

Deadlock

enrico.bacis@unibg.it

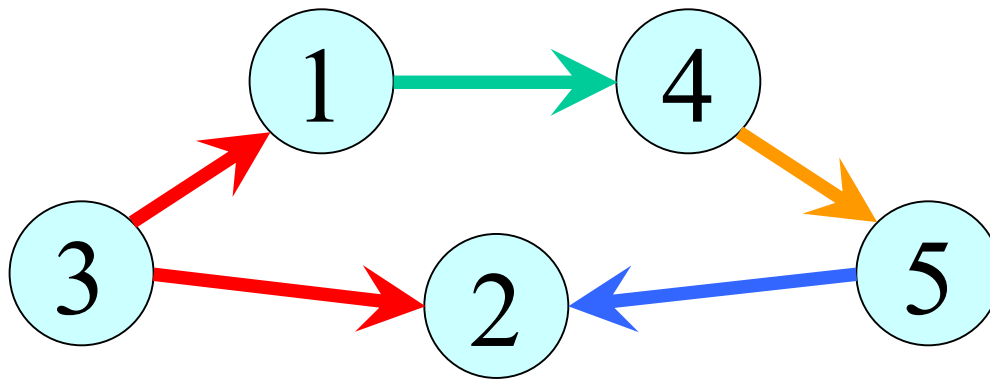
D.1 Si assuma che $L_i^r X$ ($L_i^w X$) rappresenti una richiesta di lock in lettura (scrittura) da parte della transazione i sulla risorsa X . Sia data la sequenza di richieste di lock:

$L_1^r A$ $L_2^r A$ $L_3^w B$ $L_4^r C$ $L_2^w D$ $L_3^w A$ $L_1^w C$
 $L_5^w E$ $L_5^r D$ $L_4^r E$

Si verifichi **se** la sequenza di richieste di lock porta ad una situazione di blocco critico.

Costruiamo un grafo in cui l'arco i,j indica che t_i aspetta un rilascio da t_j

$L_1^r A$ $L_2^r A$ $L_3^w B$ $L_4^r C$ $L_2^w D$
 $L_3^w A$ $L_1^w C$ $L_5^w E$ $L_5^r D$ $L_4^r E$



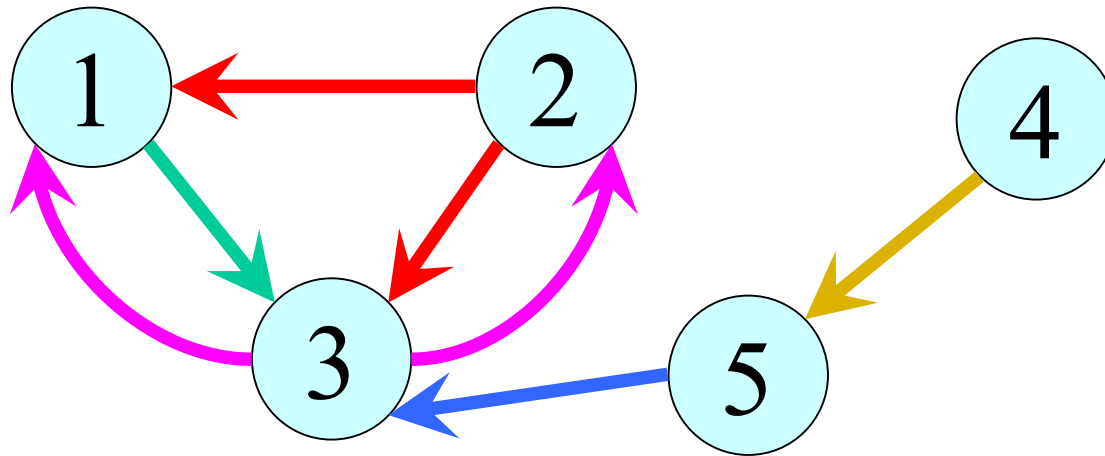
L'aciclicità del grafo garantisce l'assenza di deadlock. t_2 può terminare per prima, poi nell'ordine t_5, t_4, t_1, t_3

D.2 Si verifichi **che** la sequenza di richieste di lock porta ad una situazione di blocco critico.

$L^r_1^A$	$L^r_2^A$	$L^w_3^B$	$L^w_3^A$	$L^r_4^C$	$L^w_2^A$	$L^r_1^B$
$L^w_5^E$	$L^r_5^B$	$L^r_4^E$				

Si individui inoltre quale transazione può essere fatta terminare per rimuovere la situazione di blocco.

$L_1^r A$ $L_2^r A$ $L_3^w B$ $L_3^w A$ $L_4^r C$ $L_2^w A$ $L_1^r B$
 $L_5^w E$ $L_5^r B$ $L_4^r E$



Il grafo è ciclico. C'è un deadlock tra le transazioni t_1 , t_2 e t_3 , che si può risolvere uccidendo t_3 (non t_2 [t_1], che rimarrebbe il ciclo tra t_1 e t_3 [tra t_2 e t_3])

E.1 Siano date le seguenti condizioni di attesa sui nodi di un DBMS distribuito:

Nodo 1: $E_4 \rightarrow t_1$, $t_1 \rightarrow t_2$, $t_2 \rightarrow E_2$

Nodo 2: $E_1 \rightarrow t_2$, $t_2 \rightarrow t_4$, $t_4 \rightarrow E_3$

Nodo 3: $E_2 \rightarrow t_4$, $t_4 \rightarrow t_3$, $t_3 \rightarrow E_4$

Nodo 4: $E_3 \rightarrow t_3$, $t_3 \rightarrow t_1$, $t_1 \rightarrow E_1$

Determinare se si è in presenza di una situazione di blocco critico.

$$t_i \rightarrow t_j$$

indica un'attesa locale

(t_i attende il rilascio di una *risorsa* acquisita da t_j)

$$t_i \rightarrow E_n$$

indica che t_i attende la *terminazione* dell'esecuzione di una sottotransazione t_j sul *nodo* n
(invocazione *sincrona*)

$$E_m \rightarrow t_i$$

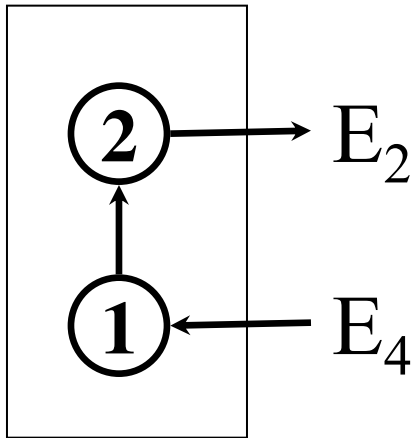
indica che t_i è stata invocata in modo sincrono da una t_j residente sul nodo m

Ogni condizione di attesa in cui una sotto-transazione t_i , attivata in remoto da un nodo m , **attende** (*anche transitivamente* a causa della situazione dei lock) un'altra transazione t_j , che a sua volta attende una sotto-transazione remota su un nodo n , è espressa da:

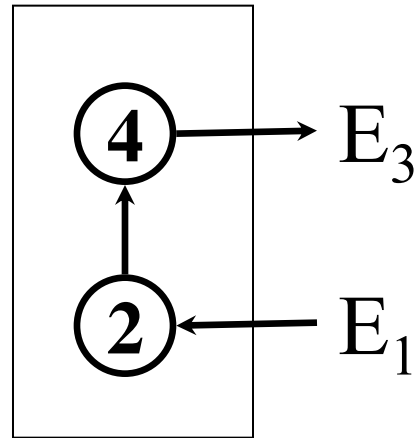
$$E_m \rightarrow t_i \rightarrow t_j \rightarrow E_n$$

Tale è anche la forma del messaggio che si scambiano i nodi del sistema

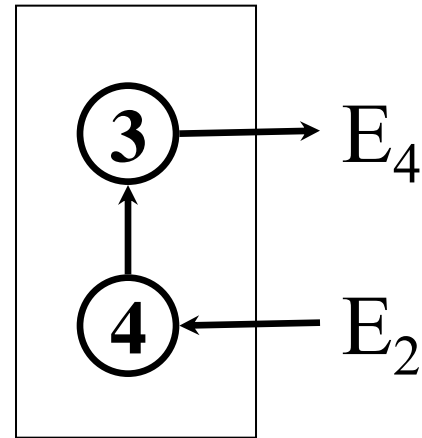
NODO 1



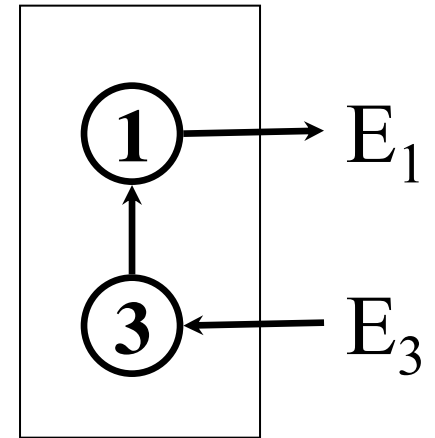
NODO 2



NODO 3



NODO 4



Nodo 1: $E_4 \rightarrow t_1, t_1 \rightarrow t_2, t_2 \rightarrow E_2$

Nodo 2: $E_1 \rightarrow t_2, t_2 \rightarrow t_4, t_4 \rightarrow E_3$

Nodo 3: $E_2 \rightarrow t_4, t_4 \rightarrow t_3, t_3 \rightarrow E_4$

Nodo 4: $E_3 \rightarrow t_3, t_3 \rightarrow t_1, t_1 \rightarrow E_1$

L'algoritmo è **distribuito**.

Ogni istanza (in esecuzione su un nodo) comunica ad altre istanze dello stesso algoritmo le sequenze di attesa:

$$E_m \rightarrow t_i \rightarrow t_j \rightarrow E_n$$

I messaggi sono inviati solo “in avanti”, cioè verso i nodi dove è attiva la sotto-transazione *attesa* da t_i , e viene inviato solo se $i > j$ (*puramente convenzionale*)

Per rispondere *simuliamo l'esecuzione* - **asincrona** e **distribuita** - dell'algoritmo di rilevazione dei deadlock. Ogni nodo decide di inviare le condizioni di attesa (da esterno su esterno) che rileva, in base alla convenzione per cui la generica condizione

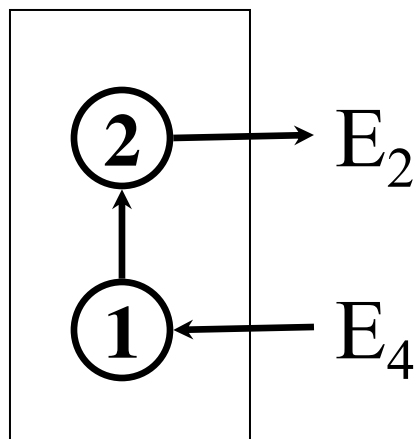
$$E_m \rightarrow t_i \rightarrow t_m \rightarrow t_n \rightarrow t_p \rightarrow t_j \rightarrow E_n$$

si traduce nel messaggio

$$E_m \rightarrow t_i \rightarrow t_j \rightarrow E_n$$

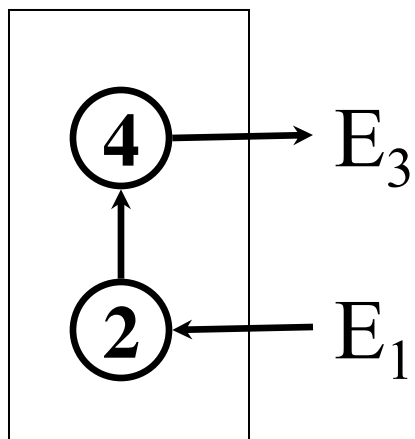
da inviare al nodo **n**, e soltanto se **i > j**

NODO 1



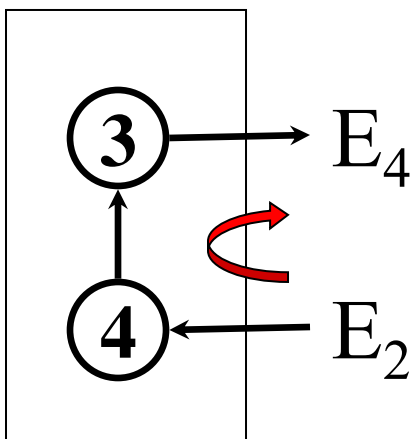
Nulla da
inviare

NODO 2



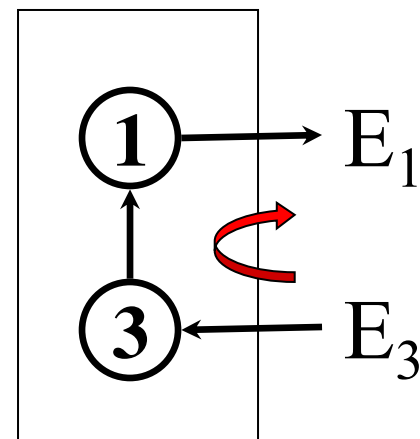
Nulla da
inviare

NODO 3



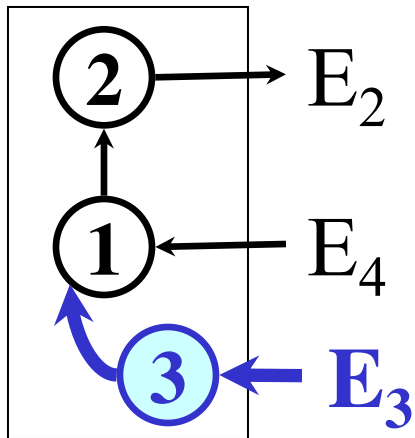
$E_2 \rightarrow t_4 \rightarrow t_3 \rightarrow E_4$
da inviare al
nodo 4

NODO 4

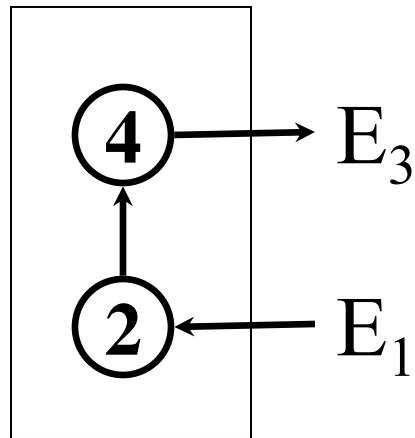


$E_3 \rightarrow t_3 \rightarrow t_1 \rightarrow E_1$
da inviare al
nodo 1

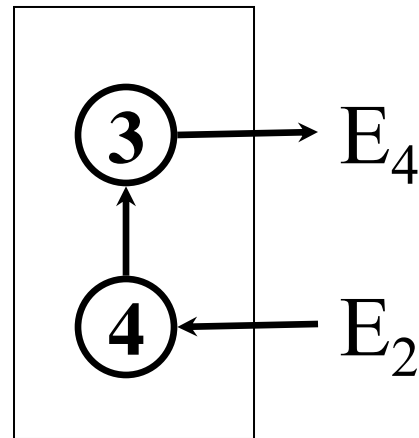
NODO 1



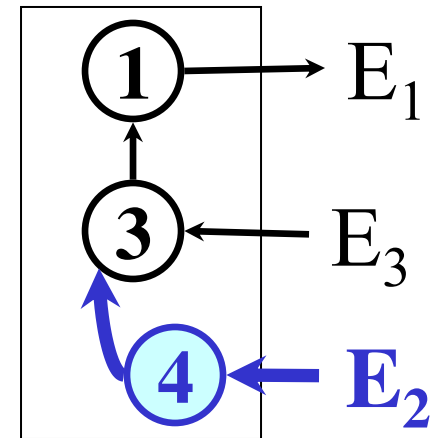
NODO 2



NODO 3



NODO 4



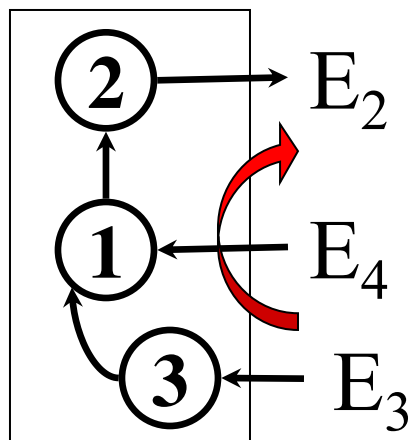
RICEVUTO

$E_3 \rightarrow t_3 \rightarrow t_1 \rightarrow E_1$

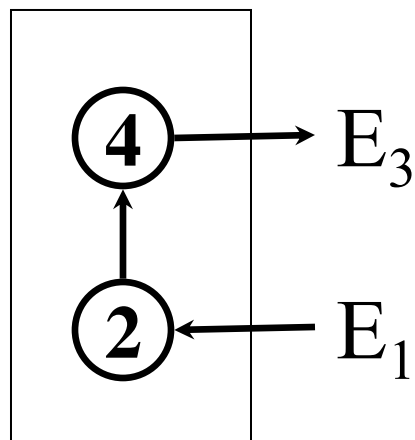
RICEVUTO

$E_2 \rightarrow t_4 \rightarrow t_3 \rightarrow E_4$

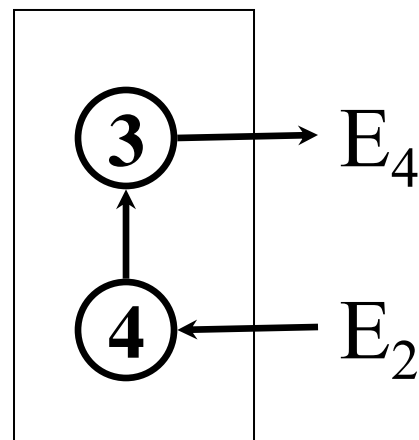
NODO 1



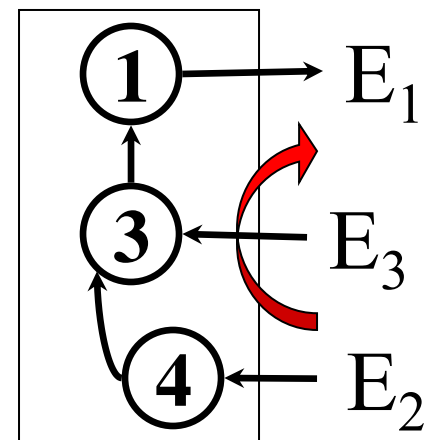
NODO 2



NODO 3



NODO 4



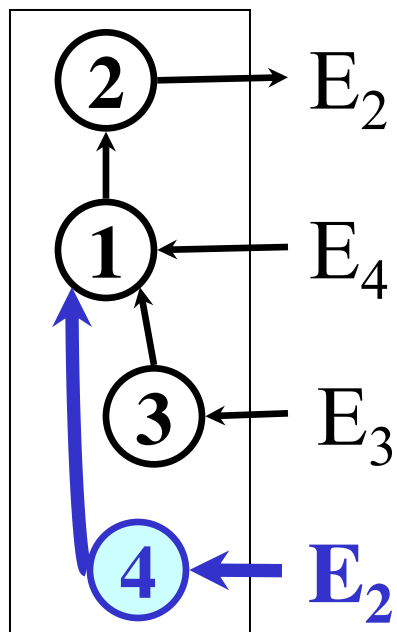
$E_3 \rightarrow t_3 \rightarrow t_2 \rightarrow E_2$
da inviare al
nodo 2

Nulla da
inviare

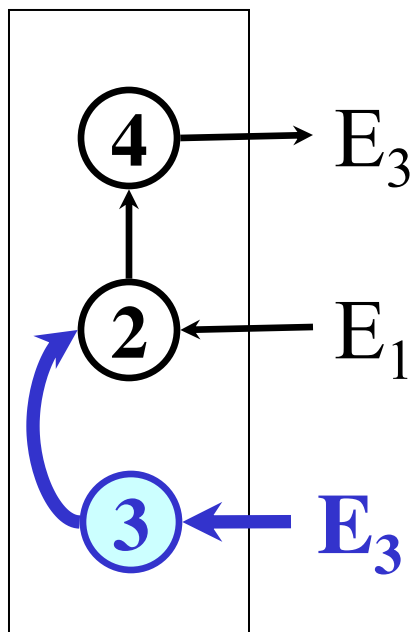
Nulla da
inviare

$E_2 \rightarrow t_4 \rightarrow t_1 \rightarrow E_1$
da inviare al
nodo 1

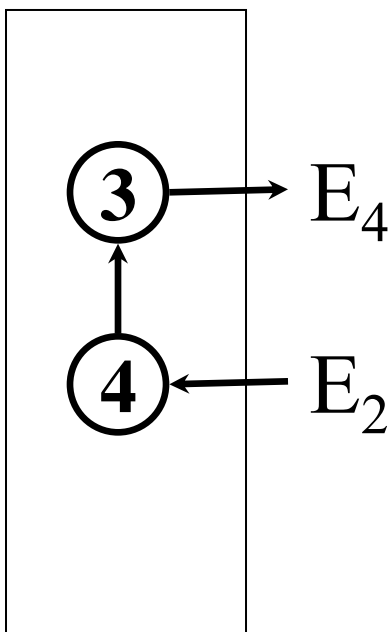
NODO 1



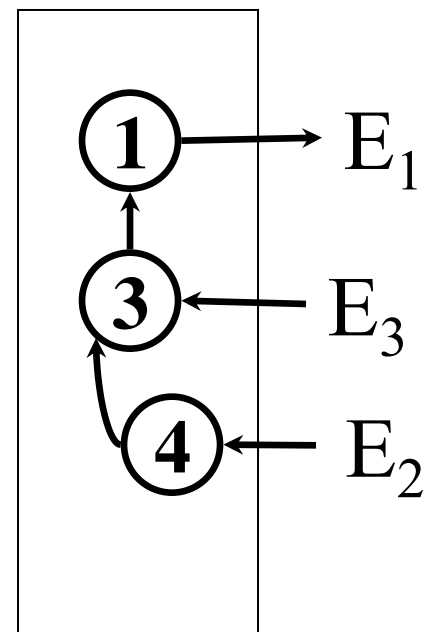
NODO 2



NODO 3



NODO 4



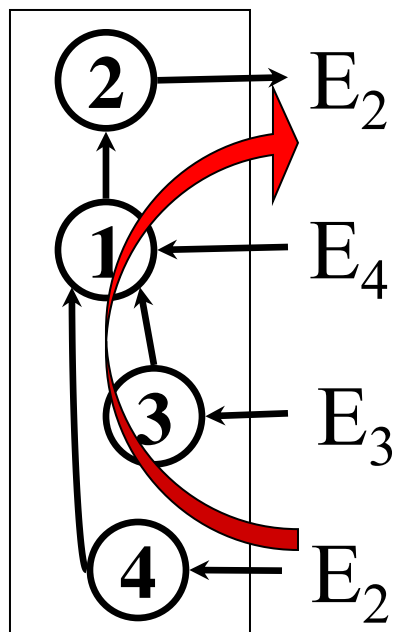
RICEVUTO

$E_2 \rightarrow t_4 \rightarrow t_1 \rightarrow E_1$

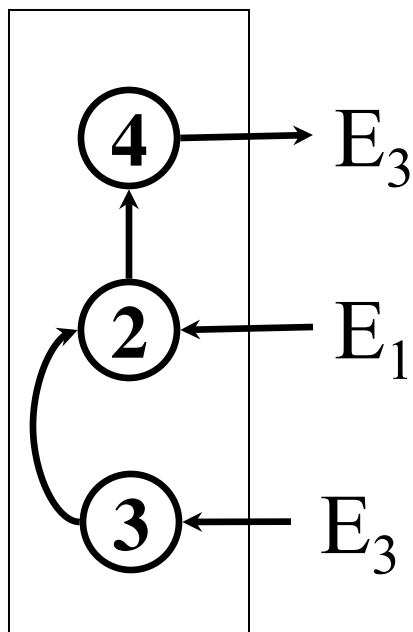
RICEVUTO

$E_3 \rightarrow t_3 \rightarrow t_2 \rightarrow E_2$

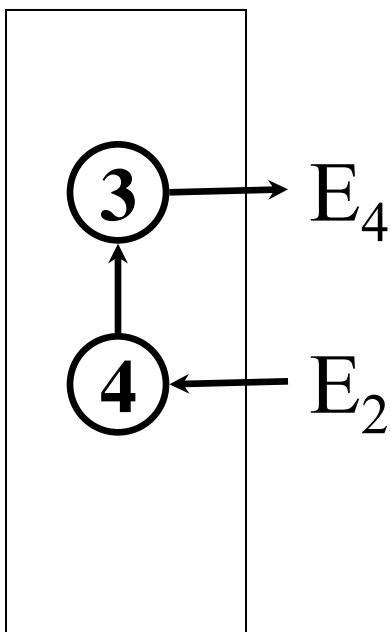
NODO 1



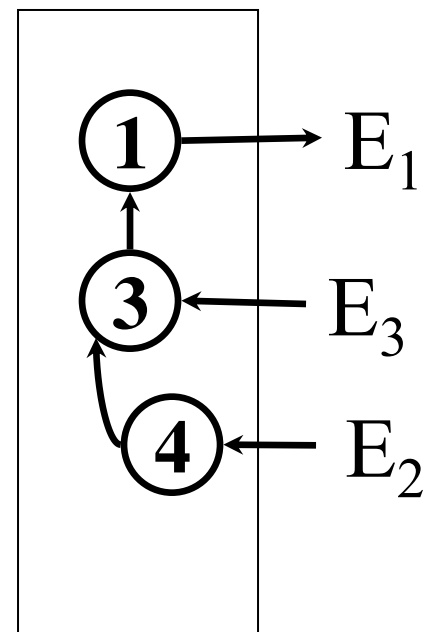
NODO 2



NODO 3



NODO 4



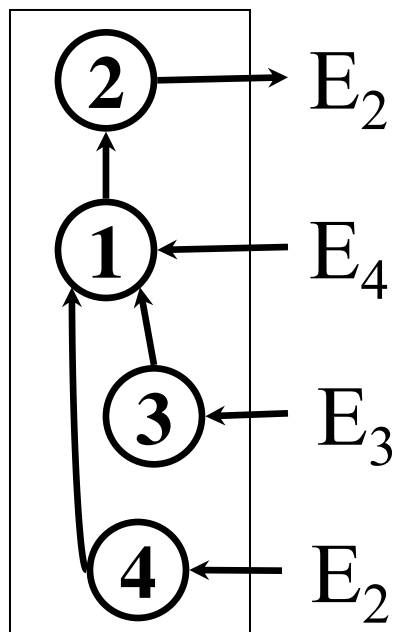
$E_2 \rightarrow t_4 \rightarrow t_2 \rightarrow E_2$
da inviare al
nodo 2

Nulla da
inviare

Nulla da
inviare

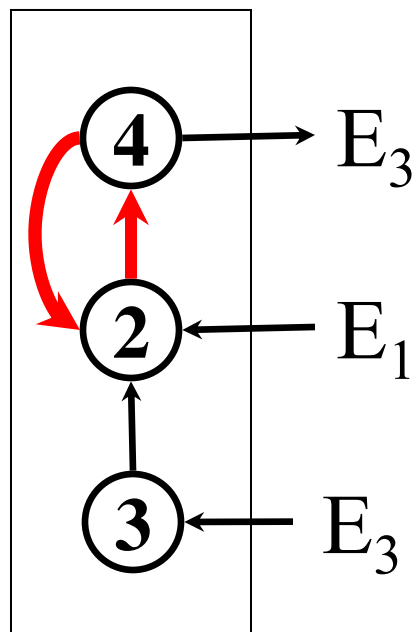
Nulla da
inviare

NODO 1



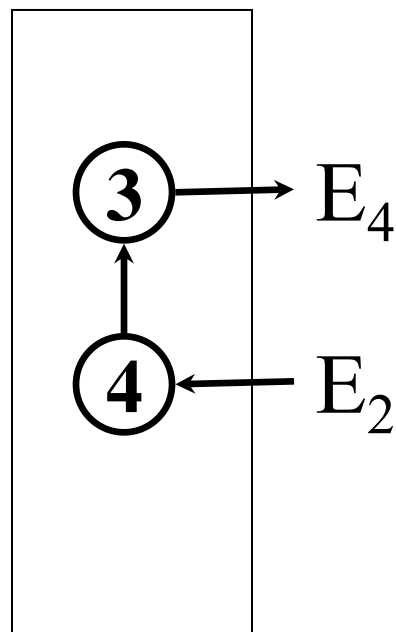
Nulla da
inviare

NODO 2



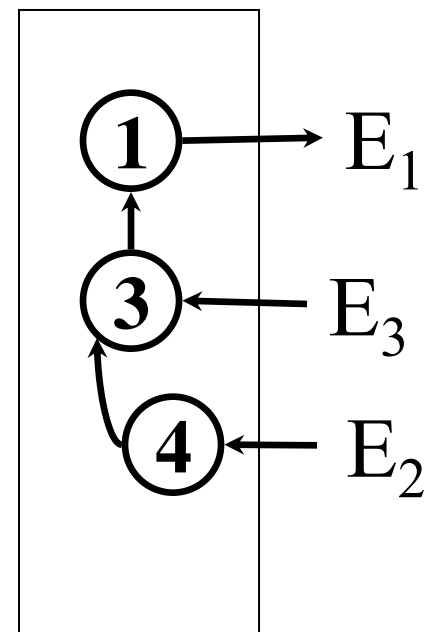
**SCOPRE
UN CICLO**

NODO 3



Nulla da
inviare

NODO 4



Nulla da
inviare

Esiste un blocco critico tra le transazioni t_2 e t_4

E.2 Dire se le seguenti condizioni di attesa determinano una situazione di blocco critico:

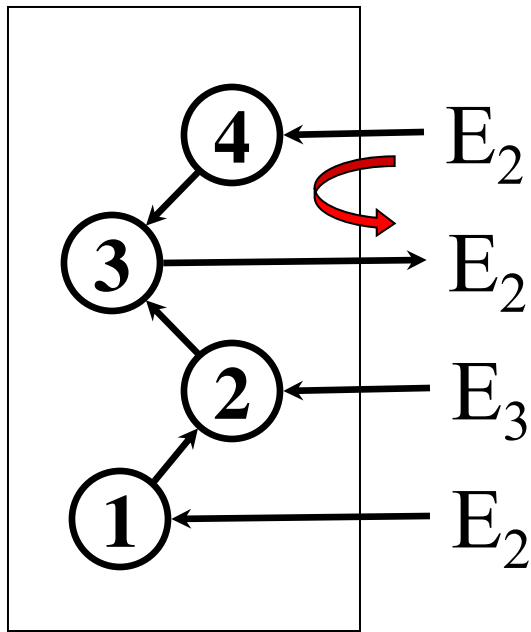
Nodo 1: $E_2 \rightarrow t_1$, $t_1 \rightarrow t_2$, $E_3 \rightarrow t_2$, $t_2 \rightarrow t_3$,
 $t_3 \rightarrow E_2$, $E_2 \rightarrow t_4$, $t_4 \rightarrow t_3$

Nodo 2: $E_1 \rightarrow t_3$, $t_3 \rightarrow t_5$, $t_5 \rightarrow t_6$, $t_6 \rightarrow E_3$, $E_3 \rightarrow t_7$,
 $t_7 \rightarrow t_6$, $t_9 \rightarrow t_4$, $t_4 \rightarrow E_1$, $t_1 \rightarrow E_1$

Nodo 3: $E_2 \rightarrow t_6$, $t_6 \rightarrow t_8$, $t_8 \rightarrow t_2$, $t_2 \rightarrow E_1$, $t_7 \rightarrow E_2$

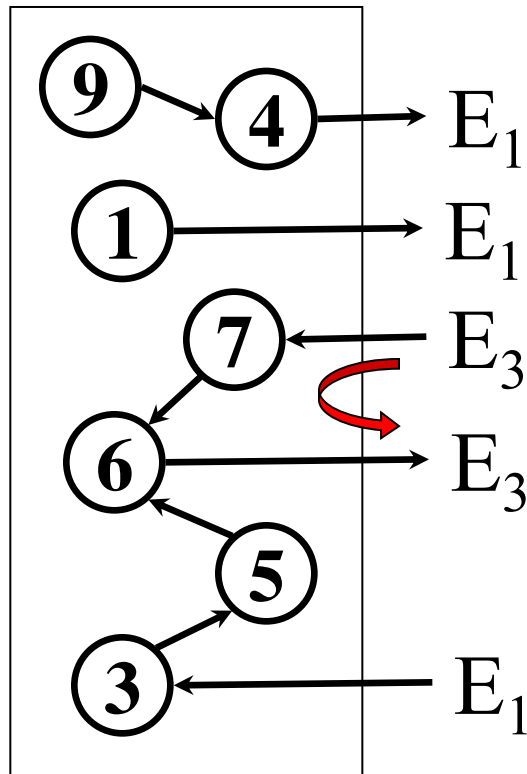
Dove $t_m \rightarrow t_n$ indica un'attesa locale (t_m attende il rilascio di una *risorsa* acquisita da t_n), $t_m \rightarrow E_n$ indica che t_m attende l'esecuzione di una sottotransazione sul *nodo* n (invocazione *sincrona*) e $E_m \rightarrow t_n$ indica che t_n è stata invocata in modo sincrono da una t_i sul nodo m

NODO 1



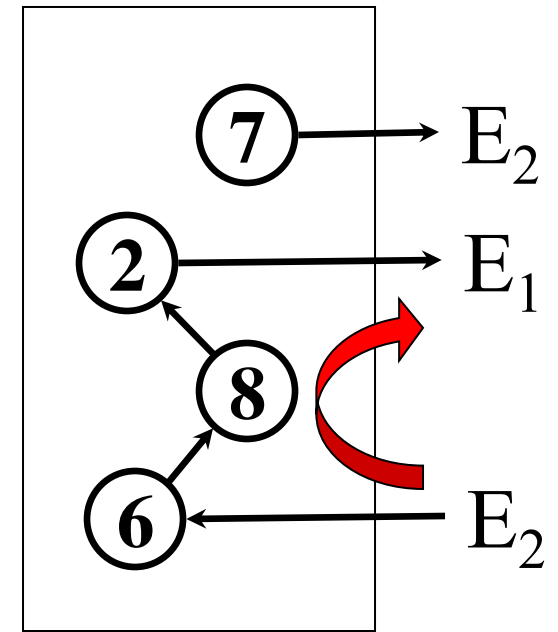
$E_2 \rightarrow t_4 \rightarrow t_3 \rightarrow E_2$
da inviare al
nodo 2

NODO 2



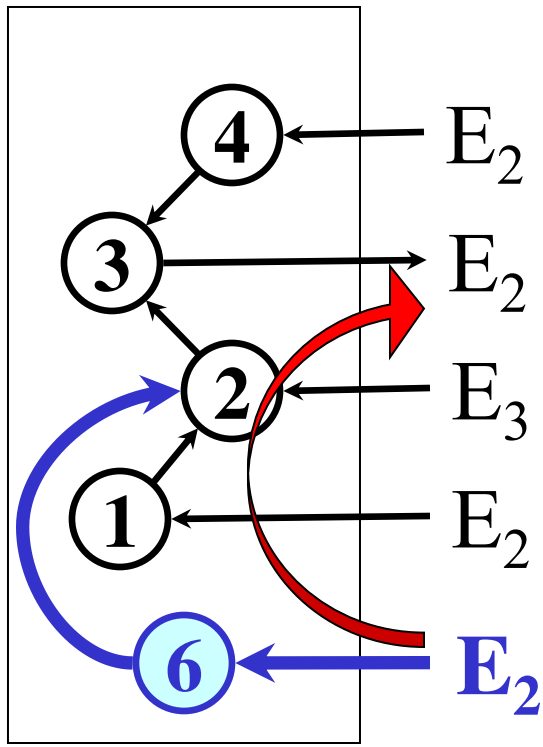
$E_3 \rightarrow t_7 \rightarrow t_6 \rightarrow E_3$
da inviare al
nodo 3

NODO 3



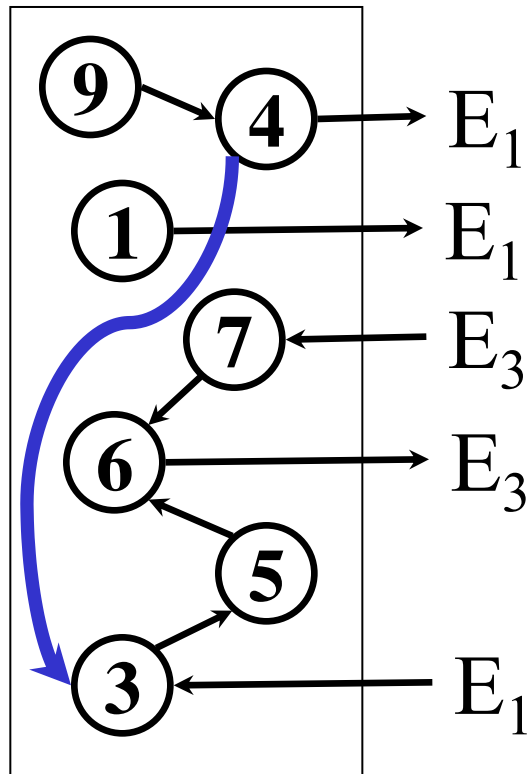
$E_2 \rightarrow t_6 \rightarrow t_2 \rightarrow E_1$
da inviare al
nodo 1

NODO 1



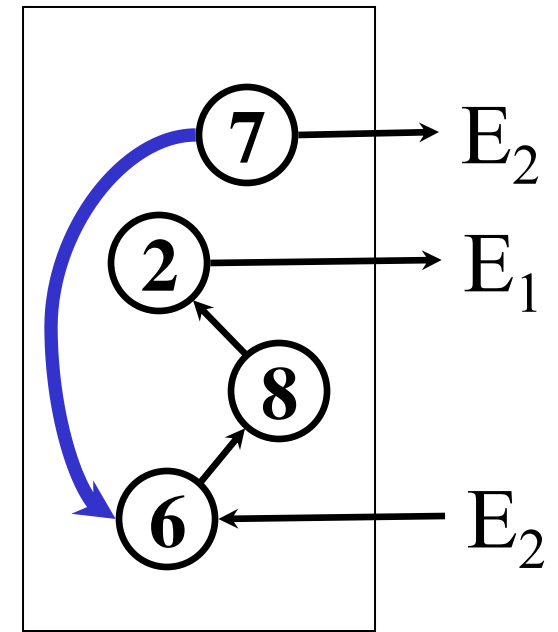
$E_2 \rightarrow t_6 \rightarrow t_3 \rightarrow E_2$
da inviare al
nodo 2

NODO 2



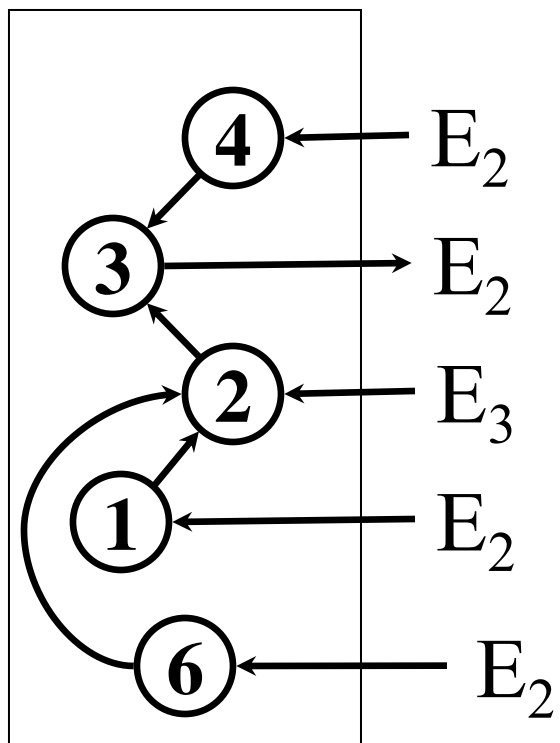
Il messaggio
arrivato non
causa nuovi
messaggi

NODO 3

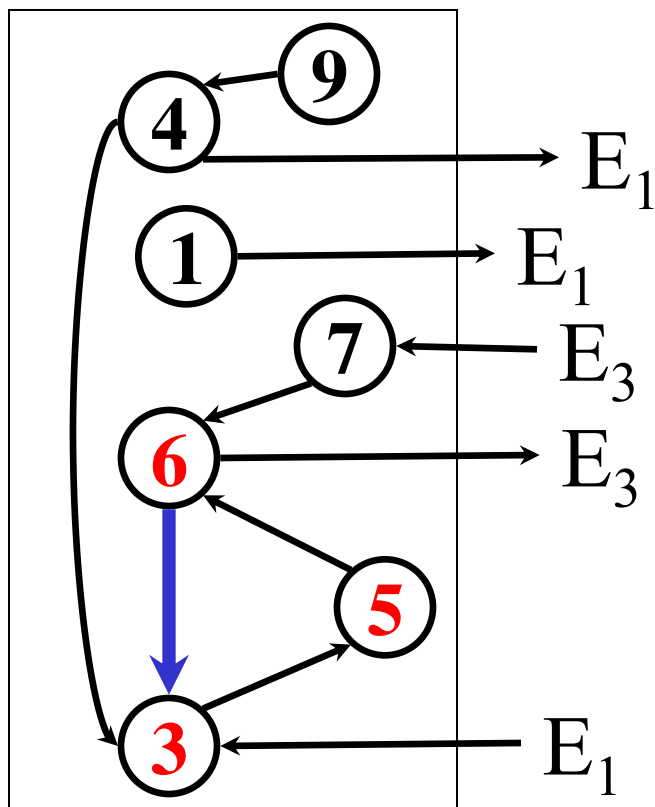


Il messaggio
arrivato non
causa nuovi
messaggi

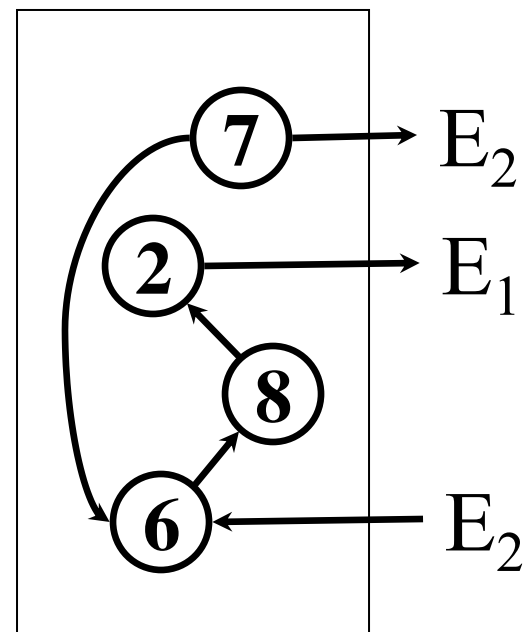
NODO 1



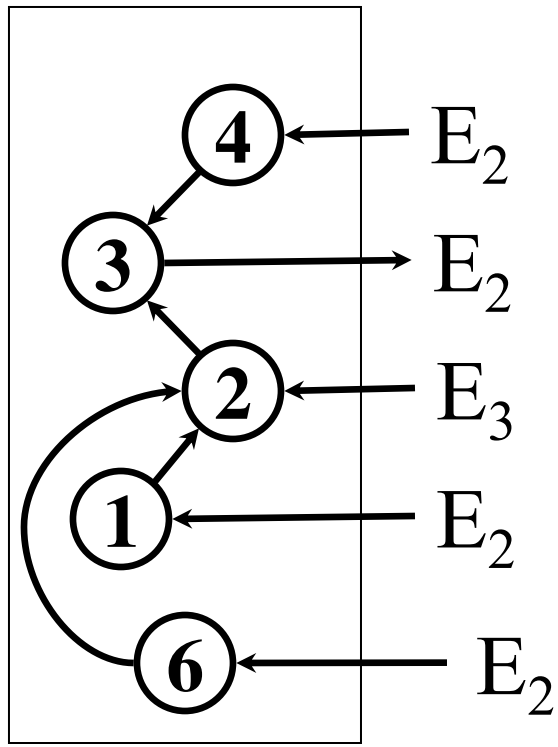
NODO 2



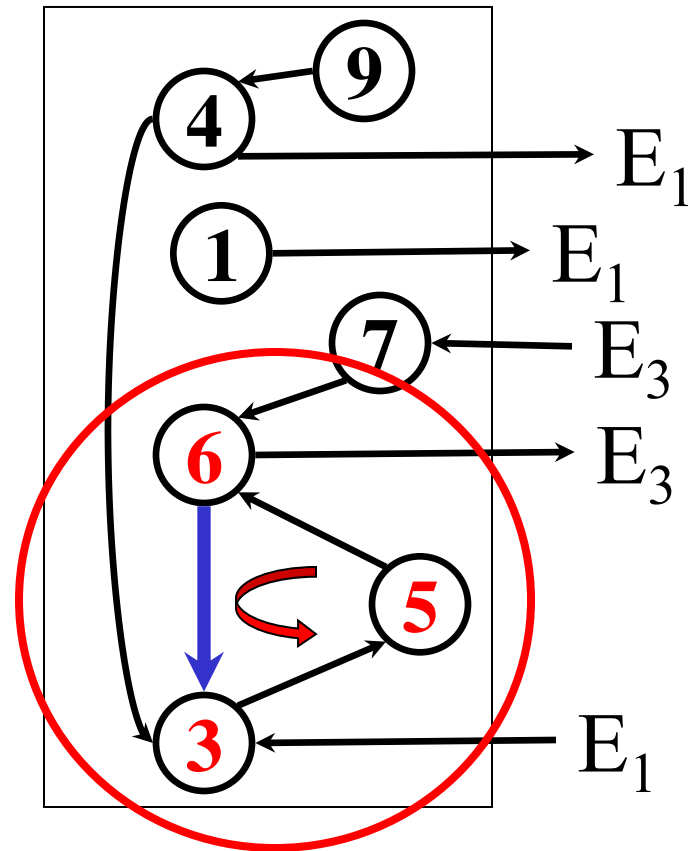
NODO 3



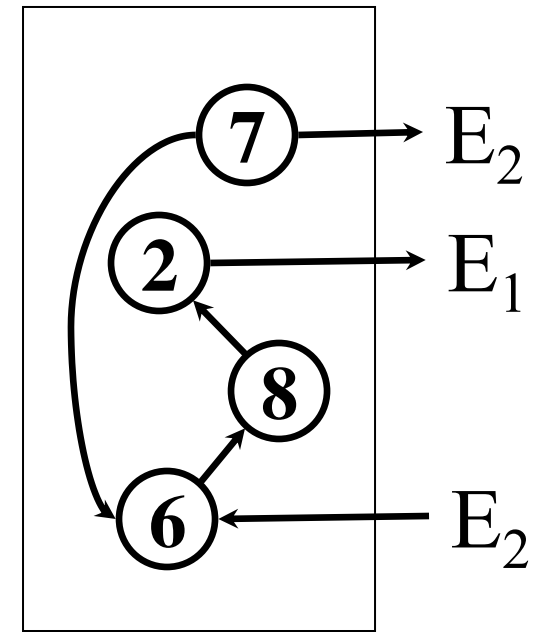
NODO 1



NODO 2



NODO 3



ESISTE UN BLOCCO CRITICO

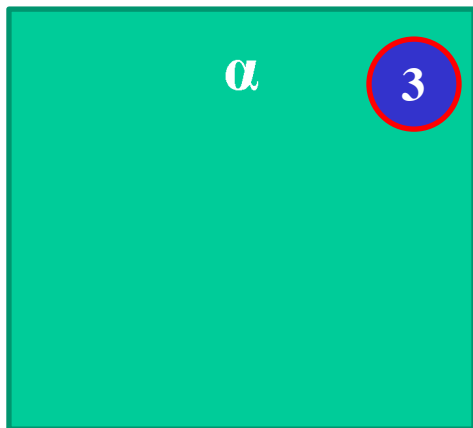
Obermark

Su una base dati distribuita su 3 nodi (α , β , γ) sono in esecuzione sei transazioni $T_1...T_6$, che operano sulle risorse $A...F$, così allocate: A,B,C sul nodo α , D sul nodo β e E,F sul nodo γ .

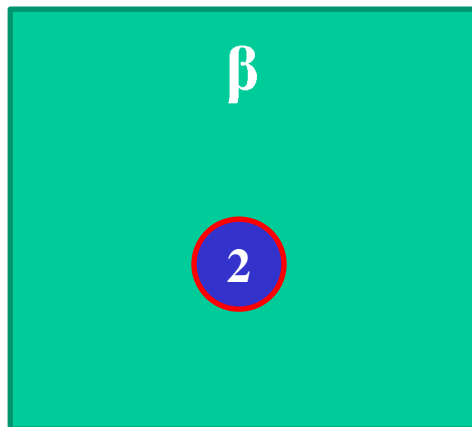
Le operazioni delle transazioni sono state registrate in questo ordine:

$r_1(E), r_2(D), r_3(A), r_2(C), w_1(B), r_4(B), w_4(A), r_3(E), r_5(D), w_1(C), w_3(F), r_6(D), w_5(E), w_6(D)$

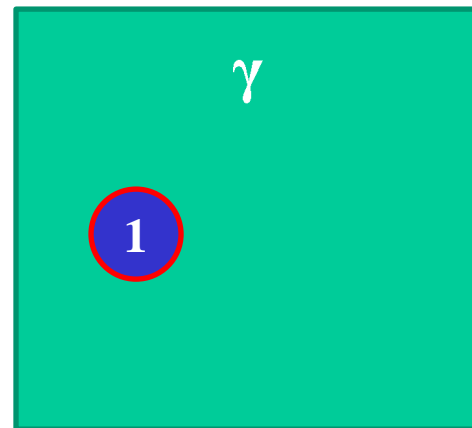
Assumendo che ogni transazione sia iniziata dal nodo su cui si trova la prima risorsa acceduta, e che si verifica l'invocazione di una sotto-transazione quando si accede a una risorsa remota, si costruiscano le condizioni di attesa e le si mostri in forma grafica. Si indichino gli eventuali messaggi da inviare secondo l'algoritmo di Obermark, e se ne simuli l'esecuzione per rilevare eventuali condizioni di deadlock.



A **B** **C**
 $\mathbf{r_3}$

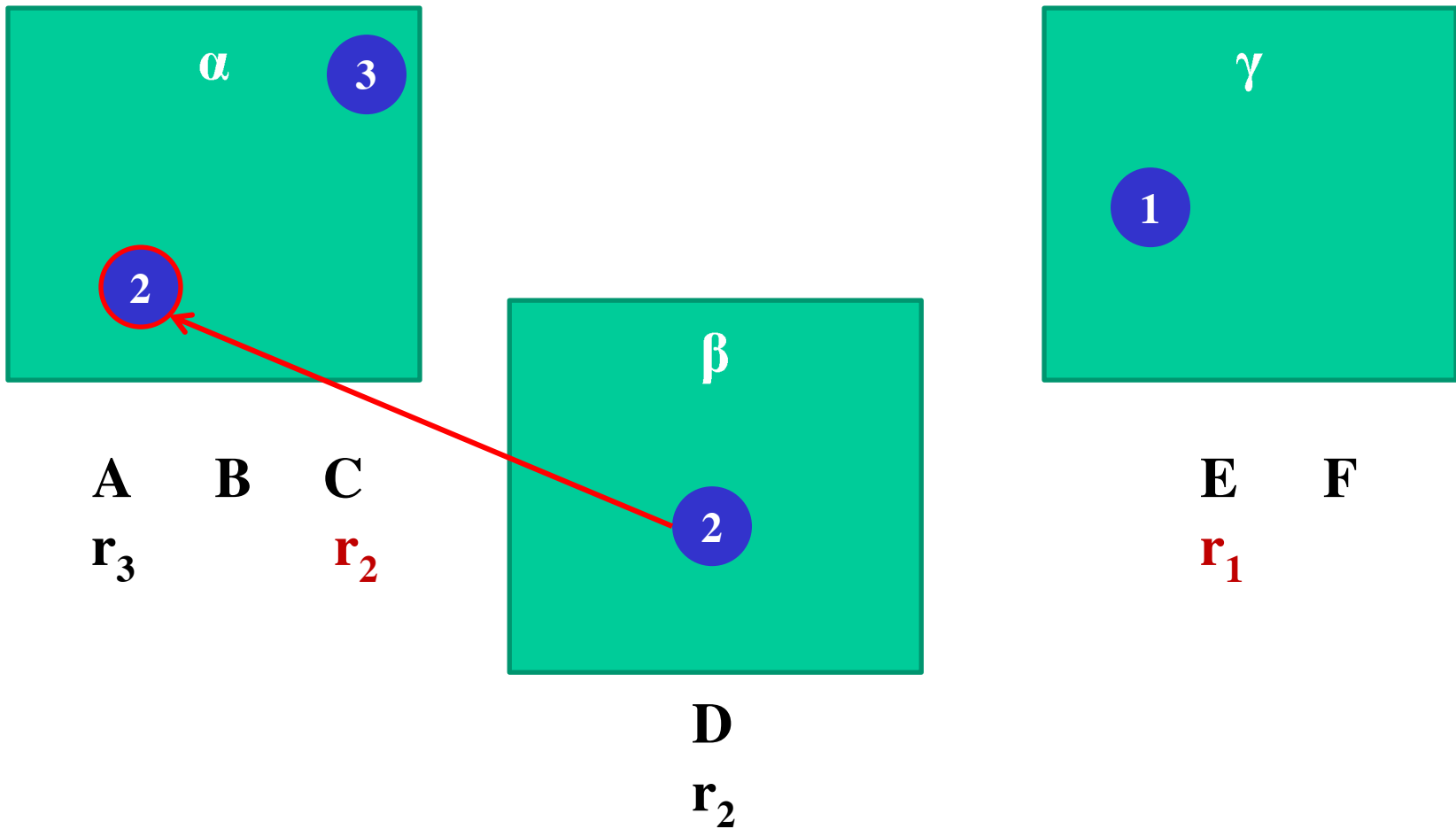


D
 $\mathbf{r_2}$

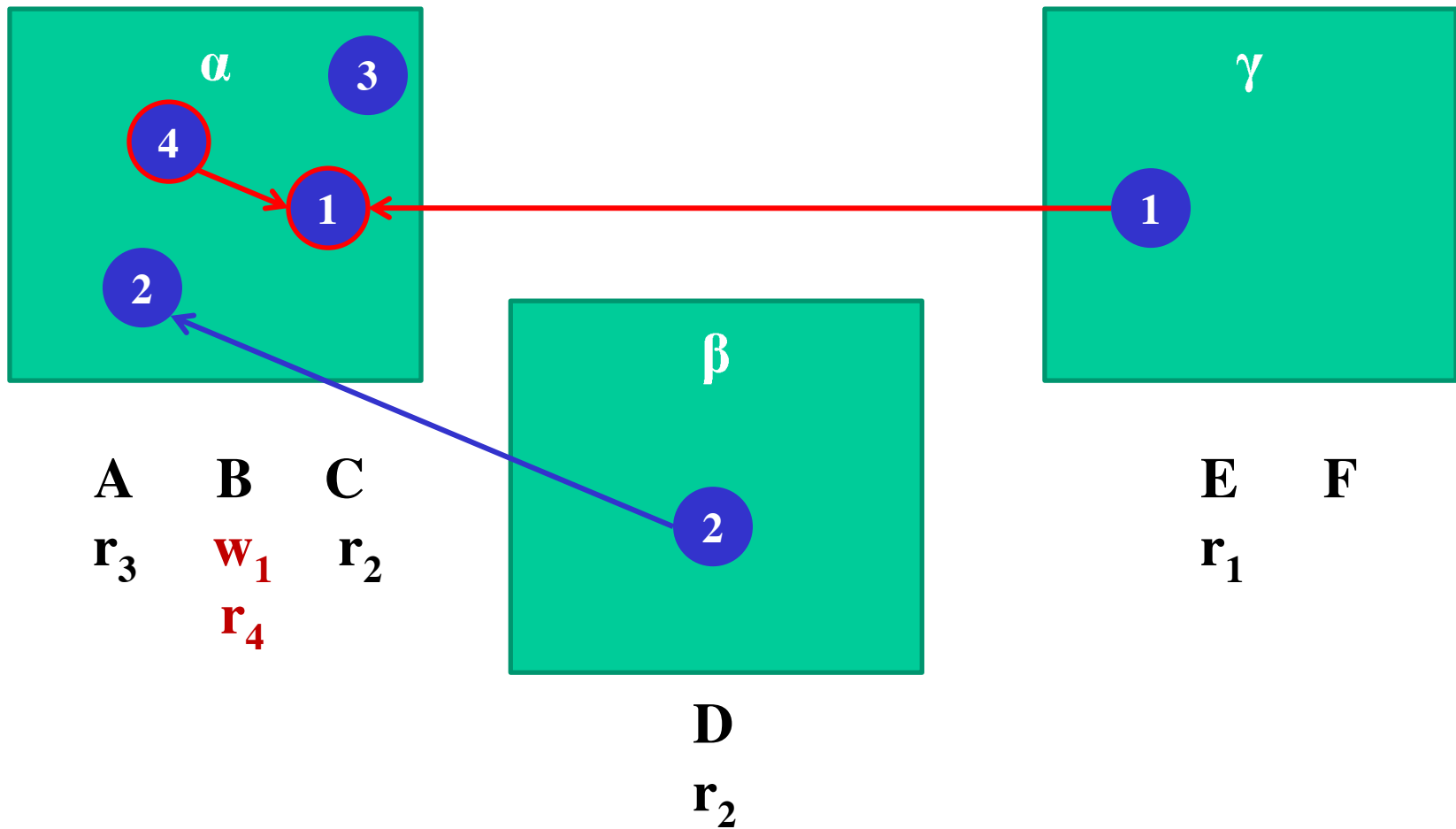


E **F**
 $\mathbf{r_1}$

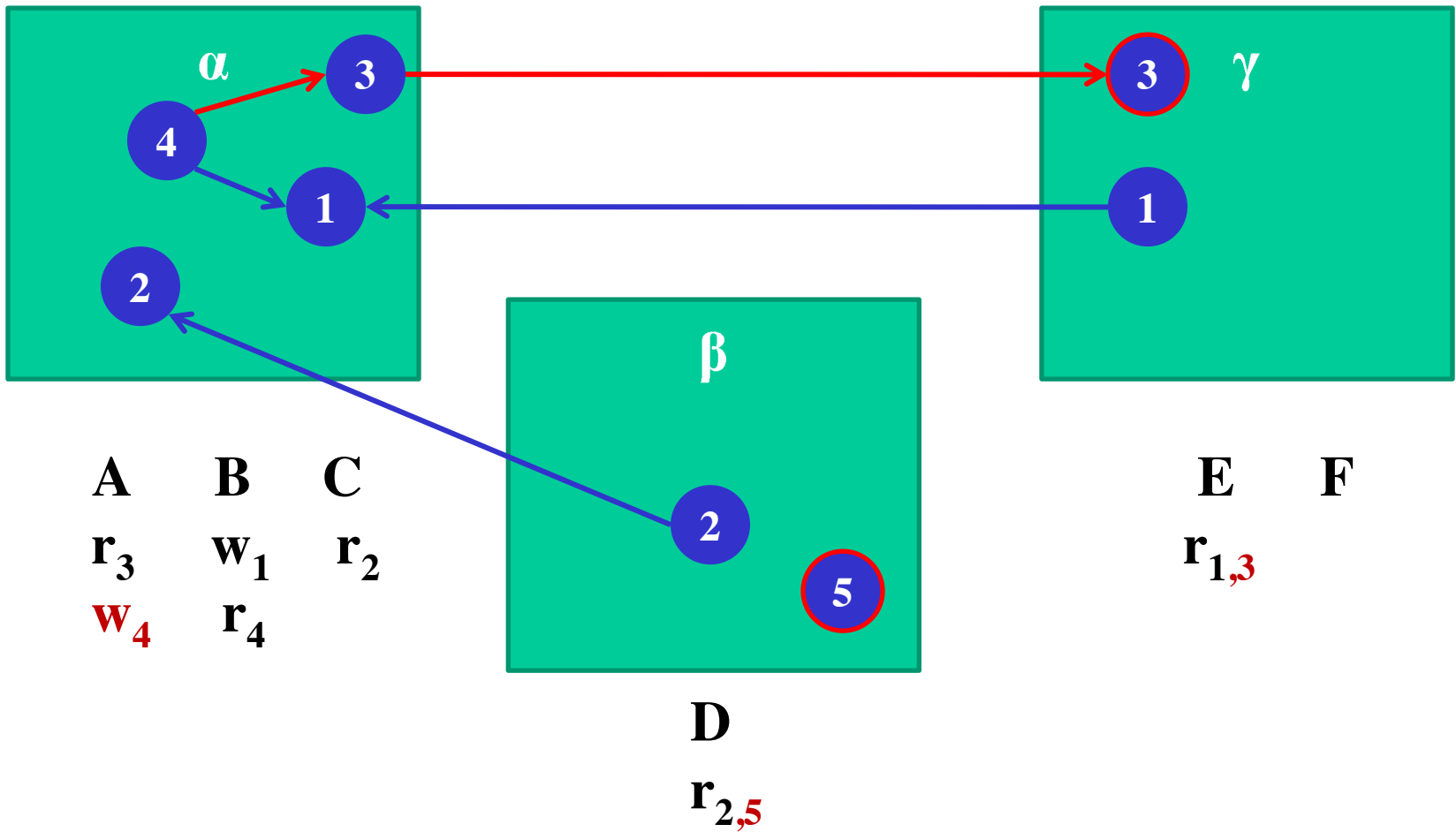
$\mathbf{r_1(E), r_2(D), r_3(A), r_2(C), w_1(B), r_4(B), w_4(A), r_3(E), r_5(D), w_1(C), w_3(F), r_6(D), w_5(E), w_6(D)}$



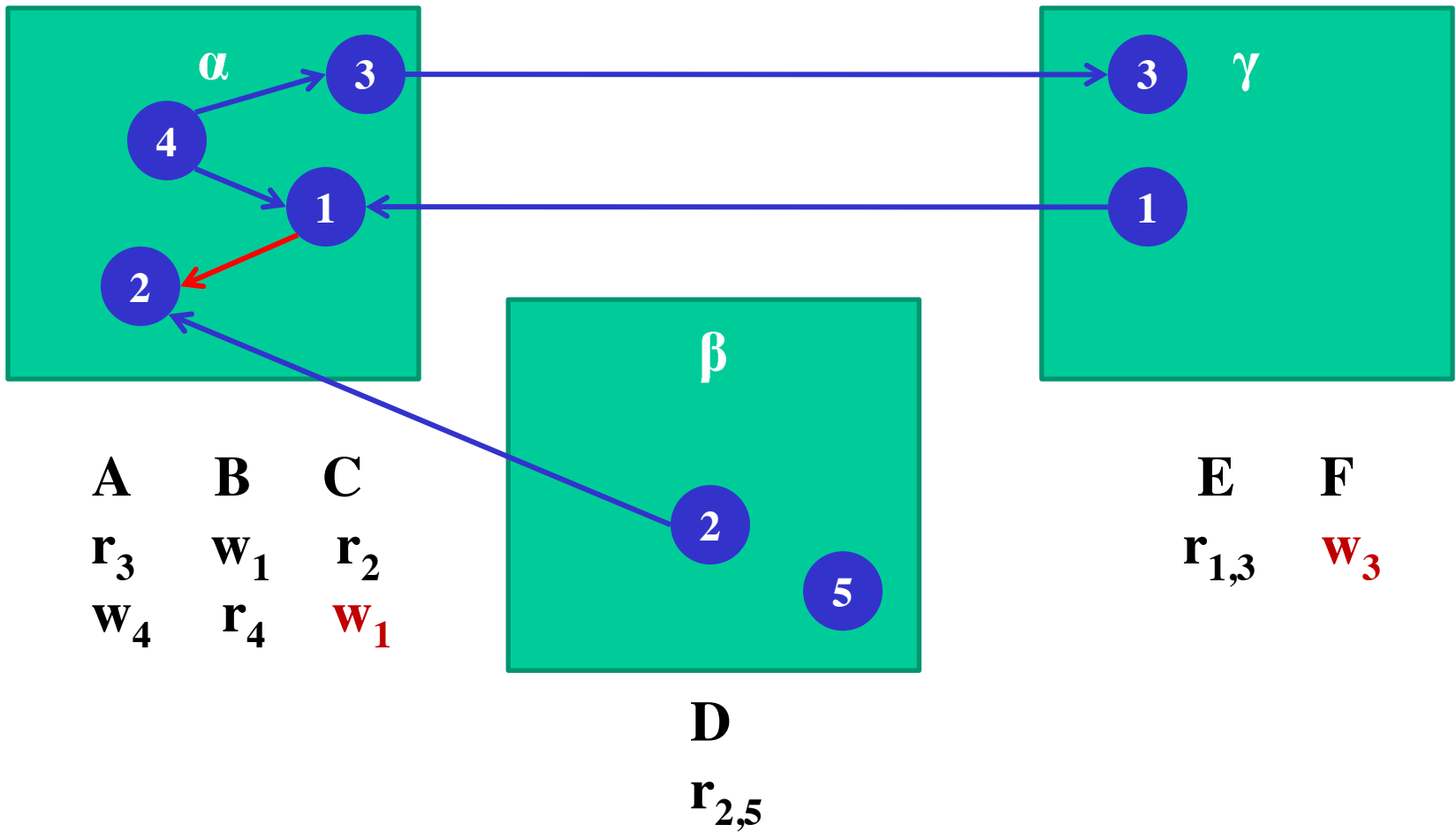
$\mathbf{r}_1(\mathbf{E}), \mathbf{r}_2(\mathbf{D}), \mathbf{r}_3(\mathbf{A}), \mathbf{r}_2(\mathbf{C}), \mathbf{w}_1(\mathbf{B}), \mathbf{r}_4(\mathbf{B}), \mathbf{w}_4(\mathbf{A}), \mathbf{r}_3(\mathbf{E}), \mathbf{r}_5(\mathbf{D}), \mathbf{w}_1(\mathbf{C}), \mathbf{w}_3(\mathbf{F}), \mathbf{r}_6(\mathbf{D}), \mathbf{w}_5(\mathbf{E}), \mathbf{w}_6(\mathbf{D})$



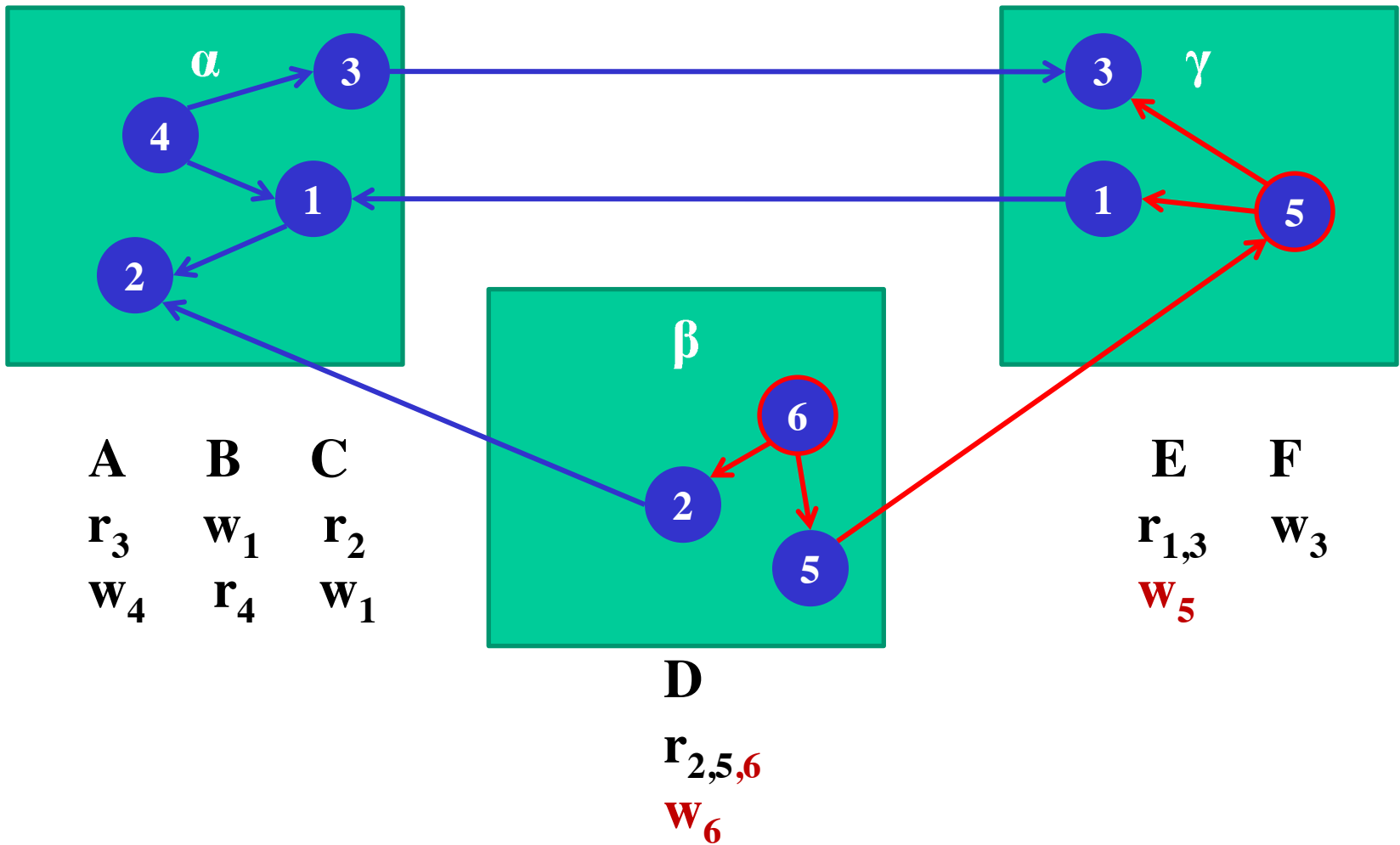
$r_1(E), r_2(D), r_3(A), r_2(C), w_1(B), r_4(B), w_4(A), r_3(E), r_5(D), w_1(C), w_3(F), r_6(D), w_5(E), w_6(D)$



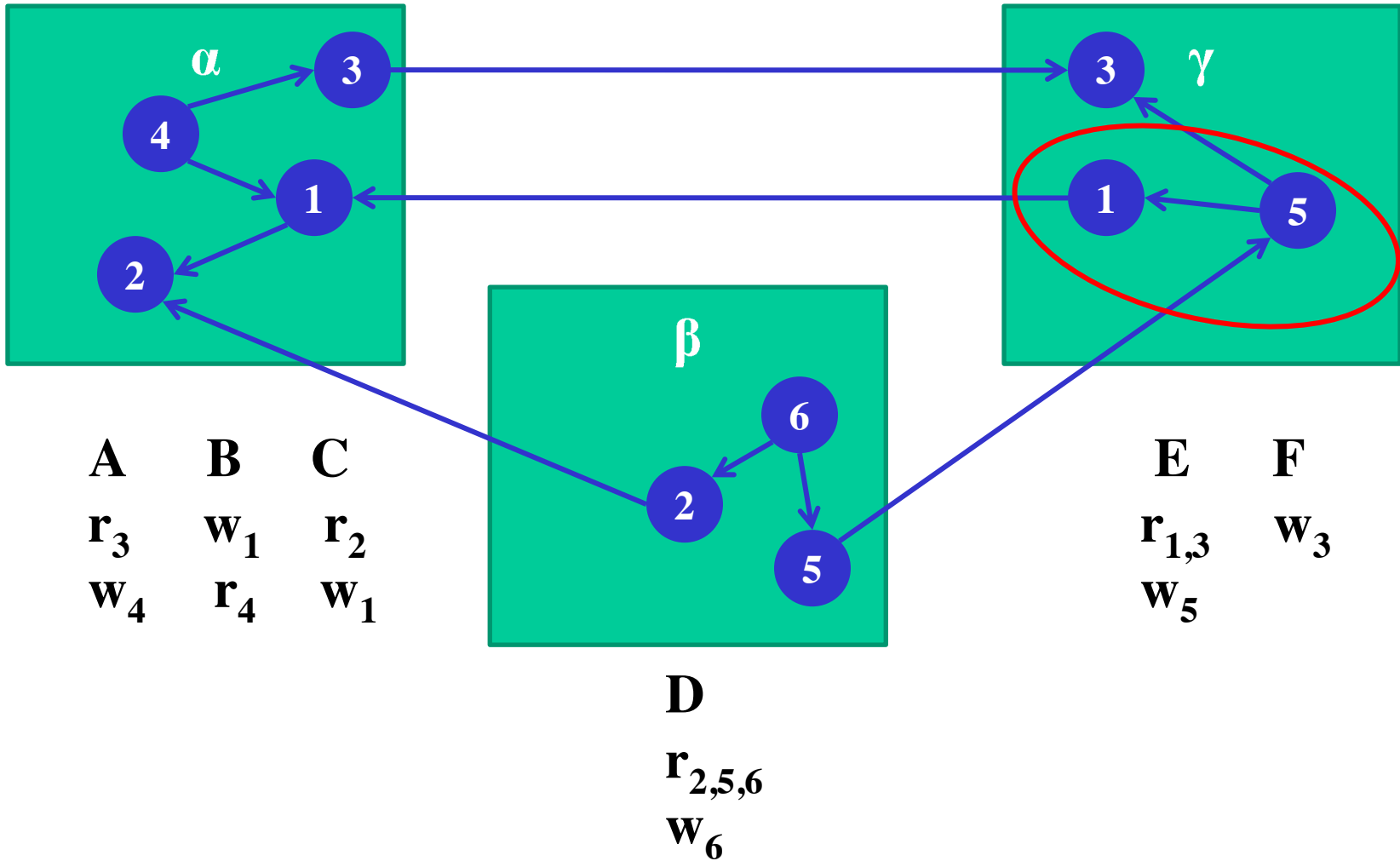
$r_1(E), r_2(D), r_3(A), r_2(C), w_1(B), r_4(B), w_4(A), r_3(E), r_5(D), w_1(C), w_3(F), r_6(D), w_5(E), w_6(D)$



$r_1(E), r_2(D), r_3(A), r_2(C), w_1(B), r_4(B), w_4(A), r_3(E), r_5(D), w_1(C), w_3(F), r_6(D), w_5(E), w_6(D)$



$r_1(E), r_2(D), r_3(A), r_2(C), w_1(B), r_4(B), w_4(A), r_3(E), r_5(D), w_1(C), w_3(F), r_6(D), w_5(E), w_6(D)$



Message: $E_\beta \rightarrow T_5 \rightarrow T_1 \rightarrow E_\alpha$
to node α

No deadlock is found