

SZYMON MACHOWSKI

Informatyczna implementacja algorytmów binarnej syntezy i dychotomizacji zdań*

Abstract (Computer Implementation of Binary Synthesis and Sentence Dichotomization Algorithms). The paper aims to show how to implement binary synthesis and sentence dichotomization algorithms in the most popular programming languages. In the first part the author discusses both algorithms from a theoretical perspective, using the concepts from Bańczerowski's axiomatic theory of syntax. The sentence dichotomization algorithm is described as a series of stages of the temporal break-up of syntactic structures, starting from sentence division into the nominal and verbal phrase, and finishing with the division of minimal hypotactic or paratactic structures into single words. The binary synthesis algorithm is described in terms of a multi-stage syntactic relation which constructs syntactic relationships, allowing for collocational acceptability. This algorithm operates on a principle contrary to the sentence dichotomization algorithm. In the second part of the paper the author argues that the computer implementation of both algorithms is an effectively decidable problem. He also postulates the construction of three balanced parallel corpuses of the Polish language. Every syntactic structure is expressed by means of an operation on declared variables, which are matched with words belonging to specific parts of speech from the corpuses of Polish. Finally, both algorithms are presented by means of a set of programming commands.

Abstrakt. Celem artykułu jest przedstawienie sposobu implementacji algorytmów binarnej syntezy i dychotomizacji zdań w najpopularniejszych językach programowania. W pierwszej części artykułu oba algorytmy przedstawione są w ujęciu teoretycznym przy użyciu pojęć z aksjomatycznej teorii składni prof. Bańczerowskiego. Algorytm dychotomizacji zdań opisany jest jako szereg etapów temporalnego rozpadu struktur syntaktycznych, począwszy od podziału zdania na frazę nominalną i werbalną, a skończywszy na podziale minimalnych konstrukcji hipotaktycznych lub parataktycznych na pojedyncze wyrazy. Algorytm binarnej syntezy zdań przedstawiono jako wieloetapową relację syntaktyczną konstruującą związki syntaktyczne z uwzględnieniem ich dopuszczalności frazeologicznej, który działa na zasadzie odwrotnej do algorytmu dychotomizacji zdań. W drugiej części artykułu informatyczną implementację obu algorytmów określono jako problem efektywnie rozstrzygalny. Postulowano opracowanie trzech zrównoważonych,

* Niektóre z tez Autora wzbudziły kontrowersje w gronie zespołu redakcyjnego, niemniej tematyka poruszana w artykule rzadko występuje na łamach naszego czasopisma, dlatego redakcja zdecydowała się na jego opublikowanie.

równoległych korpusów języka polskiego. Każdą konstrukcję syntaktyczną wyrażono za pomocą operacji na zadeklarowanych zmiennych, do których przypisano wyrazy należące do określonych części mowy z utworzonych korpusów polszczyzny. Ostatecznie oba algorytmy przedstawiono w postaci zestawu komend programistycznych.

1. Wstęp

Celem pracy jest przedstawienie algorytmów binarnej syntezy i dychotomizacji zdań z danego zbioru leksonów wyłącznie w oparciu o aksjomatyczną teorię syntaktyczną, zaprezentowaną przez prof. dr. hab. Jerzego Bańcerowskiego w trakcie wykładów ze składni ogólnej na Uniwersytecie im. Adama Mickiewicza w roku akademickim 2006/2007. Opiekę naukową nad informatyczną implementacją obu algorytmów sprawuje dr Włodzimierz Lapis.

W pierwszej części niniejszego artykułu omówione zostaną teoretyczne podstawy informatycznej implementacji algorytmów binarnej syntezy i dychotomizacji zdań, w tym w szczególności relacji syntaktycznych konstytuujących syntagmy, frazy i zdania. Podane zostaną również wstępne postulaty stosownej teorii w postaci aksjomatów umożliwiające algorytmizację temporalnego podziału i syntezy struktur syntaktycznych. Następnie omówione zostanie samo działanie obu schematów na przykładach konkretnego zbioru zdań języka naturalnego.

Druga część pracy traktuje o możliwości implementacji powyższych schematów w ograniczeniu do języka polskiego w obecnie najpowszechniejszych systemach operacyjnych, takich jak Microsoft Windows, Linux i Unix, poprzez opracowanie programu do automatycznego dychotomizowania i syntezy syntagm i fraz pochodzących z fragmentów tekstów celem dokonania analizy ich struktury kwalifikacyjnej. W pierwszej kolejności rozważone będą kwestie wdrożenia pojęć specyficznie językoznawczych do systemu komputerowego, a następnie przedstawione zostaną planowane etapy implementowania obu algorytmów.

Wyniki tej pracy mogą mieć szerokie zastosowanie, jak chociażby w przedstawionym poniżej przykładzie z glottodydaktyki. Nauczyciel języka obcego lub ojczystego może wykorzystać zaimplementowane algorytmy binarnej syntezy i dychotomizacji zdania jako pomoc dydaktyczną w nauczaniu dokonywania rozbioru zdań pojedynczych i złożonych oraz konstruowania frazeologicznie poprawnych w danym języku syntagm hipotaktycznych i parataktycznych.

2. Algorytmizacja binarnej syntezy i dychotomizacji zdań w świetle aksjomatycznej teorii syntaktycznej

Jak wspomniano we wstępie, oba schematy wykorzystują następujące pojęcia, relacje i ich symboliczny zapis w ramach aksjomatycznej teorii składni prof. Bańcerowskiego:

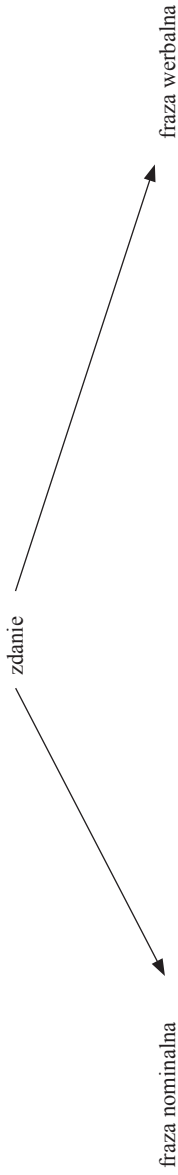
- (i) **Lex** – zbiór leksonów (zbiór aktualnych leksonów)
- (ii) **gsyf** – relacja tworzenia całości (z leksonów, syntagm bądź fraz)
- (iii) **LNG** – zbiór wszystkich języków
- (iv) **ln** – relacja bycia leksonem w zdaniu
- (v) **Wdt** – relacja determinacji wyrazowej
- (vi) **Wpt** – relacja parataksy wyrazowej
- (vii) **Cgsm** – relacja kongruencji semicznej
- (viii) **Rcsm** – relacja rekcji semicznej
- (ix) **Adlk** – relacja adjunkcji leksykalnej
- (x) **fqec** – relacja equikohezji kwalifikacyjnej
- (xi) **fqc** – relacja kohezji kwalifikacyjnej nierównego stopnia
- (xii) **shqdt** – relacja „bycia syntagmą dychotomizowalną kwalifikacyjnie w”
- (xiii) **frqdt** – relacja „bycia frazą dychotomizowalną kwalifikacyjnie w”

2.1. Algorytmizacja schematu dychotomizacji zdań

Jako wyjściowy potraktować należy algorytm dychotomizacji zdań, gdyż schemat binarnej syntezy zdań działa na zasadzie odwrotnej do owego schematu. Poniżej omówiony jest algorytm dychotomizacji zdań, a bezpośrednio pod jego opisem dla lepszego zobrazowania podany zostanie jego schemat.

W pierwszym etapie dzieli on aktualne zdanie na frazę nominalną i werbalną poprzez rozłączenie fraz, które są poprzednikami relacji bycia frazą dychotomizowalną kwalifikacyjnie **frqdt** w danym zdaniu t i połączone relacją tworzenia całości **gsyf** stanowią zdanie t . Następnie niezależnym od siebie podziałom dychotomicznym podlegają osobno obie frazy.

Jako pierwsza rozważona zostanie fraza nominalna. Użytkownik języka musi rozstrzygnąć, ile leksonów wchodzi w jej skład. Jeżeli składają się na nią tylko dwa leksony, to przystąpić należy do dychotomizacji pary minimalnej, co wiąże się z koniecznością określenia typu syntagmy, którą ona reprezentuje. Jeżeli leksony znajdują się w relacji determinacji wyrazowej **Wdt**, w której jeden z nich jest determinansem, a drugi determinatum, to należy dokonać dychotomizacji minimalnej konstrukcji hipotaktycznej na kwalifikator i kwalifikatum. Jeśli z kolei rozważana syntagma nie jest minimalną konstrukcją hipotaktyczną, to musi być minimalną konstrukcją parataktyczną, która dzieli się na 2 leksony. Jeśli zatem fraza nominalna składa się z większej liczby leksonów niż 2, to użytkownik języka musi rozstrzygnąć, czy poszczególne grupy leksonów frazy nominalnej powiązane są także relacjami równego i nierównego stopnia kohezji kwalifikacyjnej (odpowiednio **fqec** i **fqc**). Jeśli tak, to w pierwszej kolejności dychotomizacji podlega fraza nominalna przez rozbitcie relacji kohezji nierównego stopnia **fqc**, łączącej leksony będące w najbliższym otoczeniu determinacyjnym, wskutek czego otrzymujemy syntagmy z leksonami w relacji nierównego stopnia kohezji kwalifikacyjnej **fqec** lub syntagmę w relacji równego stopnia kohezji kwalifikacyjnej **fqec** i leksony. W dalszym etapie następuje dychotomizacja wszystkich konstrukcji syntaktycznych w relacji equikohezji kwalifikacyjnej **fqec** oraz wszystkich minimalnych konstrukcji hipotaktycznych na kwalifikatory i kwalifikata.



Aktualne zdanie frazowo dychotomizuje się na frazę nominalną i frazę werbalną.
 Grafy frazowej dychotomizacji frazy nominalnej i werbalnej znajdują się odpowiednio na stronach 36 i 37 oraz 38 i 39.

FRAZA NOMINALNA

Czy fraza nominalna składa się tylko z 2 leksonów?



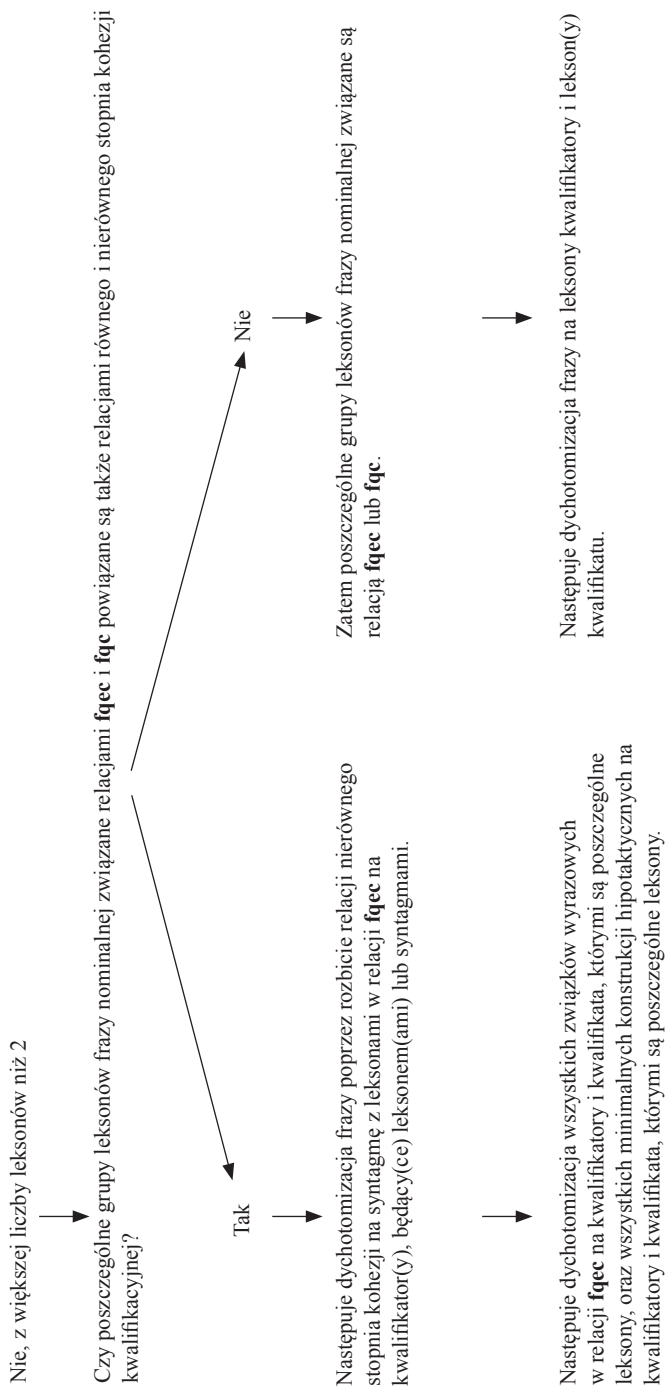
Czy 2 leksony pozostają w relacji **Wdf**?



Następuje dychotomizacja minimalnej konstrukcji hipotaktycznej na kwalifikator i kwalifikatum.

Zatem leksony pozostają w relacji **Wpt**.
 Następuje dychotomizacja minimalnej konstrukcji parataktycznej na dwa leksony.

FRAZA NOMINALNA



FRAZA WERBALNA

Czy fraza werbalna składa się tylko z 2 leksonów?

Tak



Czy 2 leksony pozostają w relacji determinacji?

Tak



Następuje dychotomizacja minimalnej konstrukcji hipotaktycznej na kwalifikator i kwalifikatum.

Nie

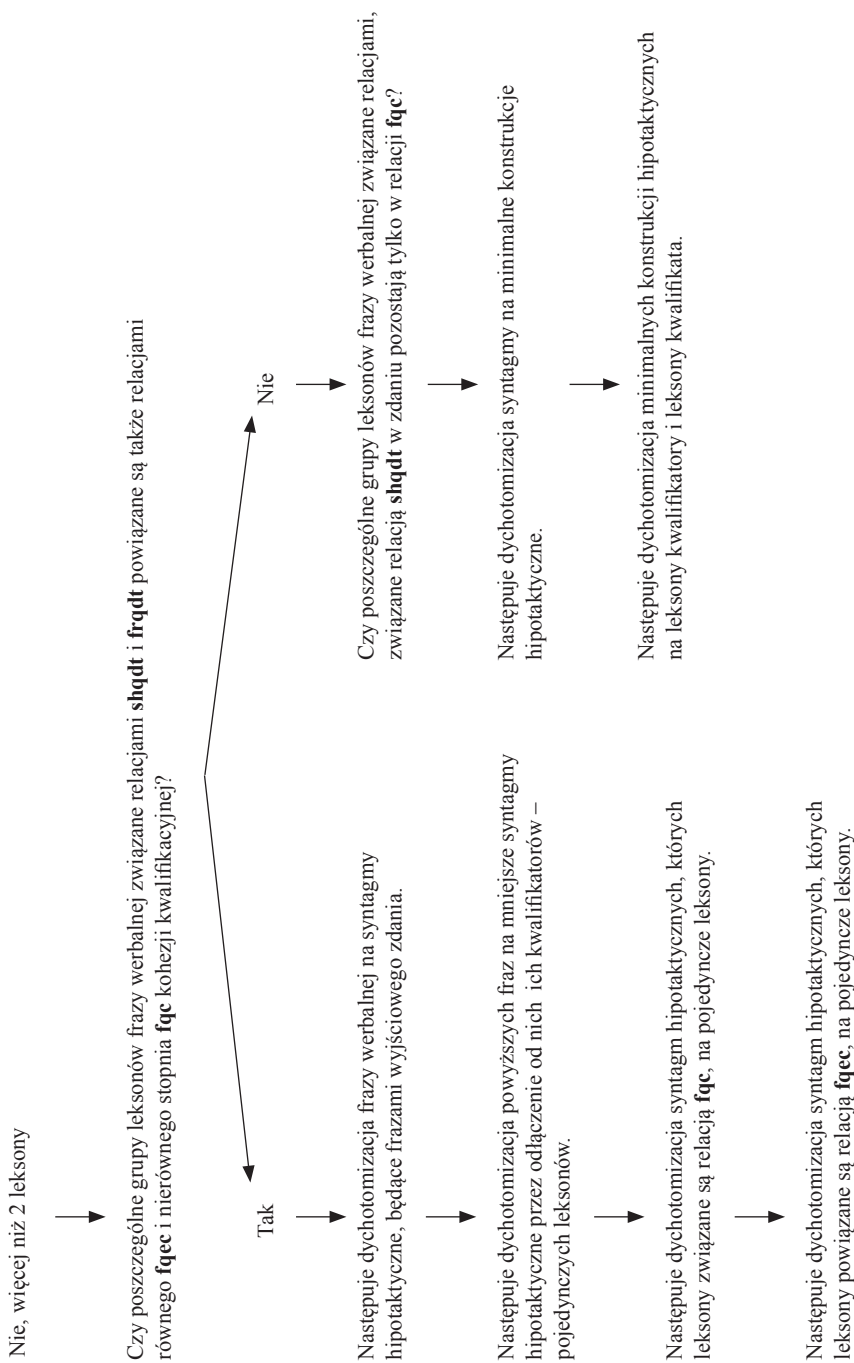
Nie, z większej liczby leksonów niż 2 (kontynuacja na następnej stronie)

Nie



Zatem leksony pozostają w relacji parataksy wyrazowej **wpt**. Następuje dychotomizacja minimalnej konstrukcji parataktycznej na 2 leksony.

FRAZA WERBALNA



Działanie powyższego schematu dychotomizacji zdania ilustruje poniższy przykład:

Doświadczeni i odważni turyści z dużymi plecakami pospiesznie wyruszyli na wycieczkę.

Fraza nominalna:

Fraza werbalna:

doświadczeni turyści z dużymi plecakami

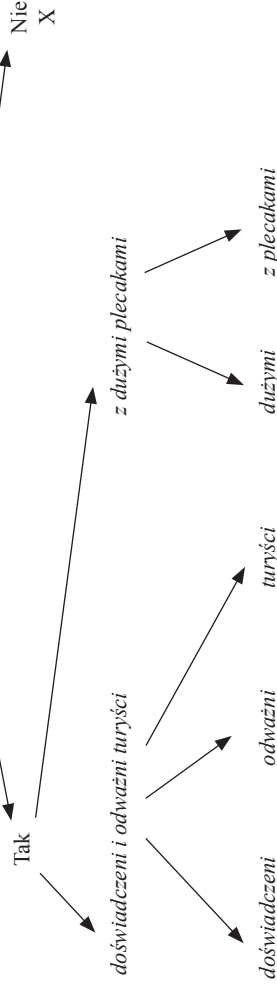
wyruszyli pospiesznie na wycieczkę

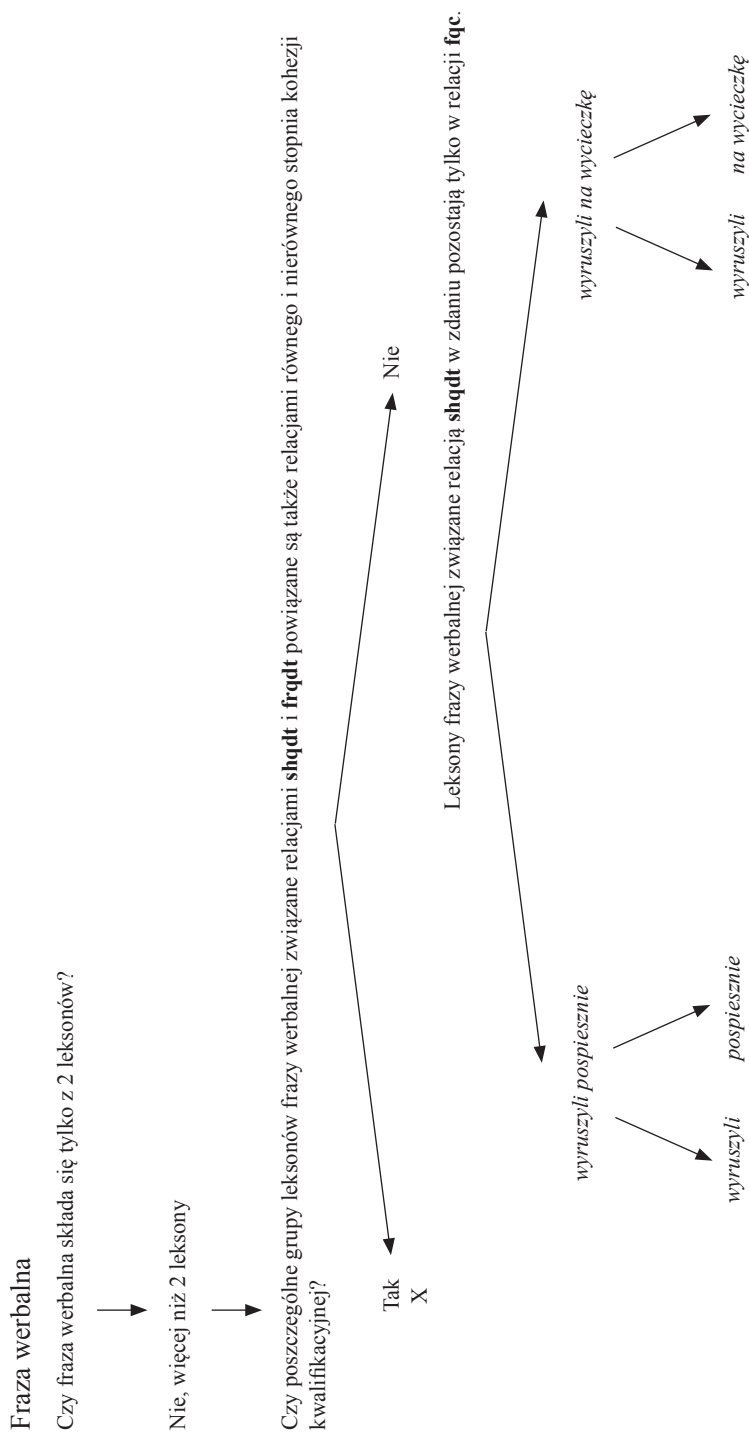
Fraza nominalna

Nie, z większej liczby leksonów niż 2



Czy poszczególne grupy leksonów frazy nominalnej związane relacjami **fqc** i **fqc** powiązane są także relacjami równego i nierównego stopnia kohezji kwalifikacyjnej?





Jeżeli z kolei leksony frazy nominalnej nie są połączone koniunkcją relacji equikohezji **fqec** i kohezji kwalifikacyjnej nierównego stopnia **fqc**, to poszczególne grupy aktualnych wyrazów połączone są alternatywnie relacją equikohezji **fqec** lub kohezji nierównego stopnia **fqc**, zatem wiąże je tylko jedna z wyżej wymienionych relacji. W związku z powyższym następuje dychotomizacja frazy nominalnej na leksony kwalifikatory i lekson(y) kwalifikatum(a).

Analogicznie do frazy nominalnej należy rozważyć dychotomizację frazy werbalnej. Użytkownik języka musi rozstrzygnąć, ile leksonów wchodzi w jej skład. Jeżeli składają się na nią tylko 2 leksony, to przystąpić należy do dychotomizacji pary minimalnej, co wiąże się z koniecznością określenia typu syntagmy, którą ona reprezentuje. Jeżeli leksony znajdują się w determinacji wyrazowej **Wdt**, w której jeden z nich jest determinansem, a drugi determinatum, to należy dokonać dychotomizacji minimalnej konstrukcji hipotaktycznej na kwalifikator i kwalifikatum. Jeśli zaś rozważana syntagma nie jest minimalną konstrukcją hipotaktyczną, to musi być minimalną konstrukcją parataktyczną, która dzieli się na dwa leksony.

Jeśli z kolei fraza werbalna składa się z większej liczby leksonów niż 2, to użytkownik języka musi rozstrzygnąć, czy jej poszczególne grupy leksonów powiązane są także relacjami równego i nierównego stopnia kohezji kwalifikacyjnej (odpowiednio **fqec** i **fqc**). Jeśli tak, to fraza werbalna dychotomizuje się na syntagmy hipotaktyczne będące frazami i zarazem będące poprzednikami relacji bycia frazą dychotomizacyjną **shqdt** wyjściowego zdania *t*. W dalszej kolejności frazy dychotomizują się na mniejsze syntagmy hipotaktyczne przez odłączenie od nich pojedynczych leksonów, kwalifikatorów, wskutek czego pozostają syntagmy z leksonami złączonymi relacją equikohezji kwalifikacyjnej **fqec** oraz leksony – ich kwalifikatory. Później następuje dychotomizacja syntagm hipotaktycznych związanych relacją equikohezji kwalifikacyjnej **fqec** na pojedyncze leksony. Na końcu dychotomizują się syntagmy hipotaktyczne, których leksony związane są relacją kohezji równego stopnia **fqec** na pojedyncze leksony. Jeżeli z kolei w grupie leksonów nie współistnieją relacje kohezji kwalifikacyjnej równego i nierównego stopnia (odpowiednio **fqec** i **fqc**), to poszczególne grupy aktualnych wyrazów związane są relacją kohezji nierównego stopnia **fqc**, które dychotomizują się na minimalne konstrukcje hipotaktyczne, podlegające potem podziałowi na leksony kwalifikatory i leksony kwalifikata.

2.2. Algorytmizacja schematu binarnej syntezy zdania

Poniżej omówiony jest algorytm binarnej syntezy zdania, a bezpośrednio pod opisem dla lepszego zobrazowania znajduje się jego schemat.

Binarna synteza zdania **Bisyn**, została zdefiniowana jako wieloetapowa temporalna relacja syntaktyczna, która w pierwszej kolejności tworzy z pojedynczych aktualnych wyrazów minimalne konstrukcje hipotaktyczne poprzez relację tworzenia całości **gsyf**, a następnie przez relację tworzenia całości **gsyf** buduje z nich frazy i syntagmy zdania *t*, w których leksony związane są z sobą również relacjami kohezji równego **fqec** i nierównego stopnia **fqc**.

Prawidłowość działania relacji binarnej syntezy zdania warunkują dwa następujące postulaty:

- (i) $\mathbf{Bisyn}_t \subset \mathbf{Lex} \times \mathbf{Lex}$
- (ii) $\mathbf{Bisyn}_t(x,y) \leftrightarrow [(x,y) \in \mathbf{Lex} \wedge \mathbf{gsyf}^*(x,y) \in \mathbf{LNG}]$

Pierwszy z nich stwierdza, że binarna synteza zdania zachodzi w zbiorze leksonów.

Drugi z nich zakłada, że zsintezowane konstrukcje hipotaktyczne lub parataktyczne muszą być dopuszczalnymi i istniejącymi w danym języku związkami syntaktycznymi.

W pierwszych trzech etapach z rozsypanki leksonów tworzone są minimalne konstrukcje hipotaktyczne, których człony znajdują się w relacji kongruencji semicznej **Cgsm**, rekcji semicznej **Rcsm** i adiunkcji leksykalnej **Adlk**. Dany lekson może wystąpić w więcej niż jednej parze i być w nich zarówno determinatorem, jak i determinansem.

W czwartym kroku następuje synteza powstałych konstrukcji hipotaktycznych, których człony znajdują się w relacji equikohezji kwalifikacyjnej **fqec**, w dychotomizowalne kwalifikacyjnie syntagmy **shqdt** aktualnego zdania *t*.

W piątym etapie syntetyzują się minimalne konstrukcje hipotaktyczne, których człony znajdują się bezpośrednio w relacji nierównego stopnia kohezji kwalifikacyjnej **fqc**, w dychotomizowalne kwalifikacyjnie syntagmy **shqdt** aktualnego zdania *t*.

W ostatnim etapie temporalnej relacji binarnej syntezy zdania **Bisyn**, dochodzi do zsintezowania się dwóch dychotomizowalnych kwalifikacyjnie syntagm **shqdt** zdania *t* w aktualne zdanie *t* za pomocą relacji tworzenia całości **gsyf**.

Tabela 1

Numer etapu Bisyn_t	Zapis symboliczny	Przykład na aktualnych wyrazach
I	gsyf [*] (a,c) ∈ Cgsm gsyf [*] (b,c) ∈ Cgsm	zdolny uczeń pracowity uczeń
II	gsyf [*] (v,y) ∈ Rcsm gsyf [*] (v,z) ∈ Rcsm	jechać autobusem jechać do szkoły
III	gsyf [*] (v,x) ∈ Adlk	jechać szybko
IV	$[c,(a,b)] \in \mathbf{fqec}$ $[v (y,z)] \in \mathbf{fqec}$	zdolny uczeń } zdolny, pracowity uczeń pracowity uczeń jechać autobusem } jechać autobusem do szkoły jechać do szkoły
V	$[v (y,z,x)] \in \mathbf{fqc}$	jechać autobusem do szkoły } jechać szybko autobusem do szkoły jechać szybko
VI	gsyf [*] $[[c(a,b)], [v (y,z,x)]] = t_t$	Zdolny, pracowity uczeń jedzie autobusem do szkoły.

3. Implementacja algorytmu binarnej syntezy i dychotomizacji zdania do postaci programu komputerowego

Algorytmizacja schematów binarnej syntezy i dychotomizacji zdania jako problem efektywnie rozstrzygalny pozwala dokonać implementacji w najpopularniejszych językach programowania, takich jak np. JavaScript, TurboPascal czy Delphi, z wykorzystaniem w przeważającej mierze programowania imperatywnego w czwartym języku generacji.

3.1. Zgromadzenie zasobów leksykalnych i preparacja narzędzi informatyczno-językoznawczych do implementacji algorytmów binarnej syntezy i dychotomizacji zdania

3.1.1. Zasoby leksykalne jako korpusy

Zadanie opracowania programu do analizy tekstu pod kątem zrekonstruowania struktury kwalifikacyjnej jego zdań wraz z możliwością zbudowania przez komputer nowych zdań z ich leksonów wymaga w pierwszej kolejności zgromadzenia odpowiednich zasobów leksykalnych, przez odniesienie do których system będzie właściwie rozpoznawał wyrazy i łączył je z sobą. Program będzie służył do pracy nad tekstami różnego rodzaju w języku polskim, począwszy od artykułów prasowych, popularnonaukowych, poprzez utwory literackie, a skończywszy na dziełach specjalistycznych. Wskazane zatem byłoby połączenie gotowych bądź opracowanie nowych trzech lingwistycznych zrównoważonych i równoległych korpusów języka polskiego. Pierwszy z nich może być zbiorem lemm wyrazów i słów języka polskiego w kolejności alfabetycznej wraz z przyporządkowaniem im nazw reprezentowanych przez nich części mowy, drugi może być zbiorem form fleksyjnych wyrazów z pierwszego korpusu sprzężonych z nim za pomocą operacji przypisania, a trzeci – zbiorem poprawnych w polszczyźnie syntagm i frazeologizmów zbudowanych z wyrazów z pierwszego i drugiego słownika.

3.1.2. Narzędzie do dystynkcji wyrazów i vocabulonów (słów)

Należy także opracować narzędzie, które w ramach zasobów leksykalnych w trzech powyższych korpusach będzie pozwalało odróżnić aktualne wyrazy od vocabulonów. Owo rozgraniczenie terminologiczne wywodzi się stąd, iż lekson rozumiany jest jako jednostka leksykalna posiadająca znacznie leksykalne, semiczne, syntaktyczne i dystrybtywne, a vocabulon jako jednostka leksykalna o znaczeniu semicznym, syntaktycznym i dystrybtywnym.

Zatem zbiór słów inkorporuje zbiór wyrazów. Ograniczenie się wyłącznie do leksyki polszczyzny pozwala wyróżnić następujący zbiór części mowy, których elementy stanowią wyrazy: rzeczownik, czasownik osobowy, przymiotnik, zaimek, liczebnik, przysłówek, imiesłów, gerundium, bezokolicznik, oraz części mowy, których elementy stanowią **tylko** vocabulony: przyimek, rodzajnik, spójnik i partykuła.

Ponadto – z praktycznego punktu widzenia – do zbioru leksonów należą także wyrażenia przyimkowe.

Ze względu na zastosowanie rozróżnienia między słowami i wyrazami w algorytmach dychotomizacji i syntezy zdania omówiony zostanie jeszcze przypadek, w którym lekson kwalifikator znajduje się w środku wyrażenia przyimkowego będącego kwalifikatorem, na przykład *w dużym domu*. W takim wypadku program musi sprawdzić z teoretycznego punktu widzenia, czy w najbliższym, a potem sąsiednim otoczeniu determinacyjnym przyimka znajduje się rzeczownik, z którym poprzez relację tworzenia całości **gsyf** stanowi lekson, który jest elementem zbioru leksykalnego korpusu.

Programista może zdefiniować zbiory wyrazów i tylko słów, wykorzystując początkowo instrukcję warunkową, a potem pętlę, która może przyjąć następującą postać: jeżeli dana zmienna x jako jednostka leksykalna należy do zbioru rzeczowników lub czasowników osobowych lub przymiotników, zaimków, liczebników, przysłówków, imiesłówów, gerundów, bezokoliczników, to wyświetl napis **wyraz**, jeżeli nie, to czy x i sąsiednie y tworzą wyrażenie przyimkowe zadeklarowane w korpusie, jeżeli nie, to czy x i kolejne z tworzą wyrażenie przyimkowe, jeżeli nie, ..., aż do końca frazy, jeżeli nie, to wyświetl napis **tylko vocabulon**.

3.1.3. Eliminacja redundantnych semantycznie słów

Celem bezbłędnego przeprowadzenia analizy tekstu pisanego, a szczególnie transkrypty dialogu, pod kątem zrekonstruowania jego struktury kwalifikacyjnej programista winien wprowadzić moduł eliminacji partykuł i wykrzykników wyrażających stosunek emocjonalny autora wypowiedzi. Może ona nastąpić poprzez zidentyfikowanie owych słów z ich listą zadeklarowaną uprzednio w korpusie oraz wymazanie ich w wersji roboczej tekstu, jak w poniższym przykładzie:

fragment zapisanego dialogu:

*Och, jest tu niesamowicie duszno! Czy mogłaby pani otworzyć okno?
Tak, zaraz je otworzę.*

gotowy do analizy fragment dialogu po eliminacji słów redundantnych semantycznie:

*Jest tu niesamowicie duszno. Mogłaby pani otworzyć okno.
Zaraz je otworzę.*

Eliminacja redundantnych semantycznie słów może się wiązać ze zmianą trybu zdań z interrogatywnego, aklamacyjnego na oznajmujący, która jednak nie wpływa na jego strukturę kwalifikacyjną.

3.2. Ekstrakcja związku głównego z poszczególnych aktualnych zdań

Jako że polszczyzna jest silnie fleksyjna oraz charakteryzuje się dość swobodnym szykiem wyrazów w zdaniach, człony związku głównego mogą się znajdować w dość dalekim otoczeniu determinacyjnym względem siebie. Wiąże się to z koniecznością

znalezienia przez program w pierwszej kolejności orzeczenia prostego lub złożonego, następnie rzeczownika o takiej samej liczbie cech semicznych: liczby i ewentualnie rodzaju. Poleca się tutaj zastosowanie instrukcji wyliczenia i warunkowej o następującej postaci:

1. wskaż czasownik osobowy,
2. znajdź rzeczownik lub rzeczowniki o takiej samej liczbie i rodzaju, co czasownik osobowy,
3. jeżeli nie ma, to znajdź przymiotnik o takiej samej liczbie i rodzaju, co czasownik osobowy,
4. jeżeli nie ma, to znajdź zaimek o takiej samej liczbie i rodzaju, co czasownik osobowy,
5. jeżeli nie ma, to znajdź liczebnik o takiej samej liczbie i rodzaju, co czasownik osobowy,
6. jeżeli nie ma, to znajdź imiesłów o takiej samej liczbie i rodzaju, co czasownik osobowy,
7. jeżeli nie ma, to znajdź bezokolicznik,
8. jeżeli nie ma, to napisz „zdanie bezpodmiotowe lub zdanie o podmiocie domyślnym”.

3.3. Informatyczna implementacja pojęć specyficznie językoznawczych

Implementacja relacji syntaktycznych polega na określeniu, jakie elementy konkretnych zbiorów części mowy wymienionych wcześniej będą wchodziły z sobą w różne rodzaje spójności semantycznej.

3.3.1. Informatyczna implementacja relacji kongruencji semicznej C_{gsm}

Relacja kongruencji semicznej wiąże dwa wyrazy, gdy należą przynajmniej do jednej tej samej kategorii semicznej, czyli są zgodne co do liczby, rodzaju i przypadku. Pod kątem implementacji są to pary wyrazów, z których jeden jest przymiotnikiem, a drugi rzeczownikiem. Można zastosować tu instrukcję warunkowania: jeśli x jest przymiotnikiem, a y rzeczownikiem, lub na odwrót, lub suma x i y to związek główny, to wyświetl C_{gsm} .

3.3.2. Informatyczna implementacja relacji rekcji semicznej R_{csm}

Relacja rekcji semicznej R_{csm} wiąże dwa wyrazy W_i i W_j wtedy i tylko wtedy, gdy W_j określa W_i , przy czym wyraz W_i rządzi kategorią semiczną W_j . Z punktu widzenia programisty związane nią są pary aktualnych wyrazów w następujących konfiguracjach:

1. Jeden wyraz jest czasownikiem osobowym, a drugi rzeczownikiem w dopełniaczu lub celowniku, lub bierniku, lub narzędniku,
2. Jeden wyraz jest rzeczownikiem, a drugi rzeczownikiem w dopełniaczu lub narzędniku, lub miejscowniku.

Rekomenduje się zastosowanie instrukcji warunkowej, która może przyjąć postać:

- Jeśli x jest czasownikiem osobowym, a y jest rzeczownikiem w dopełniaczu lub celowniku, lub bierniku, lub narzędniku, to wyświetl **Rcsm**. Jeśli x jest rzeczownikiem i y jest rzeczownikiem w dopełniaczu lub narzędniku, lub miejscowniku, to wyświetl **Rcsm**.

3.3.3. Informatyczna implementacja relacji adiunkcji leksykalnej Adlk

W teorii językoznawczej adiunkcję leksykalną **Adlk** definiuje się jako relację łączącą dwa wyrazy W_i i W_j , gdy W_j określa W_i , przy czym W_j nie należy do żadnej kategorii semicznej określonej. Oba leksy wiąże wyłącznie spójność leksykalna.

Programista może dokonać implementacji relacji adiunkcji leksykalnej **Adlk** poprzez zastosowanie, tak jak we wcześniejszych krokach, instrukcji warunkowej o następującej postaci: jeżeli x jest czasownikiem osobowym, a y jest przysłówkiem, lub na odwrót, to wyświetl **Adlk**.

3.3.4. Informatyczna implementacja relacji parataktycznych wpt

Relacja parataksy wyrazowej **wpt** łączy dwa wyrazy, które są syntaktycznie jednorodnie i równorzędne. Zalicza się do niej relacje wyrazowej kopulatywności, adwersatywności, wyrazowej dysjunkcji lub alternatywy. Programista może ją określić w postaci koniunkcji warunków: jeżeli x i y należą do tej samej syntagmy i nie są związane relacją **wdt** i jeżeli między x i y jest spójnik zadeklarowany w korpusie lub jest przecinek, to wyświetl **wpt**, jeśli nie, to wyświetl **bląd**.

3.3.5. Informatyczna implementacja relacji equikohezji kwalifikacyjnej (kohezji równego stopnia) fqc i kohezji nierównego stopnia fqc

Obie relacje syntaktyczne konstytuują pojęcie spójności semicznej par determinacyjnych.

Leksyony w parach najsilniej wiąże relacja kongruencji semicznej **Cgsm**, a relacja rekacji semicznej **Rcsm** odzwierciedla silniejszą spójność semiczną niż adiunkcja leksykalna **Adlk**.

a) Informatyczna implementacja relacji kohezji równego stopnia fqc

Wiąże ona z sobą frazy złożone co najmniej z trzech leksonów, dwóch kwalifikatorów i jednego leksonu kwalifikatum. Oba leksony kwalifikatory w równym stopniu kwalifikują lekson kwalifikatum. Całą syntagmę można zatem podzielić na dwie minimalne konstrukcje hipotaktyczne o tym samym stopniu kohezji.

Dla programisty jest to związek co najmniej trzech różnych wyrazów x, y, z w następujących konfiguracjach:

1. x i y to przymiotniki, a z jest rzeczownikiem,
2. x i y to rzeczowniki w różnych przypadkach, a z to czasownik osobowy,
3. x i y to rzeczowniki w różnych przypadkach, a z to rzeczownik,
4. x i y to przysłówki, a z jest czasownikiem.

Powyższy zapis można implementować za pomocą następującej instrukcji warunkowej:

- Jeżeli x i y to przymiotniki i z jest rzeczownikiem lub jeżeli x i y to rzeczowniki w różnych przypadkach i z jest czasownikiem osobowym, lub x i y to rzeczowniki w różnych przypadkach i z jest rzeczownikiem, to wyświetl **fqec**.

b) Informatyczna implementacja relacji kohezji nierównego stopnia **fqc**

Wiąże ona z sobą frazy złożone co najmniej z trzech leksonów: dwóch kwalifikatorów i jednego kwalifikatum. Oba leksony kwalifikatory w różnym stopniu kwalifikują lekson kwalifikatum. Całą syntagmę można zatem podzielić na dwie minimalne konstrukcje hipotaktyczne o różnych stopniach kohezji. Na potrzeby informatycznej implementacji można ją ująć jako związek co najmniej trzech różnych wyrazów x , y , z w następujących konfiguracjach:

1. x jest przymiotnikiem, y jest rzeczownikiem w określonym przypadku, a z jest czasownikiem osobowym,
2. x jest przymiotnikiem, y jest rzeczownikiem w określonym przypadku, a z jest rzeczownikiem w mianowniku,
3. x jest rzeczownikiem, y jest przysłówkiem, a z jest czasownikiem osobowym.

Powyższy zapis można wyrazić następującą instrukcją warunkową:

- Jeżeli x jest przymiotnikiem i y jest rzeczownikiem w określonym przypadku i z jest czasownikiem osobowym lub jeżeli x jest przymiotnikiem i y jest rzeczownikiem w określonym przypadku i z jest rzeczownikiem w mianowniku, lub jeżeli x jest rzeczownikiem, y jest przysłówkiem, a z jest czasownikiem osobowym, to wyświetl **fqc**.

3.3.6. Ekstrakcja frazy werbalnej z aktualnego zdania

Fraza werbalna jest frazą dychotomizowalną kwalifikacyjnie **frqdt** aktualnego zdania t , w którym kwalifikator związku głównego jest maksymalnym kwalifikatorem całej frazy. Za pomocą wcześniej zaimplementowanych relacji syntaktycznych komputer może dokonać ekstrakcji frazy werbalnej przy wykorzystaniu następującej instrukcji warunkowej: jeżeli y jest czasownikiem osobowym i kwalifikatorem w związku głównym, to wypisz frazę lub wszystkie frazy typu **fqc** i **fqec** lub **Rcsm**, lub **Adlk**, w których on występuje.

3.4. Ekstrakcja frazy nominalnej z aktualnego zdania

Fraza nominalna jest frazą dychotomizowalną kwalifikacyjnie **frqdt** aktualnego zdania t , w którym kwalifikatum związku głównego jest maksymalnym kwalifikatorem całej frazy. Wykorzystując omówione wcześniej sposoby implementacji relacji syntaktycznych, komputer może wydzielić frazę nominalną z całego zdania za pomocą instrukcji warunkowej o następującej postaci: jeżeli x jest rzeczownikiem lub przymiotnikiem, a y czasownikiem w związku głównym, to wypisz frazę lub wszystkie frazy typu **fqc** i **fqec** lub **Cgsm**, lub **Rcsm** poza x **Cgsm** y , w których x występuje.

3.6. Informatyczna implementacja algorytmu binarnej syntezy zdania

Po dokonaniu dychotomizacji utworzył się zbiór wyrazów, z których aktualne zdanie zostało pierwotnie skonstruowane. Teraz program ma za zdanie przeprowadzenie kilkakrotnej binarnej syntezy zdania z otrzymanych leksonów. Zalecane jest tymczasowe automatyczne zapisanie wyników pierwszej operacji i jej powtórzenie z wykorzystaniem operacji iteracji następującego schematu dopóty, dopóki za każdym razem powstaną inne zdania.

Ta część programu może się składać z następującego zestawu komend:

1. Wypisz leksony zdania w postaci słownikowej.
2. Połącz odpowiednie leksony w pary, aby utworzyć wszystkie możliwe minimalne konstrukcje hipotaktyczne, których człony znajdują się w relacji kongruencji semicznej.
3. Połącz odpowiednie leksony w pary, aby utworzyć wszystkie możliwe minimalne konstrukcje hipotaktyczne, których człony znajdują się w relacji rekcji semicznej.
4. Połącz odpowiednie leksony w pary, aby utworzyć wszystkie możliwe konstrukcje hipotaktyczne, których człony znajdują się w relacji adiunkcji leksykalnej.
5. Uporządkuj powstałe pary według ich stopnia spójności leksykalnej.
6. W oparciu o dowolne powyższe pary utwórz wszystkie możliwe frazy, których człony będą w relacji equikohezji kwalifikacyjnej.
7. W oparciu o wyżej wypisane frazy i pary utwórz wszystkie możliwe frazy, których człony będą w relacji kohezji nierównego stopnia.
8. Znajdź we frazach orzeczenie lub orzeczenia.
9. Zbuduj poprawne zdania przez połączenia frazy z orzeczeniem lub orzeczeniami z innymi frazami w różnej kolejności.

4. Podsumowanie

Implementacja algorytmów binarnej syntezy i dychotomizacji zdań należy niewątpliwie do zagadnień z pogranicza językoznawstwa formalnego i informatycznego. Dzięki możliwości wyrażenia formalnych relacji syntaktycznych we współczesnych językach programowania otrzymujemy nie tylko urządzenie do analizowania struktury zdań o walorach naukowo-edukacyjnych, ale również w ograniczeniu do języka polskiego jesteśmy w stanie do pewnego stopnia zbadać jego produktywność składniową poprzez przeliczenie liczby zdań, które komputer jest w stanie samodzielnie zbudować z określonej ilości leksonów.

BIBLIOGRAFIA

- Bañcerowski Jerzy, Pogonowski Jerzy, Zgółka Tadeusz, 1982, *Wstęp do językoznawstwa*, Poznań, Wydawnictwo UAM, s. 237–250.
- Lapis Włodzimierz, 2004, *Lingwistyka matematyczna i teoria automatów*, Poznań, s. 6–7.