

Logische Complexiteit

Context-vrije Talen

Jeroen Goudsmit

Dinsdag 21 Februari

Deze inleveropgave bestaat uit Opgave 1, 2 en 3. Ook is er een klein bonusonderdeel, het bewijzen van Lemma 1. Beargumenteer je antwoorden, maar houd de uitleg *kort en helder*. Opgave 3 is een wat lastigere opgave, hier kun je best een tijd over nadenken. Lemma 2.21 uit het boek kan hier wat inspiratie geven. Vaak helpt het ook om het bewijs eerst voor een klein voorbeeld te doen.

Opgave 1 — Rekenen

Beschouw de onderstaande grammatica over het alfabet $\Sigma = \{0, 1, +, \cdot, (,)\}$.

$$\begin{aligned} S &\rightarrow 0 A \\ S &\rightarrow 1 A \\ S &\rightarrow (S) \\ A &\rightarrow + S \\ A &\rightarrow \cdot S \\ A &\rightarrow 0 A \\ A &\rightarrow 1 A \\ A &\rightarrow \varepsilon \end{aligned}$$

- Staat deze grammatica in Chomsky normaalvorm? Zo nee, transformeer haar naar deze vorm.
- Geef een PDA die dezelfde taal herkent als de bovenstaande grammatica.
- Geef afleidingsbomen van de zinnen $01 + (1 \cdot 10)$ en $0 + 1 + 0$.
- Is deze grammatica ambigu? Zo ja, is de bijbehorende taal inherent ambigu?

Opgave 2 — Grammatica's ontwerpen

In deze opgave wordt je gevraagd een aantal context-vrije grammatica's te construeren. Ieder van de onderstaande talen is context vrij, je opdracht is om de bijbehorende context vrije grammatica te geven. Tevens vragen we je per taal aan te geven of de taal regulier is, met een korte redenering. Alle talen zijn over het alfabet $\Sigma = \{a, b\}$.

- $\{a^i b^j \mid i \text{ en } j \text{ geheel en } j < i\}$;
- $\{w \in \Sigma^* \mid w \text{ bevat meer } a\text{'s dan } b\text{'s}\}$;
- $\{w \in \Sigma^* \mid w \text{ is een palindroom}\}$

Opgave 3 — Greibach naar PDA

Zij $G = \langle V, \Sigma, R, S \rangle$ een grammatica waar elke regel van de vorm $A \rightarrow x B_1 \dots B_n$ is, met $A, B_1, \dots, B_n \in V$ en $x \in \Sigma$. Merk op dat er dus geen enkele regel is die naar ε gaat.

In deze opgave gaan we een lange PDA maken die deze grammatica als het ware simuleert. Een lange PDA is een PDA met een veralgemeniseerde transitiefunctie $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow \mathbf{P}(Q \times \Gamma^*)$ als behandeld op college, genoemd in het boek (pagina 119) en ook uitgelegd in de aantekening “context-vrije grammatica’s”.

Zet als toestanden $Q = \{q_0, q_*\}$, en neem als stackalfabet $\Gamma = V \cup \Sigma$. Er is maar een accepterende toestand, en dat is q_* . We definiëren nu δ als volgt:

$$\begin{aligned}\delta(q_0, \varepsilon, \varepsilon) &= \{ \langle q_*, S \rangle \} \\ \delta(q_*, x, A) &= \{ \langle q_*, B_1 \dots B_n \rangle \mid A \rightarrow x B_1 \dots B_n \text{ is een regel} \} \quad \text{per variabele } A \in V \text{ en letter } x \in \Sigma\end{aligned}$$

Neem nu een willekeurig woord $w \in \Sigma^*$. We willen aantonen dat $S \xRightarrow{*} w$ met een left-most derivation precies als je de automaat door w op te eten van q_0 naar q_* kan met op het einde een lege stack.¹

- Bewijs dat als $q_*, w, S \vdash^* q_*, \varepsilon, u$ voor $u \in (\Sigma \cup V)^*$ dan $S \xRightarrow{*} w u$. Sterker nog: de afleiding van $w u$ uit S is een *left-most derivation*. Hint: werk met inductie naar het aantal stappen dat de automaat neemt.
- Bewijs dat voor elke $u \in (\Sigma \cup V)^*$ geldt dat als er een *left-most derivation* is van $w u$ uit S , dan geldt $q_*, w, S \vdash^* q_*, \varepsilon, u$. Hint: werk met inductie naar het aantal stappen in de afleiding van $S \xRightarrow{*} w u$.

Beide bewijzen kunnen redelijk kort. Hier geldt: een uitwerking van langer dan één pagina (voor beide bewijzen) wordt *niet* geaccepteerd! Als bonus kun je het bovenstaande samenrapen om het volgende lemma te bewijzen. Hint: kijk naar de nette lange PDA’s als omschreven in de aantekening “context-vrije grammatica’s”, oftewel, PDA’s die alleen accepteren met lege stack.

Lemma 1. *Er is een PDA die de grammatica G herkent.*

¹Zie pagina 108 van het boek voor wat uitleg over left-most derivations. Informeel is het een afleiding waarin je steeds de meest linker variabele herschrijft.