Математические методы верификации схем и программ

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

Блок 21

Символьный алгоритм model checking для CTL
Преобразователи предикатов
Неподвижные точки

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

ВМК МГУ, 2022/2023, осенний семестр

Основная идея символьного алгоритма

Символьным алгоритмом решения задачи обычно называют решающий алгоритм, в котором для основных («трудоёмких») структур данных используются символьные представления в предположении о том, что работать с такими представлениями можно существенно более эффективно, чем с явными

Алгоритм model checking для CTL линеен относительно как размера формулы, так и размера модели

Но на практике возникает необходимость применять алгоритм к настолько большим моделям, что даже такая (казалось бы совсем невысокая) сложность оказывается неприемлемой

Основная идея символьного алгоритма

Например, если в системе независимо асинхронно выполняется n одинаковых процессов, в каждом из которых k состояний, то суммарное число состояний системы (k^n) экспоненциально относительно числа процессов

Такой эффект экспоненциального роста числа состояний относительно параметров системы называется комбинаторным взрывом числа состояний, и это основная проблема, возникающая в области model checking на практике

Символьный алгоритм model checking для CTL — это базовый алгоритм, адаптированный для работы с моделями Крипке в символьном представлении

Стандартное символьное представление (например, в виде ROBDD) множества или отношения X далее будем обозначать записью Φ_X

Представление Φ множества над переменными $\vec{x}$ будем обозначать записью $\Phi(\vec{x})$, и особо для двуместных отношений над комплектами переменных $\vec{x}$ для первого аргумента и $\vec{y}$ для второго — $\Phi(\vec{x}, \vec{y})$

Основная идея символьного алгоритма

Записью $\vec{x}$ будем обозначать набор переменных $x_1, \ldots, x_m$ для некоторого m, *заданного контекстом*

$\exists\vec{x}$ — так будем сокращать запись $\exists x_1 \exists x_2 \ldots \exists x_m$

Основные операции над множествами и отношениями, используемые в базовом алгоритме, естественно переформулируются на языке символьных представлений:

- Объединение множеств: $A = B \cup C \mapsto \Phi_A = \Phi_B \vee \Phi_C$
- Пересечение множеств: $A = B \cap C \mapsto \Phi_A = \Phi_B \,\&\, \Phi_C$
- Разность множеств: $A = B \setminus C \mapsto \Phi_A = \Phi_B \,\&\, \neg\Phi_C$
- Образ отношения: $A = \{y \mid \exists x : (x, y) \in R\} \mapsto \Phi_A(\vec{y}) = \exists\vec{x}\Phi_R(\vec{x}, \vec{y})$
- Прообраз отношения: $A = \{x \mid \exists y : (x, y) \in R\} \mapsto$ $\Phi_A(\vec{x}) = \exists\vec{y}\Phi_R(\vec{x}, \vec{y})$

Для «стыковки» переменных множеств понадобится также уметь переименовывать переменные в формулах: $\Phi[\vec{x}/\vec{y}] = \Phi[x_1/y_1, \ldots, x_m/y_m]$

Символьный алгоритм (начало)

Дано:

- Стандартное символьное представление конечной модели Крипке $M = (S, S_0, \to, L)$ над $\{p_1, \ldots, p_k\}$:
  $\mathfrak{M} = (\Phi_S(\vec{x}), \Phi_{S_0}(\vec{x}), \Phi_{\to}(\vec{x}, \vec{y}), \Phi_{p_1}(\vec{x}), \ldots, \Phi_{p_k}(\vec{x}))$
- Ctl-формула φ

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

Базовый алгоритм, основная процедура:

- Вычислить множество $X = Sat(M, \varphi)$ при помощи процедуры с соответствующим названием
- Проверить включение $S_0 \subseteq X$
  - То есть проверить соотношение $S_0 \setminus X = \emptyset$
- Вернуть результат проверки предыдущего пункта

Символьный алгоритм, основная процедура:

- Вычислить $\Phi_X(\vec{x}) = Sat_{\mathfrak{s}}(\mathfrak{M}, \varphi)$
- Проверить соотношение $\Phi_{S_0} \,\&\, \neg\Phi_X \sim \Phi_\emptyset$
- Вернуть результат проверки предыдущего пункта

Символьный алгоритм (начало)

Базовый алгоритм, процедура $Sat(M, \varphi)$:

- Используя известные равносильности, преобразовать φ в равносильную *упрощённую* формулу ψ в базисе **EX**, **EG**, **EU**:
$$\psi \ ::= \ \mathbb{t} \ | \ p \ | \ \psi \,\&\, \psi \ | \ \neg\psi \ | \ \mathbf{EX}\psi \ | \ \mathbf{EG}\psi \ | \ \mathbf{E}(\psi \mathbf{U} \psi)$$

- $Sat(M, \varphi) = Sat'(M, \psi)$, где процедура Sat' отличается от Sat тем, что применяется только к *упрощённым* формулам

Символьный алгоритм, процедура $Sat_{\mathfrak{s}}(\mathfrak{M}, \varphi)$: всё дословно то же, кроме начала второго пункта, $Sat(M, \varphi)$ и $Sat'(M, \psi)$ следует заменить соответственно на $Sat_{\mathfrak{s}}(\mathfrak{M}, \varphi)$ и $Sat'_{\mathfrak{s}}(\mathfrak{M}, \psi)$

Символьный алгоритм (начало)

Базовый алгоритм, процедура $Sat'(M, \varphi)$:

- Если $\varphi = \mathbb{t}$, то $Sat'(M, \varphi) = S$
- Если $\varphi = p \in \mathrm{AP}$, то $Sat'(M, \varphi) = \{s \mid s \in S,\ p \in L(s)\}$
- Если $\varphi = \psi_1 \,\&\, \psi_2$, то $Sat'(M, \varphi) = Sat'(M, \psi_1) \cap Sat'(M, \psi_2)$
- Если $\varphi = \neg\psi$, то $Sat'(M, \varphi) = S \setminus Sat'(M, \psi)$
- Если $\varphi = \mathbf{EX}\psi$, то $Sat'(M, \varphi) = Sat_{\mathbf{EX}}(M, \psi)$
- Если $\varphi = \mathbf{EG}\psi$, то $Sat'(M, \varphi) = Sat_{\mathbf{EG}}(M, \psi)$
- Если $\varphi = \mathbf{E}(\psi_1 \mathbf{U} \psi_2)$, то $Sat'(M, \varphi) = Sat_{\mathbf{EU}}(M, \psi_1, \psi_2)$

Символьный алгоритм, процедура $Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi)$:

- Если $\varphi = \mathbb{t}$, то $Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi) = \Phi_S$
- Если $\varphi = p \in \mathrm{AP}$, то $Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi) = \Phi_p$
- Если $\varphi = \psi_1 \,\&\, \psi_2$, то $Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi) = Sat'_{\mathfrak{s}}(\mathfrak{M}, \psi_1) \,\&\, Sat'_{\varphi}(\mathfrak{M}, \psi_2)$
- Если $\varphi = \neg\psi$, то $Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi) = \Phi_S \,\&\, \neg Sat'_{\mathfrak{s}}(\mathfrak{M}, \psi)$
- Если $\varphi = \mathbf{EX}\psi$, то $Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi) = Sat_{\mathfrak{s}\mathbf{EX}}(\mathfrak{M}, \psi)$
- Если $\varphi = \mathbf{EG}\psi$, то $Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi) = Sat_{\mathfrak{s}\mathbf{EG}}(\mathfrak{M}, \psi)$
- Если $\varphi = \mathbf{E}(\psi_1 \mathbf{U} \psi_2)$, то $Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi) = Sat_{\mathfrak{s}\mathbf{EU}}(\mathfrak{M}, \psi_1, \psi_2)$

Символьный алгоритм (начало)

Базовый алгоритм, множество $Pre(M, X) = \{s \mid \exists s' : s \to s',\ s' \in X\}$

Символьный алгоритм: $Pre_{\mathfrak{s}}(\mathfrak{M}, \Phi_X) = \exists \vec{y}(\Phi_{\to} \,\&\, \Phi_X[\vec{x}/\vec{y}])$

Базовый алгоритм, процедура $Sat_{\mathbf{EX}}(M, \varphi)$:

- Вычислить $X = Sat'(M, \varphi)$
- Вернуть множество $Pre(M, X)$

Символьный алгоритм, процедура $Sat_{\mathfrak{s}\mathbf{EX}}(\mathfrak{M}, \varphi)$:

- Вычислить $\Phi_X = Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi)$
- Вернуть $Pre_{\mathfrak{s}}(\mathfrak{M}, \Phi_X)$

Преобразователи предикатов

В базовом алгоритме вычисление множеств $Sat'(M, \mathbf{EX}\psi)$, $Sat'(M, \mathbf{EG}\psi)$ и $Sat'(M, \mathbf{E}(\psi_1 \mathbf{U} \psi_2))$ основывалось на преобразовании множеств состояний, или, по-другому, одноместных предикатов на множестве S

Способ преобразования предиката можно представить как преобразователь предикатов: функцию вида $\mathbb{f} : 2^S \to 2^S$, такую что $\mathbb{f}(A)$ — это результат (одного шага) преобразования предиката A

В частности, комбинации **EX** можно расценивать как преобразователь предикатов:

$$\mathbf{EX}(A) = \{s \mid \exists s' : s \to s',\ s' \in A\}$$

Преобразователи предикатов

Совокупность $(2^S, \subseteq)$ — это **решётка**, в которой:

- Точная верхняя грань $sup(A, B)$ множеств A и B — это их объединение
- Точная нижняя грань $inf(A, B)$ множеств A и B — это их пересечение
- Наибольший элемент — это можество S
- Наименьший элемент — это $\emptyset$

Таким образом, преобразователь предикатов может расцениваться как функция преобразования элементов решётки, и из этого далее будут вытекать терминология и свойства преобразователей

Отношение $\subseteq$ в такой решётке — это отношение нестрогого порядка, и для этого отношения будем применять соответствующую терминологию:

- $A \subseteq B \Leftrightarrow A$ не больше B и B не меньше A
- $A \subset B \Leftrightarrow A$ меньше B и B больше A
- ...

Преобразователи предикатов

Преобразователь предикатов $\mathbb{f} : 2^S \to 2^S$ называется

- монотонным, если для любых предикатов A, B справедлива импликация
$$A \subseteq B \Rightarrow \mathbb{f}(A) \subseteq \mathbb{f}(B)$$
- $\cup$-непрерывным, если для любой бесконечной монотонно неубывающей последовательности предикатов
$$A_1 \subseteq A_2 \subseteq \ldots$$
верно $\mathbb{f}(\bigcup\limits_{i=1}^{\infty} A_i) = \bigcup\limits_{i=1}^{\infty} \mathbb{f}(A_i)$
- $\cap$-непрерывным, если для любой бесконечной монотонно невозрастающей последовательности предикатов
$$A_1 \supseteq A_2 \supseteq \ldots$$
верно $\mathbb{f}(\bigcap\limits_{i=1}^{\infty} A_i) = \bigcap\limits_{i=1}^{\infty} \mathbb{f}(A_i)$

Преобразователи предикатов

Лемма. Любой монотонный преобразователь предикатов $\mathbb{f}$ на конечном множестве S $\cup$-непрерывен и $\cap$-непрерывен

Доказательство.

Рассмотрим бесконечную последовательность предикатов $A_1 \subseteq A_2 \subseteq \ldots$

Так как множество S конечно, для некоторого k верно $A_k = A_{k+1} = A_{k+2} = \ldots$

Так как $A_1 \subseteq \cdots \subseteq A_k$, то верно и $\bigcup\limits_{i=1}^{k} A_i = A_k$

Значит, $\bigcup\limits_{i=1}^{\infty} A_i = A_k$

Так как $\mathbb{f}$ монотонен, верно $\mathbb{f}(A_1) \subseteq \cdots \subseteq \mathbb{f}(A_k) = \mathbb{f}(A_{k+1}) = \ldots$, и аналогично верно $\bigcup\limits_{i=1}^{\infty} \mathbb{f}(A_i) = \mathbb{f}(A_k)$

Следовательно, $\mathbb{f}(\bigcup\limits_{i=1}^{\infty} A_i) = \mathbb{f}(A_k) = \bigcup\limits_{i=1}^{\infty} \mathbb{f}(A_i)$, то есть $\mathbb{f}$ $\cup$-непрерывен

$\cap$-непрерывность обосновывается аналогично ▼

Неподвижные точки

Неподвижной точкой преобразователя $\mathbb{f} : 2^S \to 2^S$ называется предикат A, такой что $f(A) = A$

Неподвижная точка A преобразователя $\mathbb{f}$ называется наименьшей ($A = \mu Z.\mathbb{f}(Z)$), если она наименьшая по включению среди всех неподвижных точек $\mathbb{f}$, и наибольшей ($A = \nu Z.\mathbb{f}(Z)$), если она наибольшая по включению среди всех неподвижных точек

$\mathbb{f}^i(A)$ — так обозначим i-кратное применение преобразователя $\mathbb{f}$ к предикату A:

- $\mathbb{f}^0(A) = A$
- $\mathbb{f}^i(A) = \mathbb{f}(\mathbb{f}^{i-1}(A))$, если $i > 0$

Неподвижные точки

Лемма. Для любого монотонного $\cup$-непрерывного преобразователя $\mathbb{f}$ верно $\mu Z.\mathbb{f}(Z) = \bigcup\limits_{i=0}^{\infty} \mathbb{f}^i(\emptyset)$

Доказательство. Пусть $A = \bigcup\limits_{i=0}^{\infty} \mathbb{f}^i(\emptyset)$

По определению наименьшей неподвижной точки, достаточно обосновать два факта:

1. $\mathbb{f}(A) = A$
2. Если $\mathbb{f}(B) = B$, то $A \subseteq B$

1. По $\cup$-непрерывности $\mathbb{f}$: $\mathbb{f}(A) = \bigcup_{i=1}^{\infty} \mathbb{f}^i(\emptyset) = \mathbb{f}^0(\emptyset) \cup \bigcup_{i=1}^{\infty} \mathbb{f}^i(\emptyset) = A$

2. Верно $\mathbb{f}^0(\emptyset) = \emptyset \subseteq B$

По монотонности $\mathbb{f}$, для любого $i \in \{0, 1, 2, \dots\}$ верно следующее: если $\mathbb{f}^i(\emptyset) \subseteq B$, то верно и $\mathbb{f}^{i+1}(\emptyset) = \mathbb{f}(\mathbb{f}^i(\emptyset)) \subseteq \mathbb{f}(B) = B$

Значит, для любого $i \in \{0, 1, 2, \dots\}$ верно $\mathbb{f}^i(\emptyset) \subseteq B$, и следовательно, $\bigcup\limits_{i=0}^{\infty} \mathbb{f}^i(\emptyset) \subseteq B$ ▼

Неподвижные точки

Лемма. **Для любого монотонного $\cap$-непрерывного преобразователя $\mathbb{f}$ на S верно $\nu Z.\mathbb{f}(Z) = \bigcap\limits_{i=0}^{\infty} \mathbb{f}^i(S)$**

Доказательство. Аналогично доказательству предыдущей леммы

Лемма. **Для любого монотонного преобразователя на S и любого i, $i \in \mathbb{N}_0$, верно $\mathbb{f}^i(\emptyset) \subseteq \mathbb{f}^{i+1}(\emptyset)$**

Доказательство.

$\mathbb{f}^0(\emptyset) = \emptyset \subseteq \mathbb{f}^1(\emptyset)$

Если $\mathbb{f}^{i-1}(\emptyset) \subseteq \mathbb{f}^i(\emptyset)$ $(i \geq 1)$, то $\mathbb{f}^i(\emptyset) = \mathbb{f}(\mathbb{f}^{i-1}(\emptyset)) \subseteq \mathbb{f}(\mathbb{f}^i(\emptyset)) = \mathbb{f}^{i+1}(\emptyset)$

Значит, *согласно принципу математической индукции*, для любого $i \in \{0, 1, 2, \dots\}$ верно $\mathbb{f}^i(\emptyset) \subseteq \mathbb{f}^{i+1}(\emptyset)$ ▼

Лемма. **Для любого монотонного преобразователя на S и любого i, $i \in \mathbb{N}_0$, верно $\mathbb{f}^i(S) \supseteq \mathbb{f}^{i+1}(S)$**

Доказательство. Аналогично доказательству предыдущей леммы

Неподвижные точки

Лемма. **Для любого монотонного преобразователя $\mathbb{f}$ на конечном множестве S существует k, $k \in \mathbb{N}_0$, такое что $\mathbb{f}^k(\emptyset) = \mathbb{f}^{k+1}(\emptyset)$**

Доказательство. *Предположим от противного*, что это не так

По доказанной ранее лемме, для любого $i \in \{0, 1, \dots\}$ верно $\mathbb{f}^i(\emptyset) \subseteq \mathbb{f}^{i+1}(\emptyset)$

По предположению, $\mathbb{f}^i(\emptyset) \neq \mathbb{f}^{i+1}(\emptyset)$, а значит, $\mathbb{f}^i(\emptyset) \subset \mathbb{f}^{i+1}(\emptyset)$

Следовательно, $|\mathbb{f}^0(\emptyset)| < |\mathbb{f}^1(\emptyset)| < |\mathbb{f}^2(\emptyset)| < \dots$

Но тогда существует m, такое что $|\mathbb{f}^m(\emptyset)| > |S|$, что *противоречит* включению $\mathbb{f}^m(\emptyset) \subseteq S$ ▼

Лемма. **Для любого монотонного преобразователя $\mathbb{f}$ на конечном множестве S существует k, $k \in \mathbb{N}_0$, такое что $\mathbb{f}^k(S) = \mathbb{f}^{k+1}(S)$**

Доказательство. Аналогично доказательству предыдущей леммы

Неподвижные точки

Лемма. **Для любого преобразователя $\mathbb{f}$, любого k, $k \in \mathbb{N}_0$, и любого предиката A верно следующее: если $\mathbb{f}^k(A) = \mathbb{f}^{k+1}(A)$, то для любого m, $m \in \mathbb{N}$, верно $\mathbb{f}^k(A) = \mathbb{f}^{k+m}(A)$**

Доказательство (индукцией по m).

База ($m = 1$): верно по условию леммы

Индуктивный переход: если $\mathbb{f}^k(A) = \mathbb{f}^{k+(m-1)}(A)$, то
$\mathbb{f}^{k+m}(A) = \mathbb{f}^{k+1}(\mathbb{f}^{m-1}(A)) = \mathbb{f}^k(\mathbb{f}^{m-1}(A)) = \mathbb{f}^{k+m-1}(A) = \mathbb{f}^k(A)$ ▼

Неподвижные точки

Объединив результаты всех предложенных лемм, можно легко получить следующие теорему и алгоритм вычисления наименьшей неподвижной точки (LFP) для преобразователя предикатов на конечном множестве S

Теорема (о поиске наименьшей неподвижной точки). Для любого монотонного преобразователя $\mathbb{f}$ на конечном множестве S существует k, $k \in \mathbb{N}_0$, такое что

$$\mathbb{f}^0(\emptyset) \subset \mathbb{f}^1(\emptyset) \subset \cdots \subset \mathbb{f}^k(\emptyset) = \mathbb{f}^{k+1}(\emptyset),$$

и верно $\mu Z.\mathbb{f}(Z) = \mathbb{f}^k(\emptyset)$

Процедура $LFP(M, \mathbb{f})$:

- Положить $X_0 = \emptyset$
- Последовательно для $i \in \{1, 2, \dots\}$:
  - Вычислить $X_i = \mathbb{f}(X_{i-1})$
  - Если $X_i = X_{i-1}$, то завершить процедуру и вернуть X_i

Для такого использования преобразователей требуется подходящее представление — оно будет введено чуть позже

Неподвижные точки

Символьным представлением преобразователя $\mathbb{f}$ назовём отображение $\mathbb{f}_\mathfrak{s}$ множества символьных представлений предикатов в него же, такое что $\mathbb{f}_\mathfrak{s}(\Phi_A) = \Phi_{\mathbb{f}(A)}$

Процедура вычисления наименьшей неподвижной точки очевидным образом переформулируется в терминах символьных представлений

Процедура $LFP_\mathfrak{s}(\mathfrak{M}, \mathbb{f}_\mathfrak{s})$:

- Положить $\Phi^0 = \Phi_\emptyset$
- Последовательно для $i \in \{1, 2, \dots\}$:
  - Вычислить $\Phi^i = \mathbb{f}_\mathfrak{s}(\Phi^{i-1})$
  - Если $\Phi^i \sim \Phi^{i-1}$, то завершить процедуру и вернуть Φ^i

Неподвижные точки

Аналогичные теорема и алгоритмы получаются и для наибольшей неподвижной точки

Теорема (о поиске наибольшей неподвижной точки). Для любого монотонного преобразователя $\mathbb{f}$ на конечном множестве S существует k, $k \in \mathbb{N}_0$, такое что

$$\mathbb{f}^0(S) \supset \mathbb{f}^1(S) \supset \cdots \supset \mathbb{f}^k(S) = \mathbb{f}^{k+1}(S),$$

и верно $\nu Z.\mathbb{f}(Z) = \mathbb{f}^k(S)$

Процедура $GFP(M, \mathbb{f})$:

- Положить $X_0 = S$
- Последовательно для $i \in \{1, 2, \dots\}$:
  - Вычислить $X_i = \mathbb{f}(X_{i-1})$
  - Если $X_i = X_{i-1}$, то завершить процедуру и вернуть X_i

Процедура $GFP_{\mathfrak{s}}(\mathfrak{M}, \mathbb{f}_{\mathfrak{s}})$:

- Положить $\Phi^0 = \Phi_S$
- Последовательно для $i \in \{1, 2, \dots\}$:
  - Вычислить $\Phi^i = \mathbb{f}_{\mathfrak{s}}(\Phi^{i-1})$
  - Если $\Phi^i \sim \Phi^{i-1}$, то завершить процедуру и вернуть Φ^i

Символьный алгоритм (продолжение)

Каждой ctl-формуле φ для предзаданной модели Крипке M можно сопоставить предикат $S_\varphi = Sat(M, \varphi)$

Этот предикат можно представить символьно: $S_{\mathfrak{S}\varphi} = \Phi_{S_\varphi}$

Для заданной модели Крипке комбинация **EX** может расцениваться как преобразователь таких предикатов:

$$\mathbf{EX}_M(S_\varphi) = S_{\mathbf{EX}\varphi}$$

Отличие этой записи от процедуры $Sat_{\mathbf{EX}}$ только в том, что модель Крипке не подаётся на вход, а считается предзаданной (*вспоминаем λ-функции и замыкания из известных языков программирования; здесь это можно считать просто техническим упрощением*)

Символьный алгоритм (продолжение)

Согласно устройству процедуры $Sat_{\mathbf{EX}}$, преобразователь $\mathbf{EX}_M$ можно задать так:

$$\mathbf{EX}_M(Z) = Pre(M, Z)$$

Преобразователь $\mathbb{f}$, такой что $\mathbb{f}(Z) = E$, где E — выражение, зависящее от Z, будем записывать в виде λ-выражения $\lambda Z.E$

Таким образом, $\mathbf{EX}_M = \lambda Z.Pre(M, Z)$

Символьное представление выражения $\lambda Z.E$ — это выражение $\lambda \Phi_Z.\Phi_E$

Например, символьное представление преобразователя $\mathbf{EX}_M$ устроено так:

$$\mathbf{EX}_{\mathfrak{M}} = \lambda \Phi_Z.Pre_{\mathfrak{s}}(\mathfrak{M}, Z)$$

Символьный алгоритм (продолжение)

Лемма. Для любого предиката C преобразователь $\mathbb{f}_C = \lambda Z.C \cap \mathbf{EX}_M(Z)$, является монотонным

Доказательство.

Рассмотрим предикаты A и B, такие что $A \subseteq B$, и покажем, что $\mathbb{f}(A) \subseteq \mathbb{f}(B)$, то есть что для любого состояния s из $\mathbb{f}(A)$ верно $s \in \mathbb{f}(B)$

Так как $s \in \mathbb{f}(A)$, верно $s \in C$ и $s \in \mathbf{EX}_M(A)$

$s \in \mathbf{EX}_M(A)$ означает, что существует состояние s', такое что $s' \in A$ и $s \to s'$

Так как $A \subseteq B$, верно и $s' \in B$

Значит, $s \in \mathbf{EX}_M(B)$, и следовательно, $s \in C \cap \mathbf{EX}_M(B) = \mathbb{f}(B)$ ▼

Символьный алгоритм (продолжение)

Лемма. **Для любой ctl-формулы φ предикат $S_{\mathbf{EG}\varphi}$ является неподвижной точкой преобразователя $\mathbb{f}_\varphi = \lambda Z.S_\varphi \cap \mathbf{EX}_M(Z)$**

Доказательство.

По определению неподвижной точки, достаточно показать, что $S_{\mathbf{EG}\varphi} = S_\varphi \cap \mathbf{EX}_M(S_{\mathbf{EG}\varphi})$

В терминах выполнимости ctl-формул это равенство переписывается как равносильность

$M, s \models \mathbf{EG}\varphi \Leftrightarrow M, s \models \varphi$ и $M, s \models \mathbf{EXEG}\varphi$

По семантике ctl-формул,

- левая часть означает, что в M из s исходит хотя бы один бесконечный путь $s_0, s_1, \ldots$ $(s_0 = s)$, такой что $M, s_0 \models \varphi$, $M, s_1 \models \varphi$, ...
- правая часть означает, что $M, s \models \varphi$ и в M из s исходит хотя бы один бесконечный путь $s_0, s_1, \ldots$ $(s_0 = s)$, такой что $M, s_1 \models \varphi$, $M, s_2 \models \varphi$, ...

Легко видеть, что эти два пункта равносильны ▼

Символьный алгоритм (продолжение)

Лемма. **Для любой ctl-формулы φ предикат $S_{\mathbf{EG}\varphi}$ является наибольшей неподвижной точкой преобразователя $\mathbb{f}_\varphi = \lambda Z.S_\varphi \cap \mathbf{EX}_M(Z)$**

Доказательство. Можете попробовать самостоятельно

Из последней леммы естественно вытекает альтернативный (по сравнению с базовым алгоритмом) вариант процедуры $Sat_{\mathbf{EG}}(M, \varphi)$:

- Вычислить $X = Sat'(M, \varphi)$
- Объявить преобразователь $\mathbb{f} = \lambda Z.X \cap \mathbf{EX}_M(Z)$
- Вернуть предикат $GFP(M, \mathbb{f})$

Эту процедуру несложно представить символьно ($Sat_{\mathfrak{s}\mathbf{EG}}(\mathfrak{M}, \varphi)$):

- Вычислить $\Phi_X = Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi)$
- Объявить символьный преобразователь $\mathbb{f}_{\mathfrak{s}} = \lambda \Phi_Z.\Phi_X \,\&\, \mathbf{EX}_{\mathfrak{M}}(\Phi_Z)$
- Вернуть предикат $GFP_{\mathfrak{s}}(\mathfrak{M}, \mathbb{f}_{\mathfrak{s}})$

Символьный алгоритм (продолжение)

Лемма. **Для любых ctl-формул φ и ψ предикат $S_{\mathbf{E}(\varphi\mathbf{U}\psi)}$ является неподвижной точкой преобразователя $\mathbb{f}_{\varphi,\psi} = \lambda Z.S_\psi \cup (S_\varphi \cap \mathbf{EX}_M(Z))$**

Доказательство.

По определению неподвижной точки, достаточно показать, что $S_{\mathbf{E}(\varphi\mathbf{U}\psi)} = S_\psi \cup (S_\varphi \cap \mathbf{EX}_M(S_{\mathbf{E}(\varphi\mathbf{U}\psi)}))$

В терминах выполнимости ctl-формул это переписывается так:

$$M, s \models \mathbf{E}(\varphi\mathbf{U}\psi) \Leftrightarrow M, s \models \psi \text{ или } (M, s \models \varphi \text{ и } M, s \models \mathbf{EXE}(\varphi\mathbf{U}\psi))$$

Аналогично доказательству такой же леммы для $\mathbf{EG}\varphi$, легко видеть, что в правой части равносильности записано то же, что и в левой ▼

Лемма. **Для любых ctl-формул φ и ψ предикат $S_{\mathbf{E}(\varphi\mathbf{U}\psi)}$ является <u>наименьшей</u> неподвижной точкой преобразователя $\mathbb{f}_{\varphi,\psi}(Z) = S_\psi \cup (S_\varphi \cap \mathbf{EX}_M(Z))$**

Доказательство. Можете попробовать самостоятельно

Символьный алгоритм (продолжение)

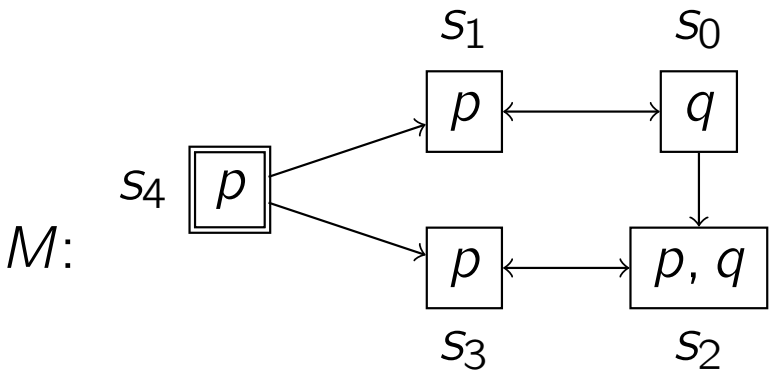
Из последней леммы естественно вытекает альтернативный (по сравнению с базовым алгоритмом) вариант процедуры $Sat_{\mathbf{EU}}(M, \varphi, \psi)$:

- Вычислить $X = Sat'(M, \varphi)$ и $Y = Sat'(m, \psi)$
- Объявить преобразователь $\mathbb{f} = \lambda Z.\, Y \cup (X \cap \mathbf{EX}_M(Z))$
- Вернуть предикат $LFP(M, \mathbb{f})$

Эту процедуру несложно представить символьно ($Sat_{\mathfrak{s}\mathbf{EU}}(\mathfrak{M}, \varphi, \psi)$):

- Вычислить $\Phi_X = Sat'_{\mathfrak{s}}(\mathfrak{M}, \varphi)$ и $\Phi_Y = Sat'_{\mathfrak{s}}(\mathfrak{M}, \psi)$
- Объявить символьный преобразователь $\mathbb{f}_{\mathfrak{s}} = \lambda \Phi_Z.\, \Phi_Y \vee (\Phi_X \,\&\, \mathbf{EX}_{\mathfrak{M}}(\Phi_Z))$
- Вернуть предикат $LFP_{\mathfrak{s}}(\mathfrak{M}, \mathbb{f}_{\mathfrak{s}})$

Символьный алгоритм: пример



$$\varphi = \mathbf{EX}p \,\&\, \neg\mathbf{E}(q\mathbf{UEG}p)$$

Для начала проиллюстрируем модифицированный базовый алгоритм

Начало — такое же, как в исходном базовом алгоритме:

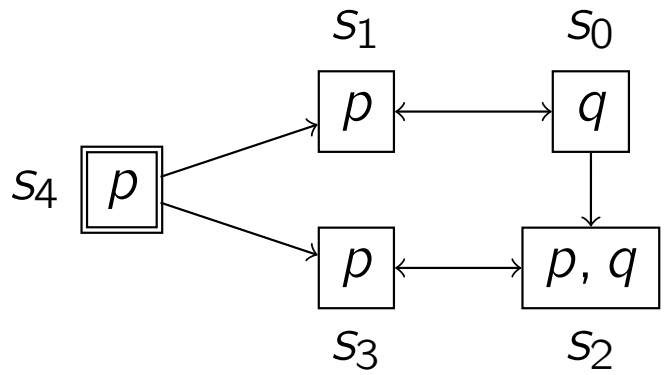
$Sat'(M, p) = \{s_1, s_2, s_3, s_4\}$

$Sat'(M, \mathbf{EX}p) = Sat_{\mathbf{EX}}(M, p) = Pre(Sat'(M, p)) = \{s_0, s_2, s_3, s_4\}$

$Sat'(M, q) = \{s_0, s_2\}$

Символьный алгоритм: пример

M:



$$\varphi = \mathbf{EX}p \,\&\, \neg\mathbf{E}(q\mathbf{UEG}p)$$

Преобразователь для $\mathbf{EG}p$:

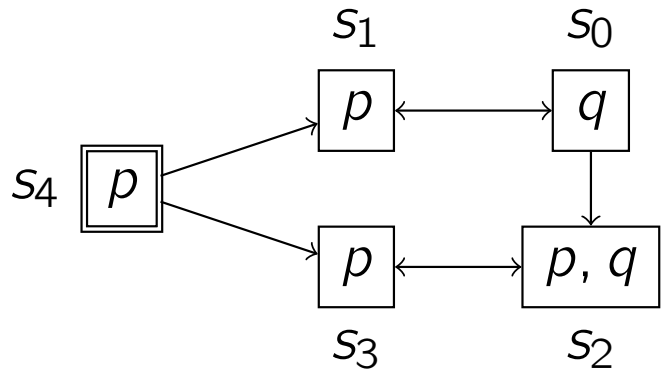
$$\mathbb{f}_1 = \lambda Z.Sat'(M, p) \cap \mathbf{EX}_M(Z) = \lambda Z.\{s_1, s_2, s_3, s_4\} \cap Pre(M, Z)$$

Вычисление наибольшей неподвижной точки $\mathbb{f}_1$:

- $X_0 = S = \{s_0, s_1, s_2, s_3, s_4\}$
- $X_1 = \mathbb{f}_1(X_0) = \{s_1, s_2, s_3, s_4\} \cap Pre(M, S) = \{s_1, s_2, s_3, s_4\} \cap S = \{s_1, s_2, s_3, s_4\}$
- $X_2 = \mathbb{f}_1(X_1) = \{s_1, s_2, s_3, s_4\} \cap \{s_0, s_2, s_3, s_4\} = \{s_2, s_3, s_4\}$
- $X_3 = \mathbb{f}_1(X_2) = \{s_1, s_2, s_3, s_4\} \cap \{s_0, s_2, s_3, s_4\} = \{s_2, s_3, s_4\} = X_2$
- $Sat'(M, \mathbf{EG}p) = \nu Z.\mathbb{f}_1(Z) = X_2 = \{s_2, s_3, s_4\}$

Символьный алгоритм: пример

M:



$$\varphi = \mathbf{EX}p \,\&\, \neg\mathbf{E}(q\mathbf{UEG}p)$$

Преобразователь для $\mathbf{E}(q\mathbf{UEG}p)$:

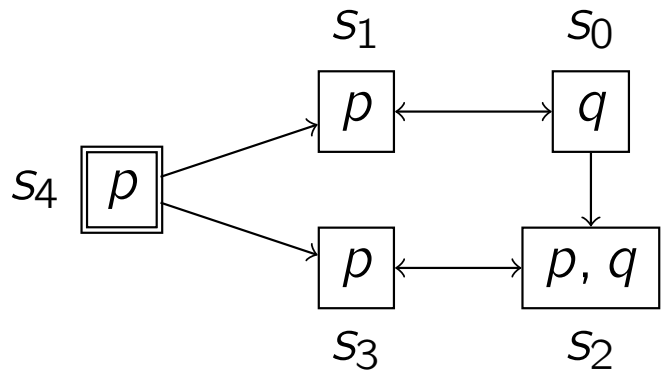
$\mathbb{f}_2 = \lambda Z.Sat'(M, \mathbf{EG}p) \cup (Sat'(M, q) \cap \mathbf{EX}_M(Z)) =$
$\lambda Z.\{s_0, s_3, s_4\} \cup (\{s_2, s_4\} \cap \mathbf{EX}_M(Z))$

Вычисление наименьшей неподвижной точки $\mathbb{f}_2$:

- $X_0 = \emptyset$
- $X_1 = \mathbb{f}_2(X_0) = \{s_2, s_3, s_4\} \cup (\{s_0, s_2\} \cap \emptyset) = \{s_2, s_3, s_4\}$
- $X_2 = \mathbb{f}_2(X_1) = \{s_2, s_3, s_4\} \cup (\{s_0, s_2\} \cap \{s_0, s_2, s_3, s_4\}) = \{s_0, s_2, s_3, s_4\}$
- $X_3 = \mathbb{f}_2(X_1) = \{s_2, s_3, s_4\} \cup (\{s_0, s_2\} \cap \{s_0, s_1, s_2, s_3, s_4\}) =$
  $\{s_0, s_2, s_3, s_4\} = X_2$
- $Sat'(M, \mathbf{E}(q\mathbf{UEG}p)) = \mu Z.\mathbb{f}_2(Z) = \{s_0, s_2, s_3, s_4\}$

Символьный алгоритм: пример

M:



$\varphi = \mathbf{EX}p \,\&\, \neg\mathbf{E}(q\mathbf{UEG}p)$

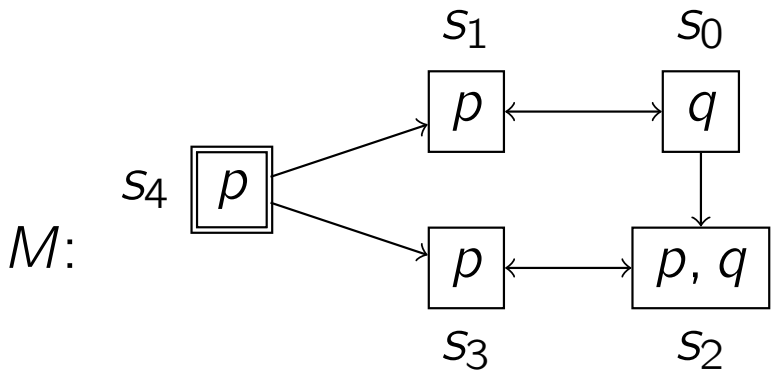
Конец — такой же, как в исходном базовом алгоритме

$Sat'(M, \neg\mathbf{E}(q\mathbf{UEG}p)) = S \setminus Sat'(M, \mathbf{E}(q\mathbf{UEG}p)) = \{s_1\}$

$Sat'(M, \varphi) = Sat'(M, \mathbf{EX}p) \cap Sat'(M, \neg\mathbf{E}(q\mathbf{UEG}p)) = \emptyset$

Так как $\{s_0\} \not\subseteq \emptyset$, можно заключить, что $M \not\models \varphi$

Символьный алгоритм: пример



$$\varphi = \mathbf{EX}p \,\&\, \neg\mathbf{E}(q\mathbf{UEG}p)$$

Теперь всё то же самое в символьном виде

Выберем ширину представления состояний 3 бита, переменные x_0, x_1, x_2 и естественное кодирование: $s_i \mapsto (i)_2^3$

Тогда символьное представление модели устроено так (*для компактности используются формулы*):

- $\Phi_S = x_2 \to \neg x_1 \,\&\, \neg x_0$
- $\Phi_{S_0} = x_2 \,\&\, \neg x_1 \,\&\, \neg x_0$
- $\Phi_\to = \Phi_S \,\&\, \neg x_2' \,\&((x_0 \leftrightarrow x_0') \,\&(x_1 \oplus x_1') \vee \neg x_1 \,\&\, \neg x_0 \,\&\, x_1' \,\&\, \neg x_0')$
- $\Phi_p = \Phi_S \,\&(x_1 \vee x_0)$
- $\Phi_q = \neg x_2 \,\&\, \neg x_0$

А переписать результаты работы модифицированного базового алгоритма в символьном виде можете попробовать сами