

# 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**

# Основная идея символьного алгоритма

**Символьным алгоритмом** решения задачи обычно называют решающий алгоритм, в котором для основных («трудоемких») 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  $\Phi_X$

Представление  $\Phi$  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  $x_1, \dots, x_m$  для некоторого  $m$ , заданного контекстом

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

Основные операции над множествами и отношениями, используемые в базовом алгоритме, естественно переформулируются на языке символических 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:  $A = \{x \mid \exists y : (y, x) \in R\} \mapsto \Phi_A(\vec{x}) = \exists \vec{y} \Phi_R(\vec{x}, \vec{y})$

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

# Символьный алгоритм (начало)

## Дано:

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

**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  $\Phi_X(\vec{x}) = Sat_s(\mathfrak{M}, \varphi)$
- ▶ 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  $\varphi$  в равносильную упрощённую формулу  $\psi$  в базисе **EX**, **EG**, **EU**:  
$$\psi ::= \top \mid p \mid \psi \& \psi \mid \neg\psi \mid \mathbf{EX}\psi \mid \mathbf{EG}\psi \mid \mathbf{E}(\psi\mathbf{U}\psi)$$
- ▶  $Sat(M, \varphi) = Sat'(M, \psi)$ , где процедура  $Sat'$  отличается от  $Sat$  тем, что применяется только к упрощённым формулам

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

# Символьный алгоритм (начало)

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

- ▶ Если  $\varphi = \top$ , то  $Sat'(M, \varphi) = S$
- ▶ Если  $\varphi = p \in AP$ , 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  $Sat'(M, \varphi) = Sat_{EX}(M, \psi)$
- ▶ Если  $\varphi = \mathbf{EG}\psi$ , то  $Sat'(M, \varphi) = Sat_{EG}(M, \psi)$
- ▶ Если  $\varphi = \mathbf{E}(\psi_1 \mathbf{U} \psi_2)$ , то  $Sat'(M, \varphi) = Sat_{EU}(M, \psi_1, \psi_2)$

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

- ▶ Если  $\varphi = \top$ , то  $Sat'_s(\mathfrak{M}, \varphi) = \Phi_S$
- ▶ Если  $\varphi = p \in AP$ , то  $Sat'_s(\mathfrak{M}, \varphi) = \Phi_p$
- ▶ Если  $\varphi = \psi_1 \& \psi_2$ , то  $Sat'_s(\mathfrak{M}, \varphi) = Sat'_s(\mathfrak{M}, \psi_1) \& Sat'_s(\mathfrak{M}, \psi_2)$
- ▶ Если  $\varphi = \neg\psi$ , то  $Sat'_s(\mathfrak{M}, \varphi) = \Phi_S \& \neg Sat'_s(\mathfrak{M}, \psi)$
- ▶ Если  $\varphi = \mathbf{EX}\psi$ , то  $Sat'_s(\mathfrak{M}, \varphi) = Sat_{sEX}(\mathfrak{M}, \psi)$
- ▶ Если  $\varphi = \mathbf{EG}\psi$ , то  $Sat'_s(\mathfrak{M}, \varphi) = Sat_{sEG}(\mathfrak{M}, \psi)$
- ▶ Если  $\varphi = \mathbf{E}(\psi_1 \mathbf{U} \psi_2)$ , то  $Sat'_s(\mathfrak{M}, \varphi) = Sat_{sEU}(\mathfrak{M}, \psi_1, \psi_2)$

## Символьный алгоритм (начало)

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

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

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

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

Символьный алгоритм, процедура  $Sat_sEX(\mathfrak{M}, \varphi)$ :

- ▶ Вычислить  $\Phi_X = Sat'_s(\mathfrak{M}, \varphi)$
- ▶ Вернуть  $Pre_s(\mathfrak{M}, \Phi_X)$



# 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  $f : 2^S \rightarrow 2^S$ , такую что  $f(A)$  — это результат (одного шага) преобразования предиката  $A$

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

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

# 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 множество  $S$
- ▶ 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  $f : 2^S \rightarrow 2^S$  называется

- ▶ **МОНОТОННЫМ**, если для любых предикатов  $A, B$  справедлива импликация

$$A \subseteq B \Rightarrow f(A) \subseteq f(B)$$

- ▶ **U-непрерывным**, если для любой бесконечной монотонно неубывающей последовательности предикатов

$$A_1 \subseteq A_2 \subseteq \dots$$

верно  $f\left(\bigcup_{i=1}^{\infty} A_i\right) = \bigcup_{i=1}^{\infty} f(A_i)$

- ▶ **∩-непрерывным**, если для любой бесконечной монотонно невозрастающей последовательности предикатов

$$A_1 \supseteq A_2 \supseteq \dots$$

верно  $f\left(\bigcap_{i=1}^{\infty} A_i\right) = \bigcap_{i=1}^{\infty} f(A_i)$

## Преобразователи предикатов

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

Доказательство.

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

Так как множество  $S$  конечно, для некоторого  $k$  верно

$$A_k = A_{k+1} = A_{k+2} = \dots$$

Так как  $A_1 \subseteq \dots \subseteq A_k$ , то верно и  $\bigcup_{i=1}^k A_i = A_k$

$$\text{Значит, } \bigcup_{i=1}^{\infty} A_i = A_k$$

Так как  $f$  монотонен, верно  $f(A_1) \subseteq \dots \subseteq f(A_k) = f(A_{k+1}) = \dots$ , и

$$\text{аналогично верно } \bigcup_{i=1}^{\infty} f(A_i) = f(A_k)$$

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

$\cap$ -непрерывность обосновывается аналогично ▼

# Неподвижные точки

**Неподвижной точкой** преобразователя  $f : 2^S \rightarrow 2^S$  называется предикат  $A$ , такой что  $f(A) = A$

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

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

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

## Неподвижные точки

**Лемма.** Для любого монотонного  $\cup$ -непрерывного преобразователя  $f$  верно  $\mu Z.f(Z) = \bigcup_{i=1}^{\infty} f^i(\emptyset)$

**Доказательство.** Пусть  $A = \bigcup_{i=0}^{\infty} f^i(\emptyset)$

По определению наименьшей неподвижной точки, достаточно обосновать два факта:

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

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

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

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

$$\bigcup_{i=0}^{\infty} f^i(\emptyset) \subseteq B \quad \blacktriangledown$$

## Неподвижные точки

**Лемма.** Для любого монотонного  $\cap$ -непрерывного

преобразователя  $f$  на  $S$  верно  $\nu Z.f(Z) = \bigcap_{i=1}^{\infty} f^i(S)$

**Доказательство.** Аналогично доказательству предыдущей леммы

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

**Доказательство.**

$$f^0(\emptyset) = \emptyset \subseteq f^1(\emptyset)$$

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

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

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

**Доказательство.** Аналогично доказательству предыдущей леммы

## Неподвижные точки

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

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

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

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

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

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

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

*Доказательство.* Аналогично доказательству предыдущей леммы



# Неподвижные точки

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

Доказательство (индукцией по  $m$ ).

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

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

# 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  $f$  на конечном множестве  $S$  существует  $k, k \in \mathbb{N}_0$ , такое что

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

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

**Процедура *LFP*( $M, f$ ):**

- ▶ Положить  $X_0 = \emptyset$
- ▶ Последовательно для  $i \in \{1, 2, \dots\}$ :
  - ▶ Вычислить  $X_i = f(X_{i-1})$
  - ▶ 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  $f$  назовём отображение  $f_s$  множества символьных представлений предикатов в него же, такое что  $f_s(\Phi_A) = \Phi_{f(A)}$

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

**Процедура**  $LFP_s(\mathfrak{M}, f_s)$ :

- ▶ Положить  $\Phi^0 = \Phi_\emptyset$
- ▶ Последовательно для  $i \in \{1, 2, \dots\}$ :
  - ▶ Вычислить  $\Phi^i = f_s(\Phi^{i-1})$
  - ▶ Если  $\Phi^i \sim \Phi^{i-1}$ , то завершить процедуру и вернуть  $\Phi^i$

# Неподвижные точки

Аналогичные теорема и алгоритмы получаются и для наибольшей неподвижной точки

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

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

и верно  $\nu Z.f(Z) = f^k(S)$

**Процедура  $GFP(M, f)$ :**

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

**Процедура  $GFP_s(\mathfrak{M}, f_s)$ :**

- ▶ Положить  $\Phi^0 = \Phi_S$
- ▶ Последовательно для  $i \in \{1, 2, \dots\}$ :
  - ▶ Вычислить  $\Phi^i = f_s(\Phi^{i-1})$
  - ▶ Если  $\Phi^i \sim \Phi^{i-1}$ , то завершить процедуру и вернуть  $\Phi^i$

## Символьный алгоритм (продолжение)

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

Этот предикат можно представить символично:  $S_{S_\varphi} = \Phi_{S_\varphi}$

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  $\text{Sat}_{\mathbf{EX}}$  только в том, что модель Крипке не подаётся на вход, а считается предзаданной (*вспоминаем  $\lambda$ -функции и замыкания из известных языков программирования; здесь это можно считать просто техническим упрощением*)

## Символьный алгоритм (продолжение)

Согласно устройству процедуры  $Sat_{EX}$ , преобразователь  $EX_M$  можно задать так:

$$EX_M(Z) = Pre(M, Z)$$

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

Таким образом,  $EX_M = \lambda Z.Pre(M, Z)$

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

**Например**, символьное представление преобразователя  $EX_M$  устроено так:

$$EX_{\mathfrak{M}} = \lambda\Phi_Z.Pre_s(\mathfrak{M}, Z)$$

## Символьный алгоритм (продолжение)

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

Доказательство.

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

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

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

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

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

## Символьный алгоритм (продолжение)

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

Доказательство.

По определению неподвижной точки, достаточно показать, что

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

В терминах выполнимости ctl-формул это равенство переписывается как равносильность

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

По семантике ctl-формул,

- ▶ левая часть означает, что в  $M$  из  $s$  исходит хотя бы один бесконечный путь  $s_0, s_1, \dots$  ( $s_0 = s$ ), 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  $s_0, s_1, \dots$  ( $s_0 = s$ ), 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  $\varphi$  предикат  $S_{EG\varphi}$  является наибольшей неподвижной точкой преобразователя

$$f_{\varphi} = \lambda Z. S_{\varphi} \cap \mathbf{EX}_M(Z)$$

**Доказательство.** Можете попробовать самостоятельно

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

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

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

- ▶ Вычислить  $\Phi_X = Sat'_s(\mathfrak{M}, \varphi)$
- ▶ Объявить символьный преобразователь  $f_s = \lambda \Phi_Z. \Phi_X \& \mathbf{EX}_{\mathfrak{M}}(\Phi_Z)$
- ▶ Вернуть предикат  $GFP_s(\mathfrak{M}, f_s)$

## Символьный алгоритм (продолжение)

**Лемма.** Для любых ctl-формул  $\varphi$  и  $\psi$  предикат  $S_{E(\varphi \cup \psi)}$  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

$$S_{E(\varphi \cup \psi)} = S_{\psi} \cup (S_{\varphi} \cap \mathbf{EX}_M(S_{E(\varphi \cup \psi)}))$$

В терминах выполнимости ctl-формул это переписывается так:

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

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

**Лемма.** Для любых ctl-формул  $\varphi$  и  $\psi$  предикат  $S_{E(\varphi \cup \psi)}$  является наименьшей 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)

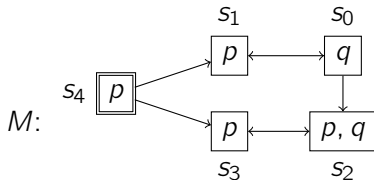
Из последней леммы естественно вытекает альтернативный (по сравнению с базовым алгоритмом) вариант процедуры  $Sat_{EU}(M, \varphi, \psi)$ :

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

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

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

## Символьный алгоритм: пример



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

Для начала проиллюстрируем модифицированный базовый алгоритм

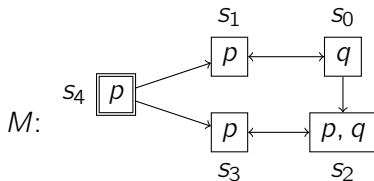
Начало — такое же, как в исходном базовом алгоритме:

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

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

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

# Символьный алгоритм: пример



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

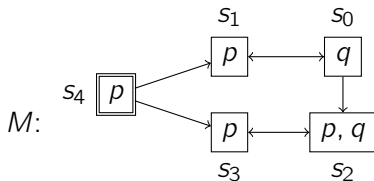
Преобразователь для  $\mathbf{EG}p$ :

$$f_1 = \lambda Z. \text{Sat}'(M, p) \cap \mathbf{EX}_M(Z) = \lambda Z. \{s_1, s_2, s_3, s_4\} \cap \text{Pre}(M, Z)$$

Вычисление наибольшей неподвижной точки  $f_1$ :

- ▶  $X_0 = S = \{s_0, s_1, s_2, s_3, s_4\}$
- ▶  $X_1 = f_1(X_0) = \{s_1, s_2, s_3, s_4\} \cap \text{Pre}(M, S) = \{s_1, s_2, s_3, s_4\} \cap S = \{s_1, s_2, s_3, s_4\}$
- ▶  $X_2 = f_1(X_1) = \{s_1, s_2, s_3, s_4\} \cap \{s_0, s_2, s_3, s_4\} = \{s_2, s_3, s_4\}$
- ▶  $X_3 = f_1(X_2) = \{s_1, s_2, s_3, s_4\} \cap \{s_0, s_2, s_3, s_4\} = \{s_2, s_3, s_4\} = X_2$
- ▶  $\text{Sat}'(M, \mathbf{EG}p) = \nu Z. f_1(Z) = X_2 = \{s_2, s_3, s_4\}$

## Символьный алгоритм: пример



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

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

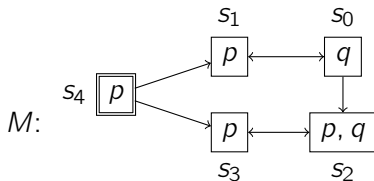
$$\mathbb{f}_2 = \lambda Z. \text{Sat}'(M, \mathbf{EG}p) \cup (\text{Sat}'(M, q) \cap \mathbf{EX}_M(Z)) =$$

$$\lambda Z. \{s_0, s_3, s_4\} \cup (\{s_2, s_4\} \cap \mathbf{EX}_M(Z))$$

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

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

## Символьный алгоритм: пример



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

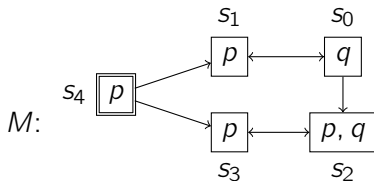
Конец — такой же, как в исходном базовом алгоритме

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

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

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):

- ▶  $\Phi_S = x_2 \rightarrow \neg x_1 \ \& \ \neg x_0$
- ▶  $\Phi_{S_0} = x_2 \ \& \ \neg x_1 \ \& \ \neg x_0$
- ▶  $\Phi_{\rightarrow} = \Phi_S \ \& \ \neg x'_2 \ \& \ ((x_0 \leftrightarrow x'_0) \ \& \ (x_1 \oplus x'_1)) \vee \neg x_1 \ \& \ \neg x_0 \ \& \ x'_1 \ \& \ \neg x'_0$
- ▶  $\Phi_p = \Phi_S \ \& \ (x_1 \vee x_0)$
- ▶  $\Phi_q = \neg x_2 \ \& \ \neg x_0$

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