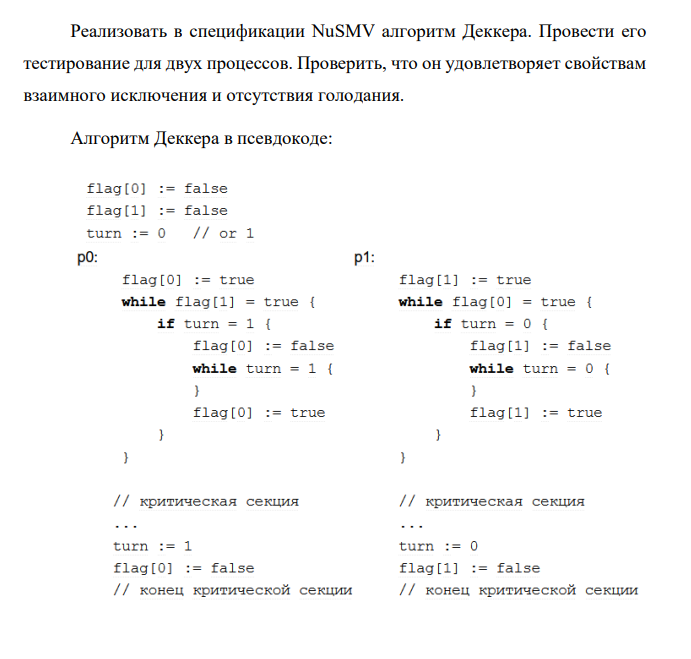
**Постановка задачи**



**Решение**

Свойство взаимного исключения:

***AG !(proc1.state = CRITICAL & proc2.state = CRITICAL)***

Свойство отсутствия голодания:

***AG (proc1.state = ENTERING -> AF proc1.state = CRITICAL) & AG (proc2.state = ENTERING -> AF proc2.state = CRITICAL)***

Оба свойства выполняются

**Код программы**

MODULE user(i, k, b)

VAR

state : { IDLE, ENTERING, CRITICAL, EXITING };

DEFINE

j := 3 - i;

ASSIGN

init(state) := IDLE;

next(state) := case

state = IDLE : { IDLE, ENTERING };

state = ENTERING & b[j] = FALSE : { CRITICAL };

state = ENTERING & b[j] = TRUE : { ENTERING };

state = CRITICAL : { CRITICAL, EXITING };

state = EXITING : IDLE;

esac;

next(b[i]) := case

state = ENTERING & b[j] = TRUE & k = j : FALSE;

state = ENTERING : TRUE;

state = EXITING : FALSE;

TRUE : b[i];

esac;

next(k) := case

state = EXITING : j;

TRUE : k;

esac;

FAIRNESS

running;

FAIRNESS

state = EXITING;

FAIRNESS

state = ENTERING;

MODULE main

VAR

k : {1, 2};

b : array 1 .. 2 of boolean;

proc1 : process user(1, k, b);

proc2 : process user(2, k, b);

ASSIGN

init(b[1]) := FALSE;

init(b[2]) := FALSE;

CTLSPEC AG !(proc1.state = CRITICAL & proc2.state = CRITICAL);

CTLSPEC AG (proc1.state = ENTERING -> AF proc1.state = CRITICAL) & AG (proc2.state = ENTERING -> AF proc2.state = CRITICAL);