Automaty i Języki Formalne

Soma Dutta

Wydział Matematyki i Informatyki, UWM w Olsztynie
soma.dutta@matman.uwm.edu.pl

Wykład - 4: Automaty skończone i języki formalne

Semestr zimowy 2023

Podsumowanie

- Poznaliśmy automaty skończone zarówno typu deterministycznego, jak i niedeterministycznego, a także dowiedzieliśmy się, że generują one ten sam zbiór języków.
- Poznaliśmy odmianę niedeterministycznego automatu skończonego, znaną jako ϵ-NDFA, która generuje ten sam język, ponieważ można go również przekonwertować na DFA.
- Wprowadzono jeszcze dwie odmiany automatów skończonych, które zamiast wskazywać, że ciąg wejściowy został zaakceptowany (lub nie akceptowane), mogą wygenerować ciąg wyjściowy. Jedna z nich znana jest jako maszyna Moore'a, a druga jako maszyna Mealey'a.

Plan

- Widzieliśmy, .ze dla danego języka mogą istnieć różne sposoby zaprojektowania automatu skończonego akceptujkacego ten język. Więc, pojawia się pytanie, czy dla danego języka istnieje sposób na zaprojektowanie automatu z minimalną liczbą stanów. W tym kontekście poznamy technikę minimalizacji automatu (minimization of automaton).
- Po drugie, ponieważ za pomocą automatów możemy generować różne języki, jednym z ważnych aspektów jest poznanie powiązania między automatami a językami formalnymi. W związku z tym osobno poznamy pojęcie gramatyki formalnej (formal grammar) i języków generowanych na jej podstawie. Następnie nauczymy się klasyfikacji języków (classification of languages). Później omówimy związek pomiędzy różnymi rodzajami automatów i językami formalnymi.

Minimalizacja automatów skończonych

W celu zbudowania automatu z minimalną liczbą stanów równoważnemu danemu automatowi M musimy poznać pojęcie równoważności pomiędzy dwoma stanami. Ponieważ interesują nas tylko ciągi akceptowane przez M, co jest ważne czy stan jest końcowy czy nie.

- Definicja: Dla danego automatu $M = (Q, \Sigma, \delta, q_0, F)$ i $k \geq 0$, dwa stany $q_1, q_2 \in Q$ są uważane za k-równoważne, oznaczane jako $q_1 \equiv_k q_2$, jeśli dla wszystkich ciągów $x \in \Sigma^*$ o długości k lub mniej, oba $\delta(q_1, x)$ i $\delta(q_2, x)$ są stanami końcowymi lub oba są stanami niekońcowymi.
- Np. dowolne dwa stany końcowe są 0-równoważne i dowolne stany niekońcowe są również 0-równoważne.

(Prop 1) $\equiv_k$ jest relacją zwrotną, symetryczną i przechodnią. Oznacza to, że jest to relacja równoważności, a zatem może podzielić cały zbiór Q na rozłączne klasy równoważności.

(Prop 2) Jeśli q_1, q_2 są $k+1$-równoważne ($q_1 \equiv_{k+1} q_2$), to są k-równoważne ($q_1 \equiv_k q_2$).

(Prop 3) Niech Π_k oznacza zbiór wszystkich klas równoważności wygenerowanych z $\equiv_k$. Wtedy $\Pi_n = \Pi_{n+1}$ dla pewnego n.

Skonstrukcja automatu minimalnego

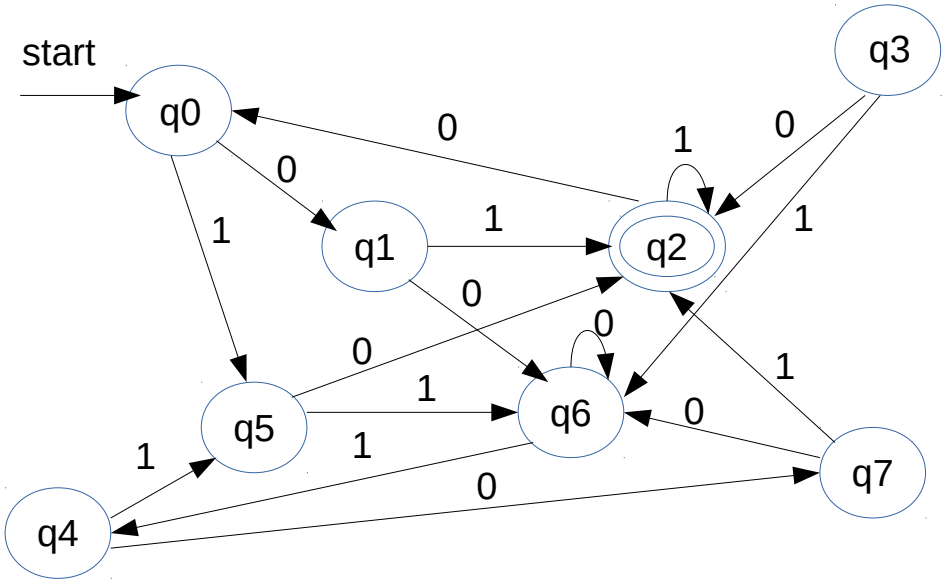
(Krok 1) Skonstrukcja Π_0: Z definicji 0-równoważności $\Pi_0 = \{Q_0, Q_1\}$ gdzie $Q_0 = F$ i $Q_1 = Q \setminus F$.

(Krok 2) Skonstrukcja Π_{k+1} z Π_k: Niech Q_i^k będzie dowolnym podzbiorem Π_k. Jeśli $q_1, q_2 \in Q_i^k$ (czyli $q_1 \equiv_k q_2$), są one $k+1$-równoważne (tzn., $q_1 \equiv_{k+1} q_2$) podanemu dla dowolnego $a \in \Sigma$, $\delta(q_1, a)$ i $\delta(q_2, a)$ należą do tej samej klasy równoważności w Π_k. W tym procesie Q_i^k jest dalej dzielony na klasy $k+1$-równoważności i proces jest kontynuowany aż do uzyskania wszystkich klas równoważności Π_{k+1}.

(Krok 3) Kontynuuj konstruowanie Π_n dla $n = 1, 2, 3, \ldots$ aż do $\Pi_n = \Pi_{n+1}$.

(Krok 4) Skonstrukcja automatu minimalnego: Za wymagane minimum automat stanowy stany są klasami równoważności uzyskanymi w kroku 3. Tablicę stanów uzyskuje się poprzez zastąpienie stanu q przez odpowiadająca klasa równoważności $[q]$. Liczba klas równoważności jest mniejsza lub równa $|Q|$. Więc, jeśli $[q_1] = \{q_1, q_2, \ldots q_k\}$, to jeśli jakiś ciąg w może dotrzeć do stanu końcowego przechodzącego przez q_1, to może również dotrzeć do stanu końcowego przechodzący przez dowolny stan $[q_1]$.

Przykład: MA1



- $M = (\{q_0, q_1, \ldots, q_7\}, \{0, 1\}, \delta, q_0, \{q_2\})$ i tabela przejściowa:

	0	1
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
$*q_2$	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

Kroki dla minimalnego automatu

- $\Pi_0 = \{\{q_2\}, \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}\}$. Nazwijmy je $Q_0^0 = \{q_2\}$ i $Q_1^0 = \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}$.
- Ponieważ Q_0^0 w Π_0 nie można dalej dzielone, pozostaje takie samo w Π_1 (tzn., $Q_0^1 = \{q_2\}$).

  Musimy podzielić $Q_1^0 = \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}$.
  - $\delta(q_0, 0), \delta(q_1, 0) \in Q_1^0$. Ale $\delta(q_0, 1) = q_5 \in Q_1^0$, a $\delta(q_1, 1) = q_2 \in Q_0^0$. Więc, q_0 i q_1 nie są 1-równoważne.
  - q_0, q_3 nie są 1-równoważne bo $\delta(q_0, 0) = q_1 \in Q_1^0$ i $\delta(q_3, 0) = q_2 \in Q_0^0$.
  - q_0, q_5 nie są 1-równoważne bo $\delta(q_0, 0) = q_1 \in Q_1^0$ i $\delta(q_5, 0) = q_2 \in Q_0^0$.
  - q_0, q_7 nie są 1-równoważne bo $\delta(q_0, 1) = q_5 \in Q_1^0$ i $\delta(q_7, 1) = q_2 \in Q_0^0$.
  - q_0, q_4, q_6 są 1-równoważne i tak więc będą należeć do jednej klasy równoważności, powiedzmy $Q_1^1 = \{q_0, q_4, q_6\}$.
  - Z drugiej strony, q_1, q_7 są 1-równoważne oraz q_3, q_5 są 1-równoważne, natomiast q_1, q_3 nie są 1-równoważne bo $\delta(q_1, 0) = q_6 \in Q_1^0$ i $\delta(q_3, 0) = q_2 \in Q_0^0$. Mamy więc dwie klasy równoważności, $\{q_3, q_5\} = Q_2^1$ (powiedzmy) i $\{q_1, q_7\} = Q_3^1$.
- Więc, $\Pi_1 = \{\{q_2\}, \{q_0, q_4, q_6\}, \{q_3, q_5\}, \{q_1, q_7\}\}$.
- Aby obliczyć Π_2, sprawdzimy stany, które są 2-równoważne. zaczniemy od $Q_1^1 = \{q_0, q_4, q_6\}$.
  - q_0, q_4 są 2-równoważne bo $\delta(q_0, 0), \delta(q_4, 0) \in Q_3^1$ i $\delta(q_0, 1), \delta(q_4, 1) \in Q_2^1$. Ale q_0, q_6 nie są 2-równoważne bo $\delta(q_0, 0) \in Q_3^1$ i $\delta(q_6, 0) \in Q_1^1$. Zatem Q_1^1 jest podzielony na dwie klasy równoważności, $\{q_0, q_4\}$ i $\{q_6\}$.
  - Innych klas równoważności z Π_1 nie można dalej dzielić, ponieważ zawierają stany 2-równoważne.
- Więc, $\Pi_2 = \{\{q_2\}, \{q_0, q_4\}, \{q_6\}, \{q_3, q_5\}, \{q_1, q_7\}\}$.
- Można sprawdzić, że Π_2 nie może być dalej dzielone, tzn., $\Pi_2 = \Pi_3$.

Diagram minimalnego automatu dla przykładu M1

- Minimalnego automatu M: $M' = (Q', \{0,1\}, \delta', q_0', F')$ gdzie

Diagram minimalnego automatu dla przykładu M1

- Minimalnego automatu M: $M' = (Q', \{0,1\}, \delta', q_0', F')$ gdzie

  $Q' = \Pi_2 = \{\{q_2\},\ \{q_0, q_4\},\ \{q_6\},\ \{q_3, q_5\},\ \{q_1, q_7\}\}$,

Diagram minimalnego automatu dla przykładu M1

- Minimalnego automatu M: $M' = (Q', \{0,1\}, \delta', q_0', F')$ gdzie

  $Q' = \Pi_2 = \{\{q_2\},\ \{q_0, q_4\},\ \{q_6\},\ \{q_3, q_5\},\ \{q_1, q_7\}\}$,

  $q_0' = [q_0] = \{q_0, q_4\}$ i $F = \{\{q_2\}\}$,

Diagram minimalnego automatu dla przykładu M1

- Minimalnego automatu M: $M' = (Q', \{0,1\}, \delta', q_0', F')$ gdzie

  $Q' = \Pi_2 = \{\{q_2\},\ \{q_0, q_4\},\ \{q_6\},\ \{q_3, q_5\},\ \{q_1, q_7\}\}$,

  $q_0' = [q_0] = \{q_0, q_4\}$ i $F = \{\{q_2\}\}$,

  $\delta'([q_i], a) = \delta(q_i, a)$ dla każdego $a \in \Sigma$
  (ponieważ wszystkie stany w klasie równoważności są identyfikowane i traktowane jako pojedynczy stan, jeśli istnieje łuk od q_i do q_j z etykietą a, to istnieje łuk z klasy równoważności $[q_i]$ do $[q_j]$ z tą samą etykietą w diagramu automatu minimalnego.)

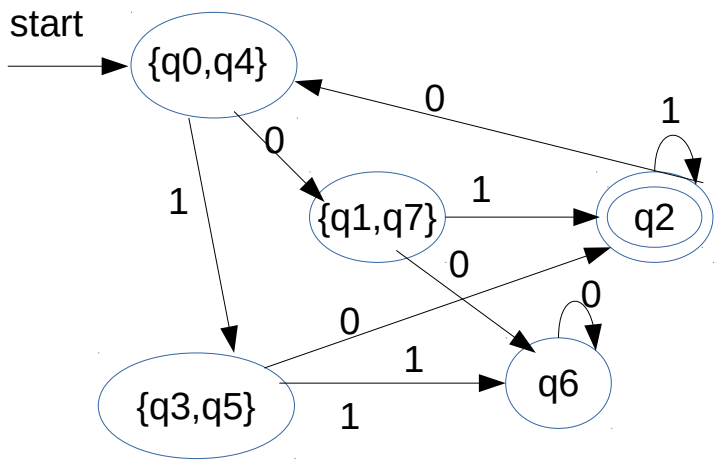
Diagram minimalnego automatu dla przykładu M1

- Minimalnego automatu M: $M' = (Q', \{0,1\}, \delta', q_0', F')$ gdzie

  $Q' = \Pi_2 = \{\{q_2\},\ \{q_0, q_4\},\ \{q_6\},\ \{q_3, q_5\},\ \{q_1, q_7\}\}$,

  $q_0' = [q_0] = \{q_0, q_4\}$ i $F = \{\{q_2\}\}$,

  $\delta'([q_i], a) = \delta(q_i, a)$ dla każdego $a \in \Sigma$
  (ponieważ wszystkie stany w klasie równoważności są identyfikowane i traktowane jako pojedynczy stan, jeśli istnieje łuk od q_i do q_j z etykietą a, to istnieje łuk z klasy równoważności $[q_i]$ do $[q_j]$ z tą samą etykietą w diagramu automatu minimalnego.)



Języki formalne

- Widzieliśmy, że mając dany język, możemy zaprojektować automat skończony generujący ten sam język. Zatem wyraźnie widzimy związek pomiędzy automatem skończonym (DFA lub NDFA) a językiem.
- Powstaje zatem pytanie, jaką rodzinę języków mogą wygenerować automaty skończone. Musimy więc najpierw określić, czym jest język formalny.
- Języki formalne pojawiają się wśród fundamentalnych pojęć logiki, matematyki i informatyki, a także teoretycznej lingwistyki. Na początku lat pięćdziesiątych (1950's) lingwiści próbowali precyzyjnie zdefiniować ważne zdania i podać strukturalne opisy zdań. Chcieli zdefiniować gramatykę formalną (tj. opisać reguły gramatyki w rygorystyczny sposób matematyczny) do opisu języka angielskiego. Myśleli, że taki opis języków naturalnych ułatwi tłumaczenie językowe za pomocą komputerów. W 1956 roku Noam Chomsky, amerykański lingwista, wprowadził pojęcie gramatyki formalnej, które okazało się przydatne w przypadku języków komputerowych. Stworzona przez niego hierarchia języków formalnych będzie jednych z naszych głównych tematów zainteresowania.

Przykład: gramatyka formalna dla języka angielskiego

- Dla uproszczenia rozważmy dwa rodzaje zdań w języku angielskim, aby sformalizować konstrukcję tych zdań.
- Rozważamy zdania zawierające rzeczownik i czasownik lub te zawierające rzeczownik-czasownik i przysłówek (np. 'John zjadł' lub 'Sam pobiegł szybko'). W zdaniu 'Sam pobiegł szybko' słowa 'Sam', 'pobiegł', 'szybko' zostały zapisane w tej kolejności.
- Jeśli zastąpimy 'John' przez 'Sam', 'Tom', 'Ala' itp., (tj. dowolnym rzeczownikiem), 'zjadł' przez 'biegał', 'chodził' itp., (tj. dowolnym czasownikiem w czasie przeszłym), a 'szybko' przez 'powoli', (tj. dowolnym przysłówek), otrzymujemy inne zdania poprawne gramatycznie.
- Mamy więc dwie proste formy zdań: <rzeczownik><czasownik> i <rzeczownik><czasownik><przysłówek>.
- Tutaj <rzeczownik>, <czasownik>, <przysłówek> są jak zmienne (variables) i jeśli zastąpimy <rzeczownik>, <czasownik> i <przysłówek> odpowiednimi słowami, otrzymamy rzeczywiste, poprawne gramatycznie zdania. Słowa takie jak 'John', 'Sam', 'zjadł', 'pobiegł', 'szybko', 'powoli', które tworzą zdania, można nazwać terminalami (terminals/constants).
- Niech S będzie zmienną oznaczającą zdanie.
- Możemy zatem utworzyć następujące reguły, aby wygenerować dwa typy zdań:

  (i) $S \rightarrow$ <rzeczownik><czasownik>
  (ii) $S \rightarrow$ <rzeczownik><czasownik><przysłówek>
  (iii) <rzeczownik> $\rightarrow$ John
  (iv) <rzeczownik> $\rightarrow$ Sam
  (v) <czasownik> $\rightarrow$ zjadł
  (vi) <czasownik> $\rightarrow$ pobiegł
  (vii) <przysłówek> $\rightarrow$ szybko

- Każde z powyższych wyrażeń jest zwykle nazywane produkcją (production) gramatyki formalnej, wskazującą, że każde wyrażenie po lewej stronie strzałki można zastąpić wyrażeniem po prawej stronie. Możemy zacząć od produkcji (i) i zastąpić <rzeczownik> używając (iii), a następnie <czasownik> używając (v). W ten sposób powstają zdania.
- Zatem gramatyka formalna zawiera cztery elementy: zbiór zmiennych (V_N), zbiór terminali/stałych (Σ), zbiór reguł produkcji (P) i zmienną początkową ($S \in V_N$).

Gramatyka formalna: przez Noama Chomsky'ego

- Definicja: Gramatyka formalna (formal grammar) to czterokrotka $G = (V_N, \Sigma, P, S)$ gdzie
  - (i) V_N jest skończonym, niepustym zbiorem zmiennych,
  - (ii) Σ jest skończonym, niepustym zbiorem terminali lub stałych,
  - (iii) $V_N \cap \Sigma = \phi$,
  - (iv) S jest specjalną zmienną (tj. $S \in V_N$) zwanym symbolem startu, oraz
  - (v) P jest skończonym zbiorem, którego elementy mają postać $\alpha \to \beta$, gdzie α, β są ciągami znaków na $V_N \cup \Sigma$ i α ma co najmniej jeden symbol z V_N. Elementy P nazywane są produkcjami, regułami produkcji lub regułami przepisywania.
- Definicja: Jeśli $\alpha \to \beta$ jest produkcją w gramatyce G i γ, ∂ są dowolnymi dwoma ciągami znaków w $V_N \cup \Sigma$, to mówimy, że $\gamma\alpha\partial$ bezpośrednio wyprowadza $\gamma\beta\partial$ z G (oznaczanego jako $\gamma\alpha\partial \Rightarrow_G \gamma\beta\partial$). Proces ten nazywany jest wyprowadzaniem jednoetapowym (one-step derivation).
- W szczególności, jeśli $\alpha \to \beta$ jest produkcją, to $\alpha \Rightarrow_G \beta$ jest wyprowadzeniem jednoetapowym.

Język generowany przez gramatykę

- Zauważmy, że $\Rightarrow$ można uważać za relację na $(V_N \cup \Sigma)^*$ (czyli $\alpha \Rightarrow_G \beta$ oznacza α, β są powiązane relacją $\Rightarrow_G$)
- Definicja: Jeśli α, β są ciągami znaków ponad $V_N \cup \Sigma$ (tzn., $\alpha, \beta \in (V_N \cup \Sigma)^*$), to mówimy, że α wyprowadza β, oznaczane jako $\alpha \Rightarrow_G^* \beta$, gdzie $\Rightarrow_G^*$ jest zwrotnym, przechodnim domknięciem $\Rightarrow_G$ (reflexive-transitive closure of $\Rightarrow_G$).
- Oznacza to, że ponieważ $\Rightarrow_G^*$ jest zwrotno-przechodnim domknięciem $\Rightarrow_G$, mamy następujące przypadki.
  - $\alpha \Rightarrow_G^* \alpha$ dla każdego ciągu $\alpha \in (V_N \cup \Sigma)^*$.
  - Jeśli $\alpha \Rightarrow_G^* \beta$ gdzie $\alpha, \beta \in (V_N \cup \Sigma)^*$ i $\alpha \neq \beta$, wtedy istnieją łańcuchy $\alpha_1, \alpha_2, \ldots, \alpha_n$ gdzie $n \geq 2$ takie, że $\alpha \Rightarrow_G \alpha_1 \Rightarrow_G \alpha_2 \Rightarrow_G \ldots \Rightarrow_G \alpha_n = \beta$.
    (Oznacza to, że wszystkie derywacje o krokach skończonych są zawarte w relacji $\Rightarrow_G^*$)
- Definicja: Jeśli $S \Rightarrow_G^* \alpha$, wtedy α nazywa się formą zdaniową.
- Definicja: Język generowany przez gramatykę G, oznaczony jako $L(G)$, jest zbiorem wszystkich ciągów terminali pochodzących od S. To znacza, $L(G) = \{w \in \Sigma^* : S \Rightarrow_G^* w\}$.

- Właściwość: Wszystkie ciągi znaków w $L(G)$ są formami zdaniowymi - ale nie odwrotnie.
- Definicja: Dwie gramatyki formalne G_1 i G_2 są równoważne jeśli $L(G_1) = L(G_2)$.
- Terminologie:
  - Jeśli $A \to \alpha$ jest produkcję w formalnym gramatyce G i $A \in V_N$, wtedy nazywamy to jako A-produkcja.
  - Jeśli $A \to \alpha_1$, $A \to \alpha_2$, $\ldots A \to \alpha_n$ są A-produkcjami, to zwykle oznaczamy je jako $A \to \alpha_1 | \alpha_2 | \ldots | \alpha_n$.

Przykład 1: język formalny

- $G = (\{S\}, \{0, 1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.

Przykład 1: język formalny

- $G = (\{S\}, \{0, 1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.
    Dla $n \geq 1$ mamy następujące wyprowadzenie.
    $S \Rightarrow_G 0S1 \Rightarrow_G 00S11 \Rightarrow_G \ldots \Rightarrow_G 0^nS1^n \Rightarrow_G 0^n1^n$.

Przykład 1: język formalny

- $G = (\{S\}, \{0, 1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.
    Dla $n \geq 1$ mamy następujące wyprowadzenie.
    $S \Rightarrow_G 0S1 \Rightarrow_G 00S11 \Rightarrow_G \ldots \Rightarrow_G 0^nS1^n \Rightarrow_G 0^n1^n$.
    Więc, $0^n1^n \in L(G)$ dla $n \geq 0$, tzn., $\{0^n1^n : n \geq 0\} \subseteq L(G)$.

Przykład 1: język formalny

- $G = (\{S\}, \{0,1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.
    Dla $n \geq 1$ mamy następujące wyprowadzenie.
    $S \Rightarrow_G 0S1 \Rightarrow_G 00S11 \Rightarrow_G \ldots \Rightarrow_G 0^nS1^n \Rightarrow_G 0^n1^n$.
    Więc, $0^n1^n \in L(G)$ dla $n \geq 0$, tzn., $\{0^n1^n : n \geq 0\} \subseteq L(G)$.
  - Musimy też pokazać $L(G) \subseteq \{0^n1^n : n \geq 0\}$.

Przykład 1: język formalny

- $G = (\{S\}, \{0, 1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.
    Dla $n \geq 1$ mamy następujące wyprowadzenie.
    $S \Rightarrow_G 0S1 \Rightarrow_G 00S11 \Rightarrow_G \ldots \Rightarrow_G 0^nS1^n \Rightarrow_G 0^n1^n$.
    Więc, $0^n1^n \in L(G)$ dla $n \geq 0$, tzn., $\{0^n1^n : n \geq 0\} \subseteq L(G)$.
  - Musimy też pokazać $L(G) \subseteq \{0^n1^n : n \geq 0\}$.
    Rozważmy dowolny ciąg $w \in L(G)$. Tzn., $S \Rightarrow_G^* w$.

Przykład 1: język formalny

- $G = (\{S\}, \{0,1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.
    Dla $n \geq 1$ mamy następujące wyprowadzenie.
    $S \Rightarrow_G 0S1 \Rightarrow_G 00S11 \Rightarrow_G \ldots \Rightarrow_G 0^nS1^n \Rightarrow_G 0^n1^n$.
    Więc, $0^n1^n \in L(G)$ dla $n \geq 0$, tzn., $\{0^n1^n : n \geq 0\} \subseteq L(G)$.
  - Musimy też pokazać $L(G) \subseteq \{0^n1^n : n \geq 0\}$.
    Rozważmy dowolny ciąg $w \in L(G)$. Tzn., $S \Rightarrow_G^* w$.
    Mamy dwie możliwości.

Przykład 1: język formalny

- $G = (\{S\}, \{0,1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.
    Dla $n \geq 1$ mamy następujące wyprowadzenie.
    $S \Rightarrow_G 0S1 \Rightarrow_G 00S11 \Rightarrow_G \ldots \Rightarrow_G 0^nS1^n \Rightarrow_G 0^n1^n$.
    Więc, $0^n1^n \in L(G)$ dla $n \geq 0$, tzn., $\{0^n1^n : n \geq 0\} \subseteq L(G)$.
  - Musimy też pokazać $L(G) \subseteq \{0^n1^n : n \geq 0\}$.
    Rozważmy dowolny ciąg $w \in L(G)$. Tzn., $S \Rightarrow_G^* w$.
    Mamy dwie możliwości.

(i) Jeśli najpierw używamy $S \to \Lambda$, otrzymamy $w = \Lambda$.

Przykład 1: język formalny

- $G = (\{S\}, \{0,1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.
    Dla $n \geq 1$ mamy następujące wyprowadzenie.
    $S \Rightarrow_G 0S1 \Rightarrow_G 00S11 \Rightarrow_G \ldots \Rightarrow_G 0^nS1^n \Rightarrow_G 0^n1^n$.
    Więc, $0^n1^n \in L(G)$ dla $n \geq 0$, tzn., $\{0^n1^n : n \geq 0\} \subseteq L(G)$.
  - Musimy też pokazać $L(G) \subseteq \{0^n1^n : n \geq 0\}$.
    Rozważmy dowolny ciąg $w \in L(G)$. Tzn., $S \Rightarrow_G^* w$.
    Mamy dwie możliwości.

  (i) Jeśli najpierw używamy $S \to \Lambda$, otrzymamy $w = \Lambda$.

  (ii) Jeśli najpierw używamy $S \to 0S1$, na dowolnym kolejnym etapie, aby uzyskać ciąg terminali, musimy zastosować $S \to \Lambda$.
  Zatem dla każdej $n \geq 1$ wyprowadzenie w ma postać $S \Rightarrow_G^* 0^nS1^n \Rightarrow_G 0^n1^n$.

Przykład 1: język formalny

- $G = (\{S\}, \{0,1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.
    Dla $n \geq 1$ mamy następujące wyprowadzenie.
    $S \Rightarrow_G 0S1 \Rightarrow_G 00S11 \Rightarrow_G \ldots \Rightarrow_G 0^nS1^n \Rightarrow_G 0^n1^n$.
    Więc, $0^n1^n \in L(G)$ dla $n \geq 0$, tzn., $\{0^n1^n : n \geq 0\} \subseteq L(G)$.
  - Musimy też pokazać $L(G) \subseteq \{0^n1^n : n \geq 0\}$.
    Rozważmy dowolny ciąg $w \in L(G)$. Tzn., $S \Rightarrow_G^* w$.
    Mamy dwie możliwości.

  (i) Jeśli najpierw używamy $S \to \Lambda$, otrzymamy $w = \Lambda$.

  (ii) Jeśli najpierw używamy $S \to 0S1$, na dowolnym kolejnym etapie, aby uzyskać ciąg terminali, musimy zastosować $S \to \Lambda$.
  Zatem dla każdej $n \geq 1$ wyprowadzenie w ma postać
  $S \Rightarrow_G^* 0^nS1^n \Rightarrow_G 0^n1^n$.
  Więc $L(G) \subseteq \{0^n1^n : n \geq 0\}$.

Przykład 1: język formalny

- $G = (\{S\}, \{0,1\}, \{S \to 0S1, S \to \Lambda\}, S)$. Znajdź $L(G)$.
- Rozwiązanie
  - Ponieważ $S \to \Lambda$ jest produkcją, $S \Rightarrow_G^* \Lambda$, a tak więc $\Lambda \in L(G)$.
    Dla $n \geq 1$ mamy następujące wyprowadzenie.
    $S \Rightarrow_G 0S1 \Rightarrow_G 00S11 \Rightarrow_G \ldots \Rightarrow_G 0^nS1^n \Rightarrow_G 0^n1^n$.
    Więc, $0^n1^n \in L(G)$ dla $n \geq 0$, tzn., $\{0^n1^n : n \geq 0\} \subseteq L(G)$.
  - Musimy też pokazać $L(G) \subseteq \{0^n1^n : n \geq 0\}$.
    Rozważmy dowolny ciąg $w \in L(G)$. Tzn., $S \Rightarrow_G^* w$.
    Mamy dwie możliwości.

  (i) Jeśli najpierw używamy $S \to \Lambda$, otrzymamy $w = \Lambda$.

  (ii) Jeśli najpierw używamy $S \to 0S1$, na dowolnym kolejnym etapie, aby uzyskać ciąg terminali, musimy zastosować $S \to \Lambda$.
  Zatem dla każdej $n \geq 1$ wyprowadzenie w ma postać
  $S \Rightarrow_G^* 0^nS1^n \Rightarrow_G 0^n1^n$.
  Więc $L(G) \subseteq \{0^n1^n : n \geq 0\}$.
  - W ten sposób udowodniliśmy $L(G) = \{0^n1^n : n \geq 0\}$. □