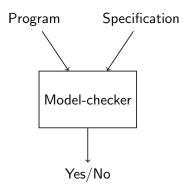
# Software Model-Checking: an algorithmic approach to prove programs correct

Damien Zufferey

IST Austria

May 25, 2011

## Push-button approach



## Running example: Lamport's bakery algorithm

There is bakery with only one clerk. The clerk can serve only one customer at the time. To avoid conflict between customers, a numbering machine gives a ticket to each customer.

#### What customers do:

- Customer enters the bakery and get a ticket;
- If he is alone or has the ticket with lowest number then he orders;
- When he is done, he leaves the bakery and throws away the ticket.

When the bakery is empty the numbering machine is reseted.

### Implementation of the algorithm for 2 customers.

```
initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0 pc values: (0 \rightarrow outside), (1 \rightarrow waiting), (2 \rightarrow ordering)
```

```
while (true) {
     if(pc1 = 0){
3
     x1 = x2 + 1:
     pc1 = 1;
5
     else\ if(pc1 = 1 \&\&
6
7
8
               (x2 = 0 | |
                 x1 < x2 )){
       pc1 = 2;
     else if(pc1 = 2)
10
      pc1 = 0;
11
      x1 = 0:
12
13
     if (pc1==2 \&\& pc2==2){
14
       ERROR:
15
16 }
```

```
while (true) {
     if(pc2 == 0){
      x2 = x1 + 1:
     pc2 = 1;
   else\ if(pc2 = 1 \&\&
               (x1 = 0 | |
                x2 < x1 )){
8
       pc2 = 2;
     else if(pc2 = 2)
10
      pc2 = 0;
      x2 = 0:
11
12
13
     if (pc1==2 \&\& pc2==2){
14
      ERROR:
15
16 }
```

### Assumptions and model

unbounded integers: from  $-\infty$  to  $\infty$ , no overflow.

atomicity: blocks of code are atomic in particular:

```
x1 = x2 + 1;
```

```
if(pc1 == 1 &&
  (x2 == 0 ||
    x1 < x2 )){
  pc1 = 2;
}</pre>
```

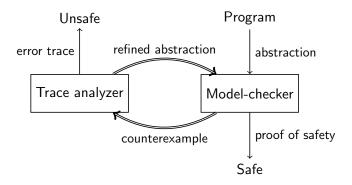
# Why do we means by correct?

reachability: (safety) no two customers fight.

liveness: (starvation-free) every customers eventually get served.

For this talk we will care only about safety property.

### General idea:



CEGAR: counterexample guided abstraction refinement

### First iteration

#### predicates:

#### initial state:

```
while (true)
      if(
2
3
4
5
                              &&
      }else if(
6
7
8
9
                                )){
      } else
                              ){
10
11
12
13
      if(
                   &&
                                ){
14
        ERROR;
15
16 }
```

```
while (true)
     if(
 3
                           &&
     }else if(
                             )){
     }else if(
                          ){
10
11
12
13
     if(
                 &&
                             ){
       ERROR;
14
15
16
```

### First iteration

#### predicates:

#### initial state:

```
while (true) {
     if(
     }else if(
                          &&
                           )){
     }else if(
                         ){
                &&
                           ){
14
       ERROR;
15
16
```

```
while (true)
     if(
3
                           &&
     }else if(
                            )){
     }else if(
10
11
12
13
                 &&
                            ){
       ERROR;
14
15
16
```

### First iteration

predicates:

```
initial state:
```

```
while (true) {
                                       while (true)
                                         if(
     if(
            if ( pc1 = 0, x1 = 0, pc2 = 0, x2 = 0;
     } else
                                                              &&
                 while (true) {
                  if(pc1==2 \&\& pc2==2){
                                                               )){
                    ERROR;
     } else
                &&
                           ){
                                         if(
                                   13
                                                    &&
                                                               ){
14
       ERROR;
                                   14
                                           ERROR;
15
                                   15
16
                                   16
```

# First counterexample

#### SSA formula:

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
  
 $pc_1 = 2 \land pc_2 = 2$ 

# First counterexample

#### SSA formula:

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
  
 $pc_1 = 2 \land pc_2 = 2$ 

Formula is unsat  $\Rightarrow$  spurious counterexample

Let A and B be two formulas such that  $A \wedge B$  unsat.

A [Craig] interpolant I has the following properties:

- I contains only AB-common symbols.
- A implies I
- $I \wedge B$  unsat.

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$

$$pc_1 = 2 \land pc_2 = 2$$

Let A and B be two formulas such that  $A \wedge B$  unsat.

A [Craig] interpolant I has the following properties:

- *I* contains only *AB*-common symbols.
- A implies I
- $I \wedge B$  unsat.

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
 $pc_1 = 2 \land pc_2 = 2$ 

### Second iteration

```
predicates: pc1 = 0
initial state: pc1 = 0
```

```
while (true) {
     if(pc1 = 0){
3
4
       pc1 = 1;
5
     else if(pc1 = 1 \&\&
6
7
8
                           )){
       pc1 = 2:
9
     else\ if(pc1 = 2)
10
       pc1 = 0:
11
12
13
     if (pc1==2 &&
14
       ERROR:
15
16 }
```

```
while (true) {
     if(
                          &&
     }else if(
                            )){
     }else if(
                         ){
10
11
12
13
     if (pc1==2 &&
                            ){
14
       ERROR:
15
16
```

### Second iteration

```
predicates: pc1 = 0
initial state: pc1 = 0
```

```
w.:le (true) {
2
     if(p_1 = 0){
       pc1 = 1:
     else if(pc1 = 1 \&\&
                         )){
       pc1 = 2;
     else\ if(pc1 = 2)
       pc1 = 0:
     if ( pc1==2 &&
14
       ERROR;
15
16
```

```
while (true) {
     if(
                          &&
     }else if(
                            )){
     }else if(
                         ){
10
11
12
13
     if (pc1==2 &&
                            ){
14
       ERROR;
15
16
```

### Second counterexample

```
\begin{array}{lll} & \text{pc1=0, x1=0, pc2=0, x2=0;} \\ & \text{assume(pc1 === 0);} \\ & \text{x1 = x2 + 1;} \\ & \text{pc1 = 1;} \\ & \text{assume(pc1==2 \&\& pc2==2);} \\ & \text{ERROR;} \end{array}
```

#### SSA formula:

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
  
 $pc_1 = 0$   
 $x'_1 = x_2 + 1$   
 $pc'_1 = 1$   
 $pc'_1 = 2 \land pc_2 = 2$ 

### Second counterexample

```
\begin{array}{lll} & \text{pc1=0, x1=0, pc2=0, x2=0;} \\ & \text{assume(pc1 == 0);} \\ & \text{x1 = x2 + 1;} \\ & \text{pc1 = 1;} \\ & \text{assume(pc1==2 \&\& pc2==2);} \\ & \text{ERROR;} \end{array}
```

#### SSA formula:

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
  
 $pc_1 = 0$   
 $x'_1 = x_2 + 1$   
 $pc'_1 = 1$   
 $pc'_1 = 2 \land pc_2 = 2$ 

Formula is unsat  $\Rightarrow$  spurious counterexample

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$

$$pc_1 = 0$$

$$x_1' = x_2 + 1$$

$$pc_{1}' = 1$$

$$pc_1'=2 \land pc_2=2$$

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$

Т

$$pc_1 = 0$$

$$x_1' = x_2 + 1$$

$$pc_{1}' = 1$$

$$pc_1' = 2 \wedge pc_2 = 2$$

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$

Т

$$pc_1 = 0$$

 $\overline{\phantom{a}}$ 

$$x_1' = x_2 + 1$$

$$pc_{1}' = 1$$

$$pc_1'=2 \land pc_2=2$$

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
 $pc_1 = 0$ 
 $x'_1 = x_2 + 1$ 

$$pc_{1}' = 1$$

$$pc_1' = 2 \land pc_2 = 2$$

г

 $\top$ 

Т

$$egin{aligned} egin{aligned} egin{aligned\\ egin{aligned} egi$$

### Third iteration

```
predicates: pc1 = 0, pc1 = 1 initial state: pc1 = 0
```

```
while (true) {
     if(pc1 = 0){
3
4
       pc1 = 1;
5
     else if(pc1 = 1 \&\&
6
7
8
                           )){
       pc1 = 2:
9
     else\ if(pc1 = 2)
10
       pc1 = 0:
11
12
13
     if (pc1==2 &&
14
       ERROR:
15
16|}
```

```
while (true) {
     if(
                          &&
     }else if(
                            )){
     }else if(
                         ){
10
11
12
13
     if (pc1==2 &&
                            ){
14
       ERROR:
15
16
```

### Third iteration

```
 \begin{array}{lll} \mbox{predicates: } \mbox{pc1} = 0, \mbox{ pc1} = 1 \\ \mbox{initial state: } \mbox{pc1} = 0 \end{array}
```

```
while (true) {
  f(pc1 == 0){
  pc1 = 1;
  else if(pc1 = 1 \&\&
                     )){
   pc1 = 2:
 else\ if(pc1 = 2)
   pc1 = 0:
  if (pc1==2 &&
   ERROR;
```

```
while (true) {
     if(
                          &&
     }else if(
                            )){
     }else if(
                         ){
10
11
12
13
     if (pc1==2 &&
                            ){
14
       ERROR:
15
16
```

### Third iteration

```
\begin{array}{ll} \mbox{predicates: } \mbox{pc1} = 0, \mbox{ pc1} = 1 \\ \mbox{initial state: } \mbox{pc1} = 0 \end{array}
```

```
while (true) {
   (pc1 = 0){
   pc1 = 1;
   lse if (pc1 == 1 &&
                      )){
   Ise if (pc1 = 2){
   pc1 = 0;
   ERR
```

```
while (true) {
     if(
     }else if(
                          &&
                           )){
     }else if(
10
11
12
13
     if (pc1==2 &&
                           ){
14
       ERROR;
15
16
```

### Third counterexample

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
 $pc_1 = 0$ 
 $x'_1 = x_2 + 1$ 
 $pc'_1 = 1$ 
 $pc'_1 = 1 \land (x_2 = 0 \lor x'_1 < x_2)$ 
 $pc''_1 = 2$ 
 $pc''_1 = 2 \land pc_2 = 2$ 

### Third counterexample

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
 $pc_1 = 0$ 
 $x'_1 = x_2 + 1$ 
 $pc'_1 = 1$ 
 $pc'_1 = 1 \land (x_2 = 0 \lor x'_1 < x_2)$ 
 $pc''_1 = 2$ 
 $pc''_1 = 2 \land pc_2 = 2$ 

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1 initial state: pc1 = 0, pc2 = 0
```

```
while (true) {
     if(pc1 = 0){
       pc1 = 1:
5
     else if(pc1 = 1 \&\&
6
                          )){
8
       pc1 = 2:
9
     else if(pc1 = 2)
10
       pc1 = 0:
11
12
13
     if (pc1==2 \&\& pc2==2){
14
       ERROR:
15
16|}
```

```
while (true) {
     if(pc2 == 0){
       pc2 = 1;
    else if(pc2 = 1 \&\&
                          )){
       pc2 = 2:
    else if(pc2 = 2)
       pc2 = 0:
10
11
12
13
     if (pc1==2 \&\& pc2==2){
14
       ERROR:
15
16
```

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1 initial state: pc1 = 0, pc2 = 0
```

```
while (true) {
 f(pc1 = 0)
  pc1 = 1:
 else if(pc1 = 1 \&\&
                     )){
   pc1 = 2:
 else if(pc1 = 2)
   pc1 = 0:
 if (pc1==2 \&\& pc2==2){
   ERROR:
```

```
while (true) {
     if(pc2 == 0){
       pc2 = 1;
    else if(pc2 = 1 \&\&
                          )){
       pc2 = 2:
    else if(pc2 = 2)
10
       pc2 = 0:
11
12
13
     if (pc1==2 \&\& pc2==2){
14
       ERROR:
15
16
```

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1 initial state: pc1 = 0, pc2 = 0
```

```
while (true) {
   (pc1 = 0){}
   pc1 = 1:
   lse if (pc1 == 1 &&
                      )){
   lse if(pc1 == 2){
   pc1 = 0:
    (pc1==2 \&\& pc2==2){
   ERROR:
```

```
while (true) {
     if(pc2 == 0){
       pc2 = 1;
    else\ if(pc2 = 1 \&\&
                          )){
       pc2 = 2:
    else if(pc2 = 2)
10
       pc2 = 0:
11
12
13
    if (pc1==2 \&\& pc2==2){
14
       ERROR:
15
16
```

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1 initial state: pc1 = 0, pc2 = 0
```

```
while (true) {
   (pc1 = 0){}
   pc1 = 1:
   lse if (pc1 == 1 &&
                      )){
   Ise if (pc1 = 2){
   pc1 = 0:
    (pc1==2 \&\& pc2==2){
   ERROR:
```

```
while (true) {
    if(pc2 == 0){
      pc2 = 1;
    else\ if(pc2 = 1 \&\&
                         )){
      pc2 = 2:
    else if(pc2 = 2)
10
      pc2 = 0:
11
12
13
    if (pc1==2 \&\& pc2==2){
      ERROR:
```

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1 initial state: pc1 = 0, pc2 = 0
```

```
while (true) {
   (pc1 = 0){}
   pc1 = 1:
   lse if (pc1 == 1 &&
                      )){
   pc1 = 2:
   Ise if (pc1 == 2){
   pc1 = 0;
    (pc1==2 && pc2==2){
   ERROR:
```

```
wille (true) {
      f(pc2 == 0){
       pc2 = 1:
      else if (pc2 = 1 \&\&
                           )){
      pc2 = 2:
      else if (pc2 = 2){
10
       pc2 = 0:
11
12
13
      (pc1==2 \&\& pc2==2){
```

# Finding out why the cex is spurious (first possibility)

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
 $pc_1 = 0 \land x'_1 = x_2 + 1 \land pc'_1 = 1$ 
 $pc'_1 = 1 \land (x_2 = 0 \lor x'_1 < x_2) \land pc''_1 = 2$ 
 $pc_2 = 0 \land x'_2 = x'_1 + 1 \land pc'_2 = 1$ 
 $pc'_2 = 1 \land (x'_1 = 0 \lor x'_2 < x'_1) \land pc''_2 = 2$ 
 $pc''_1 = 2 \land pc''_2 = 2$ 

# Finding out why the cex is spurious (first possibility)

$$pc_{1} = 0 \land x_{1} = 0 \land pc_{2} = 0 \land x_{2} = 0$$

$$pc_{1} = 0 \land x'_{1} = x_{2} + 1 \land pc'_{1} = 1$$

$$pc'_{1} = 1 \land (x_{2} = 0 \lor x'_{1} < x_{2}) \land pc''_{1} = 2$$

$$pc_{2} = 0 \land x'_{2} = x'_{1} + 1 \land pc'_{2} = 1$$

$$pc'_{2} = 1 \land (x'_{1} = 0 \lor x'_{2} < x'_{1}) \land pc''_{2} = 2$$

$$pc''_{1} = 2 \land pc''_{2} = 2$$

# Finding out why the cex is spurious (first possibility)

$$pc_{1} = 0 \land x_{1} = 0 \land pc_{2} = 0 \land x_{2} = 0$$

$$pc_{1} = 0 \land x'_{1} = x_{2} + 1 \land pc'_{1} = 1$$

$$pc'_{1} = 1 \land (x_{2} = 0 \lor x'_{1} < x_{2}) \land pc''_{1} = 2$$

$$pc_{2} = 0 \land x'_{2} = x'_{1} + 1 \land pc'_{2} = 1$$

$$pc'_{2} = 1 \land (x'_{1} = 0 \lor x'_{2} < x'_{1}) \land pc''_{2} = 2$$

$$pc''_{1} = 2 \land pc''_{2} = 2$$

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
 $x_2 = 0$ 
 $pc_1 = 0 \land x_1' = x_2 + 1 \land pc_1' = 1$ 
 $x_1' = 1 \land x_2 = 0$ 
 $pc_1' = 1 \land (x_2 = 0 \lor x_1' < x_2) \land pc_1'' = 2$ 
 $x_1' = 1$ 
 $pc_2 = 0 \land x_2' = x_1' + 1 \land pc_2' = 1$ 
 $pc_2' = 1 \land (x_1' = 0 \lor x_2' < x_1') \land pc_2'' = 2$ 
 $pc_1'' = 2 \land pc_2'' = 2$ 

$$\begin{aligned} \rho c_1 &= 0 \land x_1 = 0 \land \rho c_2 = 0 \land x_2 = 0 \\ \rho c_1 &= 0 \land x_1' = x_2 + 1 \land \rho c_1' = 1 \\ \rho c_1' &= 1 \land (x_2 = 0 \lor x_1' < x_2) \land \rho c_1'' = 2 \\ \rho c_1' &= 1 \land (x_2' = x_1' + 1 \land \rho c_2' = 1 \\ \rho c_2 &= 0 \land x_2' = x_1' + 1 \land \rho c_2' = 1 \\ \rho c_2' &= 1 \land (x_1' = 0 \lor x_2' < x_1') \land \rho c_2'' = 2 \end{aligned}$$

$$\begin{aligned} x_2 &= 0 \\ x_1' &= 1 \land x_2 = 0 \\ x_1' &= 1 \\ x_1' &= 1 \land x_2' = 2 \end{aligned}$$

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
  $x_2 = 0$   $x_2 = 0$   $pc_1 = 0 \land x_1' = x_2 + 1 \land pc_1' = 1$   $x_1' = 1 \land x_2 = 0$   $pc_1' = 1 \land (x_2 = 0 \lor x_1' < x_2) \land pc_1'' = 2$   $x_1' = 1$   $x_1' = 1$   $x_1' = 1 \land x_2' = 2$   $pc_2' = 1 \land (x_1' = 0 \lor x_2' < x_1') \land pc_2'' = 2$   $pc_1'' = 2 \land pc_2'' = 2$ 

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```

```
while (true) {
     if(pc1 = 0){
     x1 = x2 + 1:
       pc1 = 1:
5
     else\ if(pc1 = 1 \&\&
6
7
               (x2 = 0 | |
                 x1 < x2 )){
8
       pc1 = 2:
9
     else if(pc1 = 2)
10
     pc1 = 0;
11
      x1 = 0:
12
13
     if (pc1==2 \&\& pc2==2){
14
       ERROR:
15
16|}
```

```
while (true) {
     if(pc2 == 0){
      x2 = x1 + 1:
       pc2 = 1:
    else\ if(pc2 = 1 \&\&
               (x1 = 0 | |
                x2 < x1 )){
       pc2 = 2:
    else if(pc2 = 2)
      pc2 = 0;
10
11
      x^2 = 0:
12
13
     if (pc1==2 \&\& pc2==2){
14
      ERROR:
15
16
```

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```

```
while (true) {
  f(pc1 = 0)
   x1 = x2 + 1:
  pc1 = 1:
 else\ if(pc1 = 1 \&\&
            (x2 = 0 | |
             x1 < x2 )){
   pc1 = 2:
 else if(pc1 = 2)
   pc1 = 0:
   x1 = 0:
  if (pc1==2 \&\& pc2==2){
   ERROR:
```

```
while (true) {
     if(pc2 == 0){
      x2 = x1 + 1:
       pc2 = 1:
    else\ if(pc2 = 1 \&\&
               (x1 = 0 | |
                x2 < x1 )){
       pc2 = 2:
    else if(pc2 = 2)
10
      pc2 = 0:
11
      x^2 = 0:
12
13
     if (pc1==2 \&\& pc2==2){
14
      ERROR:
15
16
```

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```

```
while (true) {
   (pc1 = 0){}
   x1 = x2 + 1:
   pc1 = 1:
    Ise if (pc1 = 1 \&\&
           (x2 = 0 | |
            x1 < x2 )){
   pc1 = 2:
   Ise if (pc1 = 2){
   pc1 = 0:
   x1 = 0:
   (pc1==2 && pc2==2){
   ERROR:
```

```
while (true) {
    if(pc2 = 0){
    x2 = x1 + 1:
      pc2 = 1:
    else if(pc2 = 1 \&\&
               (x1 = 0 | |
               x2 < x1 )){
      pc2 = 2:
    else if(pc2 = 2)
    pc2 = 0;
10
11
     x^2 = 0:
12
13
    if (pc1==2 && pc2==2){
14
      ERROR:
15
16
```

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```

```
while (true) {
   (pc1 = 0){}
   x1 = x2 + 1:
   pc1 = 1:
   Ise if (pc1 = 1 \&\&
           (x2 = 0 | |
            x1 < x2 )){
   pc1 = 2:
   Ise if (pc1 == 2){
   pc1 = 0:
   x1 = 0:
   (pc1==2 && pc2==2){
   ERROR:
```

```
while (true) {
     if(pc2 == 0){
      x2 = x1 + 1;
       pc2 = 1:
    else\ if(pc2 = 1 \&\&
               (x1 = 0 | |
                x2 < x1 )){
       pc2 = 2:
    else if(pc2 = 2)
10
      pc2 = 0:
      x^2 = 0:
12
13
     if (pc1==2 \&\& pc2==2){
      ERROR:
```

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```

```
while (tille) {
    (pc1 = 0){
        f(pc1 == 1 \&\&
           (x2 = 0 | |
             x1 < x2 )){
        f(pc1 == 2){
   pc1 =
         =2 && pc2==2){
   ERRO
```

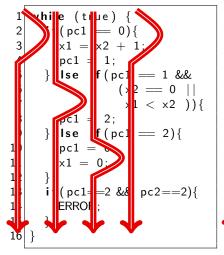
```
while (true) {
    if(pc2 == 0){
      x2 = x1 + 1;
      pc2 = 1:
    else\ if(pc2 = 1 \&\&
               (x1 = 0 | |
                x2 < x1 )){
      pc2 = 2:
    else if(pc2 = 2)
      pc2 = 0:
10
      x^2 = 0:
12
13
    if (pc1==2 \&\& pc2==2){
      ERROR:
```

```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```

```
pc1
            < x2 )){}
pc1
      2 && pc2==2){
ERRO
```

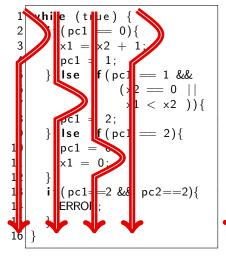
```
while (true) {
    if(pc2 == 0){
      x2 = x1 + 1:
      pc2 = 1;
    else\ if(pc2 = 1 \&\&
               (x1 = 0 | |
                x2 < x1 )){
      pc2 = 2:
    else if(pc2 = 2)
      pc2 = 0:
10
      x^2 = 0:
12
13
    if (pc1==2 \&\& pc2==2){
      ERROR:
```

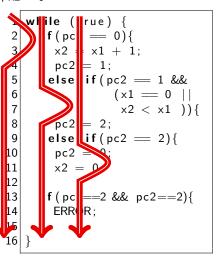
```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```



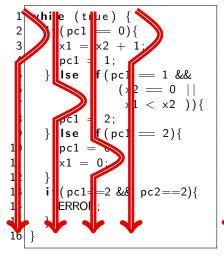
```
wllile (true) {
     f(pc2 == 0){
       x2 = x1 + 1;
       pc2 = 1;
      else if (pc2 = 1 &&
                (x1 = 0 | |
                x2 < x1 )){
      pc2 = 2;
      else if (pc2 = 2){
10
      pc2 = 0;
11
      x2 = 0:
12
13
14
      f(pc1==2 \&\& pc2==2){
       ERROR:
```

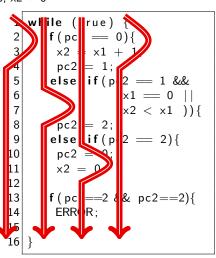
```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```



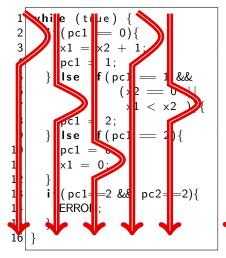


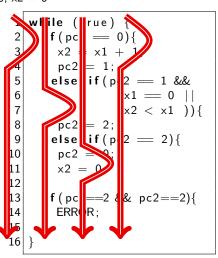
```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```



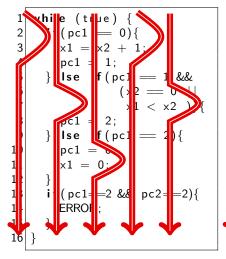


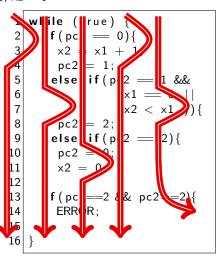
```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```





```
predicates: pc1 = 0, pc1 = 1, pc2 = 0, pc2 = 1, x1=1, x2=0, x2=2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```





$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
 $pc_1 = 0 \land x'_1 = x_2 + 1 \land pc'_1 = 1$ 
 $pc'_1 = 1 \land (x_2 = 0 \lor x'_1 < x_2) \land pc''_1 = 2$ 
 $pc_2 = 0 \land x'_2 = x'_1 + 1 \land pc'_2 = 1$ 
 $pc'_2 = 1 \land (x'_1 = 0 \lor x'_2 < x'_1) \land pc''_2 = 2$ 
 $pc''_1 = 2 \land pc''_2 = 2$ 

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$
 $pc_1 = 0 \land x_1' = x_2 + 1 \land pc_1' = 1$ 
 $pc_1' = 1 \land (x_2 = 0 \lor x_1' < x_2) \land pc_1'' = 2$ 
 $pc_2 = 0 \land x_2' = x_1' + 1 \land pc_2' = 1$ 
 $pc_2' = 1 \land (x_1' = 0 \lor x_2' < x_1') \land pc_2'' = 2$ 
 $pc_1'' = 2 \land pc_2'' = 2$ 

$$pc_{1} = 0 \land x_{1} = 0 \land pc_{2} = 0 \land x_{2} = 0$$

$$pc_{1} = 0 \land x'_{1} = x_{2} + 1 \land pc'_{1} = 1$$

$$x'_{1} > x_{2} \land x_{2} = 0$$

$$pc'_{1} = 1 \land (x_{2} = 0 \lor x'_{1} < x_{2}) \land pc''_{1} = 2$$

$$pc_{2} = 0 \land x'_{2} = x'_{1} + 1 \land pc'_{2} = 1$$

$$pc'_{2} = 1 \land (x'_{1} = 0 \lor x'_{2} < x'_{1}) \land pc''_{2} = 2$$

$$pc''_{1} = 2 \land pc''_{2} = 2$$

$$pc_1 = 0 \land x_1 = 0 \land pc_2 = 0 \land x_2 = 0$$

$$pc_1 = 0 \land x'_1 = x_2 + 1 \land pc'_1 = 1$$

$$pc'_1 = 1 \land (x_2 = 0 \lor x'_1 < x_2) \land pc''_1 = 2$$

$$pc_2 = 0 \land x'_2 = x'_1 + 1 \land pc'_2 = 1$$

$$pc'_2 = 1 \land (x'_1 = 0 \lor x'_2 < x'_1) \land pc''_2 = 2$$

$$pc''_1 = 2 \land pc''_1 = 2$$

$$\begin{aligned} \rho c_1 &= 0 \land x_1 = 0 \land \rho c_2 = 0 \land x_2 = 0 \\ \rho c_1 &= 0 \land x_1' = x_2 + 1 \land \rho c_1' = 1 \\ \rho c_1' &= 1 \land (x_2 = 0 \lor x_1' < x_2) \land \rho c_1'' = 2 \\ \rho c_2 &= 0 \land x_2' = x_1' + 1 \land \rho c_2' = 1 \\ \rho c_2' &= 1 \land (x_1' = 0 \lor x_2' < x_1') \land \rho c_2'' = 2 \end{aligned}$$

$$\begin{aligned} x_2 &= 0 \\ x_1' &> x_2 \land x_2 = 0 \\ x_1' &> 0 \\ x_1' &> 0 \\ x_1' &> 0 \land x_2' > x_1' \end{aligned}$$

$$\begin{array}{c} \rho c_{1} = 0 \wedge x_{1} = 0 \wedge \rho c_{2} = 0 \wedge x_{2} = 0 \\ \\ \rho c_{1} = 0 \wedge x_{1}' = x_{2} + 1 \wedge \rho c_{1}' = 1 \\ \\ \rho c_{1}' = 1 \wedge (x_{2} = 0 \vee x_{1}' < x_{2}) \wedge \rho c_{1}'' = 2 \\ \\ \rho c_{2}' = 0 \wedge x_{2}' = x_{1}' + 1 \wedge \rho c_{2}' = 1 \\ \\ \rho c_{2}' = 1 \wedge (x_{1}' = 0 \vee x_{2}' < x_{1}') \wedge \rho c_{2}'' = 2 \\ \\ \rho c_{1}'' = 2 \wedge \rho c_{2}'' = 2 \end{array}$$

#### Final version

```
predicates: pc1=0, pc1=1, pc2=0, pc2=1, x1=0, x2=0, x1<x2, x1>x2 initial state: pc1 = 0, x1 = 0, pc2 = 0, x2 = 0
```

```
while (true) {
     if(pc1 = 0){
     x1 = x2 + 1:
       pc1 = 1:
5
     else if(pc1 = 1 \&\&
6
7
               (x2 = 0 | |
                x1 < x2 )){
8
       pc1 = 2;
9
     else if(pc1 = 2)
10
     pc1 = 0:
11
      x1 = 0:
12
13
     if (pc1==2 \&\& pc2==2){
14
       ERROR:
15
16|}
```

```
while (true) {
     if(pc2 = 0){
      x2 = x1 + 1:
      pc2 = 1;
    else\ if(pc2 = 1 \&\&
               (x1 = 0 | |
                x2 < x1 )){
8
       pc2 = 2;
    else if(pc2 = 2)
10
      pc2 = 0:
      x^2 = 0:
11
12
13
     if (pc1==2 \&\& pc2==2){
      ERROR:
14
15
16
```

### Building an actual proof

state-space: 2 \* 16 loc and 8 predicates  $\Rightarrow 16^2*2^8 = 65536$  states

### Building an actual proof

state-space: 2 \* 16 loc and 8 predicates  $\Rightarrow$  16<sup>2</sup> \* 2<sup>8</sup> = 65536 states

#### simpler version:

$$\rho c_1 = 1 \land (x_2 = 0 \lor x_1 < x_2) \rightarrow \rho c_1' = 2$$

$$\rho c_1 = 0 \rightarrow \rho c_1' = 1 \land \rho c_1' = 0 \land x_1' = 0$$

$$\rho c_1 = 2 \rightarrow \rho c_1' = 0 \land x_1' = 0$$

$$\rho c_1 = 2 \land \rho c_2 = 2$$

$$\rho c_1 = 2 \land \rho c_2 = 2$$

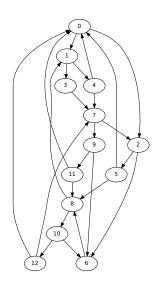
$$\begin{array}{c} pc_{1} = 1 \land (x_{2} = 0 \lor x_{1} < x_{2}) \rightarrow pc'_{1} = 2 \\ pc_{1} = 0 \rightarrow \\ pc'_{1} = 1 \land \\ x'_{1} = x_{2} + 1 \end{array} \qquad \begin{array}{c} pc_{1} = 2 \rightarrow \\ pc'_{1} = 0 \land \\ x''_{1} = 0 \\ pc_{1} = 2 \land pc_{2} = 2 \end{array} \qquad \begin{array}{c} pc_{2} = 1 \land (x_{1} = 0 \lor x_{2} < x_{1}) \rightarrow pc'_{2} = 2 \\ pc_{2} = 0 \rightarrow \\ pc'_{2} = 1 \land \\ x'_{2} = x_{1} + 1 \end{array} \qquad \begin{array}{c} pc_{2} = 2 \rightarrow \\ pc'_{2} = 0 \land \\ x'_{2} = x_{1} + 1 \end{array} \qquad \begin{array}{c} pc_{2} = 2 \rightarrow \\ pc'_{2} = 0 \land \\ x'_{2} = x_{1} + 1 \end{array} \qquad \begin{array}{c} pc_{2} = 2 \rightarrow \\ pc'_{2} = 0 \land \\ x'_{2} = 0 \rightarrow \\ pc_{1} = 2 \land pc_{2} = 2 \end{array} \qquad \begin{array}{c} pc_{2} = 2 \rightarrow \\ pc'_{2} = 1 \land \\ pc'_{2} = 1 \land \\ pc'_{3} = 1 \land \\ pc'_{4} = 0 \land \\ pc'_{5} = 0 \land$$

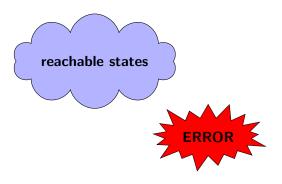
state-space:  $2^2 * 2^8 = 1024$  states

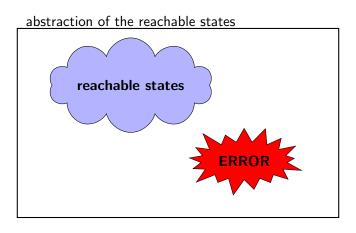
By being clever with the predicates we can go down to 432 states (12 are reachable).

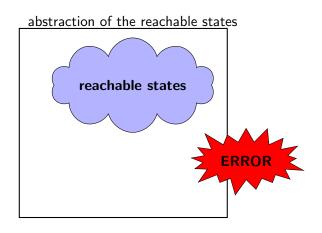
# Reachability graph

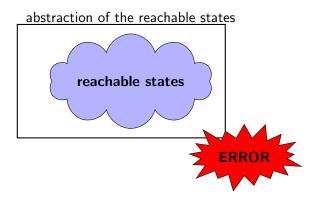
id	pc1	pc2	$x_1 = 0$	$x_2 = 0$	$x_1?x_2$
0	0	0	Т	Т	$x_1 = x_2$
1	1	0	$\perp$	Τ	$x_1 > x_2$
2	0	1	T	$\perp$	$x_1 < x_2$
3	1	1	$\perp$	$\perp$	$x_1 < x_2$
4	2	0	$\perp$	Τ	$x_1 > x_2$
5	0	2	Τ	$\perp$	$x_1 < x_2$
6	1	1	$\perp$	$\perp$	$x_1 > x_2$
7	2	1	$\perp$	$\perp$	$x_1 < x_2$
8	1	2	$\perp$	$\perp$	$x_1 > x_2$
9	0	1	T	$\perp$	$x_1 > x_2$
10	1	0	$\perp$	Τ	$x_1 < x_2$
11	0	2	Τ	$\perp$	$x_1 > x_2$
12	2	0	$\perp$	T	$x_1 < x_2$

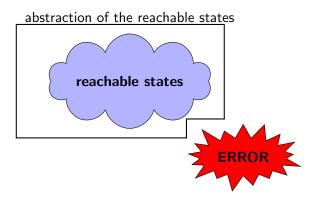


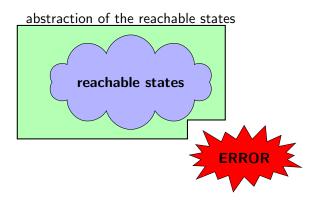












# Questions?