Detecção de Terminação

Algoritmos Distribuídos

Professora: Lúcia Drummond

Detecção de Terminação

- Os algoritmos síncronos terminam após um certo número de pulsos.
- Para alguns algoritmos assíncronos regulares e bem estruturados não há problema na detecção de que não existem mais mensagens esperadas.
- Por exemplo, o Algoritmo de Propagação de Informação.

Detecção de Terminação

- Os algoritmos assíncronos que não apresentam a regularidade que permite tal análise simples de terminação, utilizam a detecção de terminação global.
- Um algoritmo assíncrono terminou globalmente quando todos os nós estão ociosos e todas as arestas estão vazias no estado global.

Algoritmo Detect_Termination

- **Variáveis**

  $suspects_i$;

  $edge\_state_i^j$:={} para todo n_j;

  $recorded_i$:=$false$;

  $received_i^j$:=false para todo n_j;

  max_tag_i:=0;

  $terminated_i$:=$false$;

Algoritmo Detect_Termination

- **Algoritmo**

Input:

msg_i=**nil**;

Ação se $n_i \in N_0$:

Execute alguma computação;

Envie *comp_msg* em cada aresta de um subconjunto de Out_i;

Algoritmo Detect_Termination

- **Algoritmo**

if $suspects_i$ **then**
  begin
    $max\_tag_i := max\_tag_i + 1$;
    $recorded_i := true$;
    Envie *marker*(max_tag_i) para todo n_j;
  end

Algoritmo Detect_Termination

- **Algoritmo**

Input:

msg_i=$comp\_msg$ tal que $origem_i(msg_i)=(n_j \rightarrow n_i)$;

Ação:

Execute alguma computação;

Envie *comp_msg* em cada aresta de um subconjunto de Out_i;

if $recorded_i$ **then**

if not $received_i^j$ **then**

$edge\_state_i^j := edge\_state_i^j \cup \{msg_i\}$;

Algoritmo Detect_Termination

- **Algoritmo**

if *suspects*$_i$ **then**

begin

$edge\_state_i^k$:={} para todo n_k;

$received_i^k$:= *false* para todo n_k;

max_tag_i:= max_tag_i + 1;

$recorded_i$:= true;

Envie *marker*(max_tag_i) para todo n_j;

end

Algoritmo Detect_Termination

- **Algoritmo**

Input:

msg$_i$=marker(t) tal que $origem_i(msg_i)=(n_j \rightarrow n_i)$;

Ação:

if $t=max\_tag_i$ **then**

$received_i^j := true$;

if $t>max\_tag_i$ **then**

begin

$max\_tag_i:=t$;

$edge\_state_i^k := \{\}$ para todo n_k;

$recorded_i = false$;

$received_i^k := false$ para todo n_k;

Algoritmo Detect_Termination

- **Algoritmo**

if $suspects_i$ **then**
  begin
    $received_i^j := true$;
    $recorded_i := true$;
    Envie $marker(max\_tag_i)$ para todo n_j;
  end
end
If $received_i^k$ para todo n_k **then**
  Envie $edge\_state_i^k$ para todo n_k, juntamente com max_tag_i para n_1;

Algoritmo Detect_Termination

- **Algoritmo**

Input:

msgi=*terminate*;

Ação se $n_i \neq n_1$:

terminated$_i$:=*true*;

Algoritmo Detect_Termination_D

- **Terminação por Difusão**
- **Variáveis**

$expected_i := 0;$

$parent_i :=$ **nil**;

$terminated_i := false;$

Algoritmo Detect_Termination_D

- **Algoritmo**

Input:

msg_i=**nil**;

Ação se $n_i \in N_0$:

Execute alguma computação;

Envie *comp_msg* em cada aresta de um subconjunto de Inc_i;

Algoritmo Detect_Termination_D

- **Algoritmo**

Input:

$msg_i = comp\_msg$ tal que $origem(msg_i) = n_j$;

Ação:

if $expected_i > 0$ **then**

begin

Envie *ack* para n_j;

Execute alguma computação;

Envie *comp_msg* em cada aresta de um subconjunto de Inc_i;

end

Algoritmo Detect_Termination_D

- **Algoritmo**

else

begin

Execute alguma computação;

Envie *comp_msg* em cada aresta de um subconjunto de Inc_i;

if $expected_i > 0$ **then**

$parent_i := n_j$;

Else

Envie *ack* para n_j;

end

Algoritmo Detect_Termination_D

- **Algoritmo**

Input:

msg_i=*ack*;

Ação:

$expected_i$:= $expected_i$ -1;

if $expected_i$ = 0 **then**

if $parent_i \neq$ nil **then**

Envie *ack* para $parent_i$;

Algoritmo Detect_Termination_D

- **Algoritmo**

Input:

$msg_i = terminate$;

Ação se $n_i \neq n_1$:

$terminated_i := true$;

Teorema

Teorema:

Todo estado global no qual n_1 tem terminado localmente, é um estado global no qual a terminação global ocorre.

Prova:

Se n_1 terminou localmente, então todo nó deve ter enviado um número finito de mensagens de computação.

Como estas mensagens de computação e os correspondentes *acks* foram recebidos, o valor de $expected_i$ para o nó n_i, inicialmente igual a zero, se tornou positivo e zero, diversas vezes.

Teorema

Prova (cont):

Sempre que uma transição ocorresse no valor de *expected*$_i$ de zero para uma valor positivo, *parent*$_i$ seria atualizado para apontar para o nó que enviou a mensagem de computação correspondente.

Considere os estados do sistema nos quais todos os nós n_i estão:

- ou em um estado positivo de *expected*$_i$, seguindo a última transição de zero para este valor, se já enviou uma mensagem de computação durante a difusão,
- ou em qualquer outro estado, caso contrário.

Teorema

Prova (cont):

Claramente, no mínimo um destes estados de sistema é um estado global, como por exemplo, aquele no qual todo nó que já enviou mensagens de computação está no seu estado que imediatamente precede a recepção do último *ack*.

Neste estado global, somente há *acks* sobres as arestas, nenhum deles enviado como consequência da recepção de um último *ack*.

Consideremos um destes estados globais.

Teorema

Prova (cont):

Neste estado global, as variáveis *parent*$_i$ para $n_i \neq n_1$ induzem uma árvore que se espalha por todos os nós em G correspondentes aos nós que enviaram no mínimo uma mensagens de computação durante a computação por difusão.

Esta árvore muda dinamicamente com a execução do algoritmos, e *parent*$_i$ pode apontar para diversos vizinhos n_j.

A raiz dessa árvore é n_1 e suas folhas correspondem a aqueles nós dos quais nenhum outro nó n_i recebeu a mensagem de computação que disparou a última transição de zero para um valor positivo de *expected*$_i$.

Teorema

Prova (cont):

A prova é por indução na árvore.

Pela indução, a prova a ser feita é de que todo estado global no qual a raiz da sub-árvore tem terminado localmente é um estado global no qual todo outro nó na sub-árvore tem também terminado localmente.

A base da indução é dada pelas sub-árvores com raízes nas folhas.

Como nenhuma folha n_i é tal que $n_i = parent_j$ para algum nó n_j, então a afirmação vale.

Teorema

Prova (cont):

Como hipótese de indução, assuma esta afirmação para todas as sub-árvores com raízes em n_j tal que $parent_j = n_1$.

Então, n_1 recebe $expected_1$ *acks*, no momento em que terminou localmente, e pela hipótese de indução, também todos os outros nós.