die gelijk is aan s behalve dat de symbolen op plaats i en $i+1$ van plaats zijn veranderd. Gegeven een taal L, kunnen we een nieuwe taal $swapped(L)$ defineren zodat

$$swapped(L) = \{w|s \in L, \exists i : 1 \leq i \leq |s| - 1 \wedge w = swap(s, i)\}$$

Voorbeeld. Indien $L = \{$leuk, toets$\}$, dan is $swapped(L) = \{$eluk, luek, leku, otets, teots, totes, toest$\}$.

1. Bewijs dat voor elke reguliere taal L de taal $L_1 = L \cup swapped(L)$ ook regulier is.

   Antwoord L is regulier, dus er bestaat een DFA $A = (Q, \Sigma, \delta, q_0, F)$ voor deze taal. Om de taal L_1 te bepalen bouwen we een NFA $N = (Q_N, \Sigma, \delta_N, q_{N0}, F_N)$ gebruikmakende van A. We gebruiken twee kopieën van A (A^1, A^2) die we combineren om N te verkrijgen. A^1 wordt gebruikt om strings zonder fout (L dus) te aanvaarden. Maar we voegen bovendien bij elke toestand bogen toe naar buffer toestanden om ook strings de aanvaarden waar twee symbolen verwisseld zijn van plaats. De buffer toestanden $Q^B = (Q, \Sigma)$ onthouden de toestand q en het symbool σ_1 wat gezien werdt en hebben een boog voor elk mogelijk symbool σ_2 die verwijst naar de juiste toestand in $A^2 : \delta(\delta(q, \sigma_2), \sigma_1)$ (de toestand waar je in terecht zou komen als σ_1 en σ_2 van plaats veranderen). A^2 aanvaardt strings waarin al een symbool verwisseld is en omdat er geen tweede keer symbolen verwisseld mogen worden zijn er geen bijkomende uitgaande bogen. De kopieën van een toestand q noemen we q^1 voor A^1 en q^2 voor A^2.

   We kunnen N dus formeel defineren als volgt:

$$\begin{aligned} Q_N &= Q^1 \cup Q^2 \cup (Q \times \Sigma) \\ \forall q^1 \in Q^1, \sigma \in \Sigma : \delta_N(q^1, \sigma) &= \{\delta(q, \sigma)^1, (q, \sigma)\} \\ \forall (q, \sigma_1) \in Q \times \Sigma, \sigma_2 \in \Sigma : \delta_N((q, \sigma_1), \sigma_2) &= \{\delta(\delta(q, \sigma_2), \sigma_1)^2\} \\ \forall q^2 \in Q^2, \sigma \in \Sigma : \delta_N(q^2, \sigma) &= \{\delta(q, \sigma)^2\} \\ q_{N0} &= q_0^1 \\ F_N &= F^1 \cup F^2 \end{aligned}$$

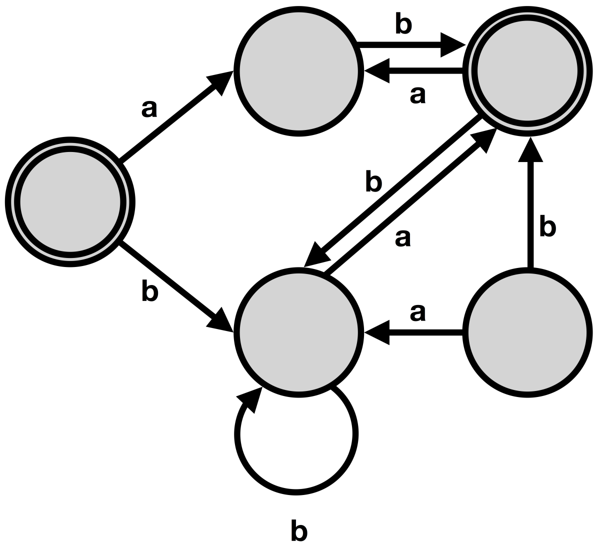
2. Wat verandert er als ik enkel de taal $swapped(L)$ wil aanvaarden?

   Antwoord Verwijder de eindtoestanden van de kopie A^1, dus met N als hierboven: $F_N = F^2$.

3. Is $swapped(swapped(L))$ nog steeds regulier?

   Antwoord $swapped(L)$ is regulier (zie boven) dus de operatie nog eens toepassen levert weer een reguliere taal op.

5 Minimale DFA



Construeer een DFA die dezelfde taal aanvaardt als de DFA in de figuur hierboven maar een minimaal aantal toestanden heeft. Toon de tussenstappen van jouw redenering.

Antwoord Volg de procedure uit de oefenzitting:

1. Verwijder nutteloze toestanden (toestand beneden rechts)
2. Vindt f-gelijke toestanden door een tabel op te stellen en f-verschillende toestanden toe te voegen, itereer tot dat de tabel niet meer verandert (de twee aanvaardende toestanden zijn f-gelijk)
3. Stel de geminimaliseerde DFA op door f-gelijke toestanden samen te voegen

Bonus: je ziet dat er geen uitgaande boog is voor a in de toestand boven-midden en je neemt de impliciete garbage-toestand mee in jouw berekeningen