

# The Bakery Algorithm

--algorithm *Bakery*

```
{ variable  $num = [i \in Procs \mapsto 0]$ ,  $flag = [i \in Procs \mapsto \text{FALSE}]$  ;  
  process (  $p \in Procs$  )  
    variables  $unchecked = \{\}$ ,  $max = 0$ ,  $nxt = 1$  ;  
    {  $nsc$ : while ( TRUE )  
      {  $e1$ : either {  $flag[self] := \neg flag[self]$  ;  
                    goto  $e1$  }  
        or {  $flag[self] := \text{TRUE}$  ;  
               $unchecked := Procs \setminus \{self\}$  ;  
               $max := 0$   
            } ;  
       $e2$ : while (  $unchecked \neq \{\}$  )  
        { with (  $i \in unchecked$  )  
          {  $unchecked := unchecked \setminus \{i\}$  ;  
            if (  $num[i] > max$  ) {  $max := num[i]$  }  
          }  
        } ;  
       $e3$ : either { with (  $k \in Nat$  ) {  $num[self] := k$  } ;  
                goto  $e3$  }  
        or { with (  $i \in \{j \in Nat : j > max\}$  )  
              {  $num[self] := i$  }  
            } ;  
       $e4$ : either {  $flag[self] := \neg flag[self]$  ;  
                goto  $e4$  }  
        or {  $flag[self] := \text{FALSE}$  ;  
               $unchecked := Procs \setminus \{self\}$   
            } ;  
       $w1$ : while (  $unchecked \neq \{\}$  )  
        { with (  $i \in unchecked$  ) {  $nxt := i$  } ;  
          await  $\neg flag[nxt]$  ;  
           $w2$ : await  $\vee num[nxt] = 0$   
                   $\vee \langle num[self], self \rangle \prec \langle num[nxt], nxt \rangle$  ;  
           $unchecked := unchecked \setminus \{nxt\}$  ;  
        } ;  
       $cs$ : skip ; the critical section;  
      exit: either { with (  $k \in Nat$  ) {  $num[self] := k$  } ;  
                   goto exit }  
        or {  $num[self] := 0$  }  
    }  
  }
```

