R O Z D Z I A Ł I

Podstawowe pojęcia i ogólne twierdzenia lingwistyki matematycznej

1. Wyrażenia, języki, gramatyki

Dowolny skończony niepusty zbiór V będziemy nazywać:

1) a l f a b e t e m, jeżeli będzie składać się on z liter,

2) s ł o w n i k i e m, gdy jego elementami będą słowa,

przyjmując jednak zarazem nazwę „alfabet" za uniwersalną.

Elementy V oznaczać będziemy małymi literami alfabetu łacińskiego (a, b, c,...) i nazywać je będziemy s y m b o l a m i a l f a b e t u (czy s ł o w n i k a) V.

W y r a ż e n i e m n a d V nazywać będziemy dowolny skończony ciąg symboli z V. Zbiór wszystkich wyrażeń nad V oznaczać będziemy przez V^*. Elementy z V^* oznaczać będziemy przez P, Q, R,..., przy czym dodatkowo obieramy symbol λ (czyt.: lambda) na oznaczenie pustego ciągu symboli z V. Zbiór wszystkich niepustych ciągów symboli nad V oznaczać będziemy przez V^+ (tj. $V^+ = V^* \setminus \{\lambda\}$).

Elementy V^* nazywamy:

1) s ł o w a m i, gdy tworzone są z liter alfabetu V,

2) z d a n i a m i, gdy tworzone są ze słów słownika V.

Przykład 1.1.

Gdy V = {a, b}, to V^* = {λ, a, b, aa, ab, ba, bb, aaa,...},

zaś V^+ = {a, b, aa, ab, ba, bb, aaa,...}. □

K o n k a t e n a c j ą w y r a ż e ń P i Q (gdzie $P = a_1...a_n$, a $Q = b_1...b_m$) nazywamy ich sklejenie, co oznaczamy i rozumiemy w następujący sposób:

$$P \cdot Q = PQ = a_1...a_n\, b_1...b_m.$$

λ jest oczywiście elementem neutralnym konkatenacji: $\lambda \cdot P = P \cdot \lambda = P$ (stąd $\lambda\lambda = \lambda$).

Bezpośrednio z definicji konkatenacji otrzymujemy, że jest ona łączna:

P(QR) = (PQ)R. Zwykle nie jest ona jednak przemienna (tj. zwykle $PQ \neq QP$).

Przykład 1.2.

Gdy P=kar, a Q=nawał, to PQ=karnawał, a QP=nawałkar. □

Wprowadźmy kolejne definicje:

1) D ł u g o ś c i ą w y r a ż e n i a $P \in V^*$ nazywamy liczbę |P| określoną indukcyjnie:

$$|P| = \begin{cases} 0, \text{ gdy } P = \lambda \\ n, \text{ gdy } P = a_1 \ldots a_n. \end{cases}$$

2) Przez P^n oznaczmy n - tą krotność wyrażenia P:
   $P^n = PP...P$, gdzie P występuje n razy, przy czym $P^0 = \lambda$.
3) Wyrażenie P nazywamy odcinkiem wyrażenia Q (co zapisujemy $P \subseteq Q$) witw, gdy istnieją pewne Q_1, Q_2 należące do V^* takie, że $Q = Q_1PQ_2$.

Zauważmy, że może być $Q_1 = \lambda$ lub $Q_2 = \lambda$.

4) Gdy w poprzedniej definicji $Q_1 = \lambda$, to mamy $Q = PQ_2$, a P nazywamy odcinkiem początkowym Q i fakt ten oznaczamy: $P \overset{\subseteq}{e} Q$.
5) Gdy z kolei w owej definicji $Q_2 = \lambda$, to mamy $Q = Q_1P$, a P nazywamy odcinkiem końcowym Q i fakt ten oznaczamy: $P \overset{\subseteq}{f} Q$.
6) Dla dowolnego słowa $P \in V^*$, przez P^{-1} określamy odbicie (zwierciadlane) wyrażenia P określone w następujący sposób: jeśli $P = a_1...a_n$, to $P^{-1} = a_n...a_1$.

Zauważmy, że zachodzi:

a) $(P^n)^{-1} = (P^{-1})^n$,

b) $(P^{-1})^{-1} = P$,

c) $(PQ)^{-1} = Q^{-1}P^{-1}$.

Zadanie 1.1: Wykaż zachodzenie powyższych zależności. □

Przykład 1.2 - c.d.

Gdy P=ma, a Q=ta, to PQ=mata, QP=tama, a $PP=P^2$=mama i $QQ=Q^2$=tata.

Nadto P^{-1}=am, a $(PQ)^{-1}$=atam=$Q^{-1}P^{-1}$. □

Załóżmy, że zbiór symboli alfabetu V jest uporządkowany (tj. zadana jest kolejność jego elementów). Fakt, że litera $x \in V$ jest wcześniejszą od litery $y \in V$, zapisywać będziemy $x \lessdot y$. Tak rozumiana relacja $\lessdot$ określa w zbiorze V porządek liniowy. Porządek ów możemy rozszerzyć na zbiór wszystkich słów w następujący sposób:

1) słowo puste poprzedza wszystkie pozostałe słowa,
2) litery alfabetu uporządkowane są według relacji $\lessdot$ porządku liter,
3) wszystkie słowa zaczynające się od pierwszej litery alfabetu są wcześniejsze od słów zaczynających się od drugiej litery, itd. Jeżeli dwa słowa zaczynają się od tej samej litery alfabetu, to wcześniejsze jest to, którego druga litera jest wcześniejszą. Jeśli i drugie litery są takie same, to należy rozpatrywać trzecie litery, itd.... Jeżeli słowo P stanowi początek słowa Q, to słowo P jest wcześniejsze od słowa Q (chyba, że są one identyczne).

Porządek ten jest zgodny z porządkiem słów w słowniku. Dlatego też nosi on nazwę p o r z ą d k u l e k s y k o g r a f i c z n e g o. Jest on bardzo wygodny dla skończonego zbioru słów (np. dla języka naturalnego i dlatego wykorzystuje się go m.in. w słownikach i encyklopediach). Jednak dla nieskończonego zbioru słów nad jakimś alfabetem jest on (jak to pokazuje poniższy przykład) dość dziwny i niepraktyczny.

Przykład 1.1 - c.d.

Dla V = {a, b} - V^* z wypisanymi elementami w kolejności leksykograficznej wyglądałby następująco:

$V^* = \{\lambda, a, aa, aaa, aaaa,..., aaab, aaaba,..., aab,...,ab,aba,abaa,...,b,ba,...\}$. □

Widzimy, że jest on co pewien czas „poprzetykany" wielokropkiem zastępującym nieskończoną liczbę słów, a zatem jest on niewłaściwy dla maszyny (tj. komputera), która nie byłaby w stanie wygenerować w tym porządku niektórych słów, nawet dysponując nieskończonym czasem (w powyższym przykładzie, chociażby dowolnego słowa zawierającego chociażby jedno b, gdyż po słowie zawierającym n a, generowałaby słowo zawierające n+1 a, następnie n+2 a,..., itd.). Dlatego też słowa nad alfabetem porządkuje się zazwyczaj w inny sposób, który nazywamy p o r z ą d k i e m z g o d n y m z d ł u g o ś c i ą, zgodnie z którym to słowa najpierw porządkuje się według długości, a potem dopiero słowa o równej długości porządkuje się leksykograficznie. Tak też zostały uporządkowane słowa w V^* w przykładzie 1.1 na początku tego rozdziału.

Zadanie 1.2: Pokaż, że:

a) dla zwykłego porządku leksykograficznego,

b) dla porządku leksykograficznego zgodnego z długością,

jeżeli x i y są takimi słowami, że $x \lessdot y$, a z jest dowolnym słowem, to $xz \lessdot yz$ oraz $zx \lessdot zy$. □

J ę z y k i e m (s e g m e n t o w y m, lub równoważnie c i ą g o w y m) L nad alfabetem V nazywamy dowolny zbiór słów utworzonych nad alfabetem V.

Język L jest więc pewnym podzbiorem zbioru V^*. Zauważmy ponadto, że tak sformułowana definicja nie precyzuje, które słowo jest, a które nie jest elementem języka. Podaje ona jedynie, że słowa w danym języku mogą być tworzone tylko nad danym alfabetem (tak więc np. słowo „vox" nie może być słowem języka polskiego, dopóki v i x nie staną się literami jego alfabetu). Ponieważ dla dowolnego V, zbiór V^* jest zawsze zbiorem nieskończonym, zatem z definicji języka wynika, że mogą być one tak zbiorami skończonymi, jak i nieskończonymi. Dany język możemy

dokładnie określić jedynie podając, z jakich słów się składa (przez ich wypisanie lub podanie ich kształtu).

Przykład 1.1 - c.d.

Nad alfabetem V = {a, b} można określić m.in. następujące języki:

1) $L_1 = \varnothing$ (język pusty, tj. język w ogóle nie posiadający słów),

2) $L_2 = \{\lambda\}$ (język, którego jedynym elementem jest słowo puste),

3) $L_3 = \{\lambda, a, b\}$ (język ze słowami o długości co najwyżej 1),

4) $L_4 = \{a^i b^i : i = 0,1,2,...\} = \{\lambda, ab, aabb, aaabbb,...\}$,

5) $L_5 = \{PP^{-1} : P \in V^*\}$,

6) $L_6 = \{ a^{n^2} : n = 0,1,2,...\}$,

7) Niech $N_a(P)$ - oznacza liczbę występowania symbolu a w P,

a $N_b(P)$ - liczbę występowania symbolu b w P.

Wówczas $L_7 = \{P: P \in \{a, b\}^+ \wedge N_a(P) = N_b(P)\}$ (np. ababba $\in L_7$). □

Zadanie 1.3: Opisz języki L_5 - L_7. □

Dla celów następnej definicji przyjmijmy oznaczenie, że gdy $A,B \subset V^*$, to

$AB = \{P_1P_2 : P_1 \in A \wedge P_2 \in B\}$.

G r a m a t y k ą (g e n e r a t y w n ą) nazywamy uporządkowaną czwórkę $G = <V_N,V_T,S,F>$, w której:

1) V_N i V_T są niepustymi, skończonymi i rozłącznymi alfabetami,

2) $S \in V_N$,

3) $F \subseteq (V_N \cup V_T)^* V_N (V_N \cup V_T)^* \times (V_N \cup V_T)^*$.

V_N - to a l f a b e t n i e t e r m i n a l n y (pomocniczy), złożony ze z m i e n n y c h s y n t a k t y c z n y c h (lub równoważnie: metajęzykowych albo pomocniczych).

V_T - to alfabet symboli terminalnych (przedmiotowych), albo krócej - a l f a b e t t e r m i n a l n y (końcowy).

S - to s y m b o l p o c z ą t k o w y.

F - to zbiór par (P,Q), takich że w P występuje zawsze przynajmniej jeden symbol nieterminalny, a Q jest dowolnym wyrażeniem o elementach należących do $V_N \cup V_T$.

Elementy F nazywamy r e g u ł a m i p r o d u k c j i (lub p r z e p i s y w a n i a).

Zamiast pisać (P,Q), piszemy zwykle $P \rightarrow Q$.

P nazywamy tu p o p r z e d n i k i e m , a Q n a s t ę p n i k i e m p r o d u k c j i .

Zasada stosowania reguł produkcji jest następująca: jeśli mamy wyrażenie, którego częścią jest wyrażenie P, a w gramatyce, którą się posługujemy, jest reguła produkcji $P \to Q$, wówczas możemy wyrażenie to zastąpić nowym wyrażeniem, w którym na miejscu P jest Q. Zobaczmy to na poniższym przykładzie:

Przykład 1.3.

Niech $G = <V_N, V_T, S, F>$, gdzie

V_N = {<zdanie>, <podmiot>, <orzeczenie>} (jest to zbiór symboli nieterminalnych, czyli pomocniczych, zwanych tu zmiennymi metajęzykowymi),

V_T ={a, ą, b,..., z, ź, ż} (alfabet języka polskiego),

S = <zdanie> (symbol początkowy, określający końcowy wynik działania gramatyki),

F = {<zdanie> → <podmiot> <orzeczenie>, <podmiot> → Jaś, <orzeczenie> → śpi} (zbiór reguł produkcji).

W gramatyce tej wyrażenie „Jaś śpi" otrzymujemy następująco:

<zdanie> → <podmiot> <orzeczenie> → Jaś <orzeczenie> → Jaś śpi □

Mówimy, że gramatyka G przekształca bezpośrednio wyrażenie $P \in (V_N \cup V_T)^*$ (co czytamy: P utworzone nad sumą alfabetów terminalnego i nieterminalnego) w wyrażenie $Q \in (V_N \cup V_T)^*$, co oznaczamy: $P \underset{G}{\to} Q$, witw, gdy stosując jeden raz jedną z reguł produkcji gramatyki G, z wyrażenia P otrzymujemy wyrażenie Q.

Uogólnijmy pojęcie przekształcania bezpośredniego.

Mówimy, że gramatyka G przekształca słowo $P \in (V_N \cup V_T)^*$ w słowo $Q \in (V_N \cup V_T)^*$ (co oznaczamy: $P \overset{*}{\underset{G}{\to}} Q$) witw, gdy istnieje ciąg wyrażeń

$P=P_0, P_1,..., P_n =Q$, taki że $P=P_0 \underset{G}{\to} P_1 \underset{G}{\to} P_2 \underset{G}{\to} ... \underset{G}{\to} P_n =Q$.

Taki ciąg $P_0,P_1,...,P_n$ nazywamy wyprowadzeniem (derywacją) wyrażenia Q z P.

Przykład 1.4.

Niech V = {a, b, c, +, ·, (,)}. Skonstruujemy gramatykę złożoną ze wszystkich poprawnie zbudowanych wyrażeń algebraicznych z elementów zbioru V:

$V_N = \{S\}$,

$V_T = V$,

$F = \{S \to S + S, S \to S \cdot S, S \to (S + S), S \to (S \cdot S), S \to a, S \to b, S \to c\}$.

W gramatyce tej wyrażenie ((a·b)+c) otrzymujemy następująco:

$$S \underset{G}{\rightarrow} (S+S) \underset{G}{\rightarrow} ((S\cdot S)+S) \underset{G}{\rightarrow} ((a\cdot S)+S) \underset{G}{\rightarrow} ((a\cdot b)+S) \underset{G}{\rightarrow} ((a\cdot b)+c).$$

Tak więc gramatyka ta przekształca bezpośrednio np. wyrażenie ((a·S)+S) w wyrażenie ((a·b)+S). Podobnie możemy powiedzieć, że przekształca ona wyrażenie (S+S) w wyrażenie ((a·b)+S). □

Zauważmy, że:

1) zmienne alfabetu nieterminalnego są rzeczywiście jedynie zmiennymi pomocniczymi. W powyższym przykładzie służą one ponadto do wyodrębnienia w wyrażeniu grup (wraz z regułami produkcji są więc tu odpowiedzialne za jego syntaktykę). Podobnie ma się sprawa w przypadku przykładu 1.3;
2) gdy wiadomo jest, z jakiej gramatyki korzystamy, to przy kolejnych krokach derywacji nie musimy pisać pod strzałką „G", specyfikującego tę gramatykę.

Zadanie 1.4: W gramatyce z przykładu 1.4 wyprowadź wyrażenie

$$(((a+b)\cdot c+(b+a)\cdot a)\cdot b).$$ □

Zadanie 1.5: Udowodnij następujące własności wynikania bezpośredniego:

a) $v \overset{*}{\rightarrow} v$ dla dowolnego $v \in V^*$,

b) jeżeli $v \overset{*}{\rightarrow} u$ i $u \overset{*}{\rightarrow} w$, to $v \overset{*}{\rightarrow} w$ dla dowolnych $u, v, w \in V^*$,

c) jeżeli $v \overset{*}{\rightarrow} u$, to $vy \overset{*}{\rightarrow} uy$ i $yv \overset{*}{\rightarrow} yu$ dla dowolnych $u, v, y \in V^*$. □

Dla P i Q takich, że $P \overset{*}{\rightarrow} Q$, do Q z P często można dojść na wiele sposobów, i to w dodatku mogą się różnić one swą długością. W takiej sytuacji za **długość derywacji** będziemy przyjmować długość najkrótszej derywacji prowadzącej z P do Q.

Językiem L(G) generowanym przez gramatykę generatywną G nazywamy zbiór

$$L(G) = \{P \in V_T^* : S \overset{*}{\underset{G}{\rightarrow}} P\}.$$

Jest to więc zbiór tych słów utworzonych nad alfabetem terminalnym gramatyki G, które są w niej wyprowadzalne z jej symbolu początkowego S.

Tak określony język generowany przez gramatykę jest oczywiście językiem (tj. spełnia definicję języka). Składa się on bowiem ze słów utworzonych nad pewnym alfabetem (tu: V_T); jest on podzbiorem (wyznaczonym przez wyprowadzalność z S) zbioru wszystkich słów z V_T^*.

Mówimy, że dwie g r a m a t y k i G_1 i G_2 s ą (s ł a b o) r ó w n o w a ż n e witw, gdy języki przez nie generowane są identyczne (tj. $L(G_1) = L(G_2)$).

Z kolei mówimy, że dwie g r a m a t y k i s ą m o c n o r ó w n o w a ż n e, gdy są one słabo równoważne i identyczne są odpowiednie drzewa derywacji w tych gramatykach (z którymi szczegółowiej zaznajomimy się w dalszej części kursu).

Poniżej podano dwa przykłady równości języków (generowanego przez gramatykę i zdefiniowanego przez podanie konstytuujących go słów).

Przykład 1.5.

Niech $G = <V_N, V_T, S, F>$, $V_N = \{S, A, B\}$, $V_T = \{a, b\}$,

$F = \{S \to aB, B \to aBB, B \to b, B \to bS, S \to bA, A \to bAA, A \to a, A \to aS\}$.

Wówczas $L(G) = L_7$ (gdzie L_7 jest językiem zdefiniowanym w przykładzie 1.1 na str. 9). □

Przykład 1.6.

Niech $G = <\{S, X, Y\}, \{a, b, c\}, S, F>$, gdzie

$F = \{S \to abc, S \to aXbc, Xb \to bX, Xc \to Ybcc, bY \to Yb, aY \to aaX, aY \to aa\}$.

Wówczas $L(G) = \{a^n b^n c^n : n \geq 1\}$. □

Zauważmy na koniec, że wyprowadzanie wyrażenia z S może się zatrzymać wyłącznie w dwóch przypadkach:

1) gdy uzyskamy wyrażenie składające się wyłącznie z elementów z V_T

($S \to ... \to P \in V_T^*$),

2) gdy dostaniemy wyrażenie $P \in (V_N \cup N_T)^*$ takie, że nie ma reguł produkcji, które to byśmy mogli do niego zastosować.

(1) jest oczywiście szczególnym przypadkiem 2)).

Zadanie 1.6: Niech $G = <V_N, V_T, S, F>$ i $G' = < V_N, V_T, S, F'>$, gdzie $F' \subset F$.

Pokaż, że $L(G') \subseteq L(G)$. □

Zadanie 1.7: Udowodnij, że jeśli $x \overset{*}{\to} y$ i $x = x_1\ x_2\ ...\ x_k$, to istnieją słowa

$y_1, y_2, ..., y_k$, takie że $x_1 \overset{*}{\to} y_1, ..., x_k \overset{*}{\to} y_k$, a $y = y_1\ y_2\ ...\ y_k$. □

Zadanie 1.8: Nad alfabetem języka polskiego zdefiniuj gramatykę generującą tylko i wyłącznie wszystkie słowa będące imionami studentek i studentów z twojej grupy. □

Zadanie 1.9: Niech symbol „_” oznacza spację. Skonstruuj gramatykę generującą tylko i wyłącznie wszystkie słowa postaci „Mam_Q.”, gdzie zamiast Q stoi dowolny niepusty ciąg liter utworzony nad alfabetem polskim. □

2. Hierarchia Chomsky'ego

Amerykański lingwista N. Chomsky sklasyfikował gramatyki ze względu na kształt dopuszczalnych w nich reguł produkcji. Zgodnie z ową klasyfikacją mówimy, że g r a m a t y k a generatywna $G = \langle V_N, V_T, S, F \rangle$ j e s t t y p u (lub równoważnie k l a s y):

1) 0 (lub g r a m a t y k ą s t r u k t u r f r a z o w y c h), gdy wszystkie jej reguły produkcji mają postać $P \to Q$, gdzie $P \in (V_N \cup V_T)^* V_N (V_N \cup V_T)^*$, zaś $Q \in (V_N \cup V_T)^*$ (tj., w stosunku do definicji gramatyki, brak jest w niej jakichkolwiek ograniczeń na kształt reguł produkcji);
2) 1 (lub g r a m a t y k ą k o n t e k s t o w ą bądź czułą na kontekst, a po angielsku: context-sensitiv grammar, lub krócej CS-grammar), gdy wszystkie jej reguły produkcji mają postać $Q_1AQ_2 \to Q_1PQ_2$, gdzie $Q_1, Q_2 \in (V_N \cup V_T)^*$, $A \in V_N$, zaś $P \in (V_N \cup V_T)^* \setminus \{\lambda\}$, bądź też $S \to \lambda$, ale wówczas S nie występuje po prawej stronie w żadnej z reguł produkcji;
3) 2 (lub b e z k o n t e k s t o w ą, a po angielsku: context-free grammar, lub krócej CF-grammar), gdy wszystkie jej reguły produkcji mają postać $A \to P$, gdzie $A \in V_N$, zaś $P \in (V_N \cup V_T)^*$ (tj. są postaci $\lambda A\lambda \to \lambda P\lambda$);
4) 3 (lub równoważnie gramatyką: r e g u l a r n ą, prawoliniową, prawostronnie liniową, automatową), gdy wszystkie jej reguły produkcji mają postać $A \to P$ lub $A \to PB$, gdzie $A,B \in V_N$, zaś $P \in V_T^*$.

Jak się przekonamy, każda gramatyka klasy i (i=0,1,2,3) jest równocześnie gramatyką klasy j, dla $0 \le j \le i$. Mamy więc rzeczywiście do czynienia z h i e r a r c h i ą (wzajemnym zawieraniem się zbiorów).

Język L nazywamy j ę z y k i e m t y p u i, gdy jest on generowany przez pewną gramatykę typu i (tj. L = L(G), gdzie G jest gramatyką typu i).

Klasę języków typu i oznaczamy $\mathcal{L}_i$.

Z powyższych definicji widzimy, że $\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_0$ oraz $\mathcal{L}_1 \subseteq \mathcal{L}_0$.

W dalszym ciągu pokażemy, że zachodzi następujące właściwe zawieranie się klas języków: $\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subseteq \mathcal{L}_0$ (tj., że ponadto $\mathcal{L}_2 \subseteq \mathcal{L}_1$ oraz że między $\mathcal{L}_1$, $\mathcal{L}_2$ i $\mathcal{L}_3$ zachodzą inkluzje właściwe). W rzeczywistości zachodzi nawet: $\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$, tj. dodatkowo $\mathcal{L}_1$ właściwie zawiera się w $\mathcal{L}_0$, ale w ramach naszego kursu faktu tego formalnie nie wykażemy.

3. Operacje na językach. Zamkniętość klas języków ze względu na operacje regularne

Ponieważ (zgodnie z definicją) języki są zbiorami, zatem możemy wykonywać na nich przynależne im operacje. Dla dowolnych języków L_1, L_2 (o słowach z V^*), mamy więc:

$L_1 \cup L_2 = \{P: P \in L_1 \vee P \in L_2\}$ (suma języków);

$L_1 \cap L_2 = \{P: P \in L_1 \wedge P \in L_2\}$ (iloczyn języków);

$L_1 \setminus L_2 = \{P: P \in L_1 \wedge P \notin L_2\}$ (różnica języków);

$\overline{L_1} = V^* \setminus L_1$ (dopełnienie języka);

Zadanie 1.10: Wykaż, że dla dowolnego języka L:

$L \cup \varnothing = \varnothing \cup L = L$, a $L \cup L = L$. □

Zadanie 1.11: Niech $V = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

a) Co to jest V^*? Jakie znaczenia mają słowa z V^*? Czy są to liczby?

b) Określ tak język L nad alfabetem V, by był on złożony (tylko i wyłącznie) ze wszystkich liczb naturalnych. Zdefiniuj język $\overline{L}$.

c) Jakie własności miałby ten język L, jeśli dodatkowo spełniałby warunek: $y \in L \Leftrightarrow y^{-1} \in L$. Co powiesz o analogicznym przypadku dla $\overline{L}$? □

$L_1 L_2 = \{P_1 P_2 : P_1 \in L_1 \wedge P_2 \in L_2\}$ (konkatenacja języków),

przy czym, ponieważ możemy tworzyć konkatenację (i to wielokrotną) tylko jednego języka (postaci: LL, LLL,...), więc definiujemy też:

$$L^i = \begin{cases} \{\lambda\}, \text{gdy } i = 0 \\ L^k L, \text{gdy } i = k+1; \end{cases}$$

Zadanie 1.12: Wykaż, że $\varnothing L = L\varnothing = \varnothing$, a z kolei $\{\lambda\}L = L\{\lambda\} = L$

(WSKAZÓWKA: „$\varnothing L$” oznacza język, w którym nic nie może stać na pierwszym miejscu). □

Zadanie 1.13: Niech K, L i M będą trzema dowolnymi językami. Udowodnij, że (LM)K= L(MK), tzn. że operacja konkatenacji języka jest łączna. □

$L^* = \bigcup_{i\geq 0} L^i$ (domknięcie Kleene'go języka L);

$L^+ = \bigcup_{i>0} L^i$ (niepełne domknięcie Kleene'go języka L);

Zadanie 1.14: Udowodnij, że zachodzi:

a) $L^* = L^+ \Leftrightarrow \lambda \in L$,

b) $L^+ = L^* \setminus \{\lambda\} \Leftrightarrow \lambda \notin L$. □

Dla dowolnego języka L: $L^{-1} = \{P^{-1}: P \in L\}$ (odbicie lustrzane języka L).

Zadanie 1.15: Wykaż, że:

a) $(L^{-1})^i = (L^i)^{-1}$,

b) $(L^{-1})^{-1} = L$. □

Operacjami regularnymi na językach nazywa się sumę języków, konkatenację języków i domknięcie Kleene'go języka.

Twierdzenie 1.1.

Każda z klas języków typu i (dla i = 0,1,2,3) jest zamknięta ze względu na operacje regularne.

Ponieważ **dowód** tego twierdzenia jest dość pracochłonny (zajmuje kilka stron) - więc go pomijamy, tym bardziej, że twierdzenie to w pewnej modyfikacji i zawężeniu zostanie podane i udowodnione w trzecim rozdziale niniejszego podręcznika (tw. 3.3.). □

4. I twierdzenie o postaci normalnej

Aby móc udowodnić tytułowe twierdzenie niniejszego paragrafu, musimy najpierw zdefiniować pojęcie homomorfizmu.

Niech mianowicie V_1 i V_2 będą dwoma alfabetami.

Odwzorowanie h: $V_1^* \rightarrow V_2^*$ nazywamy homomorfizmem, gdy:

a) $\forall\, P\in V_1^*\ \exists!\ W\in V_2^*$: W=h(P),

b) $\forall\, P,Q\in V_1^*$ h(PQ)=h(P)h(Q)

(symbol „$\exists!$ a" oznacza: istnieje dokładnie jedno a).

Tak więc jest to odwzorowanie jednoznaczne, zawsze istnieje, a ponadto zachowuje ono konkatenację wyrażeń. W związku z tym, dowolny homomorfizm wystarczy określić jedynie dla alfabetu, bo gdy $P = a_1...a_n$, to $h(P) = h(a_1)...h(a_n)$.

Oczywiście $h(\lambda)=\lambda$, bo: gdy $P\neq\lambda$, to $h(\lambda P) = h(\lambda)h(P) \rightarrow h(\lambda) = \lambda$.

$$\underset{h(P)}{\overset{h(\lambda P)}{\|}}$$

Homomorfizm nazywamy λ - *wolnym*, gdy $\forall_{P\neq\lambda}\ h(P)\neq\lambda$.

Gdy $L \subseteq V_1^*$, a $h: V_1^* \rightarrow V_2^*$, to *homomorficzny obraz języka* L definiujemy w następujący sposób: $h(L) = \{h(P): P \in L \subseteq V_1^*\}$.

Obecnie możemy przejść do podania drugiego już z naczelnych twierdzeń lingwistyki matematycznej.

Twierdzenie 1.2 (I twierdzenie o postaci normalnej)

Dla każdej gramatyki $G_i = \langle V_N, V_T, S, F\rangle$ (i=0, 1, 2, 3), istnieje równoważna jej gramatyka $G_i' = \langle V_N', V_T, S, F'\rangle$ tego samego typu, taka że symbole terminalne <u>nie</u> występują po lewej stronie reguł produkcji w F'.

Dowód.

Dla gramatyk typu 2 i 3 nie ma czego dowodzić, bo wszystkie reguły produkcji mają w nich postać:

1) w gramatykach typu 2: $A \rightarrow P$, gdzie $A \in V_N$, a $P \in (V_N \cup V_T)^*$,

2) w gramatykach typu 3: $A \rightarrow P$ lub $A \rightarrow PB$, gdzie $A, B \in V_N$, a $P \in V_T^*$,

tj. po ich lewych stronach nie występują symbole terminalne.

Wystarczy więc udowodnić twierdzenie w przypadku, gdy i=0 i gdy i=1 (gdzie symbole terminalne po lewych stronach reguł produkcji mogą wystąpić).

W naszej gramatyce $G = \langle V_N, V_T, S, F\rangle$, niech $V_T = \{a_1,\dots, a_k\}$.

Dla każdego $a_i \in V_T$, obieramy nowy symbol nieterminalny $A_i \notin V_N$. Wówczas definiujemy $V_N' = V_N \cup \{A_1,\dots, A_k\}$, a następnie tworzymy F' w następujący sposób:

1) każdą regułę produkcji $\ulcorner P \rightarrow Q \urcorner \in F$ zastępujemy nową regułą, w której wszystkie symbole terminalne występujące w starej regule zastąpione zostają odpowiadającymi im symbolami nieterminalnymi (tak więc np. regułę $\ulcorner Aa_1B \rightarrow a_1a_2a_3 \urcorner \in F$, zastępujemy regułą $\ulcorner AA_1B \rightarrow A_1A_2A_3 \urcorner \in F'$),
2) dla każdego $1 \leq i \leq k$, do F' włączamy reguły postaci $A_i \rightarrow a_i$.

W ten sposób utworzyliśmy gramatykę $G' = \langle V_N', V_T, S, F'\rangle$, w której rzeczywiście symbole terminalne nie występują po lewej stronie reguł produkcji.

Pozostaje nam jeszcze jedynie pokazać, że $L(G) = L(G')$.

a) $L(G) \subseteq L(G')$, bo jeśli $a_{i_1} \dots a_{i_n} \in L(G)$, to $S \overset{*}{\underset{G}{\rightarrow}} A_{i_1} \dots A_{i_n} \overset{*}{\underset{G}{\rightarrow}} a_{i_1} \dots a_{i_n} \in L(G')$.

Najpierw stosowaliśmy tu reguły typu 1), odpowiednie do reguł gramatyki G, gwarantujące nam uzyskanie takich samych słów, jednak zbudowanych jedynie z symboli nieterminalnych, a następnie reguły typu 2), zamieniające poszczególne nieterminały na odpowiadające im terminały. W wyniku tych operacji, w gramatyce G', uzyskiwaliśmy poszczególne słowa języka L(G).

b) Pokażemy, że $L(G) \supseteq L(G')$.

W tym celu określmy homomorfizm h: $(V_N' \cup V_T)^* \rightarrow (V_N \cup V_T)^*$ w następujący sposób:

1) $\forall_{1 \le i \le k}\ h(A_i) = a_i$,

2) $h(x) = x$ dla pozostałych elementów z alfabetu $V_N' \cup V_T$.

Tak określona funkcja h działa na symbolach alfabetu w odwrotną stronę niż w przypadku funkcji zmiany liter w regułach produkcji F na reguły produkcji F'.

Mamy wówczas, że gdy $P, Q \in (V_N' \cup V_T)^*$, a $P \overset{*}{\underset{G'}{\rightarrow}} Q$, to $h(P) \overset{*}{\underset{G}{\rightarrow}} h(Q)$ (wniosek ten otrzymujemy bezpośrednio z definicji homomorfizmu h, rozpatrując oddzielnie stosowanie obydwu typów reguł produkcji z F'). W szczególności, obierając S za P, otrzymujemy że: gdy $S \overset{*}{\underset{G'}{\rightarrow}} Q$ (gdzie $Q \in V_T^*$), to $S = h(S) \overset{*}{\underset{G}{\rightarrow}} h(Q) = Q$ (gdzie równości uzyskaliśmy na mocy 2)). Zatem $L(G) \subseteq L(G')$, co już kończy dowód. □