Jakub Szymanik

Semantyka obliczeniowa dla kwantyfikatorów monadycznych w języku naturalnym*

Tak więc tym, co pcha nas wszędzie, by od sensu sięgać do znaczenia,
jest dążenie do prawdy.
Gottlob Frege „Sinn und Bedeutung"

Dwaj ludzie rozumieją pewne wyrażenie w tym samym znaczeniu,
gdy rozumienie to uzbraja ich obu w tę samą metodę rozstrzygania,
czy wyrażenie to zastosować do jakiegoś przedmiotu, czy też nie.
Kazimierz Ajdukiewicz „Logika pragmatyczna"

1. Sformułowanie problemu

Jednym z interesujących pytań w teorii języka jest problem opisu i wyjaśnienia mechanizmów, które odpowiadają za naszą zdolność rozumienia zdań. Opis mechanizmu działania kompetencji językowej, który można określić mianem kompetencji semantycznej, jest niezbędny dla zrozumienia fenomenu języka. Posługiwanie się językiem polega bowiem nie tylko na używaniu określonego słownika i reguł gramatycznych, lecz przede wszystkim na łączeniu z wyrażeniami odpowiednich znaczeń. Kiedy na przykład mówię „rana" to o tym czy posługuję się językiem polskim czy językiem łacińskim decyduje intencja znaczeniowa z jaką użyłem tego słowa, czy miałem na myśli skaleczenie czy żabę (zob. [Ajdukiewicz 1931]).

Ważnym składnikiem tego zadania jest opis warunków prawdziwości zdań języka naturalnego i tym właśnie zajmuje się semantyka. Zazwyczaj ogranicza się ona do opisu funkcji, która każdemu poprawnie zbudowanemu wyrażeniu przypisuje pewnien obiekt teoriomnogościowy skonstruowany w uniwersum modelu i stanowiący ekstensję (denotację) tego wyrażenia. Co znajduje swój wyraz w częstym określaniu tego typu badań nad językiem mianem czysto ekstensjonalnych. Techniczne aspekty tego podejścia zostały rozwinięte przez Alfreda Tarskiego [Tarski 1933]. W odniesieniu do semantyki języka naturalnego z sukcesem zastosował je uczeń Tarskiego Richard Montague (zob. zbiór prac R. Montague [Thomason 1974]).

Istnieje tradycja, zapoczątkowana przez Gottloba Fregego [Frege 1892], myślenia o znaczeniu wyrażenia językowego jako o „sposobie reprezentacji" jego denotacji. Tak rozumiane znaczenie wyrażenia można utożsamić z procedurą znajdowania jego ekstensji. Po raz pierwszy eksplikacja Fregowskiego

* Artykuł ten w dużej mierze powstał na bazie mojej pracy magisterskiej. Chciałbym podziękować promotorowi, Marcinowi Mostowskiemu, za wskazanie tematu i pomoc w przygotowaniu pracy. Jestem również wdzięczny wielu osobom za inspirujące dyskusje. Szczególne podziękowania należą się Ninie Gierasimczuk.

rozróżnienia na *Sinn* i *Bedeutung* z wykorzystaniem aparatury teorii obliczeń pojawiła się w pracy Pavla Tichy'ego „Intension In Terms Of Turing Machines" [Tichy 1969]. W 1990 roku na Logic Colloqium Yiannis Moschovakis wygłosił referat „Sense and Denotation as Algorithm and Value" [Moschovakis 1990], w którym rozwijał idee Fregego odwołując się do pojęcia algorytmu, nie wspominając jednak o pracy czeskiego logika.

Kilka lat wcześniej Johan van Benthem w pracy „Semantic Automata" zapoczątkował rozważania nad kwantyfikatorami w języku naturalnym z perspektywy teorii obliczeń [van Benthem 1986] (zob. też: [van Benthem 1987]). Przegląd problematyki logicznej związanej z semantyką obliczeniową dla kwantyfikatorów zawiera praca J. A. Makowsky'ego i Y. B. Pnueli „Computable quantifiers and logics over finite structures" [Makowsky, Pnueli 1995]. Semantyka obliczeniowa dla kwantyfikatorów monadycznych została systematycznie potraktowana w pracy Marcina Mostowskiego „Computational semantics for monadic quantifiers" [M. Mostowski 1998].

Również w pracach czysto lingwistycznych obecne jest spojrzenie na semantykę języka naturalnego z perspektywy teorii obliczeń (zob. np. [Suppes 1982], [Piasecki 2004]). Wspomniany wyżej artykuł Tichy'ego wraz z jego kolejną pracą „An Approach to Intensional Analysis" [Tichy 1971] zainicjował cały nurt badań poświęcony lingwistyce obliczeniowej pod nazwą „Transparent Intensional Logic", którego głównym celem wydaje się odpowiedź na pytanie „Czym są znaczenia wyrażeń językowych?" (zob. [Hajicova, Materna, Sgall 1988]). Bezpośrednio do pracy Moschovakisa nawiązują lingwiści z Amsterdamu (zob. [van Lambalgen, Hamm 2004a], [van Lambalgen, Hamm 2004b]).

W niniejszym artykule rozważa się semantykę obliczeniową dla kwantyfikatorów w języku naturalnym z perspektywy opisu mechanizmów poznawczych człowieka. Procedurę wyliczającą denotację wyrażenia będziemy utożsamiać z jego znaczeniem referencyjnym. W pracy nie poruszamy zagadnienia innych możliwych sposobów określania znaczenia wyrażeń językowych (zob. np. [M. Mostowski 1994]). Przez szczegółową analizę wybranego fragmentu języka przy wykorzystaniu narzędzi logiki oraz teorii obliczeń chcemy przybliżyć się do opowiedzi na następujące pytania:

— Jak ludzie mogą rozpoznawać denotacje wyrażeń językowych?
— Dlaczego jedne zdania są trudniejsze od innych?
— Czym jest znaczenie danego wyrażenia?

Zaczynamy od omówienia na przykładach sposobów przyporządkowania kwantyfikatorom procedur, które pozwalają wyznaczyć denotację zdania oraz wyjaśniamy powody, dla których ograniczamy się tylko i wyłącznie do rozważania struktur skończonych. Następny rozdział poświęcamy omówieniu automatów skończonych oraz automatów ze stosem, a w kolejnym wprowadzamy pojęcie monadycznego kwantyfikatora uogólnionego. Elementy teorii obliczeń i logiki pozwalają nam precyzyjnie opisać związki zachodzące pomiędzy kwantyfikatorami monadycznymi w języku naturalnym a odpowiadającymi im algorytmami. Ustalenia te pozwalają nam następnie sformułować hipotezę psychologiczną na temat możliwych sposobów interpretowania przez ludzi zdań z kwantyfikatorami. Dyskutujemy również najnowsze badania neurologiczne, które wydają się potwierdzać postawioną tezę. W ostatnim rozdziale ponownie analizujemy zdania dyskutowane we wprowadzeniu. Na ich przykładzie, tym razem już w precyzyjnym języku teorii obliczeń, ilustrujemy ideę traktowania referencyjnego znaczenia zdania jako algorytmu wyliczającego wartość logiczną tego zdania w skończonych uniwersach. Omawiamy też konsekwencje utożsamienia znaczenia zdania z algorytmem wyliczającym jego ekstensję dla problemu synonimiczności zdań.

2. Kwantyfikatory a procedury

Kwantyfikatorom w języku naturalnym, np.: „każdy”, „pewien”, „co najmniej pięć”, „więcej niż”, „tyle samo co”, itp., odpowiadają procedury, które wyznaczają ich interpretacje. Rozważmy na przykład następujące zdania języka polskiego:

1. *Każda książka w bibliotece IF UW została wydana po 1900 roku.*
2. *Dokładnie trzy książki w bibliotece IF UW mają plastikowe okładki.*
3. *Większość książek w bibliotece IF UW została wydana po 1980 roku.*
4. *W bibliotece IF UW jest tyle samo książek różowych, co żółtych oraz tyle samo zielonych.*

Rozumiemy te zdania, aby zaś stwierdzić ich wartość logiczną posługujemy się pewnymi procedurami (algorytmami), które są wyznaczone przez występujące w tych zdaniach kwantyfikatory. Spróbujmy opisać przykładowe procedury zaczynając od najprostszych wyrażeń kwantyfikatorowych.

(Ad. 1) Aby ustalić wartość logiczną pierwszego zdania bierzemy do ręki kolejne kartki z kompletnego katalogu biblioteki i sprawdzamy rok wydania książki. Postępujemy tak, dopóki znajdziemy książkę wydaną przed 1900 rokiem albo wyczerpią się zasoby katalogu. Procedura ta zawsze się kończy, ponieważ jest skończenie wiele książek, każdej książce przyporządkowana jest dokładnie jedna karta i nigdy nie bierzemy do ręki więcej niż raz tej samej karty. Jeśli skończymy wykonywać tę czynność zanim przejrzymy wszystkie fiszki znaczy to, iż jedna z książek została wydana przed 1901 rokiem. W tym przypadku zdanie (1) jest fałszywe. W przeciwnym razie, czyli jeśli uda się przejrzeć wszystkie pozycje katalogu i nie znaleźć książki sprzed 1901 roku, to zdanie (1) jest prawdziwe.

(Ad. 2) Zdanie (2) będzie prawdziwe wtedy i tylko wtedy, gdy:

5. *Co najmniej trzy książki mają plastikowe okładki.*

i zarazem:

6. *Co najwyżej trzy książki mają plastikowe okładki.*

Innymi słowy musimy sprawdzić dwa warunki: (5) oraz (6). Ponownie analizujemy po kolei wszystkie książki. Jeśli przejrzeliśmy cały księgozbiór i znaleźliśmy mniej niż trzy książki z plastikową okładką, to zdanie (5) jest fałszywe więc (2) też musi być fałszywe. Jeżeli w pewnym momencie przeszukiwania znaleźliśmy trzecią książkę z plastikową okładką, to zaczynamy analizować zdanie (6), czyli przeglądamy księgozbiór dalej, szukając plastikowych okładek. Jeśli znajdziemy choćby jeszcze jedną, to zdanie (6) jest fałszywe a zatem fałszywe jest też zdanie (2). W przeciwnym przypadku (6) jest prawdziwe i (2) musi być również prawdziwe. Aby zrealizować ten algorytm trzeba przejrzeć cały księgozbiór. W trakcie przeglądania musimy pamiętać jedną z pięciu rzeczy, albo że znaleźliśmy dotychczas n książek z plastikową okładką, dla $n = 0, 1, 2, 3$, albo że znaleźliśmy już więcej niż trzy takie książki.

(Ad. 3) W celu sprawdzenia prawdziwości zdania (3) bierzemy do ręki kolejne fiszki i układamy je jedna na drugiej (w stos) na osobnym stole w następujący sposób: Jeśli na wierzchu stosu leży karta książki wydanej po (przed) 1980 roku, to jeżeli następna fiszka dokumentuje książkę również wydaną po (przed) 1980 roku, to kładę tę fiszkę na stos i teraz ona jest na szczycie. W przeciwnym przypadku, to znaczy, jeśli na stosie leży karta książki wydanej po (przed) 1980 roku a następna karta dokumentuje pozycję wydaną przed (po) 1980 rokiem, to wówczas zdejmuję znajdującą się na wierzchu fiszkę. Zdanie (3) jest prawdziwe wtedy i tylko wtedy, gdy po przejrzeniu całego księgozbioru na stosie leży co najmniej jedna karta książki wydanej po 1980 roku. Aby zrealizować opisaną procedurę trzeba przejrzeć cały księgozbiór, lecz tym razem do wykonania algorytmu wykorzystujemy dodatkową pomoc w postaci stosu fiszek, który teoretycznie może pomieścić dowolną liczbę kart bibliotecznych. Sprawdzenie wartości logicznej zdania (3) jest więc trudniejsze niż obliczenie tej wartości dla poprzednich zdań.

(Ad. 4) Procedura obliczania wartości logicznej zdania (4) przypomina algorytm dla zdania (3), lecz tym razem potrzeba przynajmniej dwóch stosów. Za pomocą pierwszego sprawdzamy jak wyżej, czy liczba książek różowych jest równa liczbie książek żółtych. Drugi stos służy nam do porównania liczby zielonych książek z liczbą książek różowych i żółtych.

Bierzemy do ręki kolejne książki i sprawdzamy kolor ich okładek. Jeśli trzymamy właśnie różową (żółtą) książkę i na pierwszym stosie leży książka tego samego koloru, to kładziemy trzymaną książkę na stos. Jeśli jednak książka na wierzchu pierwszego stosu jest innego koloru, to ją zdejmujemy i kładziemy ją na stos drugi pod warunkiem, że jest on pusty. Jeśli nie jest pusty, to zdejmujemy z tego stosu książkę.

Jeżeli natomiast bierzemy do ręki książkę zieloną, to wtedy patrzymy na szczyt drugiego stosu. Jeśli leży na nim książka zielona, to kładziemy tam przed chwilą zdjęty wolumin. W przeciwnym przypadku, czyli kiedy na szczycie drugiego stosu leży książka różowa lub żółta, zdejmujemy książkę ze szczytu drugiego stosu.

Zdanie (4) jest prawdziwe wtedy i tylko wtedy, gdy po przeanalizowaniu całego księgozbioru oba stosy są puste.

Procedury tego typu można precyzyjnie opisać oraz porównać ich złożoność przy wykorzystaniu aparatu pojęciowego teorii automatów.

3. Uniwersa skończone

Autorzy zajmujący się semantyką języka naturalnego ograniczają zazwyczaj swoją uwagę do rozważania modeli o skończonych uniwersach (zob. np. [Montague 1970], [Westerståhl, Peters 1989]). Intuicyjnie wydaje się, że to wystarcza, aby adekwatnie opisać semantykę języka naturalnego. W większości sytuacji komunikacyjnych odnosimy się do względnie małych uniwersów dyskursu. Na przykład zdania:

— *Dokładnie pięcioro dzieci Jana skończyło wyższe studia.*
— *Większość dzieci Jana otworzyło praktykę adwokacką.*
— *Jan ma tyle samo córek co synów oraz bratanków.*

mają naturalną interpretację w skończonym uniwersum, jakim jest rodzina Jana.

Innym powodem takiego ograniczenia mogą być problemy z intuicyjną interpretacją semantyki pewnych wyrażeń języka naturalnego w uniwersach nieskończonych. Na przykład rozważając wyrażenia kwantyfikatorowe, takie jak: „więcej", czy „większośc" rozumiemy wyrażenie „Więcej x-ów jest φ niż ψ" jako: „istnieje więcej x-ów spełniających formułę φ niż spełniających formułę ψ". Redukujemy zatem problem semantyki tych wyrażeń do pytania o odpowiadające im relacje pomiędzy zbiorami obiektów spełniającymi odpowiednie formuły. W skończonych uniwersach na pytanie to istnieje powszechnie akceptowana odpowiedź polegająca na porównywaniu liczb kardynalnych odpowiadających tym zbiorom. Przechodząc do przypadku uniwersów nieskończonych rozwiązanie to traci wiele ze swej intuicyjności. Rozważmy bowiem zdania:

— *Większość liczb naturalnych to liczby złożone.*

— *Więcej punktów w przestrzeni kosmicznej należy do gwiazd niż do planet.*

Zdania te są sensowne a ponadto wydają się intuicyjnie prawdziwe, lecz jeśli tak, jak poprzednio interpretujemy występujące w nich kwantyfikatory w terminach relacji pomiędzy liczbami kardynalnymi, to zdania te powinniśmy oczywiście uznać za fałszywe (zob. też: [Krynicki, M. Mostowski 1999]).

Przytoczony powyżej argument za ograniczeniem się do skończonych interpretacji nie jest oczywiście wystarczający. Chociaż ograniczenie takie istotnie upraszcza nasze rozważania teoretyczne, to jednak pomijamy w ten sposób wiele istotnych przypadków. Mimo to w niniejszej pracy bedą nas interesowały tylko modele skończone, co wydaje się dopuszczalnym założeniem w kontekście rozważań nad językiem naturalnym. Z naszej perspektywy ograniczenie do uniwersów skończonych ma jeszcze tę zaletę, że procedury szukania denotacji wyrażeń językowych nabierają charakteru algorytmicznego (efektywnego). Można więc sensownie pytać o złożoność obliczeniową pewnych konstrukcji w języku naturalnym (zob. [M. Mostowski, Wojtyniak 2004], też [Sevenster 2006], [M.Mostowski, J.Szymanik 2005]).

Zdania (1) – (4) wyraźnie różnią się pod względem trudności, każde kolejne wyraża coraz bardziej złożoną treść. Poczucie tej różnicy jest spowodowane wzrostem złożoności algorytmów, które możemy wykorzystywać do sprawdzenia prawdziwości tych zdań. Mianowicie różnym klasom wyrażeń językowych są przypisane różne mechanizmy semantyczne o różnej złożoności obliczeniowej. Opis kompetencji semantycznej musi polegać więc na podaniu zbioru algorytmów, z których każdy przypisany jest jakiejś klasie wyrażeń i formalizuje metodę znajdowania denotacji wyrażeń należących do tej klasy. W niniejszej pracy charakteryzuje się algorytmy odpowiadające znaczeniu kwantyfikatorów monadycznych pod względem ich złożoności obliczeniowej.

4. Automaty

Teoria automatów zajmuje się analizą abstrakcyjnych maszyn liczących. W latach trzydziestych w pracy „On computable numbers, with an application to the Entscheidungsproblem" Alan Turing wprowadził pojęcie deterministycznego i niedeterministycznego (*with choice*) automatu oraz zdefiniował ogólny model obliczeń nazwany później maszyną Turinga [Turing 1936]. W latach czterdziestych dwudziestego wieku rozpoczęto badania nad prostszymi urządzeniami, tzw. automatami skończonymi. W kolejnym dziesięcioleciu badania nad automatami splotły się z rozwijaną przez Noama Chomsky'ego teorią gramatyk. Gramatykom w hierarchii Chomsky'ego odpowiadają klasy automatów, które rozpoznają języki generowane przez te gramatyki [Chomsky 1957]. W latach sześćdziesiątych teoria automatów i gramatyk okazała się przydatnym narzędziem w projektowaniu języków programowania wysokiego rzędu, przyczyniając się tym samym do rozwoju informatyki (zob. [Hopcroft, Motwani, Ullman 2001], [Rosenberg, Salomaa 1997]).

4.1. Podstawowe pojęcia lingwistyki matematyczneh

Alfabet to skończony i niepusty zbiór symboli. Na przykład:

1. $A = \{a, b\}$ - alfabet binarny;
2. $B = \{0, 1\}$ - inny alfabet binarny;
3. $C = \{a, \ldots z, A, \ldots Z\}$ - zbiór znaków alfabetu łacińskiego;
4. $D = \{\boxed{\text{pies}}, \boxed{\text{jak}}, \boxed{\text{biega}}, \boxed{\text{je}}, \boxed{\text{szybko}}\}$ - alfabet fragmentu języka polskiego;

Słowo to skończony ciąg symboli wybranych z pewnego alfabetu. Na przykład ciąg "111000111010100101" jest słowem nad alfabetem B, a ciąg $\boxed{\text{pies}}\boxed{\text{biega}}\boxed{\text{szybko}}$ jest słowem nad alfabetem D.

Słowo puste to ciąg, w którym nie występują żadne symbole alfabetu. Słowo puste oznaczamy literą ε.

Długość słowa to liczba symboli w nim występujących, oznaczamy ją przez $lh()$, np. $lh(111)$=3 a $lh(\varepsilon) = 0$.

Jeśli Σ jest alfabetem przez Σ^k oznaczamy zbiór słów długości k nad alfabetem Σ. Na przykład $\{0, 1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Dla dowolnego alfabetu Σ mamy $\Sigma^0 = \{\varepsilon\}$. Dla dowolnej litery a i liczby naturalnej n przez a^n oznaczamy ciąg długości n złożony z samych liter a.

Zbiór wszystkich słów nad alfabetem Σ oznaczamy przez Σ^*, np. $\{0, 1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$. Innymi słowy $\Sigma^* = \bigcup_{n\in\omega} \Sigma^n$. Każdy taki zbiór jest nieskończony, poza przypadkami, kiedy $\Sigma = \emptyset$ lub $\Sigma = \{\varepsilon\}$.

Przez xy oznaczamy *konkatenację* słowa x oraz słowa y, czyli słowo powstałe z kopii słowa x, po której bezpośrednio następuje kopia słowa y. Jeśli $x = a_1 \ldots a_i$ a $y = b_1 \ldots b_n$, to wówczas xy jest słowem długości $i + n$, takim, że $xy = a_1 \ldots a_i b_1 \ldots b_n$. Na przykład, jeśli $x = 101$ oraz $y = 00$, to $xy = 10100$. Dla dowolnego ciągu α zachodzi $\varepsilon\alpha = \alpha\varepsilon = \alpha$, czyli ε jest elementem neutralnym dla konkatenacji.

Dowolny zbiór słów wybranych z Σ^*, dla pewnego alfabetu Σ, nazywamy *językiem*. Jeśli Σ jest alfabetem oraz $L \subseteq \Sigma^*$, to mówimy, że L jest językiem nad Σ. Na przykład podzbiór $L \subseteq A^*$, taki, że $L = \{\alpha :$ liczba wystąpień litery a oraz b w α jest parzysta$\}$ jest językiem nad alfabetem A. Zbiór J poprawnych zdań języka polskiego taki, że $J \subset D^*$ jest językiem nad alfabetem dla fragmentu języka polskiego.

4.2. Automaty skończone

Definicja 1. *Niedeterministyczny automat skończony (FA) jest to piątka postaci* (A, Q, q_s, F, δ), *gdzie:*

— *A jest alfabetem wejściowym;*
— *Q jest skończonym zbiorem stanów;*

— $q_s \in Q$ *jest wyróżnionym stanem początkowym;*
— $F \subseteq Q$ *jest zbiorem stanów akceptujących;*
— $\delta : Q \times A \longrightarrow P(Q)$ *jest funkcją przejścia.*

Jeśli $H = (A, Q, q_s, \delta, F)$ jest FA oraz dla dowolnego $a \in A$ oraz $q \in Q$ $card(\delta(q,a)) \leqslant 1$, to H jest automatem deterministycznym. Wówczas funkcję przejścia można opisać jako funkcję częściową: $\delta : Q \times A \longrightarrow Q$. 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.

Następnie zdefiniujemy indukcyjnie uogólnioną funkcję przejścia $\bar{\delta}$, która opisuje zachowanie automatu, który w stanie q zaczyna czytać słowo w:

$$\bar{\delta} : Q \times A^* \longrightarrow P(Q), \text{ gdzie:}$$

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

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

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

$$L(H) = \{w \in A^* : \bar{\delta}(q_s, w) \cap F \neq \emptyset\}.$$

Język $L \subseteq A^*$ jest regularny wtedy i tylko wtedy, gdy istnieje FA H taki, że $L = L(H)$. Co więcej wiadomo, że deterministyczne oraz niedeterministyczne automaty skończone rozpoznają dokładnie tę samą klasę języków (języki regularne).

4.2.1. Przykłady

Opiszemy kilka prostych przykładów języków regularnych wraz z akceptującymi je automatami skończonymi.

Niech $A = \{a, b\}$ i 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 $\delta_1(q_{1s}, b) = q_{1s}$.

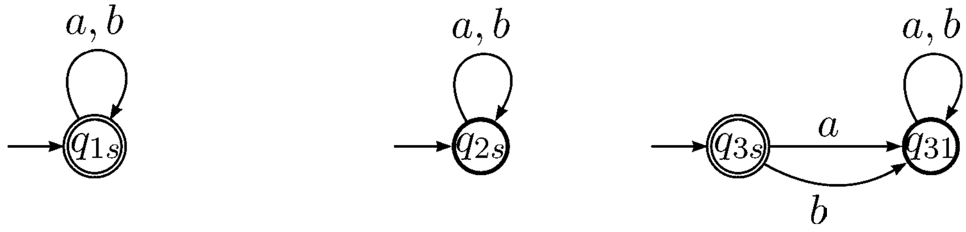
Niech $L_2 = \emptyset$; wówczas $L_2 = L(H_2)$, gdzie $H_2 = (Q_2, q_{2s}, F_2, \delta_2)$ taki, że: $Q_2 = \{q_{2s}\}$, $F_2 = \emptyset$, $\delta_2(q_{2s}, a) = q_{2s}$ oraz $\delta_2(q_{2s}, b) = q_{2s}$.

Niech $L_3 = \{\varepsilon\}$; 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 $i = a, b$.

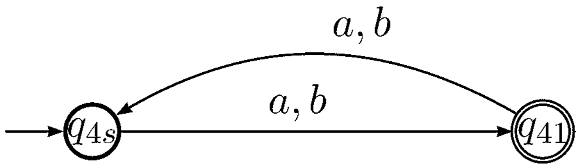
Niech symbol $n_x(w)$ oznacza liczbę wystąpień litery x w słowie w. Opiszemy teraz język, w którym występują tylko słowa o różnej, co do parzystości, liczbie wystąpień liter a i b. Zatem $L_4 = \{w \in A^* : n_a(w) \not\equiv n_b(w)(\text{mod}2)\}$. $L_4 = L(H_4)$, gdzie $H_4 = (Q_4, q_{4s}, F_4, \delta_4)$ taki, że: $Q_4 = \{q_{4s}, q_{41}\}$, $F_4 = \{q_{41}\}$, $\delta_4(q_{4s}, i) = q_{41}$ oraz $\delta_4(q_{41}, i) = q_{4s}$, dla $i = a, b$.

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

Rysunek 1. FA akceptujące L_1, L_2 oraz L_3.



Rysunek 2. FA akceptujący $L_4 = \{w \in A^* : n_a(w) \not\equiv n_b(w)(\mathrm{mod}2)\}$.



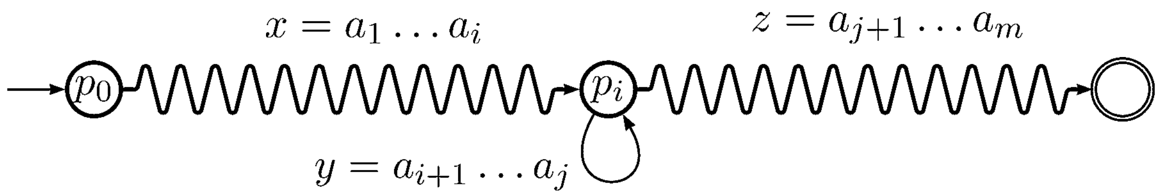
4.2.2. Prosta struktura nawiasowa i lemat o pompowaniu

Języki regularne to języki akceptowane przez automaty skończone. Nie wszystkie języki są regularne, do rozpoznania niektórych nie wystarcza skończona pamięć, jaką dysponują FA. Przykładem są języki zawierające w sobie prostą strukturę nawiasową $L_{[]} = \{[^n]^n : n \geqslant 1\}$. Słowa tego języka mogą być dowolnie długie, a ich rozpoznawanie polega na zapamiętywaniu lewych nawiasów i sprawdzaniu czy prawych nawiasów jest dokładnie tyle samo. Słowo tego języka może zaczynać się od dowolnej liczby lewych nawiasów więc odpowiedni automat musi zapamiętać dowolną liczbę n. Do tego zadania potrzebujemy maszyn z pamięcią, która pozwala na zapisanie liczby przeczytanych symboli „[". To umożliwia późniejsze porównanie jej z liczbą symboli „]". Automat skończony o k stanach może zapamiętać tylko liczby mniejsze od k. Precyzyjnie ten stan rzeczy ukazuje następujące twierdzenie:

Twierdzenie 1. *(Lemat o pompowaniu dla języków regularnych) Dla dowolnego nieskończonego języka regularnego $L \subseteq A^*$ istnieje liczba naturalna n taka, że dla dowolnego słowa $\alpha \in L$, jeśli $lh(\alpha) \geqslant n$, to istnieją x, y, $z \in A^*$ takie, że:*

1. $\alpha = xyz$
2. $y \neq \varepsilon$
3. $lh(xz) \leqslant n$
4. *Dla każdego $k \geqslant 0$ ciąg xy^kz również należy do L.*

Rysunek 3. Każde słowo dłuższe niż liczba stanów automatu powoduje, iż musi on wracać do jakiegoś stanu.



Przykładem języka, który nie jest regularny, ponieważ zawiera w sobie strukturę nawiasową, jest zbiór wszystkich poprawnie zbudowanych wyrażeń rachunku zdań. Problemem jest tutaj opis gramatyki spójników dwuargumentowych, w przypadku których poprawność formuły zależy od poprawnego rozmieszczenia nawiasów. Innym takim językiem nieregularnym jest język polski, w którym występują konstrukcje gramatyczne typu „jeśli ..., to", „albo ... albo" oraz związki gramatyczne pomiędzy podmiotem i orzeczeniem, o których można myśleć jak o obligatoryjnej zależności pomiędzy wyrażeniami.

Dyskusja nad regularnością gramatyk języków naturalnych była jednym z ważniejszych tematów w początkach lingwistyki matematycznej. Noam Chomsky pokazał, że język angielski nie jest językiem regularnym [Chomsky 1957]. Weźmy na przykład fragment języka angielskiego składający się ze zdań:

1. *The cat died.*
2. *The cat the dog chased died.*
3. *The cat the dog the rat bit chased died.*
4. *The cat the dog the rat the elephant admired bit chased died.*

Powyższe zdania są postaci:

$$\text{(noun phrase)}^n \text{(transitive verb)}^{n-1} \text{ intransitive verb}$$

Język $L = \{a^n b^{n-1} c : a \in NP, b \in TV, c \in ITV\}$, gdzie NP – noun phrase, TV – transitive verb, ITV – intransitive verb, nie jest oczywiście regularny na mocy lematu o pompowaniu.

4.3. Automaty ze stosem

Definicja 2. *Niedeterministyczny automat ze stosem (PDA) jest to struktura postaci* $(A, \Sigma, \#, Q, q_s, F, \delta)$*, gdzie:*

- *A jest alfabetem wejściowym;*
- *Σ jest alfabetem stosu;*
- *$\#$ jest symbolem początkowym stosu;*
- *Q jest skończonym zbiorem stanów;*
- *$q_s \in Q$ jest wyróżnionym stanem początkowym;*
- *$F \subseteq Q$ dowolny zbiór stanów akceptujących;*
- *$\delta : Q \times (A \cup \{\varepsilon\}) \times \Sigma \longrightarrow P(Q \times \Sigma^*)$ jest funkcją przejścia. Pojedyńczy krok obliczenia oznaczymy następująco: $(q, a, n) \xrightarrow{H} (p, \gamma)$, jeśli $(p, \gamma) \in \delta(q, a, n)$, gdzie $q, p \in Q, a \in A, n \in \Sigma, \gamma \in \Sigma^*$.*

Język rozpoznawany przez PDA H to zbiór słów w nad alfabetem A, które są akceptowane przez H, tzn. H zaczynając czytać w w stanie startowym q_0 z pustym stosem kończy pracę w stanie akceptującym $p \in F$. Jeśli ponadto stos jest pusty po przeczytaniu w, to powiemy, że H akceptuje przez pusty stos.

Powiemy, że język $L \subseteq A^*$ jest bezkontekstowy, wtedy i tylko wtedy, gdy istnieje PDA H, taki, że $L = L(H)$.

Klasa języków bezkontekstowych jest oczywiście szersza niż klasa języków regularnych. Język $L_{[]} = \{[^m]^m : m \in \omega\}$, o którym wiemy, że nie jest regularny, jest bezkontekstowy. Aby to wykazać wystarczy skonstruować PDA H, taki, że $L_{[]} = L(H)$. Niech $H = (A, \Sigma, \#, Q, q_s, F, \delta)$, gdzie $A = \{[,]\} = \Sigma$, $Q = \{q_s, q_1, q_2 q_a\}$, $F = \{q_a\}$ a funkcja przejścia jest określona następująco:

— $(q_s, [, \#) \xrightarrow{H} (q_s, \#[)$
— $(q_s, [, [) \xrightarrow{H} (q_s, [[)$
— $(q_s,], [) \xrightarrow{H} (q_1, \varepsilon)$
— $(q_s,], \#) \xrightarrow{H} (q_2, \varepsilon)$
— $(q_1, \varepsilon, \#) \xrightarrow{H} (q_a, \varepsilon)$
— $(q_1,],]) \xrightarrow{H} (q_2, \varepsilon)$
— $(q_1, [,]) \xrightarrow{H} (q_2, \varepsilon)$
— $(q_1,], \#) \xrightarrow{H} (q_2, \varepsilon)$
— $(q_1, [, \#) \xrightarrow{H} (q_2, \varepsilon)$

H rozpoznaje $L_{[]}$, czytając słowo od lewej do prawej i zapisując na stosie napotykane „[". Po znalezieniu pierwszego „]" H zdejmuje z wierzchu stosu „[", kiedy czyta „]". Jeśli po przeczytaniu całego słowa stos jest pusty, to H akceptuje to słowo. Zatem H akceptuje tylko słowa, które zawierają taką samą liczbę lewych i prawych nawiasów.

Ograniczenia dla języków bezkontekstowych opisuje odpowiednia wersja lematu o pompowaniu. Wynika z niej, że język $L_{abc} = \{a^k b^k c^k : k \geqslant 1\}$ nie jest bezkontekstowy.

Twierdzenie 2. (Lemat o pompowaniu dla języków bezkontekstowych) *Dla dowolnego języka bezkontekstowego $L \subseteq A^*$ istnieje liczba naturalna k, taka, że dla dowolnego $w \in L$, jeśli $lh(w) \geqslant k$, to istnieją $\beta_1, \beta_2, \gamma_1, \gamma_2, \eta$ takie, że:*

— $\gamma_1 \neq \varepsilon \vee \gamma_2 \neq \varepsilon$
— $w = \beta_1 \gamma_1 \eta \gamma_2 \beta_2$
— *dla dowolnego* $m \in \omega$: $\beta_1 \gamma_1^m \eta \gamma_2^m \beta_2 \in L$.

Obecnie lingwiści toczą spór czy składnia języka naturalnego jest bezkontekstowa. Większość badaczy skłania się ku poglądowi, że do opisu gramatyki języka naturalnego wystarczy siła wyrazu gramatyk bezkontekstowych, choć podaje się również kontrprzykłady (dyskusję tego tematu można znaleźć m.in. w pracach [Gazdar, Pullum 1985], [Pullum, Gazdar 1982], [Shieber 1985]). Pokażemy, że siła języków bezkontekstowych z pewnością nie wystarcza do opisania semantyki języka naturalnego.

5. Kwantyfikatory monadyczne

Wiele kwantyfikatorów w języku naturalnym nie jest definiowalnych środkami logiki elementarnej. W logice pierwszego rzędu nie jest wyrażalna na przykład parzystość czy skończoność liczby elementów, które spełniają daną formułę. Własności te możemy wyrazić w logice pierwszego rzędu rozszerzonej o dodatkowe kwantyfikatory. Pojęcie kwantyfikatora uogólnionego zostało wprowadzone przez Andrzeja Mostowskiego [A. Mostowski 1957], który badał rozszerzenia logiki elementarnej o kwantyfikatory takie, jak „istnieje nieprzeliczalnie wiele" czy „istnieje nieskończenie wiele". Kwantyfikatory badane przez A. Mostowskiego wiązały jedną zmienną w jednej formule. Ogólniejszy opis kwantyfikatorów podał P. Lindström [Lindström 1966]. Zgodnie z jego definicją każda strukturalna własność modeli nad jakimś ustalonym słownikiem wyznacza interpretację pewnego kwantyfikatora. Richard Montague zainicjował badania nad semantyką języka naturalnego przy wykorzystaniu pojęcia kwantyfikatora uogólnionego [Montague 1970]. Pojęciu kwantyfikatora uogólnionego i jego użyteczności w opisie lingwistycznym poświęcono wiele prac (zob. np. [van Benthem 1986], [Barwise, Cooper 1981], [M. Mostowski 1994], [Westerståhl, Peters 1989]). Poniżej ograniczamy się do rozważania kwantyfikatorów monadycznych w modelach skończonych.

Definicja 3. *Niech K będzie domkniętą na izomorfizm klasą modeli postaci $(U, R_1,\ldots,R_n)$, gdzie $U \neq \emptyset$ oraz $R_i \subseteq U$, dla $i=1,\ldots,n$. K wyznacza interpretację monadycznego kwantyfikatora Q_K. Dla dowolnego modelu M oraz wartościowania $\bar{a}$ w M zachodzi:*

$$M \models Q_K x(\varphi_1(x),\ldots,\varphi_n(x))[\bar{a}] \iff (|M|, \varphi_1^{M,x,\bar{a}},\ldots,\varphi_n^{M,x,\bar{a}}) \in K,$$

gdzie $|M|$ jest uniwersum modelu M, a $\varphi^{M,x,\bar{a}}$ to zbiór wyznaczony przez φ w M ze względu na zmienną x przy wartościowaniu $\bar{a}$. Kwantyfikator Q_K typu $\underbrace{(1,1,\ldots 1)}_{n}$ wiąże jedną zmienną pierwszego rzędu w n formułach. Zbiór formuł logiki $L(Q_K)$ definiujemy przyjmując standardowe reguły tworzenia formuł dla logiki elementarnej oraz dodatkowo: jeśli $\varphi_1,\ldots,\varphi_n$ są formułami oraz x jest zmienną indywiduową, to $Qx(\varphi_1,\ldots,\varphi_n)$ jest formułą logiki $L(Q)$.

5.1. Przykłady

Kwantyfikator egzystencjalny ($\exists$) Dla dowolnego modelu M zachodzi:

$$M \models \exists x\varphi(x)[\bar{a}] \iff card(\varphi^{M,x,\bar{a}}) \geqslant 1 \iff (|M|, \varphi^{M,x,\bar{a}}) \in K_E$$

Klasę modeli K_E, która wyznacza interpretację kwantyfikatora egzystencjalnego określamy następująco:

$$K_E = \{(|M|, R) : R \subseteq |M| \wedge R \neq \emptyset\}$$

Kwantyfikator ogólny ($\forall$)

$$M \models \forall x\varphi(x)[\bar{a}] \iff \varphi^{M,x,\bar{a}} = |M| \iff (|M|, \varphi^{M,x,\bar{a}}) \in K_A,$$

$$K_A = \{(|M|, R) : R = |M| \wedge R \neq \emptyset\}.$$

Kwantyfikator parzystości (D_2)

$$M \models D_2 x\varphi(x)[\bar{a}] \iff card(\varphi^{M,x,\bar{a}})\ \textit{jest podzielna przez}\ 2 \iff$$

$$\iff (|M|, \varphi^{M,x,\bar{a}}) \in K_{D_2},$$

$$K_{D_2} = \{(|M|, R) : R \subseteq |M| \wedge card(R) = 2k,\ \textit{gdzie}\ k \in \omega\}.$$

Kwantyfikator podzielności (D_n)

$$M \models D_n x\varphi(x)[\bar{a}] \iff card(\varphi^{M,x,\bar{a}})\ \text{jest podzielna przez}\ n \iff$$

$$\iff (|M|, \varphi^{M,x,\bar{a}}) \in K_{D_n},$$

$$K_{D_n} = \{(|M|, R) : R \subseteq |M| \wedge card(R) = kn,\ \textit{gdzie}\ k,\ n \in \omega\}.$$

Istnieje dokładnie m $(\exists^{=m})$

$$M \models \exists^{=m} x\varphi(x)[\bar{a}] \iff card(\varphi^{M,x,\bar{a}}) = m \iff$$

$$\iff \exists x_1, \ldots, x_m(\varphi(x_1) \wedge \ldots \wedge \varphi(x_m)) \wedge \forall z(\varphi(z) \Rightarrow z = x_1 \vee \ldots \vee z = x_m),$$

$$K_{\exists^{=m}} = \{(|M|, R) : R \subseteq |M| \wedge card(R) = m\}.$$

Większość (W)

$$M \models Wx(\varphi_1(x), \varphi_2(x))[\bar{a}] \iff card(\varphi_1^{M,x,\bar{a}}\ \cap\ \varphi_2^{M,x,\bar{a}}) > card(\varphi_1^{M,x,\bar{a}}\ -\ \varphi_2^{M,x,\bar{a}}) \iff$$

$$\iff (|M|, \varphi_1^{M,x,\bar{a}}, \varphi_2^{M,x,\bar{a}}) \in K_W,$$

$$K_W = \{(|M|, R_1, R_2) : R_1, R_2 \subseteq |M| \wedge card(R_1 \cap R_2) > card(R_1 - R_2)\}.$$

Tyle samo co (TS)

$$M \models TSx(\varphi_1(x), \ldots \varphi_n(x))[\bar{a}] \iff card(\varphi_1^{M,x,\bar{a}}) = \ldots = card(\varphi_n^{M,x,\bar{a}}) \iff$$

$$\iff (|M|, \varphi_1^{M,x,\bar{a}}, \ldots, \varphi_n^{M,x,\bar{a}}) \in K_{TS},$$

$$K_{TS} = \{(|M|, R_1, \ldots, R_n) : R_1, \ldots, R_n \subseteq |M|\ \wedge\ card(R_1) = \ldots = card(R_n)\}.$$

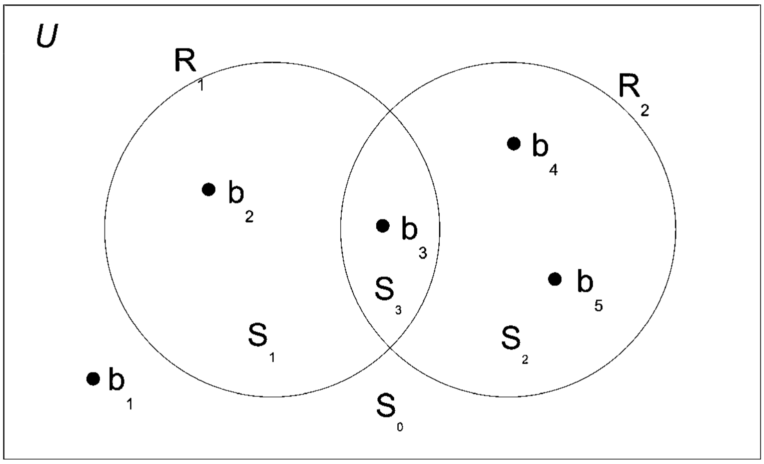
6. Kwantyfikatory i obliczenia

Klasę K_Q skończonych modeli postaci $\{M, R_1, \ldots, R_n\}$ można reprezentować za pomocą niepustego zbioru słów L_Q nad alfabetem A $= \{a_0,\ldots,a_{2^n-1}\}$, takiego, że: $\alpha \in L_Q$ wtedy i tylko wtedy, gdy istnieje $(U, R_1,\ldots,R_n) \in K_Q$ oraz liniowy porządek na $U = \{b_1,\ldots,b_k\}$ taki, że: lh$(\alpha)=k$ oraz i-ta litera α jest równa a_j dokładnie wtedy, kiedy $b_i \in S_1 \cap \ldots \cap S_n$, gdzie:

$$S_l = \begin{cases} R_l & \text{jeśli część całkowita } \frac{j}{2^{l-1}} \text{ jest liczbą nieparzystą} \\ U - R_l & \text{w przeciwnym przypadku} \end{cases}$$

Zdefiniowane powyżej przecięcia $S_1 \cap \ldots \cap S_n$ to tak zwane składowe modelu, litery $a_0, \ldots, a_{2^n-1}$ nazywają te składowe. Powyższa definicja mówi tyle, iż i-ta litera słowa α jest równa a_j wtedy i tylko wtedy, gdy element b_i należy do j-tej składowej. Zatem słowo α opisuje w jednoznaczny sposób model, to znaczy koduje informację o tym, które elementy należą do których składowych.

Dla zilustrowania powyższej idei rozważmy model $M = (U, R_1, R_2)$, gdzie $U = \{b_1, b_2, b_3, b_4, b_5\}$. Model ten będzie reprezentowało słowo $\alpha_M = a_0a_1a_3a_2a_2$ nad alfabetem $A = \{a_0, a_1, a_2, a_3\}$, które mówi, że element $b_1 \in S_1 = U-(R_1 \cup R_2)$, $b_2 \in S_2 = R_1 - R_2$, $b_3 \in S_4 = R_1 \cap R_2$, a $b_4, b_5 \in S_3 = R_2 - R_1$.



Rysunek 4. $M = (U, R_1, R_2)$

Teraz możemy już zdefiniować, co to znaczy, że pewna klasa kwantyfikatorów jest rozpoznawana przez pewną klasę automatów.

Definicja 4. *Niech $\mathcal{A}$ będzie klasą automatów a $\mathcal{Q}$ klasą kwantyfikatorów monadycznych.* $\mathcal{A}$ akceptuje $\mathcal{Q}$ *wtedy i tylko wtedy, gdy dla każdego monadycznego kwantyfikatora Q:*

$$(Q \in \mathcal{Q} \iff \textit{istnieje automat } A \in \mathcal{A} \ (A \textit{ akceptuje } L_Q)).$$

Dzięki powyższym definicjom możemy pytać o złożoność procedur rozpoznawania prawdziwości zdań z kwantyfikatorami w modelach skończonych. Odpowiedzi na tego typu pytanie jako pierwszy udzielił J. van Benthem (zob. [van Benthem 1986]), który pokazał między innymi, że skończone automaty bez powrotów akceptują klasę wszystkich kwantyfikatorów definiowalnych w logice elementarnej oraz że automaty ze stosem akceptują tzw. póliniowe kwantyfikatory (definiowalne w strukturze $(\omega, +)$) typu (1). Wyniki badań nad semantyką obliczeniową kwantyfikatorów monadycznych zostały uogólnione i uporządkowane w pracy M. Mostowskiego (zob. [M. Mostowski 1998]), w której znajdujemy m.in. następujące twierdzenie:

Twierdzenie 3. *Kwantyfikator monadyczny Q jest definiowalny w logice $FO(D_\omega)$ $\iff$ L_Q jest rozpoznawany przez automat skończony.*

Nie wszystkim kwantyfikatorom odpowiada zbiór słów, który jest regularny. Rozpatrzmy na przykład kwantyfikator „większość”, który jest wyznaczony przez klasę modeli:

$$K_W = \{(|M|, R_1, R_2) : R_1, R_2 \subseteq |M| \wedge card(R_1 \cap R_2) > card(R_1 - R_2)\}$$

K_W można opisac jako język bezkontekstowy L_W nad alfabetem $A = \{a_0, a_1, a_2, a_3\}$ (zob. rysunek 4) w następujący sposób:

$$L_W = \{\alpha \in A^* : n_{a_3}(\alpha) > n_{a_1}(\alpha)\}.$$

Do rozpoznania prawdziwości zdań z kwantyfikatorem „większość” w modelach skończonych potrzebujemy więc automatu ze stosem, który akceptuje język L_W.

Van Benthem postawił hipotezę, iż do opisu semantyki obliczeniowej dla kwantyfikatorów w języku naturalnym wystarczą maszyny o mocy automatów ze stosem [van Benthem 1986]. Łatwo wskazać kontrprzykład. Wystarczy w tym celu rozważyć kwantyfikator „tyle samo ..., co ... oraz tyle samo ...”. Kwantyfikator ten jest opisany przez klasę:

$$K_{TS} = \{(|M|, R_1, R_2, R_3) : R_1, R_2, R_3 \subseteq |M| \ \wedge \ card(R_1) = card(R_2) = card(R_3)\}$$

Odpowiada jej język L_{TS} nad alfabetem $A = \{a_0, \ldots, a_7\}$, taki, że

$$L_{TS} = \{\alpha \in A^* : n_{a_1}(\alpha) + n_{a_4}(\alpha) + n_{a_5}(\alpha) = n_{a_2}(\alpha) + n_{a_4}(\alpha) + n_{a_6}(\alpha) =$$
$$= n_{a_3}(\alpha) + n_{a_5}(\alpha) + n_{a_6}(\alpha)\}.$$

Korzystając z lematu o pompowaniu dla języków bezkontekstowych łatwo zauważyć, że nie istnieje automat ze stosem rozpoznający język L_{TS}. Przeczy to oczywiście hipotezie sformułowanej przez van Benthema. Wydaje mi się, że bardzo trudno znaleźć jakiekolwiek górne ograniczenie na złożoność obliczeniową dla kwantyfikatorów w języku naturalnym. Wraz z używaniem języka ciągle uczymy się posługiwać coraz bardziej złożonymi pod względem semantycznym konstrukcjami. Do języka naturalnego należą nie tylko wyrażenia często używane w mowie potocznej, jak np. „pewien”, „każdy”, „kilku”, lecz także semantycznie wyrafinowane konstrukcje służące nam do uprawiania nauki (zob. też [M.Mostowski, J.Szymanik 2005]). Ewentualne ograniczenia na złożoność konstrukcji semantycznych języka naturalnego mogą zostać odkryte w ramach badań nad zdolnościami obliczeniowymi mózgu.

7. Rozumienie kwantyfikatorów w języku naturalnym

Powyższe logiczne ustalenia pozwalają na sformułowanie pewnych hipotez, dotyczących procesu interpretowania przez ludzi zdań z kwantyfikatorami w skończonych modelach. Można przypuszczać, że w proces rozumienia zdań z kwantyfikatorami definiowalnymi w logice podzielności nie jest zaangażowana pamięć operacyjna człowieka, podczas gdy do zrozumienia zdań z kwantyfikatorami o większej złożoności obliczeniowej jest konieczne skorzystanie z zasobów takiej pamięci.

Metody badania neurofizjologicznego podłoża języka są rozwijane co najmniej od 1861 roku, kiedy to Paul Broca opisał pacjenta z trudnościami w ekspresji mowy, u którego po śmierci zlokalizowano uszkodzenie w lewej półkuli mózgu. Niedługo potem Carl Wernicke badał chorego z uszkodzeniami lewej półkuli i trudnościami w rozumieniu mowy. Obserwacje te pozwoliły na określenie części mózgu odpowiedzialnej za ekspresję mowy, tzw. okolicy Broki oraz za rozumienie mowy — okolicy Wernickego. Od tamtego czasu trwają zarówno obserwacje kliniczne pacjentów z uszkodzeniami mózgu i zaburzeniami w funkcjonowaniu językowym, jak i inne próby docierania do neurofizjologicznego podłoża języka.

Obecnie dostępne są techniki polegające na wizualizacji nieuszkodzonych obszarów kory mózgowej, które odpowiadają za posługiwanie się językiem. Do takich nieinwazyjnych technik neuroobrazowania należy pozytronowa tomografia emisyjna PET (*positron emission tomography*) oraz czynnościowy rezonans magnetyczny fMRI (*functional magnetic resonance imaging*). Obie techniki polegają na mierzeniu zmian w przepływie krwi w badanych obszarach mózgu pacjenta, wykonującego pewne zadanie poznawcze, i porównaniu ich ze zmianami zachodzącymi podczas wykonywania innych zadań podobnego typu. Przepływ krwi w danej części mózgu jest uznawany za dowód jej aktywności. Mierzenie aktywności mózgu przy wykorzystaniu neuroobrazowania wymaga zatem co najmniej dwóch pomiarów, aby dostrzec różnice w aktywności mózgu pomiędzy dwoma różnymi zadaniami poznawczymi. Nawet jeśli zadania eksperymentalne są dopasowane na bazie mocnych przesłanek psychologicznych, to ciągle interpretacje wyników uzyskanych przy użyciu metod neuroobrazowania należy traktować ostrożnie (zob. też [Bookheimer 2002]).

7.1. Dane neurologiczne

Pierwszą próbę empirycznego testowania implikacji obliczeniowej teorii znaczenia przy użyciu technik neuroobrazowania podjął zespół badawczy z Uniwersytetu w Pensylwani [McMillan, Clark et al. 2005].

W prezentowanych badaniach rozważa się dwie klasy kwantyfikatorów: kwantyfikatory pierwszego rzędu (np. *każdy*, *niektórzy*, *przynajmniej trzech*, *żaden*, *dokładnie dwóch*) oraz kwantyfikatory wyższych rzędów (np. *większość*, *więcej niż*, *parzyście wiele*). Autorzy stawiają następujące hipotezy:

1. Rozumienie wszystkich kwantyfikatorów zależy od kompetencji liczbowej człowieka (zob. [Dehaene 1997]), która pozwala identyfikować własności liczbowe.
2. Rozumienie kwantyfikatorów wyższych rzędów wymaga dodatkowo uaktywnienia pamięci operacyjnej (zob. [Baddeley 1986]), która przechowuje te własności liczbowe podczas przetwarzania zdania, oraz mechanizmu odpowiadającego za operowanie (porównywanie) na tych wielkościach liczbowych.
3. Różnica pomiędzy złożonością obliczeniową kwantyfikatorów pierwszego rzędu i kwantyfikatorów

wyższych rzędów wpływa na różnice w aktywności mózgu podczas przetwarzania tych kwantyfikatorów.

4. W szczególności, przetwarzanie kwantyfikatorów wyższych rzędów zależy od aktywności obszarów mózgu związanych z pamięcią operacyjną w sposób istotnie różny niż dzieje się to w przypadku kwantyfikatorów pierwszego rzędu.

7.1.1. Metoda badawcza

Przebadano dwunastu dorosłych amerykanów (ośmiu mężczyzn i cztery kobiety). Badanym prezentowano przez 10 sekund proste zdanie zawierające kwantyfikację po liczbie obiektów danego koloru (np. *Przynajmniej trzy piłki są niebieskie*). Przez następne 10 sekund wraz z wcześniej wyświetlonym zdaniem pokazywano obrazek zawierający osiem losowo wybranych przedmiotów (ze zbioru: kobiety, piłki, kwiaty, samochody, dinozaury). Badani byli proszeni o udzielenie odpowiedzi na pytanie, czy zdanie dokładnie opisuje wyświetloną sytuację. Obserwowano aktywację obszarów mózgu podczas rozwiązywania zadania przy wykorzystaniu fMRI.

Wyświetlano zdania z sześcioma różnymi kwantyfikatorami w dwudziestu próbach: w połowie prób badano kwantyfikatory pierwszego rzędu (*przynajmniej trzy*, *wszyscy*, *niektórzy*), a w połowie kwantyfikatory wyższego rzędu (*mniej niż połowa*, *parzyście wiele*, *nieparzyście wiele*). Połowa zdań była prawdziwa.

7.1.2. Wyniki

Zarówno rozumienie kwantyfikatorów pierwszego jak i wyższych rzędów aktywuje niższą korę ciemieniową (*inferior parietal cortex*), związaną z liczeniem. Tylko kwantyfikatory wyższych rzędów aktywują korę przedczołową (*prefrontal cortex*), związaną z zasobami wykonawczymi, takimi jak pamięć operacyjna. Oba typy kwantyfikatorów aktywują również dolną część kory ciemieniowej prawej półkuli (*right inferior parietal cortex*), co sugeruje, iż zdolności liczbowe są związane z przetwarzaniem kwantyfikatorów. Tylko kwantyfikatory wyższych rzędów aktywują korę okolicy przedczołowej po stronie brzuszno–bocznej prawej półkuli (*right dorsolateral prefrontal cortex*), co wskazuje na zaangażowanie pamięci roboczej do ich rozumienia. Wyniki te potwierdzają postawione hipotezy.

7.1.3. Dyskusja

Przedstawione badania są pierwszą próbą ustalenia różnic anatomicznych podczas przetwarzania kwantyfikatorów. Badania te pokazują jak ważne jest pojęcie złożoności obliczeniowej dla oceny trudności zadania oraz dostarczają częściowego potwierdzenia empirycznej trafności logiczno–lingwistycznego modelu przetwarzania kwantyfikatorów. Z drugiej strony badania te skłaniają do stawiania nowych licznych pytań i podnoszenia wątpliwości. Wszystkie one domagają się odpowiedzi i wyjaśnienia za sprawą podjęcia nowych badań. Poniżej omawiamy pokrótce niektóre problemy[1].

Logika podzielności a kwantyfikatory wyższych rzędów Nadużyciem jest zaliczenie kwantyfikatora *istnieje (nie)parzyście wiele* do tej samej grupy co kwantyfikatora *większość*. Ten pierwszy jest defi-

[1] Pełną dyskusję tych badań znajdzie czytelnik w artykule [Szymanik 2007].

niowalny w logice podzielności i rozpoznawany przez automat skończony. Innymi słowy do wyliczenia wartości tego kwantyfikatora nie trzeba się odwoływać do mechanizmu wykorzystującego pamięć operacyjną. Natomiast kwantyfikator *większość* jest typowym przykładem kwantyfikatora do rozpoznania którego potrzebujemy pamięci operacyjnej. Zatem teoretyczne przewidywania wskazują raczej na to, iż aktywacja pamięci operacyjnej powinna występować dopiero w wypadku kwantyfikatorów nie definiowalnych w logice podzielności. Wydaje się ważne przeprowadzenie szczegółowych badań nad różnicami w przetwarzaniu kwantyfikatorów definiowalnych w logice elementarnej i w logice podzielności.

Trudność kwantyfikatora a uporządkowanie uniwersum Trudność oceny prawdziwości zdania zależy również od uporządkowania elementów uniwersum. W opisanych badaniach elementy były generowane w losowym położeniu. Tymczasem porządkując uniwersum można sprawdzić różnicę pomiędzy zdaniami wykorzystującymi pamięć operacyjną i takimi, które jej nie wymagają. Na przykład, aby rozpoznać prawdziwość zdania *Większość A jest B* nad dowolnym uniwersum potrzebujemy skorzystać z pamięci operacyjnej. Jeśli elementy są uporządkowane w pary (a, b), gdzie $a \in A$, $b \in B$, to wtedy możemy łatwo rozpoznać wartość logiczną zdania bez użycia pamięci. W takim przypadku wystarczy jedynie sprawdzić, czy istnieje element a niepołączony w parę z żadnym elementem b.

8. Algorytm jako znaczenie

Poniżej zilustrujemy ideę traktowania znaczenia zdania jako algorytmu wyliczającego jego wartość logiczną w skończonym i ustalonym uniwersum. Rozważymy w tym celu zdania z kwantyfikatorami monadycznymi:

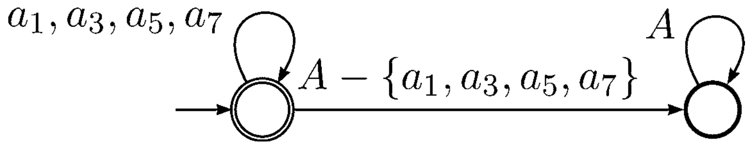
1. *Każda książka w bibliotece IF UW jest zielona.*
2. *Pewna książka w bibliotece IF UW jest czerwona.*
3. *Co najmniej dwie książki w bibliotece IF UW są niebieskie.*
4. *Większość książek w bibliotece IF UW jest zielona.*

Możemy teraz opisać ich znaczenie referencyjne (algorytm wyliczania wartości logicznej w zadanym uniwersum) za pomocą wprowadzonych pojęć. Powiedzmy, że interesujemy się wartością logiczną tych zdań w świecie $W = (U, R_1, R_2, R_3)$ wyznaczonym przez pewną ich interpretację, gdzie $U = \{k_1, \ldots, k_n\}$ – uniwersum składające się z książek biblioteki IF UW, $R_i \subseteq U$ dla $i = 1, \ldots, 3$, takie, że R_1 – zbiór książek zielonych, R_2 – zbiór książek czerwonych, R_3 – zbiór książek niebieskich. Na wejściu nasz algorytm będzie otrzymywał słowo α_W, które opisuje model W z dokładnością do izomorfizmu. Słowo α_W otrzymujemy wybierając dowolny porządek na elementach uniwersum $U = \{b_1, \ldots, b_n\}$ a następnie za i-tą literę podstawiamy j-tą literę z alfabetu $A = \{a_0, \ldots, a_7\}$ wtedy i tylko wtedy, gdy b_i jest w j-tej składowej. Algorytm będzie akceptował α_W wtedy i tylko wtedy, gdy w W będzie prawdziwe zdanie, którego znaczeniem jest ten algorytm.

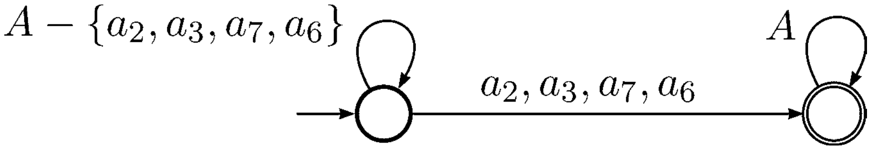
Dla trzech pierwszych zdań odpowiednie algorytmy opiszemy za pomocą automatów skończonych. Znaczeniem zdania (1.) jest algorytm rozstrzygający, czy $\alpha_W \in L_\forall$:

Znaczeniem zdania (2.) jest automat rozpoznający język $L_\exists \in A^*$:

Rysunek 5. FA akceptujący $L_\forall$



Rysunek 6. FA akceptujący $L_\exists$



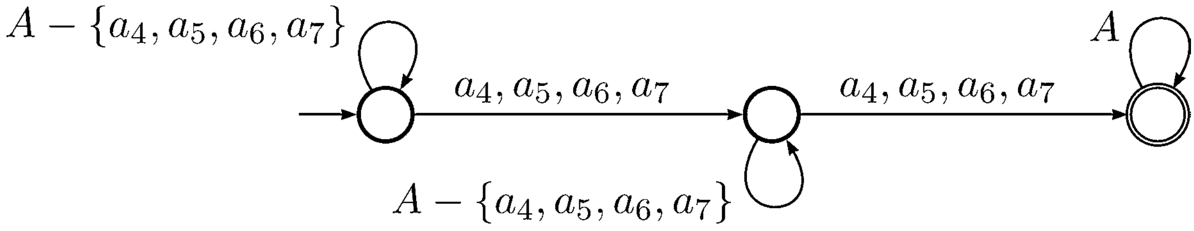
Znaczenie zdania (3.) możemy opisać następująco:

W opisie znaczenia zdania (4.) skorzystamy z automatu ze stosem $H = (A, \Sigma, \#, Q, q_s, F, \delta)$, gdzie $A = \{a_0, a_1, \ldots, a_7\}$, $\Sigma = \{1\}$, $Q = \{q_s, q_1, q_2, q_3 q_a\}$, $F = \{q_a\}$ a funkcja przejścia jest określona następująco:
— $(q_s, a_k, \#) \overset{H}{\to} (q_1, \#1)$, dla $k = 1, 3, 5, 7$;
— $(q_s, a_i, \#) \overset{H}{\to} (q_2, 1)$, dla $i = 0, 2, 4, 6$;
— $(q_1, a_k, 1) \overset{H}{\to} (q_1, 11)$, dla $k = 1, 3, 5, 7$;
— $(q_1, a_i, 1) \overset{H}{\to} (q_1, \varepsilon)$, dla dla $i = 0, 2, 4, 6$;
— $(q_1, a_i, \#) \overset{H}{\to} (q_2, \#1)$, dla dla $i = 0, 2, 4, 6$;
— $(q_1, a_k, \#) \overset{H}{\to} (q_1, \#1)$, dla $k = 1, 3, 5, 7$;
— $(q_1, \varepsilon, \#) \overset{H}{\to} (q_3, \#\varepsilon)$;
— $(q_1, \varepsilon, 1) \overset{H}{\to} (q_a, \varepsilon)$;
— $(q_2, a_i, 1) \overset{H}{\to} (q_2, 11)$, dla $i = 0, 2, 4, 6$;
— $(q_2, a_k, 1) \overset{H}{\to} (q_2, \varepsilon)$, dla $k = 1, 3, 5, 7$;
— $(q_2, a_k, \#) \overset{H}{\to} (q_1, \#1)$, dla $k = 1, 3, 5, 7$;
— $(q_2, a_i, \#) \overset{H}{\to} (q_2, \#1)$, dla dla $i = 0, 2, 4, 6$;
— $(q_2, \varepsilon, \#) \overset{H}{\to} (q_3, \varepsilon)$;
— $(q_2, \varepsilon, 1) \overset{H}{\to} (q_3, \varepsilon)$;

Oczywiście, co wynika z naszych rozważań, istnieją zdania których znaczenie referencyjne jest bardziej skomplikowane niż język bezkontekstowy, np. zdania z kwantyfikatorem TS.

8.1. Kryterium identyczności znaczeń

Różnym zdaniom przypisujemy różne znaczenia. Pojawia się tutaj problem synonimiczności, który przy eksplikacji znaczenia jako algorytmu wyliczającego denotację przybiera postać pytania o to, Rysunek 7. FA akceptujący $L_{\exists\geqslant 2}$



kiedy dwa algorytmy są identyczne. Problem ten jest również dyskutowany z perspektywy rozważań nad podstawami informatyki[2].

Innymi słowy, mając dany zbiór A fizycznych realizacji algorytmów, szukamy relacji równoważności $\approx$ na A, takiej, że dla dowolnych $f, g \in A$:

$$f \approx g \iff f \text{ i } g \text{ realizują ten sam algorytm.}$$

Minimalnym wymogiem na $\approx$ jest to, aby utożsamiała ona notacyjne warianty tego samego algorytmu. Najszersza z rozsądnych definicji identyczności dwóch algorytmów utożsamia algorytmy obliczające tę samą funkcję częściową. Powiemy więc, że algorytmy f i g są identyczne ($f \cong g$), wtedy i tylko wtedy, gdy zatrzymują się dla tych samych danych wejściowych ($\forall \alpha \in \Sigma^* \{f\}(\alpha) \downarrow \iff \{g\}(\alpha) \downarrow$), dając dla identycznych argumentów taki sam wynik ($\forall \alpha \in \Sigma^* (\{f\}(\alpha) \downarrow \wedge \{f\}(\alpha) = \beta \Rightarrow \{g\}(\alpha) = \beta)$). Zatem:

$$(f \cong g) \iff \forall \alpha, \beta \in \Sigma^* [\{f\}(\alpha) \downarrow \iff \{g\}(\alpha) \downarrow \wedge$$

$$\wedge (\{f\}(\alpha) \downarrow \wedge \{f\}(\alpha) = \beta \Rightarrow \{g\}(\alpha) = \beta)]$$

Tak rozumiane kryterium równoznaczności jest nieefektywne. Dla dowolnych dwóch algorytmów f i g problem: „Czy $f \cong g$?” jest nierozstrzygalny. Pytanie o tak pojętą identyczność algorytmów potrafimy mechanicznie rozstrzygać tylko i wyłącznie w sytuacji najprostszej, kiedy algorytmy te można utożsamić z automatami skończonymi. Co więcej, nie każde dwa identyczne algorytmy są równie dobre. Na przykład jeden potrzebuje dla zdania długości n wykonać n^2 kroków, a drugi aż 2^{2^n}. Dlatego być może rozsądnym uzupełnieniem definicji równoznaczności zdań byłoby dodanie warunku, iż dwa znaczenia f, g są równoznaczne, gdy $f \cong g$ i ponadto f oraz g są porównywalne pod względem złożoności obliczeniowej [M. Mostowski, Wojtyniak 2004]. Zatem ($f \approx g$) wtedy i tylko wtedy, gdy $f \cong g$ oraz istnieją wielomiany $p(m, n)$, $q(m, n)$ takie, że dla dowolnego słowa $\alpha \in \Sigma^*$ długości n zachodzi:

— jeśli $\{f\}(\alpha) \downarrow$ po m krokach, to $\{g\}(\alpha) \downarrow$ po $\leqslant p(m, n)$ krokach;
— jeśli $\{g\}(\alpha) \downarrow$ po m krokach, to $\{f\}(\alpha) \downarrow$ po $\leqslant q(m, n)$ krokach;

Przy takiej definicji uzyskujemy całą hierarchię znaczeń dla poszczególnych zdań, wyznaczoną przez właściwości odpowiadających znaczeniom algorytmów.

[2] Problemowi temu z perspektywy filozofi informatyki poświęcony jest tekst Kierasia i Szymanika pt. „Czym jest algorytm”, który znajduje się w tym tomie Studiów Semiotycznych.

9. Wnioski

Oto niektóre wnioski z powyższych rozważań:

1. Znaczenie wielu wyrażeń możemy trafnie opisać poprzez wskazanie procedur wyliczających ich denotacje, w szczególności w przypadku zdań ich wartość logiczną.
2. Dla różnych klas wyrażeń procedury te różnią się pod względem złożoności obliczeniowej. Prawdopodobnie dlatego pewne zdania wydają nam się trudniejsze od innych.
3. Znaczenie wyrażenia językowego można utożsamić z procedurą obliczającą jego ekstensję w ustalonym skończonym uniwersum. Przy takiej eksplikacji problem synonimiczności wyrażeń przybiera postać pytania o identyczność algorytmów.
4. Obliczeniowy model znaczenia kwantyfikatorów znajduje częściowe potwierdzenie w danych neurologicznych.

Pozostaje jeszcze wiele ciekawych i ważnych zagadnień, na które nie udzieliliśmy żadnej odpowiedzi. Należą do nich między innymi:

1. Problem adekwatnego opisu semantyki wyrażeń językowych bez ograniczania się do uniwersów skończonych.
2. Opis semantyki, innych niż kwantyfikatory, wyrażeń języka naturalnego przy wykorzystaniu aparatury teorii obliczeń.
3. Opis innych możliwych sposobów ustalania znaczenia wyrażeń językowych.
4. Opis możliwych procedur odpowiadających za uczenie się semantyki wyrażeń językowych[3].
5. Szczegółowe porównanie modeli teoretycznych z danymi pochodzącymi z badań psychologicznych i neurologicznych nad językowymi funkcjami mózgu.

Zagadnienia te wymagają osobnego opracowania na bazie interdyscyplinarnych badań z wykorzystaniem dorobku takich dyscyplin, jak: filozofia, informatyka, językoznawstwo, logika oraz neuropsychologia.

[3] Temat ten jest szczegółowo potraktowany w artykule Niny Gierasimczuk pt. „Algorytmiczne podejście do problemu uczenia się języka", który to tekst również można znaleźć w tym tomie.

Literatura

[Ajdukiewicz 1931] K. AJDUKIEWICZ *O znaczeniu wyrażeń*, w: **Księga Pamiątkowa Polskiego Towarzystwa Filozoficznego we Lwowie**, Lwów 1931, str. 31 – 77. Przedruk w: K. Ajdukiewicz **Język i poznanie**, tom I, PWN, Warszawa 1985, str. 103 – 136.

[Baddeley 1986] A. BADDELEY *Working Memory*, Oxford University Press, 1986.

[Barwise 1979] J. BARWISE *On Branching Quantifiers in English*, **Journal of Philosophical Logic** 8 (1979), str 47 – 80.

[van Benthem 1986] J. VAN BENTHEM *Essays in Logical Semantics*, Reidel Publishing Company, Amsterdam 1986.

[van Benthem 1987] J. VAN BENTHEM *Towards a Computational Semantics*, **Generalized Quantifiers**, pod red. P. Gardenförsa, D. Reidel Publishing Company, Amsterdam 1987.

[Barwise, Cooper 1981] J. BARWISE, R. COOPER *Generalized Quantifiers and Natural Language*, **Liguistics and Philosophy** 4 (1981), str. 159 –219. Przedruk w: [Portner, Partee 2002].

[Bookheimer 2002] S. BOOKHEIMER *Functional MRI of Language: New Approaches to Understanding the Cortical Organization of Semantic Processing*, **Annual Review of Neuroscience**, Vol. 25 (2002), str. 151 – 188.

[Chomsky 1957] N. CHOMSKY *Syntactic Structures*, Mouton, Haga 1957. Tłumaczenie: *Zagadnienia teorii składni*, przeł. I. Jakubcza, Ossolineum 1981.

[Dehaene 1997] S. DEHAENE, *The Number Sense: How the Mind Created Mathematics*, Oxford University Press, 1997.

[Frege 1892] G. FREGE *Über Sinn und Bedeutung*, **Zeitschrift für Philosophie und philosophische Kritik** 100, str. 25 – 50. Tłumaczenie w: G. Frege **Pisma semantyczne**, pod red. B. Wolniewicza, PWN, Warszawa 1997, str 60 – 89.

[Gazdar, Pullum 1985] , G. GAZDAR, G.K. PULLUM *Computationally relevant properties of natural languages and their grammars*, **New Generation Computing**, 3, 1985, str.273–306.

[Hajicova, Materna, Sgall 1988] E. HAJICOVA, P. MATERNA, P. SGALL *Linguistic Constructions in the Transparent Intensional Logic*, **Categorial Grammar**, pod red. W. Buszkowskiego, W. Marciszewskiego, J. van Benthema, John Benjamins Publishing Company, zob. też: http://www.phil.muni.cz/fil/logika/til/articles.html

[Hopcroft, Motwani, Ullman 2001] J. HOPCROFT, R. MOTWANI, J. ULLMAN *Introduction to Automata Theory, Languages, and Computation*, Addison-Weslley 2001.

[van Lambalgen, Hamm 2004a] M. VAN LAMBALGEN, F. HAMM *The Proper Treatment of Events*, Blackwell, 2004.

[van Lambalgen, Hamm 2004b] M. VAN LAMBALGEN, F. HAMM *Moschovaki's notion of meaning as applied to linguistics*, **Logic Colloqium '01, ASL Lecture Notes in Logic**, pod red. M. Baaza, S. Friedmana, J. Krajicka, A.K. Peters Publishers, 2004, zob. też: http://staff.science.uva.nl/ michiell/docs/MoschWien.pdf

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

[Krynicki, M. Mostowski 1999] M. KRYNICKI, M. MOSTOWSKI *Ambigous quantifiers*, **Logic at work**, pod red. E. Orłowskiej, Heidelberg 1999, str. 548 – 565.

[Krynicki, M. Mostowski, Szczerba 1995] M. KRYNICKI, M. MOSTOWSKI, L. SZCZERBA (red.) Quantifiers: Logics, Models and Computation, Kluwer 1995.

[Makowsky, Pnueli 1995] J. A. MAKOWSKY, Y. B. PNUELI *Computable quantifiers and logics over finite structures*, w: [Krynicki, M. Mostowski, Szczerba 1995], str. 313 – 357.

[McMillan, Clark et al. 2005] C. MCMILLAN, R. CLARK, P. MOORE, C. DEVITA, M. GROSSMAN *Neural Basis for Generalized Quantifiers*, **Neuropsychology**, 43, str. 1729–1737.

[Montague 1970] R. MONTAGUE *The Proper Treatment of Quantification in Ordinary English*, **Proceedings of the 1970 Stanford Workshop on Grammar and Semantics**, pod red. J. Hintikki, J. Moravcsika i P. Suppesa, D. Reidel Publishing Company, Dordrecht, 1973. Przedruk w: [Montague 1970] oraz [Portner, Partee 2002].

[Moschovakis 1990] Y. MOSCHOVAKIS *Sense and Denotation as Algorithm and Value*, **Lecture Notes in Logic 2**, pod red. J. Oikkonena oraz J. Väänänena, Springer 1994, str. 210 – 249, zob. też: http://www.math.ucla.edu/ ynm/papers/frege.ps

[A. Mostowski 1957] A. MOSTOWSKI *On a Generalization of Quantifiers*, **Funadamenta Mathematicae** 44 (1957), str. 12 – 36.

[M. Mostowski 1994] M. MOSTOWSKI *Kwantyfikatory rozgałęzione a problem formy logicznej*, **Nauka i język**, pod red. M. Omyły, BMS, Warszawa 1994, str. 201 – 242.

[M. Mostowski 1998] M. MOSTOWSKI *Computational Semantics for Monadic Quantifiers*, **Journal of Applied Non-Classical Logics** Vol. 8 (1998) no 1-2, str. 107 – 121.

[M.Mostowski, J.Szymanik 2005] M. MOSTOWSKI, J. SZYMANIK *Everyday fragment of natural language*, **Semiotica**, w druku.

[M. Mostowski, Wojtyniak 2004] M. MOSTOWSKI, D. WOJTYNIAK *Computational Complexity of the Semantics of Some Natural Language Constructions*, **Annals of Pure and Applied Logic** Vol. 127 (2004), 1 – 3, str. 219 – 227.

[Partee, ter Meulen, Wall 1993] B. H. PARTEE, A. TER MEULEN, R. E. *Mathematical Methods in Linguistics*, Kluwer Academic Publishers 1993.

[Piasecki 2004] M. PIASECKI *Selektywne wprowadzenie do semantyki formalnej*, w: **Kognitywistyka. O umyśle umyślnie i nieumyślnie**, pod red. J. Szymanika, M. Zajenkowskiego, Koło Filozoficzne prz Kolegium MISH, Warszawa 2004, zob. też: http://kf.mish.uw.edu.pl/kog/kog_mac.pdf

[Portner, Partee 2002] P. PORTNER, B. H. PARTEE (red.) *Formal Semantics. The Essential Readings*, Blackwell Publishing 2002.

[Pullum, Gazdar 1982] G.K. PULLUM, G. GAZDAR *Natural languages and context–free grammars*, **Linguistics and Philosophy** 4, 1982, str.471–504.

[Rosenberg, Salomaa 1997] G. ROSENBERG, A. SALOMAA (red.) *Handbook of Formal Languages*, Springer Verlag, Berlin Heidelberg 1997.

[Sevenster 2006] M. SEVENSTER *Branching Imperfect Information. Logic, Language, and Computation*, rozprawa doktorska, Institute for Logic, Language and Information, University of Amsterdam.

[Shieber 1985] S.M. SHIEBER *Evidence against the context–freeness of natural language*, **Linguistics and Philosophy**8, 1985, str.333–343.

[Suppes 1982] P. SUPPES *Variable-Free Semantics with Remarks on Procedural Extensions*, **Language, Mind, and Brain**, pod red. H. Bunta, I. Sluisa, R. Morante'a, 1982, str. 21 – 31.

[Szymanik 2007] J. SZYMANIK *A comment on a neuroimaging study of natural language quantifiers*, **Neuropsychology**, 45, str. 2158–2160.

[Tarski 1933] A. TARSKI *Pojęcie prawdy w językach nauk dedukcyjnych*, **Prace Towarzystwa Naukowego Warszawskiego, Wydział III Nauk Matematyczno-Fizycznych** 34, Warszawa 1933, str. 7 – 116. Przedruk w: **Pisma logiczno - filozoficzne** tom 1 „Prawda”, pod red. J. Zygmunta, PWN, Warszawa 1995.

[Tichy 1969] P. TICHY *Intension In Terms Of Turing Machines*, **Studia Logica** XXIV (1969), str. 7 – 23.

[Tichy 1971] P. TICHY *An Approach to Intensional Analysis*, **Nous** Vol. 5, No. 3 (1971).

[Thomason 1974] R. H. THOMASON (red.) *Formal Philosophy. Selected Papers of Richard Montague*, Yale University Press 1974.

[Turing 1936] A. TURING *On Computable Numbers, with an application to the Entscheidungsproblem*, **Proceedings of the London Mathematical Society** 42 (1936), str. 230 – 265.

[Westerståhl, Peters 1989] D. WESTERSTÅHL, P. STANLEY *Quantifiers in Language And Logic*, Oxford University Press, Oxford 2006.