

Algorithm

```
lock  $\rightarrow$  0
while (true) {
    a1 while (compare-and-swap(&lock, 1, 1) != 0);
        /* do nothing */
    a2 /* critical section */
    a3 lock = 0;
    a4 /* Remainder section */
}
```

① mutual exclusion :- lock \rightarrow 0

$P_0(a_1), P_0(a_2) CS, P_1(a_1), P_1(a_2)$ not satisfied

② progress :- lock \rightarrow 0, 0, 0

$P_0(a_1), P_0(a_2) CS, P_0(a_3) RS, P_1(a_1), P_1(a_2),$
 $P_1(a_3), P_1(a_4) RS$ satisfied
 $P_0(a_1), P_1(a_2) CS$

③ Bounded waiting :- lock \rightarrow 0, 0

$P_0(a_1)$ waiting, $P_1(a_1), P_1(a_2) CS$

Algorithm

lock $\Rightarrow 1$

```
while (true) {  
    a1 while (compare-and-swap (&lock, 0, 1) != 0);  
        /* do nothing */  
    a2 /* critical section */  
    a3 lock = 0;  
    a4 /* Reminder Section */  
}
```

① mutual exclusion :- lock $\Rightarrow 1$

P0(a1) waiting because $1 \neq 0 \rightarrow \text{return 1}$

② progress :-

P0(a1) waiting

③ Bounded waiting :-

P0(a1) waiting

Algorithm

waiting[4] = F, F, F, F

lock = 0

2

a₁ waiting[i] = 1;

a₂ key = 1;

a₃ while (waiting[i] && key)

a₄ key = test-and-set (&lock);

a₅ waiting[i] = 0;

a₆ /* critical section */

a₇ j = (i + 1) % 4;

a₈ while ((j != i) && (waiting[j]))

a₉ j = (j + 1) % 4;

a₁₀ if (j == i)

a₁₁ lock = 0;

~~a₁₂~~ else

a₁₂ waiting[j] = 0;

a₁₃ /* Remainder section */

3 while (true)

(i) mutual exclusion lock 0, 1, 1 F, F, F, F

key 1, 0, 1 T, T

P₀(a₁), P₀(a₂), P₀(a₃), P₀(a₄), P₀(a₃), P₀(a₅)

P₀(a₆) CS, P₁(a₁), P₁(a₂), P₁(a₃), P₁(a₄) wait

② progress & lock 0,1 key 1,0
 $P_0(a_1), P_0(a_2), P_0(a_3), P_0(a_4), P_0(a_5),$
 $P_0(a_6) \text{ CS}, P_0(a_7) \quad j = 0 + 1 \times 4 = 1$
 $P_0(a_8), P_0(a_9) \quad j = 1 + 1 \times 4 = 2, P_0(a_8)$
 $P_0(a_9) \quad j = 2 + 1 \times 4 = 5, P_0(a_5), P_0(a_9)$

③ Bounded waiting lock = 0,1,1 F,F,F,F
 key $\Rightarrow 1, 0, 1, 1$
 $P_0(a_1), P_0(a_2), P_0(a_3), P_0(a_4), P_0(a_5)$
 $P_0(a_5) \text{ waiting}, P_1(a_1), P_1(a_2), P_1(a_3)$
 $P_1(a_4), P_1(a_5), P_1(a_6) \text{ waiting}$