

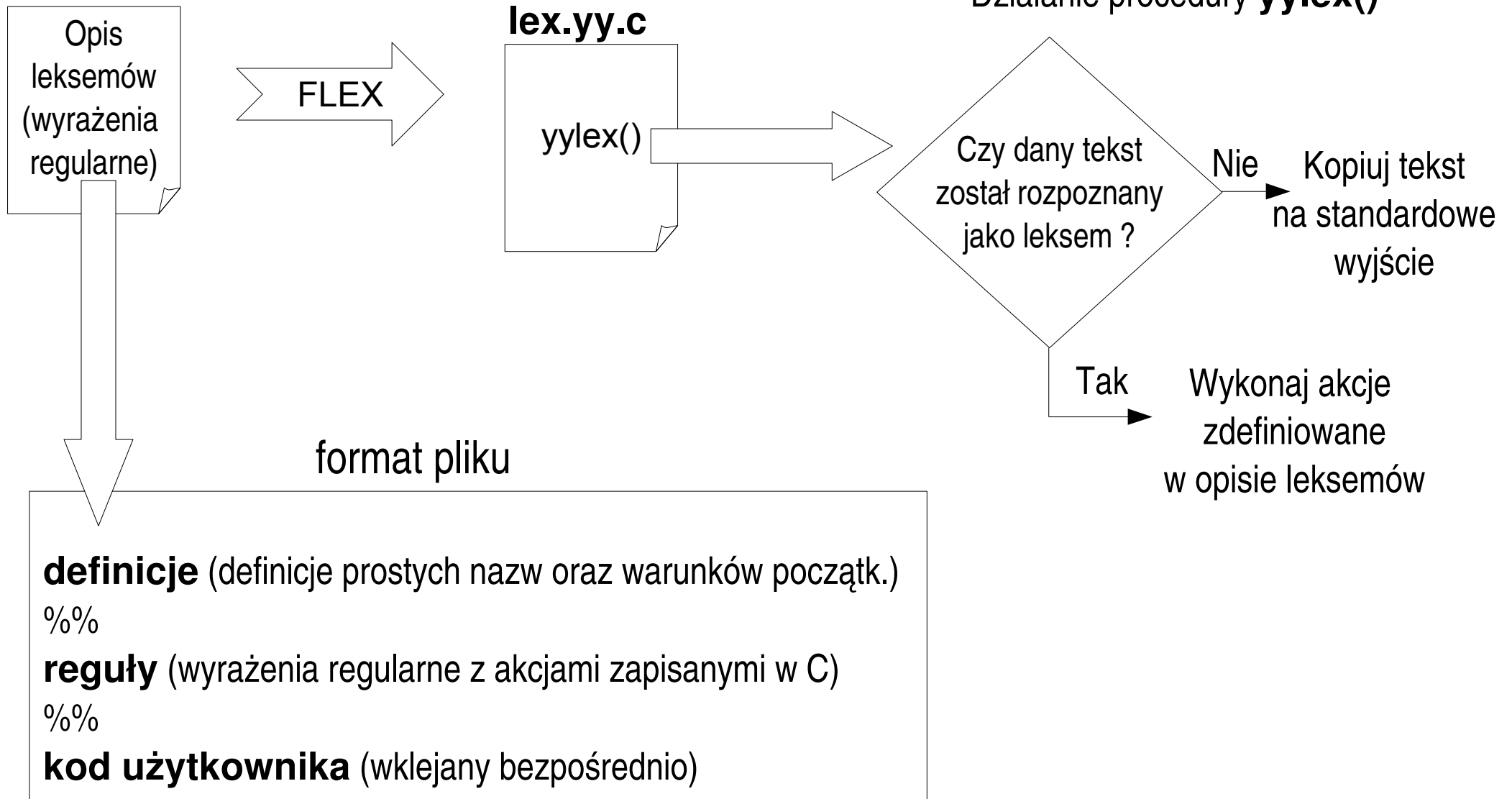
Generatory analizatorów

- Generator analizatora leksykalnego - **flex** (<http://www.gnu.org/software/flex/>)
- Generator analizatora składniowego - **bison** (<http://www.gnu.org/software/bison/>)
- Idea ogólna



- Inne generatory
 - Coco/R (<http://www.ssw.uni-linz.ac.at/Research/Projects/Coco/>) - skaner oraz parser "recursive-descent", duża ilość języków wyjściowych (C++,Java,C#)
 - ANother Tool for Language Recognition ([www.antlr.org](http://wwwantlr.org)) - skaner, parser oraz procedury przeglądania wygenerowanego drzewa wywołu

Flex



Flex c.d.

- Definicje

- "Zwykła" - pomaga użytkownikowi definiować często używane elementy np.

CYFRA [0-9]

IDENT [a-zA-Z][a-zA-Z0-9_]*

LICZBA_C {CYFRA}+

- Warunek początkowy:

%s string lub %x string

- Każdy tekst odsunięty od początku linii lub zawarty w %{ %} jest kopiowany

- Reguły - składnia bardzo podobna do popularnej składni wyrażeń regularnych

- *<warunek>wyrażenie* - wyrażenie rozpoznawane tylko wtedy, gdy skaner jest w stanie "warunek"

- <<EOF>> - oznaczenie końca pliku (wejścia)

- wyrażenia [:alnum:], [:alpha:], [:digit:] itd.

Flex c.d.

- Generowany algorytm rozpoznawania wyrażeń
 - Znajdź dopasowanie, które pobiera **najdłuższy** tekst
 - Jeżeli znalazłeś więcej niż jedno dopasowanie o takiej samej długości wybierz to, które zostało zdefiniowane w pliku jako pierwsze
 - Skopiuj tekst leksemu do zmiennej globalnej **yytext** a długość do **yylen**
 - Wykonaj zdefiniowaną akcję
- Różnica pomiędzy definicją %pointer a %array
- Akcje
 - Kawałki kodu w języku C (lub C++)
 - Specjalne: ECHO (wypisz yytext), REJECT (przejdź do następnej reguły), BEGIN(<nazwa>)
 - yymore() - dołącz tekst następnego leksemu
 - yyless(n) - wypisz leksem, ale bez n początkowych znaków
 - unput(c) - zwróć znak na wejście
 - input() - odczytaj znak z wejścia
 - yyterminate() - zakończ i wyjdź z 0

Flex c.d.

- Procedura `int yylex()` oraz jej redefinicja makrem `YY_DECL`
- **yyin** oraz **yyout**
- Warunki początkowe (start conditions) - `INITIAL` oraz `BEGIN`
 - `%s` - inclusive, `%x` - exclusive
- Przełączanie buforów wejściowych
 - `YY_CURRENT_BUFFER`
 - `YY_BUFFER_STATE yy_create_buffer(FILE *file, int size)`
 - `void yy_switch_to_buffer(YY_BUFFER_STATE new_buffer)`
 - `void yy_delete_buffer(YY_BUFFER_STATE buffer)`
- Ćwiczenie - definicja reguł dla analizatora implementowanego na pracowni

Bison

- Generuje parser typu "bottom-up" a dokładniej (Generalized LR) - w przypadku konfliktu S/R lub R/R tworzymy kopie parserów i próbujemy równolegle wykonywać analizę (funkcja **yyparse()**)
- Pojęcie wartości semantycznej (semantic value) tokenu i wyrażenia (\$)
- Lokalizacje (@)
- Format pliku wejściowego

```
%{  
Kod użytkownika  
%}  
deklaracje (symbole terminalne, nieterminalne, pierwsz. operatorów, wartości sem.)  
%%  
produkcje gramatyki (w notacji BNF)  
%%  
kod użytkownika (wklejany bezpośrednio)
```

Bison

- Deklaracje

- `%token IDENT 100` ← numer opcjonalnie

- `%token DOT "."`

- Pierwszeństwo operatorów `%left '+' '-'` `%left '*' '/'`

- Typy symboli terminalnych

```
%union { double val; symrec *tpr; }
```

```
%token <val> NUM
```

- Typy symboli nieterminalnych - `%type <val> EXPR`

- `%start PROGRAM`

- Produkcje

```
<nazwa prod>: <składowe>  
                ;
```

```
exp: term "+" term { akcja1 } |  
    term { akcja 1.5 } "-" term { akcja2 }  
    ;
```

Bison

- Akcje i symbole specjalne
 - YYABORT / YYACCEPT - natychmiastowe wyjście z błędem / bez błędu
 - YYERROR - symulacja błędu składni i rozpoczęcie procedury wydobywania
 - YYRECOVERING - wartość 1 w czasie wydobywania się z błędu
 - \$\$ - semantyczna wartość lewej strony produkcji
 - \$n - semantyczna wartość n-tej składowej prawej strony produkcji (od 1)
 - @\$, @n - jak wyżej tylko lokalizacja zamiast wartości
 - ychar - aktualny symbol na wejściu lub YYEMPTY
- Generowany algorytm analizy składniowej
- Wydobywanie się z błędów
 - yyerror(char *msg) - funkcja definiowana przez użytkownika
 - **error** - specjalny symbol terminalny do używania w produkcjach
- Konflikty
 - Shift/Reduce - wybieramy Shift, Reduce/Reduce - decyduje kolejność zapisu