

Denise[®]

Wersja 5.0.

DOKUMENTACJA

Denise[®] 5.0 jest piątą wersją programu z zakresu Sztucznej Inteligencji [SI], a ściślej mówiąc, programem z kategorii Intelligent Dialogue Systems [IDS]. Jest to typ programów, których głównym zadaniem jest prowadzenie dialogu z dowolnym użytkownikiem programu – inną osobą bądź innym programem z rodziny IDS.

Celem nadrzędnym oprogramowania typu IDS jest zdanie testu Turinga bądź testu w ograniczonej formie, czyli konkursu Loebnera. Tym samym też osiągnięcie jednego z głównych wytycznych Sztucznej Inteligencji, czyli udowodnienie, że maszyny mogą myśleć – w sensie idei angielskiego matematyka, Alana Turinga.

Program Denise[®] jest częścią projektu naukowego i służy weryfikacji opracowywanych teorii. Jego publikacja wynika z zainteresowania, które obserwujemy, oraz dobrej woli autorów. Nie jest jednak w naszym interesie przystosowywanie programu dla niedoświadczonego użytkownika.



Denise 5.0.

MYTH Group® to ogólnopolska grupa ludzi skoncentrowana wokół prac nad programem Denise®. Przy pracy nad piątą wersją programu brały udział następujące osoby:

■ **Koordinacja, teoria, dokumentacja:**

Szymon Jessa

■ **Programowanie:**

Szymon Jessa (główny), Piotr Drapich, Stanisław Szymczyk, Piotr Radej, Daniel Horecki,

■ **Metodologia, teoria, dokumentacja:**

Marek Kasperski

■ **Projektowanie stron internetowych:**

Grzegorz Dzięgielewski (główny), Marek Kasperski (design), Piotr Mazurkiewicz (programowanie), Tomasz Pacyna (design)

■ **Kontakt:**

Mariusz Czopek – duncan@denise.gnu.pl

■ **Administrator listy dyskusyjnej:**

Tomasz Michalak

Specjalne podziękowania należą się: dr hab. Andrzejowi Dyszakowi za pomoc w zakresie polskiej gramatyki oraz przyjaciółom, ludziom, którzy pomagali każdego dnia.



Wszelkie poczynania grupy i rozwój programu można śledzić na oficjalnych stronach:

- <http://www.denise.gnu.pl>
- <http://www.denise.union.pl>

1. Jakie zdania można wpisywać?

Denise[®] 5.0 jest w stanie poprawnie zidentyfikować:

- Zdania pojedyncze rozwinięte, np. "Dobry komputer umie mówić i myśleć";
- Zdania współrzędnie złożone, np. "Ala ma kota, a kot ma samochód";
- Zdania z zaimkami, np. "Ktoś jest kimś", "Co kot ma?", "Kto ma kota?";
- Zdania z podmiotem domyślnym, np. "Mam inteligentny komputer", przy czym musi być ustalony (zidentyfikowany) rozmówca; pełny przykład: jestem Szymon → następuje identyfikacja rozmówcy mam komputer → podmiotem domyślnym jest rozmówca (Szymon)

1.1. Pomocne uwagi

Nie należy zrażać się w przypadku, kiedy program nie potrafi zidentyfikować zdania. Przyczyną może być brak jakiegoś słowa w bazie fleksji (w słowniku jest ok. 1.8 mln odmian, jednak niektórych słów, także tych powszechnie używanych, nadal brakuje), mały błąd w opisie gramatyki (szczególnie we wczesnych wersjach programu), ale także okazać się może (i jest to najczęstszym przypadkiem), że nie rozumiemy istoty zdań, które program poprawnie analizuje. Na przykład mówiąc "Tam stoi rower", mamy na myśli jakieś miejsce, które człowiek owszem, może zobaczyć, ale Denise[®] nie może zrozumieć tego zdania nie posiadając tego kontekstu. Program nie jest człowiekiem, i nie można próbować rozmawiać z nim w ten sam sposób, jak z ludźmi, którzy mają oczy, uszy, i całe bardzo skomplikowane ciało będące źródłem masy informacji pochodzących od zmysłów.

2. Kwestia identyfikacji podmiotu (tematu)

Program wymaga, aby podmiot był ściśle zidentyfikowany, tj. rozpoznany (ponownie) jeśli istnieje już w bazie faktów. Denise[®] może pytać o lepsze określenie podmiotu, jeśli zachodzi taka potrzeba – kiedy np. występuje kilka osób o tym samym imieniu. Program może również podać informacje pomocne w określaniu, jeśli tego zażądamy pytając zwyczajnie "dlaczego?".

Np.:

UŻYTKOWNIK: Programista Marek pisze programy.

UŻYTKOWNIK: Kucharz Marek piecze chleb.

UŻYTKOWNIK: Marek pisze programy?

DENISE: Co za 'Marek'? 'Programista', czy jak?

UŻYTKOWNIK: Programista.

DENISE: To prawda.

2.1. Pomocne uwagi

Identyfikację rozumie się tu jako jednoznaczne rozróżnienie danego obiektu (podmiotu) wśród zbioru wszystkich obiektów (podmiotów) opisanych w bazie faktów. Jako, że jest to pierwsza wersja programu posiadająca taką możliwość, rozwiązanie tu zastosowane na pewno nie jest najlepsze – m.in. dlatego, że identyfikacja dotyczy tylko podmiotu.

3. Elementy logiki

Stosowany jest mechanizm wnioskowania oparty na metodzie rezolucji.

Np.:

UŻYTKOWNIK: Ala ma kota albo Ala ma psa.
 UŻYTKOWNIK: Ala ma psa i kota?
 DENISE: Informacja sprzeczna.
 UŻYTKOWNIK: Ala nie ma psa.
 UŻYTKOWNIK: Ktoś ma kota?
 DENISE: To prawda.
 UŻYTKOWNIK: Kto?
 DENISE: Ala ma kota.
 UŻYTKOWNIK: Dlaczego?
 DENISE: To wynika z tego, że faktem jest, że Ala ma kota albo psa, oraz tego, że Ala nie ma psa.

Możliwe jest również wykorzystanie programu do zamiany wyrażeń logicznych na postać klauzul.

Np.:

UŻYTKOWNIK: not A and (B or C)
 DENISE: not A, B or C

W zapisie logicznym:

UŻYTKOWNIK: $\neg A \wedge (B \vee C)$
 DENISE: $\neg A, B \vee C$

3.1. Pomocne uwagi

Mechanizm wnioskowania oparty na tej metodzie mają zastosowanie na poziomie zdań pojedynczych, które stanowią tu zdarzenia elementarne. Analiza porównawcza treści zdań jest realizowana w odmienny sposób (patrz plik [logic/factscmp.dat](#)). Taka konstrukcja wynika z przestarzałej podstawy teoretycznej i zostanie zmieniona w następnej wersji programu – relacyjna baza danych będzie posiadała wszystkie obiekty i relacje identyfikowane, a wnioskowanie w całości odbywać się będzie w logice predykatów.

4. Kwantyfikatory

Można korzystać z kwantyfikatorów uniwersalnych i egzystencjalnych, jak np. *wszyscy*, *ktoś* / *coś* itp.

Np.:

UŻYTKOWNIK: Ala ma kota.
 UŻYTKOWNIK: Ktoś coś ma?
 DENISE: To prawda.
 UŻYTKOWNIK: Kto?
 DENISE: Ala ma kota.
 UŻYTKOWNIK: Wszyscy mają coś?
 DENISE: Brak danych.
 UŻYTKOWNIK: Wszyscy mają komputer.
 UŻYTKOWNIK: Ktoś ma komputer?

DENISE: To prawda.

UŻYTKOWNIK: Dlaczego?

DENISE: Ponieważ faktem jest, że wszyscy mają komputer.

5. Pomijanie określonych słów

Program dokonuje wstępnej filtracji wypowiedzi – usuwa wyrażenia, które zostały określone jako zbędne dla dalszej analizy.

Np. wypowiedź: "Myślisz, że Marek jest programistą?" zostanie zrozumiana jako "Marek jest programistą?".

6. Dostępne rozkazy

Aby zobaczyć listę dostępnych rozkazów, należy wpisać: **"/help"**. Jeśli chcemy uzyskać informacje n. t. określonej instrukcji, wpisujemy: **"/help instrukcja"**, np. **"/help set"**.

/about – wyświetlenie informacji o programie,

/help – wyświetlenie listy komend,

/status – wyświetlenie informacji statusowych,

/load config nazwa – wczytanie konfiguracji o podanej nazwie (brak nazwy - konfiguracja domyślna)

/save config nazwa – zapisanie konfiguracji z podaną nazwą (brak nazwy - konfiguracja domyślna)

Grupa komend **"/set nazwa"** i **"/get nazwa"**, gdzie funkcją **"set"** ustawiamy określony parametr, a funkcja **"get"** powoduje wyświetlenie informacji o aktualnym stanie danego parametru, natomiast jako "nazwa" można podać:

minwords liczba/OFF – min. liczba wyrazów (oddzielonych spacją) w zdaniu (OFF - brak ograniczeń)

mingroups liczba/OFF – min. liczba wyrażeń w zdaniu (OFF – brak ograniczeń)

maxgroups liczba/OFF – max. liczba wyrażeń w zdaniu (OFF – brak ograniczeń)

maxfleks liczba/OFF – max. liczba wyrazów (słownikowych) w zdaniu (OFF – brak ograniczeń)

maxwords liczba/OFF – max. liczba wyrazów w wyrażeniu (OFF – brak ograniczeń)

maxclauses liczba/OFF – max. liczba zdań pojedynczych w wypowiedzi (OFF – brak ograniczeń)

firstclause on/off – założenie: zdanie pierwsze nigdy nie jest podrzędne,

subjectfirst on/off – założenie: podmiot stoi zawsze przed orzeczeniem,

minprefixlen liczba/off – min. długość przedrostka w wyrazie (OFF – brak ograniczeń)

prefixcheck on/off – sprawdzanie występowania przedrostków,

prefixcheckmode 0/1 – tryb sprawdzania występowania przedrostków: 0-prosty/1-pełny

rememberfirst on/off – analiza zdania zostanie zakończona po pierwszym zaakceptowanym rozbiore,

phrasequality liczba(0-100) – ocena gramatyczna: m.in. jakość budowy pojedynczego wyrażenia,

structurequality liczba(0-100) – ocena gramatyczna: m.in. jakość związków pomiędzy wyrażeniami,

functionquality liczba(0-100) – ocena gramatyczna: m.in. jakość funkcji wyrażeń w zdaniu,

language nazwa – wybór aktualnego języka,

viewwords on/off – wyświetlanie informacji o przyjętym podzieleniu wyrazów w zdaniu,

viewclause on/off – wyświetlanie kolejnych dokonywanych rozbiorów zdań pojedynczych,

viewsyntax on/off – wyświetlanie ostatecznego rozbioru całej wypowiedzi,

viewsyntaxtime on/off – wyświetlanie czasu analizy gramatycznej,

viewpreliminary on/off – wyświetlanie analizy wstępnej wypowiedzi,

viewtheme on/off – wyświetlanie analizy tematu wypowiedzi,

viewinterlocutory on/off – wyświetlanie analizy konwersacyjnej wypowiedzi,

viewlogic on/off – wyświetlanie analizy logicznej wypowiedzi,

viewfact on/off – wyświetlanie informacji o analizie faktu,

viewunknownwords on/off – wyświetlanie słów, których nie odnaleziono w słowniku,

supervisor on/off – tryb SUPERVISOR,

logwords on/off – zapisywanie nieznanych słów w katalogu logbook,

logclauses on/off – zapisywanie do katalogu logbook wypowiedzi, których nie udało się zidentyfikować,

logdialog on/off – zapisywanie do katalogu logbook komunikatów z okna dialogu,

logreports on/off – zapisywanie do katalogu logbook komunikatów z okna informacyjnego.

1. Rozmowa

Na początku ważna uwaga, o której zbyt często zapominają użytkownicy programu: PROGRAM NIE JEST CZŁOWIEKIEM - więc proszę go tak nie traktować, bo dojdzie do nieporozumień! Program nie posiada zmysłu wzroku, słuchu, węchu, nie jest w stanie stwierdzić "naocznie", że "TAM" coś stoi, że zmieniają się ludzie przy komputerze... Co więcej, program ten nie zna schematów, jakimi posługujemy się w komunikacji, nie zna trendów językowych, slangów, nie czuje "atmosfery" i nie może się w nią wczuwać. Trzeba sobie uświadomić, że Denise[®] nie ma służyć rekonwalescentom IRC-netu, którym zabrakło pieniędzy na Internet, ani zastąpić innego rodzaju metody do tanich pogaduszek.

Denise[®] to NIE trochę rozbudowana ELIZA, która, gdyby miała głowę, to nic by w niej nie stało na przeszkodzie między uszami, a z ust (gdyby i te miała) płynęły pomieszane nasze własne słowa. Rozmowa z Denise[®] przebiega na zupełnie innym poziomie – przekazywane informacje zawarte w zdaniu, w jego treści, znajdują swoje trwałe odbicie w pamięci programu, podobne do tego, jakim jest idea, będąca naszym wzorcem dla skonstruowanego zdania. Jest to podstawą do uważania kogoś lub coś za posiadacza jakiejś formy inteligencji. Co więcej, podczas takiej komunikacji w naszym mózgu tworzymy wizerunek Denise[®] jako programu, którego działanie jest nam bliskie, ze względu na jego logiczną istotę rozumowania. Możemy przeto snuć strategię rozmowy, której zasady są te same, lub przynajmniej bliskie tym, na jakich opieramy się w rozmowie z człowiekiem. Nie trzeba pamiętać o tym, co napisałem, gdyż wstępnie obcowanie, a następnie już rozmowa z programem, w sposób naturalny pozwoli doświadczyć tego zjawiska, być może – bez uświadamiania sobie tego.

Na zakończenie tego przydługiego wstępu w sprawie prowadzenia rozmowy, przypominam i podkreślam raz jeszcze, że Denise[®] jest PROGRAMEM, którego nie można traktować jak człowieka, czy istoty żywej, posiadającej ciało. Pomimo tych "braków" z programem tym można prowadzić konwersację – musi ona opierać się na czystości językowej, i zrozumieniu drugiej strony. Rozmowa z Denise[®] może przypominać spotkanie z inteligentną rasą przyjaźnie nastawionych istot z odległej planety AI – wyobraźmy sobie, że nie mają ciała, uczuć, i niczego nie wiedzą o naszym świecie...

Początkowo myślałem, że jeśli napiszę w instrukcji, jakie zdania program potrafi zinterpretować, jakiego rodzaju konstrukcje gramatyczne są dopuszczalne, a jakie nie, to każdy będzie w stanie poprawnie korzystać z programu. Otóż, ciężko mi będzie popełnić równie wielką pomyłkę – wszystko to na nic się bowiem zdało, a moim i kolegów udziałem stały się krytyka poparta brakiem chęci zrozumienia/czytania instrukcji. Dlatego zdecydowałem się poprowadzić z Denise[®] przykładową rozmowę, i opisać co ważniejsze fragmenty. Mam nadzieję, że ta metoda okaże się przynajmniej nie gorsza od poprzedniej, której reprezentacja w formie opisu także znajduje się w tej dokumentacji.

Program został uruchomiony bez jakiegokolwiek bazy faktów:

Denise-MYTH 5.0Beta (15.04.2002) Intelligent Dialogue System
<musisz się przedstawić: jestem Twoje_imie>

- » Jesteś Denise
- » Jestem Szymon

KOMENTARZ: na początku TRZEBA program poinformować czym jest, żeby mógł się z tym zidentyfikować; jednocześnie ja muszę się przedstawić, aby program mógł mnie "zauważyć" jako osobę – nadawcę.

» Witaj Szymon

KOMENTARZ: powitanie oznacza, że program nas zidentyfikował i traktuje jak rozmówcę

» Czy umiesz myśleć?
» Brakuje danych

KOMENTARZ: program nie może dać ani odpowiedzi twierdzącej ani przeczącej, gdyż nie posiada żadnych informacji na ten temat w swojej bazie faktów.

» Ja umiem myśleć
» Brakuje danych

KOMENTARZ: ta sama sytuacja, przy czym program zapamiętał już fakt, że ja umiem myśleć,

» A czy jesteś inteligentna?
» Brakuje danych

KOMENTARZ: proszę zwrócić uwagę, że zdanie zostało sformułowane troszkę luźniej: zwrot "a czy" jest zupełnie zbędny, ale widać – nie wpływa na pracę programu, tego rodzaju elementy są interpretowane przy pomocy pliku "polski.fil", warto również zauważyć, że w zdaniu użyto podmiotu domyślnego.

» Wydaje mi się, że ja jestem inteligentny
» Brakuje danych

KOMENTARZ: tu również ten sam przypadek z inną formułą początkową; i tutaj również zwracam uwagę na podmiot – jest w formie zaimka, który poprawnie został przez program zinterpretowany,

» Ty jesteś programem, a ja jestem człowiekiem
» Brakuje danych

KOMENTARZ: ta wypowiedź składa się z dwóch zdań współrzędnie złożonych, połączonych spójnikiem "a"; powinna być już zauważalna pewna swoboda formułowania wypowiedzi rozmówcy, z którymi program potrafi sobie radzić,

» Uważasz, że jesteś kimś?
» Brakuje danych
» A jesteś czymś?
» To prawda

KOMENTARZ: tutaj napotykamy na różnicę w sposobie interpretacji zdań przez program i człowieka, który albo by nie zwrócił uwagi czy mówimy "kimś" czy "czymś", albo by nas poprawił, będąc świadom właściwej postaci zdania; rozmawiając z programem należy pamiętać, że w TEJ CHWILI ma on bardziej rygorystyczne wymagania, co do jakości formułowanych wypowiedzi niż człowiek.

» Czym?
» Faktem jest, że <Denise,być,program>

KOMENTARZ: proszę się zainteresować tym pytaniem – wyraźnie odnosi się ono do udzielonej wcześniej odpowiedzi na wcześniejsze pytanie – można więc mówić o pewnej analizie kontekstowej, czy wiedzy o temacie prowadzonej rozmowy; w drugiej kolejności widzimy problem z odmianą zdania – program nie potrafi jeszcze odmieniać zdań z orzeczeniem imiennym.

- » Czym jest program?
- » Brakuje danych
- » Program mówi jak działać
- » Brakuje danych
- » Więc wiesz czym jest program?

KOMENTARZ: jak widać, dla programu wszystkie te pytania dotyczą czegoś innego – człowiek w tym wypadku zdecydowałby się przekazać TO CO WIE na ten temat, choćby był tylko podobny do poruszonego w pytaniu – takie zachowanie człowieka wynika jednak m. in. z głębszego interpretowania treści zdania, co nie oznacza, że kolejne wersje programu nie będą mogły się nią również charakteryzować.

- » Brakuje danych
- » A wiesz co robi program?
- » Brakuje danych
- » Czy program może być inteligentny?
- » Brakuje danych
- » Myślę, że program może być inteligentny
- » Brakuje danych
- » Wiesz kto jest inteligentny?
- » Faktem jest, że <Szymon,być,inteligentny>

KOMENTARZ: proszę zwrócić uwagę, że program posiada tę informację ODE MNIE, pamiętając ją w swojej bazie faktów i WYKORZYSTUJĄC we wnioskowaniu.

- » Czy program jest inteligentny?
- » Brakuje danych
- » A czy program może myśleć?
- » Brakuje danych
- » Program może być inteligentny?
- » To prawda
- » Programy są? (2/2)

KOMENTARZ: jest to pierwsze PYTANIE zadane przez program; Denise[®] zadaje pytania na podstawie informacji, które posiada, toteż najpierw musi je zgromadzić, by móc pytać – w dodatku, informacji tych musi być odpowiednio dużo, i urozmaiconych, by pytania stawały się odpowiednio dobre.

- » Oczywiście, że programy są
- » Brakuje danych
- » On umie myśleć?

KOMENTARZ: zwracam uwagę, że Denise[®] kontynuuje rozmowę na zaproponowany temat, i próbuje się czegoś dowiedzieć o programach; jednocześnie widać, że program potrafi posługiwać się zaimkami (a w pytaniu wcześniejszym – podmiotem domyślnym), oraz formułować poprawne gramatycznie zdania.

- » A czy ty umiesz myśleć?
- » Brakuje danych
- » Szymon mówisz? (2/2)
- » Ja piszę

KOMENTARZ: Denise® poza tematem żywo interesuje się rozmówcą próbując się o nim czegoś dowiedzieć, pamiętając jednocześnie, że tematem jest PROGRAM.

- » Brakuje danych
- » Kto pisze?
- » Faktem jest, że pisz

KOMENTARZ: oczywiście, powinno być "piszesz", jednak program nie rozróżnia formy rozkazującej, stąd pojawia się możliwość jej wykorzystania – gdyby w bazie słów nie było tej formy, słowo nie pojawiłoby się.

- » On umie myśleć?
- » Sądzę, że nie umiesz myśleć
- » Brakuje danych
- » On umie myśleć?
- » Program nie umie myśleć
- » Brakuje danych

KOMENTARZ: proszę zwrócić uwagę na ciekawy fakt – Denise® wymogła na mnie podanie informacji czy program umie myśleć; niektórym może to przypominać zachowanie małych dzieci próbujących się czegoś dowiedzieć o świecie.

- » Programy piszą? (1/2)
- » Ty piszesz, i programy też piszą

KOMENTARZ: tego rodzaju zdania złożone bardzo ładnie się prezentują, a jednocześnie widać, że nie sprawiają programowi kłopotu; zauważamy też, że zainteresowanie Denise® programem nie gaśnie.

- » Brakuje danych
- » Kto pisze?
- » Faktem jest, że Szymon pisze
- » A ty piszesz?
- » To prawda
- » Mam dla ciebie zadanie
- » Brakuje danych
- » Ty myślisz albo ty piszesz?

KOMENTARZ: podkreślam, że użycie spójnika "albo" łączącego zdania, w znaczący sposób wikła treści obu zdań – spójnik ALBO wymaga, aby TYLKO JEDNA i DOKŁADNIE JEDNA z części połączonej wypowiedzi była prawdziwa,

- » Brakuje danych
- » Dlaczego?
- » O co pytasz?

KOMENTARZ: tutaj pokazuję mechanizm wyjaśniania odpowiedzi; w tym jednak przypadku program nie wie o co chodzi, gdyż przecież zabrakło mu danych na udzielenie odpowiedzi na wcześniejsze pytanie.

- » Ty myślisz?
- » Brakuje danych
- » Ty umiesz myśleć?
- » To fałsz (informacja sprzeczna)

KOMENTARZ: pierwszy raz widać, że program potrafi się nie zgodzić,

- » Ty umiesz myśleć lub ty piszesz?

» To prawda

KOMENTARZ: wykorzystanie spójnika "lub" jest analogiczne jak działanie operatora OR w logice, i oznacza, że zdanie jest prawdziwe, jeśli chociaż jedna z połączonych części jest prawdą.

» Dlaczego?

» Ponieważ faktem jest, że piszę

KOMENTARZ: tutaj widoczne jest już wykorzystanie mechanizmu wyjaśniania odpowiedzi; proszę zwrócić uwagę, że program mówi o sobie w pierwszej osobie.

» A czy umiesz myśleć?

» Brakuje danych

» Myślisz?

» Brakuje danych

» Mam zadanie?

» To prawda

» Dlaczego?

» Ponieważ faktem jest, że masz zadania

KOMENTARZ: jeszcze inny przykład.

» Program umie pisać?

» Mam zadanie dla ciebie

» Brakuje danych

KOMENTARZ: pozwalając sobie na pewną ignorancję pytań zadawanych przez program wcale nam nie grozi, że się na nas obrazi.

» Szymon mów? (1/2)

KOMENTARZ: proszę nie myśleć, że jest to objaw zniecierpliwienia, choć trzeba przyznać, że reakcja programu jest zaskakująco naturalna – w rzeczywistości jest to zwyczajne pytanie, do którego LOSOWO została użyta forma rozkazująca.

» Mam zadanie dla ciebie?

» To prawda

» Powiedz, czy ty umiesz myśleć albo ty piszesz?

» To prawda

KOMENTARZ: proszę zauważyć: spójnik "albo".

» Dlaczego???

» Co wynika z tego, że

» Faktem jest, że <Denise,!nie,umieć,myśleć>

» Oraz umiem myśleć

» Co wynika z tego, że

» Przyjęto hipotezę, że nie jest prawdą, że piszę lub umiem myśleć

» Oraz faktem jest, że piszę

KOMENTARZ: jak widać, program zdolny jest do przeprowadzania dość skomplikowanych operacji wnioskowania, za którymi kryją się proste odpowiedzi; jednocześnie wnioskowanie to nie jest UKRYTE i może zostać przedstawione rozmówcy w dość czytelnej postaci – wymagana jest jednak pewna koncentracja jeśli chciałaby je prześledzić osoba nie przygotowana na takie uzasadnienie. Więc... Wypowiedź nie zawiera żadnej informacji oczywiście, dla programu.

- » Czy umiesz myśleć?
- » Brakuje danych
- » A kim jesteś?
- » Brakuje danych
- » Kim jesteś albo czym ty jesteś?
- » Brakuje danych

KOMENTARZ: brakuje danych, gdyż chociaż program wie czym jest, to nie wie, czy jest kimś, ani czy też kimś nie jest.

- » Dlaczego?
- » O co pytasz?
- » Czym jesteś?
- » Faktem jest, że <Denise,być,program>
- » Kim jesteś lub czym ty jesteś?

KOMENTARZ: stosowanie spójnika lub da nam dużo więcej informacji, których oczekujemy.

- » Faktem jest, że <Denise,być,program>
- » A czym jest program?
- » Brakuje danych
- » Czy program umie myśleć?
- » Brakuje danych
- » On umie pisać?
- » Program umie pisać albo ty nie umiesz pisać

KOMENTARZ: to stwierdzenie zostało przez program zapamiętane.

- » Brakuje danych
- » Ty nie umiesz pisać?
- » To fałsz (informacja sprzeczna)
- » Program nie umie pisać?
- » To fałsz (informacja sprzeczna)
- » Program umie pisać?
- » To prawda
- » Dlaczego?
- » Ponieważ program umie pisać
- » Co wynika z tego, że
- » Faktem jest, że umiem pisać
- » Oraz faktem jest, że program umie pisać lub <Denise,!nie,umieć,pisać>

KOMENTARZ: proszę jedynie zwrócić jeszcze uwagę na to, że Denise[®] nie posiada w bazie faktu stwierdzającego, że program umie pisać – wie jedynie, że Denise[®] umie pisać, oraz że albo program umie pisać, albo ona nie umie pisać, (skoro jest programem, i programy by nie umiały pisać, to jakże ona ma umieć pisać?), z czego wynika, że programy umieją pisać (przynajmniej niektóre).

UWAGA!

Z powyższej rozmowy wycięte zostały fragmenty ze zdaniem, których program nie mógł zinterpretować - były dwa powody: brak wyrazów w bazie fleksji, oraz braki w pliku "polski.fil"; rozwój obu tych części jest jednak kwestią ilościową, a nie jakościową, i pozostaje kwestią czasu wyeliminowanie tego rodzaju problemów.

2. Katalog FACTS

W tym katalogu znajdują się bazy faktów. Lista wykorzystywanych przez program baz faktów jest podana w pliku `facts-list` – w każdej linii podana jest nazwa bazy faktów, która zostanie wczytana przez program. Jeśli chcemy dodać do programu jakąś bazę faktów, to należy:

- 0) jeśli `Denise`[®] jest uruchomiona, to wyłączyć ją,
- 1) przegrać bazę do katalogu `facts`,
- 2) dopisać do pliku `facts-list` (jeśli plik nie istnieje, to należy go utworzyć) nazwę przegranej bazy faktów,
- 3) można uruchomić `Denise`[®] i sprawdzić, czy dodana baza została poprawnie odczytana,

Każda baza faktów posiada trzy pola konfiguracji, znajdujące się w liniach numer 2, 3, 4.

W linii nr 1 znajduje się słowo "DATABASE". Parametry te muszą być zmieniane ręcznie, przy użyciu dowolnego edytora tekstowego. Opis parametrów:

- 1) parametr z linii numer 2 – dotyczy dodawania nowych tematów do bazy wartości parametru:

1-ACCEPT_NEW_THEMES – do bazy mogą być dodawane nowe tematy, (domyślnie)
 2-NO_NEW_THEMES – do bazy nie mogą być dodawane nowe tematy,

- 2) parametr z linii numer 3 – dotyczy dodawania nowych faktów do bazy, wartości parametru:

1-ACCEPT_NEW_FACTS – do bazy mogą być dodawane nowe fakty, (domyślnie)
 2-NO_NEW_FACTS - do bazy nie mogą być dodawane nowe fakty,

- 3) parametr z linii numer 4 – dotyczy przekierowywania tematów, przekierowanie tematu następuje, kiedy w dwóch bazach pojawi się ten sam temat – wtedy należy jeden "wyłączyć", gdyż nie może być dwóch takich samych różnych tematów. wartości parametru:

1-ACCEPT_FORWARD_THEMES – baza umożliwia przekierowywanie tematów, (domyślnie)
 2-NO_FORWARD_THEMES – baza nie umożliwia przekierowywania tematów,

Możliwe jest skonfigurowanie danych w ten sposób, że w jednej bazie znajdują się tematy, a w drugiej fakty. Uczyni to te bazy bezużyteczne, jeśli zostaną wczytane jedna bez drugiej. Wykorzystywane specjalne identyfikatory baz zapewniają, że prawdopodobieństwo pomylenia baz i tematów jest bardzo małe (praktycznie zerowe).

3. Katalog LANGUAGES

Opis plików zawartych w tym katalogu znajduje się w osobnej dokumentacji, w pliku o nazwie `Pliki_jezykowe.txt`.

4. Katalog LOGBOOK

W tym katalogu znajdują się pliki, do których program zapisuje przebieg rozmowy. Przy czym istnieje kilka plików, z których każdy służy innym celom:

- **words.txt** – tu znajdują się wyrazy, których nie odnaleziono w słowniku,
- **clauses.txt** – tu zapisywane są zdania, których program nie zdołał zidentyfikować,
- **dialog.txt** – w tym pliku zapamiętywane są wszystkie teksty z okna dialogu,
- **reports.txt** – w tym pliku zapamiętywane są wszystkie komunikaty z okna informacyjnego.

Dzięki analizie tych plików można uzupełniać słownik, poprawiać parametry analizy zdania, korygować sposób prowadzenia rozmowy, jak również eliminować inne usterki, czy optymalizować działanie programu w krytycznych momentach.

Jeśli ktoś spotka się z błędami, lub uzna, że w czasie rozmów zaszło coś dziwnego, co powinniśmy zobaczyć, powinien wysłać cały katalog z wszystkimi plikami na nasz adres e-mail [duncan@denise.gnu.pl], lub dostarczyć w inny sposób. W przypadku ciekawych rozmów, proszę o wyrażenie zgody na ich publikację w dokumentacji, na stronie [www \[http://www.denise.gnu.pl\]](http://www.denise.gnu.pl) <http://www.denise.union.pl>], czy w innym miejscu – bez tej zgody rozmowy nie zostaną nigdzie zamieszczone.

5. Katalog LOGIC

W katalogu LOGIC znajdują się dwa pliki: **factscmp.dat** i **conversation.dat**. Oba pliki regulują kwestie rozmowy, jednak na innym poziomie: w pliku **factscmp** zawarte są informacje z zakresu porównywania zdań, są więc istotne na głębokim poziomie analizy; w pliku **conversation.dat** opisane są zasady ważne na wyższym poziomie pracy programu – dotyczą prowadzenia rozmowy. Dopuszczalne jest modyfikowanie pliku **conversation.dat** przez wszystkich, bez znaczenia na zasób wiedzy na temat działania programu. Natomiast modyfikowanie pliku **factscmp.dat** bez posiadania wiedzy na temat wykorzystania tego pliku w programie, może prowadzić do błędnego działania programu już u samych podstaw analizy wypowiedzi. Wykrycie takiego błędu jest praktycznie niewykonalne przez osobę nie posiadającą dokładnych informacji o budowie programu, i jego działaniu. Dlatego zaleca się, aby plik **factscmp.dat** pozostawić w postaci jaką dostarczają autorzy, natomiast plik **conversation.dat** może z powodzeniem służyć do manipulowania zachowaniem programu.

5.1. Opis pliku factscmp.dat

W pliku tym zawarte są opisy przejść stanów logicznych w fazie porównywania zdania pojedynczego z innym zdaniem pojedynczym. Zdania porównywane są poprzez porównywanie wyrażień sobie odpowiadających – pod względem struktury w zdaniu oraz funkcji. Każde przejście jest opisane w jednej linii, o formacie:

state, posnegA, posnegB, comparison result, new state, operation, groups

Gdzie:

state – aktualny stan porównywania zdań, może przyjmować wartości:

1-NULL – kiedy porównywane jest dopiero pierwsze wyrażenie,
 3-A_EQU_B – kiedy do tej pory zdania były identyczne,
 4-A_IN_B – kiedy zdanie A zawiera się w zdaniu B,
 5-B_IN_A – kiedy zdanie B zawiera się w zdaniu A,
 6-A_PR_B – kiedy zdanie A precyzuje zdanie B.
 7-B_PR_A – kiedy zdanie B precyzuje zdanie A,
 8-A_EQU_B+NEG – kiedy zdania były sprzeczne (np. "program myśli" i "program NIE myśli"),
 9-A_IN_B+NEG – kiedy zdanie A zawiera się w zdaniu B i jest w stosunku do niego sprzeczne,
 10-B_IN_A+NEG – kiedy zdanie B zawiera się w zdaniu A i jest w stosunku do niego sprzeczne,
 11-A_PR_B+NEG – kiedy zdanie A precyzuje zdanie B i jest z nim sprzeczne,
 12-B_PR_A+NEG – kiedy zdanie B precyzuje zdanie A i jest z nim sprzeczne,

posnegA – czy wyrażenie jest zanegowane, czy nie; przyjmuje wartości:

1-NULL – kiedy zdanie nie ma już wyrażenia na tym poziomie w strukturze składniowej,
 2-POS – kiedy wyrażenie nie jest zanegowane,
 3-NEG – kiedy wyrażenie jest zanegowane (np. "nie myśli"),

posnegB – jw. ale dotyczy wyrażenia drugiego (B),

comparison result – wynik porównania ostatnich porównywanych wyrażen, może przyjmować wartości:

2-A_DIF_B – kiedy wyrażenia są różne,
 3-A_EQU_B – kiedy są te same,
 4-A_IN_B – kiedy wyrażenie A zawiera się w B,
 5-B_IN_A – kiedy wyrażenie B zawiera się w A,
 6-A_PR_B – kiedy wyrażenie A precyzuje B,
 7-B_PR_A – kiedy wyrażenie B precyzuje A,

new state – stan wynikowy, który będzie aktualny podczas porównywania następnego wyrażenia. Przyjmuje te same wartości co 'state' z wyjątkiem 1-NULL.

operation – określa typ operacji do wykonania. Przyjmuje wartości:

1-NEXT – porównywanie będzie trwało dalej,
 2-END – porównywanie zostanie zakończone i ostatecznym wynikiem będzie "new state".

groups – mówi jakich wyrażen może dotyczyć dane przejście. Typy wyrażen:

1-ALL_GROUPS – wszystkie wyrażenia,
 2-ONLY_PREDICATE – tylko orzeczenia,
 3-WITHOUT_PREDICATE – wszystkie wyrażenia poza orzeczeniami.

5.2. Opis pliku conversations.dat

Plik ten odpowiada za rozmowę. W pliku opisane jest JAK reagować ma program, jak zachowywać się w trakcie rozmowy. To tutaj określa się, czy Denise® ma odpowiadać na pytania, jak ma na nie odpowiadać, czy i kiedy ma zadawać pytania, i jakie to mają być pytania, czy ma zapamiętywać nowe informacje itp. Łatwo zauważyć, że jest to takie małe centrum dowodzenia – można tak sterować rozmową, aby osiągnąć pożądany przez nas efekt, uzyskując bez większego problemu program zachowujący się zupełnie inaczej niż wersja oryginalna. Elastyczność jest jedną z głównych cech inteligencji, a plik ten służy właśnie zwiększeniu elastyczności programu.

W kolejnych wersjach plik ten będzie się stopniowo rozbudowywał, poszerzany o nowe akcje wykonywane przez program, oraz inne funkcjonalne elementy. Umiejętność modyfikowania tego pliku tak, aby osiągnąć oczekiwane rezultaty, jest na pewno przydatna w obcowaniu z programem.

Plik `conversation.dat` składa się z dwóch części – zbioru definicji nazw zastępczych dla operatorów opisu, i samego opisu przedstawionego z wykorzystaniem podanych definicji.

Oto operatory i przykładowe (stosowane w wersji 5.0) nazwy zastępcze:

```

*** operatory ogólne ***
# początek opisu (część warunkowa)
1:
=jeśli
=if
#
# początek opisu (część wykonawcza)
2:
=to
=then
#
# spójnik I (łączy wyrażenia)
3:
=i
=and
#
# spójnik LUB (łączy wyrażenia)
4:
=lub
=or
#
# brak parametru (NULL)
5:
=NULL
#
# *** akcje użytkownika ***
#
# użytkownik wprowadza zdanie oznajmujące
10:
=rozmówca_oznajmia
=user_announce
#
# użytkownik zadaje pytanie
11:
=rozmówca_pyta
=user_question
#
# *** akcje denise ***
#
# denise odpowiada na zdanie użytkownika
20:
=DeniseOdpowiedz
=DeniseAnswer
#
# 20 lub 20(arg1)
#
# arg1=2001,2002,2003
# odpowie w sposób, jaki oczekuje rozmówca
2001:

```



```

=LIKE_PREFERED
=JAK_OCZEKIWANO
# odpowie przedstawiając całe wnioskowanie
2002:
=WITH_PROVE
=POKAZ_DOWOD
# odpowie jednym zdaniem
2003:
=SIMPLE_ANSWER
=PROSTA_ODPOWIEDZ
#
# denise zadaje pytanie
21:
=DeniseZapytaj
=DeniseQuestion
#
# 21 lub 21(arg1) lub 21(arg1,arg2,arg3)
#
# arg1=2111,2112,2113,2114,2115,2116
# tematem będzie denise
2111:
=SUBJECT_DENISE
# tematem będzie użytkownik
2112:
=SUBJECT_USER
# tematem będzie aktualny temat rozmowy
2113:
=SUBJECT_THEME
# temat zostanie wybrany losowo
2114:
=SUBJECT_RANDOM
# zostanie wybrany temat najmniej znany przez denise
2115:
=SUBJECT_MOST_UNKNOWN
# zostanie wybrany temat najbardziej znany przez denise
2116:
=SUBJECT_MOST_KNOWN
#
# arg2=2121,2122,2123,2124,2125
# pytanie zostanie utworzone z tematu najbardziej podobnego podmiotowi
pytania
2121:
=FROM_MOST_ALIKE
# pytanie zostanie utworzone z tematu najbardziej różnego podmiotowi pytania
2122:
=FROM_MOST_DIFFERENT
# pytanie zostanie utworzone z tematu, który jest najbardziej znany przez
denise
2123:
=FROM_MOST_KNOWN
# pytanie zostanie utworzone z tematu, który jest najmniej znany przez denise
2124:
=FROM_MOST_UNKNOWN
# pytanie zostanie utworzone z tematu, wybranego losowo
2125:
=FROM_RANDOM
#
# arg3=2131,2132,2133
# w temacie, który posłuży do utworzenia pytania nie może być faktów
sprzecznych z tematem podmiotu

```

```

2131:
=MODE_WITHOUT_NEG
# w temacie, który posłuży do utworzenia pytania muszą być fakty sprzeczne z
tematem podmiotu
2132:
=MODE_WITH_NEG
# temat, który posłuży do utworzenia pytania może mieć fakty sprzeczne z
tematem podmiotu
2133:
=MODE_FREE
#
# denise ma zapamiętać w bazie faktów podana informacje (pod warunkiem, że
nie jest sprzeczna)
22:
=DeniseZapamietaj
=DeniseRemember
#
# *** zmienne wewnętrzne ***
#
# jest zidentyfikowany temat DENISE
30:
=denise_defined
=denise_zdefiniowana
#
# jest zidentyfikowany użytkownik
31:
=user_defined
=rozmówca_zdefiniowany
#
# jest zidentyfikowany temat
32:
=theme_defined
=temat_zdefiniowany
#
# WYRAŻENIA ZŁOŻONE
#
# *** akcje alternatywne ***
#
# [pp%=action1(params);qq%=action2(params)...;zz%=NULL]
#
# UWAGI
# w wyrażeniu warunkowym może występować tylko jeden rodzaj spójnika:
AND lub OR
#
Druga część pliku opisuje zachowanie programu. Schemat zapisu jest
następujący:

```

Część warunkowa część wykonawcza.

Część warunkowa to zestaw warunków, których spełnienie będzie oznaczało wykonanie zadań umieszczonych w części wykonawczej. Istotna jest kolejność umieszczania warunków (wyrażeń z częścią warunkową i wykonawczą) – program będzie poszukiwał pierwszego wyrażenia, którego część warunkowa będzie spełniona – pozostałe (zapisanie poniżej niego) nie będą sprawdzane.

Przykład najprostszego zestawu:

```

if user_question then DeniseAnswer
if user_announce then DeniseRemember and DeniseQuestion

```

Zapis ten oznacza, że program będzie odpowiadał na pytania rozmówcy, oraz będzie zapamiętywał twierdzenia i zadawał pytania w odpowiedzi na nie.

Oto bardziej skomplikowany zestaw reguł, wykorzystany w programie Denise[®] 5.0. W tym przypadku zestawy reakcji zostały uszeregowane względem posiadanych przez program informacji o rozmówcy, samym programie i temacie. Wprowadzono też element losowy i kilka innych elementów.

```
#
# *** wszystko zdefiniowane (temat, rozmówca, Denise)
#if user_defined and denise_defined and theme_defined
#
# * denise odpowiada na pytania, czasem zadaje własne
if user_defined and denise_defined and theme_defined and user_question then
DeniseAnswer and
[50%=DeniseQuestion(SUBJECT_THEME);10%=DeniseQuestion(SUBJECT_U
SER);40%=NULL]
#
# * denise zapamiętuje informacje, ocenia je i często zadaje pytanie
if user_defined and denise_defined and theme_defined and user_announce
then DeniseRemember and DeniseAnswer and
[75%=DeniseQuestion(SUBJECT_THEME);5%=DeniseQuestion(SUBJECT_U
SER);5%=DeniseQuestion(SUBJECT_RANDOM);15%=NULL]
#
# *** nie jest zdefiniowany temat (jest rozmówca i Denise)
#if user_defined and denise_defined
#
# * denise odpowiada na pytania w wybrany przez rozmówcę sposób i ew.
zadaje pytanie
if user_defined and denise_defined and user_question then DeniseAnswer and
[25%=DeniseQuestion(SUBJECT_RANDOM);75%=NULL]
#
# * denise zapamiętuje informacje i ocenia je, oraz ew. zadaje pytanie
if user_defined and denise_defined and user_announce then DeniseRemember
and DeniseAnswer and
[25%=DeniseQuestion(SUBJECT_RANDOM);25%=DeniseQuestion(SUBJECT
_MOST_UNKNOWN);25%=DeniseQuestion(SUBJECT_USER);25%=NULL]
#
# *** nie jest zdefiniowana denise (jest temat i rozmówca)
#if user_defined and theme_defined
#
# * denise nie odpowiada na pytanie (bo jej nie ma)
if user_defined and theme_defined and user_question then NULL
#
# * denise zapamiętuje informacje, i pyta o temat lub rozmówcę
if user_defined and theme_defined and user_announce then DeniseRemember
and
[50%=DeniseQuestion(SUBJECT_THEME);50%=DeniseQuestion(SUBJECT_U
SER)]
#
# *** nie jest zdefiniowany rozmówca (jest temat i denise)
#if denise_defined and theme_defined
#
# * denise nie odpowiada na pytania (bo komu?)
if denise_defined and theme_defined and user_question then NULL
#
# * denise zapamiętuje informacje i pyta o temat lub (rzadziej) o denise
if denise_defined and theme_defined and user_announce then
DeniseRemember and
```

```

[75%=DeniseQuestion(SUBJECT_THEME);25%=DeniseQuestion(SUBJECT_D
ENISE)]
#
# *** tylko rozmówca zdefiniowany
#if user_defined
#
# * denise nie odpowiada na pytania
if user_defined and user_question then NULL
#
# * denise zapamiętuje nowe informacje i pyta o rozmówcę
if user_defined and user_announce then DeniseRemember and
DeniseQuestion(SUBJECT_USER,FROM_MOST_ALIKE,MODE_FREE)
#
# *** tylko denise zdefiniowana
#if denise_defined
#
# * denise nie odpowiada na pytania
if denise_defined and user_question then NULL
#
# * denise zapamiętuje nowe informacje i pyta o sama siebie (jako, ze brak
rozmówcy, to
# denise chce dowiedzieć się czegoś o sobie – w porównaniu z innymi
podobnymi rzeczami)
if denise_defined and user_announce then DeniseRemember and
DeniseQuestion(SUBJECT_DENISE,FROM_MOST_ALIKE,MODE_FREE)
#
# *** tylko temat zdefiniowany
#if theme_defined
#
# * denise nie odpowiada na pytania
if theme_defined and user_question then NULL
#
# * denise zapamiętuje nowe informacje i pyta o temat (jako, ze brak rozmowy,
to
# informacja niejako "ze słownika", wiec denise chce rozróżnić temat)
if theme_defined and user_announce then DeniseRemember and
DeniseQuestion(SUBJECT_THEME,FROM_MOST_DIFFERENT,MODE_FREE
)
#
# *** nic nie jest zdefiniowane
#if
#
# * denise nie odpowiada na pytania
if user_question then NULL
#
# * denise zapamiętuje nowe informacje
if user_announce then DeniseRemember
#
#
# np.
# jeśli użytkownik cos stwierdzi, to denise albo zada pytanie, albo powie, czy to
prawda, czy fałsz
# prawdopodobieństwo zadania pytania wynosi 50%, odpowiedzi także 50%
# pytanie zostanie zadane na dowolny, losowo wybrany temat, na podstawie
innego, dowolnego,
# bez zwracania uwagi na istnienie faktów sprzecznych w obu tych tematach,
#if user_announce then
[50%=DeniseQuestion(SUBJECT_RANDOM,FROM_RANDOM,MODE_FREE);
50%=DeniseAnswer]
#

```

1. Technika analizy składniowej

W swojej historii Denise[®] miała zaimplementowane trzy zupełnie różne techniki analizy zdania. Pierwsza była najbardziej ograniczona, gdyż opierała się na założonych *ad hoc* zasadach budowy zdania. W języku polskim jednak dominuje raczej koniugacja wyrazów, aniżeli ich kolejność, toteż algorytm analizy musiał być bardzo skomplikowany i złożony, w rzeczywistości realizując coś bardzo prostego i ułomnego. Technika ta polegała na znajdowaniu części zdania każdej z osobna, i jej wyniki nie były imponujące, choć zwracały uwagę.

W wersji czwartej Denise[®] zastosowano algorytm oparty o analizę wszystkich możliwych struktur rozbioru zdania, i wybór najlepszego na podstawie oceny gramatycznej. Ocena gramatyczna sprawdzała, czy taki a taki rozbiór jest gramatycznie poprawny. Zdarzało się jednak, że ze względu na brak dodatkowych warunków uwzględniających specyficzne konstrukcje niektóre ze zdań posiadać mogły więcej niż jeden poprawny rozbiór logiczny – zdania rozumiane były więc niejako wieloznacznie. Z drugiej strony zdarzały się zdania, których program nie był w stanie zrozumieć. Algorytm nadal dotyczył tylko zdania pojedynczego rozwiniętego z wieloma ograniczeniami.

Obecnie stosowana technika likwiduje prawie wszystkie ograniczenia, jakie miały dwie poprzednie. Rozciąga się na zdania wielokrotnie złożone, umożliwia tworzenie wyrażen złożonych współrzędnie (tj. konstrukcji: "Darek i Marek", "pies albo koza", czy wręcz "Darek, Marek i pies"), oraz wykorzystuje inny sposób oceny gramatycznej pozwalający na rozumienie zdań, które nie są poprawne gramatycznie. Jediną wadą tej techniki jest jej czasochłonność. Z drugiej strony odpowiednie katalizatory umożliwiają znaczne przyśpieszenie działania algorytmu, tak, że w obecnych warunkach, algorytm ten może być praktycznie stosowany w komputerach domowych. Z czasem natomiast jego możliwości będą mogły być wykorzystywane w coraz większym zakresie obejmując, mam nadzieję, wszystkie poddawane analizie zdania.

W celu ostatecznej oceny funkcjonalności algorytmu należy spojrzeć na program jako całość, wraz z funkcjami gramatycznymi, obejmującymi również obsługę operacji słownikowych (fleksja).

Opisując najprościej jak można, zadaniem algorytmu jest znalezienie wszystkich możliwych struktur składniowych podanej wypowiedzi. Dotyczy to szeregu poziomów: określenia granic między zdaniami pojedynczymi wypowiedzi złożonej, określenie jak mają się do siebie te wypowiedzi (współrzędnie, podrzędnie), a następnie dokonanie podobnej analizy struktury wewnętrznej każdego ze zdań (powiązanie wyrazów w grupy, opisanie, jakie grupy są dla jakich nadrzędne, oraz określenie funkcji pełnionej przez grupę w zdaniu).

Możliwe, że powstanie wątpliwość, dlaczego tak złożony proces, dlaczego sprawdzać wszystkie możliwości, skoro można ocenić "na oko", jak wygląda struktura zdania, wiedząc na przykład, że czasownik będzie orzeczeniem, a rzeczownik podmiotem itd. Otóż, tutaj muszę rozwiać nadzieje na tak proste rozwiązanie problemu – taka technika może objąć tylko naprawdę niewielki zakres wypowiedzi, i do tego założone warunki zmuszą nas do sztywnego trzymania się formy, co ostatecznie jasno podkreśli, że mamy do czynienia tylko ze zwykłym programem, a nie czymś, co ma choćby porównywalne do człowieka możliwości rozumienia zdań. Ponadto w komunikacji naturalnej bardzo rzadko da się ze 100% pewnością dokonać poprawnej analizy składniowej, gdyż popełniamy błędy, przejęzyczenia, przestawienia, mylimy formy, zapominamy czegoś dopowiedzieć, a w szczególności – notorycznie powtarzamy niektóre błędy, których nie jesteśmy świadomi. Nowy algorytm potrafi sobie z tym wszystkim poradzić, a wspomóżony modułem ekstrakcji informacji może stanowić silnego partnera w mowie potocznej.

Pamiętać przy tym należy, że generowane przezeń zdania, oraz samo "rozumienie" ich są POPRAWNE, pomimo możliwego zafalszowania informacji wejściowej. Już tylko to mówi o jakości stosowanej metody.

Na zakończenie jedyna przykra informacja. Z praktycznego punktu widzenia, bez wykorzystania procesów katalizujących, dla pełnego wykorzystania możliwości algorytmu musielibyśmy mieć do dyspozycji bardzo szybki komputer. Dokładniejsze dane na temat jego szybkości będą mogły zostać podane przy ostatecznym testowaniu kompletnego systemu analizy składniowej. Jestem jednak przekonany, że w ciągu najbliższego czasu szybkości dostępnych komputerów domowych staną się wystarczające dla zapewnienia programowi warunków do najbardziej wydajnej pracy.

Z drugiej strony, na drodze szukania algorytmu unifikującego opisane poniżej algorytmy częściowe, można osiągnąć niepomiarne wielkie przyspieszenie działania analizy. Odbyć się to musi kosztem pracy wkładanej w algorytm, na co jednak w chwili obecnej nie mogę sobie pozwolić nie dysponując wystarczająco wielkim zespołem *developerskim*. Dlatego też osoby chętne do współpracy są zawsze mile widziane.

2. Podział wypowiedzi na zdania pojedyncze i równoważniki zdań

Wypowiedź składa się z szeregu jednostek elementarnych pod względem funkcyjnym. Są to oczywiście zdania (w tym równoważniki zdań, oraz zdania nie pełne, opierające się na kontekście). Wypowiedź jest niejako reprezentacją logiczną zależności jednostek elementarnych, czego odzwierciedleniem jest logika predykatów. Dla przykładu wypowiedź "jest już późno, więc zaraz pójdę spać", składa się z dwóch zdań pojedynczych: "jest już późno" i "zaraz pójdę spać".

Zdania te, nazwijmy je odpowiednio zdaniem A i zdaniem B, połączone są spójnikiem (są w relacji); WIĘC (wynikowy). Szczegółowy opis logiki występującej w zdaniach, wraz z charakterystyką jej implementacji w *Denise*[®], znaleźć będzie można prawdopodobnie w specjalnym artykule temu poświęconym. Teraz wystarczy wiedzieć, że wypowiedź musi zostać podzielona na zdania pojedyncze. Postępując zgodnie z opisaną wcześniej techniką, musimy znaleźć wszystkie możliwe podziały wypowiedzi. Poniżej przedstawiam garść orientacyjnych informacji z zakresu realizacji tego procesu.

Warunki:

- Między wyrazami jednego zdania nie ma wyrazów innego zdania; poza zdaniem pierwszym, które może zawierać wyrazy znajdujące się za zdaniem drugim, lecz przed kolejnymi zdaniami;
- Zdanie może mieć min. 1 wyraz;
- Np. 1 2 3 4 5 (wyrazy).

Możliwe zdania:

- | | | |
|----|---------------------|-------|
| 1. | {1},{2},{3},{4},{5} | %0000 |
| 2. | {1},{2},{3},{4 5} | %0001 |
| 3. | {1},{2},{3 4},{5} | %0010 |
| 4. | {1},{2},{3 4 5} | %0011 |

5.	{1},{2 3},{4},{5}	%0100
6.	{1},{2 3},{4 5}	%0101
7.	{1},{2 3 4},{5}	%0110
8.	{1},{2 3 4 5}	%0111
9.	{1 2},{3},{4},{5}	%1000
10.	{1 2},{3},{4 5}	%1001
11.	{1 2},{3 4},{5}	%1010
12.	{1 2},{3 4 5}	%1011
13.	{1 2 3},{4},{5}	%1100
14.	{1 2 3},{4 5}	%1101
15.	{1 2 3 4},{5}	%1110
16.	{1 2 3 4 5}	%1111

* Z uwzględnieniem rozbicia zdania pierwszego:

17.	{1 {2} 3},{4},{5}	czyli: {1 3},{2},{4},{5}
18.	{1 {2} 3},{4 5}	czyli: {1 3},{2},{4 5}
19.	{1 {2} 3 4},{5}	czyli: {1 3 4},{2},{5}
20.	{1 {2} 3 4 5}	czyli: {1 3 4 5},{2}
21.	{1 {2 3} 4},{5}	czyli: {1 4},{2 3},{5}
22.	{1 {2 3} 4 5}	czyli: {1 4 5},{2 3}
23.	{1 {2 3 4} 5}	czyli: {1 5},{2 3 4}
24.	{1 2 {3} 4},{5}	czyli: {1 2 4},{3},{5}
25.	{1 2 {3} 4 5}	czyli: {1 2 4 5},{3}
26.	{1 2 {3 4} 5}	czyli: {1 2 5},{3 4}
27.	{1 2 3 {4} 5}	czyli: {1 2 3 5},{4}

Wzór na liczbę możliwych zdań dla N wyrazów:

$$2^{(N)-1} + 2^{(N)-1} = 2^{(N)-N}$$

W przykładzie, dla N = 5: (16) + (11) = $2^{(5)-5} = 27$

Dla 10 wyrazów: $2^{(10)-10} = 1014$ możliwości.

Przykład z uwzględnieniem rozbicia zdania pierwszego: "Uczniowie, których bardzo lubię, mają u mnie dobrze."

3. Struktura zdań i równoważników zdań

Struktura zdań (w domyśle również równoważników zdań) to nic innego jak określanie które zdanie jest nadrzędne, podrzędne, a które do którego współrzędne. Dla przykładu: "Mam komputer, który umie mówić" składa się ze

zdań: "(ja) mam komputer" i "(komputer) umie mówić", które są do siebie w stosunku: zdanie nadrzędne "mam komputer" i zdanie podrzędne "umie mówić". Słowo "który" określa, że "umie mówić" dotyczy komputera, i przez to mówi nam, że jest to zdanie podrzędne przydawkowe.

Warunki:

- Każde zdanie może być dla innego nadrzędne, podrzędne, współrzędne,
- Np. 1 2 3 4 5 (zdania).

Możliwe struktury:

1. 1+2+3+4
2. 1=2+3+4
3. 1-2+3+4
4. 1+2=3+4
5. 1=2=3+4
6. 1-2=3+4
7. 1+2-3+4
8. 1=2-3+4
9. 1-2-3+4
10. 1+2+3=4
11. 1=2+3=4
12. 1-2+3=4
13. 1+2=3=4
14. 1=2=3=4
15. 1-2=3=4
16. 1+2-3=4
17. 1=2-3=4
18. 1-2-3=4
19. 1+2+3-4
20. 1=2+3-4
21. 1-2+3-4
22. 1+2=3-4
23. 1=2=3-4
24. 1-2=3-4
25. 1+2-3-4
26. 1=2-3-4
27. 1-2-3-4

Objaśnienia:

- + – połączenie zdania nadrzędnego z podrzędnym
- – połączenie zdania podrzędnego z nadrzędnym
- = – połączenie zdań współrzędnych

Liczbę możliwych struktur dla N zdań można wyznaczyć ze wzoru:

$$3^{(N-1)}$$

$$3^{(4-1)} = 27$$

Algorytm:

- Wstępnie każde kolejne zdanie jest podrzędnym dla wcześniejszego,
- Niech X będzie numerem zdania, i przyjmuje kolejno wartości 1, 2...N-1
- Jeśli zdanie nr X ma zdanie nr $X+1$ jako:
 - Podrzędne, to czynimy je współrzędnym i zwracamy wynik
 - Współrzędne, to czynimy je nadrzędnym i zwracamy wynik
 - Nadrzędne, to czynimy je podrzędnym.
- Powtarzamy operacje zwiększając X .

4. Współzależność podziału na zdania i struktury zdań

LICZBA WYRAZÓW	5	6	7	8	9
LICZBA MOŻLIWOŚCI	337	1 360	5 455	21 838	87 373
CZAS PRACY [ms]	2	8	29	111	432

LICZBA WYRAZÓW	10	11	12	13
LICZBA MOŻLIWOŚCI	349 516	1 398 091	5 592 394	2 2369 609
CZAS PRACY [s]	1.696	6.790	28.719	107.730

Powyższe czasy pracy dotyczą programu o konstrukcji dalece odbiegającej od zaimplementowanej w wersji 5.0 Denise[®]. Zostały one zmierzone na Amidze z procesorem 68040 25MHz (ok. 20 MIPSów). Na komputerach z procesorami PowerPC czasy te powinny być około 100-200 razy krótsze, umożliwiając analizę zdań złożonych z więcej niż 13 wyrazów, bez wykorzystywania funkcji katalizujących ten przebieg, zmniejszających liczbę możliwości (wersja 4.2 zmniejszała tę liczbę (dla 7 wyrazów) z 850 389 do 21 262, a więc ok. 40 razy, przy 13-to krotnym skróceniu czasu analizy).

5. Podział zdania na wyrażenia (grupy)

Pojęcie GRUPY zostało wprowadzone przeze mnie bardzo dawno temu. Określało ono to, co w najbliższym sensie opisuje się poprzez nazwę WYRAŻENIA. Sformułowanie WYRAŻENIA zostaje wprowadzone dopiero teraz, toteż z przyzwyczajenia i ze względów historycznych przypisuję sobie prawo do używania obu tych słów dla wskazywania tego samego.

Wyrażenie (grupa) to jednostka funkcyjna, która może składać się z jednego lub wielu wyrazów połączonych ze sobą i stanowiących funkcjonalną całość. Definicja ta nie jest (o ile wiem) obowiązującą w językoznawstwie, jednakże dla naszych potrzeb wystarczy. Wyrażeniem są więc tak pojedyncze wyrazy (*dom, pies, komputer, trzymać, za*), jak również połączenia wielu wyrazów (*w domu, Jaś i Małgosia, Jaś Fasola*).

Dla tych, którzy jeszcze nie rozumieją – pojęcie wyrażenia jest niezbędne dla analizy składniowej, gdyż to GRUPY pełnią funkcje w zdaniu, a nie pojedyncze wyrazy! Np. "Jaś i Małgosia zjedli za dużo czekolady" – w zdaniu tym podmiotem jest złożenie "Jaś i Małgosia", a więc aż TRZY wyrazy.

Np. dla wyrazów: A B C D możliwe są grupy (podawane są numery grup, do których przypisane są kolejne wyrazy):

	A	B	C	D	
1.	1	1	1	1	ABCD
2.	1	1	1	2	ABC,D
3.	1	1	2	1	ABD,C
4.	1	1	2	2	AB,CD
5.	1	1	2	3	AB,C,D
6.	1	2	1	1	ACD,B
7.	1	2	1	2	AC,BD
8.	1	2	1	3	AC,B,D
9.	1	2	2	1	AD,BC
10.	1	2	2	2	A,BCD
11.	1	2	2	3	A,BC,D
12.	1	2	3	1	AD,B,C
13.	1	2	3	2	A,BD,C
14.	1	2	3	3	A,B,CD
15.	1	2	3	4	A,B,C,D

Liczba możliwych do utworzenia kombinacji grup:

- 1 → 1
- 2 → 2
- 3 → 5
- 4 → 15
- 5 → 52
- 6 → 203
- 7 → 877
- 8 → 4 140
- 9 → 21 147
- 10 → 115 975
- 11 → 678 570
- 12 → 4 213 597
- 13 → 27 644 437

Wzór na liczbę możliwości wiąże się z tzw. liczbami Stirlinga drugiego rodzaju i liczbami Bella. Liczba Bella dla zbioru n -elementowego (B_n) jest szukaną przez nas liczbą wszystkich podziałów zbioru n -elementowego.

$$B_0 = 1$$

$$B_1 = 1$$

$$B_{(n+1)} = 1 \text{ suma od } i = 0 \text{ do } n \text{ iloczynów } B_i \text{ oraz kombinacji } n \text{ po } i$$

Ostateczny wzór:

$$B_{n+1} = \sum_{i=0}^n \binom{n}{i} B_1$$

Z obliczeń wynika ile jest możliwości dla ponad 13 grup:

n	B_n
14	190.899.322
15	1.382.958.545
16	10.480.142.147
17	82.864.869.804
18	682.076.806.159
19	5.832.742.205.057
20	51.724.158.235.372

Informacje na temat wzoru zostały znalezione przez Piotra Radeja, zajmującego się optymalizacją modułu składni.

Algorytm 1:

- Wstępnie każdy wyraz jest przypisany do grupy nr 1,
- Niech X będzie numerem wyrazu, i przyjmuje kolejno wartości $N, N-1, \dots, 1$
- Jeśli wyraz nr X przypisany jest grupie o najwyższym numerze nie równym innym wyrazom, to:
 - Jeśli numer tej grupy jest równy liczbie wyrazów N , to KONIEC MOŻLIWOŚCI,
 - Przypisz mu grupę numer 1,
 - Kontynuuj wykonywanie pętli dla $X = X-1$,
 - Jeśli nie jest, to:
 - Przypisz wyraz grupie o numerze jeden większym,
 - Przerwij wykonywanie pętli.

Algorytm 2:

- Wstępnie każdy wyraz jest przypisany do grupy równej numerowi wyrazu, (liczba grup równa jest liczbie wyrazów)
- Niech X będzie numerem wyrazu, i przyjmuje kolejno wartości $N, N-1, \dots, 1$
- Jeśli numer grupy wyrazu X jest większy od 1 to:
 - Jeśli liczba X jest równa numerowi grupy to zmniejszamy o 1 liczbę grup,
 - Zmniejszamy grupę wyrazu X o 1
 - Ustalamy wartości kolejnych grup wyrazów:
 - Zaczynając od wartości równej liczbie grup zwiększonej o jeden,
 - Dla każdego następnego wyrazu wpisujemy tę wartość powiększając ją o 1,
 - Zwracamy kolejną możliwość,
 - Jeśli nie to:
 - Powtarzamy pętlę.

Czas potrzeby dla znalezienia wszystkich możliwości w porównaniu z sumarycznym czasem (patrz tabela wcześniej) dzielenia wypowiedzi, jest od niego ok. dwa razy krótszy dla liczby wyrazów do ok. 10, po czym oba czasy zrównują się i przy ok. 13 wyrazach dzielenie wypowiedzi staje się szybsze od tworzenia grup.

6. Ustalenie wyrazów fleksyjnych

Wyrazem fleksyjnym nazywa się konstrukcje wyrazów stanowiących praktycznie fleksyjną całość. W języku polskim będą to np. stopnie wyższe w stopniowaniu opisowym (np. "bardzo zielony"), a także formy zwrotne (np. "myć się") oraz inne, składające się tak z jednego jak wielu pojedynczych wyrazów. Do poprawnej analizy uwzględnić należy również zrosty wyrazów, jak np. "nienaturalny", gdzie właściwie występuje zrost wyrazów "nie" i "naturalny".

Ustalenie wyrazów fleksyjnych polega na znajdowaniu kolejnych możliwości interpretacji wyrażenia i formułowania wyrazów fleksyjnych. W tym celu wykorzystywana jest baza fleksji pozwalająca stwierdzić, czy dany wyraz fleksyjny jest poprawny w sensie, że istnieją w bazie dane na jego temat. Istnieje bardzo silne powiązanie ustalania wyrazów fleksyjnych z wybieraniem fleksji, czyli z punktem następnym.

Proces ustalania wyrazów fleksyjnych jest mikro-odbiciem analizy zdania, i jest równie skomplikowany. Z tego jednak względu, że dotyczy raczej analizy wyrazu, jego liter, i jest ściśle powiązany z oceną gramatyczną i słownikiem fleksji, nie zostanie dokładniej opisany w tym miejscu. Jeśli znajdzie taka potrzeba, odpowiednie opisy znajdują się w osobnym artykule, poświęconym wyłącznie temu zagadnieniu.

7. Wybranie fleksji wyrazów fleksyjnych

W wybieraniu fleksji wyrazów fleksyjnych (całościowych), chodzi o odnalezienie wyrazu w bazie fleksji, i identyfikacji jego cech odmiany. Jako, że może istnieć wiele różnych opisów odmian, które "wyglądają" tak samo, wymagane jest sprawdzenie ich wszystkich. Rozszerzając to wszystkie wyrazy całościowe w zdaniu, otrzymujemy skończony zbiór wszystkich możliwych opisów odmian. W ujęciu programowym, algorytm opisuje wielopozycyjny licznik, cechujący się niezależną od pozostałych pozycji liczbą stanów pozycji.

Przypominam, że za wyraz całościowy uważa się konstrukcję utworzoną jak opisano w punkcie poprzednim.

8. Struktura grup

Struktura grup, wyrażeń odpowiada "drzewku" budowanemu w szkole podstawowej. Przypomnę, że drzewko to obrazowało, w jakiej zależności w stosunku do siebie występują wyrażenia użyte w zdaniu. Strzałkami wskazywano, który człon, któremu jest podrzędny, a przy strzałkach stawiano pytania ułatwiające (pozornie) określenie, jaką funkcję pełni grupa.

Stosowany tu algorytm strukturyzacji grup tworzy właśnie wszystkie możliwe takie drzewka, dla danej liczby grup, które są grupami między sobą rozróżnialnymi.

Np. dla grup ponumerowanych kolejno: 1, 2, 3, możliwe są struktury:

1.	0	1	1
2.	0	1	2
3.	0	3	1
4.	2	0	1
5.	2	0	2
6.	3	0	2
7.	2	3	0
8.	3	1	0
9.	3	3	0

1 | 2 | 3 – kolumny dotyczą poszczególnych grup.

Gdzie podane numery oznaczają numer grupy NADRZĘDNEJ dla danej grupy (dana grupa jest określona kolumną). Liczba 0 oznacza, że grupa nie ma grupy nadrzędnej, czyli, że jest nadrzędna dla wszystkich pozostałych grup.

Liczba możliwych do utworzenia kombinacji:

1 → 1
 2 → 2
 3 → 9
 4 → 64
 5 → 625
 6 → 7 776
 7 → 117 649 (6 556s)
 8 → ? (patrz – wzór)

Liczbę możliwych struktur dla N grup można wyznaczyć ze wzoru:

$$N(N-1)$$

$$5^{(5-1)} = 5^4 = 625$$

$$5^{(8-1)} = 5^7 = 2\,097\,152$$

Algorytm:

- Wstępnie ustalamy grupy nadrzędne tak, że grupa pierwsza nie ma grupy nadrzędnej, a wszystkie pozostałe mają za grupę nadrzędną grupę pierwszą,
- Niech X będzie numerem grupy, i przyjmuje kolejno wartości N, N-1...1
- Jeśli grupa nie ma grupy nadrzędnej to pomijamy,
- Sprawdzamy czy możemy zwiększyć numer grupy nadrzędnej dla grupy X (niech Y będzie zwiększonym numerem grupy nadrzędnej dla grupy X):
 - Jeśli grupa o numerze Y jest wcześniejsza od X i ma za nadrzędną grupę o numerze X, to nie możemy zwiększyć

- (grupa nie może być jednocześnie podrzędna i nadrzędna dla jednej grupy).
- Jeśli numer grupy X jest równy Y to nie możemy zwiększyć (grupa nie może być podrzędna samej sobie).
 - Jeśli liczba Y jest większa niż liczba grup, to nie możemy zwiększyć.
 - Jeśli możemy zwiększyć numer grupy, to:
 - Zwiększamy.
 - Jeśli grupa nie była ostatnią, to zmieniamy też następujące po niej,
 - Sprawdzamy również, czy istnieje "droga powrotna", tj. czy od grupy.
 - Nadrzędnej dla X dojdziemy do grupy bez grupy nadrzędnej, jeśli tak, to:
 - Zwracamy nową MOŻLIWOŚĆ,
 - Jeśli nie możemy zwiększyć numeru grupy, to kontynuujemy pętlę,
 - Jeśli po wyjściu z pętli nie udało się znaleźć nowej możliwości, to:
 - Jeśli grupa nadrzędna do wszystkich jest ostatnią, to KONIEC MOŻLIWOŚCI,
- Przepisujemy o jedno miejsce dalej grupę nadrzędną dla wszystkich,
 - Ustalamy odpowiednie numery wszystkich pozostałych grup.

9. Funkcje grup

Jest to już ostatni podział, ostatni etap tworzenia możliwości rozumienia zdania. Określone są tu funkcje pełnione przez grupy. Funkcje te są te same dla znanych mi języków, toteż mogą stanowić podstawę do ROZUMNEGO tłumaczenia zdań.

Podstawowymi funkcjami, które są powszechnie znane są podmiot (SUBJECT) i orzeczenie (PREDICATE → zwróć uwagę na LOGIKĘ PREDYKATÓW). Tworzą one w zdaniu związek główny, jednakże o gramatyce znajduje się więcej informacji w poświęconym temu zagadnieniu artykule.

Np. dla 3 grup, przy założeniu 3 możliwych funkcji, istnieją możliwości:

1.	1	1	1
2.	1	1	2
3.	1	1	3
4.	1	2	1
5.	1	2	2
6.	1	2	3
7.	1	3	1
8.	1	3	2
9.	1	3	3
10.	2	1	1
11.	2	1	2
12.	2	1	3
13.	2	2	1

14.	2	2	2
15.	2	2	3
16.	2	3	1
17.	2	3	2
18.	2	3	3
19.	3	1	1
20.	3	1	2
21.	3	1	3
22.	3	2	1
23.	3	2	2
24.	3	2	3
25.	3	3	1
26.	3	3	2
27.	3	3	3

Gdzie podane numery odpowiadają funkcji grupy o numerze odpowiadającym pozycji.

Liczbę możliwości można określić zgodnie ze wzorem:

$$X^Y$$

Gdzie: X – liczba możliwych funkcji, Y – liczba grup.

Algorytm jest oczywisty, i nie zostanie tu opisany. W programie zastosowano również pewną optymalizację kodu, który jest dostępny osobom zainteresowanym.

10. Sumaryczny wynik działania algorytmów rozbioru zdania

Poniżej przedstawiam tabelaryczne zestawienie danych testowych, powstałych po połączeniu opisanych wyżej algorytmów (pomijając ustalanie wyrazów całościowych i wybieranie fleksji). Dane te są jedynie orientacyjnymi, i nie mogą stanowić oparcia dla dalszych, praktycznych obliczeń, gdyż rzeczywisty algorytm modułu składni jest daleko bardziej skomplikowany i wykorzystuje wiele narzędzi pomocniczych, w tym również pochodzących od innych modułów. Warto jednak zwrócić uwagę na wzrost skomplikowania obliczeń dla dodania zaledwie jednego wyrazu.

LICZBA WYRAZÓW	5	6
LICZBA WYPOWIEDZI	337	1 360
LICZBA ZDAŃ	337 548	12 699 282
CZAS PRACY [ms]	42.849 (dla 040/50MHz)	29.379 (dla 060/50MHz)

UWAGA! Powyższe testy zostały przeprowadzone z uwzględnieniem jedynie 3 funkcji, jakie grupy mogą pełnić w zdaniu. Liczba funkcji w chwili obecnej nie jest ostatecznie ustalona, lecz może sięgać nawet 10. Ostateczne, realne parametry mogą zostać podane dopiero po ostatecznym zakończeniu prac nad analizą gramatyczną wypowiedzi.

Podane czasy dotyczą Amigi z procesorem 68040/25MHz, przy czym z pomocą innych osób dokonano również dodatkowych pomiarów, których wyniki i parametry sprzętu, na którym zostały przeprowadzane zamieszczono w tabeli. Należy uwzględnić, że w każdej z tych konfiguracji uruchomione było jednocześnie dodatkowe oprogramowanie, wydłużające nieco czas pracy testowanego programu.

11. Specyfikacje algorytmu – katalizatory

Algorytm analizy składniowej, działający na powszechnie dostępnych komputerach domowych mógłby być stosowany tylko do zdań składających się z zaledwie kilku (5, 6) wyrazów. Jest to spowodowane liczbą koniecznych obliczeń oraz powstałych do oceny możliwości rozbiórów. Uniemożliwiłoby to praktyczne wykorzystanie tej metody w ciągu następnych kilku lat, do czasu upowszechnienia się znacznie szybszych komputerów.

Katalizatory umożliwiają przyśpieszenie działania algorytmu. Dzięki nim możliwe jest korzystanie z opisanej techniki bez znacznej zwłoki czasowej. Czasem może się to odbywać kosztem pewnej elastyczności języka, jednakże w masie przypadków umiejętne stosowanie katalizatorów daje nadszpiewanie dobre rezultaty.

Istnieje kilka typów katalizatorów, które można rozróżnić ze względu na przesłanki katalizacji:

- a) **Katalizatory wbudowane w algorytm**; pracują w oparciu o OGÓLNE zasady składniowe dotyczące wszystkich znanych mi języków; zostały zaimplementowane bezpośrednio w algorytm tak, że algorytm te zostały przez nie ZMODYFIKOWANE, przekształcone, i utraciły swoją niejako pierwotną "czystość" kombinatoryczną, dostosowując się do efektywnego spełniania swoich funkcji – analizy składniowej,
- b) **Katalizatory opcjonalne**; katalizatory te wprowadzają do algorytmu pewne uproszczenia, ograniczenia, których zadaniem jest eliminacja rozbiórów, które z dużym prawdopodobieństwem i tak skazane są na niskie oceny jakościowe; dodatkowo zawężają one możliwości analizy zdań, definiując niejako "okno", przez które tylko niektóre konstrukcje zostają przepuszczone; dzięki temu posiadacze wolnych komputerów mogą dostosować algorytm do swoich możliwości, jednocześnie będąc świadomymi narzuconych przez to ograniczeń rozumienia wprowadzanych przez nas zdań;
- c) **Opcjonalne katalizatory gramatyczne**; są to katalizatory specyficzne dla danego języka, a mimo to, ich funkcjonalność i właściwe umieszczenie w algorytmie jest całkowicie ogólne i obejmuje różne języki; katalizatory te w oparciu o parametry fleksyjne pozwalają na wczesnym etapie analizy ocenić, które możliwe analizy zdań są mało prawdopodobne, lub też zgoła niemożliwe; zaznaczyć jednak trzeba, że zbyt ścisła i rygorystyczna kontrola, może doprowadzić do sytuacji, kiedy zdanie z błędem gramatycznym nie zostanie poprawnie zrozumiane, kiedy bez użycia katalizatorów zdanie mogłoby zostać dobrze zinterpretowane.

Bardzo ważne jest właściwe rozumienie każdej z metod katalizacji, aby poprzez ślepe modyfikowanie parametrów pracy algorytmu nie doprowadzić do całkowitego "ogłupienia" programu i uniemożliwić mu poprawne rozumienie jakichkolwiek zdań.

Przed wszelkimi modyfikacjami w programie proponuję zapoznać się z poniższymi opisami poszczególnych metod, oraz specyficznych opcji. Pozwoli to uświadomić sobie ich funkcjonalność i zrozumieć ograniczenia, jakie nakładają niektóre z katalizatorów.

Katalizatory wbudowane w algorytm:

- Wyszczególnienie podmiotu, (p. 12)
- Wyszczególnienie orzeczenia, (p. 13)
- Uwzględnianie struktur grup w ustalaniu funkcji grupy, (p. 14,15)
- Aktywne ocenianie możliwości jakości analizy składniowej (p. 16)

Katalizatory opcjonalne: (p. 17)

- Min. liczba wyrazów w zdaniu: 1-9/OFF
- Min. liczba grup w zdaniu: 1-9/OFF
- Max. liczba grup w zdaniu: 1-9/OFF
- Max. liczba wyrazów w grupie: 1-9/OFF
- Max. liczba zdań pojedynczych w wypowiedzi: 1-9/OFF
- Założenie: zdanie pierwsze nigdy podrzędne,
- Założenie: podmiot stoi zawsze przed orzeczeniem,

Specyficzne dla języka opcjonalne katalizatory gramatyczne: (p. 18)

- Kontrola części mowy (wyszukiwanie orzeczeń) w podziale na zdania pojedyncze,
- Kontrola związków pomiędzy wyrazami w grupie,
- Kontrola podmiotu jako grupy nie podrzędnej,
- Kontrola związków między grupami w strukturze grup,
- Kontrola zgodności części mowy z funkcją na etapie określania funkcji.

12. Katalizatory algorytmu – wyszczególnienie podmiotu

Wyszczególnienie podmiotu polega na założeniu, że w zdaniu musi wystąpić dokładnie jedno wyrażenie pełniące funkcję PODMIOTU (SUBJECT). Dzięki temu liczba kombinacji znacznie się zmniejsza.

Np. dla 3 grup, przy założeniu 3 możliwych funkcji, istnieją możliwości:

1.	1	2	2
2.	1	2	3
3.	1	3	2
4.	1	3	3
5.	2	1	2
6.	2	1	3
7.	3	1	2
8.	3	1	3
9.	2	2	1
10.	2	3	1
11.	3	2	1
12.	3	3	1

(Wcześniej było 27 możliwości.) Gdzie podane numery odpowiadają funkcji grupy o numerze odpowiadającym pozycji.

Liczbę możliwości można określić zgodnie ze wzorem:

$$\text{ILE_GRUP}[(\text{ILE_FUNKCJI}-1) \times (\text{ILE_GRUP}-1)]$$

Algorytm:

- Wstępnie ustalamy grupę pierwszą z funkcją 1 (PODMIOT), a pozostałe grupy z 1
 - Niech X będzie numerem grupy, i przyjmuje kolejno wartości N, N-1, ..., 1
 - Jeśli grupa X ma funkcję 1 to pomijamy tę grupę
 - Jeśli grupa pełni ostatnią z możliwych funkcji, to:
 - Ustawiamy funkcję tej grupy na 2,
 - Powtarzamy dla kolejnego X,
 - Jeśli grupa nie pełni ostatniej z możliwych funkcji, to:
 - Zwiększamy funkcję tej grupy,
 - Zwracamy kolejną możliwość <return>
- <po zakończeniu pętli liczenia X>
- Jeśli ostatnia grupa ma funkcję 1, to KONIEC MOŻLIWOŚCI
- W przeciwnym razie:
 - Ustawiamy funkcję grupy z funkcją 1 na 2,
 - Ustawiamy funkcję grupy następującej po grupie z funkcją 1 na 1,

Poniższe zestawienie może służyć do porównania pracy wyspecjalizowanego algorytmu.

LICZBA WYRAZÓW	5	6
LICZBA WYPOWIEDZI	337	1 360
LICZBA ZDAŃ	125 696	3 956 995
CZAS PRACY [s]	1.539	41.117

Testy przeprowadzono w warunkach identycznych jak w punkcie 8.

13. Katalizatory algorytmu – wyszczególnienie orzeczenia

Katalizator analogiczny do opisanego wyżej, dotyczy jednak ORZECZENIA (PREDICATE).

Np. dla 3 grup, przy założeniu 3 możliwych funkcji, istnieją możliwości:

1.	1	2	3
2.	1	3	2
3.	2	1	3
4.	3	1	2
5.	2	3	1
6.	3	2	1

Gdzie podane numery odpowiadają funkcji grupy o numerze odpowiadający pozycji.

Warunek: muszą być min. 2 grupy i min. 3 funkcje,

Algorytm:

- Wstępnie ustalamy grupę pierwszą z funkcją 1 (PODMIOT), drugą z funkcją 2 (ORZECZENIE), pozostałe na funkcję numer 3,
 - Niech X będzie numerem grupy, i przyjmuje kolejno wartości N,N-1...1
 - Jeśli grupa X ma funkcję 1 lub 2 to pomijamy tę grupę, w przeciwnym razie dalej:
 - Jeśli grupa pełni ostatnią z możliwych funkcji, to:
 - Ustawiamy funkcję tej grupy na 3,
 - Powtarzamy dla kolejnego X,
 - Jeśli grupa nie pełni ostatniej z możliwych funkcji, to:
 - Zwiększamy funkcję tej grupy,
 - Zwracamy kolejną możliwość <return>
- <po zakończeniu pętli liczenia X>
- Jeśli ostatnia grupa ma funkcję 1, a grupa wcześniejsza funkcję 2 to KONIEC MOŻLIWOŚCI (w przeciwnym razie dalej:)
 - Jeśli ostatnia grupa ma funkcje 2, to:
 - Ustawiamy funkcję ostatniej grupy na 3,
 - Ustawiamy funkcję grupy z funkcją 1 na 3,
 - Ustawiamy funkcję grupy następującej po grupie z funkcją 1 na 1,
 - Ustawiamy funkcję pierwszej grupy na 2,
 - Zwracamy kolejną możliwość <return>
 - W przeciwnym razie:
 - Ustawiamy funkcję grupy z funkcją 2 na 3,
 - Ustawiamy funkcję grupy następującej po grupie z funkcją 2 na 2,
 - Zwracamy kolejną możliwość <return>

Wynik działania programu z uwzględnieniem opisanej specyfikacji:

LICZBA WYRAZÓW	5	6
LICZBA WYPOWIEDZI	337	1 360
LICZBA ZDAŃ	43 310	968 560
CZAS PRACY [s]	1.082	24.752

14. Katalizatory algorytmu – uwzględnienie struktury grup w ustalaniu funkcji grupy (I)

Katalizator ten pogłębia założenia opierane o prawa rządzące strukturą zdania. Postulaty te dotyczą faktu, że PODMIOT (SUBJECT) jest jedyną grupą, mogącą występować bez grupy nadrzędnej, oraz, że tylko wyrażenie będące podrzędnym dla PODMIOTU, może mieć funkcję ORZECZENIA (PREDICATE). W przykładzie poniżej funkcja podmiotu nosi numer 1, a orzeczenia 2.

Np. dla 3 grup, przy założeniu 3 możliwych funkcji, oraz uwzględnieniu następującej struktury grup 0 | 1 | 1 (patrz punkt 6), istnieją możliwości:

1.	1	2	3
2.	1	3	2

Gdzie podane numery odpowiadają funkcji grupy o numerze odpowiadający pozycji.

Warunek: muszą być min. 2 grupy i min. 3 funkcje,

Algorytm:

- Wstępnie ustalamy funkcję grupy bez grupy nadrzędnej na 1 (PODMIOT), a funkcję pierwszej z grup z grupą nadrzędną bez grupy nadrzędnej (PODMIOT) na 2 (ORZECZENIE), pozostałe grupy na funkcję numer 3,
 - Niech X będzie numerem grupy, i przyjmuje kolejno wartości N,N-1...1
 - Jeśli grupa X ma funkcję 1 lub 2 to pomijamy tę grupę, w przeciwnym razie dalej;
 - Jeśli grupa pełni ostatnią z możliwych funkcji, to:
 - Ustawiamy funkcję tej grupy na 3,
 - Powtarzamy dla kolejnego X,
 - Jeśli grupa nie pełni ostatniej z możliwych funkcji, to:
 - Zwiększamy funkcję tej grupy,
 - Zwracamy kolejną możliwość <return>
- <po zakończeniu pętli liczenia X>
- Niech X będzie numerem grupy, i przyjmuje kolejno wartości M+1...N, gdzie M jest aktualnym numerem grupy z funkcją 2,
 - Jeśli grupa X ma grupę nadrzędną z funkcją 1, to:
 - Ustawiamy funkcję grupy M na 3,
 - Ustawiamy funkcję grupy X na 2,
 - Zwracamy kolejną możliwość <return>

<po zakończeniu pętli liczenia X>

KONIEC MOŻLIWOŚCI

Wynik działania programu z uwzględnieniem opisanej specyfikacji:

LICZBA WYRAZÓW	5	6	7
LICZBA WYPOWIEDZI	337	1 360	5 455
LICZBA ZDAŃ	6 580	97 154	1 681 820
CZAS PRACY [s]	0.337	9.343	160.138

15. Katalizatory algorytmu – uwzględnienie struktury grup w ustalaniu funkcji grupy (II)

Jest to kolejny katalizator opierający się ściśle o reguły składniowe, przestrzegane we wszystkich znanych mi językach. Okazuje się bowiem, że wszystkie one mają niejako "wspólną naturę".

Zakładamy 5 możliwych funkcji w zdaniu:

- 1 – podmiot [*subject*]
- 2 – orzeczenie [*predicate*]
- 3 – przydawka [*modifier*]
- 4 – dopełnienie [*object*]
- 5 – okolicznik [*adverbial*]

Prawdziwe są wtedy reguły:

- Przydawka może być podrzędna do: podmiotu, przydawki, dopełnienia, okolicznika,
- Dopełnienie może być podrzędne do: orzeczenia, przydawki,
- Okolicznik może być podrzędny do: orzeczenia, przydawki.

Wynik działania programu przy założeniu 5 funkcji w zdaniu:

LICZBA WYRAZÓW	5	6	7
LICZBA ZDAŃ	37 230	1 534 814	75 942 300
CZAS PRACY [s]	0.497	17.386	788.746

Wynik działania programu po uwzględnieniu opisanej katalizacji w ustalaniu funkcji:

LICZBA WYRAZÓW	5	6	7
LICZBA ZDAŃ	6 268	166 730	429 926
CZAS PRACY [s]	0.871	34.762	5 519.878

Z tabel widać, że katalizator co prawda znacznie (dla 7 wyrazów: 176.6 razy!) zmniejsza liczbę kombinacji, zwiększa jednak (dla 7 wyrazów: 7 razy) czas potrzebny na ich wygenerowanie. Pamiętać w tym miejscu trzeba, że ostateczny czas analizy jest w głównej mierze uzależniony od czasu oceny gramatycznej; im więc mniej będzie ocen, tym całkowity czas będzie krótszy. Na marginesie informuję, że algorytmy oceny gramatycznej zostaną również wyposażone w elementy uwzględniające aktywne katalizatory, co znaczy, że ocena gramatyczna zostanie SKRÓCONA o te części, które niejako podchodzą pod działanie katalizatorów.

16. Katalizatory algorytmu – aktywne ocenianie jakości analizy składniowej

Jakość analizy składniowej jest sumą jakości cząstkowych poszczególnych zdań pojedynczych. Przy czym ocenianie odbywa się na zasadzie naliczania punktów karnych. Jeśli w trakcie analizy nie ostatniego zdania suma ocen cząstkowych (punktów karnych) dotychczas przeanalizowanych zdań pojedynczych jest gorsza (naliczono więcej punktów karnych) niż aktualnie przechowywana w pamięci – analiza najlepsza, można zrezygnować z dalszego oceniania kolejnych zdań pojedynczych tej możliwości rozbioru.

Prezentacja wyników otrzymanych dzięki tej katalizacji nastąpić może jedynie po skonstruowaniu odpowiedniego modułu gramatycznego oceny jakości zdania pojedynczego. Tematyka ta zostanie prawdopodobnie opisana w artykule dotyczącym gramatyki.

17. Opcjonalne katalizatory algorytmu

Katalizatory opcjonalne w wyraźny sposób OGRANICZAJĄ możliwości algorytmu. Jednakże, rozsądne ich stosowanie, czyli uwzględnianie ich ustawień w trakcie prowadzenia rozmowy, może przyspieszyć analizę wypowiedzi w zauważalnym stopniu. Ponadto, pewne ograniczenia algorytmu mogą być stosowane w gruncie rzeczy jako domyślne, gdyż język pisany zwykle jest prostszy od mówionego, i składa się w większości ze zdań pojedynczych, maksymalnie dwóch. Podobne ograniczenia spotykane są w używanych wyrażeniach, które zwykle składają się z góra 2-3 wyrazów. Podobnie ma się rzecz ze strukturą zdań – zdanie pierwsze zwykle jest jednocześnie nadrzędnym/współrzednym, a nie podrzędnym. Pewna konwencja dotyczy również stawiania podmiotu przed orzeczeniem.

Katalizatory opcjonalne: (p. 15)

- Min. liczba wyrazów w zdaniu: 1-9/OFF
- Min. liczba grup w zdaniu: 1-9/OFF
- Max. liczba grup w zdaniu: 1-9/OFF
- Max. liczba wyrazów w grupie: 1-9/OFF
- Max. liczba zdań pojedynczych w wypowiedzi: 1-9/OFF
- Założenie: zdanie pierwsze nigdy podrzędne,
- Założenie: podmiot stoi zawsze przed orzeczeniem.

18. Katalizatory gramatyczne

Tak jak katalizatory algorytmu dotyczą samej składni, bez uwzględniania gramatyki i fleksji, i działają jakby w ciemno, kierując się odgórnymi założeniami. W odróżnieniu od nich, katalizatory gramatyczne opierają się ściśle na formach fleksyjnych poszczególnych wyrazów, ich dziedziną jest więc GRAMATYKA. Z tego też względu katalizatory te, wyodrębnione jedynie w module składni, dopiero przy wykorzystaniu narzędzi modułu gramatycznego stają się rzeczywiście funkcjonalne. Dlatego też informacje na ich temat zostaną zawarte w innej części dokumentacji. Poniżej została jedynie powtórzona wstępna lista katalizatorów gramatycznych, co może pozwolić zorientować się, co do pola ich działania.

Specyficzne dla języka opcjonalne katalizatory gramatyczne: (p. 15)

- Kontrola części mowy (wyszukiwanie orzeczeń) w podziale na zdania pojedyncze;
- Kontrola związków pomiędzy wyrazami w grupie;
- Kontrola podmiotu jako grupy nie podrzędnej;
- Kontrola związków między grupami w strukturze grup;
- Kontrola zgodności części mowy z funkcją na etapie określania funkcji.

1. Charakterystyka ogólna modułu

Moduł gramatyki (czy raczej gramatyk, jeśli uwzględnić wiele języków), odpowiada za ocenę utworzonej przez moduł składni struktury zdania. Ocena ta opiera się o zasady gramatyczne, jakie rządzą danym językiem. W przypadku wykorzystania wielu języków słuszny staje się podział modułu na część ogólną oraz moduły specyficzne dla danych języków. Część ogólna stanowi interfejs pomiędzy gramatyką "jako taką", czyli odpowiednimi ocenami gramatycznymi, a specyficznymi funkcjami, które te oceny realizują zależnie od rozpoznanego języka. W części ogólnej znaleźć się również mogą funkcje globalne (zwykle będzie to rodzaj funkcji pomocniczych), mające jednakową realizację w używanych językach.

Każdy moduł częściowy, odpowiadający określonemu językowi, zbudowany jest w zasadzie według jednego schematu, na który składają się funkcje: oceny wyrazów, oceny wyrażeń, funkcji wyrażeń, związków i szyku zdania. Dalsze szczegóły techniczne w chwili obecnej nie są znane na etapie projektowania.

Ocena gramatyczna polega na procentowym określeniu poprawności w danym zakresie kontroli. Katalizatory gramatyczne wynikają z możliwości ustalenia różnych dolnych granic jakości, które w czasie rzeczywistym są w stanie eliminować dalsze poszukiwania, z góry skazane na niepowodzenie. Jednocześnie stanowią rzeczywiste ryzyko ostatecznego niepowodzenia. Jednakże, stosowane rozumnie stanowią nieocenioną pomoc w przyśpieszeniu analizy.

Tryb oceniania jest następujący. Wstępnie przyjmujemy ocenę najlepszą, czyli 100% poprawność. W dalszej części sprawdzane są kolejne warunki i ewentualna ocena może zostać obniżona. Ocenianie odbywa się zwykle na zasadzie rozpoznania struktury, i oceny parametrów, które za sobą pociąga (np. odpowiednie przypadki).

Ocena wyrazów – obejmuje kontrolę zrostów, szczegółową analizę każdego wyrazu z osobna.

Ocena wyrażeń – obejmuje kontrolę jakości utworzonych wyrażeń (grup). Badane są wyrazy połączone w wyrażenie, pod kątem poprawności takiego połączenia.

Ocena funkcji wyrażeń – dotyczy sprawdzenia, czy dane wyrażenie ma prawo pełnić przypisaną mu funkcję.

Ocena związków między wyrażeniami – zgodnie z nazwą, kontrola obejmuje sprawdzenie czy dopuszczalne są proponowane przez składnię związki między określonymi wyrażeniami.

Ocena szyku zdania – na tym etapie oceniana jest kolejność występowania wyrażeń w zdaniu. Czasem ma to decydujące znaczenie do poprawnego zrozumienia tekstu, szczególnie np. w języku angielskim. [UWAGA! ocena szyku zdania została włączona w ocenę związków między wyrażeniami!]

2. Ocena gramatyczna – JĘZYK POLSKI

W tym miejscu chcę podziękować Panu dr. hab. Andrzejowi Dyszakowi, za wiele profesjonalnych uwag i pomoc w tworzeniu poniższego opisu gramatycznego. Bez tej pomocy ocena gramatyczna byłaby na dużo niższym poziomie. Raz jeszcze serdecznie Panu dziękuję.

Następnie przedstawiony jest opis poparty przykładami – nie są one jednak aktualne, i jeśli ktoś chciałby przestudiować stosowane w programie reguły, proszę sięgnąć po pliki z katalogu [languages](#), i ich opisu, który znajduje się w dalszej części dokumentacji (pliki językowe).

2.1. Ocena wyrazów

Ocena wyrazów to etap sprawdzania, czy dany wyraz nie jest w rzeczywistości zrostem dwóch wyrazów odrębnych. Jeśli tak jest, wyraz zostaje rozbity, i dalej traktowany jest w "specjalny" sposób. Dzięki temu możliwe stało się wreszcie bezproblemowe analizowanie zanegowanych partykułą "nie" rzeczowników i przymiotników.

Zapis formalny dla potrzeb programu wygląda następująco:

- Denise-MYTH[®] Artificial Intelligence (16.07.2001)
- Grammar Data: GRAMATYKA JĘZYKA POLSKIEGO – ZROSTY
- przeczenie "nie" łącznie z rzeczownikami i przymiotnikami
16-partykuła,2-przecząca+1-rzeczownik
16-partykuła,2-przecząca+7-przymiotnik
- przeczenie "nie" łącznie z imiesłowami mającymi z zaprzeczeniem znaczenie przymiotnika
16-partykuła,2-przecząca+3-imiesłów przymiotnikowy
16-partykuła,2-przecząca+4-imiesłów przymiotnikowy
- przeczenie "nie" łącznie z przysłówkami mającymi odpowiednie podstawowe przymiotniki
16-partykuła,2-przecząca+8-przysłówek

2.2. Ocena wyrażień (grup)

* Do ustalenia.

Elementy wymagające dodatkowych ustaleń:

- Kontrola: specyfikacje liczebnika w połączeniu 2 wyrazów: czasownik osobowy, liczebnik;
- Dodać ściśle reguły w parze: liczebnik, liczebnik.

* Opis reguł.

Występują 2 etapy "sklejania" wyrazów grupy. Ostateczna jakość jest określana jako stosunek liczby wyrazów sklejonych do wszystkich wyrazów w grupie. Przy czym, jeśli występuje tylko jeden wyraz, to sprawdzane jest jedynie, czy należy on do grupy samodzielnych. Wspomniane etapy to: sklejanie wyrazów niesamodzielnych z samodzielnymi (partykuły, przymyki itd.), sklejanie gramatyczne wyrazów samodzielnych (polega na łączeniu tych wyrazów, które z punktu widzenia poprawności gramatycznej mogą być połączone - zwykle przy uwzględnieniu jakiś warunków jak np. w orzeczeniu imiennym; łączone wyrazy samodzielne mogą składać się dodatkowo z wyrazów niesamodzielnych, jak również mogą to być formy wieloczłonowe, przy czym uwzględniany jest wyraz samodzielny lub cechy wspólne grupy wyrazów samodzielnych).

W analizie całkowicie pomijane są WYKRZYKNIKI. Nie są one również brane pod uwagę do określania liczby wyrazów w grupie. Powinny one zostać automatycznie wyeliminowane ze zdania, a informacja przez nie przenoszona musi zostać odebrana innymi drogami aniżeli przez analizę składniową.

1) Etap 1 – sklejanie wyrazów niesamodzielnych.

Reguły:

- przyimek: Przyimek stoi zawsze przed wyrazem nadrzędnym (samodzielnym), którym może być:
 - rzeczownik (wyrażenie przyimkowe) "(siedzę) na krześle"
 - rzeczownik występuje w przypadku narzuconym przez przyimek,
 - przymiotnik (wyrażenie przyimkowe) "(świat dąży) ku lepszemu"
 - przymiotnik występuje w przypadku narzuconym przez przyimek,
 - liczebnik (wyrażenie przyimkowe) "(liczę) do trzech"
 - liczebnik występuje w przypadku narzuconym przez przyimek,
 - zaimek rzeczowny (wyrażenie przyimkowe) "(stoję) obok niego"
 - zaimek występuje w przypadku narzuconym przez przyimek,
 - zaimek przymiotny (wyrażenie przyimkowe) "(dążymy) ku temu"
 - zaimek występuje w przypadku narzuconym przez przyimek,
 - zaimek liczebnny (wyrażenie przyimkowe) "dla ilu (osób wystarczy miejsca)"
 - zaimek występuje w przypadku narzuconym przez przyimek,
 - partykuła:
 - partykuła zostaje przyłączona do najbliższego wyrazu samodzielnego po prawej, który spełnia poniższe kryteria:
 - rzeczownik
 - jeśli to partykuła - łącznik "to" (orzeczenie imienne) "(amiga) to komputer", to rzeczownik musi występować w przypadku mianownika,
 - czasownik osobowy "(każdy) niech pije (mleko)"
 - przymiotnik "(ktoś jest) niemły"
 - przysłówek "niedobrze (że pada dziś deszcz)"
 - liczebnik "(zliczył) aż milion (krów)"
 - imiesłów przymiotnikowy "(poduszka) dla śpiącego (kolegi)"
 - imiesłów przysłówkowy "(jeśli) nie dziś (to jutro)"
 - bezokolicznik "(lepiej) nie pić (trucizny)"
 - forma -no -to "oby zabito (wszystkie smoki zanim zjedzą księżniczki)"
 - zaimek rzeczowny "li tobie (kłaniać się nam?)"
 - zaimek przymiotny "czy twój (ten notes?)"
 - zaimek przysłowny "oby tak (zrobiono)"
 - zaimek liczebnny "(naliczył) aż kilkaset (krów)"

2) Etap 2 – sklejanie gramatyczne wyrazów (samodzielných).

Uwaga! Podczas sklejania gramatycznego ważna jest kolejność, tj. najpierw sklejane są SZEREGI wyrazów reprezentujące te same części mowy, w drugiej kolejności natomiast sklejane są wyrazy reprezentujące różne części mowy.

Uwaga! Po sklejeniu dwóch wyrazów o różnych częściach mowy, nie mogą one być już dalej sklejane z innymi.

Reguły:

- <szeregi wyrazów tej samej części mowy w stosunku współrzednym>
 Wszystkie wyrazy szeregu muszą stać zawsze bezpośrednio obok siebie (muszą być oddzielone spójnikiem lub przecinkiem), przy czym wyrazy niesamodzielne, które mogą występować między nimi zostają automatycznie włączone do szeregu; dotyczy to głównie spójników.
- rzeczowniki (wraz z zaimkami rzeczownymi) "(nie ma) Marka, psa, jej ani samochodu"
 - rzeczowniki, które nie wymagają przypadku narzucanego przez przyimek, muszą mieć ten sam przypadek,
 - przymiotniki "(wiatr) słaby i ciepły (wiatr)" – wszystkie przymiotniki muszą mieć ten sam przypadek,
 - przysłówki "(spotkamy się) dziś, jutro, pojutrze" – wszystkie przysłówki muszą być tego samego typu,
 - liczebniki "(przychodzę z) dwudziestoma jeden (problemami)"
 - imiesłowy przymiotnikowe "(wygląda jak) śpiący, chrapiący (niedźwiedź)"

- imiesłowy przysłówkowe "śpiewając i tańcząc (spędzili cały wieczór)"
- bezokoliczniki "(lubie) pić i jeść (ale z umiarem)"
- formy -no -to "wyłamano, skradziono (cenne drzwi zabytkowego kościoła)"
- zaimki rzeczowne "ja, on, ona (to trzy osoby liczby pojedynczej)" – wszystkie zaimki muszą mieć ten sam przypadek,
- <połączenia wyrazów różnych części mowy – konkretnie: orzeczenie złożone>
- Połączenia obejmują tylko jeden wyraz bezpośrednio sąsiadujący (chodzi o wyraz samodzielny, lub szereg takich wyrazów tworzony w poprzedniej części).
- czasownik osobowy, rzeczownik (orzeczenie słowno-imienne) "(Mariusz) został mnichem"
- poprawna kolejność: czasownik + rzeczownik,
- czasownik osobowy musi należeć do rodziny łączników w orzeczeniu imiennym,
- rzeczownik musi być w przypadku narzuconym przez łącznik (zwykle mianownik/narzędnik)
- czasownik osobowy, przymiotnik (orzeczenie słowno-imienne) "(Darek) był miły"
- poprawna kolejność: czasownik + przymiotnik,
- czasownik osobowy musi należeć do rodziny łączników w orzeczeniu imiennym,
- przymiotnik musi być zgodny z czasownikiem co do liczby,
- przymiotnik musi być zgodny z czasownikiem co do rodzaju,
- przymiotnik musi być w przypadku mianownika,
- czasownik osobowy, przysówek (orzeczenie słowno-przysłówkowe) "przyjemnie jest (spacerować)"
- poprawna kolejność: czasownik + przysówek lub przysówek + czasownik
- czasownik osobowy musi należeć do rodziny łączników w orzeczeniu imiennym,
- czasownik osobowy, liczebnik (orzeczenie słowno-imienne) "(Karol) był trzeci (na mecie)"
- poprawna kolejność: czasownik + przymiotnik,
- czasownik osobowy musi należeć do rodziny łączników w orzeczeniu imiennym,
- liczebnik musi być zgodny z czasownikiem co do liczby,
- liczebnik musi być zgodny z czasownikiem co do rodzaju,
- liczebnik musi być w mianowniku,
- czasownik osobowy, imiesłów przymiotnikowy (orzeczenie słowno-imienne) "(ona) była zachwycająca"
- poprawna kolejność: czasownik + imiesłów,
- czasownik osobowy musi należeć do rodziny łączników w orzeczeniu imiennym,
- imiesłów przymiotnikowy musi być zgodny z czasownikiem co do liczby,
- imiesłów przymiotnikowy musi być zgodny z czasownikiem co do rodzaju,
- imiesłów przymiotnikowy musi być w mianowniku,
- czasownik osobowy, zaimek rzeczowny (orzeczenie słowno-imienne) "(Mariusz) został nim"
- poprawna kolejność: czasownik + zaimek lub zaimek + czasownik,
- czasownik osobowy musi należeć do rodziny łączników w orzeczeniu imiennym,
- zaimek musi być w przypadku narzuconym przez łącznik (zwykle mianownik/narzędnik)
- czasownik osobowy, zaimek przymiotny (orzeczenie słowno-imienne) "(Darek) był taki"
- poprawna kolejność: czasownik + zaimek lub zaimek + czasownik,
- czasownik osobowy musi należeć do rodziny łączników w orzeczeniu imiennym,
- zaimek musi być zgodny z czasownikiem co do liczby,
- zaimek musi być zgodny z czasownikiem co do rodzaju,
- zaimek musi być w przypadku mianownika,

- czasownik osobowy, zaimek liczebny (orzeczenie słowno-imienne) "ilu jest (Karolów?)"
- poprawna kolejność: czasownik + zaimek lub zaimek + czasownik,
- czasownik osobowy musi należeć do rodziny łączników w orzeczeniu imiennym,
- zaimek musi być zgodny z czasownikiem co do liczby,
- zaimek musi być zgodny z czasownikiem co do rodzaju,
- zaimek musi być w mianowniku.

3) Ostateczna ocena.

Odnajdujemy człon główny, tj. pierwszą grupę sklejonych wyrazów, w której znajduje się przynajmniej jeden wyraz samodzielny, tj.:

- rzeczownik,
- czasownik osobowy,
- przymiotnik,
- przysłówek,
- liczebnik,
- imiesłów przymiotnikowy,
- imiesłów przysłówkowy,
- bezokolicznik,
- forma -no -to,
- zaimek rzeczowny,
- zaimek przymiotny,
- zaimek przysłowny,
- zaimek liczebny.

Następnie liczymy ile wyrazów znajduje się w tym sklejeniu. Dzielimy tę liczbę przez liczbę wszystkich wyrazów w grupie powiększoną o liczbę braków w sformułowaniach formy zwrotnej i stopniowaniu opisowym. Po pomnożeniu przez 100 uzyskujemy procentową jakość wyrażenia.

2.3. Ocena funkcji wyrażień (grup)

* Do ustalenia.

Elementy wymagające dodatkowych ustaleń:

- dokładniejsza charakterystyka orzeczenia (szczególnie imiennego i złożonego),
- wyszukanie innych konkretnych parametrów cechujących daną funkcję,

* Opis reguł.

PODMIOT

Niedopuszczalne, aby wyrażenie zawierało wyrazy o części mowy:

- czasownik osobowy,
- przysłówek,
- imiesłów przysłówkowy,
- forma -no -to,
- zaimek przysłowny,
- zaimek liczebny.

ORZECZENIE

Niedopuszczalne, aby wyrażenie zawierało wyrazy o części mowy:

- imiesłów przysłówkowy,
- zaimek przysłowny.

PRZYDAWKA

Niedopuszczalne, aby wyrażenie zawierało wyrazy o części mowy:

- czasownik osobowy,
- przysłówek,
- imiesłów przysłówkowy,
- forma -no -to,
- zaimek przysłowny.

DOPEŁNIENIE

Niedopuszczalne, aby wyrażenie zawierało wyrazy o części mowy:

- czasownik osobowy,
- przysłówek,
- imiesłów przysłówkowy,
- forma -no -to,
- zaimek przysłowny,
- zaimek liczebny,
- jeśli wyrażone jest rzeczownikiem lub zaimkiem rzeczownym:
- rzeczownik lub zaimek musi być w przypadku zależnym (inny niż mianownik i wołacz).

OKOLICZNIK

Niedopuszczalne, aby wyrażenie zawierało wyrazy o części mowy:

- czasownik osobowy,
- przymiotnik,
- imiesłów przymiotnikowy,
- forma -no -to,
- zaimek przymiotny,
- zaimek liczebny.

Ostateczna ocena polega na obliczeniu stosunku uzyskanych punktów karnych do maksymalnej możliwej liczby punktów karnych przewidzianej dla danej funkcji. Wynik tego dzielenia mnożymy przez 100 i odejmujemy od 100. Otrzymujemy ocenę jakości wyrażoną w procentach.

2.4. Ocena związków między wyrażeniami (grupami)

Na dzień dzisiejszy istnieje plik [language.gr7](#) zawierający opis poszczególnych połączeń między wyrazami. Składnia opisu jest dokładnie opisana w dokumentacji, która ma służyć tworzeniu plików gramatycznych. W tym miejscu wyjaśniam jedynie, że określone są możliwe związki między wyrażeniami. Podawane mogą być specyficzne parametry, takie jak: możliwe pełnione funkcje w zdaniu, wymagana kolejność w zdaniu, czy też wymagane lub zabronione występowanie wybranych części mowy w każdej z grup.

1. Charakterystyka ogólna modułu

Moduł fleksji pełni funkcję ośrodka informacyjnego, zawierającego dane o wszystkich wyrazach. Dane te dotyczą odmiany wyrazów. Wyrazem odmienianym jest tzw. WYRAZ PODSTAWOWY. Poniżej przedstawiono dokładną strukturę fleksji, oraz w punkcie drugim – jej model stosowany w programie. Podano również odpowiednie formy dla wyrazów podstawowych.

Format danych w bazie fleksji jest następujący:

```
wyraz_podstawowy
liczba_grup_odmiany
parametry_stale_dla_grupy_odmiany1
odmiana1,odmiana2,odmiana3...
parametry_stale_dla_grupy_odmiany2
odmiana1,odmiana2,odmiana3...
...
```

Na przykład:

```
program
2
ABBAAA
program,programu,program,programowi,programem,programie,programie
ABBAAB
programy,programów,programy,programom,programami,programach,programy
```

Zestaw parametrów jest zależny od części mowy, i szczegółowy opis zamieszczony został w dalszej części dokumentacji.

Warto zwrócić uwagę, że nowy format zapisu fleksji dopuszcza wyrazy wieloczęściowe, czyli formy zwrotne, stopnie wyższe i najwyższe w stopniowaniu opisowym. W tej sytuacji jako wyraz uznaje się odpowiednio całość: "myć się", a wyrazem podstawowym jest bezokolicznik "myć".

2. Tworzenie wzorcowych plików fleksji (WED)

Pliki *.wed (np. [polski.wed](#), [english.wed](#)) umieszczane w katalogu [data/languages](#) opisują strukturę fleksji danego języka. Są też podstawą działania wewnętrznego edytora bazy fleksji.

Stosowane operatory:

liczba-tekst – liczba, która jest przypisana formie wyrazu parametru opisanego przez **tekst**
 , – oddzielenie poziomów (parametrów) opisu,
 / – znak oddzielający alternatywy,
 ... – wskazanie na kontynuację w następnej linii,
 (POS) – parametr pozycyjny (może dotyczyć tylko ostatniego zestawu parametrów!),
 # – oznacza linię komentarza,

Tworzenie plików WED (od WordEditor) umożliwia niejako zewnętrznie w stosunku do programu, definiowanie właściwości fleksyjnych danego języka. Dzięki tym plikom można "nauczyć" Denise fleksji innych języków, i "powiedzieć" programowi jak ma wyglądać ich baza fleksji (pliki *.voc, od VOCABULARY).

Zasada tworzenia wzorcowych plików jest bardzo prosta. Po prostu, w kolejnych liniach podawane są MOŻLIWOŚCI charakteryzowania wyrazu. Np.:

1-rzeczownik, 1-osobowy, 1-pospolity, 1-liczba pojedyncza, 1-rodzaj męski,
(POS) 1-mianownik
1-rzeczownik, 1-osobowy, 1-pospolity, 2-liczba mnoga, 1-rodzaj męski, (POS)
1-mianownik

Gdzie liczba przy każdym z "poziomów" opisu jest bardzo ważna, i oznacza jaka wartość jest przypisana danej możliwości z określonego poziomu. Wystarczy spojrzeć na parametr liczby: 1-liczba pojedyncza, 2-liczba mnoga. Dla skrócenia zapisu można stosować znak '/' oddzielający możliwości:

1-rzeczownik, 1-osobowy, 1-pospolity, 1-liczba pojedyncza/2-liczba mnoga, 1-rodzaj męski, (POS) 1-mianownik

Uzupełniając o pozostałe parametry i korzystając z możliwości dzielenia linii otrzymujemy:

1-rzeczownik,...
...1-osobowy/2-nieosobowy, ...
...1-pospolity/2-własny, ...
...1-liczba pojedyncza, ...
...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki, ...
...(POS)1-mianownik/2-dopełniacz/3-biernik/4-celownik/5-narzędnik/6-
miejscownik/7-wołacz

Zapis ten stanowi pełny opis fleksji rzeczownika. Zgodnie z nim układ wyrazów w bazie fleksji, dla rzeczowników wygląda następująco:

piłka
2
ABBAAA piłka...
ABBAAB ,piłki, piłkę, , piłką...

Wyraźnie widać, że parametrem pozycyjnym jest forma przypadku, co zostało podane w opisie fleksji za pomocą parametru '(POS)'.

Zasady tworzenia i modyfikacji opisu struktury fleksji są więc mało skomplikowane i stały się już zapewne jasne. W przeciwnym razie proszę wzorować się na dostarczonych z programem użytecznych plikach *.wed, lub spojrzeć niżej, gdzie zostały one wstawione w tekst (punkt 3 dowolnego języka).

3. Struktura fleksji dla języka polskiego

Część mowy:

RZECZOWNIK

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczba

POJEDYNCZA

Rodzaj (męski, żeński, nijaki).

MNOGA

Rodzaj (męskoosobowy, niemęskoosobowy).

Żywy/nieżywy.

Osobowy/nieosobowy.

Pospolity/własny.

CZASOWNIK

Typ czasownika:

CZASOWNIK OSOBOWY

Liczba

POJEDYNCZA

Czas:

PRZESZŁY/PRZYSZŁY

Rodzaj (męski, żeński, nijaki).

TERAŻNIEJSZY**MNOGA**

Czas:

PRZESZŁY/PRZYSZŁY

Rodzaj (męskoosobowy, niemęskoosobowy).

TERAŻNIEJSZY

Osoba (pierwsza, druga, trzecia).

Strona (czynna, bierna, zwrotna).

Tryb (orzekający, rozkazujący, przypuszczający).

Aspekt (dokonany, niedokonany).

Przechodni/nieprzechodni.

IMIESŁÓW PRZYMIOTNIKOWY

Czynny/bierny.

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczba

POJEDYNCZA

Rodzaj (męski, żeński, nijaki).

MNOGA

Rodzaj (męskoosobowy, niemęskoosobowy).

IMIESŁÓW PRZYSŁÓWKOWY

Współczesny/uprzedni.

BEZOKOLICZNIK

Narzuca przypadek w twierdzeniu (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Narzuca przypadek w przeczeniu (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Strona (czynna, bierna, zwrotna).

FORMA -NO, -TO.

PRZYMIOTNIK

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczba

POJEDYNCZA

Rodzaj (męski, żeński, nijaki).

MNOGA

Rodzaj (męskoosobowy, niemęskoosobowy).

Stopniowanie:

PROSTE, NIEREGULARNE

Stopień (równy, wyższy, najwyższy).

OPISOWE**PRZYSŁÓWEK**

Stopniowanie:

PROSTE, NIEREGULARNE:

Stopień (równy, wyższy, najwyższy).

OPISOWE

LICZEBNIK

Typ liczebnika:

LICZEBNIK GŁÓWNY oprócz "jeden, tysiąc, milion, miliard":

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Rodzaj (męski, żeński, nijaki).

LICZEBNIK GŁÓWNY "tysiąc, milion, miliard":

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczba (pojedyncza, mnoga).

LICZEBNIK ZBIOROWY:

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

LICZEBNIK UŁAMKOWY

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

LICZEBNIK PORZĄDKOWY, MNOŻNY, WILORAKI, LICZEBNIK GŁÓWNY "jeden":

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczba (pojedyncza, mnoga).

Rodzaj (męski, żeński, nijaki).

ZAIMEK

Znaczenie zaimka (osobowy, zwrotny, dzierżawczy, wskazujący, pytający, względny, nieokreślony, przeczący, upowszechniający).

Typ zaimka:

ZAIMEK RZECZOWNY

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczba (pojedyncza, mnoga).

Rodzaj (męski, żeński, nijaki).

Osobowy/nieosobowy.

ZAIMEK PRZYMIOTNY

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczba

POJEDYNCZA

Rodzaj (męski, żeński, nijaki).

MNOGA

Rodzaj (męskoosobowy, niemęskoosobowy).

ZAIMEK PRZYSŁOWNY

ZAIMEK LICZEBNY

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

PRZYIMEK

Narzuca przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

SPÓJNIK

Typ spójnika (łączny, rozłączny, wyłączający, przeciwstawny, wynikowy, synomiczny).

PARTYKUŁA

Typ partykuły (wzmacniająca, przecząca, pytająca, przypuszczająca, rozkazująca).

WYKRZYKNIK**4. Struktura fleksji zmodyfikowana dla potrzeb programu****4.1. Kategorie odmiany****RZECZOWNIK**

Parametry stałe: osobowość, pospolitość, liczba, rodzaj.

Szeregi odmiany względem: przypadków.

Odmiana jak w p. 1.

CZASOWNIK OSOBOWY

Parametry stałe: tryb, strona, czas, liczba, (rodzaj).

Szeregi odmiany względem: osób.

Odmiana jak w p. 1.

IMIESŁÓW PRZYMIOTNIKOWY

Parametry stałe: typ imiesłowu przymiotnikowego, liczba, rodzaj.

Szeregi odmiany względem: przypadków.

Odmiana jak w p. 1.

IMIESŁÓW PRZYSŁÓWKOWY

Parametry stałe: typ imiesłowu przysłówkowego.

Odmiana jak w p. 1.

BEZOKOLICZNIK

Parametry stałe: strona, narzucany przypadek w twierdzeniu, narzucany przypadek w przeczeniu.

Odmiana jak w p. 1.

FORMA -NO, -TO

Odmiana jak w p. 1.

PRZYMIOTNIK

Parametry stałe: liczba, rodzaj, stopniowanie, (stopień).

Szeregi odmiany względem: przypadków.

Odmiana jak w p. 1.

PRZYSŁÓWEK

Parametry stałe: stopniowanie, (stopień).

Odmiana jak w p. 1.

LICZEBNIK

Parametry stałe: (liczba), (rodzaj).

Szeregi odmiany względem: przypadków.

Typ liczebnika:

LICZEBNIK GŁÓWNY (1) oprócz "jeden, tysiąc, milion, miliard".

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczebnik główny rozróżnia:

RODZAJ (męski, żeński, nijaki).

LICZEBNIK GŁÓWNY (2) "tysiąc, milion, miliard".

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczebnik główny rozróżnia:

LICZBĘ (pojedyncza, mnoga).

LICZEBNIK GŁÓWNY (3) "jeden".

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczebnik główny rozróżnia:

LICZBĘ i RODZAJ (rodzaj: męski, żeński, nijaki, liczba: pojedyncza, mnoga).

LICZEBNIK ZBIOROWY:

Odmiana jak w p. 1.

LICZEBNIK UŁAMKOWY

Odmiana jak w p. 1.

LICZEBNIK PORZĄDKOWY, MNOŻNY, WIELORAKI

Odmiana jak w p. 1.

ZAIMEK RZECZOWNY

Parametry stałe: znaczenie zaimka, osobowość, liczba, rodzaj,

Szeregi odmiany względem: przypadków.

Znaczenie zaimka (osobowy, zwrotny, dzierżawczy, wskazujący, pytający, względny, nieokreślony, przeczący, upowszechniający).

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczba (pojedyncza, mnoga).

Rodzaj (męski, żeński, nijaki).

Osobowy/nieosobowy.

ZAIMEK PRZYMIOTNY

Parametry stałe: znaczenie zaimka, liczba, rodzaj.

Szeregi odmiany względem: przypadków.

Znaczenie zaimka (osobowy, zwrotny, dzierżawczy, wskazujący, pytający, względny, nieokreślony, przeczący, upowszechniający).

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

Liczba (pojedyncza, mnoga).

Rodzaj (męski, żeński, nijaki).

ZAIMEK PRZYSŁOWNY

Parametry stałe: znaczenie zaimka.

Znaczenie zaimka (osobowy, zwrotny, dzierżawczy, wskazujący, pytający, względny, nieokreślony, przeczący, upowszechniający).

ZAIMEK LICZEBNY

Parametry stałe: znaczenie zaimka.

Szeregi odmiany względem: przypadków.

Znaczenie zaimka (osobowy, zwrotny, dzierżawczy, wskazujący, pytający, względny, nieokreślony, przeczący, upowszechniający).

Przypadek (mianownik, dopełniacz, biernik, celownik, narzędnik, miejscownik, wołacz).

PRZYIMEK

Odmiana jak w p. 1.

SPÓJNIK

Odmiana jak w p. 1.

PARTYKUŁA

Odmiana jak w p. 1.

WYKRZYKNIK

Odmiana jak w p. 1.

4.2. Formy wyrazów podstawowych**RZECZOWNIK**

Rzeczownik.

Przypadek mianownik.

Liczba pojedyncza.

Rodzaj męski.

CZASOWNIK OSOBOWY

Bezokolicznik.

IMIESŁÓW PRZYMIOTNIKOWY

Bezokolicznik.

IMIESŁÓW PRZYSŁÓWKOWY

Bezokolicznik.

BEZOKOLICZNIK

Jak wyraz wprowadzany.

FORMA -NO, -TO

Bezokolicznik.

PRZYMIOTNIK

Przymiotnik.

Przypadek mianownik.

Liczba pojedyncza.

Rodzaj męski.

Stopień równy.

PRZYSŁÓWEK

Przysłówek.

Stopień równy.

LICZEBNIK

Liczebnik główny.

Przypadek mianownik.

Rodzaj męski (jeśli wyróżnia).

Liczba pojedyncza (jeśli wyróżnia).

ZAIMEK RZECZOWNY

Zaimek rzeczowny.

Przypadek mianownik.

Liczba pojedyncza.

Rodzaj męski.

ZAIMEK PRZYMIOTNY

Zaimek przymiotny.

Przypadek mianownik.

Liczba pojedyncza.

Rodzaj męski.

ZAIIMEK PRZYSŁOWNY
Jak wyraz wprowadzany.

ZAIIMEK LICZEBNY
Zaimiek liczebny.
Przypadek mianownik.

PRZYIMEK
Jak wyraz wprowadzany.

SPÓJNIK
Jak wyraz wprowadzany.

PARTYKUŁA
Jak wyraz wprowadzany.

WYKRZYKNIK
Jak wyraz wprowadzany.

5. Kompletna struktura fleksji w zapisie formalnym (WordEditor: polski.wed)

1-rzeczownik,...
...1-osobowy/2-nieosobowy,...
...1-pospolity/2-własny,...
...1-liczba pojedyncza,...
...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...
...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz
1-rzeczownik,...
...1-osobowy/2-nieosobowy,...
...1-pospolity/2-własny,...
...2-liczba mnoga,...
...1-rodzaj męskoosobowy/2-rodzaj niemęskoosobowy,...
...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz
2-czasownik osobowy,...
...1-tryb orzekający/2-tryb rozkazujący/3-tryb przypuszczający,...
...1-strona czynna/2-strona bierna/3-strona zwrotna,...
...1-czas przeszły/3-czas przyszły,...
...1-liczba pojedyncza,...
...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...
...1-w twierdzeniu narzuca mianownik/2-w twierdzeniu narzuca dopełniacz/3-w twierdzeniu narzuca celownik/4-w twierdzeniu narzuca biernik/5-w twierdzeniu narzuca narzędnik/6-w twierdzeniu narzuca miejscownik/7-w twierdzeniu narzuca wołacz,...
...1-w przeczeniu narzuca mianownik/2-w przeczeniu narzuca dopełniacz/3-w przeczeniu narzuca celownik/4-w przeczeniu narzuca biernik/5-w przeczeniu narzuca narzędnik/6-w przeczeniu narzuca miejscownik/7-w przeczeniu narzuca wołacz,...
...1-może być łącznikiem w orzeczeniu imiennym/2-nie może być łącznikiem,...
...(POS) 1-pierwsza osoba/2-druga osoba/3-trzecia osoba
2-czasownik osobowy,...
...1-tryb orzekający/2-tryb rozkazujący/3-tryb przypuszczający,...
...1-strona czynna/2-strona bierna/3-strona zwrotna,...
...1-czas przeszły/3-czas przyszły,...
...2-liczba mnoga,...
...1-rodzaj męskoosobowy/2-rodzaj niemęskoosobowy,...

...1-w twierdzeniu narzuca mianownik/2-w twierdzeniu narzuca dopełniacz/3-w twierdzeniu narzuca celownik/4-w twierdzeniu narzuca biernik/5-w twierdzeniu narzuca narzędnik/6-w twierdzeniu narzuca miejscownik/7-w twierdzeniu narzuca wołacz,...

...1-w przeczeniu narzuca mianownik/2-w przeczeniu narzuca dopełniacz/3-w przeczeniu narzuca celownik/4-w przeczeniu narzuca biernik/5-w przeczeniu narzuca narzędnik/6-w przeczeniu narzuca miejscownik/7-w przeczeniu narzuca wołacz,...

...1-może być łącznikiem w orzeczeniu imiennym/2-nie może być łącznikiem,...

...(POS) 1-pierwsza osoba/2-druga osoba/3-trzecia osoba

2-czasownik osobowy,...

...1-tryb orzekający/2-tryb rozkazujący/3-tryb przypuszczający,...

...1-strona czynna/2-strona bierna/3-strona zwrotna,...

...2-czas teraźniejszy,...

...1-liczba pojedyncza/2-liczba mnoga,...

...1-w twierdzeniu narzuca mianownik/2-w twierdzeniu narzuca dopełniacz/3-w twierdzeniu narzuca celownik/4-w twierdzeniu narzuca biernik/5-w twierdzeniu narzuca narzędnik/6-w twierdzeniu narzuca miejscownik/7-w twierdzeniu narzuca wołacz,...

...1-w przeczeniu narzuca mianownik/2-w przeczeniu narzuca dopełniacz/3-w przeczeniu narzuca biernik/4-w przeczeniu narzuca celownik/5-w przeczeniu narzuca narzędnik/6-w przeczeniu narzuca miejscownik/7-w przeczeniu narzuca wołacz,...

...1-może być łącznikiem w orzeczeniu imiennym/2-nie może być łącznikiem,...

...(POS) 1-pierwsza osoba/2-druga osoba/3-trzecia osoba

3-imiesłów przymiotnikowy,...

...1-imiesłów przymiotnikowy czynny/2-imiesłów przymiotnikowy bierny,...

...1-liczba pojedyncza,...

...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

3-imiesłów przymiotnikowy,...

...1-imiesłów przymiotnikowy czynny/2-imiesłów przymiotnikowy bierny,...

...2-liczba mnoga,...

...1-rodzaj męskoosobowy/2-rodzaj niemęskoosobowy,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

4-imiesłów przysłówkowy,...

...1-imiesłów przysłówkowy współczesny/2-imiesłów przysłówkowy uprzedni

5-bezokolicznik,...

...1-strona czynna/2-strona bierna/3-strona zwrotna,...

...1-w twierdzeniu narzuca mianownik/2-w twierdzeniu narzuca dopełniacz/3-w twierdzeniu narzuca celownik/4-w twierdzeniu narzuca biernik/5-w twierdzeniu narzuca narzędnik/6-w twierdzeniu narzuca miejscownik/7-w twierdzeniu narzuca wołacz,...

...1-w przeczeniu narzuca mianownik/2-w przeczeniu narzuca dopełniacz/3-w przeczeniu narzuca celownik/4-w przeczeniu narzuca biernik/5-w przeczeniu narzuca narzędnik/6-w przeczeniu narzuca miejscownik/7-w przeczeniu narzuca wołacz

6-forma -no -to,...

...1-w twierdzeniu narzuca mianownik/2-w twierdzeniu narzuca dopełniacz/3-w twierdzeniu narzuca celownik/4-w twierdzeniu narzuca biernik/5-w twierdzeniu narzuca narzędnik/6-w twierdzeniu narzuca miejscownik/7-w twierdzeniu narzuca wołacz,...

...1-w przeczeniu narzuca mianownik/2-w przeczeniu narzuca dopełniacz/3-w przeczeniu narzuca celownik/4-w przeczeniu narzuca biernik/5-w przeczeniu narzuca narzędnik/6-w przeczeniu narzuca miejscownik/7-w przeczeniu narzuca wołacz

7-przymiotnik,...

...1-liczba pojedyncza,...

...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...

...1-stopniowanie proste/2-stopniowanie nieregularne/3-stopniowanie opisowe,...

...1-stopień równy/2-stopień wyższy/3-stopień najwyższy,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

7-przymiotnik,...

...2-liczba mnoga,...

...1-rodzaj męskoosobowy/2-rodzaj niemęskoosobowy,...

...1-stopniowanie proste/2-stopniowanie nieregularne/3-stopniowanie opisowe,...

...1-stopień równy/2-stopień wyższy/3-stopień najwyższy,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

8-przysłówek,...

...1-stopniowanie proste/2-stopniowanie nieregularne/3-stopniowanie opisowe,...

...1-stopień równy/2-stopień wyższy/3-stopień najwyższy

9-liczebnik,...

...1-liczebnik główny oprócz "jeden tysiąc milion miliard",...

...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

9-liczebnik,...

...2-liczebnik główny "tysiąc milion miliard",...

...1-liczba pojedyncza/2-liczba mnoga,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

9-liczebnik,...

...3-liczebnik główny "jeden",...

...1-liczba pojedyncza/2-liczba mnoga,...

...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

9-liczebnik,...

...4-liczebnik zbiorowy/5-liczebnik ułamkowy,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

9-liczebnik,...

...6-liczebnik porządkowy mnożny wieloraki,...

...1-liczba pojedyncza/2-liczba mnoga,...

...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

10-zaimek rzeczowny,...

...1-osobowy/3-dzierżawczy,...

...1-osobowy/2-nieosobowy,...

...1-liczba pojedyncza,...

...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...

...1-pierwsza osoba/2-druga osoba/3-trzecia osoba,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

10-zaimek rzeczowny,...

...1-osobowy/3-dzierżawczy,...

...1-osobowy/2-nieosobowy,...

...2-liczba mnoga,...

...1-rodzaj męskoosobowy/2-rodzaj niemęskoosobowy,...

...1-pierwsza osoba/2-druga osoba/3-trzecia osoba,...

...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz

10-zaimek rzeczowny, ...
 ...2-zwrotny/4-wskazujący/5-pytający/6-względny/7-nieokreślony/8-przeczący/9-upowszechniający, ...
 ...1-osobowy/2-nieosobowy, ...
 ...1-liczba pojedyncza, ...
 ...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki, ...
 ...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz
 10-zaimek rzeczowny, ...
 ...2-zwrotny/4-wskazujący/5-pytający/6-względny/7-nieokreślony/8-przeczący/9-upowszechniający, ...
 ...1-osobowy/2-nieosobowy, ...
 ...2-liczba mnoga, ...
 ...1-rodzaj męskoosobowy/2-rodzaj niemęskoosobowy, ...
 ...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz
 11-zaimek przymiotny, ...
 ...1-osobowy/2-zwrotny/3-dzierżawczy/4-wskazujący/5-pytający/6-względny/7-nieokreślony/8-przeczący/9-upowszechniający, ...
 ...1-liczba pojedyncza, ...
 ...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki, ...
 ...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz
 11-zaimek przymiotny, ...
 ...1-osobowy/2-zwrotny/3-dzierżawczy/4-wskazujący/5-pytający/6-względny/7-nieokreślony/8-przeczący/9-upowszechniający, ...
 ...2-liczba mnoga, ...
 ...1-rodzaj męskoosobowy/2-rodzaj niemęskoosobowy, ...
 ...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz
 12-zaimek przysłowny, ...
 ...1-osobowy/2-zwrotny/3-dzierżawczy/4-wskazujący/5-pytający/6-względny/7-nieokreślony/8-przeczący/9-upowszechniający/10-"dlaczego"
 13-zaimek liczebnny, ...
 ...1-osobowy/2-zwrotny/3-dzierżawczy/4-wskazujący/5-pytający/6-względny/7-nieokreślony/8-przeczący/9-upowszechniający, ...
 ...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-miejscownik/7-wołacz
 14-przyimek, ...
 ...1-narzuca mianownik/2-narzuca dopełniacz/3-narzuca celownik/4-narzuca biernik/5-narzuca narzędnik/6-narzuca miejscownik/7-narzuca wołacz
 15-spójnik, ...
 ...1-łączny/2-wyłaczający/3-przeciwstawny/4-wynikowy/5-alternatywny
 16-partykuła, ...
 ...1-wzmacniająca/2-przecząca/3-pytająca/4-przypuszczająca/5-rozkazująca/6-łącznik "to"
 17-wykrzyknik

1. Pliki językowe [language] – słowo wstępne

Pliki językowe są podstawą do komunikacji z programem. Zawierają one wszystkie niezbędne do analizy wypowiedzi informacje. Są jednocześnie furtką umożliwiającą nauczenie programu komunikowania w prawie dowolnym języku, takim, którego opis gramatyczny może zostać zawarty w opracowanej strukturze.

Utworzenie plików językowych umożliwia swobodną pracę językoznawcom, którzy mogą w sposób bezpośredni wpływać na dokonywaną analizę zdania. Uniezależnia to rozwój programu od programistów, i jednocześnie zdejmuje z nich ciężar opracowywania reguł gramatycznych, co nie powinno być ich głównym zadaniem.

Zapraszam wykwalifikowaną kadrę językoznawców do wspólnego podjęcia trudu opracowania reguł gramatycznych ich narodowych języków. Po zebraniu pierwszych doświadczeń, podjęte zostaną próby utworzenia modułu translacyjnego. Wspólnymi siłami może uda nam się postawić silny krok w stronę łatwego porozumienia między ludźmi posługującymi się różnymi językami.

2. Pliki językowe – opis szczegółowy

2.1. LANGUAGE.WED – struktura fleksji

Opis.

- Plik zawiera pełny opis fleksji języka.
- Cały dalszy opis gramatyczny musi być oparty o strukturę opisaną w tym miejscu.
- Plik ten jest wykorzystywany przez program WordEditor do edycji słów w bazie, dlatego inaczej niż w plikach pozostałych, ważne są podawane w nim nazwy.

Składnia – operatory.

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,
... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,
... – na początku linii oznaczają, że jest ona kontynuacją linii,
/ – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,
, – oddziela kolejne pozycje,
(POS) – może być postawiony jedynie w ostatniej linii, na samym jej początku i oznacza, że parametr w tym miejscu jest parametrem pozycyjnym (patrz LANGUAGE.VOC),
- – oddziela NUMER parametru od jego słownego OPISU,
NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,
OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

Przykładowy zapis:

Fragment odmiany rzeczowników

1-rzeczownik,...
 ...1-osobowy/2-nieosobowy,...
 ...1-pospolity/2-własny,...
 ...1-liczba pojedyncza,...
 ...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...
 ...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-
 miejscownik/7-wołacz
 1-rzeczownik,...
 ...1-osobowy/2-nieosobowy,...
 ...1-pospolity/2-własny,...
 ...2-liczba mnoga,...
 ...1-rodzaj męskoosobowy/2-rodzaj niemęskoosobowy,...
 ...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-
 miejscownik/7-wołacz

Oznacza, że istnieje część mowy o numerze 1, nazwana jako RZECZOWNIK. Na drugiej pozycji zawarta jest informacja, że ta część mowy może być OSOBOWA lub NIEOSOBOWA. Parametrowi OSOBOWA został przypisany numer 1, a NIEOSOBOWA numer 2.

Na pozycji trzeciej następuje określenie, czy rzeczownik jest POSPOLITY (numer 1) czy WŁASNY (numer 2). Analogicznie, pozycja czwarta mówi, że RZECZOWNIK może występować w LICZBIE POJEDYNCZEJ (numer 1) lub LICZBIE MNOGIEJ (numer 2). Jeśli występuje w LICZBIE POJEDYNCZEJ, to na pozycji piątej jest informacja o rodzaju rzeczownika, który może mieć RODZAJ MĘSKI, RODZAJ ŻEŃSKI lub RODZAJ NIJAKI.

Jeśli rzeczownik występuje w LICZBIE MNOGIEJ, to na pozycji piątej jest informacja o tym, że może występować w RODZAJU MĘSKOOSOBOWYM lub RODZAJU NIEMĘSKOOSOBOWYM. Bez względu na parametry na wcześniejszych pozycjach, pozycja szósta informuje, że RZECZOWNIK może występować w siedmiu różnych przypadkach.

Dodatkowo zapis '(POS)' informuje, że forma przypadku jest parametrem POZYCYJNYM co oznacza, że np. RZECZOWNIK w formie przypadku piątego (NARZĘDNIK) zostanie zapisany na pozycji piątej.

Przy tak zdefiniowanych parametrach odmiany, można podać opis, jaki zostanie zawarty w bazie fleksji np. dla wyrazu "człowieki":

AAAAA „człowieki”...

Zapis ten należy rozumieć jako: AAAAAC, gdzie "C" oznacza, że wyraz "człowieki" znajduje się na trzeciej pozycji na liście odmian – trzecia pozycja, oznacza, że wyraz "człowieki" ma formę przypadku celownika. Pozostałe parametry mówią, że wyraz jest odpowiednio: rzeczownikiem osobowym, pospolitym, w liczbie pojedynczej, i ma rodzaj męski.

2.2. LANGUAGE.VOC – słownik fleksyjny

Opis.

- Plik zawiera informacje o słowach wykorzystywanych w języku (jest to słownik fleksyjny).
- Program może analizować tylko te zdania, których wyrazy są zapisane w tym słowniku.
- Wyrazy są poukładane alfabetycznie, względem WYRAZÓW PODSTAWOWYCH, gdzie WYRAZ PODSTAWOWY to wyraz "bazowy" dla odmiany danego wyrazu. Dla rzeczownika będzie to mianownik liczby pojedynczej, dla przymiotnika mianownik liczby pojedynczej rodzaju męskiego w stopniu równym, a dla czasownika bezokolicznik. Pozostałe wyrazy to WYRAZY POCHODNE, które zostały utworzone z

wyrazu podstawowego przez jego odmianę np. przez przypadki i liczby (rzeczownik), przypadki, liczby, rodzaje i stopnie (przymiotnik), czy osoby, liczby, czasy i inne parametry dla czasownika (definicja ta została wzięta z dokumentacji programu "External Vocabulary Editor for Denise^{®n} (xVED), tam też znajdują się odpowiednie przykłady).

- Plik ten w odpowiedniej formie (patrz przykład), będącej skondensowanym zapisem parametrów odmiany wyrazów, tworzony jest przez program WordEditor, dostarczony wraz z programem Denise[®], dlatego nie jest wymagana znajomość jego struktury.
- Pojęcie POZYCJI wyrazu oznacza miejsce na liście odmian, w którym zapisany zostanie wyraz w danej formie – jest to wykorzystywane, aby zmniejszyć ilość potrzebnej do zapamiętania odmian pamięci, i uprościć zapis w bazie fleksji, umożliwiając łatwe modyfikacje; proponuje się stosowanie parametru pozycyjnego, gdy wyraz posiada więcej form na danej pozycji; oto przykład, gdzie obserwujemy 3 pozycje – osoby czasownika:

mieć

1

BAABADBB mam, masz, ma

Na pozycji pierwszej jest wyraz "mam", na drugiej "masz", a na trzeciej "ma". Ciąg "BAABADBB" należy rozumieć analogicznie jak podano w przykładzie w punkcie 1.

2.3. LANGUAGE.GR1 – przedrostki i przyrostki

Opis.

- Plik zawiera informacje o tym, jakie wyrazy są pisane łącznie.
- Dopuszczalne są połączenia najwyżej dwóch wyrazów.
- OPIS w analizie jest pomijany i służy jedynie edycji.

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,

... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,

... – na początku linii oznaczają, że jest ona kontynuacją linii,

/ – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,

, – oddziela kolejne pozycje,

- – oddziela NUMER parametru od jego słownego OPISU,

NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,

OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

+ – oddziela fleksję pierwszego z połączonych wyrazów od drugiego,

Przykładowy zapis:

Przeczenie "nie" jest pisane łącznie z rzeczownikami.

16-partykuła,2-przecząca+1-rzeczownik

Oznacza, że do części mowy o NUMERZE 16, której parametr na pozycji drugiej ma NUMER 2 może być połączona część mowy o NUMERZE 1.

2.4. LANGUAGE.GR2 – lista wyrazów samodzielnych

Opis.

- Wyraz samodzielny to taki wyraz, który może sam jeden tworzyć wyrażenie, w przeciwieństwie do wyrazów niesamodzielnych, które mogą jedynie towarzyszyć innym wyrazom samodzielnym, nie mogąc tworzyć wyrażenia bez wyrazu samodzielnego,
- Wyraz samodzielny może pełnić funkcję w zdaniu, wyraz niesamodzielny bez wyrazu samodzielnego nie może pełnić funkcji w zdaniu.

Składnia – operatory.

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,

- – oddziela NUMER parametru od jego słownego OPISU,

NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,

OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

Przykładowy zapis:

1-rzeczownik
 2-czasownik osobowy
 3-imiesłów przymiotnikowy
 4-imiesłów przysłówkowy

Oznacza, że części mowy o NUMERACH 1,2,3,4 są częściami samodzielnymi. Wszystkie inne części mowy będą uważane, za części niesamodzielne.

2.5. LANGUAGE.GR3 – łączenie wyrazów niesamodzielnych z samodzielnymi

2.6. LANGUAGE.GR4 – szeregi wyrazów samodzielnych

2.7. LANGUAGE.GR5 – łączenie w wyrażenie wyrazów samodzielnych różnych części mowy

Opis.

- Te trzy pliki gramatyczne posiadają tę samą składnię,
- Istnieje ściśle określona kolejność łączenia wyrazów przez program: najpierw sprawdzane są tylko połączenia wyrazów niesamodzielnych z samodzielnymi (opisane w pliku LANGUAGE.GR3), dopiero potem szeregi (LANGUAGE.GR4), a na końcu wyrazy samodzielne różnych części mowy (LANGUAGE.GR5),
- Dla poprawności działania programu bezwzględnie należy pamiętać o kolejności wykorzystywania przez program informacji zawartych w tych plikach.

Składnia.

Kolejność zapisu wyrazów mówi o kolejności ich występowania w zdaniu,

Operatory:

- # – na początku linii oznacza, że jest to linia komentarza, pomijana przez program,
- ... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,
- ... – na początku linii oznaczają, że jest ona kontynuacją linii,
- / – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,
- , – oddziela kolejne pozycje w odmianie,
- – oddziela NUMER parametru od jego słownego OPISU,
- NUMER** – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,
- OPIS** – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.
- + – oddziela fleksję pierwszego z połączonych wyrazów od drugiego,
- ? – oznacza, że parametr na danej pozycji może być dowolny, (nie jest brany pod uwagę)
- \$nr** – oznacza jakiś parametr, ten sam w każdym wyrazie (co do wartości), nr to numer parametru,
- (ENV-nr)** – oznacza, że jeśli nie jest ustawiona wewnętrzna zmienna (listę zmiennych podano niżej) o numerze nr, to parametr na danej pozycji nie jest brany pod uwagę
- 1-ZWIĄZEK: opis+opis** – oznacza połączenie dwóch wyrazów o częściach mowy podanych w opisach; kolejność jest następująca: opis wyrazu nadrzędnego + opis wyrazu nadrzędnego,
- 2-MIĘDZY NIE MOŻE BYĆ: część mowy;...;część mowy** – lista części mowy, które nie mogą występować MIĘDZY wyrazami łączonymi,

Opis zmiennych wewnętrznych (ENV-nr):

- (ENV-1) - TWIERDZENIE
- (ENV-2) - NEGACJA

Zapis prostego przykładu:

- # rzeczownik (wyrażenie przyimkowe) "(siedzę) na krześle"
- # rzeczownik występuje w przypadku narzuconym przez przyimek,
- 1-ZWIĄZEK: 14-przyimek,\$1+1-rzeczownik,?,?,?,\$1
- 2-MIĘDZY NIE MOŻE BYĆ: 2-czasownik osobowy

Z powyższego wynika, że dopuszczalny jest związek wyrazów o części mowy z numerem 14 (przyimek) i części mowy z numerem 1 (rzeczownik). Dodatkowo wymagane jest, aby parametr na pozycji drugiej przyimka i parametr na pozycji szóstej rzeczownika były co do wartości te same. Ponadto określono, że między przyimkiem i rzeczownikiem nie może występować część mowy: czasownik osobowy.

Inny przykład:

- # czasownik osobowy, rzeczownik (orzeczenie słowno-imiennie) "(Mariusz) został mnichem"
- # – poprawna kolejność: czasownik+rzeczownik,
- # – czasownik osobowy musi należeć do rodziny łączników w orzeczeniu imiennym,
- # – rzeczownik musi być w przypadku narzuconym przez łącznik (zwykle mianownik/narzędnik)
- 1-ZWIĄZEK: 2-czasownik osobowy,?,?,1-czas przeszły/3-czas przyszły,?,?,(ENV-1) \$1,(ENV-2) \$1,1-może być łącznikiem w orzeczeniu imiennym+1-rzeczownik,?,?,?,\$1

1-ZWIĄZEK: 2-czasownik osobowy,?,?,2-czas terażniejszy,?,(ENV-1) \$1,(ENV-2) \$1,1-może być łącznikiem w orzeczeniu imiennym+1-rzeczownik,?,?,?,,\$1

Ten zapis określa, jakie warunki muszą być spełnione, aby części mowy o numerach 2 i 1 zostały połączone w jednym wyrażeniu. Zapis '1-czas przeszły/3-czas przyszły' oznacza, że czasownik musi występować w jednej z tych dwóch form. Druga linia natomiast jest odpowiednia dla czasowników występujących w czasie terażniejszym. Takie rozdzielanie jest konieczne ze względu na to, że w czasie terażniejszym czasownik inaczej się odmienia niż w pozostałych dwóch czasach (patrz opis fleksji).

Zapis '(ENV-1) \$1,(ENV-2) \$1' początkowo może budzić sprzeciw, jednak jest zupełnie poprawny. Oznacza mianowicie, że jeśli jest ustawiona zmienna wewnętrzna ENV-1 (TWIERDZENIE), to parametr na tej pozycji musi być taki sam jak odpowiedni parametr drugiej części mowy. Analogicznie wygląda rzecz z drugą częścią: '(ENV-2) \$1'. Zapis ten został wprowadzony dlatego, że rzeczownik musi występować w określonej formie przypadku, która może być jednak różna w TWIERDZENIU i NEGACJI. Informacja o wymaganej przez czasownik formie znajduje się na dwóch sąsiednich pozycjach "(ENV-1) \$1,(ENV-2) \$1", jednak w danym momencie wymagane jest, aby wziąć pod uwagę tylko JEDNĄ z tych dwóch pozycji. To właśnie zapewnia powyższy zapis.

Wymagane jest dobre zrozumienie dostępnej składni, gdyż tylko wtedy otrzymuje się możliwość pełnego wykorzystania narzędzi oferowanych przez program. Jest to kluczem do utworzenia poprawnego modułu gramatycznego.

2.8. LANGUAGE.GR6 – lista części mowy, które nie mogą występować w danej funkcji zdania

Opis.

- Lista służy przyśpieszeniu pracy programu i jest wstępną oceną analizy zdania,
- W liście ważne są jedynie NUMERY, wszelkie OPISY są dla wygody osoby opracowującej plik i nie mają wpływu na pracę programu,

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,
... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,
... – na początku linii oznaczają, że jest ona kontynuacją linii,
; – oddziela kolejne części mowy,
!= – oddziela nazwę
- – oddziela NUMER parametru od jego słownego OPISU,
NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer części mowy,
OPIS – słowny opis – nazwa części mowy oznaczonej przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

Składnia:

CZĘŚĆ_ZDANIA!=CZĘŚĆ_MOWY;CZĘŚĆ_MOWY...;CZĘŚĆ_MOWY;CZĘŚĆ_MOWY

Gdzie:

CZĘŚĆ_ZDANIA to zapis: NUMER_CZĘŚCI_ZDANIA-OPIS, przy czym NUMER_CZĘŚCI_ZDANIA przyjmuje wartości odpowiednio:

- 1-podmiot (SUBJECT)
- 2-orzeczenie (PREDICATE)
- 3-przydawka (MODIFIER)
- 4-dopełnienie (OBJECT)
- 5-okolicznik (ADVERBIAL)

Przykładowy zapis:

1-podmiot!=2-czasownik osobowy;8-przysłówek;4-imiestów przysłówkowy;6-forma -no -to;...
...12-zaimek przysłowny;13-zaimek liczebny

Oznacza, że grupa, która ma pełnić funkcję o NUMERZE 1 (nazwaną jako PODMIOT), nie może zawierać wyrazu o częściach mowy z NUMERAMI: 2,8,4,6,12,13.

2.9. LANGUAGE.GR7 – opis związków między wyrażeniami

Opis.

- Jest to praktycznie najważniejszy spośród wszystkich plików językowych, gdyż w największym stopniu wpływa na liczbę poprawnie analizowanych zdań,
- W pliku tym opisane są dopuszczalne związki między wyrażeniami,
- Opis w pliku dotyczy połączeń między wyrazami jako częściami mowy oraz jako częściami zdania, uwzględniając również kolejność w zdaniu,
- W opisie związku, w przypadku, gdy wyrażenie składa się z wielu części mowy pod uwagę brana jest część mowy określona jako samodzielna,
- W określaniu opisów istotne są jedynie NUMERY - opisy następujące po myślniku są istotne jedynie dla twórców modułu, dla których sam zapis liczbowy nie byłby tak jasny.

Składnia – operatory:

- # – na początku linii oznacza, że jest to linia komentarza, pomijana przez program,
- ... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,
- ... – na początku linii oznaczają, że jest ona kontynuacją linii,
- / – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,
- , – oddziela kolejne pozycje w odmianie,
- – oddziela NUMER parametru od jego słownego OPISU,
- NUMER** – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,
- OPIS** – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.
- + – oddziela fleksję pierwszego z połączonych wyrazów od drugiego,
- ? – oznacza, że parametr na danej pozycji może być dowolny, (nie jest brany pod uwagę)
- \$nr** – oznacza jakiś parametr, ten sam w każdym wyrazie (co do wartości), nr to numer parametru,
- (ENV-nr)** – oznacza, że jeśli nie jest ustawiona wewnętrzna zmienna o numerze nr, to parametr na danej pozycji nie jest brany pod uwagę

1-ZWIĄZEK: opis+opis – oznacza połączenie dwóch wyrazów o częściach mowy podanych w opisach; kolejność jest następująca: opis wyrazu nadrzędnego + opis wyrazu podrzędnego,

2-FUNKCJE: funkcja+funkcja;...;funkcja+funkcja – oznacza możliwe kombinacje funkcji pełnionych w zdaniu; kolejność j. w.

3-MUSI BYĆ W NADRZĘDNEJ: część mowy;...;część mowy – jest to lista części mowy, które MUSZĄ występować w wyrażeniu nadrzędnym

4-MUSI BYĆ W PODRZĘDNEJ: część mowy;...;część mowy – j. w. dla wyrażenia podrzędnego,

5-NIE MOŻE BYĆ W NADRZĘDNEJ: część mowy;...;część mowy – analogicznie – lista wyrazów, które nie mogą występować w wyrażeniu nadrzędnym

6-NIE MOŻE BYĆ W PODRZĘDNEJ: część mowy;...;część mowy – j. w. dla wyrażenia podrzędnego,

7-KOLEJNOŚĆ: numer_grupy_mającej_wystąpić_wcześniej – określa wymaganą kolejność występowania wyrażen w zdaniu: 1-nadrzędna,2-podrzędna,

8-MIĘDZY NIE MOŻE BYĆ: funkcja;...;funkcja – lista funkcji, których nie mogą pełnić grupy znajdujące się MIĘDZY grupą nadrzędną i podrzędną,

9-MIĘDZY NIE MOŻE BYĆ WYRAZU: opis;opis;...;opis – oznacza, że między grupą nadrzędną a podrzędną nie może być wyrazów o danej fleksji,

Opis zmiennych wewnętrznych (ENV-nr):

(ENV-1) - TWIERDZENIE

(ENV-2) - PRZECZENIE

Przykłady stosowania poszczególnych opisów:

1-ZWIĄZEK: 1-rzeczownik+2-czasownik osobowy

Oznacza prosty związek między rzeczownikiem (nadrzędnym) i czasownikiem (podrzędnym).

2-FUNKCJE: 1-podmiot+2-orzeczenie;1-podmiot+3-przydawka

Oznacza, że możliwe są sytuacje, kiedy grupa nadrzędna jest podmiotem, a podrzędna orzeczeniem, lub grupa nadrzędna jest podmiotem a podrzędna przydawką.

3-MUSI BYĆ W NADRZĘDNEJ: 14-przyimek;1-rzeczownik

Oznacza, że w wyrażeniu nadrzędnym muszą być: przyimek i rzeczownik.

5-NIE MOŻE BYĆ W NADRZĘDNEJ: 2-czasownik osobowy

Oznacza, że w wyrażeniu nadrzędnym NIE MOŻE być czasownika.

7-KOLEJNOŚĆ: 1-nadrzędna

Oznacza, że w zdaniu pierwsza musi występować grupa nadrzędna.

8-MIĘDZY NIE MOŻE BYĆ: 2-orzeczenie

Oznacza, że między grupą nadrzędną i podrzędną NIE MOŻE być orzeczenia.

Pełny przykład:

RZECZOWNIK	CZASOWNIK OSOBOWY	DOPUSZCZALNE ZWIĄZEK GŁÓWNY (ORZEKAJĄCY)
Dziecko	Plakało	Wspólne: liczba, rodzaj

1-ZWIĄZEK: 1-rzeczownik,?,?,\$1,\$2+2-czasownik osobowy,?,?,1-czas przeszły/3-czas przyszły,\$1,\$2

2-FUNKCJE: 1-podmiot+2-orzeczenie

#

1-ZWIĄZEK: 1-rzeczownik,?,?,\$1+2-czasownik osobowy,?,?,2-czas teraźniejszy,\$1

2-FUNKCJE: 1-podmiot+2-orzeczenie

Jak widać, nie jest wymagane podawanie WSZYSTKICH opisów – często wystarczają choćby dwa. Opis związku jest analogiczny jak w punkcie 7.

2.10. LANGUAGE.LO1 – plik z wyrazami negującymi wyrażenie

Opis.

- Plik zawiera informacje o tym jakie człony negują wyrażenie,
- OPIS w analizie jest pomijany i służy jedynie edycji,
- Negacja służy m. in. dostosowaniu form gramatycznych różnych w twierdzeniu i przeczeniu.

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,

... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,

... – na początku linii oznaczają, że jest ona kontynuacją linii,

, – oddziela kolejne pozycje,

- – oddziela NUMER parametru od jego słownego OPISU,

NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,

OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

Przykładowy zapis:

16-partykuła,2-przecząca

Oznacza, że część mowy o numerze 16 (partykuła) o parametrze na pozycji drugiej równym 2 (przecząca) powoduje, że grupa, w której wyraz ten się znajduje zostaje zanegowana.

2.11. LANGUAGE.LO2 – plik z łącznikami (operatorami logicznymi)

Opis.

- Plik zawiera informacje o wyrazach, które są operatorami logicznymi: AND (i – koniunkcja), OR (lub – alternatywa), IF (jeśli – implikacja),

- Dane te wykorzystywane są w rozumieniu relacji logicznych między zdaniem,
- OPIS w analizie jest pomijany i służy jedynie edycji.

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,
 ... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,
 ... – na początku linii oznaczają, że jest ona kontynuacją linii,
 , – oddziela kolejne pozycje,
 / – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,
 - – oddziela NUMER parametru od jego słownego OPISU,
NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,
OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.
1-AND: opis fleksji – oznacza, że wyraz o fleksji zgodnej z opisem fleksji jest rozumiany jako logiczny operator AND (i),
2-NXOR: opis fleksji – oznacza, że wyraz o fleksji zgodnej z opisem fleksji jest rozumiany jako logiczny operator NXOR ($A \text{ NXOR } B = A \text{ AND NOT } B \text{ OR NOT } A \text{ AND } B$)
3-OR: opis fleksji – oznacza, że wyraz o fleksji zgodnej z opisem fleksji jest rozumiany jako logiczny operator OR,

Przykładowy zapis:

1-AND: 15-spójnik,1-łączny/3-przeciwstawny

Oznacza, że część mowy z numerem 15 (spójnik), która ma na pozycji drugiej parametr 1 lub 3 (spójnik łączny lub przeciwstawny), jest rozumiana jako logiczne AND (i).

2.12. LANGUAGE.LO3 – plik z kwantyfikatorami (ogólnymi i szczegółowymi)

Opis.

- Plik praktycznie opisuje zastępowanie wyrazów przez zaimki,
- Dotyczy to zaimków, które są z logicznego punktu widzenia kwantyfikatorami,
- Informacje z tego pliku wykorzystywane są w analizie logicznej,
- OPIS w analizie jest pomijany i służy jedynie edycji.

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,
 ... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,
 ... – na początku linii oznaczają, że jest ona kontynuacją linii,
 , – oddziela kolejne pozycje,
 / – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,
 - – oddziela NUMER parametru od jego słownego OPISU,
NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,
OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

? – oznacza, że parametr na danej pozycji może być dowolny, (nie jest brany pod uwagę)

\$nr – oznacza jakiś parametr, ten sam w każdym wyrazie (co do wartości), nr to numer parametru,

1-ALL: opis fleksyjny kwantyfikatora – opis fleksyjny zastępowany (kwantyfikator uniwersalny) pokazuje kiedy wyraz opisany po lewej stronie równania, jest uznawany za kwantyfikator uniwersalny, zastępujący wyraz opisany po prawej stronie równości, gdzie kwantyfikatorem uniwersalnym jest np. wyraz "wszyscy" i zastępuje np. "Marek"

2-EXIST: opis fleksyjny kwantyfikatora – opis fleksyjny zastępowany (kwantyfikator egzystencjalny) pokazuje kiedy wyraz opisany po lewej stronie równania, jest uznawany za kwantyfikator egzystencjalny, zastępujący wyraz opisany po prawej stronie równości, gdzie kwantyfikatorem egzystencjalnym jest np. wyraz "ktoś" i zastępuje np. "lingwista"

3-NO_EXIST: opis fleksyjny kwantyfikatora – opis fleksyjny zastępowany (zanegowany kwantyfikator egzystencjalny) pokazuje kiedy wyraz opisany po lewej stronie równania, jest uznawany za zanegowany kwantyfikator egzystencjalny, zastępujący wyraz opisany po prawej stronie równości, gdzie zanegowanym kwantyfikatorem jest np. wyraz "nobody", i neguje np. wyraz "denise",

4-NO_EXIST_NO: opis fleksyjny kwantyfikatora – opis fleksyjny zastępowany (zanegowany kwantyfikator egzystencjalny wymagający dodatkowego zaprzeczenia) pokazuje kiedy wyraz opisany po lewej stronie równania, jest uznawany za zanegowany kwantyfikator egzystencjalny wymagający dodatkowej negacji, zastępujący wyraz opisany po prawej stronie równości, gdzie zanegowanym kwantyfikatorem jest np. wyraz "nikt", i neguje np. wyraz "Marek"; kwantyfikator ten wymaga dodatkowego przeczenia, np.: "nikt NIE jest..."

Przykładowy zapis:

1-ALL: 10-zaimek rzeczowny,9-upowszechniający,\$1,?, \$2=1-rzeczownik,\$1,?,?, \$2

Oznacza, że zaimek rzeczowny upowszechniający, jest kwantyfikatorem ogólnym, zastępującym rzeczownik, jeśli oba wyrazy mają wspólny parametr 3 (w zaimku) i 2 (w rzeczowniku), oraz 5 (w zaimku) i 5 (w rzeczowniku). W tym przypadku będą to odpowiednio: wspólny rodzaj i zgodna osobowość/pospolitość. Mówimy to np. o zastępowaniu (logicznym) wyrazu "Marek" przez zaimek (kwantyfikator) "wszyscy".

Inny zapis:

2-EXIST: 10-zaimek rzeczowny,7-nieokreślony,\$1,\$2=10-zaimek rzeczowny,1-osobowy/4-wskazujący,\$1,?, \$2

Opisuje zależność między "ktoś" i "on/ten". W tym przypadku wymagana jest zgodność w osobowości i liczbie.

Ostatni przykład:

4-NO_EXIST_NO: 10-zaimek rzeczowny,8-przeczący,\$1=1-rzeczownik,\$1

Przedstawia zależność między "nikt" a "Marek" (na przykład), i wymaga zgodności osobowości.

2.13. LANGUAGE.LO4 – plik z zaimkami wymuszającymi opisowy format odpowiedzi

Opis.

- Plik zawiera informacje o interpretacji zaimków pytających,
- Zawarte są też informacje na temat odpowiedzi wymuszanej przez dany zaimek,
- Określone jest też, które zaimki powinny zostać usunięte przed dalszą analizą.

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,

... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,

... – na początku linii oznaczają, że jest ona kontynuacją linii,

, – oddziela kolejne pozycje,

/ – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,

- – oddziela NUMER parametru od jego słownego OPISU,

NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,

OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

? – oznacza, że parametr na danej pozycji może być dowolny, (nie jest brany pod uwagę)

\$nr – oznacza jakiś parametr, ten sam w każdym wyrazie (co do wartości), nr to numer parametru,

1-PROVE_WITH_CON: opis fleksyjny – opis fleksyjny zastępowany pokazuje, że wyraz o podanym z lewej strony opisie fleksyjnym, zastępuje wyraz o opisie z prawej strony, przy czym zaimek wymusza odpowiedź w postaci pełnego dowodu, i zaimek nie zostaje wykasowany przed dalszą analizą,

2-PROVE_WITHOUT_CON: opis fleksyjny – pokazuje, że wyraz o podanym opisie fleksyjnym narzuca, wymaganie przedstawienia pełnego dowodu w odpowiedzi, a sam wyraz musi być usunięty ze zdania przed dalszą analizą,

3-RESULT_WITH_CON: opis fleksyjny – opis fleksyjny zastępowany pokazuje, że wyraz o podanym z lewej strony opisie fleksyjnym, zastępuje wyraz o opisie z prawej strony, przy czym zaimek, wymusza odpowiedź w postaci twierdzenia wynikowego, i zaimek nie zostaje wykasowany przed dalszą analizą,

4-RESULT_WITHOUT_CON: opis fleksyjny – pokazuje, że wyraz o podanym opisie fleksyjnym określa, wymaganie odpowiedzi wynikowej, a sam wyraz musi być usunięty ze zdania przed dalszą analizą.

Przykładowy zapis:

3-RESULT_WITH_CON: 10-zaimek rzeczowny,5-pytający,\$1,?,?,\$2=1-rzeczownik,\$1,?,?,\$2

Oznacza, że zaimek rzeczowny pytający, zastępujący rzeczownik pod warunkiem wspólnej formy osobowości i przypadku, wymaga odpowiedzi wynikowej i nie jest usuwany ze zdania przed dalszą analizą.

2.14. LANGUAGE.LO5 – plik z opisem identyfikatorów słownych

Opis.

- Plik zawiera informacje na temat wyrazów, które są podstawą identyfikacji tematu,
- Opisane w pliku identyfikatory są identyfikatorami słownymi, nie zdaniowymi (identyfikatory słowne są pojedynczymi wyrazami zajmującymi się identyfikacją podmiotu, a identyfikatory zdaniowe to całe zdania (podrzędne) zajmujące się identyfikacją podmiotu),
- Identyfikatorem głównym jest zawsze tylko identyfikator z grupy podmiotu.

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,

... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,

... – na początku linii oznaczają, że jest ona kontynuacją linii,

, – oddziela kolejne pozycje,

/ – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,

- – oddziela NUMER parametru od jego słownego OPISU,

NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,

OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

? – oznacza, że parametr na danej pozycji może być dowolny, (nie jest brany pod uwagę)

\$nr – oznacza jakiś parametr, ten sam w każdym wyrazie (co do wartości), nr to numer parametru,

1-CZĘŚĆ MOWY: część mowy – określa część mowy, która jest identyfikatorem (identyfikatorami są wyrazy, które uznajemy za podmiot, np. Marek, ktoś, samochód),

2-PEŁNI FUNKCJĘ: funkcja – określa funkcję, którą musi pełnić grupa, do której należy wyraz,

3-W GRUPIE MUSI BYĆ: część mowy;...;część mowy – mówi jakie wyrazy muszą być w grupie, aby wyraz mógł być identyfikatorem,

4-W GRUPIE NIE MOŻE BYĆ: część mowy;...;część mowy – mówi jakich wyrazów nie może być w grupie, aby wyraz był identyfikatorem,

Przykładowy zapis:

1-CZĘŚĆ MOWY: 1-rzeczownik

2-PEŁNI FUNKCJĘ: 1-podmiot

Oznacza, że każdy rzeczownik, który jest w grupie podmiotu będzie identyfikatorem. Np. w zdaniu "Marek Nowak ma samochód", identyfikatorami będą wyrazy "Marek" i "Nowak".

2.15. LANGUAGE.LO6 – plik określający rozpoznawanie osoby gramatycznej użytej w zdaniu

Opis.

- Plik zawiera informacje na temat miejsc w zdaniu, służących określeniu użytej osoby,
- Aby zdanie zostało poprawnie zrozumiane wymagane jest określenie osoby gramatycznej.

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,
 ... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,
 ... – na początku linii oznaczają, że jest ona kontynuacją linii,
 , – oddziela kolejne pozycje,
 / – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,
 - – oddziela NUMER parametru od jego słownego OPISU,
NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,
OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.
 ? – oznacza, że parametr na danej pozycji może być dowolny, (nie jest brany pod uwagę)
 \$nr – oznacza jakiś parametr, ten sam w każdym wyrazie (co do wartości), nr to numer parametru,
1-CZĘŚĆ MOWY: opis fleksji – określa fleksję wyrazu,
2-PEŁNI FUNKCJĘ: funkcja – określa funkcję grupy, do której należy
3-DETERMINUJE OSOBĘ: numer osoby – określa numer osoby, który wynika z istnienia wyrazu o podanych parametrach w zdaniu.

Przykładowy zapis:

1-CZĘŚĆ MOWY: 2-czasownik osobowy,?,?,1-czas przeszły/3-czas
 przyszyły,?,?,?,?,1-pierwsza osoba
 2-PEŁNI FUNKCJĘ: 2-orzeczenie
 3-DETERMINUJE OSOBĘ: 1-pierwsza osoba

Oznacza, że czasownik osobowy w czasie przeszłym lub przyszłym w pierwszej osobie, pełniący funkcję orzeczenia jednoznacznie określa, że w zdaniu została użyta pierwsza osoba ("ja").

2.16. LANGUAGE.LO7 – plik z wyrazami wymagającymi ścisłej identyfikacji (nazwy własne)

Opis.

- Dane z tego pliku wykorzystywane są w procesie identyfikacji tematu zdania (temat jest tu rozumiany jako podmiot, np. "MAREK jedzie samochodem"),
- Podane w pliku wyrazy (w postaci opisów fleksyjnych) wymagają ścisłej identyfikacji,
- Praktycznie wystarczy wiedzieć, że należy tu podać parametry fleksyjne nazw własnych (zgodnie z utworzonym plikiem LANGUAGE.WED).

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,
 ... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,
 ... – na początku linii oznaczają, że jest ona kontynuacją linii,
 , – oddziela kolejne pozycje,
 / – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,
 - – oddziela NUMER parametru od jego słownego OPISU,
NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,

OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

? – oznacza, że parametr na danej pozycji może być dowolny, (nie jest brany pod uwagę)

Przykładowy prosty zapis:

1-rzeczownik,?,2-własny

Oznacza, że rzeczownik własny jest nazwą własną, tzn. rzeczowniki "Marek, Polska, Denise[®]", są nazwami własnymi i wymagają ścisłej identyfikacji.

2.17. LANGUAGE.LO8 – plik z informacjami o identyfikowaniu zaimków (rzeczownych: osobowych, dzierżawczych)

Opis.

- Dane z tego pliku wykorzystywane są w procesie identyfikacji zaimków rzeczownych osobowych (ja, ty, on...) i dzierżawczych (mój, twój, jego...),
- Tylko pełna identyfikacja tych zaimków umożliwia przeanalizowanie informacji zawartej w zdaniu, po dokonaniu rozbioru gramatycznego.

Składnia – operatory:

–na początku linii oznacza, że jest to linia komentarza, pomijana przez program,

... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,

... – na początku linii oznaczają, że jest ona kontynuacją linii,

, – oddziela kolejne pozycje,

/ – (LUB) oddziela możliwe parametry dostępne na aktualnej pozycji,

- – oddziela NUMER parametru od jego słownego OPISU,

NUMER – jest to liczba przyjmująca wartości większe lub równe 1, oznaczająca numer parametru na aktualnej pozycji,

OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

? – oznacza, że parametr na danej pozycji może być dowolny, (nie jest brany pod uwagę)

\$nr – oznacza jakiś parametr, ten sam w każdym wyrazie (co do wartości), nr to numer parametru,

1-IDENTYFIKUJ: opis fleksji zaimka+opis fleksji identyfikatora – pierwszy opis fleksji jest opisem zaimka, który ma być zidentyfikowany i zastąpiony wyrazem (identyfikatorem) o podanej fleksji identyfikatora

2-DETERMINUJE OSOBE: numer osoby – określa numer osoby, który wynika z istnienia wyrazu o podanych parametrach w zdaniu,

3-TYP: 1-OSOBOWY lub 2-DZIERŻAWCZY - określa typ zaimka.

Przykładowy zapis:

1-IDENTYFIKUJ: 10-zaimek rzeczowny,1-osobowy,?,\$,1,?,1-pierwsza osoba,\$2+1-rzeczownik,?,?,\$,1,?,\$,2

2-DETERMINUJE OSOBE: 1-pierwsza osoba

3-TYP: 1-OSOBOWY

Oznacza, że zaimek rzeczowny, osobowy w pierwszej osobie, wymaga zidentyfikowania rzeczownikiem, który będzie miał tę samą liczbę, rodzaj i

przypadek, co zaimek użyty w zdaniu. Dodatkowo, wyraz ten musi identyfikować rozmówcę, który jest nadawcą występującym w pierwszej osobie.

2.18. LANGUAGE.WGN, LANGUAGE-???.WGN – pliki do generowania form fleksyjnych (drugi słownik)

Opis.

- Dane w tych plikach opisują, w jaki sposób program może generować różne formy fleksyjne na bazie wyrazów podstawowych, umieszczonych w tych plikach.
- Jest plik *.wgn (plik główny), w którym są nazwy plików, w których umieszczone są właściwe informacje słownikowe (pliki słownikowe – mogą być dowolnie nazwane, choć sugeruje się: *-???.wgn, gdzie ??? – to skrót typu: "przym." dla pliku z przymiotnikami).

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,
 , – oddziela kolejne pozycje,
 = – oznacza początek symbolicznego opisu formy fleksyjnej,
 - – na początku linii – oznacza początek nowej grupy wyrazów zakończonych podanymi po tym znaku literami, w opisie formy fleksyjnej po tym znaku umieszczana jest końcówka zastępcza, odpowiednia dla danej formy fleksyjnej,

Przykładowy zapis w pliku głównym:

```
# WordGenerator Vocabulary Files
# czasowniki (ponad 588tys.)
polski-czas.wgn
# rzeczowniki (ponad 376tys.)
polski-rzecz.wgn
```

Oznacza, że wczytane mają zostać pliki: "polski-czas.wgn" i "polski-rzecz.wgn". Dodatkowo w komentarzach zawarto informacje o tych plikach.

Przykładowy zapis w pliku słownikowym:

```
#
# *** -cja ***
# RZECZOWNIK, ZENSKI, MIANOWNIK, LICZBA POJEDYNCZA,
# POSPOLITY, NIEOSOBOWY
#
-cja
#l. poj. r. z.
=ABAAB -cja,-cji,-cję,-cja,-cji,-cjo
#
inteligencja
pólinteligencja
```

Zapis ów pokazuje, w jaki sposób należy odmieniać wyrazy zakończone na -cja. W linii =ABAAB... w szczegółowy sposób podano jak zmienia się końcówka -cja zależnie od formy przypadku. Jest to przykład dotyczący form przypadku dla liczby pojedynczej i rodzaju żeńskiego. Następnie, po przedstawieniu sposobu tworzenia odmian, następuje lista wyrazów, które mogą być w ten sposób odmieniane.

Symboliczny opis: "ABAAB" jest zakodowaną informacją, którą można odczytać za pomocą pliku *.wed lub korzystając z programu WordEditor. Tworzenie tego opisu należy realizować w ten sam sposób – tworząc listę parametrów na podstawie pliku *.wed lub posłużyć się programem WordEditor.

Opierając się na pliku *.wed opis symboliczny tworzy się w następujący sposób. Oto fragment pliku *.wed:

```
1-rzeczownik,...
...1-osobowy/2-nieosobowy,...
...1-pospolity/2-własny,...
...1-liczba pojedyncza,...
...1-rodzaj męski/2-rodzaj żeński/3-rodzaj nijaki,...
...(POS) 1-mianownik/2-dopełniacz/3-celownik/4-biernik/5-narzędnik/6-
miejscownik/7-wołacz
```

Jeśli chcemy utworzyć opis symboliczny dla rzeczownika osobowego, pospolitego, w liczbie pojedynczej, w rodzaju nijakim i formie przypadku celownika, to wystarczy zamienić wszystkie '1' na 'A', '2' na 'B' itd. i następnie wybrać odpowiedni zestaw: AAAAC,,,,,, – gdzie przecinki oznaczają istnienie 7 parametrów pozycyjnych.

2.19. LANGUAGE.FIL – plik opisujący proces filtrowania zdania ze zbędnych części

Opis.

- W pliku opisane są zasady filtrowania wprowadzonych zdań ze zbędnych zwrotów, np. ze zdania "myślę, że człowiek ma duszę" wycięte może zostać "myślę, że" i program będzie analizował zdanie "człowiek ma duszę",
- Informacje zawarte w pliku umożliwiają również prostą modyfikację zdania, przed przystąpieniem do właściwej analizy gramatycznej,

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,

... – na końcu linii oznaczają, że jest ona kontynuowana w następnej linii,

... – na początku linii oznaczają, że jest ona kontynuacją linii,

, – oddziela kolejne pozycje,

/ – (LUB) oddziela wiele możliwości alternatywnych,

- – oddziela NUMER parametru od jego słownego OPISU,

NUMER – jest to liczba przyjmująca wartości większe lub równe 1,

OPIS – słowny opis zawierający zwykle informację o charakterze pozycji i typie parametru oznaczonego przez dany NUMER, np. 1-RZECZOWNIK, 1-rodzaj męski itp.

1-ZNAJDŹ: 'ciąg 1'/'ciąg 2'/'...'/'ciąg n' – (znak '/' oznacza LUB) w tym miejscu podawane są zwroty, których odnalezienie w zdaniu jest powodem realizowania określonej operacji (patrz poniżej),

2-ARGUMENT: 'opcjonalny ciąg argumentu' – jest to miejsce na parametr, który może być potrzebny dla wykonania wybranej operacji,

3-OPERACJA: <operacja-numer> – określa operację, która zostanie wykonana w przypadku znalezienia ciągu zdefiniowanego w ZNAJDŹ, operacje:

1-SKASUJ ZNALEZIONĄ CZĘŚĆ – kasuje znaną część,

2-ZAMIEŃ ZNALEZIONĄ CZĘŚĆ NA ARGUMENT – zastępuje argumentem,

3-ZAMIEŃ CAŁE ZDANIE NA ARGUMENT – zastępuje całe zdanie argumentem,

Przykładowy zapis:

1-ZNAJDŹ: ' '
 2-ARGUMENT: ''
 3-OPERACJA: 2-ZAMIEŃ ZNALEZIONĄ CZĘŚĆ NA ARGUMENT

Oznacza, że każde dwa znaki spacji zostaną zamienione na pojedynczy znak spacji.

Natomiast inny zapis:

1-ZNAJDŹ: 'myślę, że'/'myślę że'
 3-OPERACJA: 1-SKASUJ ZNALEZIONĄ CZĘŚĆ

Mówi, że ze zdania bezwzględnie skasowane zostaną wszystkie zwroty "myślę, że" i "myślę że". Tak spreparowane zdanie zostanie poddane dalszej analizie.

2.20. LANGUAGE.CATALOG – plik z wersjami językowymi komunikatów programu

Opis.

- W tym pliku znajdują się komunikaty programu w danej wersji językowej,
- W przypadku braku tego pliku, program będzie korzystał z ustawień domyślnych i komentarze będą w języku polskim.

Składnia – operatory:

– na początku linii oznacza, że jest to linia komentarza, pomijana przez program,

NUMER – jest to liczba oznaczająca numer komunikatu, po liczbie może nastąpić opis słowny, który zostanie potraktowany jako zwykły komentarz,

KOMUNIKAT – komunikat, który zostanie wyświetlony przez program; w komunikacie często wykorzystywane są dodatkowe informacje, wstawiane przez program:

%s – wstawiony zostanie ciąg tekstowy,

%d – wstawiona zostanie liczba.

Parametry te są odpowiednikami parametrów w funkcji printf() języka C.

Przykładowy zapis:

```
10 MSG_PROGNAME
Denise-MYTH %s Artificial Intelligence
```

Oznacza, że program zamiast komunikatu domyślnego numer 10 (mówiącego o nazwie programu) wyświetli komunikat "Denise-MYTH %s Artificial Intelligence", gdzie w miejsce %s zostanie wpisana wersja programu.

Tekst i skład:

Szymon Jessa & Marek Kasperski