



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: $\underline{id}_1 := (\underline{id}_2 + \underline{id}_3) * \underline{num}_4$

Tablica symboli:

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

Zadanie analizy leksykalnej

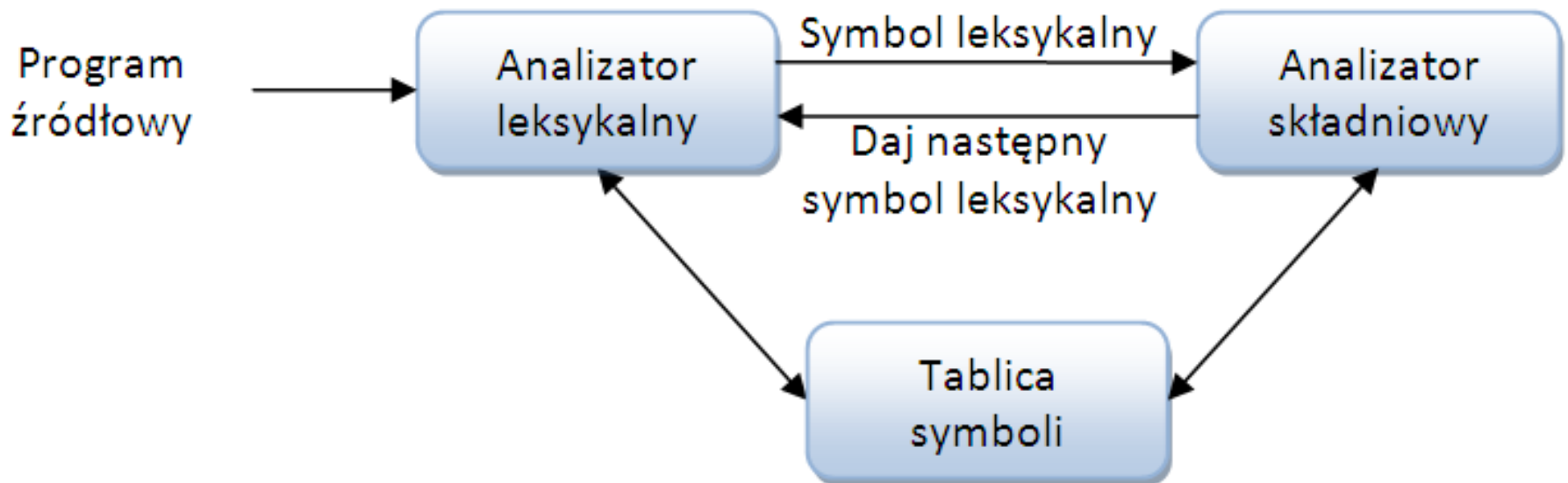
Przykład:

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

Wy: $\underline{id}_1 \underline{:=} (\underline{id}_2 + \underline{id}_3) * \underline{num}_4$

Wyjście skanera:	<=	Wejście skanera:
(<u>id</u> , 1)		COST
(<u>equ</u> ,)		:=
(<u>left-par</u> ,)		(
(<u>id</u> , 2)		PRICE
(<u>plus</u> ,)		+
(<u>id</u> , 3)		TAX
(<u>right-par</u> ,))
(<u>mult</u> ,)		*
(<u>num</u> , 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)
<u>const</u> (słowo kluczowe)	const	const
<u>id</u>	pi2	<i>litera (litera cyfra)*</i>
<u>relop</u>	=	< > <= >= = <>
<u>num</u>	6.2832	<i>cyfra⁺ (. cyfra⁺)? ((E e) (+ -)? cyfra⁺)?</i>

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.

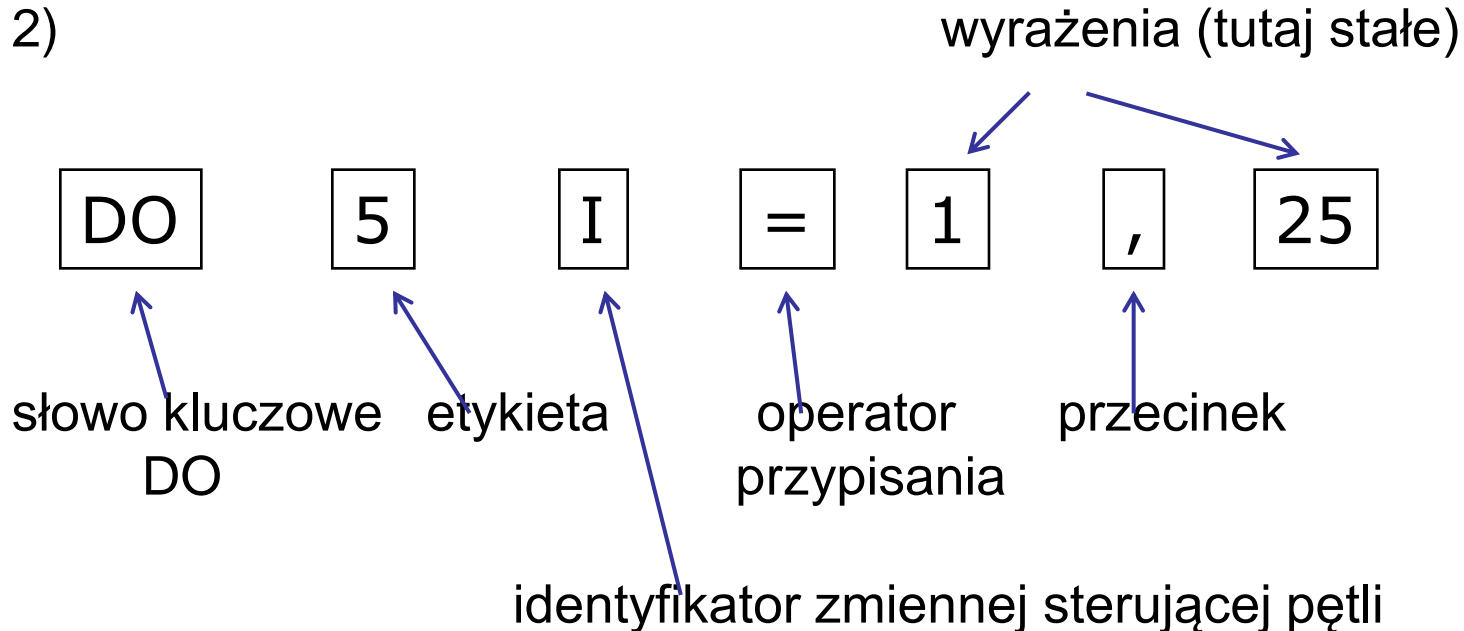
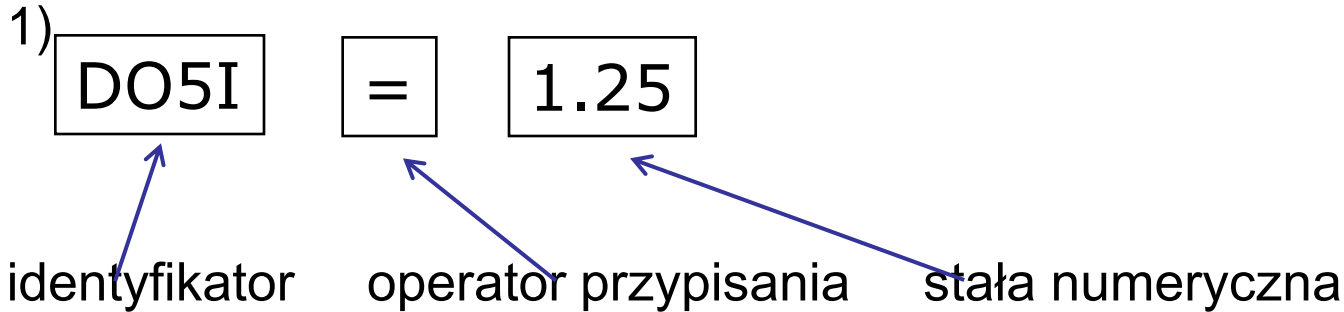
Porównujemy:

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.:



Trudności w budowaniu analizatora leksykalnego

Przykład c.d.:

DO 5 l = 1 { \n }



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

$$\Sigma \cup \{d_1, d_2, \dots, d_{i-1}\}$$

Przykład definicji regularnych (1)

Stałe bez znaku w Pascal'u:

cyfra $\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

cyfry $\rightarrow \underline{\text{cyfra}} \underline{\text{cyfra}}^*$

część-ułamkowa $\rightarrow \cdot \underline{\text{cyfry}} \mid \varepsilon$

wykładnik $\rightarrow (E \mid e) (+ \mid - \mid \varepsilon) \underline{\text{cyfry}} \mid \varepsilon$

num $\rightarrow \underline{\text{cyfry}} \underline{\text{część-ułamkowa}} \underline{\text{wykładnik}}$

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 $\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

cyfry \rightarrow cyfra $^+$

część-ułamkowa $\rightarrow (\cdot \text{ cyfry }) ?$

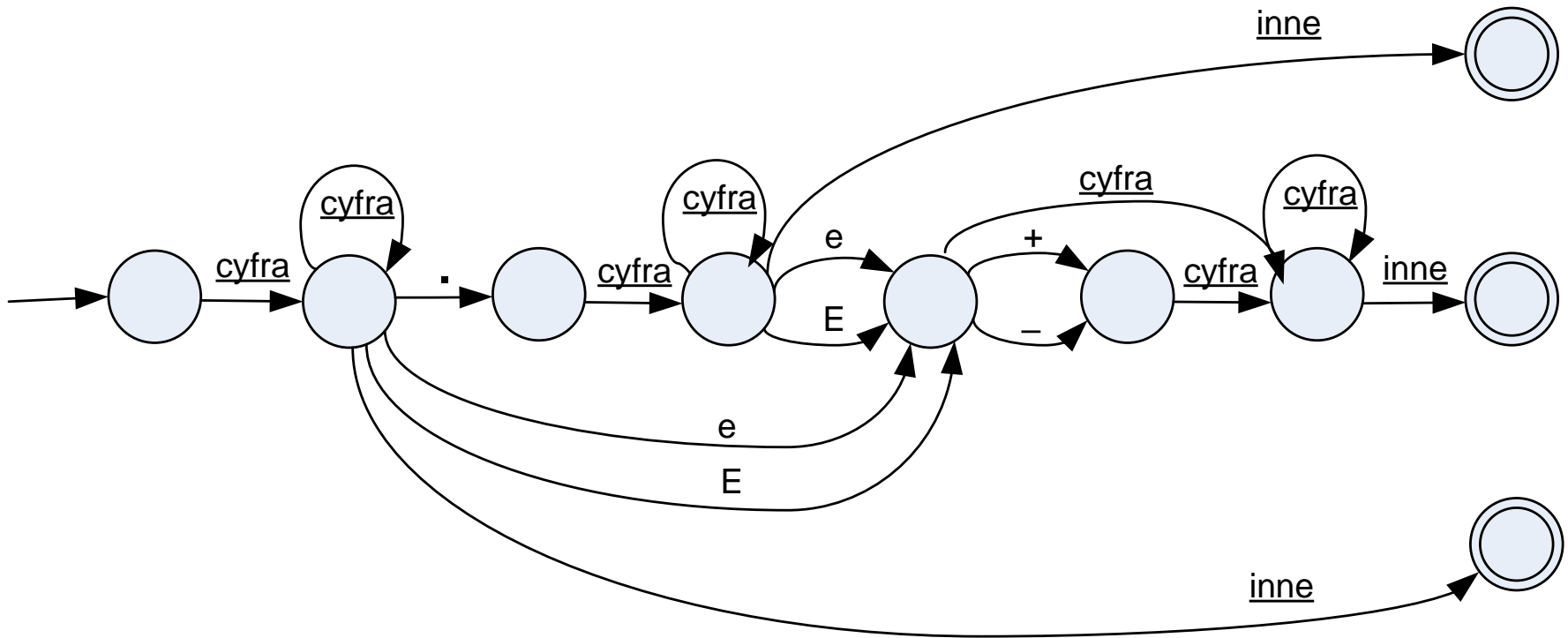
wykładnik $\rightarrow ((E \mid e) (+ \mid -) ? \text{ cyfry }) ?$

num \rightarrow cyfry część-ułamkowa wykładnik

Diagramy przejść (1)

Liczba bez znaku w Pascalu

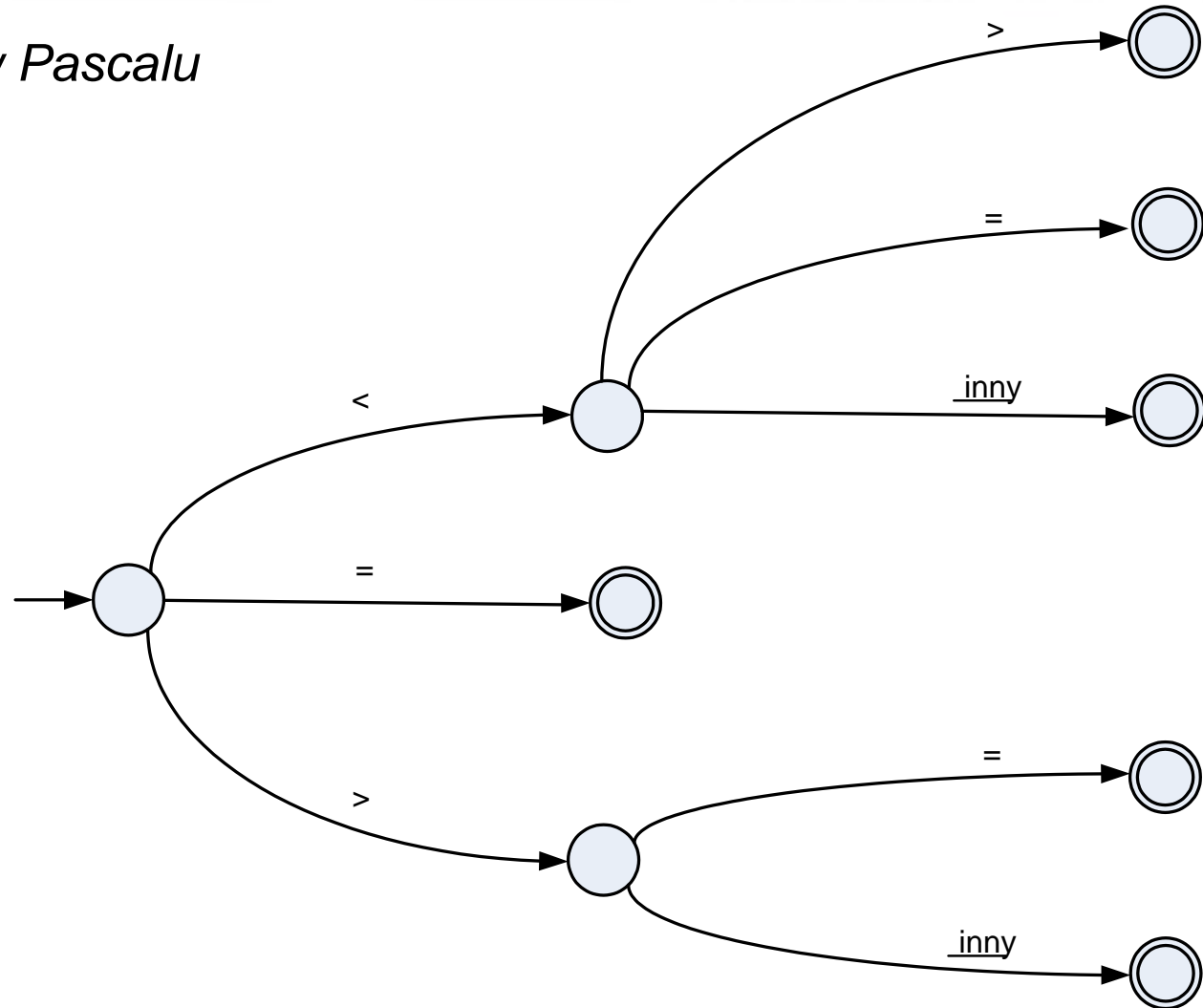
cyfra⁺ (cyfra⁺)? ((e|E) (+|-)? cyfra⁺)?



Diagramy przejść (2)

Operatory relacyjne w Pascalu

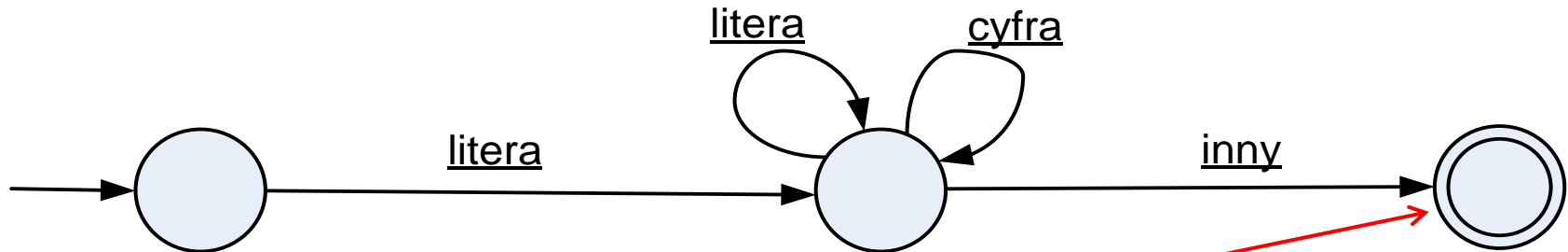
< | <= | <> | = | >= | >



Diagramy przejść (3)

Identyfikatory

litera (litera | cyfra)*



„oddaj” ostatni przeczytany symbol na wejście;
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.