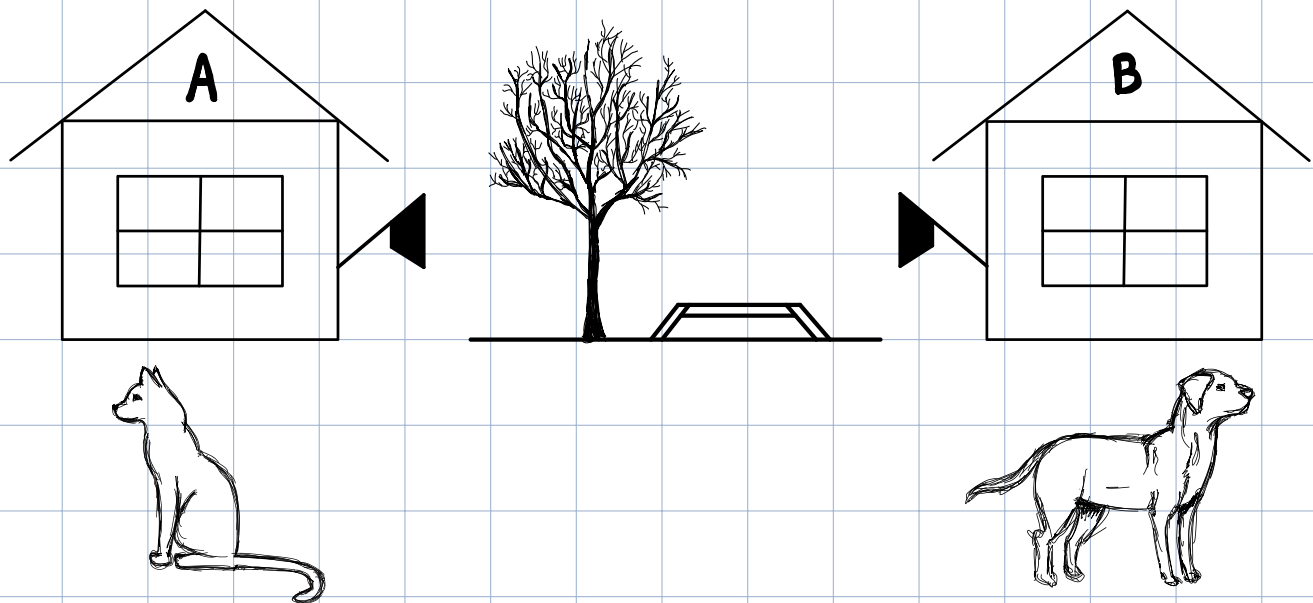


## Практика 09.02.22



- А:
- 1) поднимает ФЛАГ (механизм синхронизации)
  - 2) если ФЛАГ Боба опущен, то она выпускает кота
  - 3) когда кот возвращается, то она опускает ФЛАГ

- В:
- 1) поднимает ФЛАГ
  - 2) если ФЛАГ А поднят, то:
    - он опускает ФЛАГ
    - ждет, когда А опустит
    - снова поднимает
  - 3) когда его ФЛАГ поднят, а у А опущен, тогда он выпускает собаку
  - 4) когда собака возвращается, он опускает ФЛАГ

}} препрокол

распределенный алгоритм

основной  
алгоритм

}} постпрокол

распределенный алгоритм

Правильный алгоритм

Неправильный алгоритм

Alice - loop forever {

a1: flagA = 1

a2: while (flagB) {

a3: skip

}

a4: K.C.

a5: flagA = 0

}

Bob - loop forever {

b1: flagB = 1

b2: if (flagA) {

b3: B = 0

}

b4: while (flagA) {

b5: skip

}

b6: flagB = 1

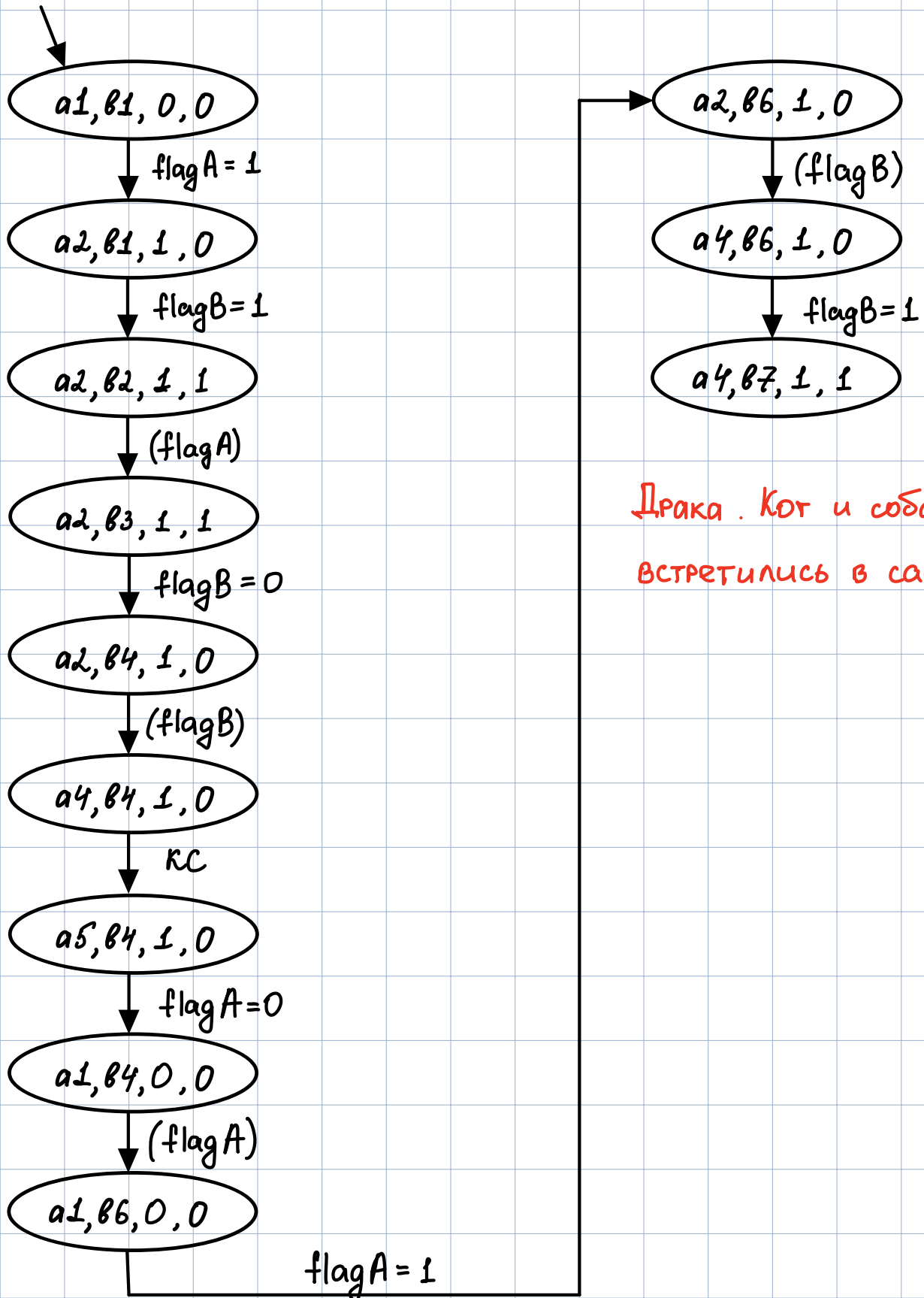
b7: K.C.

b8: flagB = 0

}

Кортеж состояний  $\langle a, b, \text{flagA}, \text{flagB} \rangle$

Проверка построенных алгоритмов



Драка. Кот и собака  
встретились в саду.

Правильный алгоритм	Правильный алгоритм
<pre> Alice_loopforever { a1:  flagA = 1 a2:  while (flagB) { a3:    skip       } a4:  К.С. a5:  flagA = 0     }       } </pre>	<pre> Bob_loopforever { b1:  flagB = 1 b2:  while (flagA) { b3:    B = 0 b4:    while(flagA){ b5:      skip         } b6:    flagB = 1       } b7:  К.С. b8:  flagB = 0     } </pre>

Количество состояний в системе переходов:

5 (состояний A)  $\times$  8 (состояний B)  $\times$  2 (значения флага A)  $\times$  2 (значения флага B)  $\leq$  160 состояний

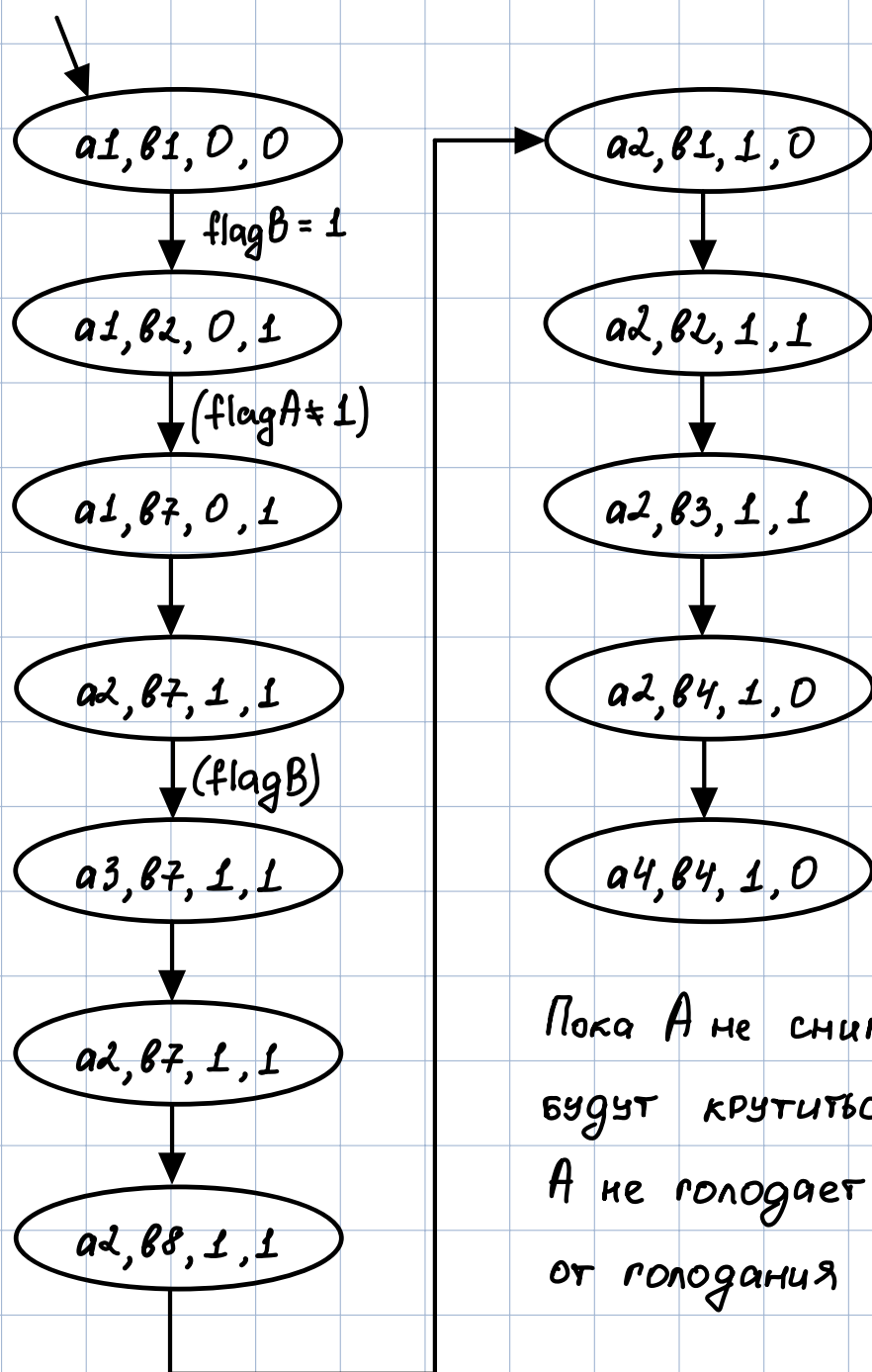
- Свойство взаимно исключаящего доступа

Ищем, что нет состояния a4, b7, ..., ...

Если не строить систему переходов, то можно проверить от противного.

- Свойство отсутствия голодания

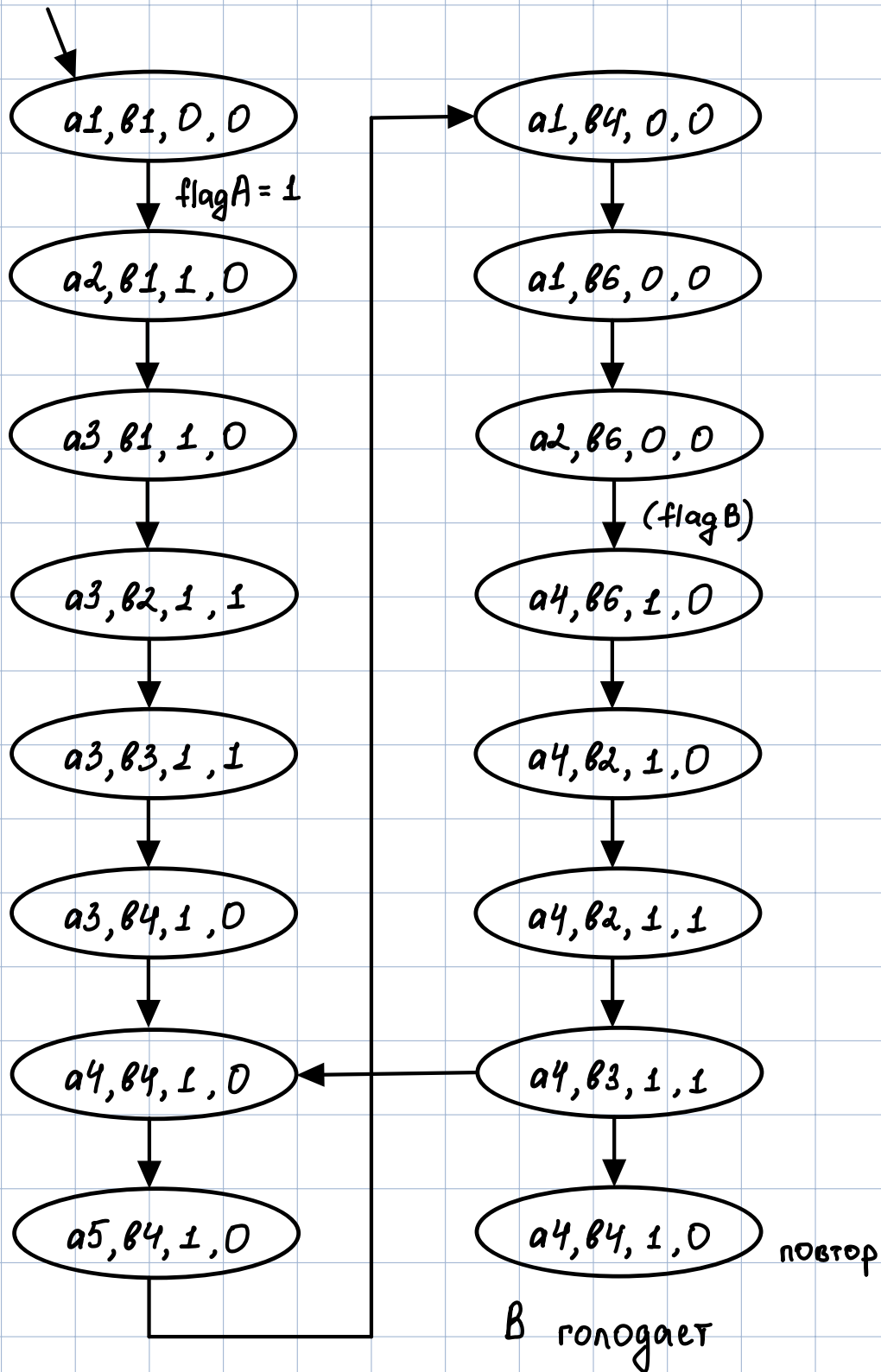
Проверка для процесса A:



Пока A не снимется, b4 и b5  
будут крутиться

A не голодает => свойство свободы  
от голодания выполняется

# Проверка для процесса В :



## Задача из Ben-Ari

int  $n := 0$

P::

P1: do 10 times {

P2:  $n := n + 1$

}

Q::

Q1: do 10 times {

Q2:  $n := n + 1$

}

Если процессы

атомарны, то

$n$  будет равно 20

int  $n := 0$

P:: int  $t_p := 0$

P1: do 10 times {

P2:  $t_p := n$

P3:  $n := t_p + 1$

}

Q:: int  $t_q := 0$

Q1: do 10 times {

Q2:  $t_q := n$

Q3:  $n := t_q + 1$

}

$2 \leq n \leq 20$

$n = 20$ :

$(P1P2P3)^{10\text{раз}} (Q1Q2Q3)^{10\text{раз}}$

$n = 10$ :

$(P1P2P3 Q1Q2Q3)^{10\text{раз}}$

P	Q	$t_p$	$t_q$	$n$
P1	Q1	0	0	0
P2	Q1	0	0	0
P3	Q1	0	0	0
P3	Q2	0	0	0
P3	Q3	0	0	0
P1	Q3	0	0	1
P2	Q3	0	0	1

и т.д.

## Дома :

- 2.9. Алгоритм

$n=2$  получить

- 2.14. Раздел

Лягушки. Построить систему переходов для 2-ух пар.

- 2.19.

Система переходов. задача 9

- 2.6. задача

$P_2$ , а не  $P_3$

- 12 задание. Вычисление и система переходов