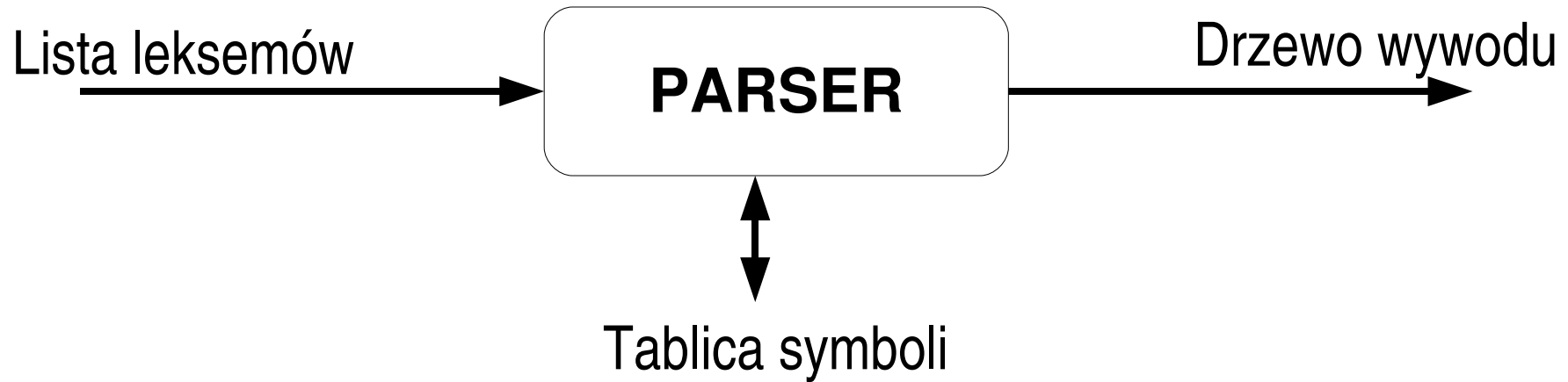


# Analizator składniowy (parser)



Parsery:

- zstępujące ( top-down )

$S ::= cAd$

$A ::= ab \mid a$

słowo: **cabd**

- wstępujące ( bottom-up )

$S ::= aABe$

$A ::= Abc \mid b$

$B ::= d$

słowo: **abbcde**

## Analizatory top-down

- Z powrotami ( backtracking )

$S ::= aBc$

słowo: **abc**

$B ::= bc \mid b$

- Przewidujące ( predictive )
  - Gramatyka musi posiadać własność LL(1) – patrz produkcja z alternatywą
  - Jak rozwiązać problem epsilon-produkcji ?
  - Schemat implementacji przewidującego kompilatora rekurencyjnego:
    - Każdemu nieterminalowi przypisz oddzielną procedurę
    - Zdefiniuj funkcję sprawdzającą, czy aktualne wejście pasuje do oczekiwanego terminala (*accept*)
    - Rozwiązania dla: alternatywy, pętli(listy), konstrukcji opcjonalnej
- Parsery LL(1) – przewidujące ale bez rekurencji

# Parser LL(1)

*Parser iteracyjny = Wejście + Tablica parsingu (TP) +  
Stos + Algorytm analizy*

## Algorytm działania:

**WS:** symbol z wierzchołka stosu    **WE:** aktualny symbol na wejściu    **TP:** tablica parsingu

- Początkowo, na stosie znajdują się: znak końca danych i wyżej – symbol startowy gramatyki.
- Jeżeli symbol na wejściu i na wierzchu stosu to symbol końca danych, oznacza to, że analiza zakończyła się sukcesem
- Jeżeli symbole na wierzchu stosu i na wejściu są takie same ( i nie jest to koniec danych), to zdejmij element ze stosu oraz przesunij się do następnego symbolu na wejściu.
- Jeżeli symbole są różne, to sprawdź, czy symbol na stosie jest nieterminalem (powinien być) i sprawdź zapis  $TP[WS,WE]$ :
  - a)  $TP[WS,WE] = \text{err}$  – błąd, wypisz informację o błędzie
  - b)  $TP[WS,WE]$  to produkcja postaci : **WS ::= ABC**  
Zdejmij symbol WS i zastąp go symbolami ABC. Wypisz użytą produkcję.

## Parser LL(1) - przykład

(1)  $E ::= TE'$

(2)  $E' ::= +TE' \mid \epsilon$

(3)  $T ::= FT'$

(4)  $T' ::= *FT' \mid \epsilon$

(5)  $F ::= (E) \mid id$

|    | id   | +    | *    | (    | )    | \$   |
|----|------|------|------|------|------|------|
| E  | (1)  | err  | err  | (1)  | err  | err  |
| E' | err  | (2a) | err  | err  | (2b) | (2b) |
| T  | (3)  | err  | err  | (3)  | err  | err  |
| T' | err  | (4b) | (4a) | err  | (4b) | (4b) |
| F  | (5b) | err  | err  | (5a) | err  | err  |

# Błędy składniowe

Tryby wydobywania się z błędów:

## 1. **Panic mode:**

Pomijamy część symboli do momentu, aż znajdziemy takie, od których ponownie można poprawnie rozpoznawać tekst.

## 2. **Phrase level:**

Dokonujemy lokalnej korekcji wyrażenia (ostrożnie!)

## 3. **Error productions:**

W gramatyce uwzględniamy błędy składniowe

*Rozwiązanie proponowane: "panic mode" z wykorzystaniem zbiorów "poprzedników" i "następników" (FIRST, FOLLOW)*