

POLITECHNIKA ŁÓDZKA

**Wydział Elektrotechniki, Elektroniki,
Informatyki i Automatyki
Instytut Informatyki Stosowanej**

Lingwistyka Matematyczna

Laboratorium

Rok akademicki 2022/2023

Zadanie 5

Analizator składniowy LL(1)

Dominik Bujnowicz

249073@edu.p.lodz.pl

SRIMP-1

1. Treść zadania

Zadaniem 5 laboratorium będzie napisanie programu komputerowego w dowolnym języku programowania będącego analizatorem składniowym LL(1) realizującym algorytm rozbioru generacyjnego zstępującego z wyprzedzeniem o jeden symbol LL(1) do następującej gramatyki:

```
S ::= W ; Z
Z ::= W ; Z | ε
W ::= P | POW
P ::= R | (W)
R ::= L | L.L
L ::= C | CL
C ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
O ::= * | : | + | - | ^
```

Powyższa gramatyka umożliwia budowanie ciągów składających się z liczb rzeczywistych i operacji arytmetycznych (zdań arytmetycznych) zakończonych średnikiem:

$$(1.2 * 3) + 5 - (23.4 + 3)^3; 8: 3;$$

2. Zgodność gramatyki z LL(1)

Aby gramatyka mogła być wykorzystana w analizatorze składniowym LL(1), do wybrania kolejnej produkcji powinno wystarczyć podejrzenie jednego symbolu terminalnego. Jest to równoznaczne ze spełnianiem przez gramatykę warunków I i II reguły LL(1).

Zgodność z I regułą LL(1) oznacza, że dla każdej pary alternatyw produkcji, część wspólna zbiorów FIRST tych alternatyw jest zbiorem pustym. Zgodność z II regułą LL(1) oznacza natomiast, że dla każdej produkcji zawierającej symbol pusty ϵ , część wspólna zbiorów FIRST i FOLLOW tej produkcji jest zbiorem pustym.

Do sprawdzenia zgodności z regułami LL(1) będą potrzebne zbiory FIRST produkcji, które są zbiorami możliwych symboli pierwszych oraz zbiory FOLLOW, które są zbiorami możliwych symboli następujących.

Wyprowadzanie potrzebnych zbiorów FIRST produkcji

$$\text{FIRST}(O) = \{*,.,+,-,\wedge\}$$

$$\text{FIRST}(C) = \{0,1,2,3,4,5,6,7,8,9\}$$

$$\text{FIRST}(L) = \text{FIRST}(C) = \{0,1,2,3,4,5,6,7,8,9\}$$

$$\text{FIRST}(R) = \text{FIRST}(L) = \{0,1,2,3,4,5,6,7,8,9\}$$

$$\text{FIRST}(P) = \text{FIRST}(R) \cup \{(\} = \{0,1,2,3,4,5,6,7,8,9,(\}$$

$$\text{FIRST}(W) = \text{FIRST}(P) \cup \text{FIRST}(P) = \text{FIRST}(P) = \{0,1,2,3,4,5,6,7,8,9,(\}$$

$$\text{FIRST}(Z) = \text{FIRST}(W) \cup \{\epsilon\} = \{0,1,2,3,4,5,6,7,8,9,(\epsilon\}$$

$$\text{FIRST}(S) = \text{FIRST}(W) = \{0,1,2,3,4,5,6,7,8,9,(\}$$

Sprawdzanie poprawności z I regułą gramatyki

S: reguła jest spełniona, nie ma alternatywy

Z: $\text{FIRST}(W) \cap \{\epsilon\} = \emptyset$ reguła jest spełniona, część wspólna to zbiór pusty

W: $\text{FIRST}(P) \cap \text{FIRST}(P) = \text{FIRST}(P) \neq \emptyset$ reguła nie jest spełniona, część wspólna nie jest zbiorem pustym

P: $\text{FIRST}(R) \cap \{(\} = \emptyset$ reguła jest spełniona, część wspólna to zbiór pusty

R: $\text{FIRST}(L) \cap \text{FIRST}(L) = \text{FIRST}(L) \neq \emptyset$ reguła nie jest spełniona, część wspólna nie jest zbiorem pustym

L: $\text{FIRST}(C) \cap \text{FIRST}(C) = \text{FIRST}(C) \neq \emptyset$ reguła nie jest spełniona, część wspólna nie jest zbiorem pustym

C: \emptyset reguła jest spełniona, wszystkie alternatywy to różne od siebie symbole, część wspólna z dowolnej ich pary jest zbiorem pustym

O: \emptyset reguła jest spełniona, wszystkie alternatywy to różne od siebie symbole, część wspólna z dowolnej ich pary jest zbiorem pustym

Wyprowadzanie potrzebnych zbiorów FOLLOW produkcji

$$\text{FOLLOW}(Z) = \text{FOLLOW}(Z) \cup \text{FOLLOW}(S) = \text{FOLLOW}(S) = \emptyset$$

Sprawdzanie poprawności z II regułą gramatyki

S: reguła jest spełniona, nie występuje pusty symbol ϵ

Z: $\text{FIRST}(Z) \cap \text{FOLLOW}(Z) = \text{FIRST}(Z) \cap \emptyset = \emptyset$ reguła jest spełniona, część wspólna jest zbiorem pustym

W: reguła jest spełniona, nie występuje pusty symbol ϵ

P: reguła jest spełniona, nie występuje pusty symbol ϵ

R: reguła jest spełniona, nie występuje pusty symbol ϵ

L: reguła jest spełniona, nie występuje pusty symbol ϵ

C: reguła jest spełniona, nie występuje pusty symbol ϵ

O: reguła jest spełniona, nie występuje pusty symbol ϵ

Przekształcanie gramatyki

Ponieważ produkcje W, R i L nie są zgodne z I regułą gramatyki LL(1), należy w pewien sposób przekształcić gramatykę, aby każda alternatywa dla tych produkcji miała oddzielne symbole pierwsze. W tej gramatyce, można to osiągnąć używając lewostronnej faktoryzacji, wyciągając część wspólną produkcji do osobnej produkcji.

$S ::= W ; Z$
 $Z ::= W ; Z \mid \varepsilon$
 $W ::= PW'$
 $W' ::= \varepsilon \mid OW$
 $P ::= R \mid (W)$
 $R ::= LR'$
 $R' ::= \varepsilon \mid .L$
 $L ::= CL'$
 $L' ::= \varepsilon \mid L$
 $C ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
 $O ::= * \mid : \mid + \mid - \mid ^$

Po przekształceniu gramatyki należy ponownie sprawdzić jej zgodność z I i II regułą LL(1).

Wyprowadzanie potrzebnych zbiorów FIRST produkcji

$FIRST(O) = \{*, :, +, -, ^\}$
 $FIRST(C) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
 $FIRST(L') = \{\varepsilon\} \cup FIRST(L) = \{\varepsilon, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
 $FIRST(L) = FIRST(C) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
 $FIRST(R') = \{\varepsilon\} \cup \{.\} = \{\varepsilon, .\}$
 $FIRST(R) = FIRST(L) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
 $FIRST(P) = FIRST(R) \cup \{(\} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (\}$
 $FIRST(W') = \{\varepsilon\} \cup FIRST(O) = \{\varepsilon, *, :, +, -, ^\}$
 $FIRST(W) = FIRST(P) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (\}$
 $FIRST(Z) = FIRST(W) \cup \{\varepsilon\} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (\}, \varepsilon\}$
 $FIRST(S) = FIRST(W) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (\}$

Sprawdzanie poprawności z I regułą gramatyki zmienionych produkcji

W: reguła jest spełniona, nie ma alternatywy
W': $\{\varepsilon\} \cap FIRST(O) = \emptyset$ reguła jest spełniona, część wspólna to zbiór pusty
R: reguła jest spełniona, nie ma alternatywy
R': $\{\varepsilon\} \cap \{.\} = \emptyset$ reguła jest spełniona, część wspólna to zbiór pusty
L: reguła jest spełniona, nie ma alternatywy
L': $\{\varepsilon\} \cap FIRST(L) = \emptyset$ reguła jest spełniona, część wspólna to zbiór pusty

Wyprowadzanie potrzebnych zbiorów FOLLOW produkcji

$\text{FOLLOW}(W') = \text{FOLLOW}(W) = \{\}\} \cup \text{FOLLOW}(W') \cup \{;\} = \{\},;\}$

$\text{FOLLOW}(R') = \text{FOLLOW}(R) = \text{FOLLOW}(P) = \text{FIRST}(W') \cup \text{FOLLOW}(W') = \{\epsilon, *, :, +, -, ^,), ;\}$

$\text{FOLLOW}(L') = \text{FOLLOW}(L) = \text{FOLLOW}(L') \cup \text{FOLLOW}(R') \cup \text{FIRST}(R') \cup \text{FOLLOW}(R') = \text{FIRST}(R') \cup \text{FOLLOW}(R') = \{\epsilon, *, :, +, -, ^,), ;\}$

Sprawdzanie poprawności z II regułą gramatyki zmienionych produkcji

W: reguła jest spełniona, nie występuje pusty symbol ϵ

W': $\text{FIRST}(W') \cap \text{FOLLOW}(W') = \emptyset$ reguła jest spełniona, część wspólna to zbiór pusty

R: reguła jest spełniona, nie występuje pusty symbol ϵ

R': $\text{FIRST}(R') \cap \text{FOLLOW}(R') = \{\epsilon\} = \emptyset$ reguła jest spełniona, część wspólna to zbiór pusty

L: reguła jest spełniona, nie występuje pusty symbol ϵ

L': $\text{FIRST}(L') \cap \text{FOLLOW}(L') = \{\epsilon\} = \emptyset$ reguła jest spełniona, część wspólna to zbiór pusty

Wszystkie produkcje przekształconej gramatyki spełniają I i II regułę LL(1).

3. Diagram składniowy

Diagramy składni dla poszczególnych produkcji

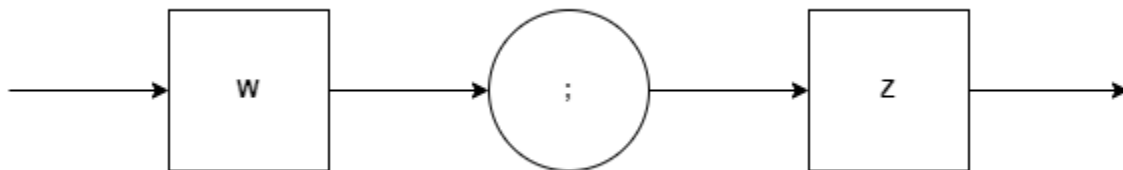


Diagram 1. Produkcja S

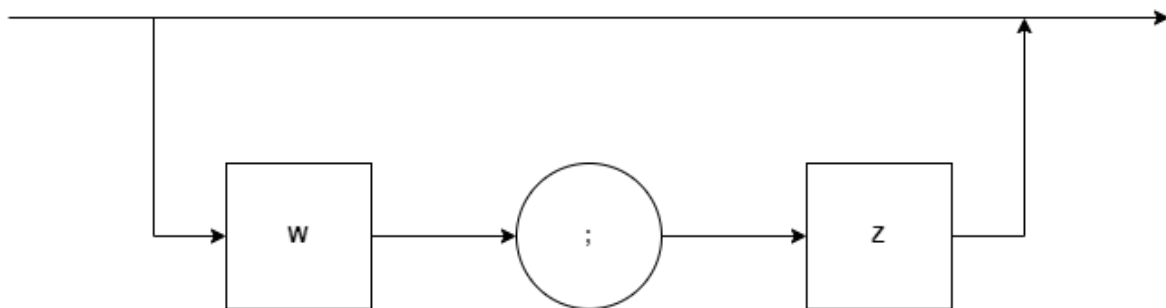


Diagram 2. Produkcja Z

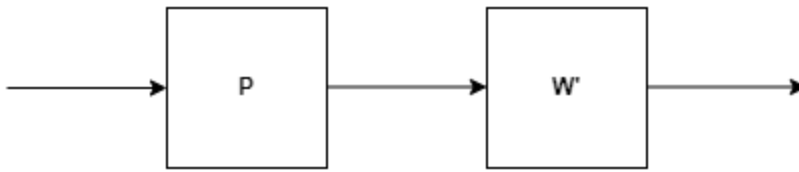


Diagram 3. Produkcja W

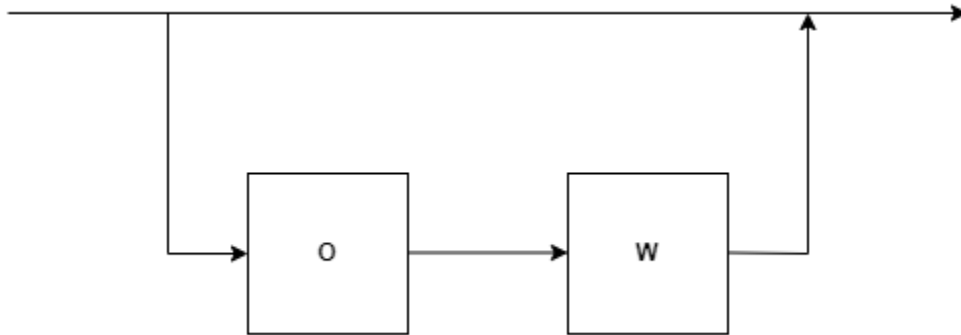


Diagram 4. Produkcja W'

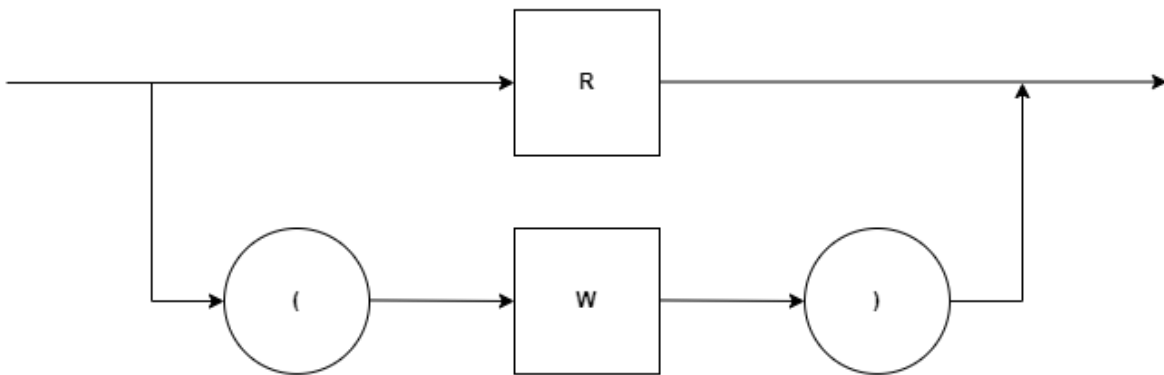


Diagram 5. Produkcja P

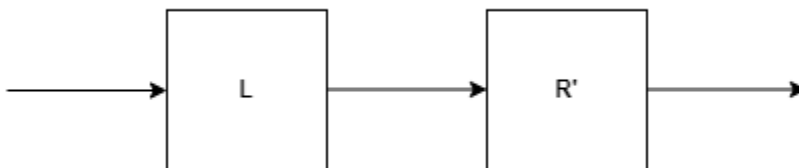


Diagram 6. Produkcja R

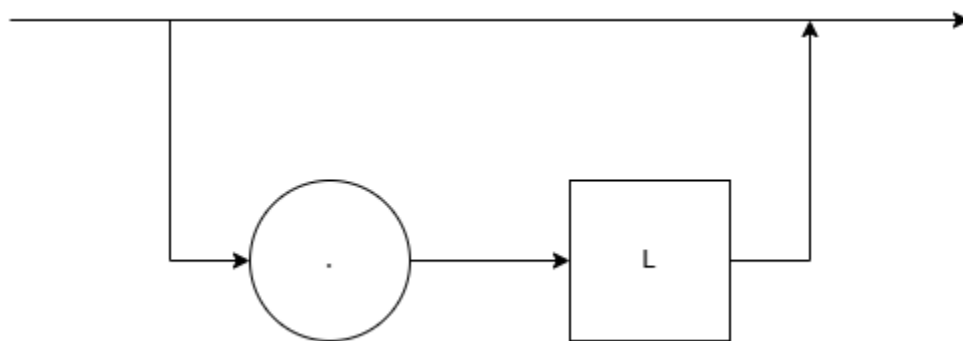


Diagram 7. Produkcja R'

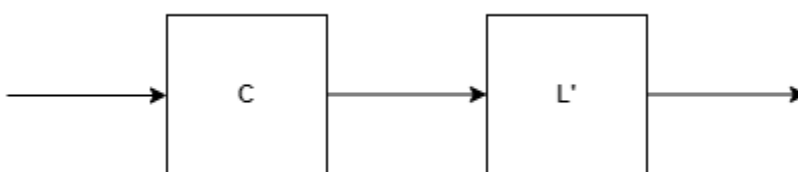


Diagram 8. Produkcja L

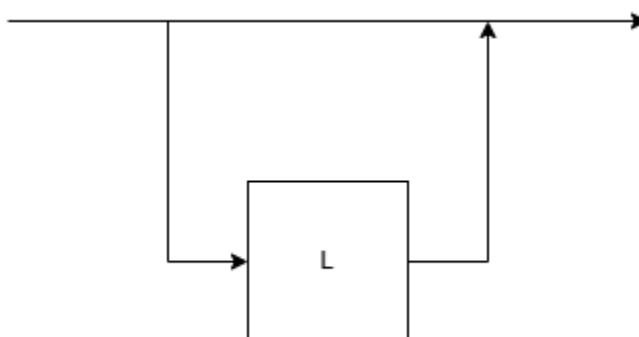


Diagram 9. Produkcja L'

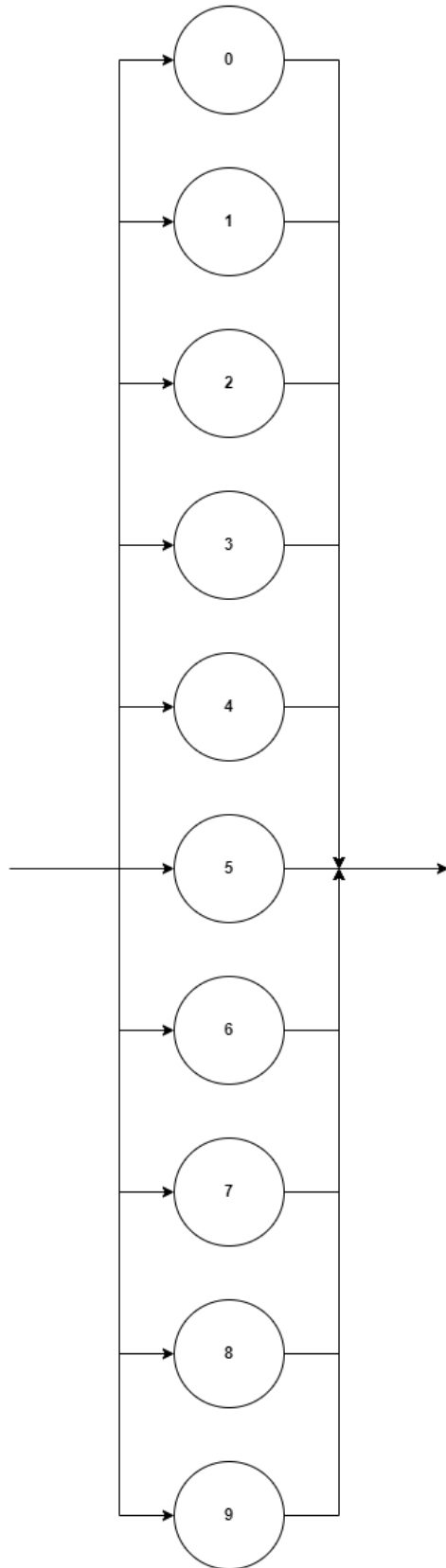


Diagram 10. Produkcja C

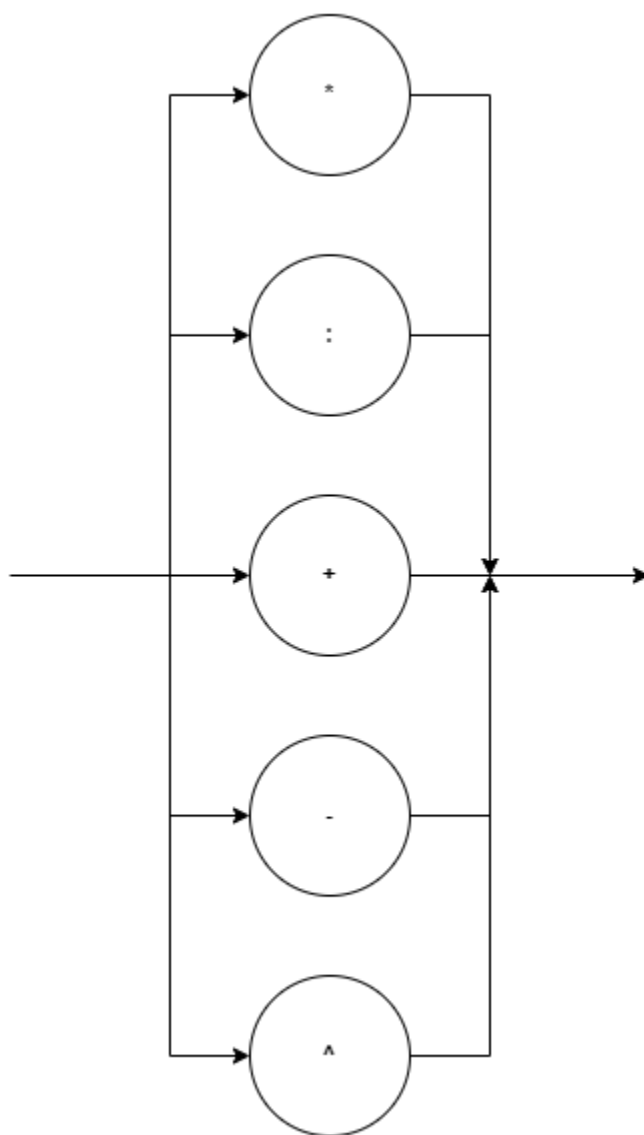
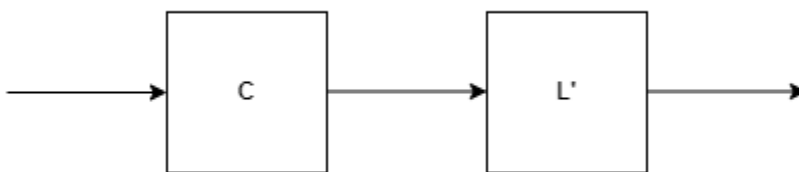


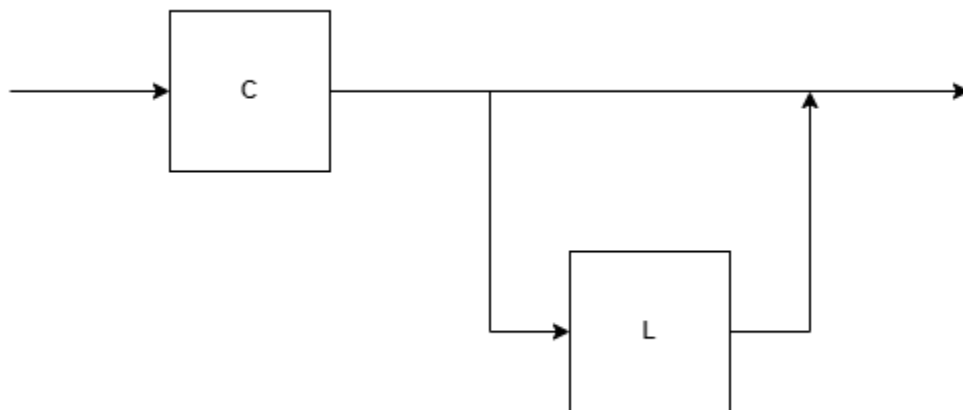
Diagram 11. Produkcja O

Aby zmniejszyć ilość potrzebnych produkcji w końcowym diagramie i w programie należy uprościć diagramy produkcji, dla których jest to możliwe.

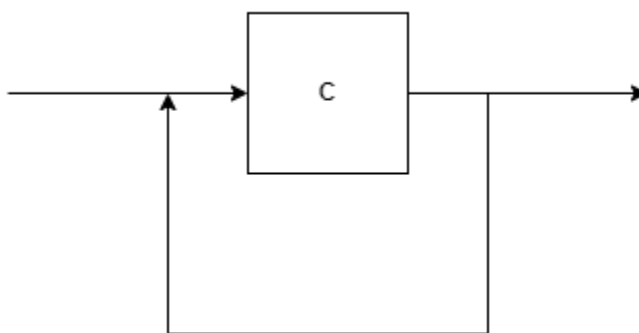
Upraszczenie produkcji L



Upraszczenie produkcji L 1. Produkcja L

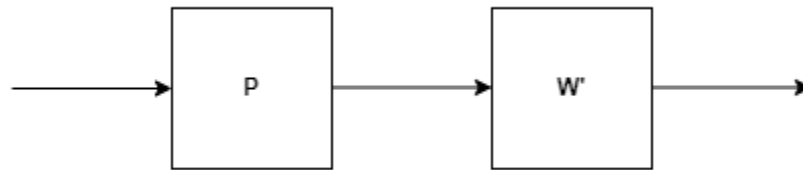


Upraszczenie produkcji L 2. Zastąpienie produkcji L' jej diagramem

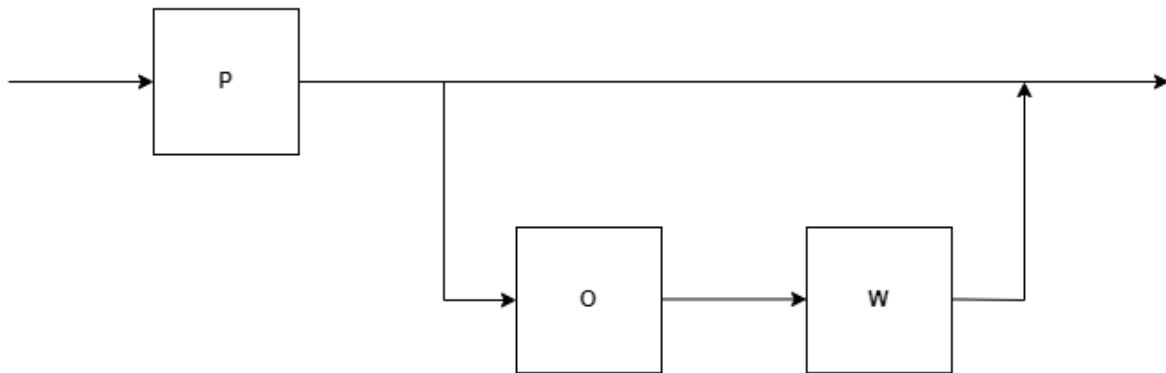


Upraszczenie produkcji L 3. Usunięcie duplikacji dla produkcji L używając rekurencji

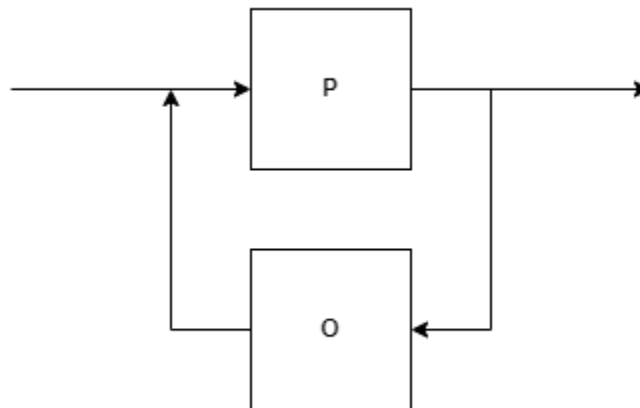
Upraszczenie produkcji W



Upraszczenie produkcji W 1. Produkcja W

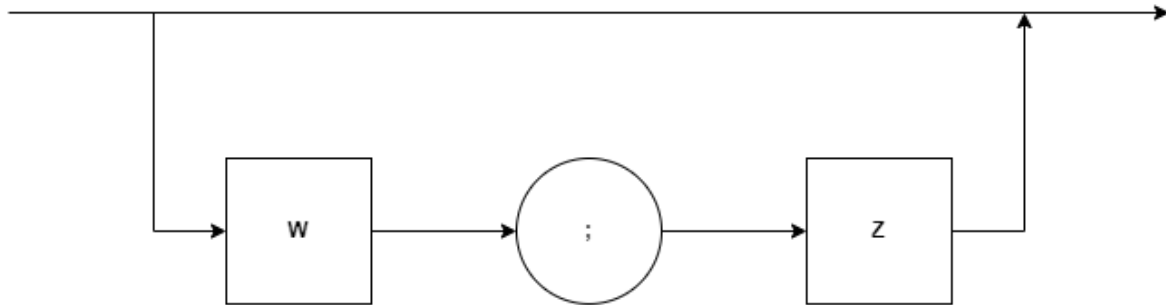


Upraszczenie produkcji W 2. Zastąpienie produkcji W' jej diagramem

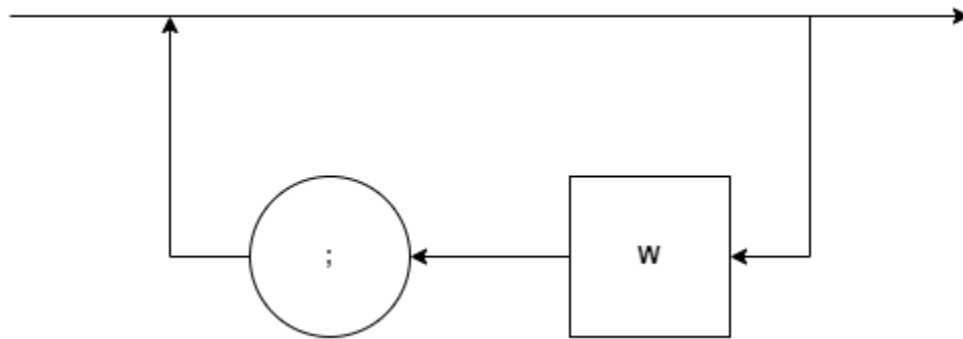


Upraszczenie produkcji W 3. Usunięcie duplikacji dla produkcji W używając rekurencji

Upraszczenie produkcji Z

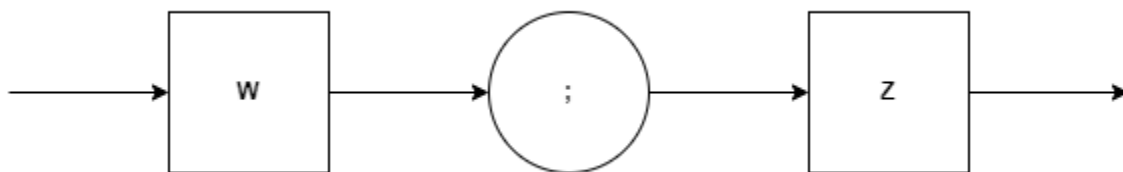


Upraszczenie produkcji Z 1. Produkcja Z

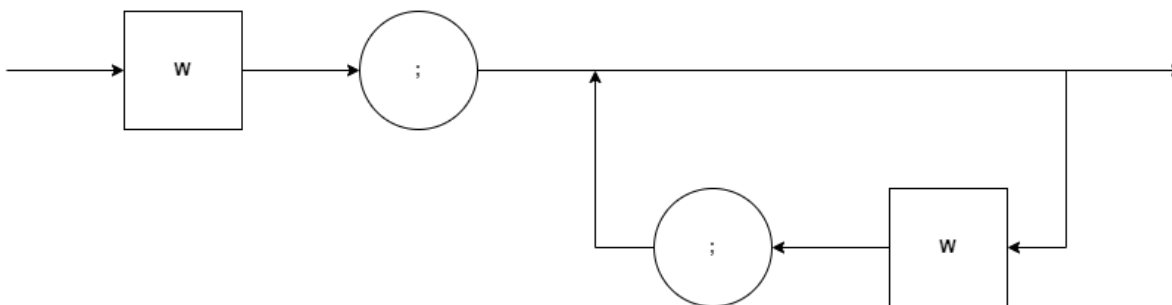


Upraszczenie produkcji Z 2. Usunięcie duplikacji dla produkcji Z używając rekurencji

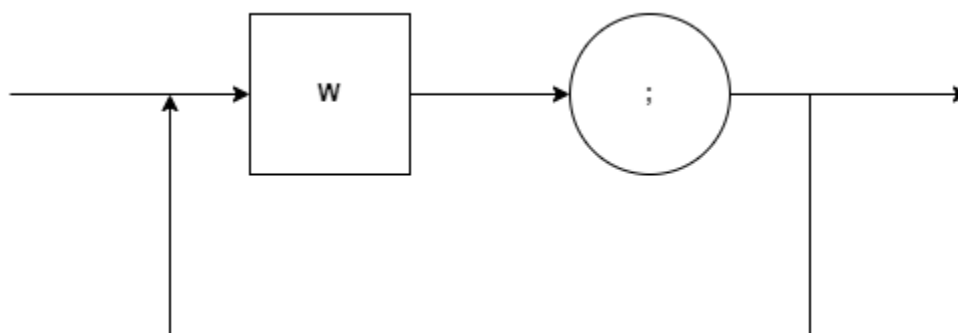
Upraszczenie produkcji S



Upraszczenie produkcji S 1. Produkcja S



Upraszczanie produkcji S 2. Zastąpienie produkcji Z jej diagramem



Upraszczanie produkcji S 3. Usunięcie duplikacji dla produkcji W i symbolu ; używając rekurencji

Po uproszczeniu wszystkich możliwych produkcji, należy zaczynając od korzenia gramatyki, po kolei podstawiać potrzebne produkcje aż otrzymamy końcowy diagram gramatyki.

Wprowadzenie końcowego diagramu gramatyki

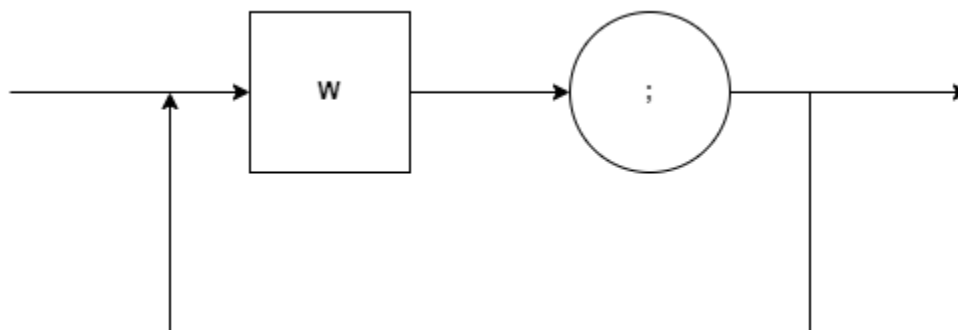


Diagram końcowy gramatyki 1. Produkcja S (korzeń gramatyki)

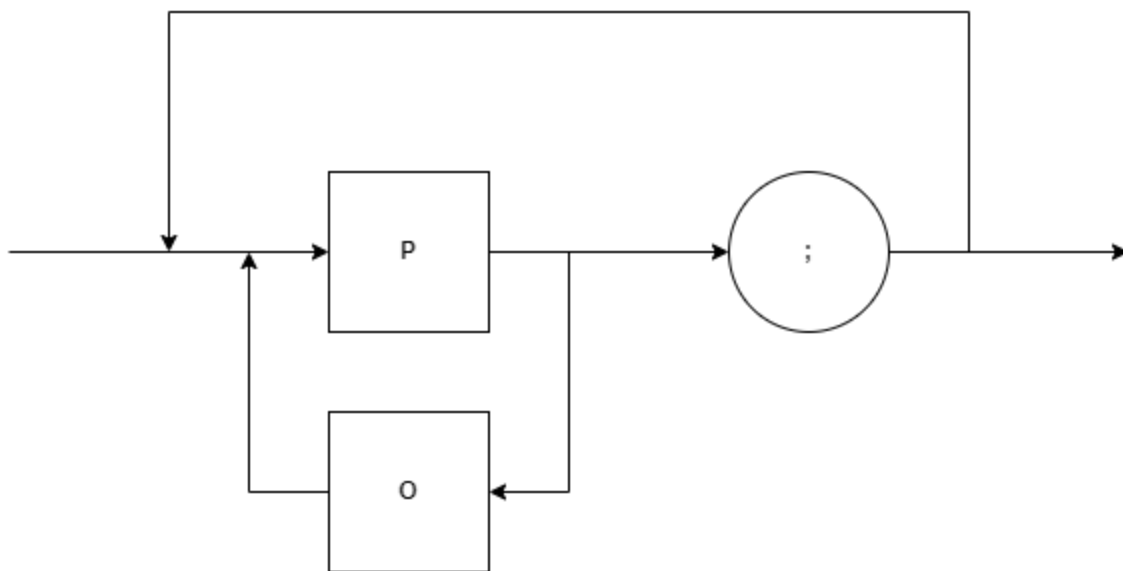


Diagram końcowy gramatyki 2. Zastąpienie produkcji W jej diagramem

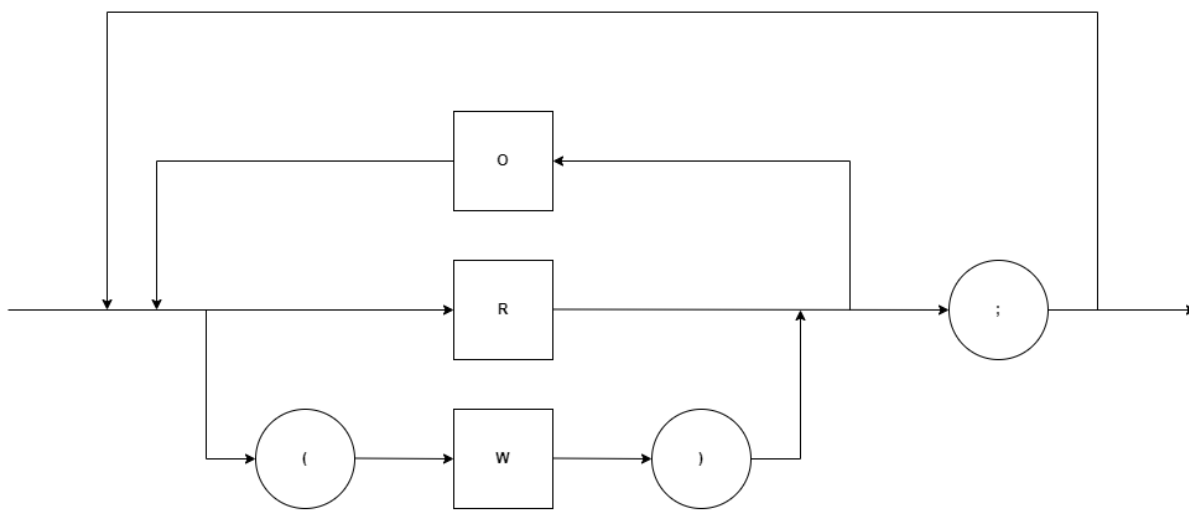


Diagram końcowy gramatyki 3. Zastąpienie produkcji P jej diagramem

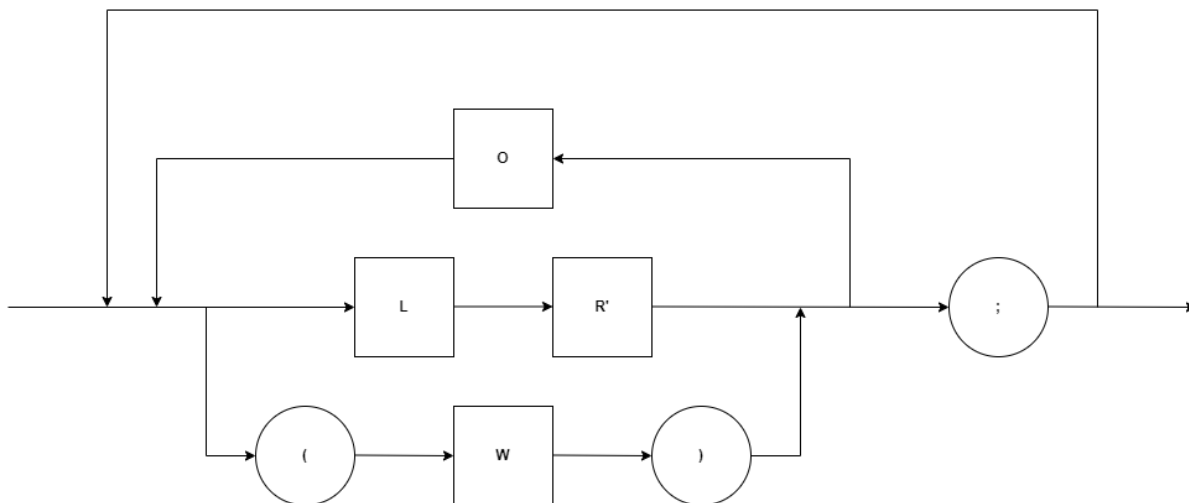


Diagram końcowy gramatyki 4. Zastąpienie produkcji R jej diagramem

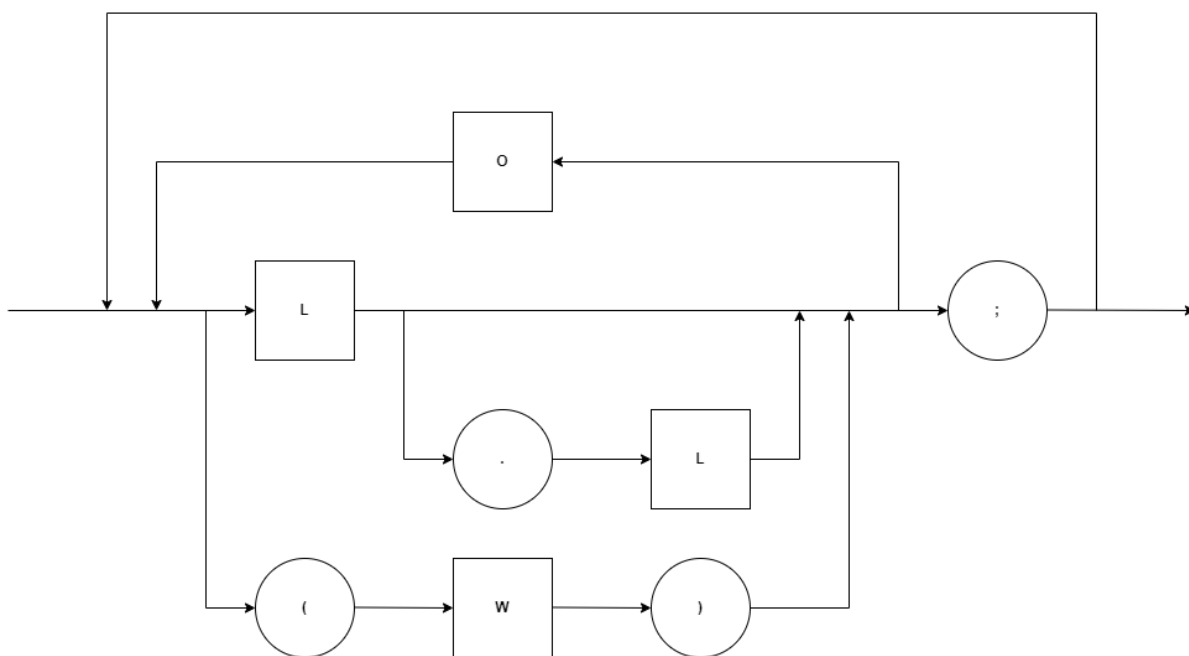


Diagram końcowy gramatyki 5. Zastąpienie produkcji R' jej diagramem

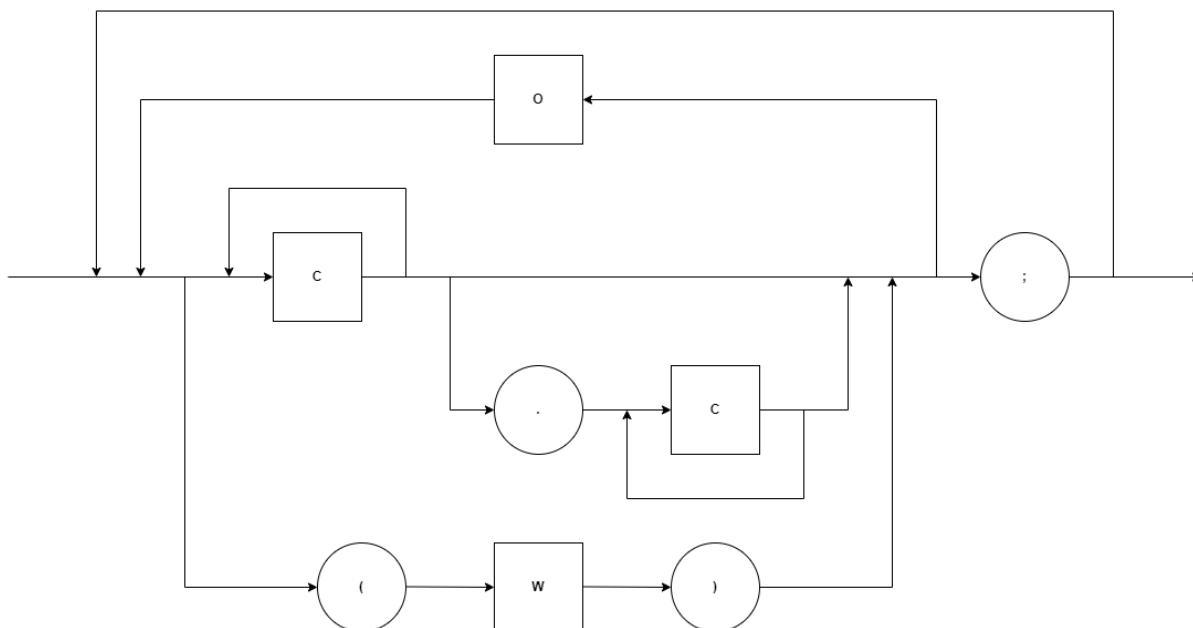


Diagram końcowy gramatyki 6. Zastąpienie produkcji L ich diagramami. Końcowy diagram gramatyki.

Po podstawieniu uproszczonych produkcji, otrzymujemy końcowy diagram gramatyki. Przez występowanie rekurencji w gramatyce dalsze podstawianie diagramu produkcji W nie uprościłoby końcowego diagramu, a w produkcjach O i C możliwe są jedynie alternatywy terminali.

4. Analiza przykładowych wyrażeń

a) $1+1$

Wyrażenie arytmetyczne jest niepoprawne, nie jest zakończone ;.

```
Wpisz wyrażenie/wyrażenia arytmetyczne:  
1+1  
Wyrażenie arytmetyczne nie jest zgodne z gramatyką!  
Wyrażenie arytmetyczne nie jest zakończone ;!
```

b) $(1.1-3.3):2;$

Wyrażenie arytmetyczne jest poprawne.

```
Wpisz wyrażenie/wyrażenia arytmetyczne:  
(1.1-3.3):2;  
Wyrażenie arytmetyczne jest zgodne z gramatyką!
```

c) $(1.2*3)+5-(23.4+3)^3;8:3;$

Wyrażenie arytmetyczne jest poprawne.

```
Wpisz wyrażenie/wyrażenia arytmetyczne:  
(1.2*3)+5-(23.4+3)^3;8:3;  
Wyrażenie arytmetyczne jest zgodne z gramatyką!
```

d) $((2.22^1)+5;$

Wyrażenie arytmetyczne jest niepoprawne, zewnętrzne nawiasy są niedomknięte.

```
Wpisz wyrażenie/wyrażenia arytmetyczne:  
((2.22^1)+5;  
Wyrażenie arytmetyczne nie jest zgodne z gramatyką!  
Niedomknięty nawias w wyrażeniu!
```

e) $1.123*4.;$

Wyrażenie arytmetyczne jest niepoprawne, brakuje cyfr po kropce.

```
Wpisz wyrażenie/wyrażenia arytmetyczne:  
1.123*4.;  
Wyrażenie arytmetyczne nie jest zgodne z gramatyką!  
Wczytano niepoprawny symbol pierwszy produkcji !
```

f) $((3+2):4)3.22;$

Wyrażenie arytmetyczne jest niepoprawne, brakuje operatora pomiędzy wyrażeniami.

```
Wpisz wyrażenie/wyrażenia arytmetyczne:  
((3+2):4)3.22;  
Wyrażenie arytmetyczne nie jest zgodne z gramatyką!
```