AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Analiza leksykalna – 1

Języki formalne i automaty

Dr inż. Janusz Majewski
Katedra Informatyki

Zadanie analizy leksykalnej

Przykład:

We: COST := (PRICE + TAX) * 0.98

Wy: id_1 := (id_2 + id_3) * num_4

Tablica symboli:

Adres	Nazwa/wartość	Charakter	Dodatkowa informacja
1	COST	zmienna	….
2	PRICE	zmienna	….
3	TAX	zmienna	….
4	0.98	stała	….

Zadanie analizy leksykalnej

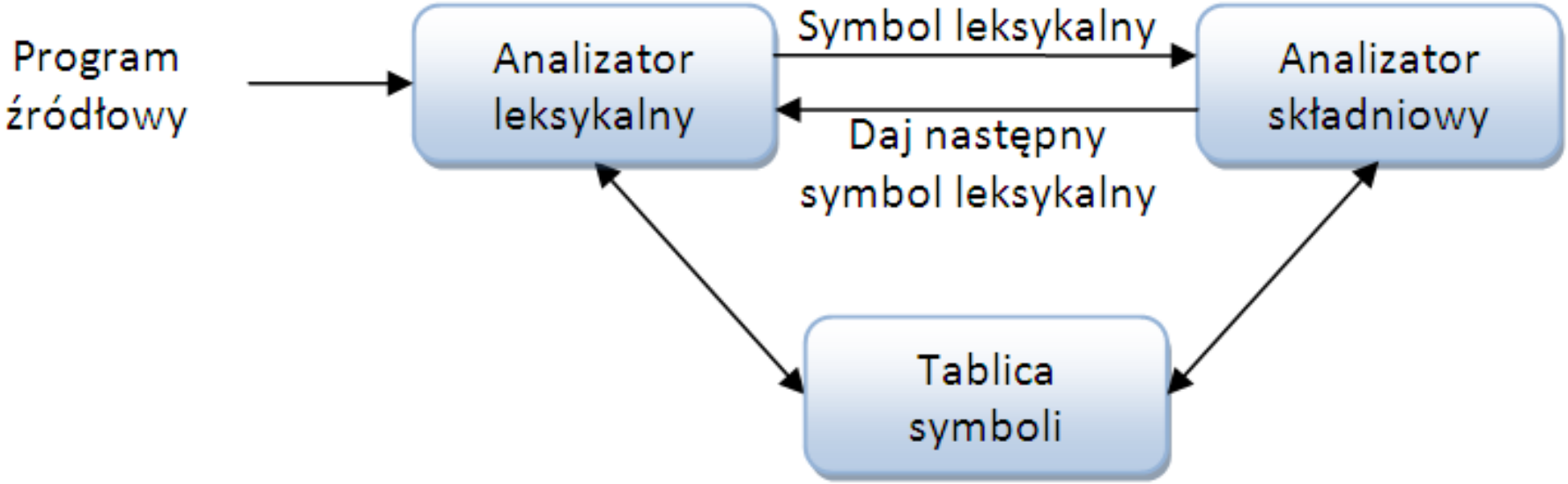
Przykład:

We: COST:= (PRICE + TAX) * 0.98

Wy: id_1 := (id_2 + id_3) * num_4

Wyjście skanera:	<=	**Wejście skanera:**
(id, 1)		COST
(equ,)		:=
(left-par,)		(
(id, 2)		PRICE
(plus,)		+
(id, 3)		TAX
(right-par,)		)
(mult,)		*
(num, 4)		0.98

Współpraca z parserem



Zadania analizatora leksykalnego

Zadania analizatora leksykalnego:

- wyodrębnianie symboli leksykalnych (tokenów)
- ignorowanie komentarzy
- ignorowanie białych znaków (spacji, tabulacji, znaków nowej linii…)
- korelowanie błędów zgłaszanych przez kompilator z numerami linii
- tworzenie kopii zbioru wejściowego (źródłowego) łącznie z komunikatami o błędach
- czasami realizacja funkcji preprocessingu, rozwijanie makrodefinicji

Rozdzielenie etapu analizy na dwie odrębne funkcje: analizę leksykalną i analizę syntaktyczną sprawia, że jedna i druga mogą być wykonywane przy użyciu bardziej efektywnych algorytmów, gdyż algorytmy te istotnie się różnią, wykorzystując inne pryncypia formalne i realizacyjne.

Podstawowe pojęcia: token, leksem, wzorzec

Przykład:

const pi2 = 6.2832;

token	leksem	wzorzec (pattern)
const (słowo kluczowe)	const	const
id	pi2	*litera* (*litera* | *cyfra*)*
relop	=	< | > | <= | >= | = | <>
num	6.2832	*cyfra*$^{+}$(. *cyfra*$^{+}$)? ((E|e) (+|–)? *cyfra*$^{+}$)?

źródło: `const pi2 = 6.2832`
leksemy: const pi2 = 6.2832
tokeny: const id relop num

Trudności w budowaniu analizatora leksykalnego

Przykład:

W niektórych językach programowania słowa kluczowe nie są zastrzeżone (FORTRAN, PL/I). W języku PL/I poprawny jest zapis:

IF THEN **THEN** THEN = ELSE ; **ELSE** ELSE = THEN;

Trudności w budowaniu analizatora leksykalnego

Przykład:

W języku FORTRAN spacje (z wyjątkiem spacji wewnątrz łańcuchów) są zawsze ignorowane. Nazwy zmiennych nie wymagają deklaracji. Typ zmiennej ustalany jest na podstawie pierwszej litery nazwy.

Porównujemy:

1) DO 5 I = 1 **.** 25 równoznaczne DO5I = 1.25
(instrukcja przypisania, DO5I – zmienna rzeczywista)

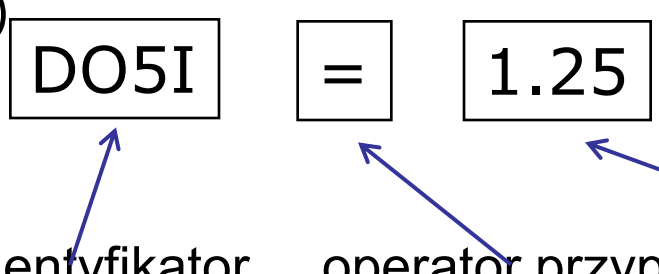
2) DO 5 I = 1**,** 25 instrukcja pętli wyliczanej typu "for", odpowiednik:

```
for I := 1 to 25 do
  begin
        ...
        ...
etykieta ...
        ...
5: end;
```

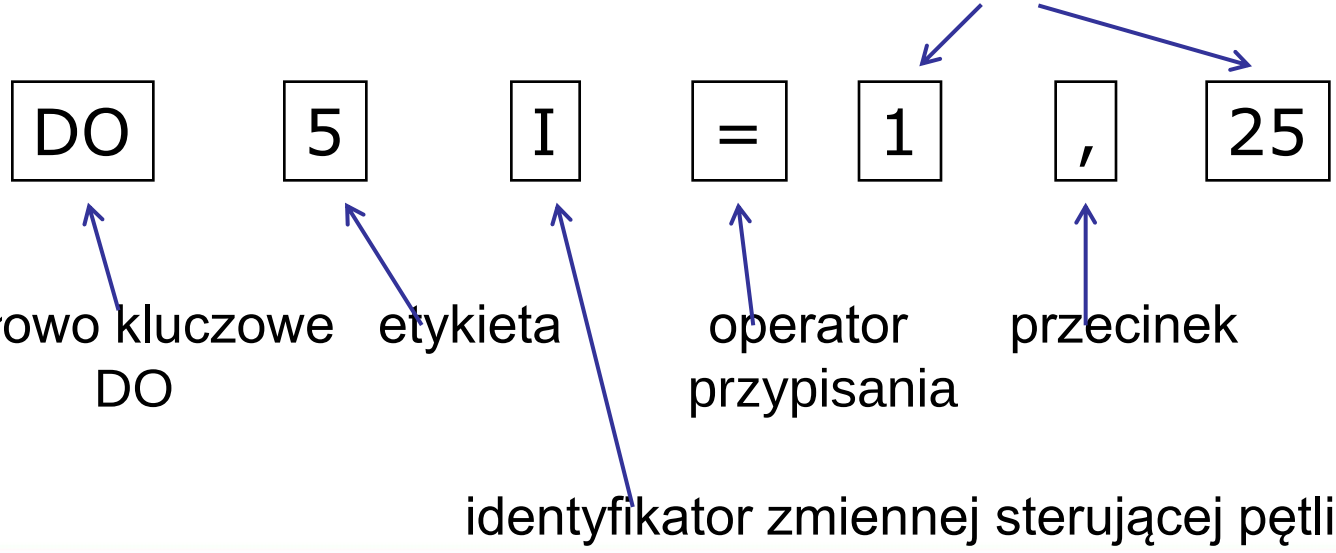
Trudności w budowaniu analizatora leksykalnego

Przykład c.d.:

1)

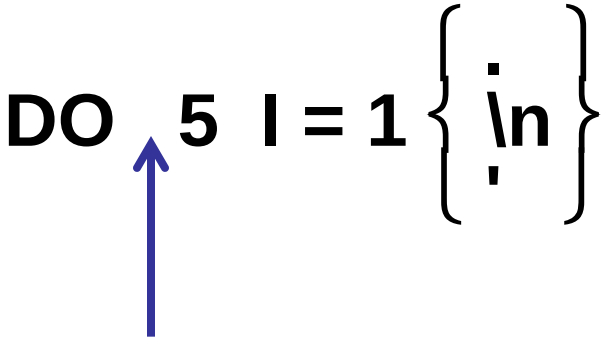


2)



Trudności w budowaniu analizatora leksykalnego

Przykład c.d.:



Po przeczytaniu znaków DO nie można dokonać uzgodnienia tokenu "Słowo kluczowe DO" dopóki nie zbada się prawego kontekstu i nie znajdzie się przecinka (wtedy rzeczywiście uzgadnia się "DO") lub kropki bądź znaku nowej linii (wtedy mamy instrukcje podstawienia).

Definicje regularne

Do opisu wzorców dla skanera stosujemy definicje regularne:

$d_1 \rightarrow r_1$

$d_2 \rightarrow r_2$

.....

$d_n \rightarrow r_n$

gdzie:

d_i - unikalna nazwa

r_i - wyrażenie regularne nad symbolami alfabetu $\Sigma \cup \{d_1, d_2, ..., d_{i-1}\}$

Przykład definicji regularnych (1)

Stałe bez znaku w Pascal'u:

cyfra → *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9*

cyfry → *cyfra* *cyfra* *

część-ułamkowa → . *cyfry* | ε

wykładnik → (*E* | *e*) (+ | – | ε) *cyfry* | ε

num → *cyfry* *część-ułamkowa* *wykładnik*

Rozszerzenie zbioru operatorów

Dla ułatwienia wprowadza się nowe operatory w wyrażeniach regularnych, np.:

w^+ - oznacza jedno lub więcej wystąpień wzorca w

$$w^+ = w\, w^*$$

$w?$ - oznacza zero lub jedno wystąpienie wzorca w

$$w? = w \mid \varepsilon$$

Przykład definicji regularnych (2)

Stałe bez znaku w Pascal'u zapisane po rozszerzeniu zbioru operatorów w wyrażeniach regularnych:

cyfra → *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9*

cyfry → *cyfra* $^{+}$

część-ułamkowa → (**.** *cyfry*) ?

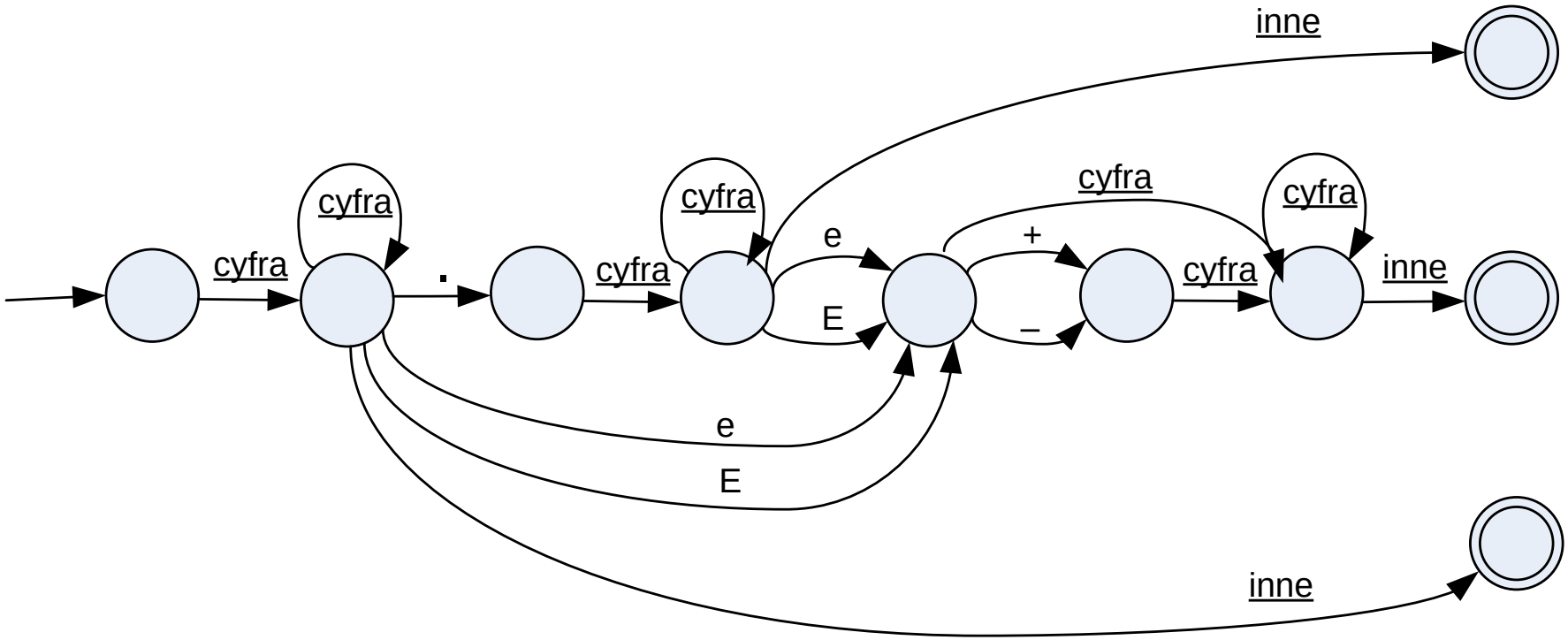
wykładnik → ((*E* | *e*) (+ | –) ? *cyfry*) ?

num → *cyfry* *część-ułamkowa* *wykładnik*

Diagramy przejść (1)

Liczba bez znaku w Pascalu

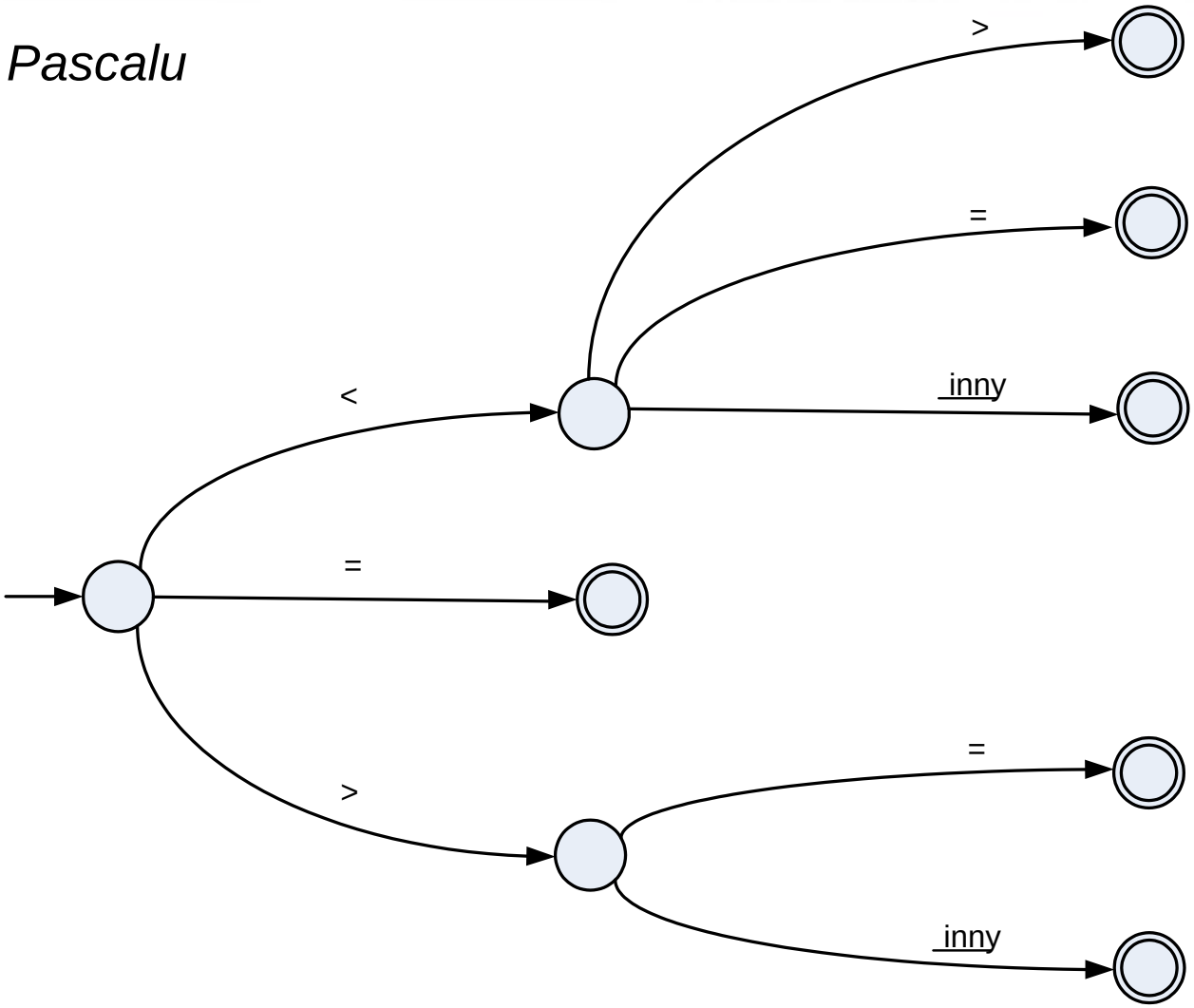
cyfra$^+$ (. cyfra$^+$)? ((e|E) (+|–)? cyfra$^+$)?



Diagramy przejść (2)

Operatory relacyjne w Pascalu

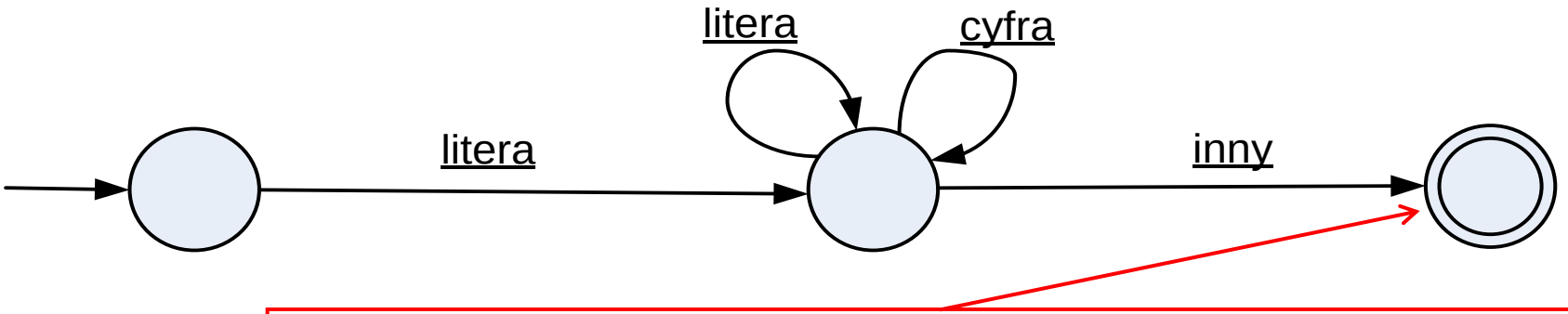
< | <= | <> | = | >= | >


>
=
inny
<
=
=
inny
>

Diagramy przejść (3)

Identyfikatory

litera (litera | cyfra)*



„oddaj” ostatni przeczytany symbol na wejscie;
sprawdź, czy leksem to słowo kluczowe;
jeśli tak – zwróć odpowiednie słowo kluczowe;
jeśli nie – sprawdź, czy identyfikator jest
już w tablicy symboli;
jeśli jest – zwróć adres jego pozycji;
jeśli nie ma – utwórz nową pozycje
i wpisz identyfikator do tablicy symboli.