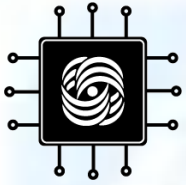


# **ПРАКТИКУМ НА ЭВМ**

## **Задание 2:**

### **Размеченные системы переходов (LTS)**

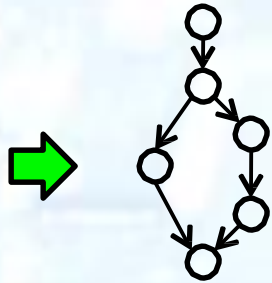
ВМК МГУ им. М.В. Ломоносова,  
Кафедра АСВК  
к.ф.-м.н., доцент Волканов Д.Ю.  
аспирант Степанов Е.П.



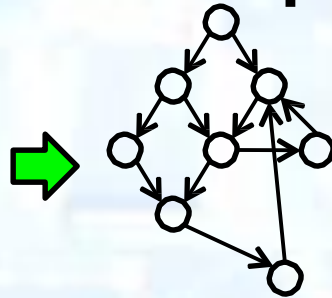
# Различные представления программы

```
int main
() {
printf(
}
```

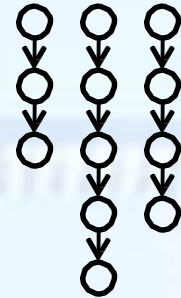
Исходный код программы



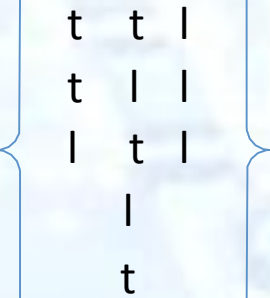
Граф программы (ACFG)



Размеченная система переходов



Множество вычислений



Множество трасс

Взлетает,  
не падает,  
приземляется

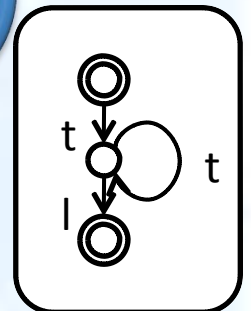
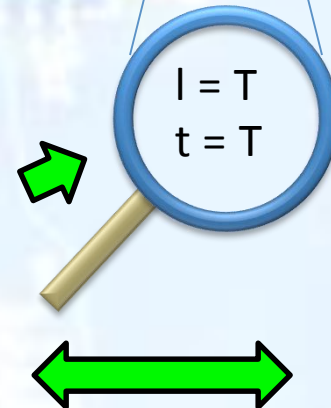
Требования

□ ( TAKEOFF → (!FALL) U (LANDED) )

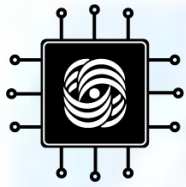
Спецификация (линейного времени)

|  |   |   |   |
|--|---|---|---|
|  |   |   | t |
|  | t | t |   |
|  |   |   |   |
|  |   |   |   |

Допустимые последовательности атомарных высказываний



Язык допустимых трасс

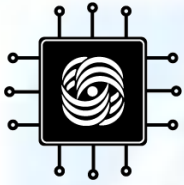


# Размеченные системы переходов (LTS)

$$TS = \langle S, Act, \xrightarrow{a}, I, AP, L \rangle$$

- $S$  – множество состояний,
- $Act$  – множество действий,  $\tau$  – невидимое действие,
- $\xrightarrow{a} \subseteq S \times Act \times S$  – **тотальное** отношение переходов,
- $I \subseteq S$  – **множество** начальных состояний,
- $AP$  – множество атомарных высказываний,
- $L: S \rightarrow 2^{AP}$  – функция разметки.

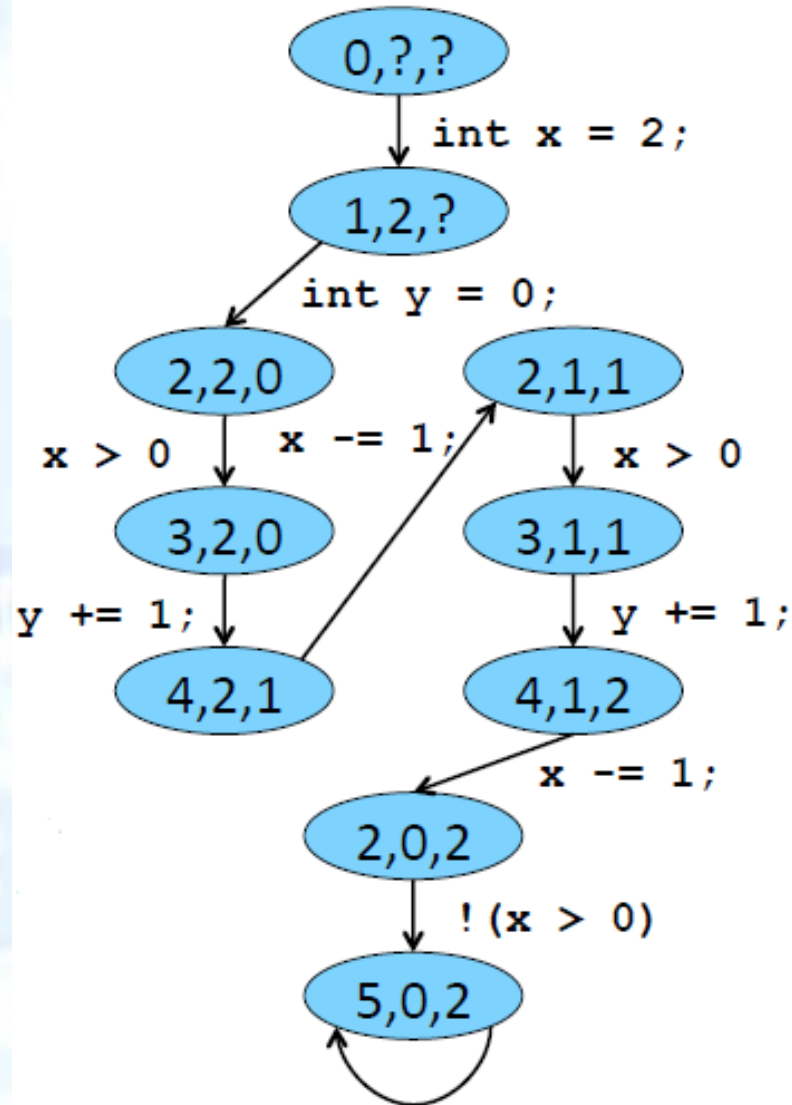
$$\left( \begin{array}{l} S \text{ – конечное или счётное множество, } Act \text{ – конечное} \\ \text{Нотация: } \langle s, a_0, s' \rangle \in \xrightarrow{a} \equiv s \xrightarrow{a_0} s' \end{array} \right)$$

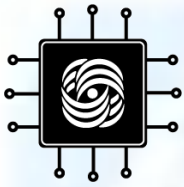


# Пример LTS

```
0: int x = 2;  
1: int y = 0;  
2: while (x>0)  
  {  
3:   y += 1;  
4:   x -= 1;  
  }  
5:
```

( Состояние: (счётчик, x, y) )





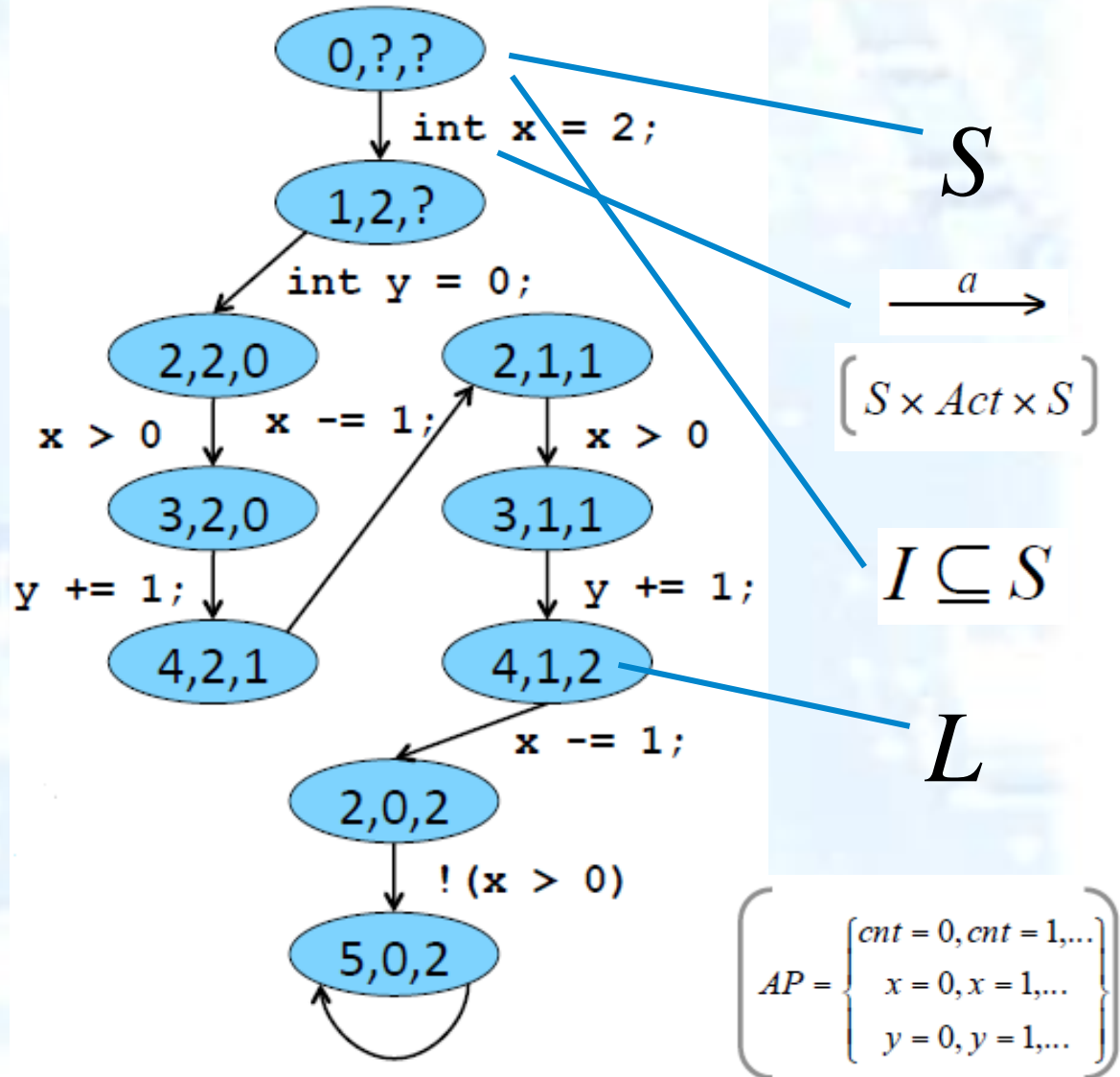
# Пример LTS

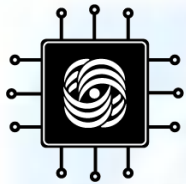
```

0: int x = 2;
1: int y = 0;
2: while (x>0)
{
3:   y += 1;
4:   x -= 1;
}
5:

```

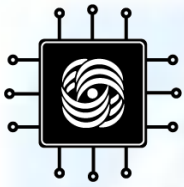
( Состояние: (счётчик, x, y) )





# Что включать в состояние

- Набор атомарных высказываний **AP** определяется свойствами, которые нужно проверить
- Изменение состояния связано с изменением выполнимости хотя бы одного атомарного высказывания
- Исходя из этого мы определяем, что включать в состояние программы
- Главное – не «потерять» ни одного изменения атомарных состояний



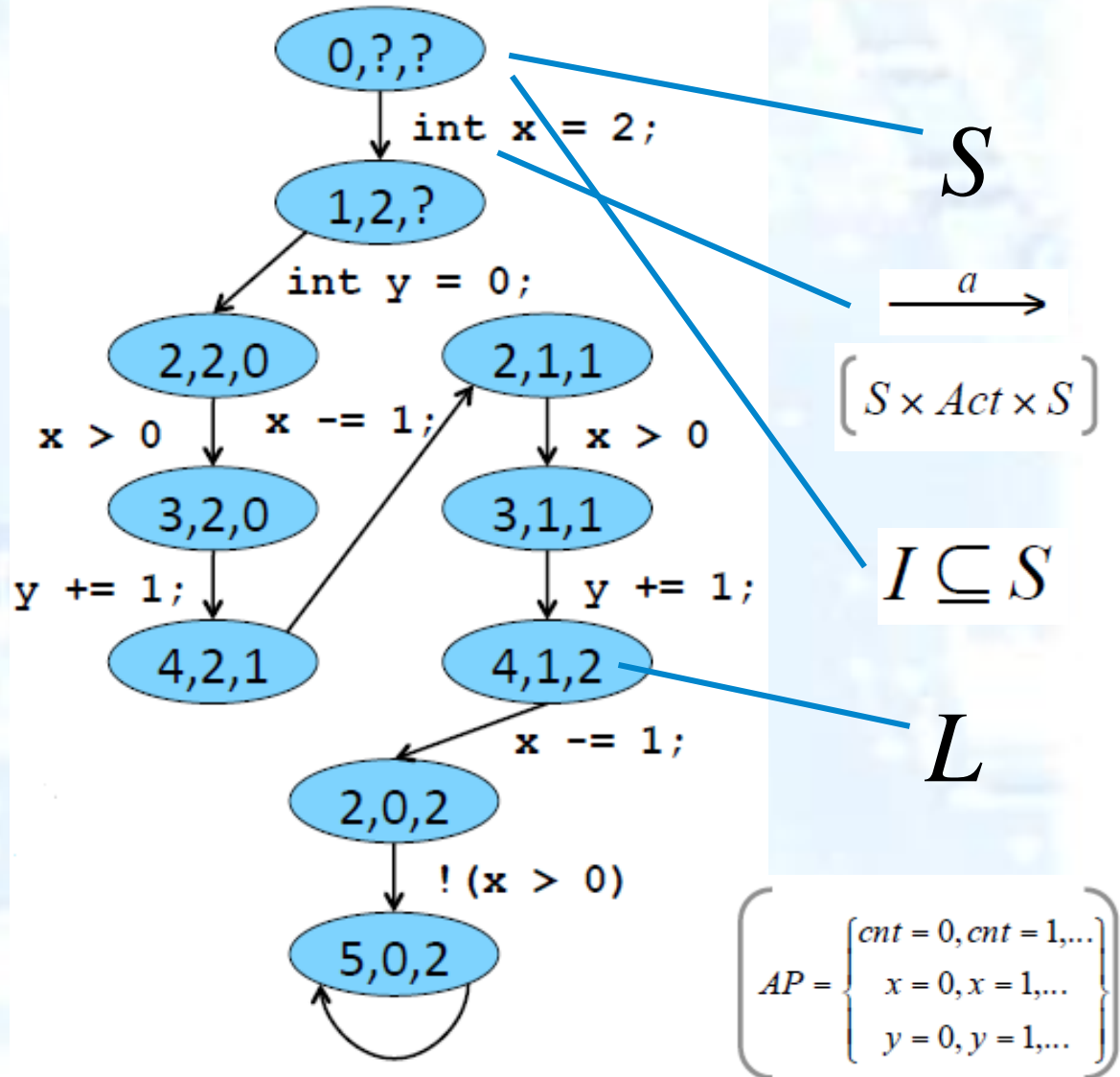
# Пример LTS

```

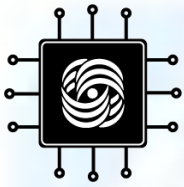
0: int x = 2;
1: int y = 0;
2: while (x>0)
{
3:   y += 1;
4:   x -= 1;
}
5:

```

( Состояние: (счётчик, x, y) )



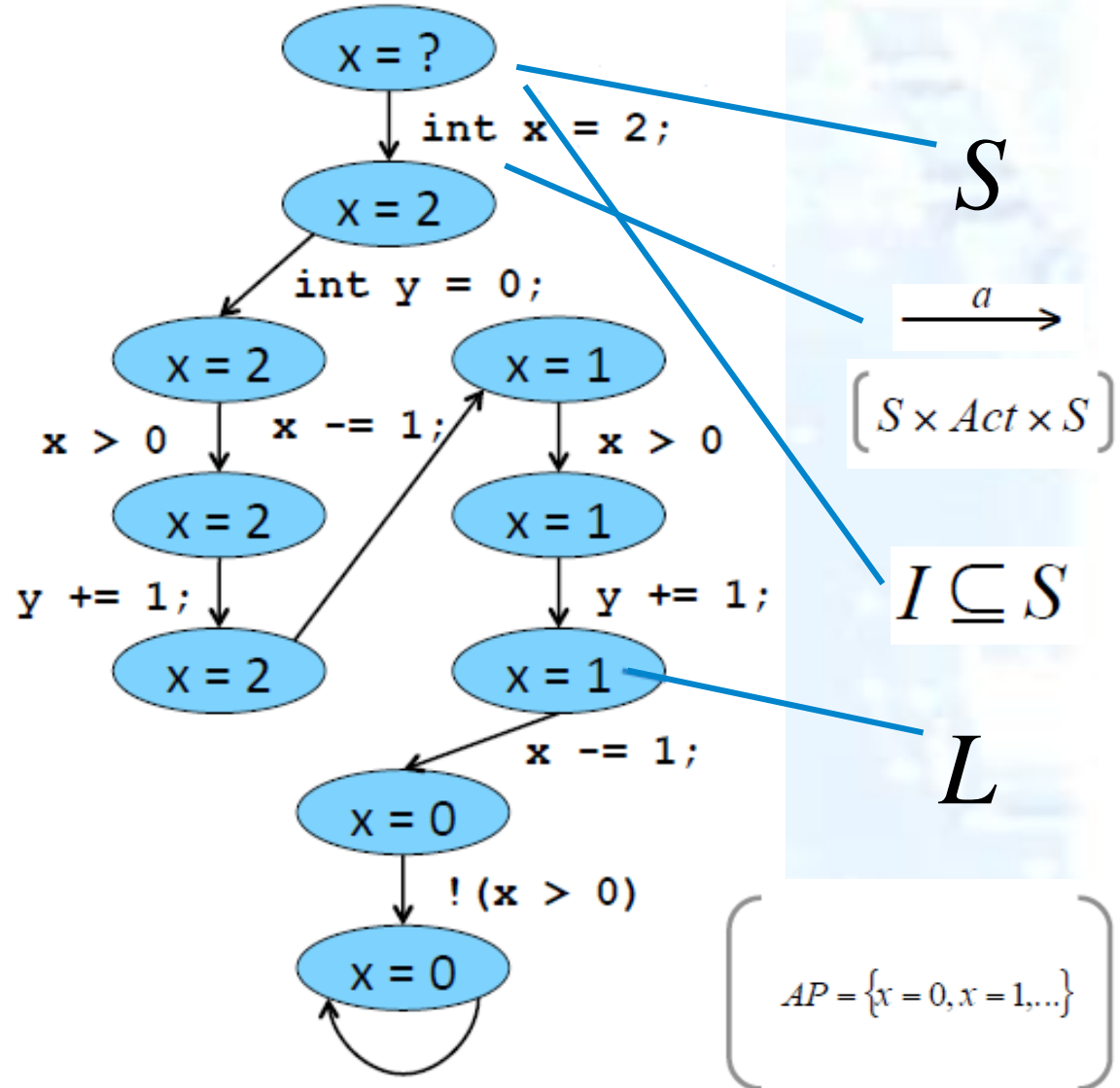




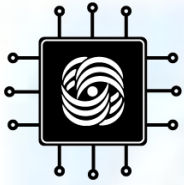
# Пример LTS

```
0: int x = 2;  
1: int y = 0;  
2: while (x>0)  
  {  
3:   y += 1;  
4:   x -= 1;  
  }  
5:
```

Состояние: (счётчик, x, y)



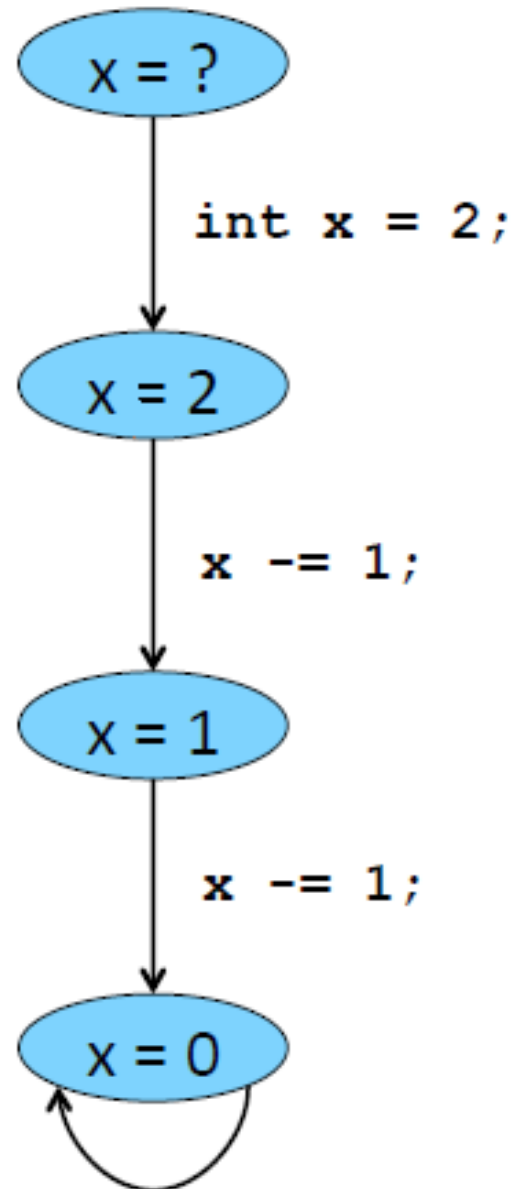




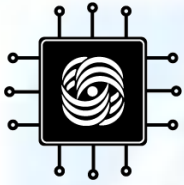
# Пример LTS

```
0: int x = 2;  
1: int y = 0;  
2: while (x>0)  
   {  
3:   y += 1;  
4:   x -= 1;  
   }  
5:
```

( Состояние: (счётчик, x, y) )



$AP = \{x = 0, x = 1, \dots\}$



# Пример LTS

```
int p;
```

```
Process Prod() {
```

```
  while(1)
```

```
1.1:   if(p < 2)
```

```
1.2:   p += 1;
```

```
}
```

```
process Cons() {
```

```
  while(1)
```

```
2.1:   if(p > 0)
```

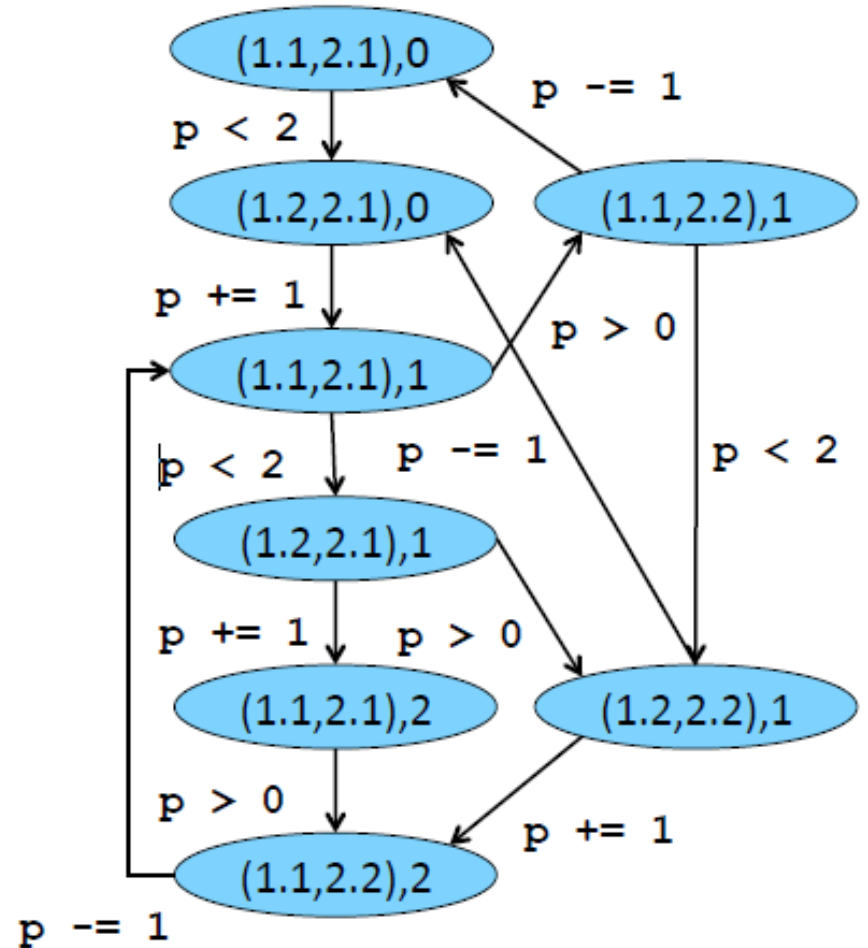
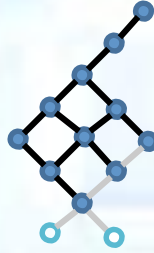
```
2.2:   p -= 1;
```

```
}
```

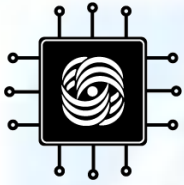
Состояние: (счётчик Prod,  
счётчик Cons, p)

Часть состояний не показана  
(оператор беск. цикла,  
стартовые состояния)

Prod ↔ Cons

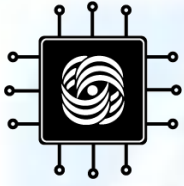


Бесконечное количество вычислений, однако  
размеченная система переходов конечна<sup>10</sup>



# Сроки

- Задача сдаётся в МЗ 758 14:35
  - 05.11 (+50%)
  - 12.11 (+20%)
  - 19.11, 26.11 (+0%)
- Ссылка на **сданную** задачу присылается на [volkanvov@lvk.cs.msu.su](mailto:volkanvov@lvk.cs.msu.su) письмом с темой Praktikum-Task2-Solution
- Присланная до 23:59 (московское время) 26 ноября задача принимается (**БЕЗ ШТРАФА**), далее -2 балла за каждую неделю
- В **ИСКЛЮЧИТЕЛЬНЫХ СЛУЧАЯХ** возможна сдача по почте



**Спасибо за внимание!**