

## Desenvolvimento de Software Concorrente - 2020-1

### Atividade 4

1 – Faça o diagrama de estados para o programa abaixo:

boolean wantp false, wantq false	
q	p
loop forever q1: await wantp = false q2: wantq true q3: wantq false	loop forever p1: await wantq = false p2: wantp true p3: wantp false

2 - Construa o diagrama de estados para o programa abaixo e use-o para mostrar que a exclusão mútua é válida:

boolean wantp false, wantq false	
p	q
loop forever p1: non-critical section p2: wantp true p3: await wantq = false p4: critical section p5: wantp false	loop forever q1: non-critical section q2: wantq true q3: await wantp = false q4: critical section q5: wantq false

3 – Para o programa abaixo, responda os itens:

boolean wantp false, wantq false	
p	q
loop forever p1: non-critical section p2: wantp true p3: while wantq p4: wantp false p5: wantp true p6: critical section p7: wantp false	loop forever q1: non-critical section q2: wantq true q3: while wantp q4: wantq false q5: wantq true q6: critical section q7: wantq false

- Construa a forma tabular (ou diagrama de estados) do cenário para starvation como mostrado em 3.5;
- Explique o porquê desse cenário ser considerado starvation e não uma livelock como na terceira tentativa.
- Construa um cenário onde um processo entra em starvation e outro entra na SC infinitamente.

**Referências: Ben-Ari, M. (2006). Principles of Concurrent and Distributed Programming. Boston: Addison-Wesley (Second Edition). ISBN 978-0-321-31283-9.**