# QLOCK

Daniel Gaina

JAIST
Research Center for Software Verification

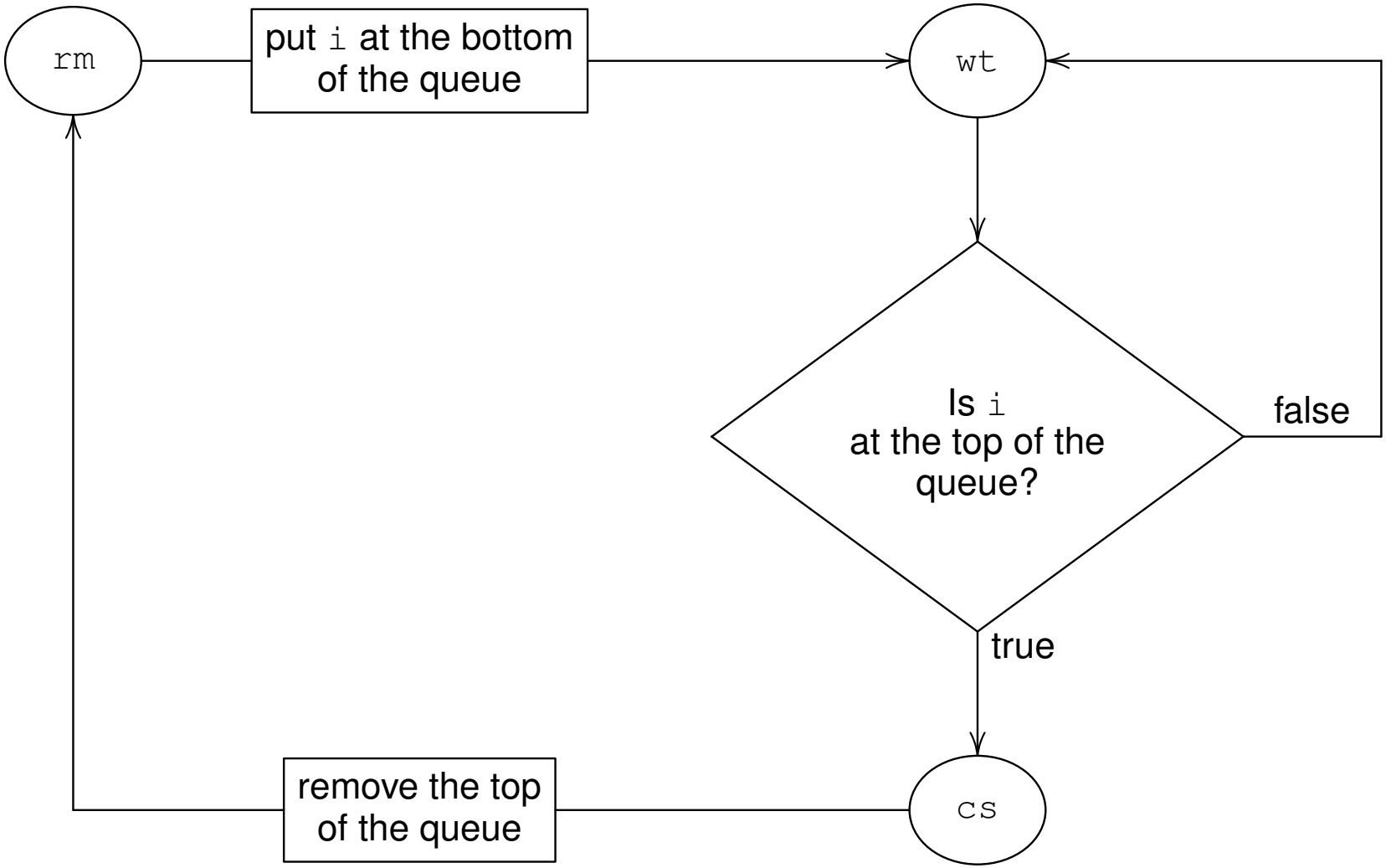
January 8, 2014

# Description

- Initially, each process `i` is in the `reminder section (rm)` .
- After process `i` puts its name at the bottom of a waiting queue, `i` is in the `waiting section (wt)`.
- Process `i` will be in the critical section and have access to the information when it will be the first in the queue.
- When it leaves the source process `i` will be removed from the queue entering again in the remainder section.

**Property to show**

At most one agent is in the critical section (or at the label `cs`) at any moment.


rm
put i at the bottom of the queue
wt
Is i at the top of the queue?
false
true
remove the top of the queue
cs

# Properties to prove

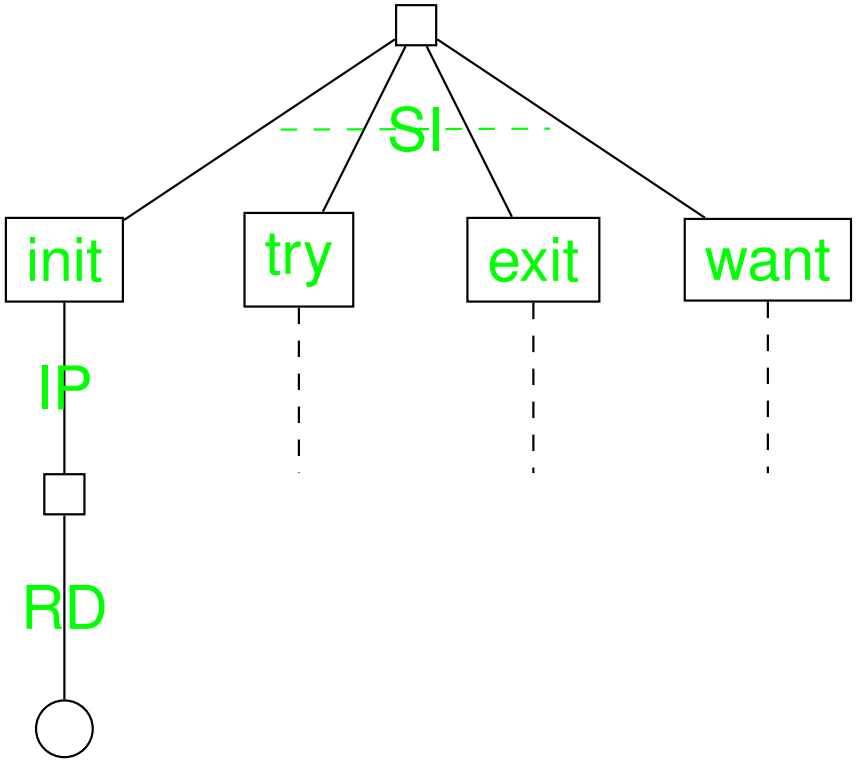
See `http://www.jaist.ac.jp/~danielmg/examples/qlock/qlock.maude` for the specification QLOCK.

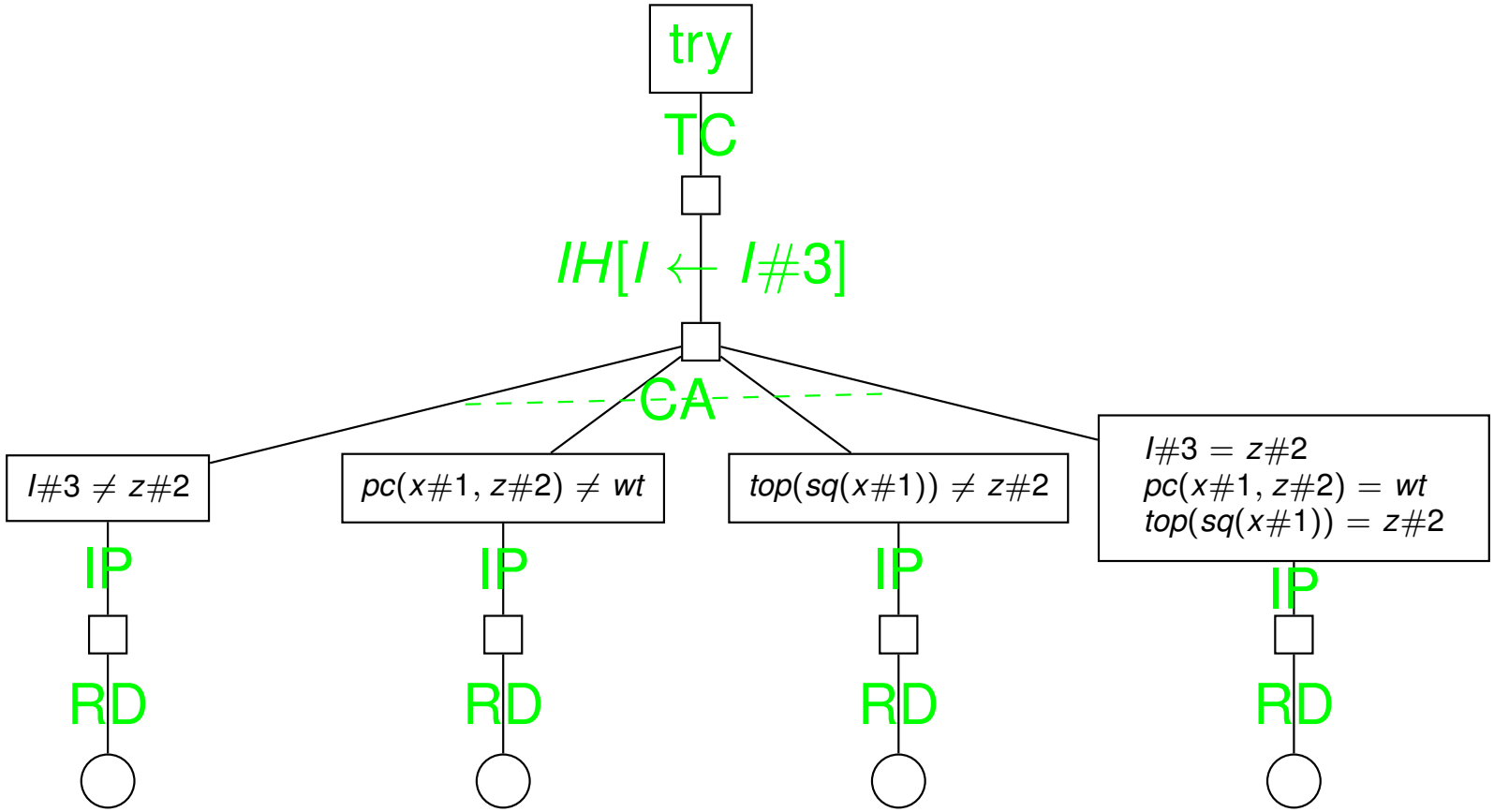
See `http://www.jaist.ac.jp/~danielmg/examples/qlock/proofs.html` for the CITP proofs of the following properties:

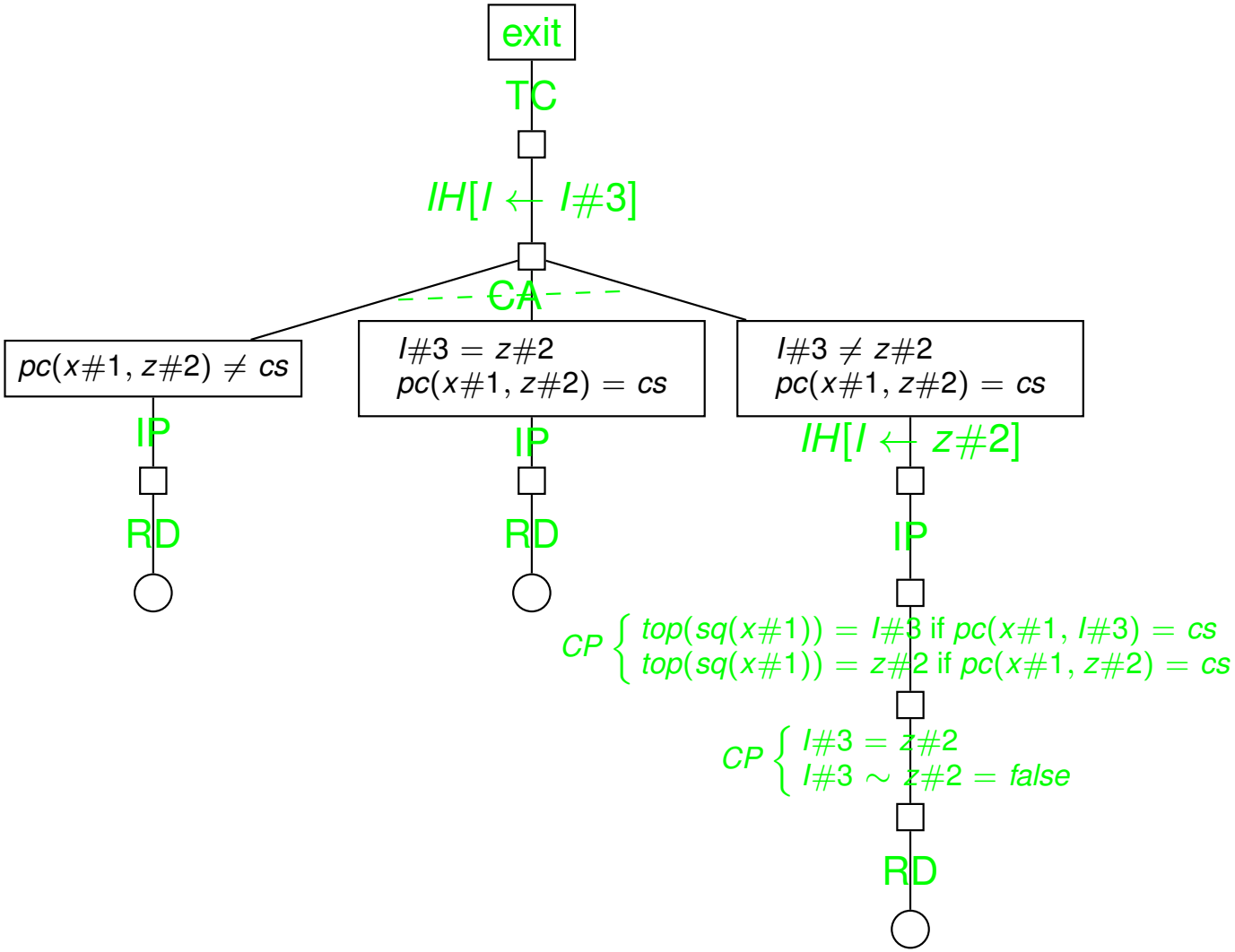
qinv:= `ceq top(sq(S:Sys)) = I:X$Pid if pc(S:Sys,I:X$Pid) = cs .`

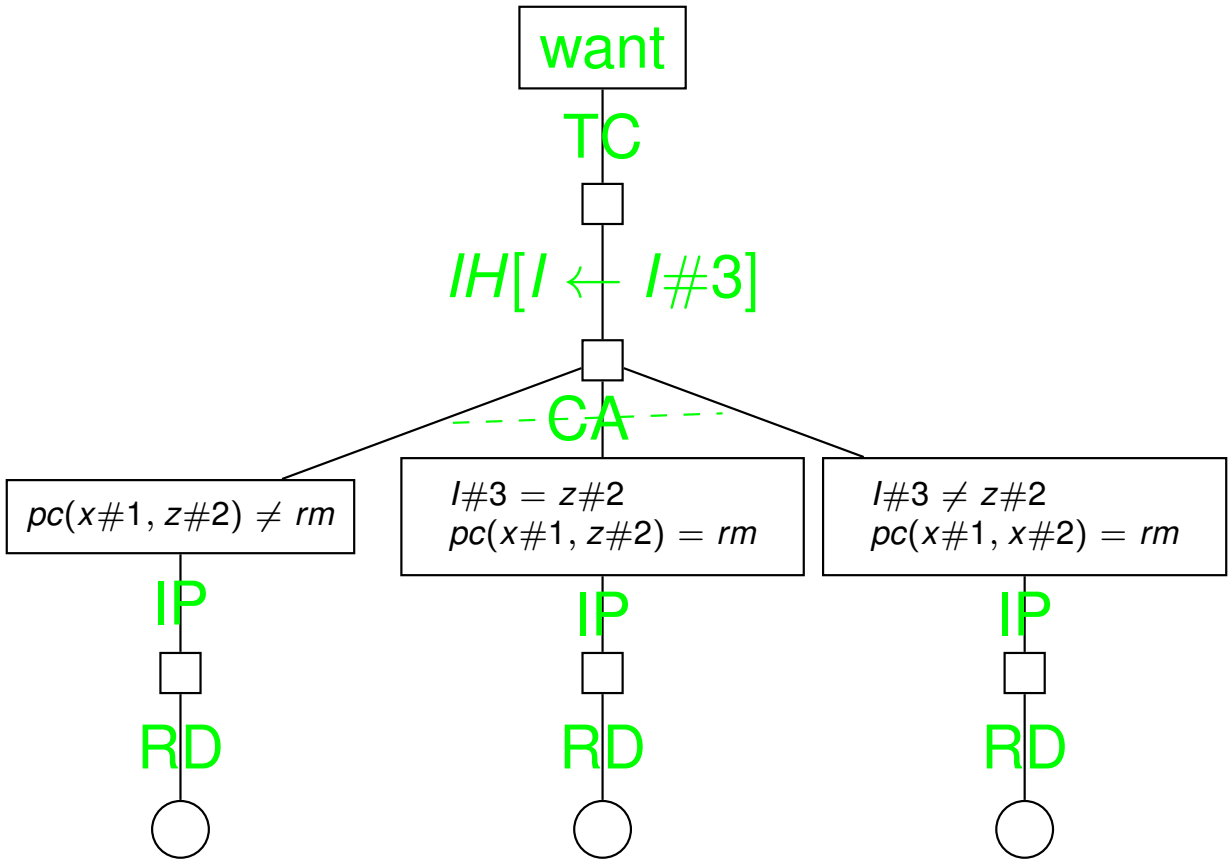
qinv:= `ceq I:X$Pid = J:X$Pid if pc(S:Sys,I:X$Pid) = cs` $\bigwedge$
`pc(S:Sys,J:X$Pid) = cs .`

# QLOCK |- qinv

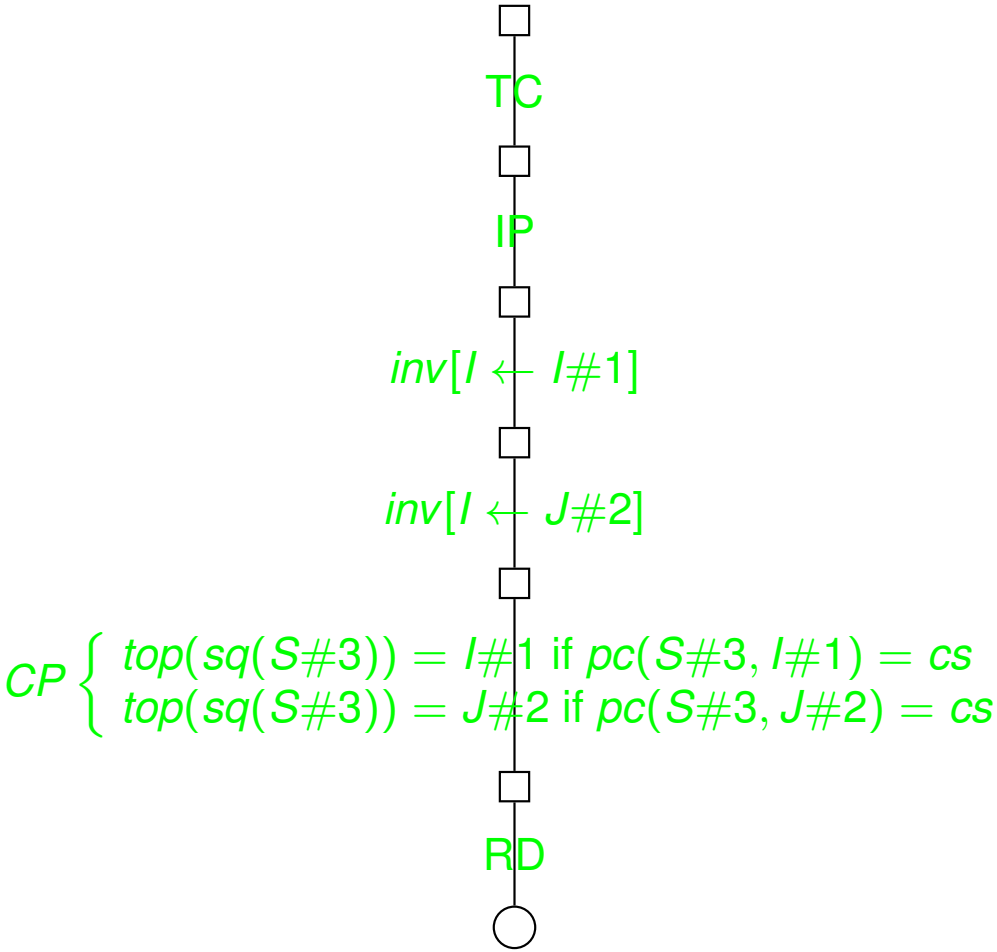



try
TC
$IH[I \leftarrow I\#3]$
CA
$I\#3 \neq z\#2$
$pc(x\#1, z\#2) \neq wt$
$top(sq(x\#1)) \neq z\#2$
$I\#3 = z\#2$
$pc(x\#1, z\#2) = wt$
$top(sq(x\#1)) = z\#2$
IP
IP
IP
IP
RD
RD
RD
RD


exit
TC
IH[I ← I#3]
CA
pc(x#1, z#2) ≠ cs
I#3 = z#2
pc(x#1, z#2) = cs
I#3 ≠ z#2
pc(x#1, z#2) = cs
IP
RD
IP
RD
IH[I ← z#2]
IP
CP { top(sq(x#1)) = I#3 if pc(x#1, I#3) = cs
CP { top(sq(x#1)) = z#2 if pc(x#1, z#2) = cs
CP { I#3 = z#2
I#3 ∼ z#2 = false
RD


want
TC
IH[I ← I#3]
CA
pc(x#1, z#2) ≠ rm
I#3 = z#2
pc(x#1, z#2) = rm
I#3 ≠ z#2
pc(x#1, x#2) = rm
IP
IP
IP
RD
RD
RD

# QLOCK |- qgoal


TC
IP
$inv[I \leftarrow I\#1]$
$inv[I \leftarrow J\#2]$
$CP \begin{cases} top(sq(S\#3)) = I\#1 \text{ if } pc(S\#3, I\#1) = cs \\ top(sq(S\#3)) = J\#2 \text{ if } pc(S\#3, J\#2) = cs \end{cases}$
RD