Математические методы верификации схем и программ

Лекторы:

Захаров Владимир Анатольевич

Подымов Владислав Васильевич

e-mail рассказчика:

valdus@yandex.ru

Осень 2017

Лекция 7

Задача model checking для LTL

Алгоритмы верификации LTL-формул: табличный алгоритм, автоматный алгоритм

Автоматы Бюхи, их свойства и обобщения

Задача model checking для LTL

Напоминание

AP — множество атомарных высказываний

Модель Крипке (*над множеством* AP) — это система $M = (S, S_0, R, L)$, где

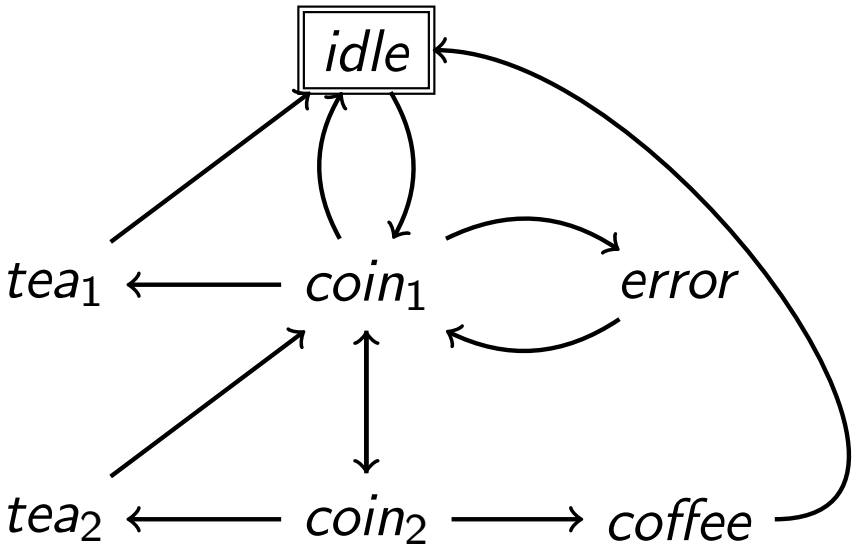
- S — конечное множество состояний
- $S_0 \subseteq S$ — множество начальных состояний
- $R \subseteq S \times S$ — **тотальное** отношение переходов
- $L : S \to 2^{AP}$ — функция разметки

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

$s \to s'$ — синоним $R(s, s')$

Задача model checking для LTL

Пример: кофейный автомат



$L(idle) = \{idle\}$
$L(coin_1) = \{OneCoin, ReadyForTea\}$
$L(error) = \{OneCoin, ShowErrorMessage\}$
$L(tea_1) = \{GiveTea\}$
$L(coin_2) = \{TwoCoins, ReadyForTea, ReadyForCoffee\}$
$L(coffee) = \{GiveCoffee\}$
$L(tea_2) = \{GiveTea, OneCoin\}$

Задача model checking для LTL

Напоминание

БНФ, определяющая синтаксисс LTL-формул $ltlf$ (над AP):

$$ltlf ::= \quad a \mid ltlf \vee ltlf \mid ltlf \,\&\, ltlf \mid \neg ltlf \mid ltlf \to ltlf \mid \mathbf{X}ltlf \mid \mathbf{F}ltlf \mid \mathbf{G}ltlf \mid ltlf\,\mathbf{U}\,ltlf \qquad (a \in AP)$$

В лекции 4 давалось немного другое определение LTL-формул

Отличия:

- отсутствует символ **A**, с которого *в лекции 4* начиналась каждая LTL-формула
  - этот символ избыточен и потому опускается
- отсутствует оператор **R**
  - *лекция 4:* $\varphi\mathbf{R}\psi \equiv \neg(\neg\varphi\mathbf{U}\neg\psi)$

Задача model checking для LTL

Примеры LTL-требований:

- $\mathbf{F}\,GiveCoffee$
- $\mathbf{G}(TwoCoins \to \neg\mathbf{X}\,ShowErrorMessage)$
- $\mathbf{G}\neg(OneCoin \,\&\, \mathbf{X}\,GiveCoffee)$
- $\mathbf{GF}(GiveTea \vee GiveCoffee)$
- $\mathbf{G}\neg ShowErrorMessage \to \mathbf{GF}(GiveTea \vee GiveCoffee)$
- $\mathbf{G}(OneCoin \vee TwoCoins \vee GiveTea \vee GiveCoffee\,\mathbf{U}\,idle)$

А какие из этих требований выполнены для предложенного кофейного автомата?

Задача model checking для LTL

LTL-формулой φ определяется *свойство трасс* $Tr(\varphi)$:
$(tr = A_0, A_1, A_2, \dots)$

- $tr \in Tr(p)$, $p \in AP \quad \Leftrightarrow \quad p \in A_0$
- $Tr(\psi \vee \chi) = Tr(\psi) \cup Tr(\chi)$, $Tr(\psi \,\&\, \chi) = Tr(\psi) \cap Tr(\chi)$, $Tr(\neg\psi) = (2^{AP})^\omega \setminus Tr(\psi)$, $Tr(\psi \to \chi) = Tr(\neg\psi \vee \chi)$
- $tr \in Tr(\mathbf{X}\psi) \quad \Leftrightarrow \quad tr^1 \in Tr(\psi)$
- $tr \in Tr(\mathbf{F}\psi) \quad \Leftrightarrow \quad$ существует k, $k \in \mathbb{N}_0$, такое что $tr^k \in Tr(\psi)$
- $tr \in Tr(\mathbf{G}\psi) \quad \Leftrightarrow \quad$ для любого k, $k \in \mathbb{N}_0$, верно $tr^k \in Tr(\psi)$
- $tr \in Tr(\psi \mathbf{U} \chi) \quad \Leftrightarrow \quad$ существует k, $k \in \mathbb{N}_0$, такое что
  - $tr^k \in Tr(\chi)$
  - для любого m, $m \in \mathbb{N}_0$, $m < k$, верно $tr^m \in Tr(\psi)$

Задача model checking для LTL ($M \overset{?}{\models} \varphi$) может быть переформулирована так: $Tr(M) \overset{?}{\subseteq} Tr(\varphi)$

Табличный алгоритм model checking для LTL

Можно ли проверить выполнимость LTL-формулы на модели Крипке, размечая состояния модели подформулами аналогично тому, как это делалось для CTL?

CTL-формулы — это *формулы состояния*: значение такой формулы в каждом состоянии несложно выводится из значения её подформул в каждом состоянии

LTL-формулы φ — это *формулы пути*: какие-то пути, исходящие из состояния, могут удовлетворять свойству $Tr(\varphi)$, а какие-то — нет

Чтобы проверить, выполнена ли LTL-формула в состоянии, может потребоваться перебрать **всевозможные варианты** того, какие её подформулы верны, а какие — нет, для **каждого** состояния

Небольшая пауза

1. Все ли помнят, что табличный алгоритм model checking для LTL уже рассказывался?
2. Все ли помнят, как выглядело описание алгоритма?
3. Все ли представляют, как этот алгоритм работает хоть для каких-нибудь LTL-формулы и модели Крипке?
4. Все ли помнят, **почему** этот алгоритм работает?

Всё это рассказывалось давно, и тогда ещё не было такого глубокого понимания задачи model checking, как есть сейчас

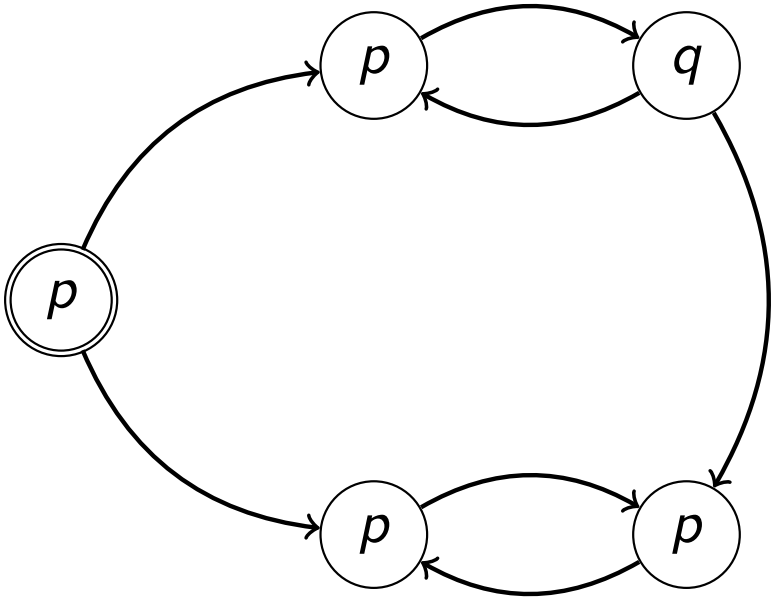
Поэтому имеет смысл напомнить, как выглядит табличный алгоритм проверки LTL-формул

Табличный алгоритм model checking для LTL

Пример:

φ: $p\mathbf{U}q$

M:



Верно ли соотношение $M \models \varphi$?

Табличный алгоритм model checking для LTL

Попытаемся доказать, что $M \not\models \varphi$

Для этого

- упростим формулу φ
- разметим каждое состояние s модели всевозможными "хорошими" **предположениями** о том, какие подформулы формулы φ истинны и какие ложны хотя бы для одного пути, исходящего из s
- проведём "хорошим" образом дуги от предположений, сформулированных для текущего момента времени и текущего состояния, к предположениям, сформулированным для следующего момента времени и следующего состояния
- попробуем найти бесконечную последовательность предположений, начинающуюся с φ и демонстрирующую путь в M, не удовлетворяющий свойству φ

Упрощение формулы

LTL-формулы φ и ψ равносильны, если $Tr(\varphi) = Tr(\psi)$

LTL-формулу будем называть упрощённой, если она

- содержит только связки $\&$, $\neg$ и операторы **X**, **U**
- не содержит двойных отрицаний (двух подряд идущих связок $\neg$)

Утверждение. Для любой LTL-формулы существует равносильная упрощённая LTL-формула

Доказательство. Достаточно поочерёдно исключить из формулы все

- **G**: $\mathbf{G}\varphi \equiv \neg\mathbf{F}\neg\varphi$
- **F**: $\mathbf{F}\varphi \equiv \mathbf{true}\ \mathbf{U}\ \varphi$
- $\rightarrow$: $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$
- $\vee$: $\varphi \vee \psi \equiv \neg(\neg\varphi \,\&\, \neg\psi)$
- $\neg\neg$: $\neg\neg\varphi \equiv \varphi$ ▼

Далее проверяемая формула полагается упрощённой

Предположения

При разметке состояний модели M для проверки формулы φ используются формулы из замыкания Фишера-Ладнера $[\varphi]_{fl}$, содержащего

- все подформулы формулы φ, кроме подформул вида $\neg\psi$
- $\mathbf{X}(\psi\mathbf{U}\chi)$ для каждой подформулы $\psi\mathbf{U}\chi$ формулы φ
  - *пояснение того, зачем включать в $[\varphi]_{fl}$ такие* $\mathbf{X}$*-формулы, будет дальше*

Предположение (для формулы φ) — это подмножество множества $[\varphi]_{fl}$, в котором каждая формула помечена как истинная или ложная

Для краткости будем использовать запись $\neg\psi$ как синоним ψ, и запись $\neg\psi$ как синоним ψ

Хорошие и плохие предположения

Предположение H согласованно с множеством атомарных высказываний A, если для любого a, $a \in AP$, верно

$$a \in H \quad \Leftrightarrow a \in A$$

Хорошее предположение H, помечающее состояние s модели Крипке (S, S_0, R, L), удовлетворяет двум условиям:

1. согласованность с состоянием s, то есть согласованность с атомарными высказываниями $L(s)$
2. внутренняя согласованность: для любых формул $\psi, \psi_1 \,\&\, \psi_2, \chi_1 \mathbf{U} \chi_2 \in [\varphi]_{FL}$ верно:
   - $\psi \in H \Leftrightarrow \psi \notin H$
   - $\psi_1 \,\&\, \psi_2 \in H \Leftrightarrow \psi_1, \psi_2 \in H$
   - $\chi_1 \mathbf{U} \chi_2 \in H \Leftrightarrow \chi_2 \in H$ или $\{\chi_1, \mathbf{X}(\chi_1 \mathbf{U} \chi_2)\} \subseteq H$

Табличный алгоритм: разметим **каждое** состояние модели Крипке **всевозможными** внутренне согласованными предположениями, согласованными с размечаемым состоянием

Хорошие и плохие предположения

- внутренняя согласованность: для любых формул $\psi, \psi_1 \,\&\, \psi_2, \chi_1 \mathbf{U} \chi_2 \in [\varphi]_{FL}$ верно:
  - $\psi \in H \Leftrightarrow \psi \notin H$
  - $\psi_1 \,\&\, \psi_2 \in H \Leftrightarrow \psi_1, \psi_2 \in H$
  - $\chi_1 \mathbf{U} \chi_2 \in H \Leftrightarrow \chi_2 \in H$ или $\{\chi_1, \mathbf{X}(\chi_1 \mathbf{U} \chi_2)\} \subseteq H$

Пояснение: предположение H внутренне согласованно $\Leftrightarrow$ существует трасса, удовлетворяющая истинным и не удовлетворяющая ложным свойствам предположения

В частности, справедлива следующая равносильность:

$$\psi \mathbf{U} \chi \equiv \chi \vee \psi \,\&\, \mathbf{X}(\psi \mathbf{U} \chi)$$

Это означает, что

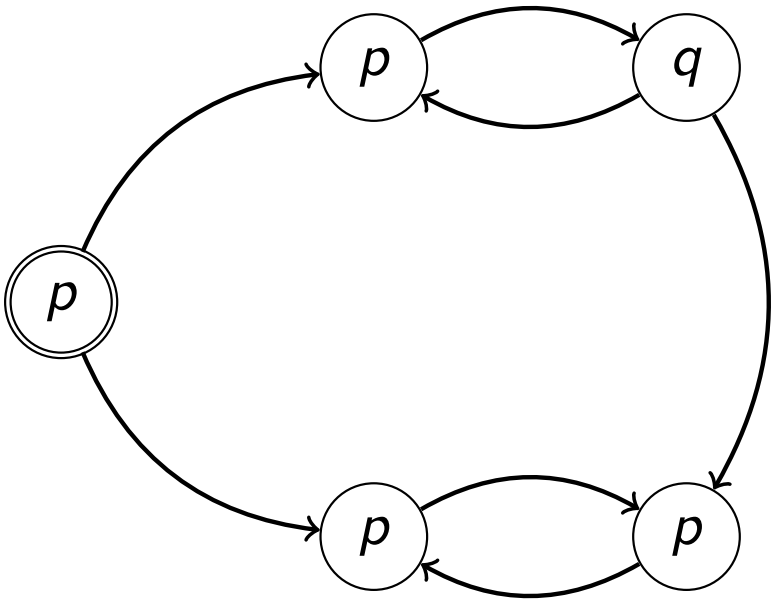
- если χ, то $\psi \mathbf{U} \chi$
- если ψ и χ, то $\psi \mathbf{U} \chi$
- если ψ и χ, то $\psi \mathbf{U} \chi \Leftrightarrow \mathbf{X}(\psi \mathbf{U} \chi)$

Хорошие и плохие предположения

Вернёмся к примеру:

φ: $p\mathbf{U}q$

M:

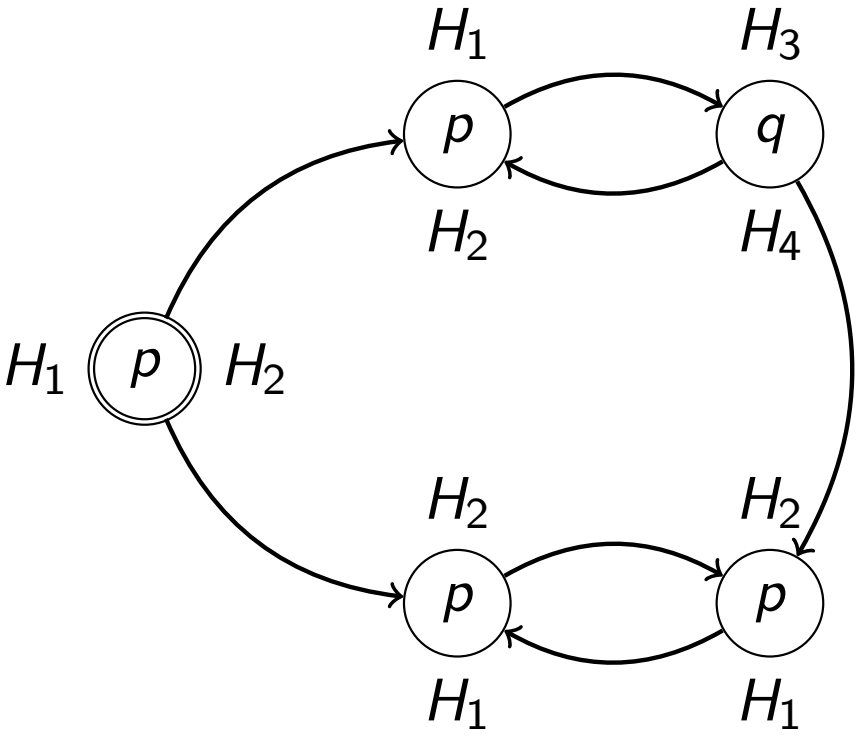


Разметим состояния модели M хорошими предположениями

Хорошие и плохие предположения

φ: $p\mathbf{U}q$

M:



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

Хорошие и плохие обещания на завтра

Упорядоченная пара предположений (H_1, H_2) внешне согласованна, если для любой формулы вида $\mathbf{X}\psi$ из $[\varphi]_{fl}$ верно:

$$\mathbf{X}\psi \in H_1 \quad \Leftrightarrow \quad \psi \in H_2$$

Табличный алгоритм: соединим дугой все пары внешне согласованных предположений H_1, H_2, помечающих состояния s_1, s_2 соответственно, такие что $s_1 \to s_2$

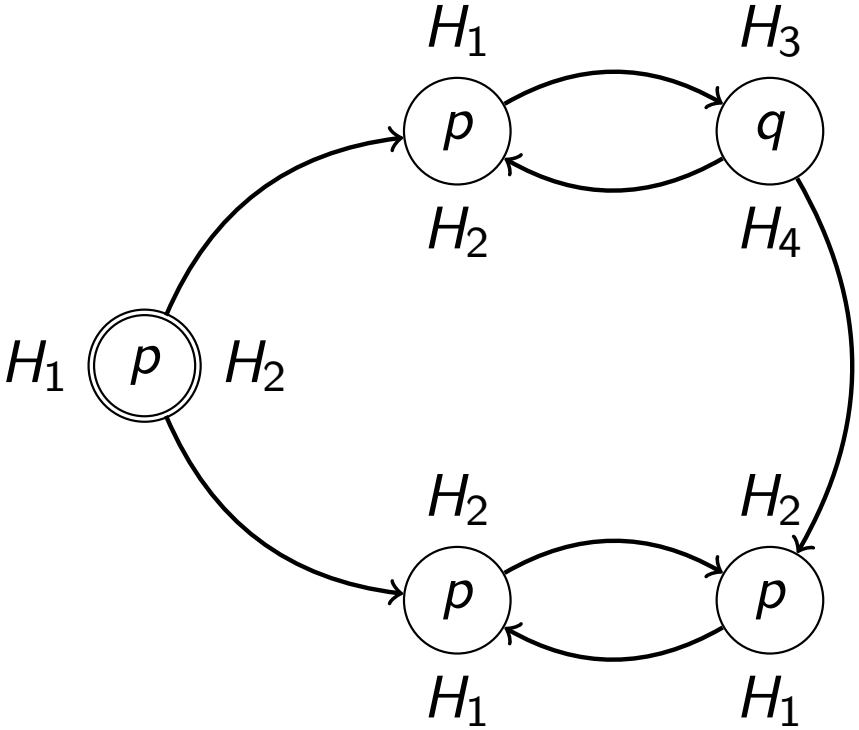
Пояснение: если трасса tr удовлетворяет истинным и не удовлетворяет ложным свойствам предположения H_1, а трасса tr^1 удовлетворяет истинным и не удовлетворяет ложным свойствам предположения H_2, то пара (H_1, H_2) обязательно является внешне согласованной

Хорошие и плохие обещания на завтра

Вернёмся к примеру:

φ: $p\mathbf{U}q$

M:



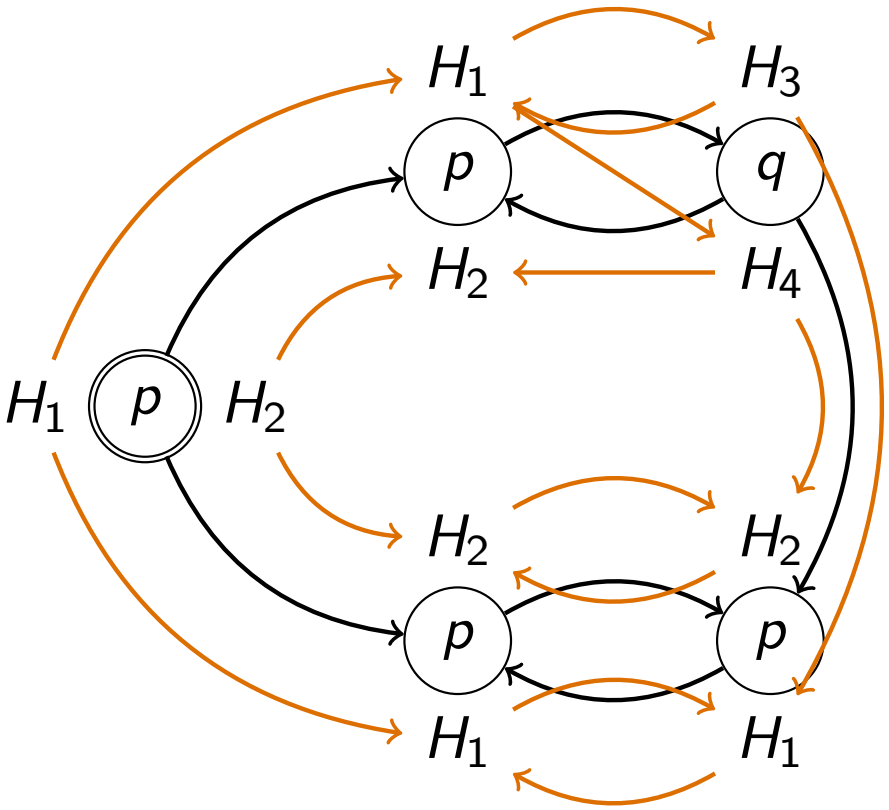
$$H_1 = \{p, q, \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\} \quad H_3 = \{p, q, \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$
$$H_2 = \{p, q, \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\} \quad H_4 = \{p, q, \mathbf{X}(p\mathbf{U}q), p\mathbf{U}q\}$$

Соединим дугами соседние внешне согласованные предположения

Хорошие и плохие обещания на завтра

φ: $p\mathbf{U}q$

M:



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

Система Хинтикки

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

В некоторых случаях бесконечная последовательность внешне согласованных предположений демонстрирует наличие пути в модели Крипке, удовлетворяющего истинным и не удовлетворяющего ложным формулам первого предположения:

- $H_1 \to H_1 \to H_3 \to \ldots$:
  трасса пути, размеченного таким образом, входит в $Tr(p\mathbf{U}q)$
- $H_2 \to \ldots$:
  трасса пути, размеченного таким образом, не входит в $Tr(p\mathbf{U}q)$

К сожалению, это работает не всегда:

- $H_1 \to H_1 \to \cdots \to H_1 \to \ldots$: несмотря на то что $p\mathbf{U}q \in H_1$, трасса пути, размеченного таким образом, не входит в $Tr(p\mathbf{U}q)$

"Плохая" особенность бесконечной последовательности предположений H_1:

Система Хинтикки

Для формулы $\psi \mathbf{U} \chi$ в предположении H звенит звонок, верно хотя бы одно из двух:

- $\chi \in H$
- $\mathbf{X}(\varphi \mathbf{U} \psi) \in H$

Система Хинтикки для модели Крипке M и формулы φ — это граф следующего вида:

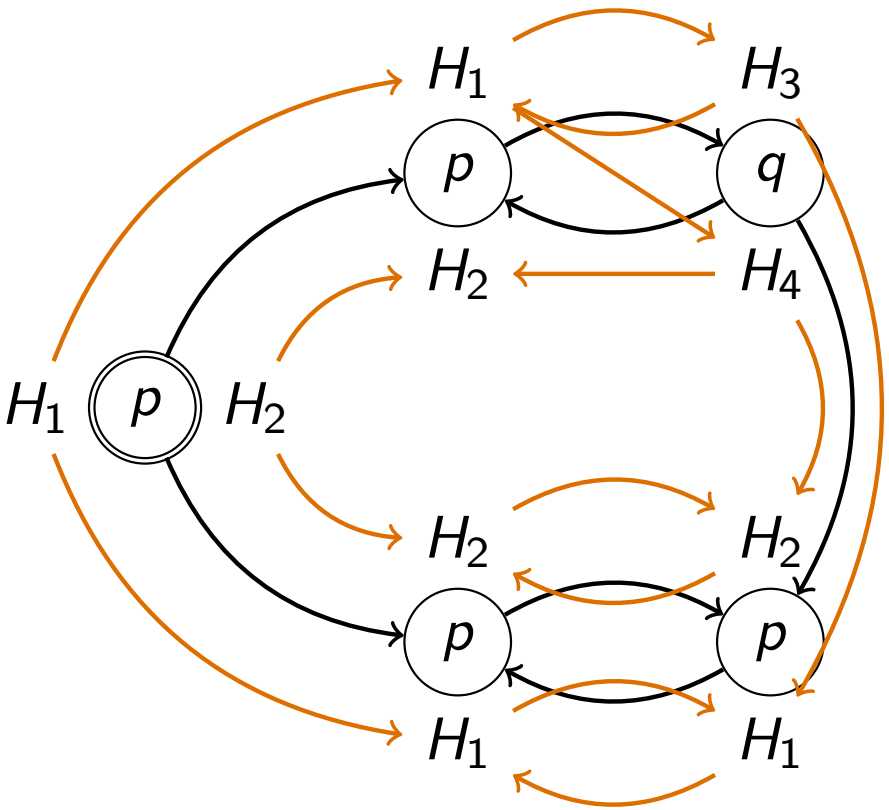
- вершины — всевозможные пары (s, H), где s — состояние модели и H — предположение, которым состояние помечается согласно табличному алгоритму
- $(s, H) \to (s', H') \Leftrightarrow s \to s'$ и $H \to H'$
- вершина (s, H) объявляется начальной, если s — начальное состояние модели и $\varphi \in H$
- каждой подформуле Φ вида $\psi \mathbf{U} \chi$ формулы φ присваивается уникальный цвет c_Φ
- вершина (s, H) помечается цветом c_Φ, если в H звенит звонок для формулы Φ

Система Хинтикки

Вернёмся к примеру:

φ: $p\mathbf{U}q$

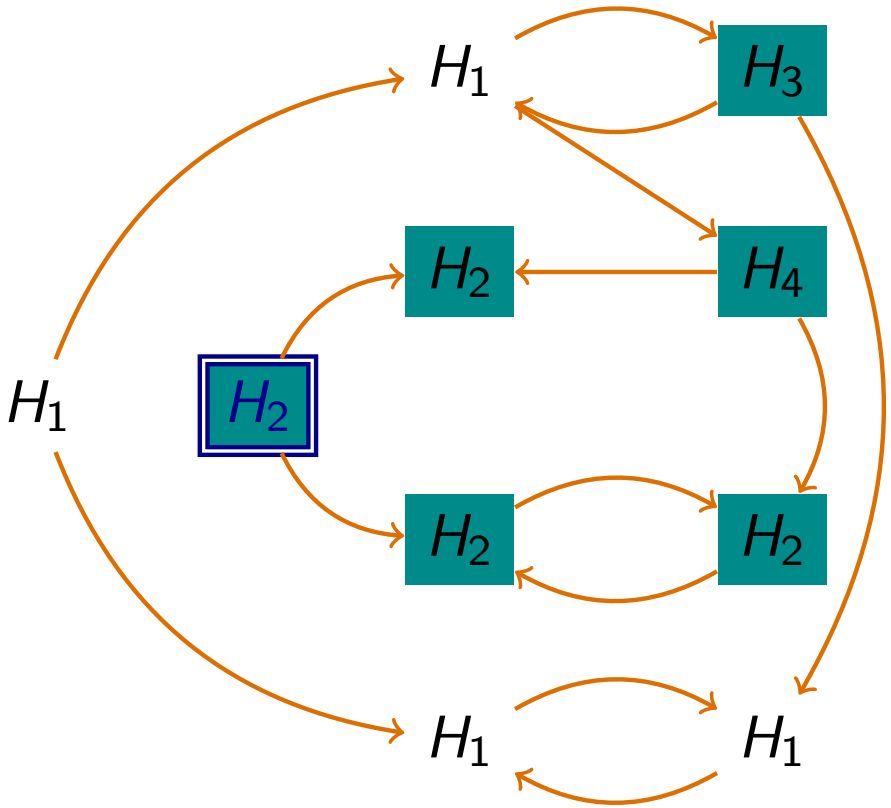
M:



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

Система Хинтикки

Система Хинтикки для M и φ:



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

Система Хинтикки

Бесконечный маршрут в системе Хинтикки является опровергающим, если он

- начинается в начальной вершине
- для каждого цвета c_Φ содержит бесконечно много вершин, окрашенных в этот цвет

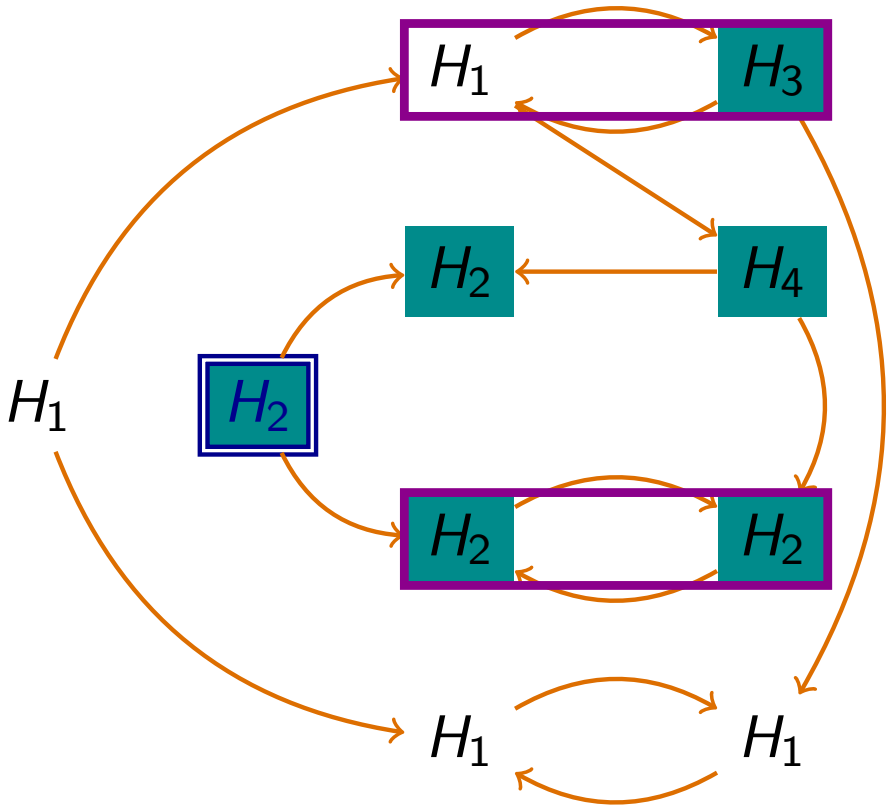
Теорема. $M \not\models \varphi \Leftrightarrow$ **в системе Хинтикки для** M **и** φ **существует опровергающий маршрут**

Теорема. В системе Хинтикки существует опровергающий маршрут $\Leftrightarrow$ в этой системе из начальной вершины достижима компонента сильной связности, в вершинах которой встречаются все цвета

Обоснование этих утверждений смотрите в курсе "Математическая логика и логическое программирование" для кафедр 3-го потока факультета

Система Хинтикки

Обратно к примеру:



Две подходящие компоненты сильной связности обведены прямоугольниками

Одна из них достижима из начальной вершины, а значит, $M \not\models p\mathbf{U}q$

Система Хинтикки

Достоинства табличного алгоритма: . . .

Недостатки табличного алгоритма:

- труден для понимания
- требуется явный обход графа, полученного из модели Крипке разметкой состояний предположениями (а можно ли этого избежать?)
- состояния могут размечаться большими множествами формул (а насколько большими?)

Алгоритмы model checking

Задача model checking для CTL имеет эффективную альтернативу табличному алгоритму: *символьный алгоритм*

Задача model checking для LTL также имеет эффективную альтернативу табличному алгоритму: автоматный алгоритм

Если программное средство производит верификацию для CTL, то *скорее всего* в основе реализованного алгоритма верификации лежит символьный алгоритм

Если программное средство производит верификацию для LTL, то *скорее всего* в основе реализованного алгоритма верификации лежит автоматный алгоритм

Автоматный алгоритм model checking для LTL

Общая схема работы автоматного алгоритма:

- имеются модель Крипке M и LTL-формула φ
- построить автомат A_M, распознающий множество $Tr(M)$
- построить автомат $A_{\neg\varphi}$, распознающий множество $Tr(\neg\varphi)$
- построить автомат $A_M \oplus A_{\neg\varphi}$, распознающий множество $Tr(M) \cap Tr(\neg\varphi)$
- проверить, пусто ли множество, распознаваемое автоматом $A_M \oplus A_{\neg\varphi}$
  - если да, то $M \models \varphi$
  - если нет, то $M \not\models \varphi$

И что же это за автоматы?

Автоматы Бюхи

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

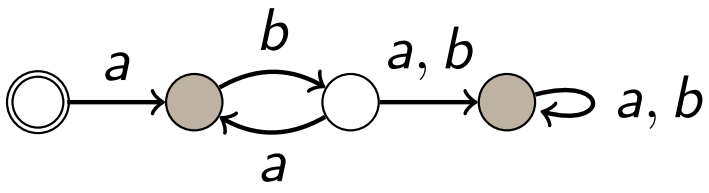
- S — конечное множество состояний
- $S_0 \subseteq S$ — множество начальных состояний
- $\delta \subseteq S \times \Sigma \times S$ — отношение переходов
- $F \subseteq S$ — множество заключительных состояний

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

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

Автоматы Бюхи

Пример



◎ — начальное состояние
● — заключительное состояние

Синтаксис автоматов Бюхи в точности совпадает с синтаксисом конечных недетерминированных автоматов-распознавателей

Автомат Бюхи, прочитывая конечные слова, изменяет свои состояния точно так же, как и автоматы распознаватели

Отличие этих двух моделей — в семантике: распознаваемом **языке** (множестве слов)

Автоматы Бюхи

Автоматы Бюхи строятся для того, чтобы проверять, содержит ли модель Крипке **трассы**, не удовлетворяющие заданному **свойству**

И трассы, и свойства основаны на **бесконечных** словах

Язык, распознаваемый автоматом Бюхи, — это множество **бесконечных** слов в заданном алфавите

Множество всех бесконечных слов в алфавите Σ будем обозначать так: Σ^ω

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

Автоматы Бюхи

$inf(t)$ — множество состояний, встречающихся бесконечно часто в трассе t автомата Бюхи

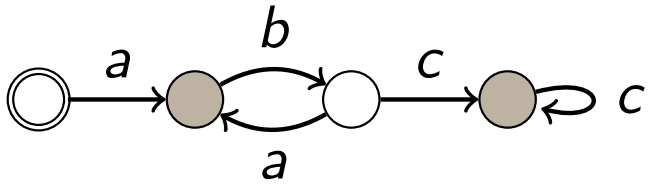
Язык $L(A)$, распознаваемый автоматом Бюхи $A = (S, S_0, \delta, F)$, определяется так:

$$L(A) = \{w \mid \exists t \in Tr(A, w) : inf(t) \cap F \neq \emptyset\}$$

Проще говоря, бесконечное слово w распознаётся автоматом Бюхи, если среди всех трасс автомата, получаемых при прочитывании этого слова, есть такая, в которой бесконечно часто повторяется хотя бы одно заключительное состояние

Автоматы Бюхи

Пример



Язык, распознаваемый этим автоматом Бюхи, состоит в точности из всех бесконечных слов следующего вида:

- $(ab)^\omega$ — слово, получающееся бесконечным повторением слова ab
- $(ab)^+c^\omega$ — слово, начинающееся с конечного ненолевого числа повторений ab, после которого бесконечно повторяется буква c

Автоматы Бюхи и модели Крипке

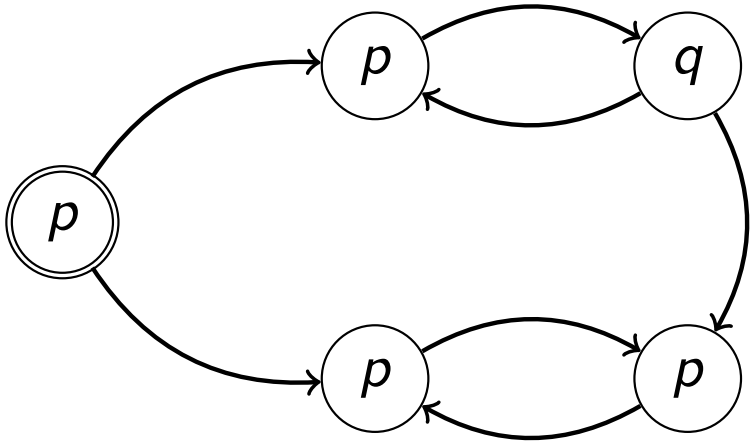
Как по модели Крипке M построить автомат Бюхи, распознающий язык $Tr(M)$?

Очень просто:

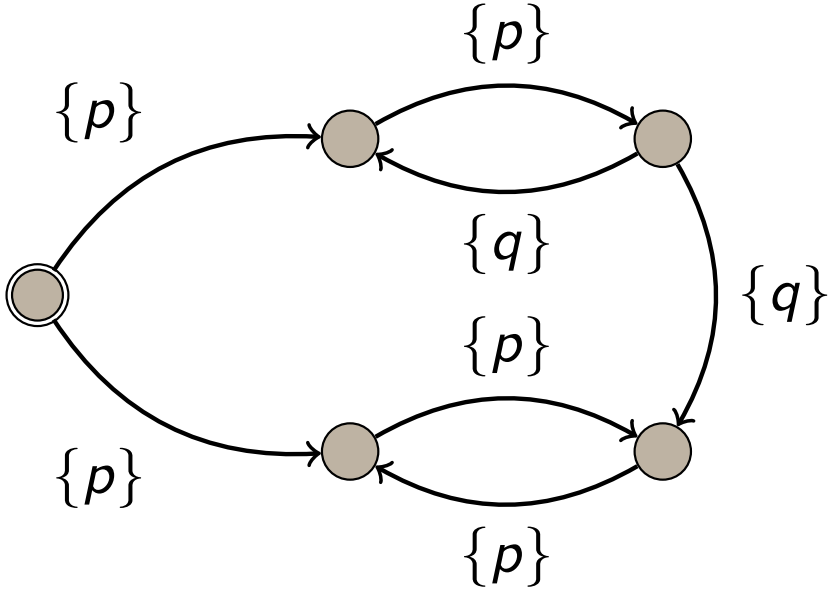
- алфавит, над которым строится автомат: 2^{AP}
- объявим состояния модели состояниями автомата Бюхи
- начальные состояния модели объявим начальными состояниями автомата
- соединим состояния автомата дугами так же, как они были соединены в модели
- множество высказываний, помечавшее состояние модели, переместим на каждую исходящую из состояния дугу в автомате
- объявим **все** состояния автомата заключительными

Автоматы Бюхи и модели Крипке

Пример



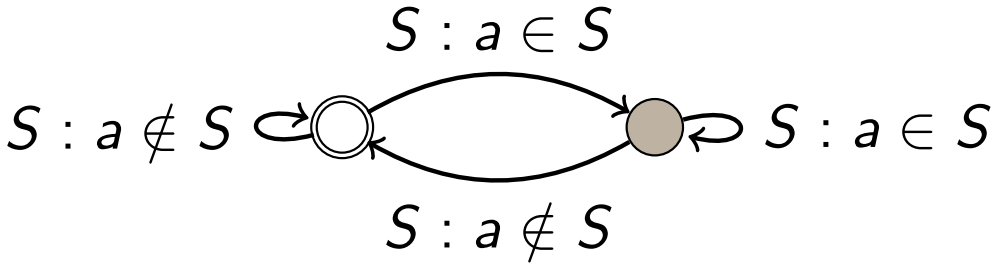
Автомат, распознающий в точности множество всех трасс этой модели Крипке, может выглядеть так:



Автоматы Бюхи и LTL-формулы

Как по LTL-формуле φ построить автомат Бюхи, распознающий множество $Tr(\varphi)$?

Начнём с примеров

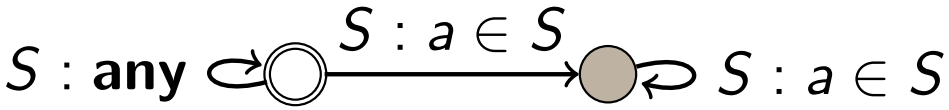


Этот автомат распознаёт множество $Tr(\mathbf{GF}a)$

Автоматы Бюхи и LTL-формулы

Как по LTL-формуле φ построить автомат Бюхи, распознающий множество $Tr(\varphi)$?

Начнём с примеров

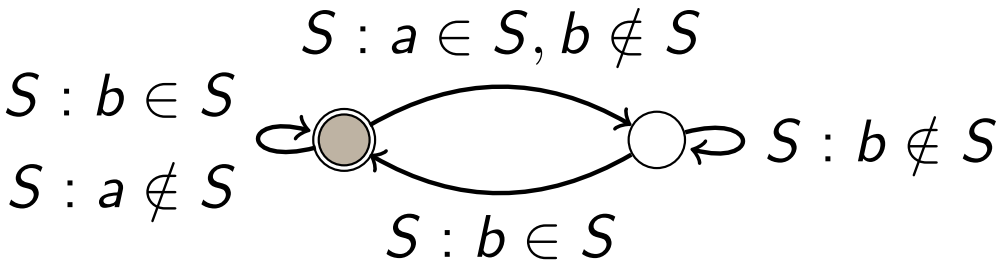


Этот автомат распознаёт множество $Tr(\mathbf{FG}a)$

Автоматы Бюхи и LTL-формулы

Как по LTL-формуле φ построить автомат Бюхи, распознающий множество $Tr(\varphi)$?

Начнём с примеров

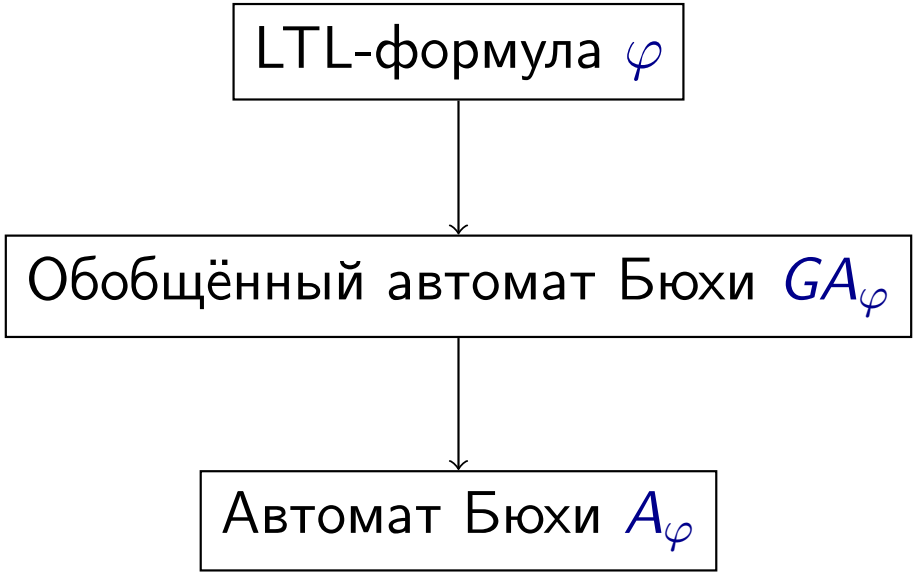


Этот автомат распознаёт множество $Tr(\mathbf{G}(a \to \mathbf{F}b))$

А как построить такой автомат Бюхи для **произвольной** LTL-формулы?

Построение автомата Бюхи по LTL-формуле

Общая схема построения автомата Бюхи по LTL-формуле:



Обобщённые автоматы Бюхи

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

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

Основное отличие от *необобщённого* автомата Бюхи состоит в том, как трактуются заключительные состояния

Бесконечное слово w распознаётся обобщённым автоматом Бюхи GA, если среди трасс, порождаемых словом w, существует трасса t, такая что $inf(t) \cap F \neq \emptyset$ для **любого** множества F из семейства $\mathcal{F}$

Обобщённые автоматы Бюхи

Утверждение

Для любого обобщённого автомата Бюхи существует *необобщённый* автомат Бюхи, распознающий тот же язык

Доказательство.

Рассмотрим произвольный обобщённый автомат Бюхи $GA = (S, S_0, \delta, \{F_0, \ldots, F_{k-1}\})$

Требуемый автомат Бюхи $A = (S', S'_0, \delta, F)$ имеет следующее устройство:

- $S' = S \times \{0, \ldots, k-1\}$
- $S'_0 = S_0 \times \{0\}$
- $F = \{F_i \times \{i\} \mid 0 \leq i < k\}$
- $\delta'((s, i), \sigma, (s', i)) \Leftrightarrow \delta(s, \sigma, s')$ и $s \notin F_i$
- $\delta'((s, i), \sigma, (s', i + 1(\mathbf{mod}\ k))) \Leftrightarrow \delta(s, \sigma, s')$ и $s \in F_i$ ▼

Обобщённые автоматы Бюхи и LTL-формулы

Как по LTL-формуле φ построить обобщённый автомат Бюхи, распознающий множество $Tr(\varphi)$?

Например, такой обобщённый автомат Бюхи $A = (S, S_0, \delta, \mathcal{F})$ распознаёт язык $Tr(\varphi)$:

- S — это всевозможные *внутренне согласованные предположения*, построенные на основе *замыкания Фишера-Ладнера* формулы φ
- S_0 — это всевозможные предположения H, такие что $\varphi \in H$
- Каждой формуле $\psi \mathbf{U} \chi \in [\varphi]_{FL}$ соответствует множество заключительных состояний $F_{\psi \mathbf{U} \chi}$, состоящее в точности из тех предположений H, для которых *звенит звонок* для формулы $\psi \mathbf{U} \chi$
- $\delta(H_1, L, H_2) \Leftrightarrow$
  - H_1 и H_2 — *внешне согласованные предположения*
  - предположение H_1 *согласованно с атомарными высказываниями* L

Автоматный алгоритм model checking для LTL

Промежуточный итог

Для заданных модели Крипке M и LTL-формулы φ требуется проверить соотношение

$$M \models \varphi$$

Что для этого уже сделано:

- для модели M построен автомат Бюхи A_M, такой что $L(A) = Tr(M)$
- для формулы $\neg\varphi$ построен автомат Бюхи $A_{\neg\varphi}$, такой что $L(A_{\neg\varphi}) = Tr(\neg\varphi)$

Что осталось сделать:

- построить автомат $A_M \oplus A_{\neg\varphi}$, распознающий язык $L(A_M) \cap L(A_{\neg\varphi})$
- проверить, пуст ли язык $L(A_M \oplus A_{\neg\varphi})$

Пересечение автоматов Бюхи

Утверждение

Для любой пары автоматов Бюхи A_1, A_2 существует автомат Бюхи A, распознающий язык $L(A_1) \cap L(A_2)$

Доказательство.

Рассмотрим произвольные автоматы Бюхи $A_1 = (S_1, S_0^1, \delta_1, F_1)$, $A_2 = (S_2, S_0^2, \delta_2, F_2)$

Достаточно построить **обобщённый** автомат Бюхи, распознающий язык $L(A_1) \cap L(A_2)$

Требуемый обобщённый автомат Бюхи GA имеет следующее устройство:

$$GA = (S_1 \times S_2, S_0^1 \times S_0^2, \delta, \{F_1 \times S_2, S_1 \times F_2\}), \text{ где}$$

$$\delta((s_1, s_2), \sigma, (s_1', s_2')) \Leftrightarrow \delta_1(s_1, s_1') \text{ и } \delta_2(s_2, s_2')$$

▼

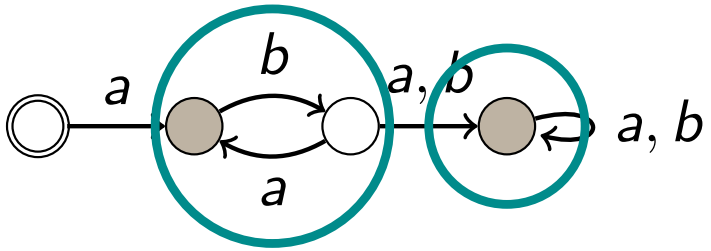
Проверка пустоты автомата Бюхи

Утверждение

Язык, распознаваемый автоматом Бюхи A, непуст в том и только том случае, если из какого-либо начального состояния достижимо заключительное состояние, принадлежащее какой-либо компоненте сильной связности

Доказательство.Очевидно?

Пример



Язык, распознаваемый этим автоматом, непуст

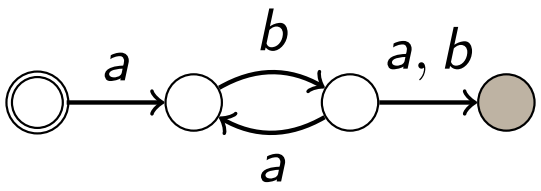
Проверка пустоты автомата Бюхи

Утверждение

Язык, распознаваемый автоматом Бюхи A, непуст в том и только том случае, если из какого-либо начального состояния достижимо заключительное состояние, принадлежащее какой-либо компоненте сильной связности

Доказательство.Очевидно?

Пример



Язык, распознаваемый этим автоматом, пуст

Сведение задачи model checking для LTL к проблеме пустоты для автоматов Бюхи

Итог

Пусть M — произвольная модель Крипке и φ — произвольная LTL-формула

Тогда

$$M \models \varphi$$

$$\Leftrightarrow$$

$$L(A_M \oplus A_{\neg\varphi}) = \emptyset$$

Конец лекции 7