DISTRIBUTED ALGORITHMS 2014

SEMINAR

Anastasija Efremovska and István Haller

Vrije Universiteit Amsterdam

December 4, 2014
In Safra’s algorithm, certain messages do not need to color the receiver black. Only messages that are sent after a token visits the sender and that are received before this same token visits the receiver have to be taken into account. Propose an optimization of Safra’s algorithm based on this observation.
Safra’s algorithm detects termination using token passing

Termination: All processes hit by the token are passive

Messages in channels are monitored using #received and #sent values at each process

Terminology: in front of token - processes not yet reached by token, behind token - processes already traversed by current token

A message sent from behind the token to in front of the token can repair inconsistencies in message count, while processes behind the token got woken up, need to color receiving process black
Exercise 2

Explain in detail how the Dijkstra-Scholten algorithm detects termination in the Chandy-Misra algorithm.
Chandy-Misra computes minimum path towards given node
Dijkstra-Sholten detections termination with given initiator
Both use sink-tree structure
Membership in Dijkstra-Sholten based on first message to wake up process
Membership in Chandy-Misra based on minimum distance
Dijkstra-Sholten tree shrinks when processes go passive
Chandy-Misra tree gets reorganized, but never shrinks
Let $N$ range over the natural numbers. Generalize Example 8.1 to a network with $2N + 1$ nodes and $3N$ weighted edges, for which the number of messages sent by the Chandy-Misra algorithm in the worst case grows exponentially (in $N$). Explain why this is the case.
Chandy-Misra algorithm triggers broadcast for every improvement in distance

Every improvement in one part of the network can trigger a whole new set of messages in the connected elements

Use induction to build up network from simplest 3 node example
Exercise 4

Give an acyclic orientation cover $G_1, G_2$ of a set of paths in the graph below that contains for each pair of nodes $u, v$ a minimum-hop path from $u$ to $v$. Describe how the buffers are linked in the corresponding acyclic orientation cover controller.
Need to build 2 directed graphs, each of which is acyclic
Routing using the edges of $G_1$ first, then $G_2$
Once an edge from $G_2$ taken, can never go back to $G_1$
Each process has a buffer for each $G_i$
Buffers belonging to the same $G_i$ are connected via the edges in $G_i$
Buffers belonging to $G_i$ are connected to buffers belonging to $G_{i+1}$ via edges in $G_{i+1}$ (going to the next level)
Routing in mesh has 4 directions: left, right, up, down
Routing only requires one horizontal and one vertical direction!
Consider a ring with FIFO channels of size $p$, where $p$ is an odd prime number. Determine the exact chance that the Itai-Rodeh ring size algorithm computes a ring size of $p$. 
Itai-Rodeh algorithm computes ring size by trying different estimates starting from 2.

At each "estimate round" select random identifier for each process from a range of $1 \ldots R$.

Algorithm stops: if all pairs of processes $E$ apart from each other have the same identifier, $E$ is the network size.

Example: $N = 6$, $E = 2$, $ID_0 = ID_2 = ID_4$ and $ID_1 = ID_3 = ID_5$, algorithm thinks that size is 2.

What does it mean that $N$ is odd prime?