

Математические методы верификации схем и программ

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

Блок 11

Свойства трасс
Безопасность и живость

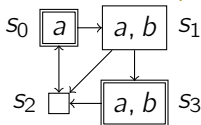
Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2024/2025, осенний семестр

Напоминание



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



Вычисление τ модели M (бесконечный начальный путь):

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

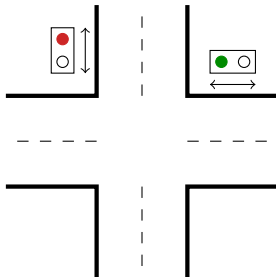
Трасса вычисления τ :

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

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

Свойства трасс

Пример: перекрёсток со светофорами

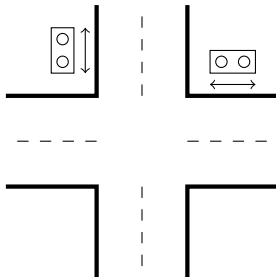


Пример вычисления этой системы:

$$\left(\begin{array}{|c|} \hline \bullet \\ \hline \circ \\ \hline \end{array}, \begin{array}{|c|c|} \hline \bullet & \circ \\ \hline \end{array} \right) \rightarrow$$

Свойства трасс

Пример: перекрёсток со светофорами

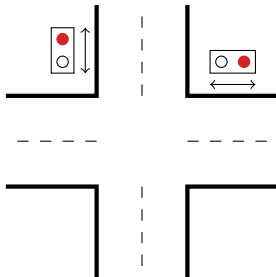


Пример вычисления этой системы:

$$\left(\begin{array}{|c|} \hline \bullet \\ \hline \circ \\ \hline \end{array}, \begin{array}{|c|c|} \hline \bullet & \circ \\ \hline \end{array} \right) \rightarrow \left(\begin{array}{|c|} \hline \circ \\ \hline \circ \\ \hline \end{array}, \begin{array}{|c|c|} \hline \circ & \circ \\ \hline \end{array} \right) \rightarrow$$

Свойства трасс

Пример: перекрёсток со светофорами

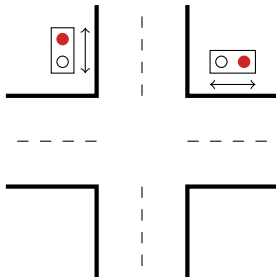


Пример вычисления этой системы:

$$\left(\begin{array}{|c|} \hline \bullet \\ \hline \circ \\ \hline \end{array}, \begin{array}{|c|} \hline \bullet \\ \hline \circ \\ \hline \end{array} \right) \rightarrow \left(\begin{array}{|c|} \hline \circ \\ \hline \circ \\ \hline \end{array}, \begin{array}{|c|} \hline \circ \\ \hline \circ \\ \hline \end{array} \right) \rightarrow \left(\begin{array}{|c|} \hline \bullet \\ \hline \circ \\ \hline \end{array}, \begin{array}{|c|} \hline \circ \\ \hline \bullet \\ \hline \end{array} \right) \rightarrow$$

Свойства трасс

Пример: перекрёсток со светофорами

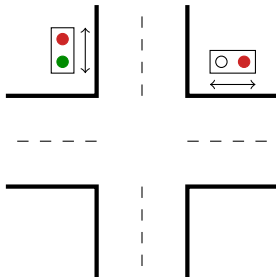


Пример вычисления этой системы:



Свойства трасс

Пример: перекрёсток со светофорами

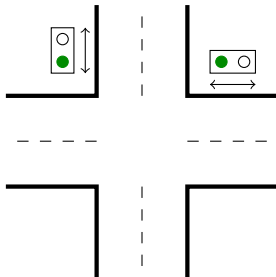


Пример вычисления этой системы:



Свойства трасс

Пример: перекрёсток со светофорами

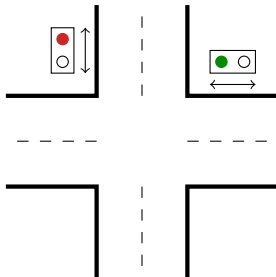


Пример вычисления этой системы:

$$\begin{pmatrix} \text{red} \\ \text{green} \end{pmatrix}, \begin{pmatrix} \text{green} \end{pmatrix} \rightarrow \begin{pmatrix} \text{red} \\ \text{green} \end{pmatrix}, \begin{pmatrix} \text{green} \end{pmatrix} \rightarrow \begin{pmatrix} \text{red} \\ \text{green} \end{pmatrix}, \begin{pmatrix} \text{red} \end{pmatrix} \rightarrow \begin{pmatrix} \text{red} \\ \text{green} \end{pmatrix}, \begin{pmatrix} \text{red} \end{pmatrix} \rightarrow \begin{pmatrix} \text{red} \\ \text{green} \end{pmatrix}, \begin{pmatrix} \text{red} \end{pmatrix} \rightarrow \begin{pmatrix} \text{red} \\ \text{green} \end{pmatrix}, \begin{pmatrix} \text{red} \end{pmatrix} \rightarrow \dots$$

Свойства трасс

Пример: перекрёсток со светофорами

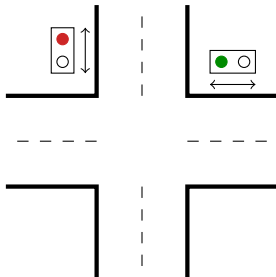


Какие **требования** было бы разумно предъявить к этой системе:

- ▶ 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:

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

Свойства трасс

Пример: перекрёсток со светофорами

Тогда трасса вычисления



устроена так:

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

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 ● одновременно» отвечает свойству

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

и трасса, изображённая выше, не обладает этим свойством

Свойства трасс

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

- ▶ $\Pi(M, s)$ — множество всех бесконечных путей в 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 путей 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)
- ▶ Tr_f — множество всех конечных трасс
- ▶ M удовлетворяет свойству трасс P ($M \models P$), если $\text{Tr}(M) \subseteq P$

Свойства трасс

Пояснение соотношения $M \models P$

для модели Крипке 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 Крипке 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 из 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

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

$$\sigma_1 \rightarrow \sigma_2 \rightarrow \sigma_3 \rightarrow \dots$$

опасна, то существует обозримая (конечная) совокупность событий

$$\sigma_1 \rightarrow \sigma_2 \rightarrow \sigma_3 \rightarrow \dots \rightarrow \sigma_n,$$

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

То есть **опасность**, однажды наступив, не может быть устранена дальнейшими событиями

Свойства безопасности и живости

Примеры требований, отвечающих свойствам безопасности:

- ▶ Два процесса не обратятся 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 прочесть так:

как бы ни работала система, обязательно есть возможность ей выполняться 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* больше не может зажечь зелёный свет
- ▶ После завершения печати принтер стирает содержимое буфера
 - ▶ *Мёртвая система* ни при каких условиях не опустошит буфер
- ▶ Рано или поздно наступит следующий такт работы процессора
 - ▶ *Мёртвая система* потеряла возможность осциллировать тактовым сигналом
- ▶ Процесс бесконечно часто обращается к заданной ячейке памяти
 - ▶ *Мёртвая система* может обратиться к ячейке памяти лишь конечное число раз

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 не очень трудно

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

Доказательство.

Можете попробовать самостоятельно (и это может быть трудно)

Свойства безопасности и живости

Пример

**Процесс при запуске открывает файл
и затем бесконечно часто обращается к этом 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 позже невозможно сделать его «открытым при запуске»

Но это пересечение свойства безопасности

Процесс при запуске открывает файл
и СВОЙСТВА ЖИВОСТИ

Процесс бесконечно часто обращается к файлу