

Математические методы верификации схем и программ

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

Блок 26

Символьный алгоритм проверки моделей для CTL
(начало)

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

ВМК МГУ, 2024/2025, осенний семестр

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 проверки моделей для CTL имеет невысокую сложность относительно как размера формулы, так и размера модели

Но на практике возникает необходимость применять алгоритм к настолько большим моделям, что даже теоретически очень низкая 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 проверки моделей на практике

Символьный алгоритм проверки моделей 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 $x_1, \dots, x_m$ 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 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: $\Phi[\vec{x}/\vec{y}] = \Phi[x_1/y_1, \dots, x_m/y_m]$

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 φ

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

Базовый алгоритм, основная процедура:

1. Вычислить множество $X = \text{Sat}(M, \varphi) = \mathfrak{P}_{\text{sat}}(M, \varphi)$
2. Проверить включение $S_0 \subseteq X$
 - ▶ То есть проверить соотношение $S_0 \setminus X = \emptyset$
3. Вернуть результат проверки предыдущего пункта

Символьный алгоритм, основная процедура:

1. Вычислить $\Phi_X(\vec{x}) = \mathfrak{F}_{\text{sat}}(\mathfrak{M}, \varphi)$ ($\mathfrak{F}_{\text{sat}}$ описана далее)
2. Проверить соотношение $\Phi_{S_0} \& \neg \Phi_X \sim \Phi_{\emptyset}$
3. Вернуть результат проверки предыдущего пункта

Символьный алгоритм (начало)

Базовый алгоритм, процедура $\mathfrak{P}_{sat}(M, \varphi)$:

1. Используя известные равносильности, преобразовать φ в равносильную упрощённую формулу ψ в базисе **EX**, **EG**, **EU**:
$$\psi ::= \mathfrak{t} \mid p \mid \psi \& \psi \mid \neg\psi \mid \mathbf{EX}\psi \mid \mathbf{EG}\psi \mid \mathbf{E}(\psi\mathbf{U}\psi)$$
2. $\mathfrak{P}_{sat}(M, \varphi) = \mathfrak{P}'_{sat}(M, \psi)$

Символьный алгоритм, процедура $\mathfrak{F}_{sat}(\mathfrak{M}, \varphi)$:

1. Дословно как выше
2. $\mathfrak{F}_{sat}(\mathfrak{M}, \varphi) = \mathfrak{F}'_{sat}(\mathfrak{M}, \psi)$

Символьный алгоритм (начало)

Базовый алгоритм, процедура $\mathfrak{P}'_{sat}(M, \varphi)$:

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

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

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

Символьный алгоритм (начало)

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

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

Базовый алгоритм, процедура $\mathfrak{P}_{EX}(M, \varphi)$:

- ▶ Вычислить $X = \mathfrak{P}'_{sat}(M, \varphi)$
- ▶ Вернуть множество $Pre(M, X)$

Символьный алгоритм, процедура $\mathfrak{F}_{EX}(\mathfrak{M}, \varphi)$:

- ▶ Вычислить $\Phi_X = \mathfrak{F}'_{sat}(\mathfrak{M}, \varphi)$
- ▶ Вернуть $\mathfrak{F}_{pre}(\mathfrak{M}, \Phi_X)$

Перед описанием процедур $\mathfrak{F}_{EU}$ и $\mathfrak{F}_{EG}$ символьного алгоритма сформулируем ещё несколько понятий и утверждений, позволяющих устроить эти процедуры «более умно»