

Символьная верификация моделей

Ю. Лифшиц*

25 октября 2005 г.

План лекции

1. Двоичные разрешающие диаграммы
2. Вычисление неподвижной точки
3. Символьный алгоритм верификации STL

1 Двоичные разрешающие диаграммы

1.1 Определения и свойства

Рассмотрим булеву функцию вида

$$F : \{0, 1\}^k \rightarrow \{0, 1\}.$$

(Заметим, что таких функций — 2^{2^k} .) Рассмотрим ориентированное корневое дерево, такое что:

1. исходящая степень любой внутренней вершины равна двум;
2. каждая внутренняя вершина помечена како-либо переменной;
3. одно исходящее ребро любой внутренней вершины помечено 0, другое — 1;
4. на листьях записаны значения функции.

Такое дерево является одним из способов задания булевой функции F и называется **двоичным разрешающим деревом**. Внутренние вершины такого дерева называются **нетерминалами**, а листья — **терминалами**.

Рисунок 1. Пример ДРД. Функция $(a_1 = b_1) \wedge (a_2 = b_2)$

Легко убедиться в том, что любой булевой функции соответствует какое-то ДРД и по любому ДРД можно построить функцию. Действительно, если

*Законспектировал С. Вишняков.

задать какой-нибудь порядок переменных, то любая последовательность нулей и единиц при спуске по дереву (вариантов спуска 2^k) будет взаимно однозначно соответствовать строке из таблицы истинности функции.

Теперь рассмотрим ациклический ориентированный граф, удовлетворяющий тем же условиям, что и ДРД. Такой граф называется **двоичной разрешающей диаграммой**. Если переменные упорядочены каким-либо образом и на любом пути из корня к листьям переменные встречаются именно в таком порядке (не обязательно все), то такая диаграмма называется **упорядоченной двоичной разрешающей диаграммой**.

Далее будем использовать аббревиатуру OBDD (Ordered Binary Decision Diagram).

Рисунок 2. Два варианта OBDD для той же функции. Стрелки можно не рисовать — порядок задается высотой узлов.

Для каждого порядка переменных существует минимальная OBDD. Для разных порядков размеры OBDD могут отличаться. К примеру, для функции $(a_1 \oplus b_1) \& \dots \& (a_n \oplus b_n)$ при разных порядках размер меняется от $3n + 2$ до $3 \cdot 2^n - 1$. (Порядки $a_1, b_1, \dots, a_n, b_n$ и $a_1, \dots, a_n, b_1, \dots, b_n$, соответственно). Есть функции, при любом порядке дающие OBDD экспоненциального размера. Найти оптимальный порядок — NP-трудная задача. Существуют эвристические алгоритмы, подбирающие по формуле порядок близкий к оптимальному (идея, примерно, следующая: переменные, “взаимодействующие” между собой в формуле, ставить близко в OBDD).

Вопрос: OBDD, по сути, — детерминированный конечный автомат (стартовая вершина — самая верхняя, входное слово — набор значений переменных, допускающие состояния — терминалы со значением 1). Можно ли добиться минимализации OBDD, применив алгоритм минимализации ДКА?

Ответ: OBDD можно отождествить с ДКА только зафиксировав predetermined порядок переменных. Применив алгоритм для ДКА, мы не найдем оптимальный порядок, а всего лишь получим минимальную OBDD только для определенного порядка переменных.

1.2 Операции над OBDD

Пусть есть OBDD для функций f и f' . Построим OBDD для

1. $\neg f$;
2. $f \vee f'$;
3. $f \wedge f'$;
4. любой бинарной булевой функции от f и f' .

OBDD для функции $\neg f$ строится очень просто: заменить значения во всех терминальных вершинах на их отрицание.

Для построения OBDD бинарной функции от двух булевых функций используем **алгоритм Бриана**.

Идея алгоритма.

Пусть даны функции f и f' и бинарная операция $*$. Для каждой вершины a OBDD для f рассмотрим OBDD “висящую” на ней. Эта “висящая” на a OBDD задает функцию f_a . Для каждой пары $(a; a')$ будем строить диаграмму для $f_a * f'_{a'}$, где a — вершина OBDD для f , а a' — вершина OBDD для f' . То есть придется подсчитать $|OBDD(f)| * |OBDD(f')|$ диаграмм: сначала диаграммы для нижних вершин, потом используя уже посчитанные диаграммы — для более высоких, пока не доберемся до $f_s * f'_{s'}$, где s и s' — корни диаграмм для f и f' , соответственно.

Формулы Шеннона.

Эти формулы вдохновили Бриана на создание своего алгоритма.

$$f * f' = (x \wedge (f|_{x \rightarrow 1} * f')) \vee (\neg x \wedge (f|_{x \rightarrow 0} * f'))$$

$$f * f' = (x \wedge (f|_{x \rightarrow 1} * f'|_{x \rightarrow 1})) \vee (\neg x \wedge (f|_{x \rightarrow 0} * f'|_{x \rightarrow 0}))$$

Алгоритм Бриана

База. Вычислить OBDD для пар: терминал из f -OBDD, терминал из f' -OBDD. Таким образом, после этого шага у нас есть все комбинации вида $f_t * f'_{t'}$, где t и t' — терминальные вершины диаграмм для f и f' соответственно.

Переход. Пусть посчитаны все OBDD функций $f_a * f'_{a'}$ для всех комбинаций a и a' , высота которых (считая от терминальных вершин) меньше или равна t и t' соответственно. Заметим, что эта высота определяется неоднозначно, так как можно идти по разным путям. Будем считать, что способ определения высоты мы зафиксировали. Теперь можно посчитать OBDD функций $f_a * f'_{a'}$, где высота a равна $t + 1$, а высота a' — меньше или равна t' .

Случай 1. Вершины a и a' помечены одинаковой переменной x . Пусть a_0, a_1, a'_0, a'_1 — их дети (индекс соответствует значению, которым помечено ребро). Тогда рисуем вершину, помеченную x , слева (переход по 0) “подвешиваем” OBDD для $f_{a_0} * f'_{a'_0}$, справа (переход по 1) — для $f_{a_1} * f'_{a'_1}$. Эти OBDD посчитаны по индукционному предположению.

Случай 2. Вершины a и a' помечены разными переменными x и y (соответственно). Тогда OBDD будет выглядеть следующим образом: рисуем вершину (x), слева — OBDD для $f_{a_0} * f'_{a'}$, справа — для $f_{a_1} * f'_{a'}$. Эти OBDD также посчитаны по индукционному предположению.

Каждую вновь полученную OBDD упрощаем.

Работу алгоритма иллюстрирует рисунок.

1.3 OBDD и модель Крипке

Напомним, что **модель Крипке** — один из способов описания программ. Пусть AP — множество атомарных высказываний. Моделью Крипке над AP называется четверка $M = (S, S_0, R, L)$, где:

1. S — конечное множество состояний;
2. $S_0 \subseteq S$ — множество начальных состояний;
3. $R \subseteq S \times S$ — функция переходов;
4. $L : S \rightarrow 2^{AP}$ — функция истинности (задает, какие атомарные высказывания верны в каждом конкретном состоянии).

Пусть $|AP| = n$, тогда можно считать, что $S \subseteq \{0, 1\}^n$.

Для описания модели Крипке зададим:

1. характеристическую функцию f_S для множества S ;
2. характеристическую функцию f_R для отношения $R(x_1, \dots, x_n, x'_1, \dots, x'_n)$.

Рассмотрим пример:

$$S = (a \wedge b') \vee (a \wedge \neg b)$$

$$R = (a \wedge b \wedge a' \wedge \neg b') \vee (a \wedge \neg b \wedge a' \wedge b') \vee (a \wedge \neg b \wedge a' \wedge \neg b')$$

Диаграмма для R :

Диаграмма для S :

2 Вычисление неподвижной точки

2.1 Определения

Рассмотрим отображение $\tau : 2^U \rightarrow 2^U$. Множество $S \subseteq U$ называется **неподвижной точкой** относительно τ , если $\tau(S) = S$.

Множество S — минимальная неподвижная точка, если:

- 1) S — неподвижная точка;
- 2) Любая неподвижная точка содержит S .

Аналогично определяется максимальная неподвижная точка. Минимальная и максимальная неподвижные точки обозначаются как $\mu S . \tau(S)$ и $\nu S . \tau(S)$.

Пусть U конечно. τ называется монотонным, если $X \subset Y \Rightarrow \tau(X) \subset \tau(Y)$.

2.2 Теорема

[Тарский:]

Если отображение τ монотонно, то существуют минимальная и максимальная неподвижные точки.

Доказательство. Предъявим минимальную и максимальную неподвижные точки. Докажем следующие факты:

- 1) $\mu S . \tau(S) = \bigcup_{i=1}^{\infty} \tau^i(\emptyset)$
- 2) $\nu S . \tau(S) = \bigcap_{i=1}^{\infty} \tau^i(U)$

Во-первых, для любого i верно $\tau^i(\emptyset) \subseteq \tau^{i+1}(\emptyset)$ (база: $\emptyset \subseteq \tau(\emptyset)$, индукционный переход — по монотонности τ).

Пусть $S = \bigcup_{i=1}^{\infty} \tau^i(\emptyset)$. Покажем, что $S \subseteq \tau(S)$ и $\tau(S) \subseteq S$. Действительно, поскольку U — конечное множество, $\tau^i(\emptyset) = \tau^{i+1}(\emptyset)$, начиная с какого-то m . Таким образом, $S = \tau^m(\emptyset)$. Но тогда $\tau(S) = \tau^{m+1}(\emptyset) = \tau^m(\emptyset) = S$.

Покажем, что S содержится в любой неподвижной точке S' . Понятно, что $\emptyset \subseteq S'$. Применим τ к обеим частям, учитывая, что S' — неподвижная точка. Получим $\tau(\emptyset) \subseteq S'$. Применим τ m раз, получим $\tau^m(\emptyset) \subseteq S'$. А это и означает, что $S \subseteq S'$. Аналогично можно провести доказательство для максимальной неподвижной точки.

3 Символьный алгоритм верификации CTL

3.1 Постановка задачи

Напомним, что логика CTL строится из конструкций вида:

1. $\neg f$, $f \vee g$;
2. **EX** f (из данной вершины существует ребро в другую вершину, в которой выполнено свойство f);
3. **EG** f (существует путь из данной вершины, на котором в каждой вершине выполнено f);
4. **E** $[fUg]$ (существует путь из данной вершины, на котором до какого-то момента всегда выполняется f , а потом — всегда g).

Задача верификации CTL формулируется следующим образом: даны модель Крипке $M = (S, R, L)$ и CTL-формула f ; необходимо найти множество $\{s \in S \mid M, s \models f\}$ (множество вершин s , для которых выполняется p).

3.2 Алгоритм

Задаем модель Крипке в виде OBDD функций f_S и f_R . Для каждой подформулы формулы f будем строить OBDD состояний, ее выполняющих.

При этом начинаем с атомарных высказываний. Для них делаем подстановку в f_S . Например, для атомарного высказывания A в OBDD для f_S обрезаем все поддеревья, где $A = 0$, и упрощаем OBDD.

Для операций \vee и \neg используем алгоритм Бриана. Осталось разобраться с операциями $\mathbf{EX}f$, $\mathbf{EG}f$ и $\mathbf{E}[fUg]$.

Множество состояний $\mathbf{EX}f$ определяется следующей формулой:

$$\exists v' [f(v') \wedge R(v, v')].$$

OBDD для $f(v')$ у нас есть по индукционному предположению. Используя алгоритм Бриана, получаем диаграмму для $f(v') \wedge R(v, v')$. Из этой диаграммы несложно получить требуемую. Действительно, чтобы получить диаграмму $\exists v' \dots$ достаточно в старой диаграмме найти вершины, помеченные переменной v' , и их ее поддеревьям применить операцию \vee , используя алгоритм Бриана (в этих поддеревьях некоторые вершины могут совпадать — такие вершины надо предварительно продублировать).

Теперь построим OBDD для $\mathbf{EG}f$. Рассмотрим оператор $\tau(Z) = (f \wedge \mathbf{EX})(Z)$. Этот оператор действует на подмножествах S следующим образом: по множеству состояний Z он возвращает состояния, где

1. выполнена формула f ;
2. есть исходящее ребро в множество Z .

Лемма 1. Отображение τ монотонно.

Если $Z \subseteq Z'$, то множество вершин, из которых есть ребро в Z , содержится в множестве вершин, из которых есть ребро в Z' . То же справедливо для пересечений этих множеств с множеством вершин, в которых выполнено f . Значит, $\tau(Z) \subseteq \tau(Z')$.

Лемма 2. Для любого состояния из $F = \bigcap_{i=1}^{\infty} \tau^i(S)$ выполнена формула $\mathbf{EG}f$.

Заметим, что F — неподвижная точка оператора τ (τ монотонен). Значит, $\tau(F) = F$. Иными словами, в каждой вершине F выполняется f и из каждой вершины F есть ребро в другую вершину F . Тогда взяв любую вершину $v_0 \in F$, мы можем перейти из нее в другую вершину $v_1 \in F$ и так далее. Значит, для любой вершины v_0 существует какой-то путь (очевидно, в нем будут циклы) $v_0 \dots v_i \dots$, где $\forall! v_j \in F$, значит, для каждой вершины этого пути выполняется f .

Лемма 3. Все состояния, для которых выполнено $\mathbf{EG}f$ попали в $\bigcap_{i=1}^{\infty} \tau^i(S)$. Рассмотрим произвольную вершину $x \in \mathbf{EG}f$. Во-первых, $x \in S$. Из x есть бесконечный путь, в каждой вершине которого выполняется f . Значит, из x есть ребро в S . Значит, $x \in \tau(S)$. Но из x есть ребро в вершину, в которой выполняется f и из которой тоже есть бесконечный путь, на всех вершинах которого выполняется f . Значит, $x \in \tau^2(S)$. Таким образом, по индукции доказывается, что $\forall x \in \tau^i(S)$. А из этого утверждения следует доказываемая лемма.

Объединив Леммы 2 и 3, получим, что выполняющее множество $\mathbf{EG}f$ является наибольшей неподвижной точкой оператора $\tau(Z) = f \wedge \mathbf{EX}Z$.

На основе полученных результатов построим OBDD для $\mathbf{EG}f$:

1. Начинаем с OBDD для f_S

2. Последовательно вычисляем OBDD для $(f \wedge \mathbf{EX})^i(S)$

3. Останавливаемся, когда степень $i + 1$ равна i -ой

Аналогичным образом строится OBDD для функции $\mathbf{E}[fUg]$. Таким образом, мы научились строить выполняющее множество любой формулы CTL для заданной модели Крипке.