

The Atomic Bakery Algorithm

--algorithm *AtomicBakery*

```
{ variable  $num = [i \in Procs \mapsto 0]$ ,  $flag = [i \in Procs \mapsto \text{FALSE}]$  ;  
  process (  $p \in Procs$  )  
    variables  $unchecked = \{\}$ ,  $max = 0$ ,  $nxt = 1$  ;  
    {  $ncs$ : while ( TRUE )  
      { e1:  $flag[self] := \text{TRUE}$  ;  
           $unchecked := Procs \setminus \{self\}$  ;  
           $max := 0$  ;  
          e2: while (  $unchecked \neq \{\}$  )  
              { with (  $i \in unchecked$  )  
                  {  $unchecked := unchecked \setminus \{i\}$  ;  
                      if (  $num[i] > max$  ) {  $max := num[i]$  }  
                  }  
              } ;  
          e3: with (  $i \in \{j \in Nat : j > max\}$  ) {  $num[self] := i$  } ;  
          e4:  $flag[self] := \text{FALSE}$  ;  
               $unchecked := Procs \setminus \{self\}$  ;  
          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:  $num[self] := 0$  ;  
      }  
    }  
}
```