

Математические методы верификации схем и программ

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

Блок 13

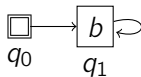
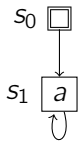
Размеченные системы переходов
Справедливость для систем переходов
Справедливость в LTL

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

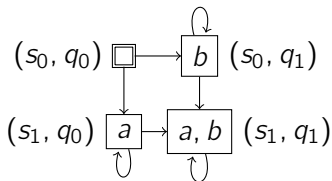
ВМК МГУ, 2024/2025, осенний семестр

Вступительный пример

Рассмотрим такие две модели Крипке:

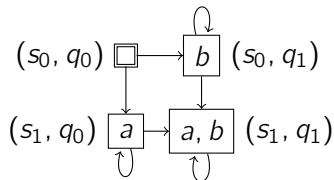


Асинхронная композиция этих моделей Крипке устроена так:



Насколько «реалистична» такая композиция?

Вступительный пример



Представим себе, что исходные модели отвечают программам π_1 и π_2 , асинхронная композиция — их параллельному выполнению, a означает, что π_1 выполнила своё единственное действие и b означает что π_2 выполнила своё единственное действие

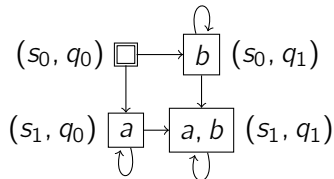
«Реальность» говорит, что если наблюдать за независимым параллельным выполнением π_1 и π_2 достаточно долго, то каждая из программ выполнит своё действие

При этом в асинхронной композиции есть «нереалистичное» вычисление

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

в котором π_2 не выполняет ни одного действия

Вступительный пример



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

При этом каждый конечный префикс ρ реалистичен:

в зависимости от темпа выполнения π_2 , программа π_1 может выполнить любое число своих действий перед первым действием π_2

Можно сказать, что ρ **несправедливо** по отношению к π_2 , так как не даёт этой программе 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 π_1

Так в моделях появляется понятие **справедливости**

Системы переходов

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

Размеченная система переходов (с.п.)

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

$$\blacktriangleright \rightarrow \subseteq S \times \text{Act} \times S$$

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 α**

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

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

Виды справедливости

Принято рассматривать три вида справедливости:

1. Безусловная справедливость:

система бесконечно часто выполняет действия множества A

- ▶ Соответствующая несправедливость:
с некоторого момента действия из A перестают выполняться
- ▶ Пример несправедливости:
переходы, отвечающие программе π в асинхронной композиции,
с некоторого момента никогда более не выполняются

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

- ▶ Соответствующая несправедливость:
с некоторого момента система бесконечно часто может выполнить действия из A , но ни разу не выполняет
- ▶ Пример несправедливости:
с некоторого момента принтер бесконечно часто (регулярно время от времени) оказывается свободным, но программе ни разу не удаётся его занять

Виды справедливости

Принято рассматривать три вида справедливости:

3. Слабая справедливость:

если система почти всегда

имеет возможность выполнить действия множества A ,
то она бесконечно часто выполняет эти действия

- ▶ Соответствующая несправедливость:
с некоторого момента система
всегда может выполнить действия из A ,
но ни разу не выполняет
- ▶ Пример несправедливости:
с некоторого момента принтер постоянно
(без перерывов, на которые можно было бы всё списать)
ожидает данные на печать, но никогда их не получает

Справедливость в системах переходов

Рассмотрим бесконечный путь ρ вида

$$s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} s_3 \xrightarrow{\alpha_3} \dots$$

в системе переходов TS

Этот путь для заданного множества действий A будем называть

- ▶ **безусловно A -справедливым**, если действия из A выполняются в ρ бесконечно часто
- ▶ **сильно A -справедливым**, если верно хотя бы одно из двух:
 - ▶ число моментов времени i , таких что $\text{Act}(TS, s_i) \cap A \neq \emptyset$, конечно
 - ▶ ρ безусловно A -справедлив
- ▶ **слабо A -справедливым**, если верно хотя бы одно из двух:
 - ▶ число моментов времени i , таких что $\text{Act}(TS, s_i) \cap A = \emptyset$, бесконечно
 - ▶ ρ безусловно A -справедлив

Справедливость в системах переходов

Рассмотрим бесконечный путь ρ вида

$$s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} s_3 \xrightarrow{\alpha_3} \dots$$

в системе 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 ρ будем называть \mathcal{F} -справедливым для ограничений справедливости $\mathcal{F} = (\mathcal{F}_u, \mathcal{F}_s, \mathcal{F}_w)$, если он

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

Обозначим записями $\Pi_{\mathcal{F}}(TS)$ и $\text{Tr}_{\mathcal{F}}(TS)$ соответственно множество всех \mathcal{F} -справедливых вычислений с.п. TS и множество всех трасс таких путей

Справедливость в 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 Φ_A , а выполнение действия A на следующем переходе отвечает формуле Ψ_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 φ 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 $\mathbf{GF}\psi$