Asynchronous ASMs

Exclusive Use of Shared Resources (Dining Philosophers Problem)

Egon Börger

Dipartimento di Informatica, Universita di Pisa
http://www.di.unipi.it/~boerger

Exclusive use of shared resources: problem statement

- Goal: Design a distributed algorithm that
  - allows every interested agent to eventually use some shared resource in exclusive use
  - prevents two users to use the shared resource simultaneously
- Agent: set of agents each equipped with
  - resource: Resource (different agents may have the same resource)
- owner: Resource $\rightarrow$ Agent recording the current user of a given resource
- The algorithm has to manipulate updates of owner in such a way that every attempt by an agent to become owner of his resource will eventually succeed

Exclusive use of shared resources: single agent view

- The algorithm has to manipulate updates of owner in such a way that every attempt by an agent to become owner of his resource will eventually succeed
- Every single agent would like to execute alternately the following two rules:

if owner(resource) = none
then owner(resource):= self

if owner(resource) = self
then owner(resource):= none

Conflicts in using resource shared among different agents are risolved by defining a partial order among possibly conflicting moves in a distributed (appropriately initialized) run of an async ASM. Realizing such an order reflects appropriate scheduling and priority policies.

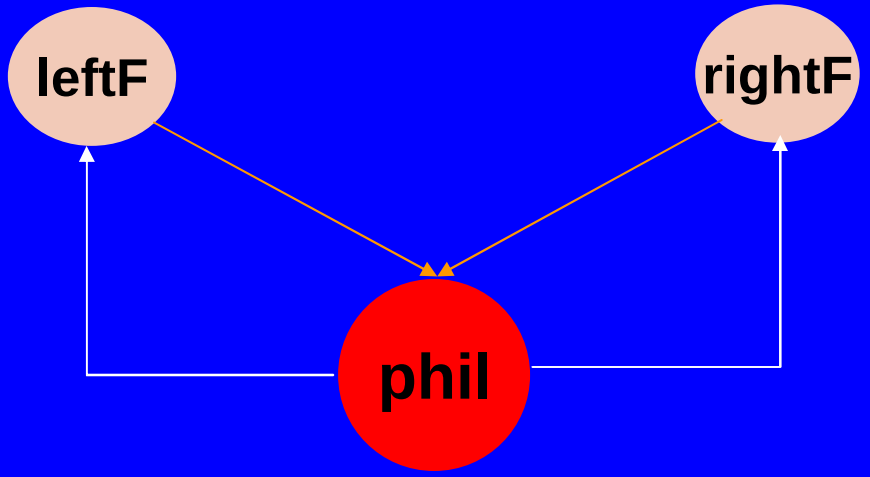
Dining Philosophers
illustrating exclusive use of shared resources

Shared resources represented as “forks” used by “philosophers”

Resource assigning function F: depends on the neighbourhood in the underlying graph

e.g. each agent may have one left and one right neighbour sharing the resource:

Fork = (leftF,rightF) with elementwise defined Owner



if owner(leftF) = owner(rightF) = none
then owner(leftF):= self
owner(rightF):= self

if owner(leftF) = owner(rightF) = self
then owner(leftF):= none
owner(rightF):= none

Reference

- W.Reisig: Elements of Distributed Algorithms
  Springer-Verlag 1998
  - See Section 10.
- E. Börger, R. Stärk: Abstract State Machines. A Method for High-Level System Design and Analysis Springer-Verlag 2003, see http://www.di.unipi.it/AsmBook
  - See Chapter 6.1