

Математические методы верификации схем и программ

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

Блок 8

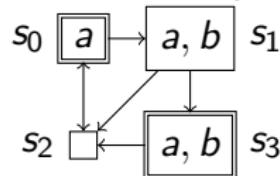
Свойства трасс
Безопасность и живость

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

Напоминание



Модель Кripке M над множеством атомарных высказываний $\{a, b\}$:



Вычисление τ модели M (начальный путь):

$$s_3 \rightarrow s_2 \rightarrow s_0 \rightarrow s_2 \rightarrow s_0 \rightarrow s_1 \rightarrow s_3 \rightarrow s_2 \rightarrow \dots$$

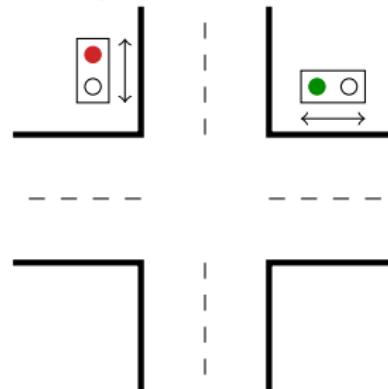
Трасса вычисления τ :

$$\{a, b\}, \emptyset, \{a\}, \emptyset, \{a\}, \{a, b\}, \{a, b\}, \emptyset, \dots$$

Перейдём к тому, как могут быть устроены **формальные спецификации** моделей Кripке и соответствующие **требования**, предъявляемые к вычислительным системам

Свойства трасс

Пример: перекрёсток со светофорами

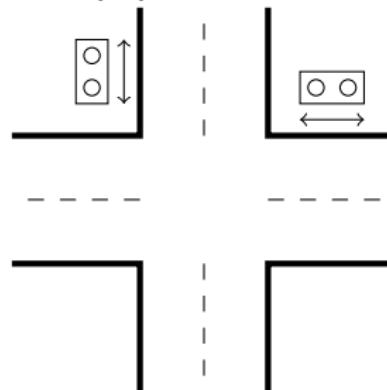


Пример вычисления этой системы:

$$(\boxed{\bullet}, \boxed{\circ}) \rightarrow$$

Свойства трасс

Пример: перекрёсток со светофорами

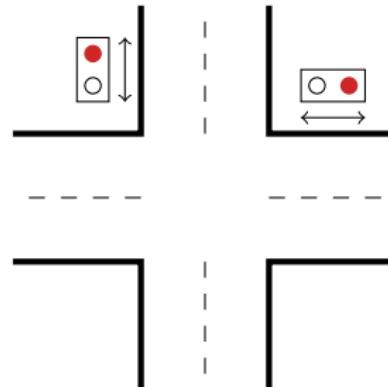


Пример вычисления этой системы:

$$(\boxed{\bullet}, \boxed{\circ}) \rightarrow (\boxed{\circ}, \boxed{\circ\circ}) \rightarrow$$

Свойства трасс

Пример: перекрёсток со светофорами

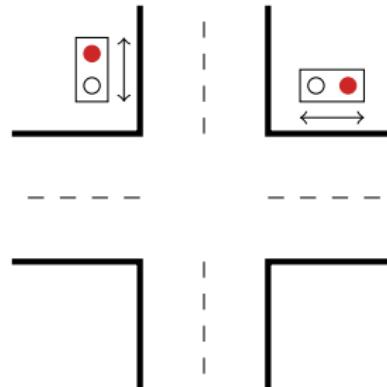


Пример вычисления этой системы:

$$(\boxed{\bullet}, \boxed{\circ}) \rightarrow (\boxed{\circ}, \boxed{\circ\circ}) \rightarrow (\boxed{\bullet}, \boxed{\circ\bullet}) \rightarrow$$

Свойства трасс

Пример: перекрёсток со светофорами

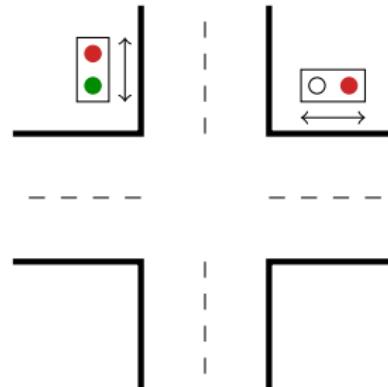


Пример вычисления этой системы:

$$([\text{red}, \text{green}], [\text{green}, \text{red}]) \rightarrow ([\text{green}, \text{green}], [\text{green}, \text{red}]) \rightarrow ([\text{green}, \text{red}], [\text{red}, \text{red}]) \rightarrow ([\text{red}, \text{red}], [\text{red}, \text{red}]) \rightarrow$$

Свойства трасс

Пример: перекрёсток со светофорами

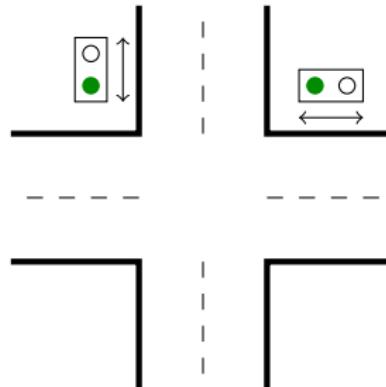


Пример вычисления этой системы:

$$(\boxed{\textcolor{red}{\circ}}, \boxed{\textcolor{green}{\circ} \textcolor{white}{\circ}}) \rightarrow (\boxed{\textcolor{white}{\circ}}, \boxed{\textcolor{white}{\circ} \textcolor{red}{\circ}}) \rightarrow (\boxed{\textcolor{red}{\circ}}, \boxed{\textcolor{white}{\circ} \textcolor{red}{\circ}}) \rightarrow (\boxed{\textcolor{red}{\circ}}, \boxed{\textcolor{red}{\circ} \textcolor{red}{\circ}}) \rightarrow (\boxed{\textcolor{green}{\circ}}, \boxed{\textcolor{red}{\circ} \textcolor{red}{\circ}}) \rightarrow$$

Свойства трасс

Пример: перекрёсток со светофорами

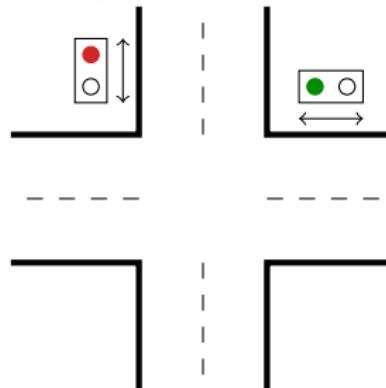


Пример вычисления этой системы:

$$(\boxed{\bullet}, \boxed{\circ}) \rightarrow (\boxed{\circ}, \boxed{\circ\bullet}) \rightarrow (\boxed{\bullet}, \boxed{\circ\bullet}) \rightarrow (\boxed{\bullet}, \boxed{\bullet\circ}) \rightarrow (\boxed{\bullet}, \boxed{\circ\bullet}) \rightarrow (\boxed{\circ}, \boxed{\bullet\circ}) \rightarrow \dots$$

Свойства трасс

Пример: перекрёсток со светофорами

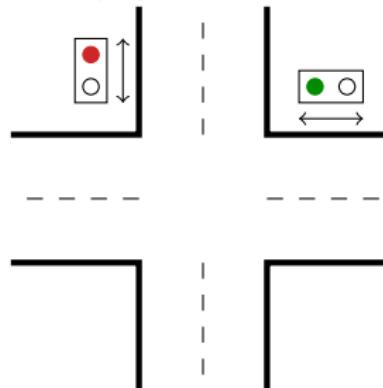


Какие **требования** было бы разумно предъявить к этой системе:

- ▶ Никакой светофор никогда не может быть ● и ● одновременно
- ▶ Сколько бы ни выполнялась система, каждый светофор рано или поздно ещё хотя бы раз станет ●
- ▶ Никогда светофоры не будут ● одновременно
- ▶ Каждый светофор бесконечно часто бывает и ●, и ●

Свойства трасс

Пример: перекрёсток со светофорами



Модель Крипке, составленную с учётом этих требований, разумно строить над такими атомарными высказываниями:

- ▶ r_{\uparrow} : светофор \uparrow красный
- ▶ g_{\uparrow} : светофор \uparrow зелёный
- ▶ r_{\leftrightarrow} : светофор \leftrightarrow красный
- ▶ g_{\leftrightarrow} : светофор \leftrightarrow зелёный

Свойства трасс

Пример: перекрёсток со светофорами

Тогда трасса вычисления

$$(\text{red}, \text{green}) \rightarrow (\text{green}, \text{yellow}) \rightarrow (\text{red}, \text{yellow}) \rightarrow (\text{green}, \text{yellow}) \rightarrow (\text{red}, \text{yellow}) \rightarrow (\text{green}, \text{yellow}) \rightarrow \dots$$

устроена так:

$$\{r_{\uparrow\downarrow}, g_{\leftrightarrow}\}, \quad \emptyset, \quad \{r_{\uparrow\downarrow}, r_{\leftrightarrow}\}, \quad \{r_{\uparrow\downarrow}, r_{\leftrightarrow}\}, \quad \{r_{\uparrow\downarrow}, g_{\uparrow\downarrow}, r_{\leftrightarrow}\}, \quad \{g_{\uparrow\downarrow}, g_{\leftrightarrow}\}, \quad \dots$$

Свойством трасс будем называть любое множество трасс

Будем говорить, что трасса τ **обладает свойством P** , если $\tau \in P$

Например, требование «Никогда светофоры не будут \bullet одновременно»
отвечает свойству

$$\{\tau \mid \tau \in (2^{\text{AP}})^\omega, \quad \forall \sigma \in \tau : \{g_{\leftrightarrow}, g_{\uparrow\downarrow}\} \not\subseteq \sigma\},$$

и трасса, изображённая выше, не обладает этим свойством

Обозначения:

- ▶ X^ω — множество всех бесконечных последовательностей,
составленных из элементов множества X
- ▶ $x \in \Sigma$ для последовательности $\Sigma \Leftrightarrow x$ хотя бы раз встречается в Σ

Свойства трасс

Для модели Кripке $M = (S, S_0, \rightarrow, L)$ над AP и её состояния s будем использовать такие понятия и обозначения:

- ▶ $\Pi(M, s)$ — множество всех бесконечных путей M , исходящих из s
- ▶ $\Pi(M)$ — множество всех вычислений M
 - ▶ То есть $\Pi(M) = \bigcup_{s_0 \in S_0} \Pi(M, s_0)$
- ▶ $\text{Tr}(M, s)$ — множество всех трасс вычислений из $\Pi(M, s)$
- ▶ $\text{Tr}(M)$ — множество всех трасс вычислений из $\Pi(M)$
 - ▶ То есть $\text{Tr}(M) = \bigcup_{s_0 \in S_0} \text{Tr}(M, s_0)$
- ▶ Tr — множество всех трасс (для заданного множества AP)
- ▶ M удовлетворяет свойству трасс P ($M \models P$), если $\text{Tr}(M) \subseteq P$

Свойства трасс

Пояснение соотношения $M \models P$

для модели Кripке M и свойства трасс P :

- ▶ Всевозможные трассы делятся свойством P на **хорошие** (обладающие свойством P) и **плохие** (не обладающие свойством P)
- ▶ Соотношение $M \models P$ означает, что все трассы модели M **хорошие** (т.е. что в модели M нет ни одной **плохой** трассы)

Утверждение. Для любых моделей Кripке M, M' и свойства трасс P верно: если $\text{Tr}(M) \subseteq \text{Tr}(M')$ и $M' \models P$, то $M \models P$

Доказательство. Очевидным образом следует из определений и свойства транзитивности включения множеств

Свойства безопасности и живости

При анализе поведения систем зачастую рассматриваются свойства трасс двух классов:

- ▶ Свойства безопасности
 - ▶ *Safety properties*
- ▶ Свойства живости
 - ▶ *Или, по-другому, — свойства живучести*
 - ▶ *Liveness properties*

Свойства безопасности и живости

Ещё немного понятий и обозначений ($\Sigma, \Sigma', \Sigma''$ — последовательности)

- ▶ Если $\Sigma' = (x_0, \dots, x_n)$ и $\Sigma'' = (y_0, y_1, y_2, \dots)$, то
 $\Sigma'\Sigma'' = (x_0, \dots, x_n, y_0, y_1, y_2, \dots)$
- ▶ Если Σ' бесконечна и Σ'' пуста, то $\Sigma'\Sigma'' = \Sigma'$
- ▶ Σ' — префикс Σ , если существует Σ'' , такая что $\Sigma = \Sigma'\Sigma''$
- ▶ Σ — продолжение Σ' , если Σ' — префикс Σ
- ▶ X^* — множество всех конечных последовательностей элементов множества X
- ▶ Конечная трасса — это конечная последовательность событий
- ▶ Tr_f — множество всех конечных трасс (для заданного множества AP)

Свойства безопасности и живости

Свойство трасс P называется **свойством безопасности**, если у любой трассы τ , не обладающей этим свойством, существует конечный префикс, любое бесконечное продолжение которого не обладает этим свойством

$$(\forall \tau \in \text{Tr} \setminus P : \exists \tau_1 \in \text{Tr}_f : \exists \tau_2 \in \text{Tr} : \tau = \tau_1\tau_2 \text{ и } \forall \tau_3 \in \text{Tr} : \tau_1\tau_3 \notin P)$$

Пояснение

Трассы, обладающие свойством P , считаются **безопасными**, а не обладающие — **опасными**

При этом понятие **(без)опасности** подобрано так, что если трасса

$$\sigma_0 \rightarrow \sigma_1 \rightarrow \sigma_2 \rightarrow \dots$$

опасна, то существует обозримая (конечная) совокупность событий

$$\sigma_0 \rightarrow \sigma_1 \rightarrow \sigma_2 \rightarrow \dots \rightarrow \sigma_n,$$

от начала работы системы до некоторого (конечного) момента времени, по которой и все другие трассы можно признать **опасными**

То есть понятие **опасности** в свойстве P сформулировано так, что её или её отсутствие можно констатировать в каждый момент времени, и однажды наступив, **опасность** не может быть устранена

Свойства безопасности и живости

Примеры требований, отвечающих свойствам безопасности:

- ▶ Два процесса не обращаются одновременно к одной ячейке памяти
 - ▶ *Опасность:* сейчас два процесса обращаются к одной ячейке
- ▶ Пока принтер не завершит печать, он не доступен для других устройств
 - ▶ *Опасность:* сейчас принтер занят печатью для одного устройства и при этом доступен для другого
- ▶ Команда выполняется процессором не более трёх тактов подряд
 - ▶ *Опасность:* команда выполняется четыре последних такта
- ▶ Красный свет загорается только после жёлтого
 - ▶ *Опасность:* раньше жёлтый свет не загорался, а сейчас горит красный

Свойства безопасности и живости

Свойство трасс P называется **свойством живости**, если для любой конечной трассы существует бесконечное продолжение, обладающее этим свойством

$$(\forall \tau_1 \in \text{Tr}_f : \exists \tau_2 \in \text{Tr} : \tau_1 \tau_2 \in P)$$

Пояснение

Определение живости можно прочитать так:
как бы ни работала система, обязательно есть возможность ей выполнять дальше так, чтобы она была признана **живой** (не сломавшейся, не зависшей, не отключившейся, ...)

Способ продолжения произвольной трассы до входящей в P — это способ **подтверждения живости**, не зависящий от истории событий до текущего момента и задающийся в терминах как конечного, так и бесконечного продолжения работы системы

Мёртвая в этом смысле система — это такая, которая после выполнения некоторых действий оказалась неспособной ни при каких обстоятельствах подтвердить свою **живость**

Свойства безопасности и живости

Примеры требований, отвечающих свойствам живости:

- ▶ Рано или поздно загорится зелёный свет
 - ▶ *Мёртвая система*: невозможно зажечь зелёный свет
- ▶ После завершения печати принтер стирает содержимое буфера
 - ▶ *Мёртвая система*: невозможно добиться пустого буфера
- ▶ Рано или поздно наступит следующий такт работы процессора
 - ▶ *Мёртвая система*: тактовый сигнал перестал осциллировать
- ▶ Процесс бесконечно часто обращается к заданной ячейке памяти
 - ▶ *Мёртвая система*: от процесса можно добиться только конечного числа обращений к ячейке

Свойства безопасности и живости

Утверждение. Если свойство трасс P является и свойством безопасности, и свойством живости, то $P = \text{Tr}$

Доказательство. Можете попробовать самостоятельно, это нетрудно получить из определений

Утверждение. Для любого свойства трасс P существуют такие свойство безопасности P_s и свойство живости P_l , для которых верно $P = P_s \cap P_l$

Доказательство. Можете попробовать самостоятельно (но это трудно!)

Свойства безопасности и живости

Пример

Процесс при запуске открывает файл
и затем бесконечно часто обращается к этому файлу

Это не свойство безопасности: если файл открыт при запуске, то для этого случая невозможно подобрать подходящее понятие [опасности](#)

Это не свойство живости: если файл не открыт при запуске, то невозможно добиться выполнения этого свойства

Но это пересечение свойства безопасности

Процесс при запуске открывает файл

и свойства живости

Процесс бесконечно часто обращается к файлу