The proof

```
  |-------
{1}   is_P_reachable_valid(G(reachable)), pfs)

Rule? (BINV "reachable")
```

Rule BINV

I1. $\Theta \rightarrow \varphi$

I2. $\varphi(V) \wedge \rho(V, V') \rightarrow \varphi(V')$

$G\varphi$

```
 ;;;The inductive step of the BINV rule
{-1,(rho)}
     loc(current!1)(p!1) = 0 AND y(next!1) = y(current!1) AND
     loc(next!1) = loc(current!1) WITH [(p!1) := 1]
  OR loc(current!1)(p!1) = 1 AND y(next!1) = y(current!1) AND
     loc(next!1) = loc(current!1) WITH [(p!1) := 2]
  OR loc(current!1)(p!1) = 2 AND y(current!1) = 1 AND
     y(next!1)=0 AND loc(next!1) = loc(current!1) WITH [(p!1) := 3]
  OR loc(current!1)(p!1) = 3 AND y(next!1) = y(current!1) AND
     loc(next!1) = loc(current!1) WITH [(p!1) := 4]
  OR loc(current!1)(p!1) = 4 AND y(next!1) = 1 AND
     loc(next!1) = loc(current!1) WITH [(p!1) := 0]
{-2,(reachable invariant)}
    FORALL (i: PROC_ID): FORALL (j: PROC_ID):
     (loc(current!1)(i) > 2 IMPLIES y(current!1) = 0) AND
     (i = j OR loc(current!1)(i) < 3 OR loc(current!1)(j) < 3)
  |-------
{1,(rtp)}
      (loc(next!1)(i!1) > 2 IMPLIES y(next!1) = 0) AND
       (i!1 = j!1 OR loc(next!1)(i!1) < 3 OR loc(next!1)(j!1) < 3)
```

```
 Rule? (split-rho)

this yields  5 subgoals:
reachableInv.1 :
{-1,(rho)} loc(current!1)(p!1) = 0
{-2,(rho)} y(next!1) = y(current!1)
{-3,(rho)}
     loc(next!1) = loc(current!1) WITH [(p!1) := 1]
{-4,(reachable invariant)}
     FORALL (i: PROC_ID): FORALL (j: PROC_ID):
      IF loc(current!1)(i) > 2 THEN y(current!1) =0 ELSE TRUE ENDIF
      AND (i = j OR loc(current!1)(i) < 3 OR loc(current!1)(j) < 3)
  |-------
{1,(rtp)}
      IF loc(current!1) WITH [(p!1) := 1](i!1) > 2
         THEN y(current!1) = 0
      ELSE TRUE ENDIF
       AND
       (i!1 = j!1 OR
         loc(current!1) WITH [(p!1) := 1](i!1) < 3 OR
          loc(current!1) WITH [(p!1) := 1](j!1) < 3)



 Rule? (inst - "i!1" "j!1")

[-1,(rho)] loc(current!1)(p!1) = 0
[-2,(rho)] y(next!1) = y(current!1)
[-3,(rho)] loc(next!1) = loc(current!1) WITH [(p!1) := 1]
{-4,(reachable invariant)}
    IF loc(current!1)(i!1)>2 THEN y(current!1)=0 ELSE TRUE ENDIF AND
      (i!1 = j!1 OR loc(current!1)(i!1)<3 OR loc(current!1)(j!1)<3)
  |-------
[1,(rtp)]
      IF loc(current!1) WITH [(p!1) := 1](i!1) > 2
        THEN y(current!1) = 0
      ELSE TRUE
      ENDIF
       AND
       (i!1 = j!1 OR
         loc(current!1) WITH [(p!1) := 1](i!1) < 3 OR
          loc(current!1) WITH [(p!1) := 1](j!1) < 3)

Rule? (grind)
This completes the proof of reachableInv.1.
```

```
reachableInv.5 :

{-1,(rho)} loc(current!1)(p!1) = 4
{-2,(rho)} y(next!1) = 1
{-3,(rho)} loc(next!1) = loc(current!1) WITH [(p!1) := 0]
{-4,(reachable invariant)}
     FORALL (i: PROC_ID): FORALL (j: PROC_ID):
      IF loc(current!1)(i) > 2 THEN y(current!1) = 0 ELSE TRUE ENDIF
      AND (i = j OR loc(current!1)(i) < 3 OR loc(current!1)(j) < 3)
  |-------
{1,(rtp)}
      IF loc(current!1) WITH [(p!1) := 0](i!1) > 2
        THEN FALSE
      ELSE TRUE
      ENDIF
       AND
       (i!1 = j!1 OR
         loc(current!1) WITH [(p!1) := 0](i!1) < 3 OR
          loc(current!1) WITH [(p!1) := 0](j!1) < 3)




Rule? (inst - "i!1" "j!1")

[-1,(rho)]
      loc(current!1)(p!1) = 4
[-2,(rho)]
      y(next!1) = 1
[-3,(rho)]
      loc(next!1) = loc(current!1) WITH [(p!1) := 0]
{-4,(reachable invariant)}
    IF loc(current!1)(i!1)>2 THEN y(current!1)=0 ELSE TRUE ENDIF AND
       (i!1 = j!1 OR loc(current!1)(i!1)<3 OR loc(current!1)(j!1)<3)
  |-------
[1,(rtp)]
      IF loc(current!1) WITH [(p!1) := 0](i!1) > 2
        THEN FALSE
      ELSE TRUE
      ENDIF
       AND
       (i!1 = j!1 OR
         loc(current!1) WITH [(p!1) := 0](i!1) < 3 OR
          loc(current!1) WITH [(p!1) := 0](j!1) < 3)
```

```
 Rule? (grind)
this yields  2 subgoals:
reachableInv.5.1 :

[-1,(rho)]
      loc(current!1)(p!1) = 4
[-2,(rho)]
      y(next!1) = 1
[-3,(rho)]
      loc(next!1) = loc(current!1) WITH [(p!1) := 0]
{-4}  y(current!1) = 0
{-5}  i!1 = j!1
{-6}  loc(current!1)(j!1) > 2
  |-------
{1}   p!1 = j!1

Rule?
```

next!1 violates reachable as $y(\text{next!1}) = 1$ and $\text{loc(next!1)(j!1)} > 2$.

current!1 violates reachable:
$\text{loc(current!1)(p!1)} = 4$ and $\text{loc(current!1)(j!1)} > 2$

```
 Rule? (undo inst)

reachableInv.5 :
{-1,(rho)} loc(current!1)(p!1) = 4
{-2,(rho)} y(next!1) = 1
{-3,(rho)} loc(next!1) = loc(current!1) WITH [(p!1) := 0]
{-4,(reachable invariant)}
     FORALL (i: PROC_ID): FORALL (j: PROC_ID):
      IF loc(current!1)(i) > 2 THEN y(current!1) = 0 ELSE TRUE ENDIF
      AND (i = j OR loc(current!1)(i) < 3 OR loc(current!1)(j) < 3)
  |-------
{1,(rtp)}
      IF loc(current!1) WITH [(p!1) := 0](i!1) > 2
        THEN FALSE
      ELSE TRUE
      ENDIF
       AND
       (i!1 = j!1 OR
         loc(current!1) WITH [(p!1) := 0](i!1) < 3 OR
          loc(current!1) WITH [(p!1) := 0](j!1) < 3)
```

```
Rule? (inst-cp - "i!1" "j!1")

[-1,(rho)] loc(current!1)(p!1) = 4
[-2,(rho)] y(next!1) = 1
[-3,(rho)] loc(next!1) = loc(current!1) WITH [(p!1) := 0]
[-4,(reachable invariant)]
    FORALL (i: PROC_ID): FORALL (j: PROC_ID):
     IF loc(current!1)(i)>2 THEN y(current!1)=0 ELSE TRUE ENDIF
     AND (i = j OR loc(current!1)(i)<3 OR loc(current!1)(j)<3)
{-5,(reachable invariant)}
    IF loc(current!1)(i!1)>2 THEN y(current!1)=0 ELSE TRUE ENDIF AND
    (i!1 = j!1 OR loc(current!1)(i!1)< 3 OR loc(current!1)(j!1) < 3)
  |-------
[1,(rtp)]
      IF loc(current!1) WITH [(p!1) := 0](i!1) > 2
        THEN FALSE ELSE TRUE
      ENDIF
       AND (i!1 = j!1 OR
         loc(current!1) WITH [(p!1) := 0](i!1) < 3 OR
          loc(current!1) WITH [(p!1) := 0](j!1) < 3)




Rule? (grind :if-match nil)

this yields  2 subgoals:
reachableInv.5.1 :

[-1,(rho)]
      loc(current!1)(p!1) = 4
[-2,(rho)]
      y(next!1) = 1
[-3,(rho)]
      loc(next!1) = loc(current!1) WITH [(p!1) := 0]
{-4,(reachable invariant)}
      FORALL (i: PROC_ID):
        FORALL (j: PROC_ID):
          (i = j OR loc(current!1)(i) < 3 OR loc(current!1)(j) < 3)
{-5}  y(current!1) = 0
{-6}  i!1 = j!1
{-7}  loc(current!1)(j!1) > 2
  |-------
{1}   p!1 = j!1
```

```
Rule? (inst - "p!1" "j!1")

[-1,(rho)]
      loc(current!1)(p!1) = 4
[-2,(rho)]
      y(next!1) = 1
[-3,(rho)]
      loc(next!1) = loc(current!1) WITH [(p!1) := 0]
{-4,(reachable invariant)}
      (p!1 = j!1 OR loc(current!1)(p!1) < 3 OR loc(current!1)(j!1) < 3)
[-5]  y(current!1) = 0
[-6]  i!1 = j!1
[-7]  loc(current!1)(j!1) > 2
  |-------
[1]   p!1 = j!1


Rule? (assert)

This completes the proof of reachableInv.5.1.
```