



# **Teoria współbieżności**

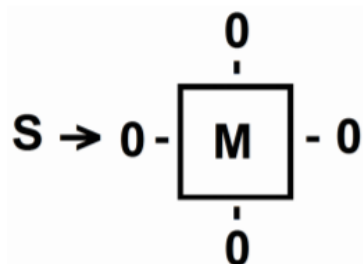
## **Generowanie siatek 2D**

Kamil Koczera

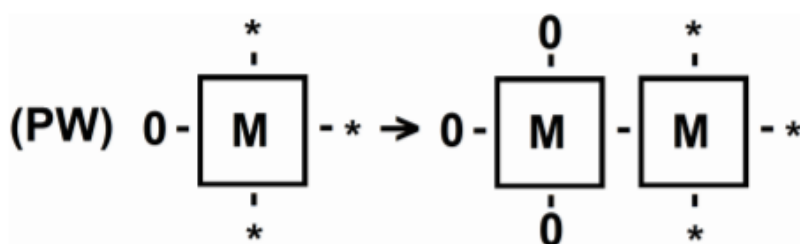
## Produkcje:

W celu wykonania zadania, do produkcji zawartych w instrukcji:

(PI):

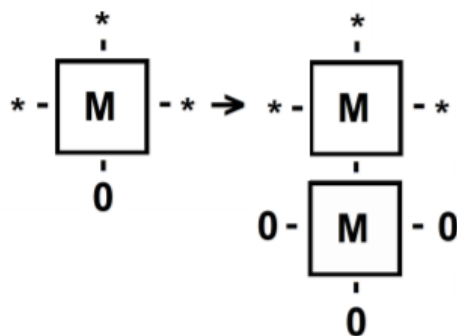


(PW):

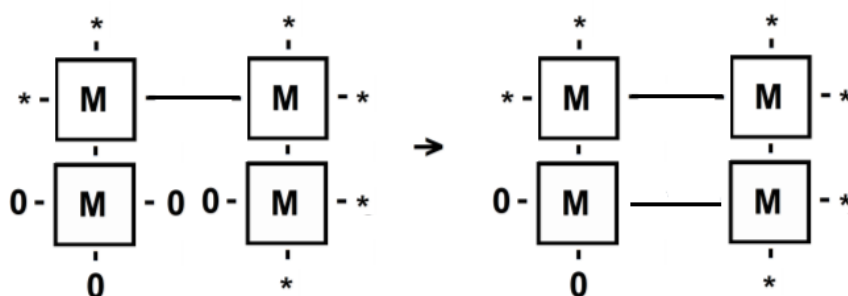


Dodałem następujące:

(PS):



(PC):



## Generacja siatki 3x3:

Możliwa np. za pomocą ciągu produkcji:

(PI) -> (PW) -> (PS) -> (PW) -> (PS) -> (PC) -> (PS) -> (PC) -> (PS) -> (PS) -> (PC) -> (PS)  
-> (PC)

Generacja przebiega następująco (uproszcilem nieco zapis, ze względu na trudności związane z przedstawianiem operacji w dokumencie elektronicznym):

S -> (PI)

M

-> (PW)

M - M

-> (PS)

M - M  
|  
M

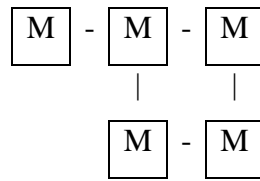
-> (PW)

M - M - M  
|  
M

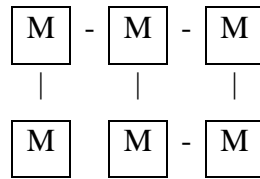
-> (PS)

M - M - M  
| |  
M M

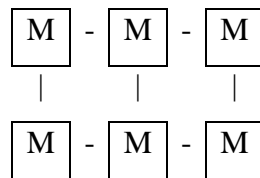
-> (PC)



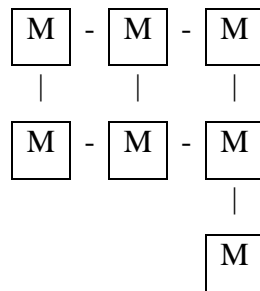
-> (PS)



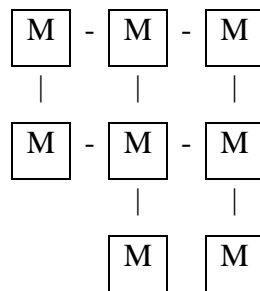
-> (PC)



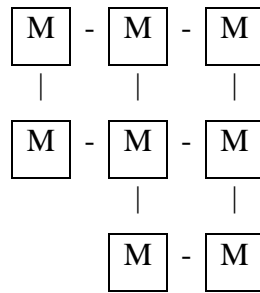
-> (PS)



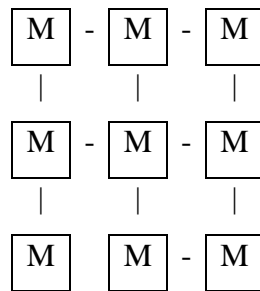
-> (PS)



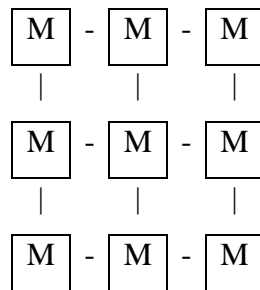
-> (PC)



-> (PS)



-> (PC)



## Alfabet w sensie teorii śladów:

Może zostać zdefiniowany w następujący sposób (powtarzające się produkcje rozróżnione za pomocą indeksów dolnych):

$$A = \{PI\} \cup \{PW_n, n \in \{1, 2\}\} \cup \{PS_n, n \in \{1, 2, 3, 4, 5, 6\}\} \cup \{PC_n, n \in \{1, 2, 3, 4\}\}$$

## Słowo odpowiadające generacji siatki:

PI, PW<sub>1</sub>, PS<sub>1</sub>, PW<sub>2</sub>, PS<sub>2</sub>, PC<sub>1</sub>, PS<sub>3</sub>, PC<sub>2</sub>, PS<sub>4</sub>, PS<sub>5</sub>, PC<sub>3</sub>, PS<sub>6</sub>, PC<sub>4</sub>

## Relacja (nie)zależności dla alfabetu A:

$D = \text{sym}\{ \{ (PI, PW_1), (PI, PS_1), (PW_1, PW_2), (PW_1, PS_2), (PS_2, PC_1), (PW_2, PS_3), (PS_3, PC_2), (PS_1, PS_4), (PS_2, PS_5), (PS_5, PC_3), (PS_3, PS_6), (PS_6, PC_4) \}^+ \} \cup I_A$

$$I = A \times A - D$$

## Postać normalna Foaty:

$$FNF = [PI][PW_1, PS_1][PW_2, PS_2, PS_4][PS_3, PC_1, PS_5][PC_2, PS_6, PC_3][PC_4]$$

Posługując się powyższymi klasami Foaty otrzymana zostaje generacja o następującym przebiegu (dla siatki 3x3):

S -> (PI)

M

-> (PW) -> (PS)

M - M  
|  
M

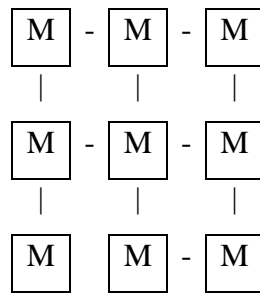
-> (PW) -> (PS) -> (PS)

M - M - M  
|     |  
M     M  
      |  
      M

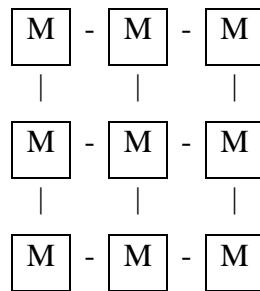
-> (PS) -> (PC) -> (PS)

M - M - M  
|     |     |  
M     M - M  
      |     |  
      M     M

-> (PC) -> (PS) -> (PC)



-> (PC)



## Algorytm współbieżny

Zaprojektowany został algorytm, który prezentuje się następująco (dla generowania siatki NxN):

- Do elementu wysuniętego najbardziej na zachód przyłącz nowy element z jego lewej (zachodniej) strony. Następnie do tego samego elementu (nie nowo przyłączonego) przyłącz nowy element z dołu (południa). Do każdego elementu znajdującego się na przekątnej (w kierunku południowo-wschodnim) (w stosunku do elementu, na którym wykonywaliśmy operacje przyłączania) przyłącz jego prawego (wschodniego) sąsiada oraz przyłącz nowy element z dołu (południa). Powtarzaj ten krok do osiągnięcia żądanej wysokości (szerokości).
- Do elementu wysuniętego najbardziej na zachód (niemającego sąsiada z dołu (południa)) przyłącz do niego jego prawego (wschodniego) sąsiada oraz nowy element z dołu (południa). Powtórz tę operację dla każdego elementu znajdującego się na przekątnej (w kierunku południowo-wschodnim) (w stosunku do elementu, na którym wykonywaliśmy operacje przyłączania), chyba, że jest to ostatni element (najbardziej wysunięty na południe) – wówczas pomiń operację przyłączania z dołu (południa).

Powtarzaj ten krok do momentu ukończenia siatki (powinien zostać wykonany łącznie N razy).

Implementacja powyższego algorytmu znajduje się w dołączonym archiwum. Jest to zmodyfikowana wersja programu z laboratorium.

Przykład generacji siatki dla  $N = 5$ :

S -> (PI)

M

-> (PW) -> (PS)

M - M  
|  
M

-> (PS) -> (PW) -> (PS)

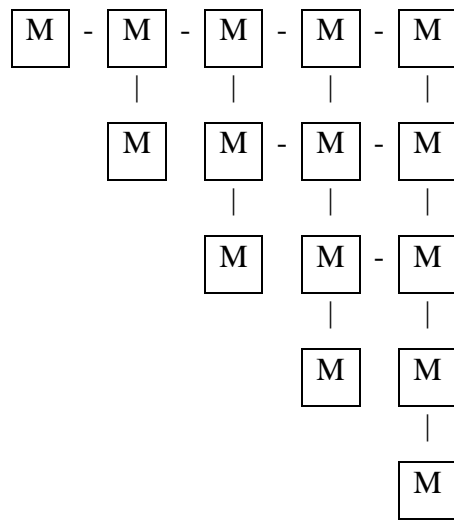
M - M - M  
| |  
M M  
|  
M

-> (PW) -> (PS) -> (PS) -> (PS) -> (PC)

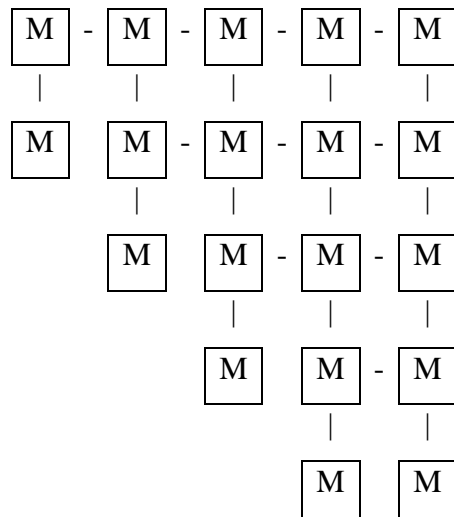
M - M - M - M  
| | |  
M M - M  
| |  
M M  
|  
M



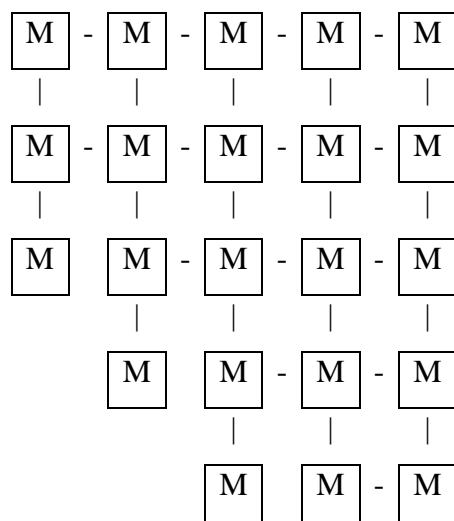
-> (PW) -> (PS) -> (PC) -> (PC) -> (PS) -> (PS) -> (PS)



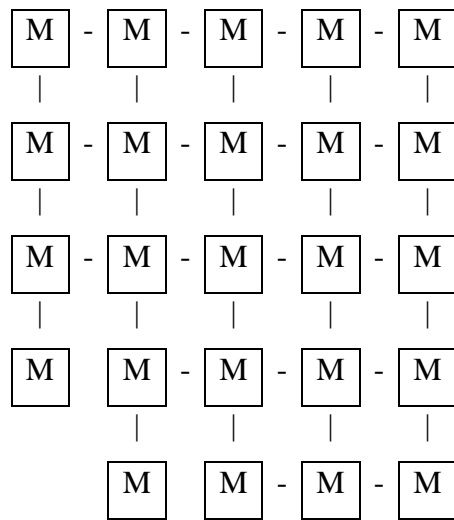
-> (PS) -> (PS) -> (PS) -> (PS) -> (PC) -> (PC) -> (PC)



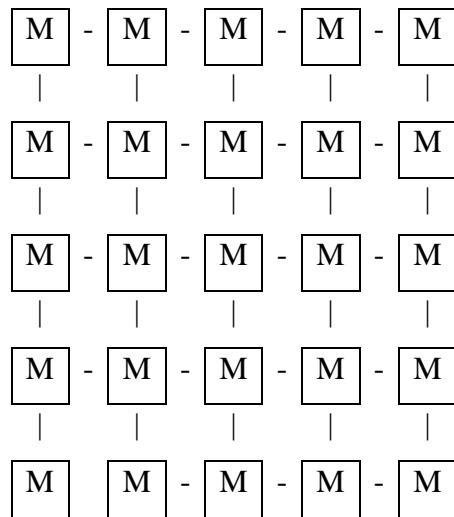
-> (PS) -> (PS) -> (PS)-> (PC) -> (PC) -> (PC) -> (PC)



-> (PS) -> (PS) -> (PC) -> (PC) -> (PC)



-> (PS) -> (PC) -> (PC)



-> (PC)

