

Programowanie współbieżne i rozproszone

WYKŁAD 11

Graf osiągalności

- Graf osiągalności – graf przedstawiający wszystkie możliwe stany w jakich może znaleźć się sieć.
 - Węzły grafu → znakowania sieci osiągalne ze znakowania początkowego.
 - Łuki grafu etykietowane są nazwami tranzycji powodującymi określone zmiany znakowania sieci.

Graf osiągalności

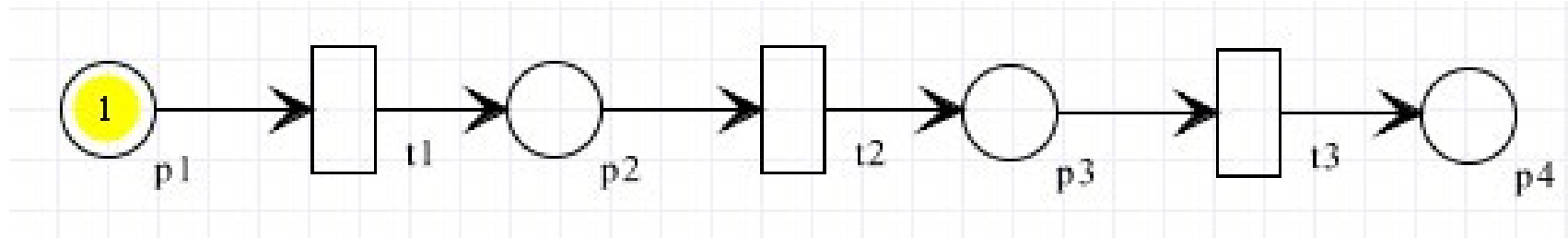
- Sieć Petriego: $N = (P, T, F, W, M_0)$
- Graf osiągalności: $G = (V, A)$

$$V = [M_0 >$$

$$A = \{ (M, t, M') : M, M' \in [M_0 > \wedge M [t > M'] \}$$

$M [t > M'$ - tranzycja t powoduje zmianę znakowania sieci z M na M'

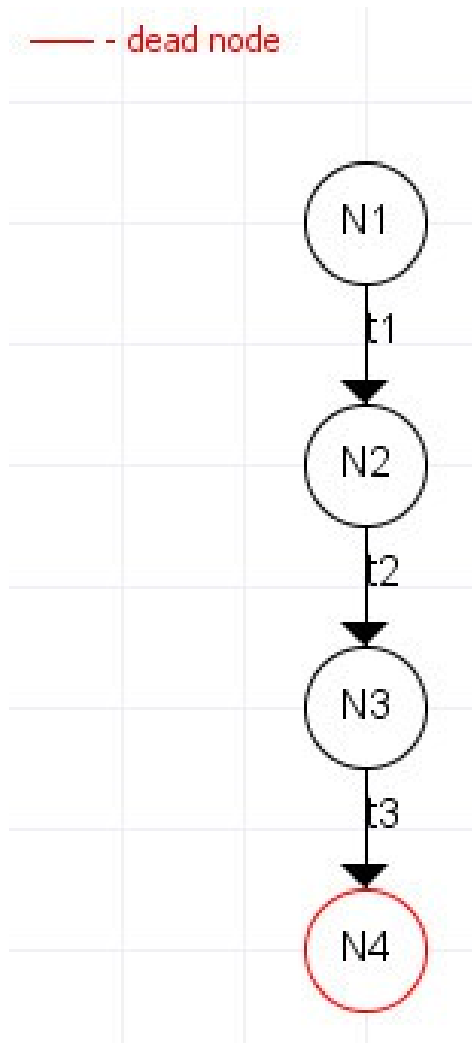
Modelowanie sekwencyjności



Modelowanie sekwencyjności

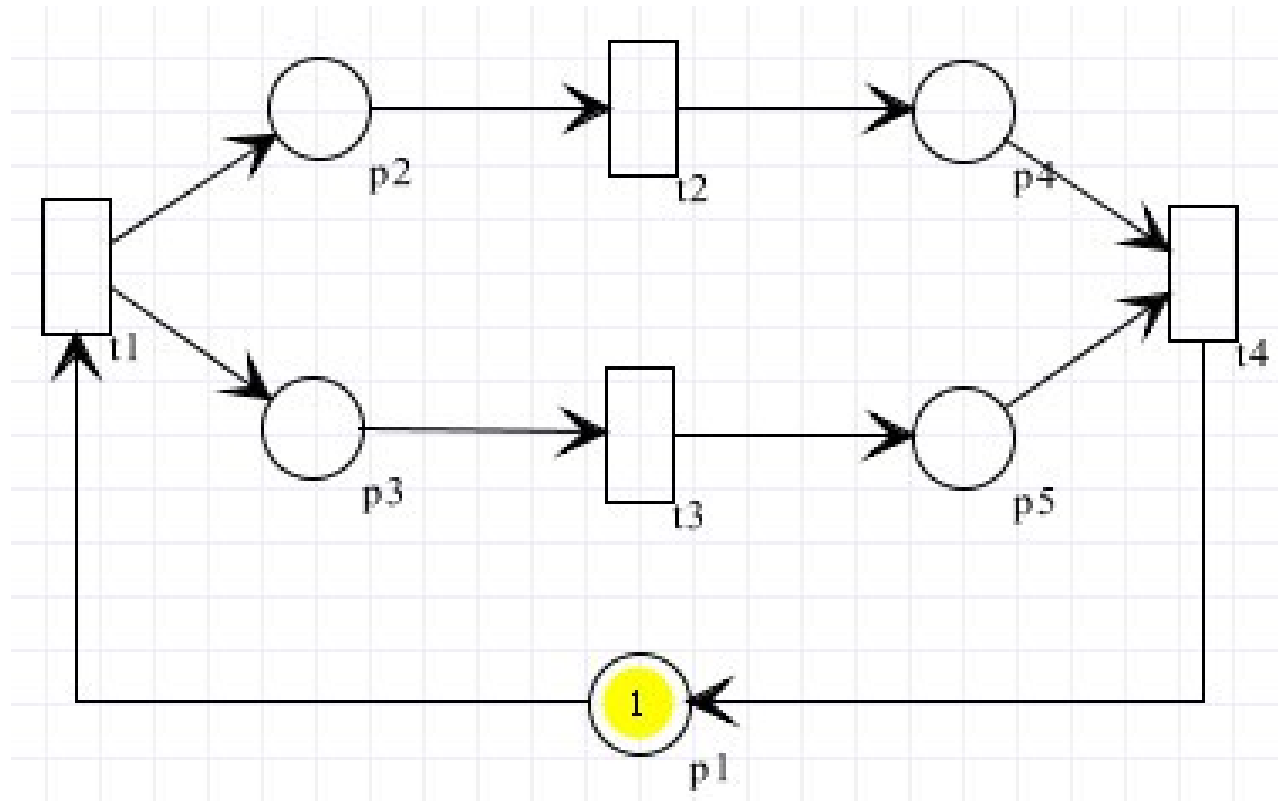
Graf osiągalności:

— - dead node



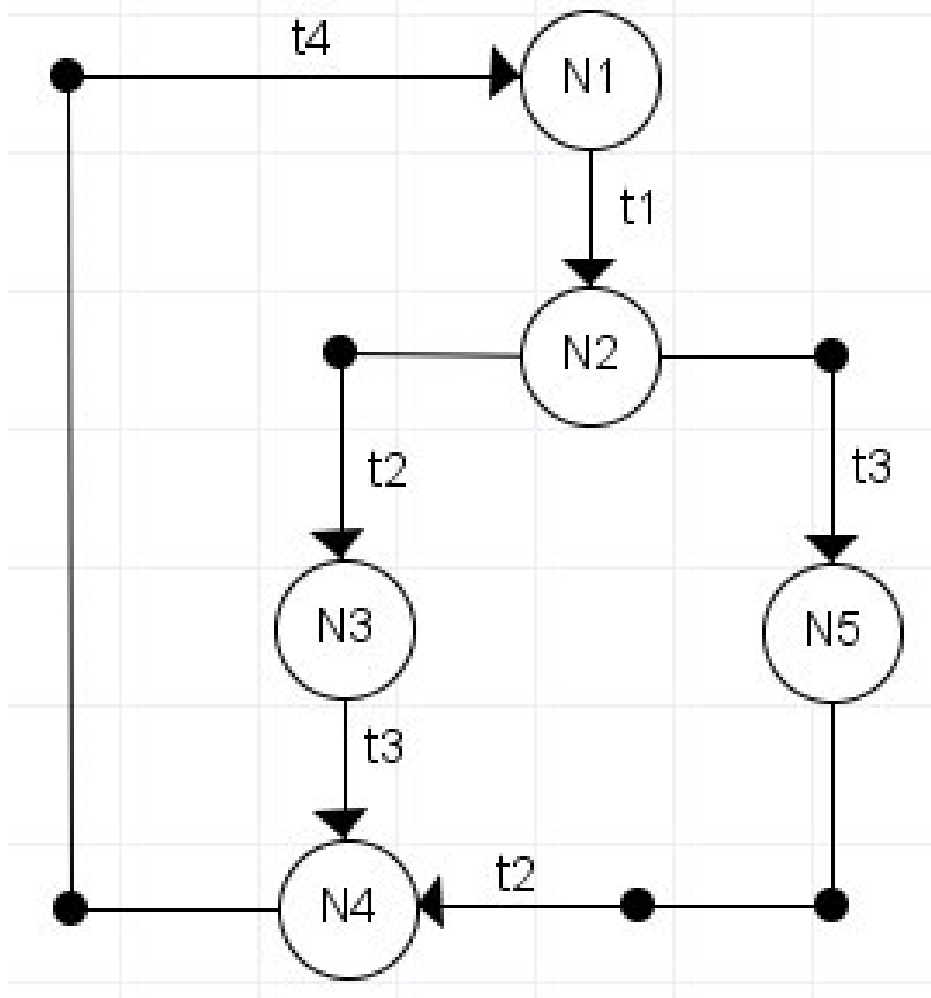
N/P	p1	p2	p3	p4
N1	1.0	0.0	0.0	0.0
N2	0.0	1.0	0.0	0.0
N3	0.0	0.0	1.0	0.0
N4	0.0	0.0	0.0	1.0

Modelowanie współbieżności



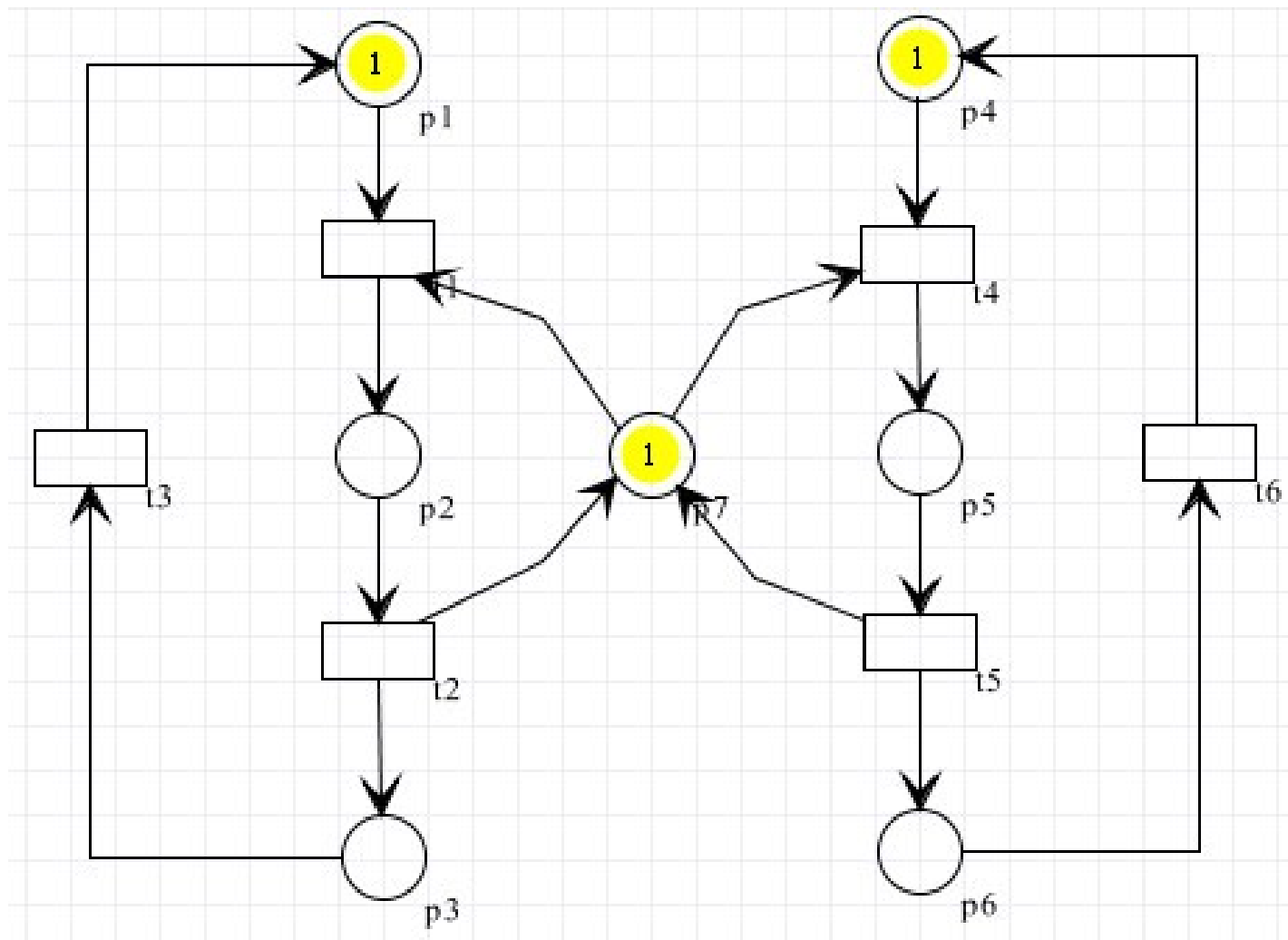
Modelowanie współbieżności

Graf osiągalności:



N/P	p1	p2	p3	p4	p5
N1	1.0	0.0	0.0	0.0	0.0
N2	0.0	1.0	1.0	0.0	0.0
N3	0.0	0.0	1.0	1.0	0.0
N4	0.0	0.0	0.0	1.0	1.0
N5	0.0	1.0	0.0	0.0	1.0

Modelowanie wzajemnego wykluczania



Krzysztof Panczerz

Programowanie współbieżne i rozproszone

Modelowanie wzajemnego wykluczania

- Stany procesu 1: p1, p2, p3.
- Stany procesu 2: p4, p5, p6.
- Miejsce p7 reprezentuje semafor.
- Interpretacja znaczników w miejscach:
 - p2, p5 → przebywanie procesów w sekcjach krytycznych
 - p1, p3, p4, p6 → przebywanie procesów w sekcjach lokalnych

Modelowanie wzajemnego wykluczania

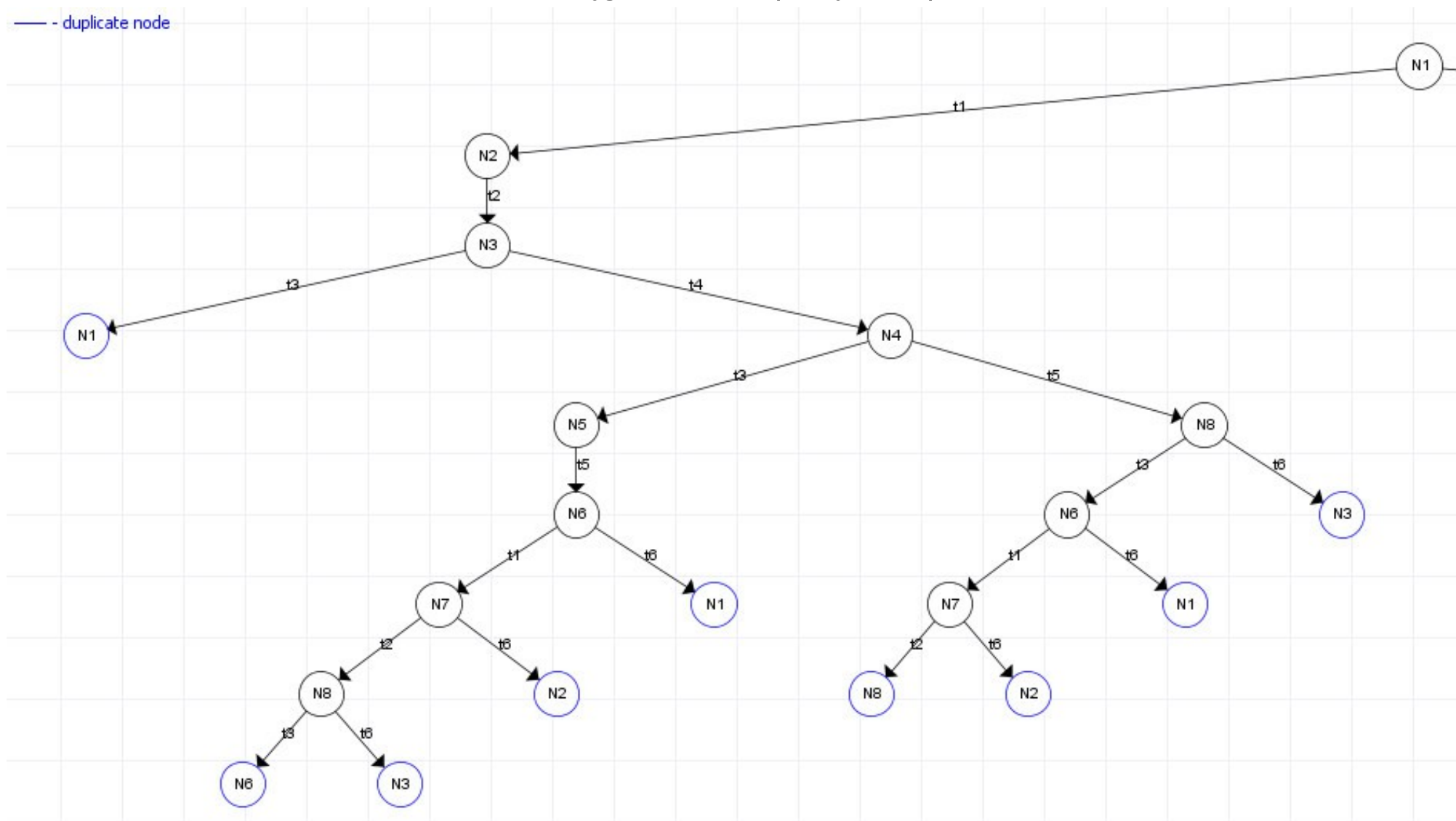
Graf osiągalności (węzły):

N/P	p1	p2	p3	p4	p5	p6	p7
N1	1.0	0.0	0.0	1.0	0.0	0.0	1.0
N2	0.0	1.0	0.0	1.0	0.0	0.0	0.0
N3	0.0	0.0	1.0	1.0	0.0	0.0	1.0
N4	0.0	0.0	1.0	0.0	1.0	0.0	0.0
N5	1.0	0.0	0.0	0.0	1.0	0.0	0.0
N6	1.0	0.0	0.0	0.0	0.0	1.0	1.0
N7	0.0	1.0	0.0	0.0	0.0	1.0	0.0
N8	0.0	0.0	1.0	0.0	0.0	1.0	1.0

Modelowanie wzajemnego wykluczania

Graf osiągalności (część 1):

— - duplicate node

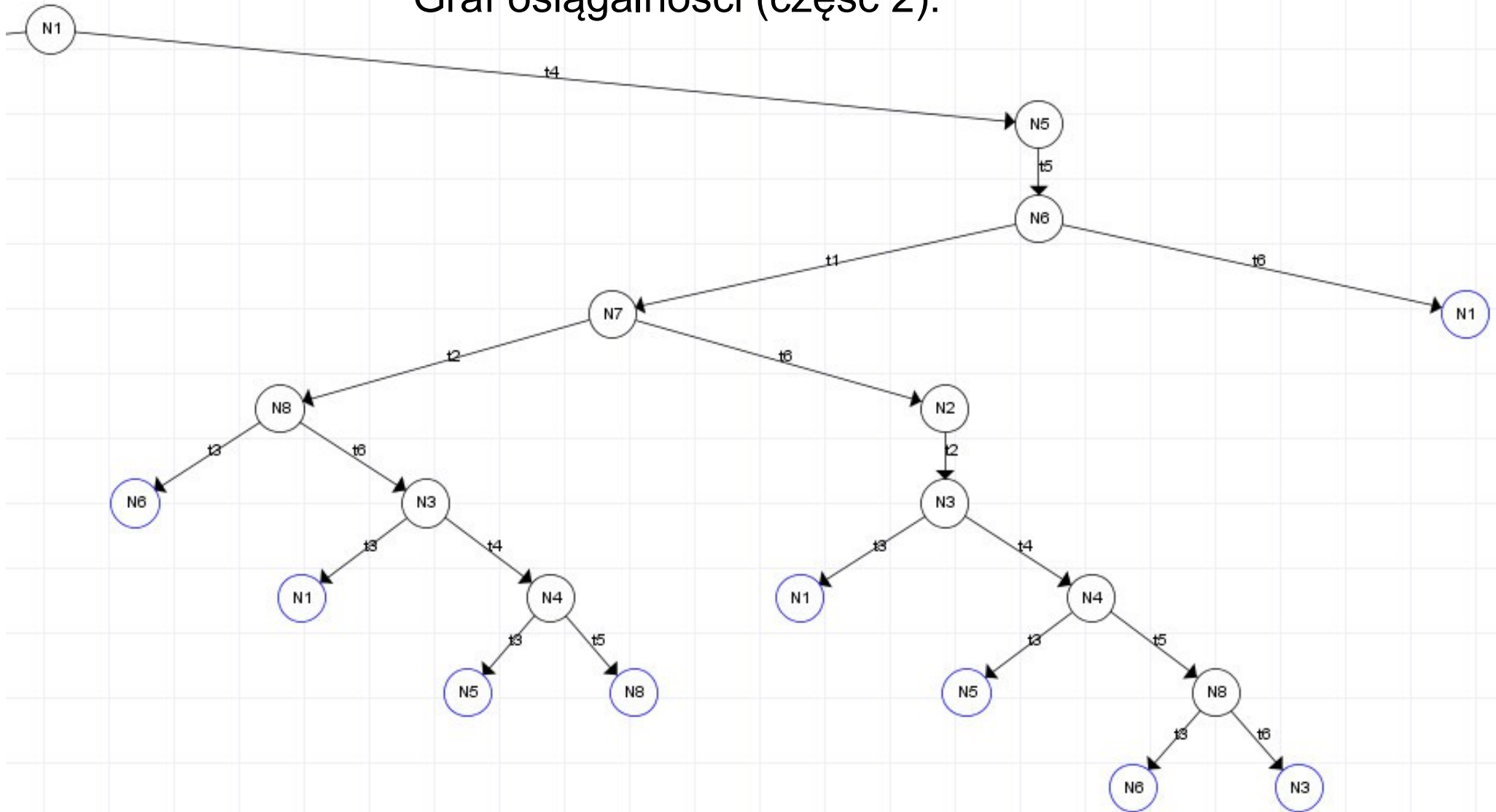


Krzysztof Pancierz

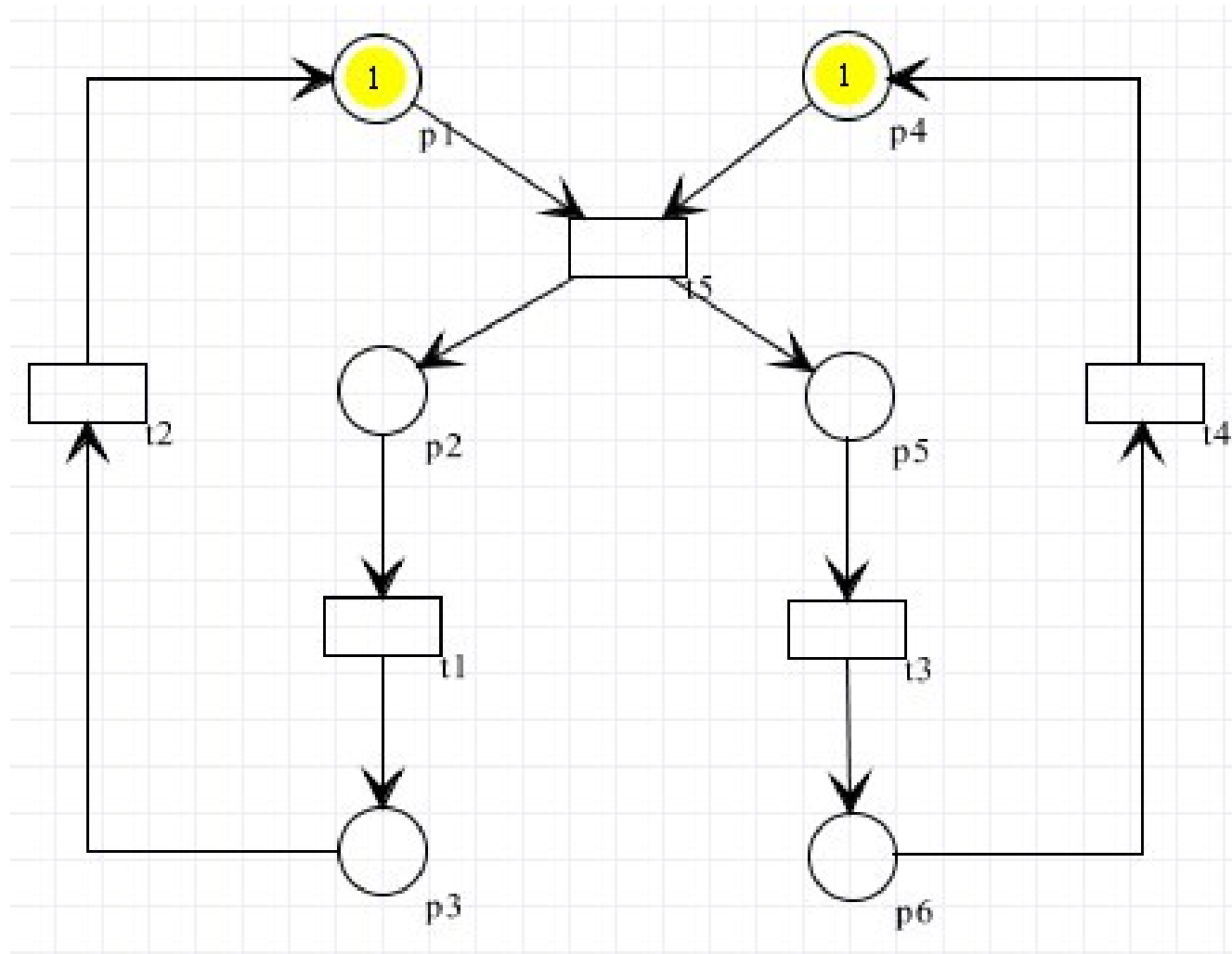
Programowanie współbieżne i rozproszone

Modelowanie wzajemnego wykluczania

Graf osiągalności (część 2):



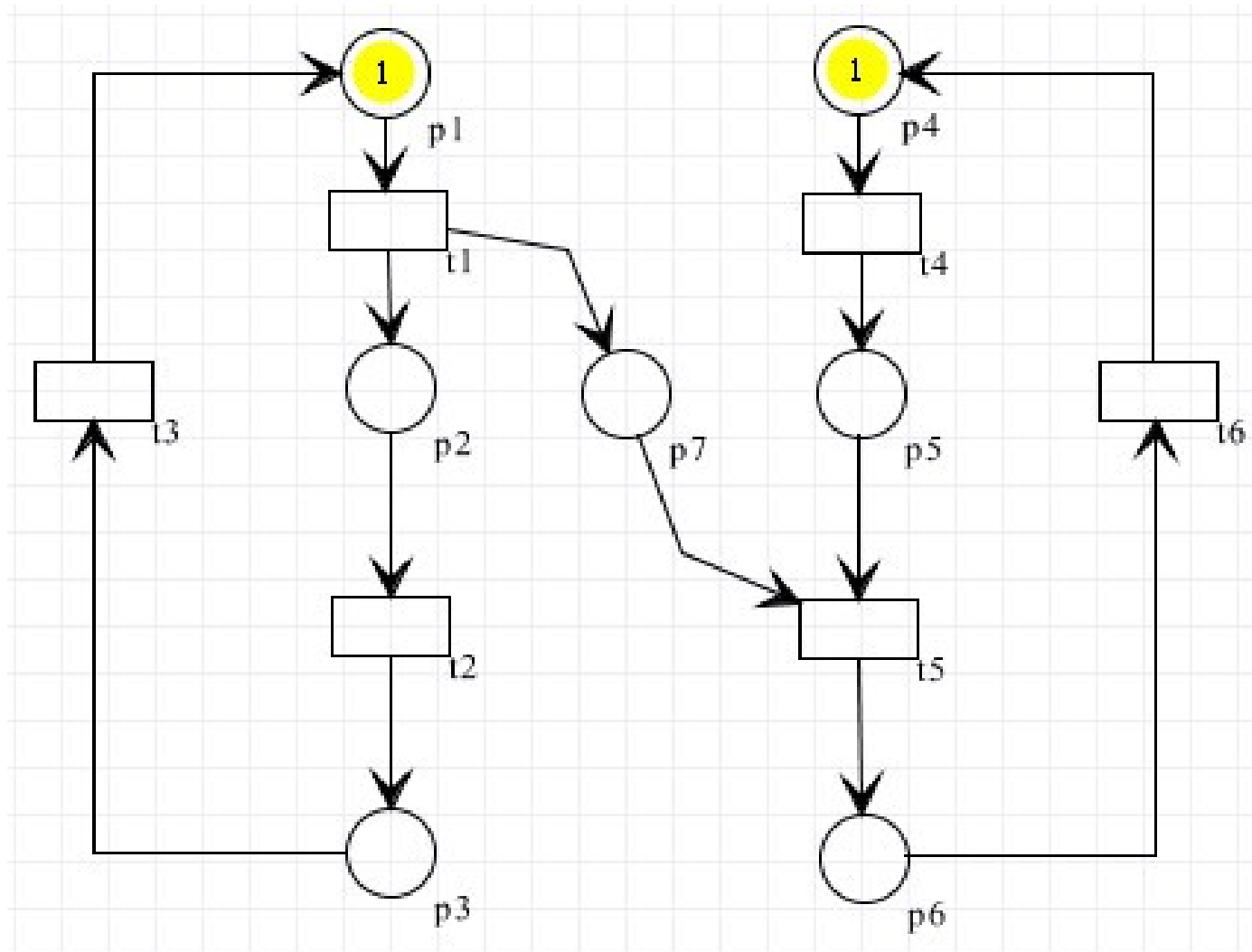
Modelowanie komunikacji synchronicznej procesów



Modelowanie komunikacji synchronicznej procesów

- Stany procesu 1: p1, p2, p3.
- Stany procesu 2: p4, p5, p6.
- Tranzycja t5 synchronizuje procesy.
- Miejsca p1, p4 reprezentują dotarcie procesów (odpowiednio 1 i 2) do miejsca synchronizacji.

Modelowanie komunikacji asynchronicznej procesów

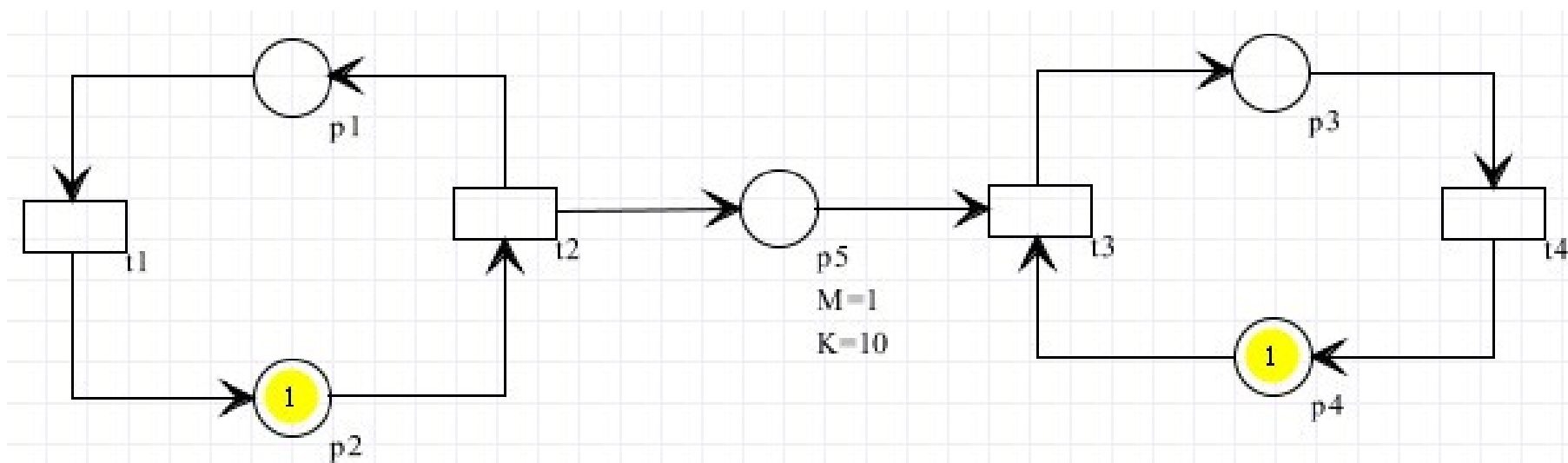


Modelowanie komunikacji asynchronicznej procesów

- Stany procesu 1: p1, p2, p3.
- Stany procesu 2: p4, p5, p6.
- Miejsce p7 reprezentuje bufor dla komunikatów przekazywanych z procesu 1 do procesu 2.

Model producent-konsument

Jeden producent – jeden konsument

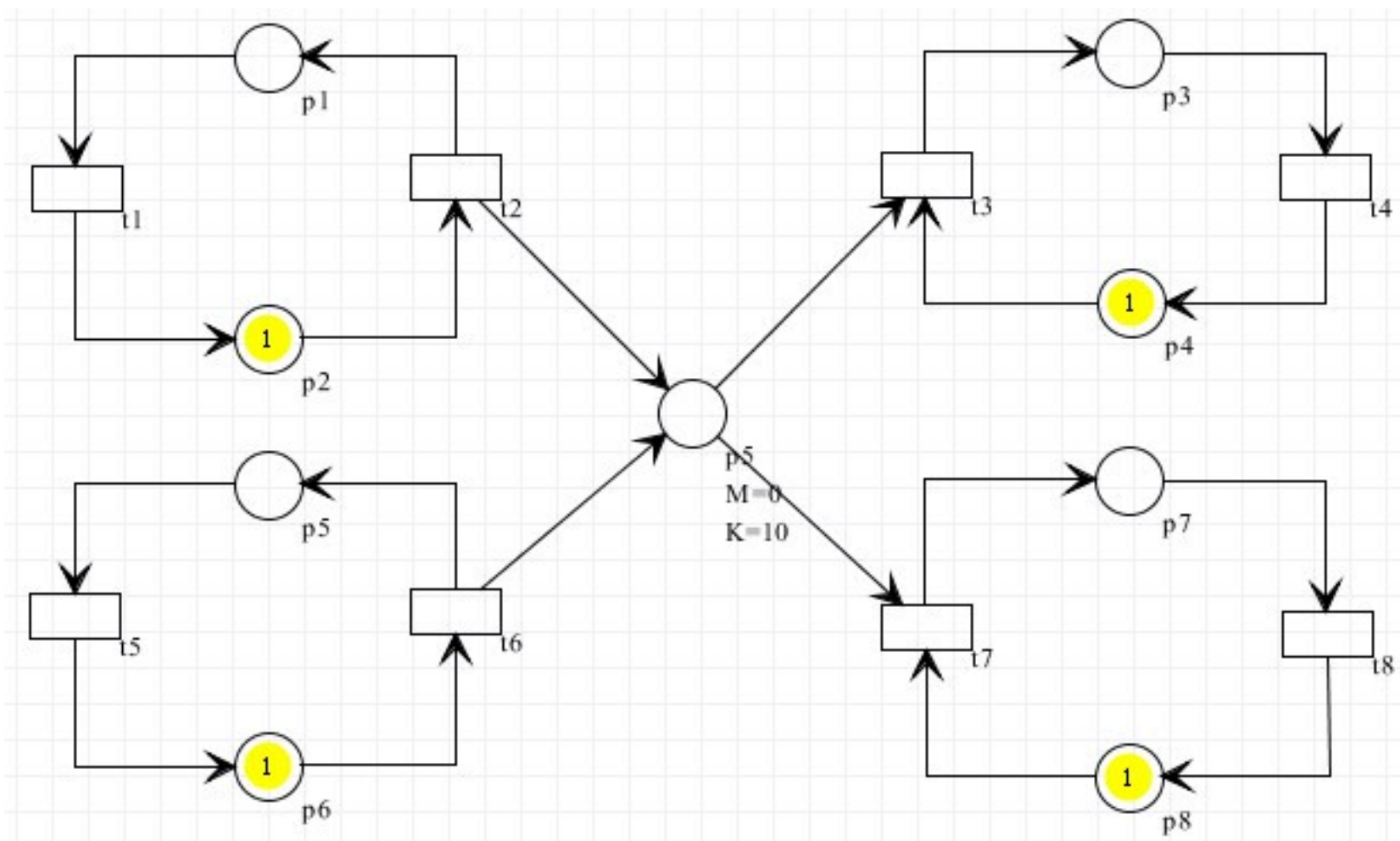


Model producent-konsument

- Stany producenta: p_1 , p_2 .
- Stany konsumenta: p_3 , p_4 .
- Miejsce p_5 (o zadanej pojemności) reprezentuje bufor.

Model producent-konsument

Dwóch producentów – dwóch konsumentów

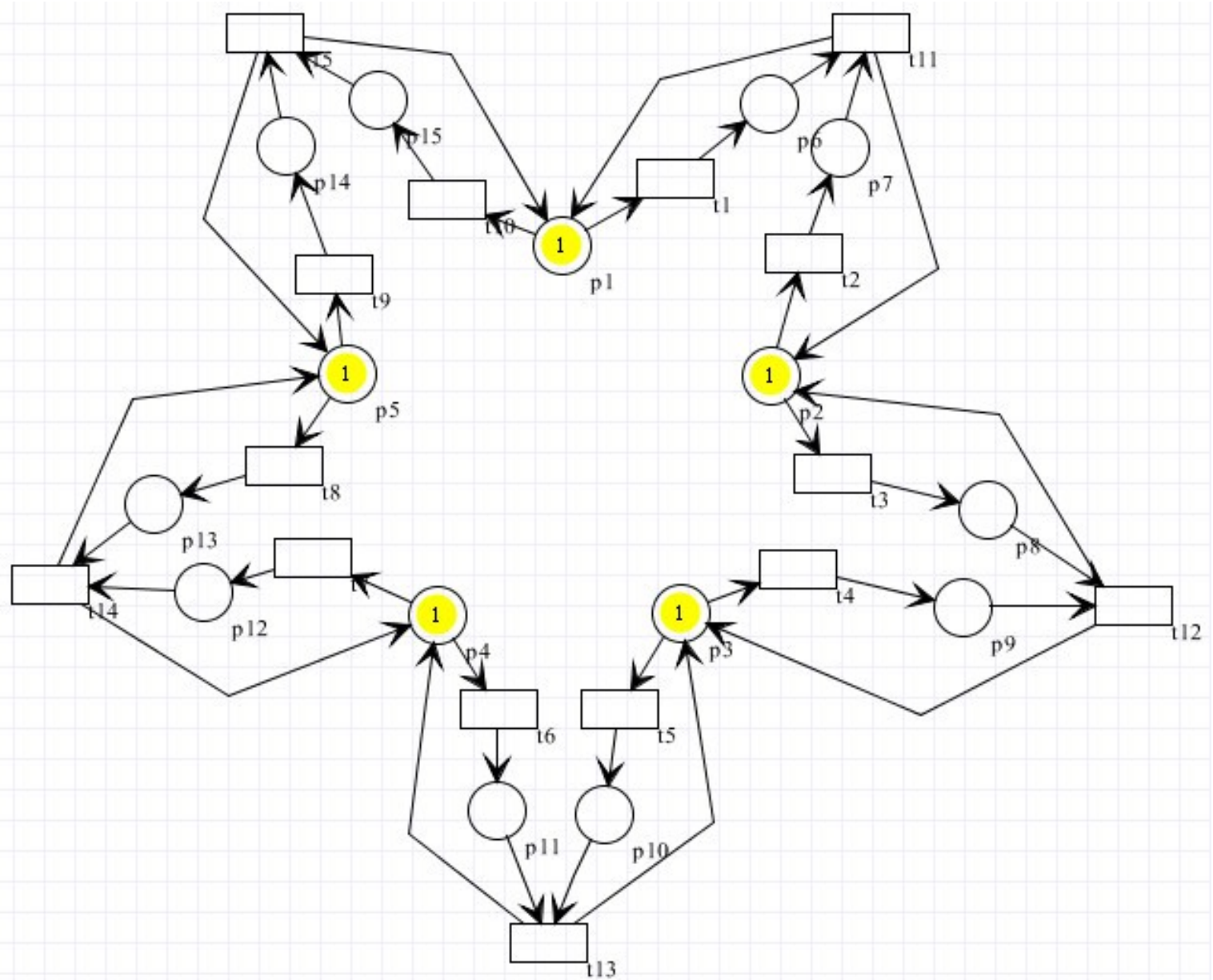


Krzysztof Pancierz

Programowanie współbieżne i rozproszone

Model pięciu filozofów

Wersja 1



Krzysztof Pancierz

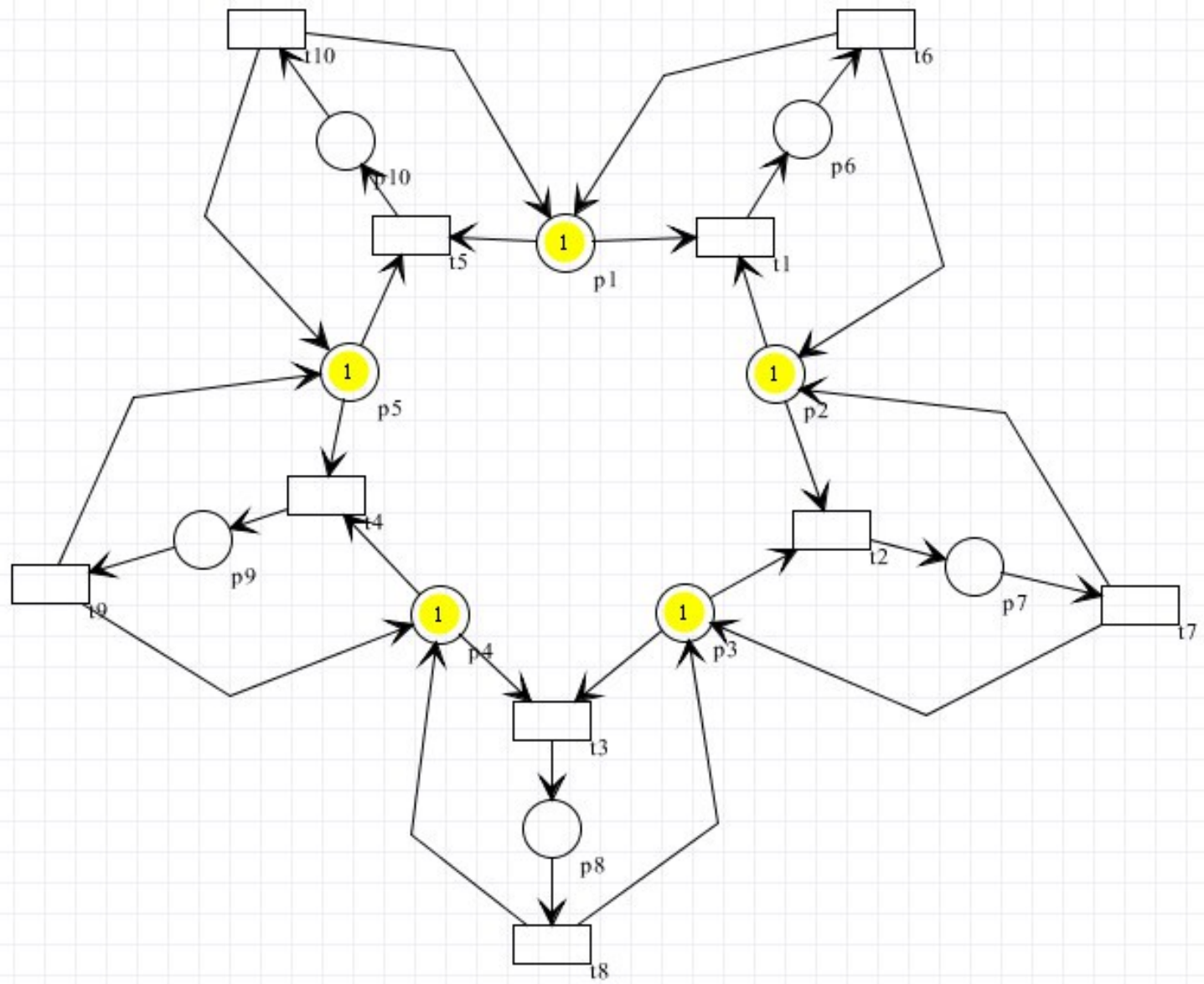
Programowanie współbieżne i rozproszone

Model pięciu filozofów

- Interpretacja znaczników w miejscach:
 - p1, p2, p3, p4, p5 → widelce znajdują się na stole
 - pozostałe miejsca → widelce znajdują się w posiadaniu określonych filozofów
- **UWAGA:** Sieć nie jest żywa (istnieje znakowanie martwe w sieci gdy każdy z filozofów podniósł tylko jeden widelec).

Model pięciu filozofów

Wersja 2



Krzysztof Pancierz

Programowanie współbieżne i rozproszone

Model pięciu filozofów

- Interpretacja znaczników w miejscach:
 - p1, p2, p3, p4, p5 → widelce znajdują się na stole
 - pozostałe miejsca → określani filozofowie myślą.
- **UWAGA:** Sieć jest żywa.