Uniwersytet Warszawski
Wydział Filozofii i Socjologii

Nina Gierasimczuk
Nr albumu: 189173

Algorytmiczne podejście do problemu uczenia się języka

Praca magisterska
na kierunku Filozofia
w zakresie Logika

Praca wykonana pod kierunkiem
prof. dr hab. Marcina Mostowskiego
Instytut Filozofii, Uniwersytet Warszawski

Warszawa, czerwiec 2005

Oświadczenie kierującego pracą

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data | Podpis kierującego pracą

Oświadczenie autora pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data | Podpis autora pracy

Streszczenie

W pracy opisujemy algorytmiczne podejście do problemu uczenia się języka, opierając się na pojęciach językoznawstwa generatywnego (zob. (Chomsky, 1957)). Koncentrujemy się głównie na problemie nabywania wiedzy składniowej. Analizujemy model identyfikowalności w granicy (Gold, 1967), który ukazuje możliwości ogólnie pojętej wyuczalności. Rozważamy również model uczenia się przez zapytania (Angluin, 1987), (Sakakibara, 1990). Opisujemy algorytm L^*, który uczy się języków regularnych oraz algorytm LA uczący się gramatyk bezkontekstowych. Oba algorytmy wykorzystują w uczeniu się jednocześnie negatywne i pozytywne dane na temat poszukiwanego języka. Analizujemy założenia wyżej wymienionych modeli pod kątem ich psychologicznej adekwatności oraz badamy w jakim stopniu wyjaśniają naturalny proces uczenia się języka. Podkreślamy znaczenie modelowania procesu przyswajania języka dla epistemologii. Końcowa część pracy poświęcona jest problemowi użyteczności omówionych w pracy narzędzi do modelowania procesu przyswajania semantyki języka.

Słowa kluczowe

algorytmy uczące się, generatywizm, gramatyki bezkontekstowe, hierarchia Chomsky'ego, identyfikowalność w granicy, języki regularne, teoria uczenia się, twierdzenie Golda, uczenie się z danych pozytywnych i negatywnych

Dziedzina pracy

(kody według programu Socrates Erasmus)

08100

It will not be possible to apply exactly the same teaching process to the machine as to a normal child. It will not, for instance, be provided with legs, so that it could not be asked to go out and fill the coal scuttle. Possibly it might not have eyes. But however well these deficiencies might be overcome by clever engineering, one could not send the creature to school without the other children making excessive fun of it.

Alan Turing „Computer Machinery and Intelligence”

Spis treści

1. Przedmowa

Jednym z wielkich problemów epistemologii jest wyjaśnienie zjawiska przyswajania wiedzy i umiejętności, a w szczególności nabywania zdolności językowych. W jaki sposób człowiek uczy się swego pierwszego języka? Jak to możliwe, że korzystając ze szczątkowych i często wadliwych danych udaje mu się wykryć strukturę języka na tyle poprawnie, by móc się skutecznie porozumiewać? W jaki sposób zdobywa umiejętność konstruowania zdań, z którymi nigdy wcześniej się nie zetknął? Jak wreszcie opisać fragment umysłu odpowiadający za zdolności językowe, aby wyjaśnić fakt, że człowiek równie dobrze może jako pierwszego języka nauczyć się chińskiego jak i polskiego? Z tymi pytaniami mierzył się w swoich badaniach Noam Chomsky — twórca współczesnego paradygmatu badań nad językiem. Postulował on istnienie wrodzonego mechanizmu umysłowego, umożliwiającego nabywanie zdolności językowych. W pracach *Syntactic Structures* oraz *The Logical Stucture of Linguistic Theory* zaproponował generatywistyczny model języka, który umożliwia analizowanie składni. Problem wyuczalności języka podjął E.M. Gold w pracy *Identification in the Limit.* Jego propozycja daje wyobrażenie o możliwym kształcie wrodzonego mechanizmu nabywania zdolności językowych.

Niniejsza praca pozostaje w paradygmacie i w zasięgu problematyki wyznaczonych przez Chomsky'ego. Opisano tutaj model nabywania języka zaproponowany przez Golda. Ponadto przedstawiono inne rozwiązania mieszczące się w tym samym paradygmacie. Przedyskutowano je od strony metodologicznej, rozważono ich psychologiczną adekwatność oraz implikacje epistemologiczne. W końcowej części pracy przeanalizowano możliwość wykorzystania tych narzędzi do modelowania procesu uczenia się semantyki.

2. Wstęp

Istotne z naszego punktu widzenia cechy lingwistyki Chomsky'ego można określić następująco: rozróżnienie na semantykę i składnię w badaniach nad językiem oraz podkreślenie związków językoznawstwa z psychologią i filozofią umysłu.

Rozróżnienie na semantykę i składnię w badaniach nad językiem Rozróżnienie na semantykę i składnię wprowadzone zostało do językoznawstwa przez Chomsky'ego. We wczesnym stadium rozwoju językoznawstwa semantyka i syntaktyka albo nie były rozróżniane albo granica pomiędzy nimi nie była wyznaczona (Lyons, 1977).

Noam Chomsky nie zajmował się zagadnieniami semantyki języka naturalnego. Ścisłe zdefiniowanie pojęć składni pozwoliło jednak na precyzyjne wyznaczenie obszaru syntaktyki i w ten sposób oddzielenie go od kwestii znaczeń językowych. Wprowadzenie ścisłego, matematycznego aparatu służącego badaniom składni stworzyło możliwość ujmowania języka naturalnego jako zbioru (przynajmniej potencjalnie) nieskończonego. Pozostając w zgodzie z poprawnością gramatyczną można bowiem dowolnie wydłużać zdania języka naturalnego, np.:

(1) Genji popatrzył na swoich ludzi.

(2) Piękny Genji popatrzył na swoich ludzi.

(3) (...), posępny, piękny Genji popatrzył na swoich ludzi.

albo:

(4) Posępny, piękny Genji popatrzył na swoich ludzi, którzy spali (...).

W ten sposób uzyskamy nieskończony zbiór poprawnie zbudowanych zdań. Istnieją oczywiście praktyczne ograniczenia na długość zdań. W ogólnym przypadku nie wiadomo jednak, w którym miejscu granicę tę umieścić. Dlatego, do opisu języka potrzebujemy narzędzia, które (potencjalnie) będzie produkować nieskończony zbiór zdań. Takim narzędziem jest gramatyka, która określa dopuszczalne sposoby łączenia elementów słownika. Dla potrzeb modelu zakładamy, że słownik jest ustalony, skończony i niezmienny. Model ten nie wyjaśnia więc zjawisk związanych z ewolucją języków — w tym paradygmacie zawsze badamy język w pewnym jego stadium. Przyjmujemy ponadto, że do opisu języka wystarcza skończenie wiele podstawowych reguł gramatycznych o charakterze rekurencyjnym. Wszak zdania 1–3 powstają wskutek powtarzania wciąż tej samej operacji — rozszerzania frazy rzeczownikowej przez dodanie przymiotnika z lewej strony. W dalszej części pracy znajdują się ścisłe definicje gramatyk oraz odpowiadających im klas języków.

Podkreślenie związków z psychologią, epistemologią i filozofią umysłu Zainicjowany przez Chomsky'ego przełom można wyrazić w stwierdzeniu: lingwistyka jest częścią psychologii, bowiem badania nad językiem przyczyniają się do głębszego zrozumienia procesów umysłowych (zob. np. (Chomsky, 1968)). Moduł umysłu, który umożliwia posługiwanie się językiem cechuje się dużą dostępnością badawczą. Język jako zewnętrzny wytwór procesów umysłowych staje się ważnym przedmiotem dociekań psychologicznych (zob. (Pinker, 2000), (Macnamara, 1986)).

W swej teorii języka Noam Chomsky wyraża przekonanie, że jesteśmy wyposażeni w szereg swoistych zdolności (którym nadajemy nazwę „umysłu"), odgrywających istotną rolę w przyswajaniu wiedzy i umożliwiających nam działania dowolne, niezdeterminowane przez zewnętrzne bodźce (choć podlegające pewnym wpływom z ich strony) (Chomsky, 1965). Jedną z tych zdolności ma być możliwość przyswojenia języka, aktywowania tzw. kompetencji językowej. Warunkiem uaktywnienia tej wrodzonej potencjalności jest obecność odpowiedniego bodźca. W jednej ze swych prac cytuje Leibniza: „idee i prawdy są nam wrodzone jako skłonności, dyspozycje, nawyki i naturalne potencjalności", od siebie dodaje: „doświadczenie służy wydobywaniu, a nie formowaniu tych wrodzonych struktur" (Chomsky, 1959). Jednym z celów lingwistyki Chomsky'ego jest odkrycie struktury ludzkiej kompetencji językowej wraz z jej wrodzonymi elementami. Teza o wrodzoności pewnych zdolności językowych wspierana jest następującymi argumentami.

Po pierwsze, językiem posługujemy się w sposób twórczy. Ze względu na nieskończony zbiór zdań, który może być generowany przez człowieka, języka nie można traktować ściśle behawiorystycznie, jako „nawyku" lub „zbioru dyspozycji do reagowania". Zbiór zdań używanych przez człowieka nie zawiera się w zbiorze zdań wcześniej przez niego zasłyszanych. Jak już wcześniej wspomnieliśmy, postulowana przez Chomsky'ego kompetencja językowa ma postać zbioru reguł rekurencyjnych. Podmiot, wyposażony w zestaw takich reguł jest w stanie stwierdzać poprawność wyrażeń lub tworzyć nowe wyrażenia, bez względu na to, czy zetknął się z nimi wcześniej czy nie. Ten aspekt lingwistyki Chomsky'ego sprawił, że określa się ją mianem generatywnej.

Po drugie, według Chomsky'ego, struktura języka ma charakter uniwersalny. Chomsky twierdzi, że tzw. „struktury głębokie" są bardzo podobne w różnych językach, a reguły rządzące przekształceniami „struktur głębokich" w „struktury powierzchniowe" dają się zamknąć w wąskiej klasie operacji formalnych (Chomsky, 1995). Zjawisko to początkowo wyjaśniane jest przez odwołanie się do wrodzonego mechanizmu nabywania kompetencji językowej (Chomsky, 1965). W późniejszych pracach Chomsky zakłada istnienie podstawowego zrębu wszystkich gramatyk, obecnego w każdym umyśle. Ten podstawowy zbiór reguł nazywa gramatyką uniwersalną (Chomsky, 1995). Chomsky odrzuca inne możliwe przyczyny strukturalnego podobieństwa języków: postulat maksymalnej prostoty, do której mogły dążyć języki podczas swojego rozwoju, postulat „powszechnej logiczności" oraz koncepcję wspólnego pochodzenia wszystkich języków ludzkich. Nie tylko fakt podobieństwa różnych istniejących języków naturalnych jest dla niego zastanawiający. Równie zaskakujące jest podobieństwo występujące wśród zdolności językowych poszczególnych użytkowników tego samego języka.

W pracy opiszemy generatywny model języka, który daje możliwość ścisłego rozważania zagadnień składni. Przyjmiemy też tezę o istnieniu wrodzonego mechanizmu służącego przyswajaniu języka i zastanowimy się nad jego możliwą strukturą. Chomsky opisuje mechanizm nabywania tzw. „wiedzy językowej" w następujący sposób:

> [...] dany schemat gramatyki określa klasę możliwych hipotez; metoda interpretacji umożliwia testowanie każdej hipotezy na danych wejścia; miara ewaluacyjna wybiera najwyżej ocenianą gramatykę zgodną z danymi. Od momentu, gdy hipoteza [...] zostanie wybrana, uczący się zna język przez nią definiowany. W szczególności [...] [jest on w stanie generować] nieskończony zbiór zdań z którymi nigdy się nie zetknął. (Chomsky, 1967)

Wrodzonym elementem kompetencji językowej ma więc być według Chomsky'ego aparat służący nabywaniu zdolności językowej, *LAD* (skrót od *Language Acquisition Device*)(Chomsky, 1965). Problem opisu *LAD* podjął E. M. Gold w pracy *Identification in the Limit.* Zaproponował tam model nabywania wiedzy składniowej, zwany identyfikacją w granicy. Postulowane przez niego rozwiązanie ma charakter hipotetyczny, jest jedną z możliwych odpowiedzi na problem postawiony przez Chomsky'ego. Jednocześnie propozycja Golda jest bardzo ogólna — modeluje wyuczalność wskazując na jej ograniczenia, bez odwołania do poszczególnych praktycznych realizacji algorytmów uczących się. Model ten nie wywarł wpływu ani na współczesną psychologię poznawczą ani epistemologię. Stało się tak prawdopodobnie z powodu matematycznie zaawansowanego zestawu narzędzi jakie wykorzystuje. Propozycja ta spotkała się zaś z potężnym odzewem w informatyce teoretycznej, językoznawstwie komputerowym i badaniach nad sztuczną inteligencją. Jej aspekt psycholingwistyczny pozostał praktycznie niezbadany.

3. Wprowadzenie terminologii

Poniżej przedstawimy podstawowe definicje i ustalenia notacyjne.

Alfabet to skończony i niepusty zbiór symboli.

Na przykład: $A = \{a, b\}$ — alfabet binarny;

Słowo to skończony ciąg symboli wybranych z pewnego alfabetu.

Na przykład: ciąg „$aaabbbaababaa$" jest słowem nad alfabetem A.

Długość słowa to liczba symboli w nim występujących, oznaczamy ją przez $lh()$, np. $lh(aa)$=2.

Słowo puste, oznaczane symbolem λ, to ciąg zerowej długości, czyli: $lh(\lambda) = 0$.

Jeśli A jest alfabetem, to przez A^k oznaczamy zbiór słów długości k nad alfabetem A. Na przykład $\{a, b\}^2 = \{aa, ab, ba, bb\}$. Dla dowolnego alfabetu A mamy $A^0 = \{\lambda\}$.

Zbiór wszystkich słów nad alfabetem A oznaczamy przez A^*, np. $\{a, b\}^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, \ldots\}$. Innymi słowy $A^* = \bigcup_{n\in\omega} A^n$.

Przez xy, dla $x, y \in A^*$, oznaczamy *konkatenację* słowa x oraz słowa y, czyli słowo powstałe z kopii słowa x, po której bezpośrednio następuje kopia słowa y. Na przykład: jeśli $x = bab$ oraz $y = a$, to $xy = baba$. Dla dowolnego ciągu $\alpha \in A^*$ zachodzi $\lambda\alpha = \alpha\lambda = \alpha$, czyli λ jest elementem neutralnym dla konkatenacji.

Dla dowolnych języków $X, Y \subseteq A^*$ określamy ich konkatenację XY, jako $XY = \{\alpha\beta : \alpha \in X,\ \beta \in Y\}$.

Niech $x, y \in A^*$. Wtedy:

— x jest *prefiksem* y wtedy i tylko wtedy, gdy istnieje $z \in A^*$ takie, że $y = xz$.

— x jest *sufiksem* y wtedy i tylko wtedy, gdy istnieje $z \in A^*$ takie, że $y = zx$.

Niech $B \subseteq A^*$. Wtedy:

— Zbiór B jest *domknięty na prefiksy* wtedy i tylko wtedy, gdy:

$$\forall x, y \in A^* \ [(y \in B \wedge x \text{ jest prefiksem } y) \Longrightarrow x \in B].$$

— Zbiór B jest *domknięty na sufiksy* wtedy i tylko wtedy, gdy:

$$\forall x, y \in A^* \ [(y \in B \wedge x \text{ jest sufiksem } y) \Longrightarrow x \in B].$$

Dowolny zbiór słów wybranych z A^*, dla pewnego alfabetu A, nazywamy *językiem*. Jeśli A jest alfabetem oraz $L \subseteq A^*$, to mówimy, że L jest językiem nad A^*.

Definicja 1. *Gramatyka G jest to struktura postaci (A, Σ, S, P), gdzie:*

— *A jest alfabetem (alfabetem terminalnym);*
— *Σ jest zbiorem zmiennych (alfabetem nieterminalnym);*
— *$S \in \Sigma$ jest symbolem początkowym;*
— *P jest skończonym zbiorem par postaci $\alpha_i \longrightarrow \beta_i$ dla $i = 1, \ldots, n$ oraz $\alpha_i, \beta_i \in (A \cup \Sigma)^*$.*

Definicja 2. *Dla dowolnych $\gamma, \gamma' \in (A \cup \Sigma)^*$ powiemy, że γ' jest w gramatyce G bezpośrednio wyprowadzalna z γ, czyli $\gamma \longrightarrow_G \gamma'$ wtedy i tylko wtedy, gdy istnieją η_1, η_2 oraz $i = 1, \ldots, n$, takie, że $\gamma = \eta_1 \alpha_i \eta_2$ oraz $\gamma' = \eta_1 \beta_i \eta_2$.*

Definicja 3. *γ' jest w gramatyce G wyprowadzalna z γ, czyli $\gamma \longrightarrow_G^* \gamma'$ wtedy i tylko wtedy, gdy istnieje ciąg $\gamma_1, \ldots, \gamma_m \in (A \cup \Sigma)^*$ taki, że $\gamma = \gamma_1$, $\gamma' = \gamma_m$ oraz $\gamma_i \longrightarrow_G \gamma_{i+1}$, dla $i = 1, \ldots, m-1$.*

Definicja 4. *Język generowany przez gramatykę G jest to zbiór: $L(G) = \{\gamma \in A^* : S \longrightarrow_G^* \gamma\}$.*

Hierarchia języków

— Język L jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy istnieje maszyna Turinga e taka, że

$$\alpha \in L \iff \{e\}(\alpha) \downarrow, \text{dla dowolnego } \alpha \in A^*,$$

czyli: maszyna Turinga e zatrzymuje się tylko na słowach należących do języka L.

— Język L jest rekurencyjny wtedy i tylko wtedy, gdy istnieje maszyna Turinga M licząca funkcję χ_L taką, że dla dowolnego $\alpha \in L$:

$$\chi_L(\alpha) = \begin{cases} 0 & \text{jeśli } \alpha \notin L \\ 1 & \text{jeśli } \alpha \in L. \end{cases}$$

— Język L jest pierwotnie rekurencyjny wtedy i tylko wtedy, gdy jego funkcja charakterystyczna jest pierwotnie rekurencyjna (w sprawie pojęć teorii rekursji zob. (Shoenfield, 1991), (Cutland, 1980)).

— Języki kontekstowe są postaci $L(G)$, gdzie wszystkie produkcje gramatyki G są postaci:

$$\eta_1 Y \eta_2 \longrightarrow_G \eta_1 \beta \eta_2, \text{ dla } Y \in \Sigma, \eta_1, \eta_2, \beta \in (A \cup \Sigma)^*.$$

— Języki bezkontekstowe są postaci $L(G)$, gdzie wszystkie produkcje gramatyki G są postaci:

$$Y \longrightarrow_G \beta, \text{ dla } Y \in \Sigma, \ \beta \in (A \cup \Sigma)^*.$$

— Języki regularne są postaci $L(G)$, gdzie wszystkie produkcje gramatyki G są postaci:

$$Y \longrightarrow_G \alpha Z \text{ lub } Y \longrightarrow_G \alpha, \text{ dla } Y,\ Z \in \Sigma,\ \alpha \in A^*.$$

4. Identyfikowalność w granicy

Człowiek nie uczy się pierwszego języka poprzez zapamiętywanie reguł gramatycznych podanych *explicite*. Odbywa się to przez pewien rodzaj generalizacji zasłyszanych elementów języka. Od algorytmu uczącego się oczekujemy tego samego — przyswajania zbioru reguł gramatycznych na podstawie szczątkowych danych, przypadkowych elementów języka.

Model identyfikacji w granicy ukazuje możliwy przebieg tego procesu. Model działa dla ustalonej klasy języków. Zostaje z niej wybrany jeden język. Uczeń otrzymuje informacje na jego temat. Ukazane są one na jeden z możliwych sposobów — według przyjętej metody prezentacji danych. Zadanie ucznia polega na odgadnięciu nazwy wybranego języka. Nazwami języków są ich opisy, na przykład gramatyki. Model identyfikacji w granicy polega więc na wyszukiwaniu odpowiedniej gramatyki opisującej zaprezentowany ciąg informacji dotyczącej wybranego języka. W każdym kolejnym kroku procedury uczniowi prezentowana jest jednostka danych dotycząca nieznanego języka. Każdorazowo uczeń ma więc do dyspozycji tylko skończony zbiór danych. W każdym kroku uczeń wybiera nazwę języka. Ta procedura przebiega w nieskończoność. Język jest identyfikowany w granicy, gdy po pewnym czasie uczeń wybiera poprawnie ciągle tę samą nazwę (gramatykę). Właściwa identyfikowalność dotyczy klas języków. Cała klasa języków jest identyfikowalna w granicy, gdy istnieje algorytm zgadujący (uczeń) taki, że identyfikuje on w granicy dowolny język z tej klasy. Warto zaznaczyć, że uczeń nie ma dostępu do informacji, kiedy jego zgadnięcia są prawidłowe. Przetwarza informacje w nieskończoność, ponieważ nie może stwierdzić, czy w kolejnym kroku nie dostanie informacji, która zmusi go do zmiany wyboru.

Identyfikowalność zależy trzech czynników: wybranej klasy języków, metody prezentacji danych dotyczących języka oraz przyjętej relacji nazywania. Najważniejszą częścią modelu jest zaś algorytm zgadujący (uczeń).

4.1. Klasa języków

Zanim uruchomimy procedurę uczenia się należy wybrać pewną klasę języków $\Omega = \{L : L \subseteq A^*\}$. Będziemy w pracy rozważać klasy języków skończonych, regularnych, bezkontekstowych, kontekstowych, pierwotnie rekurencyjnych, rekurencyjnych i rekurencyjnie przeliczalnych. Klasy te zostały zdefiniowane wcześniej. Z ustalonej klasy wybieramy jeden język, dla ustalenia uwagi L_{44}. Zadaniem ucznia będzie odszukanie nazwy opisującej L_{44}. Wybierać będzie spośród udostępnionego mu zbioru nazw wszystkich języków z klasy Ω.

4.2. Metoda prezentacji danych

Informacja o języku może być prezentowanana uczniowi na dwa sposoby: jako tekst lub jako informant. Zdefiniowane poniżej ciągi danych nazywamy inaczej ciągami treningowymi.

Definicja 5. ***TEKST dla*** *L jest ω – ciągiem I słów $\alpha_1, \alpha_2, \ldots \in L$, takim, że każde słowo $\alpha \in L$ pojawia się przynajmniej raz w I.*

Tekst dla danego języka L składa się z elementów L. Dla każdego języka jest ciągiem nieskończonym, ponieważ dopuszczamy powtarzanie się jego elementów. Identyfikowalność przy pomocy tekstu nazywana jest też identyfikowalnością z danych pozytywnych.
W zależności od sposobu wyliczenia i–tego elementu tekstu wyróżniamy następujące rodzaje tekstu:
— arbitralny, to taki, którego i–ty element wyliczany jest przez dowolną funkcję;
— rekurencyjny, to taki, którego i–ty element wyliczany jest przez dowolną rekurencyjną funkcję od numeru kroku w którym element się pojawia;
— pierwotnie rekurencyjny, to taki, którego i–ty element jest wyliczany przez dowolną pierwotnie rekurencyjną funkcję od numeru kroku w którym element się pojawia.

Definicja 6. ***INFORMANT dla*** *L jest ω – ciągiem I elementów z $(A^* \times \{0,1\})$, takim, że dla każdego $\alpha \in A^*$:*

$$\begin{array}{ll} (\alpha,1) \in I & \textit{jeśli } \alpha \in L \\ (\alpha,0) \in I & \textit{jeśli } \alpha \notin L. \end{array}$$

Informant to uporządkowanie wszystkich możliwych ciągów nad danym alfabetem wraz z informacjami, czy elementy te należą, czy nie należą do danego języka. Tak więc, I jest identyczna z funkcją charakterystyczną χ_L zbioru L względem zbioru A^*. Identyfikowalność przy pomocy informantu nazywana jest też identyfikowalnością poprzez dane pozytywne i negatywne.
W zależności od sposobu wyliczenia i–tego ciągu wyróżniamy następujące rodzaje informantu:
— arbitralny, to taki, którego i–ty element jest wyliczany przez dowolną funkcję, o ile każdy ciąg z A^* pojawia się w informancie przynajmniej raz;
— metodyczny, to taki, który jest apriorycznym uporządkowaniem A^*, zaś i–ty element to element w porządku o numerze i;
— żądający, to taki, którego i–ty element wybierany jest przez ucznia w kroku i na podstawie dotychczasowych danych.

4.3. Relacja nazywania

Uczeń może używać dwóch tzw. relacji nazywania, czyli dwóch sposobów decydowania o adekwatności danej nazwy dla poszukiwanego języka:

Definicja 7. ***Generator dla*** *L jest Maszyną Turinga e taką, że:*

$$L = \{\alpha \in A^* : \{e\}(\alpha) \downarrow\}.$$

Jak już wspomnieliśmy nazwy mają charakter opisowy. Z gramatyki – nazwy opisowej danego języka, łatwo otrzymujemy maszynę Turinga będącą generatorem. Nazywanie przebiega poprzez generowanie z wybranej gramatyki odpowiednich ciągów oraz porównywanie ich z otrzymanymi wcześniej danymi. Jeśli są zgodne, uczeń decyduje się na wybór tej maszyny Turinga. Jeśli nie, wybiera inną i z nią postępuje tak samo.

Definicja 8. ***Tester dla*** *L jest Maszyną Turinga, która oblicza funkcję* χ_L *taką, że dla każdego* $\alpha \in A^*$*:*

$$\chi_L(\alpha) = \begin{cases} 0 & \textit{jeśli } \alpha \notin L \\ 1 & \textit{jeśli } \alpha \in L. \end{cases}$$

Nazywanie przebiega w tym przypadku poprzez sprawdzanie wybranym testerem przynależności do języka wszystkich elementów otrzymanego ciągu danych. Jeśli dla wszystkich elementów odpowiedź testera była pozytywna, uczeń wybiera ten tester. Jeśli dla któregoś wynik postępowania był negatywny, uczeń odrzuca tę hipotezę i wybiera inny tester.

4.4. Algorytm zgadujący

Algorytmowi zgadującemu odpowiada funkcja $Zg : I \longrightarrow \mathcal{N}$, gdzie I jest zbiorem wszystkich możliwych skończonych ciągów informacji, $\mathcal{N}$ jest zbiorem wszystkich nazw języków z danej klasy.

Definicja 9. *Niech* $I_L = i_1, i_2, i_3, \ldots$ *będzie* ω*–ciągiem treningowym dla L. Wtedy:*
L jest identyfikowany w granicy przez funkcję Zg *względem ciągu* $I_L \iff$ *istnieje nazwa N języka L taka, że* $\exists k \forall u \geqslant k \Rightarrow Zg(i_1, \ldots, i_u) = N$.

Definicja 10. *Niech* Ω *będzie klasą par postaci* (L, N)*, gdzie N jest nazwą języka L. Wtedy:*
Ω *jest identyfikowalna w granicy* $\iff$ *dowolny język L z klasy* Ω *jest identyfikowany w granicy (oczywiście względem odpowiedniego systemu nazw).*

Z reguły ważniejsze klasy języków mają pewne kanoniczne systemy nazw, na przykład: języki regularne można nazywać za pomocą automatów skończonych, gramatyk regularnych lub wyrażeń regularnych. Odpowiednie systemy nazw są tutaj wzajemnie efektywnie przekładalne. Nie ma więc znaczenia, który system wybieramy. Podobne systemy nazw mają języki bezkontekstowe (gramatyki bezkontekstowe lub automaty ze stosem), języki pierwotnie rekurencyjne (termy definiujące pierwotnie rekurencyjne funkcje charakterystyczne). Maszyny Turinga liczące funkcje charakterystyczne są kanonicznym systemem nazw dla języków rekurencyjnych. W tym przypadku sam system nazw nie jest jednak rekurencyjny.

W pracach poświęconych teorii uczenia przyjmuje się, że każda z tych podstawowych klas języków zadana jest wraz z odpowiadającym jej kanonicznym systemem nazw. Klasy języków rozpatrywane z punktu widzenia wyuczalności muszą być reprezentowane wraz z odpowiednim systemem nazw, czyli przyporządkowaniem każdemu językowi zbioru jego nazw lub, równoważnie, należy rozpatrywać klasy par (L, N), gdzie N jest nazwą języka L. Przypomnijmy, że nazwa zawiera skończony opis odpowiedniego języka. Przyjęte jest, że tam, gdzie istnieje taki kanoniczny system nazw pomija się go w opisie odpowiedniej klasy języków. Tak więc, mówimy po prostu o wyuczalności klas języków regularnych, bezkontekstowych, kontekstowych, pierwotnie rekurencyjnych, rekurencyjnych, milcząco zakładając, że określony jest odpowiedni kanoniczny system nazw.

Przyjmując powyższy aparat możemy udowodnić następujące twierdzenia:

Twierdzenie 1. *Następujące warunki są równoważne:*

1. *klasa języków jest identyfikowalna w granicy przy pomocy arbitralnego informantu;*
2. *klasa języków jest identyfikowalna w granicy przy pomocy metodycznego informantu;*
3. *klasa języków jest identyfikowalna w granicy przy pomocy żądającego informantu.*

Twierdzenie 2. *Klasa języków rekurencyjnych nie jest identyfikowalna w granicy przez metodyczny informant z generatorem.*

Dowód twierdzenia Niech Zg będzie algorytmem zgadującym. Załóżmy, że Zg identyfikuje w granicy dowolny język z klasy języków rekurencyjnych R.

Zauważmy, że do klasy R należą wszystkie języki skończone i koskończone (czyli takie, których dopełnienie jest skończone). Skonstruujemy język $L \in R$ poprzez jego funkcję charakterystyczną $\chi_L : A^* \to \{0, 1\}$. Przy ustalonym ω – porządku $\alpha_0, \alpha_1, \ldots$ na A^*, χ_L można utożsamić z ciągiem $\omega \to \{0, 1\}$ takim, że $\chi_L(\alpha_0), \chi_L(\alpha_1), \ldots$

Naszą konstrukcję rozpoczynamy od pustego ciągu. W n–tym kroku skonstruowany został pewien ciąg $\beta = i_0 \dots i_m$. Ciąg $\beta 0^\infty$ opisuje funkcję charakterystyczną pewnego skończonego języka K_0. Ponieważ $K_0 \in R$ więc dla pewnego s istnieje N będąca nazwą języka K_0 taką, że $Zg(\beta 0^w) = N$, dla dowolnego $w \geqslant s$. Zauważmy, że procedura znalezienia takiego s może być nierekurencyjna. Załóżmy, że znaleźliśmy takie s. Rozważmy teraz język koskończony K_1 opisany przez ciąg $\beta 0^s 1^\infty$. Istnieje więc M, nazwa K_1, oraz t takie, że $Zg(\beta 0^s 1^u) = M$, dla dowolnego $u \geqslant t$. $K_0 \neq K_1$, gdyż K_0 jest skończony, zaś K_1 jest koskończony. Z tego powodu również $N \neq M$. Wiemy więc, że istnieją s, t takie, że:

$$(*)\ Zg(\beta 0^s) \neq Zg(\beta 0^s 1^t).$$

Możliwe, że nie istnieje algorytm znajdujący parę liczb s i t takich, jak opisane powyżej. Dla dowodu wystarczy jednak istnienie jakiejkolwiek pary s i t spełniającej $(*)$.

Znajdujemy teraz parę s, t za pomocą następującego algorytmu A, przy zadanym przeliczeniu ω^2:

```
p := (1,1)
done:=false
while not done do
  {s:=pierwszy element p
   t:=drugi element p
   if Zg(β0^s) ≠ Zg(β0^s 1^t) then
     RETURN p
   else p:= następna para po p
  }
```

Istnienie pary (s, t) spełniającej $(*)$ gwarantuje, że dla zadanego Zg oraz β algorytm A da na wyjściu parę również spełniającą ten warunek. Funkcja Zg jest rekurencyjna i całkowita, więc A jest poprawnym algorytmem, który zawsze kończy pracę. Określamy teraz, w kroku $n+1$ ciąg β' jako $\beta' = \beta 0^s 1^t$. W ten sposób konstruujemy ω–ciąg I. Warto zauważyć, że identyfikowalność w granicy nie zależy od rodzaju informantu (Twierdzenie 1). Możemy więc przyjąć, że I opisuje informant. Z konstrukcji wynika, że dla dowolnego n, jeśli γ jest początkowym słowem I o długości n, to istnieją δ_0 oraz δ_1 takie, że:

$$Zg(\gamma\delta_0) \neq Zg(\gamma\delta_0\delta_1).$$

□

Dodajmy, że z powyższego dowodu nie wynika, że $Zg(\gamma\delta_0)$ i $Zg(\gamma\delta_0\delta_1)$ są nazwami różnych języków. Dowód nie wyklucza sytuacji, w której kolejno wymieniane nazwy opisują ten sam język. Mielibyśmy w takim przypadku do czynienia z identyfikacją języka w słabszym, niż proponowany przez Golda sensie[1].

Twierdzenie 3. *Klasa języków skończonych jest identyfikowalna w granicy przez arbitralny tekst z testerem.*

Dowód twierdzenia Niech $C = \{L_1, L_2, \ldots\}$ będzie klasą wszystkich języków skończonych. Niech I będzie tekstem dla pewnego L_i, gdzie $i \in \omega$. Określmy teraz odpowiedni algorytm zgadujący Zg. W każdym kroku Zg zgaduje nazwę języka, który składa się jedynie z wymienionych w tekście elementów. Ponieważ L_i jest skończony, każdy tekst w pewnym momencie wymieni wszystkie elementy tego języka. Zg zidentyfikuje więc L_i w granicy.
$\square$

Twierdzenie 4. *Żadna klasa języków zawierająca wszystkie języki skończone i przynajmniej jeden nieskończony (tzw. nadskończona klasa języków) nie jest identyfikowalna w granicy przez rekurencyjny tekst z generatorem.*

Dowód twierdzenia Niech C będzie klasą wszystkich języków skończonych wraz z pewnym nieskończonym językiem P. Przypuśćmy, że klasa C jest identyfikowalna w granicy. Istnieje więc algorytm Zg rozpoznający dowolny język z klasy C na podstawie tekstu. Skonstruujemy tekst I taki, że Zg będzie nieskończoną liczbę razy zmieniał zdanie czytając I. Niech $P_1 \subsetneq P_2 \subsetneq \ldots \subsetneq P$ będzie ciągiem języków takim, że P_i jest skończony, $i \in \omega$ oraz P jest nieskończony.

Niech $P_1 = \{\alpha_1, \ldots, \alpha_n\}$. Wówczas określamy pierwsze n słów tekstu I jako $\alpha_1, \ldots, \alpha_n$. W n–tym kroku Zg może nie rozpoznać P_1. Generujemy więc w dowolnym porządku słowa z P_1 aż Zg rozpozna P_1. Tak stanie się w skończonej liczbie kroków, gdyż $P_1 \in C$, zaś C jest identyfikowalna w granicy przez Zg. Gdy Zg zidentyfikuje P_1, kontynuujemy konstruowanie I przez określenie następnych jego elementów jako elementów P_2. Elementy P_2 powtarzają się w tekście tak długo, aż Zg zidentyfikuje P_2. Postępujemy tak dla kolejnych języków skończonych z klasy C. Widać wyraźnie, że skonstruowany w ten sposób tekst nie pozwoli Zg zidentyfikować wszystkich języków z klasy C. Z założenia o identyfikowalności wszystkich języków z klasy C wynika, że Zg będzie zmuszony zgadywać kolejno języki skończone, nie mogąc zidentyfikować języka nieskończonego P należącego do klasy C. Sprzeczność. $\square$

[1] Spostrzeżenie to pochodzi od Marcina Mostowskiego

Poniżej znajduje się zestawienie dotyczące wyuczalności w modelu identyfikowalności w granicy.

Pierwotnie rekurencyjny	rekurencyjnie przeliczalne
tekst z generatorem	rekurencyjne
Informant	pierwotnie rekurencyjne
	kontekstowe
	bezkontekstowe
	regularne
	nadskończona
Tekst	języki skończonej mocy

Tabela 1. Wyuczalność i niewyuczalność klas języków w modelu Golda

4.5. Podsumowanie

Przeanalizujemy teraz podstawowe założenia modelu Golda. Porównamy je z naturalnym procesem przyswajania zdolności językowych.

1. Uczenie jest procesem nieskończonym. Założenie to odpowiada intuicji na temat nabywania zdolności językowych. Nie tylko w dzieciństwie rozwijamy zdolności lingwistyczne. Możliwe, że edukacja językowa nie kończy się w jakimś ustalonym momencie, że wciąż dowiadujemy się nowych rzeczy na temat języka. Z czasem dane językowe coraz rzadziej są istotne dla rozwoju naszej kompetencji językowej. Nawet jeśli uznamy, że proces nabywania wiedzy składniowej jest skończony i upływa wraz z np. trzynastym rokiem życia (zob. (Kurcz, 2000)), to wciąż nieskończony model identyfikacji możemy z powodzeniem odnosić do procesu przyswajania semantycznej wiedzy językowej. Ponadto tak jak w naturalnym procesie tak i w modelu Golda uczeń nigdy nie wie, kiedy gramatyka, którą się posługuje jest „prawidłowa”. Wciąż istnieje bowiem możliwość, że niektóre z danych językowych spowodują zmiany w przyswojonej gramatyce.
2. Efektem uczenia się jest ustabilizowanie ciągu wyników na jednej, poprawnej odpowiedzi. Założenie to można osłabić oczekując jedynie stabilizacji odpowiedzi na jednym języku, dopuszczając wahania w identyfikacji konkretnej gramatyki. Jak widzieliśmy w dowodzie twierdzenia 2 osłabienie definicji identyfikowalności w granicy może zmienić własności wyuczalności. Co więcej, niewykluczone, że może doprowadzić do nowych wyników w dziedzinie teorii uczenia się.

   Uznajemy czyjąś znajomość języka na podstawie sprawnego porozumiewania się. Na drugi plan schodzi wtedy kwestia, jaką gramatyką dany osobnik się posługuje. Nie musimy zakładać, że język naturalny posiada jeden prawidłowy opis gramatyczny i że właśnie ten opis jest standardowo przyswajany przez wszystkich użytkowników języka. Co więcej, nie musimy nawet zakładać, że pojedynczy człowiek ustala swą wiedzę językową raz na zawsze. Stabilizacja ta nie jest konieczna, o ile jej brak nie zakłóca sprawnego porozumiewania się. Takie złagodzenie warunku przyswojenia języka nie jest zgodne ze stanowiskiem Chomsky'ego. Według niego język posiada jedną prawidłową strukturę gramatyczną, która każdorazowo podlega wyuczeniu (Chomsky, 1995) (por. (Quine, 1990)).

3. Pojęcie identyfikowalności dotyczy określonych klas języków. Jeżeli uczenie miałoby przebiegać na zasadzie identyfikacji języka wybranego z pewnej określonej klasy, mielibyśmy poszlakę w postaci górnej granicy złożoności języka naturalnego. Na przykład, jeśli mechanizm uczenia umożliwia wyuczalność tylko klasy języków kontekstowych, to górnym ograniczeniem na złożoność struktury języka naturalnego byłaby właśnie kontekstowość. W drugą stronę ta zależność nie zachodzi. Przyjmijmy, że udało się przeanalizować dostępne języki naturalne pod kątem ich struktury składniowej, a następnie wydać sąd na temat górnego ograniczenia ich złożoności. Wciąż możliwe jest to, że mechanizm wyuczalności działa również dla języków z wyższych klas. Co więcej możliwe, że ten sam mechanizm wyuczalności stosuje się do rozmaitych umiejętności pozajęzykowych. Uczymy się przecież nie tylko języka, lecz także gestykulacji, mimiki czy prowadzenia samochodu. Aktywności tego rodzaju mogą mieć dowolną złożoność, której nie jesteśmy w stanie w tej chwili oszacować. Daleko nam bowiem do zrozumienia i matematycznego opisania zasad rządzących tymi zdolnościami. Dlatego możliwe, że mechanizm wyuczalności jest silniejszy niż oczekiwalibyśmy, gdyby dotyczył tylko uczenia się języka naturalnego.
4. Uczeń posiada umiejętność testowania ciągów za pomocą automatów i generowania ciągów za pomocą gramatyk. Jest to założenie dotyczące struktury algorytmu zgadującego (relacji nazywania), które można rozważyć w kontekście wrodzonych mechanizmów poznawczych. Nie wydaje się ono zbyt kontrowersyjne. Można je bez większych zastrzeżeń akceptować.
5. Zaletą propozycji Golda jest fakt, że model ten nie wymaga istnienia jakichkolwiek wrodzonych uczniowi struktur gramatycznych. Struktura gramatyczna jest w całości przyswajana z próbek językowych przez pewnego rodzaju uogólnienie. Przyjmując model Golda pomija się więc kwestię wspólnego zrębu wszystkich języków ludzkich — tzw. gramatyki uniwersalnej. W założeniu miałaby ona być tym, co łączy wszystkie języ-

ki ludzkie, a zarazem wszystkie ludzkie umysły w aspekcie kompetencji językowej (Chomsky, 1995).

Ciekawym wynikiem pracy *Identification in the limit* jest twierdzenie 4. Głosi ono, że żadna klasa, która mogłaby zawierać język naturalny, nie jest wyuczalna z danych pozytywnych (tekstu). Język naturalny zwykle umieszcza się w hierarchii na wysokości języków bezkontekstowych, niekiedy kontekstowych (zob. (Partee et al., 1993)). Przyjęcie modelu Golda jako trafnie opisującego mechanizm nabywania języka daje niebanalny wynik istotny dla psychologii poznawczej, epistemologii, językoznawstwa, jak również badań nad sztuczną inteligencją. Okazuje się bowiem, że dla identyfikacji (wyuczalności) języka naturalnego potrzeba czegoś więcej niż tylko danych pozytywnych. Wymaga ona informacji na temat granic wyuczanego języka w postaci informacji negatywnej: *to zdanie nie należy do poszukiwanego języka.* Czy wynik ten zgodny jest ze świadectwami empirycznymi dotyczącymi naturalnego przyswajania języka? Niektórzy uważają, że informacja negatywna jest znikoma i ignorowana przez dziecko (Chomsky, 1965). Badacze wciąż spierają się na ten temat, proponując nowe ograniczenia na model wyuczalności tak, aby rozszerzyć identyfikowalność z danych pozytywnych na klasy języków wyższe niż skończone (zob. np. (Angluin, 1980)). Ograniczenie wynikające z twierdzenia 4 nie wydaje nam się kontrowersyjnie. Dziecko we wczesnym stadium lingwistycznym otrzymuje dane pozytywne w postaci zasłyszanych zdań, negatywne zaś w postaci korekt pochodzących od nauczyciela. Pojawia się tutaj pytanie o nieodzowność udziału nauczyciela w procesie przyswajania języka. Jego rola będzie bardzo wyraźna w kolejnym prezentowanym w pracy modelu uczenia się przez zapytania.

5. Uczenie się języków regularnych z danych pozytywnych i negatywnych

Dana Angluin w pracy *Learning Regular Sets from Queries and Counterexamples* rozpatruje możliwość skończonej identyfikacji języka regularnego z danych pozytywnych i negatywnych. Identyfikacja odbywa się poprzez znalezienie odpowiadającego poszukiwanemu językowi deterministycznego automatu skończonego. Nabywanie wiedzy o nieznanym języku przebiega pod kontrolą nauczyciela zwanego minimalnie adekwatnym (*Minimally Adequate Teacher*). Nauczyciel odpowiada na pytania dotyczące należenia wyrażeń do języka oraz sprawdza, czy wysunięta przez ucznia hipoteza (uzyskany na podstawie danych automat skończony) jest poprawna. Jeśli hipoteza okazuje się błędna, to nauczyciel podaje kontrprzykład, który należy do symetrycznej różnicy zbioru poszukiwanego i zbioru generowanego przez automat–hipotezę. W dalszej części pracy zaprezentujemy algorytm L^*, który przyswaja dowolny język regularny przy pomocy dowolnego minimalnie adekwatnego nauczyciela. L^* działa w czasie wielomianowym względem liczby stanów minimalnego deterministycznego automatu skończonego akceptującego język oraz względem maksymalnej długości kontrprzykładów podawanych przez nauczyciela.

5.1. Deterministyczne automaty skończone

Jak wspomnieliśmy na początku tego rozdziału, uczeń uzyskuje wiedzę na temat nieskończonego zbioru (języka) regularnego dzięki konstrukcji automatu odpowiadającego poszukiwanemu zbiorowi. Automat skończony odpowiadający danemu zbiorowi L to taki automat, który akceptuje wszystkie i tylko elementy zbioru L. Poniżej znajduje się definicja deterministycznego automatu skończonego oraz języka rozpoznawanego przez ten model obliczeń.

Definicja 11. *Deterministyczny automat skończony (DFA) jest to piątka postaci* (A, Q, q_0, F, δ), *gdzie:*

— A *jest alfabetem wejściowym;*
— Q *jest skończonym zbiorem stanów;*
— $q_0 \in Q$ *jest wyróżnionym stanem początkowym;*
— $F \subseteq Q$ *jest zbiorem stanów akceptujących;*
— $\delta : Q \times A \longrightarrow Q$ *jest funkcją przejścia.*

Język rozpoznawany (akceptowany) przez DFA H to zbiór tych słów nad alfabetem A, które są akceptowane przez H, czyli:

$$L(H) = \{w \in A^* : \bar{\delta}(q_0, w) \in F\},$$

gdzie $\bar{\delta}$ to uogólniona funkcja przejścia określona następującymi warunkami rekurencyjnymi:

$$\bar{\delta}(q, \lambda) = q$$

$$\text{oraz dla dowolnego } w \in A^* \text{ i } a \in A, \ \bar{\delta}(q_0, wa) = \delta(\bar{\delta}(q_0, w), a).$$

Przykład Poniżej znajduje się kilka prostych przykładów języków regularnych wraz z akceptującymi je automatami skończonymi.

Niech $A = \{a, b\}$. Rozważmy język $L_1 = A^*$. $L_1 = L(H_1)$, gdzie $H_1 = (Q_1, q_{1s}, F_1, \delta_1)$ taki, że: $Q_1 = \{q_{1s}\}$, $F_1 = \{q_{1s}\}$, $\delta_1(q_{1s}, a) = q_{1s}$ oraz $\delta_1(q_{1s}, b) = q_{1s}$.

Niech $L_2 = \emptyset$. Wówczas $L_2 = L(H_2)$, gdzie $H_2 = (Q_2, q_{2s}, F_2, \delta_2)$ taki, że: $Q_2 = \{q_{2s}\}$, $F_2 = \emptyset$, $\delta_2(q_{2s}, a) = q_{2s}$ oraz $\delta_2(q_{2s}, b) = q_{2s}$.

Niech $L_3 = \{\lambda\}$. Wówczas $L_3 = L(H_3)$, gdzie $H_3 = (Q_3, q_{3s}, F_3, \delta_3)$ taki, że: $Q_3 = \{q_{3s}, q_{31}\}$, $F_3 = \{q_{3s}\}$, $\delta_3(q_{3s}, i) = q_{31}$ oraz $\delta_3(q_{31}, i) = q_{31}$ dla $i = a, b$.

Automaty skończone często reprezentuje się w postaci grafu, w którym wierzchołki symbolizują stany, stan początkowy jest zaznaczony strzałką, stan akceptujący jest wyróżniony przez podwójne kółko a strzałki pomiędzy stanami opisują funkcję przejścia na literach reprezentowanych przez etykiety tych strzałek.

a, b q_{1s} q_{2s} q_{3s}

Rysunek 1. DFA akceptujące L_1, L_2 oraz L_3.

Niech symbol $n_x(w)$ oznacza liczbę wystąpień litery x w słowie w. Opiszemy teraz język, w którym występują tylko słowa o równej, co do parzystości, liczbie wystąpień liter a i b. Zatem $L_4 = \{w \in A^* : n_a(w) \equiv n_b(w)(\mathrm{mod}\, 2)\}$. $L_4 = L(H_4)$, gdzie $H_4 = (Q_4, q_{4s}, F_4, \delta_4)$ taki, że: $Q_4 = \{q_{4s}, q_{41}\}$, $F_4 = \{q_{4s}\}$, $\delta_4(q_{4s}, i) = q_{41}$ oraz $\delta_4(q_{41}, i) = q_{4s}$, dla $i = a, b$.

a, b q_{4s} q_{41}

Rysunek 2. FA akceptujący $L_4 = \{w \in A^* : n_a(w) \equiv n_b(w)(\mathrm{mod}2)\}$.

Zauważmy, iż język ten możemy opisać inaczej jako zbiór wszystkich słów o nieparzystej długości nad alfabetem binarnym.

Wcześniej zdefiniowaliśmy języki regularne jako języki generowane przez gramatyki regularne. Zachodzi następujące twierdzenie:

Twierdzenie 5. *(Kleene, 1956) Język $L \subseteq A^*$ jest regularny, wtedy i tylko wtedy, gdy istnieje DFA H taki, że $L = L(H)$.*

5.2. Zapytania algorytmu L*

Algorytm L^* (uczeń) w sposób istotny korzysta ze wskazówek nauczyciela, który zawsze odpowiada poprawnie (dlatego czasem nazywany jest wyrocznią) na zadawane przez ucznia pytania. Rozważymy dwa rodzaje pytań: pytania o należenie oraz pytania o równoważność. Pierwsza klasa pytań dotyczy poszczególnych ciągów nad pewnym ustalonym alfabetem. Pytania są postaci: Czy dany ciąg α należy do poszukiwanego języka? Nauczyciel odpowiada TAK, jeśli ciąg α należy do tego języka, w przeciwnym przypadku odpowiada przecząco. Drugi rodzaj pytań dotyczy automatu M skonstruowanego przez algorytm L^*. Pytania te mają następującą postać: Czy zbiór, który jest akceptowany przez automat M jest identyczny z poszukiwanym zbiorem? Nauczyciel odpowiada TAK, jeśli zbiory te są identyczne, w przeciwnym przypadku pada odpowiedź negatywna. Jest ona każdorazowo stowarzyszona z kontrprzykładem — ciągiem należącym do symetrycznej różnicy zbioru poszukiwanego oraz zbioru odpowiadającego skonstruowanemu przez ucznia automatowi. Warto zauważyć, że o ile istnieje jedna poprawna odpowiedź na pytanie o należenie, o tyle poprawnych negatywnych odpowiedzi na pytanie o równoważność jest wiele. Pociąga to za sobą konkluzję, że różni nauczyciele będą tak samo odpowiadali na pytania pierwszego rodzaju. Różnice zaś będą się pojawiały w odpowiedziach na pytania drugiego rodzaju, nauczyciel może bowiem wybrać dowolny kontrprzykład spośród nieskończenie wielu możliwych.

Poniżej ścisły zapis obu kategorii zapytań:

1. Zapytania o należenie danej struktury do szukanej gramatyki:

$$T(\alpha) = \begin{cases} 1 & \text{jeśli } \alpha \in L \\ 0 & \text{jeśli } \alpha \notin L. \end{cases}$$

2. Zapytania o ekstensjonalną równoważność automatu skończonego otrzymanego przez algorytm L^* oraz automatu nauczyciela:

$$R(M) = \begin{cases} 1 & \text{jeśli } \mathbf{L}(M) = L \\ \langle 0, \alpha \rangle, \alpha \in L \div \mathbf{L}(M) & \text{jeśli } \mathbf{L}(M) \neq L. \end{cases}$$

5.3. Tablica obserwacyjna

W każdym kroku swego działania algorytm L^* dysponuje danymi na temat skończonego zbioru ciągów nad alfabetem. Ciągi te są sklasyfikowane według odpowiedzi nauczyciela na pytania o należenie. W każdym kroku działania algorytmu L^* mamy więc do czynienia z pewnym skończonym fragmentem informantu (w sensie Golda) dla poszukiwanego języka. Informant ten jest specyficzny ze względu na ciągi do niego należące — są one konstruowane z dotychczasowych danych językowych przez algorytm uczący się. Cechują się maksymalną prostotą względem otrzymywanych kontrprzykładów. Przechowywaną przez algorytm L^* informację można przedstawić w postaci dwuwymiarowej tabeli, zwanej tablicą obserwacyjną.

Definicja 12. ***Tablica obserwacyjna*** *jest to* (S, E, T)*, gdzie:*

1. *S jest niepustym skończonym zbiorem ciągów domkniętym na prefiksy;*
2. *E jest niepustym skończonym zbiorem ciągów domkniętym na sufiksy;*
3. *T jest skończoną funkcją $(S \cup SA)E \to \{0, 1\}$, gdzie*[2]*:*

$$T(s) = 1 \iff s \in L.$$

(S, E, T) przedstawiamy za pomocą dwuwymiarowej tablicy, w której:

1. Wiersze są oznaczone elementami zbioru $(S \cup SA)$.
2. Kolumny są oznaczone elementami zbioru E.
3. Wartość w komórce o współrzędnych (s, e), gdzie $s \in (S \cup SA)$, $e \in E$, jest równa $T(se)$.
4. Niech $s \in (S \cup SA)$, wtedy $row(s)$ jest skończoną funkcją $f : E \longrightarrow \{0, 1\}$ taką, że $f(e) = 1 \iff T(se) = 1$.

	T	e	E
		$\vdots$	
S	$s\dots$	$1(= T(se))$	
(SA)	s_1		

Tabela 2. Tablica obserwacyjna

Definicja 13. *Tablica obserwacyjna* (S, E, T) *jest* ***domknięta*** *wtedy i tylko wtedy, gdy:*

$$\forall t \in SA\ \exists s \in S\ row(t) = row(s).$$

[2] Zgodnie z powszechnym zwyczajem zakładamy, że operator konkatenacji wiąże silniej niż operator sumy.

Definicja 14. *Tablica obserwacyjna* (S, E, T) *jest* ***spójna*** *wtedy i tylko wtedy, gdy:*

$$\forall s_1, s_2 \in S\ \forall a \in A\ [row(s_1) = row(s_2) \Longrightarrow row(s_1 a) = row(s_2 a)].$$

5.4. Konstrukcja automatu skończonego przy użyciu tablicy obserwacyjnej

Algorytm L^* używa tablicy obserwacyjnej do stworzenia odpowiedniego automatu skończonego. Wiersze tablicy obserwacyjnej oznaczone elementami zbioru S są kandydatami na stany automatu. Kolumny zaś, oznaczone elementami zbioru E, to wejścia wczytywane przez automat. Wiersze oznaczone elementami zbioru SA używane są w konstrukcji funkcji przejścia tego automatu.

Definicja 15. *Niech* (S, E, T) *będzie domkniętą i spójną tablicą obserwacyjną. Wówczas* $M(S, E, T)$ *jest automatem skończonym* $M = (A, Q, q_0, F, \delta)$*, gdzie:*
- $Q = \{row(s) : s \in S\}$ *jest zbiorem stanów automatu* M*;*
- $q_0 = row(\lambda)$ *jest stanem początkowym automatu* M*;*
- $F = \{row(s) : s \in S \land T(s) = 1\}$ *jest zbiorem stanów akceptujących;*
- $\delta(row(s), a) = row(sa)$ *jest funkcją przejścia automatu* M*.*

Lemat 1. *Powyższy automat jest dobrze zdefiniowany.*

Dowód lematu Trzeba wykazać, że funkcja przejścia δ jest dobrze zdefiniowana. Niech $s_1, s_2 \in S$ takie, że: $row(s_1) = row(s_2)$. Skoro (S, E, T) jest spójna, to dla dowolnego $a \in A$, $row(s_1 a) = row(s_2 a)$. Ponieważ zaś (S, E, T) jest domknięta, to wartość ta jest równa $row(s)$ dla pewnego $s \in S$. $\square$

Definicja 16. *Automat skończony* $M(S, E, T)$ *jest zgodny ze skończoną funkcją* T *wtedy i tylko wtedy, gdy:*

$$\forall s \in (S \cup SA)\ \forall e \in E\ \bar{\delta}(q_0, se) \in F \Longleftrightarrow T(se) = 1.$$

Definicja 17. *Niech* M *oraz* M^1 *będą automatami deterministycznymi nad alfabetem* A *takimi, że:* $M = (Q^M, q_0^M, \delta^M, F^M)$ *oraz* $M^1 = (Q^{M^1}, q_0^{M^1}, \delta^{M^1}, F^{M^1})$*. Funkcja* $\Gamma : Q^M \longrightarrow Q^{M^1}$ *jest izomorfizmem automatów* M *i* M^1*, jeśli spełnione są następujące warunki:*

- $\Gamma : Q^M \longrightarrow Q^{M^1}$ *jest bijekcją;*
- $\Gamma(q_0^M) = q_0^{M^1}$*;*
- *dla dowolnego* $q \in Q^M, a \in A$ $[\Gamma(\delta^M(q, a)) = \delta^{M^1}(\Gamma(q), a)]$*;*

— $F^M = \Gamma(F^{M^1})$.

Twierdzenie 6. *Jeśli tablica obserwacyjna (S, E, T) jest domknięta i spójna to automat skończony $M(S, E, T)$ skonstruowany jak wyżej jest zgodny ze skończoną funkcją T. Każdy inny automat skończony zgodny z tą funkcją, lecz nieizomorficzny z automatem $M(S, E, T)$ musi mieć więcej stanów.*

W celu udowodnienia powyższego twierdzenia przeprowadzimy dowody trzech lematów.

Lemat 2. *Niech (S, E, T) będzie domkniętą i spójną tablicą obserwacyjną. Dla automatu skończonego $M(S, E, T)$ i dowolnego $s \in (S \cup SA)$ zachodzi: $\bar{\delta}(q_0, s) = row(s)$.*

Dowód lematu Nasze rozumowanie przeprowadzimy przez indukcję po długości s.

1. Zauważmy, że: $\bar{\delta}(q_0, \lambda) = row(\lambda) = q_0$.
2. Załóżmy indukcyjnie, że $s \in (S \cup SA)$ takie, że dla s zachodzi teza lematu. Niech $a \in A$, $t \in (S \cup SA)$ takie, że $t = sa$. Wówczas następujące warunki są równoważne:
   — $\bar{\delta}(q_0, t)$;
   — $\bar{\delta}(q_0, sa)$;
   — $\delta(\bar{\delta}(q_0, s), a)$;
   — $\delta(row(s), a)$ — z założenia indukcyjnego;
   — $row(sa)$ — z definicji δ;
   — $row(t)$ — ponieważ $t = sa$.

□

Lemat 3. *Niech (S, E, T) będzie domkniętą i spójną tablicą obserwacyjną. Wtedy automat skończony $M(S, E, T)$ jest zgodny ze skończoną funkcją T. To znaczy:*

$$\forall s \in (S \cup SA)\ \forall e \in E\ \bar{\delta}(q_0, se) \in F \Longleftrightarrow T(se) = 1.$$

Dowód lematu Dowód przeprowadzimy przez indukcję po długości e.

1. Niech $s \in (S \cup SA)$, $e = \lambda$. Wtedy $\delta(q_0, se) = row(s)$. Rozpatrzmy dwa przypadki:
   a) jeżeli $s \in S$, to z definicji F mamy: $row(s) \in F \Longleftrightarrow T(s) = 1$;
   b) jeżeli $s \in SA - S$, to $row(s) = row(s_1)$ dla pewnego $s_1 \in S$ — ponieważ (S, E, T) jest domknięta. Wtedy $row(s_1) \in F \Longleftrightarrow T(s_1) = 1 \Longleftrightarrow T(s) = 1$.
2. Założenie indukcyjne: Niech $e \in E$ takie, że dla dowolnego $s \in (S \cup SA)$ zachodzi:

$$\bar{\delta}(q_0, se) \in F \Longleftrightarrow T(se) = 1.$$

Niech $e' \in E$ takie, że $e' = ae$, dla pewnego $a \in A$. Ponadto niech s dowolny element $(S \cup SA)$. Ponieważ (S, E, T) jest domknięta, istnieje $s_1 \in S$ takie, że $row(s) = row(s_1)$. Skoro tak, to następujące warunki są równoważne:
— $\bar{\delta}(q_0, se') \in F$;
— $\bar{\delta}(\bar{\delta}(q_0, s), ae) \in F$;
— $\bar{\delta}(row(s), ae) \in F$ — z lematu 2;
— $\bar{\delta}(row(s_1), ae) \in F$ — ponieważ $row(s) = row(s_1)$;
— $\bar{\delta}(\delta(row(s_1), a), e) \in F$;
— $\bar{\delta}(row(s_1 a), e) \in F$ — z definicji δ;
— $\bar{\delta}(\bar{\delta}(q_0, s_1 a), e) \in F$ — z lematu 2;
— $\bar{\delta}(q_0, s_1 ae) \in F$;
— $T(s_1 ae) = 1$ — z założenia indukcyjnego;
— $T(sae) = 1$ — ponieważ $row(s) = row(s_1)$;
— $T(se') = 1$ — ponieważ $e' = ae$.
Stąd $\bar{\delta}(q_0, se') \in F \iff T(se') = 1$. □

Lemat 4. *Niech (S, E, T) będzie domkniętą i spójną tablicą obserwacyjną. Załóżmy, że automat skończony $M(S, E, T)$ ma n stanów. Niech $M' = (Q', q_0', F', \delta')$ będzie dowolnym automatem skończonym zgodnym z funkcją T, który ma n lub mniej stanów. Wtedy M' jest izomorficzny z $M(S, E, T)$.*

Dowód lematu Niech będą spełnione założenia lematu. Dowód przebiega przez pokazanie izomorfizmu M' oraz $M(S, E, T)$.

Definiujemy: dla każdego $q' \in Q'$, $\overline{row}(q')$ jest skończoną funkcją $f : E \longrightarrow \{0, 1\}$ taką, że:

$$f(e) = 1 \iff \bar{\delta}'(q', e) \in F.$$

Zauważmy, że M' jest zgodny z funkcją T, więc:

$$\forall s \in (S \cup SA) \forall e \in E \ \delta'(q_0', se) \in F \iff T(se) = 1.$$

Stąd:

$$\overline{row}(\bar{\delta}'(q_0', s)) = row(s).$$

Skoro s przebiega cały zbiór S, to $\overline{row}(\bar{\delta}'(q_0', s))$ przebiega wszystkie elementy zbioru Q, czyli:

$$Q = \{\overline{row}(\bar{\delta}'(q_0', s)) : s \in S \cup SA\}.$$

Tak więc M' musi mieć przynajmniej n stanów. Tak więc M' musi mieć dokładnie n stanów:

$$Q' = \{\bar{\delta}'(q_0', s) : s \in S\}.$$

Określmy więc $\Psi : Q' \longrightarrow Q$ taką, że: $\Psi(q') = q$, jeśli istnieje $s \in S$ spełniające $\bar{\delta}'(q'_0, s) = q'$ oraz $\overline{row}(q') = q$. Poprawność tej definicji wynika stąd, że dla każdego $q' \in Q'$ istnieje $s \in S$ takie, że $\bar{\delta}'(q'_0, s) = q'$.
Dla każdego $s \in S$ istnieje więc dokładnie jedno $q' \in Q'$ takie, że:

$$row(s) = \overline{row}(q') \text{ oraz } q' = \bar{\delta}'(q'_0, s).$$

Określamy teraz funkcję $\Phi : Q \longrightarrow Q'$ następująco: $\forall s \in S\ \Phi(row(s)) = \bar{\delta}'(q'_0, s)$.
Zarówno funkcja Ψ jak i Φ jest „1–1” i „na”. Pokażemy teraz, że Φ jest izomorfizmem $M(S, E, T)$ i M', zaś Ψ jest funkcją odwrotną do Φ czyli: $\Phi^{-1} = \Psi$. Pokażemy najpierw, że $\Phi^{-1} = \Psi$. W tym celu wystarczy wykazać, że dla dowolnego $q \in Q$ zachodzi $\Psi(\Phi(q)) = q$.
Niech $q \in Q$. Wówczas $q = \bar{\delta}(q_0, s)$, dla pewnego $s \in S \cup SA$. Wtedy następujące obiekty są identyczne:
— $\Psi(\Phi(q))$;
— $\Psi(\Phi(\bar{\delta}(q_0, s)))$ — ponieważ $q = \bar{\delta}(q_0, s)$;
— $\Psi(\Phi(row(s)))$ — z lematu 2;
— $\Psi(\bar{\delta}'(q'_0, s)))$ — z definicji Φ;
— $\Psi(\bar{\delta}'(q'_0, s'))$ — dla pewnego $s' \in S$ takiego, że: $row(s) = row(s')$, ponieważ (S, E, T) jest domknięta;
— $\bar{\delta}(q_0, s')$ — z definicji Ψ;
— $\bar{\delta}(q_0, s)$ — ponieważ $row(s) = row(s')$;
— q — ponieważ $q = \bar{\delta}(q_0, s)$.
Pokażemy teraz, że Φ spełnia pozostałe warunki izomorfizmu:
1. $\Phi(q_0) = \Phi(row(\lambda)) = \bar{\delta}(q_0, \lambda) = q_0$;
2. Niech $s \in S$ oraz $e \in E$ będą dowolne. Niech $s_1 \in S$ takie, że: $row(sa) = row(s_1)$. Następujące obiekty są identyczne:
   — $\Phi(\delta(row(s), a))$;
   — $\Phi(row(sa))$;
   — $\Phi(row(s_1))$;
   — $\bar{\delta}'(q'_0, s_1)$;
   — $\bar{\delta}'(q'_0, sa)$;
   — $\delta'(\bar{\delta}'(q'_0, s)), a)$;
   — $\delta'(\Phi(row(s), a)$.
3. Niech $s \in S$ takie, że $T(s) = 1$ czyli $row(s) \in F$. Wtedy $\Phi(row(s)) = q'$ oraz $\overline{row}(q') = row(s)$, więc $q' \in F'$.

□

Dowód twierdzenia Załóżmy, że tablica obserwacyjna (S, E, T) jest domknięta i spójna. Należy wykazać, że:

1. automat skończony $M(S, E, T)$ jest zgodny ze skończoną funkcją T;
2. każdy inny automat skończony zgodny z tą funkcją, lecz nieizomorficzny z automatem $M(S, E, T)$ musi mieć więcej stanów.

Pokazaliśmy już:

1. w dowodzie lematu 3, że $M(S, E, T)$ jest zgodny z funkcją T;
2. w dowodzie lematu 4, że każdy inny automat skończony zgodny z T jest albo izomorficzny z $M(S, E, T)$, albo zawiera przynajmniej jeden stan więcej.

Stąd mamy już: $M(S, E, T)$ jest minimalnym automatem skończonym zgodnym z funkcją T. □

5.5. Algorytm L*

Algorytm L^* działa na podstawie danych pozytywnych i negatywnych. Konstruuje początkową tablicę, następnie dąży do jej uspójnienia i domknięcia stosując zapytania o należenie. Gdy tablica stanie się domknięta i spójna, konstruuje z niej automat skończony na sposób opisany wcześniej. Wysuwa zapytanie o ekstensjonalną równoważność skonstruowanego automatu i automatu reprezentującego poszukiwany język. Jeśli otrzymuje odpowiedź pozytywną, to kończy pracę podając na wyjściu skonstruowany automat. Jeśli odpowiedź nauczyciela jest negatywna, to otrzymany kontrprzykład dodawany jest do tablicy obserwacyjnej, po czym algorytm wraca do poleceń uspójnienia i domknięcia tablicy. Poniżej znajduje się ścisły opis algorytmu L^* zapisany w tzw. pseudokodzie opartym na języku *Pascal* z elementami *C*.

<u>begin</u>
$S := \{\lambda\} \cup A$;
$E := \{\lambda\}$;
Stosując zapytania o należenie utwórz tablicę obserwacyjną
(S, E, T):
  kolumna etykietowana λ
  wiersze etykietowane wszystkimi $a \in A$
  komórka o współrzędnych (a, λ) etykietowana wartościami $T(a)$,
  dla każdego $a \in A$
<u>repeat</u>
  <u>while</u> ((S, E, T) nie jest domknięta lub nie jest spójna) <u>do</u>
    { <u>if</u> ((S, E, T) nie jest spójna) <u>then</u>
      { znajdź $s_1, s_2 \in S$, $a \in A$, $e \in E$ takie, że:
        $row(s_1) = row(s_2)$ i $T(s_1ae) \neq T(s_2ae)$
        dodaj s_1a do zbioru E

```
            rozszerz T do (S ∪ SA)E używając zapytań o należenie
          }
          if ((S, E, T) nie jest domknięta) then
          { znajdź s₁ ∈ S i a ∈ A takie, że:
            dla dowolnego s ∈ S row(s₁a) ≠ row(s)
            dodaj s₁a do zbioru S
            rozszerz T do (S ∪ SA)E używając zapytań o należenie
          }
        }
      M:=M(S, E, T);
      Zapytaj o poprawność M
      if odpowiedź=⟨0, kontrprzykład t⟩ then
        { dodaj kontrprzykład t i wszystkie jego prefiksy do S
          rozszerz T do (S ∪ SA)E używając zapytań o należenie
        }
until odpowiedź=TAK;
RETURN M;
end.
```

Warto zaznaczyć, że dla dowolnego tzw. minimalnie adekwatnego nauczyciela prezentującego nieznany regularny zbiór U, algorytm L^* podaje na wyjściu automat skończony H izomorficzny z minimalnym automatem akceptującym zbiór U.

Co więcej, jeśli n jest liczbą stanów minimalnego automatu dla zbioru U, zaś m jest górnym ograniczeniem długości kontrprzykładów, to całkowity czas działania algorytmu L^* jest wielomianowy względem n i m. Ponadto algorytm kończy obliczenie po najwyżej n zapytaniach o równoważność oraz najpóźniej po $n - 1$. wykonaniu głównej pętli.

5.6. Podsumowanie

Model Angluin eksplikuje pojęcie nauczyciela w teorii uczenia się języka. Taki opis stwarza możliwość modelowania roli nauczyciela w procesie uczenia się.

Ważną częścią modelu Angluin są dwojakiego rodzaju zapytania. Pytania o należenie nie wprowadzają niczego nowego w stosunku do modelu Golda. Jest to rozwinięcie idei przedstawiania danych językowych w postaci funkcji charakterystycznej danego języka, czyli informantu. Istotną nowością jest wprowadzenie zapytań o równoważność w celu uczynienia procesu uczenia się konkluzywnym. Założenie o udzielaniu przez nauczyciela poprawnej odpowiedzi na ten rodzaj pytań jest realistyczne. Problem ekstensjonalnej równoważności dwóch automatów skończonych jest bowiem rozstrzygalny (zob. (Hopcroft et al., 2001)).

Zadajmy sobie następujące pytania.

1. Czy możliwa jest redukcja zapytań?
2. Czy taki model wyuczalności można zastosować do języków wyższych klas, w szczególności do języków bezkontekstowych?

Ad. 1. Jeśli zrezygnujemy z zapytań o równoważność w procedurach identyfikacji języków nieskończonych, to uzyskamy algorytmy z dowolną dokładnością aproksymujące poszukiwany automat. Jak już wspomnieliśmy, taki aproksymujący model wydaje się adekwatny do opisu naturalnego przyswajania języka. Rezygnacja z zapytań o równoważność prowadzi do zastosowania metod probabilistycznych w inferencji gramatyk. Modelom probabilistycznym poświęcono wiele uwagi; szczególnie istotnym rozwiązaniem jest tzw. PAC – learning (*probably approximately correct learning*) (zob. np. (Pitt, 1989), (Valiant, 1984), (Parekh Honavar, 1997)).

Gold wykazał, że niemożliwe jest identyfikowanie jakiejkolwiek klasy języków zawierającej wszystkie języki skończone oraz jakikolwiek język nieskończony przy użyciu jedynie pozytywnych danych. Tak więc, redukcja zapytań o należenie mogłaby być użyteczna tylko o tyle, o ile wśród danych językowych, z których korzysta uczeń są wadliwe dane wraz z informacją, że są wadliwe. Rozważania naturalnego procesu przyswajania języka nie prowadzą do konieczności redukcji tych zapytań. Pochodzenie informacji negatywnej wydaje się nieodłącznie związane z postacią wspomagającego ten proces nauczyciela. Co więcej forma zaproponowana przez Angluin: konstruowania wadliwych struktur przez ucznia, a nie przez nauczyciela, również odpowiada naturalnej sytuacji uczenia. Ciągi generowane przez ucznia są oceniane przez nauczyciela pod względem ich poprawności. Tak założenie o konieczności danych pozytywnych i negatywnych, jak i metoda prezentacji w postaci zapytań jest adekwatna do opisu naturalnego procesu przyswajania języka.

Ad. 2. Problem skończonej wyuczalności języków wyższych klas rozważymy w następnym rozdziale.

6. Uczenie się gramatyk bezkontekstowych ze strukturalnych danych pozytywnych i negatywnych

Nie można opisać składniowej struktury języka naturalnego środkami gramatyk regularnych (Chomsky, 1957). Zatem próby utożsamienia LAD z algorytmem uczącym się języków regularnych, opisywanym w poprzednim rozdziale muszą być nieskuteczne. Ludzką zdolność przyswajania języka można naśladować tylko przy pomocy algorytmu uczącego się klas języków, które zawierają język naturalny. Często twierdzi się, że język naturalny jest bezkontekstowy, to znaczy, że właśnie środkami gramatyk bezkontekstowych można opisać jego strukturę składniową (zob. (Gazdar Pullum, 1985), (Pullum Gazdar, 1982), (Shieber, 1985)).

W tym rozdziale opiszemy algorytmy uczące się języków bezkontekstowych, rozważymy trudności pojawiające się przy konstruowaniu takich algorytmów oraz zbadamy ich moc wyjaśniającą.

Algorytm $\mathbf{L^{cf}}$ Angluin w pracy (Angluin, 1987) zaproponowała metodę identyfikacji gramatyk bezkontekstowych przy pomocy algorytmu L^{cf}. Poniżej opiszemy pokrótce tę propozycję.

Niech G będzie poszukiwaną gramatyką bezkontekstową. Algorytm uczący się L^{cf} zna w momencie inicjacji:

1. zbiór A symboli terminalnych gramatyki G;
2. zbiór Σ symboli nieterminalnych gramatyki G;
3. symbol początkowy S gramatyki G.

Algorytm korzysta z pomocy nauczyciela odpowiadającego na pytania dwóch kategorii:

1. Pytania o należenie. Niech x będzie ciągiem symboli terminalnych, A zaś symbolem nieterminalnym. Pytanie brzmi: Czy x może zostać wyprodukowany z A za pomocą G? Pytanie to będziemy skrótowo oznaczać: $MEMBER(x, A)$.
2. Pytania o równoważność. Niech H będzie gramatyką bezkontekstową. Pytanie brzmi: Czy H jest równoważna G? Skrótowo: $EQUIV(H)$. Odpowiedź negatywna na pytanie o równoważność stowarzyszona jest z kontrprzykładem z symetrycznej różnicy zbioru generowanego przez H i zbioru generowanego przez G.

W momencie inicjacji algorytm umieszcza wszystkie możliwe produkcje G w zbiorze hipotetycznych produkcji — P. Następnie zadaje pytanie $EQUIV(H)$ gdzie $H = \{A, V, S, P\}$. Jeśli $EQUIV(H) = 1$, to L^{cf} podaje na wyjściu gramatykę H. Jeśli $EQUIV(H) = 0$, to podany przez nauczyciela kontrprzykład powoduje, że ze zbioru produkcji P usunięta zostaje przynajmniej jedna produkcja. Łatwo zauważyć, że H do momentu osiągnięcia równoważności z G zawsze będzie nadzbiorem G. Kontrprzykład będzie więc zawsze pochodził ze zbioru $H - G$.

Pozostaje wyjaśnić jak przebiega analiza kontrprzykładu t zakończona decyzją, która z produkcji powinna zostać wyeliminowana ze zbioru P. Otóż L^{cf} znajduje drzewo derywacyjne t z symbolu startowego S w gramatyce H. Załóżmy, że $t = t_1 t_2$ oraz, że drzewo derywacyjne ciągu t w gramatyce H wygląda tak:

$$\begin{array}{ccccc} & & S & & \\ & \swarrow & & \searrow & \\ A & & & & B \\ \downarrow & & & & \downarrow \\ t_1 & & & & t_2 \end{array}$$

Rysunek 3. Drzewo derywacyjne słowa t z S w gramatyce H

W takiej sytuacji L^{cf} zadaje pytanie: $MEMBER(t_1, A)$. Jeśli odpowiedź jest negatywna, to t_1 może być derywowane z A w H, ale nie w G. L^{cf} usuwa z P produkcję $A \longrightarrow t_1$. Jeśli zaś odpowiedź jest pozytywna, to L^{cf} zadaje pytanie $MEMBER(t_2, B)$. Jeśli na to pytanie odpowiedź jest negatywna, to L^{cf} usuwa z P produkcję $B \longrightarrow t_2$. Jeśli odpowiedź jest pozytywna znaczy to, że do G należą produkcje $A \longrightarrow t_1$ oraz $B \longrightarrow t_2$, nie zaś $S \longrightarrow t$. Pozostaje więc usunąć z P produkcję $S \longrightarrow AB$. W ten sposób algorytm L^{cf} dochodzi do wyznaczenia wszystkich i tylko produkcji G.

Rozwiązanie opisane powyżej jest problematyczne. Największą trudność sprawia fakt, że pytanie $EQUIV(H)$ jest nierozstrzygalne (zob.(Sudkamp, 1988), (Hopcroft et al., 2001)). W związku z tym nie można liczyć na realistyczną implementację nauczyciela poprawnie odpowiadającego na to pytanie. Kolejny problem związany jest ze złożonością kontrprzykładów. W ogólnym przypadku nie ma rekurencyjnego oszacowania rozmiaru wymaganych kontrprzykładów.

Ponadto założenie o tak dużej wiedzy początkowej ucznia (choćby znajomość symboli nieterminalnych) jest zbyt mocne by adekwatnie opisywać naturalny proces uczenia się języka. To samo tyczy się zapytań o symbole nieterminalne.

Podsumowując algorytm L^{cf} jest mniej realistyczny niż L^* z powodów, których z zaproponowanego przez Angluin modelu nie można wyeliminować.

Algorytm LA Rozwinięcie koncepcji skończonego uczenia się zaproponował Yasubumi Sakakibara (Sakakibara, 1990). Jest to bezpośrednie przełożenie algorytmu L^* opisanego w rozdziale 5. na problem uczenia się gramatyk bezkontekstowych. Zaproponowany algorytm uczy się danej gramatyki bezkontekstowej G na podstawie tzw. danych strukturalnych. Są nimi szkielety drzew derywacyjnych gramatyki G. Szereg następujących faktów pozwoli przybliżyć w zarysie zasadę działania zapowiadanego algorytmu.

1. Zbiór drzew derywacyjnych danej gramatyki bezkontekstowej jest regularnym zbiorem drzew.
2. Regularny zbiór drzew to zbiór drzew rozpoznawany przez pewien automat drzewiasty.
3. Procedura tworzenia z drzew derywacyjnych ich opisów strukturalnych zachowuje regularność zbioru.
4. Problem uczenia się gramatyki bezkontekstowej z opisów strukturalnych jest więc redukowalny do problemu uczenia się pewnego automatu drzewiastego.

Należy podkreślić, iż celem „kształcenia” naszego algorytmu L^* w wersji dla gramatyk bezkontekstowych nie jest język bezkontekstowy, lecz pewna gramatyka bezkontekstowa. Wiadomo bowiem, że dla każdego języka bezkontekstowego istnieje nieskończenie wiele gramatyk go opisujących.

6.1. Gramatyki bezkontekstowe

Od gramatyk regularnych, którymi zajmowaliśmy się w poprzednim rozdziale gramatyki bezkontekstowe różnią się postacią produkcji. Dla przypomnienia:

1. Produkcje w gramatykach regularnych:

$$Y \longrightarrow_G \alpha Z \text{ lub } Y \longrightarrow_G \alpha \text{ dla } Y,\ Z \in \Sigma,\ \alpha \in A^*.$$

2. Produkcje w gramatykach bezkontekstowych:

$$Y \ \longrightarrow_G \beta \text{ dla } Y \in \Sigma,\ \beta \in (A \cup \Sigma)^*.$$

Definicja 18. *Niech G_1, G_2 będą gramatykami bezkontekstowymi, wtedy:*

$$G_1 \equiv G_2 \text{ wtw } L(G_1) = L(G_2).$$

Czyli: Gramatyki bezkontekstowe są równoważne wtedy i tylko wtedy, gdy generują ten sam język.

Przykład Gramatyka bezkontekstowa G^{PAL}.

$$G^{PAL} = \{A, \Sigma, R, P\}, \text{ gdzie:}$$

1. $A = \{0, 1\}$ jest zbiorem symboli terminalnych,
2. $\Sigma = \{R\}$ jest zbiorem symboli nieterminalnych,
3. R jest nieterminalnym symbolem początkowym,
4. P jest zbiorem produkcji takim, że:

$$P = \{R \longrightarrow \lambda, R \longrightarrow 0, R \longrightarrow 1, R \longrightarrow 0R0, R \longrightarrow 1R1\}.$$

Nietrudno zauważyć, że gramatyka G^{PAL} generuje wszystkie palindromy nad alfabetem $\{0, 1\}$.

6.2. Drzewa derywacyjne gramatyki bezkontekstowej

Niech $G = (A, \Sigma, P, S)$. Każdemu ciągowi z języka generowanego przez gramatykę G można przypisać drzewo derywacyjne. Drzewa derywacyjne gramatyki G to drzewa spełniające następujące warunki:

1. W korzeniu drzewa znajduje się symbol S.
2. Każdy wierzchołek nie będący liściem jest etykietowany symbolem nieterminalnym ze zbioru Σ.
3. Każdy liść jest etykietowany symbolem terminalnym ze zbioru A lub λ.
4. Jeśli wierzchołek nie będący liściem jest etykietowany symbolem B zaś jego synowie etykietowani są kolejno od lewej strony symbolami $X_1, X_2, \ldots, X_k$, to $B \longrightarrow X_1X_2 \ldots X_k$ jest produkcją w P.

Przykład Aby podać przykład drzewa derywacyjnego pewnego ciągu w danej gramatyce bezkontekstowej możemy wykorzystać gramatykę G^{PAL} zaprezentowaną na początku tego rozdziału. Wybierzmy w tym celu prosty przykład palindromu nad alfabetem $\{0, 1\}$: 01110. Ciąg ten ma w gramatyce G^{PAL} drzewo derywacyjne następującej postaci:

$$\begin{array}{ccc}
 & R & \\
\swarrow & \downarrow & \searrow \\
0 & R & 0 \\
\swarrow & \downarrow & \searrow \\
1 & R & 1 \\
 & \downarrow & \\
 & 1 &
\end{array}$$

Rysunek 4. Drzewo derywacyjne ciągu 01110 w gramatyce G^{PAL}

Oczywiście, elementy zbioru A, czyli symbole terminalne, znajdziemy w etykietach liści, zaś elementy zbioru Σ, czyli symbole nieterminalne, w pozostałych węzłach drzewa derywacyjnego.

W dalszych rozważaniach będzie nas interesował zbiór wszystkich drzew derywacyjnych gramatyki G, zwany dalej $D_S(G)$, gdzie S jest symbolem początkowym gramatyki G. Będziemy dalej pomijać indeks dolny S. Pisząc $D(G)$ mamy na myśli zbiór drzew derywowanych w gramatyce G z symbolu początkowego S.

Operacja

Definicja 19. *Niech V będzie skończonym zbiorem symboli funkcyjnych takim, że:*

$$V = V_0 \cup V_1 \cup \ldots \cup V_n, \text{ gdzie:}$$

1. *V_0 jest zbiorem stałych, odpowiadającym zbiorowi symboli terminalnych gramatyki;*
2. *V_i jest zbiorem symboli funkcyjnych arności i, dla $i = 1, \ldots, n$, odpowiadających symbolom nieterminalnym gramatyki.*

Etykietami węzłów w drzewie derywacyjnym są elementy zbioru V. Etykietami liści mogą być jedynie elementy zbioru V_0, czyli stałe funkcyjne. Skończone drzewa nad V mogą być identyfikowane z dobrze zbudowanymi termami nad V. Jako takie mogą być zapisywane w formie liniowej.

Definicja 20. *Niech V^T będzie zbiorem wszystkich skończonych drzew nad V. Niech $\$ \notin V$ będzie nowym symbolem funkcyjnym o arności 0. Wtedy:*

$$V_\$^T = \{t \in (V \cup \{\$\})^T : t \text{ zawiera dokładnie jeden symbol } \$\}.$$

Definicja 21. *Niech $s \in V_\$^T$ oraz $t \in (V^T \cup V_\$^T)$. Definiujemy operację podmieniania liścia etykietowanego symbolem $\$$ w drzewie s przez drzewo t:*

$$s\#t(x) = \begin{cases} s(x) & \text{jeśli } x \in s \text{ oraz } s(x) \neq \$ \\ t(y) & \text{jeśli } x = z \cdot y, s(z) = \$, y \in t. \end{cases}$$

Przykład Poniżej znajduje się ilustracja operacji podmieniania liścia etykietowanego \$ w drzewie s drzewem t.

$$\begin{array}{ccccc} & & R & & \\ & \swarrow & \downarrow & \searrow & \\ 0 & & R & & 0 \\ & \swarrow & \downarrow & \searrow & \\ 1 & & \underline{\$} & & 1 \end{array} \qquad\qquad \begin{array}{ccccc} & & \underline{R} & & \\ & \swarrow & \downarrow & \searrow & \\ 0 & & R & & 0 \\ & & \downarrow & & \\ & & 1 & & \end{array}$$

Drzewo s Drzewo t

$$\begin{array}{ccccc} & & R & & \\ & \swarrow & \downarrow & \searrow & \\ 0 & & R & & 0 \\ & \swarrow & \downarrow & \searrow & \\ 1 & & \underline{R} & & 1 \\ & \swarrow & \downarrow & \searrow & \\ 0 & & R & & 0 \\ & & \downarrow & & \\ & & 1 & & \end{array}$$

Drzewo $s\#t$

Rysunek 5. Konstrukcja drzewa $s\#t$

Definicja 22. *Niech $S \subseteq V^T_\$$ oraz $T \subseteq (V^T \cup V^T_\$)$. Definiujemy:*

$$S\#T = \{s\#t : s \in S, t \in T\}.$$

Szkielety drzew derywacyjnych Symbolem $K(D(G))$ będziemy określać zbiór szkieletów drzew derywacyjnych gramatyki G. Szkielety różnią się od drzew derywacyjnych jedynie tym, że w miejscach symboli nieterminalnych występuje symbol σ. Struktura i etykiety liści drzewa derywacyjnego pozostają bez zmian. Każdy taki szkielet drzewa derywacyjnego gramatyki G nazywać będziemy **opisem strukturalnym tej gramatyki**.

Definicja 23. *Niech $Sk = \{\{\sigma\} \times \{1, 2, \ldots, m\}\}$, m jest maksymalną arnością symboli funkcyjnych w Sk.*

Definicja 24. *Niech $t \in V^T$. Definiujemy opis strukturalny (szkielet) drzewa t:*

$$s(t)(x) = \begin{cases} t(x) & \text{jeśli } x \text{ jest liściem w } t \\ \sigma & \text{jeśli } x \text{ jest różnym od liścia węzłem w drzewie } t. \end{cases}$$

Definicja 25. *Niech T będzie zbiorem drzew. Odpowiadający mu zbiór szkieletów to:*

$$K(T) = \{s(t) : t \in T\}.$$

Przykład Opisem strukturalnym omawianej gramatyki G^{PAL} odpowiadającym przedstawionemu wcześniej drzewu derywacyjnemu będzie:

$$\begin{array}{ccc} & \sigma & \\ \swarrow & \downarrow & \searrow \\ 0 & \sigma & 0 \\ \swarrow & \downarrow & \searrow \\ 1 & \sigma & 1 \\ & \downarrow & \\ & 1 & \end{array}$$

Rysunek 6. Opis strukturalny (szkielet) gramatyki G odpowiadający ciągowi 01110

Definicja 26. *Niech G_1, G_2 będą gramatykami bezkontekstowymi, wtedy:*

$$G_1 \equiv_{str} G_2 \ wtw \ K(D(G_1)) = K(D(G_2)).$$

Czyli: Dwie gramatyki bezkontekstowe są strukturalnie równoważne wtedy i tylko wtedy, gdy odpowiadające im zbiory szkieletów drzew derywacyjnych są identyczne.

Rozważanie strukturalnej równoważności gramatyk pozwoli nam ominąć nierozstrzygalny problem zwykłej równoważności gramatyk bezkontekstowych.

6.3. Automaty drzewiaste

Algorytm L^* w wersji dla gramatyk bezkontekstowych identyfikuje nieskończony zbiór szkieletów drzew derywacyjnych gramatyki bezkontekstowej dzięki konstrukcji automatu drzewiastego odpowiadającego poszukiwanemu zbiorowi. Automat drzewiasty odpowiadający danemu zbiorowi szkieletów $K(D(G))$ to taki automat, który akceptuje wszystkie i tylko elementy zbioru $K(D(G))$. Poniżej znajduje się definicja deterministycznego automatu drzewiastego.

Definicja 27. *Deterministyczny automat drzewiasty (inaczej: DFTA), jest to:*

$$A = [(Q, F_1, \ldots, F_n, q_1, \ldots, q_m), Q_f], \ gdzie:$$

— $F_i : Q^{k_i} \mapsto Q$, *dla* $i = 1, \ldots, n$;

— $q_1, \ldots, q_m \in Q$; $Q_f \subseteq Q$;

— *Niech t będzie termem nad słownikiem* $(\{f_1, \ldots, f_n\}, k, \{c_1, \ldots, c_m\})$ $wart_A(t)$ *jest wartością termu t w A.*

Funkcja przejścia w DFTA A:

1. $wart_A(c_i) = Q_i$;

2. $wart_A(f_i(t_1, \ldots, t_{k_i})) = F_i(wart_A(t_1), \ldots, wart_A(t_{k_i}))$.

Fakt 1. *DFTA A akceptuje term t wtedy i tylko wtedy, gdy* $wart_A(t) \in Q_f$.

6.4. Zapytania algorytmu LA

Algorytm LA (uczeń) w sposób istotny korzysta ze wskazówek nauczyciela, który odpowiada poprawnie na zadawane przez ucznia pytania. Tak jak w przypadku algorytmu LA dla języków regularnych mamy tutaj do czynienia z dwoma rodzajami pytań: o należenie oraz o równoważność. Pierwsza klasa pytań dotyczy poszczególnych szkieletów drzew. Pytania są postaci: Czy dany szkielet s należy do zbioru szkieletów drzew poszukiwanej gramatyki? Nauczyciel odpowiada TAK, jeśli s należy do tego zbioru, w przeciwnym przypadku odpowiada przecząco. Drugi rodzaj pytań dotyczy automatu drzewiastego B skonstruowanego przez algorytm LA. Pytania te mają następującą postać: Czy zbiór szkieletów drzew, który jest akceptowany przez automat B jest identyczny z poszukiwanym zbiorem szkieletów? Nauczyciel odpowiada TAK, jeśli zbiory te są identyczne, w przeciwnym przypadku pada odpowiedź negatywna. Jest ona każdorazowo stowarzyszona z tzw. kontrprzykładem — ciągiem należącym do symetrycznej różnicy poszukiwanego zbioru szkieletów oraz zbioru odpowiadającego skonstruowanemu przez ucznia automatowi drzewiastemu. Warto zauważyć, że tak jak w przypadku algorytmu uczącego się języków regularnych różni nauczyciele będą różnili się między sobą w odpowiedziach na pytania drugiego rodzaju, nauczyciel może bowiem wybrać dowolny kontrprzykład spośród nieskończenie wielu możliwych. Poniżej opis obu kategorii pytań:

1. Zapytania o należenie danej struktury do szukanej gramatyki (G_u):

$$T(s) = \begin{cases} 1 & \text{jeśli } s \in K(D(G_u)) \\ 0 & \text{jeśli } s \notin K(D(G_u)). \end{cases}$$

2. Zapytania o równoważność struktury wyjściowej (G') algorytmu i struktury szukanej (G_u):

$$R(A) = \begin{cases} 1 & \text{jeśli } K(D(G_u)) = K(D(G')) \\ \langle 0, s \rangle, s \in K(D(G_u)) \div K(D(G')) & \text{jeśli } K(D(G_u)) \neq K(D(G')). \end{cases}$$

6.5. Tablica obserwacyjna

Podczas swego działania algorytm LA dysponuje danymi na temat skończonego zbioru szkieletów drzew derywacyjnych. Szkielety te są sklasyfikowane według odpowiedzi nauczyciela na pytania o należenie. Przechowywaną przez algorytm LA informację można przedstawić w postaci dwuwymiarowej tabeli, zwanej tablicą obserwacyjną.

Definicja 28. *Zbiór drzew A nazywamy domkniętym na poddrzewa wtedy i tylko wtedy, gdy jeśli s należy do A, to wszystkie poddrzewa drzewa s o głębokości niemniejszej niż 1 należą do zbioru A.*

Definicja 29. *Zbiór drzew B nazywamy domkniętym na \$–prefiksy ze względu na zbiór A wtedy i tylko wtedy, gdy jeśli $e \in B - \{\$\}$, to istnieje $e' \in B$, $s_1, \ldots, s_{k-1} \in A \cup \Sigma$ takie, że: $e = e' \# \sigma(s_1, \ldots, s_{i-1}, \$, s_i, \ldots, s_{k-1})$.*

Definicja 30. ***Tablica obserwacyjna*** *jest to (S, E, T), gdzie:*

1. *S jest niepustym skończonym zbiorem szkieletów drzew derywacyjnych o głębokości $\geqslant 1$;*
2. *$X(S) = \{\sigma(u_1, \ldots, u_k) : \ \sigma \in Sk; \ u_1, \ldots, u_k \in S \cup \Sigma, \sigma(u_1, \ldots, u_k) \notin S, \ k \geqslant 1\}$;*
3. *E jest niepustym skończonym zbiorem szkieletów ze zbioru $(Sk \cup \Sigma)^T_\$$ domkniętym na \$–prefiksy ze względu na zbiór S;*
4. *T jest skończoną funkcja $(E\#(S \cup X(S))) \to \{0, 1\}$, gdzie $T(s) = 1$ wtedy i tylko wtedy, gdy s jest opisem strukturalnym poszukiwanej gramatyki G.*

(S, E, T) można przedstawić za pomocą dwuwymiarowej tablicy:

1. Wiersze oznaczone elementami zbioru $(S \cup X(S))$.
2. Kolumny oznaczone elementami zbioru E.
3. Wartość w komórce o współrzędnych (s, e), gdzie $s \in (S \cup X(S))$, $e \in E$, jest równa $T(s\#e)$.
4. Niech $s \in (S \cup X(S))$, wtedy $row(s)$ jest wektorem złożonym z wartości $T(s\#e)$, dla wszystkich $e \in E$.

	T	e	E
S	$s\ldots$	$\vdots$ $1(= T(s\#e))$	
$X(S)$	s_1		

Tabela 3. Tablica obserwacyjna

Definicja 31. *Tablica obserwacyjna (S, E, T) jest **domknięta** wtedy i tylko wtedy, gdy:*

$$\forall t \in X(S)\ \exists s \in S(row(t) = row(s)).$$

Definicja 32. *Tablica obserwacyjna (S, E, T) jest **spójna** wtedy i tylko wtedy, gdy:*

$$\forall s_1, s_2 \in S, \sigma \in Sk, u_1, \ldots, u_k \in S \cup \Sigma$$

$$[(row(s_1) = row(s_2)) \Rightarrow (row(\sigma(u_1, \ldots, u_{i-1}, s_1, u_i, \ldots, u_{k-1}))$$

$$=$$

$$row(\sigma(u_1, \ldots, u_{i-1}, s_2, u_i, \ldots, u_{k-1})].$$

6.6. Konstrukcja skończonego automatu drzewiastego przy użyciu tablicy obserwacyjnej

Algorytm LA używa tablicy obserwacyjnej do stworzenia odpowiedniego deterministycznego automatu drzewiastego.

Definicja 33. *Niech (S, E, T) będzie domkniętą i spójną tablicą obserwacyjną. Możemy zdefiniować skończony automat drzewiasty $A(S, E, T)$ nad $Sk \cup \Sigma$ taki, że:*

1. *$Q = \{row(s) : s \in S\}$ jest zbiorem stanów automatu A;*
2. *$F = \{row(s) : s \in S \wedge T(s) = 1\}$ jest zbiorem stanów akceptujących;*
3. *$\delta(row(s_1), \ldots, row(s_k)) = row(\sigma(s_1, \ldots, s_k))$ oraz $\delta(a) = a$, dla $a \in \Sigma$ jest funkcją przejścia automatu A.*

6.7. Algorytm LA

Algorytm LA działa na podstawie pozytywnych i negatywnych danych na temat szkieletów drzew derywacyjnych. Konstruuje początkową tablicę, następnie dąży do jej uspójnienia i domknięcia stosując zapytania o należenie. Gdy tablica stanie się domknięta i spójna, konstruuje z niej automat drzewiasty na sposób opisany wcześniej. Wysuwa zapytanie o ekstensjonalną równoważność skonstruowanego automatu i automatu reprezentującego poszukiwany zbiór szkieletów. Jeśli otrzymuje odpowiedź pozytywną, to kończy pracę podając na wyjściu skonstruowany automat. Jeśli odpowiedź nauczyciela jest negatywna, to otrzymany kontrprzykład dodawany jest do tablicy obserwacyjnej, po czym algorytm wraca do poleceń uspójnienia i domknięcia tablicy. Poniżej znajduje się ścisły opis algorytmu LA.

```
begin
S := ∅;
E := $;
G := ({S}, Σ, ∅, S)
Zapytaj o poprawność G
if (odpowiedź=TAK) then
   RETURN G
else dodaj kontrprzykład t i wszystkie jego poddrzewa o
głębokości przynajmniej 1 do S
Stosując zapytania o należenie utwórz tablicę obserwacyjną
(S, E, T):
   kolumna etykietowana $
   wiersze etykietowane kontrprzykładem t i wszystkimi jego
   poddrzewami o głębokości ⩾ 1
   komórka o współrzędnych (s, $) etykietowana wartościami T(s),
   dla każdego s ∈ S
repeat
  while ((S, E, T) nie jest domknięta lub nie jest spójna) do
     { if ((S, E, T) nie jest spójna) then
        { znajdź s1, s2 ∈ S, e ∈ E, u1, ..., u(k-1) ∈ S ∪ Σ
          oraz i ∈ ℕ takie, że:
          row(s1) = row(s2) i
          T(e#σ(u1, ..., u(i-1), s1, ui, ..., u(k-1)))
          ≠ T(e#σ(u1, ..., u(i-1), s2, ui, ..., u(k-1)))
          dodaj (e#σ(u1, ..., u(i-1), $, ui, ..., u(k-1))) do zbioru E
          rozszerz T do E#(S ∪ X(S))
          używając zapytań o należenie
        }
       if ((S, E, T) nie jest domknięta) then
        { znajdź s1 ∈ X(S) takie, że:
          dla dowolnego s ∈ S row(s1) ≠ row(s)
          dodaj s1 do zbioru S
          rozszerz T do E#(S ∪ X(S))
          używając zapytań o należenie
        }
     }
   G:=G(A(S, E, T))
   Zapytaj o poprawność G
   if odpowiedź=⟨0, kontrprzykład t⟩ then
     { dodaj kontrprzykład t i wszystkie jego poddrzewa
```

```
        o głębokości ⩾ 1 do S
        rozszerz T do E#(S ∪ X(S))
        używając zapytań o należenie.
      }
until odpowiedź=TAK
RETURN G
end.
```

Warto zaznaczyć, że dla dowolnej gramatyki bezkontekstowej G_U algorytm LA zatrzymuje się podając na wyjściu gramatykę bezkontekstową strukturalnie równoważną gramatyce G_U.

Co więcej, jeśli n jest liczbą stanów minimalnego automatu drzewiastego dla zbioru opisów strukturalnych G_U, zaś m jest górnym ograniczeniem długości kontrprzykładów, to całkowity czas działania algorytmu LA jest wielomianowy względem n i m (Sakakibara, 1990).

6.8. Podsumowanie

Powyższy model jest pierwszym, który daje efektywną możliwość wyuczalności całej klasy gramatyk bezkontekstowych. Wcześniejsze próby dotyczyły pewnych podklas języków bezkontekstowych (Crespi–Reghizzi, 1972) lub były nieefektywne (Levy Joshi, 1978), (Fass, 1983). Poniżej zastanowimy się, czy model wyuczalności gramatyk bezkontekstowych opisany w tym rozdziale może mieć znaczenie w eksplikacji naturalnego procesu uczenia się.

Zaletą omawianego modelu jest to, że nie zakłada on „wrodzonej" wiedzy ucznia o symbolach nieterminalnych. Nie zakłada również, że symbole nieterminalne są uczniowi *explicite* podawane. Cechy te sprawiają, że w ograniczonym zakresie model Sakakibary odpowiada naturalnemu procesowi nabywania umiejętności językowych. Dziecko najpierw uczy się budować poprawnie ciągi, dużo później zaś dowiaduje się co jest nazwą a co zdaniem. Alternatywą mogłoby być przypuszczenie, że symbole nieterminalne (kategorie gramatyczne) są człowiekowi wrodzone. Jest to jednak założenie bardzo ryzykowne i zbyt daleko idące. Okazuje się bowiem, iż istnieje możliwość przyswojenia gramatyki bezkontekstowej bez uprzedniej znajomości jej symboli nieterminalnych.

Algorytm LA uczy się gramatyki bezkontekstowej w postaci strukturalnej. Nie operuje na ciągach, a raczej na termach zbudowanych z symboli terminalnych i symboli funkcyjnych różnej arności. Zakłada więc wcześniejsze wyuczenie się ciągów terminalnych, z których będzie można później korzystać budując zdania. To założenie również wydaje się zgodne z naturalnym procesem uczenia się języka. Najpierw dziecko przyswaja poszczególne słowa ostensywnie wiążąc je z przedmiotami, dopiero później buduje konstrukcje składniowe. Ponadto eliminacja symboli nieterminalnych z procesu uczenia się języka sprzyja adekwatności modelu. Wydaje się, że wiedza jaki konkretnie nieterminalny symbol reprezentuje dany ciąg nie jest dziecku udostępniana.

Właśnie ten strukturalny aspekt uczenia się gramatyk bezkonstekstowych rodzi pewną trudność przy porównywaniu z naturalnym procesem przyswajania języka. Model Angluin, zaprezentowany w rozdziale 5. w przypadku zapytań o należenie przypisywał nauczycielowi i uczniowi realistyczne role. W modelu Sakakibary również ten rodzaj zapytań staje się nierealistyczny. Otóż dotyczą one tutaj szkieletów gramatyk. Pytanie stawiane jest w formie: Czy dany szkielet należy do zbioru opisów strukturalnych? Trudno znaleźć jego odpowiednik w naturalnym procesie uczenia się języka.

Rozwiązanie zaproponowane przez Sakakibarę może być użyteczne do opisu związków pomiędzy uczeniem się semantyki a uczeniem się składni. Przyswajanie szkieletów drzew derywacyjnych umożliwia przyporządkowanie wyrażeniom językowym struktury, która pozwala wyznaczyć znaczenia tych wyrażeń. Zakładając bowiem zasadę kompozycyjności (zob. (Janssen, 2003)) istnieje odpowiedniość pomiędzy strukturą składniową wyrażenia a jego strukturą semantyczną. Zatem szkielety drzew derywacyjnych reprezentują wyrażenia nie tylko pod względem składniowym, lecz również zawierają część informacji o znaczeniu (semantyce) wyrażeń.

7. Uczenie się semantyki

W poprzednich rozdziałach omówiliśmy modele wyuczalności rozmaitych klas języków. Wyuczalność ta dotyczyła jedynie składniowego aspektu języka. Naturalnym kierunkiem dalszych rozważań nad wyuczalnością jest kwestia uczenia się semantyki języka. Jest to problem stosunkowo słabo zbadany, poświęcono mu dotychczas niewiele uwagi. Jako przykład konstrukcji semantycznych badanych pod względem wyuczalności podać można konstrukcje kwantyfikatorowe. Problem wyuczalności konstrukcji kwantyfikatorowych po raz pierwszy podjęty został przez Johana van Benthema (van Benthem, 1986), a następnie rozważany w kilku innych pracach (zob. (Clark, 1996), (Florêncio, 2002), (Tiede, 1999)). Były to próby zastosowania modelu uczenia się składni zaprezentowanego w niniejszej pracy do problemu uczenia się semantyki języka.

W jaki sposób przyswajamy znaczenie wyrażeń kwantyfikatorowych w języku naturalnym? Znaczenie wyrażeń języka naturalnego można utożsamić z procedurami rozpoznawania ich ekstensji (Moschovakis, 1994), (Szymanik, 2004). Rozsądnym wydaje się przyjęcie, że przyswajanie znaczenia wyrażeń kwantyfikatorowych odbywa się poprzez uczenie się procedur wyznaczania ich denotacji. Opierając się na definicji kwantyfikatora monadycznego zaproponowanej przez Lindströma (Lindström, 1966), możemy zakodować klasy modeli skończonych odpowiadających danemu kwantyfikatorowi. Takie klasy modeli skończonych mogą być reprezentowane przez odpowiednie języki (Mostowski, 1998). Jako takie, mogą podlegać wyuczalności w tym samym sensie, co języki w ujęciu składniowym.

Poniżej znajduje się kilka faktów dotyczących klas kwantyfikatorów monadycznych i modeli obliczeń rozpoznających te klasy.

— Kwantyfikator Q jest definiowalny w logice pierwszego rzędu $\iff$ L_Q jest akceptowany przez pewien acykliczny automat skończony (van Benthem, 1986).
— Monadyczny kwantyfikator Q jest definiowalny w logice podzielności $FO(D_\omega)$ $\iff$ L_Q jest akceptowany przez pewien automat skończony (Mostowski, 1998).
— Kwantyfikator typu (1) jest półliniowy $\iff$ L_Q jest akceptowany przez automat ze stosem (van Benthem, 1986).
— W języku naturalnym istnieje wiele kwantyfikatorów niedefiniowalnych środkami gramatyk bezkontekstowych.

Oto kilka rezultatów połączenia idei kodowania konstrukcji kwantyfikatorowych oraz wyuczalności składniowej.

— Klasy FO, $FO(D_\omega)$ oraz półliniowych kwantyfikatorów nie są identyfikowalne w granicy z danych pozytywnych, ale są identyfikowalne w granicy z danych pozytywnych i negatywnych.

— Istnieją podklasy klas wymienionych powyżej, które są identyfikowalne w granicy przy pomocy tekstu, np. zbiór kwantyfikatorów *left upward monotone* (Tiede, 1999).
— Kwantyfikatory definiowalne środkami $FO(D_\omega)$ są wyuczalne przy użyciu algorytmu L^*.
— Kwantyfikatory półliniowe typu (1) są wyuczalne przy użyciu LA.

Kwestia adekwatności narzędzi teorii uczenia się składni do opisu uczenia się semantyki jest oczywiście dyskusyjna. Wątpliwości mogą dotyczyć adekwatności tego aparatu do kształtowania kompetencji semantycznej. W obrębie tej kompetencji wyróżnić można kilka powiązanych ze sobą mechanizmów: zdolność do sprawdzania wartości logicznej zdania w danym modelu, zdolność do rozpoznawania relacji wynikania pewnego zdania z innych zdań oraz zdolność do generowania adekwatnych opisów, czyli zdań prawdziwych w danym modelu (Mostowski, 1994), (Bucholc, 2004), (Mostowski Wojtyniak, 2004). Można więc kompetencję semantyczną przedstawić w następujący sposób.

	zdolność sprawdzania wartości logicznej wejście: model M oraz zdanie φ; Czy $M \models \varphi$?
Kompetencja semantyczna	zdolność do rozpoznawania relacji wynikania wejście: zdania φ, ψ; Czy $\varphi \models \psi$?
	zdolność do generowania adekwatnych opisów wejście: model M; znajdź zdanie φ takie, że $M \models \varphi$

Rysunek 7. Kompetencja semantyczna

Faktem jest, że rozpatrywane wcześniej modele wyuczalności nie są wrażliwe na rozróżnienie pomiędzy rozpoznawaniem a generowaniem adekwatnych opisów. Można argumentować , że istnieje wzajemny przekład pomiędzy automatami a gramatykami (zob. (Hopcroft et al., 2001)). Nie wydaje się jednak aby taka redukcja poprawnie zdawała sprawę z procesu przyswajania konstrukcji semantycznych. Wydaje się, że najpierw człowiek rozumie konstrukcje semantyczne, a dopiero później sam zaczyna je konstruować. Generowanie opisów jest prawdopodobnie bardziej skomplikowane niż testowanie ich trafności.

8. Wnioski

Zaprezentowane w pracy modele wyuczalności języka w różnym stopniu odpowiadają naturalnemu procesowi przyswajania języka przez człowieka. Model identyfikacji w granicy szczególnie dobrze oddaje to, że uczeń nigdy nie dowiaduje się kiedy przyswojona przez niego gramatyka jest „prawidłowa". Inną zaletą metodologiczną modelu Golda jest fakt, że pojęcie identyfikowalności dotyczy określonych klas języków. Założenie to pozwala wnioskować o strukturze języka naturalnego (jego miejscu w hierarchii języków) na podstawie struktury kompetencji językowej.

Model wyuczalności przez zapytania podkreśla rolę nauczyciela w procesie uczenia się. Jej istotność wykazana została pośrednio przez Golda, w dowodzie nieidentyfikowalności nadskończonych klas języków z danych pozytywnych. Eksplikacja pojęcia nauczyciela stwarza możliwość formalnego opisu i modelowania roli nauczyciela w procesie uczenia się. Daje również ciekawą wskazówkę ogólnopoznawczą: tzw. informacja negatywna okazuje się niezbędna dla procesów poznawczych.

Algorytm Sakakibary ukazuje możliwość rozszerzania zasięgu modeli uczenia się oraz wskazuje na możliwość korzystania z drzew derywacyjnych w badaniu wyuczalności języka, co może okazać się użyteczne w modelowaniu uczenia się semantyki.

Ważną cechą wszystkich zaprezentowanych modeli jest to, że nie zakładają one istnienia wrodzonego zrębu wszystkich gramatyk, w sensie gramatyki uniwersalnej postulowanej przez Chomsky'ego. Okazuje się więc, że przyjęcie algorytmicznego modelu kompetencji językowej nie wymaga postulowania części wspólnej wszystkich języków naturalnych. Ich podobieństwo może być spowodowane przynależnością do takiej klasy, którą wrodzony mechanizm uczenia się jest w stanie identyfikować, na przykład klasy wszystkich języków bezkontekstowych.

Poniżej wymieniamy narzucające się kierunki dalszych badań dotyczących modeli wyuczalności.

1. Sprawdzenie adekwatności modeli wyuczania składni do opisu uczenia się semantyki.
2. Empiryczne zbadanie adekwatności przyjętego w rozdziale 7. modelu kompetencji semantycznej.
3. Empiryczne zbadanie psychologicznej adekwatności modeli uczenia się języka zarówno w aspekcie składniowym, jak i semantycznym.
4. Uszeregowanie konstrukcji semantycznych ze względu na wyuczalność.
5. Zbadanie użyteczności algorytmu LA do modelowania procesu uczenia się semantyki kompozycyjnej.

Literatura

ANGLUIN, D. (1980): *Inductive inference of formal languages from positive data*. Information and Control 45(1980), 117–135.

——— (1987): *Learning regular sets from queries and counterexamples*. Information and Computation 75/2(1987), 87–106.

VAN BENTHEM, J. (1986): *Essays in logical semantics*, Reidel Publishing Company, 1986.

BUCHOLC, P. (2004): *Kompetencja logiczna a poprawność logiczna. Analiza na przykładzie terminów pustych*, [w:] Kognitywistyka. O umyśle umyślnie i nieumyślnie, (J. Szymanik , M. Zajenkowski, red.), Koło Filozoficzne przy Kolegium MISH, 2004, Praca magisterska w Instytucie Filozofii UW, promotor: prof. Marcin Mostowski, 2001, str. 1–28.

CHOMSKY, N. (1957): *Syntactic structures*, Mouton, 1957.

——— (1959): *Review of Verbal Behavior by B.F. Skinner*. Language 35/1(1959), 26–57.

——— (1965): *Aspects of the theory of syntax*, The M.I.T. Press, 1965.

——— (1965): *Cartesian linquistics*, Harper & Row, 1965.

——— (1968): *Language and mind*, Harcourt Brace, 1968.

——— (1967): *Recent contributions to the theory of innate ideas*. In Syntheses 17, 2–11.

——— (1985): *The logical structure of linguistic theory*, University of Chicago Press, 1985.

——— (1995): *The minimalist program*, The MIT Press, 1995.

CICHOSZ, P. (2000): *Systemy uczące się*, Wydawnictwa Naukowo–Techniczne, 2000.

CLARK, R. (1996): *Learning first–order quantifiers denotations. An essay in semantic learnability*. IRCS Technical Report, University of Pennsylvania, 19–96.

CUTLAND, N. (1980): *Computability. An introduction to recursive function theory*, Cambridge University Press, 1980.

CRESPI–REGHIZZI, S. (1972): *An effective model for grammatical inference*, [w:] Information Processing 71, (B. Gilchrist, red.), Elsevier Science Publishers, 1972, str. 524–529.

FASS, L. F. (1983): *Learning context–free languages from their structured sentences*. SIGACT News 15/3(1983), 24–35.

FLORÊNCIO, C. C. (2002): *Learning generalized quantifiers*. Proceedings of Seventh ESSLLI Student Session, 2002, 1–9.

GAZDAR, G., G. K. PULLUM (1985): *Computationally relevant properties of natural languages and their grammars*. New Generation Computing 3(1985), 273–306.

GOLD, E. M. (1965): *Limiting recursion*. The Journal of Symbolic Logic 30(1965), 28–48.

——— (1967): *Language identification in the limit*. Information and Control 10(1967), 447–474.

Hopcroft, J., R. Motwani, J. Ullman (2001): *Introduction to automata theory, languages, and computation*, Addison-Weslley Publishing Company, 2001.

Janssen, T. (2003): *Compositionality*, [w:] Handbook of Logic and Linguistics, (J. van Benthem , A. ter Meulen, red.), Elsevier Science Publishers, 2003, str. 417–473.

Kleene, S.C. (1956): *Representation of events in nerve nets and finite automata*, [w:] Automata Studies, (C. E. Shannon , J. McCarthy, red.), Princeton University Press, 1956, str. 3–40.

Kurcz, I. (2000): *Psychologia języka i komunikacji*, Wydawnictwo Naukowe „Scholar", 2000.

Levy, L. S., A. K. Joshi (1978): *Skeletal structural descriptions*. Information and Control 39/2(1978), 192–211.

Lindström, P. (1966): *First order predicate logic with generalized quantifiers*. Theoria 32(1966), 186–195.

Lyons, J. (1977): *Semantics*, Cambridge University Press, 1977.

Macnamara, J. (1986): *A Border dispute. The place of logic in psychology*, The MIT Press, 1986.

Moschovakis (1994): *Sense and denotation as algorithm and value*, [w:] Lecture Notes in Logic 2, (J. Oikkonen , J. Väänänen, red.), Springer–Verlag, 1994, str. 210–249.

Mostowski, M. (1994): *Kwantyfikatory rozgałęzione a problem formy logicznej*, [w:] Nauka i język, (M. Omyła, red.), vol. 8(1994), Biblioteka Myśli Semiotycznej, str. 201–242.

——— (1998): *Computational semantics for monadic quantifiers*. Journal of Applied Non-Classical Logics 8, 107–121.

Mostowski, M., D. Wojtyniak (2004): *Computational Complexity of the Semantics of Some Natural Language Constructions*. Annals of Pure and Applied Logic 127.

Parekh, Y., V. Honavar (1997): *Learning DFA from simple examples*, [w:] Algorithmic Learning Theory, Proceedings from 8th International Workshop, Sendai, Japan, vol. 1316, Springer–Verlag, 1997, str. 116–131.

Partee, B., A. ter Meulen, R. E. Wall (1993): *Mathematical methods in linguistics*, Kluwer Academic Publishers, 1993.

Pinker, S. (2000): *The language instinct. The new science of language and mind*, Penguin Books, 2000.

Pitt, L. (1989): *Inductive inference, DFA's and computational complexity*, [w:] Analogical and Inductive Inference, Lecture Notes in Artificial Intelligence, 397, Springer–Verlag, 1989, str. 18–44.

Pullum, G. K., G. Gazdar (1982): *Natural languages and context–free grammars*. Linguistics and Philosophy 4, 471–504.

Quine, W. V. O. (1990): *Pursuit of Truth*, Harvard University Press, 1990.

Sakakibara, Y. (1990): *Learning context-free grammars from structural data in polynomial time*. Theoretical Computer Science 75(1990), 223 – 242.

Shieber, S. M. (1985): *Evidence against the context–freeness of natural language*. Linguistics and Philosophy 8(1985), 333–343.

Shoenfield, J. R. (1991): *Recursion theory*, Springer–Verlag, 1991.

Sudkamp, T.A. (1988): *Languages and machines. An introduction to the theory of computer science*, Addison–Wesley Publishing Company, 1988.

Szymanik, J. (2004): *Semantyka obliczeniowa dla kwantyfikatorów monadycznych w języku naturalnym*, Praca magisterska w Instytucie Filozofii UW, promotor: prof. Marcin Mostowski.

Tiede, H. J. (1999): *Identifiability in the limit of context–free generalized quantifiers*. Journal of Language and Computation 1, 93–102.

Turing, A.M. (1950): *Computing machinery and intelligence*. Mind 49(1950), 433–460.

Valiant, L. (1984): *A theory of the learnable*. Computation of the ACM 27/11(1984), 1134–1142.