

Samenvatting Automaten en Formele Talen

Lucas Riedstra

31 mei 2018

Opmerking. De nieuwe stof begint op bladzijde 10.

Inhoudsopgave

1	Strings en verzamelingen	2
2	Eindige Automaten	3
3	Pattern matching en Reguliere Expressies	6
4	Homomorfismes	7
5	Beperkingen van eindige automaten	7
6	Automaten versimpelen	8
7	Myhill-Nerode Relaties	8
8	Contextvrije talen	10
9	Pushdown Automaten	12
10	Turingmachines	14

1 Strings en verzamelingen

Definitie 1.1. • Een *alfabet* is een eindige verzameling. We noteren een alfabet vaak met Σ . De elementen uit Σ noemen we *letters* of *symbolen*.

- Een *string* over Σ is een eindige rij elementen uit Σ .
- De *lengte* van een string x is het aantal letters in x .
- De unieke string van lengte 0 heet de *lege string*, genoteerd met ϵ .
- Als a een letter is definiëren we $a^0 = \epsilon$ en $a^n = a^{n-1}a$.
- De verzameling van alle strings over een alfabet Σ noteren we met Σ^* .

Definitie 1.2. • Als x een string is definiëren we $x^0 = \epsilon$ en $x^n = x^{n-1}x$.

- Als $a \in \Sigma$ en $x \in \Sigma^*$, schrijven we $\#a(x)$ voor de hoeveelheid a 's in x .
- Een string x is een *substring* van een string y als y geschreven kan worden als $v x w$, voor $v, w \in \Sigma^*$.
- Een string x is een *prefix* van een string y als y geschreven kan worden als $x w$ voor $w \in \Sigma^*$.
- Een string x is een *suffix* van een string y als y geschreven kan worden als $v x$ voor $v \in \Sigma^*$.

Definitie 1.3. We specificeren een aantal operaties op verzamelingen:

- De *vereniging* van twee verzamelingen: $A \cup B = \{x \mid x \in A \vee x \in B\}$;
- De *doorsnede* van twee verzamelingen: $A \cap B = \{x \mid x \in A \wedge x \in B\}$;
- Het *complement* van een verzameling: $A^c = \{x \in \Sigma^* \mid x \notin A\}$;
- De *concatenatie* van twee verzamelingen: $AB = \{xy \mid x \in A \wedge y \in B\}$;
- De *machten* van een verzameling A : $A^0 = \{\epsilon\}$ en $A^n = AA^{n-1}$.
- De *Kleene-ster* van een verzameling: $A^* = \bigcup_{n \geq 0} A^n$.
- De vereniging van alle natuurlijke machten van A : $A^+ = AA^*$.

Stelling 1.4. *Enkele nuttige eigenschappen van operaties op verzamelingen:*

- *Vereniging, doorsnede, en concatenatie zijn associatief;*
- *Vereniging en doorsnede zijn commutatief;*
- *Vereniging en doorsnede zijn distributief over elkaar;*
- *Concatenatie is distributief over vereniging;*
- $A \cup B = (A^c \cap B^c)^c$ en $A \cap B = (A^c \cup B^c)^c$;
- $A^* A^* = A^{**} = A^*$, $\emptyset^* = \{\epsilon\}$, en $A^* = \{\epsilon\} \cup A^+$.

2 Eindige Automaten

Definitie 2.1. Een *deterministische eindige automaat* (DFA) is een vijf-tupel

$$M = (Q, \Sigma, \delta, s, F).$$

De elementen betekenen het volgende:

- Q is een eindige verzameling, waarvan de elementen *staten* heten;
- Σ is een eindige verzameling, het *input-alfabet*;
- $\delta: Q \times \Sigma \rightarrow Q$ is de *transitiefunctie*;
- $s \in Q$ is de *startstaat*;
- F is een deelverzameling van Q , waarvan de elementen *eindstaten* of *accepterende staten* heten.

Definitie 2.2. Zij $M = (Q, \Sigma, \delta, s, F)$ een DFA. Dan definiëren we inductief de functie $\hat{\delta}: Q \times \Sigma^* \rightarrow Q$ door

$$\begin{aligned}\hat{\delta}(q, \epsilon) &= q, \\ \hat{\delta}(q, xa) &= \delta(\hat{\delta}(q, x), a).\end{aligned}$$

We zeggen dat een string x *geaccepteerd* wordt door M als $\hat{\delta}(s, x) \in F$, en dat x *verworpen* wordt door M als $\hat{\delta}(s, x) \notin F$.

De *taal* geaccepteerd door M is de verzameling van alle strings die M accepteert, en wordt genoteerd met $L(M)$:

$$L(M) = \{x \in \Sigma^* \mid \hat{\delta}(s, x) \in F\}.$$

Een verzameling $A \subseteq \Sigma^*$ heet *regulier* als er een automaat M bestaat zodat $A = L(M)$.

Definitie 2.3. Zij $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ en $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ DFA's. We definiëren het *product* van M_1 en M_2 als de automaat $M_3 = (Q_3, \Sigma, \delta_3, s_3, F_3)$ met de eigenschap dat

$$\begin{aligned}Q_3 &= Q_1 \times Q_2 \\ F_3 &= F_1 \times F_2 \\ s_3 &= (s_1, s_2),\end{aligned}$$

en met $\delta_3: Q_3 \times \Sigma \rightarrow Q_3$ de functie gedefinieerd door

$$\delta_3((p, q), a) = (\delta_1(p, a), \delta_2(q, a)).$$

Stelling 2.4. Zij A, B *regulier* met $A = L(M_1)$ en $B = L(M_2)$. Dan is $A \cap B$ ook *regulier*, en geldt dat $A \cap B$ *geaccepteerd* wordt door het *product* van M_1 en M_2 .

Stelling 2.5. *Zij A regulier met $A = L(M_1)$. Dan is A^c ook regulier, en geldt dat A^c geaccepteerd wordt door M_2 , waarbij M_2 gelijk is aan M_1 behalve dat de accepterende staten van M_2 het complement van die van M_1 zijn.*

Stelling 2.6. *Zij A, B regulier. Dan is $A \cup B$ ook regulier, en kan een bijbehorende DFA geconstrueerd worden met de regel $A \cup B = (A^c \cap B^c)^c$.*

Definitie 2.7. Een niet-deterministische eindige automaat (NFA) is een vijftupel

$$N = (Q, \Sigma, \Delta, S, F).$$

De elementen betekenen hetzelfde als bij een DFA op twee verschillen na:

- $\Delta: Q \times \Sigma \rightarrow 2^Q$ is een transitiefunctie op de machtsverzameling van Q ;
- S is een verzameling startstaten.

Definitie 2.8. Zij $N = (Q, \Sigma, \Delta, S, F)$ een NFA. Dan definiëren we de functie $\hat{\Delta}: 2^Q \times \Sigma^* \rightarrow 2^Q$ door

$$\begin{aligned}\hat{\Delta}(A, \epsilon) &= A \\ \hat{\Delta}(A, xa) &= \bigcup_{q \in \hat{\Delta}(A, x)} \Delta(q, a).\end{aligned}$$

We zeggen dat N een string $x \in \Sigma^*$ *accepteert* als $\hat{\Delta}(S, x) \cap F \neq \emptyset$, en net zoals bij DFA's definiëren we $L(N)$ als de verzameling strings die N accepteert.

Stelling 2.9. *Zij $N = (Q_N, \Sigma, \Delta_N, S_N, F_N)$ een NFA, en zij $M = (Q_M, \Sigma, \delta, s, F)$ een DFA zodanig dat*

$$\begin{aligned}Q_M &= 2^{Q_N} \\ \delta_M(A, a) &= \hat{\Delta}_N(A, a) \\ s_M &= S_N \\ F_M &= \{A \subseteq Q_N \mid A \cap F_N \neq \emptyset\}.\end{aligned}$$

Dan geldt $L(M) = L(N)$. Oftewel: voor elke NFA bestaat er DFA die dezelfde taal accepteert.

Definitie 2.10. Een NFA met ϵ -transities is een zes-tupel $N = (Q, \Sigma, \epsilon, \Delta, S, F)$ zodanig dat

- $\epsilon \notin \Sigma$,
- $N_\epsilon = (Q, \Sigma \cup \{\epsilon\}, \Delta, S, F)$ is een NFA over het alfabet $\Sigma \cup \{\epsilon\}$.

We definiëren de ϵ -afsluiting van een verzameling $A \subseteq Q$ dan als

$$C_\epsilon(A) = \bigcup \{\hat{\Delta}(A, x) \mid x \in \{\epsilon\}^*\},$$

en we definiëren $\hat{\Delta}: 2^Q \times \Sigma^* \rightarrow 2^Q$ inductief door

$$\begin{aligned}\hat{\Delta}(A, \epsilon) &= C_\epsilon(A) \\ \hat{\Delta}(A, xa) &= \bigcup \{C_\epsilon(\Delta(q, a)) \mid q \in \hat{\Delta}(A, x)\}.\end{aligned}$$

Stelling 2.11. Zij $N_1 = (Q, \Sigma, \epsilon, \Delta_1, S, F_1)$ een NFA_ϵ , en zij $N_2 = (Q, \Sigma, \Delta_2, S, F_2)$ een NFA zodanig dat

$$\begin{aligned}\Delta_2(q, a) &= \hat{\Delta}_1(\{q\}, a) \\ F_2 &= \{q \mid C_\epsilon(\{q\}) \cap F_1 \neq \emptyset\}.\end{aligned}$$

Dan geldt $L(N_2) = L(N_1)$. Oftewel: elke taal die geaccepteerd wordt door een NFA_ϵ is regulier.

Stelling 2.12. Reguliere verzamelingen zijn effectief gesloten onder vereniging, concatenatie, en de Kleene-ster.

Bewijs. Vereniging: schrijf de twee bijbehorende automaten naast elkaar op, dit geeft een NFA .

Concatenatie van A en B : maak alle eindstaten in de automaat bij A niet meer accepterend en plaats ϵ -transities naar de startstaten van de automaat bij B .

Kleene-ster: Voeg een nieuwe staat toe en maak daarvan de enige startstaat en eindstaat. Voeg ϵ -transities toe van de nieuwe startstaat naar alle oude startstaten, en voeg ϵ -transities toe van alle oude eindstaten naar de nieuwe startstaat. $\square$

3 Pattern matching en Reguliere Expressies

Definitie 3.1. Zij Σ een eindig alfabet. Een *pattern* is een string van symbolen van een bepaalde vorm die een deelverzameling van Σ^* representeren. Als α een patroon is, definiëren we $L(\alpha)$ als de strings die α *matchen*.

We definiëren patterns inductief, beginnend met de atomaire patterns:

- a voor elke $a \in \Sigma$, met $L(a) = \{a\}$;
- ϵ (de lege string), met $L(\epsilon) = \{\epsilon\}$;
- $\emptyset$, met $L(\emptyset) = \emptyset$;
- $\#$, met $L(\#) = \Sigma$;
- $@$, met $L(@) = \Sigma^*$.

Dan stellen we dat als α en β patterns zijn, dan ook $\alpha + \beta, \alpha \cap \beta, \alpha^*, \alpha^+, \alpha^{\mathbb{G}}$, en $\alpha\beta$, met

$$\begin{aligned} L(\alpha + \beta) &= L(\alpha) \cup L(\beta); \\ L(\alpha \cap \beta) &= L(\alpha) \cap L(\beta); \\ L(\alpha^*) &= (L(\alpha))^*; \\ L(\alpha^+) &= (L(\alpha))^+; \\ L(\alpha^{\mathbb{G}}) &= (L(\alpha))^{\mathbb{G}}; \\ L(\alpha\beta) &= L(\alpha)L(\beta). \end{aligned}$$

Definitie 3.2. Een pattern dat alleen de atomaire patterns $a \in \Sigma, \epsilon, \emptyset$ en de operatoren $+, *, \cap$, en concatenatie gebruikt, heet een *reguliere expressie*.

Stelling 3.3. De volgende drie statements zijn equivalent:

1. A is regulier;
2. $A = L(\alpha)$ voor een pattern α ;
3. $A = L(\alpha)$ voor een reguliere expressie α .

4 Homomorfismes

Definitie 4.1. Een *homomorfisme* is een afbeelding $h: \Sigma^* \rightarrow \Gamma^*$ zodat voor elke $x, y \in \Sigma^*$ geldt dat

$$h(xy) = h(x)h(y).$$

Stelling 4.2. Zij $h: \Sigma^* \rightarrow \Gamma^*$ een homomorfisme.

- Als $B \subseteq \Gamma^*$ regulier is, dan is $h^{-1}(B) \subseteq \Sigma^*$ dat ook;
- Als $A \subseteq \Sigma^*$ regulier is, dan is $h(A) \subseteq \Gamma^*$ dat ook.

In andere woorden: reguliere talen zijn gesloten onder (inverse) beelden van homomorfismes.

5 Beperkingen van eindige automaten

Stelling 5.1 (Pumping lemma). Zij A een reguliere verzameling. Dan geldt:

Er bestaat een $k \geq 0$ zodat voor alle strings x, y, z met $xyz \in A$ en $|y| \geq k$ er strings u, v, w bestaan met $y = uvw$ en $v \neq \epsilon$, zodat voor elke $i \geq 0$ geldt dat $xuv^i w \in A$.

Stelling 5.2 (Pumping lemma, contrapositief). Zij A een verzameling waarvoor het volgende geldt:

Voor elke $k \geq 0$ bestaan er strings x, y, z met $xyz \in A$ en $|y| \geq k$ zodat voor alle strings u, v, w met $y = uvw$ en $v \neq \epsilon$ geldt dat er een $i \geq 0$ bestaat zodat $xuv^i w \notin A$.

Dan is A niet regulier.

Definitie 5.3. Zij U een deelverzameling van $\mathbb{N}_0$. Dan noemen we U *uiteindelijk periodiek* als er getallen $n \geq 0$ en $p > 0$ bestaan zodat voor alle $m \geq n$ geldt dat

$$m \in U \iff m + p \in U.$$

Stelling 5.4. Zij $A \subseteq \{a\}^*$. Dan is A regulier als en slechts als de verzameling $\{m \mid a^m \in A\}$ uiteindelijk periodiek is.

Stelling 5.5. Zij A een reguliere verzameling over een eindig alfabet Σ . Dan is de verzameling van lengtes van strings in A ,

$$\{|x| \mid x \in A\},$$

uiteindelijk periodiek.

6 Automaten versimpelen

Definitie 6.1. Zij $M = (Q, \Sigma, \delta, s, F)$ een eindige automaat. Dan definiëren we een equivalentierelatie op Q als volgt:

$$p \approx q \iff \forall x \in \Sigma^* : (\hat{\delta}(p, x) \in F \iff \hat{\delta}(q, x) \in F).$$

De equivalentieklasse van een staat $p \in Q$ geven we aan met $[p]$. Met behulp van deze equivalentierelatie definiëren we de *quotiëntautomaat* $M/\approx = (Q', \Sigma, \delta', s', F')$ als volgt:

$$\begin{aligned} Q' &= \{[p] \mid p \in Q\} \\ \delta'([p], a) &= [\delta(p, a)] \\ s' &= [s] \\ F' &= \{[p] \mid p \in F\} \end{aligned}$$

Stelling 6.2. Zij M en $M/\approx$ automaten zoals in de vorige definitie. Dan geldt $L(M) = L(M/\approx)$.

Stelling 6.3 (Minimalisatiealgoritme). *Het volgende algoritme geeft aan welke staten in een DFA $M = (Q, \Sigma, \delta, s, F)$ equivalent zijn, als M geen onbereikbare staten heeft:*

1. Maak een tabel van alle paren van staten $\{p, q\}$.
2. Markeer $\{p, q\}$ als $p \in F$ en $q \notin F$ of vice versa.
3. Als er een ongemarkeerd paar $\{p, q\}$ bestaat zodat $\{\delta(p, a), \delta(q, a)\}$ gemarkeerd is voor een $a \in \Sigma$, markeer $\{p, q\}$ dan.
4. Herhaal de vorige stap totdat het niet meer kan.
5. Voor elk ongemarkeerd paar $\{p, q\}$ geldt nu $p \approx q$.

7 Myhill-Nerode Relaties

Definitie 7.1. Twee DFA's $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$ en $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$ worden *isomorf* genoemd als er een bijectie $f: Q_M \rightarrow Q_N$ bestaat zodanig dat

$$\begin{aligned} f(s_M) &= s_N; \\ f(\delta_M(p, a)) &= \delta_N(f(p), a) \text{ voor alle } p \in Q_M \text{ en } a \in \Sigma; \\ p \in F_M &\iff f(p) \in F_N. \end{aligned}$$

In woorden: M en N zijn isomorf als ze gelijk zijn, op naamgeving van de staten na.

Definitie 7.2. Zij $M = (Q, \Sigma, \delta, s, F)$ een DFA, met $L(M) = R$, waarbij M geen onbereikbare staten heeft. Zij $\equiv_M$ nu een equivalentierelatie op Σ^* met de volgende eigenschappen:

1. *Rechtscongruentie*: Voor alle $x, y \in \Sigma^*$ en $a \in \Sigma$ geldt

$$x \equiv_M y \iff xa \equiv_M ya;$$

2. $\equiv_M$ *verfijnt* R : Voor alle $x, y \in \Sigma^*$ geldt

$$x \equiv_M y \implies (x \in R \iff y \in R);$$

3. $\equiv_M$ heeft *eindige index*: Er zijn een eindig aantal equivalentieklassen.

Dan noemen we $\equiv_M$ een *Myhill-Nerode relatie* voor R .

Stelling 7.3. Zij Σ een alfabet, $R \subseteq \Sigma^*$, en zij $\equiv$ een Myhill-Nerode relatie voor R . Definieer de automaat $M_{\equiv} = (Q, \Sigma, \delta, s, F)$ als volgt:

$$\begin{aligned} Q &= \{[x] \mid x \in \Sigma^*\}, \\ s &= [\epsilon], \\ F &= \{[x] \mid x \in R\}, \\ \delta([x], a) &= [xa]. \end{aligned}$$

Dan geldt $L(M_{\equiv}) = R$. Oftewel: een taal is regulier als er een Myhill-Nerode relatie voor deze taal bestaat.

Definitie 7.4. Een relatie $\equiv_1$ *verfijnt* een relatie $\equiv_2$ als

$$x \equiv_1 y \implies x \equiv_2 y.$$

In dit geval noemen we $\equiv_1$ *fijner* dan $\equiv_2$ en $\equiv_2$ *groffer* dan $\equiv_1$.

Stelling 7.5. Zij $R \subseteq \Sigma^*$. Dan definiëren we de equivalentierelatie $\equiv_R$ op Σ^* als volgt:

$$x \equiv_R y \iff \forall z \in \Sigma^* : (xz \in R \iff yz \in R).$$

De relatie $\equiv_R$ is rechtscongruent, verfijnt R , en is de grofste equivalentierelatie op Σ^* met die eigenschappen.

Stelling 7.6 (Stelling van Myhill-Nerode). Zij $R \subseteq \Sigma^*$. Dan zijn de volgende drie uitspraken equivalent:

1. R is regulier;
2. Er bestaat een Myhill-Nerode-relatie voor R ;
3. De relatie $\equiv_R$ heeft eindige index.

8 Contextvrije talen

Definitie 8.1. Een *contextvrije grammatica* (CFG) is een vier-tupel

$$G = (N, \Sigma, P, S),$$

waarbij geldt dat

- N een eindige verzameling *niet-terminalen* is;
- Σ een eindige verzameling *terminalen* is (disjunct van N);
- P een verzameling *producties* is, een deelverzameling van $N \times (N \cup \Sigma)^*$;
- $S \in N$ het startsymbool is.

Voor niet-terminalen gebruiken we hoofdletters, voor terminalen kleine letters, en voor strings in $(N \cup \Sigma)^*$ griekse kleine letters.

Producties schrijven we niet als (A, α) maar als $A \rightarrow \alpha$.

We voegen producties van dezelfde niet-terminaal ook samen: Als $A \rightarrow \alpha$, $A \rightarrow \beta$, en $A \rightarrow \gamma$ producties zijn schrijven we ook $A \rightarrow \alpha \mid \beta \mid \gamma$.

Als we strings $\alpha, \beta \in (N \cup \Sigma)^*$ hebben zodat we β kunnen verkrijgen uit α door één productie toe te passen op één niet-terminaal in α , schrijven we $\alpha \xrightarrow[G]{1} \beta$.

Inductief definiëren we

- $\alpha \xrightarrow[G]{0} \alpha$ voor elke α ;
- $\alpha \xrightarrow[G]{n+1} \beta$ als er een γ bestaat zodat $\alpha \xrightarrow[G]{n} \gamma$ en $\gamma \xrightarrow[G]{1} \beta$;
- $\alpha \xrightarrow[G]{*} \beta$ als er een $n \in \mathbb{N}_0$ bestaat zodat $\alpha \xrightarrow[G]{n} \beta$.

Als er een $n \in \mathbb{N}_0$ bestaat zodat $\alpha \xrightarrow[G]{n} \beta$ zeggen we dat β *afleidbaar* is uit α in n stappen.

Een string die afleidbaar is uit S heet een *zinvorm*. Een zinvorm die enkel terminalen bevat heet een *zin*. De *taal gegenereerd door G* , genoteerd met $L(G)$, is de verzameling van alle zinnen

$$L(G) = \{x \in \Sigma^* \mid S \xrightarrow[G]{*} x\}.$$

Een verzameling $B \subseteq \Sigma^*$ heet een *contextvrije taal* (CFL) als er een CFG G bestaat zodat $B = L(G)$.

Definitie 8.2. Zij $G = (N, \Sigma, P, S)$ een CFG. We zeggen dat G in de *Chomsky-normaalvorm* staat (CNF) als alle producties van G van de vorm $A \rightarrow BC$ of $A \rightarrow a$ zijn, waarbij $A, B, C \in N$ en $a \in \Sigma$.

We zeggen dat G in de *Greibach-normaalvorm* (GNF) staat als alle producties van G van de vorm $A \rightarrow aB_1B_2 \cdots B_k$ voor een $k \geq 0$ zijn, waarbij $A, B_i \in N$ en $a \in \Sigma$.

Definitie 8.3. Een productie $A \rightarrow \epsilon$ heet een ϵ -productie, een productie $A \rightarrow B$ heet een *eenheidsproductie*.

Stelling 8.4. Voor elke CFG $G = (N, \Sigma, P, S)$ is er een CFG G' zonder ϵ - of eenheidsproducties zodat $L(G') = L(G) \setminus \{\epsilon\}$.

Bewijs. We passen twee regels toe:

- Als $A \rightarrow \alpha B \beta$ en $B \rightarrow \epsilon$ producties zijn, voegen we de productie $A \rightarrow \alpha \beta$ toe;
- Als $A \rightarrow B$ en $B \rightarrow \gamma$ producties zijn, voegen we de productie $A \rightarrow \gamma$ toe.

Als we hierna alle ϵ - en eenheidsproducties weghalen zijn we klaar. $\square$

Stelling 8.5. Voor elke CFG G is er een CFG G' zodat G' in de Chomsky-normaalvorm staat, en

$$L(G') = L(G) \setminus \{\epsilon\}.$$

Bewijs. Haal eerst alle ϵ - en eenheidsproducties weg. Haal dan alle producties $A \rightarrow a$ weg voor elke $A \in N$ en $a \in \Sigma$.

Voeg voor elke terminaal a de niet-terminaal A_a en de productie $A_a \rightarrow a$ toe. Vervang nu bij elke productie (behalve bij de zojuist gedefinieerde) elk voorkomen van een terminaal a door zijn nieuwe bijbehorende niet-terminaal A_a .

Vervolgens gaan we inductief verder: Elke productie van de vorm $A \rightarrow B_1 B_2 \cdots B_k$ (met $k > 2$) vervangen we door $A \rightarrow B_1 C$ en $C \rightarrow B_2 \cdots B_k$. Dit herhalen we totdat we niet meer verder kunnen, en dan zijn we klaar. $\square$

Stelling 8.6. Voor elke CFG G is er een CFG G' zodat G' in de Greibach-normaalvorm staat, en

$$L(G') = L(G) \setminus \{\epsilon\}.$$

Stelling 8.7. Als A een reguliere taal is, is A een contextvrije taal.

Bewijs. Neem een DFA voor A . Maak van de staten niet-terminalen, en maak van elke transitie $(q_1, a) \rightarrow q_2$ een productie $Q_1 \rightarrow a Q_2$. Voeg bij de eindstaat F de productie $F \rightarrow \epsilon$ toe. Deze contextvrije grammatica heet *sterk rechtstreeks*: elke productie is van de vorm $X \rightarrow a Y$. $\square$

Stelling 8.8 (Pumping Lemma). Zij A een taal waarvoor geldt:

Voor alle $k \geq 0$ bestaat er een $z \in A$ met $|z| \geq k$, zodat voor elke splitsing $z = uvwxy$ met $vx \neq \epsilon$ en $|vwx| \leq k$ er een $i \geq 0$ bestaat zodanig dat $uv^iwx^i y \notin A$.

Dan is A geen contextvrije taal.

Stelling 8.9 (Ogden's Lemma). Zij A een taal waarvoor geldt:

Voor alle $k \geq 0$ bestaat er een $z \in A$ met ten minste k gemarkeerde letters, zodat voor elke splitsing $z = uvwxy$ met $vx \neq \epsilon$ waarbij vwx ten hoogste k gemarkeerde letters bevat, er een $i \geq 0$ bestaat zodat $uv^iwx^i y \notin A$.

9 Pushdown Automaten

Definitie 9.1. Een *niet-deterministische pushdown automaat* (NPDA) is een zeven-tupel

$$M = (Q, \Sigma, \Gamma, \delta, s, \perp, F),$$

waarbij

- Q een eindige verzameling staten is;
- Σ een eindig input-alfabet is;
- Γ een eindig *stack-alfabet* is;
- $\delta \subseteq (Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma) \times (Q \times \Gamma^*)$ de *transitielerelatie* is (ook eindig);
- $s \in Q$ de startstaat is;
- $\perp \in \Gamma$ het *start-stacksymbool* is;
- $F \subseteq Q$ de verzameling einstaten is.

Een *configuratie* van M is een element uit $Q \times \Sigma^* \times \Gamma^*$ dat de huidige staat van de automaat beschrijft: de huidige staat, het huidige overgebleven stuk van de input, en de huidige stack. De *startconfiguratie* voor een willekeurige input x is $(s, x, \perp)$.

Als er een transitie $((p, a, A), (q, \gamma)) \in \delta$ bestaat dan beschrijven we zo'n stap van de automaat met de *configuratielerelatie* $\xrightarrow[M]{1}$, dus

$$((p, a, A), (q, \gamma)) \in \delta \implies (p, ay, A\beta) \xrightarrow[M]{1} (q, y, \gamma\beta),$$

en

$$((p, \epsilon, A), (q, \gamma)) \in \delta \implies (p, y, A\beta) \xrightarrow[M]{1} (q, y, \gamma\beta).$$

We definiëren nu analoog aan bij CFG's de stappenrelaties:

- $C \xrightarrow[M]{0} D \iff C = D$;
- $C \xrightarrow[M]{n+1} D \iff \exists E : C \xrightarrow[M]{n} E \wedge E \xrightarrow[M]{1} D$;
- $C \xrightarrow[M]{*} D \iff \exists n \geq 0 : C \xrightarrow[M]{n} D$.

We zeggen dat een woord *geaccepteerd* wordt *door eindstaat* als $(s, x, \perp) \xrightarrow[M]{*} (q, \epsilon, \gamma)$ voor een $q \in F, \gamma \in \Gamma^*$. De verzameling door eindstaat geaccepteerde woorden noemen we $L_f(G)$. We zeggen dat een woord geaccepteerd wordt *door lege stack* als $(s, x, \perp) \xrightarrow[M]{*} (q, \epsilon, \epsilon)$ voor een $q \in Q$. De verzameling door lege stack geaccepteerde woorden noemen we $L_e(G)$.

Stelling 9.2. *Zij A een verzameling. De volgende drie statements zijn equivalent:*

- $A = L(G)$ voor een CFG G ;
- $A = L_f(M_1)$ voor een NPDA M_1 ;
- $A = L_e(M_2)$ voor een NPDA M_2 .
- $A = L_e(M_3) = L_f(M_3)$ voor een NPDA M_3 .

Definitie 9.3. Een *deterministische pushdown automaat* (DPDA) is een acht-tupel $M = (Q, \Sigma, \Gamma, \delta, \perp, \neg, s, F)$, waarbij

- $Q, \Sigma, \Gamma, \perp, s, F$ allemaal hetzelfde betekenen als bij NPDA's;
- δ een *deterministische* transitiefunctie is: bij elke staat is elke transitie voor elk mogelijk stacksymbool op één manier gedefinieerd, of een staat heeft alleen een ϵ -transitie;
- δ nooit $\perp$ van de stack af mag halen.

Acceptatie bij DPDA's gebeurt alleen op eindstaat, en een taal die geaccepteerd wordt door een DPDA noemen we een *deterministische contextvrije taal* (DCFL).

Stelling 9.4. *Elke CFL is een DCFL, maar niet elke DCFL is een CFL.*

Stelling 9.5. *Om te bepalen of een woord wel of niet geaccepteerd wordt kan het Cocke-Kasami-Younger-algoritme gebruikt worden. Dit staat beschreven in college 8 en op bladzijde 194-196 van het boek.*

Stelling 9.6. *CFL's zijn gesloten onder vereniging, concatenatie, Kleene-ster, doorsnede met reguliere verzamelingen, en homomorfe (inverse) beelden.*

CFL's zijn niet gesloten onder doorsnede en complement.

Bewijs. Voor alle afsluitingseigenschappen gaan we er vanuit dat de verzameling nietterminalen van twee CFL's disjunct zijn.

Vereniging: Voeg de producties samen en voeg een nieuwe startproductie $S \rightarrow S_1 \mid S_2$ toe.

Concatenatie: Voeg de producties samen en voeg een nieuwe startproductie $S \rightarrow S_1 S_2$ toe.

Kleene-ster: Voeg een nieuwe startproductie $S \rightarrow S_1 S \mid \epsilon$ toe. □

Stelling 9.7. *DCFL's zijn gesloten onder complement en homomorfe inverse beelden, maar niet onder vereniging, doorsnede, concatenatie, Kleene-ster, en homomorfe beelden.*

10 Turingmachines

Definitie 10.1. Een *turingmachine* (TM) is een negen-tupel $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$, met

- Q een verzameling staten, Σ het eindige inputalfabet en Γ het eindige tape-alfabet (met $\Sigma \subseteq \Gamma$);
- $\sqcup$ het blank-symbool en $\vdash$ de linker-eindmarker (beide niet in Σ);
- δ een transitiefunctie van $Q \times \Sigma$ naar $Q \times \Sigma \times \{L, R\}$ (links/rechts);
- s de startstaat, t de accepterende staat, en r de verwerpende staat.

Verder leggen we nog de beperking op dat de linker-eindmarker nooit overschreven mag worden, en dat zodra de machine in een eind- of verwerpende staat komt, deze nooit verlaten wordt.

Een *configuratie* van een TM is een element uit $Q \times \Gamma^* \times \mathbb{N}_0$ (huidige staat, tape, en plek op tape). We definiëren een configuratierelatie $\xrightarrow[M]{1}$ en de relaties $\xrightarrow[M]{0}$, $\xrightarrow[M]{n}$ en $\xrightarrow[M]{*}$ net zoals bij PDA's, alleen houden we in de transitierelatie rekening met links (L) of rechts (R).

Als een TM een string x accepteert of verwerpt *stopt* deze op x , anders *loopt* de machine op x . Een TM die nooit loopt op input heet een *totale turingmachine*.

Definitie 10.2. Zij A een verzameling strings. Als geldt dat $A = L(M_1)$ voor een turingmachine M_1 noemen we A *recursief opsombaar* (r.e.). Als het complement van A recursief opsombaar is noemen we A *co-r.e.*, en als $A = L(M_2)$ voor een totale turingmachine M_2 noemen we A *recursief*.

Een eigenschap P op strings wordt *beslisbaar* genoemd als de verzameling $\{x \mid P(x)\}$ recursief is, en wordt *semi-beslisbaar* genoemd als de verzameling $\{x \mid P(x)\}$ r.e. is. Een niet-beslisbare taaal heet *onbeslisbaar*.

Stelling 10.3. *Elke recursieve taal en elke contextvrije taal is r.e., maar niet elke recursief opsombare taal is contextvrij of recursief.*

Stelling 10.4. *Een taal A is recursief als en slechts als deze zowel r.e. als co-r.e. is.*

Stelling 10.5. *Turingmachines kunnen op een aantal manieren uitgebreid worden, die geen grotere klasse opleveren, namelijk: Nietdeterminisme, een tape die zowel naar links als naar rechts oneindig is, meerdere tapes, en een multidimensionale tape.*

Definitie 10.6. We representeren een Turingmachine $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$ met bits als volgt:

We nummeren de staten $0, 1, \dots, n-1$, en zo ook het input-alfabet en tape-alfabet. Vervolgens geven we die weer met een aantal nullen gescheiden door

enen. Dit geeft als beginstring $0^n 10^m 10^k 10^s 10^t 10^r 10^u 10^v 1$, waarbij n het aantal staten, m het aantal letters uit Γ en k het aantal letters uit Σ representeren, waarbij s, t , en r de nummers van die staten weergeven, en u en v de eind-marker en blank-symbool. Daarachteraan plakken we alle transities uit δ . Als bijvoorbeeld $((p, a), (q, b, L))$ een transitie is uit δ , dan plakken we de string $0^p 10^a 10^q 10^b 10$ achter onze huidige string (bij R plakken we een 1 achteraan ipv 0).

Definitie 10.7. Gegeven verzamelingen $A \subseteq \Sigma^*$ en $B \subseteq \Delta^*$ definiëren we een *reductie* van A naar B als een berekenbare functie $\sigma: \Sigma^* \rightarrow \Delta^*$ met als eigenschap dat voor elke $x \in \Sigma^*$ geldt dat $x \in A \iff \sigma(x) \in B$. Sigma moet hierbij totaal berekenbaar zijn.

Als zo een σ bestaat, schrijven we $A \leq_m B$. v

Stelling 10.8. • Als $A \leq_m B$ en B is r.e., dan A ook. Als A niet r.e. is, dan B ook niet.

• Als $A \leq_m B$ en B is recursief, dan A ook. Als A niet recursief is, dan B ook niet.

Definitie 10.9. We noemen een eigenschap P op de recursief opsombare talen *monotoon* als geldt dat

$$A \subseteq B \implies P(A) \leq P(B),$$

met de regel dat $\perp < \top$.

Stelling 10.10. • Elke niettriviale eigenschap van recursief opsombare talen is onbeslisbaar.

• Geen enkele niet-monotone eigenschap van recursief opsombare talen is semi-beslisbaar.

Opmerking. Buiten reductie hebben we ook diagonalisatie gezien om onbeslisbaarheid te bepalen. Zie hiervoor de slides van college 10.