Математические методы верификации схем и программ

Лекторы:

Захаров Владимир Анатольевич

Подымов Владислав Васильевич

e-mail рассказчика:

valdus@yandex.ru

Осень 2018

Лекция 7

Задача model checking для LTL

Алгоритмы верификации LTL-формул: табличный алгоритм, автоматный алгоритм

Автоматы Бюхи, их свойства и обобщения

LTL-формулы (напоминание и новые определения)

AP — множество атомарных высказываний, полагающееся заданным во всех дальнейших определениях

В лекции используется синтаксисс LTL-формул, задаваемый следующей формой Бэкуса-Наура:

$$\varphi \quad ::= \quad a \mid (\varphi \,\&\, \varphi) \mid (\neg\varphi) \mid (\mathbf{X}\varphi) \mid (\varphi \mathbf{U} \varphi),$$

где φ — LTL-формула и $a \in AP$

Также в записи LTL-формул могут использоваться связки $\vee$, $\rightarrow$ и операторы **F**, **G**, **R**:

$$\varphi \vee \psi \equiv \neg(\neg\varphi \,\&\, \neg\psi) \qquad \varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$$

$$\mathbf{F}\varphi \equiv \mathbf{true}\mathbf{U}\varphi \qquad \mathbf{G}\varphi \equiv \neg\mathbf{F}\neg\varphi \qquad \varphi\mathbf{R}\psi \equiv \neg(\neg\varphi\mathbf{U}\neg\psi)$$

В лекции 4 все LTL-формулы начинались с квантора пути **A**

В этой лекции квантор **A** избыточен и не записывается явно

LTL-формулы (напоминание и новые определения)

Событие — это подмножество множества AP

Трасса τ — это бесконечная последовательность событий

$\tau[i]$ — i-е событие трассы τ (события нумеруются с ноля)

τ^i — суффикс трассы τ, начинающийся с i-го события

Формула φ выполняется на трассе τ ($\tau \models \varphi$) в следующих случаях:

- $\tau \models a$, где $a \in AP \quad \Leftrightarrow \quad a \in tr[0]$
- $\tau \models \psi_1 \,\&\, \psi_2 \quad \Leftrightarrow \quad \tau \models \psi_1$ и $\tau \models \psi_2$
- $\tau \models \mathbf{X}\psi \quad \Leftrightarrow \quad \tau^1 \models \chi$
- $\tau \models \psi_1 \mathbf{U} \psi_2 \quad \Leftrightarrow \quad$ существует k, $k \geq 0$, такое что:
  - $\tau^k \models \psi_2$
  - $\tau^m \models \psi_1$ для каждого m, такого что $0 \leq m < k$

$Tr(\varphi)$ — множество всех трасс τ, таких что $\tau \models \varphi$

(*свойство трасс*, задаваемое формулой φ)

LTL-формулы (напоминание и новые определения)

Примеры свойств, задаваемых LTL-формулами:

1. Когда-нибудь цель будет достигнута
$$\mathbf{F}\mathit{goal}$$
2. Когда-нибудь наступит осень, а до тех пор будет тепло
$$\mathit{warm}\,\mathbf{U}\,\mathit{autumn}$$
3. Двух подряд плохих дней не бывает
$$\mathbf{G}(\mathit{bad_day} \to \mathbf{X}\neg\mathit{bad_day})$$
4. Если запрос посылается бесконечно часто,
то хотя бы раз появится отклик
$$\mathbf{GF}\mathit{request} \to \mathbf{F}\mathit{response}$$
5. Мне уготована вечность в раю или в аду
$$\mathbf{F}(\mathbf{G}\mathit{heaven} \vee \mathbf{G}\mathit{hell})$$

Модели Крипке (напоминание)

Модель Крипке — это система $M = (S, S_0, \rightarrow, L)$, где

- S — конечное множество состояний
- $S_0 \subseteq S$ — множество начальных состояний
- $\rightarrow \subseteq S \times S$ — **тотальное** отношение переходов
- $L : S \rightarrow 2^{AP}$ — функция разметки состояний событиями

Отношение переходов тотально, если для любого состояния $s \in S$ существует состояние $s' \in S$, такое что $R(s, s')$

Путь π из состояния s в M — это бесконечная последовательность состояний вида

$$s \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$$

Трасса $\alpha(\pi)$ пути π — это бесконечная последовательность событий вида

$$L(s), L(s_1), L(s_2), \dots$$

$\Pi(M)$ — множество всех путей из начальных состояний M

$Tr(M) = \{\alpha(\pi) \mid \pi \in \Pi(M)\}$

Задача model checking для LTL (напоминание)

Рассмотрим LTL-формулу φ и модель Крипке M

Формула φ выполняется на пути π в M ($M, \pi \models \varphi$), если

$$\alpha(\pi) \models \varphi$$

Формула φ выполняется на модели M ($M \models \varphi$), если она выполняется на каждом пути π множества $Tr(M)$, то есть если

$$Tr(M) \subseteq Tr(\varphi)$$

Постановка задачи model checking для LTL:

для заданной модели Крипке M
и заданной LTL-формулы φ
проверить справедливость соотношения

$$M \models \varphi$$

Табличный алгоритм model checking для LTL

Табличный алгоритм решения задачи model checking для LTL обсуждался в курсе "*Математическая логика и логическое программирование*" бакалавриата

Так как это слушали давно и не все, далее приводится описание табличного алгоритма

Так как этот курс всё-таки был, в целях экономии времени все утверждения, относящиеся к табличному алгоритму, приводятся без доказательства

Начало рассуждений

Без ограничения общности полагаем, что φ — формула без двойных отрицаний: не содержит подформул вида $\neg\neg\psi$

$(\neg\neg\psi \equiv \psi)$

Предположим, что $M \not\models \varphi$

Тогда в M существует путь π из начального состояния, такой что $M, \pi \not\models \varphi$

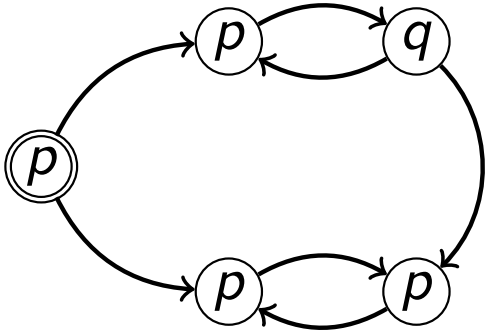
Попробуем доказать или опровергнуть наличие такого пути π

Табличный алгоритм model checking для LTL

Краткое описание алгоритма

Иллюстрация

φ: $p\mathbf{U}q$ M:



Табличный алгоритм model checking для LTL

Краткое описание алгоритма

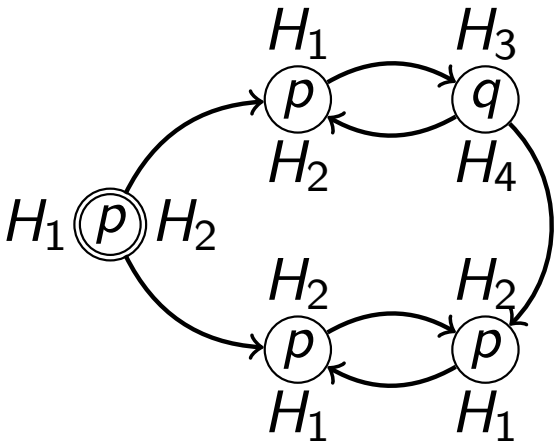
Состояния модели M размечаются **гипотезами** — особыми конечными множествами формул, содержащими одну из формул

$$\varphi, \qquad \neg\varphi, \qquad \psi \text{ (если } \varphi = \neg\psi)$$

Каждое состояние может быть помечено любым числом гипотез

Иллюстрация

φ: $p\mathbf{U}q$



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

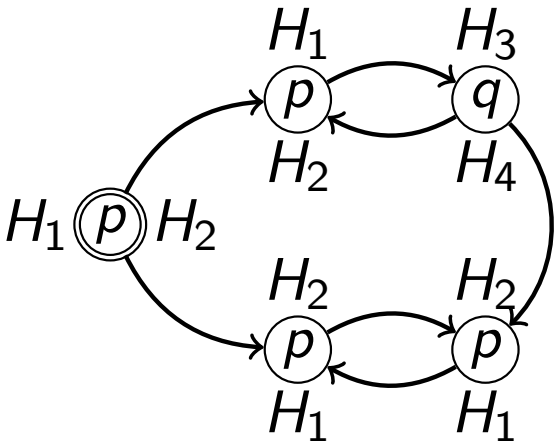
Табличный алгоритм model checking для LTL

Краткое описание алгоритма

Содержательная трактовка гипотезы H, помечающей состояние s:
возможно, что в M из состояния s исходит путь, на трассе τ которого выполнены все формулы из H $(\tau \models H)$

Иллюстрация

φ: $p\mathbf{U}q$



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

Табличный алгоритм model checking для LTL

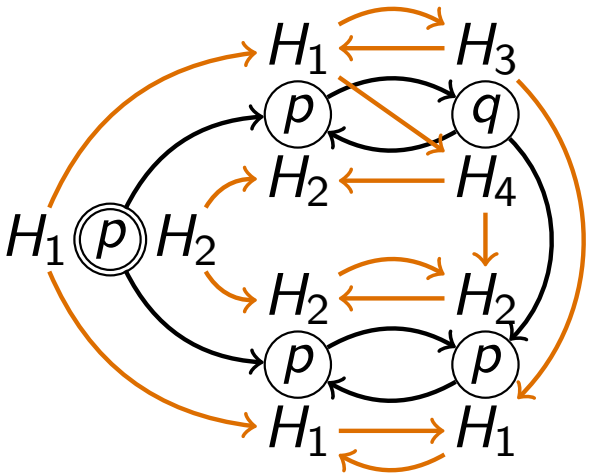
Краткое описание алгоритма

Некоторые пары (состояние, гипотеза) соединяются дугами

Содержательная трактовка дуги $(s_1, H_1) \to (s_2, H_2)$:
возможно, что в M содержится путь $s_1 \to s_2 \to \ldots$, для трассы τ которого справедливы соотношения $\tau \models H_1$ и $\tau^1 \models H_2$

Иллюстрация

φ: $p\mathbf{U}q$



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

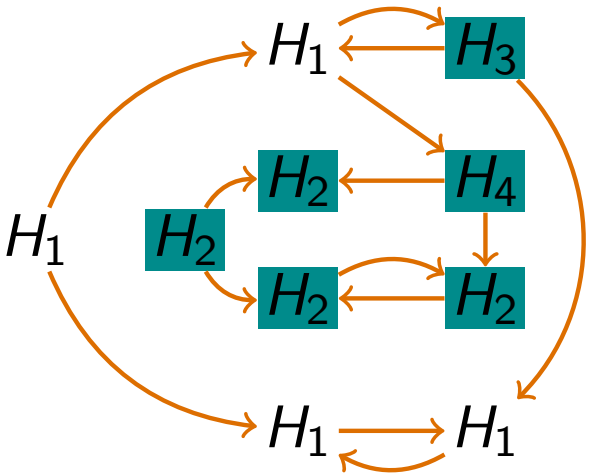
Табличный алгоритм model checking для LTL

Краткое описание алгоритма

В получившемся графе соединения пар (состояние, гипотеза) — **системе Хинтикки** — рассматриваются бесконечные пути, на которых **действительно** выполняются все формулы соответствующих гипотез

Иллюстрация

φ: $p\mathbf{U}q$



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

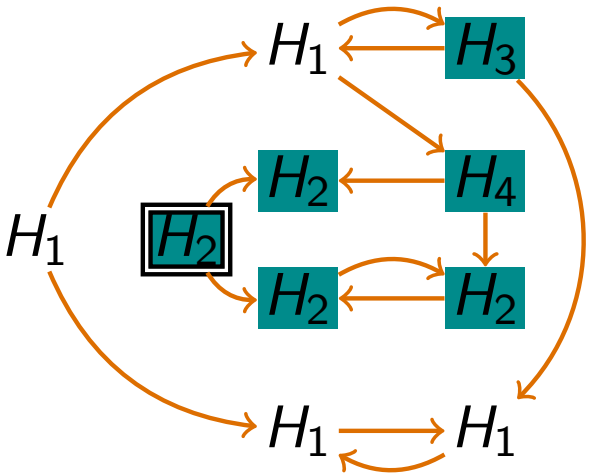
Табличный алгоритм model checking для LTL

Краткое описание алгоритма

$M \not\models \varphi$ тогда и только тогда, когда существует рассматриваемый путь в системе Хинтикки, начинающийся с пары (s, H), где s — начальное состояние M и $\varphi \notin H$

Иллюстрация

φ: $p\mathbf{U}q$



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

Табличный алгоритм model checking для LTL

Формулы вида $\neg\psi$ далее называются негативными,
а все остальные — позитивными

Замыкание Фишера-Ладнера $[\varphi]_{fl}$ формулы φ — это
множество формул, содержащее в точности:

1. все позитивные подформулы формулы φ
2. $\mathbf{X}(\psi\mathbf{U}\chi)$ для каждой подформулы вида $\psi\mathbf{U}\chi$ формулы φ

Пример: $[\neg(p\mathbf{U}\neg q)]_{fl} = \{p, q, p\mathbf{U}\neg q, \mathbf{X}(p\mathbf{U}\neg q)\}$

Гипотеза (для формулы φ) — это множество формул вида

$$F \cup \{\neg\psi \mid \chi \in [\varphi]_{fl} \setminus F\},$$

где $F \subseteq [\varphi]_{fl}$

Пример: $\{\neg p, \neg q, p\mathbf{U}\neg q, \neg\mathbf{X}(p\mathbf{U}\neg q)\}$ — гипотеза для $\neg(p\mathbf{U}\neg q)$

Табличный алгоритм model checking для LTL

Гипотеза H совместна, если
для любых формул $\psi_1 \,\&\, \psi_2$ и $\chi_1 \mathbf{U} \chi_2$ из $[\varphi]_{fl}$ верно следующее:

1. $\psi_1 \,\&\, \psi_2 \in H \quad \Leftrightarrow \quad \{\psi_1, \psi_2\} \subseteq H$
2. $\chi_1 \mathbf{U} \chi_2 \in H \quad \Leftrightarrow \quad \chi_2 \in H$ или $\{\chi_1, \mathbf{X}(\chi_1 \mathbf{U} \chi_2)\} \subseteq H$

Пояснение: $\chi_1 \mathbf{U} \chi_2 = \chi_2 \vee \chi_1 \,\&\, \mathbf{X}(\chi_1 \mathbf{U} \chi_2)$
(*закон неподвижной точки для* $\mathbf{U}$, *отражённый в пункте 2*)

Пример:

- $\{\neg p, q, \neg \mathbf{X}(p \mathbf{U} q), \quad p \mathbf{U} q\ \}$ — совместная гипотеза
- $\{\neg p, q, \quad \mathbf{X}(p \mathbf{U} q), \neg(p \mathbf{U} q)\}$ — несовместная гипотеза

$[\varphi]_{fl}^{-} = \{\neg\psi \mid \psi \in [\varphi]_{fl}\}$

H_τ^i — множество всех формул ψ из $[\varphi]_{fl} \cup [\varphi]_{fl}^{-}$, таких что $\tau^i \models \psi$
(τ — *трасса*, $i \geq 0$)

Утверждение. **Для любой трассы τ и любого индекса i, $i \geq 0$, множество H_τ^i является совместной гипотезой**

Табличный алгоритм model checking для LTL

Гипотезы H_1, H_2 локально согласованны, если для любой формулы вида $\mathbf{X}\chi$ из $[\varphi]_{fl}$ верно: $\mathbf{X}\chi \in H_1 \Leftrightarrow \chi \in H_2$

Утверждение. **Для любой трассы τ и любого индекса i, $i \geq 0$, гипотезы H_τ^i и H_τ^{i+1} локально согласованны**

Гипотеза H завершает формулу $\psi\mathbf{U}\chi$ из $[\varphi]_{fl}$, если выполнено хотя бы одно из двух условий:

1. $\chi \in H$
2. $\mathbf{X}(\psi\mathbf{U}\chi) \notin H$

Бесконечная последовательность гипотез $\mathfrak{H} = (H_0, H_1, \dots)$ глобально согласованна, если для каждой формулы вида $\psi\mathbf{U}\chi$ из $[\varphi]_{fl}$ в $\mathfrak{H}$ бесконечно часто встречаются гипотезы, завершающие эту формулу

Утверждение. **Для любой трассы τ последовательность гипотез $H_\tau^0, H_\tau^1, H_\tau^2, \dots$ глобально согласованна**

Табличный алгоритм model checking для LTL

Если $\mathfrak{H} = H_0, H_1, H_2, \dots$ — бесконечная последовательность гипотез, то $\tau_{\mathfrak{H}}$ — трасса вида $H_0 \cap AP, H_1 \cap AP, H_2 \cap AP, \dots$

Утверждение. **Для любой бесконечной глобально согласованной последовательности совместных гипотез $\mathfrak{H} = (H_0, H_1, H_2, \dots)$, такой что каждая пара гипотез H_j, H_{j+1} локально согласованна, и для каждого индекса i, $i \geq 0$, справедливо соотношение $H_i = H^i_{\tau_{\mathfrak{H}}}$**

Контрпример:

- гипотеза $H = \{p, \neg q, \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$ совместна
- пара гипотез H, H локально согласованна
- бесконечная последовательность гипотез $H, H, H, \dots$ **не является** глобально согласованной
- $\tau_{H,H,H,\dots} = (\{p\}, \{p\}, \{p\}, \dots) \not\models p\mathbf{U}q$

Табличный алгоритм model checking для LTL

Гипотеза H адекватна состоянию s модели крипке $M = (S, S_0, \to, L)$, если $L(s) = H \cap AP$

Система Хинтикки $\mathcal{HS}(M, \varphi)$ для модели Крипке M и LTL-формулы φ — это размеченный ориентированный граф следующего вида:

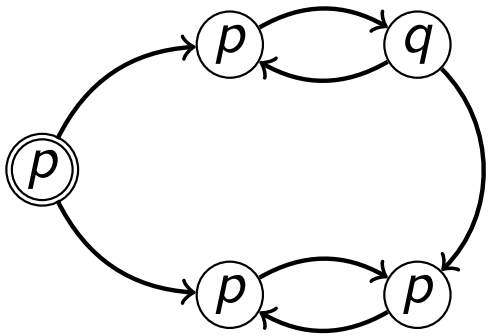
- вершинами являются всевозможные пары (s, H), где $s \in S$ и H — совместная гипотеза, адекватная состоянию s
- дуга $(s_1, H_1) \to (s_2, H_2)$ содержится в графе $\Leftrightarrow$ $s_1 \to s_2$ и гипотезы H_1, H_2 локально согласованны
- вершина (s, H) является начальной $\Leftrightarrow$ $s \in S_0$ и $\varphi \notin H$
- каждой формуле вида $\psi \mathbf{U} \chi$ из $[\varphi]_{fl}$ сопоставлен уникальный цвет $c_{\psi \mathbf{U} \chi}$
- вершина (s, H) помечена цветом c_Φ $\Leftrightarrow$ гипотеза H завершает формулу Φ

Табличный алгоритм model checking для LTL

Пример

φ: $p\mathbf{U}q$

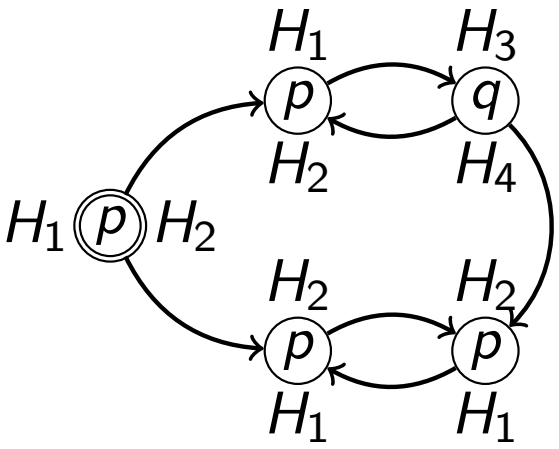
M:



Табличный алгоритм model checking для LTL

Пример

$\varphi\colon p\mathbf{U}q$



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

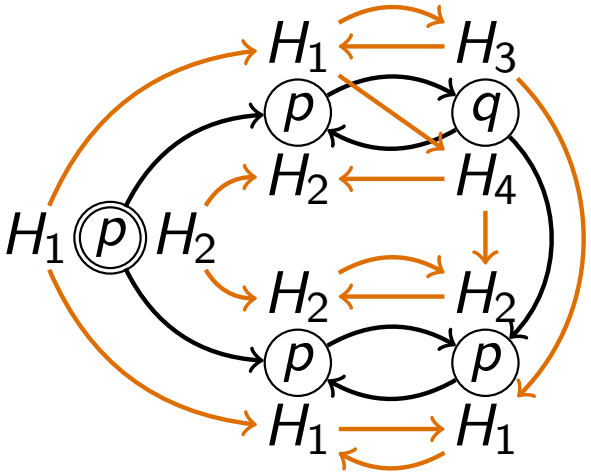
H_1 и H_2 — все совместные гипотезы, адекватные состоянию, помеченному множеством $\{p\}$

H_3 и H_4 — все совместные гипотезы, адекватные состоянию, помеченному множеством $\{q\}$

Табличный алгоритм model checking для LTL

Пример

φ: $p\mathbf{U}q$



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

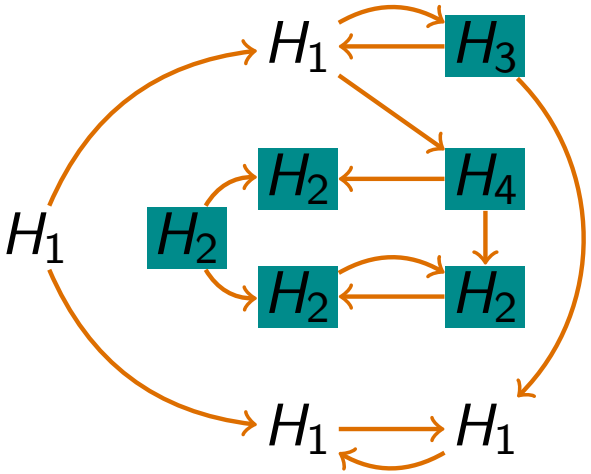
Локально согласованная пара гипотез H_i, H_j:

$$\mathbf{X}(p\mathbf{U}q) \in H_i \Leftrightarrow p\mathbf{U}q \in H_j$$

Табличный алгоритм model checking для LTL

Пример

φ: $p\mathbf{U}q$



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

Гипотезы H_3, H_4 завершают формулу $p\mathbf{U}q$: $q \in H_3 \cap H_4$

Гипотеза H_2 завершает формулу $p\mathbf{U}q$: $\mathbf{X}(p\mathbf{U}q) \notin H_2$

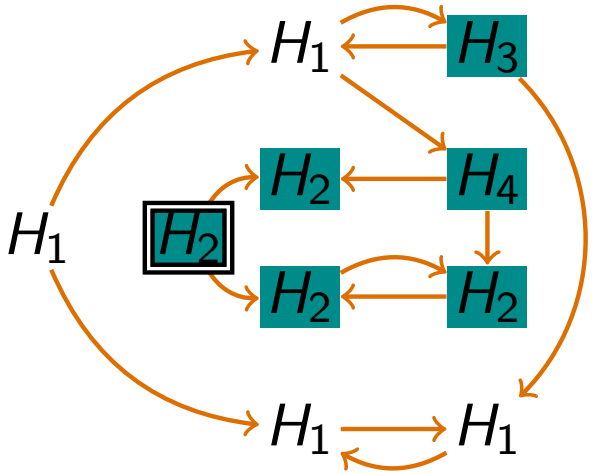
Гипотеза H_1 **не** завершает формулу $p\mathbf{U}q$:

- $q \notin H_1$
- $\mathbf{X}(p\mathbf{U}q) \in H_1$

Табличный алгоритм model checking для LTL

Пример

$\varphi\colon p\mathbf{U}q$ $\qquad$ $\mathcal{HS}(M,\varphi)\colon$



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

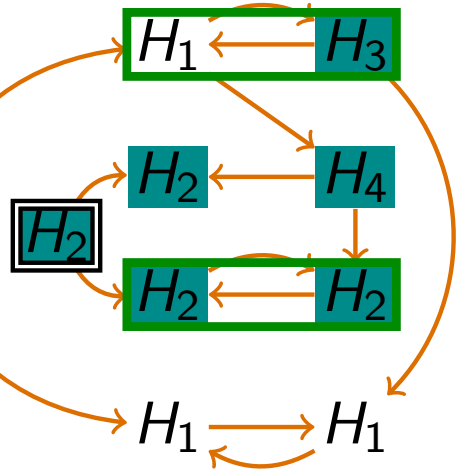
Начальная вершина (s, H):

- s — начальное состояние модели M
- $p\mathbf{U}q \notin H$

Табличный алгоритм model checking для LTL

Пример

φ: $p\mathbf{U}q$ $\qquad$ $\mathcal{HS}(M,\varphi)$:



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

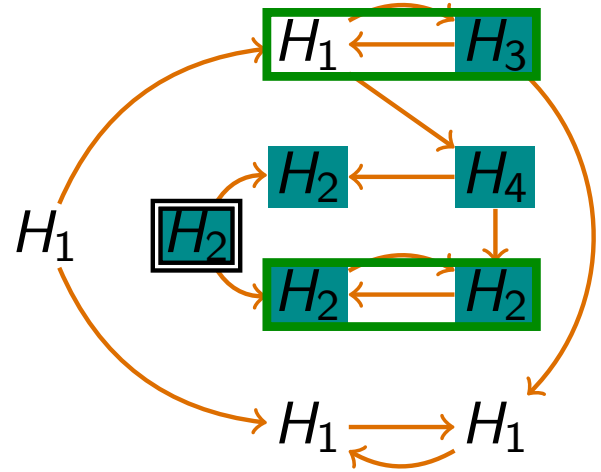
Теорема. $M \not\models \varphi \Leftrightarrow$
в $\mathcal{H}(M,\varphi)$ содержится бесконечный путь из начальной вершины, в котором каждый цвет встречается бесконечно часто $\Leftrightarrow$
в $\mathcal{H}(M,\varphi)$ содержится начальная вершина, из которой достижима нетривиальная компонента сильной связности, в вершинах которой встречаются все цвета

Табличный алгоритм model checking для LTL

Пример

φ: $p\mathbf{U}q$

$\mathcal{HS}(M,\varphi)$:



$$H_1 = \{p, \neg q, \quad \mathbf{X}(p\mathbf{U}q), \quad p\mathbf{U}q\ \} \qquad H_3 = \{\neg p, q, \quad \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q), \neg(p\mathbf{U}q)\} \qquad H_4 = \{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

Итог: $M \not\models \varphi$

Интерлюдия

Табличный алгоритм model checking для CTL сам по себе неэффективен, но идеи табличного алгоритма лежат в основе эффективного *символьного алгоритма*

В ядре средства верификации для CTL *скорее всего* реализована вариация символьного алгоритма

Табличный алгоритм model checking для LTL тоже неэффективен сам по себе, но предоставляет набор понятий и методов, использующихся в эффективных аналогах этого алгоритма

Один из таких аналогов, *часто* так или иначе использующийся в ядрах средств верификации — это автоматный алгоритм

Общая схема автоматного алгоритма

Дано: модель Крипке M и LTL-формула φ

Требуется: проверить справедливость соотношения $M \models \varphi$

Схема проверки соотношения:

1. по модели M строится автомат A_M, распознающий множество (**бесконечных**!) трасс $Tr(M)$
2. по формуле φ строится автомат $A_{\neg\varphi}$, распознающий множество трасс $Tr(\neg\varphi)$
3. по автоматам A_M и $A_{\neg\varphi}$ строится автомат A, распознающий множество трасс $Tr(M) \cap Tr(\neg\varphi)$
4. проверяется, распознаёт ли автомат A **пустое** множество трасс:
   - если да, то $M \models \varphi$
   - если нет, то $M \not\models \varphi$

Автоматы Бюхи

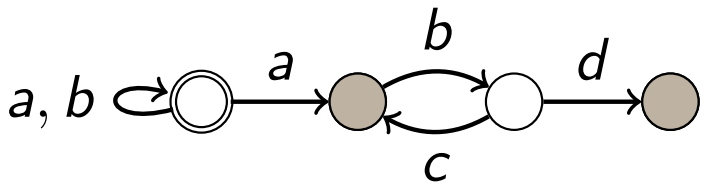
Автомат Бюхи (над алфавитом Σ) — это система $A = (S, S_0, \rightarrow, F)$, где:

- S — конечное множество состояний
- $S_0 \subseteq S$ — множество начальных состояний
- $\rightarrow \subseteq S \times \Sigma \times S$ — отношение переходов
- $F \subseteq S$ — множество заключительных состояний

$s \xrightarrow{\sigma_1,\ldots,\sigma_k} s'$ — обозначение семейства переходов

$$(s, \sigma_1, s'), \qquad \ldots, \qquad (s, \sigma_k, s')$$

Пример



◎ — начальное состояние

● — заключительное состояние

Автоматы Бюхи

Трасса автомата Бюхи $A = (S, S_0, \to, F)$ (порождаемая бесконечным словом $\sigma_1\sigma_2\sigma_3\ldots$) — это любая бесконечная последовательность состояний автомата вида

$$s_0 \xrightarrow{\sigma_1} s_1 \xrightarrow{\sigma_2} s_2 \xrightarrow{\sigma_3} \ldots,$$

такая что $s_0 \in S_0$

$Tr(A, w)$ — множество всех трасс автомата A, порождаемых бесконечным словом w

$inf(\tau)$ — множество состояний, встречающихся бесконечно часто в трассе τ автомата A

Трасса τ автомата A — принимающая, если в ней бесконечно часто повторяется хотя бы одно заключительное состояние:

$$inf(\tau) \cap F \neq \emptyset$$

Бесконечное слово w принимается автоматом A, если хотя бы одна трасса, порождаемая этим словом, является принимающей:

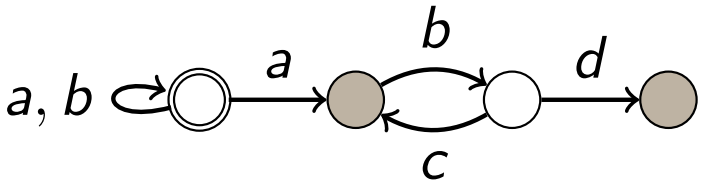
$$\exists \tau \in Tr(A, w) : inf(\tau) \cap F \neq \emptyset$$

$L(A)$ — язык автомата A:
множество всех бесконечных слов, принимаемых A

Автоматы Бюхи

Пример

Рассмотрим такой автомат A:



Язык этого автомата:

$$L(A) = \{Wabcbcbc \ldots bc \cdots \mid W \in \{a, b\}^*\}$$

Автоматы Бюхи и модели Крипке

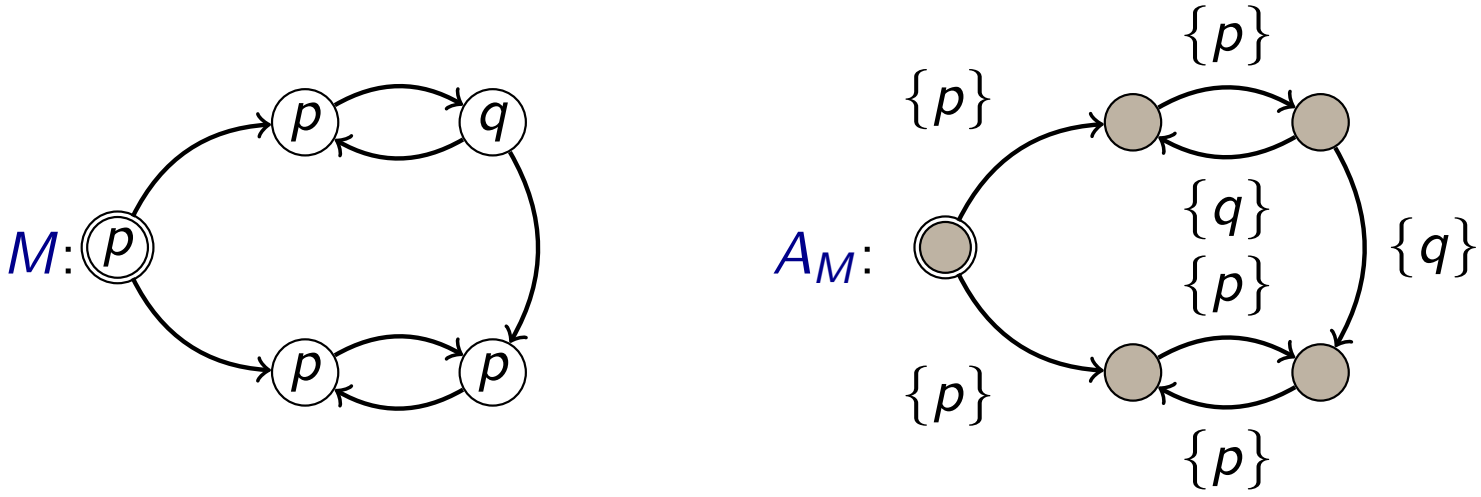
Дано: модель Крипке $M = (S, S_0, \rightarrow, L)$

Требуется: построить автомат Бюхи $A_M = (S', S'_0, \mapsto, F)$ над алфавитом 2^{AP}, такой что $L(A_M) = Tr(M)$

Решение:

- $S' = F = S$
- $S'_0 = S_0$
- $s_1 \overset{\sigma}{\mapsto} s_2 \Leftrightarrow s_1 \rightarrow s_2$ и $\sigma = L(s_1)$

Пример

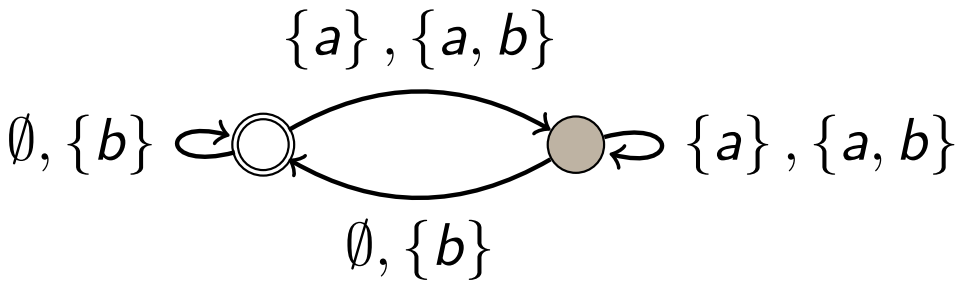


Автоматы Бюхи и LTL-формулы

Дано: LTL-формула φ

Требуется: построить автомат Бюхи $A_\varphi = (S, S_0, \to, F)$ над алфавитом 2^{AP}, такой что $L(A_\varphi) = Tr(\varphi)$

Начнём с примеров $(AP = \{a, b\})$



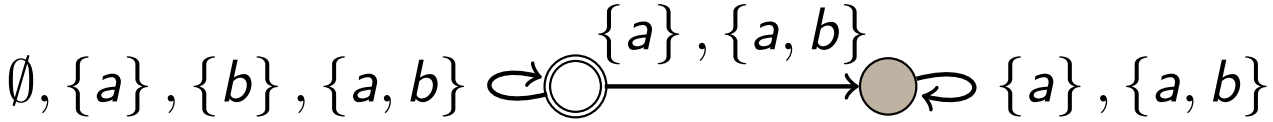
$$L(A) = Tr(\mathbf{GF}a)$$

Автоматы Бюхи и LTL-формулы

Дано: LTL-формула φ

Требуется: построить автомат Бюхи $A_\varphi = (S, S_0, \to, F)$ над алфавитом 2^{AP}, такой что $L(A_\varphi) = Tr(\varphi)$

Начнём с примеров $(AP = \{a, b\})$



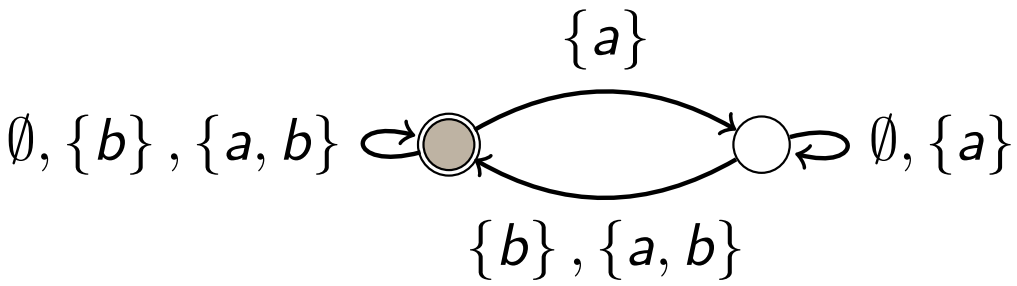
$$L(A) = Tr(\mathbf{FG}a)$$

Автоматы Бюхи и LTL-формулы

Дано: LTL-формула φ

Требуется: построить автомат Бюхи $A_\varphi = (S, S_0, \to, F)$ над алфавитом 2^{AP}, такой что $L(A_\varphi) = Tr(\varphi)$

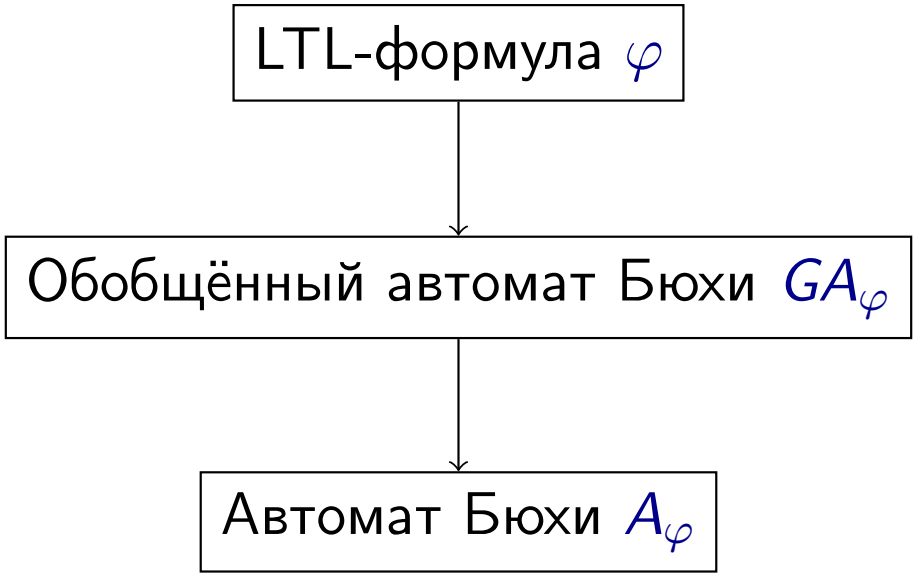
Начнём с примеров $(AP = \{a, b\})$



$$L(A) = Tr(\mathbf{G}(a \to \mathbf{F}b))$$

Автоматы Бюхи и LTL-формулы

Общая схема построения автомата Бюхи по LTL-формуле:



Обобщённые автоматы Бюхи

Обобщённый автомат Бюхи (над алфавитом Σ) — это система $GA = (S, S_0, \rightarrow, \mathcal{F})$, где

- S — конечное множество состояний
- $S_0 \subseteq S$ — множество начальных состояний
- $\rightarrow \subseteq S \times \Sigma \times S$ — отношение переходов
- $\mathcal{F} \subseteq 2^S$ — семейство заключительных множеств состояний

Все понятия и обозначения напрямую переносятся с автоматов Бюхи на обобщённые автоматы Бюхи, кроме определения *принимающей трассы*

Трасса τ обобщённого автомата GA — принимающая, если в ней бесконечно часто повторяется хотя бы одно состояние **каждого** заключительного множества:

$$\forall F \in \mathcal{F} : inf(\tau) \cap F \neq \emptyset$$

Условие принятия слова w автоматом GA переформулируется так:

$$\exists \tau \in Tr(GA, w) : \forall F \in \mathcal{F} : inf(\tau) \cap F \neq \emptyset$$

Обобщённые автоматы Бюхи

Теорема

Для любого обобщённого автомата Бюхи GA существует *необобщённый* автомат Бюхи A, такой что $L(A) = L(GA)$

Доказательство.

Пусть, для определённости, $GA = (S, S_0, \to, \mathcal{F})$

Произвольно упорядочим заключительные множества состояний:

$$\mathcal{F} = \{F_0, \ldots, F_{k-1}\}$$

Рассмотрим такой автомат Бюхи $A = (S', S'_0, \mapsto, F)$:

- $S' = \{(s, i) \mid s \in S; i \in \{0, \ldots, k-1\}\}$
- $S'_0 = \{(s, 0) \mid s \in S_0\}$
- $F = \{(s, i) \mid s \in F_i; i \in \{0, \ldots, k-1\}\}$
- $(s, i) \overset{\delta}{\mapsto} (s', i) \Leftrightarrow s \overset{\delta}{\to} s'$ и $s \notin F_i$
- $(s, i) \overset{\delta}{\mapsto} (s', (i+1) \,\mathbf{mod}\, k) \Leftrightarrow s \overset{\delta}{\to} s'$ и $s \in F_i$

Покажем, что $L(GA) = L(A)$

Обобщённые автоматы Бюхи

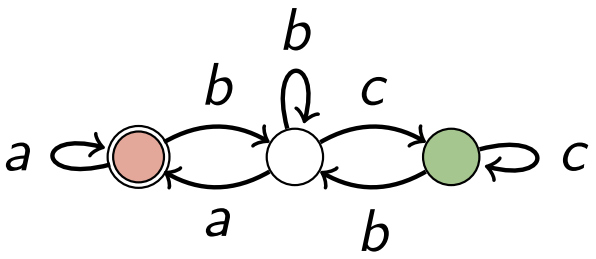
Теорема

Для любого обобщённого автомата Бюхи GA существует *необобщённый* автомат Бюхи A, такой что $L(A) = L(GA)$

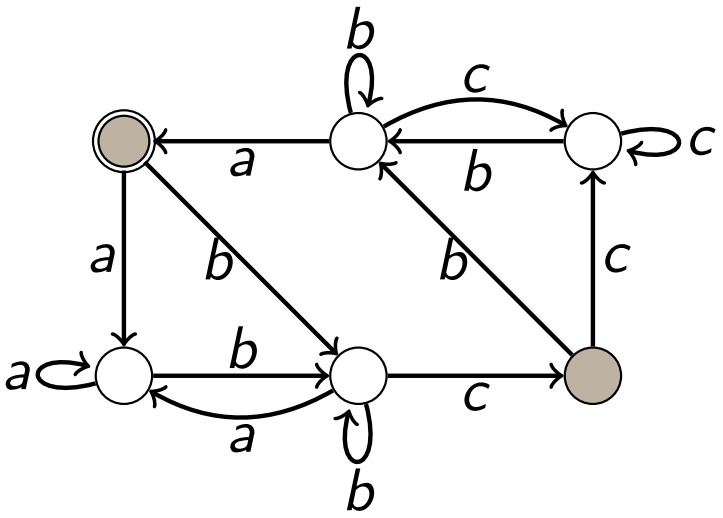
Доказательство.

Иллюстрация:

GA:



A:



Обобщённые автоматы Бюхи

Теорема

Для любого обобщённого автомата Бюхи GA существует *необобщённый* автомат Бюхи A, такой что $L(A) = L(GA)$

Доказательство. $(L(A) \subseteq L(GA))$

Рассмотрим произвольное бесконечное слово $\delta_0\delta_1\ldots$, принимаемое автоматом A, и произвольную трассу τ автомата A, принимающую это слово: $(s_0, i_0) \xmapsto{\delta_0} (s_1, i_1) \xmapsto{\delta_1} \ldots$

В τ обязательно содержится бесконечная подпоследовательность

$$(q_0, 0), (q_1, 1), \ldots, (q_{k-1}, k-1), (q_k, 0), (q_{k+1}, 1), \ldots,$$

такая что $(q_i, i \,\mathbf{mod}\, k) \in F$, $i \geq 0$

По заданию автомата A:

- $\tau' = (s_0 \xrightarrow{\delta_0} s_1 \xrightarrow{\delta_1} \ldots)$ — трасса автомата GA
- $q_i \in F_{i \,\mathbf{mod}\, k}$, а значит, в трассе τ' бесконечно часто повторяются состояния **каждого** заключительного множества

Значит, слово $\delta_0\delta_1\ldots$ принимается трассой τ'

Обобщённые автоматы Бюхи

Теорема

Для любого обобщённого автомата Бюхи GA существует *необобщённый* автомат Бюхи A, такой что $L(A) = L(GA)$

Доказательство. $(L(GA) \subseteq L(A))$

Рассмотрим произвольное бесконечное слово $\delta_0\delta_1\ldots$, принимаемое автоматом GA, и произвольную трассу τ автомата GA, принимающую это слово: $s_0 \xrightarrow{\delta_0} s_1 \xrightarrow{\delta_1} \ldots$

Выделим из τ бесконечную подпоследовательность $q_0, q_1, q_2, \ldots$ следующего вида:

- q_0 — самое первое состояние трассы τ, такое что $q_0 \in F_0$
- q_{i+1} — самое первое состояние трассы τ, следующее за q_i и такое что $q_{i+1} \in F_{i \,\mathbf{mod}\, k}$, $i \geq 0$

По заданию автомата A, слово $\delta_0\delta_1\ldots$ принимается трассой вида

$$(s_0, 0) \mapsto \cdots \mapsto (q_0, 0) \mapsto (s_{i_1}, 1) \mapsto \cdots \mapsto (q_1, 1) \mapsto \cdots \mapsto (s_{i_2}, 2) \mapsto \cdots \mapsto \cdots \mapsto (q_{k-1}, k-1) \mapsto (s_{i_k}, 0) \mapsto \cdots \mapsto (q_k, 0) \mapsto (s_{k+1}, 1) \mapsto \ldots$$

▼

Обобщённые автоматы Бюхи и LTL-формулы

Дано: LTL-формула φ

Требуется: построить **обобщённый** автомат Бюхи $GA_\varphi = (S, S_0, \to, \mathcal{F})$ над алфавитом 2^{AP}, такой что $L(GA_\varphi) = Tr(\varphi)$

Решение (*не самое лучшее, но самое простое для понимания*):

- S — множество всех совместных гипотез для формулы φ
- $H \in S_0 \Leftrightarrow \varphi \in H$
- $H_1 \xrightarrow{X} H_2 \Leftrightarrow X = H_1 \cap AP$ и пара гипотез H_1, H_2 локально согласованна
- $\mathcal{F} = \{F_1, \ldots, F_k\}$, где
  - $\psi_1, \ldots, \psi_k$ — все подформулы формулы φ вида $\chi_1 \mathbf{U} \chi_2$, перечисленные в любом порядке
  - $H \in F_i \Leftrightarrow$ гипотеза H завершает формулу ψ_i

Описанный автомат GA — это доразмеченная система Хинтикки для наименьшей модели Крипке M', содержащей всевозможные трассы ($Tr(M') = (2^{2^{AP}})^\omega$), и формулы φ, и корректность описанного решения обосновывается свойствами систем Хинтикки

Пересечение автоматов Бюхи

Теорема

Для любой пары автоматов Бюхи A', A'' существует обобщённый автомат Бюхи GA, такой что $L(GA) = L(A') \cap L(A'')$

Доказательство.

Пусть, для определённости, $A' = (S', S'_0, \to, F')$ и $A'' = (S'', S''_0, \mapsto, F'')$

Рассмотрим такой автомат $GA = (S, S_0, \Rightarrow, \mathcal{F})$

(*декартово произведение автоматов A' и A''*):

- $S = \{(s', s'') \mid s' \in S'; s'' \in S''\}$
- $S_0 = \{(s', s'') \mid s' \in S'_0; s'' \in S''_0\}$
- $(s'_1, s''_1) \overset{\delta}{\Rightarrow} (s'_2, s''_2) \Leftrightarrow s'_1 \overset{\delta}{\to} s'_2$ и $s''_1 \overset{\delta}{\mapsto} s''_2$
- $\mathcal{F} = \{F_1, F_2\}$, где
  $F_1 = \{(s', s'') \mid s' \in F'; s'' \in S''\}$ и
  $F_2 = \{(s', s'') \mid s' \in S'; s'' \in F''\}$

Покажем, что $L(GA) = L(A') \cap L(A'')$

Пересечение автоматов Бюхи

Теорема

Для любой пары автоматов Бюхи A', A'' существует обобщённый автомат Бюхи GA, такой что $L(GA) = L(A') \cap L(A'')$

Доказательство. $(L(GA) \subseteq L(A') \cap L(A''))$

Рассмотрим произвольную принимающую трассу τ автомата GA:

$$(s'_0, s''_0) \overset{\delta_0}{\Rightarrow} (s'_1, s''_1) \overset{\delta_1}{\Rightarrow} \dots$$

По заданию автомата GA:

- $\tau' = (s'_0 \xrightarrow{\delta_0} s'_1 \xrightarrow{\delta_1} \dots)$ — трасса автомата A'
- $\tau'' = (s''_0 \overset{\delta_0}{\longmapsto} s''_1 \overset{\delta_1}{\longmapsto} \dots)$ — трасса автомата A''
- τ' — принимающая трасса, так как в τ бесконечно часто повторяются состояния множества F_1
- τ'' — принимающая трасса, так как в τ бесконечно часто повторяются состояния множества F_2

Пересечение автоматов Бюхи

Теорема

Для любой пары автоматов Бюхи A', A'' существует обобщённый автомат Бюхи GA, такой что $L(GA) = L(A') \cap L(A'')$

Доказательство. $(L(A') \cap L(A'') \subseteq L(GA))$

Рассмотрим произвольные принимающие трассы τ', τ'' автоматов A', A'' соответственно:

$$s'_0 \xrightarrow{\delta_0} s'_1 \xrightarrow{\delta_1} \dots$$
$$s''_0 \xmapsto{\delta_0} s''_1 \xmapsto{\delta_1} \dots$$

По заданию автомата GA:

- $\tau = ((s'_0, s''_0) \overset{\delta_0}{\Rightarrow} (s'_1, s''_1) \overset{\delta_1}{\Rightarrow} \dots)$ — трасса автомата GA
- в τ бесконечно часто встречаются состояния множества F_1, так как τ' — принимающая трасса
- в τ бесконечно часто встречаются состояния множества F_2, так как τ'' — принимающая трасса

Значит, τ — принимающая трасса ▼

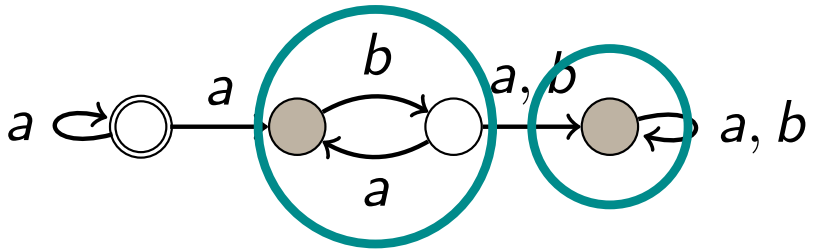
Проверка пустоты автомата Бюхи

Теорема. **Язык автомата Бюхи A непуст $\Leftrightarrow$ в A содержится начальное состояние, из которого достижима нетривиальная компонента сильной связности, содержащая хотя бы одно заключительное состояние**

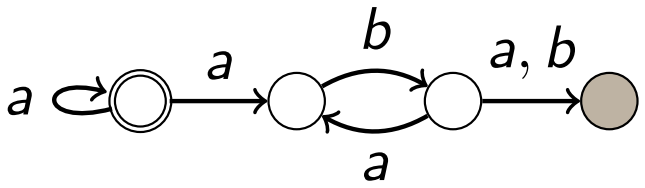
Доказательство. Очевидно?

Пример

A_1:



A_2:



$$L(A_1) \neq \emptyset$$
$$L(A_2) = \emptyset$$

Автоматный алгоритм model checking для LTL

Дано: модель Крипке M, LTL-формула φ

Требуется: проверить справедливость соотношения $M \models \varphi$

Решение:

$$
\begin{tikzcd}
M \arrow[d] & & \varphi \arrow[d] \\
A_M\colon L(A_M) = Tr(M) \arrow[dr] & & GA_{\neg\varphi}\colon L(GA_{\neg\varphi}) = Tr(\neg\varphi) \arrow[d] \\
& & A_{\neg\varphi}\colon L(A_{\neg\varphi}) = L(GA_{\neg\varphi}) \arrow[dl] \\
& GA\colon L(GA) = L(A_M) \cap L(A_{\neg\varphi}) \arrow[d] & \\
& A\colon L(A) = L(GA) \arrow[dl, "L(A) = \emptyset"'] \arrow[dr, "L(A) \neq \emptyset"] & \\
M \models \varphi & & M \not\models \varphi
\end{tikzcd}
$$