ACTA UNIVERSITATIS WRATISLAVIENSIS No 3397
Kształcenie Językowe 10 (20) Wrocław 2012

Agnieszka Kułacka
King's College London

Język naturalny a język regularny

1. Wstęp

Na początku ubiegłego stulecia językoznawcy podjęli pierwsze próby sformalizowania składni języka naturalnego. W latach 30. XX wieku Kazimierz Ajdukiewicz stworzył rachunek zdań, który nazwał gramatyką kategorialną[1]. Konstruowanie modeli matematycznych, mających na celu opisanie fragmentów składni języka naturalnego, a później tworzenie języków sztucznych, stało się przedmiotem badań nowo powstałej dyscypliny, która początkowo była dziedziną matematyki. W chwili, gdy językoznawstwo matematyczne stało się niezależną dyscypliną, powstały dwa trendy badawcze: syntaktyczny i analityczny. Naukowcy podejmujący badania w pierwszej gałęzi zajmują się gramatykami formalnymi, ich typami oraz cechami języków, które generują, a także badają automaty, czyli maszyny formalne rozpoznające język wygenerowany przez gramatykę. Gałąź analityczna językoznawstwa matematycznego podczas konstruowania modeli składni języka naturalnego korzysta z osiągnięć dziedzin matematycznych, takich jak teoria zbiorów, logika, teoria grafów itp. Osiągnięcia syntetycznej poddyscypliny językoznawstwa matematycznego wspierają badania jego gałęzi analitycznej.

Jedną z pierwszych gramatyk formalnych, jaką wykorzystano do opisu fragmentu składni języka naturalnego, była gramatyka skonstruowana dla języków regularnych, tj. takich, które rozpoznawane są przez automaty skończone. Noam Chomsky podjął wiele prób[2], by udowodnić, że język angielski nie jest językiem regularnym. Kolejną próbę podjęli Barbara Partee, Alice Ter Meulen, Robert Wall[3]. W niniejszym artykule omówię ważne pojęcia i twierdzenie potrzebne, by zrozumieć argumentację przedstawioną w poprawionym przeze mnie wspo-

[1] Zob. A. Kułacka, *Syntax of a montague grammar*, „LingVaria" 6 (2011) | 1 (11), s. 9–23.

[2] N. Chomsky, *Three models for the description of language*, „IRE Translations on Information Theory" 2, nr 3, 1956, s. 113–124; *idem*, *Syntactic Structures*, Berlin 1957 (reedycja 2002); *idem*, *Introduction to the Formal Analysis of Natural Languages*, [w:] *Handbook of Mathematical Psychology*, t. 2, red. R.D. Luce, R.R. Bush, E. Galanter, New York 1967.

[3] B. Partee, A. Ter Meulen, R.E. Wall, *Mathematical Methods in Linguistics*, London 1990.

mnianym dowodzie. Wskażę też na podobne konstrukcje gramatyczne istniejące w języku polskim, co dowodzi, że język ten jest również niereguralny.

2. Teoria automatów

Teoria automatów jest dziedziną informatyki zajmującą się maszynami obliczeniowymi nazywanymi automatami. Powstała w latach 30. ubiegłego wieku wraz z pracami Alana Turinga nad możliwościami tych maszyn. Od tego czasu badania w zakresie teorii automatów zostały wykorzystane w modelowaniu funkcji mózgu, projektowaniu układów cyfrowych, opracowywaniu gramatyk formalnych itd. W tym rozdziale wprowadzę kluczowe pojęcia teorii, konieczne do zrozumienia pozostałej części artykułu, oraz omówię wyrażenia regularne, wzorce opisujące ciągi symboli rozpoznawane przez automat, z którym są związane.

2.1. Kluczowe pojęcia teorii automatów

Jednym z kluczowych pojęć teorii jest alfabet Σ, który jest skończonym, niepustym zbiorem symboli. Przyjrzyjmy się kilku alfabetom, takim jak: (a) alfabet binarny, gdzie $\Sigma_1 = \{0,1\}$, (b) alfabet złożony z liter drukowanych, $\Sigma_2 = \{A, B, ..., Z\}$ itd. Możemy również zdefiniować alfabet, który składa się z czterech słów: $\Sigma_3 = \{Maria, Jan, Smiles, Likes\}$.

Kolejnym ważnym pojęciem jest ciąg, nazywany również słowem. Ciąg jest skończoną sekwencją symboli alfabetu. W powyższym akapicie podałam przykłady trzech alfabetów. Korzystając z alfabetu binarnego Σ_1, możemy utworzyć następujące ciągi: 101, 111, 10001, ale nie możemy utworzyć 102, gdyż 2 nie jest symbolem należącym do Σ_1. Ciągi ALPHA i OMEGA powstały z liter alfabetu Σ_2. Alfabet Σ_3 składający się ze słów symbolicznych pozwala na utworzenie następujących ciągów: *LikesMariaMariaJanSmiles* itp. W tej chwili nie zajmujemy się gramatycznością, rozumianą w konwencjonalnym sensie, tych ciągów wyrazów, a tylko tym, czy mogą one powstać z liter danego alfabetu. Jest jeden ciąg, który może zostać zawsze utworzony niezależnie od alfabetu. Oznaczony jest symbolem ε i nazywany ciągiem pustym, gdyż nie występują w nim żadne symbole alfabetu.

Możemy podzielić zbiór złożony z ciągów utworzonych z alfabetu na grupy, w zależności od ich długości. Zbiór złożony ze wszystkich ciągów z alfabetu Σ o długości n, oznaczany Σ^n, jest n-tą potęgą zbioru Σ. Niezależnie od alfabetu $\Sigma^0 = \{\varepsilon\}$. Rozważmy teraz niektóre potęgi alfabetu $\Sigma = \{Maria, Jan, Smiles, Likes\}$.

(a) Jak wspomniałam $\Sigma^0 = \{\varepsilon\}$.

(b) $\Sigma^1 = \{Maria, Jan, Smiles, Likes\}$. Należy rozróżnić Σ^1 i Σ. Pierwszy jest zbiorem ciągów o długości 1, a drugi jest alfabetem.

(c) Drugą potęgą Σ jest następujący zbiór:

Σ^2 = {*MariaMaria; MariaJan; MariaSmiles; MariaLikes; JanMaria; JanJan; JanSmiles; JanLikes; SmilesMaria; SmilesJan; SmilesSmiles; SmilesLikes; LikesMaria; LikesJan; LikesSmiles; LikesLikes*}.

Możemy dalej wymieniać kolejne potęgi alfabetu Σ. Ogólnie suma wszystkich potęg Σ jest oznaczana przez Σ*. Jeśli wyłączymy z Σ* ciąg pusty, to otrzymamy zbiór wszystkich ciągów o długości co najmniej 1, oznaczany przez Σ^+. Związek pomiędzy tymi dwoma zbiorami oddany jest przez równanie: $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$. Możemy też łączyć dwa ciągi *x* i *y* w jeden ciąg *xy*, tj. ustawić ciąg *y* za ciągiem *x*. Konkatenacja ciągów *y* i *x* różni się od konkatenacji ciągów *x* i *y*. Rezultatem pierwszej jest ciąg *yx*, a drugiej *xy*. Konkatenacja ciągu pustego i ciągu *y* w tej czy odwrotnej kolejności daje ciąg *y*.

Językiem L nazywamy podzbiór złożony z ciągów należących do Σ*, gdzie Σ jest alfabetem. Mówimy, że L jest językiem nad alfabetem Σ. Należy zauważyć, że możemy nie wykorzystać wszystkich symboli z alfabetu Σ w ciągach, które należą do danego języka. Ponadto, nawet jeśli alfabet jest skończony, język może składać się z nieskończonej liczby ciągów. Przyjrzyjmy się następującym przykładom.

Trzema najważniejszymi językami utworzonymi nad alfabetem Σ są: (1) Σ*, (2) ∅, język pusty, (3) {ε}, język składający się tylko z ciągu pustego ε. Zwróćmy uwagę na to, że języki (2) i (3) są różne. Język (2) nie ma ani jednego ciągu, natomiast na język (3) składa się jeden ciąg, ciąg pusty.

Każdy z poniższych zbiorów jest językiem nad alfabetem Σ = {*Maria, Jan, Smiles, Likes*}.

L_1 = {*MariaSmiles; JanSmiles; SmilesJan; LikesSmiles; Jan; ε*};

(a) L_2 = {*JanLikesMaria; JanJan; JanJanJan; Maria*};

L_3 = {*ε; Jan*; *JanJan; JanJanJan; JanJanJanJan; ...*}.

Język w ostatnim przykładzie jest językiem nieskończonym. Możemy zdefiniować elementy tego języka rekurencyjnie, czyli za pośrednictwem definicji odwołującej się do samej siebie, do elementu o mniejszym stopniu komplikacji: (1) $\varepsilon \in L_3$, (2) każdy z ciągów o długości *n* jest utworzony za pomocą konkatenacji ciągu o długości *n-1* i symbolu *Jan*, (3) żaden inny ciąg nie należy do L_3.

2.2. Wyrażenia regularne

Ze względu na to, że można udowodnić, iż język jest regularny wtedy i tylko wtedy, gdy opisuje go jakieś wyrażenie regularne[4], pominę definicję automatu. Nie jest ona konieczna do zrozumienia dowodu, który jest główną treścią tego artykułu, a w zamian przedstawię definicję wyrażenia regularnego. W następnym

[4] Zob. J.E. Hopcroft, R. Motwani, J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Boston 2001, s. 90 n.; M. Sipser, *Introduction to the Theory of Computation*, Boston 1997, s. 66 n.

rozdziale pokażę, jak można wyprowadzić z wyrażenia regularnego pewną gramatykę formalną.

Wyrażenie regularne R tworzy się w sposób rekurencyjny. Podstawą definicji jest następujące zdanie (1), wyrażeniami regularnymi są: a dla pewnego $a \in \Sigma$, gdzie Σ jest pewnym alfabetem, oraz ε i $\varnothing$. Krok indukcyjny definiuje się w następujący sposób: (2) jeśli R_1 i R_2 są wyrażeniami regularnymi, to $R_1 + R_2$, R_1R_2 i R_1^* są również wyrażeniami regularnymi. Wyrażania zdefiniowane w punkcie (2) opisują następujące języki: $L(R_1 + R_2) = L(R_1) \cup L(R_2)$, $L(R_1R_2) = L(R_1)L(R_2)$ (konkatenacja ciągów ze zbiorów R_1 i R_2 w tej kolejności) oraz $L(R_1^*) = (L(R_1))^*$.

Rozważmy następujące przykłady. Załóżmy, że $\Sigma_1 = \{0,1\}$. Symbol w oznacza dowolny ciąg danego języka.

(a) $L(\mathbf{01}^*) = \{w | w$ *zaczyna się od* 0*, po którym następuje zero lub więcej syboli* 1$\}$.

(b) $L(\mathbf{0 + 1}) = \{0,1\}$.

(c) $L(\mathbf{0 + 1}^*) = \{w | w$ *jest* 0 *lub ciągiem zera lub więcej symboli* 1$\}$.

(d) $L((\mathbf{01})^*) = \{w | w$ *składa się z zera lub więcej ciągów symboli* 01$\}$.

Zauważmy, że te języki regularne są językami wygenerowanymi przez gramatyki formalne typu 3, najniższymi w hierarchii Chomskiego, co oznacza, że języki te mogą być wygenerowane przez gramatyki typów wyższych (typu 2, typu 1 i typu 0), ale niekoniecznie odwrotnie. W niniejszym artykule zdefiniuję tylko gramatyki typu 3.

3. Gramatyka formalna dla języków regularnych

Rozpocznę od zdefiniowania gramatyki formalnej w sensie ogólnym, by później podać warunki, jakie musi ona spełniać, by generować języki regularne[5]. Gramatyką formalną nazywamy czwórkę $G = (\Sigma, V, S, P)$, gdzie Σ jest skończonym zbiorem symboli terminalnych (symboli wykorzystanych do budowania ciągów języka), V skończonym zbiorem symboli nieterminalnych (symboli pomocniczych), $S \in V$ jest symbolem startowym, a P jest skończonym zbiorem reguł wyprowadzania postaci $S_1 \rightarrow S_2$, gdzie S_1 i S_2 są ciągami symboli terminalnych i nieterminalnych.

Jeśli x, y, z są ciągami symboli terminalnych i nieterminalnych, A jest symbolem nieterminalnym, wtedy $A \rightarrow x$ jest regułą wyprowadzania i mówimy, że z yAz wyprowadzamy yxz oraz piszemy $yAz \Rightarrow yxz$. Niech $x_1, x_2, ..., x_k$, $k \geq 0$ będą ciągami symboli terminalnych i nieterminalnych. Piszemy wtedy $y \overset{*}{\Rightarrow} z$ jeśli $y = z$ (zero kroków) lub $y \Rightarrow x_1 \Rightarrow x_2 \Rightarrow ... \Rightarrow x_k \Rightarrow z$ (k + 1 kroków).

Język generowany przez gramatykę formalną $G = (\Sigma, V, S, P)$ to $\{w \in \Sigma^* | S \overset{*}{\Rightarrow} w\}$, gdzie w jest pewnym ciągiem symboli terminalnych, a S jest symbolem startowym.

[5] Por. A. Kułacka, *Metodologiczne założenia semantyki komputerowej*, [w:] *Metodologie językoznawstwa*, red. P. Stelmaszczyk, Łódź 2011.

Rozważmy przykłady gramatyk formalnych i ciągów, jakie generują. Niech G_1 = (Σ, *V, S, P*) będzie gramatyką formalną, gdzie Σ = {*Maria, Jan, Smiles, Likes*}, *V* = {*S*, *VP*, *IV, TV*} oraz *P* jest zbiorem następujących reguł wyprowadzania (kreska | oznacza „lub"):

$S \rightarrow Maria\ VP \mid Jan\ VP$;

$VP \rightarrow IV \mid TV\ Maria \mid TV\ Jan$;

$IV \rightarrow Smiles$;

$TV \rightarrow Likes$.

Możemy wygenerować ciąg *Maria Likes Jan* w następujący sposób:

$S \Rightarrow Maria\ VP \Rightarrow Maria\ TV\ Jan \Rightarrow Maria\ Likes\ Jan$.

To wyprowadzenie pokazuje, że *Maria Likes Jan* należy do języka wygenerowanego przez gramatykę formalną G_1. Skonstruuję teraz gramatyki formalne dla przykładów wyrażeń regularnych podanych w poprzednim rozdziale. W opisie języka symbol *w* oznacza dowolny ciąg.

Niech G_2 = (Σ, *V, S, P*) będzie gramatyką formalną, gdzie Σ = {0,1}, *V* = {*S*, *A*} i *P* jest zbiorem następujących reguł wyprowadzania: $S \rightarrow 0A$, $A \rightarrow \varepsilon|1\ A$. Język wygenerowany przez G_2 to {*w*|*w zaczyna się od* 0*, po którym następuje zero lub więcej syboli* 1}.

Niech G_3 = (Σ, *V, S, P*) będzie gramatyką formalną, gdzie Σ = {0,1}, *V* = {*S*} i *P* jest zbiorem złożonym z reguły wyprowadzania: $S \rightarrow 0|1$. Język wygenerowany przez G_3 to {0,1}.

Niech G_4 = (Σ, *V, S, P*) będzie gramatyką formalną, gdzie Σ = {0,1}, *V* = {*S*, *A*} i *P* jest zbiorem następujących reguł wyprowadzania: $S \rightarrow \varepsilon|0|1A$, $A \rightarrow \varepsilon|1\ A$. Język wygenerowany przez G_4 to {*w*|*w jest* 0 *lub ciągiem zera lub więcej syboli* 1}.

Niech G_5 = (Σ, *V, S, P*) będzie gramatyką formalną, gdzie Σ = {0,1}, *V* = {*S*, *A, B*} i *P* jest zbiorem następujących reguł wyprowadzania: $S \rightarrow \varepsilon|0|A$, $A \rightarrow 1B$, $B \rightarrow \varepsilon|0A$. Język wygenerowany przez G_5 to {*w*|*w składa się z zera lub więcej ciągów symboli* 01}.

Jak możemy zauważyć, reguły wyprowadzania dla gramatyk formalnych G_1, G_2, G_3, G_4, G_5 mają jedną z form: $A \rightarrow yB$ lub $A \rightarrow x$, gdzie *A* i *B* są symbolami nieterminalnymi, *x* jest albo symbolem terminalnym, albo ciągiem pustym ε, a *y* jest symbolem terminalnym. Są to jedyne formy reguł wyprowadzania występujące w gramatykach typu 3, które generują języki regularne[6]. Należy zauważyć, że poszerzyłam definicję podaną przez Partee, Ter Meulena, Walla o regułę, dzięki której można wyprowadzić ciąg pusty ε. W przeciwnym razie języki opisane przez wyrażenia regularne i te generowane przez gramatyki formalne typu 3 nie byłyby takie same. Jak wspomniałam, gramatyki wyższych typów również generują języki regularne, na przykład G_6 = (Σ, *V, S, P*), gdzie Σ = {0,1}, *V* = {*S*} i *P* jest zbiorem złożonym z reguły wyprowadzania: $S \rightarrow \varepsilon|01S$, generuje ten sam język co gramatyka formalna G_5, tj. {*w*|*w składa się z zera lub więcej ciągów symboli* 01}, gdzie *w* jest dowolnym ciągiem. Reguła występująca w tej gramatyce nie ma jednej z dwóch form wymaganych dla gramatyki formalnej typu 3.

[6] B. Partee, A. Ter Meulen, R.E. Wall, *op. cit.*, s. 451.

Poniżej podam możliwe przejście z opisu języka za pomocą wyrażenia regularnego do jego gramatyki formalnej. Alfabety symboli terminalnych dla gramatyki i wyrażenia regularnego są równe. Krok podstawowy można zdefiniować w następujący sposób (duże litery oznaczają symbole nieterminalne, małe — terminalne):

Reguła wyprowadzania	Wyrażenie regularne, które opisuje ten sam język co język wygenerowany przez regułę po lewej stronie
$A \rightarrow a$	a
$A \rightarrow \varepsilon \mid aA$	a^*
$A \rightarrow a \mid b$	$a + b$
$A \rightarrow ab$	ab

Załóżmy, że R_1 i R_2 są wyrażeniami regularnymi związanymi ze zbiorem reguł w odpowiedniej formie, typowej dla gramatyk formalnych języków regularnych. Reguła wyprowadzania odpowiednia dla wyrażenia regularnego $R_1 + R_2$ to $B \rightarrow R_1 \mid R_2$, gdzie po prawej stronie reguły $B \rightarrow R_1 \mid R_2$ skopiowaliśmy prawe strony reguł R_1 i R_2. Dla wyrażeń regularnych R_1^* i R_1R_2 najprawdopodobniej będzie należało wprowadzić więcej symboli, by zachować odpowiednią formę reguły wyprowadzania, tak jak to zrobiłam w konstrukcji gramatyki G_5. Odpowiednie schematy reguł wyprowadzania dla tych ostatnich wyrażeń regularnych to $B \rightarrow \varepsilon \mid R_1B$ i $B \rightarrow R_1R_2$.

By udowodnić, że język jest nieregularny, wykorzystuje się twierdzenie zwane lematem o pompowaniu dla języków regularnych. Jest to twierdzenie, które zastosowali Partee, Ter Meulen, Wall. w swoim pomyśle na dowód, że język angielski nie jest językiem regularnym. Należy zwrócić uwagę, że wykazanie, iż jeden lub kilka języków są językami nieregularnymi, nie dowodzi, że wszystkie są nieregularne. Wskazuje to tylko na fakt, że jeśli chcemy zdefiniować ogólną gramatykę, która będzie miała zastosowanie do opisu wszystkich języków, to gramatyka typu 3 nie spełnia tego wymogu.

4. Lemat o pompowaniu dla języków regularnych

W tym rozdziale przedstawię lemat o pompowaniu dla języków regularnych bez podawania jego dowodu[7], przedstawię tylko jego zastosowanie w dowodach na to, że dany język jest nieregularny.

Twierdzenie to brzmi: Niech L będzie językiem regularnym. Istnieje taka liczba naturalna *n*, że każdy ciąg *w* języka L o długości (liczbie symboli terminal-

[7] Dowód można znaleźć między innymi w: J.E. Hopcroft, R. Motwani, J.D. Ullman, *op. cit.*, s. 126 n.; M. Sipser, *op. cit.*, s. 78 n.

nych) równej lub większej od n (piszemy $|w| \geq n$) możemy podzielić na trzy ciągi x, y, z, a więc, spełniające następujące warunki:

(a) $y \neq \varepsilon$;

(b) $|xy| \leq n$;

(c) Dla każdego $k \geq 0$, ciąg xy^kz również należy do L.

Należy dodać, że twierdzenie to stosuje się tylko do języków nieskończonych. Język regularny opisany wyrażeniem regularnym **0 + 1** to {0,1}. Jeśli będziemy chcieli zastosować twierdzenie, pod zmienną n możemy podstawić tylko 1. Oba ciągi są o długości równej lub większej od 1. Jeśli je podzielimy na ciągi x, y, z, a więc $w = xyz$, gdzie x i z są ciągami pustymi i y to 0 lub 1, to warunki (a) i (b) będą spełnione; $y \neq \varepsilon$ i $|xy| \leq 1$, gdyż jest to długość ciągu y. Jednakże warunek (c) nie może być spełniony na przykład dla $k \geq 2$, gdyż $y^2 \notin$ L.

Rozważmy teraz nieskończony język regularny opisany przez wyrażenie regularne **01*.** Istnieje taka liczba n, $n = 2$, że każdy ciąg w spełniający warunek $|w| \geq 2$ może zostać podzielony na ciągi x, y, z, a więc $w = xyz$, gdzie $x = 0$, $y = 1$, oraz $z = 1^p$, $p \geq 0$, zastępuje resztę ciągu w. Warunki (a) i (b) są spełnione; $y \neq \varepsilon$ oraz $|xy| \leq 2$. Warunek (c) jest również spełniony dla każdego $k \geq 0$, ciąg 01^k1^p jest także w L.

Zaprezentuję teraz procedurę dowodzenia, że dany język jest nieregularny i każdy punkt tej procedury zilustruję przykładem. Udowodnię, że język $L_1 = \{0^l1^l | l \geq 1\}$ nie jest regularny.

A. Wybieramy język, o którym chcemy powiedzieć, że jest nieregularny.

AA. Wybraliśmy język $L_1 = \{0^l1^l | l \geq 1\}$.

B. Wybieramy dowolną liczbę naturalną n.

C. Wybieramy ciąg języka L_1, w, taki że $|w| \geq n$.

CC. Niech ciągiem tym będzie $w = 0^n1^n$, $|w| = 2n$.

D. Podzielimy w na ciągi x, y, z, takie że warunki (a) i (b) lematu o pompowaniu będą spełnione.

DD. By warunek (b) był spełniony, ciągi x i y muszą zawierać tylko wielokrotności symbolu 0. Niech $x = 0^{p_1}$, $y = 0^{p_2}$, $z = 0^{p_3}1^n$; $p_2 \geq 1$, by warunek (a) był spełniony i $p_1 + p_2 \leq n$, by warunek (b) był spełniony oraz $p_1 + p_2 + p_3 = n$, gdzie $p_3 \geq 0$.

E.Wybieramy liczbę k, taką że xy^kz nie jest w języku L.

EE. Niech $k = 2$. Wtedy $w_1 = 0^{p_1}\ 0^{2p_2}\ 0^{p_3}1^n$ nie jest w L_1, ponieważ $p_1 + 2p_2 + p_3 = n + p_2 \neq n$, a $p_2 \geq 1$.

Niektóre operacje na językach regularnych dają wynik, który również jest językiem regularnym. Dowody, które wykraczają poza zawartość niniejszego artykułu, są umieszczone u Hopcrofta, Motwaniego, Ullmana oraz Sipsera[8]. Do mojego dowodu potrzebne są fakty: (1) homomorfizm (podstawienie za ciągi symboli jednego języka symboli drugiego) języka regularnego jest również języ-

[8] J.E. Hopcroft, R. Motwani, J.D. Ullman, *op. cit.*, s. 131 n.; M. Sipser, *op. cit.*, s. 58 n.

kiem regularnym i (2) część wspólna dwóch języków regularnych jest językiem regularnym.

Po prezentacji wiedzy koniecznej do zrozumienia poniższego dowodu powstałego na podstawie pomysłów Chomskiego i Partee, Ter Meulena, Walla możemy przejść do tej nowej, poprawionej wersji. Oryginalne dowody można znaleźć w pozycjach Chomskiego[9] i Partee, Ter Meulena, Walla[10].

5. Dowód

Skonstruujemy gramatykę formalną fragmentu języka angielskiego $G_7 = (\Sigma, V, S, P)$, gdzie $\Sigma = \{if, grass, is, green, then\}$, $V = \{S, A\}$, a P jest zbiorem złożonym z dwóch reguł wyprowadzania: $\{S \rightarrow if\ S\ then\ A\ |if\ A\ then\ A, A \rightarrow grass\ is\ green\}$. Zauważmy, że do zbioru symboli terminalnych włączyłam spację, by zachować klarowność znaczenia słów języka. Możemy wygenerować następujące ciągi, które stanowią gramatyczne zdania języka angielskiego, pomijając oczywiście możliwość ich interpretacji:

(a) *if grass is green then grass is green*;

(b) *if if grass is green then grass is green then grass is green*;

(c) *if if if grass is green then grass is green then grass is green then grass is green* itp.

Zdefiniuję teraz homomorfizm h:{*if*, *grass is green*, *then grass is green*}$\rightarrow$ {a, b, c} w następujący sposób:

h (*if*) = a, h (*grass is green*) = b, h (*then grass is green*) = c.

Z poprzedniego rozdziału wiemy, że własność regularności języka zachowana jest w homomorfizmie. Dlatego też, jeśli język wygenerowany przez G_7 jest regularny i ciągi otrzymane jako wartości homomorfizmu h mają formę a^nbc^n, $n \geq 1$, to regularny jest również język $\{a^nbc^n|n \geq 1\}$. Kontrapozycją powyższego twierdzenia jest: jeśli język $\{a^nbc^n|n \geq 1\}$ nie jest regularny, nieregularny jest też język $L(G_7)$. Korzystając z procedury omówionej w powyższym rozdziale, udowodnię, że język $\{a^nbc^n|n \geq 1\}$ nie jest regularny. Po zastosowaniu zasady wnioskowania *modus ponens* dowiodę, że nieregularny jest też język $L(G_7)$.

A. Wybieram język: $L_2 = \{a^nbc^n|n \geq 1\}$.

B. Wybieram dowolną liczbę naturalną n.

C. Niech ciąg wybrany to $w = a^nbc^n$, $|w| = 2n + 1$.

D. By warunek (b) był spełniony, ciągi x i y zawierają tylko symbole a. Niech $x = a^{p_1}$, $y = a^{p_2}$, $z = a^{p_3}bc^n$; $p_2 \geq 1$, by warunek (a) był spełniony, $p_1 + p_2 \leq n$ tak, by warunek (b) był spełniony oraz $p_1 + p_2 + p_3 = n$, gdzie $p_3 \geq 0$.

[9] N. Chomsky, *Three models for the description of language*..., s. 113–124; *idem*, *Syntactic Structures*...; *idem*, *Introduction to the formal analysis of natural languages*...

[10] B. Partee, A. Ter Meulen, R.E. Wall, *op. cit*.

E. Niech $k = 2$. Wtedy $w_1 = a^{p_1}a^{2p_2}a^{p_3}b\,c^n$ nie należy do języka L_2, gdyż $p_1 + 2p_2 + p_3 = n + p_2 \neq n$, jako że $p_2 \geq 1$.

Pokażę teraz, że język {*w*|*w* = *if***grass is green* (*then grass is green*)*} jest regularny. Nie jest ważne, czy ciągi należące do tego języka są wyrażeniami języka angielskiego czy innego języka naturalnego, czy też żadnego. Wyrazy należące do tego języka da się opisać wyrażeniem regularnym *if***grass is green* (*then grass is green*)*. Możemy również skonstruować gramatykę formalną typu 3, która generować będzie ciągi należące do tego języka: $G_8 = (\Sigma, V, S, P)$, gdzie Σ = {*if*, *grass*, *is*, *green*, *then*}, V = {*S*, *A*, *B*, *C*}, i *P* jest zbiorem dwóch reguł wyprowadzania: {$S \rightarrow$ *if S*|*grass A*, $A \rightarrow$ *is B*, $B \rightarrow$ *green C*, $C \rightarrow \varepsilon$|*then A*}.

W ostatniej części dowodu wykorzystam fakt, że część wspólna (zwana też przekrojem) dwóch języków regularnych jest również językiem regularnym. Przetnijmy język angielski z językiem {*if***grass is green* (*then grass is green*)*}. Częścią wspólną tych zbiorów jest zbiór if^n *grass is green* (*then grass is green*)$^n|n \geq 1$}. Też ostatni zbiór jest nieregularny, jak wykazałam to powyżej, oraz regularnym jest język {*if***grass is green* (*then grass is green*)*}, a więc język angielski jest nieregularny dzięki kontrapozycji: jeśli przekrój dwóch zbiorów jest nieregularny, to co najmniej jeden z języków musi być nieregularny. Język {*if***grass is green* (*then grass is green*)*} jest regularny, a więc nieregularny jest język angielski.

W powyższym dowodzie wykorzystałam jeden z pomysłów Chomskiego na konstrukcje językowe, które pokazują, że język naturalny nie może być generowany przez gramatykę typu 3. Należy więc tworzyć gramatyki o większej mocy, a i to może nie wystarczyć ze względu na cechy, jakie język naturalny posiada[11]. Inne konstrukcje, które mogą służyć temu samemu celowi, czyli wykazaniu, że język angielski nie jest językiem regularnym, to „Either S or S", „The man who said that S is arriving today", gdzie S to zdanie, oraz zdania zawierające konstrukcje osadzone, takie jak angielskie zdanie: (the rat (the cat (the dog chased) killed) ate the malt)[12].

6. Język polski

W języku polskim mamy podobne schematy zdań jak te opisane dla języka angielskiego. W następujących schematach zdaniowych S oznacza zdanie osadzone w konstrukcji: „Jeżeli S, to S", a także „albo S, albo S". Zdanie „Mężczyzna, który powiedział, że S przyjeżdża dzisiaj" jest polskim odpowiednikiem zdania „The man who said that S is arriving today". Dlatego też dla języka polskiego można przeprowadzić dowody podobne do tego, jaki przedstawiłam dla języka angielskiego. Konstrukcje osadzone, takie jak (the

[11] Zob. A. Kułacka, *Syntax of a Montague Grammar*...

[12] N. Chomsky, *Introduction to the formal analysis of natural languages*..., s. 286.

rat (the cat (the dog chased) killed) ate the malt)[13], nie są obecne w języku polskim. Innym schematem zdaniowym stanowiącym podstawę podobnych dowodów jest angielskie „neither S nor S" i jego polski odpowiednik „ani S, ani S".

7. Dalsze badania

Należy brać pod uwagę dwa fakty: (1) dowiedzenie twierdzenia, że niektóre języki naturalne są nieregularne nie uprawnia nas do stwierdzenia, że wszystkie języki naturalne są takie, jak zakłada się w literaturze[14]; (2) udowodniono, że słuchacz przetwarza język tak, jak przetwarzałby język regularny[15]. Fakt (2) można uzasadnić ograniczeniami pamięci krótkiej, która nie może przechowywać zbyt dużo informacji[16]. Te ograniczenia mogą być do pewnego stopnia pominięte w przypadku komputerów i języków wygenerowanych przez implementację gramatyk.

Interesujące byłoby też przyjrzenie się językom spoza indoeuropejskiej rodziny języków, żeby zobaczyć, czy na ich temat można wyciągnąć podobne wnioski. Innym możliwym projektem dalszej pracy byłoby dookreślenie cech charakterystycznych tych konstrukcji językowych, które pozwalają na dowodzenie, że dany język jest nieregularny, by umożliwić ich wyszukiwanie w innych językach.

Miesza się również język, który człowiek produkuje i rozumie, z teoretycznie możliwym językiem naturalnym posiadającym abstrakcyjnego użytkownika języka. Jasne jest, że ten pierwszy jest językiem regularnym, ponieważ jest językiem skończonym, podczas gdy ten teoretycznie możliwy język jest nieskończony, a przez to możliwie nieregularny. W tym artykule rozważałam język teoretycznie możliwy. Innym pytaniem jest, czy generujemy wyrażenia językowe, implementując mentalną gramatykę formalną, czy odtwarzamy zbitki wyrazów i tylko okazjonalnie produkujemy wyrażanie zupełnie nowe, do tworzenia którego wykorzystujemy ową gramatykę.

Bibliografia

Chomsky N., *Introduction to the Formal Analysis of Natural Languages*, [w:] *Handbook of Mathematical Psychology*, t. 2, red. R.D. Luce, R.R. Bush, E. Galanter, New York 1967.

[13] *Ibidem*.

[14] Zob. *ibidem* oraz G. Gazdar, Ch. Mellish, *Natural Languages Processing in PROLOG*, Reading 1989, s. 135.

[15] *Ibidem*.

[16] Zob. A. Kułacka, *The Necessity of the Menzerath-Altmann Law*, „Anglica Wratislaviensia" 47, 2009, s. 55–60.

Chomsky N., *Syntactic Structures*, Berlin 1957 (reedycja 2002).
Chomsky N., *Three models for the description of language*, „IRE Translations on Information Theory" 2, 1956, nr 3, s. 113–124.
Gazdar G., Mellish Ch., *Natural Languages Processing in PROLOG*, Reading 1989.
Hopcroft J.E., Motwani R., Ullman J.D., *Introduction to Automata Theory, Languages, and Computation*, Boston 2001.
Kułacka A., *Intensional logic for a montague grammar*, „LingVaria" 6 (2011) | 2 (2012).
Kułacka A., *Metodologiczne założenia semantyki komputerowej*, [w:] *Metodologie językoznawstwa*, red. P. Stelmaszczyk, Łódź 2011.
Kułacka A., *The necessity of the Menzerath-Altmann Law*, „Anglica Wratislaviensia" 47, 2009, s. 55–60.
Kułacka A., *Syntax of a montague grammar*, „LingVaria" 6 (2011) | 1 (11), s. 9–23.
Partee B., Ter Meulen A., Wall R.E., *Mathematical Methods in Linguistics*, London 1990.
Sipser M., *Introduction to the Theory of Computation*, Boston 1997.