Teoretyczne podstawy informatyki

Andrzej M. Borzyszkowski

Instytut Informatyki
Uniwersytet Gdański

sem. letni 2025/26

inf.ug.edu.pl/~amb/TPI

Gramatyki bezkontekstowe

Przykłady języków bezkontekstowych i nie bezkontekstowych

- były już gramatyki wyrażeń arytmetycznych, regularnych, w podobny sposób można definiować wyrażenia logiczne
- język $\{a^n b^n \mid n \in \mathbb{N}\}$, gramatyka $S \to \varepsilon \mid aSb$
- język $\{a^n b^m \mid n \geq m \geq 0\}$, gramatyka $S \to \varepsilon \mid aS \mid aSb$
- język palindromów $\{w \in \{a,b\}^* \mid w = w^R\}$, gramatyka $S \to \varepsilon \mid a \mid b \mid aSa \mid bSb$ (bez a i b można wymusić postać ww^R)
- język palindromów z rozgranicznikiem $\{wcw^R \mid w \in \{a,b\}^*\}$, gramatyka $S \to c \mid aSa \mid bSb$
- język "powtórzeń" $\{ww \mid w \in \{a,b\}^*\}$, nie spełnia lematu o pompowaniu, niech $z = a^n b^n a^n b^n$, fragment vwx może leży w całości w jednej połówce słowa, po napompowaniu "połówki" będą różne; albo leży w środku w $b^n a^n$, wówczas też nie zachowa się równości
- było już: język $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ też nie spełnia lematu o pompowaniu

Operacje na językach bezkontekstowych 2

- było już, że klasa języków bezkontekstowych jest zamknięta na sumę, konkatenację i iterację
- klasa języków bezkontekstowych nie jest zamknięta na przekrój
  - dw.: świadczy o tym przykład $\{a^n b^n c^n \mid n \in \mathbb{N}\} = \{a^n b^n c^k \mid n,k \in \mathbb{N}\} \cap \{a^k b^n c^n \mid n,k \in \mathbb{N}\}$
- ani na dopełnienie
  - dw.: prawa deMorgana pozwalają wyrazić przekrój przez sumę i na odwrót:
    $$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$
  - $\overline{\{a^n b^n \mid n \in \mathbb{N}\}} = \{a^n b^m \mid n \neq m\} \cup (a+b)^* ba (a+b)^*$
  - dopełnienia powyższych języków L_i są bezkontekstowe, ich suma $\overline{L_1} \cup \overline{L_2}$ też, ale kolejne dopełnienie już nie jest
- słowo jest akceptowane jeśli istnieje wyprowadzenie w gramatyce, negacja nie daje żadnej konstruktywnej informacji o budowie słowa

Automaty ze stosem

Automaty ze stosem

- automat ze stosem jest to automat (być może niedeterministyczny, z cichymi przejściami) zaopatrzony w dodatkową pamięć w postaci stosu, na który kładzione są pewne elementy (PDA – *push-down automata*)
- w każdym ruchu automat odczytuje symbol z taśmy wejściowej i pobiera symbol z wierzchu stosu, następnie zmienia stan i kładzie na stos słowo (ciąg symboli)
- może być również możliwy ruch zależny tylko od wierzchołka stosu, bez czytania taśmy wejściowej
- automat zaczyna działanie w stanie początkowym z pustym stosem (jego pustość jest widoczna specjalnym symbolem Z)
- dwie definicje akceptowania:
  1. stany akceptujące, jak dla zwykłych automatów
  2. akceptowane są słowa w momencie gdy stos jest pusty

Automaty ze stosem, definicja

- $\mathcal{A} = \langle Q, A, \Gamma, \delta, q_0, F, Z\rangle$ gdzie:
  Q – skończony zbiór stanów
  $q_0 \in Q$ – stan początkowy
  $F \subseteq Q$ – zbiór stanów końcowych (akceptujących)
  A – alfabet wejściowy, którym etykietowane są przejścia
  Γ – alfabet symboli na stosie
  $Z \in \Gamma$ – symbol pustego stosu
  δ funkcja niedeterministyczna określająca przejście stanów pod wpływem odczytania symbolu w zależności od wierzchołka stosu:
  $\delta : Q \times (A \cup \{\varepsilon\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma^*)$
- funkcja przejścia odczytuje symbol wejściowy, ale może też wykonać ruch bez odczytania
  zawsze zdejmuje wierzchołek stosu
  w szczególności rozpoznaje, że stos jest pusty – leży na nim symbol Z
  nie ma możliwości ruchu, jeśli stos jest naprawdę pusty

Konfiguracje i obliczenia

- definiujemy podobnie jak dla automatów niedeterministycznych
- jeśli $q, p \in Q$, $a \in A$, $w \in A^*$, $X \in \Gamma$, $\alpha, \beta \in \Gamma^*$ oraz $\langle p, \alpha\rangle \in \delta(q, a, X)$ to

$$\langle q, aw, X\beta\rangle \to \langle p, w, \alpha\beta\rangle$$

  automat zmienił stan, została odczytana jedna litera ze słowa wejściowego, zdjęty jeden symbol ze stosu i zastąpiony słowem
- tw.: obliczenia są lokalne w tym sensie, że przyrostek czytanego słowa nie ma znaczenia na przebieg obliczeń
  $\langle p, u, \alpha\rangle \stackrel{*}{\Rightarrow} \langle q, v, \beta\rangle$ w.t.w. $\langle p, uw, \alpha\rangle \stackrel{*}{\Rightarrow} \langle q, vw, \beta\rangle$
  $\langle p, u, \alpha\rangle \stackrel{*}{\Rightarrow} \langle q, v, \beta\rangle$ implikuje $\langle p, u, \alpha\gamma\rangle \stackrel{*}{\Rightarrow} \langle q, v, \beta\gamma\rangle$
- uwaga: druga implikacja nie może być odwrócona, być może zmiana stosu $\alpha\gamma$ na $\beta\gamma$ wymagała zdjęcia α i pewnych symboli z γ i położenia tych symboli z powrotem

Automaty ze stosem, akceptowanie

- def.: słowo w jest akceptowane przez stan, jeśli istnieje obliczenie dla w prowadzące do stanu akceptującego
- def.: słowo w jest akceptowane przez stos, jeśli istnieje obliczenie dla w zakończone pustym stosem
- tw.: te definicje są równoważne, tzn. dla każdego automatu akceptującego przez stan istnieje równoważny automat akceptujący przez stos i na odwrót
- dw.: jeśli automat akceptuje przez pusty stos, tworzymy nowy, jedyny, stan akceptujący i ε-przejście do tego stanu gdy odczytany został symbol Z na stosie
  na odwrót, jeśli automat akceptuje przez stany akceptujące, dodajemy dodatkowy symbol dna stosu, przeprowadzamy wszystkie obliczenia jak dotychczas a po przejściu do stanu akceptującego opróżniamy stos do osiągnięcia nowego symbolu dna

Przykład $\{a^n b^n \mid n \in \mathbb{N}\}$

- słowo jest akceptowane przez pusty stos
- w stanie początkowym czytamy serię liter a, każda litera odkładana jest na stos, $\delta(q_0, a, X) = \langle q_0, aX\rangle$
  pierwsza odczytana litera b zmienia stan, dalej można odczytać jedynie litery b, każdy odczyt zdejmuje literę a ze stosu, $\delta(q_i, b, a) = \langle q_1, \varepsilon\rangle$
- trójki $\langle$ stan, słowo, wierzchołek stosu $\rangle$ zmieniają się następująco:
  - $\langle q_0, abb, Z\rangle \Rightarrow \langle q_0, bb, aZ\rangle \Rightarrow \langle q_1, b, Z\rangle$
    odczyt kolejnej litery jest niemożliwy, bo stos jest pusty
  - $\langle q_0, aab, Z\rangle \Rightarrow \langle q_0, ab, aZ\rangle \Rightarrow \langle q_0, b, aaZ\rangle \Rightarrow \langle q_1, \varepsilon, aZ\rangle$
    słowo zostało odczytane do końca ale stos pozostał niepusty
  - $\langle q_0, aabb, Z\rangle \Rightarrow \langle q_0, abb, aZ\rangle \Rightarrow \langle q_0, bb, aaZ\rangle \Rightarrow \langle q_1, b, aZ\rangle \Rightarrow \langle q_1, \varepsilon, Z\rangle \Rightarrow \langle q_1, \varepsilon, \varepsilon\rangle$
    słowo zostało odczytane do końca i stos jest pusty – zostało zaakceptowane

Przykład $\{wcw^R \mid w \in \{a, b\}^*\}$

- słowo jest akceptowane przez pusty stos
- w stanie początkowym czytamy serię liter a i b, każda litera odkładana jest na stos, $\delta(q_0, x, X) = \langle q_0, xX\rangle$
  litera c zmienia stan, $\delta(q_0, c, X) = \langle q_1, X\rangle$
  dalej sprawdzamy, czy przeczytana litera jest równa wierzchołkowi stosu, pozytywny odczyt zdejmuje literę ze stosu, $\delta(q_1, x, x) = \langle q_1, \varepsilon\rangle$
- brak litery c spowoduje, że nie dojdziemy do drugiego stanu, jeśli jest i dojdziemy, możemy nie móc wykonać kolejnego kroku (gdy pozostała część słowa nie jest równa w^R albo pojawi się kolejne c) albo może okazać się, że stos pozostał niepusty

Przykład $\{ww^R \mid w \in \{a, b\}^*\}$

- są to palindromy parzystej długości
- słowo jest akceptowane jeśli stos został opróżniony
- nie ma wskaźnika połowy słowa, decyzja przejścia do drugiego stanu wykonana jest niedeterministycznie
- wszystkie palindromy, również nieparzystej długości, $\{w \in \{a, b\}^* \mid w = w^R\}$, będą rozpoznane, jeśli dopuścimy, że odczytanie litery nie zmienia stosu ale zmienia stan

Gramatyki bezkontekstowe a automaty ze stosem

- tw.: języki bezkontekstowe są to dokładnie języki akceptowane przez automaty ze stosem (CFL=PDA)
- dw.: ($\Rightarrow$) budujemy automat, akceptujący przez pusty stos, na podstawie wywodów lewostronnych gramatyki
  wywód lewostronny oznacza $wA\alpha \Rightarrow w\beta\alpha$ dla pewnych $w \in T^*$, $A \in V$, $\alpha, \beta \in (T \cup V)^*$ gdzie $A \to \beta$ jest produkcją w gramatyce
- automat będzie miał jeden stan $\{q\}$, S na dnie stosu, funkcja przejścia $\delta(q, \varepsilon, A) = \{\langle q, \beta\rangle \mid A \to \beta\}$ dla $A \in V$ oraz $\delta(q, a, a) = \{\langle q, \varepsilon\rangle\}$, $a \in A$
  tzn. albo zdejmuje ze stosu symbol nieterminalny i zamienia go wynik produkcji w gramatyce, albo odczytuje kolejną literę i sprawdza, że zgadza się z tą na stosie, którą jednocześnie zdejmuje
- niech w ma wywód lewostronny w gramatyce czyli $S \to \gamma_1 \Rightarrow \ldots \Rightarrow \gamma_\ell = w$, pośrednie słowa są postaci $\gamma_i = u_i\alpha_i$, $u_i \in T^*$, i produkują przedrostki w: $w = u_i y_i$
  z konstrukcji wynika obliczenie $\langle q, w, S\rangle \stackrel{*}{\Rightarrow} \langle q, y_i, \alpha_i\rangle \stackrel{*}{\Rightarrow} \langle q, \varepsilon, \varepsilon\rangle$

Gramatyki a automaty ze stosem, c.d. dowodu

- w drugą stronę, załóżmy obliczenie $\langle q, w, A\rangle \stackrel{*}{\Rightarrow} \langle q, \varepsilon, \varepsilon\rangle$
  - jeśli ma długość 1, to musi być produkcja $A \to \varepsilon$ oraz $w = \varepsilon$
  - w innym przypadku pierwszym krokiem jest $\langle q, w, A\rangle \to \langle q, w, \beta\rangle$ dla produkcji $A \to \beta$ w gramatyce
  - kolejne symbole w β_i będą konsumowane i rozwijane w kolejnych krokach, z założenie indukcyjnego będą istnieć wywody lewostronne z odpowiadających symboli
  - wszystkie one złożą się na wywód $A \stackrel{*}{\Rightarrow} w$

czyli słowo akceptowane obliczeniem zaczynającym się od symbolu początkowego S ma wywód w tej gramatyce

Gramatyki a automaty ze stosem, przykład $\{a^n b^n \mid n \in \mathbb{N}\}$

- gramatyka $S \to \varepsilon \mid aSb$
- tym razem automat ma tylko jeden stan, kroki obliczenia wkładają na stos aSb w pierwszej fazie albo tylko zdejmują ze stosu w drugiej
- obliczenie akceptujące $aabb$ wygląda następująco
  - $\langle q, aabb, S\rangle \Rightarrow \langle q, aabb, aSb\rangle \Rightarrow \langle q, abb, Sb\rangle \Rightarrow$
    $\langle q, abb, aSbb\rangle \Rightarrow \langle q, bb, Sbb\rangle$ (niedeterministyczna decyzja produkcji $S \to \varepsilon$)
    $\Rightarrow \langle q, bb, bb\rangle \Rightarrow \langle q, b, b\rangle \Rightarrow \langle q, \varepsilon, \varepsilon\rangle$
    słowo zostało odczytane do końca i stos jest pusty – zostało zaakceptowane
- automat jest niedeterministyczny, mimo, że potrafiliśmy znaleźć inny deterministyczny

Automaty ze stosem a gramatyki bezkontekstowe

- tw.: języki bezkontekstowe są to dokładnie języki akceptowane przez automaty ze stosem
- dw.: ($\Leftarrow$) definiujemy gramatykę na podstawie automatu akceptującego przez pusty stos
  niech $\mathcal{A} = \langle Q, A, \Gamma, \delta, q_0, Z\rangle$ będzie takim automatem
  - symbolami nieterminalnymi będą wszystkie zestawy $[p, X, q]$ dla $p, q \in Q$, $X \in \Gamma$ (zdjęcie symbolu X przy przejście od p do q) i dodatkowo symbol początkowy S
  - produkcjami będą $S \to [q_0, Z, p]$ dla $p \in Q$ oraz $[q, X, r_k] \to a[r, Y_1, r_1] \ldots [r_{k-1}, Y_k, r_k]$ jeśli $\langle r, Y_1 \ldots Y_k\rangle \in \delta(q, a, X)$, $a \in A \cup \{\varepsilon\}$, $r, r_1, \ldots, r_k \in Q$
    w szczególności $[q, X, r] \to a$ jeśli $\langle r, \varepsilon\rangle \in \delta(q, a, X)$

czyli z dużym nadmiarem przygotowujemy opis, że produkcja spowodowała położenie na stosie słowa oraz zmianę stanów

Automaty ze stosem a gramatyki, c.d. dowodu

- sprawdzamy teraz (przez indukcję względem długości), że
  - obliczenie w automacie $\langle q, w, X\rangle \stackrel{*}{\Rightarrow} \langle p, \varepsilon, \varepsilon\rangle$ odpowiada wywodowi $[q, X, p] \stackrel{*}{\Rightarrow} w$ w gramatyce
  - i na odwrót, wywód w gramatyce odpowiada obliczeniu w automacie
- w szczególności obliczenia zaczynające się od stanu początkowego i pustego stosu odpowiadają wywodom zaczynającym się od symbolu początkowego
- gramatyka będzie miała dużo niepotrzebnych symboli nieterminalnych, które można usunąć standardową procedurą

Automaty ze stosem – konsekwencje dla bezpieczeństwa aplikacji

- rozpoznanie słowa z języka regularnego wymaga przygotowania automatu o ustalonej z góry wielkości
- rozpoznanie słowa z języka bezkontekstowego wymaga użycia stosu o nieograniczonej z góry wielkości
- np. w aplikacji Webowej użytkownik ma podać wyrażenie arytmetyczne, złośliwy użytkownik podaje nieskończony ciąg symboli $2 * ($
  - po każdej iteracji na stosie jest co najmniej jeden więcej symbol oczekujący wyrażenia arytmetycznego
  - w końcu dochodzi do przepełnienia stosu (*stack overflow*) i całkowitego zakończenia działania aplikacji
  - czyli udany atak DOS (*denial of service* – odmowa usługi)
- aplikacje powinny analizować wejście użytkownika i nie pozwalać na takie wyczerpanie zasobów

Operacje na językach bezkontekstowych 3

- suma, konkatenacja i iteracja języków bezkontekstowych również jest w tej klasie
- dowód poprzez konstrukcję automatu dla powyższych operacji: konstrukcje dla języków regularnych działają i tutaj, automat dodatkowo ma stos
- konstrukcja dla dopełnienia nie działa, dla języków regularnych automat był deterministyczny, tutaj na pewno nie jest
- przekrój języków bezkontekstowych nie musi być bezkontekstowy
  - próba dowodu analogicznego do języków regularnych zawodzi – musimy uruchomić oba automaty, a do dyspozycji jest jeden stos
- ale przekrój jest językiem bezkontekstowym jeśli jeden z języków jest regularny
  - dw.: rozpatrujemy iloczyn kartezjański automatów, jeden z nich ma dodatkowy stos, a drugi nie potrzebuje stosu