Wykłady z Języków Formalnych i Teorii Automatów

Antoni Kościelski

1 Wykład 1

1.1 Podstawowe pojęcia

Alfabetem nazywamy dowolny zbiór skończony. Elementy alfabetu nazywamy **znakami** lub **literami**.

Niech A będzie ustalonym alfabetem. Niżej podajemy aksjomatyczną definicję pojęcia słowa nad alfabetem A. Pojęcie to nie zostanie więc w pełni zdefiniowane. Zostaną tylko przytoczone jego pewne własności. Przyjmiemy, że słowa to coś, co ma przytoczone własności.

Zakładamy, że A jest podzbiorem pomocniczego zbioru P, w którym jest określone pewne działanie. Jeżeli $x,\ y \in P$, to wynik tego działania zastosowanego do x i y oznaczamy symbolem xy. Działanie to będziemy nazywać **konkatenacją**. Zakładamy, że konkatenacja jest działaniem łącznym, a więc

$$(xy)z \quad = \quad x(yz)$$

dla dowolnych $x,\ y,\ z \in P$, oraz że zachodzi następujące prawo skracania

$$xa = yb \Longrightarrow x = y \wedge a = b$$

dla dowolnych $x,\ y \in P$ oraz $a,\ b \in A$. Przyjmujemy także, że w zbiorze P jest element ε spełniający

$$x\varepsilon \ = \ \varepsilon x \ = \ x$$

dla wszystkich $x \in P$. Nietrudno zauważyć, że jest najwyżej jeden taki element. Będziemy nazywać go **słowem pustym**.

Interesują nas zbiory $X \subseteq P$ o następujących własnościach

- $A \subseteq X$ oraz $\varepsilon \in X$,
- jeżeli $x \in X$ i $a \in A$, to $xa \in X$,

a więc zbiory zawierające alfabet A i słowo puste ε, zamknięte ze względu na konkatenację z pojedyńczymi literami alfabetu A. Takie zbiory istnieją, jednym

z nich jest zbiór P. Dowodzi się, że jest wśród nich zbio najmniejszy, zawarty we wszystkich innych takich zbiorach. Ten zbiór będziemy oznaczać symbolem A^* i nazywać zbiorem wszystkich słów nad albetem A. Natomiast elementy zbioru A^* nazywamy **słowami** nad alfabetem A. Z przytoczonej definicji, w oczywisty sposób wynika następujący schemat indukcji.

Twierdzenie 1.1 (o dowodzeniu przez indukcję) *Jeżeli zbiór X ma wyżej podane własności, to każde słowo nad alfabetem A należy do X.*

2 Wykład 2

2.1 Formalna definicja automatu skończonego

Automatem skończonym M nazywamy piątkę

$$\langle Q, \Sigma, \delta, q_0, F \rangle,$$

gdzie

- Q jest skończonym zbiorem, nazywamy go **zbiorem stanów**,
- Σ jest alfabetem, nazywamy go **alfabetem wejściowym**,
- $\delta : Q \times \Sigma \to Q$ jest funkcją i nazywamy ją **funkcją przejścia**,
- $q_0 \in Q$ jest jednym ze stanów, nazywamy go **stanem początkowym**,
- $F \subseteq Q$ jest pewnym zbiorem stanów, nazywamy go zbiorem **stanów końcowych lub akceptujących**.

Dla dowolnego automatu skończonego M, funkcję przejścia δ tego automatu rozszerzamy do funkcji $\widehat{\delta} : Q \times \Sigma^* \to Q$ tak, aby

$$\begin{aligned} \widehat{\delta}(q, \varepsilon) &= q, \\ \widehat{\delta}(q, sa) &= \delta(\widehat{\delta}(q, s), a) \end{aligned}$$

dla dowolnych $q \in Q$, $s \in \Sigma^*$ i $a \in \Sigma$. Nietrudno zauważyć, że funkcja $\widehat{\delta}$ może zostać zinterpretowana jako funkcja, która stanowi q automatu skończonego i słowu s przyporządkowuje stan, w którym znajdzie się automat po przeczytaniu słowa s, jeżeli zostanie uruchomiony w stanie q. Często obie funkcje δ i $\widehat{\delta}$ będziemy oznaczać tym samym symbolem δ.

Słowo $s \in \Sigma^*$ jest **akceptowane** przez automat skończony M, jeżeli $\widehat{\delta}(q_0, s) \in F$. **Językiem akceptowanym** przez automat M nazywamy

$$L(M) = \{s \in \Sigma^* : \widehat{\delta}(q_0, s) \in F\}.$$

3 Wykład 3

3.1 Automaty niedeterministyczne są równoważne deterministycznym

Twierdzenie 3.1 *Języki akceptowane przez niedeterministyczne automaty skończone są regularne.*

Dowód. Przypuśćmy, że $L = L(M)$ jest językiem akceptowanym przez niedeterministyczny automat skończony

$$M \quad = \quad \langle Q, \Sigma, \delta, q_0, F \rangle.$$

W szczególności δ jest funkcją przyporządkowującą stanowi i literze pewien zbiór stanów. Rozważmy automat

$$M_d \quad = \quad \langle 2^Q, \Sigma, \Delta, \{q_0\}, \{X \in 2^Q : \ X \cap F \neq \emptyset\} \rangle,$$

gdzie $\Delta : 2^Q \times \Sigma \to 2^Q$ jest funkcją zdefiniowaną wzorem

$$\Delta(X, a) \quad = \quad \bigcup_{q \in X} \delta(q, a).$$

Przypuśćmy, że automat M po przeczytaniu pewnego słowa może się znaleźć w stanach ze zbioru X, a po przeczytaniu tego słowa z dopisaną literą a - w stanach ze zbioru Y. Zauważmy, że automat M_d znajdujący się w stanie X i obserwujący literę a po wykonaniu jednego ruchu znajdzie się w stanie Y.

Funkcję Δ jak zwykle rozszerzamy do funkcji $\widehat{\Delta} : 2^Q \times \Sigma^* \to 2^Q$. Funkcja ma następującą własność

$$\widehat{\Delta}(X, s) \quad = \quad \bigcup_{q \in X} \widehat{\delta}(q, s) \tag{1}$$

dla $s \in \Sigma^*$. Wzór ten dowodzimy przez indukcję ze względu na postać słowa s. Druga część dowodu indukcyjnego wynika z następujących równości.

$$\begin{aligned}
\widehat{\Delta}(X, sa) &= \Delta(\widehat{\Delta}(X, s), a) \\
&= \bigcup_{q' \in \widehat{\Delta}(X,s)} \delta(q', a) \\
&= \bigcup_{q' \in \bigcup_{q \in X} \widehat{\delta}(q,s)} \delta(q', a) \\
&= \bigcup_{q \in X} \bigcup_{q' \in \widehat{\delta}(q,s)} \delta(q', a) \\
&= \bigcup_{q \in X} \widehat{\delta}(q, sa).
\end{aligned}$$

Pierwsza z tych równości wynika z definicji $\widehat{\Delta}$. Druga – z definicji Δ. Trzecia – z założenia indukcyjnego dla słowa s. Piąta – z definicji $\widehat{\delta}$. Natomiast czwarta równość jest konsekwencją łączości i przemienności sumy mnogościowej. Z tych praw wynika w szczególności, że $A \cup B \cup C \cup D = (A \cup B) \cup (D \cup C)$. Czwarta równość jest prawdziwa z tych samych powodów, co ta ostatnia.

Wzór 1 w przypadku $X = \{q_0\}$ przyjmuje postać

$$\widehat{\Delta}(\{q_0\}, s) \quad = \quad \widehat{\delta}(q_0, s) \tag{2}$$

(suma z wzoru 1 jest jednoskładnikowa i jest równa temu składnikowi). Z wzoru 2 już łatwo wyprowadzić, że $L(M) = L(M_d)$. □

Dowód powyższego twierdzenia podaje sposób przekształcania automatu niedeterministycznego w deterministyczny, akceptujący ten sam język.

3.2 Dodatkowe wnioski z dowodu

Funkcja Δ jest funkcją przejścia automatu deterministycznego. Z zadania 3 z listy 1 wynika, że rozszerzenie tej funkcji na zbiór $2^Q \times \Sigma^*$, czyli funkcja $\widehat{\Delta}$ spełnia dla wszystkich $X \in 2^Q$ i $s_1, s_2 \in \Sigma^*$

$$\widehat{\Delta}(X, s_1 s_2) \quad = \quad \widehat{\Delta}(\widehat{\Delta}(X, s_1), s_2).$$

Stąd, stosując wzór 1 dla $X = \{q\}$ otrzymujemy

$$\widehat{\delta}(q, s_1 s_2) \quad = \quad \widehat{\Delta}(\widehat{\delta}(q, s_1), s_2).$$

Jeżeli jeszcze raz zastosujemy wzór 1, to otrzymamy

$$\widehat{\delta}(q, s_1 s_2) \quad = \quad \bigcup_{q' \in \widehat{\delta}(q, s_1)} \widehat{\delta}(q', s_2). \tag{3}$$

Wzór 3 podaje zależność między zbiorem stanów, w jakich może się znaleźć automat niedeterministyczny po przeczytaniu konkatenacji $s_1 s_2$ i zbiorami stanów, w których może się znaleźć automat po przeczytaniu słow s_1 i s_2. W szczególności wynikają z niego dwie własności:

- jeżeli $\widehat{\delta}(q, s_1) = \emptyset$, to $\widehat{\delta}(q, s_1 s_2) = \emptyset$,
- jeżeli $\widehat{\delta}(q, s_1) = \{q'\}$, to $\widehat{\delta}(q, s_1 s_2) = \widehat{\delta}(q', s_2)$

dla wszystkich $q, q' \in Q$ i $s_1, s_2 \in \Sigma^*$.

3.3 Diagramy automatów niedeterministycznych

Automaty niedeterministyczne, podobnie jak deterministyczne, możemy przedstawiać w postaci diagramów. W tym przypadku, rysując diagram dla automatu z funkcją przejścia δ, z wierzchołka odpowiadającego stanowi q_1 prowadzimy krawędź do wierzchołka odpowiadającego stanowi q_2 i etykietujemy ją literą a wtedy i tylko wtedy, gdy $q_2 \in \delta(q_1, a)$.

Wykład 4.

4.1 Zamkniętość ze względu na działania mnogościowe

Twierdzenie 4.1 *Klasa języków regularnych jest zamknięta ze względu na dopełnienie.*

Dowód. Jeżeli $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ jest deterministycznym automatem skończonym akceptującym język L, to automat $M^c = \langle Q, \Sigma, \delta, q_0, Q \setminus F\rangle$ akceptuje $\Sigma^* \setminus L$. □

Lemat 4.2 *Następujące warunki są równoważne:*

1. *język L jest regularny,*
2. *język $L \cup \{\varepsilon\}$ jest regularny,*
3. *język $L \setminus \{\varepsilon\}$ jest regularny*

Dowód. Wystarczy dowieść, że jeżeli język L jest regularny, to języki wymienione w treści lematu też są regularne. Przypuśćmy, że deterministyczny automat skończony $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ akceptuje język L. Rozważmy dwa automaty:

$$\begin{aligned} M_1 &= \langle Q \cup \{q_0'\}, \Sigma, \delta', q_0', F \cup \{q_0'\}\rangle \\ M_2 &= \langle Q \cup \{q_0'\}, \Sigma, \delta', q_0', F\rangle \end{aligned}$$

gdzie q_0' jest stanem nie należącym do Q, a funkcja przejścia δ' jest zdefiniowana wzorem

$$\delta'(q,a) = \begin{cases} \delta(q,a) & \text{jeżeli } q \in Q \\ \delta(q_0,a) & \text{jeżeli } q = q_0', \end{cases}$$

dla $a \in \Sigma$. Nietrudno zauważyć, że dla niepustych słów s rozszerzenie tej funkcji spełnia

$$\begin{aligned} \widehat{\delta}'(q_0', s) &\subseteq Q \\ \widehat{\delta}'(q_0', s) &= \widehat{\delta}(q_0, s) \end{aligned}$$

Stąd możemy wywnioskować, że $L(M_1) = L \cup \{\varepsilon\}$ oraz $L(M_2) = L \setminus \{\varepsilon\}$. □

Twierdzenie 4.3 *Suma mnogościowa dwóch języków regularnych jest językiem regularnym.*

Dowód. Dowód zostanie podany przy dodatkowym założeniu, że oba języki są akceptowane przez automaty z tym samym alfabetem wejściowym. Przypuśćmy, że dane są dwa niedeterministyczne automaty

$$\begin{aligned} M_1 &= \langle Q_1, \Sigma, \delta_1, q_{0,1}, F_1\rangle \\ M_2 &= \langle Q_2, \Sigma, \delta_2, q_{0,2}, F_2\rangle \end{aligned}$$

o rozłącznych zbiorach stanów. Niech

$$M \quad = \quad \langle Q_1 \cup Q_2 \cup \{q_0\}, \Sigma, \delta, q_0, F_1 \cup F_2 \rangle,$$

gdzie $q_0 \notin Q_1 \cup Q_2$, funkcja przejścia δ jest zdefiniowana wzorem

$$\delta(q,a) \quad = \quad \begin{cases} \delta_1(q,a) & \text{jeżeli } q \in Q_1 \\ \delta_2(q,a) & \text{jeżeli } q \in Q_2 \\ \delta_1(q_{0,1},a) \cup \delta_2(q_{0,2},a) & \text{jeżeli } q = q_0, \end{cases}$$

Zauważmy, że rozszerzenie funkcji δ spełnia dla niepustych słów s

$$\begin{aligned} \widehat{\delta}(q_0,s) &\subseteq Q_1 \cup Q_2 \\ \widehat{\delta}(q_0,s) &= \delta_1(q_{0,1},s) \cup \delta_2(q_{0,2},s). \end{aligned}$$

Stąd, dla niepustych $s \in \Sigma^*$ otrzymujemy, że

$$s \in L(M) \iff s \in L(M_1) \vee s \in L(M_2).$$

Teza twierdzenia wynika z Lematu 4.2. □

4.2 Konkatenacja języków

Jeżeli L_1 i L_2 są językami, to język

$$L_1 L_2 \quad = \quad \{xy : \ x \in L_1 \wedge y \in L_2\}$$

nazywamy konkatenacją języków L_1 i L_2.

Twierdzenie 4.4 *Konkatenacja języków regularnych jest językiem regularnym.*

Dowód. Jak poprzednio załóżmy, że dane są dwa niedeterministyczne automaty

$$\begin{aligned} M_1 &= \langle Q_1, \Sigma, \delta_1, q_{0,1}, F_1 \rangle \\ M_2 &= \langle Q_2, \Sigma, \delta_2, q_{0,2}, F_2 \rangle \end{aligned}$$

o rozłącznych zbiorach stanów. Niech

$$M \quad = \quad \langle Q_1 \cup Q_2, \Sigma, \delta, q_{0,1}, F_2 \rangle,$$

gdzie funkcja przejścia δ jest zdefiniowana wzorem

$$\delta(q,a) \quad = \quad \begin{cases} \delta_1(q,a) \cup \delta_2(q_{0,2},a) & \text{jeżeli } q \in F_1, \\ \delta_1(q,a) & \text{jeżeli } q \in Q_1 \setminus F_1 \\ \delta_2(q,a) & \text{jeżeli } q \in Q_2 \end{cases}$$

Automaty M i M_1 pracują tak samo pod warunkiem, że automat M nie korzysta ze stanów z Q_2. Fakt ten można wyrażić wzorem

$$\widehat{\delta}(q_{0,1}, s) \cap Q_1 \quad = \quad \widehat{\delta}_1(q_{0,1}, s) \tag{4}$$

prawdziwym dla wszystkich słów $s \in \Sigma^*$. Dowodzimy go przez indukcję.

$$\begin{aligned}
\widehat{\delta}(q_{0,1}, sa) \cap Q_1 &= \left(\bigcup_{q \in \widehat{\delta}(q_{0,1}, s)} \delta(q, a) \right) \cap Q_1 \\
&= \left(\bigcup_{q \in \widehat{\delta}(q_{0,1}, s) \cap Q_1} \delta(q, a) \cup \bigcup_{q \in \widehat{\delta}(q_{0,1}, s) \setminus Q_1} \delta(q, a) \right) \cap Q_1 \\
&= \bigcup_{q \in \widehat{\delta}_1(q_{0,1}, s)} (\delta(q, a) \cap Q_1) \cup \bigcup_{q \in \widehat{\delta}(q_{0,1}, s) \setminus Q_1} (\delta_2(q, a) \cap Q_1) \\
&= \bigcup_{q \in \widehat{\delta}_1(q_{0,1}, s)} \delta_1(q, a) \\
&= \widehat{\delta}_1(q_{0,1}, s).
\end{aligned}$$

Ponadto, dla wszystkich $s \in \Sigma^*$

$$\widehat{\delta}(q_{0,1}, s) \cap Q_2 \quad = \bigcup_{\substack{\exists x \; xy = s \wedge |x| < |s| \\ \wedge \widehat{\delta}(q_{0,1}, x) \cap F_1 \neq \emptyset}} \widehat{\delta}_2(q_{0,2}, y). \tag{5}$$

Suma po prawej stronie wzoru przebiega po wszystkich słowach y spełniających warunki podane pod znakiem sumy. Jest zbiorem pustym, jeżeli nie ma słów spełniających podane warunki. Wzór ten mówi, że automat M kończy pracę w stanie należącym do Q_2 wtedy i tylko wtedy, gdy przed zakończeniem czytania danego słowa znajdzie się w stanie z F_1 (po raz ostatni) i dalej pracuje dokładnie, jak automat M_2. Dowód tego jest – jak zwykle – indukcyjny.

$$\widehat{\delta}(q_{0,1}, sa) \cap Q_2 \quad = \quad \left(\bigcup_{q \in \widehat{\delta}(q_{0,1}, s)} \delta(q, a) \right) \cap Q_2$$

$$= \left(\bigcup_{q \in \widehat{\delta}(q_{0,1}, s) \cap F_1} \delta(q, a) \cup \bigcup_{q \in \widehat{\delta}(q_{0,1}, s) \cap (Q_1 \setminus F_1)} \delta(q, a) \right.$$

$$
\begin{aligned}
&\qquad\qquad \cup \bigcup_{q\in\widehat{\delta}(q_{0,1},s)\cap Q_2} \delta(q,a)\Bigg) \cap Q_2 \\
&= \bigcup_{q\in\widehat{\delta}(q_{0,1},s)\cap F_1} ((\delta_1(q,a)\cup\delta_2(q_{0,2},a))\cap Q_2) \\
&\qquad\qquad \cup \bigcup_{q\in\widehat{\delta}(q_{0,1},s)\cap(Q_1\setminus F_1)} (\delta_1(q,a)\cap Q_2) \\
&\qquad\qquad\qquad \cup \bigcup_{q\in\widehat{\delta}(q_{0,1},s)\cap Q_2} (\delta(q,a)\cap Q_2) \\
&= \bigcup_{q\in\widehat{\delta}(q_{0,1},s)\cap F_1} \delta_2(q_{0,2},a) \cup \bigcup_{q\in\widehat{\delta}(q_{0,1},s)\cap Q_2} \delta_2(q,a) \\
&= \bigcup_{q\in\widehat{\delta}(q_{0,1},s)\cap F_1} \delta_2(q_{0,2},a) \cup \bigcup_{\substack{\exists x\ xy=s\wedge|x|<|s| \\ \wedge\widehat{\delta}(q_{0,1},x)\cap F_1\neq\emptyset}} \bigcup_{q\in\widehat{\delta}_2(q_{0,2},y)} \delta_2(q,a) \\
&= \bigcup_{q\in\widehat{\delta}(q_{0,1},s)\cap F_1} \delta_2(q_{0,2},a) \cup \bigcup_{\substack{\exists x\ xy=s\wedge|x|<|s| \\ \wedge\widehat{\delta}(q_{0,1},x)\cap F_1\neq\emptyset}} \widehat{\delta}_2(q_{0,2},ya) \\
&= \bigcup_{\substack{\exists x\ xy=sa\wedge|x|<|sa| \\ \wedge\widehat{\delta}(q_{0,1},x)\cap F_1\neq\emptyset}} \widehat{\delta}_2(q_{0,2},y).
\end{aligned}
$$

Teraz zauważmy, że słowa z języka $L_1(L_2\setminus\{\varepsilon\})$ są akceptowane przez automat M. Jeżeli $x\in L_1$ i $y\in L_2\setminus\{\varepsilon\}$, to istnieje stan $q\in\widehat{\delta}_1(q_{0,1},x)\cap F_1$ i – na mocy wzoru 5 – mamy

$$\widehat{\delta}(q_{0,1},xy) \quad\supseteq\quad \widehat{\delta}_2(q_{0,2},y).$$

W drugim z tych zbiorów jest stan ze zbioru F_2, a więc stan akceptujący automatu M. Stan ten należy również do pierwszego zbioru, a więc automat M akceptuje xy.

Jeżeli słowo s jest akceptowane przez automat M, to w zbiorze $\widehat{\delta}(q_{0,1},s)$ jest stan akceptujący $q\in F_2\subseteq Q_2$. Zgodnie ze wzorem 5, $q\in\widehat{\delta}_2(q_{0,2},y)$ dla pewnego $y\neq\varepsilon$ takiego, że $s=xy$ i dla słowa x zachodzi $\widehat{\delta}(q_{0,1},x)\cap F_1\neq\emptyset$. Jeżeli $q\in\widehat{\delta}_2(q_{0,2},y)$, to oczywiście $y\in L(M_2)=L_2$. Jeżeli $\widehat{\delta}(q_{0,1},x)\cap F_1\neq\emptyset$, to pewien stan akceptujący $q'\in F_1$ automatu M_1 należy do $\widehat{\delta}(q_{0,1},x)$. Na mocy wzoru 4 stan $q'\in\widehat{\delta}_1(q_{0,1},x)$. Oznacza to, że słowo x jest akceptowane przez automat M_1. Wobec tego $s=xy\in L_1(L_2\setminus\{\varepsilon\})$ i ostatecznie $L(M)=L_1(L_2\setminus\{\varepsilon\})$.

Ostatecznie teza wynika z Twierdzenia 4.3. Język L_1L_2 jest równy bowiem $L_1(L_2\setminus\{\varepsilon\})$ (jeżeli $\varepsilon\notin L_2$) lub $L_1(L_2\setminus\{\varepsilon\})\cup L_1$ (w przeciwnym razie). □

4.3 Definicja domknięcia Kleene'ego

Dla dowolnego języka L definiujemy domknięcie Kleene'ego L^*. Domknięcie Kleene'ego L^* jest najmniejszym spośród zbiorów X spełniających

1. $\varepsilon \in X$,
2. $L \subseteq X$,
3. jeżeli $s \in X$ oraz $x \in L$, to $sx \in X$.

Jeżeli przez L^0 oznaczymy $\{\varepsilon\}$ i przyjmniemy, że $L^{n+1} = L^n L$ dla dowolnego $n \in N$, to domknięcie Kleene'ego możemy zdefiniować wzorem

$$L^* = \bigcup_{n \in N} L^n.$$

Jeszcze inaczej domknięcie Kleene'ego definiujemy jako zbiór słów będących konkatenacjami wyrazów skończonych ciągów słów z języka L (za konkatenację wyrazów ciągu pustego uznajemy słowo puste). Można się przekonać, że trzy wyżej podane definicje rzeczywiście definiują ten sam język.

"Dodatnie" domknięcie Kleene'ego L^+ języka L definiujemy podobnie jako najmniejszy spośród zbiorów X spełniających

1. $L \subseteq X$,
2. jeżeli $s \in X$ oraz $x \in L$, to $sx \in X$

lub wzorem

$$L^+ = \bigcup_{n > 0} L^n.$$

Wobec Lematu 4.2, albo języki L^* i L^+ są jednocześnie regularne, albo jednocześnie nie są regularne.

Zauważmy też, że $L^* L \subseteq L^*$ oraz $L^+ L \subseteq L^+$.

4.4 Regularność domknięcia Kleene'ego

Twierdzenie 4.5 *Dla dowolnego regularnego języka L, języki L^* oraz L^+ są regularne.*

Dowód. Przypuśćmy, że niedeterministyczny automat

$$M = \langle Q, \Sigma, \delta, q_0, F \rangle$$

akceptuje język L. Udowodnimy, że język L^+ jest regularny. Niech

$$M^+ = \langle Q, \Sigma, \delta^+, q_0, F \rangle$$

będzie automatem z funkcją przejścia zdefiniowaną wzorem

$$\delta^+(q,a) \quad = \quad \begin{cases} \delta(q,a) & \text{jeżeli } q \in Q \setminus F \\ \delta(q,a) \cup \delta(q_0,a) & \text{jeżeli } q \in F \end{cases}$$

($a \in \Sigma$). Automat M^+ działa tak, jak automat M, ale dodatkowo, w stanach akceptujących może działać tak, jak automat M znajdujący się w stanie początkowym. Automat M^+ może więc pracować albo dokładnie jak M (nie korzystając z dodatkowych możliwości), albo dokładnie jak automat M, ale po skorzystaniu po raz ostatni z dodatkowej możliwości przejścia automatu M^+. Można to opisać precyzyjniej wzorem

$$\widehat{\delta}^+(q_0,s) \quad = \quad \widehat{\delta}(q_0,s) \cup \bigcup_{\substack{\exists x\ xy=s \wedge |x|<|s| \\ \wedge \widehat{\delta}^+(q_0,x) \cap F \neq \emptyset}} \widehat{\delta}(q_0,y). \tag{6}$$

Przyjmijmy, że

$$\Delta_s(a) = \bigcup_{q \in \widehat{\delta}^+(q_0,s) \cap F} \delta(q_0,a) = \begin{cases} \delta(q_0,a) & \text{jeżeli } \widehat{\delta}^+(q_0,s) \cap F \neq \emptyset \\ \emptyset & \text{w przeciwnym razie} \end{cases} .$$

Drugi krok dowodu indukcyjnego tego wzoru prowadzimy w następujący sposób:

$$\begin{aligned}
\widehat{\delta}^+(q_0,sa) &= \bigcup_{q \in \widehat{\delta}^+(q_0,s)} \delta^+(q,a) \\
&= \bigcup_{q \in \widehat{\delta}^+(q_0,s) \cap F} (\delta(q,a) \cup \delta(q_0,a)) \cup \bigcup_{q \in \widehat{\delta}^+(q_0,s) \setminus F} \delta(q,a) \\
&= \bigcup_{q \in \widehat{\delta}^+(q_0,s) \cap F} \delta(q_0,a) \cup \bigcup_{q \in \widehat{\delta}^+(q_0,s)} \delta(q,a) \\
&= \Delta_s(a) \cup \bigcup_{q \in \widehat{\delta}^+(q_0,s)} \delta(q,a) \\
&= \Delta_s(a) \cup \bigcup_{q \in \widehat{\delta}(q_0,s)} \delta(q,a) \cup \bigcup_{\substack{\exists x\ xy=s \wedge |x|<|s| \\ \wedge \widehat{\delta}^+(q_0,x) \cap F \neq \emptyset}} \bigcup_{q \in \widehat{\delta}(q_0,y)} \delta(q,a) \\
&= \Delta_s(a) \cup \widehat{\delta}(q_0,sa) \cup \bigcup_{\substack{\exists x\ xy=s \wedge |x|<|s| \\ \wedge \widehat{\delta}^+(q_0,x) \cap F \neq \emptyset}} \widehat{\delta}(q_0,ya)
\end{aligned}$$

$$
\begin{aligned}
&= \widehat{\delta}(q_0, sa) \cup \left(\Delta_s(a) \cup \bigcup_{\substack{\exists x\ xy=s \wedge |x|<|s| \\ \wedge \widehat{\delta^+}(q_0,x) \cap F \neq \emptyset}} \widehat{\delta}(q_0, ya) \right) \\
&= \widehat{\delta}(q_0, sa) \cup \bigcup_{\substack{\exists x\ xy=sa \wedge |x|<|sa| \\ \wedge \widehat{\delta^+}(q_0,x) \cap F \neq \emptyset}} \widehat{\delta}(q_0, ya).
\end{aligned}
$$

Nietrudno zauważyć, że $L^+ \subseteq L(M^+)$. Z wzoru 6 otrzymujemy, że

$$\widehat{\delta^+}(q_0, s) \supseteq \widehat{\delta}(q_0, s)$$

oraz

$$\widehat{\delta^+}(q_0, s) \supseteq \widehat{\delta}(q_0, y)$$

dla dowolnego $y \neq \varepsilon$ takiego, że $s = xy$ i $\widehat{\delta^+}(q_0, x) \cap F \neq \emptyset$. Stąd łatwo wywnioskować, że $L(M^+)$ zawiera L oraz konkatenacja słowa x akceptowanego przez automat M^+ i słowa $x \in L$ jest akceptowana przez automat M^+. Tak więc $L^+ \subseteq L(M^+)$.

Przeciwne zawieranie dowodzimy przez indukcję ze względu na długość słowa. Przypuśćmy, że słowo s jest akceptowane przez automat M^+ oraz, że słowa krótsze od s i akceptowane przez M^+ należą do L^+. W zbiorze $\widehat{\delta^+}(q_0, s)$ jest stan akceptujący, należący do F. Zgodnie ze wzorem 6 są możliwe dwa przypadki. Albo stan ten należy do $\widehat{\delta}(q_0, s)$ i w konsekwencji $s \in L$, albo należy on do $\widehat{\delta}(q_0, y)$ dla pewnych x i y takich, że $s = xy$, $|x| < |y|$ i $\widehat{\delta^+}(q_0, x) \cap F \neq \emptyset$. Takie słowa spełniają $x \in L^+$ (na mocy założenia indukcyjnego) i $y \in L$. Ostatecznie $s = xy \in L^+L \subseteq L^+$. Oznacza to, że $L(M^+) \subseteq L^+$. □

Wykład 5.

5.1 Charakteryzacja języków regularnych

Twierdzenie 5.1 *Klasa języków regularnych nad alfabetem Σ jest najmniejszą klasą języków spośród klas $\mathcal{X}$ spełniających*

1. *$\emptyset \in \mathcal{X}$ oraz $\{\varepsilon\} \in \mathcal{X}$,*
2. *jeżeli $a \in \Sigma$, to $\{a\} \in \mathcal{X}$,*
3. *jeżeli L_1, $L_2 \in \mathcal{X}$, to $L_1 \cup L_2 \in \mathcal{X}$ oraz $L_1L_2 \in \mathcal{X}$,*
4. *jeżeli $L \in \mathcal{X}$, to $L^* \in \mathcal{X}$.*

Dowód. Niech $\mathcal{R}$ oznacza klasę języków regularnych nad alfabetem Σ, a $\mathcal{X}_0$ – najmniejszą klasę języków mającą własności wymienione w tezie. Jest oczywiste, że języki wymienione w tezie Twierdzenia 5.1 są regularne. Wiemy, że klasa $\mathcal{R}$ jest zamknięta ze względu na sumę mnogościową, konkatenację i domknięcie Kleene'ego. Tak więc $\mathcal{R}$ jest jedną z klas o wymienionych własnościach i, w konsekwencji, $\mathcal{X}_0 \subseteq \mathcal{R}$.

Przypuśćmy więc, że $L \subseteq \Sigma^*$ jest językiem akceptowanym przez deterministyczny automat

$$M \;=\; \langle Q, \Sigma, \delta, q_0, F\rangle.$$

Pokażemy, że $L \in \mathcal{X}_0$. Przypuśćmy, że stany automatu M zostały ponumerowane liczbami mniejszymi od n, a więc $q = \{q_0, q_1, \ldots, q_{n-1}\}$. Niech

$$\begin{aligned} L^m_{k,l} \;=\; \{s \in \Sigma^* :\; & \widehat{\delta}(q_k, s) = q_l \wedge \forall i < n \forall s_1, s_2 \in \Sigma^* \setminus \{\varepsilon\} \\ & (s_1 s_2 = s \wedge \widehat{\delta}(q_k, s_1) = q_i \Longrightarrow i < m)\} \end{aligned}$$

dla dowolnych k, $l < n$ i $m \leq n$. Mówiąc inaczej, język $L^m_{k,l}$ składa się ze słów, których czytanie powoduje przejście automatu M od stanu q_k do q_l w taki sposób, że z wyjątkiem najwyżej pierwszego i ostatniego momentu, automat stale znajduje się w stanach o numerach mniejszych od m.

Prawdziwe są następujące wzory:

$$\begin{aligned} L^0_{k,l} \;&=\; \begin{cases} \{a \in \Sigma :\; \delta(q_k, a) = q_l\} & \text{jeżeli } q_k \neq q_l, \\ \{a \in \Sigma :\; \delta(q_k, a) = q_l\} \cup \{\varepsilon\} & \text{jeżeli } q_k = q_l, \end{cases} \\ L^{m+1}_{k,l} \;&=\; L^m_{k,l} \cup L^m_{k,m} L^{m+1}_{m,m} L^m_{m,l}, \\ L^{m+1}_{m,m} \;&=\; (L^m_{m,m})^*. \end{aligned}$$

Aby dowieść te wzory wystarczy przeanalizowazować ciągi stanów, w których znajduje się automat M podczas czytania danego słowa. Wzory te pozwalają bez trudu dowieść przez indukcję ze względu na m, że

$$\forall m \leq n \forall k, l < n \; L^m_{k,l} \in \mathcal{X}_0.$$

W szczególności, zachodzi

$$\forall k, l < n \; L^n_{k,l} \in \mathcal{X}_0.$$

Nietrudno zauważyć, że

$$L \;=\; L(M) \;=\; L^n_{0,i_1} \cup L^n_{0,i_2} \cup \ldots \cup L^n_{0,i_t},$$

gdzie i_1, i_2, …, i_t są wszystkimi stanami akceptującymi automatu M (w przypadku $F = \emptyset$, suma z powyższego wzoru jest zbiorem pustym). Stąd oczywiście wynika, że $L \in \mathcal{X}_0$. □

5.2 Wyrażenia regularne

Niech

$$\Sigma_r \quad = \quad \{\emptyset, \varepsilon, +, ^*, (,)\} \cup \Sigma$$

(przez pewien czas symbol $\emptyset$ i pozostałe uważamy za znaki pewnego alfabetu i nie precyzujemy, co te symbole oznaczają). Wyrażeniem regularnym nazywamy słowo nad alfabetem Σ_r należące do zbioru wyrażeń regularnych. Z kolei zbiór wyrażeń regularnych definiujemy jako najmniejszy spośród zbiorów X spełniających

1. $\emptyset \in X$, $\varepsilon \in X$ oraz $\Sigma \subseteq X$,
2. jeżeli r_1, $r_2 \in X$, to $(r_1 + r_2) \in X$ oraz $r_1 r_2 \in X$,
3. jeżeli $r \in X$, to $(r)^* \in X$.

Na przykład wyrażeniem regularnym jest słowo $((0 + 1)(01)^*)^*0$ (w przypadku $\Sigma = \{0, 1\}$). Wyrażenia regularne są właściwie wzorami definiującymi pewien język. Postać tych wzorów została ukształtowana przed wieloma laty i odbiega trochę od naszych przyzwyczajeń. Podane wyrażenie regularne to wzór oznaczający język $((\{0\} \cup \{1\})(\{0\}\{1\})^*)^*\{0\}$, czyli konkatenację domknięcia Kleene'ego języka $(\{0\} \cup \{1\})(\{0\}\{1\})^*$ oraz języka $\{0\}$. Aby to sformalizować, definiujemy pojęcie języka reprezentowanego przez wyrażenie regulrane r. Indukcyjna definicja języka reprezentowanego przez wyrażenie ma postać

$$L(r) \quad = \quad \begin{cases} \emptyset & \text{jeżeli } r = \emptyset, \\ \{\varepsilon\} & \text{jeżeli } r = \varepsilon, \\ \{a\} & \text{jeżeli } r = a \in \Sigma, \\ L(r_1) \cup L(r_2) & \text{jeżeli } r = (r_1 + r_2), \\ L(r_1)L(r_2) & \text{jeżeli } r = r_1 r_2, \\ (L(r_1))^* & \text{jeżeli } r = (r_1)^*. \end{cases}$$

Jeżeli r jest wyrażeniem regularnym, to mówimy, że słowo s ma postać r wtedy i tylko wtedy, gdy $s \in L(r)$.

Korzystając z Twierdzenia 5.1 można dowieść

Twierdzenie 5.2 *Język $L \subseteq \Sigma^*$ jest regularny wtedy i tylko wtedy, gdy L jest reprezentowany przez pewne wyrażenie regularne $r \in \Sigma_r$.*□

5.3 Lemat o nadymaniu

Twierdzenie 5.3 (Lemat o nadymaniu) *Dla dowolnego języka regularnego L istnieje liczba naturalna n taka, że dowolne słowo $s \in L$ o długości przynajmniej n można podzielić na trzy części u, v i w (a więc $s = uvw$) tak, aby $|uv| \leq n$, $|v| > 0$ oraz, aby każde słowo postaci $uv^i w$ ($i \in N$) należało do L.*

Przed dowodem lematu o nadymaniu pokażemy, jak można z niego wyprowadzić, że język

$$L \quad = \quad \{0^k 1^k \in \{0,1\}^* : \ n \in N\}$$

nie jest regularny.

Przypuśćmy, że język L jest regularny. Korzystając z Twierdzenia 5.3 bierzemy liczba naturalna n o własnościach podanych w tezie. Tak więc każde słowo z języka L, o długości równej przynajmniej n, a więc także słowo $s = 0^n 1^n$, można odpowiednio podzielić na trzy części tak, aby

$$\begin{aligned} 0^n 1^n &= uvw, \\ |uv| &\leq n, \\ |v| &> 0 \end{aligned}$$

oraz, aby – między innymi – słowo uv^2w należało do L. Nietrudno zauważyć, że $v = 0^m$ dla pewnej liczby $m > 0$. Tak więc $uv^2w = 0^{n+m}1^n \in L$. Ponieważ $m > 0$, jest to sprzeczne z definicją L. Uzyskana sprzeczność świadczy o tym, że język L nie jest regularny.

Wykład 6.

6.1 Dowód Lematu 5.3 o nadymaniu

Dowód. Przypuśćmy, że L jest językiem regularnym. Weźmy deterministyczny automat skończony

$$M \quad = \quad \langle Q, \Sigma, \delta, q_0, F \rangle$$

akceptujący język L. Liczbę n, która ma spełniać tezę Twierdzenia 5.3, definiujemy jako liczbę stanów automatu M.

Aby przekonać się liczba n ma żadane własności weźmy słowo $x \in L$ o długości m, $m \geq n$. Przypuśćmy, że

$$x \quad = \quad a_1 a_2 \ldots a_m$$

gdzie a_i są literami z alfabetu Σ. Rozważmy ciąg stanów, w których znajdzie się automat M podczas czytania słowa x, a więc

$$q_0, \ q_1 = \widehat{\delta}(q_0, a_1), \ q_2 = \widehat{\delta}(q_0, a_1 a_2), \ \ldots, \ q_n = \widehat{\delta}(q_0, a_1 a_2 a_3 \ldots a_n).$$

Ciąg ten ma $n+1$ elementów. Pewien stan występuje więc w nim dwukrotnie. Niech $0 \leq k < l \leq n$ oraz

$$q_k \ = \ \widehat{\delta}(q_0, a_1 \ldots a_k) \ = \ \widehat{\delta}(q_0, a_1 \ldots a_k a_{k+1} \ldots a_l) \ = \ q_l.$$

Zdefiniujmy $u = a_1 \dots a_k$, $v = a_{k+1} \dots a_l$ i $w = a_{l+1} \dots a_m$. Jest oczywiste, że $|uv| \leq n$, $|v| \geq 1$ oraz $x = uvw$, a także

$$\widehat{\delta}(q_0, u) \quad = \quad \widehat{\delta}(q_0, uv).$$

Aby zakończyć dowód wystarczy pokazać, że każde ze słów $uv^i w$ należy do języka L.

Zauważmy, że dla dowolnej liczby $i \in N$,

$$\widehat{\delta}(q_0, uv^i) \quad = \quad \widehat{\delta}(q_0, u).$$

Jest to oczywiste dla $i = 0$. Dalej dowodzimy to przez indukcję w następujący sposób

$$\begin{aligned}
\widehat{\delta}(q_0, uv^{i+1}) &= \widehat{\delta}(q_0, uv^i v) \\
&= \widehat{\delta}(\widehat{\delta}(q_0, uv^i), v) \\
&= \widehat{\delta}(\widehat{\delta}(q_0, u), v) \\
&= \widehat{\delta}(q_0, uv) \\
&= \widehat{\delta}(q_0, u).
\end{aligned}$$

Wobec powyższego wzoru mamy także

$$\begin{aligned}
\widehat{\delta}(q_0, uv^i w) &= \widehat{\delta}(\widehat{\delta}(q_0, uv_i), w) \\
&= \widehat{\delta}(\widehat{\delta}(q_0, u), w) \\
&= \widehat{\delta}(\widehat{\delta}(q_0, uv), w) \\
&= \widehat{\delta}(q_0, uvw) \\
&= \widehat{\delta}(q_0, x) \in F.
\end{aligned}$$

Tak więc wszystkie słowa $uv^i w$ są akceptowane przez automat M i należą do języka L. □

Wniosek 6.1 *Jeżeli deterministyczny automat M o n stanach akceptuje słowo x o długości przynajmniej równej n, to akceptuje także pewne słowo y takie, że*

$$|x| - n \leq |y| < |x|.$$

Dowód. Jeżeli słowo x podzielimy w sposób spełniający tezę Lematu o nadymaniu na części u, v i w, to słowo y może zostać zdefiniowane jako uw. □

6.2 Wnioski dotyczące problemów decyzyjnych

Lemat 6.2 *Niech M będzie deterministycznym automatem skończonym o n stanach. Język $L(M)$ jest niepusty wtedy i tylko wtedy, gdy automat M akceptuje pewne słowo o długości mniejszej od n. Oznacza to, że istnieje algorytm pozwalający na stwierdzenie, czy dany automat skończony akceptuje przynajmniej jedno słowo.*

Dowód. Oczywiście, jeżeli automat akceptuje pewne słowo, to język $L(M)$ jest niepusty. Odwrotnie, jeżeli język $L(M)$ jest niepusty, to jest najkrótsze słowo akceptowane przez automat M. Na mocy Wniosku 6.1 nie może mieć ono długości większej lub równej n. Ma więc długość mniejszą niż n. □

Lemat 6.3 *Niech M będzie deterministycznym automatem skończonym o n stanach. Język $L(M)$ jest nieskończony wtedy i tylko wtedy, gdy automat M akceptuje pewne słowo o długości większej lub równej n i mniejszej od $2 \cdot n$. Oznacza to, że jest algorytm pozwalający na stwierdzenie, czy dany automat skończony akceptuje nieskończenie wiele słów.*

Dowód. Z Lematu 5.3 o nadymaniu wynika, że jeżeli automat M akceptuje słowo o długości równej przynajmniej liczbie stanów automatu M, to do $L(M)$ należy nieskończenie wiele słów wynienionych w tezie wspomnianego lematu.

Natomiast jeżeli automat M akceptuje nieskończenie wiele słów, to także akceptuje pewne słowo o długości większej lub równej n. Z Wniosku 6.1 wynika, że najkrótsze takie słowo nie może mieć długości większej lub równej $2 \cdot n$. Jego długość jest więc mniejsza niż $2 \cdot n$. □

Lemat 6.4 *Istnieje algorytm, który dla danych automatów skończonych M_1 i M_2 stwierdza, czy te automaty akceptują dokładnie ten sam język.*

Dowód. Przypuśćmy, że automaty M_1 i M_2 mają odpowiednio n_1 i n_2 stanów. Można w efektywny sposób skonstruować automat M o n stanach akceptujący różnicę symetryczną języków $L(M_1)$ i $L(M_2)$. Istotne jest właściwie tylko to, ze potrafimy efektywnie ustalić liczbę stanów tego automatu. Dalej, zgodnie z Lematem 6.2 sprawdzamy, czy pewne słowo o długości mniejszej niż n jest akceptowane przez jeden i nie jest akceptowane przez drugi z automatów M_1 i M_2. □

Wykład 7.

7.1 Automaty skończone a relacje niezmiennicze

Relację $R \subseteq (\Sigma^*)^2$ nazywamy **niezmienniczą**, jeżeli dla dowolnych słów x, y i $z \in \Sigma^*$, warunek xRy implikuje, że $xzRyz$.

Lemat 7.1 *Niech*

$$M \quad = \quad \langle Q, \Sigma, \delta, q_0, F \rangle$$

będzie automatem skończonym (deterministycznym – dla ustalenia uwagi – ale lemat jest także prawdziwy dla automatów niedeterministycznych). Relacja $R_\delta \subseteq (\Sigma^*)^2$ zdefiniowana przez

$$xR_\delta y \iff \widehat{\delta}(q_0, x) = \widehat{\delta}(q_0, y)$$

jest niezmienniczą relacją równoważności o skończonej liczbie klas abstrakcji. Ponadto albo $L(M)$ jest pustym językiem, albo jest sumą mnogościową pewnych klas abstrakcji tej relacji.

Dowód. Jest oczywiste, że R_δ jest relacją równoważności. Aby przekonać się o niezmienniczości tej relacji zauważmy, że jeżeli $\widehat{\delta}(q_0, x) = \widehat{\delta}(q_0, y)$, to także

$$\widehat{\delta}(q_0, xz) = \widehat{\delta}(\widehat{\delta}(q_0, x), z) = \widehat{\delta}(\widehat{\delta}(q_0, y), z) = \widehat{\delta}(q_0, yz).$$

Dla $q \in Q$ zdefiniujmy

$$X_q \quad = \quad \{x \in \Sigma^* : \widehat{\delta}(q_0, x) = q\}.$$

Jest więc oczywiste, że zbiorów postaci X_q jest najwyżej tyle, co stanów automatu M i, wobec tego, jest ich skończenie wiele. Wśród nich są wszystkie klasy abstrakcji relacji R_δ. Aby się o tym przekonać, obliczmy klasę abstrakcji słowa x_0:

$$\{x \in \Sigma^* : xR_\delta x_0\} = \{x \in \Sigma^* : \widehat{\delta}(q_0, x) = \widehat{\delta}(q_0, x_0)\} = X_{\widehat{\delta}(q_0, x_0)}.$$

Łatwo się także przekonać, że jeżeli $x_0 \in X_q$, to $q = \widehat{\delta}(q_0, x_0)$ i w konsekwencji $X_q = X_{\widehat{\delta}(q_0, x_0)}$. Oznacza to, że zbiór X_q jest albo zbiorem pustym, albo jest klasą abstrakcji relacji R_δ. Stąd bez trudu otrzymujemy ostatnią część tezy, gdyż

$$L(M) = \{x \in \Sigma^* : \widehat{\delta}(q_0, x) \in F\} = \bigcup_{q \in F} X_q. \square$$

Lemat 7.2 *Jeżeli $R \subseteq (\Sigma^*)^2$ jest niezmienniczą relacją równoważności o skończonej liczbie klas abstrakcji i L jest sumą pewnej rodziny klas abstrakcji relacji R, to L jest akceptowany przez deterministyczny automat skończony mający tyle stanów, ile relacja R ma klas abstrakcji. Tak więc L jest językiem regularnym.*

Dowód. Niech $\mathcal{K}$ będzie zbiorem klas abstrakcji relacji R i niech F będzie podzbiorem $\mathcal{K}$. Przyjmijmy, że L jest sumą mnogościową klas należących do F,

$$L \quad = \quad \bigcup_{X \in F} X.$$

Pokażemy, że język L jest akceptowany przez automat

$$M \quad = \quad \langle \mathcal{K}, \Sigma, \delta, [\varepsilon], F \rangle,$$

gdzie $[\varepsilon]$ jest klasą abstrakcji relacji R wyznaczoną przez słowo puste (ogólniej, $[x]$ będzie oznaczać klasę abstrakcji relacji R wyznaczoną przez słowo x), a δ jest funkcją zdefiniowaną wzorem

$$\delta([x], a) \quad = \quad [xa]$$

dla dowolnego słowa $x \in \Sigma^*$ i dowolnej litery $a \in \Sigma$. W przypadku takiej definicji musimy sprawdzić, czy jest ona poprawna, a więc, czy $[xa] = [ya]$ wtedy, gdy $[x] = [y]$ (sprawdzamy, że wartość funkcji δ nie zależy od sposobu przedstawienia pierwszego argumentu). Poprawność definicji wynika z niezmienniczości relacji R. Jeżeli $[x] = [y]$, to słowa x i y są równoważne w sensie relacji R. Ponieważ jest to relacja niezmiennicza, także słowa xa i ya są równoważne w sensie relacji R. Wobec tego, klasy abstrakcji relacji R, wyznaczone przez słowa xa i ya są identyczne.

Prosty dowód indukcyjny pozwala na wykazanie, że dla dowolnych słów x i y zachodzi

$$\widehat{\delta}([x], y) \quad = \quad [xy],$$

i w szczególności

$$\widehat{\delta}([\varepsilon], x) \quad = \quad [x].$$

Aby zakończyć dowód zauważmy, że

$$x \in L(M) \Longleftrightarrow \widehat{\delta}([\varepsilon], x) \in F \Longleftrightarrow [x] \in F \Longleftrightarrow x \in L.$$

Wątpliwości może budzić najwyżej ostatnia równoważność. Jeżeli $[x] \in F$, to klasa $[x]$ jest jednym ze składników sumy równej L, jest więc zawarta L. Tak więc wszytkie elementy $[x]$, w tym x, należą do L. I odwrotnie, jeżeli $x \in L$, to $x \in [y] \in F$. W takim przypadku $[x] = [y]$ i w konsekwencji $[x] \in F$. □

Powyższe lematy w nowy sposób pozwalają uzasadnić, że języki akceptowane przez automaty niedeterministyczne są regularne.

7.2 Przykład relacji niezmienniczej

Niech $L \subseteq \Sigma^*$ będzie dowolnym językiem. Zdefiniujmy relację $R_L \subseteq (\Sigma^*)^2$ tak, aby

$$xR_Ly \quad \Longleftrightarrow \quad \forall z \in \Sigma^* \; (xz \in L \Longleftrightarrow yz \in L)$$

dla dowolnych słów x, $y \in L$.

Lemat 7.3 *Dla dowolnego języka L, relacja R_L jest niezmienniczą relacją równoważności i L jest sumą pewnej rodziny klas abstrakcji tej relacji.*

Dowód. Jest to właściwie oczywiste. Zauważmy tylko, że dwa słowa równoważne w sensie relacji R_L albo jednocześnie należą do języka L, albo jednocześnie do niego nie należą. Oznacza to, że klasy abstrakcji relacji R_L są albo zawarte w L, albo rozłącze z L. Tak więc L jest sumą klas abstrakcji relacji R_L wyznaczonych przez słowa należace do L. □

Twierdzenie 7.4 *Jeżeli język L jest akceptowany przez deterministyczny automat skończony o n stanach, to relacja R_L ma najwyżej n klas abstrakcji.*

Dowód. Niech $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ będzie deterministycznym automatem akceptującym język L. Zdefiniujmy funkcję f, która stanom automatu M przyporządkowuję klasy abstrakcji relacji R_L. Funkcja f stanowi q przyporządkowuje klasę abstrakcji wyznaczoną przez słowo puste, jeżeli $\widehat{\delta}(q_0, x) \neq q$ dla dowolnego $x \in \Sigma^*$. W przeciwnym razie, jeżeli $\widehat{\delta}(q_0, x) = q$, to $f(q)$ jest klasą abstrakcji wyznaczoną przez słowo x. Także tym razem musimy sprawdzić poprawność powyższej definicji. Wynika ona stąd, że dowolne słowa spełniające $\widehat{\delta}(q_0, x) = \widehat{\delta}(q_0, y)$ są równoważne w sensie relacji R_L. Jeżeli $\widehat{\delta}(q_0, x) = \widehat{\delta}(q_0, y)$, to również $\widehat{\delta}(q_0, xz) = \widehat{\delta}(q_0, yz)$ i, wobec tego, $\widehat{\delta}(q_0, xz) \in F$ wtedy i tylko wtedy, gdy $\widehat{\delta}(q_0, yz) \in F$.

Ponieważ $f(\widehat{\delta}(q_0, x))$ jest klasą abstrakcji wyznaczoną przez słowo x, funkcja f przyjmuje jako wartości wszystkie możliwe klasy abstrakcji relacji R_L. Aby uzasadnić tezę wystarczy jeszcze zauważyć, że dziedzina dowolnej funkcji typu "na" ma przynajmniej tyle elementów, co zbiór wartości.

Inny dowód Twierdzenia 7.4 można otrzymać z zawierania $R_\delta \subseteq R_L$, które zostało udowodnione podczas sprawdzania poprawności definicji funkcji f. Oznacza ono, że podział zbioru Σ^* na klasy abstracji relacji R_δ jest drobniejszy niż podział na klasy abstrakcji relacji R_L. Podział drobniejszy jest podziałem na więcej części, a podział na klasy abstracji relacji R_δ dzieli Σ^* na skończenie wiele zbiorów. □

Deterministyczny automat skończony akceptujący język L nazywamy **minimalnym**, jeżeli żaden deterministyczny automat o mniejszej liczbie stanów nie akceptuje języka L.

Wniosek 7.5 *Deterministyczny automat skończony akceptujący język L jest minimalny wtedy i tylko wtedy, gdy ma tyle stanów, ile jest klas abstrakcji relacji R_L.*

Dowód. Jest to wniosek z Twierdzenia 7.4 oraz Lematów 7.3 i 7.2. □

7.3 Minimalizacja automatów skończonych

Teraz przedstawimy konstrukcję automatu minimalnego akceptującego ten sam język, co dany automat skończony. Nie wnikając w szczegóły przedstawimy też argumenty świadczące o tym, że w podany sposób rzeczywiście konstruujemy automat minimalny.

Przypuśćmy, że mamy dany deterministyczny automat skończony

$$M \quad = \quad \langle Q, \Sigma, \delta, q_0, F\rangle.$$

Niezbędnymi nazywamy te stany automatu M, które należą do najmniejszego spośród zbiorów $X \subseteq Q$ takich, że

1. $q_0 \in X$,
2. jeżeli $q \in X$ oraz $a \in \Sigma$, to $\delta(q, a) \in X$.

Pozostałe stany automatu M nazywamy **zbędnymi**. Nietrudno zauważyć, że wszystkie stany niezbędne są postaci $\widehat{\delta}(q_0, x)$, dla pewnego $x \in \Sigma^*$, oraz po usunięciu z automatu M stanów zbędnych otrzymujemy automat akceptujący język $L(M)$. Dalej będziemy zakładać, że automat M nie ma stanów zbędnych.

Przyjmijmy, że $L(M) = L$. Zdefiniujmy relację $R \subseteq Q^2$ tak, aby

$$q_1 R q_2 \iff \exists u \in \Sigma^* \exists w \in \Sigma^* \; u R_L w \wedge \widehat{\delta}(q_0, u) = q_1 \wedge \widehat{\delta}(q_0, w) = q_2.$$

Dla automatów bez stanów zbędnych, R jest relacją równoważności, która ma tyle samo klas abstracji, co relacja R_L. Aby się o tym przekonać, wystarczy sprawdzić, że warunek "$f(\widehat{\delta}(q_0, x))$ jest klasą abstrakcji relacji R_L, wyznaczoną przez słowo x" definiuje (poprawnie) pewną różnowartościową funkcję określoną w zbiorze Q i przyjmującą jako wartości wszystkie klasy abstrakcji relacji R_L.

Minimalizacja automatu M wymaga znalezienia relacji R. Zwykle, zamiast relacji R obliczamy najpierw jej dopełnienie korzystając z następującej charakteryzacji: $Q^2 \setminus R$ jest najmniejszym spośród zbiorów $X \subseteq Q^2$ spełniających

1. $F \times Q \setminus F \subseteq X$,
2. jeżeli $(q_1, q_2) \in X$, to $(q_2, q_1) \in X$,
3. jeżeli $q_1 \neq q_2$ i $(\widehat{\delta}(q_1, a), \widehat{\delta}(q_2, a)) \in X$ dla pewnego $a \in \Sigma$, to $(q_1, q_2) \in X$.

Przypuśćmy, że oprócz automatu M mamy "popsuty" automat M. Popsuty automat po przeczytaniu litery zmienia swój stan w sposób, który nie musi być zgodny z funkcją przejścia; czasem popełnia błędy polegające na tym, że zamiast znaleźć się w stanie q_1 (zgodnie z funkcją przejścia), przechodzi do innego stanu q_2. Zakładamy jednak, że błędy automatu nie są całkowicie dowolne. Robiąc błąd przechodzi on do stanu q_2 tylko wtedy, gdy działając poprawnie powinien się znaleźć w stanie q_1 takim, że $q_1 R q_2$.

Taki "popsuty" automat M także akceptuje język $L(M)$. Przypuśćmy, że czytając słowo x popełnia tylko jeden błąd polegający na przejściu do stanu q_2 zamiast q_1. Wobec tego $q_1 R q_2$. Niech u i w będą słowami takimi, że $\widehat{\delta}(q_0, u) = q_1$, $\widehat{\delta}(q_0, w) = q_2$ oraz $u R_L w$. Przyjmijmy, że błąd ten miał miejsce w momencie, gdy zostało przeczytane słowo y i do przeczytania zostało jeszcze słowo z. Tak więc $\widehat{\delta}(q_0, y) = q_1$ i $x = yz$. Poprawnie działający automat M kończy pracę w stanie

$$\widehat{\delta}(q_0, x) = \widehat{\delta}(q_0, yz) = \widehat{\delta}(\widehat{\delta}(q_0, y), z) = \widehat{\delta}(q_1, z) = \widehat{\delta}(\widehat{\delta}(q_0, u), z) = \widehat{\delta}(q_0, uz).$$

Natomiast błędnie działający automat M zakończy czytanie słowa x w stanie

$$\widehat{\delta}(q_2, z) = \widehat{\delta}(\widehat{\delta}(q_0, w), z) = \widehat{\delta}(q_0, wz).$$

Ponieważ uR_Lw, słowa uz i wz albo jednocześnie należą do języka L, albo jednocześnie do niego nie należą. Wobec tego

$$\widehat{\delta}(q_0, uz) \in F \iff \widehat{\delta}(q_0, wz) \in F,$$

a to oznacza, że albo poprawnie i niepoprawnie działające automaty M akceptują słowo x, albo jednocześnie go odrzucają. Korzystając z zasady indukcji, przedstawione rozumowanie bez trudu można uogólnić na przypadek dowolnej liczby błędów.

Przypuśćmy, że automat M ma dwa różne stany q_1 i q_2 takie, że q_1Rq_2. W takiej sytuacji możemy popsuć automat M tak, aby przechodził do stanu q_2 zawsze wtedy, gdy zgodnie z funkcją przejścia powinien znaleźć się w stanie q_1. Automat M i popsuty w taki sposób automat M będą akceptować ten sam język. Co więcej, w tym przypadku popsuty automat M jest automatem skończonym

$$M' = \langle Q, \Sigma, \delta', q_0, F\rangle$$

z odpowiednio zdefiniowaną funkcją przejścia δ':

$$\delta'(q,a) = \begin{cases} \delta(q,a) & \text{jeżeli } \delta(q,a) \neq q_1, \\ q_2 & \text{jeżeli } \delta(q,a) = q_1, \end{cases}$$

Łatwo zauważyć, że jeżeli $q_0 \neq q_1$, to stan q_1 automatu M' jest zbędny i, wobec tego, można go z automatu M' usunąć. Opisaną konstrukcję można uogólnić na przypadek większej liczby par stanów zachowując podane własności.

Aby skonstruować automat minimalny akceptujący język $L(M)$

1. obliczamy relację R,
2. z każdej klasy abstrakcji relacji R wybieramy po jednym stanie (dobrze jest wybrać stan początkowy z klasy, do której on należy),
3. a następnie stosujemy wyżej opisaną konstrukcję do każdych dwóch różnych stanów q_1, q_2 takich, że stan q_2 jest wybranym z klasy abstrakcji relacji R, wyznaczonej przez stan q_1,
4. i w końcu usuwamy z otrzymanego automatu stany zbędne.

Po wykonaniu opisanych wyżej czynności otrzymujemy automat M_m akceptujący język $L = L(M)$, o liczbie stanów najwyżej równej liczbie klas abstrakcji relacji R, a więc także najwyżej równej liczbie klas abstrakcji relacji R_L. Ponieważ automat M_m akceptuje język L, ma tyle stanów, ile klas abstrakcji ma relacja R_L. Teraz z Wniosku 7.5 wynika, że M_m jest minimalnym automatem akceptującym język L.

Wykład 8.

8.1 Gramatyki

Gramatyką nazywamy czwórkę

$$G \quad = \quad \langle V, T, P, S \rangle \tag{7}$$

taką, że

1. V jest skończonym alfabetem; elementy zbioru V nazywamy **zmiennymi** lub **symbolami nieterminalnymi**, a sam zbiór V – zbiorem zmiennych lub zbiorem symboli terminalnych,

2. T jest skończonym alfabetem rozłącznym z V; elementy T nazywamy **symbolami końcowymi** lub **terminalnymi**, a zbiór T zbiorem symboli końcowych lub symboli terminalnych,

3. P jest skończonym zbiorem par słów nad alfabetem $V \cup T$, o niepustej pierwszej współrzędnej (a więc $P \subseteq (V \cup T)^+ \times (V \cup T)^*$); P nazywamy zbiorem produkcji, elementy P – **produkcjami**,

4. S jest jedną ze zmiennych; nazywamy ją **symbolem początkowym**.

Umawiamy się, że będziemy pisać $s_1 \to s_2$ zamiast $(s_1, s_2) \in P$. Jeżeli będziemy rozważać jednocześnie kilka gramatyk, to wzór $s_1 \to s_2$ będziemy uzupełniać o symbol gramatyki lub zbioru produkcji, na przykład $s_1 \to_P s_2$ lub $s_1 \to_G s_2$.

Słowo $y \in (V \cup T)^*$ **wyprowadzamy** w gramatyce G (patrz (7)) ze słowa $x \in (V \cup T)^*$ **w jednym kroku**, jeżeli istnieją słowa z_1, z_2, s_1, $s_2 \in (V \cup T)^*$ takie, że $s_1 \to_G s_2$ oraz

$$x = z_1 s_1 z_2 \quad \text{i} \quad y = z_1 s_2 z_2.$$

Relację wyprowadzalności w jednym kroku oznaczamy symbolem $\Rightarrow$ albo – jeżeli jest konieczne – symbolem $\Rightarrow_G$. Tak więc zapis $x \Rightarrow y$ oznacza, że y wyprowadzamy w jednym kroku ze słowa x (w gramatyce G).

Symbolem $\Rightarrow^*$ oznaczamy przechodnie domknięcie relacji $\Rightarrow$. Jest to najmniejsza przechodnia i zwrotna relacja w zbiorze słów nad alfabetem $V \cup T$ zawierająca relację $\Rightarrow$. Inaczej relację $\Rightarrow^*$ definiujemy jako najmniejszą spośród relacji $R \subseteq ((V \cup T)^*)^2$ spełniających dla dowolnych słów x, y, $z \in (V \cup T)^*$

1. xRx,

2. jeżeli xRy i $y \Rightarrow z$, to xRz.

Jeżeli $x \Rightarrow^* y$, to o słowie y mówimy, że daje się **wyprowadzić** (w gramatyce G) ze słowa x. Tę własność będziemy zapisywać jako $\Rightarrow^*_G$ w sytuacjach, gdy będziemy zajmować się jednocześnie kilkoma gramatykami.

Relacja $\Rightarrow^*$ jest oczywiście przechodnia.

O słowie x mówimy, że jest **generowane przez gramatykę** G, jeżeli $S \Rightarrow_G^* x$. Językiem **generowanym przez gramatykę** G nazywamy

$$L(G) \quad = \quad \{x \in T^* : \ S \Rightarrow_G^* x\}.$$

Tak więc do $L(G)$ należę dokładnie te słowa utworzone z symboli terminalnych, które można wyprowadzić z symbolu początkowego S gramatyki G.

Przykład 8.1 Przypuśćmy, że istnieje algorytm stwierdzający, czy dane słowo $x \in \Sigma^*$ należy do języka L. Wtedy $L = L(G)$ dla pewnej gramatyki G. Uzasadnienie tego wykracza poza ramy wykładu z języków formalnych. Ma to dość ważne konsekwencje. Aby rozwiązywać z pomocą komputera dowolne problemy wystarczy umieć odpowiadać na pytanie dane słowo x jest generowane przez daną gramatykę G. Natomiast programowanie może polegać na definiowaniu gramatyki generującej interesujący nas język.

Przykład 8.2 Język

$$\{0^n 1^n : \ n \in N\}$$

jest generowany przez gramatykę

$$G \quad = \quad \langle \{S\}, \{0,1\}, P, S \rangle$$

ze zbiorem P z dwoma produkcjami

$$S \to 0S1,$$
$$S \to \varepsilon.$$

Uzasadnienie tego wymaga spostrzeżenia, że $S \Rightarrow^* 0^n S 1^n$ dla dowolnej liczby naturalnej n. Fakt ten bez trudu dowodzimy przez indukcję. Stąd natychmiast otrzymujemy, że wszystkie słowa postaci $0^n 1^n$ są generowane przez gramatykę G.

Aby wykazać, że gramatyka G nie generuje żadnych innych słów nad alfabetem T rozważmy relację R taką, że xRy jest równoważne

$$x \neq S \vee \exists n \in N \ (y = 0^n 1^n \vee y = 0^n S 1^n).$$

Można dowieść, że ta relacja ma obie własności podane w definicji relacji $\Rightarrow^*$. Jeżeli tak jest, to $\Rightarrow^* \subseteq R$, a także

$$S \Rightarrow^* y \quad \Longrightarrow \quad \exists n \in N \ (y = 0^n 1^n \vee y = 0^n S 1^n).$$

Wynika stąd, że słowa $y \in T^*$ generowane przez gramatykę G są postaci $0^n 1^n$ dla pewnej liczby $n \in N$.

8.2 Gramatyki a języki regularne

Niech

$$M \quad = \quad \langle Q, \Sigma, \delta, q_0, F \rangle$$

będzie automatem skończonym. Zdefiniujemy gramatykę generującą język $L(M)$. Będzie to gramatyka

$$G \quad = \quad \langle Q, \Sigma, P, q_0 \rangle,$$

gdzie P składa się z produkcji $q \to aq'$ dla wszystkich q, $q' \in Q$ i $a \in \Sigma$ takich, że $\delta(q, a) = q'$ oraz z produkcji $q \to \varepsilon$ dla wszystkich $q \in F$. Tak więc

$$P \quad = \quad \{(q, aq') \in Q \times \Sigma Q : \ \delta(q, a) = q'\} \cup \{(q, \varepsilon) : \ q \in F\}.$$

Aby wykazać, że $L(M) \subseteq L(G)$ zauważmy, że jeżeli $\widehat{\delta}(q, x) = q'$, to $q \Rightarrow^* xq'$ dla dowolnych q, $q' \in Q$ i $x \in \Sigma^*$. Tę własność dowodzimy przez indukcję ze względu na $x \in \Sigma^*$. Teraz zawieranie $L(M) \subseteq L(G)$ jest już łatwe do udowodnienia. Jeżeli $x \in L(M)$, to także $\widehat{\delta}(q_0, x) = q'$ dla pewnego stanu $q' \in F$. Z powyższej własności otrzymujemy, że $q_0 \Rightarrow^* xq'$. Zachodzi także $xq' \Rightarrow x$, gdyż $q' \in F$. Wobec tego $q_0 \Rightarrow^* x$, a to oznacza, że $x \in L(G)$.

Aby wykazać zawieranie przeciwne rozważmy relację $R \subseteq ((Q \cup \Sigma)^*)^2$ taką, że

$$xRy \quad \Longleftrightarrow \quad x \notin Q \vee y \in \Sigma^* \vee \exists y' \in \Sigma^* \exists q' \in Q \ (y = y'q' \wedge \widehat{\delta}(x, y') = q'). \tag{8}$$

O relacji R dowodzimy, że ma własności podane w definicji relacji $\Rightarrow^*$. Stąd otrzymujemy, że $\Rightarrow^* \subseteq R$.

Przypuśćmy, że $x \in L(G)$. Słowo x wyprowadzamy z q_0 w przynajmniej jednym kroku ponieważ $x \neq q_0$. Istnieje więc słowo $y \in (Q \cup \Sigma)^*$ takie, że

$$q_0 \Rightarrow^* y \Rightarrow x.$$

Ponieważ $x \in \Sigma^*$, więc wyprowadzając x z y musimy stosować jedną z produkcji postaci $q \to \varepsilon$ dla pewnego $q \in F$. Wobec tego stan q występuje w słowie y. Własność $q_0 \Rightarrow^* y$ implikuje, że $q_0 R y$. Nietrudno zauważyć, że rozważanej sytuacji nie zachodzą dwa pierwsze człony alternatywy z definicji relacji R (patrz (8)). Zachodzi więc trzeci z tych członów. Tak więc $y = y'q$ dla pewnego $y' \in \Sigma^*$ spełniającego $\widehat{\delta}(q_0, y') = q$ (stan q' z definicji R musi być równy q). Ale $q \in F$, więc mamy $y' \in L(M)$. Natomiast warunek $y'q = y \Rightarrow x$ implikuje, że słowo y' jest równe x. Tak więc $x \in L(M)$.

Powyższe rozumowanie dowodzi, że $L(G) = L(M)$.

8.3 Najprostsze własności

Czasem nieco dokładniej będziemy analizować pojęcie wyprowadzenia. Wprowadźmy więc relację $\Rightarrow^n$. Dla dowolnych słów α i β, relacja $\alpha \Rightarrow^0 \beta$ zachodzi wtedy i tylko wtedy, gdy $\alpha = \beta$. Dalej relację $\Rightarrow^n$ definiujemy przez indukcję tak, aby

$$\alpha \Rightarrow^{n+1} \beta$$

było równoważne istnieniu słowa γ spełniającego

$$\alpha \Rightarrow^n \gamma \ \wedge \ \gamma \Rightarrow \beta.$$

Jeżeli $\alpha \Rightarrow^n \beta$, to będziemy mówić, że β daje się **wyprowadzić z** α **w** n **krokach**.

Sformulujemy teraz najprostsze własności pojęcia wyprowadzenia. Dowody tych własności zostaną pominięte.

Lemat 8.3 *Jeżeli* $x \Rightarrow^n y$, *to* $z_1xz_2 \Rightarrow^n z_1yz_2$, *dla dowolnych słów* x, y, z_1 *i* z_2.

Wniosek 8.4 *Jeżeli* $x \Rightarrow^* y$, *to* $z_1xz_2 \Rightarrow^* z_1yz_2$, *dla dowolnych słów* x, y, z_1 *i* z_2.

Wniosek 8.5 *Jeżeli* $x_1 \Rightarrow^n y_1$ *i* $x_2 \Rightarrow^m y_2$, *to* $x_1x_2 \Rightarrow^{n+m} y_1y_2$, *dla dowolnych słów* x_1, y_1, x_2 *i* y_2.

Wniosek 8.6 *Jeżeli* $x_1 \Rightarrow^* y_1$ *i* $x_2 \Rightarrow^* y_2$, *to* $x_1x_2 \Rightarrow^* y_1y_2$, *dla dowolnych słów* x_1, y_1, x_2 *i* y_2.

8.4 Gramatyki bezkontekstowe

Gramatykę $G = \langle V, T, P, S\rangle$ nazywamy **bezkontekstową**, jeżeli wszystkie produkcje gramatyki G są postaci $A \rightarrow_G \alpha$ dla $A \in V$.

Język L jest **bezkontekstowy**, jeśli jest generowany przez gramatykę bezkontekstową.

W rozdziale 8.2 zostały zdefiniowane gramatyki regularne, generujące języki regularne. Jest oczywiste, że są to gramatyki bezkontekstowe. Tak więc języki regularne są bezkontekstowe.

Natomiast w Przykładzie 8.2 zostało wykazane, że język $\{0^n1^n : \ n \in N\}$ jest bezkontekstowy. Wiadomo, że nie jest to język regularny. Istnieją więc języki bezkontekstowe, które nie są regularne.

Bardzo ważna własność języków bezkontekstowych jest treścią poniższego lematu.

Lemat 8.7 *Jeżeli gramatyka* G *jest bezkontekstowa i* $x_1x_2 \Rightarrow_G^n y$, *to istnieją słowa* y_1 *i* y_2 *oraz liczby* n_1 *i* n_2 *takie, że* $y = y_1y_2$, $n = n_1 + n_2$, $x_1 \Rightarrow_G^{n_1} y_1$ *oraz* $x_2 \Rightarrow_G^{n_2} y_2$.

Dowód. Dowód tego Lematu zostanie pominięty. Dowodzimy go przez indukcję ze względu na n. Teza Lematu 8.7 wynika ze spostrzeżenia, że w gramatyce bezkontekstowej, w jednym kroku ze słowa x_1x_2 wyprowadzamy słowa postaci yx_2 lub x_1y, dla odpowiedniego słowa y. □

Wniosek 8.8 *Jeżeli gramatyka G jest bezkontekstowa i $x_1x_2 \Rightarrow_G^* y$, to istnieją słowa y_1 i y_2 takie, że $y = y_1y_2$, $x_1 \Rightarrow_G^* y_1$ oraz $x_2 \Rightarrow_G^* y_2$.*

Wykład 9.

Naszym celem jest teraz pokazanie, że gramatykom bezkontekstowym można nadać prostą w pewnym sensie lub wygodną z jakiś względów postać. Wymaga to dość żmudnego przekształcania gramatyk.

9.1 Usuwanie symboli zbędnych

Konstrukcja 9.1 *Przypuśćmy, że dana jest bezkontekstowa gramatyka*

$$G \quad = \quad \langle V, T, P, S\rangle.$$

Niech V' będzie najmniejszym spośród zbiorów $X \subseteq V$ spełniających

jeżeli $A \rightarrow_G \gamma$ jest produkcją taką, że $\gamma \in (X \cup T)^*$, to $A \in X$.

Niech P' będzie zbiorem tych wszystkich produkcji ze zbioru P, w których lewa i prawa strona są słowami nad alfabetem $V' \cup T$. Chcielibyśmy rozważać gramatykę

$$G' \quad = \quad \langle V', T, P', S\rangle.$$

Aby G' było gramatyką, symbol początkowy S powinien należeć do zbioru zmiennych V'. Ten warunek nie musi być spełniony. Tak jest na przykład w przypadku gramatyki $G = \langle\{S\}, \{a\}, \{S \rightarrow S\}, S\rangle$. Dla tej gramatyki zbiór V' oraz język przez nią generowany są puste. Symbol początkowy jednak jest potrzebny tylko wtedy, gdy rozważany język generowany przez gramatykę. Nie jest on potrzebny, gdy analizujemy pojęcie wyprowadzenia.

Zauważmy, że

- jeżeli $\alpha \Rightarrow_G^* \beta$ i $\beta \in (V' \cup T)^*$, to $\alpha \in (V' \cup T)^*$ i $\alpha \Rightarrow_{G'}^* \beta$.

Aby to uzasadnić, wystarczy dowieść analogiczną własność relacji $\Rightarrow_G$ i $\Rightarrow_{G'}$. Jeżeli $\beta \in (V' \cup T)^*$ zostało wyprowadzone z α, to jest produkcja $A \rightarrow_G \gamma$ i są słowa α' i α'' takie, że $\alpha = \alpha' A \alpha''$ oraz $\beta = \alpha'\gamma\alpha''$. Ponieważ $\beta \in (V' \cup T)^*$, więc także α', γ i $\alpha'' \in (V' \cup T)^*$, a – na mocy definicji zbioru V' – również $A \in V'$. Stąd oczywiście otrzymujemy, że $\alpha \in (V' \cup T)^*$ oraz $A \rightarrow_{G'} \gamma$. A to także implikuje, że $\alpha \Rightarrow_{G'} \beta$.

Udowodniona własność ma dwie konsekwencje. Po pierwsze, jeżeli język $L(G)$ jest niepusty, to $S \in V'$ i w konsekwencji nic nie stoi na przeszkodzie, aby zajmować się językiem $L(G')$ Po drugie, przy tym założeniu mamy także zawieranie $L(G) \subseteq L(G')$.

Natomiast zawieranie przeciwne $L(G') \subseteq L(G)$ jest konsekwencją następującego lematu będącego prostym wnioskiem z definicji relacji wyprowadzenia.

Lemat 9.2 *Jeżeli symbole terminalne i nieterminalne oraz produkcje gramatyki G' są odpowiednio symbolami bądź produkcjami gramatyki G, a ponadto obie gramatyki mają ten sam symbol początkowy, to $L(G') \subseteq L(G)$.* □

Z definicji zbioru V' otrzymujemy także

- jeżeli $A \in V'$, to $A \Rightarrow_G^* \alpha$ dla pewnego $\alpha \in T^*$.

Oznacza to, że $L(G)$ jest niepusty zawsze wtedy, gdy $S \in V'$. Tak więc należenie symbolu początkowego S do V' jest równoważne niepustości języka $L(G)$.

Z obu powyższych własności wynika, że w gramatyce G' z dowolnego symbolu nieterminalnego można wyprowadzić pewne słowo zapisane przy użyciu tylko symboli terminalnych. Tak więc zachodzi następujący

Lemat 9.3 *Dla dowolnej bezkontekstowej gramatyki G generującej niepusty język istnieje gramatyka G' generująca język $L(G)$, w której z dowolnego symbolu nieterminalnego można wyprowadzić pewne słowo zapisane przy użyciu wyłącznie symboli terminalnych.*

Konstrukcja 9.4 *Przypuśćmy, że dana jest bezkontekstowa gramatyka*

$$G \quad = \quad \langle V, T, P, S \rangle.$$

Niech teraz U będzie najmniejszym spośród zbiorów $X \subseteq V \cup T$ spełniających

1. *$S \in X$,*
2. *jeżeli $A \in X$ i $A \rightarrow_G \gamma$, to $\gamma \in X^*$.*

Niech P' będzie zbiorem tych wszystkich produkcji ze zbioru P, w których lewa i prawa strona są słowami nad alfabetem U. Zbiór P' można więc zdefiniować jako $P \cap (U^*)^2$. Rozważmy gramatykę

$$G' \quad = \quad \langle V \cap U, T \cap U, P', S \rangle.$$

Tak zdefiniowana gramatyka ma następujące własności

- jeżeli $\alpha \Rightarrow_G^* \beta$ i $\alpha \in U^*$, to $\beta \in U^*$ i $\alpha \Rightarrow_{G'}^* \beta$.
- jeżeli $x \in U$, to $S \Rightarrow_G^* \alpha x \beta$ dla pewnych α, $\beta \in U^*$,

Własności te dowodzimy tak, jak analogiczne własności poprzedniej konstrukcji. Pierwsza z tych własności implikuje, że $L(G) \subseteq L(G') \subseteq (T \cap U)^*$. Prawdziwe jest też zawieranie $L(G') \subseteq L(G)$ i jest to konsekwencja Lematu 9.2. Zauważmy też, że zbiór U jest sumą alfabetów złożonych z symboli terminalnych i nieterminalnych gramatyki G' ($U = (V \cap U) \cup (T \cap U)$).

Dla danej gramatyki $G = \langle V, T, P, S\rangle$, o symbolu $x \in V \cup T$ mówimy, że jest **niezbędny**, jeżeli

- istnieje słowo $\alpha \in T^*$ takie, że $x \Rightarrow_G^* \alpha$ oraz
- istnieją słowa α, $\beta \in (V \cup T)^*$ takie, że $S \Rightarrow_G^* \alpha x \beta$.

Symbole, które nie są niezbędne nazywamy **zbędnymi**.

Lemat 9.5 *Dla dowolnej bezkontekstowej gramatyki G generującej niepusty język istnieje bezkontekstowa gramatyka bez symboli zbędnych generująca ten sam język $L(G)$.*

Dowód. Wobec Lematu 9.3, możemy dodatkowo założyć, że w gramatyce $G = \langle V, T, P, S\rangle$ z dowolnego symbolu nieterminalnego daje się wyprowadzić pewne słowo zapisane wyłącznie przy użyciu symboli terminalnych. Do takiej gramatyki stosujemy Konstrukcję 2. W otrzymana w ten sposób gramatyka $G' = \langle V', T', P', S\rangle$ generuje język $L(G)$ i nie ma symboli zbędnych.

Aby przekonać się o tej ostatniej własności zauważmy, że dla dowolnego symbolu $x \in V' \cup T'$, w gramatyce G z symbolu początkowego S daje się wyprowadzić słowo postaci $\alpha x \beta$, a z symbolu x daje się wyprowadzić słowo $\gamma \in T^*$, tak więc

$$S \Rightarrow_G^* \alpha x \beta \wedge x \Rightarrow_G^* \gamma \in T^*$$

dla pewnych słów α, β i γ. Konstrukcja 9.4 gwarantuje jednak, że wyprowadzenia w gramatyce G z symboli gramatyki G' są także wyprowadzeniami w gramatyce G'. Ostatecznie

$$S \Rightarrow_{G'}^* \alpha x \beta \wedge x \Rightarrow_{G'}^* \gamma \in (T')^*$$

dla pewnych słów α, β i γ. Oznacza to, że w gramatyce G' nie am symboli zbędnych. □

9.2 Usuwanie ε-produkcji

Produkcje postaci $\alpha \to \varepsilon$ nazywamy **ε-produkcjami**. Nietrudno zauważyć, że ε-produkcje są potrzebne do wyprowadzenia słowa pustego. Dla dowolnej gramatyki $G = \langle V, T, P, S\rangle$, gramatyka

$$G' \quad = \quad \langle V \cup \{S'\}, T, P \cup \{S' \to \varepsilon, S' \to S\}, S'\rangle$$

z nowym symbolem początkowym $S' \notin V$ generuje język $L(G') = L(G) \cup \{\varepsilon\}$. Pokażemy, że w innych sytuacjach ε-produkcje nie są potrzebne.

Symbol nieterminalny X nazywamy **zerowalnym**, jeżeli $X \Rightarrow^* \varepsilon$. Mówimy, że słowo α' **różni się** od słowa α symbolami zerowalnymi, jeżeli α' powstaje z α przez usunięcie niektórych symboli zerowalnych. Na przykład, jeżeli Z jest symbolem zerowalnym, to $abaZbbZ$ różni się od $ZaZbaZbbZZ$ symbolami zerowalnymi. Zakładamy też, że każde słowo różni się symbolami zerowalnymi od siebie.

Konstrukcja 9.6 *Przypuśćmy, że mamy gramatykę*

$$G \quad = \quad \langle V, T, P, S \rangle.$$

Niech

$$P' \quad = \quad \{\alpha \to \gamma : \ \gamma \neq \varepsilon \wedge \exists \beta \in (V \cup T)^* \ (\alpha \to_G \beta \wedge \\ \gamma \textit{ różni się symbolami zerowalnymi w } G \textit{ od } \beta)\}$$

Będziemy rozważać gramatykę

$$G' \quad = \quad \langle V, T, P', S \rangle.$$

Jest oczywiste, że gramatyka G' nie zawiera ε-produkcji i jest bezkontekstowa, jeżeli bezkontekstowa jest gramatyka G. Łatwo zauważyć, że jeżeli γ różni się od β symbolami zerowalnymi w G, to $\beta \Rightarrow_G^* \gamma$. Wynika stąd, że jeżeli $\alpha \to G'\gamma$, to $\alpha \Rightarrow_G^* \gamma$. I co więcej, jeżeli $\alpha \Rightarrow_{G'}^* \gamma$, to $\alpha \Rightarrow_G^* \gamma$. Wobec tego $L(G') \subseteq L(G)$. Ponieważ wyprowadzenia słowa pustego wymaga użycia przynajmniej raz ε-produkcji, więc także $L(G') \subseteq L(G) \setminus \{\varepsilon\}$.

Również prawdziwe jest zawieranie odwrotne. Aby je wykazać zauważmy, że jeżeli $S \Rightarrow_G^* \beta$ dla niepustego słowa β, to także $S \Rightarrow_{G'}^* \beta$. Dowodzimy to przez indukcję. Jest to prawda, jeżeli β jest równe S. Ponadto, w sytuacji, gdy $S \Rightarrow_G^* \beta' \Rightarrow_G \beta$, słowo β' jest niepuste i na mocy założenia indukcyjnego możemy przyjąć, że $S \Rightarrow_{G'}^* \beta'$ i rozważamy dwa przypadki zależnie od produkcji wykorzystanej do wyprowadzenia β z β'. Jeżeli wyprowadzenie to nie wymagało stosowania ε-produkcji, to korzystaliśmy z produkcji, która należy także P', i w konsekwencji $\beta' \Rightarrow_{G'} \beta$. Jeżeli wyprowadzenie β z β' w gramatyce G wymagało zastosowania ε-produkcji, to długość β' jest większa lub równa 2, a więc wyprowadzenie β' z S w gramatyce G' wymaga użycia przynajmniej jednej produkcji. przyjmijmy więc, że $S \Rightarrow_{G'}^* \beta'' \Rightarrow_{G'} \beta'$. Nietrudno zauważyć, że wyprowadzenie polegające na zastosowaniu producji ze zbioru P' i ε-produkcji z gramatyki G może zostać wykonane przy użyciu jednej produkcji ze zbioru P'. Tak więc $\beta'' \Rightarrow_{G'} \beta$ i ostatecznie $S \Rightarrow_{G'}^* \beta$.

Powyższa implikacja pozwala na wykazanie, że niepuste słowa z języka $L(G)$ należą do $L(G')$. Tak więc $L(G') = L(G) \setminus \{\varepsilon\}$.

Lemat 9.7 *Dla dowolnego bezkontekstowego języka L, język $L \setminus \{\varepsilon\}$ jest generowany przez bezkontekstową gramatykę bez ε-produkcji.* □

Wykład 10.

10.1 Dodawanie produkcji jednostkowych

Konstrukcja 10.1 *Przypuśćmy, że mamy gramatykę*

$$G = \langle V, T, P, S \rangle.$$

Będziemy rozważać gramatykę

$$G' = \langle V \cup \{B\}, T, P', S \rangle,$$

w której B jest nową zmienną, a

$$P' = P \cup \{B \to \beta_1, B \to \beta_2, \ldots, B \to \beta_n\}$$

dla pewnych słów β_1, β_2, …, $\beta_n \in (V \cup \{B\} \cup T)^*$.

Nietrudno zauważyć, że jeżeli $\alpha \in V \cup T$ i $\alpha \Rightarrow^*_{G'} \beta$, to $\beta \in V \cup T$ i $\alpha \Rightarrow^*_G \beta$. Wynika to stąd, że żadna produkcją gramatyki G' pozwalająca na zastąpienie zmiennej ze zbioru V nie wprowadza do przekształcanego słowa zmiennej B i, wobec tego, nie pozwala na późniejsze zastosowanie produkcji należących do P' i nie należących do P. Oczywistą konsekwencją tej własności jest zawieranie $L(G') \subseteq L(G)$. Zawieranie przeciwne jest prostym wnioskiem z zawierania $P \subseteq P'$. Tak więc obie gramatyki generują ten sam język.

Konstrukcja 10.2 *Przypuśćmy, że mamy gramatykę*

$$G = \langle V, T, P, S \rangle,$$

oraz produkcję $A \to_G \alpha_1 B \alpha_2$. Będziemy rozważać gramatykę

$$G' = \langle V, T, P', S \rangle$$

taką, że

$$\begin{aligned} P' = {} & (P \setminus \{A \to_G \alpha_1 B \alpha_2\}) \cup \\ & \{A \to \alpha_1 \beta_1 \alpha_2, A \to \alpha_1 \beta_2 \alpha_2, \ldots, A \to \alpha_1 \beta_n \alpha_2\} \end{aligned}$$

gdzie β_1, β_2, …, β_n są wszystkimi słowami $\beta \in (V \cup T)^*$ takimi, że $B \to_G \beta$.

Obie gramatyki generują ten sam język. Jest proste do zaobserwowowania, że jeżeli $\alpha \to_{G'} \beta$, to $\alpha \Rightarrow^*_G \beta$. Implikuje to, że $\alpha \Rightarrow^*_G \beta$ zawsze wtedy, gdy $\alpha \Rightarrow^*_{G'} \beta$. Tak więc $L(G') \subseteq \text{Ł}(G)$.

Dowód zawierania przeciwnego jest bardziej skomplikowany. Najpierw zauważmy, że

Lemat 10.3 *Przypuśćmy, że symbole terminalne i nieterminalne gramatyki G są odpowiednio symbolami terminalnymi i nieterminalnymi gramatyki G'. Jeżeli G jest gramatyką bezkontekstową oraz dla wszystkich $n \leq N$, zmiennych X i słów β warunek $X \Rightarrow_G^n \beta$ implikuje $X \Rightarrow_{G'}^* \beta$, to dla wszystkich $n \leq N$ i słów α, β warunek $\alpha \Rightarrow_G^n \beta$ implikuje $\alpha \Rightarrow_{G'}^* \beta$.*

Dowód. Zgodnie z Lematem 8.7, wyprowadzenie w gramatyce G słowa β ze słowa α można podzielić na wyprowadzenia z poszczególnych liter słowa α fragmentów słowa β, przy czym wyprowadzenia fragmentów nie są dłuższe od wyprowadzenia całego słowa β. Wobec tego, te fragmenty dają się wyprowadzić także z odpowiednich liter w gramatyce G' i ich wyprowadzenia w gramatyce G' dają się połączyć w wyprowadzenie słowa β ze słowa α (patrz Wniosek 8.5). □

Teraz pokażemy przez indukcję ze względu na n, że jeżeli $\beta \in T^*$ i $X \Rightarrow_G^n \beta$, to $X \Rightarrow_{G'}^* \beta$. Załóżmy więc, że własność ta jest prawdziwa dla liczb mniejszych od n. Ponieważ dla $n = 0$ dowodzona własność jest oczywista, możemy jeszcze założyć, że $n > 0$. Dla pewnej produkcji $X \rightarrow_G \alpha$ z gramatyki G zachodzi $X \rightarrow_G \alpha \Rightarrow_G^{n-1} \beta$. Jeżeli jest to produkcja różna od $A \rightarrow_G \alpha_1 B \alpha_2$, to jest to także produkcja z gramatyki G' oraz, korzystając z założenia indukcyjnego i Lematu 10.3, mamy $X \rightarrow_{G'} \alpha \Rightarrow_{G'}^{n-1} \beta$.

Dowód należy więc jeszcze przeprowadzić w przypadku, gdy $X = A$ i $A \rightarrow_G \alpha_1 B \alpha_2 \Rightarrow_G^{n-1} \beta$. Korzystając z Lematu 8.7 słowo β przedstawiamy jako $\beta'\beta''\beta'''$ tak, że $\alpha_1 \Rightarrow_G^{m'} \beta'$, $B \Rightarrow_G^{m''} \beta''$ oraz $\alpha_2 \Rightarrow_G^{m'''} \beta'''$. Ponieważ $\beta'' \in T^*$, wyprowadzenie β'' z B wymaga przynajmniej jednego kroku. Dla pewnego i mamy więc $B \rightarrow_G \beta_i \Rightarrow_G^{m''-1} \beta''$. Na podstawie założenia indukcyjnego, korzystając także z Lematu 10.3, otrzymujemy, że $\alpha_1 \Rightarrow_{G'}^{m'} \beta'$, $\beta_i \Rightarrow_{G'}^{m''-1} \beta''$ i $\alpha_2 \Rightarrow_{G'}^{m'''} \beta'''$. Wobec tego

$$X = A \rightarrow_{G'} \alpha_1\beta_i\alpha_2 \Rightarrow_{G'}^* \beta'\beta_i\alpha_2 \Rightarrow_{G'}^* \beta'\beta''\alpha_2 \Rightarrow_{G'}^* \beta'\beta''\beta''' = \beta.$$

Z udowodnionej własności wynika, że $L(G) \subseteq L(G')$. Tak więc mamy $L(G) = L(G')$.□

Konstrukcje 10.1 i 10.2 pozwalają na dodawanie produkcji jednostkowych. W kolejnym lemacie wykorzystamy to usunięcia z gramatyki niejednostkowych produkcji zawierających symbole terminalne.

Lemat 10.4 *Każdy niepusty język bezkontekstowy L taki, że $\varepsilon \notin L$ jest generowany przez bezkontekstową gramatykę $G = \langle V, T, P, S\rangle$ spełniającą dla wszystkich $X \in V$ i $\alpha \in (V \cup T)^*$*

$$\textit{jeżeli } X \rightarrow_G \alpha, \textit{ to } \alpha \in V^+ \vee \alpha \in T.$$

Dowód. Bezkontekstowy język L spełniający założenia Lematu 10.4 jest generowany przez gramatykę bez ε-produkcji. Niech T będzie zbiorem symboli terminalnych tej gramatyki. Powiększamy zbiór jej symboli nieterminalnych i

produkcji dodając dla każdego $a \in T$ nowy symbol T_a oraz produkcję $T_a \to a$. Przypuśćmy, że otrzymaliśmy w ten sposób gramatykę G. Korzystając z własności konstrukcji 10.1 stwierdzamy, że $L = L(G)$. Jest oczywiste, że w gramatyce G każdy z symboli T_a możemy zastąpić w jednym kroku tylko słowem (symbolem) a.

Zdefiniujemy teraz pewną operację, która przekształca gramatykę G bez ε-produkcji, pozwalającą na zastąpienie w jednym kroku symboli T_a tylko symbolami a, w gramatykę G_1 o tych samych własnościach, generującą ten sam język. W gramatyce G wybieramy niejednostkową produkcję $X \to_G \alpha_1 a \alpha_2$ z symbolem terminalnym a po prawej stronie i zastępujemy ją produkcją $X \to \alpha_1 T_a \alpha_2$. Otrzymaną w ten sposób gramatykę oznaczamy przez G_1. Oczywiście, przekształcając w ten sposób gramatykę G nie dodajemy produkcji postaci $T_a \to \ldots$, ani ε-produkcji. Jeżeli do gramatyki G_1 i produkcji $X \to_{G_1} \alpha_1 T_a \alpha_2$ zastosujemy konstrukcję 10.2, to otrzymamy znowu gramatykę G. Wobec tego $L(G) = L(G_1)$. Nietrudno też zauważyć, że liczba symboli terminalnych w niejednostkowych produkcjach gramatyki G_1 jest mniejsza od liczby takich symboli w gramatyce G.

Teraz łatwo opisać algorytm przekształcający gramatykę G w gramatykę o własnościach podanych w tezie Lematu 10.4. Może to być następujący algorytm (gramatyka G jest początkową wartością zmiennej G):

- **dopóki w gramatyce G jest niejednostkowa produkcja z symbolem terminalnym,**
    - **zastąp G przez wyżej zdefiniowaną gramatykę G_1.**

Algorytm ten kończy pracą i wtedy wartością zmiennej G jest gramatyka o żądanych własnościach. □

10.2 Usuwanie produkcji jednostkowych

Najpierw zauważmy, że

Lemat 10.5 *Dla dowolnej gramatyki $G = \langle V, T, P, S\rangle$ i dla dowolnego słowa α, gramatyka $G' = \langle V, T, P \setminus \{\alpha \to \alpha\}, S\rangle$ generuje język $L(G)$.*

Dowód. Oczywiście, każde wyprowadzenie w gramatyce G' jest także wyprowadzeniem w gramatyce G, więc $L(G') \subseteq L(G)$.

Nietrudno też zauważyć, że w najkrótszym wyprowadzeniu w gramatyce G słowa γ z słowa β, $\beta \neq \gamma$, nie korzystamy z produkcji $\alpha \to \alpha$. Ponadto wyprowadzenia, w których nie korzystamy z tej produkcji, są także wyprowadzeniami w gramatyce G'. Tak więc każde słowo nad alfabetem T, które można wyprowadzić z symbolu początkowego S w gramatyce G, daję się także wyprowadzić z symbolu S w gramatyce G'. Tak więc $L(G) \subseteq L(G')$. □

Lemat 10.6 *Każdy niepusty język bezkontekstowy L taki, że $\varepsilon \notin L$ jest generowany przez bezkontekstową gramatykę $G = \langle V, T, P, S\rangle$ spełniającą dla wszystkich $X \in V$ i $\alpha \in (V \cup T)^*$ warunek*

$$\text{jeżeli } X \to_G \alpha, \text{ to } \alpha \in VV^+ \vee \alpha \in T.$$

Dowód. Niech $G\langle V, T, P, S\rangle$ będzie gramatykę, której istnienie wynika z Lematu 10.4. Podamy teraz algorytm pozwalający na przekształcenie gramatyki G w gramatykę o własnościach podanych w Lemacie 10.6. Zakładam, że mam zmienną G, której wartością jest gramatyka G.

- **dla każdej zmiennej** $A \in V$
  - **zastąp gramatykę** G **gramatykę powstającą z** G **przez usunięcie produkcji** $A \to A$**,**
  - **dopóki w gramatyce** G **jest produkcja postaci** $X \to A$
    - **zastąp gramatykę** G **gramatyką** G' **zgodnie z konstrukcją 10.2 zastosowaną do produkcji** $X \to A$**.**

10.3 Gramatyki w postaci normalnej Chomsky'ego

Twierdzenie 10.7 *Każdy niepusty język bezkontekstowy L taki, że $\varepsilon \notin L$, jest generowany przez bezkontekstową gramatykę w postaci normalnej Chomsky'ego.*

Dowód. Dla języka L weźmy gramatykę G spełniającą tezę Lematu 10.6. Indeksem produkcji $X \to \alpha$ będziemy nazywać większą z liczb 0 i $|\alpha| - 2$. Tak więc indeks każdej produkcji w gramatyce w postaci normalnej Chomsky'ego jest liczba 0. Indeksem gramatyki G będziemy nazywać sumę indeksów produkcji gramatyki G. Wobec tego, indeks gramatyki w postaci normalnej Chomsky'ego jest równy 0. Prawdziwa jest też następująca implikacja odwrotna. Jeżeli indeks bezkontekstowej gramatyki G jest równy 0 i G spełnia tezę Lematu 10.6, to G ma postać normalną Chomsky'ego. Tak implikacja jest oczywista. Jeżeli indeks gramatyki jest równy zero, to prawe strony produkcji tej gramatyki mają długość nie większą niż dwa. Jeżeli dodatkowo jest spełniona teza Lematu 10.6, to prawe strony produkcji mogą być albo dwuliterowymi słowami nad alfabetem zmiennych, albo symbolami terminalnymi.

Przekształćmy gramatykę G w następujący sposób:

- **dopóki indeks gramatyki** G **jest większy od 0**
  - **wybierz produkcję** $X \to A_1A_2A_3 \dots A_n$ **taką, że** $n > 2$
  - **zastąp** G **gramatyką powstającą z** G **przez dodanie do alfabetu zmiennych nowego symbolu** B **i powiększenie zbioru produkcji o** $B \to A_1A_2$**,**

- **zastąp G gramatyką powstającą z G przez usunięcie ze zbioru produkcji $X \to A_1A_2A_3 \dots A_n$ i dodanie do niego produkcji $X \to BA_3 \dots A_n$.**

Języki generowane przez gramatykę G przed i po wykonaniu każdego z podstawień są identyczne. W przypadku pierwszego podstawienia wynika to z własności konstrukcji 10.1. W przypadku drugiego podstawiania wynika to z własności konstrukcji 10.2 z tym, że stosujemy ją do gramatyki otrzymanej w wyniku podstawiania. Tak więc gramatyki pamiętane w G przed i po wykonaniu powyższego algorytmu generują ten sam język. Jest oczywiste, że pierwsze podstawianie nie zmienia indeksu gramatyki, a drugie – zmiejsza go o 1. Wobec tego, algorytm zawsze kończy pracę, a po jej zakończeniu gramatyka G ma indeks 0. Jest więc gramatyką w postaci normalnej Chomsky'ego. □

Wykład 11.

11.1 Konstrukcja pomocnicza

Konstrukcja 11.1 *Przypuśćmy, że dana jest gramatyka*

$$G \quad = \quad \langle V, T, P, S \rangle$$

i jedna ze zmiennych $A \in V$. Produkcje pozwalające na zastępowanie zmiennej A dzielimy na dwa rodzaje

$$A \to_G A\alpha_1,\ A \to_G A\alpha_2,\ \dots,\ A \to_G A\alpha_n$$

oraz

$$A \to_G \beta_1,\ A \to_G \beta_2,\ \dots,\ A \to_G \beta_m,$$

przy czym słowa β_1, $\beta_2, \dots$, β_m, nie zaczynają się symbolem A. Przyjmijmy, że P' oznacza zbiór pozostałych produkcji gramatyki G. Będziemy rozważać trzy gramatyki

$$G_i \quad = \quad \langle V \cup \{B\}, T, P_i, S \rangle$$

dla $i = 1, 2, 3$, gdzie B jest nową zmienną ($B \notin V$) oraz

$$\begin{aligned}
P_1 &= P' \cup \{A \to \beta_i B : i = 1, \dots, m\} \cup \{B \to B\alpha_i : i = 1, \dots, n\} \cup \{B \to \varepsilon\} \\
P_2 &= P' \cup \{A \to \beta_i B : i = 1, \dots, m\} \cup \{B \to \alpha_i B : i = 1, \dots, n\} \cup \{B \to \varepsilon\} \\
P_3 &= P' \cup \{A \to \beta_i : i = 1, \dots, m\} \cup \{A \to \beta_i B : i = 1, \dots, m\} \\
&\quad \cup \{B \to \alpha_i : i = 1, \dots, n\} \cup \{B \to \alpha_i B : i = 1, \dots, n\}.
\end{aligned}$$

Pokażemy, że wyżej zdefiniowane gramatyki generują język $L(G)$.

Dowód, że $L(G) \subseteq L(G_1)$**.** Pokażemy przez indukcję ze względu na k, że

$$X \Rightarrow_G^k \gamma \quad \Longrightarrow \quad X \Rightarrow_{G_1}^* \gamma$$

dla dowolnego $X \in V$ i dla dowolnego $\gamma \in T^*$. Jest oczywiste, że implikacja ta pozwala na uzasadnienie podanego zawierania i zachodzi dla $k = 0$. Załóżmy, że $k > 0$ i implikacja zachodzi dla liczb mniejszych od k.

Najpierw zajmiemy się przypadkiem, gdy $X \neq A$. Jeżeli $X \Rightarrow_G^k \gamma$, to $X \rightarrow_G \xi \Rightarrow_G^{k-1} \gamma$ dla pewnego ξ. Wtedy $X \rightarrow_{G_1} \xi$, a na mocy założenia indukcyjnego i Lematu 10.3 mamy także $\xi \Rightarrow_{G_1}^* \gamma$. Wobec tego $X \Rightarrow_{G_1}^* \gamma$.

Jeżeli $A \Rightarrow_G^k \gamma$, to albo $A \rightarrow_G \beta_i \Rightarrow_G^{k-1} \gamma$, albo $A \rightarrow_G A\alpha_i \Rightarrow_G^{k-1} \gamma$ dla pewnego i. Jeżeli zachodzi pierwsza z tych możliwości, dowód prowadzimy jak w poprzednim przypadku i korzystamy z oczywistego faktu, że $A \Rightarrow_{G_1}^2 \beta_i$. Jeżeli najpierw zastosowaliśmy produkcję $A \rightarrow_G A\alpha_i$, to słowo γ można podzielić na dwie części γ_1 i γ_2 tak, że $A \Rightarrow_G^{k_1} \gamma_1$ i $\alpha_i \Rightarrow_G^{k_2} \gamma_2$ dla pewnych liczb k_1 i k_2, mniejszych od k. Korzystając z założenia indukcyjnego i ewentualnie z Lematu 10.3 otrzymujemy $A \Rightarrow_{G_1}^* \gamma_1$ i $\alpha_i \Rightarrow_{G_1}^* \gamma_2$. Teraz analizujemy wyprowadzenie słowa γ_1 z symbolu A w gramatyce G_1. Najpierw zastosowaliśmy produkcję postaci $A \rightarrow_{G_1} \beta_j B$. Tak więc istnieją słowa γ_{11} i γ_{12} takie, że $\beta_j \Rightarrow_{G_1}^* \gamma_{11}$, $B \Rightarrow_{G_1}^* \gamma_{12}$. Aby zakończyć ten fragment rozumowania wystarczy zauważyć, że

$$A \rightarrow_{G_1} \beta_j B \Rightarrow_{G_1} \beta_j B\alpha_i \Rightarrow_{G_1}^* \gamma_{11} B\alpha_i \Rightarrow_{G_1}^* \gamma_{11}\gamma_{12}\alpha_i = \gamma_1\alpha_i \Rightarrow_{G_1}^* \gamma_1\gamma_2 = \gamma.$$

Tak więc $A \Rightarrow_{G_1}^* \gamma$.

Dowód, że $L(G_1) \subseteq L(G)$**.** Pokażemy przez indukcję ze względu na k, że

$$X \Rightarrow_{G_1}^k \gamma \quad \Longrightarrow \quad X \Rightarrow_G^* \gamma$$

dla dowolnego $X \in V$ i dla dowolnego $\gamma \in T^*$. Tak jak poprzednio zauważmy, że wystarczy dowieść tę implikację i to w przypadku, gdy X jest równe A.

Najpierw, korzystając z Lematu 8.7 i postępując jak w poprzedniej części dowodu, wyprowadzeniu słowa γ nadajemy postać

$$A \rightarrow_{G_1} \beta_j B \Rightarrow_{G_1} \beta_j B\alpha_i \Rightarrow_{G_1}^{k_1} \gamma_1 B\alpha_i \Rightarrow_{G_1}^{k_2} \gamma_1\gamma_2\alpha_i \Rightarrow_{G_1}^{k_3} \gamma_1\gamma_2\gamma_3 = \gamma.$$

Są możliwe także wyprowadzenia, w których do symbolu B stosujemy wyłącznie produkcję $B \rightarrow_{G_1} \varepsilon$. Dla takich wyprowadzeń dalszy dowód jest prosty i zostaje pominięty. Zauważmy, że

$$A \rightarrow_{G_1} \beta_j B \Rightarrow_{G_1}^{k_1} \gamma_1 B \Rightarrow_{G_1}^{k_2} \gamma_1\gamma_2.$$

Oznacza to, że w gramatyce G_1 słowo $\gamma_1\gamma_2$ daje się wyprowadzić z symbolu A i wymaga to mniejszej liczby kroków niż wyprowadzenie słowa γ. Z założenie indukcyjnego mamy, że $A \Rightarrow_G^* \gamma_1\gamma_2$, a także $\alpha_i \Rightarrow_G^* \gamma_3$. Teraz już łatwo wyprowadzić γ w gramatyce G:

$$A \rightarrow_G A\alpha_i \Rightarrow_G^* \gamma_1\gamma_2\alpha_i \Rightarrow_G^* \gamma_1\gamma_2\gamma_3 = \gamma.$$

Dowód, że $L(G_1) = L(G_2)$**.** Aby dowieść tę równość, pokażemy przez indukcję ze względu na k, że

$$X \Rightarrow_{G_1}^{k} \gamma \quad \Longleftrightarrow \quad X \Rightarrow_{G_2}^{k} \gamma$$

dla dowolnego $X \in V \cup \{B\}$ i dla dowolnego $\gamma \in T^*$. Nietrudno zauważyć, że różnice między gramatykami G_1 i G_2 są związane z symbolem B. Pozostałe symbole w obu gramatykach przekształcamy w taki sam sposób. Jak w poprzednich dowodach, otrzymujemy stąd dowodzoną równoważność w przypadku $X \neq B$.

Przypuśćmy, że $B \Rightarrow_{G_1}^{k} \gamma$. Ponieważ $B \neq \gamma$, więc $k > 0$. Nie ma czego dowodzić, jeżeli pierwszą wykorzystaną w wyprowadzeniu produkcją jest $B \rightarrow_{G_1} \varepsilon$. Możemy więc założyć, że wyprowadzenie γ ma postać

$$B \rightarrow_{G_1} B\alpha_i \Rightarrow_{G_1}^{k-1} \gamma.$$

Słowo γ daje się rozbić na dwa podsłowa γ_1 i γ_2, $\gamma = \gamma_1\gamma_2$, spełniające

$$B \Rightarrow_{G_1}^{k_1} \gamma_1 \text{ oraz } \alpha_i \Rightarrow_{G_1}^{k_2} \gamma_2$$

dla liczb k_1 i k_2 o sumie równej $k-1$. Ponieważ liczby k_1 i k_2 są mniejsze od k, więc z założenia indukcyjnego mamy

$$B \Rightarrow_{G_2}^{k_1} \gamma_1 \text{ oraz } \alpha_i \Rightarrow_{G_2}^{k_2} \gamma_2.$$

Dalej interesuje nas pierwsza produkcja w wyprowadzeniu γ_1. Jeżeli tą produkcją jest $B \rightarrow_{G_2} \varepsilon$, to $\gamma_1 = \varepsilon$, $k_1 = 1$ i w gramatyce G_2 słowo γ można wyprowadzić w następujący sposób:

$$B \rightarrow_{G_2} \alpha_i B \Rightarrow_{G_2} \alpha_i \Rightarrow_{G_2}^{k_2} \gamma_2 = \gamma.$$

Jeżeli natomiast jest to produkcja $B \rightarrow_{G_2} \alpha_j B$, to słowo γ_1 jest konkatenacją słów γ_{11} i γ_{12} takich, że

$$\alpha_j \Rightarrow_{G_2}^{k_{11}} \gamma_{11} \text{ oraz } B \Rightarrow_{G_2}^{k_{12}} \gamma_{12}$$

dla liczb k_{11} i k_{12} spełniających $k_{11} + k_{12} = k_1 - 1$. Z założenia indukcyjnego otrzymujemy, że $B \Rightarrow_{G_1}^{k_{12}} \gamma_{12}$ i, wobec tego,

$$B \rightarrow_{G_1} B\alpha_i \Rightarrow_{G_1}^{k_{12}} \gamma_{12} \Rightarrow_{G_1}^{k_2} \gamma_{12}\gamma_2.$$

Korzystając po raz kolejny z założenia indukcyjnego mamy

$$B \Rightarrow_{G_2}^{1+k_{12}+k_2} \gamma_{12}\gamma_2$$

i w konsekwencji

$$B \rightarrow_{G_2} \alpha_j B \Rightarrow_{G_2}^{k_{11}} \gamma_{11} B \Rightarrow_{G_2}^{1+k_{12}+k_2} \gamma_{11}\gamma_{12}\gamma_2 = \gamma.$$

Oznacza to, że

$$B \Rightarrow_{G_2}^{k} \gamma$$

i kończy dowód pierwszej implikacji. Implikację odwrotną dowodzimy w ten sam sposób.

Dowód, że $L(G_2) = L(G_3)$. Fakt ten wynika ze spostrzeżenia, że Konstrukcja 9.6 zastosowana do gramatyki G_2 daje gramatykę G_3.

Udowodniliśmy więc, że gramatyki G i G_3 generują ten sam język.

11.2 Gramatyki w postaci normalnej Greibach

Bezkontekstowa gramatyka $G = \langle V, T, P, S\rangle$ na **postać normalną Greibach**, jeśli dla dowolnego $X \in V$ i $\alpha \in (V \cup T)^*$ spełnia warunek

jezeli $X \rightarrow_G \alpha$, to $\alpha \in TV^*$.

Twierdzenie 11.2 *Jeżeli L jest niepustym językiem bezkontekstowym takim, że $\varepsilon \notin L$ to L jest generowany przez bezkontekstową gramatykę w postaci normalnej Greibach.*

Dowód. Niech L będzie niepustym językiem bezkontekstowym bez słowa pustego. Na mocy Twierdzenia 10.7, język L jest generowany przez gramatykę w postaci normalnej Chomsky'ego. Niech $G = \langle V, T, P, S\rangle$ będzie taką gramatykę generującą L.

Najpierw ustalmy dowolną numerację zmiennych ze zbioru V. Niech więc

$$V \;=\; \{A_1,\ A_2,\ \ldots,\ A_n\}.$$

W dalszym ciągu będziemy rozważać wyłącznie gramatyki postaci

$$G' \;=\; \langle V \cup \{B_1,\ B_2,\ \ldots,\ B_n\}, T, P', S\rangle,$$

gdzie $B_1,\ B_2,\ \ldots,\ B_n$ są nowymi zmiennymi, a więc nie należącymi do V. Oczywiście gramatyka G ma taką postać.

Konstrukcja gramatyki w postaci normalnej Greibach, generującej język L wymaga trzech kroków. W dwóch pierwszych krokach będziemy rozważać tylko gramatyki o trzech rodzajach produkcji

$$\begin{aligned} A_i &\rightarrow A_j A_k \gamma, \\ A_i &\rightarrow a\gamma, \\ B_i &\rightarrow A_j \gamma, \end{aligned} \tag{9}$$

gdzie $\gamma \in (V \cup \{B_1,\ B_2,\ \ldots,\ B_n\})^*$.

Aby zdefiniować gramatykę w postaci normalnej Greibach będziemy przekształcać gramatykę G wielokrotnie wykonywać Konstrukcje 10.2 i 11.1 Z udowodnionych własności tych konstrukcji wynika, że wszystkie konstruowane w

tym dowodzie gramatyki będą generować język L. Jest oczywiste, że gramatyka G ma produkcje postaci (9).

Aby zastosować Konstrukcję 10.2 wskazujemy gramatykę i produkcję $A \to \alpha_1 B \alpha_2$ ze wyróżnionym wystąpieniem zmiennej B. Zastosowanie Konstrukcji 11.1 wymaga wskazania gramatyki i jednej z jej zmiennych. Umawiamy się, że podczas wykonywania dwóch pierwszych kroków definiowania gramatyki w postaci Greibach będziemy stosować Konstrukcję 10.2 do produkcji postaci $A_i \to A_j \alpha$ i do wskazanego wystąpienia zmiennej A_j, natomiast Konstrukcję 11.1 – do zmiennych ze zbioru V. Nietrudno sprawdzić, że przy tych ograniczeniach, Konstrukcje 10.2 i 11.1 zachowują postać produkcji określoną w (9).

Krok 1. W pierwszym kroku przekształcimy gramatykę G tak, aby produkcje otrzymanej gramatyki, mające postać $A_i \to A_j A_k \gamma$ spełniały warunek $i < j$. Aby to uzyskać przekształcamy gramatykę G zgodnie z następującym algorytmem:

- **dla** i **od 1 do** n **wykonaj**
  - **dla** j **od 1 do** $i-1$

    {jeżeli $u < i$ i jest produkcja $A_u \to A_v A_k \gamma$, to $u < v$}

    {jeżeli jest produkcja $A_i \to A_v A_k \gamma$, to $v \geq j$}
    - **dopóki w gramatyce** G **są produkcje postaci** $A_i \to A_j A_k \gamma$
      - **zastąp** G **gramatyką otrzymaną w wyniku zastosowania Konstrukcji 10.2 do** G **i do produkcji** $A_i \to A_j A_k \gamma$**,**

      {jeżeli $u < i$ i jest produkcja $A_u \to A_v A_k \gamma$, to $u < v$}

      {jeżeli jest produkcja $A_i \to A_v A_k \gamma$, to $v \geq j+1$}

    {jeżeli $u < i$ i jest produkcja $A_u \to A_v A_k \gamma$, to $u < v$}

    {jeżeli jest produkcja $A_i \to A_v A_k \gamma$, to $v \geq i$}
  - **zastąp** G **gramatykę otrzymaną przez zastosowanie Konstrukcji 11.1 do** G **i do zmiennej** A_i

    {jeżeli $u < i+1$ i jest produkcja $A_u \to A_v A_k \gamma$, to $u < v$}

{jeżeli jest produkcja $A_u \to A_v A_k \gamma$, to $u < v$}

{jeżeli jest produkcja $A_n \to a\gamma$, to $a \in T$}

Krok 2. W drugim kroku usuwamy z konstruowanej gramatyki wszystkie produkcje postaci $A_i \to A_j A_k \gamma$. W tym celu przekształcamy gramatykę w następujący sposób:

{jeżeli jest produkcja $A_u \to A_v A_k \gamma$, to $u < v$}

{jeżeli jest produkcja $A_n \to a\gamma$, to $a \in T$}

- **dla** i **od** $n-1$ **do 1**

{jeżeli $u > i$ oraz jest produkcja $A_u \to a\gamma$, to $a \in T$}

- **dopóki w gramatyce G są produkcje postaci $A_i \to A_j A_k \gamma$**

  {jeżeli jest produkcja $A_u \to A_v A_k \gamma$, to $u < v$}

  - **zastąp G gramatykę otrzymaną przez zastosowanie Konstrukcji 11.1 do G i do zmiennej A_j w produkcji $A_i \to A_j A_k \gamma$.**

    {jeżeli jest produkcja $A_u \to A_v A_k \gamma$, to $u < v$}

  {jeżeli $u > i - 1$ oraz jest produkcja $A_u \to a\gamma$, to $a \in T$}

{jeżeli jest produkcja $A_u \to a\gamma$, to $a \in T$}

Krok 3. W trzecim kroku usuwamy z konstruowanej gramatyki wszystkie produkcje postaci $B_i \to A_j \gamma$. W tym celu przekształcamy gramatykę w następujący sposób:

{jeżeli jest produkcja $A_u \to a\gamma$, to $a \in T$}

- **dopóki w gramatyce G są produkcje postaci $B_i \to A_j \gamma$**

  - **zastąp G gramatykę otrzymaną przez zastosowanie Konstrukcji 10.2 do G i do zmiennej A_j w produkcji $B_i \to A_j \gamma$.**

{jeżeli jest produkcja $A_u \to a\gamma$, to $a \in T$}

{jeżeli jest produkcja $B_u \to a\gamma$, to $a \in T$}

Jak wynika z komentarzy zamieszczonych w opisie algorytmów, po wykonaniu opisanych przekształceń otrzymujemy gramatykę w postaci normalnej Greibach. □

Wykład 12.

12.1 Automaty skończone ze stosem

Automatem skończonym ze stosem nazywamy siódemkę

$$M \quad = \quad \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle,$$

gdzie

1. Q jest skończonym zbiorem, nazywamy go **zbiorem stanów** automatu M,

2. Σ jest skończonym zbiorem, nazywamy go **alfabetem wejściowym** automatu M,

3. Γ jest skończonym zbiorem, nazywamy go **alfabetem stosowym** automatu M,
4. δ jest funkcją spełniającą

$$\delta : \ Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \longrightarrow 2^{Q \times \Gamma^*},$$

   nazywamy ją **funkcją przejścia** automatu M,
5. $q_0 \in Q$, nazywamy go **stanem początkowym** automatu M,
6. $Z_0 \in \Gamma$, nazywamy go **symbolem początkowym** automatu M,
7. $F \subseteq Q$, nazywamy go **zbiorem stanów akceptujących** automatu M.

Automat skończony ze stosem wykonuje dwa rodzaję ruchów. Ruchy pierwszego rodzaju są opisane przez wartości $\delta(q, a, \gamma)$ funkcji przejścia dla argumentów z a ze zbioru Σ. Ruchy drugiego rodzaju nazywamy ε-ruchami. Są opisane przez wartości $\delta(q, \varepsilon, \gamma)$ funkcji przejścia dla argumentu z ε. Automat wykonuje określony ruch wtedy, gdy wartość funkcji δ jest niepusta. Dokonuje wyboru rodzaju ruchu w sposób niedeterministyczny. Także niedeterministycznie jest wybierany ruch spośród możliwych do wykonania ruchów danego rodzaju.

Automat M znajduje się w pewnym stanie i zwykle jego głowica obserwuje jedną z liter danego słowa, z alfabetu Σ. Może też nie obserwować żadnej litery. Tak jest, gdy dane słowo jest puste lub automat przeczytał już wszystkie litery danego słowa. Działanie automatu zależy także od zawartości stosu, którą interpretujemy jako pewne słowo γ nad alfabetem stosowym Γ. Umawiamy się, że stos rośnie od prawej strony do lewej, więc pierwsza litera słowa γ znajduje się w wierzchołku stosu, a złożenie kolejnego symbolu na stosie polega na dopisaniu go na początku słowa γ. Poszczególne ruchy zależą od stanu automatu, obserwowanej litery i zawartości wierzchołka stosu. Jeżeli stos jest pusty, to automat nie może wykonać żadnego ruchu.

Jeżeli automat M jest w stanie $q \in Q$, obserwuje literę $a \in \Sigma$, wierzchołek stosu zawiera literę $Z \in \Gamma$ i $(q', \gamma) \in \delta(q, a, Z)$, to może wykonać ruch (pierwszego rodzaju) polegający na

1. przejściu do stanu q',
2. przesunięciu głowicy w prawo,
3. usunięciu z wierzchołka stosu symbolu Z i
4. dopisaniu do stosu słowa γ.

Jeżeli automat M jest w stanie $q \in Q$, wierzchołek stosu zawiera literę $Z \in \Gamma$ i $(q', \gamma) \in \delta(q, \varepsilon, Z)$ (w tym przypadku automat może obserwować dowolną literę lub nie obserwować żadnej), to ma prawo wykonać **ε-ruch** polegający na

1. przejściu do stanu q',
2. usunięciu z wierzchołka stosu symbolu Z i
3. dopisaniu do stosu słowa γ

nie powodujący ruchu głowicy.

Konfiguracją automatu M (czasem nazywaną opisem chwilowym automatu) nazywamy trójkę (q, s, γ) taką, że $q \in Q$, $s \in \Sigma^*$ i $\gamma \in \Gamma^*$. Mówimy, że automat M **znajduje się w konfiguracji** (q, s, γ), jeżeli jest w stanie q, zostało mu do przeczytania słowo s (więc obserwuje pierwszą literę słowa s, jeżeli $s \neq \varepsilon$) i stos zawiera słowo γ.

Relacją przejścia automatu M nazywamy binarną relację w zbiorze konfiguracji, oznaczaną symbolem $\vdash$ (lub $\vdash_M$), zdefiniowaną wzorem

$$\begin{aligned} \vdash \quad = \quad & \{((q, as, Z\gamma), (q', s, \gamma'\gamma)) : \ (q', \gamma') \in \delta(q, a, \gamma)\} \\ & \cup \{((q, s, Z\gamma), (q', s, \gamma'\gamma)) : \ (q', \gamma') \in \delta(q, \varepsilon, \gamma)\} \end{aligned}$$

Intuicyjnie, dwie konfiguracje (q, s, γ) i (q', s', γ') są w relacji przejścia, a więc zachodzi $(q, s, \gamma) \vdash (q', s', \gamma')$, jeżeli automat znajdujący się w konfiguracji (q, s, γ), po wykonaniu jednego ruchu może się znaleźć w konfiguracji (q', s', γ'). Zwróćmy uwagę, że definicja relacji przejścia zawiera także pewne umowy dotyczące sposobu zapisywania informacji o zmianach na stosie i o samym stosie.

Relacją przejścia będziemy nazywać także binarną relację $\vdash^*$ (lub $\vdash^*_M$) w zbiorze konfiguracji automatu M, będącą przechodnim (i zwrotnym) domknięciem relacji $\vdash$.

12.2 Akceptowanie przez automaty skończone ze stosem

Będziemy rozważać dwa sposoby akceptowania słów przez automat ze stosem. Słowo $s \in \Sigma^*$ jest **akceptowane przez przejście do stanu akceptującego**, jeżeli $(q_0, s, Z_0) \vdash^* (q, \varepsilon, \gamma)$ dla pewnych $q \in F$ i $\gamma \in \Gamma^*$. Mówiąc intuicyjnie, automat ze stosem akceptuje słowo s przez przejście do stanu akceptującego, jeżeli uruchomiony w stanie początkowym q_0, ze stosem zawierającym tylko symbol początkowy Z_0 może tak pracować, aby przeczytać całe słowo s i znaleźć się w stanie ze zbioru F. Automat ze stosem **akceptuje** słowo $s \in \Sigma$ **przez opróżnienie stosu**, jeżeli $(q_0, s, Z_0) \vdash^* (q, \varepsilon, \varepsilon)$ dla pewnego $q \in Q$. Innymi słowy, automat ze stosem akceptuje słowo s przez opróżnienie stosu, jeżeli uruchomiony w stanie początkowym q_0, ze stosem zawierającym tylko symbol początkowy Z_0 może tak pracować, aby przeczytać całe słowo s powodując jednocześnie opróżnienie stosu.

W związku z tym, że mamy dwa pojęcia akceptowania słowa, będziemy rozważać dwa języki akceptowane przez automat ze stosem. Język $L(M)$ nazywamy językiem akceptowanym przez automat M przez przejście do stanów akceptujących. Zgodnie z definicją mamy

$$L(M) \quad = \quad \{s \in \Sigma^* : \ \exists q \in F \exists \gamma \in \Gamma^* \ (q_0, s, Z_0) \vdash^*_M (q, \varepsilon, \gamma)\}.$$

Język $N(M)$ nazywamy językiem akceptowanym przez automat M przez opróżnienie stosu. Język ten definiujemy jako

$$N(M) \quad = \quad \{s \in \Sigma^* : \exists q \in Q \ (q_0, s, Z_0) \vdash_M^* (q, \varepsilon, \varepsilon)\}.$$

Przykład 12.1 Rozważmy automat skończony ze stosem

$$M \quad = \quad \langle \{q_0, q_1, q_2\}, \{0, 1\}, \{Z, Z_0\}, \delta, q_0, Z_0, \{q_2\} \rangle,$$

z funkcja przejścia δ zdefiniowaną dla wymienionych argumentów wzorami

$$\begin{aligned} \delta(q_0, 0, Z_0) &= \{(q_0, ZZ_0)\}, \\ \delta(q_0, 0, Z) &= \{(q_0, ZZ)\}, \\ \delta(q_0, 1, Z) &= \{(q_1, \varepsilon)\}, \\ \delta(q_1, 1, Z) &= \{(q_1, \varepsilon)\}, \\ \delta(q_1, \varepsilon, Z_0) &= \{(q_2, \varepsilon)\}, \end{aligned}$$

a dla pozostałych argumentów przyjmującą jako wartość zbiór pusty.

Nietrudno zauważyć, że dla $n > 0$ (we wzorach zostało założone, że $n > 1$) mamy

$$\begin{gathered} (q_0, 0^n 1^n, Z_0) \vdash_M (q_0, 0^{n-1} 1^n, ZZ_0) \vdash_M (q_0, 0^{n-2} 1^n, ZZZ_0) \vdash_M^* \\ (q_0, 1^n, Z^n Z_0) \vdash_M (q_1, 1^{n-1}, Z^{n-1} Z_0) \vdash_M (q_1, 1^{n-2}, Z^{n-2} Z_0) \vdash_M^* \\ (q_1, \varepsilon, Z_0) \vdash_M (q_2, \varepsilon, \varepsilon). \end{gathered}$$

Oznacza to, że automat M akceptuje na oba sposoby wszystkie słowa postaci $0^n 1^n$ dla $n > 0$.

Automat M uruchomiony ze słowem pustym nie wykonuje żadnego ruchu, a więc słowo puste nie jest akceptowane ani przez przejście do stanu akceptującego, ani przez opróżnienie stosu. Przypuśćmy, że automat M zdołał przeczytać słowo $s \in \Sigma^+$. Słowo s musi zaczynać się literą 0. Słowa postaci 0^n, $n > 0$, zostają przez automat M przeczytane, ale wtedy kończy on pracę w stanie q_0 i ze stosem zawierającym $Z^n Z_0$, a więc nie są one akceptowane. Jeżeli uruchomimy automat M ze słowem postaci $0^n 1^{n+1} s$, to będzie pracować w następujący sposób:

$$\begin{gathered} (q_0, 0^n 1^{n+1} s, Z_0) \vdash_M^* (q_0, 1^{n+1} s, Z^n Z_0) \vdash_M (q_1, 1^n s, Z^{n-1} Z_0) \\ \vdash_M^* (q_1, 1s, Z_0) \vdash_M (q_2, 1s, \varepsilon). \end{gathered}$$

Słowa tej postaci nie zostaną więc przeczytane i, tym bardziej, nie zostaną zaakceptowane. Także słowa postaci $0^n 1^m 0s$, $0 < m \leq n$, nie zostaną przeczytane przez automat M. Czytając te słowa znajdzie się on w konfiguracji $(q_1, 0s, Z^{n-m} Z_0)$ i wtedy albo stanie, albo wykona ε-ruch (w przypadku $n = m$) przechodząc do konfiguracji $(q_2, 0s, \varepsilon)$. Mogą więc zostać jeszcze przeczytane słowa postaci $0^n 1^m$, $0 < m \leq n$. Jeżeli $0 < m < n$, to automat M zakończy czytanie słowa $0^n 1^m$ w konfiguracji $(q_1, \varepsilon, Z^{n-m} Z_0)$, a więc słowa te nie

zostaną zaakceptowane. Mogą więc zostać zaakceptowane i, jak wiemy, zostaną zaakceptowane słowa postaci 0^n1^n, $n > 0$.

Ostatecznie, mamy $L(M) = N(M) = \{0^n1^n : n > 0\}$.

Lemat 12.2 *Jeżeli M jest automatem skończonym ze stosem, to język $L(M)$ jest akceptowany przez pewien skończony automat ze stosem przez opróżnienie stosu.*

Dowód. Niech

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle.$$

Dodamy do automatu M dwa nowe stany, nowy symbol stosowy oraz odpowiednio zmienimy funkcję przejścia, symbol i stan początkowy. Niech więc

$$M' = \langle Q \cup \{q_0', q_e\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, F\rangle,$$

gdzie δ' jest funkcją spełniającą

$$\begin{aligned}
\delta'(q_0', \varepsilon, X_0) &= \{(q_0, Z_0X_0)\},\\
\delta'(q, \varepsilon, X_0) &= \{(q_e, \varepsilon)\}, \text{ jeżeli } q \in F,\\
\delta'(q, a, Z) &= \delta(q, a, Z) \cup \{(q_e, \varepsilon)\}, \text{ jeżeli } q \in F,\ a \in \Sigma \cup \{\varepsilon\},\ Z \in \Gamma,\\
\delta'(q, a, Z) &= \delta(q, a, Z), \text{ jeżeli } q \in Q \setminus F,\ a \in \Sigma \cup \{\varepsilon\},\ Z \in \Gamma,\\
\delta'(q_e, \varepsilon, Z) &= \{(q_e, \varepsilon)\}, \text{ jeżeli } q \in Q \setminus F,\ Z \in \Gamma \cup \{X_0\},
\end{aligned}$$

a we wszystkich pozostałych przypadkach przyjmującą jako wartość zbiór pusty. Przeglądając definicję funkcji δ' można stwierdzić, że w stanie początkowym automat M' może tylko dopisać do stosu symbol początkowy Z_0 automatu M i przejść do stanu q_0. Dalej automat M' działa jak automat M, ale ma też dodatkowe możliwości. W stanach końcowych (akceptujących), bez względu na symbol znajdujący się na stosie, może jeszcze wymazać ten symbol i przejść do stanu q_e. Po znalezieniu się w stanie q_e, może jeszcze usuwać symbole ze stosu, bez zmiany stanu i położenia głowicy.

Nietrudno zauważyć, że

$$(q, s, \gamma) \vdash_M^* (q', s', \gamma') \iff (q, s, \gamma X_0) \vdash_{M'}^* (q', s', \gamma' X_0) \tag{10}$$

dla dowolnych $q, q' \in Q$, $s, s' \in \Sigma^*$, $\gamma, \gamma' \in \Gamma^*$. Implikacja w prawą stronę jest oczywista i wynika stąd, że automat M' wykonuje wszystkie ruchy automatu M. Implikacja odwrotna jest konsekwencją tego, że po przejściu do stanu ze zbioru Q, automat M' albo wykonuje ruchy automatu M, albo przechodzi do stanu q_e i do końca pracy znajduje się w tym stanie. Ponadto, wykonując ruchy automatu M nie może usunąc z dna stosu symbolu X_0.

Stąd łatwo otrzymujemy, że $L(M) \subseteq N(M')$. Jeżeli $s \in L(M)$, to automat M przeprowadza konfigurację (q_0, s, Z_0) w konfigurację $(q, \varepsilon, \gamma')$ dla pewnego $q \in F$ i $\gamma' \in \Gamma^*$ i, wobec tego,

$$(q_0', s, X_0) \vdash_{M'} (q_0, s, Z_0X_0) \vdash_{M'}^* (q, \varepsilon, \gamma'X_0) \vdash_{M'} (q_e, \varepsilon, \gamma''X_0) \vdash_{M'}^* (q_e, \varepsilon, \varepsilon).$$

Tak więc automat M' akceptuje słowo s przez opróżnienie stosu.

Przypuśćmy teraz, że automat M' akceptuje słowo s przez opróżnienie stosu. Po rozpoczęciu pracy w konfiguracji (q_0', s, X_0) automat M' przechodzi do konfiguracji (q_0, s, Z_0X_0), a więc jest w kofniguracji ze stanem ze zbioru Q i na dnie stosu znajduje się symbol X_0. Jeżeli akceptuje słowo s przez opróżnienie stosu, to musi usunąc także symbol X_0 z dna stosu. Jest możliwe tylko w stanie q_e, albo w stanie $q \in F$ równocześnie z przejściem do stanu q_e. Wobec tego, automat M', aby zaakceptować, musi znaleźć się w stanie q_e. Przejście do stanu q_e jest możliwe tylko ze stanów $q \in F$. Automat M' znalazł się więc w stanie $q \in F$, przeszedł do stanu q_e i po pewnym czasie przeczytał dane słowo do końca i opróżnił stos. Ale przejście ze stanu q do q_e i dalsze ruchy w stanie q_e są ε-ruchami. Oznacza, to że przejście ze stanu q do q_e mogło mieć miejsce dopiero po przeczytaniu danego słowa. Tak więc, jeżeli automat M' akceptuje słowo s przez opróżnienie stosu, to

$$(q_0, s, Z_0X_0) \vdash_{M'}^* (q, \varepsilon, \gamma X_0)$$

dla pewnego $q \in F$ i $\gamma \in \Gamma^*$. Wyżej już zauważyliśmy, że wtedy

$$(q_0, s, Z_0) \vdash_M^* (q, \varepsilon, \gamma),$$

a to oznacza, że automat M akceptuje słowo s przez przejście do stanu akceptującego. □

Lemat 12.3 *Jeżeli M jest automatem skończonym ze stosem, to język $N(M)$ jest akceptowany przez pewien skończony automat ze stosem przez przejście do stanu akceptującego.*

Dowód. Dowód jest bardzo podobny do dowodu Lematu 12.2. Niech

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\rangle.$$

Jak poprzednio, dodamy do automatu M dwa nowe stany, nowy symbol stosowy i odpowiednio zmienimy funkcję przejścia, symbol i stan początkowy. Niech więc

$$M' = \langle Q \cup \{q_0', q_e\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, \{q_e\}\rangle,$$

gdzie δ' jest funkcją taką, że

$$\begin{aligned} \delta'(q_0', \varepsilon, X_0) &= \{(q_0, Z_0X_0)\}, \\ \delta'(q, \varepsilon, X_0) &= \{(q_e, \varepsilon)\}, \\ \delta'(q, a, Z) &= \delta(q, a, Z), \quad \text{jeżeli } q \in Q,\ a \in \Sigma \cup \{\varepsilon\},\ Z \in \Gamma, \end{aligned}$$

i przyjmującą we wszystkich pozostałych przypadkach jako wartość zbiór pusty. Automat M' uruchomiony w stanie początkowym składa na stosie symbol początkowy Z_0 automatu M i przechodzi do stanu początkowego q_0. Dalej pracuje jak automat M. Dodatkowo może usunąć ze stosu symbol X_0 przechodząc jednocześnie do stanu q_e. W stanie q_e automat M' nie wykonuje żadnego ruchu.

Dla tak zdefiniowanego automatu zachodzi także własność (10). Dowodzimy ją podobnie, jak w dowodzie Lematu 12.2.

Z własności (10) łatwo daje się wyprowadzić, że $N(M) = L(M')$. Można też dowieść, że $L(M') = N(M')$. □

Wykład 13.

13.1 Pewne własności automatów skończonych ze stosem

W dalszym ciągu będziemy korzystać z tego, że obliczenia przez automat skończony ze stosem mają następujące własności:

Lemat 13.1 *Dla dowolnego skończonego automatu ze stosem*

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$$

i dla dowolnych $q, q' \in Q$, $s, s', t \in \Sigma^*$, $\gamma, \gamma', \xi \in \Gamma^*$ *oraz* $n \in N$ *zachodzi*

1. *jeżeli* $(q, s, \gamma) \vdash_M^n (q', s', \gamma')$, *to* $(q, st, \gamma\xi) \vdash_M^n (q', s't, \gamma'\xi)$,
2. *jeżeli* $(q, st, \gamma) \vdash_M^n (q', s't, \gamma')$, *to* $(q, s, \gamma) \vdash_M^n (q', s', \gamma')$,
3. *jeżeli* $\gamma \neq \varepsilon$ *i* $(q, s, \gamma\xi) \vdash_M (q', s', \gamma'\xi)$, *to* $(q, s, \gamma) \vdash_M (q', s', \gamma')$.

Krótko mówiąc, obliczenia automatu skończonego ze stosem są prowadzone w tak samo niezależnie od jeszcze nie przeczytanych symboli i niezależnie od symboli znajdujących się na dnie stosu. Nieco inna sytuacja ma miejsce w trzecim przypadku. Usuwanie symboli z dna stosu nie może doprowadzić do opróżnienia stosu. Konieczne więc jest dodatkowe założenie i nieco ostrożniejsze sformułowanie.

13.2 Automaty skończone ze stosem akceptują języki bezkontekstowe

Twierdzenie 13.2 *Jeżeli G jest gramatyką w postaci normalnej Greibach, to język L(G) jest akceptowany przez opróżnienie stosu przez pewien automat skończony ze stosem.*

Dowód. Dla gramatyki

$$G = \langle V, T, P, S \rangle$$

w postaci normalnej Greibach definiujemy automat

$$M \quad = \quad \langle \{q_0\}, T, V, \delta, q_0, S, \emptyset \rangle,$$

z funkcją przejścia δ określoną wzorami

$$\begin{aligned} \delta(q_0, a, Z) &= \{(q_0, \gamma) : \ Z \to_G a\gamma\}, \\ \delta(q_0, \varepsilon, Z) &= \emptyset \end{aligned}$$

dla wszystkich $a \in T$ i $Z \in V$.

Zauważmy, że dla tak zdefiniowanego automatu mamy

$$(q_0, s, Z) \vdash_M^* (q_0, \varepsilon, \gamma) \quad \Longleftrightarrow \quad Z \Rightarrow_G^* s\gamma \tag{11}$$

dla dowolnych $s \in T^*$ i $\gamma \in V^*$.

Dowód implikacji od lewej strony do prawej prowadzimy przez indukcję ze względu na s. Przypuśćmy, że

$$(q_0, sa, Z) \vdash_M^* (q_0, a, X\gamma) \vdash (q_0, \varepsilon, \gamma'\gamma) \quad .$$

Nietrudno zauważyć, że sposób pracy automatu M wymagający przynajmniej jednego ruchu można przedstawić w takiej postaci. Jest to konsekwencja nie wykonywania ε-ruchów przez automat M. Z założenia indukcyjnego i Lematu 13.1.2 otrzymujemy, że $Z \Rightarrow_G^* sX\gamma$. Natomiast na podstawie definicji funkcji przejścia δ stwierdzamy, że wykonanie ostatniego ruchu jest możliwe tylko wtedy, gdy $X \to_G a\gamma'$. Tak więc $Z \Rightarrow_G^* sa\gamma'\gamma$ i to kończy dowód rozważanej implikacji.

Implikację odwrotną udowodnimy przez indukcję ze względu na długość wyprowadzenia. Załóżmy, że $Z \Rightarrow_G^* s\gamma$. Jeżeli w pierwszym kroku wyprowadzenia $s\gamma$ korzystaliśmy z produkcji $Z \to_G a$ dla pewnego $a \in T$, to $s = a$, $\gamma = \varepsilon$ i, zgodnie z definicją funkcji δ, $(q_0, \varepsilon) \in \delta(q_0, a, Z)$. Wobec tego $(q_0, s, Z) \vdash_M (q_0, \varepsilon, \gamma)$.

Przyjmijmy teraz, że w pierwszym kroku tego wyprowadzenia korzystaliśmy z produkcji $Z \to_G aZ_1 \dots Z_k$. Na mocy Lematu 8.7 oznacza to, że słowo $s\gamma$ jest konkatenacją $a\alpha_1 \dots \alpha_k$ słów takich, że $Z_i \Rightarrow_G^* \alpha_i$ dla $i = 1, \dots, k$. Porównując słowa $s\gamma$ i $a\alpha_1 \dots \alpha_k$ stwierdzamy, że najwyżej jedno ze słów α_i zawiera symbole terminalne i nieterminalne. Ustalmy, że jest to słowo α_{i_0} i jest ono konkatenacją $\beta_1 \in T^*$ i $\beta_2 \in V^*$. W dalszym dowodzie pominiemy prostsze przypadki zakładając, że β_1 i β_2 są niepuste. Oczywiście, zachodzi

$$\begin{aligned} s &= a\alpha_1 \dots \alpha_{i_0 - 1}\beta_1, \\ \gamma &= \beta_2\alpha_{i_0+1} \dots \alpha_k. \end{aligned} \tag{12}$$

Słowa α_i należą do T^* dla dowolnego $i < i_0$. Dla wyprowadzeń słów α_i, $i \leq i_0$, korzystamy z założenia indukcyjnego. Mamy więc, że

$$(q_0, \alpha_i, Z_i) \vdash_M^* (q_0, \varepsilon, \varepsilon)$$

dla $i < i_0$ oraz

$$(q_0, \beta_1, Z_{i_0}) \vdash_M^* (q_0, \varepsilon, \beta_2).$$

Stąd, korzystając z Lematu 13.1.1 otrzymujemy

$$\begin{aligned}
(q_0, a\alpha_1 \dots \alpha_{i_0-1}\beta_1, Z) &\vdash_M (q_0, \alpha_1 \dots \alpha_{i_0-1}\beta_1, Z_1 \dots Z_k) \\
(q_0, \alpha_1 \dots \alpha_{i_0-1}\beta_1, Z_1 \dots Z_k) &\vdash_M^* (q_0, \alpha_2 \dots \alpha_{i_0-1}\beta_1, Z_2 \dots Z_k) \\
&\vdots \\
(q_0, \alpha_{i_0-1}\beta_1, Z_{i_0-1} \dots Z_k) &\vdash_M^* (q_0, \beta_1, Z_{i_0} \dots Z_k) \\
(q_0, \beta_1, Z_{i_0} \dots Z_k) &\vdash_M^* (q_0, \varepsilon, \beta_2 Z_{i_0+1} \dots Z_k).
\end{aligned}$$

Korzystając z faktu, że w gramatyce w postaci normalnej Greibach wyprowadzenie słowa $\alpha_i \in V^*$ ze zmiennej Z_i jest możliwe tylko wtedy, gdy $\alpha_i = Z_i$, oraz z równości (12) otrzymujemy

$$(q_0, s, Z) \vdash_M^* (q_0, \varepsilon, \beta_2 Z_{i_0+1} \dots Z_k) = (q_0, \varepsilon, \beta_2 \alpha_{i_0+1} \dots \alpha_k) = (q_0, \varepsilon, \gamma)$$

i tym samym dowiedliśmy równoważność (11).

Postawiając w równoważności (11) S zamiast Z i ε zamiast γ otrzymujemy, że

$$(q_0, s, S) \vdash_M^* (q_0, \varepsilon, \varepsilon) \iff S \Rightarrow_G^* s,$$

a więc $N(M) = L(G)$. □

13.3 Bezkontekstowość języków akceptowanych przez automaty ze stosem

Twierdzenie 13.3 *Jeżeli M jest automatem skończonym ze stosem, to $N(M)$ jest językiem bezkontekstowym.*

Dowód. Niech

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$$

będzie automatem skończonym ze stosem. Zdefiniujemy gramatykę

$$G = \langle Q \times \Gamma \times Q \cup \{S\}, \Sigma, P, S \rangle.$$

Zmiennymi tej gramatyki, oprócz zmiennej S, są trójki, w których pierwsza i ostatnia współrzędna jest stanem ze zbioru Q, a w środku znajduje się symbol stosowy. Można się spodziewać, że w dalszym dowodzie będą pojawiać się ciągi takich zmiennych. Przyjmijmy, że

$$[q_1, Z_1 Z_2 \dots Z_k, q_{k+1}]$$

oznacza ciąg zmiennych postaci

$$(q_1, Z_1, q_2)(q_2, Z_2, q_3)\dots(q_k, Z_k, q_{k+1})$$

dla pewnych stanów $q_2, \dots, q_k$. Oznaczenie to nie jest w pełni precyzyjne, ale będziemy się nim posługiwać zachowując trochę więcej ostrożności.

Zbiór produkcji P gramatyki G składa się z następujących rodzajów produkcji:

$S \to (q_0, Z_0, q)$	dla dowolnego $q \in Q$,
$(q, Z, q') \to a[q_1, Z_1 Z_2 \dots Z_k, q']$	dla dowolnych q, q', $q_1 \in Q$ i wszystkich ciągów postaci $[q_1, Z_1 Z_2 \dots Z_k, q']$ pod warunkiem, że $(q_1, Z_1 Z_2 \dots Z_k) \in \delta(q, a, Z)$, dla $k > 0$ i $a \in \Sigma$,
$(q, Z, q') \to a$	dla dowolnych q, q' pod warunkiem, że $(q', \varepsilon) \in \delta(q, a, Z)$ i $a \in \Sigma$,
$(q, Z, q') \to [q_1, Z_1 Z_2 \dots Z_k, q']$	dla dowolnych q, q', $q_1 \in Q$ i wszystkich ciągów postaci $[q_1, Z_1 Z_2 \dots Z_k, q']$ o ile $(q_1, Z_1 Z_2 \dots Z_k) \in \delta(q, \varepsilon, Z)$,
$(q, Z, q') \to \varepsilon$	dla dowolnych q, q' o ile $(q', \varepsilon) \in \delta(q, \varepsilon, Z)$.

Zauważmy, że

$$(q, Z, q') \Rightarrow_G^* s \quad \Longrightarrow \quad (q, s, Z) \vdash_M^* (q', \varepsilon, \varepsilon). \tag{13}$$

dla $s \in T^*$. Podstawiając w implikacji (13) q_0 zamiast q i Z_0 zamiast Z otrzymujemy

$$(q_0, Z_0, q') \Rightarrow_G^* s \quad \Longrightarrow \quad (q_0, s, Z_0) \vdash_M^* (q', \varepsilon, \varepsilon).$$

W gramatyce G, słowa wyprowadzane z S są wyprowadzane także ze zmiennej postaci (q_0, Z_0, q'). Wobec tego mamy $L(G) \subseteq N(M)$.

Aby dowieść (13) zastosujemy indukcję ze względu na długość wyprowadzenia w gramatyce G. Wyprowadzając słowo $s \in T^*$ ze zmiennej (q, Z, q') musimy zastosować przynajmniej jedną produkcję. Jeżeli pierwszą zastosowaną jest produkcja trzeciego lub piątego rodzaju, to prawdziwość implikacji sprawdzamy bezpośrednio i nie jest to trudne. Przypuśćmy więc, że pierwszą zastosowaną jest produkcja $(q, Z, q') \to a[q_1, Z_1 Z_2 \dots Z_k, q']$, to na mocy Lematu 8.7,

$$s \quad = \quad a s_1 \dots s_k$$

dla pewnych słów s_i oraz stanów q_i takich, że

$$(q_i, Z_i, q_{i+1}) \Rightarrow_G^* s_i.$$

Wyprowadzenia słów s_i są krótsze od wyprowadzeń słowa s, więc możemy zastosować założenie indukcyjne. Stąd otrzymujemy, że

$$(q_i, s_i, Z_i) \vdash_M^* (q_{i+1}, \varepsilon, \varepsilon)$$

dla $i \leq k$ (q_{k+1} jest równe q') oraz, na mocy definicji P i Lematu 13.1.1,

$$\begin{aligned}
(q, as_1 \dots s_k, Z) &\vdash_M (q_1, s_1 \dots s_k, Z_1 \dots Z_k) \\
(q_1, s_1 \dots s_k, Z_1 \dots Z_k) &\vdash_M^* (q_2, s_2 \dots s_k, Z_2 \dots Z_k) \\
&\vdots \\
(q_{k-1}, s_{k-1}s_k, Z_{k-1}Z_k) &\vdash_M^* (q_k, s_k, Z_k) \\
(q_k, s_k, Z_k) &\vdash_M^* (q', \varepsilon, \varepsilon)
\end{aligned}$$

i tym samym implikacja (13) została w tym przypadku udowodniona. Dowodzimy ją tak samo, jeżeli pierwszą wykorzystaną jest produkcja czwartego rodzaju.

Pozostało do wykazania, że $N(M) \subseteq L(G)$. Wynika to z własności

$$(q, s, \gamma) \vdash_M^n (q', \varepsilon, \varepsilon) \quad \Longrightarrow \quad [q, \gamma, q'] \Rightarrow_G^* s \text{ dla pewnego ciągu zmiennych } [q, \gamma, q'] \tag{14}$$

prawdziwej dla dowolnego, niepustego γ. Wykażemy to przez indukcję ze względu na n rozważając dwa przypadki.

Najpierw zakładamy, że γ jest zmienną Z i analizujemy pierwszy ruch automatu M. Przyjmijmy, że pracuje on w następujący sposób

$$(q, as', Z) \vdash_M (q_1, s', \gamma') \vdash_M^{n-1} (q', \varepsilon, \varepsilon).$$

dla niepustego γ'. Wtedy, na mocy definicji P i założenia indukcyjnego, w gramatyce G można wyprowadzać w następujący sposób

$$(q, Z, q') \Rightarrow_G a[q_1, \gamma', q'] \Rightarrow_G^* as'.$$

Oczywiście, automat M może także wykonać inny pierwszy ruch Wtedy implikację (14) uzasadniamy podobnie.

Przypuśćmy więc, że γ ma długość większą niż 1. Niech $\gamma = Z\gamma'$. Będziemy teraz analizować długość słowa zapisanego na stosie podczas pracy automatu. Automat ze stosem może skrócić zawartość stosu, ale tylko o 1 w pojedyńczym ruchu. Wynika stąd, że jeżeli automat opróżnia stos po rozpoczęciu pracy ze stosem długości m, to jego stos musi mieć w pewnym momencie długość i dla dowolnego $i < m$. Niech (q_1, s_2, γ') będzie pierwszą konfiguracją, w której automat M, po uruchomieniu w konfiguracji $(q, s, Z\gamma')$, ma stos o długości $|Z\gamma'| - 1$. Jest oczywiste, że stosu będzie wtedy zawierać słowo γ'. Wobec tego, automat M pracuje w następujący sposób

$$(q, s, Z\gamma') \vdash_M^* (q_1, s_2, \gamma') \vdash_M^* (q', \varepsilon, \varepsilon).$$

Załóżmy, że podczas pierwszych ruchów czyta słowo s_1, czyli $s = s_1 s_2$. Wtedy też jest zawsze dłższy od słowa γ', więc Lemat 13.1.3 implikuje, że

$$(q, s_1, Z) \vdash_M^* (q_1, \varepsilon, \varepsilon).$$

Do obu fragmentów pracy automatu możemy zastosować założenie indukcyjne. W ten sposób otrzymujemy, że

$$(q, Z, q_1) \Rightarrow_G^* s_1 \quad \text{oraz} \quad [q_1, \gamma', q'] \Rightarrow_G^* s_2$$

dla pewnego ciągu $[q_1, \gamma', q']$. Tak więc, dla odpowiednio dobranego słowa $[q, Z\gamma', q']$ mamy

$$[q, Z\gamma', q'] = (q, Z, q_1)[q_1, \gamma'.q'] \Rightarrow_G^* s_1[q_1, \gamma'.q'] \Rightarrow_G^* s_1 s_2 = s.$$

Jest oczywiste, że własność (14) implikuje zawieranie $N(M) \subseteq L(G)$. Tym samym konczymy dowód twierdzenia 13.3.

Twierdzenia 13.2 i 13.3 pozwalają na wykazanie, że

Twierdzenie 13.4 *Język L jest bezkontekstowy wtedy i tylko wtedy, gdy jest akceptowany przez automat skończony ze stosem (przez przejście do stanów akceptujących lub przez opróżnienie stosu).* □

Wykład 14.

14.1 Lemat o nadymaniu dla języków bezkontekstowych

Przypuśćmy, że mamy gramatykę $G = \langle V, T, P, S\rangle$ w postaci normalnej Chomsky'ego. Niech $\alpha \in T^*$ będzie słowem spełniającym $S \Rightarrow_G^* \alpha$. Najpierw będziemy rozważać pewne binarne drzewo $\mathcal{D}_\alpha$ wyznaczone przez słowo α (lub przez jego wyprowadzenie).

Wierzchołki drzewa $\mathcal{D}_\alpha$ będą etykietowane parami (X, β) spełniającymi $X \Rightarrow_G^* \beta$ i $\beta \in T^*$. Korzeń drzewa $\mathcal{D}_\alpha$ będzie etykietowany parą (S, α). Dalsza definicja drzewa będzie indukcyjna. Przypuśćmy więc, że w drzewie $\mathcal{D}_\alpha$ jest wierzchołek etykietowany parą (X, β). Jeżeli $\beta \in T$, to taki wierzchołek będzie liściem. Jeżeli $\beta \notin T$, to wyprowadzenia β ze zmiennej X wymaga użycia przynajmniej jednej produkcji postaci $X \to_G X_1 X_2$. W takim przypadku, korzystając z Lematu 8.7, słowo β rozbijamy na dwie części β_1, β_2 (niepuste, a więc różne od β!) takie, że

$$\begin{aligned} \beta &= \beta_1\beta_2, \\ X_1 \Rightarrow_G^* \beta_1 \quad &\text{i} \quad X_2 \Rightarrow_G^* \beta_2 \end{aligned}$$

i do drzewa $\mathcal{D}_\alpha$ dodajemy dwóch synów wierzchołka (X, β): lewego, etykietowanego parą (X_1, β_1) i prawego, etykietowanego (X_2, β_2).

Otrzymane drzewo ma kilka własności. Po pierwsze, jego liście są uporządkowane. Dla każdych dwóch liści l_1 i l_2 jest wierzchołek taki, że l_1 jest potomkiem lewego syna tego wierzchołka, a l_2 – prawego syna. Liść l_1 będący potomkiem lewego syna tego wierzchołka uważamy za znajdujący się po lewej stronie liścia l_2 będącego potomkiem prawego syna (l_1 jest wcześniejszy od l_2).

Po drugie, jeżeli utworzymy konkatenację liter z par etykietujących liście, w kolejności zgodnej z porządkiem liści, od lewej do prawej strony, to otrzymamy słowo α. Przez indukcję łatwo dowieść analogiczną własność poszczególnych wierzchołków drzewa $\mathcal{D}_\alpha$: konkatenacja liter z etykiet liści podrzewa o korzeniu z etykietą (X, β) jest równa β. Jest to oczywiste w przypadku liści. Nie ulega też wątpliwości, że wierzchołek ma tę własność, jeżeli mają ją jego synowie. Tak więc drzewo $\mathcal{D}_\alpha$ ma tyle liści, ile liter ma słowo α.

Trzecia własność stwierdza, że jeżeli wierzchołek z etykietą (Z, γ) jest potomkiem wierzchołka z etykietą (X, β), to dla pewnych słów β_1 i β_2 mamy

$$X \Rightarrow_G^* \beta_1 Z \beta_2 \quad \text{oraz} \quad \beta = \beta_1 \gamma \beta_2.$$

Tę własność dowodzimy bez trudu przez indukcję ze względu na długość drogi między wierzchołkami (X, β) i (Z, γ).

Twierdzenie 14.1 (Lemat o nadymaniu) *Jeżeli L jest językiem bezkontekstowym, to dla pewnej stałej $n \in N$, każde ze słów $s \in L$, o długości przynajmniej równej n daje się podzielić na pięć części u, v, w, x i y (a więc $s = uvwxy$) tak, aby słowo vx było niepuste, słowo vwx miało długość mniejszą od n lub równą n oraz tak, aby dla dowolnego $i \in N$, słowo uv^iwx^iy należało do języka L.*

Dowód. Lemat o nadymaniu wystarczy dowieść dla niepustych języków bezkontekstowych nie zawierającego słowa pustego. Każdy taki język jest generowany przez gramatykę w postaci normalnej Chomsky'ego. Będziemy więc dowodzić lemat o nadymaniu (Twierdzenie 14.1) dla języka $L(G)$, gdzie $G = \langle V, T, P, S\rangle$ jest gramatyką w postaci normalnej Chomsky'ego.

Niech $n = 2^{|V|}$ ($|V|$ oznacza liczbę elementów V).

Weźmy słowo $s \in L(G)$ o długości większej od n. Opisane wyżej drzewo $\mathcal{D}_s$ ma więcej niż $2^{|V|-1}$ liści. Jedna z gałęzi tego drzewa musi składać się z przynajmniej $|V|$ krawędzi. Znajduje się więc na niej więcej niż $|V|$ wierzchołków. Z zasady Dirichleta otrzymujemy, że dwa wierzchołki tej gałęzi są etykietowane parami z tą samą zmienną. W rozważanym drzewie jest więc wierzchołek z etykietą (X, β), który ma potomka z etykietą (X, γ). Wybierzmy dwa takie wierzchołki żądając jeszcze, że poddrzewo z korzeniem z etykietą (X, β) ma możliwie małą wysokość.

Nietrudno zauważyć, że wysokość poddrzewa o korzeniu z etykietą (X, β) nie przekracza $|V|$. Bowiem w przeciwnym razie, w tym poddrzewie też można by znaleźć takie dwa wierzchołki. Poddrzewo to ma więc najwyżej $2^{|V|}$ liści. Tym samym długość słowa β jest mniejsza lub równa n.

Fakt, że wierzchołek z etykietą (X, β) jest potomkiem korzenia z etykietą (S, s) implikuje, że

$$S \Rightarrow_G^* uXy$$

dla pewnych słów u i y spełniających

$$s = u\beta y.$$

W ten sam sposób uzasadniamy, że

$$X \Rightarrow_G^* vXx$$

dla pewnych słów v i x spełniających

$$\beta = v\gamma x,$$

gdyż wierzchołek z etykietą (X, γ) jest potomkiem wierchołka z etykietą (X, β). Jeżeli przyjmiemy, że $w = \gamma$, to zdefiniujemy podział słowa s na pięć części, taki jak w tezie Twierdzenia 14.1.

Słowo vx jest niepuste, gdyż $\beta \neq \gamma$.

Długość słowa vwx jest mniejsza lub równa n, ponieważ $vwx = \beta$.

Bez trudu też pokażemy, że wszystkie słowa uv^iwx^iy należą do $L(G)$. Mamy bowiem

$$\begin{aligned} S \Rightarrow_G^* uXy \Rightarrow_G^* uvXxy \Rightarrow_G^* uv^2Xx^2y \Rightarrow_G^* \ldots \\ \Rightarrow_G^* uv^{i-1}Xx^{i-1}y \Rightarrow_G^* uv^iwx^iy. \end{aligned}$$

To kończy dowód Lematu o nadymaniu. □

Przykład 14.2 Lemat o nadymaniu pozwala wykazać o wielu językach, że nie są bezkontekstowe. Na przykład pozwala to dowieść o języku

$$L \quad = \quad \{0^i1^i2^i : \ i \in N\}.$$

Jeżeli język L jest bezkontekstowy, to spełnia tezę lematu o nadymaniu (Twierdzenia 14.1). Znajdujemy więc stałą n o własnościach wymienionych w tezie Twierdzenia 14.1.

Weźmy słowo $0^n1^n2^n$. Słowo to należy do L i daje się podzielić na 5 części,

$$0^n1^n2^n \quad = \quad uvwxy$$

zgodnie z Twierdzeniem 14.1.

Jeżeli $|u| \geq n$, to słowo uv^2wx^2y ma mniej zer niż jedynek lub dwójek.

Jeżeli $|u| < n$, to słowo $uvwx$ ma długość mniejszą niż $2 \cdot n$ i, wobec tego, słowo uv^2wx^2y ma mniej dwójek niż zer lub jedynek.

W obu przypadkach słowo uv^2wx^2y nie należy do L wbrew lematowi o nadymaniu. Sprzeczność ta dowodzi, że język L nie jest bezkontekstowy.