

Marek Krężolek\*

## KONTEKSTOWE ROZSTRZYGANIE WIELOZNACZNOŚCI W TEKŚCIE POLSKIM

*Tekst opisuje program zajmujący się problemem kontekstowego rozstrzygnięcia wieloznaczności w tekście polskim. Wieloznaczność to możliwość przypisania wyrazowi kilku różnych interpretacji, np. wyraz „zamek” może oznaczać budowlę, zamknięcie, zapieczęcie.*

*Opisany w tekście program korzysta ze słownika znaczeń, który dołączony do słownika fleksyjnego określa znaczenia dowolnego wyrazu tekstu. Przy rozstrzygnięciu wieloznaczności program stosuje dwie metody kontekstowego rozpoznawania znaczeń.*

- 1) Rozpoznanie znaczenia wyrazu wieloznacznego za pomocą wyrazów powiązanych z wyrazem wieloznacznym.*
- 2) Dostosowywanie znaczenia do tematu tekstu, w którym dany wyraz się znajduje.*

**Słowa kluczowe:** przetwarzanie języka naturalnego, rozpoznawanie znaczeń, wieloznaczność, dyzambiguacja

### RESOLVING AMBIGUITIES OF MEANING FROM CONTEXTUAL CLUES IN POLISH TEXTS

*This article describes a computer programme designed to assign the appropriate meaning to homonyms based on the context in which they appear. Homonyms can be defined as words with a number of different, often unrelated, meanings (for example a bank – a business that lends or keeps money, the land along the side of river, a large pile of earth, sand, snow).*

*The programme described in the paper exploits a monolingual dictionary, which, together with and supported by declension dictionary assigns the appropriate meaning for any given word from the text. In order to assign the correct and appropriate meaning for the given context two methods are used and exploited:*

- 1) Recognition of the meaning of a particular homonym based on the occurrence of its collocations in the immediate context.*
- 2) Applying the correct meaning of the word based on the overall theme or topic of the text in which appears.*

**Keywords:** natural language processing, recognition of the meaning, homonyms, disambiguation

## 1. Problem wieloznaczności

Wieloznaczność w języku naturalnym to – mówiąc najogólniej – możliwość przypisania kilku różnych interpretacji jakiemuś wyrażeniu.

\* Doktorant wydziału EAIiE AGH

W języku polskim mamy do czynienia z dwoma rodzajami wieloznaczności:

- 1) syntaktyczną,
- 2) semantyczną.

Z wieloznacznością syntaktyczną mamy do czynienia wtedy, gdy na podstawie analizy formy fleksyjnej występującej w tekście potrafimy jednoznacznie rozpoznać znaczenie wyrazu, lecz sam kształt formy fleksyjnej nie wskazuje na funkcję zdaniową rozpoznanego wyrazu. Przykładowo, gdy w tekście pojawi się napis *zegar*, jednoznacznie rozpoznajemy wyraz, ale nie jesteśmy w stanie powiedzieć, czy ten wyraz pełni funkcję podmiotu, np. *Zegar bije*, czy dopełnienia np. *Marek rozbił zegar*.

Wieloznaczność semantyczna polega na tym, że pewne formy fleksyjne pojawiające się w tekście mogą reprezentować różne wyrazy. Zatem wieloznaczna semantycznie jest zarówno forma *piec*, mogąca oznaczać rzeczownik lub czasownik, jak i forma *zamek* reprezentująca trzy różne rzeczowniki: „budowlę”, „zamknięcie” i „zapięcie”.

Praca niniejsza zajmuje się problemem rozstrzygnięcia wieloznaczności semantycznej.

## 2. Przetwarzanie tekstu a wieloznaczność

Program przetwarzania tekstu powinien działać tak jak człowiek, który musi przeczytać i zrozumieć tekst w nieznanym sobie języku obcym. Człowiek taki zaopatruje się w słownik dwujęzyczny, tj. słownik tłumaczący wyrażenia języka tekstu na wyrażenia języka znanego człowiekowi. Mając do dyspozycji taki słownik, człowiek czyta tekst napis po napisie, identyfikując każdy napis w słowniku. Poprawna identyfikacja wyrażenia wejściowego pozwala zrozumieć znaczenie tego wyrażenia poprzez odesłanie do znaczenia w języku znanym człowiekowi. Proces taki jest efektywny do momentu, w którym napotkamy wyrażenie wieloznaczne. Wówczas człowiek musi przeprowadzić rozumowanie pozwalające wybrać jedno z możliwych znaczeń wyrażenia wejściowego. Rozumowanie takie opiera się na zasadzie badania zgodności znaczeń wyrazu analizowanego ze znaczeniami pozostałych wyrazów tekstu, co jest możliwe dzięki posiadanej przez człowieka wiedzy o świecie.

Przedstawiony tu – w ogromnym uproszczeniu – proces przetwarzania tekstu przez człowieka jest zjawiskiem tak złożonym, że ciągle jeszcze stanowi przedmiot badań i eksperymentów. Niniejszy tekst przedstawi opis eksperymentu, którego celem było skonstruowanie algorytmów rozstrzygających pewne przypadki semantycznej wieloznaczności wyrazów polskich.

## 3. Wiedza językowa potrzebna algorytmom do rozstrzygnięcia wieloznaczności

### 3.1. Słownik

Ważną częścią każdego programu z zakresu przetwarzania języka naturalnego jest słownik fleksyjny, którego zadaniem jest weryfikacja faktu, czy dana forma należy do języka, oraz rozpoznanie tego, jaki wyraz dana forma reprezentuje. Zatem słownik fleksyjny to zbiór form oraz zbiór informacji na temat form.

Do podstawowych informacji należą:

- 1) informacja wskazująca na to, jaki wyraz (wyrazy) jest reprezentowany przez daną formę fleksyjną;
- 2) informacja o tym, którą z form danego wyrazu reprezentuje konkretna forma;
- 3) klasyfikacja semantyczna wyrazu.

Zatem dla np. występującej w tekście formy *kota* ze słownika powinniśmy uzyskać informację, że jest to dopełniacz lub biernik liczby pojedynczej wyrazu *kot* oraz że forma ta reprezentuje rzeczownik męski żywotny. W programie przedstawione wyżej informacje przechowujemy w tzw. słowniku fleksyjnym [ 2, 3].

Podczas wyszukiwania wyrazu dla formy może się okazać, że dana forma reprezentuje więcej niż jeden wyraz. Wtedy mamy do czynienia z wieloznacznością. Przykładem może być forma *piec*, która reprezentuje zarówno czasownik, jak i rzeczownik.

Jednak przy rozstrzygnięciu wieloznaczności potrzebna jest dodatkowa informacja o znaczeniach. Informację tę zawiera skonstruowany specjalnie dla potrzeb niniejszego programu słownik (opisu) znaczeń. W słowniku znaczeń opisane są znaczenia danego wyrazu, wraz z własnościami danego znaczenia oraz powiązania danego znaczenia z innymi wyrazami lub znaczeniami wyrazów. Budowa słownika znaczeń została przedstawiona w rozdziale 8.

### 3.2. Opis zależności syntaktycznych

Przy przetwarzaniu języka naturalnego, w szczególności przy rozpoznawaniu znaczeń, ważną rolę odgrywa gramatyka języka, gdyż dzięki niej możemy poznać zależności między wyrazami w zdaniu, co z kolei może pomóc przy rozpoznawaniu znaczeń, ponieważ w pewnym związku wyrazowym może wystąpić tylko jedno ze znaczeń. Na przykład, jeśli wieloznacznemu wyrazowi *zamek* przysługuje określenie *kamienny*, to możemy stwierdzić, że w tym przypadku chodzi o budowlę. Natomiast analizując związek syntaktyczny między wyrazami *zamek* i *kamienny*, możemy rozpoznać, że wyraz *kamienny* jest określeniem wyrazu *zamek*.

W języku polskim wyróżnia się kilka podstawowych części mowy: czasownik, rzeczownik, zaimek, przymiotnik, przysłówek i liczebnik. Każda z tych części mowy pełni w zdaniu określoną rolę (jako część zdania) i jest powiązana z innymi częściami zdania.

Rozpatrując zależności syntaktyczne występujące w zdaniu, możemy wyróżnić dwie grupy wyrazów: grupę podmiotu i grupę orzeczenia. W grupie podmiotu główną rolę pełni podmiot, którym jest rzeczownik (lub zaimek) w mianowniku, natomiast w grupie orzeczenia główną rolę pełni orzeczenie, którym jest czasownik w formie osobowej. Podmiot i orzeczenie, poza pewnymi szczególnymi przypadkami, mają taką samą liczbę i rodzaj. Oznacza to, że jeśli podmiot jest w liczbie pojedynczej i rodzaju męskim, to orzeczenie także musi być w liczbie pojedynczej i rodzaju męskim. Wyjątek stanowią sytuacje, gdy występuje kilka podmiotów: wtedy orzeczenie zawsze jest w liczbie mnogiej, natomiast o rodzaju decyduje to, czy występuje wśród nich podmiot w rodzaju męskoosobowym. Jeśli tak, to orzeczenie także jest w rodzaju męskoosobowym, w przeciwnym wypadku orzeczenie jest w rodzaju niemęskoosobowym. W opisanych poniżej algorytmach przyjęte zostało założenie, że orzeczenie jest wyrazem podrzędnym w stosunku do podmiotu.

W grupie podmiotu jako określenia podmiotu występują przydawki, którymi mogą być najczęściej przymiotniki, liczebniki lub rzeczowniki. Jeśli przydawką jest przymiotnik lub

liczebnik, to zgadzają się one z rzeczownikiem, który określają pod względem liczby, rodzaju i przypadku. Można powiedzieć, że przydawka jest wyrazem podrzędnym w stosunku do podmiotu. Jeśli przydawka jest rzeczownikiem, to może on tworzyć własną grupę, w której występują przydawki. W wyrażeniu *zamek bogatego człowieka* wyraz *człowieka* jest przydawką dla wyrazu *zamek*, a wyraz *bogatego* jest przydawką dla wyrazu *człowieka*.

W grupie orzeczenia jako określenia orzeczenia występują dopełnienia i okoliczniki. Dopełnienie jest rzeczownikiem w przypadku innym niż mianownik. Ponieważ dopełnienie jest rzeczownikiem, może także posiadać przydawki. Okolicznikami mogą być najczęściej przysłówki oraz rzeczowniki określające między innymi czas, miejsce, sposób wykonania danej czynności. W wyrażeniu *szybko zamknąłem piękny zamek*, wyraz *zamknąłem* jest orzeczeniem, *zamek* dopełnieniem, *piękny* przydawką wyrazu *zamek*, a *szybko* okolicznikiem sposobu. Okoliczniki i dopełnienia są wyrazami podrzędnymi w stosunku do orzeczenia.

Wykorzystując opisane zależności można dla każdego zdania w tekście zbudować powiązania pomiędzy wyrazami. Powiązania te znajdują zastosowanie przy rozpoznawaniu znaczeń wyrazów.

#### 4. Opis działania programu

Przedstawiony w pracy program działa według opisanego poniżej algorytmu. Początkowo program dokonuje tokenizacji, tj. dzieli tekst na jednostki składowe (wyrazy, zdania, akapity). Tokenizacja jest przeprowadzana w ten sposób, że ignoruje się pewne szczególne sytuacje, i np. w tekście przetwarzanym przez program każda kropka oznacza koniec zdania.

Kolejno następuje etap pobrania ze słownika informacji na temat każdego wyrazu występującego w tekście. W słowniku, z którego korzysta ten program, oprócz wyrazów prostych mogą występować wyrazy złożone, tu traktowane jako osobne znaczenia dla każdego z wyrazów składowych.

Następnym elementem programu są algorytmy wyszukiwania powiązań pomiędzy wyrazami<sup>1)</sup>, te powiązania są użyte przy rozpoznawaniu znaczeń wyrazów.

Jako ostatni element jest uruchamiany algorytm rozpoznawania znaczeń, który w pierwszej kolejności sprawdza, czy dany wyraz nie jest częścią wyrazu złożonego lub frazy<sup>2)</sup>, następnie stosowane są algorytmy rozpoznawania znaczeń z użyciem wyrazów nadrzędnych i podrzędnych<sup>3)</sup>, a na końcu stosowany jest algorytm „kontekstowego“ rozpoznawania znaczeń<sup>4)</sup>.

#### 5. Algorytmy wyszukiwania powiązań pomiędzy wyrazami

W zdaniu występują zależności pomiędzy wyrazami<sup>5)</sup>. Zależności te polegają na tym, że danym wyrazom można przyporządkować wyrazy nadrzędne i podrzędne. Dla wielu wyrazów można znaleźć zarówno wyrazy nadrzędne, jak i podrzędne, ale w związku z tym, że każdy wyraz ma co najwyżej jeden wyraz nadrzędny (poza pewnymi szczególnymi przy-

---

1) Porównaj rozdział 5.

2) Porównaj podrozdział 6.1.

3) Porównaj podrozdział 6.3.

4) Porównaj rozdział 7.

5) Porównaj podrozdział 3.2.

padkami), stosujemy tu algorytmy wyszukiwania wyrazów nadrzędnych, a po znalezieniu wyrazu nadrzędnego przyporządkowujemy mu wyraz podrzędny.

W artykule opisano nie jeden algorytm, ale zbiór algorytmów, gdyż dla każdej części mowy stosowany jest inny algorytm wyszukiwania wyrazu nadrzędnego.

Algorytmy zostały zbudowane dla zdań prostych w stronie czynnej, natomiast w zdaniach innego typu także wyszukiują zależności, ale wyniki wyszukiwania mogą być błędne.

Dzięki tym algorytmom możemy zbudować strukturę, która jest bardzo pomocna przy rozpoznawaniu znaczeń wyrazów.

## 5.1. Wyszukiwanie wyrazu nadrzędnego dla czasownika

Algorytm wyszukuje wyraz nadrzędny dla podanego czasownika. Rozróżniane są dwa rodzaje czasowników: bezokoliczniki i pozostałe. Dla bezokoliczników szukany jest wyraz nadrzędny, którym musi być czasownik, natomiast dla pozostałych szukamy rzeczownika lub zaimka w mianowniku zgadzających się z czasownikiem pod względem liczby i rodzaju (czyli dla orzeczenia jest wyszukiwany podmiot). Poszukiwanie rozpoczynamy od najbliższego wyrazu na lewo od danego wyrazu, przeglądane są kolejno wszystkie wyrazy i sprawdzane jest, czy któryś z nich spełnia warunki dla wyrazu nadrzędnego. Jeśli nie znajdziemy wyrazu nadrzędnego po lewej stronie, to przenosimy się na prawo od danego wyrazu.

## 5.2. Wyszukiwanie wyrazu nadrzędnego dla rzeczownika

Rozróżnianych jest kilka form rzeczownika, dla których szukamy wyrazów nadrzędnych (podział jest według przypadków, w których występują):

- rzeczownik w mianowniku to podmiot, więc nie ma on wyrazów nadrzędnych;
- rzeczowniki w dopełniaczu, narzędniku lub miejscowniku mogą mieć za wyraz nadrzędny zarówno czasownik, jak i rzeczownik;
- pozostałe rzeczowniki mogą mieć za wyraz nadrzędny tylko czasownik.

Podobnie jak dla czasowników, rozpoznawanie rozpoczyna się na lewo od danego wyrazu, a jeśli tam nie znajdziemy wyrazu nadrzędnego, to kontynuujemy poszukiwania na prawo od danego wyrazu, aż do napotkania wyrazu, który może być wyrazem nadrzędnym dla danego rzeczownika.

## 5.3. Wyszukiwanie wyrazu nadrzędnego dla przymiotnika i liczebnika

Szukanie wyrazu nadrzędnego dla przymiotnika rozpoczynamy od szukania wśród wyrazów znajdujących się po jego prawej stronie. Szukamy rzeczownika lub zaimka, który jest zgodny pod względem formy z danym przymiotnikiem (tzn. zgadza się pod względem przypadku, liczby i rodzaju). Przymiotnik może zostać przyporządkowany grupie rzeczowników i wtedy on może być w liczbie mnogiej, natomiast każdy z tych rzeczowników lub zaimków może być w liczbie pojedynczej lub mnogiej (np. w wyrażeniu *piękne: czapka i rękawiczki* przymiotnik *piękne* podlega grupie *czapka i rękawiczki*).

## 5.4. Wyszukiwanie wyrazu nadrzędnego dla przysłówka

Wyrazu nadrzędnego dla przysłówka także zaczynamy szukać na prawo od niego, a jeśli to nie da efektów, to analizujemy wyrazy znajdujące się po jego lewej stronie. Wyszukiwany jest dla niego jako wyraz nadrzędny czasownik, przymiotnik lub imiesłów przymiotnikowy.

## 5.5. Wyszukiwanie wyrazu nadrzędnego dla przyimka

Wyrazem nadrzędnym dla przyimka może być rzeczownik lub zaimek, przeszukiwanie rozpoczynane jest wśród wyrazów znajdujących się w zdaniu po jego prawej stronie.

## 6. Algorytm rozpoznawania znaczeń

Znaczenie wyrazu jest to treść kryjąca się za danym wyrazem. Wyraz w słowniku może mieć przyporządkowanych kilka znaczeń, które są określone razem z danym wyrazem w słowniku znaczeń. Zadanie programu polega na wybraniu znaczenia dla wyrazu występującego w zdaniu spośród wszystkich znaczeń danego wyrazu występujących w słowniku (wraz z wyrazami złożonymi).

Można powiedzieć, że ten algorytm dopasowuje kontekst wyrazu do kontekstu znaczenia występującego w słowniku. Kontekstem wyrazu nazywamy jego otoczenie w tekście, w którym występuje. Jego najbliższym otoczeniem są jego wyrazy nadrzędne i podrzędne. Dalszym kontekstem wyrazu są pozostałe wyrazy występujące w zdaniu i akapicie. Natomiast kontekstem znaczenia nazywamy zbiór cech określonych w słowniku znaczeń dla danego znaczenia wyrazu, które powinien posiadać wyraz w zdaniu odpowiadający danemu wyrazowi słownikowemu, do którego należy znaczenie, aby to znaczenie mogło zostać przypisane danemu wyrazowi w zdaniu. Dzięki porównaniu kontekstu wyrazu w zdaniu i kontekstu znaczenia zdefiniowanego w słowniku, możemy dane znaczenie przypisać wyrazowi przyjąć lub odrzucić.

Algorytm działa w ten sposób, że idąc od początku analizowanego tekstu, rozpoznaje znaczenie dla kolejnych wyrazów, wykorzystując do tego już rozpoznane znaczenia innych wyrazów. Jeśli dojdzie do końca tekstu, program wraca do początku i ponawia rozpoznawanie zadaną ilość razy. Dzieje się tak dlatego, że algorytm w pierwszym przejściu pętli ma już częściowo zidentyfikowane znaczenia wyrazów poprzedzających badany wyraz, natomiast nie ma rozpoznanych znaczeń wyrazów występujących za nim, co poważnie ogranicza zdolność tego algorytmu, natomiast w kolejnych przejściach pętli ma już więcej znaczeń rozpoznanych.

Algorytm rozpoznając znaczenie wyrazu, korzysta z kilku metod opisanych poniżej. W każdej z tych metod wykorzystywane są pewne elementy opisane w słowniku znaczeń, a czasami także we fleksyjnym. W algorytmie jest założenie, że każdy wyraz w zdaniu ma jakieś znaczenie, więc jeśli liczba znaczeń dla danego wyrazu spadnie podczas przeszukiwania do zera, to wyrazowi są przypisywane od nowa wszystkie znaczenia i algorytm ponownie próbuje rozpoznawania znaczeń.

Istnieją dwie główne metody rozpoznawania znaczeń:

- 1) niezależna,
- 2) zależna.

Metoda niezależna polega na tym, że dla każdego spośród potencjalnych znaczeń danego wyrazu analiza kontekstu odbywa się osobno (niezależnie od pozostałych znaczeń) i niezależnie podejmowana jest decyzja, czy w danej sytuacji to znaczenie jest możliwe, czy nie. Metoda zależna polega na tym, że spośród dopuszczalnych znaczeń wybierane jest najbardziej prawdopodobne w tej sytuacji<sup>6)</sup>. Z tą metodą mamy do czynienia przy dostoso-

---

<sup>6)</sup> Porównaj rozdział 7.

wywaniu znaczenia do tematu akapitu, natomiast z metodą niezależną mamy do czynienia przy rozpoznawaniu znaczenia z użyciem form i odmiany, lub z użyciem wyrazów nadrzędnych i podrzędnych. Wyjątkiem są sytuacje identyfikowania danego znaczenia. Występuje to wtedy, gdy dla jednego ze znaczeń jest prawdziwa flaga identyfikująca, co powoduje odrzucenie wszystkich znaczeń, dla których żadna flaga identyfikująca nie jest prawdziwa. W pierwszej kolejności sprawdzane jest (wszystkimi metodami), czy dane znaczenie nie jest odrzucane w danym kontekście, oraz czy jest możliwe, a dopiero później jest sprawdzana identyfikacja. Szczegóły dotyczące ustawiania poszczególnych wartości i flag znajdują się w rozdziale 8.

## 6.1. Wyrazy złożone i algorytmy ich wyszukiwania

Jeśli jakieś znaczenie jest związane z jakimś wyrazem złożonym (lub frazą), to w pierwszej kolejności są analizowane flagi dla wyrazów złożonych. Uruchamiany jest wtedy algorytm sprawdzania, czy w danym zdaniu jest związek wyrazowy odpowiadający danemu znaczeniu; jeśli związek ten zostanie zidentyfikowany, to odrzucane są wszystkie znaczenia nie związane z wyrazami złożonymi.

Związek wyrazowy (lub wyraz złożony czy złożenie wyrazów) jest specyficznym rodzajem wyrazu w słowniku, jest to ciąg wyrazów prostych, na które nałożone są pewne dodatkowe warunki. Często znaczenie wyrazu złożonego jest różne od znaczeń poszczególnych wyrazów wchodzących w jego skład.

Związek wyrazowy może być definiowany w słowniku znaczeń, nie definiuje się dla niego odmiany, gdyż każdy z jego składników ma zdefiniowaną odmianę, ponieważ samodzielnie także stanowi wyraz w słowniku. Wyraz taki definiujemy poprzez wprowadzenie pewnych ograniczeń dla jego składników, natomiast w zdaniu jest identyfikowany za pomocą określonych własności. Jest on traktowany jako osobne znaczenie dla każdego z jego wyrazów składowych.

Frazą natomiast nazywamy specyficzny rodzaj złożenia wyrazów, w którym może wystąpić wyraz zastępczy, powodujący, że dana fraza może występować w tekście w wielu postaciach.

Wyraz zastępczy jest to wyraz specjalnego typu zdefiniowany w słowniku znaczeń na potrzeby fraz, jest on używany do ich definiowania. Wyrazy zastępcze nie pojawiają się w tekstach, a użycie takiego wyrazu w definicji frazy określa, że jeśli dana fraza pojawi się w tekście, to zamiast wyrazu zastępczego pojawi się w niej dowolny wyraz spełniający warunki określone przy definicji frazy<sup>7)</sup>.

Wyraz zastępczy można poznać po tym, że jest ujęty w nawiasy  $\langle \rangle$ . Wyrazy zastępcze nie są widoczne poza słownikiem znaczeń. Dzięki wyrazom zastępczym możemy definiować pewne własności jakiegoś wyrazu we frazie bez precyzowania dokładnie, o jaki wyraz nam chodzi (np. możemy określić, że zamiast tego wyrazu może wystąpić rzeczownik w dopełniaczu). Przykładem może być wyrażenie *Zbić  $\langle$ kogoś $\rangle$  z *pantałyku* zdefiniowane w słowniku, natomiast w analizowanym tekście pojawi się odpowiadające mu wyrażenie *Zbiłem Jasia z *pantałyku**.*

---

<sup>7)</sup> Porównaj podrozdział 8.2.4.1.

Algorytm wyszukiwania związków wyrazowych jest uruchamiany, zanim zostaną uruchomione inne algorytmy rozpoznawania znaczeń. Algorytm dla każdego słowa występującego w związku wyszukuje jego odpowiednik w zdaniu (lub kilka odpowiedników, jeśli jest kilka takich samych wyrazów w zdaniu). Jeśli choć jeden wyraz nie ma odpowiednika w zdaniu, to algorytm kończy się z wynikiem negatywnym. Następnie na podstawie ułożenia wyrazów w związku i zdaniu oraz tego, które formy danego wyrazu są dopuszczalne w tym związku, wybierany jest dla każdego wyrazu ze związku jeden odpowiednik w zdaniu. Otrzymujemy w zdaniu związek, który jest następnie poddawany analizie, czy pasuje do warunków określonych dla danego złożenia wyrazów w słowniku znaczeń. Warunki te określają dopuszczalne formy dla każdego wyrazu, czy jest wymagana kolejność wyrazów, czy nie, oraz czy wyrazy z danego związku muszą się znajdować obok siebie. Dany związek może zostać odrzucony lub zidentyfikowany. Przykładowo, jeśli w słowniku mamy zdefiniowany związek wyrazowy<sup>8)</sup> *panna młoda* i mamy zdefiniowane w słowniku znaczeń, że ten związek jest identyfikowany, gdy wyrazy go tworzące występują obok siebie w określonej kolejności, a w każdym innym przypadku jest odrzucany, to jeśli w zdaniu pojawia się *...panna młoda...*, to ten wyraz złożony jest identyfikowany, natomiast gdyby w zdaniu wystąpił ciąg *...młoda panna...*, to byłby on odrzucony.

## 6.2. Rozpoznawanie za pomocą form wyrazu i odmiany

Forma wyrazu jest to postać, pod jaką dany wyraz może wystąpić w zdaniu. Natomiast odmiana to określenie formy wyrazu nie bezpośrednio, ale przez formę fleksyjną, jaką ona reprezentuje (np. zamiast powiedzieć, że wymagana jest forma *pantafyku*, możemy powiedzieć, że wymagany jest dopełniacz liczby pojedynczej). Odmianą musimy się posługiwać, gdy nie wiemy, jaki to będzie wyraz (przy wyrazach zastępczych<sup>9)</sup>.

W słowniku może zostać określone, które formy są możliwe, niedozwolone lub identyfikujące dla danego znaczenia. Jeśli w zdaniu akurat występuje forma niedozwolona (odrzucająca), to dane znaczenie jest odrzucane, jeśli natomiast trafi na formę identyfikującą, to identyfikuje to znaczenie. Jeśli dla danego znaczenia są określone formy możliwe, a w zdaniu występuje forma nie należąca do tej grupy, to dane znaczenie jest odrzucane.

Podobne zależności można określić dla odmiany wyrazu.

Ta metoda nie jest związana z rozpoznawaniem kontekstowym, ale jest prostsza do stosowania, pewniejsza i dlatego jest wykonywana w pierwszej kolejności.

## 6.3. Rozpoznawanie z użyciem powiązań między wyrazami

Po wykonaniu algorytmu poszukiwania wyrazów nadrzędnych<sup>10)</sup>, dla wyrazów z analizowanego tekstu są przydzielone wyrazy nadrzędne i podrzędne. W słowniku dla danego znaczenia mogą być określone wyrazy nadrzędne i podrzędne odrzucające, możliwe i identyfikujące. Porównując je z wyrazami nadrzędnymi i podrzędnymi dla danego wyrazu w zdaniu, możemy odrzucić, zidentyfikować lub pozostawić dane znaczenie. Przykładowo w zdaniu: *Mam zielony beret* zidentyfikowane zostanie dla wyrazu *zielony* to znaczenie, dla którego jest określony wyraz nadrzędny identyfikujący jako *beret*, a odrzucone to, dla którego jest określony wyraz nadrzędny odrzucający jako *beret*.

---

<sup>8)</sup> Porównaj podrozdział 8.2.4.1.

<sup>9)</sup> Porównaj podrozdział 6.1.

<sup>10)</sup> Porównaj rozdział 5.



## 6.4. Rozpoznawanie z użyciem własności wyrazów

Możemy także określić w słowniku dla danego znaczenia własności odrzucające, identyfikujące i wymagane wyrazów nadrzędnych i podrzędnych. Porównując te własności z własnościami, jakie w danym zdaniu posiadają wyrazy nadrzędne i podrzędne, możemy odrzucić, zidentyfikować lub pozostawić dane znaczenie.

Własności są to cechy określone w słowniku fleksyjnym<sup>11)</sup>, jakie posiada dane znaczenie, cechy te są przypisywane wyrazowi w zdaniu, jeśli dla danego wyrazu zostanie zidentyfikowane dane znaczenie. Każda własność składa się z nazwy własności i wartości (może ich być dowolna ilość). Własności mogą być dziedziczone<sup>12)</sup>, tzn. możemy określić własności dla wyrazu *człowiek*, a następnie definiując np. wyraz *żołnierz*, możemy wskazać, że dziedziczy on wszystkie własności od wyrazu *człowiek* (oraz możemy dodożyć własności specyficzne dla wyrazu *żołnierz*) bez specyfikowania, o jakie własności nam chodzi. Struktura własności przypomina strukturę sieci semantycznych.

Jeśli chcemy, aby dane znaczenie zostało zidentyfikowane, gdy wyrazem nadrzędnym jest wyraz określający *człowieka*, to wystarczy podać, że własnością identyfikującą wyrazu nadrzędnego jest *jest = człowiekiem*, i wtedy niezależnie od tego, które z określeń *człowieka* zostanie użyte (np. *żołnierz*), znaczenie zostanie zidentyfikowane właściwie.

Ponieważ własności są związane ze znaczeniem, to aby ich użyć, dany wyraz, posiadający te własności musi mieć zidentyfikowane znaczenie. Czyli, aby algorytm rozpoznający znaczenia mógł dla danego wyrazu użyć cech odnoszących się do własności wyrazu nadrzędnego, to wyraz nadrzędny musi mieć rozpoznane znaczenie. Natomiast przy metodzie odwołującej się bezpośrednio do wyrazów nie ma znaczenia to, czy dla wyrazu nadrzędnego zostało zidentyfikowane znaczenie. Przewagą metody używającej własności jest to, że nie odwołujemy się do konkretnego wyrazu, tylko do pewnego zbioru wyrazów, który może zostać powiększony lub pomniejszony (bez ingerencji w definicję badanego wyrazu) poprzez przypisanie lub zabranie tej własności jakiemuś wyrazowi.

## 6.5. Rozpoznawanie za pomocą innych wyrazów występujących w zdaniu i akapicie

Ta metoda polega na określeniu wyrazów identyfikujących i odrzucających, które mogą zidentyfikować lub odrzucić dane znaczenie w ramach zdania lub akapitu. Metoda ta bazuje na tym, że jeśli w danym tekście pojawi się wyraz związany z danym znaczeniem, to w tekście pojawią się wyrazy związane tematycznie z danym wyrazem, lepszą wersją takiego rozpoznawania jest metoda kontekstowego rozpoznawania znaczeń<sup>13)</sup>, gdyż tutaj o zidentyfikowaniu lub odrzuceniu decyduje wystąpienie pojedynczego wyrazu, natomiast w tamtej metodzie temat całego akapitu.

---

<sup>11)</sup> Porównaj podrozdział 3.1.

<sup>12)</sup> Porównaj podrozdział 8.2.6.

<sup>13)</sup> Porównaj rozdział 6.

## 7. Rozpoznawanie kontekstowe znaczenia wyrazu

Zaimplementowany w program jest także algorytm wyboru najlepszego znaczenia na podstawie akapitu. Wykorzystywane jest tu spostrzeżenie, że wyrazy występujące w jednym akapicie najczęściej są ze sobą powiązane tematycznie.

W słowniku dla każdego znaczenia powinny być określone grupy tematyczne, do jakich należy dane znaczenie. Powinno to być zrobione dla wszystkich wyrazów, nie tylko dla wyrazów wieloznacznych. Algorytm ten oblicza dla każdego znaczenia badanego wyrazu współczynnik pewności i współczynnik prawdopodobieństwa.

Algorytm działa w ten sposób, że ze wszystkich wyrazów występujących w danym akapicie pobiera grupy tematyczne i dla każdej grupy tematycznej oblicza ilość jej wystąpień. Do obliczenia współczynnika pewności brane są pod uwagę tylko te wyrazy w akapicie, dla których już jednoznacznie zidentyfikowano znaczenie, natomiast do obliczenia współczynnika prawdopodobieństwa brane są wszystkie wyrazy ze wszystkimi znaczeniami, jakie jeszcze pozostały, z tym że punkty za wystąpienie grupy dzieli się przez ilość znaczeń danego wyrazu. Jeśli jakieś znaczenie badanego wyrazu ma zdecydowanie więcej punktów niż pozostałe, to ono jest identyfikowane. Jeśli nie ma takiego znaczenia, ale najlepsze znaczenie zidentyfikowane za pomocą współczynnika pewności i współczynnika prawdopodobieństwa jest takie samo, to sprawdzane są inne wystąpienia tego wyrazu w akapicie. Jeżeli jest jeszcze jakieś wystąpienie tego wyrazu i wszystkie wystąpienia posiadają to znaczenie, które uzyskało najwięcej punktów, to jest ono identyfikowane.

## 8. Opis struktury słownika znaczeń

W słowniku znaczeń zapisujemy tylko te wyrazy, których nie ma w słowniku fleksyjnym, lub te, które mogą mieć kilka znaczeń. Musimy zdefiniować tutaj także wszystkie wyrazy złożone i frazy. Słownik znaczeń (i rozpoznawanie znaczeń) możemy zdefiniować na wiele sposobów, które daje nam ten program. Sposób definicji słownika znaczeń określa metodę rozpoznawania znaczeń zastosowaną w tekście. To, którą metodę należy wybrać, zależy od użytkownika i specyfiki tekstów, które chce on interpretować.

Struktura słownika jest ściśle powiązana z algorytmami użytymi do rozpoznawania znaczeń.

### 8.1. Podstawowe pojęcia związane ze słownikiem

Słownik ma budowę blokową, tzn. składa się z odrębnych bloków, w skład których wchodzi inne bloki oraz zmienne. Bloki posiadają nagłówek bloku i zakończenie. Nagłówek bloku to nazwa bloku ujęta w nawiasy kwadratowe []. Natomiast koniec bloku to nazwa bloku ujęta w nawiasy kwadratowe i poprzedzona słowem *KONIEC*.

Zmienne podaje się w ten sposób, że podaje się jej nazwę (wielkość liter nie jest rozróżniana), a następnie znak równa się i wartości danej zmiennej oddzielone średnikami. Wiele wartości danej zmiennej można podać również, podając nazwę zmiennej dla każdej wartości osobno w osobnej linii. Niektóre zmienne wymagają, aby dla nich była dokładnie jedna wartość (np. zmienna *CzęśćMowy*) i wtedy możemy podać tylko jedną wartość dla zmiennej.

W blokach mogą występować także *operacje*. Różnią się one tym od zmiennej, że składają się z dwóch części: atrybutu i jego wartości. W blokach wypisuje się je w ten sposób, że podaje się nazwę operacji, a następnie w nawiasach (), po przecinku najpierw

atrybut, a później jego wartość. Jeśli w nawiasie będą więcej niż dwa argumenty, to pozostałe argumenty będą traktowane jako inne wartości tego samego atrybutu. *Operacje* wykonywane są na przykład przy *własnościach*.

## 8.2. Charakterystyka bloków słownika

### 8.2.1. Blok WYRAZ

Najbardziej zewnętrznym blokiem jest blok *WYRAZ*. W bloku tym występują zmienne: *Wyraz* oraz *CzęśćMowy*. Zmienna *Wyraz* określa formy podstawowe wyrazu, który definiujemy w danym bloku, zmiennej tej może zostać przypisanych kilka wartości (wtedy pierwsza jest wartością identyfikującą dany wyraz), jeśli kilka wyrazów z słownika fleksyjnego chcemy połączyć tu w jeden wyraz (przykładowo wyraz *on* ma podane wartości w polu *on*, *ona*, *ono*, *oni*, *one*, gdyż w słowniku fleksyjnym są to osobne wyrazy). Natomiast *CzęśćMowy* określa, jaką część mowy reprezentuje dany wyraz. Każdy wyraz lub złożenie wyrazów zdefiniowany w słowniku musi znajdować się w osobnym bloku *WYRAZ*.

Przykład bloku *WYRAZ*:

[WYRAZ]

Wyraz = zobaczyć

CzęśćMowy = czasownik

[ODMIANA]

...

[KONIEC ODMIANA]

[ZNACZENIE]

...

[KONIEC ZNACZENIE]

[ZNACZENIE]

...

[KONIEC ZNACZENIE]

...

[KONIEC WYRAZ]

### 8.1.2. Blok ODMIANA

Blok ten wchodzi w skład bloku *WYRAZ* i definiuje odmianę dla każdego wyrazu. Dla każdej części mowy ten blok wygląda inaczej. Jeśli korzystamy z fleksyjnego słownika na serwerze, ten blok może być pominięty lub może służyć jako uzupełnienie tamtego słownika, dlatego też dokładniejszy opis tego bloku zostanie pominięty.

### 8.1.3. Blok ZNACZENIE

Blok *ZNACZENIE* opisuje pojedyncze znaczenie danego wyrazu. Znajduje się wewnątrz bloku *WYRAZ* i może wystąpić dowolna liczba tych bloków, bo każdy wyraz w słowniku może mieć jedno lub więcej znaczeń. Jeśli znaczenie nie zostanie podane dla danego wyrazu, to przy wczytywaniu słownika przez program jest tworzone dla tego wyrazu znaczenie domyślne. Działanie programu polega na wybraniu spośród wszystkich znaczeń pasujących do danej formy zdaniowej jednego najbardziej pasującego w danym kontekście. Blok *ZNACZENIE* składa się z trzech bloków opisujących dane znaczenie: *KONTEKST*, *INFO*, *WŁASNOŚCI* oraz posiada jedną zmienną: *NazwaZnaczenia*, określająca nazwę danego znaczenia.

Przykład bloku *ZNACZENIE*:

[ZNACZENIE]

NazwaZnaczenia = budowla

[KONTEKST]

...

[KONIEC KONTEKST]

[INFO]

...

[KONIEC INFO]

[WŁASNOŚCI]

...

[KONIEC WŁASNOŚCI]

[KONIEC ZNACZENIE]

### 8.1.4. Blok KONTEKST

Blok *KONTEKST* wchodzi w skład bloku *ZNACZENIE* i może istnieć tylko jeden taki blok w jednym bloku *ZNACZENIE*. Blok *KONTEKST* zawiera zbiór własności danego znaczenia, które pozwalają zaakceptować lub odrzucić dane znaczenie w pewnym kontekście. W bloku tym przede wszystkim są określone parametry do rozpoznawania danego znaczenia z użyciem wyrazów nadrzędnych i podrzędnych.

Cechy charakteryzujące dane znaczenie mogą odnosić się do:

- 1) cech samej formy zdaniowej, dla której identyfikujemy dane znaczenie; własności te mogą dotyczyć tego, jaka forma danego wyrazu jest dopuszczalna dla danego znaczenia;
- 2) cech wyrazów nadrzędnych i podrzędnych, czyli tego, jakie to są wyrazy, jakie posiadają własności;
- 3) wszystkich wyrazów w zdaniu, w którym jest badany wyraz;
- 4) wszystkich wyrazów, jakie występują w akapicie, w którym występuje badany wyraz;
- 5) ustawienia, że dane znaczenie jest znaczeniem domyślnym.

W pierwszej grupie mamy cechy dotyczące formy i odmiany. Forma określa dokładną postać danego wyrazu w zdaniu (np. *zamka* jest formą wyrazu *zamek*), natomiast odmiana określa, co reprezentuje ta forma (np. mianownik liczby pojedynczej). Jeśli chcemy określić, że dana forma wyrazu zidentyfikuje nam to znaczenie, to przypisujemy ją zmiennej *FormaIdentyfikująca*; analogicznie – jeśli dana forma nigdy nie będzie występowała dla danego znaczenia, to możemy ją przypisać zmiennej *FormaOdrzucająca*, natomiast jeśli formy danego wyrazu dla danego znaczenia są ograniczone do pewnego zbioru (i spoza niego formy na pewno nie będą związane z danym znaczeniem), to możemy je przypisać zmiennej *FormaMożliwa*.

Analogiczne zmienne, zachowujące się w podobny sposób, mamy dla:

- *OdmianyIdentyfikującej*,
- *OdmianyOdrzucającej*,
- *OdmianyMożliwej*.

W drugiej grupie mamy zmienne dotyczące wyrazów nadrzędnych i podrzędnych. Jeśli chcemy określić, jaki wyraz nadrzędny spowoduje zidentyfikowanie danego znaczenia, to przypisujemy go zmiennej

*WyrazNadrzędnyIdentyfikujący.*

Podobnie, jeśli jakiś wyraz nadrzędny nie pasuje do danego znaczenia wyrazu, to przypisujemy go zmiennej

*WyrazNadrzędnyOdrzucający,*

a jeśli dane znaczenie wyrazu może występować tylko razem z wyrazami, które możemy wyliczyć, to możemy je wszystkie przypisać zmiennej

*WyrazNadrzędnyMożliwy.*

Podobnie możemy uczynić ze zmiennymi dotyczącymi wyrazów podrzędnych, takimi jak:

- *WyrazPodrzędnyIdentyfikujący,*
- *WyrazPodrzędnyOdrzucający,*
- *WyrazPodrzędnyMożliwy.*

W drugiej grupie znajdują się także operacje dotyczące własności wyrazów. Jeśli chcemy określić zależność jakiegoś znaczenia nie od konkretnego wyrazu, ale od wyrazu posiadającego jakąś własność, to możemy użyć jednej z tych operacji. Operacje w odróżnieniu od zmiennych składają się z dwóch części: atrybutu i jego wartości. Własności są podawane w bloku *WŁASNOŚCI*.

Operacje:

- *WłasnośćWyrazuNadrzędnegoIdentyfikująca,*
- *WłasnośćWyrazuPodrzędnegoIdentyfikująca,*
- *WłasnośćWyrazuNadrzędnegoOdrzucająca,*
- *WłasnośćWyrazuPodrzędnegoOdrzucająca,*

mogą być używane analogicznie jak zmienne opisane powyżej.

Natomiast operacje:

- *WłasnośćWyrazuNadrzędnegoWymagana,*
- *WłasnośćWyrazuPodrzędnegoWymagana,*

określają, jakie własności (atrybut plus wartość) muszą posiadać wyrazy nadrzędne lub podrzędne, aby dane znaczenie było możliwe.

Trzecia grupa zawiera zmienne dotyczące identyfikacji na poziomie zdania za pomocą innych wyrazów. Jeśli chcemy, aby wystąpienie jakiegoś wyrazu w zdaniu identyfikowało jakieś znaczenie, to musimy je przypisać zmiennej

*WyrazIdentyfikujący,*

analogicznie jeśli chcemy odrzucić znaczenie, jeśli w danym zdaniu wystąpi jakiś wyraz, to musimy go przypisać zmiennej

*WyrazOdrzucający.*

Podobnie zachowują się zmienne z czwartej grupy:

- WyrazIdentyfikującyWakapicie,
- WyrazOdrzucającyWakapicie,

tylko że dotyczą identyfikacji za pomocą wyrazów znajdujących się w akapicie.

Jeśli chcemy, żeby dane znaczenie było znaczeniem domyślnym (tzn. aby było identyfikowane, jeśli po zastosowaniu innych metod identyfikacji pozostało kilka znaczeń dla danego wyrazu – grupa piąta), to musimy zmienną *DomyślneZnaczenie* ustawić na TAK. Można to wykorzystać, jeżeli dany wyraz ma kilka znaczeń, ale jedno z nich występuje w znaczącej liczbie przypadków.

Przykład bloku *KONTEKST*:

[KONTEKST]

WyrazNadrzędnyIdentyfikujący = zwiedzać

WyrazNadrzędnyOdrzucający = podnieść

WyrazPodrzędnyIdentyfikujący = średniowieczny

WyrazIdentyfikującyWakapicie = zwiedzać

WyrazIdentyfikujący = zwiedzać

WłasnośćWyrazuNadrzędnegoIdentyfikująca = (wykonują, zwiedzanie)

[KONIEC KONTEKST]

#### 8.2.4.1. Blok *KONTEKST* dla wyrazów złożonych i fraz

W bloku *KONTEKST* dla wyrazów złożonych i fraz mogą występować wszystkie elementy, jakie są w bloku *KONTEKST* dla zwykłych wyrazów. Dodatkowo mogą wystąpić tu zmienne dotyczące kolejności i sposobu występowania poszczególnych składników oraz opisy wyrazów wchodzących w skład wyrazów złożonych.

Jeśli chcemy wymusić, aby dany wyraz złożony był identyfikowany zawsze, gdy wyrazy są ułożone kolejno, należy ustawić flagę

*IdentyfikacjaZawszeGdyKolejno*

na TAK. Jeśli ta flaga nie jest ustawiona, to brane są pod uwagę inne własności wyrazu w celu jego identyfikacji.

Natomiast jeśli chcemy, aby dany wyraz był identyfikowany zawsze, gdy wyrazy składające się na niego złożone są obok siebie, to należy ustawić flagę

*IdentyfikacjaZawszeGdyObokSiebie*

na TAK. Jeśli uważamy, że dany wyraz złożony ma sens, tylko wówczas, gdy wyrazy są ułożone kolejno (np. przy nazwach takich, jak *Polskie Koleje Państwowe*), to ustawiamy flagę

*MusząZachowaćKolejność*

na TAK. Natomiast jeśli uważamy, że poszczególne wyrazy muszą występować obok siebie w zdaniu (wykorzystywane np. przy nazwach), to ustawiamy flagę

*MusząByćObokSiebie*.

Istnieje jeszcze flaga

*IdentyfikacjaZawsze*,

która oznacza, że identyfikujemy dane złożenie wyrazów zawsze, gdy tylko można je zidentyfikować (tzn. istnieją wszystkie składowe i nie zachodzi sprzeczność z żadną z ustawionych flag).

Poniżej przedstawiono przykład bloku *KONTEKST* dla dla frazy: *obiecować <komuś> gruszki na wierzbie*.

[KONTEKST]

IdentyfikacjaZawszeGdyKolejno = TAK  
IdentyfikacjaZawszeGdyObokSiebie = TAK  
MusząByćObokSiebie = NIE  
MusząZachowaćKolejność = NIE

[KONTEKST]

Wyraz = gruszka  
FormaMożliwa = gruszki, gruszek

[KONIEC KONTEKST]

[KONTEKST]

Wyraz = wierzba  
FormaMożliwa = wierzbie

[KONIEC KONTEKST]

[KONTEKST]

Wyraz = <ktoś>  
OdmianaMożliwa = Celownik  
MożeByćNieobecny = TAK

[KONIEC KONTEKST]

[KONIEC KONTEKST]

#### 8.2.4.2. Blok *KONTEKST* wyrazu składowego w bloku *KONTEKST* dla wyrazów złożonych i fraz

W bloku tym mamy zmienną *Wyraz*, która określa, który z wyrazów składowych jest tu opisywany. A poza tym możemy używać zmiennych zawartych w bloku *KONTEKST* dla normalnego znaczenia. Najczęściej używana jest zmienna *FormaMożliwa* do określenia, która z form wyrazu może występować w danym związku. Przykład bloków dla wyrazu składowego można zaobserwować w przykładzie podanym w podrozdziale 8.2.4.1.

#### 8.2.5. Blok *INFO*

Blok ten znajduje się w bloku *ZNACZENIE* i zawiera informacje potrzebne do identyfikacji danego znaczenia, jak również informacje, które mogą zostać użyte do identyfikacji innych wyrazów za pomocą danego. Blok ten powinien być definiowany, jeśli chcemy użyć kontekstowego rozpoznawania znaczeń<sup>14)</sup>. W bloku tym może wystąpić zmienna *GrupyTematyczne*, zawierająca informacje, do jakich grup tematycznych należy dane znaczenie. Na podstawie grup tematycznych wszystkich wyrazów w akapicie można określić, do jakiego tematu zalicza się cały akapit (może to być kilka tematów), a następnie na tej podstawie jest dla danego wyrazu identyfikowane to znaczenie, którego grupy tematyczne najlepiej pasują do grup tematycznych danego akapitu. Jeśli jakaś grupa tematyczna jest szczególnie ważna dla identyfikowanego znaczenia, to można ją wypisać dwa razy, wtedy przy obliczaniu grupy tematycznej akapitu jest punktowana podwójnie.

<sup>14)</sup> Porównaj rozdział 7.

Przykład bloku *INFO*:

[INFO]

GrupyTematyczne = budowle

[KONIEC INFO]

## 8.2.6. Blok **WŁASNOŚCI**

Blok ten znajduje się w bloku *ZNACZENIE* i zawiera własności, jakie posiada dane znaczenie. Własność to atrybut razem z wartością. W bloku tym atrybuty traktowane są jak zmienne, ich wartości jak wartości tych zmiennych. Przykładowo, aby ustawić danemu znaczeniu atrybut *kolor* na wartość *zielony*, piszemy

*kolor = zielony.*

Dany atrybut może być definiowany tylko dla niektórych znaczeń, dla innych znaczeń może być pominięty. Możemy zadeklarować dowolne własności o dowolnych nazwach. Oprócz tego w bloku tym dostępnych jest kilka operacji wykonujących pewne operacje na własnościach.

Własności znaczenia powinniśmy definiować, jeśli zdecydowaliśmy się używać metody pośredniej<sup>15)</sup> rozpoznawania znaczeń za pomocą wyrazów powiązanych z danym w zdaniu, z tym że definiujemy go nie tylko dla wyrazów niejednoznacznych (czyli tych, które będziemy rozpoznawać), ale dla wszystkich wyrazów, które potencjalnie mogą być użyte do rozpoznawania znaczeń.

Myślę, że ta metoda może przynieść dobre efekty, jeśli nadawanie własności wyrazom będzie się odbywało w sposób przemyślany i własności zostaną zdefiniowane dla większości istotnych wyrazów w słowniku znaczeń.

Pierwszą z operacji występujących w tym bloku jest operacja *PobierzWłasnościOd*, która powoduje pobranie wszystkich własności od znaczenia określonego przez argumenty tej operacji (pierwszy argument to wyraz, od którego mamy pobrać własności, a drugi to nazwa znaczenia dla danego wyrazu, od którego mamy pobrać własności; jeśli nie podamy drugiego argumentu, to operacja pobierze własności od wszystkich znaczeń podanych dla wyrazu o identyfikatorze określonym przez pierwszy argument).

Drugą operacją jest *Usuń*, która powoduje usunięcie pewnych wartości dla danej własności. Pierwszym argumentem jest nazwa własności, z której usuwamy wartości, a następnie podane są wartości, które są usuwane. Jeśli jest podana tylko nazwa własności, to usuwana jest cała własność wraz ze wszystkimi wartościami.

Trzecią operacją jest operacja *Dodaj*, która zachowuje się analogicznie jak zwykle przypisanie wartości do nazwy własności.

Kolejną operacją jest operacja *Ustaw*, która powoduje dla danej własności wykasowanie wszystkich jej dotychczasowych wartości i ustawienie nowych podanych jak kolejne argumenty tej operacji.

Ostatnią operacją jest operacja *Zmień*, która powoduje zmianę jednej wartości dla danej własności na inną. Jako pierwszy argument podaje się nazwę własności, na której chcemy wykonać operację, następnie podajemy, jaką wartość chcemy zamienić, ostatnim argumentem jest nowa wartość danej własności.

W bloku *WŁASNOŚCI*, w odróżnieniu od innych bloków, ważna jest kolejność zapisania operacji. Operacje na własnościach są wykonywane w takiej kolejności w jakiej zostały zapisane.

---

<sup>15)</sup> Porównaj podrozdział 6.4



Operacje takie, jak *Usuń*, *Zmień*, *Ustaw* są przydatne, jeśli chcemy dziedziczyć własności z jakiegoś znaczenia, ale nie wszystkie (wtedy niektóre z nich możemy usunąć).

Przykład bloku *WŁASNOŚCI* (np. własności dla znaczenia *żołnierz* wyrazu *admiral*):

[WŁASNOŚCI]

PobierzWłasnościOd = żołnierz

Usuń(musi, słuchać)

Jest = dowódcą

[KONIEC WŁASNOŚCI]

## 9. Podsumowanie

### 9.1. Zastosowania programu

Program ten może zostać wykorzystany w systemach przetwarzania języka naturalnego w różnych celach. Jednym z zastosowań tego programu po niewielkiej modyfikacji jest automatyczne wstawianie w tekście linków do stron opisujących jakieś obiekty, rzeczy. Przykładowo, mamy tekst HTML będący częścią przewodnika po Krakowie, w którym pojawia się wyraz *Wisła*. Dzięki temu programowi i odpowiednio skonstruowanemu słownikowi znaczeń możemy określić, czy chodzi nam o rzekę *Wisłę*, czy o klub sportowy *Wisła*, czy też może o jakąś restaurację o nazwie *Wisła*. Po zidentyfikowaniu odpowiedniego znaczenia do danego tekstu wkładamy link do strony opisującej obiekt, który został zidentyfikowany (dokładniej zostało zidentyfikowane znaczenie wyrazu *Wisła*, które określa dany obiekt).

Program ten można także wykorzystać jako element systemu tłumaczącego teksty z języka polskiego na angielski. Niektóre wyrazy w języku polskim mają kilka swoich odpowiedników w języku angielskim w zależności od tego, co oznaczają. Dzięki temu programowi (i oczywiście odpowiedniemu słownikowi znaczeń) możemy wybrać znaczenie (a więc i odpowiednią formę angielską), która najlepiej pasuje do danego kontekstu. Przykładowo dla polskiego wyrazu *zamek* w zależności od znaczenia możemy uzyskać angielski wyraz: *zip*, *castle*, *lock*.

Innym zastosowaniem tego programu może być użycie go w wyszukiwarce internetowej, w której oprócz słowa kluczowego możemy określić jego znaczenie, które nas interesuje i na podstawie słownika znaczeń jesteśmy w stanie sprawdzić, czy znalezione na stronie internetowej słowo ma takie znaczenie, o jakie nam chodzi, i w ten sposób możemy ograniczyć liczbę znalezionych stron do tych, które nam są potrzebne.

### 9.2. Ograniczenia programu

Program posiada wiele ograniczeń, które wynikają z wielkości i złożoności problemu, jakim jest przetwarzanie języka naturalnego, a rozpoznawanie znaczeń w szczególności. Programowi mogą sprawić trudność sytuacje, gdy dana forma w zdaniu reprezentuje dwa różne wyrazy, będące różnymi częściami mowy (np. *piec*), gdyż wtedy pojawiają się poważne problemy z określeniem zależności w zdaniu.

Kolejne ograniczenia są związane z tworzeniem zależności pomiędzy wyrazami w zdaniu. Algorytmy użyte w tym programie są dostosowane do zdań najprostszych: pojedynczych w stronie czynnej.

Dla innych zdań (na przykład zdań złożonych czy w stronie biernej) są stosowane te same algorytmy, co może spowodować, że pewne zależności nie zostaną wykryte lub zostaną źle określone.

### 9.3. Możliwe kierunki rozwoju

Jednym z ulepszeń, które można zastosować w programie, jest modyfikacja tego, co zostało opisane w podrozdziale poprzednim jako ograniczenia programu. Stworzenie programu analizującego poprawnie zdania w stronie biernej czy zdania złożone jest trudne do zrealizowania, ale mogłoby znacznie wzbogacić i zwiększyć możliwości tego programu.

Innym kierunkiem rozwoju może być napisanie algorytmów specjalnych dla niektórych wyrazów w naszym języku (na razie algorytmy są stworzone dla poszczególnych części mowy), bo niektóre wyrazy w języku polskim zachowują się trochę inaczej niż inne wyrazy reprezentujące tę samą część mowy. Do takich wyrazów można zaliczyć *być*, *zostać* itp. Można także przejrzeć i udoskonalić algorytmy wyszukujące wyrazy nadrzędne.

W miarę rozwoju słownika fleksyjnego można pójść także w kierunku głębszej z nim współpracy i gdy do tego słownika zostaną wprowadzone wyrazy wielosegmentowe (złożone), można próbować stworzyć algorytmy rozpoznające te wyrazy zdefiniowane w słowniku fleksyjnym bez potrzeby ich definiowania w słowniku znaczeń.

### Literatura

- [1] Lyons J.: *Wstęp do językoznawstwa*. PWN 1975
- [2] Lubaszewski W.: *O słownikach komputerowych i przetwarzaniu tekstu*. Gdańsk, PTSK 2001
- [3] Lubaszewski W.: *Gramatyka leksykalna w maszynowym słowniku fleksyjnym języka polskiego*. Kraków, IJP PAN 1997
- [4] Gajęcki M.: *Serwer leksykalny – narzędzie wspomagające przetwarzanie języka naturalnego* (raport z 25.01.2000)
- [5] Miller G.A., Beckwith R., Fellbaum C., Gross D., Miller K.: *Introduction to WordNet: An Online Lexical Database* (dostępne pod adresem [www.cogsci.princeton.edu](http://www.cogsci.princeton.edu))

Program działający według zasad opisanych w tej pracy można obejrzeć: <http://winnie.ics.agh.edu.pl/ls>