



NATURALNY SYSTEM OPERACYJNY

Gramatyka regularno-pojęciowa

Waldemar Wietrzykowski

Instytut Inteligencji Stosowanej, marzec 2019

Streszczenie Gramatyka regularno-pojęciowa, będąca pełnym systemem syntaktyczno-semantycznym, stanowi połączenie gramatyki regularnej z wbudowanym mechanizmem automatycznego („podświadomego”) nabywania gramatyki z szeregów terminalnych oraz gramatyki pojęciowej z wbudowanym mechanizmem interaktywnego („świadomego”) pozyskiwania gramatyki od wiarygodnego nauczyciela lub jego wiedzy z wbudowanym mechanizmem automatycznego („podświadomego”) wnioskowania w ramach pozyskanej gramatyki pojęciowej. Współpraca gramatyki regularnej z gramatyką pojęciową polega na tym, że nowo pozyskane reguły pojęciowe stają się zarodkami osadzania się struktur terminalnych w drodze ich konsolidowania w ramach gramatyki regularnej, powodując bardziej uniwersalny charakter tych reguł i dostosowanie do wejściowych szeregów terminalnych. Gramatyka regularno-pojęciowa pretenduje więc do syntaktyczno-semantycznego modelu naturalnego systemu operacyjnego.

Gramatyka według Chomsky’ego

Ze standardowej teorii Noama Chomsky’ego zawartej w *"Aspects of the Theory of Syntax"* i *"Syntactic Structures"* wynika, że zdania języka naturalnego, które należy traktować jako hierarchiczne struktury danych, składają się z szeregów *symboli terminalnych* (zwanych też *terminalami*). Struktury te są hierarchicznym złożeniem podstruktur, gdzie węzłami (znacznikami frazowymi) są *symbole nieterminalne* (zwane też *nieterminalami*), a liśćmi *terminale* tego zdania. Tą hierarchiczną strukturę zdania można porównać do wielopoziomowego zagnieżdżania mniejszych podstruktur zdania w większych na wzór *nawiasowania*. Struktura ta określa zarówno zasady poprawnego łączenia terminali jak i gramatyczne znaczenie powiązania ich w grupy w zdaniu.

Zdaniem Chomsky’ego, wrodzona znajomość gramatyki języka naturalnego umożliwia uczenie się języka naturalnego.

Przeciwna koncepcja głosi, że język naturalny jest umiejętnością nabytą w procesie socjalizacji, co otwiera drogę na maszynowe, automatyczne jego nabywanie.

Maszynowe nabywanie i przetwarzanie języka naturalnego wymaga jego sformalizowania. Według Chomsky’ego formalizacja języka polega na wskazaniu dla danego języka takich elementów jak: słownik symboli terminalnych, słownik symboli nieterminalnych oraz zbiór reguł gramatyki zwany zbiorem produkcji. Reguły te umożliwiają maszynowe przetwarzanie formuł zdaniowych (szeregów symboli terminalnych i nieterminalnych) oparte zarówno na wnioskowaniu dedukcyjnym (od ogółu do szczegółu) jak i wnioskowaniu indukcyjnym (od szczegółu do ogółu).

Założmy, że mamy określoną formułę zdaniową. Korzystając z wnioskowania dedukcyjnego możemy przetworzyć ją na inną formułę zdaniową lub zdanie poprzez zamianę symbolu nieterminalnego, pasującego do lewej strony pewnej produkcji, na prawą stronę tej produkcji. Stosowanie produk-

cji do zamiany symboli nieterminalnych zostaje zakończone, kiedy nie zostaje odszukana żadna produkcja pozwalająca na dalszą zamianę formuły lub kiedy formuła zdaniowa przekształci się w zdanie, złożone z samych symboli terminalnych. Końcową formułę zdaniową nazywamy *formułą szczegółową* lub *szczegółem*. Proces ten nazywamy procesem *generacji zdań* lub *generacji szczegółu*.

Przy korzystaniu z wnioskowania indukcyjnego możemy przetworzyć formułę zdaniową lub zdanie na inną formułę poprzez zamianę symbolu lub szeregu symboli pasujących do prawej strony pewnej produkcji, na lewą stronę tej produkcji. Stosowanie produkcji do zamiany symboli zawartych w formule zdaniowej lub zdaniu zostaje zakończone, kiedy nie zostaje odszukana żadna produkcja pozwalająca na dalszą zamianę. Końcową formułę zdaniową nazywamy *formułą ogólną* lub *ogółem*. Proces ten nazywamy procesem *generacji ogółu*.

Wnioskowanie dedukcyjne i indukcyjne wykorzystujące reguły gramatyki nazywamy *wnioskowaniem gramatycznych* (należy odróżnić od *wnioskowania gramatyk*).

Praktyczna formalizacja języka naturalnego polega na doborze odpowiednich produkcji gramatycznych, przy pomocy których można przeprowadzać przetwarzanie języka przy pomocy wnioskowania gramatycznego.

Dobór produkcji gramatycznych dla danego języka może być przeprowadzony manualnie przez doświadczonego lingwistę. Dobór ten można też przeprowadzać maszynowo i nosi on wówczas nazwę *indukcji gramatyk* lub *wnioskowania gramatyk* lub też *maszynowym uczeniem gramatyk*. Celem indukcji gramatyk jest automatyczne znalezienie reguł gramatycznych (produkcji) określających język zastosowany w zdaniach trenujących.

Sformalizowanie języka naturalnego jest niezwykle trudne ze względu na kontekstowość tego języka. Oznacza to, że zastosowana produkcja gramatyki języka naturalnego może mieć postać zależną od kontekstu tzn. zależeć od symboli otaczający przetwarzany symbol w formule zdaniowej.

Gramatyka regularna

Najprostszą gramatyką jest gramatyka regularna, którą Chomsky zaklasyfikował jako gramatyka typu 4, określając ją jako skończenie stanowa. Pomimo tego, że gramatyka ta nie została przez niego uznana jako potencjalny model języka naturalnego, to jednak posiada szczególną cechę, jaką jest równoważność z najprostszym z możliwych automatów. Automat ten przebiega tylko po terminalach, a wybór kolejnego terminala zależy wyłącznie od terminala poprzedzającego. Powoduje to, że automat buduje strukturę regularną (liniową) opartą na następstwie możliwych terminali. Ponadto automat taki, zwany skończenie stanowym, umożliwia automatyczną i bardzo prostą indukcję gramatyki regularnej wprost z szeregów terminalnych. Mechanizm indukcji gramatyki regularnej opracowany przy użyciu automatu skończonego został opisany jako rachunek sekwentowy przedstawiony w pracach: „*Teoria świadomej maszyny*”, „*Wystarczalność rachunku sekwentowego. Rachunek Tautologiczny*” oraz „*Implementacja rachunku sekwentowego*”.

W rachunku sekwentowym reguły gramatyczne są wyszukiwane automatycznie na podstawie pozytywnych szeregów trenujących w następujący sposób:

1. najpierw szereg terminali zamieniany jest na szereg par typu {terminal poprzedzający, terminal następny} zwany szeregiem reguł elementarnych¹.

¹ Implementacja rachunku sekwentowego, str 2

2. następnie grupowane są lewostronnie reguły elementarne w ten sposób, że wszystkie reguły elementarne posiadające identyczny terminal po lewej stronie należą do jednej grupy. Po grupowaniu mogą pozostać samodzielne reguły elementarne, nie związane z żadną grupą.
3. otrzymane grupy reguł elementarnych są następnie konsolidowane (scalane) w ten sposób, że każda grupa reguł elementarnych jest zamieniana w jedną regułę złożoną (zwana też konstrukcją) o wspólnym terminalu po lewej stronie i listą alternatywnych terminali po prawej stronie jakie występują w grupie².
4. później grupowane są prawostronnie otrzymane reguły złożone i pozostałe reguły elementarne w ten sposób, że jeżeli lista terminali lub terminal po prawej stronie pewnej reguły ma co najmniej jeden wspólny terminal z listą terminali lub terminalem należących do prawej strony innej reguły to reguły te należą do jednej grupy.³
5. otrzymane grupy reguł są powtórnie konsolidowane (scalane) w ten sposób, że każda grupa reguł zamieniana jest w jedną regułę złożoną (zwana też konstrukcją) z listą alternatywnych terminali po lewej i listą alternatywnych terminali po prawej stronie jakie występują w grupie.⁴

Końcowy wynik konsolidacji ma następujące właściwości:

1. konsolidacja otrzymana z pewnego szeregu terminali może posłużyć do generacji wielu innych szeregów terminali nawet nieskończony ich zbiór, które dają tą samą konsolidację. Procedura generacji szeregu terminali z konsolidacji jest opisana w 12 prawie rachunku sekwentowego⁵.
2. konsolidacja jest rekursywna (może generować nieskończenie wiele szeregów terminali).
3. może być konsolidowana z innym szeregiem terminali. Jeżeli szereg należy do tej samej gramatyki, konsolidacja wypadkowa nie zmienia się. Konsolidacja może się też uzupełnić, jeżeli poprzedni szereg nie wyczerpywał wszystkich możliwości tej gramatyki.

Przykład 1

Sprawdzenie działania rachunku sekwentowego przeprowadzono w pracy „Teoria świadomej maszyny”⁶ w temacie „Reguły scalania predykatów i produkcji”. W kolejnych przykładach w tej pracy uzyskano następujące wyniki:

W przykładzie 1: z ciągu liczb 3,5,0,6,4,2,1,0,3,6,4,7,1,0,6,2,1,0,6,1,0,6,7,2,1,5,3,3,6,1,0 uzyskano konsolidację (3,5,0,1) (5,6,3,0),(6,4,2,7) (4,2,1,7)⁷.

² Teoria świadomej maszyny, str 16; Implementacja rachunku sekwentowego, str 3

³ Teoria świadomej maszyny, str 16; Implementacja rachunku sekwentowego, str 5

⁴ Teoria świadomej maszyny, str 16; Implementacja rachunku sekwentowego, str 5

⁵ Implementacja rachunku sekwentowego, str 13

⁶ Teoria Świadomej Maszyny; str 16 - 19

W przykładzie 2: z ciągu liczb 3,5,0,6,4,2,1,0,3,6,4,7,1,0,6,2,1,0,6,1,0 uzyskano konsolidację **(3, 0)** (5,6,3),(6,4,2,7) (4,2,1,7),(5,1) (0).

W przykładzie 3: strukturę **(3, 0)** (5,6,3),(6,4,2,7) (4,2,1,7),(5,1) (0) połączono z ciągiem liczb 6,7,2,1,5,3,3,6,1,0 i uzyskano konsolidację **(3,1,0,5)** (5,6,3,0),(6,4,2,7) (4,2,1,7).

W przykładzie 4: strukturę **(3, 0)** (5,6,3),(6,4,2,7) (4,2,1,7),(5,1) (0) połączono z ciągiem 0 5 6 3 i otrzymano konsolidację **(1,5,4,7,2,3,0,6)** (0,7,2,4,1,5,6,3).

W przykładzie 5: ciąg liczb 3,5,0,6,4,2,1,0,3,6,4,7,1,0,6,2,1,0,6,1,0,6,7,2,1,5,3,3,6,1,0 zakłócono ciągiem liczb 11,13,8,14,12,10,9,8,11,14,12,15,9,8,14,10,9,8,14,9,8 i otrzymano konsolidację **(0,3,5,1)** (3,6,5,0), **(4,7,2,6)** (7,2,4,1) oraz dodatkowe struktury zakłócające **(11,8)** (13,11,14), **(12,15,10,14)** (15,10,12,9), **(13,9)** 8.

Przykład 2

Sprawdzenie działania rachunku sekwentowego przeprowadzono również w pracy *“Implementacja rachunku sekwentowego”*⁸.

Operację dodawania $3566 + 4972 = 8538$ przedstawiono w postaci szeregu terminali 0, 5, 3, 7, 4, 3, 7, 1, 7, 2, 1, 1, 2, 4, 0⁹ i otrzymano konsolidację **0 5,(3,1,7,2)** (7,2,1,4),(5,4) (3,0).

Operację dodawania $4632 + 965 = 5597$ przedstawiono w postaci szeregu 6, 0, 6, 5, 5, 0, 6, 6, 6, 3, 4, 0, 5, 0 i otrzymano konsolidację **3 4,(4,0,6,5)** (3,6,5,0).

Operację dodawania $5828 + 2722 = 8550$ przedstawiono w postaci szeregu 0, 6, 5, 0, 0, 6, 5, 3, 4, 3, 1, 2, 1, 4, 0 i otrzymano konsolidację **6 5,(2,3,1)** (2,1,4),(0,5,4) (3,6,0).

Operację dodawania $6732 + 7644 = 14376$ przedstawiono w postaci szeregu 0, 0, 3, 7, 2, 4, 3, 2, 2, 1, 2, 7, 7, 4, 0 i otrzymano konsolidację **(1,3,2,7)** (1,7,2,4),(0,4) (3,0).

Ciąg kolejnych operacji dodawania ($6453 + 4449 = 10902$), ($1055 + 1237 = 2292$), ($4084 + 1686 = 5770$) przedstawiono w postaci szeregu (3, 4, 5, 0, 5, 3, 2, 4, 3, 4, 0, 5, 3, 4, 0), (3, 1, 7, 1, 7, 4, 6, 6, 0, 0, 3, 4, 0), (0, 6, 3, 4, 3, 1, 1, 7, 1, 7, 7, 1, 4, 0) i otrzymano konsolidację **(7,2,3,1)** (7,1,2,4),(6,0,5,4) (6,3,5,0).

Ciąg kolejnych innych operacji dodawania ($8353 + 1334 = 9687$), ($1046 + 9885 = 10931$), ($1122 + 2783 = 3905$) przedstawiono w postaci szeregu (5, 6, 6, 0, 6, 3, 4, 5, 6, 0, 6, 0, 0, 5, 0), (6, 5, 3, 2, 7, 4,

⁷ kolejność alternatywnych terminali w nawiasach poprzedzielanych przecinkami nie ma istotnego znaczenia. Część wytłuszczona oznacza predykat a część niewytłuszczona produkcję.

⁸ *Implementacja rachunku sekwentowego*, str 13 - 14

⁹ dla wyjaśnienia: każdy terminal z tego szeregu jest liczbą trzy bitową. Pierwszy bit jest kolejnym bitem z pierwszego składnika dodawania, drugi bit jest kolejnym bitem z drugiego składnika dodawania a trzeci bit jest kolejnym bitem z sumy

0, 6, 0, 6, 3, 4, 0, 6, 0), (6, 3, 2, 2, 2, 1, 7, 2, 4, 6, 5, 6, 0) i otrzymano taką samą, jak poprzednio konsolidację **(1,7,3,2)** (1,7,2,4),(4,0,5,6) (5,3,6,0)¹⁰.

Wniosek z przykładu 2

W przykładzie 2 konsolidacja jest strukturą gramatyczną (gramatyką) otrzymaną z szeregu terminalnego lub ciągu szeregów terminalnych przedstawiających operację dodawania dowolnych liczb naturalnych. Charakterystyczne jest to, że każda konsolidacja operacji dodawania dąży do tej samej granicznej struktury gramatycznej **(1,7,3,2)** (1,7,2,4),(4,0,5,6) (5,3,6,0), niezależnie od długości szeregu terminalnego lub ilości ciągów terminalnych.

Wskazaną strukturę można przedstawić w nieco innej równoważnej postaci **(3,1,2,7)** (4,1,2,7), **(6,5,0,4)** (0,5,6,3), ponieważ w zapisie sekwentowym w obrębie nawiasów występują alternatywy przedzielone znakiem „,” i zmiana położenia poszczególnych elementów w obrębie nawiasów nie ma wpływu na strukturę. Jeżeli predykat **(6,5,0,4)** oznaczmy jako stan 0, a predykat **(3,1,2,7)** oznaczmy jako stan 1, to wspomnianą strukturę gramatyczną można przedstawić też następująco:

Stan

0: **(6,5,0,4)** (0,5,6,3)
 0 -> 0; przejdź do stanu 0
 5 -> 0; przejdź do stanu 0
 6 -> 0; przejdź do stanu 0
 3 -> 1 ; przejdź do stanu 1

1: **(3,1,2,7)** (4,1,2,7)
 4 -> 0; przejdź do stanu 0
 1 -> 1; przejdź do stanu 1
 2 -> 1; przejdź do stanu 1
 7 -> 1; przejdź do stanu 1

Po zamianie na zapis binarny mamy:

0:
 000 -> 0 (*)
 101 -> 0
 011 -> 0
 110 -> 1

1:
 001 -> 0
 100 -> 1
 010 -> 1
 111 -> 1

(*) – zapis binarny od lewej do prawej. Od lewej mniej - znaczący bit. [012]

¹⁰ pozycja alternatywnym terminali poprzedzielanych przecinkami w konsolidacji nie ma istotnego znaczenia.

otrzymujemy listę rozkazów (czyli program) jednokierunkowej maszyny Turinga dla dodawania dwóch liczb całkowitych. Wynika z tego, że rachunek sekwentowy pozwala na samo-programowanie maszyny Turinga na podstawie danych wejściowych.

Przykład 3

Sprawdzenie działania rachunku sekwentowego przeprowadzono także w pracy „Wystarczalność rachunku sekwentowego. Rachunek Tautologiczny”¹¹. Na podstawie kolejnych przykładów zamieszczonych w tej pracy przeprowadzono następujące możliwości wykorzystania rachunku sekwentowego:

a/ Do pamięci krótkotrwałej (ang. *short-term memory*, STM) został przesłany szereg terminalny „p, q, r, s, q, s, r, t, u, w”. Po konsolidacji w STM uzyskano strukturę gramatyczną „(p, q, r, s) (q, r, s, t), t u, u w”. Składa się ona ze struktury „(p, q, r, s) (q, r, s, t)” i szeregu terminali „t, u, w”, którego nie da się połączyć z tą strukturą. Konsolidacja ta została zapamiętana w pamięci średniotrwałej (ang. *medium-term memory*, MTM). Wykonano tutaj następujące operacje:

STM = p, q, r, s, q, s, r, t, u, w
 consolid(STM) (p, q, r, s) (q, r, s, t), t u, u w
 MTM = STM

Następnie do STM przesłano szereg terminalny „a, z, u, z, a, y”. Po konsolidacji zawartości STM i MTM uzyskano w STM strukturę „(p,q,r,s) (q,r,s,t), (z,t) (u,a), a z, u z, a y, u w”. W wyniku tej konsolidacji powstała dodatkowa struktura „(z,t) (u,a)” oraz szereg „a z, u z, a y, u w”. Konsolidację tą zapamiętano w MTM. Wykonano tutaj następujące operacje:

STM = a, z, u, z, a, y
 consolid(STM, MTM) (p,q,r,s) (q,r,s,t), (z,t) (u,a), a z, u z, a y, u w
 MTM = STM

Z kolei do STM przesłano terminal „a” będący w kontekście terminala „u”. W STM została utworzona struktura „u a”. Po konsolidacji zawartości STM i MTM uzyskano w STM strukturę „(p,q,r,s) (q,r,s,t), (z,t,u) (u,a), a z, u z, a y, u w”. Konsolidację tą zapamiętano w MTM. Wykonano tutaj następujące operacje:

STM = u a
 consolid(STM, MTM) (p,q,r,s) (q,r,s,t), (z,t,u) (u,a), a z, u z, a y, u w
 MTM = STM

Otrzymaną zawartość MTM rozdysponowano. W wyniku tego strukturę „(z,t,u) (u,a)” znajdującą się w MTM skopiowano do STM i wygenerowano z niej szereg terminali „t, u, u”. Największą strukturę „(p,q,r,s) (q,r,s,t)” przeniesiono do pamięci długotrwałej (ang. *long-term memory*, LTM), a pozostałe struktury elementarne „a z, u z, a y, u w” usunięto w procesie zapominania. Wykonano tutaj operacje.

¹¹ *Wystarczalność rachunku sekwentowego. Rachunek Tautologiczny*, str. 11 - 13

MTM $(p,q,r,s) (q,r,s,t), (z,t,u) (u,a), a z, u z, a y, u w$

STM = $(z,t,u) (u,a)$

STM $\Rightarrow t, u, u$

MTM $(p,q,r,s) (q,r,s,t), (z,t,u) (u,a), a z, u z, a y, u w$

LTM = $(p,q,r,s) (q,r,s,t)$

MTM $(z,t,u) (u,a), a z, u z, a y, u w$

MTM $(z,t,u) (u,a)$

Wniosek z przykładu 3

Przy pomocy rachunku sekwentowego można zarządzać pamięcią krótkotrwałą (ang. short-term memory, STM), pamięcią średniotrwałą (ang. medium-term memory, MTM) oraz pamięcią długotrwałą (ang. long-term memory, LTM), w szczególności prowadzić szeregi terminali do STM, przetwarzać szeregi terminali w STM (konsolidacja), przetwarzać zawartości MTM i LTM w STM, jak również generować szeregi terminali z STM.

Podsumowanie

Rachunek sekwentowy opisany w pracach “*Wystarczalność rachunku sekwentowego*”, “*Teoria maszyn świadomych. Rachunek Tautologiczny*” i “*Implementacja rachunku sekwentowego*” może zarządzać procesami w sposób automatyczny („podświadomy”).

Ogólnie rachunek sekwentowy jest systemem wydzielania struktur gramatyki regularnej z ciągów szeregów terminalnych.

Gramatyka pojęciowa

Maszynowe nabywanie i przetwarzanie pojęć, podobnie jak języka naturalnego, wymaga jego sformalizowania. Formalizacja ta polega na wskazaniu takich elementów jak: słownik symboli terminalnych reprezentujących nazwy pojęć oraz zbiór nazw i definicji pojęć (zwanych regułami pojęciowymi), będących odpowiednikiem zbioru reguł gramatycznych, zwany zbiorem produkcji. Zbiór tych reguł pojęciowych (nazw i definicji) umożliwi maszynowe przetwarzanie formuł pojęciowych (szeregów symboli terminalnych reprezentujących nazwy pojęć) za pomocą wnioskowania dedukcyjnego (od ogółu do szczegółu) oraz wnioskowania indukcyjnego (od szczegółu do ogółu).

W odróżnieniu od gramatyki języka w gramatyce pojęciowej nie ma symboli nieterminalnych, które w gramatyce języka służą do określenia zasad poprawnego łączenia terminali i gramatycznego opisywania znaczenia powiązania ich w grupy w zdaniu. Poza tym symbole nieterminalne nie są świadomie doświadczane przez użytkownika języka. W gramatyce pojęciowej występują jedynie symbole terminalne, a ich szeregi stanowią definicje innych symboli terminalnych doświadczanych w sposób świadomy dla posiadacza pojęć.

Założmy, że mamy określoną formułę pojęciową złożoną z szeregu nazw pojęć (symboli terminalnych). Korzystając z wnioskowania dedukcyjnego możemy przetworzyć ją na inną formułę pojęciową poprzez zamianę symbolu terminalnego, pasującego do lewej strony pewnej reguły pojęciowej, na prawą stronę tej reguły. Stosowanie reguł pojęciowych do zamiany symboli terminalnych zostaje zakończone, kiedy nie zostaje odzyskana reguła pojęciowa pozwalająca na dalszą zamianę formuły

pojęciowej. Końcową formułę pojęciową nazywamy *formułą szczegółową* lub *szczegółem*. Proces ten nazywamy *uszczegółowieniem pojęciowym*.

Przy korzystaniu z wnioskowania indukcyjnego możemy przetworzyć formułę pojęciową złożoną z szeregu nazw pojęć (symboli terminalnych) na inną formułę pojęciową poprzez zamianę pojęcia lub szeregu pojęć pasujących do prawej strony pewnej reguły pojęciowej, na lewą stronę tej reguły. Stosowanie reguł pojęciowych do zamiany pojęć zawartych w formule pojęciowej zostaje zakończone, kiedy nie zostanie odszukana reguła pojęciowa pozwalająca na dalszą zmianę formuły pojęciowej. Końcową formułę pojęciową nazywamy *formułą ogólną* lub *streszczeniem*.

Przetwarzanie formuł pojęciowych od ogółu do szczegółu nazywamy *analizą pojęciową*, a przetwarzanie formuł pojęciowych od szczegółu do ogółu nazywamy *syntezą pojęciową*.

Gramatyka regularno-pojęciowa

Gramatyka regularno-pojęciowa, będąca pełnym systemem syntaktyczno-semantycznym, stanowi połączenie gramatyki regularnej z wbudowanym mechanizmem automatycznego („podświadomego”) nabywania gramatyki z szeregów terminalnych oraz gramatyki pojęciowej z wbudowanym mechanizmem interaktywnego („świadomego”) pozyskiwania gramatyki od wiarygodnego nauczyciela lub jego wiedzy z wbudowanym mechanizmem automatycznego („podświadomego”) wnioskowania w ramach pozyskanej gramatyki pojęciowej. Współpraca gramatyki regularnej z gramatyką pojęciową polega na tym, że nowo pozyskane reguły pojęciowe stają się zarodkami osadzania się struktur terminalnych w drodze ich konsolidowania w ramach gramatyki regularnej, powodując bardziej uniwersalny charakter tych reguł i dostosowanie do wejściowych szeregów terminalnych. Gramatyka regularno-pojęciowa pretenduje więc do syntaktyczno-semantycznego modelu naturalnego systemu operacyjnego.

Użytkownik pojęć nabywa nowe pojęcia i ich definicje w sposób świadomy od wiarygodnego nauczyciela lub jego wiedzy i przeprowadza ich analizę w oparciu o wnioskowanie w zbiorze reguł pojęciowych (pojęć i ich definicji) już nabytych. Gramatyka pojęciowa zajmuje się znaczeniem symboli terminalnych, dlatego można uważać, że gramatyka regularno-pojęciowa jest połączeniem semantyki i syntaktyki w zakresie gramatyki regularnej. Niedostateczność gramatyki regularnej do modelowania języka naturalnego ma być zrekompensowana obecnością wbudowanego systemu semantycznego i tworzyć równowagę co najmniej bezkontekstową.

Przetwarzanie w gramatyce regularnej opiera się na wbudowanym systemie nabywania gramatyki zawartej w szeregach terminalnych. Mechanizm ten nosi nazwę rachunku sekwentowego i został już opisany w pracach „*Teoria świadomej maszyny*”, „*Wystarczalność rachunku sekwentowego. Rachunek Tautologiczny*” oraz „*Implementacja rachunku sekwentowego*”.

Pozyskiwanie nowych pojęć i ich definicji (reguł pojęciowych) odbywa się przy kontakcie osobnika wyposażonego w system regularno-pojęciowy z innym takim osobnikiem lub jego wiedzą w procesie socjalizacji i ma charakter „świadomy”. Pozyskiwanie to jest wspomagane procesem wnioskowania w nabytym już zbiorze pojęć i ich definicji. Proces wnioskowania ma charakter automatyczny „podświadomy”.

Gramatyka języka naturalnego opracowana przez Chomsky’ego zorientowana była raczej na syntaktyczne właściwości języka naturalnego, natomiast gramatyka regularno-pojęciowa pretenduje do syntaktyczno-semantycznego ujmowania języka naturalnego.

Bibliografia

1. Waldemar Wietrzykowski, *Dwie maszyny*, IIS 2018
2. Waldemar Wietrzykowski, *Uczenie języka naturalnego*, IIS 2018
3. Waldemar Wietrzykowski, *Implementacja rachunku sekwentowego*, DIL 2018
4. Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018
5. Waldemar Wietrzykowski, *Teoria Świadomej Maszyny*, DIL 2018

Copyright © 19.03.2019 IIS