

Metody Kompilacji

Wykład 8

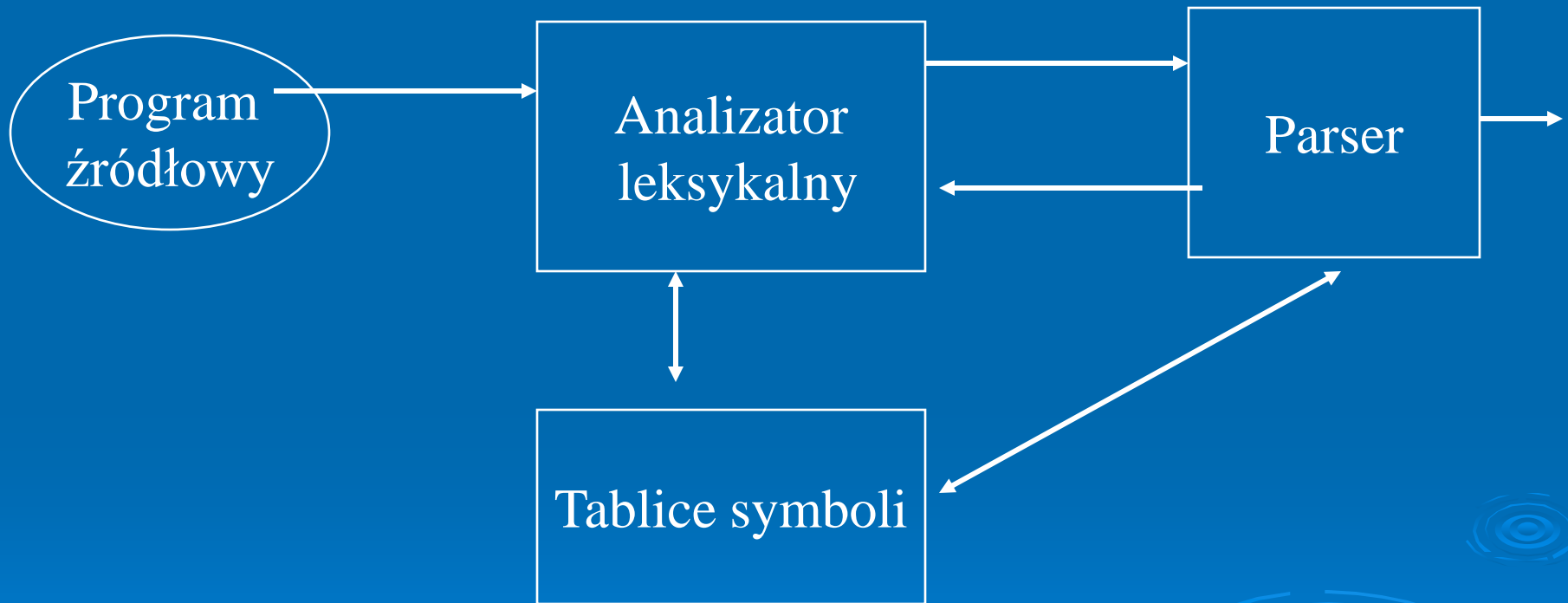
Analiza Syntaktyczna cd

Analiza Syntaktyczna

Wstęp

- Parser dostaje na wejściu ciąg tokenów od analizatora leksykalnego i sprawdza: czy ciąg ten może być generowany przez gramatykę.

Analiza Syntaktyczna



Analiza Syntaktyczna

➤ Wstęp

- Istnieją trzy główne rodzaje parserów dla gramatyk: uniwersalny, analiza zstępująca (*top-down*) oraz analiza wstępująca (*bottom-up*).

Analiza Syntaktyczna

- **Wstęp**
- Metoda uniwersalna pozwala na zastosowanie dowolnej gramatyki.
- Takie ogólne metody jednak mają dużą złożoność obliczeniową i nie mogą być stosowane do tworzenia kompilatorów przemysłowych.

Analiza Syntaktyczna

➤ Wstęp

- Metody stosowane do tworzenia kompilatorów przemysłowych są oparte na analizie zstępującej (*top-down*) oraz analizie wstępującej (*bottom-up*).

Analiza Syntaktyczna

Wyprowadzenia

- Rozważmy następującą gramatykę

$$E \rightarrow E+E \mid E*E \mid -E \mid (E) \mid \text{id}$$

- Zamianę symbolu E przez napis $-E$ zapisujemy jako:

$$E \Rightarrow -E$$

i czytamy: " E wyprowadza $-E$."

Analiza Syntaktyczna

- Wyprowadzenia
- Inne przykłady wyprowadzeń:

$$E * E \Rightarrow (E) * E \quad \text{lub}$$

$$E * E \Rightarrow E * (E)$$

Analiza Syntaktyczna

➤ Wyprowadzenia

Możemy wziąć pojedynczy symbol E i wielokrotnie stosować produkcje w dowolnej kolejności.

Na przykład,

$$E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(\mathbf{id})$$

Analiza Syntaktyczna

➤ Wyprowadzenia

- Załóżmy, że nieterminal A znajduje się w środku napisu:

$$\alpha A \beta,$$

gdzie α and β są to dowolne podnapisy.

- Przypuśćmy, że $A \rightarrow \gamma$ jest to produkcja, wtedy wyprowadzenie ma postać:

$$\alpha A \beta \Rightarrow \alpha \gamma \beta.$$

- Symbol \Rightarrow oznacza wyprowadzenie bezpośrednie.

Analiza Syntaktyczna

➤ Wyprowadzenia

➤ Gdy mamy sekwencję wyprowadzeń:

$$a_1 \Rightarrow a_2 \Rightarrow \dots \Rightarrow a_n,$$

mówimy, że a_1 wyprowadza a_n co można zapisać również jako

$$a_1 \Rightarrow^* a_n$$

gdzie symbol \Rightarrow^* oznacza „zero lub więcej wyprowadzeń”.

Analiza Syntaktyczna

➤ Wyprowadzenia

➤ Będziemy stosować również

symbol $\Rightarrow+$, który oznacza „co najmniej jedno lub więcej wyprowadzeń”.

Analiza Syntaktyczna

➤ Wyprowadzenia

- Jeśli $S \Rightarrow^+ a$, gdzie S jest to symbol startowy w gramatyce G , to mówimy, że a jest to forma zdaniowa gramatyki G .
- Uwaga: forma zdaniowa może zawierać jednocześnie symbole terminalne jak i nieterminalne .

Analiza Syntaktyczna

➤ Wyprowadzenia

- Zdaniem gramatyki G jest forma zdaniowa, której wszystkie symbole są terminalami.
- Język, generowany przez gramatykę G , jest to zbiór wszystkich zdań generowanych przez G .

Analiza Syntaktyczna

➤ Rodzaje Wyprowadzeń

1. W wyprowadzeniach lewostronnych(*leftmost derivations*), zawsze wybieramy do zamiany pierwszy nieterminal z lewej strony formy zdaniowej.

➤ Jeśli $\alpha \Rightarrow \beta$ w kroku, w którym zamieniany jest skrajnie lewy nieterminal z α , to wyprowadzenie takie zapisujemy: $\alpha \Rightarrow_{lm} \beta$.

Analiza Syntaktyczna

➤ Rodzaje Wyprowadzeń

2. W wyprowadzeniach prawostronnych (*right most derivations*), zawsze wybieramy do zamiany pierwszy nieterminal z prawej strony formy zdaniowej.

Jeśli $\alpha \Rightarrow \beta$ w kroku, w którym zamieniany jest skrajnie prawy nieterminal z α , to wyprowadzenie takie zapisujemy: $\alpha \Rightarrow_{rm} \beta$.

Analiza Syntaktyczna

Zbiór First

- Zbiór $FIRST(\alpha)$, gdzie α jest dowolnym ciągiem symboli gramatycznych, jest to zbiór terminali, od których zaczynają się ciągi wyprowadzane z α .
- Jeśli $\alpha \Rightarrow^* \varepsilon$, to ε należy do $FIRST(\alpha)$.

Zbiór First : Przykład

1. $exp \rightarrow term\ exp'$

2. $exp' \rightarrow addop\ term\ exp' \mid \varepsilon$

3. $addop \rightarrow + \mid -$

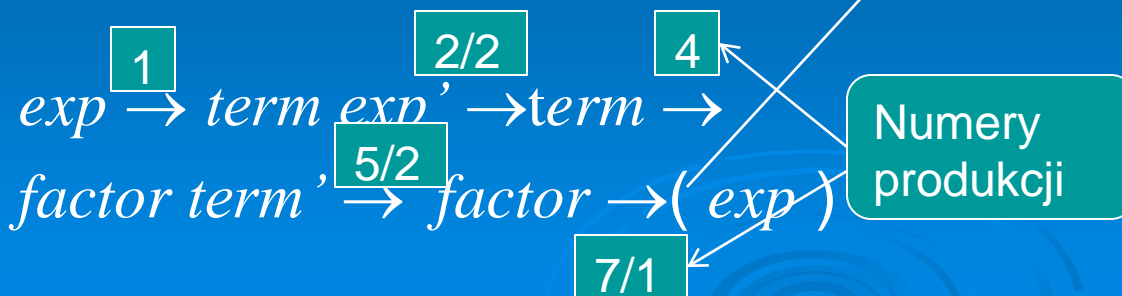
4. $term \rightarrow factor\ term'$

5. $term' \rightarrow mulop\ factor\ term' \mid \varepsilon$

6. $mulop \rightarrow *$

7. $factor \rightarrow (exp) \mid num$

| | First |
|----------|----------------------|
| exp | $(, \dots$ |
| exp' | ε, \dots |
| $addop$ | $+, -, \dots$ |
| $term$ | $(, num, \dots$ |
| $term'$ | ε, \dots |
| $mulop$ | $*, \dots$ |
| $factor$ | $(, num, \dots$ |



Zbiór First : Przykład

1. $exp \rightarrow term\ exp'$

2. $exp' \rightarrow addop\ term\ exp' \mid \varepsilon$

3. $addop \rightarrow + \mid -$

4. $term \rightarrow factor\ term'$

5. $term' \rightarrow mulop\ factor\ term' \mid \varepsilon$

6. $mulop \rightarrow *$

7. $factor \rightarrow (exp) \mid \mathbf{num}$

| | First |
|---------------|----------------------|
| <i>exp</i> | (, ... |
| <i>exp'</i> | ε, \dots |
| <i>addop</i> | + , - , ... |
| <i>term</i> | (, num , ... |
| <i>term'</i> | ε, \dots |
| <i>mulop</i> | * , ... |
| <i>factor</i> | (, num , ... |

Zbiór First : Przykład

1. $exp \rightarrow term\ exp'$

2. $exp' \rightarrow addop\ term\ exp' \mid \varepsilon$

3. $addop \rightarrow + \mid -$

4. $term \rightarrow factor\ term'$

5. $term' \rightarrow mulop\ factor\ term' \mid \varepsilon$

6. $mulop \rightarrow *$

7. $factor \rightarrow (exp) \mid \mathbf{num}$

$term \rightarrow factor\ term' \rightarrow$

$factor \rightarrow (exp) \mid \mathbf{num}$

| | First |
|----------|----------------------|
| exp | (, ... |
| exp' | ε, \dots |
| $addop$ | + , - , ... |
| $term$ | (, num , ... |
| $term'$ | ε, \dots |
| $mulop$ | * , ... |
| $factor$ | (, num , ... |

Zbiór First : Przykład

1. $exp \rightarrow term\ exp'$

2. $exp' \rightarrow addop\ term\ exp' \mid \varepsilon$

3. $addop \rightarrow + \mid -$

4. $term \rightarrow factor\ term'$

5. $term' \rightarrow mulop\ factor\ term' \mid \varepsilon$

6. $mulop \rightarrow *$

7. $factor \rightarrow (\exp) \mid \mathbf{num}$

| | First |
|----------|------------------------|
| exp | (, ... |
| exp' | ε, \dots |
| $addop$ | + , - , ... |
| $term$ | (, num , ... |
| $term'$ | ε, \dots |
| $mulop$ | * , ... |
| $factor$ | (, num , ... |

Analiza Syntaktyczna

Zbiór FOLLOW

- Zbiór $FOLLOW(A)$ dla nieterminala A , jest to zbiór terminali a , które mogą wystąpić bezpośrednio na prawo od A w pewnej formie zdaniowej.

Analiza Syntaktyczna

Zbiór FOLLOW

- Ponadto, jeśli A jest skrajnym symbolem w formie zdaniowej, to dodajemy symbol $\$$ do zbioru $FOLLOW(A)$,
gdzie $\$$ jest to symbol specjalny oznaczający koniec napisu ("*endmarker* ") i nie należący do żadnej gramatyki.

Analiza Syntaktyczna

Zbiór FOLLOW

- Ponadto, jeśli A jest symbolem startowym, to dodajemy symbol $\$$ do zbioru $\text{FOLLOW}(A)$, gdzie $\$$ jest to symbol specjalny oznaczający koniec napisu ("*endmarker*,") i nie należący do żadnej gramatyki.

Zbiór Follow : Przykład

expr jest symb. startowym

$exp \rightarrow term\ exp'$

$exp' \rightarrow addop\ term\ exp' \mid \varepsilon$

$addop \rightarrow + \mid -$

$term \rightarrow factor\ term'$

$term' \rightarrow mulop\ factor\ term' \mid \varepsilon$

$mulop \rightarrow *$

$factor \rightarrow (exp) \mid \mathbf{num}$

| | First | Follow |
|---------------|-------------------|----------|
| <i>exp</i> | (num | \$) |
| <i>exp'</i> | ε + - | \$) |
| <i>addop</i> | + - | |
| <i>term</i> | (num | + - \$) |
| <i>term'</i> | ε * | |
| <i>mulop</i> | * | |
| <i>factor</i> | (num | |

Zbiór Follow : Przykład

expr' jest symb. skrajnym

$exp \rightarrow term\ exp'$
 $exp' \rightarrow addop\ term\ exp' \mid \varepsilon$
 $addop \rightarrow + \mid -$
 $term \rightarrow factor\ term'$
 $term' \rightarrow mulop\ factor\ term'$
 $\mid \varepsilon$
 $mulop \rightarrow *$
 $factor \rightarrow (exp) \mid \mathbf{num}$
 $factor \rightarrow (exp) \rightarrow (term\ exp')$

| | First | Follow |
|---------------|-------------------|----------|
| <i>exp</i> | (num | \$) |
| <i>exp'</i> | ε + - | \$) |
| <i>addop</i> | + - | |
| <i>term</i> | (num | + - \$) |
| <i>term'</i> | ε * | |
| <i>mulop</i> | * | |
| <i>factor</i> | (num | |

Zbiór Follow : Przykład

$exp \rightarrow term\ exp'$

$exp' \rightarrow addop\ term\ exp' \mid \varepsilon$

$addop \rightarrow + \mid -$

$term \rightarrow factor\ term'$

$term' \rightarrow mulop\ factor\ term'$
 $\mid \varepsilon$

$mulop \rightarrow *$

$factor \rightarrow (exp) \mid \mathbf{num}$

| | First | Follow |
|---------------|-------------------|----------|
| <i>exp</i> | (num | \$) |
| <i>exp'</i> | ε + - | \$) |
| <i>addop</i> | + - | |
| <i>term</i> | (num | + - \$) |
| <i>term'</i> | ε * | |
| <i>mulop</i> | * | |
| <i>factor</i> | (num | |

$exp \rightarrow term\ exp' \rightarrow term\ addop\ term\ exp'$
 $\rightarrow term\ +(-)\ term\ exp'$

+(-) oznacza + lub -

Zbiór Follow : Przykład

$exp \rightarrow term\ exp'$

$exp' \rightarrow addop\ term\ exp' \mid \varepsilon$

$addop \rightarrow + \mid -$

$term \rightarrow factor\ term'$

$term' \rightarrow mulop\ factor\ term'$
 $\mid \varepsilon$

$mulop \rightarrow *$

$factor \rightarrow (exp) \mid \mathbf{num}$

| | First | Follow |
|----------|-------------------|----------|
| exp | (num | \$) |
| exp' | ε + - | \$) |
| $addop$ | + - | |
| $term$ | (num | + - \$) |
| $term'$ | ε * | |
| $mulop$ | * | |
| $factor$ | (num | |

$exp \rightarrow term\ exp' \rightarrow term \leftarrow$

term jest symb. skrajnym

Zbiór Follow : Przykład

$exp \rightarrow term\ exp'$

$exp' \rightarrow addop\ term\ exp' \mid \varepsilon$

$addop \rightarrow + \mid -$

$term \rightarrow factor\ term'$

$term' \rightarrow mulop\ factor\ term' \mid \varepsilon$

$mulop \rightarrow *$

$factor \rightarrow (exp) \mid \mathbf{num}$

$factor \rightarrow (exp) \rightarrow (term\ exp') \rightarrow (term)$

| | First | Follow |
|----------|-------------------|----------|
| exp | (num | \$) |
| exp' | ε + - | \$) |
| $addop$ | + - | |
| $term$ | (num | + - \$) |
| $term'$ | ε * | |
| $mulop$ | * | |
| $factor$ | (num | |

Analiza Syntaktyczna

Gramatyki LL(1)

- Pierwsze "L" w LL (1) oznacza skanowanie wejścia od lewej do prawej strony, drugie "L" oznacza zastosowanie wyprowadzenia lewostronnego, i "1" oznacza, że w każdym kroku parsowania tylko jeden symbol wejściowy jest brany pod uwagę.

Analiza Syntaktyczna

Gramatyki LL(1)

Klasa gramatyk LL(1) jest dość bogata, aby uwzględnić większość konstrukcji programistycznych.

Ale są również ograniczenia: gramatyki rekurencyjne lewostronnie oraz gramatyki niejednoznaczne nie należą do gramatyk LL(1).

Analiza Syntaktyczna

Gramatyki LL(1)

- Gramatyka G należy do LL(1) wtedy i tylko wtedy gdy dla dwóch różnych produkcji $A \rightarrow \alpha/\beta$, należących do G , są spełnione następujące warunki:
 1. Dla każdego terminala a , z α i β nie daje się jednocześnie wyprowadzić ciągu rozpoczynającego się od a .

Analiza Syntaktyczna

Gramatyki LL(1)

- 2. Co najwyżej z jednego z α i β daje się wyprowadzić napis pusty.
- 3. Jeśli $\beta \Rightarrow^* \varepsilon$ to α nie wyprowadza napisu, którego pierwszy symbol jest terminalem i który znajduje się w zbiorze FOLLOW(A).
- Jeśli $\alpha \Rightarrow \varepsilon$ to β nie wyprowadza napisu, którego pierwszy symbol jest terminalem i który znajduje się w zbiorze FOLLOW(A).

Analiza Syntaktyczna

Gramatyki LL(1)

- Pierwsze dwa warunki oznaczają, że zbiory $FIRST(\alpha)$ i $FIRST(\beta)$ są rozłączne.
- Trzeci warunek oznacza, że jeśli ε należy do zbioru $FIRST(\beta)$, to zbiory $FIRST(\alpha)$ i $FOLLOW(A)$ są rozłączne; podobnie gdy ε należy do zbioru $FIRST(\alpha)$.

Tworzenie tablicy do parsowania przewidującego

Parsowanie z wykorzystaniem gramatyk LL(1) wymaga w pierwszej kolejności utworzenia tablicy parsowania.

Tworzenie tablicy do parsowania przewidującego

Następujące reguły są stosowane do tworzenia tablic parsowania

1. Dla każdej produkcji $A \rightarrow \alpha$ z gramatyki wykonaj kroki podane niżej.
2. Dla każdego terminala a z $\text{FIRST}(\alpha)$, dodaj produkcję $A \rightarrow \alpha$ do macierzy $M[A,a]$
3. Jeśli ϵ należy do $\text{FIRST}(\alpha)$, to dla każdego terminala b z $\text{FOLLOW}(A)$, dodaj $A \rightarrow \alpha$ do macierzy $M[A,b]$

Tworzenie tablicy do parsowania przewidującego

Następujące reguły są stosowane do tworzenia tablic

4. Jeśli ϵ należy do $\text{FIRST}(\alpha)$ oraz $\$$ należy do $\text{FOLLOW}(A)$, to dodaj $A \rightarrow \alpha$ do $M[A, \$]$.

5. Jeśli jakieś pole tablicy jest puste po wykonaniu kroków 2-4, oznacza to, że jeśli zostanie wybrane takie pole, to analizator zwraca błąd i kończy parsowanie.

Przykład: Tworzenie Tablicy parsowania

| | First | Follow |
|---------------|-----------|------------------|
| <i>exp</i> | {(, num} | {\$,)} |
| <i>exp'</i> | {+, -, ε} | {\$,)} |
| <i>addop</i> | {+, -} | {(, num} |
| <i>term</i> | {(, num} | {+, -,), \$} |
| <i>term'</i> | {*, ε} | {+, -,), \$} |
| <i>mulop</i> | {*} | {(, num} |
| <i>factor</i> | {(, num} | {*, +, -,), \$} |

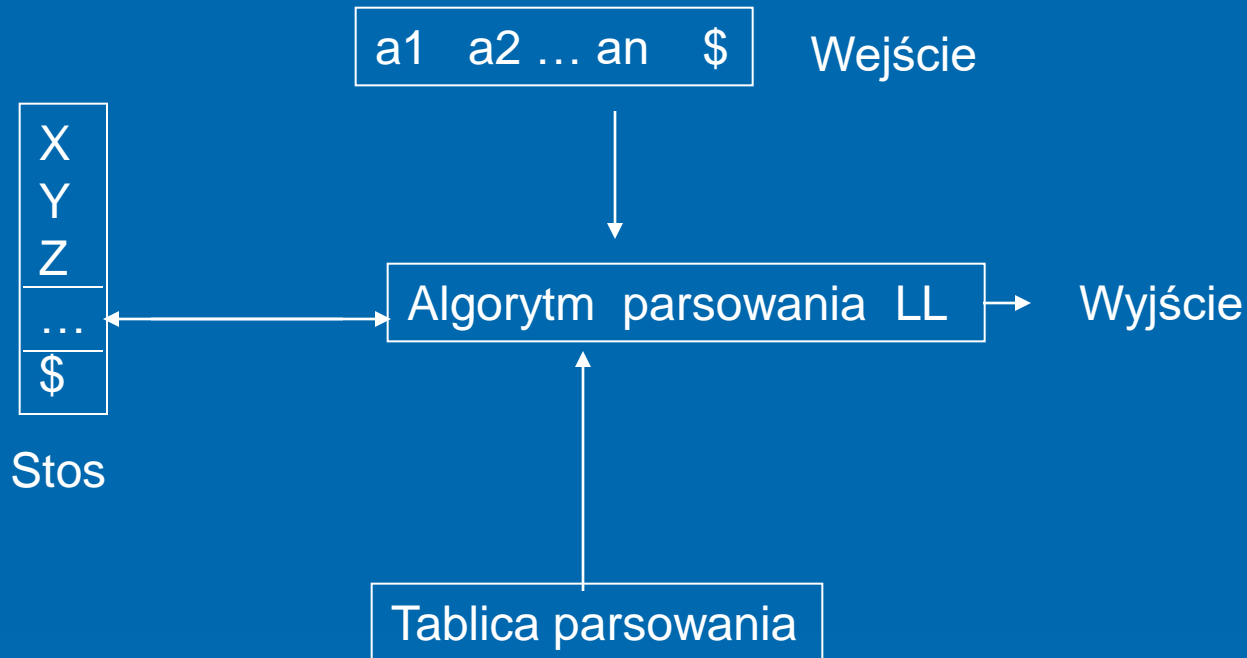
- 1 *exp* → *term exp'*
- 2 *exp'* → *addop term exp'*
- 3 *exp'* → ε
- 4 *addop* → +
- 5 *addop* → -
- 6 *term* → *factor term'*
- 7 *term'* → *mulop factor term'*
- 8 *term'* → ε
- 9 *mulop* → *
- 10 *factor* → (*exp*)
- 11 *factor* → **num**

| | (|) | + | - | * | n | \$ |
|---------------|----|---|---|---|---|----|----|
| <i>exp</i> | 1 | | | | | 1 | |
| <i>exp'</i> | | 3 | 2 | 2 | | | 3 |
| <i>addop</i> | | | 4 | 5 | | | |
| <i>term</i> | 6 | | | | | 6 | |
| <i>term'</i> | | 8 | 8 | 8 | 7 | | 8 |
| <i>mulop</i> | | | | | 9 | | |
| <i>factor</i> | 10 | | | | | 11 | |

Tworzenie tablicy do parsowania przewidującego

- Dla każdej gramatyki LL(1), każde pole(wpis) w tablicy parsowania posiada tylko jedną produkcję lub jest puste (oznacza błąd).
- Jeśli dla danej gramatyki jakieś pole tablicy zawiera dwie lub więcej produkcji, to taka gramatyka nie należy do klasy LL(1).

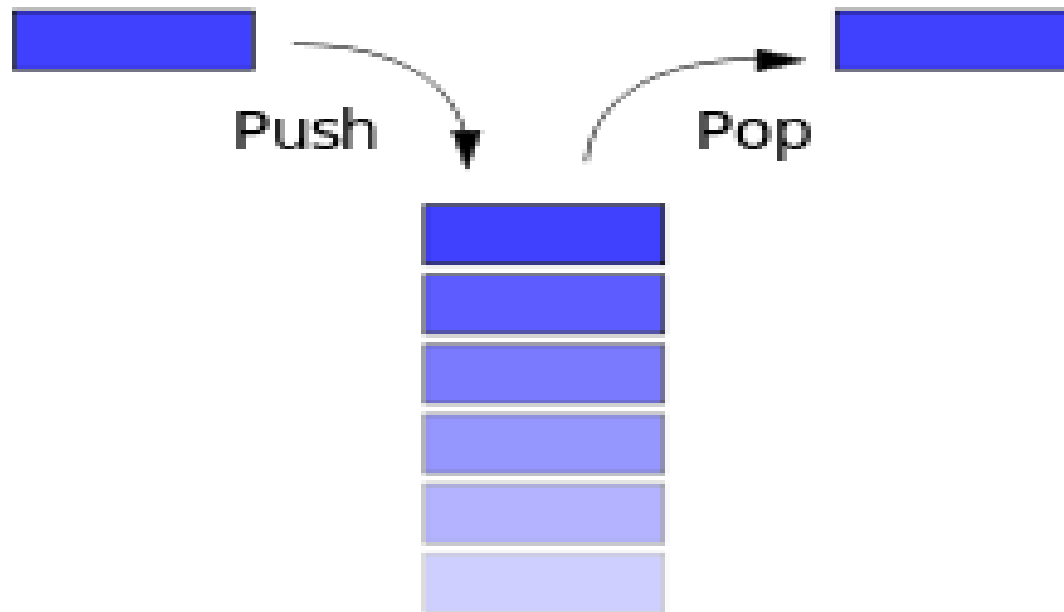
Architektura parsera LL(1)



Analiza Syntaktyczna

Operacje na stosie

- Push -- odłożenie obiektu na stos;
- Pop – ściągnięcie obiektu ze stosu i zwrócenie jego wartości;



Algorytm parsowania

- Załóżmy, że:

zmienna o nazwie **top** zawiera zawartość, która znajduje się na szczycie stosu (pierwsza wartość do odczytu);

zmienna o nazwie **input** przechowuje bieżący terminal zdania wejściowego;

$M[\text{top}, \text{input}]$ oznacza zawartość wpisu tablicy parsowania dla argumentów **top** i **input**.

Wtedy algorytm parsowania przedstawia następujący slajd.

Algorytm parsowania

odłóż na stos dwa symbole: '\$ Symbol Startowy'
(\$ oznacza koniec napisu).

- 1) if top == input == \$ then akceptuj - zdanie wejściowe jest poprawne;
- 2) if top == input then
zdejmij wartość ze szczytu stosu; odczytaj następny symbol wejściowy i podstaw go pod zmienną input; goto 1;
- 3) If top is nonterminal
if $M[\text{top}, \text{input}]$ jest to produkcja, to zastąp wartość na szczycie stosu przez ciało tej produkcji; goto 1;
else error // zawartość $M[\text{top}, \text{input}]$ jest polem pustym
Else error //top zawiera terminal

- Przykład:

(1) $E \rightarrow TE'$

(2) $E' \rightarrow +TE'$

(3) $E' \rightarrow \varepsilon$

(4) $T \rightarrow FT'$

(5) $T' \rightarrow *FT'$

(6) $T' \rightarrow \varepsilon$

(7) $F \rightarrow (E)$

(8) $F \rightarrow id$

Stos
\$E

| | id | + | * | (|) | \$ |
|----|-----|-----|-----|-----|-----|-----|
| E | (1) | | | (1) | | |
| E' | | (2) | | | (3) | (3) |
| T | (4) | | | (4) | | |
| T' | | (6) | (5) | | (6) | (6) |
| F | (8) | | | (7) | | |

Wejście
id+id*id\$

Produkcja

Wyprowadzenie lewostronne:

$E \Rightarrow TE'$

• Przykład:

| | | id | + | * | (|) | \$ |
|----------------------------------|----|-----|-----|-----|-----|-----|-----|
| (1) $E \rightarrow TE'$ | E | (1) | | | (1) | | |
| (2) $E' \rightarrow +TE'$ | E' | (2) | (2) | | | (3) | (3) |
| (3) $E' \rightarrow \varepsilon$ | T | (4) | | | (4) | | |
| (4) $T \rightarrow FT'$ | T' | (6) | (6) | (5) | | (6) | (6) |
| (5) $T' \rightarrow *FT'$ | F | (8) | | | (7) | | |
| (6) $T' \rightarrow \varepsilon$ | | | | | | | |
| (7) $F \rightarrow (E)$ | | | | | | | |
| (8) $F \rightarrow id$ | | | | | | | |

Stos
\$E
\$E'T

Wejście
id+id*id\$
id+id*id\$

Produkcja
 $E \rightarrow TE'$

Wyprowadzenie lewostronne:
 $E \Rightarrow TE' \Rightarrow FT'E$

• Przykład:

| | | id | + | * | (|) | \$ |
|----------------------------------|----|-----|-----|---|-----|-----|-----|
| (1) $E \rightarrow TE'$ | E | (1) | | | (1) | | |
| (2) $E' \rightarrow +TE'$ | E' | | (2) | | | (3) | (3) |
| (3) $E' \rightarrow \varepsilon$ | T | (4) | | | (4) | | |
| (4) $T \rightarrow FT'$ | T' | (6) | (5) | | | (6) | (6) |
| (5) $T' \rightarrow *FT'$ | F | (8) | | | (7) | | |
| (6) $T' \rightarrow \varepsilon$ | | | | | | | |
| (7) $F \rightarrow (E)$ | | | | | | | |
| (8) $F \rightarrow id$ | | | | | | | |

Stos
 \$E
 \$E'T
 \$E'T'F

Wejście
 id+id*id\$
 id+id*id\$
 id+id*id\$

Produkcja
 E->TE'
 T->FT'

Wyprowadzenie lewostronne:

$E \Rightarrow TE' \Rightarrow FT'E' \Rightarrow idT'E$

- Przykład:

| | id | + | * | (|) | \$ |
|----------------------------------|-------|-----|-----|-----|-----|-----|
| (1) $E \rightarrow TE'$ | E (1) | | | (1) | | |
| (2) $E' \rightarrow +TE'$ | E' | (2) | | | (3) | (3) |
| (3) $E' \rightarrow \varepsilon$ | T | | | (4) | | |
| (4) $T \rightarrow FT'$ | T' | (6) | (5) | | (6) | (6) |
| (5) $T' \rightarrow *FT'$ | F | (8) | | (7) | | |
| (6) $T' \rightarrow \varepsilon$ | | | | | | |
| (7) $F \rightarrow (E)$ | | | | | | |
| (8) $F \rightarrow id$ | | | | | | |

Stos

\$E

\$E'T

\$E'T'F

\$E'T'id

Wejście

id+id*id\$

id+id*id\$

id+id*id\$

id+id*id\$

Produkcja

$E \rightarrow TE'$

$T \rightarrow FT'$

$F \rightarrow id$

Wyprowadzenie lewostronne:

$E \Rightarrow TE' \Rightarrow FT'E' \Rightarrow idT'E'$

• Przykład:

- (1) $E \rightarrow TE'$
- (2) $E' \rightarrow +TE'$
- (3) $E' \rightarrow \varepsilon$
- (4) $T \rightarrow FT'$
- (5) $T' \rightarrow *FT'$
- (6) $T' \rightarrow \varepsilon$
- (7) $F \rightarrow (E)$
- (8) $F \rightarrow id$

| | id | + | * | (|) | \$ |
|----|-----|-----|-----|-----|-----|-----|
| E | (1) | | | (1) | | |
| E' | | (2) | | | (3) | (3) |
| T | (4) | | | (4) | | |
| T' | | (6) | (5) | | (6) | (6) |
| F | (8) | | | (7) | | |

Stos

\$E
\$E'T
\$E'T'F
\$E'T'id
\$E'T'

Wejście

id+id*id\$
id+id*id\$
id+id*id\$
id+id*id\$
+id*id\$

Produkcja

$E \rightarrow TE'$
 $T \rightarrow FT'$
 $F \rightarrow id$
 $T' \rightarrow \varepsilon$

Wyprowadzenie lewostronne:
 $E \Rightarrow TE' \Rightarrow FT'E' \Rightarrow idT'E' \Rightarrow idE'$

• Przykład:

| | id | + | * | (|) | \$ |
|----------------------------------|-----|-----|-----|-----|-----|-----|
| (1) $E \rightarrow TE'$ | (1) | | | (1) | | |
| (2) $E' \rightarrow +TE'$ | | (2) | | | (3) | (3) |
| (3) $E' \rightarrow \varepsilon$ | (4) | | | (4) | | |
| (4) $T \rightarrow FT'$ | | (6) | (5) | | (6) | (6) |
| (5) $T' \rightarrow *FT'$ | (8) | | | (7) | | |
| (6) $T' \rightarrow \varepsilon$ | | | | | | |
| (7) $F \rightarrow (E)$ | | | | | | |
| (8) $F \rightarrow id$ | | | | | | |

Stos
 $\$E$
 $\$E'T$
 $\$E'T'F$
 $\$E'T'id$
 $\$E'$

Wejście
 $id+id*id\$$
 $id+id*id\$$
 $id+id*id\$$
 $id+id*id\$$
 $+id*id\$$

Produkcja
 $E \rightarrow TE'$
 $T \rightarrow FT'$
 $F \rightarrow id$
 $T' \rightarrow \varepsilon$

Wyprowadzenie lewostronne:
 $E \Rightarrow TE' \Rightarrow FT'E' \Rightarrow idT'E' \Rightarrow idE'$

• Przykład:

| | id | + | * | (|) | \$ |
|----------------------------------|-------|-----|-----|-----|-----|-----|
| (1) $E \rightarrow TE'$ | E (1) | | | (1) | | |
| (2) $E' \rightarrow +TE'$ | E' | (2) | | | (3) | (3) |
| (3) $E' \rightarrow \varepsilon$ | T | | | (4) | | |
| (4) $T \rightarrow FT'$ | T' | (6) | (5) | | (6) | (6) |
| (5) $T' \rightarrow *FT'$ | F | (8) | | (7) | | |
| (6) $T' \rightarrow \varepsilon$ | | | | | | |
| (7) $F \rightarrow (E)$ | | | | | | |
| (8) $F \rightarrow id$ | | | | | | |

Stos
 \$E
 \$E'T
 \$E'T'F
 \$E'T'id
 \$E'
 \$E'T+



Wejście
 id+id*id\$
 id+id*id\$
 id+id*id\$
 id+id*id\$
 +id*id\$
 +id*id\$



Produkcja
 E->TE'
 T->FT'
 F->id
 T'-> ε
 E'->+TE'

Wyprowadzenie lewostronne:
 $E \Rightarrow TE' \Rightarrow FT'E' \Rightarrow idT'E' \Rightarrow id + TE'$

- Przykład:

| | id | + | * | (|) | \$ |
|----------------------------------|----|-----|-----|-----|-----|-----|
| (1) $E \rightarrow TE'$ | E | (1) | | (1) | | |
| (2) $E' \rightarrow +TE'$ | E' | (2) | | | (3) | (3) |
| (3) $E' \rightarrow \varepsilon$ | T | (4) | | (4) | | |
| (4) $T \rightarrow FT'$ | T' | (6) | (5) | | (6) | (6) |
| (5) $T' \rightarrow *FT'$ | F | (8) | | (7) | | |
| (6) $T' \rightarrow \varepsilon$ | | | | | | |
| (7) $F \rightarrow (E)$ | | | | | | |
| (8) $F \rightarrow id$ | | | | | | |

Stos

\$E

\$E'T

\$E'T'F

\$E'T'id

\$E'

\$E'T

.....

Wejście

id+id*id\$

id+id*id\$

id+id*id\$

id+id*id\$

+id*id\$

+id*id\$

Produkcja

$E \rightarrow TE'$

$T \rightarrow FT'$

$F \rightarrow id$

$T' \rightarrow \varepsilon$

$E' \rightarrow +TE'$

Wyprowadzenie lewostronne:

$E \Rightarrow TE' \Rightarrow FT'E' \Rightarrow idT'E' \Rightarrow \dots \Rightarrow id+id*id$

Dziękuję za uwagę