**Gramatyka:**S ::= W ; S | e  
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 ::= \* | : | + | - | ^

**Firsts:**FIRST(S) = FIRST(W) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ( , e}  
FIRST(W) = FIRST(P) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ( }  
FIRST(P) = FIRST(R) u { ( } = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ( }  
FIRST(R) = FIRST(L) = {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(C) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}  
FIRST(O)= {\*, : , + , - , ^ }

**I reguła:**S: spełniona bo nie ma alternatywy

W: FIRST(P) and FIRST(P) != pusty zbiór - niespełniona

P: FIRST(R) and { ( } = FIRST(L) and { ( } = pusty zbiór - spełniona

R: FIRST(L) and FIRST(L) != pusty zbiór - niespełniona

L: FIRST(C) and FIRST(C) != pusty zbiór - niespełniona

C: spełniona

O: spełniona

**II reguła:**Spełniona, bo nie ma znaku pustego.

**Poprawiona Gramatyka:**

S ::= W ; Z

Z ::= W ; Z | ε

W ::= P W’

W’ ::= O W | ε

L ::= C L’

L’ ::= L | ε

P ::= R | ( W )

R ::= L R’

R’ ::= . L | ε

C ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

O ::= \* | : | + | - | ^

**I reguła:**W: reg. spełniona bo nie ma alternatywy

W': FIRST(O) and { e } = zbiór pusty -> reg. spełniona

R: : reg. spełniona bo nie ma alternatywy

R': { . } and { e } = zbiór pusty -> reg. spełniona

L: reg. spełniona bo nie ma alternatywy

L': FIRST(L) and { e } = zbiór pusty -> reg.spełniona

**II reguła:**

FOLLOW(W) = { ; } oraz { ) } = { ';' , ')' }

W': FIRST(W') and FOLLOW(W') = FIRST(O) and FOLLOW(W) = {\*, : , + , - , ^ } and { ';' , ')' } = zbiór pusty - spełniona

FOLLOW(R') = FOLLOW(R) = FOLLOW(P) = FOLLOW(W') = FOLLOW(W) = { ';' , ')' }

R': FIRST(R') and FOLLOW(R') = { . } and { ';' , ')' } = zbiór pusty - spełniona  
FIRST(L') = FIRST(C) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

FOLLOW(L') = FOLLOW(L) = FOLLOW(R') oraz FOLLOW(R) = { ';' , ')' }

L': FIRST(L') and FOLLOW(L') = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} and { ';' , ')' } = zbiór pusty - spełniona





