



Algorytm Maekawy wzajemnego wykluczania

Algorytm Maekawy – wprowadzenie

- 1) Jeżeli proces chce wejść do sekcji krytycznej, potrzebuje pozwolenia tylko od pewnego podzbioru wszystkich procesów
- 2) Proces może naraz wysłać tylko jedną odpowiedź
- 3) Wiadomości są dostarczane w skończonym czasie
- 4) FIFO order



Zbiór żądań

Warunki, które musi spełnić zbiór żądań (N – liczba procesów):

(a) $(\forall i \forall j : i \neq j, 1 \leq i, j \leq N :: R_i \cap R_j \neq \emptyset)$.

(b) $(\forall i : 1 \leq i \leq N :: S_i \in R_i)$

(c) $(\forall i : 1 \leq i \leq N :: |R_i| = K)$

(d) Dowolny proces S_j zawarty jest w K zbiorach $R_i, 1 \leq i, j \leq N$



Zbiór żądań

Warunki, które musi spełnić zbiór żądań (N – liczba procesów):

(a) $(\forall i \forall j : i \neq j, 1 \leq i, j \leq N :: R_i \cap R_j \neq \emptyset)$.

(b) $(\forall i : 1 \leq i \leq N :: S_i \in R_i)$

(c) $(\forall i : 1 \leq i \leq N :: |R_i| = K)$

(d) Dowolny proces S_j zawarty jest w K zbiorach $R_i, 1 \leq i, j \leq N$

$$N = K(K-1)+1$$

$$|R_i| = \sqrt{N}$$



Algorytm Maekawy

Żądanie sekcji krytycznej

- 1) Węzeł S_i wysyła wiadomość $REQUEST(ts,i)$ do wszystkich węzłów ze zbioru R_i



Algorytm Maekawy

Żądanie sekcji krytycznej

- 1) Węzeł S_i wysyła wiadomość $REQUEST(ts,i)$ do wszystkich węzłów ze zbioru R_i
- 2) Gdy węzeł S_j otrzyma $REQUEST(ts,i)$ i nie wysłał żadnej wiadomości $RESPONSE$ od czasu otrzymania wiadomości $RELEASE$, to wysyła do węzła S_i $RESPONSE(j)$. W przeciwnym przypadku dodaje $REQUEST(ts,i)$ do kolejki żądań oczekujących.



Algorytm Maekawy

Żądanie sekcji krytycznej

- 1) Węzeł S_i wysyła wiadomość $REQUEST(ts,i)$ do wszystkich węzłów ze zbioru R_i
- 2) Gdy węzeł S_j otrzyma $REQUEST(ts,i)$ i nie wysłał żadnej wiadomości $RESPONSE$ od czasu otrzymania wiadomości $RELEASE$, to wysyła do węzła S_i $RESPONSE(j)$. W przeciwnym przypadku dodaje $REQUEST(ts,i)$ do kolejki żądań oczekujących.

Wykonanie sekcji krytycznej

- 3) Węzeł S_i uzyskuje dostęp do sekcji krytycznej tylko w przypadku otrzymania wiadomości $RESPONSE$ od wszystkich węzłów ze zbioru R_i



Algorytm Maekawy

Zwolnienie sekcji krytycznej

- 4) Po wykonaniu sekcji krytycznej, węzeł S_i wysyła wiadomość $RELEASE(i)$ do wszystkich węzłów ze zbioru R_i



Algorytm Maekawy

Zwolnienie sekcji krytycznej

- 4) Po wykonaniu sekcji krytycznej, węzeł S_i wysyła wiadomość $RELEASE(i)$ do wszystkich węzłów ze zbioru R_i
- 5) Gdy S_j otrzyma wiadomość $RELEASE(i)$, wysyła wiadomość $RESPONSE$ do następnego węzła w kolejce i usuwa to żądanie z kolejki.



Dowód poprawności

wzajemne wykluczanie

Założmy, że dwa węzły S_i i S_j znajdują się jednocześnie w sekcji krytycznej



Dowód poprawności

wzajemne wykluczanie

Założmy, że dwa węzły S_i i S_j znajdują się jednocześnie w sekcji krytycznej

- a) Węzeł S_i musiał otrzymać wiadomości RESPONSE od wszystkich węzłów ze zbioru R_i i jednocześnie węzeł S_j musiał otrzymać wiadomości RESPONSE od wszystkich węzłów ze zbioru R_j



Dowód poprawności

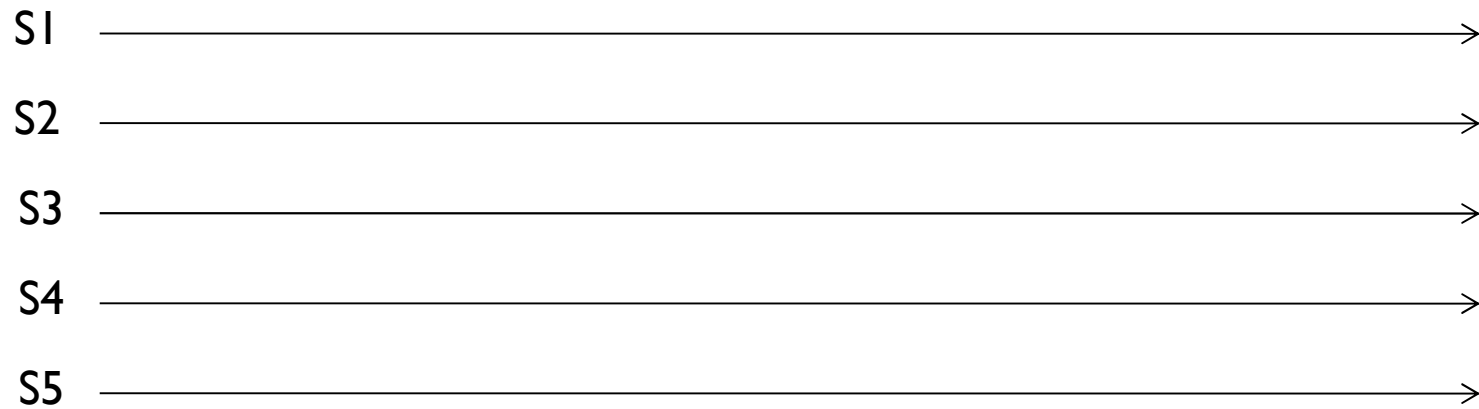
wzajemne wykluczanie

Założmy, że dwa węzły S_i i S_j znajdują się jednocześnie w sekcji krytycznej

- a) Węzeł S_i musiał otrzymać wiadomości RESPONSE od wszystkich węzłów ze zbioru R_i i jednocześnie węzeł S_j musiał otrzymać wiadomości RESPONSE od wszystkich węzłów ze zbioru R_j
- b) Jeśli $S_i \cap S_j = \{S_k\}$, to węzeł S_k musiał wysłać wiadomość RESPONSE zarówno do S_i i S_j , co jest sprzecznością

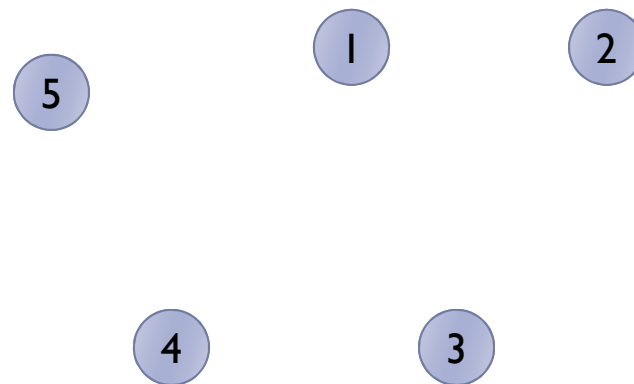


Algorytm Maekawy – przykład

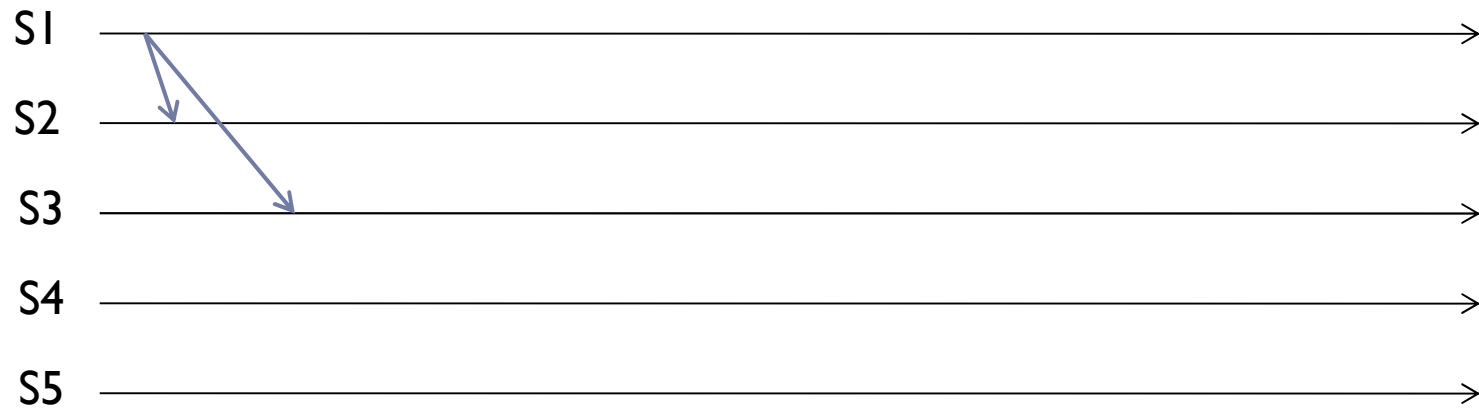


$$R_1 = \{1,2,3\}$$

$$R_5 = \{3,4,5\}$$

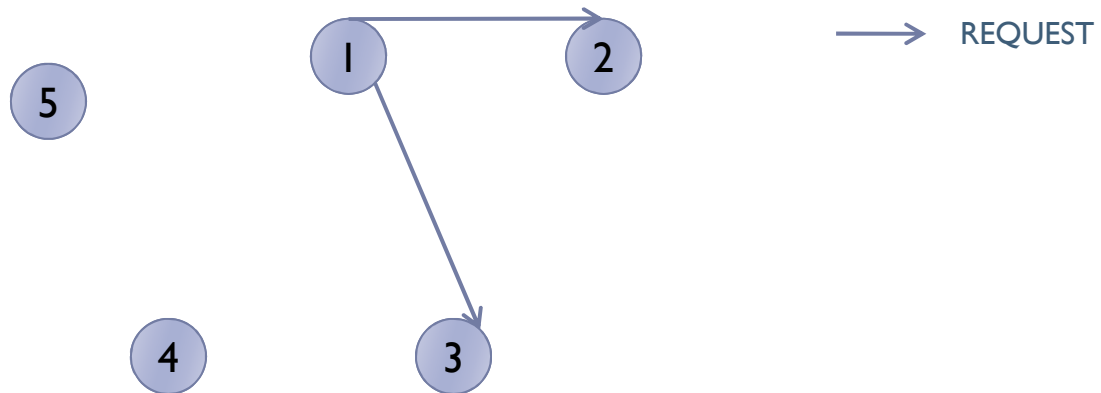


Algorytm Maekawy – przykład

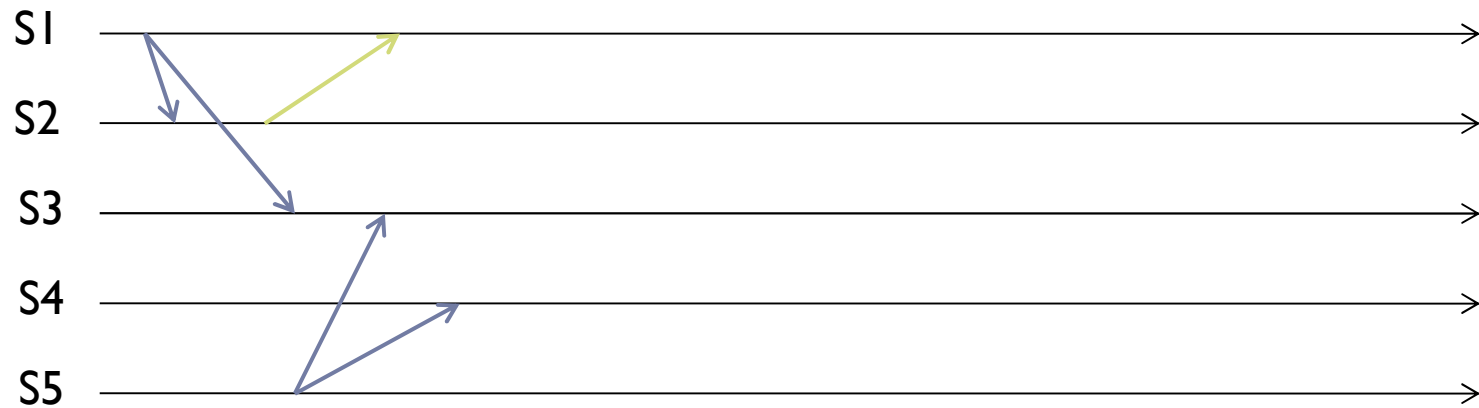


$R_1 = \{1,2,3\}$

$R_5 = \{3,4,5\}$

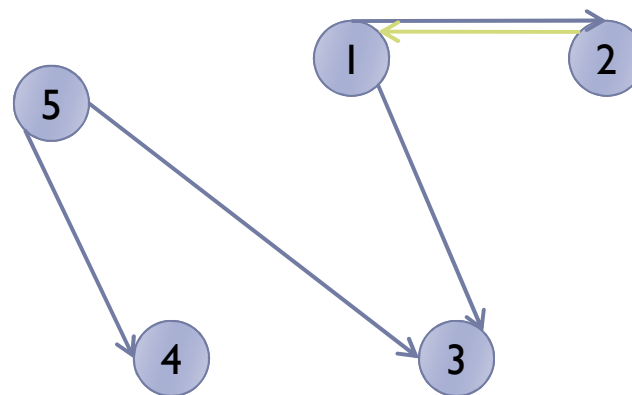


Algorytm Maekawy – przykład



$R_1 = \{1,2,3\}$

$R_5 = \{3,4,5\}$

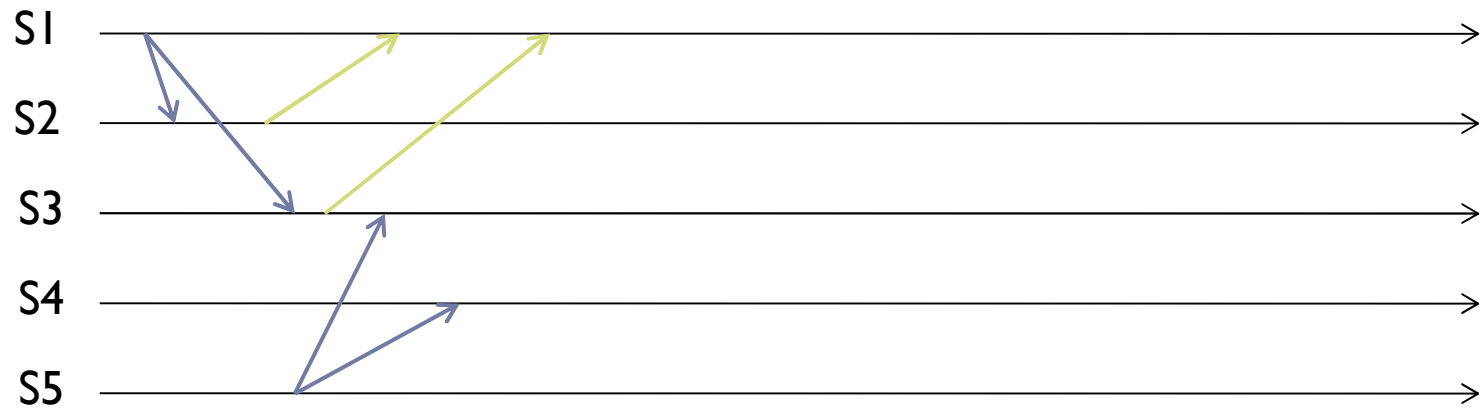


→ REQUEST

→ RESPONSE

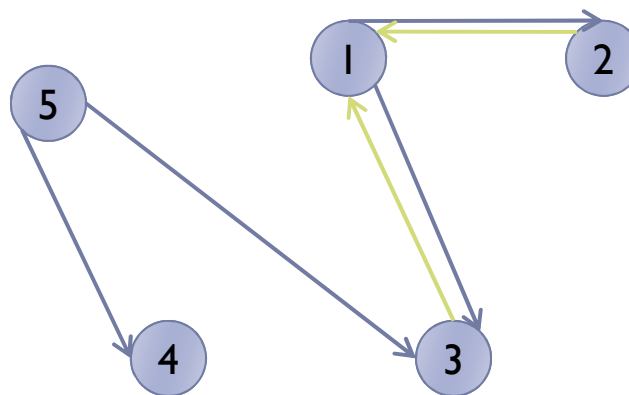


Algorytm Maekawy – przykład



$R_1 = \{1,2,3\}$

$R_5 = \{3,4,5\}$

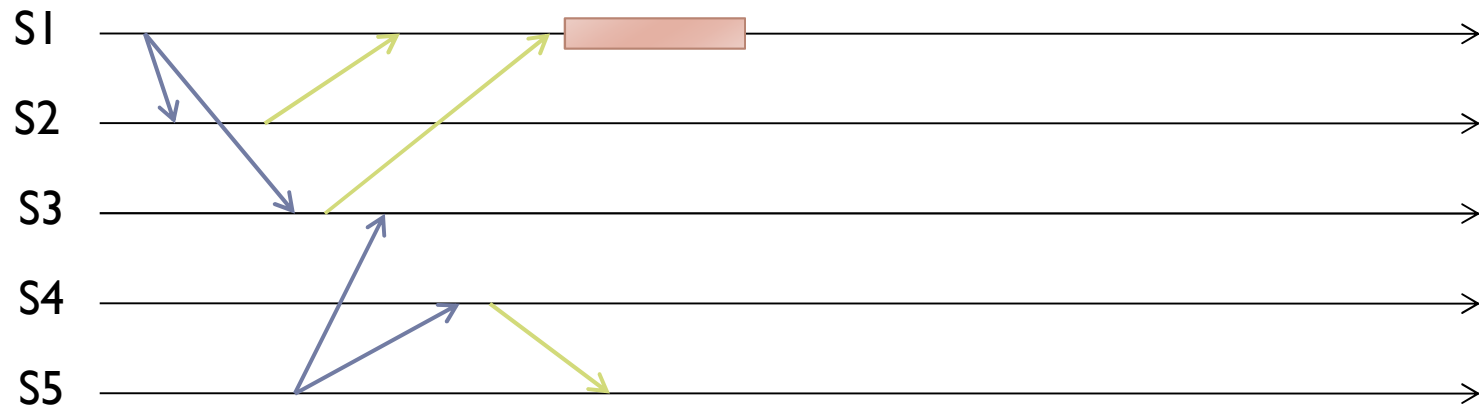


→ REQUEST

→ RESPONSE

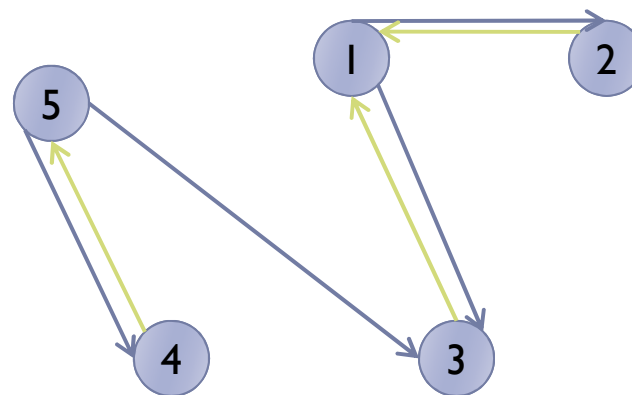


Algorytm Maekawy – przykład



$R_1 = \{1,2,3\}$

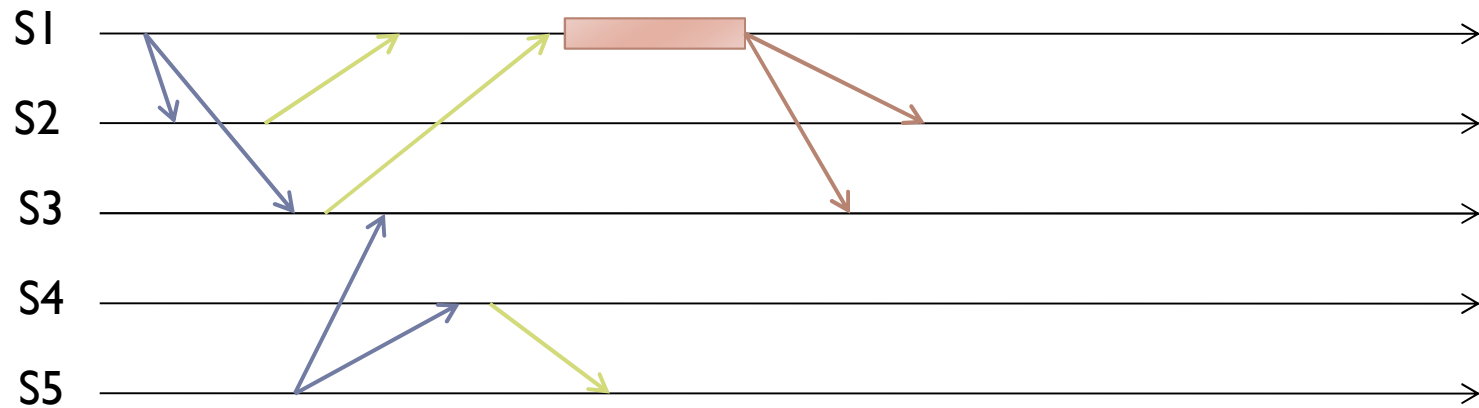
$R_5 = \{3,4,5\}$



→ REQUEST

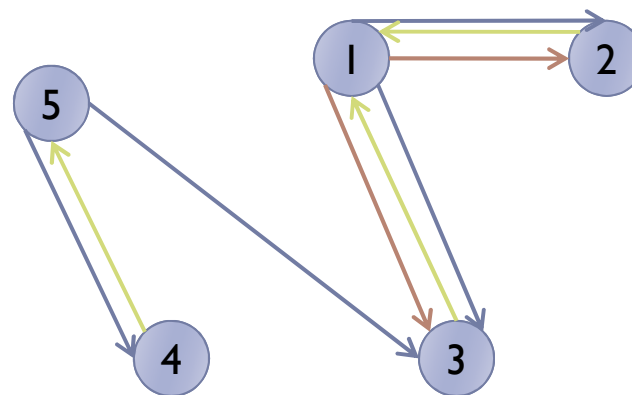
→ RESPONSE

Algorytm Maekawy – przykład



$R_1 = \{1,2,3\}$

$R_5 = \{3,4,5\}$

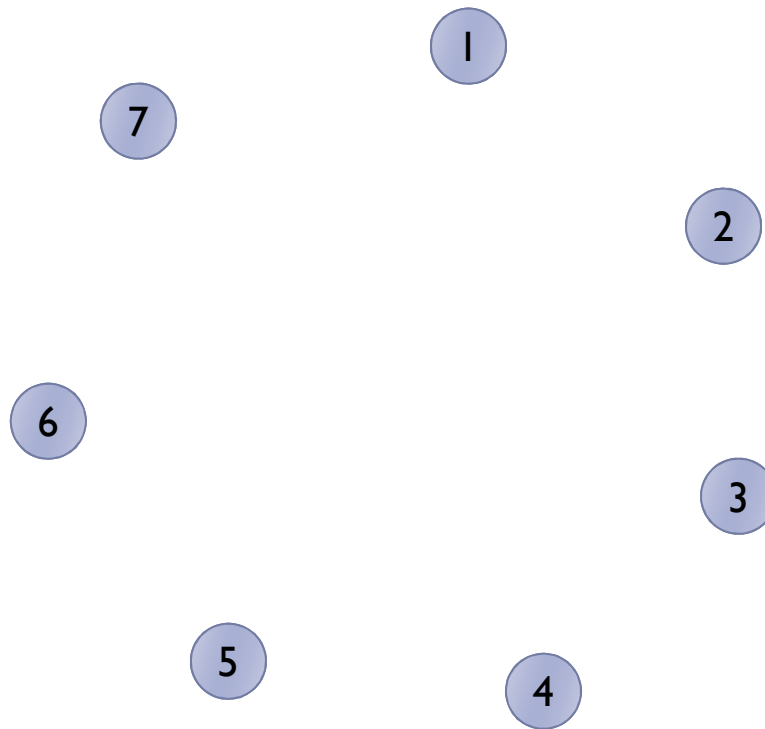


→ REQUEST

→ RESPONSE

→ RELEASE

Zakleszczenie - przykład



$$R_1 = \{1,2,3\}$$

$$R_2 = \{2,4,6\}$$

$$R_3 = \{3,5,6\}$$

$$R_4 = \{1,4,5\}$$

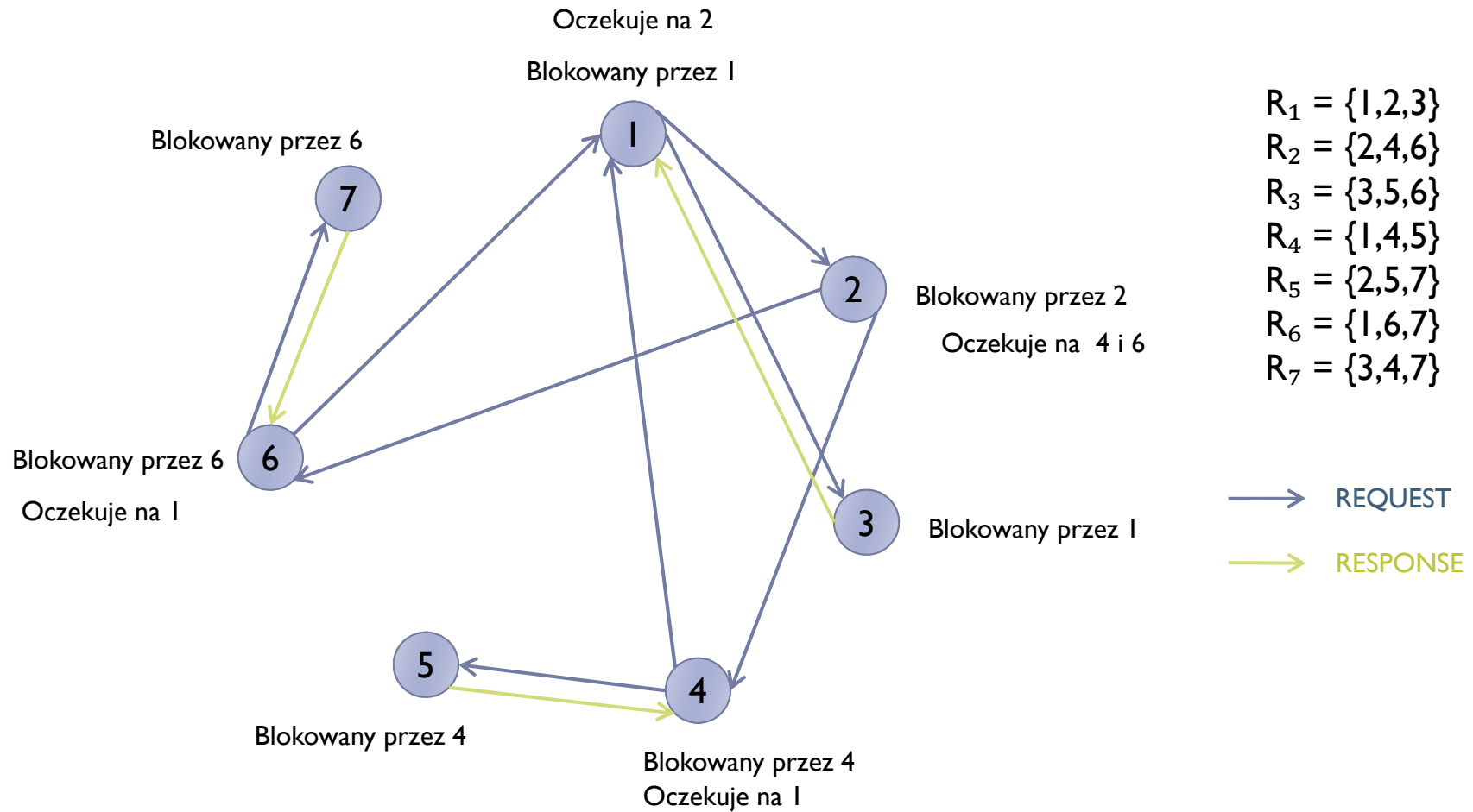
$$R_5 = \{2,5,7\}$$

$$R_6 = \{1,6,7\}$$

$$R_7 = \{3,4,7\}$$



Zakleszczenie - przykład



Algorytm Maekawy c.d.

Przeciwdziałanie zakleszczeniu

- 7) Gdy węzeł S_j otrzyma $REQUEST(ts,i)$ od węzła S_i , a S_j jest już blokowany na rzecz węzła S_k , to jeśli $REQUEST(ts,i)$ ma mniejszy priorytet, węzeł S_j wysyła $FAILED(j)$ do S_i , a w przeciwnym przypadku S_j wysyła $INQUIRE(j)$ do S_k



Algorytm Maekawy c.d.

Przeciwdziałanie zakleszczeniu

- 7) Gdy węzeł S_j otrzyma $REQUEST(ts,i)$ od węzła S_i , a S_j jest już blokowany na rzecz węzła S_k , to jeśli $REQUEST(ts,i)$ ma mniejszy priorytet, węzeł S_j wysyła $FAILED(j)$ do S_i , a w przeciwnym przypadku S_j wysyła $INQUIRE(j)$ do S_k
- 8) Gdy S_k otrzyma $INQUIRE(j)$ od węzła S_j , to wysyła $RELINQUISH(k)$ do S_j , jeśli S_k otrzymał $FAILED$ od któregośkolwiek węzła ze zbioru R_k , bądź wysłał do któregoś z tych węzłów wiadomość $RELINQUISH$ lecz nie otrzymał od nich $RESPONSE$



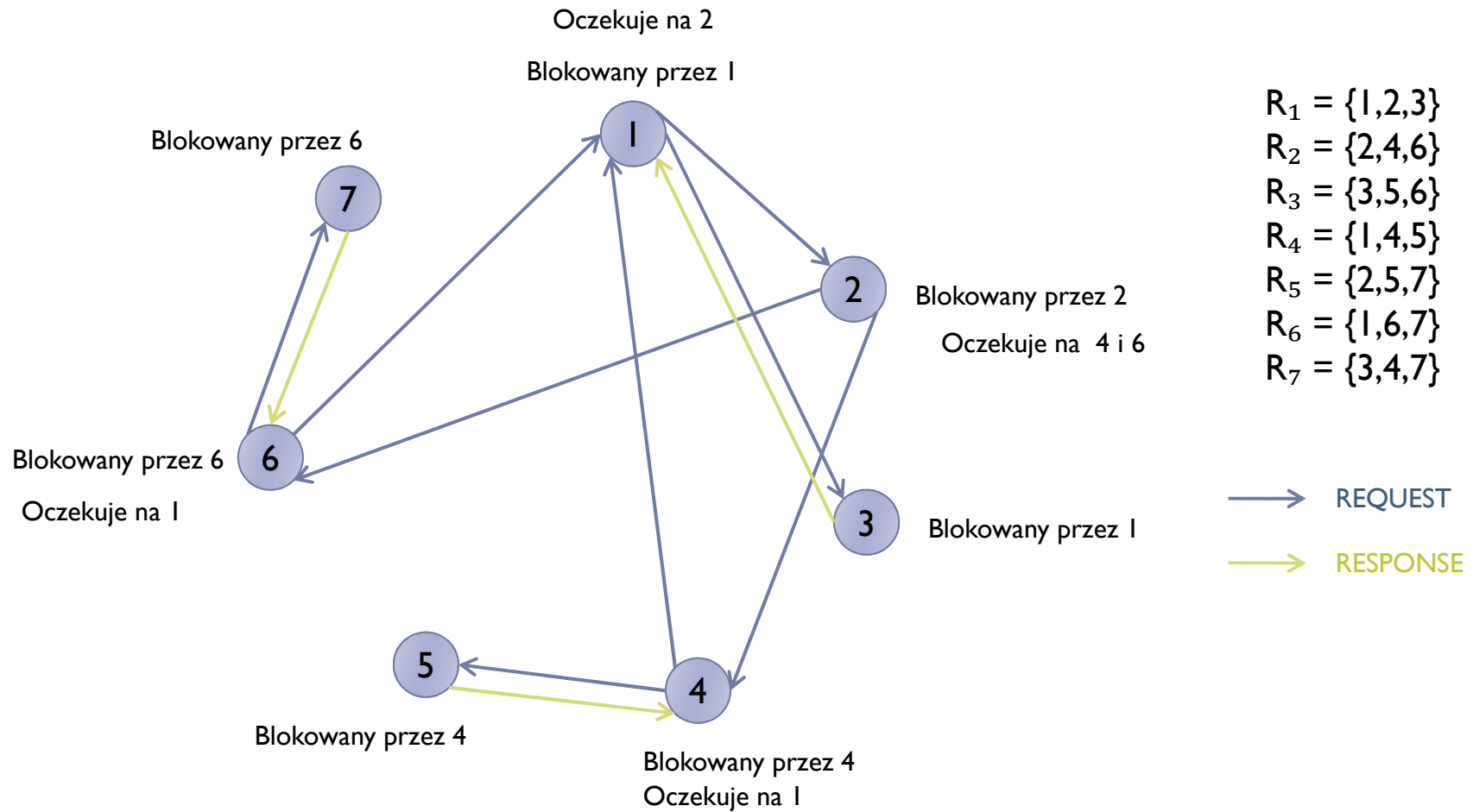
Algorytm Maekawy c.d.

Przeciwdziałanie zakleszczeniu

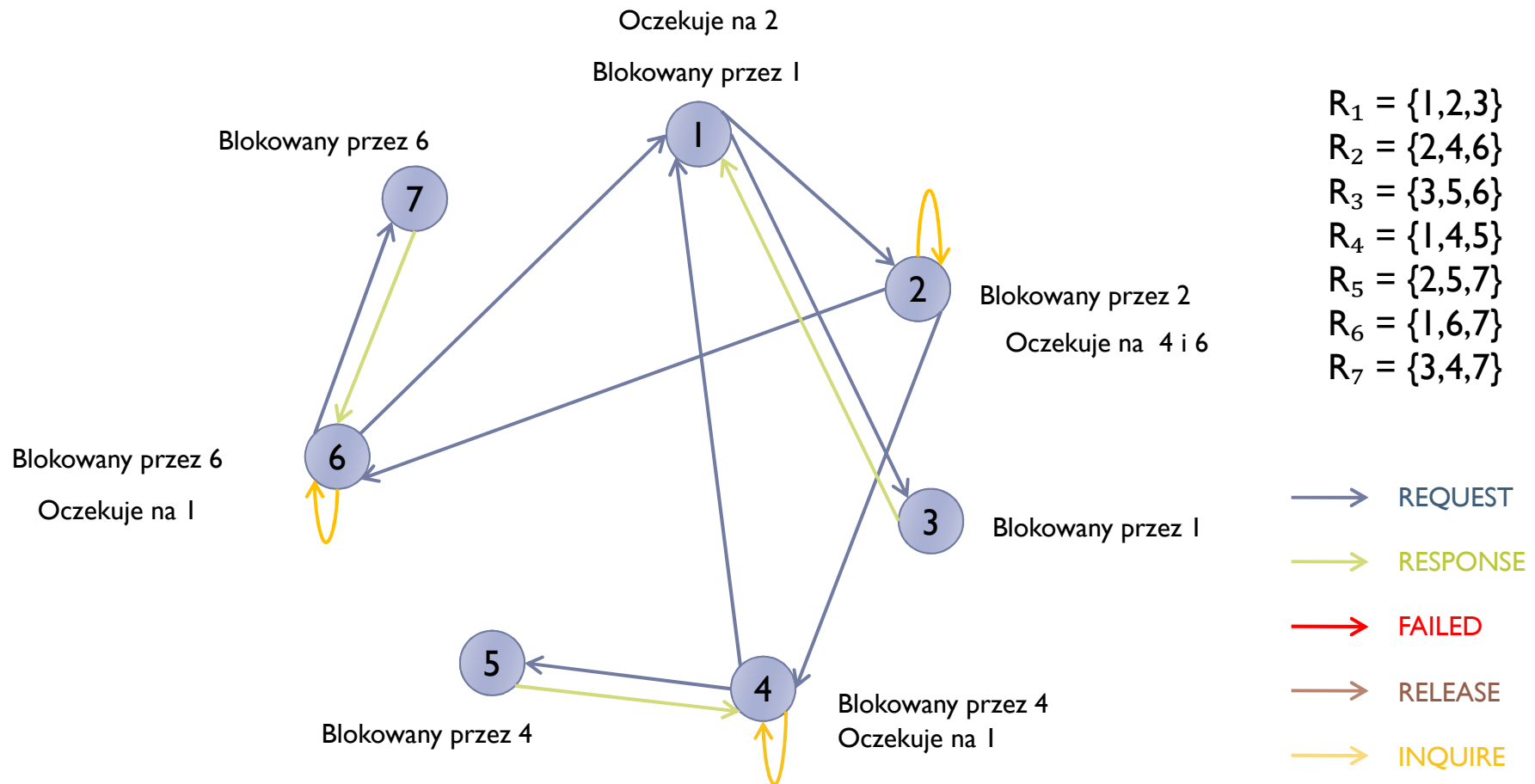
- 7) Gdy węzeł S_j otrzyma $REQUEST(ts,i)$ od węzła S_i , a S_j jest już blokowany na rzecz węzła S_k , to jeśli $REQUEST(ts,i)$ ma mniejszy priorytet, węzeł S_j wysyła $FAILED(j)$ do S_i , a w przeciwnym przypadku S_j wysyła $INQUIRE(j)$ do S_k
- 8) Gdy S_k otrzyma $INQUIRE(j)$ od węzła S_j , to wysyła $RELINQUISH(k)$ do S_j , jeśli S_k otrzymał $FAILED$ od któregośkolwiek węzła ze zbioru R_k , bądź wysłał do któregoś z tych węzłów wiadomość $RELINQUISH$ lecz nie otrzymał od nich $RESPONSE$
- 9) Gdy S_j otrzyma $RELINQUISH(k)$ zwalnia blokowanie na rzecz S_k , umieszcza żądanie od S_k w odpowiednim miejscu w kolejce, i wysyła $RESPONSE(j)$ do węzła, którego żądanie znajduje się na pierwszym miejscu w kolejce.



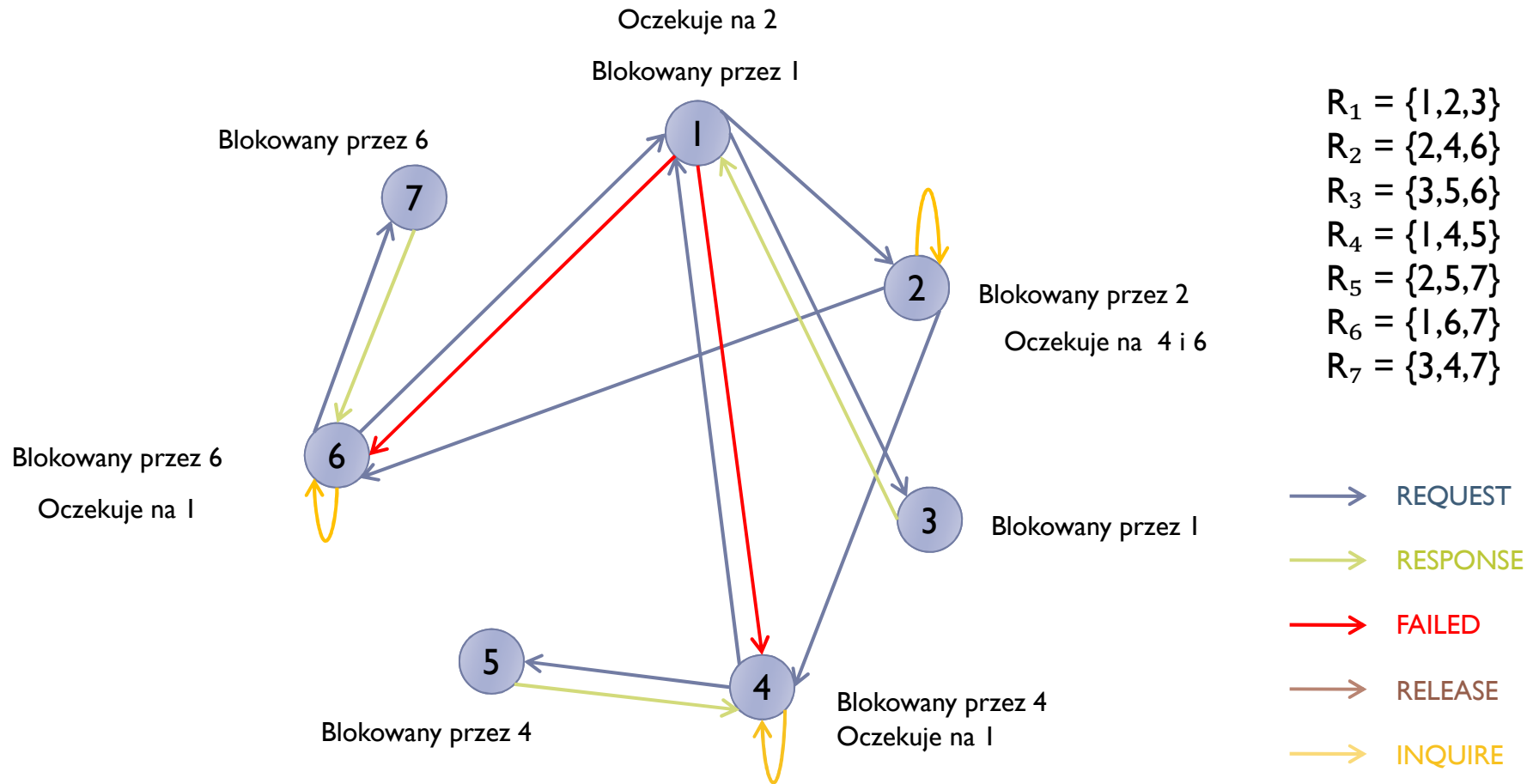
Zakleszczenie - przykład



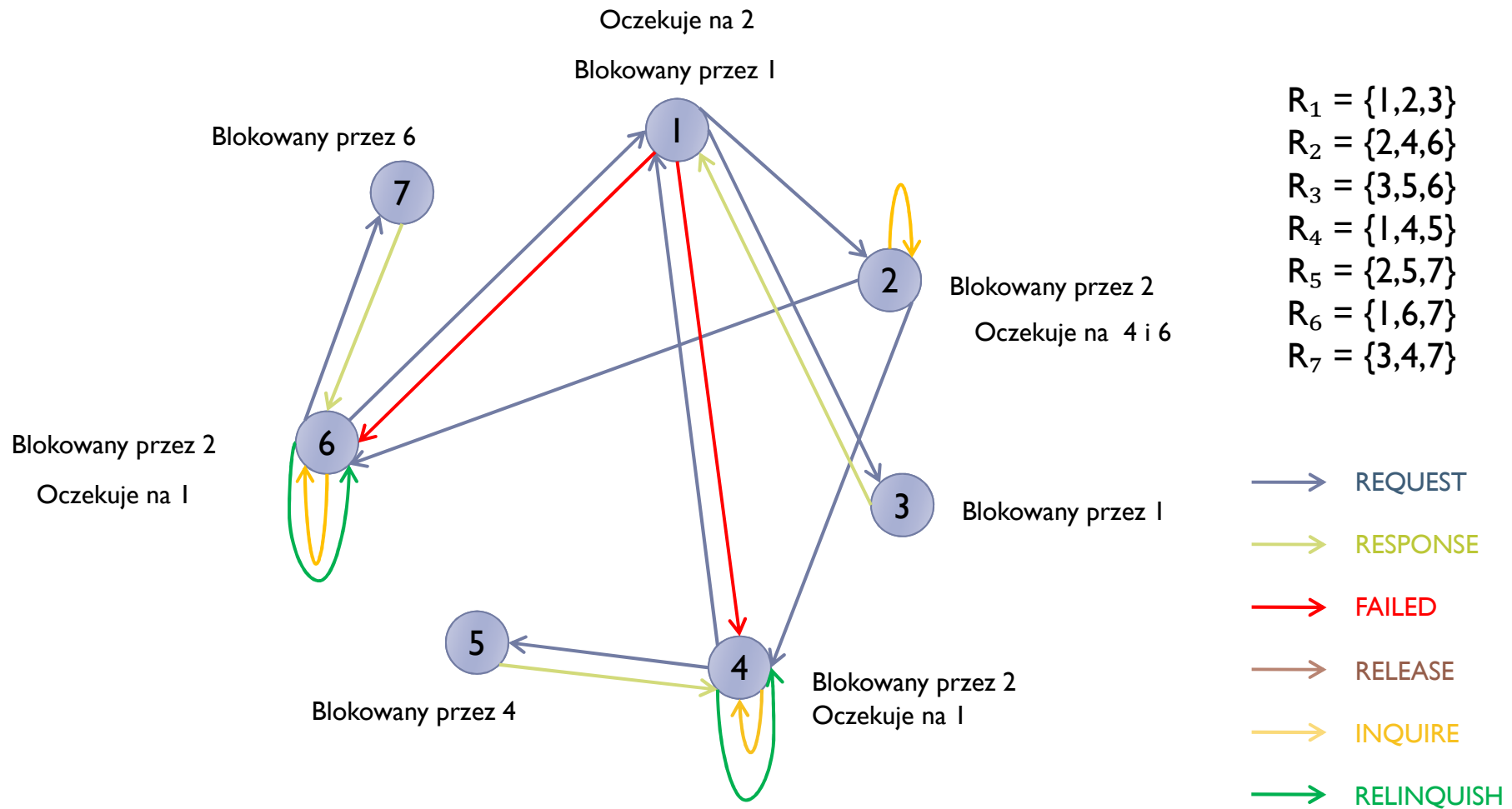
Zakleszczenie - przykład



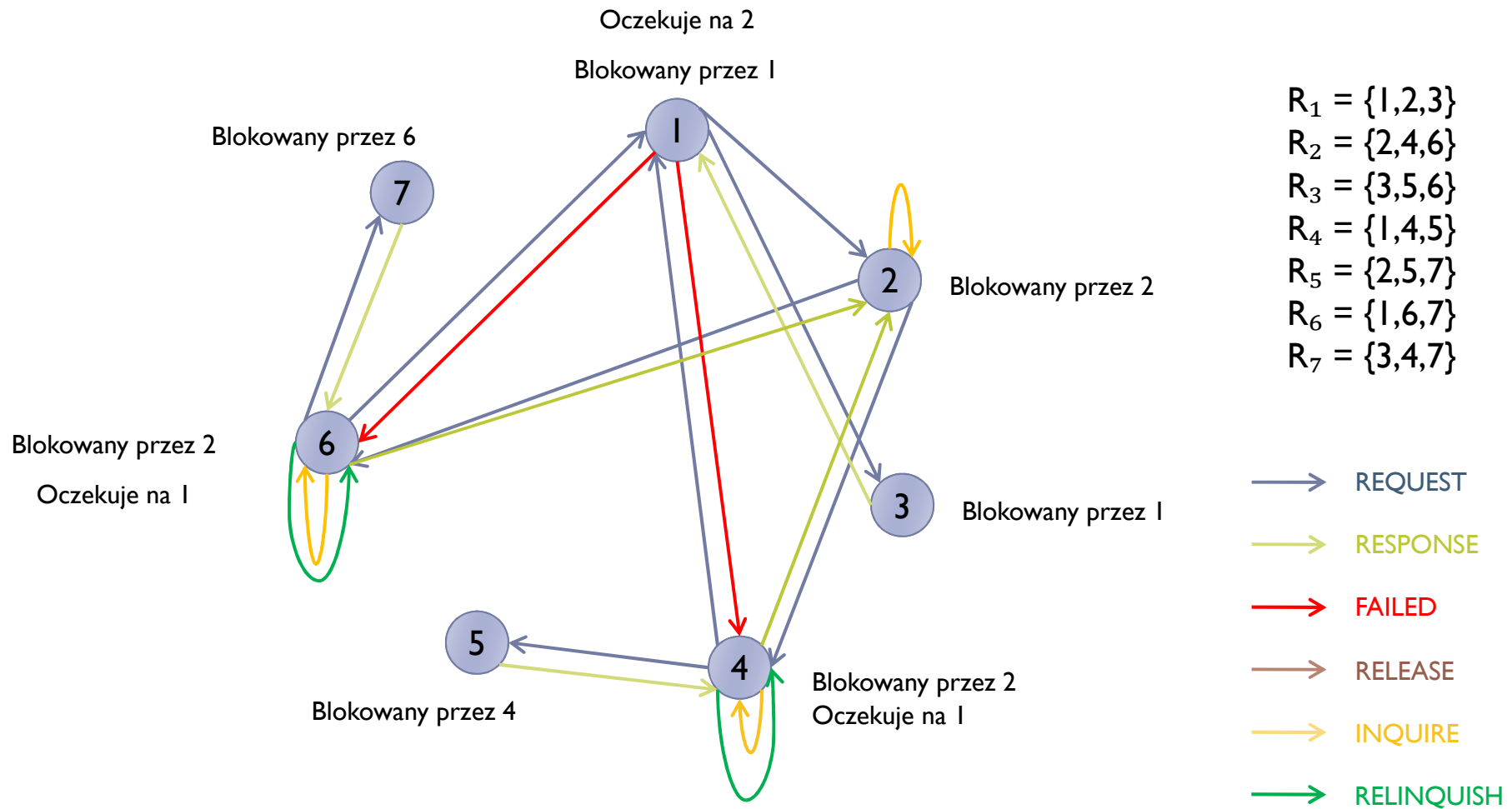
Zakleszczenie - przykład



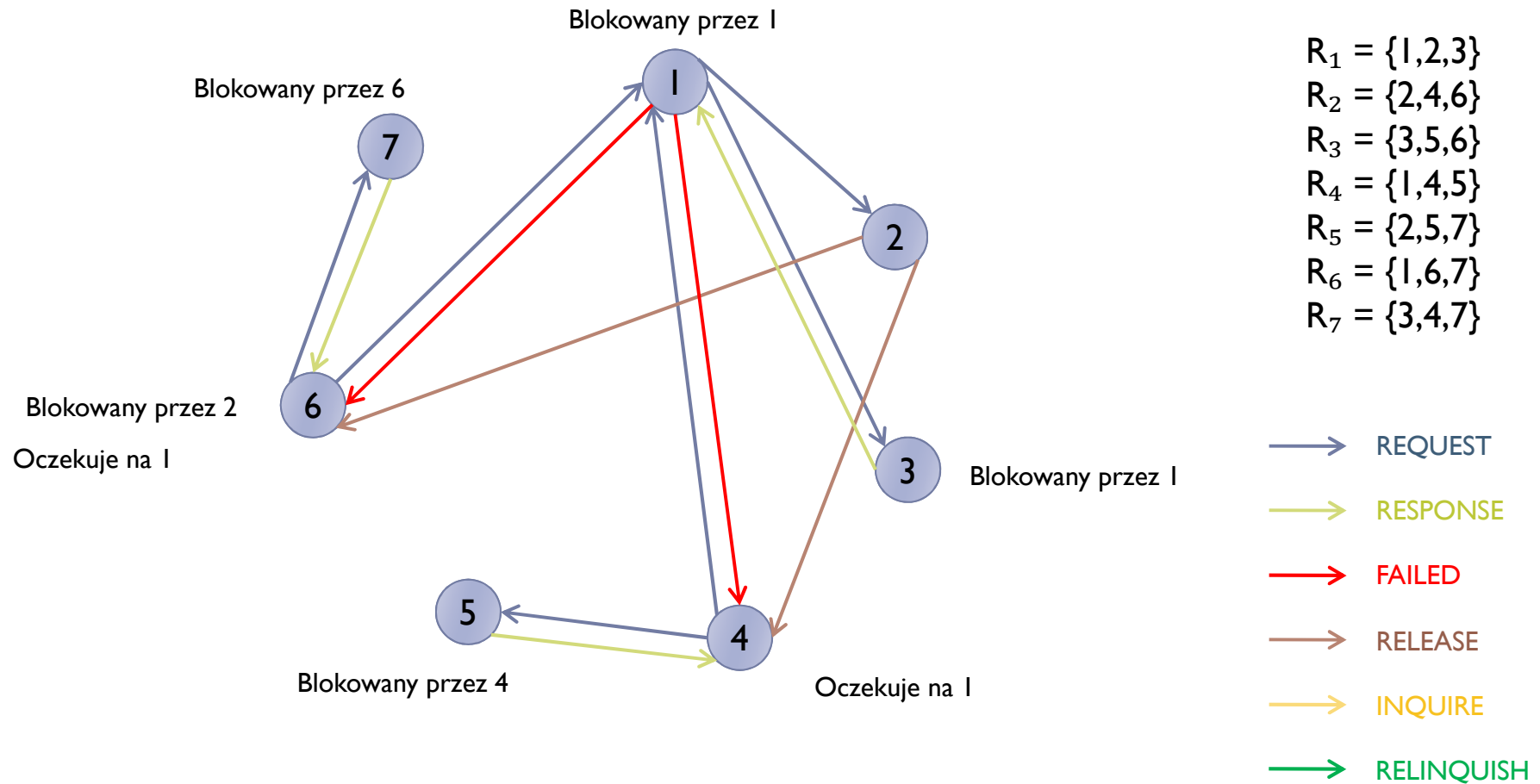
Zakleszczenie - przykład



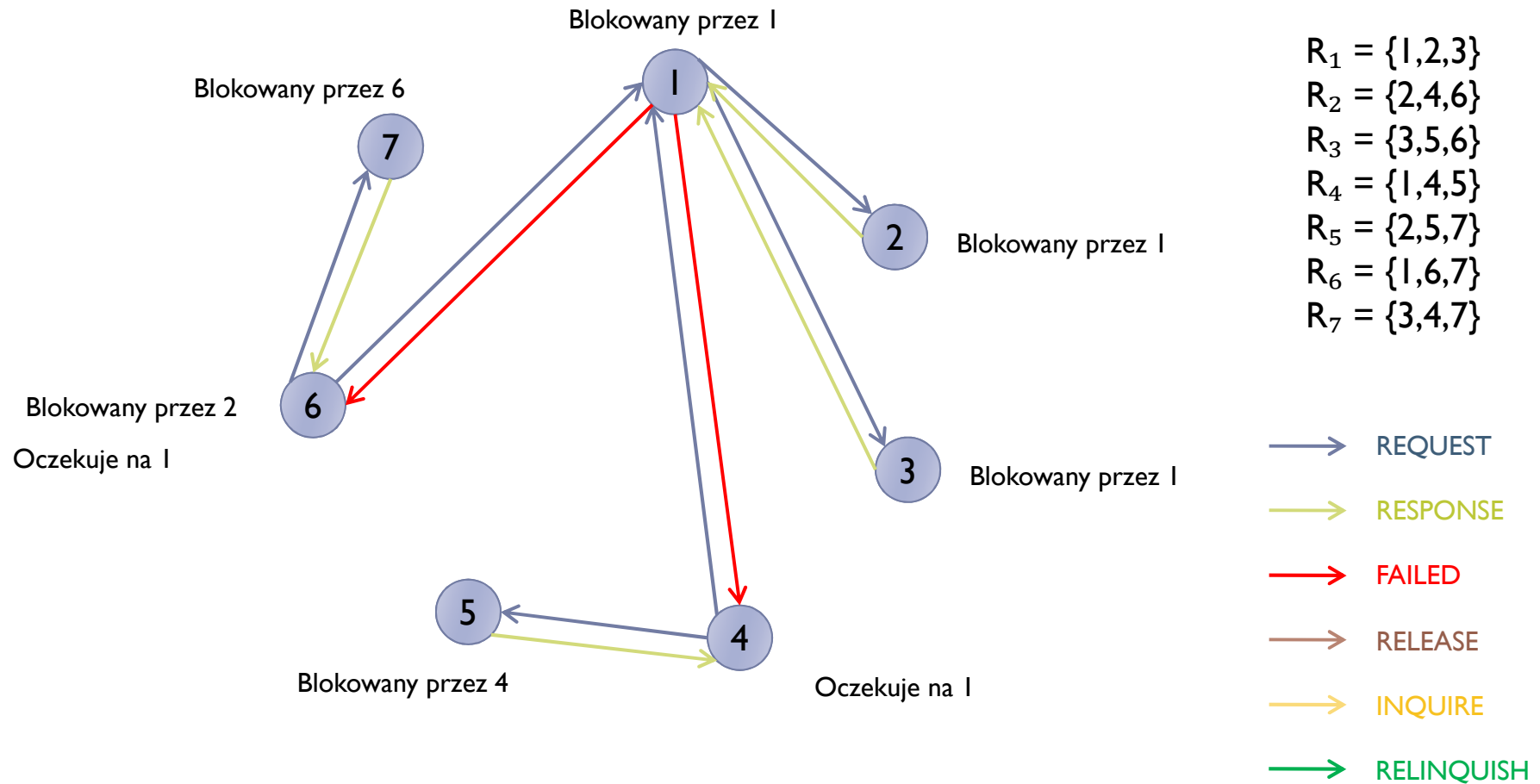
Zakleszczenie - przykład



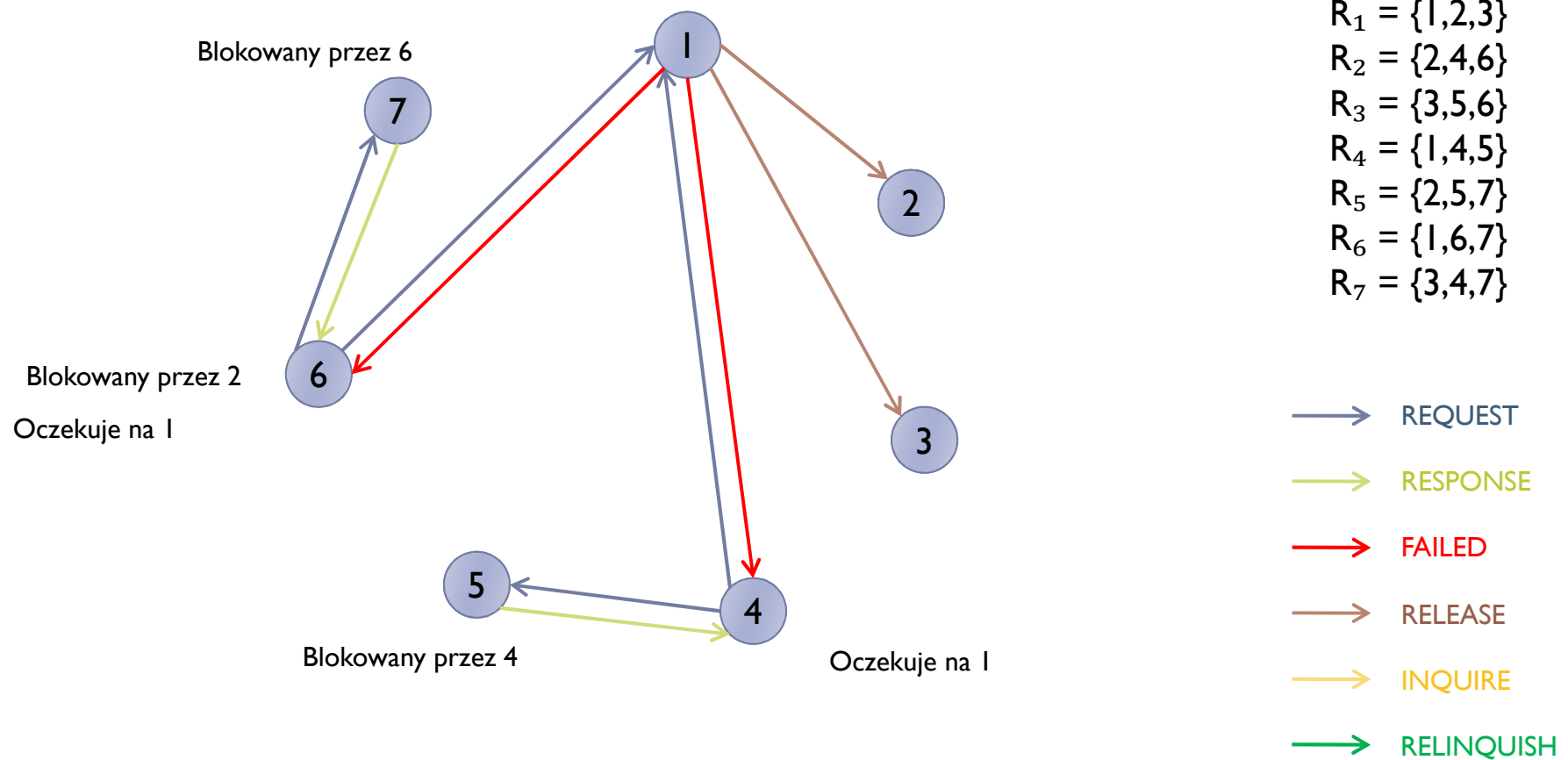
Zakleszczenie - przykład



Zakleszczenie - przykład

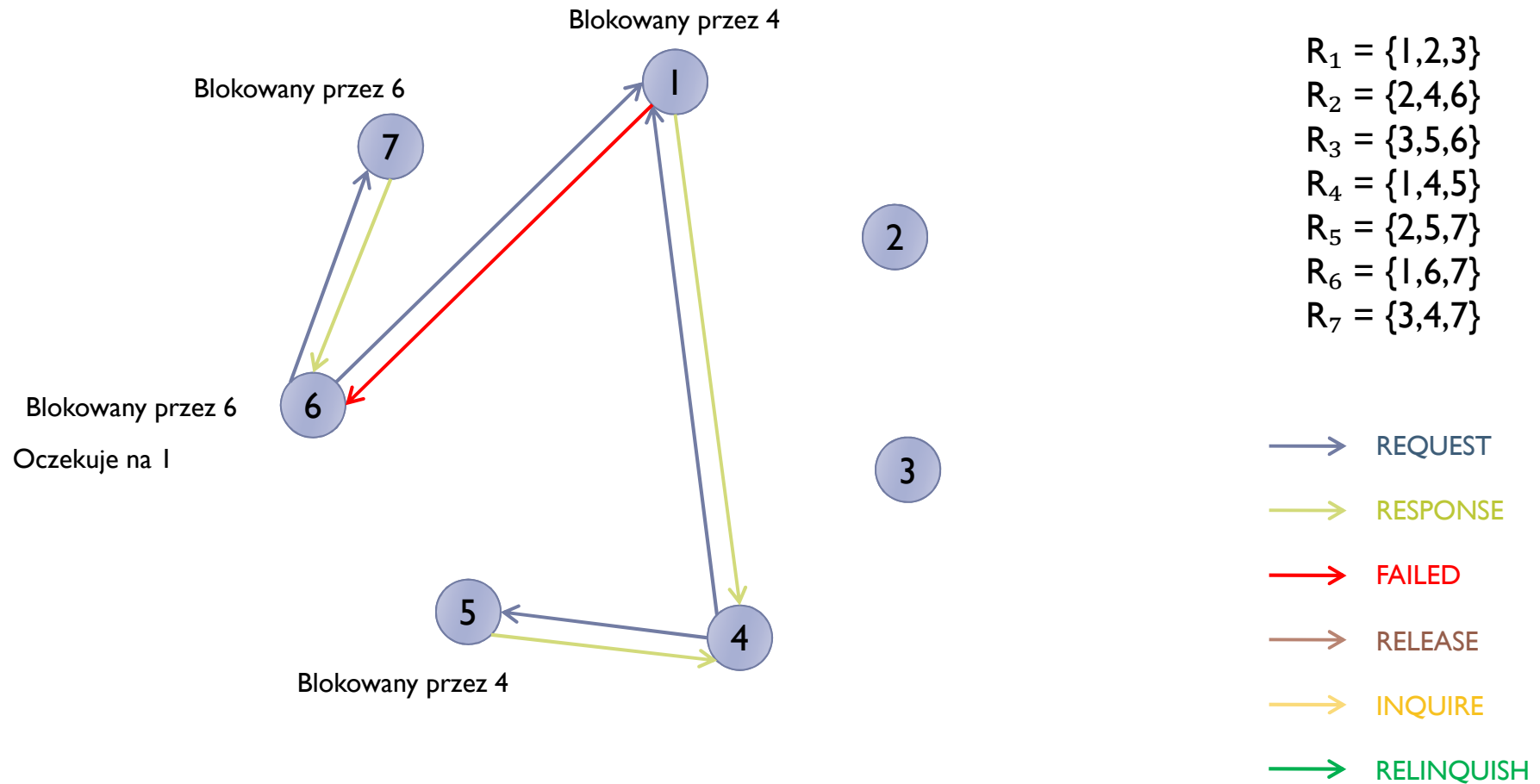


Zakleszczenie - przykład

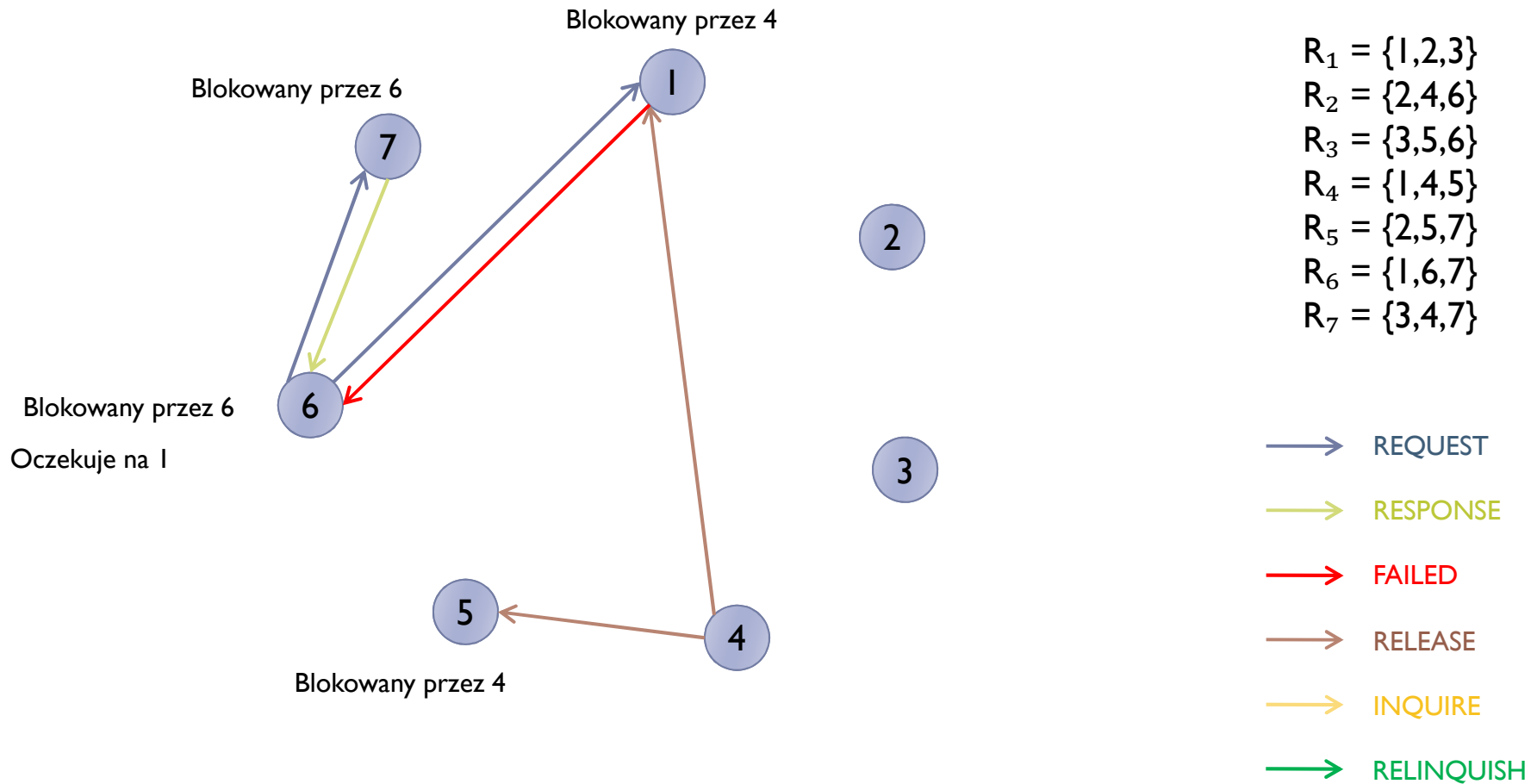


- $R_1 = \{1,2,3\}$
- $R_2 = \{2,4,6\}$
- $R_3 = \{3,5,6\}$
- $R_4 = \{1,4,5\}$
- $R_5 = \{2,5,7\}$
- $R_6 = \{1,6,7\}$
- $R_7 = \{3,4,7\}$

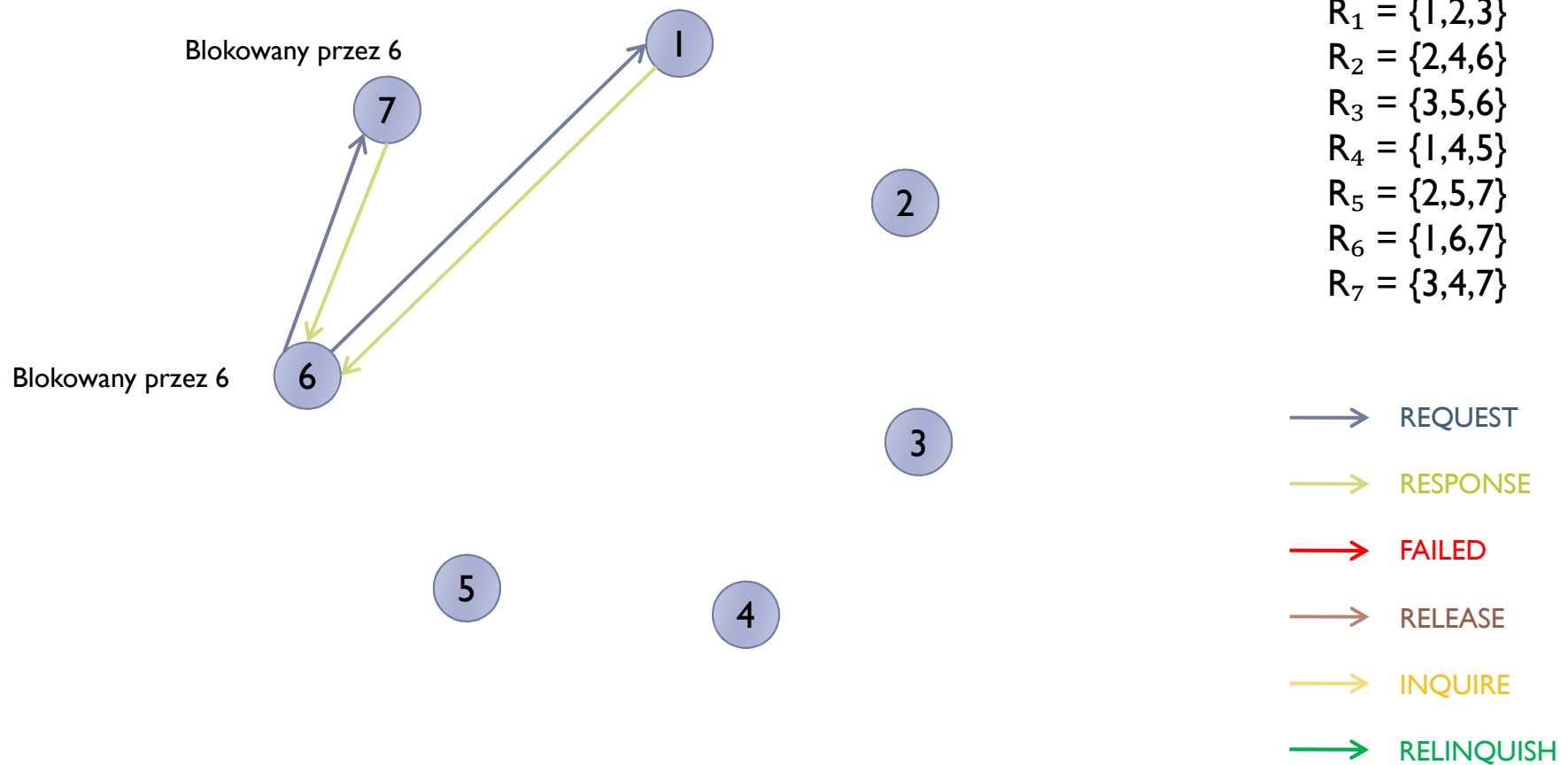
Zakleszczenie - przykład



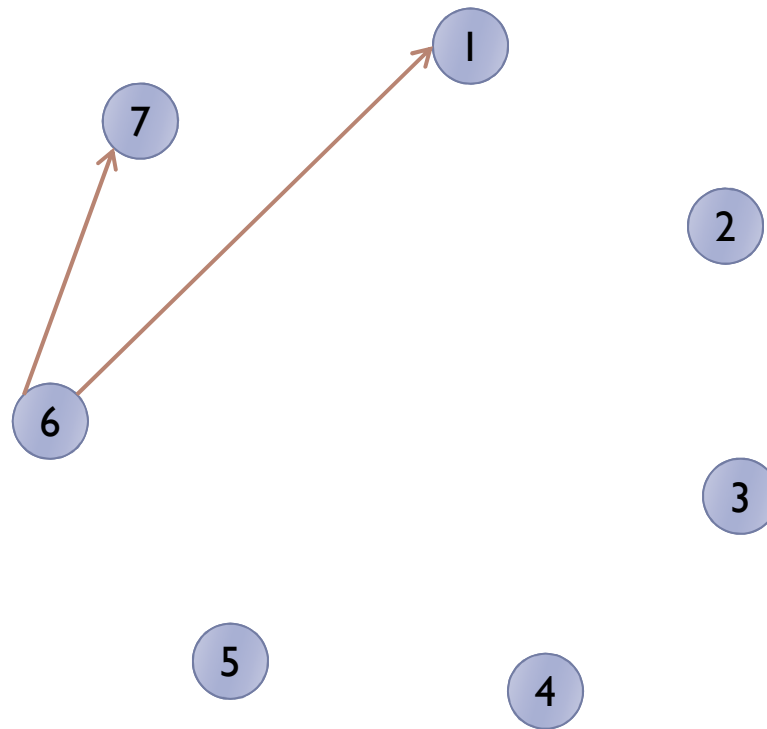
Zakleszczenie - przykład



Zakleszczenie - przykład



Zakleszczenie - przykład



$R_1 = \{1,2,3\}$
 $R_2 = \{2,4,6\}$
 $R_3 = \{3,5,6\}$
 $R_4 = \{1,4,5\}$
 $R_5 = \{2,5,7\}$
 $R_6 = \{1,6,7\}$
 $R_7 = \{3,4,7\}$

→ REQUEST
→ RESPONSE
→ FAILED
→ RELEASE
→ INQUIRE
→ RELINQUISH



Złożoność komunikacyjna

Optymistyczny przypadek:

K-1 REQUEST, K-1 LOCKED, K-1 RELEASE

$3 * \sqrt{N}$



Złożoność

Optymistyczny przypadek:

K-1 REQUEST, K-1 LOCKED, K-1 RELEASE

$3 * \sqrt{N}$

Pesymistyczny przypadek:

K-1 REQUEST, K-1 FAILED, K-1 LOCKED, K-1 RELEASE

Lub

K-1 REQUEST, K-1 INQUIRE, K-1 RELINQUISH, K-1
LOCKED, K-1 RELEASE

