



# UCZENIE JĘZYKA NATURALNEGO

Waldemar Wietrzykowski

**Instytut Inteligencji Stosowanej, listopad 2018**

Streszczenie *Zasada działania każdego komputera, tego wolnego i tego najszybszego, jest zawsze taka sama, jest to maszyna Turinga. Cała kwestia sprowadza się do pytania, jak zaprogramować maszynę Turinga, aby maszyna stała się inteligentna?. Maszyna Turinga jest dostatecznym automatem realizacji różnych modeli gramatyk języka naturalnego. Język naturalny może być w postaci pisanej lub mówionej, ale może być też formą wewnętrznego myślenia, wnioskowania i rozumienia. Można się tu posłużyć przykładem z pewnego artykułu: "maszyna Summit warta 200 milionów dolarów w jednej sekundzie potrafi dokonać obliczeń, jakie zajęłyby człowiekowi 6,3 miliarda lat". A jednak, ten człowiek, od urodzenia do 7 lat potrafi sam opanować sztukę mówienia, myślenia i rozumienia, czego najpotężniejszy i najdroższy komputer nie potrafi.*

Niniejsza praca opisuje, jakie jest powiązanie między rachunkiem sekwentowym, gramatykami Chomsk'owego i językiem naturalnym. Jest to dobra okazja na pokazanie, jak maszyna uczy się języka naturalnego.

## Modele gramatyki języka naturalnego

---

Standardowa teoria Noama Chomsky'ego zawarta w jego pracy "*Aspects of the Theory of Syntax*" (z 1965 roku), powstała na gruncie wcześniejszej jego pracy "*Syntactic Structures*" (z 1957 roku), dotyczy formalnych aspektów języka naturalnego. W niniejszym opracowaniu pominięto filozoficzne rozważania jakie towarzyszą innym opracowaniom poświęconym pracom Chomsky'ego, a skupiono się jedynie na aspektach technicznych.

Najistotniejszym elementem każdej gramatyki jest *zbiór reguł*, który według standardowej teorii Chomsky'ego można podzielić na części: *syntaktyczną*, *semantyczną* i *fonologiczną*.

Do części *syntaktycznej* zaliczamy *reguły bazowe* będące *regułami frazowymi* i *reguły transformacyjne* będące połączeniem *reguł frazowych* i *transformacyjnych*.

Pod względem funkcjonalnym *reguły transformacyjne* przekształcają *struktury frazowe*, pochodzące od *reguł bazowych* i identyfikowane przez *znaczniki frazowe*, w inne *struktury frazowe* poprzez zamianę, usuwanie i inną przemianę *znaczników frazowych*.

*Reguły bazowe* generują *zdania jądrowe* reprezentujące *głęboką strukturę zdań*, wspólną dla wszystkich języków. Natomiast *reguły transformacyjne* przekształcają *głęboką strukturę zdań* do *powierzchniowej struktury zdań*, które generują różne języki. Liczba reguł potrzebnych do generacji *zdań jądrowych* jest względnie mała, dlatego *zdania jądrowe* odnoszą się tylko do zdań prostych, twierdzących oraz oznajmujących w stronie czynnej. Zdania bardziej złożone otrzymuje się ze *struktury powierzchniowej*.

Część *semantyczną* reguł tworzą *reguły semantyczne*. Za pomocą *reguł semantycznych* z *głębokiej struktury zdań* wyprowadza się *reprezentację semantyczną* przedstawiającą znaczenie każdego zdania. Każdemu zdaniu z *głębokiej struktury zdań* jest przypisywana *reprezentacja semantyczna* (lub

kilka alternatywnych *reprezentacji semantycznych* dla zdania wieloznacznego). *Reprezentację semantyczną* tworzy zbiór symboli wyznaczających *uniwersalne pojęcie atomiczne*.

*Część fonologiczną* reguł tworzą *reguły fonologiczne*. Za pomocą reguł fonologicznych z *powierzchnowej struktury* zdań wyprowadza się *reprezentację fonetyczną*. Każdemu zdaniu z *powierzchnowej struktury* zdań jest przypisywana *reprezentacja fonetyczna*. *Reprezentację fonetyczną* tworzy zbiór symboli, wyznaczających pewną *uniwersalną cechę fonetyczną*.

*Symbole semantyczne* i *fonetyczne* są z kolei *elementami pierwotnymi* semantycznego i fonetycznego *poziomu reprezentacji*, które są odpowiednikami *schematów mentalnych* użytkownika rodzimego języka. Są one *psychologicznymi korelatami* w wykonaniu językowym.

*Reprezentacje semantyczne* i *fonetyczne* są przez użytkownika rodzimego języka naturalnego interpretowane w znaczenie i dźwięk dzięki znajomości semantycznej i fonetycznej części *gramatyki uniwersalnej*.

Cechą charakterystyczną *reguł transformacyjnych* jest to, że nie zmieniają one znaczenia zdań, dlatego wyprowadzenie semantyczne odbywa się przed zastosowaniem *reguł transformacyjnych*, a identyfikacja *głębokiej struktury* jest wystarczającym warunkiem semantycznej identyfikacji znaczenia.

Najistotniejszym elementem w przedstawionym wyżej opisie dotyczącym *gramatyki generacyjnej* Chomsky'ego jest to, że *reguły bazowe* generują *zdania jądrowe* reprezentujące *głęboką strukturę zdań*, wspólną dla wszystkich języków, z której za pomocą *reguł semantycznych* są wyprowadzane *reprezentacje semantyczne*, interpretowane następnie przez użytkownika rodzimego języka naturalnego na *znaczenie* za pomocą semantycznej części *gramatyki uniwersalnej*.

Z faktu, że *głęboka struktura zdań* języków naturalnych jest taka sama dla wszystkich języków wynika, że można sformalizować gramatykę odpowiadającą *głębokiej strukturze zdań*. Taka sformalizowana gramatyka jest modelem syntaktycznej części *gramatyki uniwersalnej* jaką posiada użytkownik rodzimego języka określaną jako jego kompetencję.

Ogólnie, każdy język jest zbiorem zdań złożonych z szeregu *symboli terminalnych* (końcowych), zwanych też *terminalami*, pochodzących ze skończonego słownika. Oprócz tego, z każdym zdaniem może być związana abstrakcyjna struktura zbudowana z *symboli nieterminalnych* (pomocniczych) zwanych też *nieterminalami*.

*Nieterminal* jest zmienną oznaczającą, że może on być zastąpiony (przepisany), przy pomocy istniejącej *reguły gramatycznej*, na odpowiadający jemu inny szereg terminali i nieterminali. Jeżeli czynność przepisywania będzie powtarzana po kolei dla wszystkich uzyskanych drogą przepisywania nieterminali, ostatecznie uzyska się zdanie, w którym wszystkie symbole będą terminalami. Proces ten nosi nazwę *generacji zdań* na podstawie gramatyki.

Najistotniejszym elementem w procesie generacji zdań jest zbiór *reguł gramatycznych* zwanych też *regułami przepisywania* lub krótko *produkcjami*. Reguła przepisywania definiuje, co ma być w derywowanym zdaniu zastąpione. Decyduje o tym lewa strona reguły. Natomiast o tym, na co ma być zastąpione, decyduje prawa strona reguły. Po dokonaniu zastąpienia z poprzedniego szeregu symboli powstaje inny szereg symboli, przepisany (derywowany), przy pomocy tej reguły.

Chomsky przebadał różne modele sformalizowanej gramatyki pod względem przydatności do reprezentowania gramatyki języka naturalnego i uporządkował je hierarchicznie według typów ponumerowanych od 0 do 4.

Okazuje się, że typ gramatyki uzależniony jest tylko od ograniczenia nałożonego na postać reguł zawartych w tych gramatykach.

Jeżeli na reguły gramatyki nie nałożono żadnych ograniczeń polegających na tym, że po prawej i lewej stronie reguły występują dowolne szeregi złożone z terminali i nieterminali, wówczas powstają

gramatyki nieograniczonego przepisywania, które Chomsky nazwał gramatykami *rekursywnie przeliczalnymi*. Tym gramatykom Chomsky nadał typ 0.

Jeżeli na reguły nakładane są następujące ograniczenia:

- (1) po prawej i lewej stronie reguły nie może wystąpić pusty szereg symboli,
- (2) po lewej stronie reguły musi wystąpić tylko jeden symbol pomocniczy,
- (3) reguła nie może być *rekursywna* tzn. symbol po lewej stronie reguły nie może być częścią prawej strony reguły (tzw. jawne lub właściwe powtórzenie symbolu) ani też częścią prawej strony reguły występującej w kolejnej derywacji (tzw. niejawne lub niewłaściwe powtórzenie symbolu).

to powstają gramatyki, które Chomsky nazwał *gramatykami bezkontekstowymi struktur frazowych* (CF - context free phrase structure), w skrócie *gramatykami bezkontekstowymi*, a generowany przez te gramatyki język nosi nazwę *języka bezkontekstowego*. Tym gramatykom Chomsky nadał typ 3.

Jeżeli na reguły *gramatyk bezkontekstowych* nałożymy dodatkowe ograniczenie:

- (4) prawa strona reguły musi zawierać co najwyżej jeden symbol pomocniczy (nieterminalny).

to powstają gramatyki, które Chomsky nazwał *gramatykami skończenie stanowymi*, zwane też *gramatykami regularnymi* lub *liniowymi*. Tym gramatykom Chomsky nadał typ 4.

Jeżeli w regułach *gramatyk bezkontekstowych* zmienimy (2) ograniczenie na następujące:

- (5) symbol pomocniczy wskazany w lewej stronie reguły może być wtedy użyty, jeżeli wystąpi w szeregu symboli w określonym prawo i lewostronnym kontekście zdefiniowanym w regule.

to powstają gramatyki, które Chomsky nazwał *gramatykami kontekstowymi struktur frazowych* (CF – context free phrase structure), w skrócie *gramatykami kontekstowymi*, a generowany przez te gramatyki język nosi nazwę *języka kontekstowego*. Tym gramatykom Chomsky nadał typ 2.

Oczywiście gramatyka bezkontekstowa jest szczególnym przypadkiem gramatyki kontekstowej, dla której konteksty: lewy i prawy są puste, oznaczające, że zastosowanie reguły nie zależy od kontekstu przepisywanego nieterminala.

Gramatykom typu 2 i typu 3, czyli *gramatykom kontekstowym* i *bezkontekstowym* Chomsky nadał ogólną nazwę *gramatyki struktur frazowych* (PS phase-structure).

Jeżeli w regułach *gramatyk struktur frazowych* zmienimy (3) ograniczenie (które nie pozwala na wystąpienie rekursywności) na następujące:

- (6) dozwala się wystąpienie reguły rekursywnej w ten sposób, że symbol po lewej stronie reguły może być częścią prawej strony reguły (tzw. jawne lub właściwe powtórzenie symbolu) lub też częścią prawej strony reguły występującej w kolejnej derywacji (niejawne lub niewłaściwe powtórzenie symbolu).

to powstają gramatyki, które Chomsky nazwał *rekursywnymi gramatykami struktur frazowych*, w skrócie *gramatykami rekursywnymi*. Tym gramatykom Chomsky nadał typ 1. Cechą charakterystyczną tych gramatyk jest to, że ze skończonego słownika i skończonego zbioru reguł gramatycznych można wprowadzić nieskończenie wiele zdań.

Podsumowując, Chomsky przedstawił następującą hierarchię gramatyk:

- typ 0: gramatyki rekursywnie przeliczalne,
- typ 1: gramatyki rekursywne,
- typ 2: gramatyki kontekstowe struktur frazowych lub gramatyki kontekstowe,
- typ 3: gramatyki bezkontekstowe struktur frazowych lub gramatyki bezkontekstowe,
- typ 4: gramatyki skończenie stanowe lub regularne (liniowe).

Podział ten oznacza, że jeżeli pewien język jest generowany przez gramatykę określonego typu, to nie musi być również generowany przez gramatykę typu wyższego, ale nie odwrotnie.

Należy zwrócić uwagę na fakt, że gramatyki typu 4 (skończenie stanowe) nie zostały przez Chomsky'ego uznane, jako potencjalne modele języków naturalnych.

Można też zauważyć, że rekursję (reguły rekursywne) można dopuścić już na poziomach gramatyki skończenie stanowej typu 4 oraz gramatyki bezkontekstowej i kontekstowej typów 3 i 2. Wówczas można ograniczyć ilość gramatyk do 3 typów (od 0 do 3) w postaci gramatyk rekursywnie przeliczalnych, kontekstowych, bezkontekstowych i skończenie stanowych. Tak też uczyniono w niektórych opracowaniach.

### **Automaty realizujące modele języka naturalnego**

---

Następną sprawą jaką Chomsky rozważył to kwestia, jakie automaty mogą realizować poszczególne gramatyki, czyli rozpoznawać generowany przez gramatykę język.

Zdaniem Chomsky'ego gramatyki skończenie stanowe typu 4 realizują automaty skończenie stanowe (ang. finite state machine), które są niczym innym jak jednokierunkowymi maszynami Turinga. Gramatyki bezkontekstowe typu 4 realizują automaty skończenie stanowe ze stosem (ang. pushdown automation). Gramatyki kontekstowe typu 3 realizują maszyny Turinga liniowo ograniczone. Gramatyki rekursywne typu 1 realizują wszystkie wyżej wymienione automaty w zakresie gramatyk typu 4, 3 i 2, gdyż wszystkie dopuszczają możliwość rekursji. Gramatyki rekursywnie przeliczalne typu 0 realizują maszyny Turinga, które są równoważne automatom skończenie stanowym z dwoma stosami (ang. two stack pushdown automaton).

Należy podkreślić, że jeżeli automat realizuje gramatykę niższego typu, np. rekursywnie przeliczalną typu 0, to realizuje również gramatykę wyższego typu, np. bezkontekstową typu 3. Oznacza to, że wszystkie gramatyki wymienione w hierarchii Chomsky'ego mogą być realizowane przez automat skończenie stanowy, który w razie potrzeby można zmodyfikować o obsługę jednego lub dwóch stosów. Jest to bardzo cenny wniosek, pozwalający na stwierdzenie, że na przykład mechanizm świadomości w zakresie języka i myślenia może być zrealizowany w oparciu o najprostszy z możliwych do skonstruowania automatów, jakim jest automat skończenie stanowy.

W dalszej części pracy przedstawiono automat skończenie stanowy w sposób odmienny niż to uczyniono w innych opracowaniach.

Na wstępie należy podkreślić, że zasada działania automatu polega na przyjmowaniu przez automat kolejnych stanów terminalnych lub nieterminalnych.

Stan terminalny oznacza, że jest z nim związany zbiór alternatywnych reguł terminalnych, z których tylko jedna reguła może być uaktywniona przez terminal podany na wejście maszyny (wtedy część warunkowa reguły pasuje do tego terminala).

Stan nieterminalny oznacza, że jest z nim związana jedna reguła nieterminalna, wyznaczona przez poprzednią regułę.

Zadaniem automatu jest realizacja procesu kolejnego przepisywania nieterminali zawartych w szeregach terminalnych i nieterminalnych gramatyki pewnego języka w oparciu o dostępne reguły tej gramatyki.

Należy zauważyć, że począwszy od gramatyk bezkontekstowych, ze zdaniem terminalnym związana jest pewna struktura, w której można realizować wnioskowanie indukcyjne (od terminalnych szczegółów do nieterminalnego ogółu) oraz wnioskowanie dedukcyjne (od nieterminalnego ogółu do terminalnych szczegółów), stąd podział automatów na indukcyjne i dedukcyjne.

Opis reguł terminalnych i nieterminalnych uzależniony jest od sposobu ich przetwarzania przez automat.

Ze względu na sposób przetwarzania reguł przez automat wyróżniamy następujące automaty:

1. Automaty skończenie stanowe dla gramatyk rekursywnych skończenie stanowych typu 4.
2. Automaty indukcyjne dla gramatyk rekursywnych: bezkontekstowych, kontekstowych i rekursywnie przeliczalnych, typów 3, 2, 0.
3. Automaty dedukcyjne dla gramatyk rekursywnych: bezkontekstowych, kontekstowych i rekursywnie przeliczalnych, typów 3, 2, 0.

Ogólnie, każda reguła realizowana przez automat, składa się z dwóch części: warunku i akcji. Jeżeli warunek jest spełniony reguła zostaje uaktywniona skutkiem czego automat wykonuje działania wymienione w akcji reguły.

Należy nadmienić, że występuje tu rozróżnienie między regułą gramatyczną a regułą realizowaną przez automat. Dla gramatyk skończenie stanowych jedna reguła gramatyczna odpowiada jednej regule automatu skończenie stanowego, natomiast dla gramatyk bezkontekstowych, kontekstowych i rekursywnie przeliczalnych jedna reguła gramatyczna składa się z wielu reguł automatu skończenie stanowego.

W akcji reguły zapisane jest działanie automatu, a od warunku reguły zależy jej uaktywnienie.

Poniższa tabela pokazuje, jakie działania automatu są zapisane w akcji reguły w zależności od stanów automatu i rodzaju gramatyk:

Stan automatu	Akcja reguły automatu skończenie stanowego	Akcja reguły automatu indukcyjnego	Akcja reguły automatu dedukcyjnego
Stan terminalny	- następny terminal wejściowy - następny stan automatu	- następny terminal wejściowy - następny stan automatu - stan powrotu automatu na stos, jeżeli jest różny od stanu na szczycie stosu	- następny terminal wejściowy - powrót automatu do stanu zdjętego ze szczytu stosu
Stan nieterminalny		- następny stan auto-	- następny stan auto-

		- stanu powrotu automatu na stos, jeżeli jest różny od stanu na szczycie stosu	- stanu powrotu automatu na stos
Stan końcowy	- akceptacja zdania terminalnego - zatrzymanie się automatu		
Stan końcowy zapisany w akcji reguły uaktywnionej		- powrót automatu do stanu zdjętego ze szczytu stosu	
Pusty stos		- akceptacja zdania terminalnego - zatrzymanie się automatu	- akceptacja zdania terminalnego - zatrzymanie się automatu

Automat gramatyk skończenie stanowych przebiega tylko stany terminalne, a wybór kolejnego terminala zależy wyłącznie od terminala poprzedzającego (stąd też inna nazwa gramatyki: regularna albo liniowa). W akcji reguły automatu skończenie stanowego zapisane są następujące działania: wybór następnego terminala wejściowego i przełączenie automatu na stan wskazany w akcji tej reguły.

Po osiągnięciu przez automat stanu końcowego (koniec zdania terminalnego) automat akceptuje to zdanie i zatrzymuje się.

Automat gramatyk indukcyjnych przebiega stany terminalne jak i stany nieterminalne.

Dla stanów terminalnych w akcji reguły zapisane są następujące działania: wybór następnego terminala wejściowego, przełączenie automatu na stan wskazany w akcji reguły, przesłanie na stos stanu powrotu automatu wskazanego w akcji reguły pod warunkiem, że stan ten jest różny od stanu na szczycie stosu. Stany powrotów dotyczą stanów wyższego poziomu (bardziej ogólnych).

Dla stanów nieterminalnych akcja automatu jest taka sama jak dla stanu terminalnego z wyłączeniem wyboru następnego terminala wejściowego.

Jeżeli w akcji reguły jest zapisany stan następny jako końcowy, to dla gramatyk indukcyjnych zamiast przejścia do stanu końcowego i wyłączenia automatu, jak dla gramatyk skończenie stanowych, następuje przejście do stanu powrotu zdjętego ze szczytu stosu. Oznacza to kontynuację pracy automatu na wyższym poziomie struktury stanów nieterminalnych.

W przypadku, gdy stos jest pusty, dla gramatyk indukcyjnych następuje akceptacja zdania terminalnego i zatrzymanie się automatu.

Automat gramatyk dedukcyjnych przebiega podobnie jak automat gramatyk indukcyjnych stany terminalne i stany nieterminalne. Występuje tu jednak różnica.

Dla stanów terminalnych w akcji reguły zapisane są następujące działania: wybór następnego terminala wejściowego, przełączenie automatu na stan powrotu zdjęty ze szczytu stosu. Oznacza to kontynuację pracy automatu dla kolejnej prawej gałęzi struktury stanów nieterminalnych (od góry gałęzi do dołu).

Dla stanów nieterminalnych w akcji reguły zapisane są następujące działania: przełączenie automatu na stan wskazany w akcji reguły, przesłanie na stos stanu powrotu automatu wskazanego w akcji reguły.

W przypadku, gdy stos jest pusty, dla gramatyk dedukcyjnych podobnie jak dla indukcyjnych następuje akceptacja zdania terminalnego i zatrzymanie się automatu.

Charakterystyczne jest to, że dla automatu gramatyk indukcyjnych akcja uaktywnionej reguły wskazuje stan następny na tym samym poziomie a stan powrotu umieszczany na stosie dotyczy stanu bezpośrednio wyżej leżącego w strukturze stanów nieterminalnych. Natomiast dla automatu gramatyk dedukcyjnych akcja reguły wskazuje jako następny stan bezpośrednio niżej leżący w strukturze stanów nieterminalnych, a stan powrotu umieszczany na stosie dotyczy stanu następnego na tym samym poziomie w strukturze stanów nieterminalnych.

Podsumowując, wszystkie gramatyki wymienione przez Chomsky'ego można przetwarzać bardzo prostym automatem skończenie stanowym modyfikując go, w razie potrzeby, o obsługę pamięci stosowej lub dwóch pamięci stosowych, a modelem właściwym języka naturalnego wydaje się być gramatyka rekursywnie przeliczalna, zaś automatem do jej obsługi automat skończenie stanowy z dwoma stosami.

Drugi stos automatu skończenie stanowego służy do przeszukiwania drzewa terminalno-nieterminalnego dotyczącego części warunkowej reguł gramatyk rekursywnie przeliczalnych.

### **Indukcja gramatyk języka naturalnego**

---

Postępowanie dotyczące przetwarzania języka naturalnego w oparciu o jego gramatykę zakładało, że mamy gotową strukturę gramatyczną, którą użytkownik języka rodzimego już zna (jest jego kompetencją językową), a zadaniem lingwisty jest badanie tej kompetencji w postaci konstruowania różnych formalnych modeli gramatyki języka naturalnego. Takie tworzenie formalnych modeli języka naturalnego odbywa się zwykle w oparciu o wiedzę, doświadczenie, intuicję, zgadywanie, testowanie, zasadę - prób i błędów, upór, pracowitość, inteligencję, spryt, itp. jego twórcy. Sprawia to tyle wysiłku umysłowego co programowanie komputerów. Zazwyczaj zakres prac obejmuje: analizę, projektowanie, testowanie, modyfikowanie, utrzymanie i rozwój.

Do tej pory rozważaliśmy automaty, które przetwarzają zdania języka, operujące na gotowych już gramatykach tego języka. Takie gramatyki rozważał Chomsky zakładając, że użytkownik języka zna doskonale jego gramatykę (ma kompetencję posługiwania się tym językiem).

Do opracowania gramatyki języka naturalnego można dochodzić metodą manualną. Chomsky dochodził do gramatyki języka angielskiego metodą manualno – ewolucyjną. Utworzoną manualnie gramatykę języka angielskiego, w ramach opracowanej przez siebie teorii, modyfikował następnie w ramach kolejnej teorii, aby zmodyfikowana gramatyka była lepszym modelem gramatyki języka angielskiego. Była to praca bardzo żmudna i możliwa tylko do zrealizowania przez doskonałych językoznawców.

Inny sposób postępowania pokazują prace projektowe dotyczące samouczenia się maszyn. Najczelniejszym zadaniem jest opracowanie najprostszego z możliwych modeli maszyn, który nie tylko mógłby być wykorzystany w samouczeniu się maszyn, ale stanowić najprostszy i najłatwiejszy potencjalny wzór do naśladowania przez naturę w organizmach żywych, w tym mózgu człowieka.

Dobrym modelem dla tych maszyn jest samoucząca się jednokierunkowa maszyna Turinga (samouczący się automat skończenie stanowy), która, na podstawie dowolnych danych wejściowych, tworzy swój własny program, w postaci zestawu reguł swojego działania, za pomocą którego potrafi nie tylko akceptować te dane wejściowe, z którymi się spotkała, ale również te dane, z którymi maszyna się jeszcze nie zetknęła. To zadanie realizuje samouczący się rekursywny automat skończenie stanowy.

Zasadniczą cechą tej maszyny jest to, że ilość danych wejściowych, przy pomocy których maszyna samoprogramuje się jest skończona (zazwyczaj niewielka), natomiast ilość danych, które maszyna nauczy się akceptować jest nieskończona ilość (dzięki obecności reguł rekurencyjnych).

Pierwszy model samouczącego się automatu skończenie stanowego nosi nazwę rachunku sekwentowego, w którym wbudowano mechanizm samouczenia w postaci reguł. Rachunek ten jest uproszczeniem rachunku tautologicznego, który powstał z rachunku zdań. Najistotniejszą regułą jest reguła konsolidacji sekwentów w struktury oraz konsolidacji struktur w większe struktury. Konsolidacja posiada bardzo prostą i jasną formułę, na której mogłaby się potencjalnie wzorować natura w organizmach żywych, w tym mózg człowieka.

Następnie rachunek sekwentowy ewoluował do samouczącego się automatu bezkontekstowego i kontekstowego ponieważ, według poglądu Chomsky'ego, gramatyka skończenie stanowa nie może być odpowiednim modelem dla gramatyki języka naturalnego. Nowa wersja rachunku bazuje już na modelach gramatyki języka naturalnego, tym samym samouczenie się maszyny obejmuje też samouczenie się języka naturalnego.

Zasadniczym elementem nowego rachunku sekwentowego jest obsługa pamięci stosowych w regułach i obsługa bazy pojęć. Bazę pojęć tworzą reguły w przestrzeni symboli nieterminalnych z najniższym poziomem symboli terminalnych.

Można zaznaczyć, że mechanizm uczenia się nowych pojęć przypomina proces indukcji automatu skończenie stanowego ze stosem na poziomie najbardziej ogólnym.

Podkreślić należy, że rachunek sekwentowy z obsługą bazy pojęć nadaje się do obsługi języka naturalnego. Jest on równoważny formalnemu modelowi gramatyki języka bezkontekstowego, kontekstowego, a nawet rekursywnie przeliczalnego.

Należy zauważyć, że istnieją inne metody indukcji gramatycznych, jak: statystyczne, genetyczne i neuronowe (oparte o sztuczną sieć neuronową), jednak metoda symboliczna, jaką jest rachunek sekwentowy, jest najprostszym i najłatwiejszym modelem indukcji gramatyk, mogącym mieć potwierdzenie w sekwencyjnych procesach świadomych.

## Bibliografia

1. John Lyons, *Chomsky*, Prószyński i Sówka, Warszawa 1998
2. Noam Chomsky, *Syntactic structures*, New York 2002
3. Noam Chomsky, *Aspects of the theory of syntax*, MIT 1965
4. Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018