Relacja: klasy gramatyk - klasy automatow

Podstawowymi narzędziami abstrakcyjnymi do opisu języków formalnych są gramatyki i automaty. Gramatyka jest formalnie czwórką $G = (N, T, P, S)$, gdzie

- N - zbiór zmiennych syntaktycznych, inaczej zwanych symbolami nieterminalnymi (w skrócie nieterminalami)
- T - zbiór stałych, inaczej zwanych symbolami terminalnymi (w skócie terminalami)
- P - zbiór reguł postaci $\alpha \to \beta$, inaczej zwanych produkcjami, gdzie:
$$\alpha \in (N \cup T)^+, \beta \in (N \cup T)^*.$$
- $S \in N$ - symbol początkowy gramatyki

Język $L(G)$ *generowany* przez G to zbiór słow nad alfabetem T które można otrzymać startując z S i stosując produkcje (reguły) gramatyki.

- **typu(0)** kombinatoryczne (bez ograniczeń)
- **typu (1)** kontekstowe, produkcje postaci
  $\alpha \cdot A \cdot \beta \to \alpha \cdot \gamma \cdot \beta$, gdzie $|\gamma| \geq 1$, $A \in N$.
  Wyjątkiem mogą być produkcje $S \to \epsilon$, ale wtedy S nie może wystąpić po prawej stronie żadnej produkcji
- **typu(2)** bezkontekstowe, produkcje postaci
  $$A \to \beta \text{ gdzie } A \in N.$$
- **typu(3)** jednostronnie liniowe. W szczególności prawostronnie liniowe gdy produkcje postaci
  $$A \to wB \text{ lub } A \to \epsilon, \text{ gdzie } w \in T^*, \ A, B \in N.$$

  Lewostronnie liniowe definiujemy analogicznie.

Generowanie $L_1 = \{ a^k : k \text{ jest dodatnią potęgą } 2\}$

$$S \to AaB, \; A \to AA, \; A \to C,$$

$$Ca \to aaC, \; CB \to B, \; B \to \epsilon$$

Przykład generacji:

$$S \to AaB \to^* A^k aB \to^* C^k aB$$

$$\to C^{k-1} a^2 C^1 B \to^* C^{k-2} a^4 C^2 B$$

$$\to^* a^{2^k} C^k B \to^* a^{2^k}.$$

Generowanie $L_2 = \{ a^n b^n c^n : n \geq 1 \}$

$$S \to ABSc \mid Abc, \quad BA \to CA, \quad CA \to CB$$

$$CB \to AB, \quad Bb \to bb, \quad A \to a$$

Zauważmy, że $AB \to^* BA$, oraz

$$(AB)^k b \to^* A(BA)^{k-1} Bb \to^* A^k B^{k-1} b \to^* a^k b^k$$

Przykład generacji:

$$S \to *ABSc \to^* (AB)^{n-1} Abc^n =$$

$$A(BA)^{n-1} bc^n \to^* a(BA)^{n-1} bc^n \to^* a^n b^n c^n$$

Produkcje postaci $\alpha \to \beta$, gdzie $|\alpha| \neq \epsilon$ oraz $|\alpha| \leq |\beta|$.

Fakt

Każda gramatyka kontekstowa nie używająca słowa pustego jest równoważna pewnej gramatyce monotonicznej i odwrotnie.

Ten sam język $L_2 = \{ a^n b^n c^n : n \geq 1 \}$.

$$S \to aBSc \mid abc \quad Ba \to aB \quad Bb \to bb.$$

Zauważmy, że $B^{n-1}b \to^* b^n$.
Mamy wyprowadzenie:

$$S \to aBSc \to aBaBScc \to\to aBaBaBSccc$$

$$\to aBaBaBabcccc \to^* (aB)^{n-1}abc^n$$

$$a^n B^{n-1} bc^n \to^* a^n b^n c^n.$$

Ogólnie automaty to abstrakcyjne modele algorytmu decyzyjnego. Zaletą i jednocześnie wadą automatów jest uproszczony model obliczeń.

Nieformalny opis automatu z pamięcią typu $\mathcal{T}$:

- automat *czyta* słowo wejściowe symbol po symbolu,
- ma skończoną liczbę stanów
- ma dodatkową pamięć typu $\mathcal{T}$
- język $L(A)$ słów *akceptowanych* przez automat składa sie z ciągów po wczytaniu których, po pewnej liczbie kroków, automat znajdzie się w stanie *akceptującym*

Klasa automatów odpowiada rodzajowi dodatkowej pamięci $\mathcal{T}$:

- **typ (0)** maszyny Turinga
  (pamięć $\mathcal{T}$ = taśma)
- **typ (1)** liniowo ograniczone
  ($\mathcal{T}$ = taśma o długości liniowej)
- **typ (2)** stosowe, inaczej zwane "ze stosem",
  ($\mathcal{T}$ = stos)
- **typ (3)** skończone ($\mathcal{T} = \emptyset$).

Może też być $\mathcal{T}$ = kolejka (kolejki), licznik (liczniki).

Twierdzenie

Gramatyki typu i odpowiadają automatom typu i, dla $i = 0 \ldots 3$.

Do opisu języka/słowa często używane są morfizmy (kodowania). Morfizm to funkcja

$$h:\ \Sigma_1 \to \Sigma_2^*$$

Funkcja h rozszerza sie automatycznie na wszystkie słówa, zapisujemy tę funkcje tak samo, choć (czysto formalnie) jest to już inna funkcja (rozszerzenie h):

$$h:\ \Sigma_1^* \to \Sigma_2^*$$

Rozszerzenie jest zdefiniowane tak aby zachodziło

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

Zdefiniujmy **język równościowy**:

$$EQ(h,g)\ =\ \{\, w \neq \varepsilon\ :\ h(w) = g(w)\,\}$$

Na przykład łatwo sie opisuje za pomocą zbiorów równościowych język

$$L = \{w \in \{a,b\}^+ \ : \ \#_a(w) = 2 \cdot \#_a(w)\}$$

Weźmy:

$$h(a) = 11; \ h(b) = \varepsilon, \ g(a) = \varepsilon, \ g(b) = 1.$$

Wtedy $L = EQ(h,g)$. Języki równościowe, jako problemy decyzyjne, są bardzo trudne.

Problem odpowiedniości Posta: sprawdzić czy $EQ(g,h) = \emptyset$ dla danych morfizów g, h.

Twierdzenie

Problem odpowiedniości Posta jest nierozstrzygalny.

Problem można zdefiniować inaczej. Dane są dwie listy słów:

$$(\,x_1, x_2, x_3, \dots x_k\,),\ \ (\,y_1, y_2, y_3, \dots y_k\,)$$

Pytamy czy istnieje niepusty ciąg $i_1, i_2, i_3, \dots i_m$ taki, że

$$x_{i_1} x_{i_2} x_{i_3} \dots x_{i_m} \ = \ y_{i_1} y_{i_2} y_{i_3} \dots y_{i_m}$$

Taki ciąg $i_1, i_2, i_3, \dots i_m$ nazywamy rozwiązaniem problemu.

Przykłady. Weźmy listy

$$(10,\ 011,\ 101),\ \ (\,101,\ 11,\ 011)$$

Wtedy problem nie ma rozwiązania. Dla list $(1, 10111, 10),\ \ (111, 10, 0)$ mamy (krótkie) rozwiązanie $x_2 x_1 x_1 x_3 \ = \ y_2 y_1 y_1 y_3$. Natomiast dla list: $(\,0,\ 1,\ 011\,),\ (\,1,\ 011,\ 0\,)$ istnieje rozwiązanie, ale najkrótsze ma długość $m = 75$.

Najpierw sprowadzamy problem do przypadku gdy rozwiązanie musi zaczynać się od $i_1 = 1$.
Wprowadzamy morfizmy $\alpha(s) = s*$, $\beta(s) = *s$.
Dla list

$$(x_1, x_2, \ldots x_k), \ (y_1, y_2, \ldots y_k)$$

Niech $*$ będzie nowym symbolem. Konstruujemy nowe listy:

$$(*\,\alpha(x_1), \alpha(x_1), \alpha(x_2), \alpha(x_3), \ldots \alpha(x_k), \ *)$$

$$(\beta(y_1), \ \beta(y_1), \ \beta(y_2), \ \beta(y_3), \ldots \beta(y_k), **)$$

Orginalny system ma rozwiązanie wtw. gdy nowy ma zaczynające się od pierwszej pary. Udowodnimy że tak zmodyfikowany problem Posta jest nierozstrzygalny.

Przypuśćmy, że mamy gramatykę G typu (0) i chcemy sprawdzić czy $u \xrightarrow{G}{}^{*} v$. Dołączamy symbole $\#$, \$ do alfabetu. Tworzymy listę (x_i, y_i) par złożoną z kilku podlist.

(x_1, y_1), gdzie $x_1 = \# \cdot u \cdot \#$, $y_1 = \#$ (a, a) dla każdego symbolu z $N \cup \{\#\}$.

(β, α) dla każdej produkcji $\alpha \to \beta$ gramatyki.

$(\$, \#v\$)$.

Wtedy zmodyfikowany problem odpowiedniości Posta ma rozwiązanie zaczynające się od 1 wtw. gdy $u \xrightarrow{G}{}^{*} v$. Z nierozstrzygalności sprawdzania $u \xrightarrow{G}{}^{*} v$ wynika nierozstrzygalność problemu Posta.

Przypuśćmy, że jedyną regułą gramatyki typu (0) jest

$$ab \to ba$$

oraz

$$u = aabb,\ v = bbaa$$

Wtedy wyprowadzeniu $aabb \to^* bbaa$ odpowiada następujący system:

1	2	3	4	5	6
$\#aabb\#$	a	b	ba	$\$$	$\#$
$\#$	a	b	ab	$\#bbaa\$$	$\#$

Najkrótszym rozwiązaniem tego systemu, zaczynającym sie od 1, jest ciąg indeksów:

$$1\,2\,4\,3\,6\,4\,4\,4\,6\,3\,4\,2\,5.$$

Twierdzenie

Problem jednoznaczności gramatyki bezkontekstowej jest nierozstrzygalny.

Twierdzenie

Problem pustości języka kontekstowego jest nierozstrzygalny.

W przeciwieństwie do poprzedniego faktu mamy:

Twierdzenie

Problem przynależności (ang. membership) do języka kontekstowego jest rozstrzygalny.