Model Checking of Parameterized Systems on Weak Memory

David Declerck

Laboratoire de Recherche en Informatique

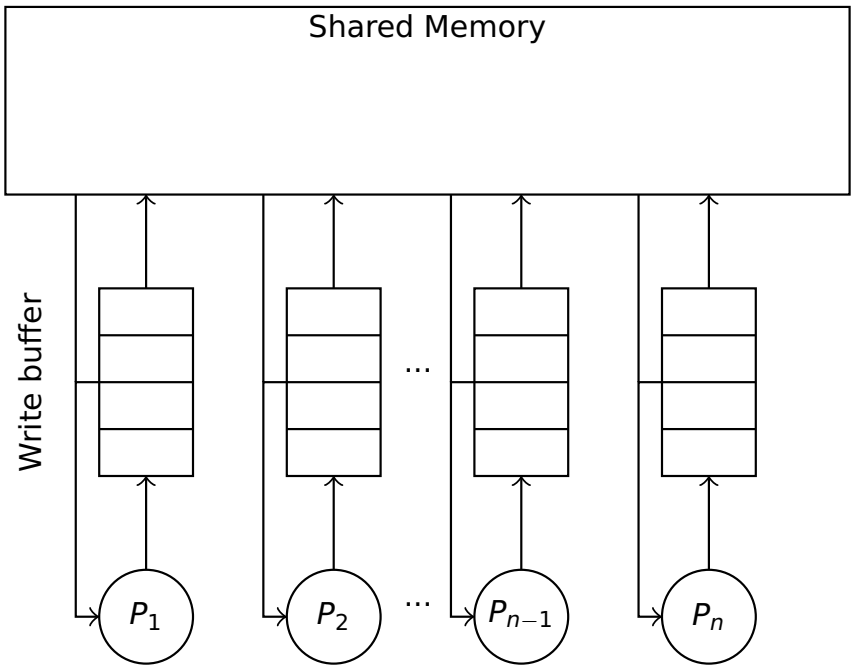
Université Paris-Sud

October 3rd, 2017

Work supported by French
ANR project PARDI (DS0703)

Weak Memory

- order of memory access
  $\neq$ interleaving of memory instructions
- we choose a TSO-like model
- reorderings can be prevented using fences
- harder to reason about concurrent programs



Parameterized Systems

- concurrent systems
- unbounded number of processes
- unbounded process-indexed arrays

Example : naive mutual exclusion

```
type loc = Idle | Want | Crit
array PC[proc] : loc
weak array X[proc] : bool

init (p) { PC[p] = Idle
        && X[p] = False }

unsafe (p1 p2) { PC[p1] = Crit
              && PC[p2] = Crit }
```

```
transition t_req ([p])
requires { PC[p] = Idle }
{ PC[p] := Want; X[p] := True }

transition t_enter ([p])
requires { PC[p] = Want && fence(p)
      && forall_other p. X[p] = False }
{ PC[p] := Crit }
```

Our approach

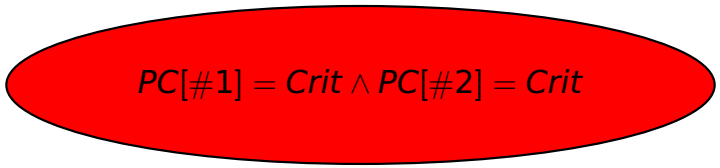
Base framework :

- Model Checking Modulo Theories
- check safety properties of parameterized systems
- assumes a sequentially consistent memory
- relies on a backward reachability algorithm

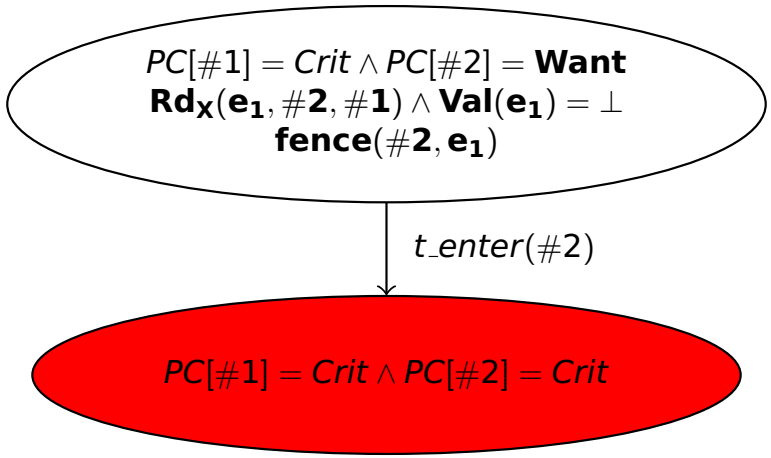
Our extension :

- add TSO reasoning using an axiomatic model
- maps memory instructions to read/write events
- builds a *global happens-before* relation over events

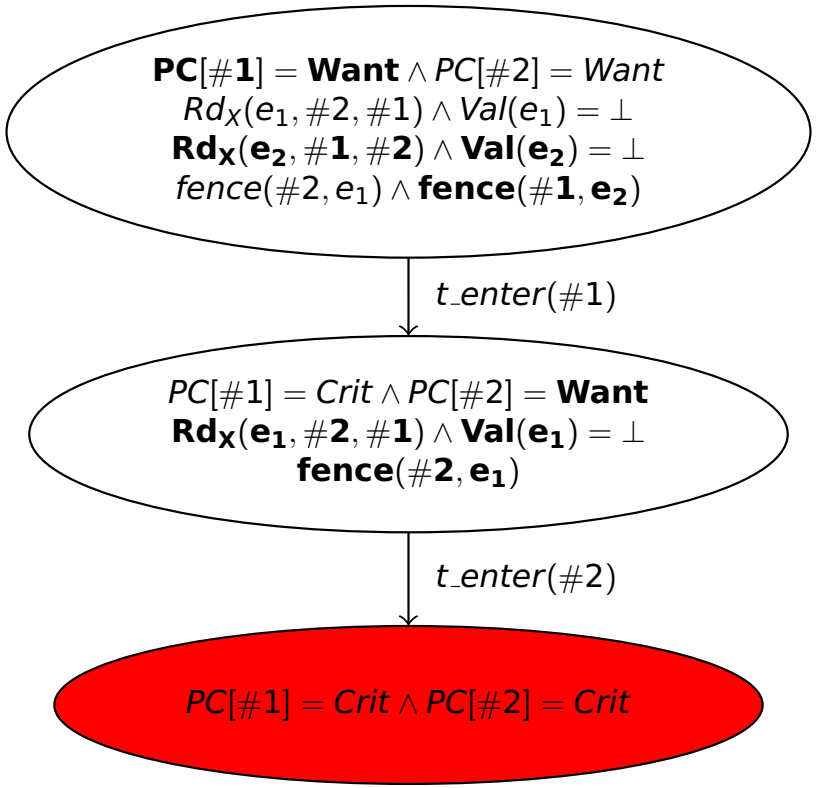
Backward Reachability Example



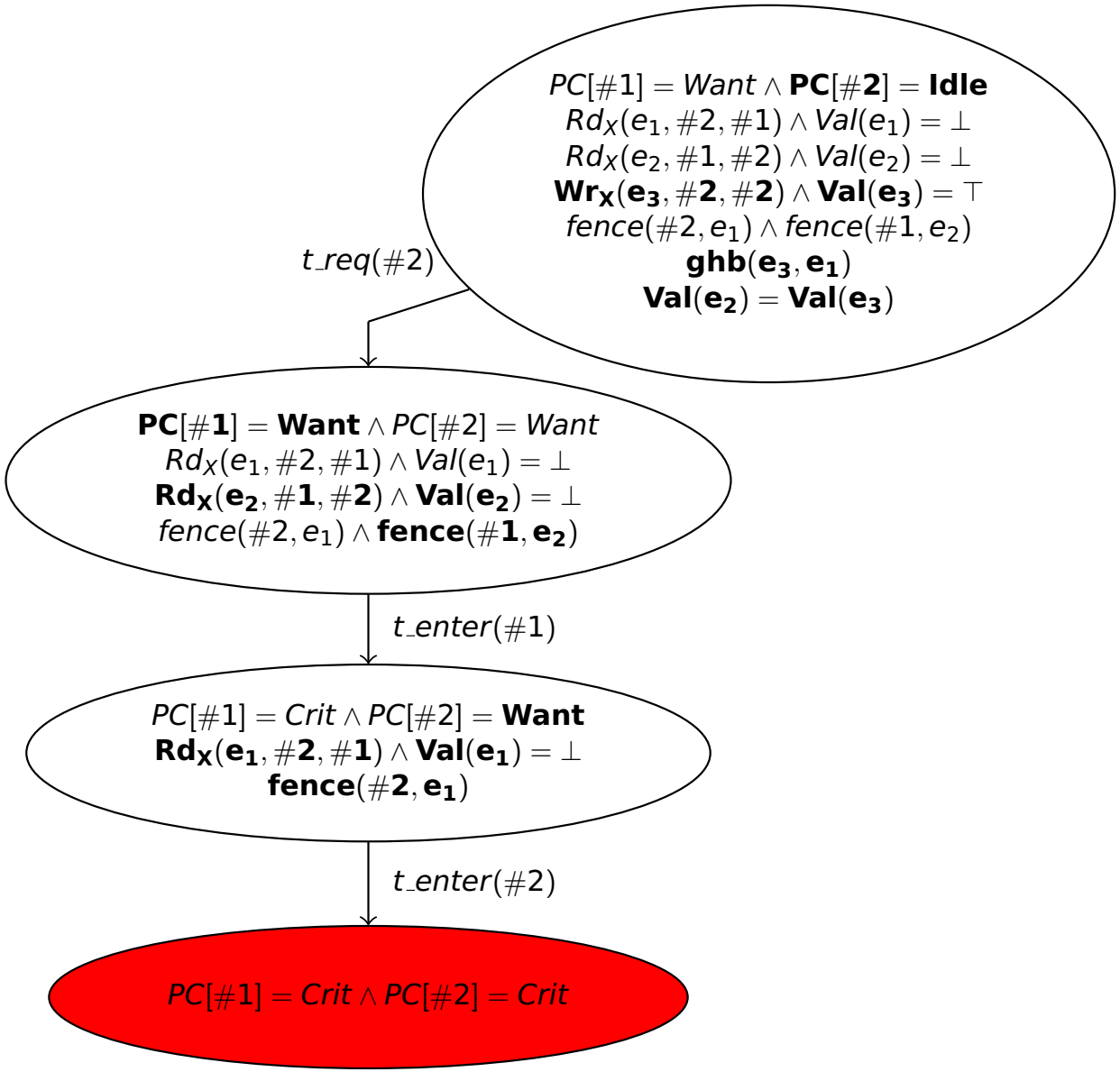
Backward Reachability Example



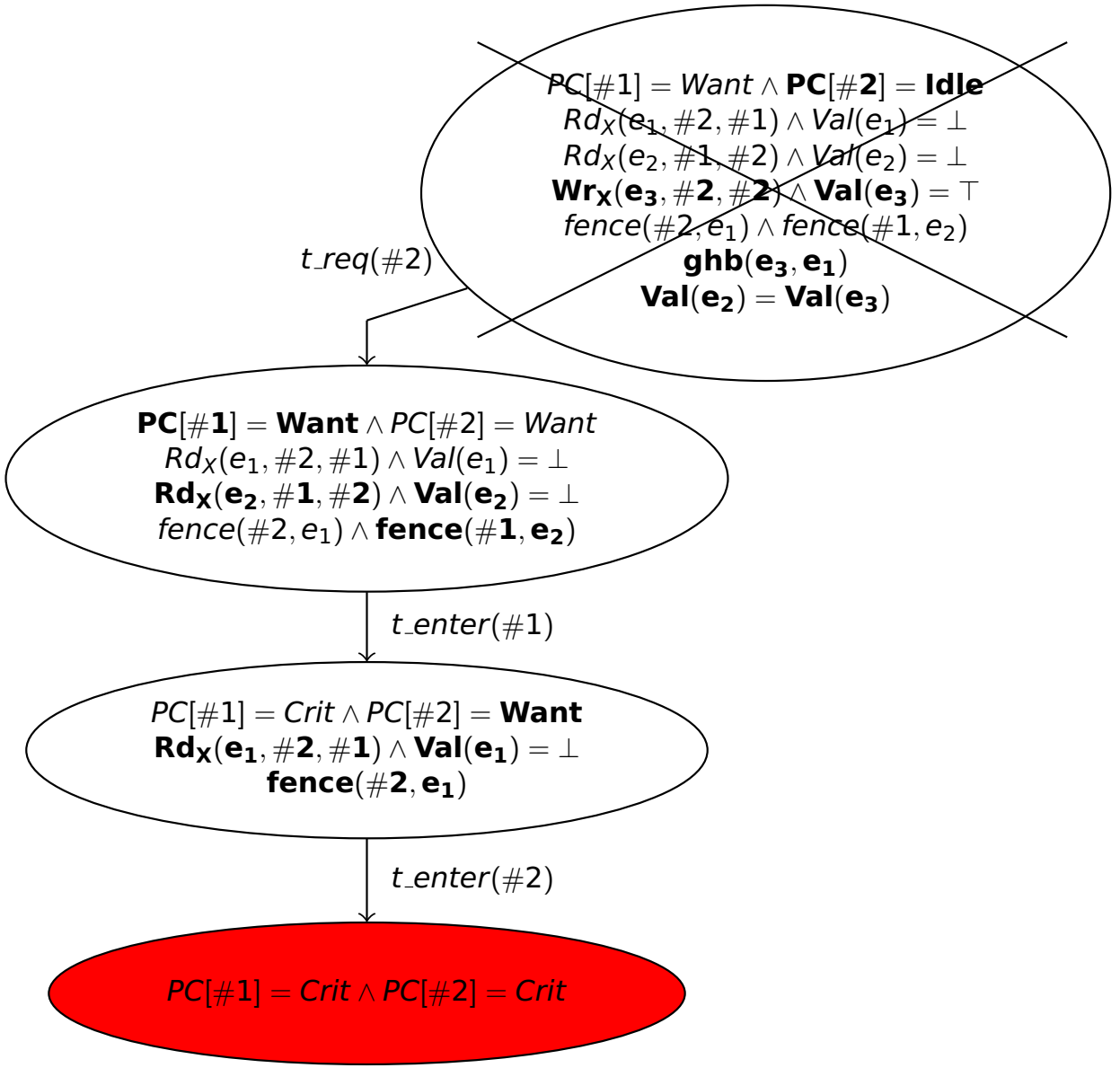
Backward Reachability Example



Backward Reachability Example



Backward Reachability Example



Backward Reachability Example