

# Математические методы верификации схем и программ

mk.cs.msu.ru → Лекционные курсы  
→ Математические методы верификации схем и программ

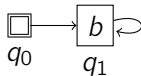
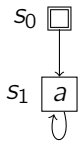
## Блок 10

Размеченные системы переходов  
Справедливость для систем переходов  
Справедливость и LTL

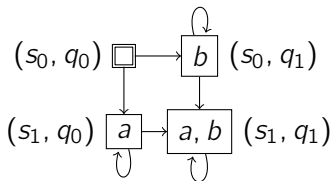
Лектор:  
**Подымов Владислав Васильевич**  
E-mail:  
**valdus@yandex.ru**

# Вступительный пример

Рассмотрим такие две модели Крипке:

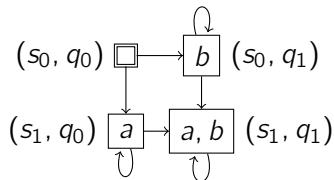


Асинхронная композиция этих моделей Крипке устроена так:



Насколько «реалистична» такая композиция?

## Вступительный пример



Представим себе, что исходные модели отвечают программам  $\pi_1$  и  $\pi_2$ , асинхронная композиция — их параллельному выполнению,  $a$  означает, что  $\pi_1$  выполнила своё единственное действие и  $b$  — что  $\pi_2$  выполнила своё единственное действие

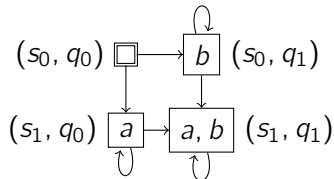
«Реальность» говорит, что если наблюдать за независимым параллельным выполнением  $\pi_1$  и  $\pi_2$  достаточно долго, то каждая из программ выполнит своё действие

При этом в асинхронной композиции есть вычисление

$$(s_0, q_0) \rightarrow (s_1, q_0) \rightarrow (s_1, q_0) \rightarrow (s_1, q_0) \rightarrow \dots,$$

в котором  $\pi_2$  не выполняет ни одного действия

## Вступительный пример



Каждый конечный начальный путь в модели можно признать реалистичным: в зависимости от медлительности  $\pi_2$ , программа  $\pi_1$  может выполнить своё действие любое наперёд заданное число раз. Но бесконечный путь, в котором  $\pi_2$  не выполняет ни одного действия, не соответствует реальности.

Можно сказать, что такой путь **несправедлив** по отношению к  $\pi_2$ , так как не даёт ей права выполнить даже одно действие.

Точно так же вычисление

$$(s_0, q_0) \rightarrow (s_0, q_1) \rightarrow (s_0, q_1) \rightarrow (s_0, q_1) \rightarrow \dots$$

**несправедливо** по отношению к  $\pi_1$ .

Так в моделях появляется понятие **справедливости**.

# Системы переходов

Для наиболее полного строгого задания справедливости обобщим модель Крипке, добавив обозначение выполняемых **действий** на переходы

**Размеченная система переходов** (с.п.) над множеством атомарных высказываний  $AP$  и множеством **действий**  $Act$  — это система  $TS = (S, S_0, \rightarrow, L)$ , отличающаяся от модели Крипке только устройством множества переходов  $\rightarrow$ :

$$\blacktriangleright \rightarrow \subseteq S \times Act \times S$$

С.п. будем называть **конечной**, если конечны множества  $S$ ,  $AP$  и  $Act$

Переход  $(s, \alpha, s') \in \rightarrow$  будем также понимать как помеченную дугу  $s \xrightarrow{\alpha} s'$

Будем говорить, что при выполнении перехода  $s \xrightarrow{\alpha} s'$  **выполняется действие  $\alpha$**

Записью  $Act(TS, s)$  для с.п.  $TS$  и состояния  $s$  обозначим множество действий, которые могут выполняться в с.п. из состояния  $s$ :

$$Act((S, S_0, \rightarrow, L), s) = \{\alpha \mid \exists s' : s \xrightarrow{\alpha} s'\}$$

# Виды справедливости

Принято рассматривать три вида справедливости:

1. **Безусловная справедливость**: система бесконечно часто выполняет действия множества  $A$ 
  - ▶ Пример справедливости: программа  $\pi_1$  должна бесконечно часто выполнять свои переходы
  - ▶ Соответствующая несправедливость: начиная с некоторого момента  $\pi_1$  всегда может выполнить переход, но ни разу не выполняет

# Виды справедливости

Принято рассматривать три вида справедливости:

2. **Сильная справедливость**: если система бесконечно часто получает возможность выполнить действия множества  $A$ , то она бесконечно часто выполняет эти действия
  - ▶ Пример справедливости: если  $\pi_1$  бесконечно часто получает возможность послать данные на печать, то она будет бесконечно часто посылать данные на печать
  - ▶ Соответствующая несправедливость: принтер бесконечно часто освобождается (и появляется возможность послать ему данные), но  $\pi_1$  ни разу не отправляет данные на печать

# Виды справедливости

Принято рассматривать три вида справедливости:

3. **Слабая справедливость**: если система **почти всегда** имеет возможность выполнить действия из  $A$ , то она бесконечно часто выполняет эти действия
  - ▶ Пример справедливости: если принтер почти всегда готов получать данные, то он будет бесконечно часто получать эти данные
  - ▶ Соответствующая несправедливость: начиная с некоторого момента принтер всегда ожидает данные на печать, но так их и не получает



# Справедливость в системах переходов

Рассмотрим бесконечный путь  $\rho$  вида

$$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \dots$$

системы переходов  $TS$

Этот путь для заданного множества действий  $A$  будем называть

- ▶ **безусловно  $A$ -справедливым**, если действия из  $A$  выполняются в  $\rho$  бесконечно часто
- ▶ **сильно  $A$ -справедливым**, если верно хотя бы одно из двух:
  - ▶ соотношение  $\text{Act}(TS, s_i) \cap A \neq \emptyset$  выполняется для не более чем конечного числа моментов времени  $i$
  - ▶  $\rho$  безусловно справедлив относительно  $A$
- ▶ **слабо  $A$ -справедливым**, если верно хотя бы одно из двух:
  - ▶ число моментов времени  $i$ , для которых верно  $\text{Act}(TS, s_i) \cap A = \emptyset$ , бесконечно
  - ▶  $\rho$  безусловно справедливо относительно  $A$

# Справедливость в системах переходов

Рассмотрим бесконечный путь  $\rho$  вида

$$s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \dots$$

системы переходов  $TS$

**Ограничениями справедливости** назовём тройку  $\mathcal{F} = (\mathcal{F}_u, \mathcal{F}_s, \mathcal{F}_w)$ , где  $\mathcal{F}_u, \mathcal{F}_s, \mathcal{F}_w \subseteq 2^{\text{Act}}$

Путь  $\rho$  будем называть  $\mathcal{F}$ -справедливым, если он

- ▶ безусловно  $A$ -справедлив относительно каждого  $A$  из  $\mathcal{F}_u$ ,
- ▶ сильно  $A$ -справедлив относительно каждого  $A$  из  $\mathcal{F}_s$  и
- ▶ слабо  $A$ -справедлив относительно  $A$  из  $\mathcal{F}_w$

Обозначим записью  $\text{Tr}_{\mathcal{F}}(TS)$  множество трасс всех  $\mathcal{F}$ -справедливых путей с.п.  $TS$

# Справедливость и LTL

Будем говорить, что ltl-формула **выполняется на с.п.  $TS$  в ограничениях справедливости  $\mathcal{F} = (\mathcal{F}_u, \mathcal{F}_s, \mathcal{F}_w)$**  ( $TS, \mathcal{F} \models \varphi$ ), если  $\text{Tr}_{\mathcal{F}}(TS) \subseteq \text{Tr}(\varphi)$

Пусть возможность выполнить действие из  $A$  на следующем переходе отвечает ltl-формуле  $\Phi_A$ , а выполнение действия  $A$  на следующем переходе отвечает формуле  $\Psi_A$

Сопоставим ограничениям  $\mathcal{F}$  формулу  $\Phi_{\mathcal{F}}$  следующего вида:

$$\left( \bigwedge_{A \in \mathcal{F}_u} \mathbf{GF}\Psi_A \right) \& \left( \bigwedge_{A \in \mathcal{F}_s} (\mathbf{GF}\Phi_A \rightarrow \mathbf{GF}\Psi_A) \right) \& \left( \bigwedge_{A \in \mathcal{F}_w} (\mathbf{FG}\Phi_A \rightarrow \mathbf{GF}\Psi_A) \right)$$

**Утверждение (о справедливости в LTL).** Для любых конечной с.п.  $TS$ , ограничений справедливости  $\mathcal{F}$  и формулы  $\varphi$  верно:

$$TS, \mathcal{F} \models \varphi \Leftrightarrow TS \models \Phi_{\mathcal{F}} \rightarrow \varphi$$

**Доказательство.** Можете попробовать самостоятельно, вспомнив семантику ltl-формул и утверждения о формулах вида  $\mathbf{FG}\psi$  и  $\mathbf{GF}\psi$