

Analizator morfologiczny SAM-95  
opis użytkowy<sup>1</sup>

Krzysztof Szafran  
kszafran@mimuw.edu.pl

TR 96-05 (226)  
maj 1996

---

<sup>1</sup> Praca częściowo finansowana przez grant KBN nr 8 S503 032 07 pt. „Analizator morfologiczno-syntaktyczny dla obszernego podzbioru języka polskiego” realizowany w Instytucie Informatyki UW w latach 1994-96 pod kierunkiem dra hab. Janusz S. Bienia.

Instytut Informatyki  
Uniwersytetu Warszawskiego  
ul. Banacha 2  
02-097 Warszawa  
telefon: (48-22) 658-31-65  
fax: (48-22) 658-31-64

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Przeznaczenie programu</b>	<b>1</b>
<b>3</b>	<b>Wymagania sprzętowe i programowe</b>	<b>2</b>
<b>4</b>	<b>Instalacja</b>	<b>2</b>
4.1	Wersja wykonywalna . . . . .	2
4.2	Wersja źródłowa . . . . .	2
<b>5</b>	<b>Komunikacja z użytkownikiem — listy poleceń</b>	<b>3</b>
5.1	Polecenia poziomu podstawowego . . . . .	4
5.2	Polecenia poziomu obsługi bazy . . . . .	4
5.3	Polecenia poziomu analizy . . . . .	5
<b>6</b>	<b>Kwalifikatory gramatyczne</b>	<b>8</b>
6.1	Oznaczenia klas leksemów . . . . .	8
6.2	Kwalifikatory wykładników form analizowanych . . . . .	9
6.2.1	Leksemy nieodmienne . . . . .	9
6.2.2	Leksemy rzeczownikowe . . . . .	9
6.2.3	Leksemy przymiotnikowe . . . . .	10
6.2.4	Leksemy liczebnikowe . . . . .	11
6.2.5	Leksemy zaimkowe . . . . .	13
6.2.6	Leksemy czasownikowe . . . . .	14
6.2.7	Leksemy pseudoprzymiotnikowe . . . . .	15
6.2.8	Leksemy przysłówkowe . . . . .	15
6.2.9	Leksemy przyimkowe odmienne . . . . .	15
<b>7</b>	<b>Lista zbiorów związanych z analizatorem SAM-95</b>	<b>15</b>
<b>8</b>	<b>Bibliografia</b>	<b>16</b>

## SAM-95 morfological analyser users manual

Krzysztof Szafran

### **Abstract**

SAM-95 is a program for morfological analysis of Polish. Although the user interface is modest, it is a tool ready to be used by linguists and computer scientists.

It covers almost all the vocabulary of the largest existing Polish dictionary, i.e. the 10-volume dictionary of Doroszewski containing about 120000 entries. If needed, new entries can be added to the analyser's dictionary by a user.

SAM-95 is available on Internet by anonymous FTP. It's URL is <ftp://ftp.mimuw.edu.pl/pub/polszczyzna/SAM-95>.

# Analizator morfologiczny SAM-95

## opis użytkowy

Krzysztof Szafran

maj 1996

### 1 Wstęp

Analizator morfologiczny SAM-95 jest znacznie zmienioną wersją programu SAM-93 opisanego w pracy: Szafran, K., *Automatyczna analiza fleksyjna tekstu polskiego (na podstawie schematycznego indeksu a tergo Jana Tokarskiego)*, Wydział Polonistyki UW, Warszawa 1993. Wprowadzono w nim liczne zmiany i modyfikacje i chociaż wykazuje nadal pewne cechy programu eksperymentalnego, takie jak stosunkowo skromny interfejs użytkownika, szcątkowa kontrola i diagnostyka sytuacji błędnych, operowanie „zaszytymi na stałe” nazwami zbiorów i niektórych katalogów używanych przez program itp. stał się jednak pełnowartościowym narzędziem umożliwiającym prowadzenie analizy morfologicznej tekstów polskich i mogącym znaleźć zastosowanie w różnego typu badaniach językowych. Zarówno wybrany kompilator jak i pewne inne rozwiązania, takie jak interfejs użytkownika praktycznie niezależny od typu używanego terminala, mają umożliwić w przyszłości w miarę łatwe przeniesienie programu na inną platformę sprzętową i systemową (UNIX).

### 2 Przeznaczenie programu

Podstawową funkcją programu SAM-95 jest dokonywanie analizy morfologicznej słów tekstowych w języku polskim, to znaczy określanie — na podstawie dostępnego słowa tekstowego — formy wyrazowej, charakterystyki gramatycznej tej formy oraz postaci wykładnika formy podstawowej leksemu, do którego należy analizowana forma<sup>1</sup>.

W tym celu wykorzystuje on informacje zawarte w pracy: Tokarski, J., *Schematyczny indeks a tergo polskich form wyrazowych*, opracowanie i redakcja: Zygmunt Saloni, Wydawnictwo Naukowe PWN, Warszawa, 1993. Wykorzystanie *Indeksu Tokarskiego* prowadzi do generowania, obok faktycznie istniejących w języku polskim, również leksemów wyłącznie potencjalnych. Fakt ten jest pewnym utrudnieniem, o ile interesują nas wyłącznie leksemy potwierdzone w języku, bowiem do odrzucenia pozostałych konieczne jest wykorzystanie dodatkowego źródła informacji. Takim źródłem może być lista faktycznie istniejących leksemów języka polskiego. Z drugiej strony, możliwość tworzenia form potencjalnych w połączeniu z możliwością rozszerzania słownika istniejących leksemów powoduje, że analizator jest otwarty na wszelkie zmiany zachodzące w rozwijającym się przecież nieustannie języku.

Jako wynik prac prowadzonych w ramach tematu badawczego *Słownik gramatyczny języka polskiego* finansowanego przez KBN i kierowanego przez profesora Z.Saloniego powstała, między innymi, lista

---

<sup>1</sup>Zarówno w pracy [Szafran 93] jak też w niniejszym opisie używana jest terminologia zaczerpnięta z pracy: Saloni, Z., Świdziński, M. *Składnia współczesnego języka polskiego*, PWN, Warszawa, 1985.

wszystkich hasel zawartych w *Słowniku języka polskiego* pod redakcją W.Doroszewskiego. Na jej podstawie utworzony został słownik hasel dla analizatora SAM, zwany dalej w skrócie słownikiem. Zawiera on nieco ponad 100 tysięcy hasel.

### 3 Wymagania sprzętowe i programowe

Zarówno do kompilacji jak i wykonania programu SAM-95 niezbędny jest kompilator GCC w wersji emx<sup>2</sup> pracujący pod systemem DOS na komputerach typu IBM/PC. Wymaga on procesora AT/386 lub silniejszego oraz koprocesora 387 lub jego emulatora.

Wykorzystywane przez program zbiory zajmują około 5.5MB miejsca na twardym dysku.

Program operuje polskimi znakami w standardzie Latin2 (strona kodowa 852), co pociąga za sobą konieczność wykorzystywania odpowiedniego pakietu programowych sterowników ekranu i klawiatury. W praktyce oznacza to konieczność wyposażenia komputera w kartę VGA i odpowiedni monitor.

### 4 Instalacja

Procedura instalacji programu zależy w istotny sposób od tego, czy instalujemy program w wersji wykonywalnej, czy też dysponujemy wersją źródłową.

#### 4.1 Wersja wykonywalna

Wykonywalna wersja analizatora SAM-95 dostępna jest bezpłatnie<sup>3</sup> poprzez anonimowe ftp w katalogu /pub/polszczyzna/SAM-95 pod adresem sieciowym serwera ftp Instytutu Informatyki UW, to jest ftp.mimuw.edu.pl. W katalogu tym, obok pewnych zbiorów dodatkowych (zależających, między innymi niniejszy tekst), znajduje się zbiór `sam95.exe`. Jest to samorozpakowywalne archiwum zawierające wszystkie podstawowe zbiory niezbędne do użycia analizatora. Tak więc do jego zainstalowania wystarczy wydanie polecenia `sam95`. Po rozpakowaniu archiwum w katalogu bieżącym umieszczony zostanie program analizatora (w zbiorze o nazwie `s27.exe`) oraz w podkatalogu **BAZA** pozostałe potrzebne do pracy zbiory.

#### 4.2 Wersja źródłowa

Wersja źródłowa analizatora SAM-95 składa się z następujących elementów:

1. Wersji źródłowej programu SAM-95 — zbiór `semx2_7.c`.
2. Wersji źródłowej programu pomocniczego `makeslow` — zbiór `makeslow.c`.
3. Wersji źródłowej *Indeksu Tokarskiego* — zbiór `baza.x`.
4. Wersji źródłowej słownika — zbiór `slo_1.txt`.

---

<sup>2</sup>Kompilator ten dostępny jest bezpłatnie poprzez anonimowe ftp, między innymi, w katalogu /pub/systems/os2/emx-0.8h/, pod adresem ftp.uni-stuttgart.de.

<sup>3</sup>Pod warunkiem niewykorzystywania jej do celów komercyjnych.

Jak zostało wspomniane wcześniej program SAM-95 (oraz `makeslow`) operują zbiorami o ustalonych nazwach. Z tego względu zmienianie nazw wymienionych wyżej zbiorów może spowodować problemy w trakcie instalacji.

Przed przystąpieniem do instalowania analizatora wskazane jest założenie oddzielnego katalogu, w którym będzie używany i przeniesienie tam wszystkich związanych z nim zbiorów. Nazwa tego katalogu może być dowolnie wybrana przez osobę instalującą. Katalog taki będziemy dalej nazywać *katalogiem roboczym analizatora*.

Przygotowanie analizatora do pracy polega na wykonaniu następujących czynności:

1. Założeniu w katalogu roboczym analizatora podkatalogu o nazwie **BAZA**. Nazwa ta jest wpisana na stałe w tekst programu i dlatego nie może być zmieniona. Do podkatalogu tego należy przenieść zbiory `baza.x` oraz `slo_1.txt` i `makeslow.c`.
2. Skompilowanie programu analizatora. W tym celu należy wykonać instrukcję  
`gcc semx2_7.c -o sam.exe`.  
Nazwa zbioru zawierającego program wykonywalny może być zmieniona.
3. Skompilowanie programu `makeslow`. W tym celu należy w katalogu **BAZA** wykonać instrukcję  
`gcc makeslow.c -o makeslow.exe`.  
Tu również nazwa zbioru z programem w postaci wynikowej nie jest krytyczna.
4. Przygotowanie do pracy zbiorów zawierających *Indeks Tokarskiego*. W tym celu należy:
  - Uruchomić program analizatora w katalogu roboczym analizatora.
  - Po pojawieniu się napisu `SAM>` wprowadzić polecenie `b` (*baza*).
  - Po pojawieniu się zachęty `SAM baza>` wprowadzić polecenie `d` *zapisac bazę*.

Operacja tworzenia zbiorów z bazą trwa dłuższą chwilę. W trakcie jej trwania na podstawie zbioru `baza.x` tworzone są w podkatalogu **BAZA** dwa zbiory `baza.sam` oraz `baza.hsh`. Zbiór `baza.x` nie jest więcej wykorzystywany w trakcie pracy analizatora.

5. Przygotowanie do pracy słownika hasel. Do tego celu służy program pomocniczy `makeslow`. Po jego uruchomieniu w katalogu **BAZA**, zbiór `slo_1.txt` zostanie zapisany w zbiorze `slo_1.sam` w postaci wymaganej przez analizator. Ponieważ program SAM-95 wymaga, aby słownik był zapisany w zbiorze o nazwie `słownik.sam` należy następnie przemianować (lub skopiować) utworzony zbiór `slo_1.sam` na `słownik.sam`. Zbiór `slo_1.txt` nie będzie dalej wykorzystywany przez analizator.

## 5 Komunikacja z użytkownikiem — listy poleceń

Sterowanie pracą programu SAM-95 odbywa się za pomocą niewielkiego zestawu prostych, jednoliterowych poleceń. Wprowadzenie polecenia o większej liczbie znaków powoduje zignorowanie wszystkich poza pierwszym z nich. Podanie nie istniejącego polecenia nie wywołuje żadnej akcji. W każdej chwili możliwe jest wprowadzenie jako polecenia znaku `?`. Powoduje to wypisanie na ekranie listy dostępnych w danym momencie poleceń. O tym jakie polecenia są dostępne w danym momencie decyduje poziom

na jakim w danej chwili pracuje analizator. Możliwe są trzy takie poziomy. Pierwszy — podstawowy jest poziomem pomocniczym pozwalającym na przejście na jeden z właściwych poziomów pracy: poziom sterowania bazą lub poziom analizy.

O tym na jakim poziomie analizator pracuje użytkownik informowany jest poprzez treść napisu zachęty.

Napis ten może mieć jedną z trzech postaci:

1. **SAM>** — postać zachęty na poziomie podstawowym,
2. **SAM baza>** — postać zachęty na poziomie sterowania bazą,
3. **SAM analiza>** — postać zachęty na poziomie analizy.

### 5.1 Polecenia poziomu podstawowego

Na poziomie podstawowym (zachęta postaci **SAM>**) dostępne są polecenia, których znaczenie jest oczywiste. Są to polecenia:

1. **?** — polecenie wypisania listy dostępnych na tym poziomie poleceń,
2. **b** — polecenie przejścia na poziom obsługi bazy danych,
3. **a** — polecenie przejścia na poziom analizy,
4. **e,q** — polecenia zakończenia pracy programu SAM-95.

### 5.2 Polecenia poziomu obsługi bazy

Na poziomie obsługi bazy (zachęta **SAM baza>**, przejście z poziomu podstawowego poleceniem **b**) dostępne są następujące polecenia:

1. **?** — polecenie wypisania listy dostępnych na tym poziomie poleceń,
2. **i** — polecenie wypisania pewnych informacji o zbiorach bazy i słowniku,
3. **d** — polecenie utworzenia zbiorów bazy,
4. **b** — polecenie zmiany stanu przełącznika pisania do słownika,
5. **e,q** — zakończenie pracy na poziomie obsługi bazy.

Znaczenie niektórych poleceń dostępnych na tym poziomie jest — podobnie jak poprzednio — oczywiste, niektóre wymagają jednak pewnych wyjaśnień.

**Ad 2.** Polecenie to pozwala na uzyskanie pewnych informacji o trzech podstawowych zbiorach niezbędnych do poprawnej pracy analizatora. Dla każdego ze zbiorów podawana jest data i godzina jego stworzenia oraz liczba haseł dostępnych w słowniku i wielkość pozostałych zbiorów w bajtach. O ile któryś ze zbiorów jest niedostępny wypisywana jest stosowna informacja.

Polecenie **i** odgrywało znacznie większą rolę w pierwotnej wersji analizatora, gdzie baza przechowywana była w kilkudziesięciu parach zbiorów, które tworzone były pojedynczo na życzenie użytkownika.



**Ad 3.** Zapisanie bazy odbywa się zawsze do zbiorów o ustalonych nazwach `baza.sam` i `baza.hsh` (umieszczonych w podkatalogu o ustalonej nazwie BAZA) niezależnie od tego, czy istniały one poprzednio, czy też nie. Ewentualne poprzednie wersje są zamazywane już w momencie rozpoczęcia tworzenia nowych. Oba zbiory tworzone są jednocześnie i stanowią nierozzerwalną parę funkcjonalną. Oznacza to, że powinny mieć praktycznie identyczny czas powstania. W przypadku jakichkolwiek wątpliwości należy bazę odtworzyć na nowo.

Baza tworzona jest na podstawie zbioru o nazwie `baza.x`.

**Ad 4.** Polecenie to jest w zasadzie pozostałością z poprzedniej wersji analizatora. Ponieważ daje jednak pewne ciekawe możliwości, zostało zachowane.

Zbiór tekstowy zawierający *Indeks Tokarskiego* zawiera również formy przykładowe występujące w czwartej kolumnie tego indeksu. W chwili tworzenia analizatora słownik hasel znajdujących się w słowniku Doroszewskiego nie był jeszcze dostępny, dlatego też konieczne było znalezienie innego sposobu stworzenia słownika hasel faktycznie istniejących. W chwili tworzenia zbiorów bazy program tworzył z przykładów zawartych w Indeksie zaczątek takiego słownika. W połączeniu z możliwością dodawania do słownika zaproponowanych przez program rozwiązań (patrz następny paragraf) uzyskuje się mechanizm tworzenia słownika rozbudowywanego w miarę wykorzystywania analizatora.

Obecnie zapisywanie do słownika hasel wygenerowanych na podstawie danych z indeksu jest wyłączone w chwili uruchamiania analizatora. Polecenie `b` pozwala włączać (i wyłączać) odpowiedni przełącznik.

### 5.3 Polecenia poziomu analizy

Na poziomie analizy (zachęta `SAM analiza>`, przejście z poziomu podstawowego poleceniem `a`) dostępne są następujące polecenia:

1. `?` — polecenie wypisania listy dostępnych na tym poziomie poleceń,
2. `k` — polecenie wczytania z klawiatury jednej formy do analizy,
3. `a` — polecenie zanalizowania ostatnio wczytanej formy,
4. `f` — polecenie czytania tekstu do analizy ze zbioru dyskowego,
5. `p` — polecenie zmiany stanu przełącznika sterującego proponowaniem rozwiązań,
6. `b` — polecenie zmiany stanu przełącznika sterującego dopisywaniem rozwiązań do słownika,
7. `e,q` — polecenia zakończenia pracy na poziomie analizy.

Większość wymienionych poleceń wymaga pewnego komentarza. Zanim do tego przejdziemy nieco uwag ogólnych.

1. Analiza może odbywać się w dwóch istotnie różniących się między sobą trybach.

Pierwszy z nich to tryb ściśle interakcyjny. W tym trybie analizowane są pojedyncze słowa tekstowe wprowadzane z klawiatury. Wyniki analizy uwidaczniane są na ekranie (dodatkowo wpisuje się je do zbioru `wyniki.sam`).

Analiza w drugim trybie polega na analizowaniu kolejnych form czytanych ze zbioru tekstowego o nazwie `dane.sam`. Wyniki tej analizy zapisywane są wyłącznie w zbiorze `wyniki.sam`.

W każdym z wymienionych wyżej trybów analizator może pracować w jednym z dwóch podtrybów. W pierwszym lista rozwiązań zawiera wyłącznie hasła, których istnienie potwierdzone jest wystąpieniem w słowniku. W drugim podawane są również propozycje leksemów, które na liście nie występują, ale których potencjalne istnienie dopuszcza *Indeks Tokarskiego*. Ten tryb, zwłaszcza w połączeniu z trybem interakcyjnym, umożliwi kontrolę poprawności działania analizatora w przypadkach wątpliwych, a w konsekwencji eliminowanie błędów występujących w *Indeksie Tokarskiego* oraz, w połączeniu z opcją pozwalającą na uzupełnianie słownika, pozwala na rozszerzanie tego słownika.

2. Przy czytaniu form do analizy, i to zarówno wprowadzanych z klawiatury, jak też ze zbioru tekstowego na dysku, obowiązują pewne konwencje.

Jako forma do analizy interpretowany jest ciąg znaków literowych zgodnie z tabelą polskich znaków w kodzie latin2 (tzw. strona kodowa 852). Ogranicznikami kończącymi formę są znaki odstępu, końca wiersza i końca zbioru oraz znaki '[' i '%'. Dwa ostatnie znaki są jednocześnie znacznikami początku komentarza. Pozostałe znaki występujące w ciągu wejściowym są wczytywane i przepisywane do zbioru wynikowego, ale nie wchodzi w skład ciągu tworzącego formę do analizy. Np. pojawienie się na wejściu napisu *Kowalski`emu* spowoduje wczytanie formy *Kowalskiemu* natomiast napisu *abc[xyz]def* spowoduje wczytanie formy *abc* przy wprowadzaniu danych z klawiatury oraz wczytanie dwóch form *abc* i *def* przy wprowadzaniu tekstu ze zbioru.

Istnieją dwa rodzaje komentarzy. Pierwszy rozpoczyna się znakiem '%' i kończy znakiem końca wiersza. Jest to komentarz typowy dla programu składania tekstów TeX. Drugi, to ciąg znaków zawartych pomiędzy '[' i ']' — umożliwia dopisywanie na końcu formy do analizy pewnych informacji dodatkowych.

Wszelkie komentarze napotkane w danych wejściowych są przepisywane do zbioru zawierającego wyniki.

3. Wielkie litery występujące w danych wejściowych są zachowywane w zbiorze wynikowym, natomiast we wczytywanej formie wielkie litery zastępowane są małymi. Wynika to z faktu, że analizator operuje wyłącznie małymi literami. Oznacza to w szczególności, że hasło znalezione dla wykładnika formy zapisanej w zbiorze wejściowym wielką literą zostanie wypisane wyłącznie małymi literami.

4. Program analizatora umieszcza uzyskane wyniki w zbiorze **wyniki.sam** w postaci odpowiednich list. Przykładowy fragment wyników pokazany jest niżej. Ich postać ma umożliwić łatwe ich wykorzystanie przez analizator składniowy. Przy odrobinie wprawy są równie łatwo czytelne dla człowieka.

```
%% minikorpus ...
```

```
[W1; GFJP A5- ; ]
```

```
Ja %
```

```
{{ (N) < ja(Zb)+ } }%
```

```
zostałem.%
```

```
{{ (I) < zostałem(mIV)- } }%
```

```
{ (I) < zostało(nIII)- }%
```

```
{ (m1) < zostać(XII)+ }%
```

```
{ (m1) < zostać(I)- }%
```

```
{ (m1) < zostać(IX)- }%
```

```
{ () < zostałem()- }%
```

```
{*em(lG) < została(żIV)- }%
```

```
{ (4) < zostałe(A(n))- }%
{*m() < zostałe()- }%
{*m(5) < zostały(A)- }%
{*m() < zostały()- } }%
```

Jest to wynik analizy wypowiedzenia: *Ja zostałem*. wykonanej w trybie czytania ze zbioru i podtrybie podawania również leksemów niepotwierdzonych w słowniku. Mogą to być zarówno leksemy nie istniejące jak też takie, których w słowniku z jakichś powodów brakuje.

Każda analizowana forma wypisywana jest z zachowaniem kasztowości oraz znaków ignorowanych, począwszy od nowego wiersza. W kolejnych wierszach umieszczone są — w postaci listy ujętej w nawiasy klamrowe — wykładniki tekstowe form podstawowych leksemów, do których ewentualnie należy forma reprezentowana przez analizowany wykładnik. Wykładniki te — również ujęte w nawiasy klamrowe — uzupełnione są kwalifikatorami gramatycznymi ujętymi w nawiasy okrągłe, odpowiednio formy analizowanej (przed znakiem '<') oraz znalezionej leksemu. Po kwalifikatorze leksemu występuje znak '+', jeśli leksem jest potwierdzony w słowniku i '-' w przeciwnym przypadku. Niekiedy kwalifikatory formy poprzedzone są znakiem '\*', po którym następuje pewien ciąg liter odpowiadających literom końcowym analizowanego wykładnika. Taki element listy opisuje leksem znalezionej dla formy uzyskanej w wyniku odcięcia od wykładnika formy wyjściowej końcowego ciągu liter wskazanego po znaku \*. Jest to efekt wykorzystania tak zwanego wiersza (jednego lub kilku) specjalnego w *Indeksie Tokarskiego*.

Na końcu każdego wiersza wynikowego dopisywany jest znak komentarza '%'

Postać kwalifikatorów gramatycznych zarówno dla wykładników form hasłowych, jak też dla wykładników form analizowanych nie zawsze jest łatwa do zinterpretowania dla osoby nie mającej pewnej wprawy. Pelen opis przyjętych oznaczeń i ich interpretacji można znaleźć w pracy [Tokarski 93] oraz, w wersji nieco skróconej, w punkcie 6 niniejszego tekstu.

Przejdźmy do opisu niektórych poleceń.

**Ad 2.** Wczytywanie z klawiatury odnosi się do pojedynczych wykładników tekstowych form poddawanych analizie. Z technicznego punktu widzenia znaki wprowadzane są buforowane (przez procedury wejścia-wyjścia należące do tzw. *standardowej biblioteki wejścia-wyjścia* — nie przez program analizatora) i przekazywane do programu dopiero po naciśnięciu klawisza ENTER. W skład formy wchodzi znaki tylko do napotkania pierwszego znaku rozpoznawanego jako ogranicznik (patrz uwaga 2. wyżej). Po naciśnięciu ENTER analizator wypisuje tekst, który został przezeń przyjęty do analizy.

**Ad 3.** Analiza dotyczy ostatnio wczytanej pojedynczej formy. Forma analizowana pozostaje w buforze programu aż do ponownego wydania polecenia *k*, co umożliwia powtórzenie analizy bez wprowadzania formy (np. po zmianie stanu przełącznika sterującego przedstawianiem propozycji).

**Ad 5.** Program analizatora pracuje zasadniczo w trybie podawania tych form hasłowych, które zostały znalezione w słowniku. Możliwe jest rozszerzanie zbioru rozwiązań o propozycje niepotwierdzone. Wymaga to użycia polecenia *p* sterującego odpowiednim przełącznikiem. W podtrybie składania propozycji, o ile analizowany jest tekst ze zbioru, lista rozwiązań zawiera obok leksemów potwierdzonych, oznaczonych znakiem '+' również niepotwierdzone (oznaczone znakiem '-'). Przy analizie pojedynczej formy wprowadzonej z klawiatury, po przedstawieniu kolejnej propozycji program oczekuje na reakcję użytkownika, który może zlecić podawanie dalszych propozycji (wprowadzając znak spacji) albo zaakceptować ostatnią (naciskając klawisz ENTER). Powoduje to przerwanie poszukiwań i ewentualne dopisanie leksemu do słownika form hasłowych, o ile odpowiednio ustawiony jest przełącznik sterowany poleceniem *b*.

**Ad 6.** Przełącznik sterowany tym poleceniem jest do pewnego stopnia sprzężony z przełącznikiem sterującym podtrybem składania propozycji. Mianowicie polecenie włączające opcję dopisywania nowych leksemów do słownika powoduje automatycznie uruchomienie tego właśnie trybu.

Należy tu podkreślić, że dopisywanie do słownika jest możliwe *wyłącznie* w trybie analizy pojedynczych form czytanych z klawiatury. Podczas analizy tekstu ze zbioru nie działa ono niezależnie od stanu odpowiedniego przełącznika.

**Ad 7.** Zakończenie pracy na poziomie analizy. O ile w trakcie analizy dopisywane były do słownika nowe leksemy zbiór *słownik.sam* jest rozszerzany o hasła przechowywane w słowniku w pamięci programu. W trakcie tej operacji używany jest zbiór roboczy o nazwie **słownik.bak**.

## 6 Kwalifikatory gramatyczne

Ze względu na nie zawsze oczywiste znaczenie używanych przez program kwalifikatorów gramatycznych, wzorowanych na oznaczeniach użytych w trudno dostępnej pracy [Tokarski 93] poniżej przedstawimy znaczenie najważniejszych z nich.

### 6.1 Oznaczenia klas leksemów

Napis ujęty w nawiasy okrągłe występujący po wykładniku tekstowym formy podstawowej leksemu określa do jakiej klasy i ewentualnie podklasy lub grupy fleksyjnej leksem ten należy. Składa się on z dwóch części pisanych łącznie (bez odstępu). Druga część napisu, określająca podklasę, zależy od klasy, do której leksem należy.

Należy tu wyraźnie podkreślić, że wyodrębnione dla potrzeb Indeksu klasy służą wyłącznie do ogólnej charakterystyki typu odmiany leksemu. Typ ten często różni się od rodzaju gramatycznego (por. np. rzeczowniki rodzaju męskiego odmieniające się jak żeńskie — *dozorca, socjalista*).

- a) Napis pusty — leksemy nieodmienne.  
Klasa ta nie dzieli się na podklasy.
- b) Napis rozpoczynający się od **m, n, ż, b1p** — leksemy rzeczownikowe.  
Wymienione litery oznaczają odpowiednio typ odmiany *męski, nijaki, żeński, plurale tantum*. Jako druga część napisu może wystąpić liczba rzymska oznaczająca grupę deklinacyjną. Nie występuje ona dla leksemów klasy **b1p**. Wreszcie jako element końcowy może pojawić się, w nawiasach okrągłych, informacja określająca faktyczny rodzaj gramatyczny. Np. leksem **wróżbita** zostanie scharakteryzowany jako (żIV(m)).
- c) Napis rozpoczynający się literą **A** — leksemy przymiotnikowe.  
Klasa ta nie dzieli się na podklasy. Sporadycznie po literze **A** może pojawić się, w nawiasach okrągłych, jedna z liter **m, ż, n**. Oznacza ona rodzaj gramatyczny, i może wystąpić w sytuacji, gdy opisywany leksem przymiotnikowy jest rzeczownikiem.
- d) Napis rozpoczynający się literą **K** — leksemy liczebnikowe.  
Bezpośrednio po literze **K** może pojawić się jedna z liter **a, b, c, d** określająca typ odmiany leksemu liczebnikowego. Znaczenie liter jest następujące: **a** — liczebniki główne bez *dwa, oba, obydwa, trzy, cztery*; **b** — liczebniki główne wymienione wyżej; **c** — liczebniki zbiorowe; **d** — liczebniki ułamkowe.

- e) Napis rozpoczynający się literą Z — leksemy zaimkowe.  
Bezpośrednio po literze Z może pojawić się jedna z liter a, b, c określająca typ odmiany leksemu zaimkowego. Oznaczają one: a — zaimki o jednej kategorii fleksyjnej, np. *kto, my*; b — zaimki o dwóch kategoriach fleksyjnych, np. *ja, ty, się*; c — zaimki o pięciu kategoriach fleksyjnych.
- f) Napis rozpoczynający się od liczby rzymskiej z przedziału I - XII — leksemy czasownikowe.  
Klasa ta nie dzieli się na podklasy.
- g) Napis rozpoczynający się literą R — leksemy pseudoprzymiotnikowe.  
Klasa ta nie dzieli się na podklasy.
- h) Napis rozpoczynający się literą J — leksemy przysłówkowe.  
Klasa ta nie dzieli się na podklasy.
- i) Napis rozpoczynający się literą E — leksemy przyimkowe odmienne.  
Klasa ta nie dzieli się na podklasy.

## 6.2 Kwalifikatory wykładników form analizowanych

Napisy w nawiasach okrągłych umieszczone po lewej stronie znaku '<' określają charakterystykę gramatyczną analizowanej formy jako odpowiedniej formy leksemu opisanego w tym samym wierszu, po prawej stronie znaku '>'. Inaczej mówiąc znaczenie ich zależne jest bezpośrednio od charakterystyki gramatycznej tego leksemu.

### 6.2.1 Leksemy nieodmienne

Charakterystyka form dla leksemów nieodmiennych jest napisem pustym.

### 6.2.2 Leksemy rzeczownikowe

Jeśli wykładnik tekstowy odpowiada różnym formom tego samego leksemu napis zawiera ciąg odpowiednich oznaczeń podanych bez odstępów. Oznaczenia umieszczane są w kolejności tradycyjnie odpowiadającej kolejności przypadków, najpierw dla liczby pojedynczej, po czym jeśli zachodzi taka potrzeba, po literze l występują oznaczenia form liczby mnogiej. Znaczenie poszczególnych symboli jest następujące:

- l - liczba mnoga,
- N - mianownik, forma deprecjatywna,
- H - mianownik, forma niedeprecjatywna,
- G - dopełniacz,
- G' - wariantowa forma dopełniacza, np. *racyj*,
- D - celownik,
- T - biernik,
- I - narzędnik,

- L - miejscownik,
- V - wołacz,

Tak więc napis (GLN) oznacza, że słowo tekstowe może być interpretowane, co najmniej, jako wykładnik formy dopełniacza liczby pojedynczej i jako wykładnik mianownika liczby mnogiej.

Niekiedy może się zdarzyć, że podany napis nie podaje pełnej informacji w sposób jawny. Ze względu na regularne synkretyzmy form leksemów rzeczownikowych pewne oznaczenia mogą być (i zwykle są) pominięte.

Oto obowiązujące w takich wypadkach reguły:

1. Dla leksemów o odmianie nijakiej pomija się oznaczenia T i V.
2. Dla leksemów o odmianie żeńskiej, przed literą 1 pomija się oznaczenia L.
3. Dla leksemów o odmianie męskiej pomija się litery T.
4. Po literze 1 pomija się litery T i V.
5. W przypadku synkretyzmu formy deprecjatywnej i niedeprecjatywnej leksemu nie wprowadza się litery H.

### 6.2.3 Leksemy przymiotnikowe

Zakładamy, że wszystkie leksemy przymiotnikowe mają kategorie fleksyjne liczby (o wartościach: *sg*, *pl*), rodzaju (o wartościach: *m1*, *m2*, *m3*, *n*, *f*), przypadku (o wartościach: *N*, *G*, *D*, *T*, *I*, *L*) i deprecjatywności (o wartościach: *h*, *-h*). Rozkład wykładników poszczególnych form i przypisane im numery ilustruje następująca tabela<sup>4</sup>:

	<i>sg</i>					<i>pl</i>		
	<i>m1</i>	<i>m2</i>	<i>m3</i>	<i>n</i>	<i>ż</i>	<i>m1</i>	<i>-m1</i>	
						<i>h</i>	<i>-h</i>	
<i>N</i>	1	1	1	5(5')	6	9	5(5'')	5(5'')
<i>G</i>	2	2	2	2	7	10	10	
<i>D</i>	3	3	3	3	7	4	4	
<i>T</i>	2	2	1	5(5')	8(8'/8'')	10	5(5'')	
<i>I</i>	4	4	4	4	8(8'')	11	11	
<i>L</i>	4	4	4	4	7	10	10	

Napis charakteryzujący analizowaną formę leksemu przymiotnikowego składa się z fakultatywnego określenia oznaczającego formę stopnia wyższego (*com*) oraz numeru homonimicznej formy przypadkowo-liczbowo-rodzajowej. W wypadku, gdy dwie spośród form oznaczonych liczbami mają synkretyczny wykładnik w postaci tego samego słowa podane są kolejno oba numery, oddzielone przecinkiem.

Obok liczb widocznych w tabeli mogą pojawić się pewne znaki sygnalizujące możliwość istnienia, najczęściej obok regularnych, form w jakimś sensie wyjątkowych. Znaki te to (widoczne w tabeli) ' i " oraz '!' (wykrzyknik), '+' (plus) oraz litera V. Ich znaczenie jest opisane szczegółowo we wstępie do pracy [Tokarski 93].

<sup>4</sup>Ta i następne tabele pochodzą ze wstępu do pracy [Tokarski 93] i zostały wykorzystane za zgodą jej redaktora prof. dra hab. Z. Saloniego.

### 6.2.4 Leksemy liczebnikowe

Wyróżnia się cztery typy leksemów liczebnikowych:

- a — liczebniki główne, poza należącymi do typu b,
- b — liczebniki główne **dwa, oba, obydwaj, trzy, cztery**,
- c — liczebniki zbiorowe,
- d — liczebniki ułamkowe, np. **półtora** (typ d).

Sposób określenia słowa tekstowego jako wykładnika jednej z form leksemu zależy od typu leksemu liczebnikowego.

**Liczebniki typu a.** Dla liczebników typu a charakterystyka gramatyczna form, których wykładnikiem jest analizowane słowo tekstowe podawana jest w postaci liczby zdefiniowanej następująco:

- 1 — forma typu *pięć*,
- 2 — forma typu *pięciu*,
- 3 — forma typu *pięcioma*.

Rozkład wykładników powyższych form przypadkowo-rodzajowych podaje poniższa tabela:

	<i>m 1</i>	<i>-m 1</i>
<i>N</i>	2	1
<i>G</i>	2	2
<i>D</i>	2	2
<i>T</i>	2	1
<i>I</i>	2/3	2/3
<i>L</i>	2	2

Zapis *-m 1* jest skrótem oznaczającym wszystkie wartości rodzajowe z wyjątkiem *m 1*. W narzędniku występują wykładniki wariantywne (dla pewnych leksemów, np. **sześćset**, forma oznaczona 3 może nie istnieć).

**Liczebniki typu b.** Dla liczebników typu b *nie podaje się* określenia formy analizowanej, tylko napis **por.** będący odesłaniem do niniejszego opisu (lub właściwego punktu wstępu do pracy [Tokarski 93]), gdzie wypisane są odpowiednie paradygmaty. Wyjątkami są formy *czterema, trzema, czterem, trzem*, które, jak widać z poniższych tabel, są łatwo charakteryzowalne — wyłącznie przez określenie przypadku, i nie wymagają odsyłania do opisu pradygmatu. Na przykład forma *trzem* zostanie opisana jako D.

Paradygmat leksemu **dwa**:

	m1		m2, m3, n	ż
cong	-cong			
N	<i>dwaj</i>	<i>dwu/dwóch</i>	<i>dwa</i>	<i>dwie</i>
G			<i>dwu/dwóch</i>	
D			<i>dwu/dwom/dwóm</i>	
T	<i>dwu/dwóch</i>		<i>dwa</i>	<i>dwie</i>
I	<i>dwu/dwoma</i>		<i>dwu/dwoma</i>	<i>dwie/dwoma</i>
L			<i>dwu/dwóch</i>	

Symbol **cong** oznacza formę występującą w tzw. związku zgody, tj. łączącą się z niedeprecjatywną formą rzeczownika w mianowniku (np. *Dwaj chłopcy przyszli*); symbol **-cong** — formę łączącą się z formą rzeczownika w dopełniaczu (np. *Dwu chłopców przyszło*). Formy oddzielone kreską ukośną stanowią warianty swobodne.

Paradygmat leksemu **obydwa** powstaje po dopisaniu do każdego z wykładników leksemu **dwa** początkowej cząstki *oby-*.

Paradygmat leksemu **oba**:

	m1		m2, m3, n	ż
cong	-cong			
N	<i>obaj</i>	<i>obu</i>	<i>oba</i>	<i>obie</i>
G			<i>obu</i>	
D			<i>obu</i>	
T	<i>obu</i>		<i>oba</i>	<i>obie</i>
I	<i>obu/oboma</i>		<i>obu/oboma</i>	<i>obiema/oboma</i>
L			<i>obu</i>	

Paradygmat leksemu **trzy**:

	m1		--m1
cong	-cong		
N	<i>trzej</i>	<i>trzech</i>	<i>trzy</i>
G		<i>trzech</i>	
D		<i>trzem</i>	
T	<i>trzech</i>		<i>trzy</i>
I		<i>trzema</i>	
L		<i>trzech</i>	

Paradygmat leksemu **cztery** powstaje po zastąpieniu w każdym z wykładników leksemu **trzy** początkowej cząstki *trz-* przez cząstkę *czter-*.



**Liczebniki typu c.** Jako napis charakteryzujący formę liczebników zbiorowych (mają one tylko kategorię przypadku) stosuje się określenie wartości przypadkowej formy zgodne z ogólnie przyjętą symboliką: **N**, **G**, **D**, **T**, **I**, **L**.

**Liczebniki typu d.** Dla liczebników ułamkowych, np. **półtora**, **półtrzecia** itd. (mają one nie zneutralizowaną tylko kategorię fleksyjną rodzaju) podawane jest określenie wartości rodzajowej formy analizowanej: **mn** — rodzaj męski i nijaki, **ż** — rodzaju żeński.

### 6.2.5 Leksemy zaimkowe

Jak powiedziano wcześniej wyróżnia się trzy typy leksemów zaimkowych:

- typ a — zaimki o jednej kategorii fleksyjnej (np. **kto**, **my**),
- typ b — zaimki o dwóch kategoriach fleksyjnych (**ja**, **ty**, **się**),
- typ c — zaimki o pięciu kategoriach fleksyjnych (**on**).

Sposób określenia formy zależy od typu leksemu zaimkowego.

**Zaimki typu a.** Jako napis charakteryzujący formę występuje określenie wartości przypadkowej formy analizowanej zgodne z ogólnie przyjętą symboliką: **N**, **G**, **D**, **T**, **I**, **L**.

**Zaimki typu b.** Na napis charakteryzujący formę leksemu zaimkowego typu b składają się oznaczenia wartości przypadkowej formy zgodne z ogólnie przyjętą symboliką: **N**, **G**, **D**, **T**, **I**, **L**, oraz wartości kategorii akcentowości: **akc** (dla form akcentowanych) oraz **-akc** (dla form nieakcentowanych). Jeśli wykładnik tekstowy odpowiada różnym formom tego samego leksemu podaje się ciąg odpowiednich oznaczeń. Oznaczenie akcentowości odnosi się do wszystkich symboli przypadków, które po nim następują. Podanie wartości przypadku nie poprzedzonej oznaczeniem wartości kategorii akcentowości oznacza brak zróżnicowania form akcentowanej i nieakcentowanej, przy ustalonej wartości przypadku.

Na przykład, napis **LakcD** wskazuje, że analizowane słowo tekstowe może być interpretowane jako forma miejscownika (o niezróżnicowanej kategorii akcentowości) albo akcentowana forma celownikowa.

**Zaimki typu c.** Dla form zaimka **on** (jedyne zaimki typu c) nie podaje się określenia formy objaśnianej, tylko napis **por.** oznaczający odesłanie do niniejszego opisu, zawierającego paradygmat tego leksemu.

Oprócz klasycznych kategorii przypadku, liczby i rodzaju leksem **on** ma kategorie fleksyjne poprzymkowości i akcentowości. W wypisanym niżej paradygmacie formy poprzymkowe (*praep*) i niepoprzymkowe (*-praep*) zostały podane w oddzielnych wierszach — dla tych przypadków, w których są zróżnicowane. Jeśli przy ustalonych wartościach kategorii przypadku, liczby, rodzaju i poprzymkowości występuje zróżnicowanie form akcentowanych i nieakcentowanych, w danej rubryce tabeli podaje się dwie formy rozdzielone kreską pionową: po lewej stronie kreski stoi forma akcentowana, a po prawej — nieakcentowana.

Paradygmat leksemu **on**:

	sg			pl		
	m	n	ż	m1	-m1	
				h	-h	
N	<i>on</i>	<i>ono</i>	<i>ona</i>	<i>oni</i>	<i>one</i>	<i>one</i>
G praep	<i>niego -ń</i>	<i>niego</i>	<i>niej</i>	<i>nich</i>		
G -praep	<i>jego go</i>	<i>jego go</i>	<i>jej</i>	<i>ich</i>		
D praep	<i>niemu</i>	<i>niemu</i>	<i>niej</i>	<i>nim</i>		
D -praep	<i>jemu mu</i>	<i>jemu mu</i>	<i>jej</i>	<i>im</i>		
T praep	<i>niego -ń</i>	<i>nie</i>	<i>nią</i>	<i>nich</i>	<i>nie</i>	
T -praep	<i>jego go</i>	<i>je</i>	<i>ją</i>	<i>ich</i>	<i>je</i>	
I	<i>nim</i>	<i>nim</i>	<i>nią</i>	<i>nimi</i>		
L	<i>nim</i>	<i>nim</i>	<i>niej</i>	<i>nich</i>		

### 6.2.6 Leksemy czasownikowe

Napis określający wartości kategorii gramatycznych analizowanej formy jest stosunkowo skomplikowany ze względu na wysoki stopień złożoności polskiej koniugacji. Specyfikuje on formę, przy czym w pierwszej kolejności jest określony, bezpośrednio lub pośrednio, jej typ. Stosowane nazwy typów są zrozumiałe i na ogół powszechnie używane.

Konwencje oznaczeniowe są następujące:

- Formy czasu teraźniejszego (przyszłego prostego) nie są specjalnie oznaczone (brak oznaczenia). Następnie podane jest oznaczenie osoby: 1, 2 lub 3 dla liczby pojedynczej, poprzedzone literą *l* dla form liczby mnogiej.

Formy czasu teraźniejszego leksemu **być**, zbudowane nietypowo, opatruje się dodatkowo znakiem ' (np. 3' dla „jest”) — dla odróżnienia od form czasu przyszłego prostego tego leksemu (np. „będzie” oznaczone jest 3”).

- Formy czasu przeszłego oraz formy od nich morfologicznie pochodne opatrzone są pośrednim oznaczeniem typu formy poprzez określenie za pomocą symbolu literowego rodzaju i liczby:

<i>m</i>	— rodzaj męski liczby pojedynczej,
<i>ż</i>	— rodzaj żeński liczby pojedynczej,
<i>n</i>	— rodzaj nijaki liczby pojedynczej,
<i>mo</i>	— rodzaj męskoosobowy liczby mnogiej,
<i>rz</i>	— rodzaj niemęskoosobowy liczby mnogiej.

Następnie może wystąpić symbol **C** oznaczający tryb warunkowy (brak oznaczenia równoważny jest trybowi oznajmującemu) oraz oznaczenie osoby, przy czym brak tego oznaczenia oznacza osobę trzecią. W wypadku gdy opisywana forma jest homonimiczna z inną formą tego samego leksemu, po przecinku następuje opis tej formy homonimicznej.

- Formy bezokolicznika oznaczone są literą **B**.
- Formy bezosobnika (formy na *-no*, *-to*) oznaczone są literą **b**.

- Formy trybu rozkazującego oznaczone są literą **i**. Brak dalszych oznaczeń odpowiada drugiej osobie liczby pojedynczej. Wystąpienie 1 lub 2 oznacza odpowiednią formę pierwszej lub drugiej osoby liczby mnogiej.
- Formy imiesłowu przysłówkowego uprzedniego oznacza się literą **u**.
- Formy imiesłowu przymiotnikowego czynnego (współczesnego) oznacza się literą **w**, po której występuje numer wykładnika formy przymiotnikowej, podany zgodnie z tabelą w punkcie *Leksemy przymiotnikowe*.
- Formy imiesłowu przysłówkowego współczesnego oznacza się literą **w**, bez dalszych oznaczeń.
- Formy imiesłowu przymiotnikowego biernego oznacza się literą **A**, po której występuje numer wykładnika formy przymiotnikowej, podany zgodnie z tabelą w punkcie *Leksemy przymiotnikowe*.
- Formy imiesłowu przeszłego przymiotnikowego oznacza się literę **z**, po której występuje numer wykładnika formy przymiotnikowej, podany zgodnie z tabelą w punkcie *Leksemy przymiotnikowe*.
- Formy odsłownika (gerundium) oznacza się literą **g**, po której występuje określenie przypadku, podane zgodnie z systemem przyjętym dla leksemów rzeczownikowych, opisanym wyżej.

### 6.2.7 Leksemy pseudoprzymiotnikowe

Napis charakteryzujący formę leksemu pseudoprzymiotnikowego w sposób konwencjonalny określa wartości kategorii rodzaju i liczby dla opisywanej formy. Podane niżej symbole wyczerpują wszystkie możliwości opisu form leksemów pseudoprzymiotnikowych.

<i>m</i>	— rodzaj męski liczby pojedynczej,
<i>z</i>	— rodzaj żeński liczby pojedynczej,
<i>n</i>	— rodzaj nijaki liczby pojedynczej,
<i>mo</i>	— rodzaj męskoosobowy liczby mnogiej,
<i>rz</i>	— rodzaj niemęskoosobowy liczby mnogiej.

W wypadku, gdy analizowane słowo tekstowe jest wykładnikiem dwóch spośród wymienionych form wprowadza się oznaczenia obydwu, oddzielone przecinkiem.

### 6.2.8 Leksemy przysłówkowe

Formy przysłówkowe w stopniu równym nie są specjalnie oznaczone. Oznaczenie **com** pojawia się w wypadku form stopnia wyższego.

### 6.2.9 Leksemy przyimkowe odmienne

Formy leksemów tego typu mogą być oznaczone literą **e** dla form zawierających końcówkę *-e* (np. *nade*, *pode*). Formy nie mające *-e* podane są bez dodatkowego oznaczenia.

## 7 Lista zbiorów związanych z analizatorem SAM-95

1. **baza.hsh** — zbiór zawierający tablicę *hash* niezbędną do obsługi dostępu do *Indeksu*. Powinien znajdować się w podkatalogu **BAZA**.

2. `baza.sam` — zbiór zawierający *Indeks Tokarskiego* gotowy do wykorzystania przez analizator. Powinien znajdować się w podkatalogu `BAZA`.
3. `baza.x` — zbiór zawierający postać źródłową *Indeksu Tokarskiego*. Zbiór ten potrzebny jest tylko na etapie instalowania analizatora i wtedy powinien być umieszczony w podkatalogu `BAZA`.
4. `dane.sam` — zbiór tekstowy, który będzie czytany podczas pracy analizatora w trybie czytania ze zbioru.
5. `logfile.sam` — zbiór zawierający informacje o przebiegu pracy analizatora oraz o uzyskanych wynikach. Do pewnego stopnia dubluje informacje zapisywane w zbiorze `wyniki.sam`.
6. `makeslow.c` — zbiór zawierający postać źródłową programu pomocniczego przewidzianego do tworzenia słownika. Program napisany jest w języku C i przewidziany do kompilacji kompilatorem `emx-gcc`.
7. `out$$$.sam` — zbiór zawierający wyniki pośrednie, usuwany w momencie kończenia pracy przez analizator. Może pozostać w katalogu roboczym w przypadku awaryjnego przerwania pracy analizatora.
8. `semx2_7.c` — zbiór zawierający postać źródłową programu analizatora (wersja `emx_2.7`). Program napisany jest w języku C i przewidziany do kompilacji kompilatorem `emx-gcc`. Zbiór potrzebny wyłącznie na etapie instalowania analizatora.
9. `slo_1.txt` — zbiór zawierający postać tekstową słownika. Do wykorzystania przez analizator powinien zostać przetworzony programem `makeslow` i umieszczony w podkatalogu `BAZA`. Zbiór potrzebny wyłącznie na etapie instalowania analizatora.
10. `sloownik.bak` — zbiór roboczy używany w czasie rozszerzania zbioru `sloownik.sam`. Może pozostać w katalogu o ile program został przerwany awaryjnie podczas tej operacji.
11. `sloownik.sam` — zbiór zawierający gotowy do wykorzystania słownik form hasłowych.
12. `wyniki.sam` — zbiór zawierający wyniki pracy analizatora.

## 8 Bibliografia

- Doroszewski, W. (red.) (1959-1969): *Słownik języka polskiego PAN*, Wiedza Powszechna — Państwowe Wydawnictwo Naukowe, Warszawa.
- Gladney, F.F. (1994): *Jan Tokarski Redivivus*, In: *Journal of Slavic Linguistic*, vol.2, no.2, pp304-317.
- Saloni, Z., Świdziński, M. (1985): *Składnia współczesnego języka polskiego*, PWN, Warszawa.
- Szafran, K. (1993): *Automatyczna analiza fleksyjna tekstu polskiego (na podstawie schematycznego indeksu a tergo Jana Tokarskiego)*, Rozprawa doktorska, Wydział Polonistyki UW, Warszawa.
- Szpakowicz, S. (1978): *Automatyczna analiza składniowa polskich zdań pisanych*, Rozprawa doktorska, Instytut Informatyki UW, Warszawa.

Świdziński, M. (1992): *Gramatyka formalna języka polskiego*, Rozprawy Uniwersytetu Warszawskiego **349**, Wydawnictwa UW, Warszawa.

Tokarski, J. (1993): *Schematyczny indeks a tergo polskich form wyrazowych*, opracowanie i redakcja: Zygmunt Saloni, Wydawnictwo Naukowe PWN, Warszawa.

**RAPORTY**  
**INSTYTUTU INFORMATYKI UW**  
**od 1994 roku**

198. M.Iglewski, J.Madey, K.Stencel: *On Fundamentals of the Trace Assertion Method*
199. M.Benke: *Efficient Type Reconstruction in the Presence of Inheritance*
200. S. van Bakel, L.Liquori, S.Ronchi, P.Urzyczyn: *Comparing Cubes of Typed and Type Assignment Systems*
201. T.Nowicki: *A dynamical model of behavioural specifications*
202. W.M.Turski: *Behavioural specifications*
203. P.Urzyczyn: *Positive Recursive Type Assignment*
204. J.Tiuryn: *Equational Axiomatization of Bicoercibility for Polymorphic Types*
205. M.Iglewski, M.Kubica, J.Madey: *Trace Specifications of Non-deterministic Multi - object Modules*
206. L.Czaja: *Lattice of cause-effect structures and their set-theoretic representation*
207. L.Czaja: *Decomposition of cause-effect structures*
208. FMTA'95 Formal Specifications: Foundations, Methods, Tools and Applications
209. ANNUAL REPORT 1995
210. L.Czaja: *Behaviour of cause-effect structures (Properties of four semantics)*
211. A.Szałas: *Relational Calculus with Recursion Quantifiers*
212. L.Czaja: *Representing CSP-like system as cause-effect structures*
213. J.Tiuryn, M.Wand: *Untyped Lambda-Calculus with Input-Output*
214. W.Ogryczak: *On the Lexicographic Minimax Approach to Location Problems*
215. V.Pratt, J.Tiuryn: *Satisfiability of Inequalities in a Poset*
216. D.Niwiński: *Fixed points characterization of infinite behaviour of finite state systems*
217. K.Stencel: *Refined Simulation Techniques for the Trace Assertion Method*
218. M.Iglewski, M.Kubica, J.Madey, J.Mincer-Daszkiewicz, K.Stencel: *The Fun-Project: From Requirements Specification to Program Presentation*
219. J.Tiuryn, P.Urzyczyn: *The Subtyping Problem for Second-Order Types is Undecidable*
220. M.Benke: *Some Complexity Bounds for Subtype Inequalities*
221. G.Grudziński: *Intersection Types Spell Polymorphic Invariance for Strictness Analysis*
222. J.Tiuryn: *A Sequent Calculus for Subtyping Polymorphic Types*
223. W.Ogryczak: *Equitable Multiple Criteria Programming*
224. L.Czaja: *Cause-effect structure processes and their link with Mazurkiewicz's traces*
225. M.Jurdziński, M.Konarski, A.Schubert: *The EML Kit Version 1*