

Практика 16.02.22

int $n := 0$

bool $flag := 0$

P:

P1: while ($\neg flag$) {

P2: $n := 1 - n$

}

(end)

Q:

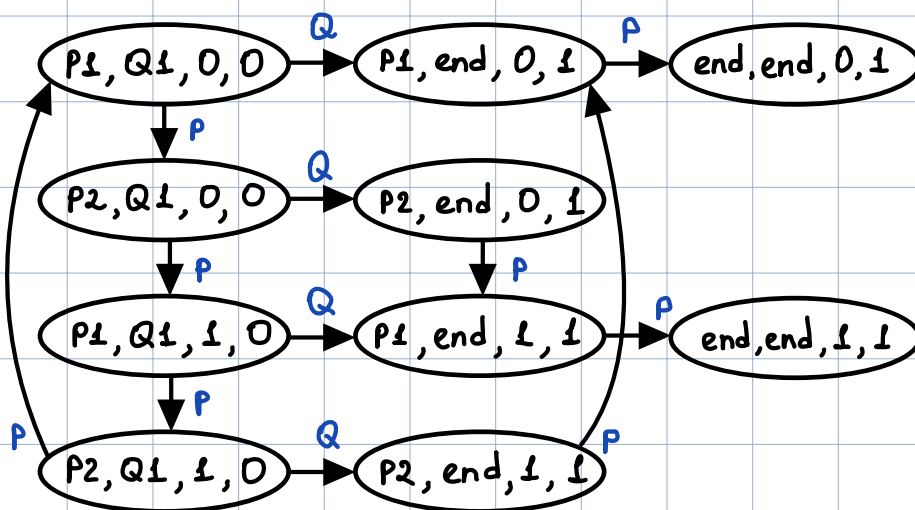
Q1: $flag := true$

(end)

Картеж состояний

$\langle p_i, q_i, n, flag \rangle$

$$3 \cdot 2 \cdot 2 \cdot 2 = 24$$



$P_1 P_2 \dots P_{n-1} P_n$

Р/м алгоритм Петерсона

Как сформулировать на языке LTL свойство отсутствия голодания
и свойство пропуска хода?

acquire_mutex(i) {

(1) FLAG[i] := 1

(2) AFTER_YOU := i

(3) while ((FLAG[i] = 1) & (AFTER_YOU = i)) {

(5) skip

}

(6) }

release_mutex(i) {

(7) FLAG[i] := 0

}

LTL = { \neg , \wedge , x , \vee }

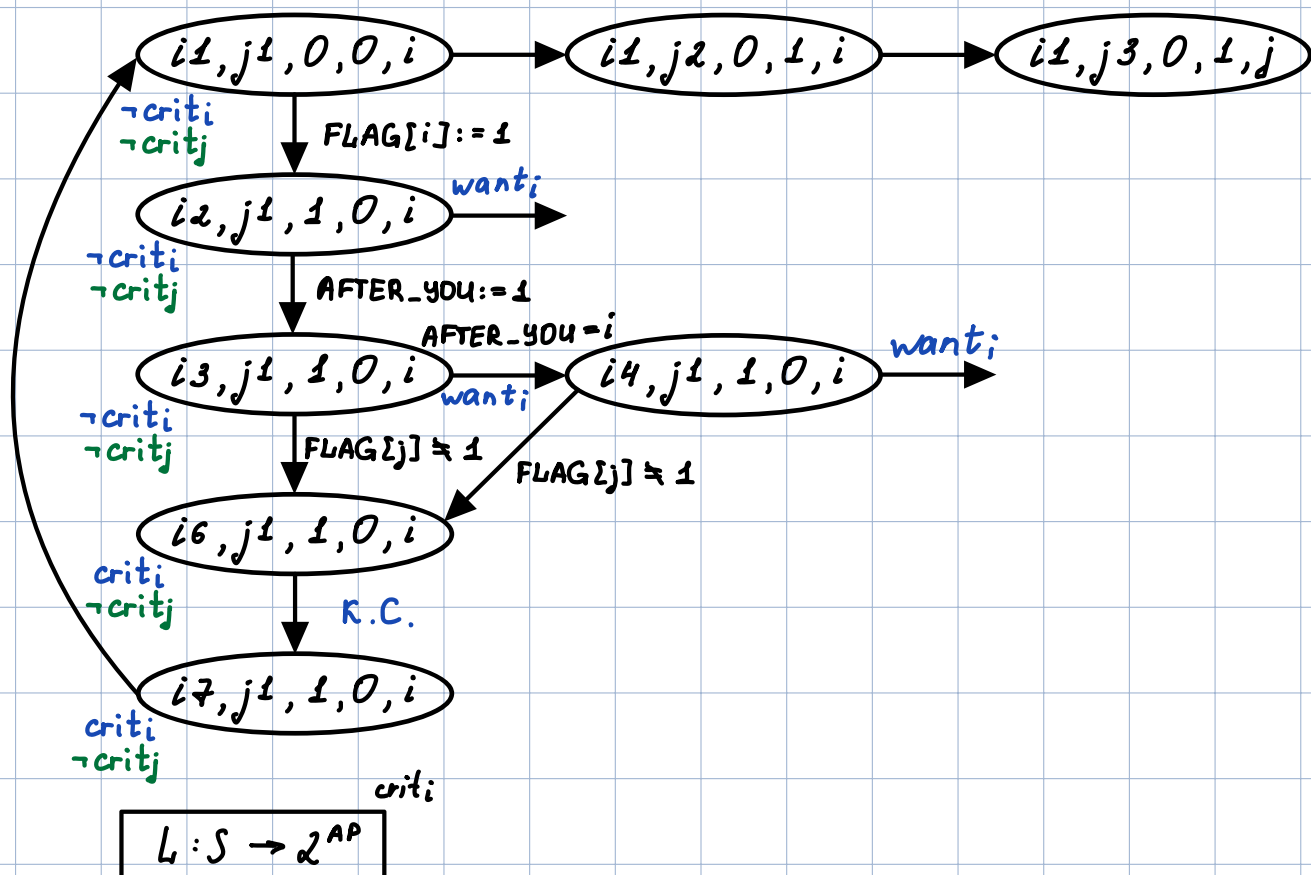
Как сказать, что оба процесса не попадают в КС?

Нужно ввести пометки.

AP = {crit_i, crit_j}

Пока мы находимся в строке (6), то мы уже находимся в СП

@i6 = crit_i (окончание препротоккола)



$crit_i = @i6 \vee @i7$

$want_i = @i2 \vee @i3 \vee @i4 \vee @i5$

$crit_j = @j6 \vee @j7$

Ошибка: $crit_j \& crit_j$

Свойство взаимно исключаящего доступа на языке LTL:

$G \neg (crit_i \& crit_j)$

Свойство отсутствия голодания

$AP = \{ crit_i, crit_j, want_i, want_j \}$

Всегда, если процесс i хочет войти в КС, то он когда-нибудь в нее войдет

Свойство отсутствия голодания на языке LTL

$G(want_i \rightarrow Fcrit_i)$

$want_i \vee crit_i$