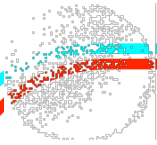


Detecção de término



- **Autor:**

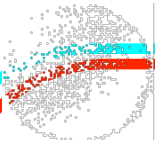
- Tiago Degani Veit
- Revisões: C. Geyer

- **Local**

- Instituto de Informática
- UFRGS
- disciplina: Programação Distribuída e Paralela

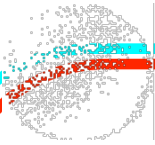
- **Versão atual**

- V12.1, abril 2013

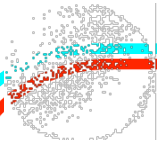


- **Bibliografia:**

- Andrews, G. Concurrent Programming: Principles and Practice. Benjamin-Cummings, 1991.
- Andrews, G. Foundations of Multithreaded, Parallel, and Distributed . Addison-Wesley, 2000.
- Lynch, N. Distributed Algorithms. 1996.

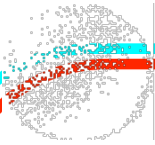


- **Súmula:**
 - Detecção de término: conceitos
 - Detecção de término em anel
 - Detecção de término em grafo

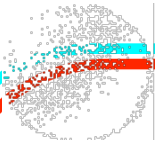


Detecção de término

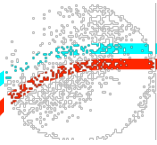
Conceitos



- **Detecção em programa seqüencial:**
 - trivial.
- **Detecção em programa concorrente com 1 processador:**
 - todos os processos terminados ou bloqueados
 - e sem operações de E/S pendentes.

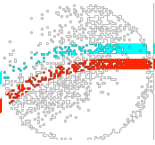


- **Detecção de término em programa distribuído:**
 - estado global do processamento não visível
 - pode haver mensagens em trânsito
 - Para término, essas mensagens devem ser recebidas e processadas
 - Como saber que não há mais mensagens em trânsito?
 - Solução: “empurrar” as mensagens não recebidas



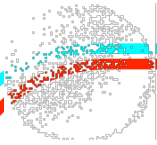
Detecção de término em Anel

Algoritmo passagem de token



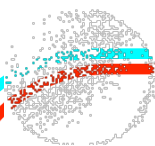
- **Algoritmo de passagem de token**

- Token: mensagem especial
 - ❑ não faz parte da computação propriamente dita.
- o processo que tem o token o passa quando se torna inativo.
 - ❑ problema chave: como e quando detectar a inatividade?
- se ao dar uma volta completa no anel
 - ❑ o nodo inicial ainda estiver inativo, é porque
 - ❑ todos os nodos estão inativos
 - ❑ e não existem mensagens em trânsito
 - ❑ e então a computação termina.

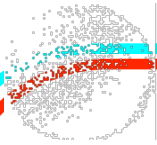


- **Algoritmo de passagem de token**

- toda comunicação é por anel unidirecional
- processos: dois estados possíveis:
 - ❑ ativo ou inativo (idle)
- inicialmente todos ativos
- ao receber token, torna-se inativo, e repassa token no anel
- processos inativos não enviam mensagens mas podem estar esperando por alguma
- recebendo uma mensagem um processo inativo torna-se ativo
- canais são FIFO
 - ❑ mensagens de mesma origem e destino são ordenadas



Código



- **Nodo geral:**
 - dois estados possíveis: vermelho(ativo) e azul(inativo)
- **Nodo P[1] determina o final da computação.**
 - ele começa a passagem do token e nele ela acaba.
- **Estilo de código**
 - Ação ao receber uma mensagem

- **Código:**

```
/* propriedade do anel */
```

```
{ RING: P[1] is blue → (P[1] ... P[token+1] are blue
```

```
^
```

```
ch[2] ... Ch[(token mod n)+1] are empty ) }
```

actions of P[1] when it first becomes idle:

```
color[1] := blue;  
token := 0;  
send ch[2](token);
```

actions of P[i: 1..n] upon receiving a regular message:

```
color[i] := red;
```

- **Código:**

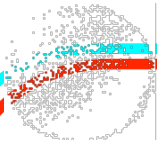
`actions` of $P[i: 2..n]$ upon receiving the token

```
color[i] := blue;  
token := token + 1;  
send ch[(i mod n) + 1] (token);
```

`actions` of $P[1]$ upon receiving the token

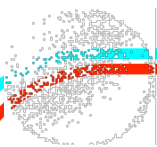
```
if color[1] = blue → announce termination and halt  
fi
```

```
color[1] := blue;  
token := 0;  
send ch[2] (token);
```



Exemplo Gráfico

Anel de 3 nós



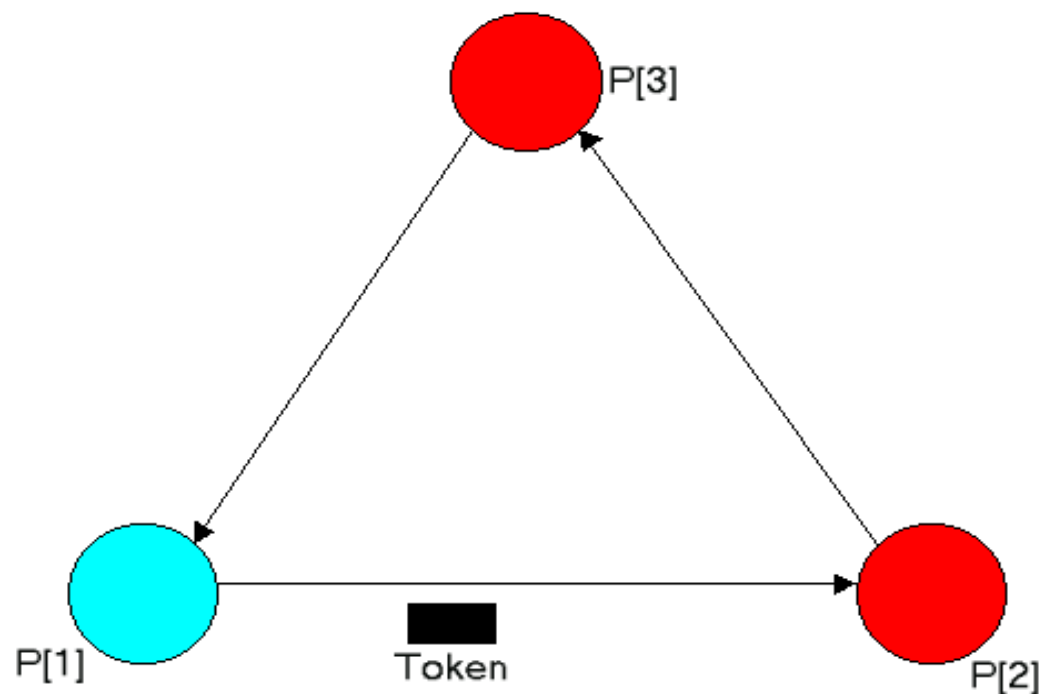
Algoritmo para detecção em Anel

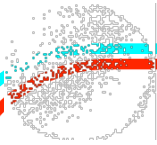
O nodo P[1]:

termina sua computação

fica inativo e

lança o token para o
próximo nodo



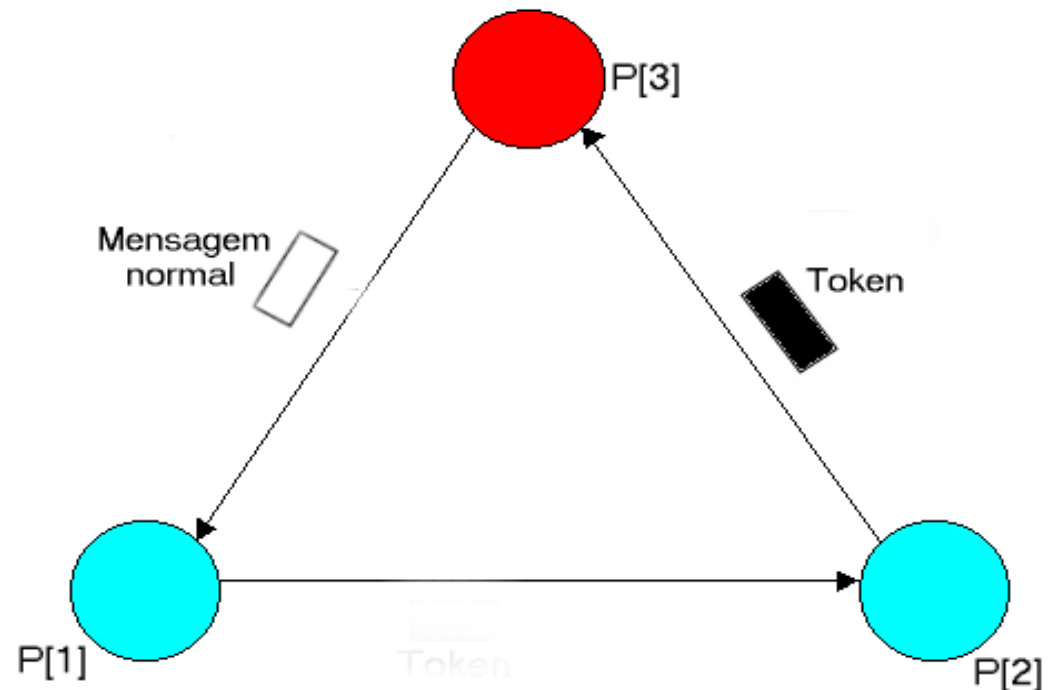


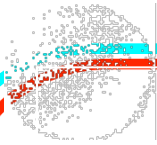
O nodo P[1] :

não está livre de receber mensagens normais.

Quando o token chegar a P[1], este estará ativo.

Então o token deverá girar outra vez pelo anel até chegar em P[1] novamente.



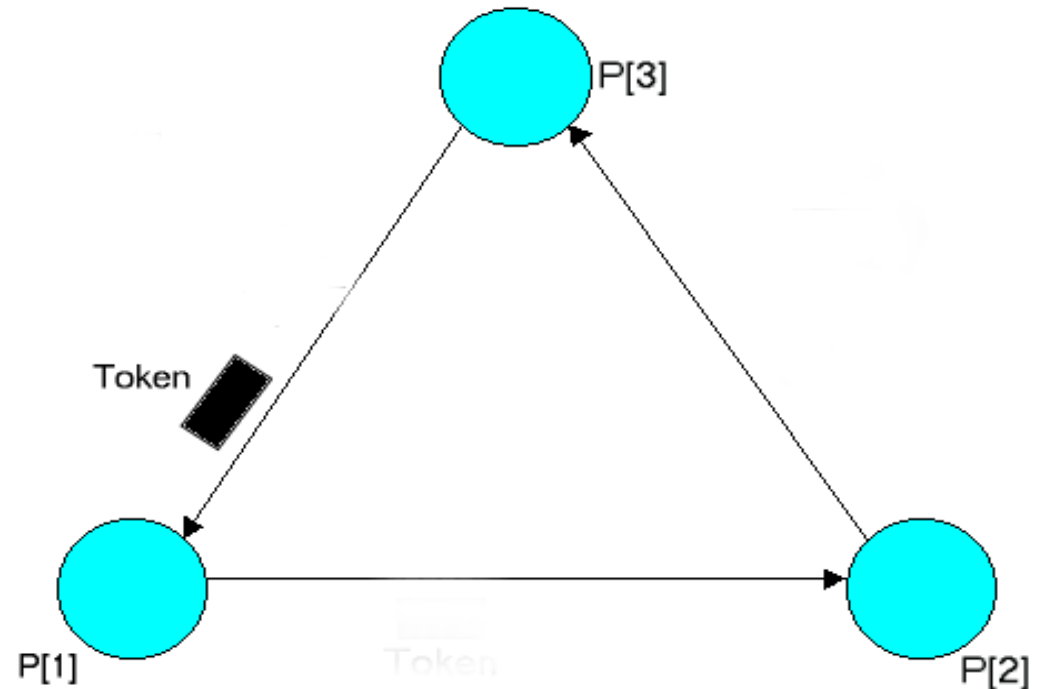


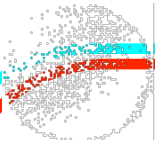
Algoritmo para detecção em Anel

. token:

deu a volta inteira no anel e
não existem mais
mensagens em trânsito.

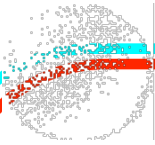
Chegando em P[1], é
detectado o final da
computação.





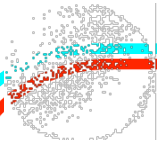
- **Complexidade**

- Em caso de término:
 - n mensagens
 - n passos
- Em caso de não término
 - Complexidade de término * #tentativas



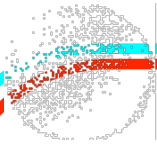
- **Exercícios**

- A) mostre as mensagens de um caso com estados e mudanças para cada nó considerando
 - ❑ ao menos 4 nós
 - ❑ mensagens de aplicações são emitidas concorrentemente ao token



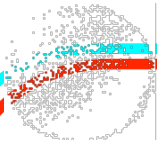
● Revisão

- explique as diferenças entre verificação de término de
 - ❑ programa seqüencial
 - ❑ programa concorrente em memória compartilhada
 - ❑ programa distribuído (com troca de mensagens)
- quais as condições de término do PD
- qual a principal dificuldade da verificação para PD?
- qual a idéia central do algoritmo de verificação (por anel)?



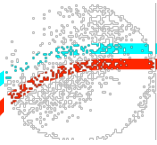
- **Revisão**

- quem e quando lança o token?
- quais os estados dos nós?
 - inicial
 - quando são alterados?
- qual a condição para término?
 - justifique quando é verdadeira e falsa

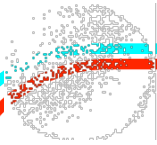


Token em Anel

Fim

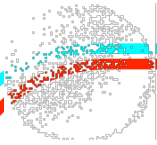


Token em Grafo Completo



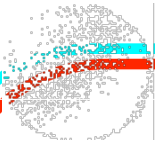
- **Súmula**

- problema do token simples em grafo completo
 - contra-exemplo
- conceito de caminho completo
 - comprimento
- estados e mudanças
- condição de término

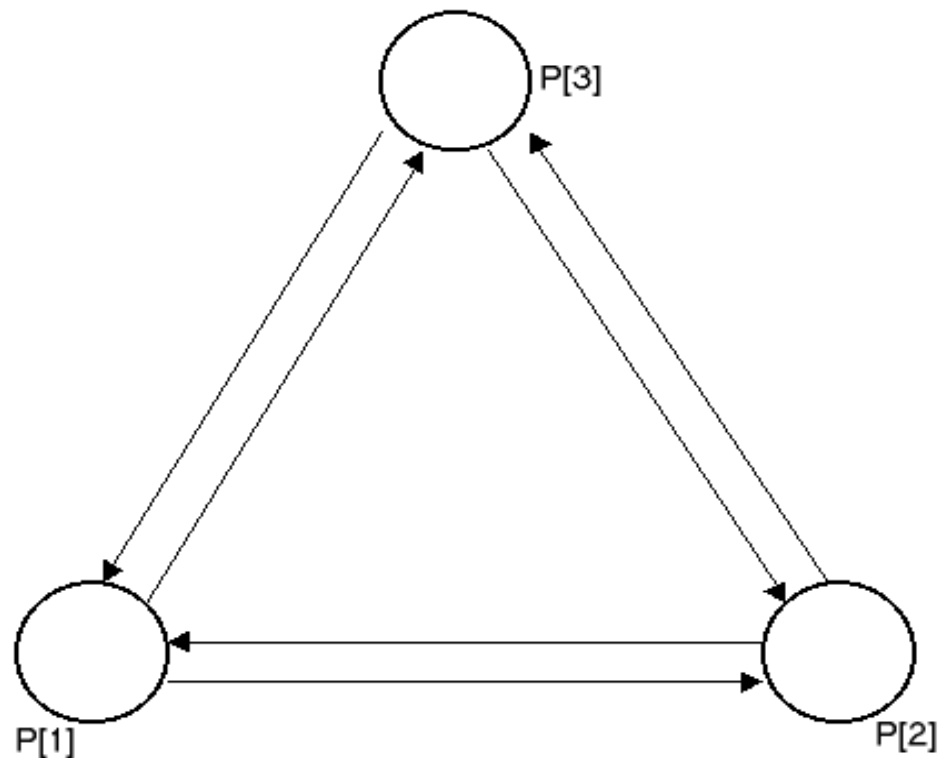


Términação em Grafo

Problema algoritmo anel sobre grafo

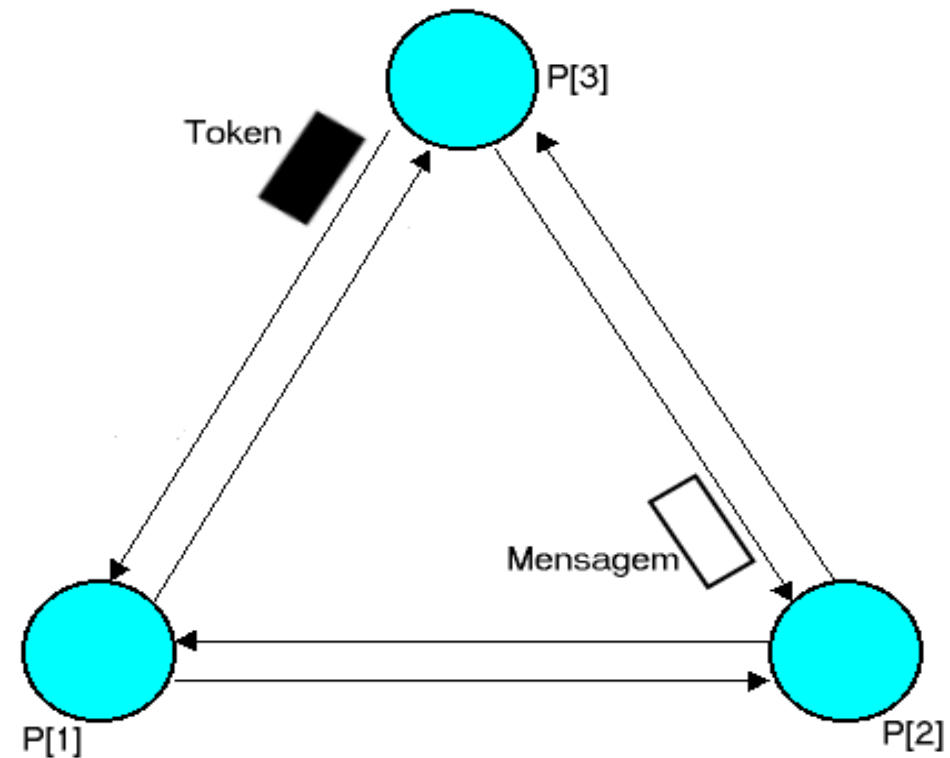


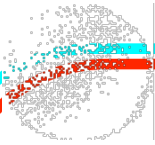
- **Grafo completo:**
mensagens podem chegar por
qualquer canal de comunicação
(detecção mais difícil)



- **Contra-exemplo do algoritmo anel**

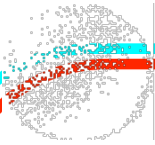
- Aqui, P[3], antes de passar o token para P[1], manda uma mensagem regular para P[2]. Esta mensagem irá acordar P[2]
- Portanto, não se garante que, se após o token completar o ciclo, P[1] estiver inativo, a computação terminou





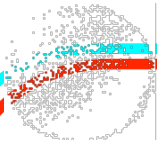
Término em Grafo

Algoritmo



- **Idéia geral do algoritmo para grafo**

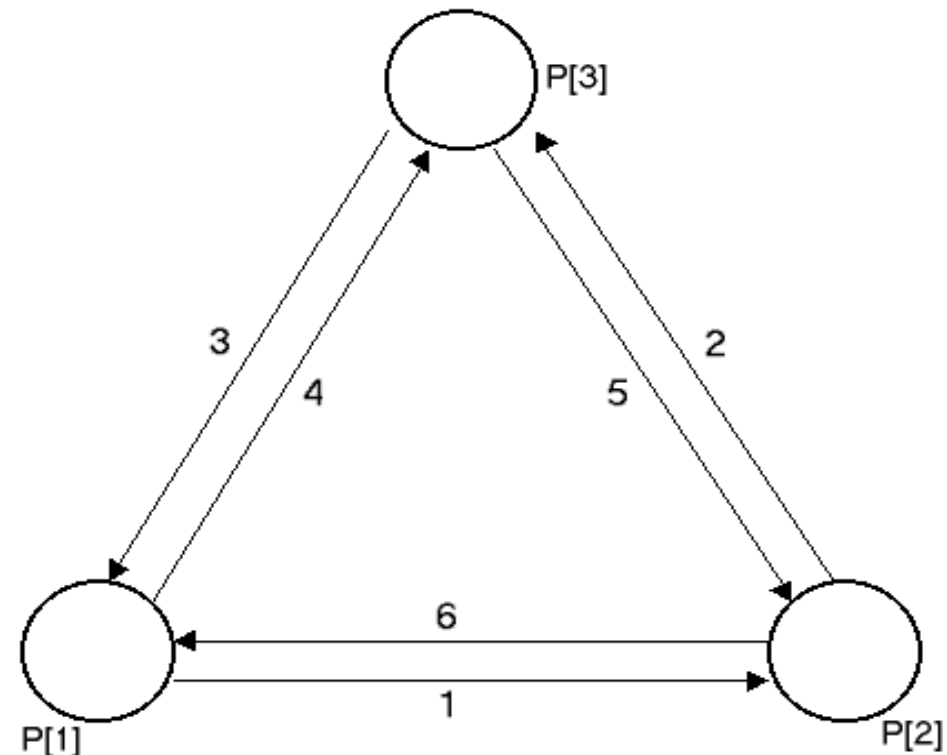
- extensão do algoritmo de detecção em anel
- o token passa por todos os arcos do grafo, visitando cada nodo múltiplas vezes
- todos os canais serão percorridos
 - ❑ todas as mensagens de aplicação, atrasadas, serão “empurradas”
 - ❑ canais são bidirecionais, independentes -> percorrer nos 2 sentidos
 - ❑ ou considerar como 2 canais unidirecionais
- se todos os processos continuaram inativos desde a primeira visita do token, a computação pode ser dada como terminada



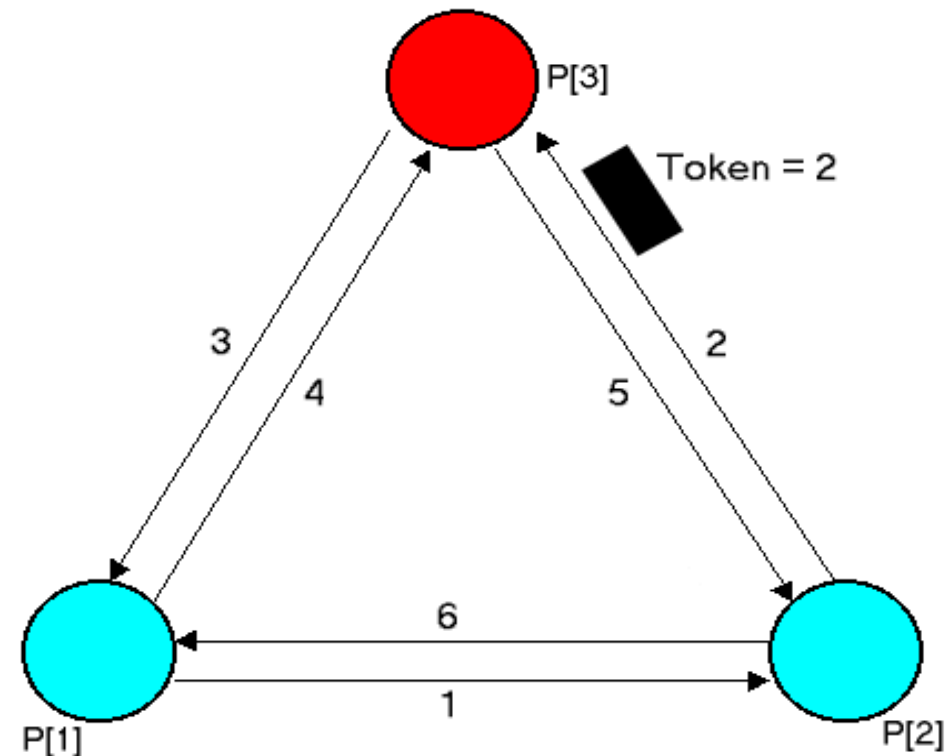
Exemplo Gráfico

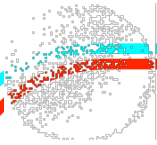
Grafo com 3 nós

- Qualquer grafo orientado contém um ciclo que inclui todos os arcos do grafo. Um exemplo pode ser o da figura ao lado
- Cada processo guarda a ordem dos arcos do ciclo
- Recebendo o token de um arco, o processo o passa pelo arco seguinte no ciclo, garantindo o completo percurso do ciclo

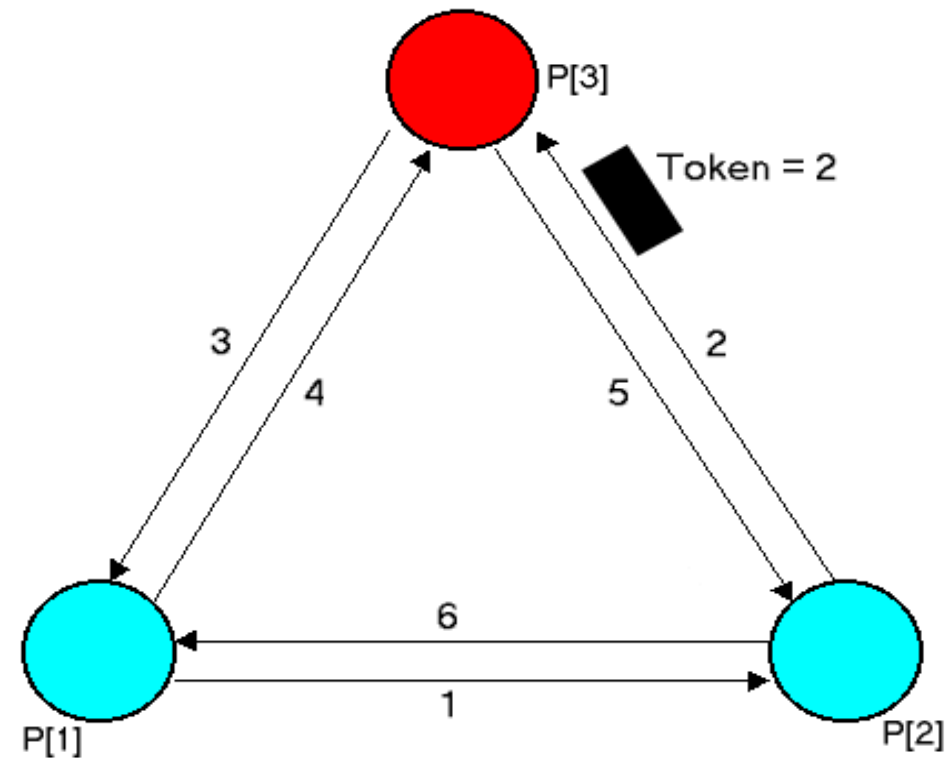


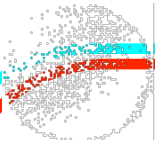
- Ao token é adicionado um contador. A cada nodo azul o token é incrementado
- Se encontrar um nodo vermelho, o contador é zerado
- Se o contador for igual ao tamanho do ciclo, a computação termina





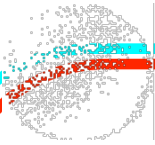
- Quando o token atinge $P[3]$, ele é zerado e passado para o canal 3
- Neste caso, a detecção deve acabar em $P[3]$





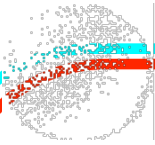
- **Complexidade**

- #mensagens se término
- # passos se ...
- Dica: considere que todos os nós inicialmente estão ativos



- **Resumo**

- Motivação para o algoritmo
- Algoritmo para topologia em anel
 - ❑ Simples: código, prova, cálculo da complexidade
- Algoritmo sobre grafo completo
 - ❑ Problema com algoritmo token em anel
 - ❑ Exige cálculo de caminho por todos os arcos unidirecionais
 - ❑ Complexidade maior: 2 fases, maior # de mensagens, ...

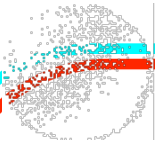


● Exercícios

- A) mostre as mensagens de um caso com estados e mudanças para cada nós considerando
 - ❑ defina o caminho completo
 - ❑ ao menos 4 nós
 - ❑ 1- sem nenhuma mensagem de aplicação
 - ❑ 2- com uma ou duas mensagens de aplicação atingindo um nó após ter recebido o token
- B) explique (justifique) porque o contador deve ser zerado cada vez que o token atinge um nó ativo

• Revisão

- Porque a detecção de término em PD pode ser complexa? (mais que em PC sobre SM)
- Detecção em anel
 - ❑ topologia das mensagens da aplicação?
 - ❑ quais os passos típicos de um processo aplicação (padrão de comportamento)?
 - ❑ quando o token é disparado?
 - ❑ qual o estado inicial dos nós?
 - ❑ qual o estado após token?
 - ❑ qual o estado ao receber mensagem de aplicação?
 - ❑ qual o critério de término?
 - ❑ porque?



● Revisão

➤ Detecção em grafo completo

- ❑ porque algoritmo anel não pode ser usado?
- ❑ como se calcula um ciclo?
- ❑ qual o valor de um token?
- ❑ ação de processo ativo ao receber token?
- ❑ ação de processo inativo ao receber token?
- ❑ ação de processo ina/ativo ao receber mensagem aplicação?
- ❑ critério de término?