

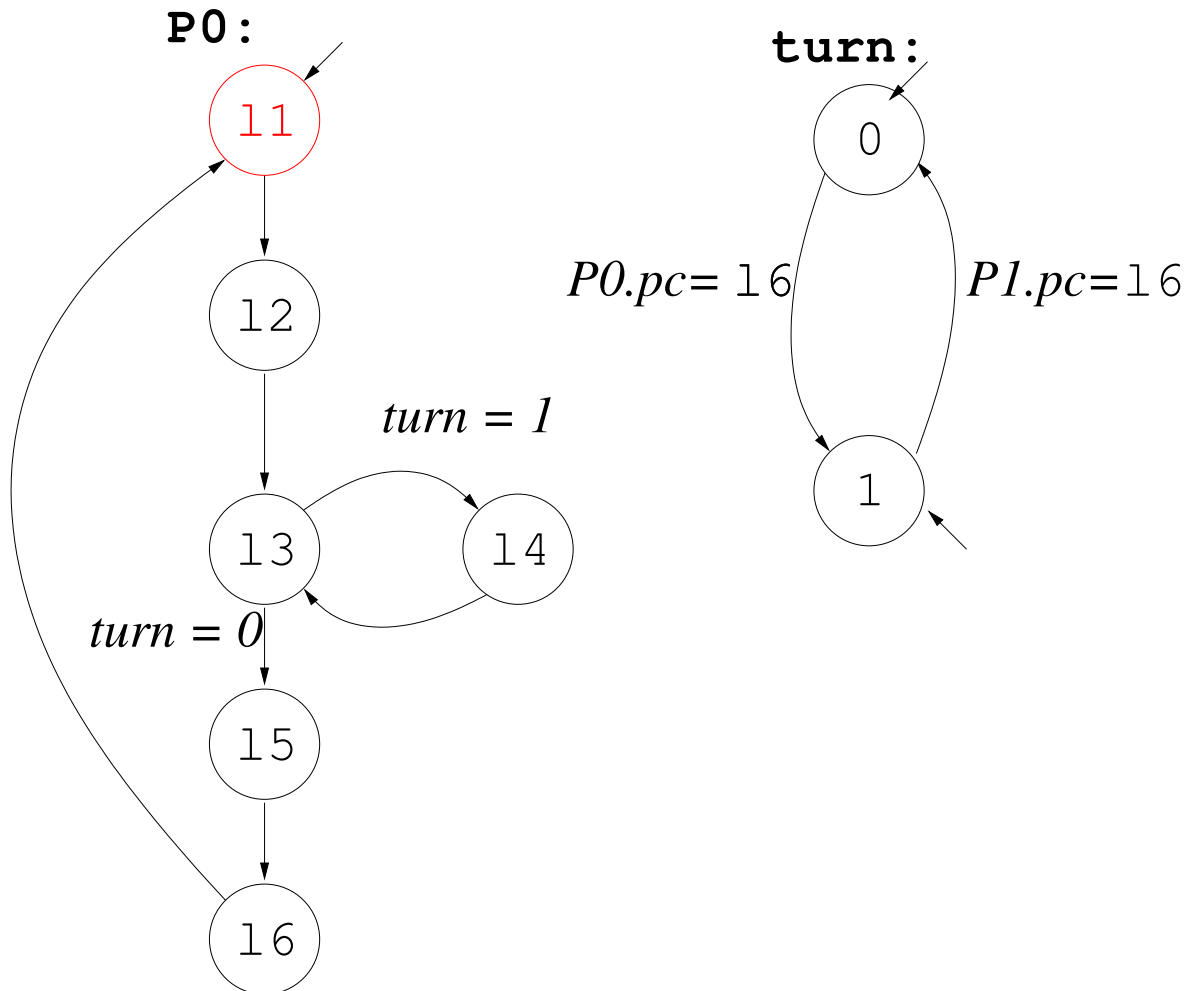
```
bool turn;  
  
procedure p0  
{  
  while true do  
  {  
    NONCRITICAL;  
    while (turn = 1) do skip;  
    CRITICAL;  
    turn := 1;  
  }  
  
procedure p1  
{  
  while true do  
  {  
    NONCRITICAL;  
    while (turn = 0) do skip;  
    CRITICAL;  
    turn := 0;  
  }  
}
```

```
procedure p0
{
11: while true do
    {
12:  NONCRITICAL;
13:  while (turn = 1) do 14: skip;
15:  CRITICAL;
16:  turn := 1;
    }
```

```

procedure P0
{
  11: while true do
    {
      12: NONCRITICAL;
      13: while (turn = 1) do 14 skip;
      15: CRITICAL;
      16: turn := 1;
    }
}

```



```
MODULE p(turn, myturn)
```

```
VAR
```

```
  s:{N, I, C, X};
```

```
ASSIGN
```

```
  init(s) := N;
```

```
  next(s) :=
```

```
    case
```

```
      (s = N) : {N, I};
```

```
      (s = I) & (turn = myturn) : C;
```

```
      (s = C) : X;
```

```
      (s = X) : N;
```

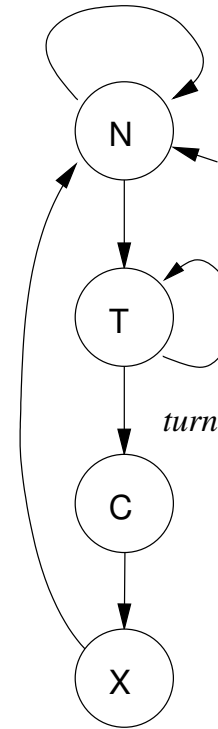
```
    1: s;
```

```
  esac;
```

*true*

*O.W.*

*turn = myturn*



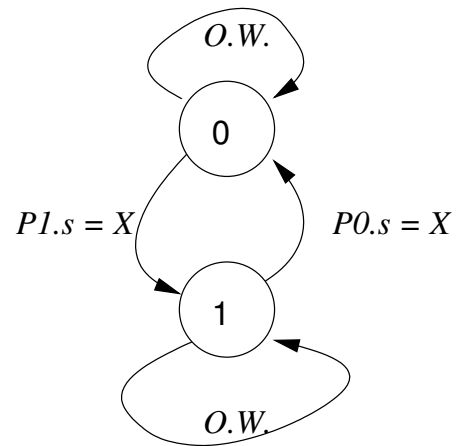
```
next(whoseturn) :=
```

```
  case
```

```
    (s = X) : !myturn;
```

```
    1 : turn;
```

```
  esac;
```



FAIRNESS running

```
MODULE main
```

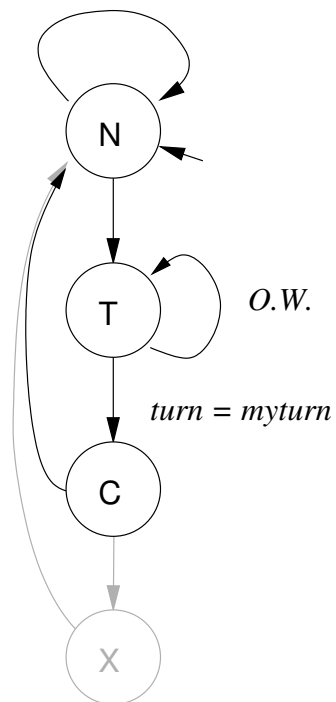
```
VAR
```

```
turn: boolean;
```

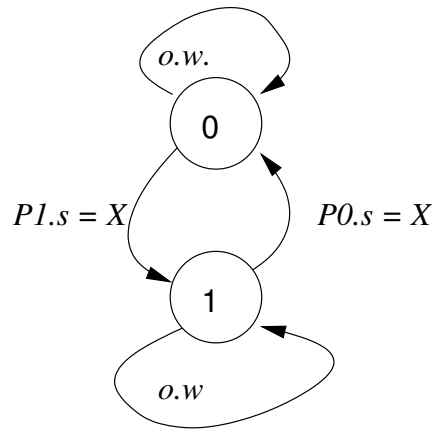
```
p0: process p(turn, 0);
```

```
p1: process p(turn, 1);
```

P0



turn



P1

