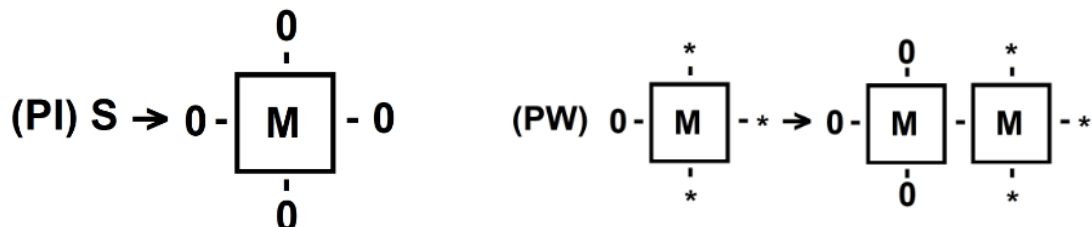


Sprawozdanie TW – Teoria śladów (CW4_zd)

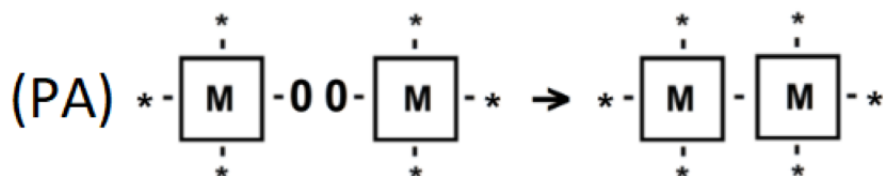
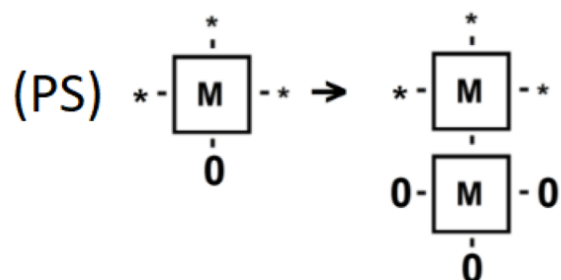
Gabriel Cyganek

1. Produkcje

Podstawowe produkcje:



Dodane przeze mnie produkcje:



(PS) – dołączenie elementu czworokątnego po południowej stronie

(PA) – połączenie dwóch sąsiadujących elementów czworokątnych w poziomie

2. Ciąg produkcji generujący siatkę prostokątną 3x3

Możliwy ciąg produkcji generujący siatkę prostokątną 3x3:

(PI) – (PW) – (PW) – (PS) – (PS) – (PS) – (PS) – (PS) – (PS) – (PA)
 – (PA) – (PA) – (PA)

Generacja:

S

(PI)



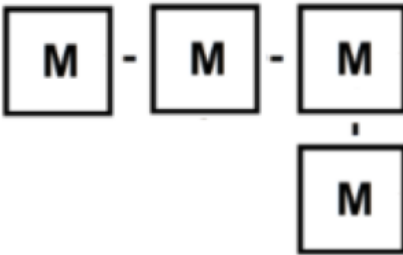
(PW)



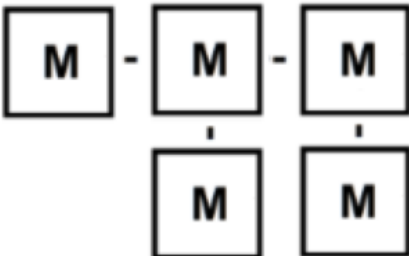
(PW)



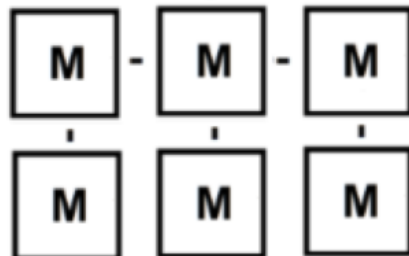
(PS)



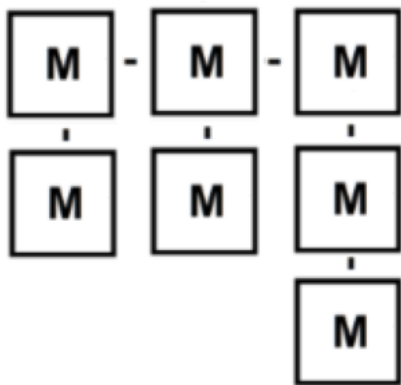
(PS)



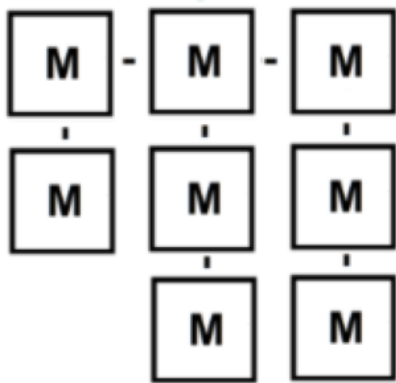
(PS)



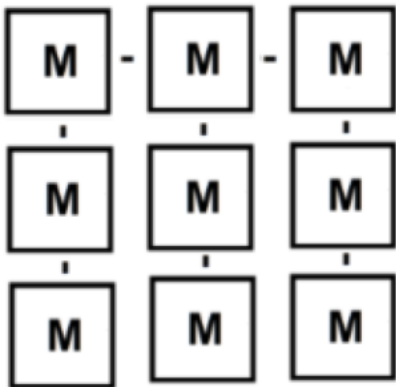
(PS)



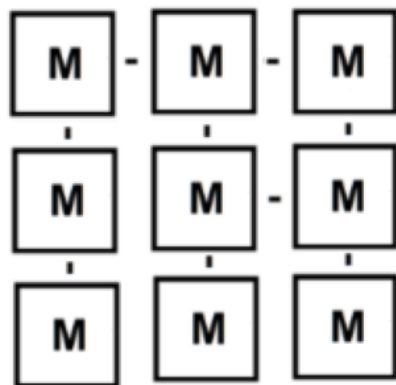
(PS)



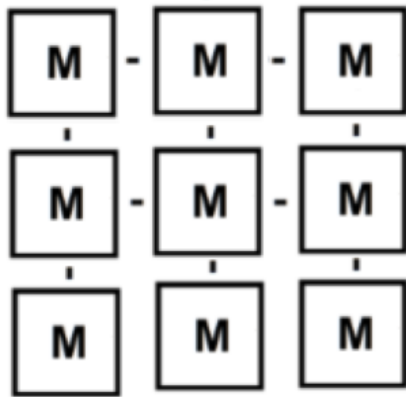
(PS)



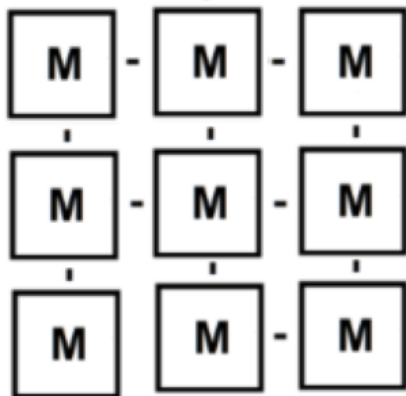
(PA)



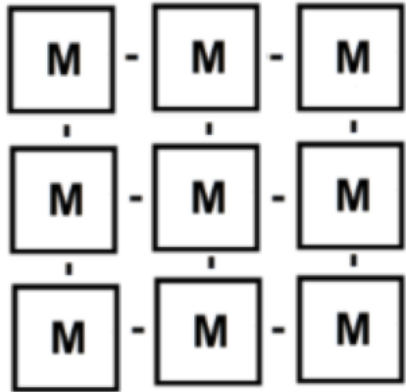
(PA)



(PA)



(PA)



3. Alfabet w sensie teorii śladów

Bazując na ciągu produkcji w gramatyce generującej przedstawioną wyżej siatkę:

$$A = \{PI, PW, PW, PS, PS, PS, PS, PS, PS, PS, PA, PA, PA, PA\}$$

4. Słowo odpowiadające generacji siatki prostokątnej

$$PI, PW, PW, PS, PS, PS, PS, PS, PS, PA, PA, PA, PA$$

5. Relacje zależności i niezależności dla alfabetu w sensie teorii śladów

$$D = \{ \text{sym}\{(PI, PW), (PI, PS), (PW, \textcolor{red}{PW}), (PW, \textcolor{red}{PS}), (\textcolor{red}{PS}, PA), (\textcolor{red}{PS}, \textcolor{blue}{PS}), (PS, \textcolor{green}{PS}), (\textcolor{blue}{PS}, \textcolor{yellow}{PA}), (\textcolor{red}{PW}, \textcolor{yellow}{PS}), (\textcolor{yellow}{PS}, \textcolor{red}{PA}), (\textcolor{yellow}{PS}, \textcolor{purple}{PS}), (\textcolor{purple}{PS}, \textcolor{green}{PA})\} \} \cup I_A$$

$$I = A^2 \setminus D$$

6. Postać normalna Foaty

$$FNF = [PI][PW, PS][\textcolor{red}{PW}, \textcolor{red}{PS}, \textcolor{green}{PS}][\textcolor{yellow}{PS}, \textcolor{blue}{PS}, PA][\textcolor{red}{PA}, \textcolor{yellow}{PA}, \textcolor{purple}{PS}][\textcolor{green}{PA}]$$

7. Algorytm współbieżny w oparciu o postać normalną Foaty

Wykorzystując wyznaczoną postać normalną Foaty można uszeregować wątki wykonujące poszczególne produkcje w efektywny sposób. W mojej implementacji na początku wykonuje się produkcja PI . Następnie do momentu wypełnienia całej siatki $N \times N$ wykonują się równolegle niezależnymi grupami produkcja PW (dopóki siatka nie osiągnie zadanej szerokości N), produkcje PA , które łączą dodane przez poprzednią grupę produkcji PS elementy z elementami po ich wschodnich stronach, oraz produkcje PS wykonane w każdej kolumnie na elemencie, który nie ma połączenia na południowej stronie (jeśli wysokość kolumny nie przekroczy N).

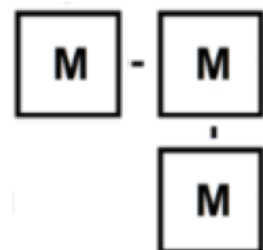
Przykładowo dla siatki 3×3 procedury wykonają się w następujących grupach:

S

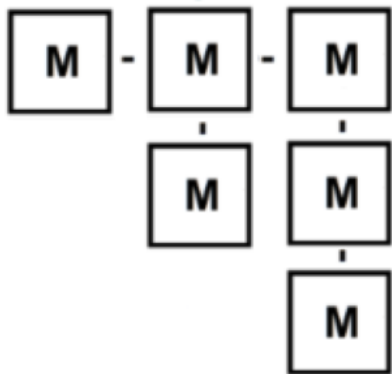
[PI]



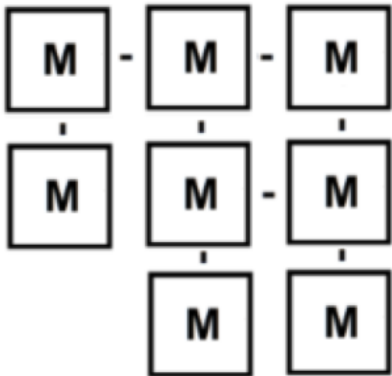
[PW, PS]



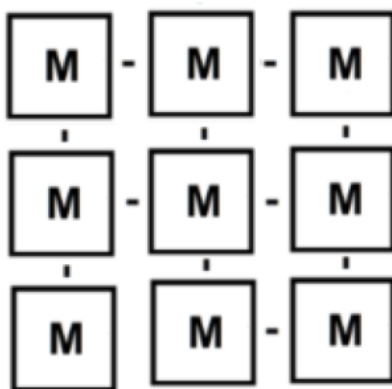
[PW, PS, PS]



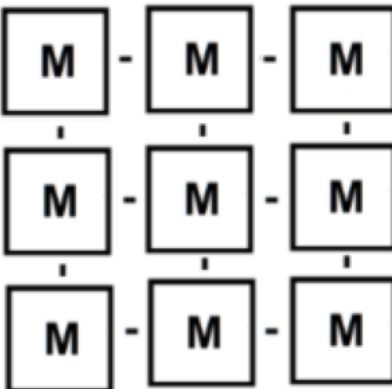
[PS, PS, PA]



[PA, PA, PS]



[PA]



Po prawej wynik wykonania
zaimplementowanego programu
tworzącego siatkę $N \times N$ dla $N = 3$
według powyższego algorytmu

```
PI
M

PW
PS
M---M
|
M

PW
PS
PS
M---M---M
| |
M M
|
M

PA
PS
PS
M---M---M
| | |
M M---M
| |
M M

PA
PA
PS
M---M---M
| | |
M---M---M
| | |
M M---M

PA
M---M---M
| | |
M---M---M
| | |
M---M---M

done
```