

PUCRS – Escola Politécnica
Modelos de Computação Concorrente

O problema da Seção Crítica – Exercício

Prof. Fernando Dotti

Exercício

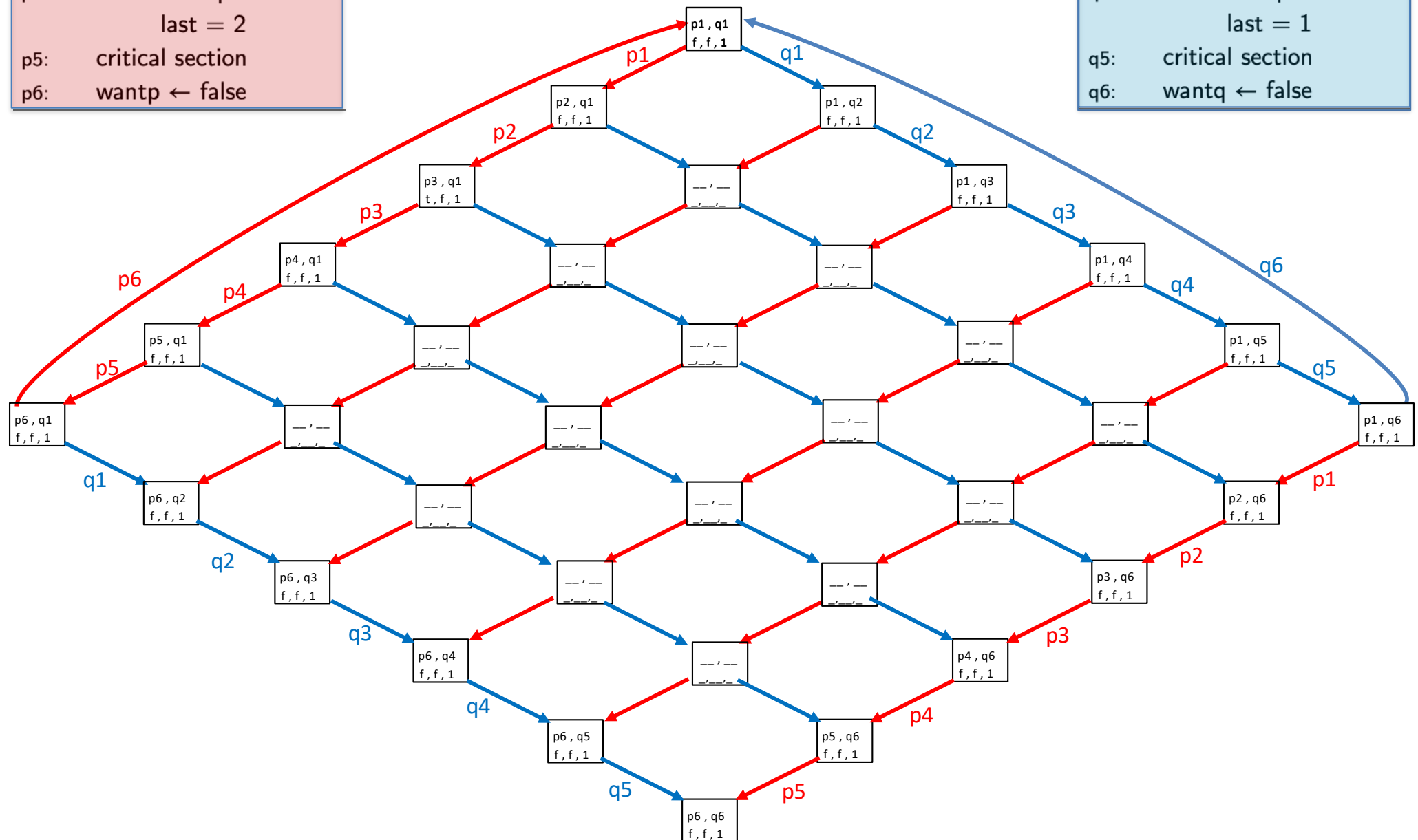
- Para o algoritmo de Peterson para seção crítica para dois processos, por SW:
 1. monte o diagrama de estados e transições completar - e modificar se necessário - o diagrama iniciado – ver próx. slide.
neste processo, transições poderão ser retiradas.
 2. com o diagrama, argumente sobre:
 1. exclusão mútua
 2. progresso
 3. espera limitada

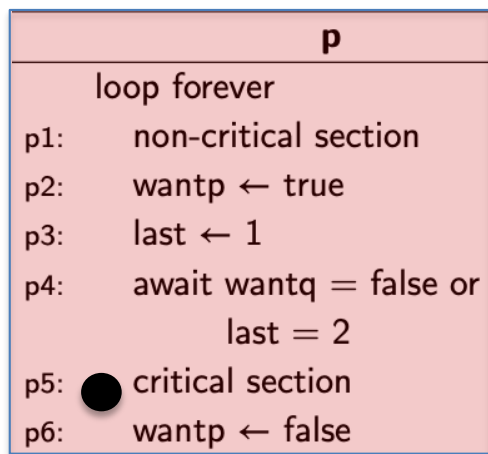
p	
loop forever	
p1:	non-critical section
p2:	wantp \leftarrow true
p3:	last \leftarrow 1
p4:	await wantq = false or last = 2
p5:	critical section
p6:	wantp \leftarrow false

boolean wantp \leftarrow false, wantq \leftarrow false
integer last \leftarrow 1

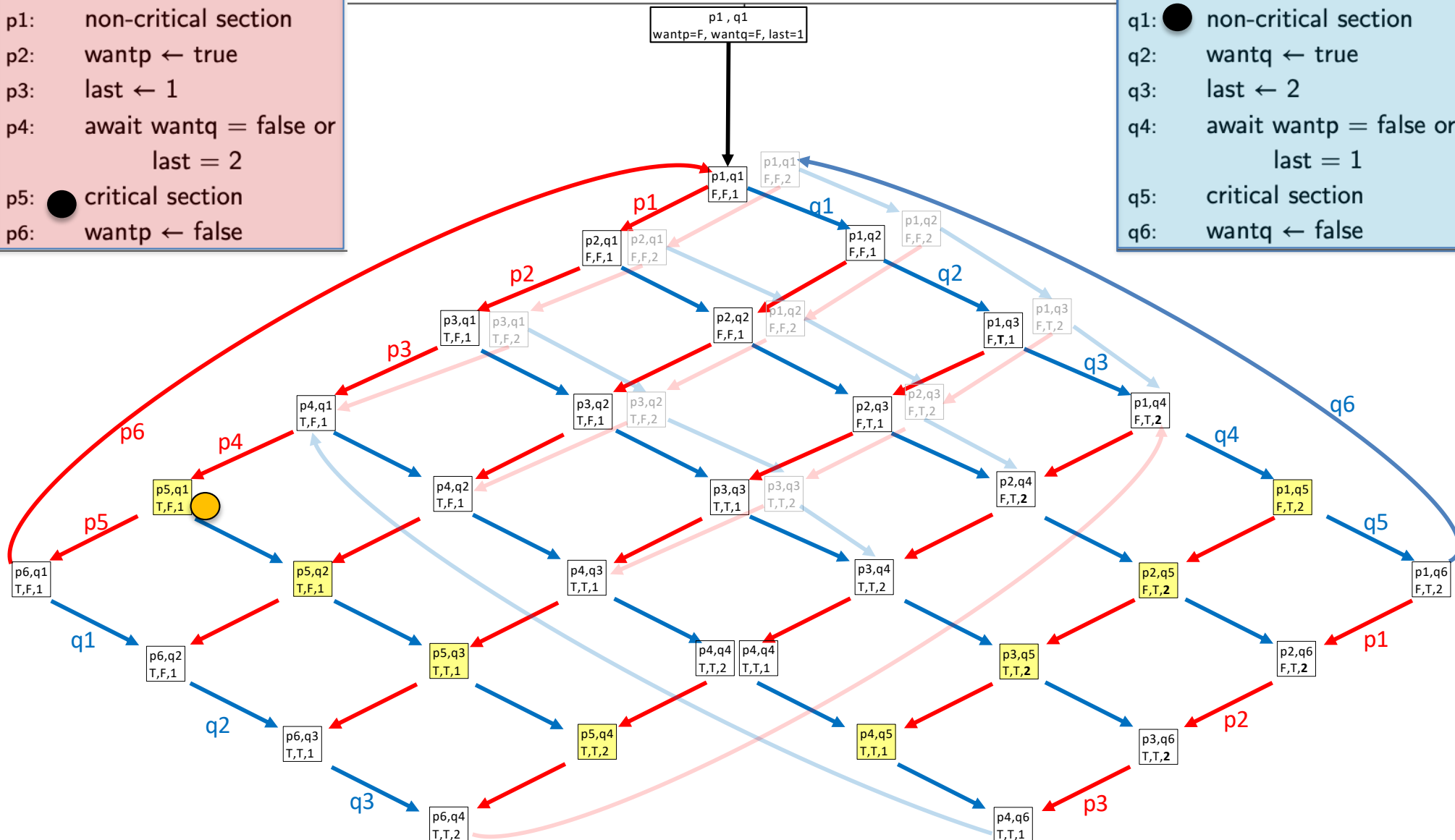
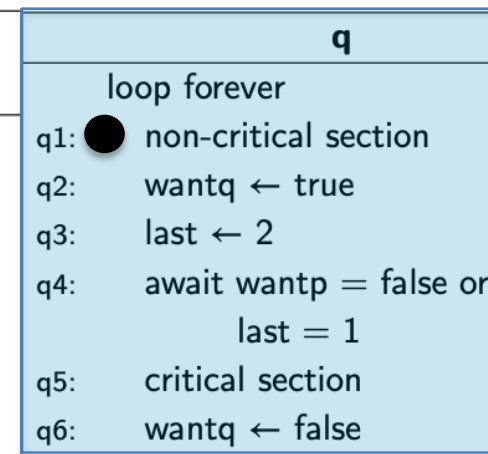
p____, q____
wantp, wantq, last

q	
loop forever	
q1:	non-critical section
q2:	wantq \leftarrow true
q3:	last \leftarrow 2
q4:	await wantp = false or last = 1
q5:	critical section
q6:	wantq \leftarrow false





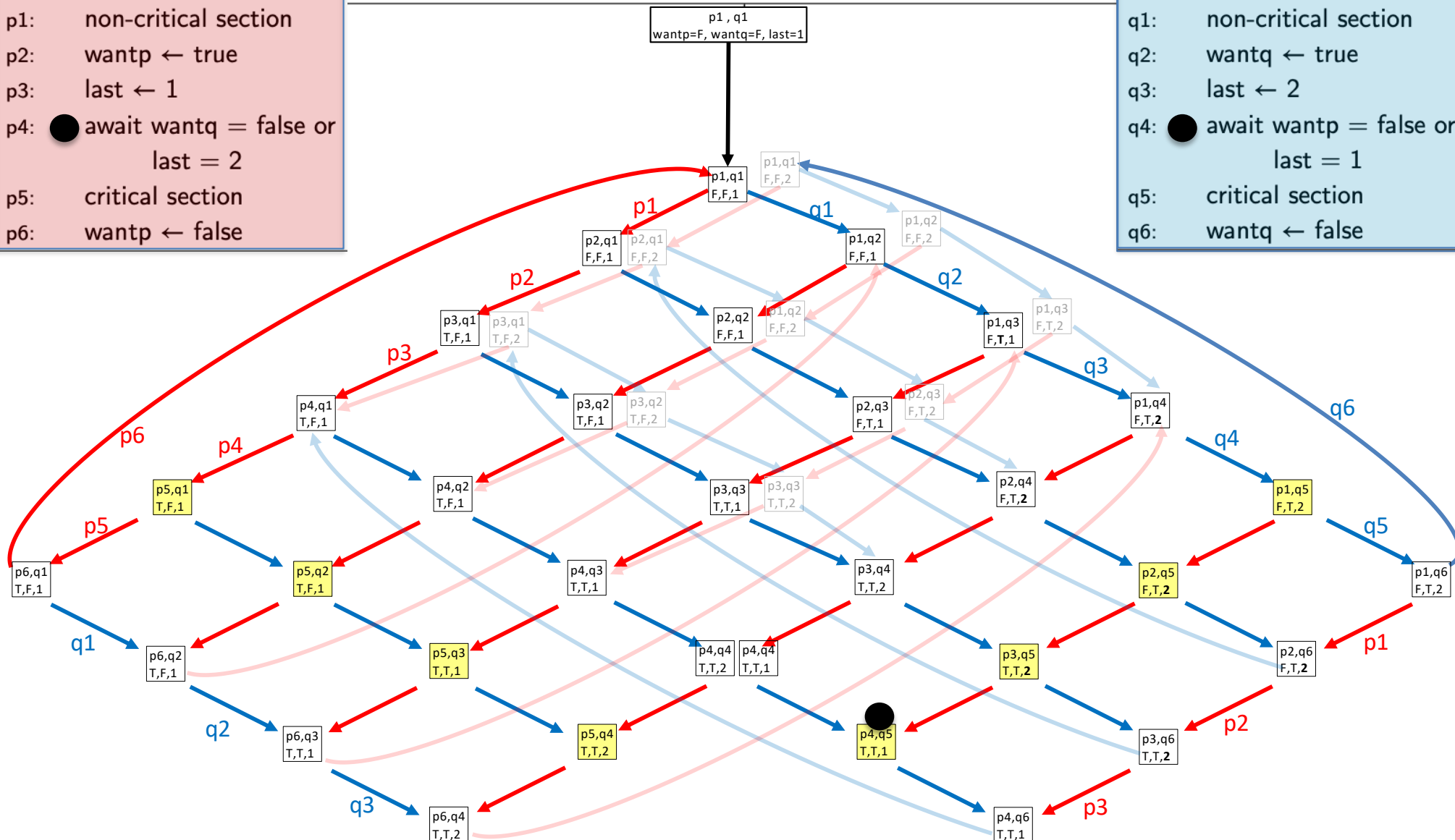
boolean wantp ← false, wantq ← false
integer last ← 1



p	
loop forever	
p1:	non-critical section
p2:	wantp ← true
p3:	last ← 1
p4:	● await wantq = false or last = 2
p5:	critical section
p6:	wantp ← false

boolean wantp ← false, wantq ← false
integer last ← 1

q	
loop forever	
q1:	non-critical section
q2:	wantq ← true
q3:	last ← 2
q4:	● await wantp = false or last = 1
q5:	critical section
q6:	wantq ← false



$$9 + 9 + 12 + 12 = 42 \text{ estados}$$

$$6 \times 4 + 6 \times 4 + 9 + 9 = 66 \text{ transições}$$

Exercício

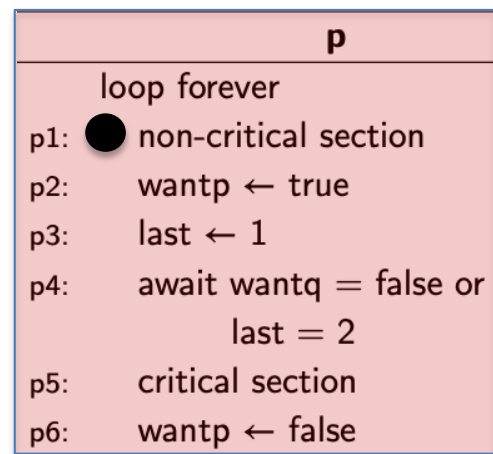
- Para o algoritmo de Peterson para seção crítica para dois processos, por SW:
 - ~~1. monte o diagrama de estados e transições
completar e modificar se necessário o diagrama
iniciado – ver próx. slide.
neste processo, transições poderão ser retiradas.~~
 2. com o diagrama, argumente sobre:
 1. exclusão mútua
 2. progresso
 3. espera limitada
- Entregue seu diagrama e os seus argumentos por escrito na sala aberta para tal.

Exercício

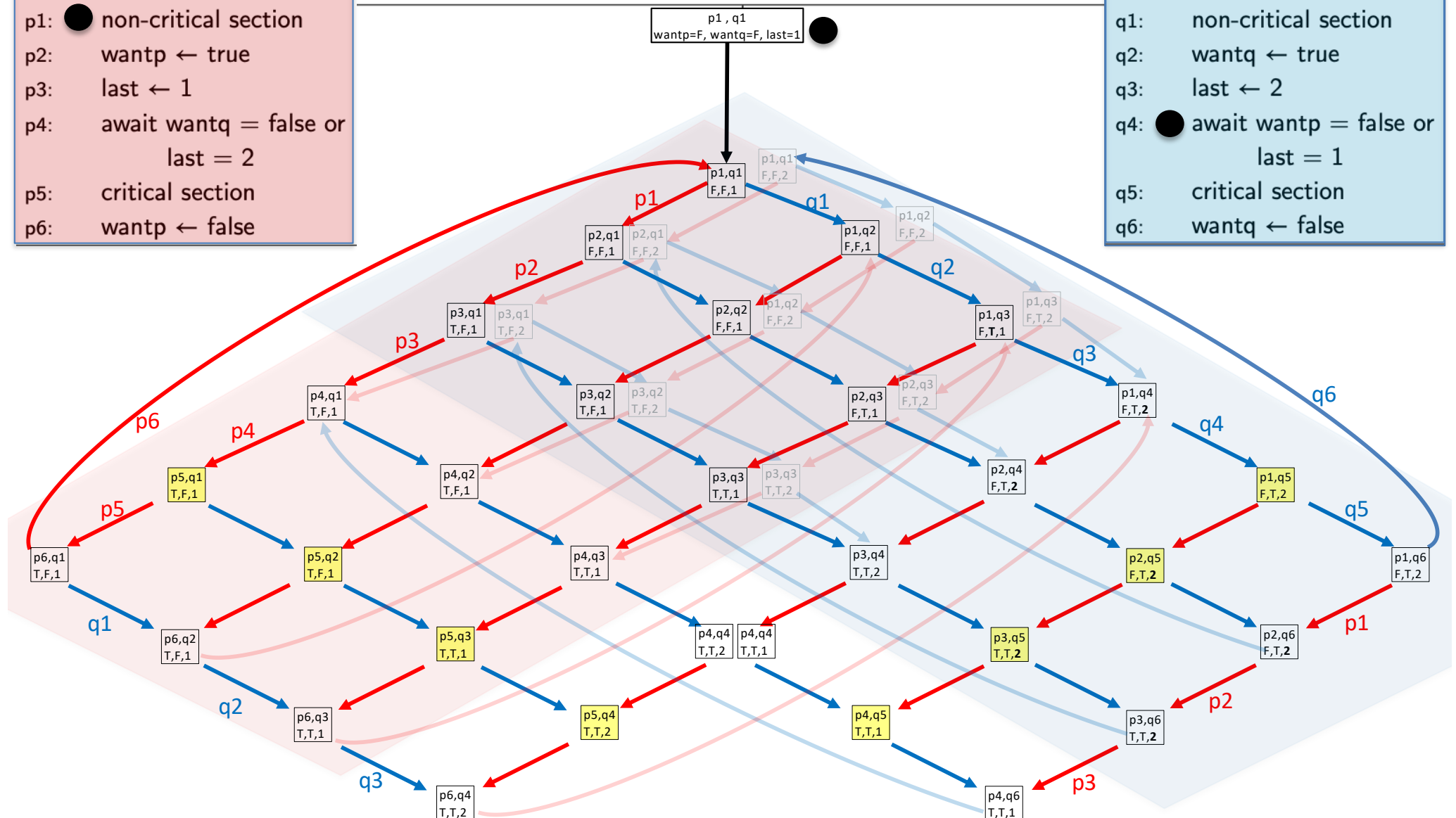
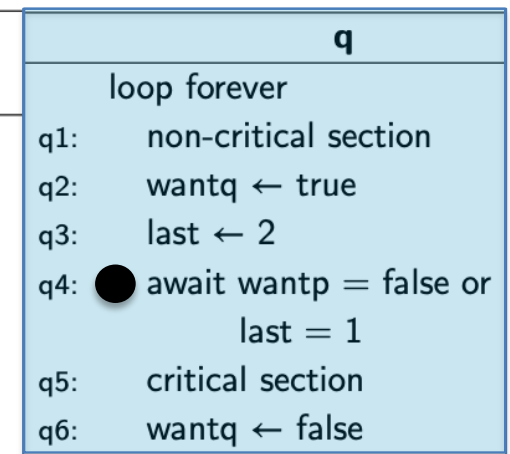
- exclusão mútua
 - não existe estado (p_5, q_5)
 - eles não são alcançáveis

Exercício

- progresso: um processo não é impedido de entrar na SC crítica se outro processo não a disputa
 - pode ser visto pela possibilidade de um processo, por exemplo p, progredir independentemente se q está em q1, q2, ou q3. E vice versa.



boolean wantp ← false, wantq ← false
integer last ← 1



$$9 + 9 + 12 + 12 = 42 \text{ estados}$$

$$6 \times 4 + 6 \times 4 + 9 + 9 = 66 \text{ transições}$$

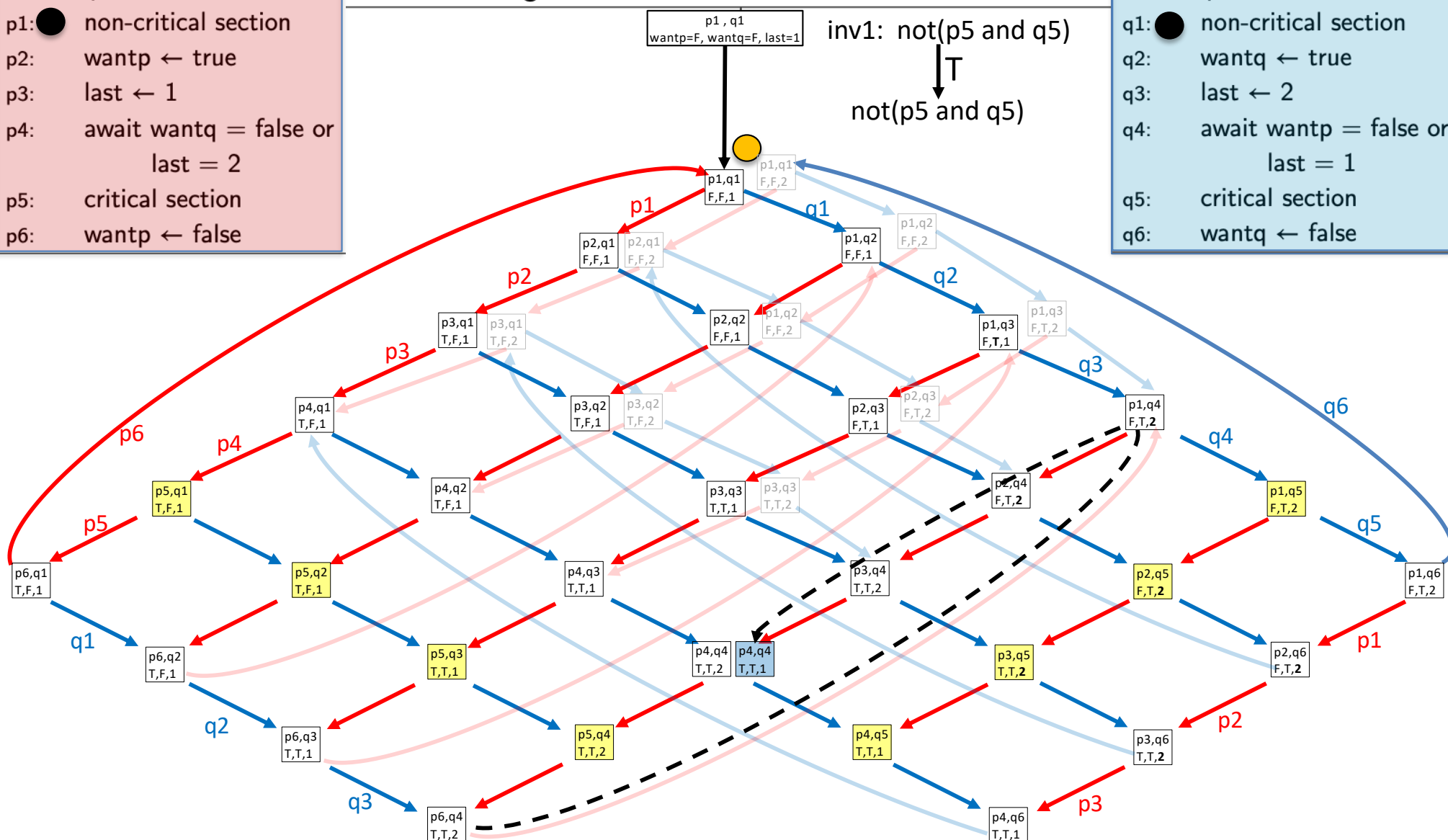
Exercício

- espera limitada: se um processo disputa a SC então ele recebe o direito de acessar em um tempo limitado (não é indefinidamente postergado)
 - pode ser visto pelo desdobramento de todas as computações a partir do ponto que os processos disputam a SC e um recebe o direito de acesso. Nos desdobramentos posteriores, o outro processo sempre prosseguirá para a SC em um número finito de passos.

p	
loop forever	
p1:	● non-critical section
p2:	wantp ← true
p3:	last ← 1
p4:	await wantq = false or last = 2
p5:	critical section
p6:	wantp ← false

boolean wantp ← false, wantq ← false
integer last ← 1

q	
loop forever	
q1:	● non-critical section
q2:	wantq ← true
q3:	last ← 2
q4:	await wantp = false or last = 1
q5:	critical section
q6:	wantq ← false



$$9 + 9 + 12 + 12 = 42 \text{ estados}$$

$$6 \times 4 + 6 \times 4 + 9 + 9 = 66 \text{ transições}$$