

# Математические методы верификации схем и программ

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

## Блок 25

Двоичные решающие диаграммы:  
BDD, OBDD, ROBDD

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

ВМК МГУ, 2025, сентябрь–декабрь

# Вступление

**Двоичная решающая диаграмма** (Binary Decision Diagram; BDD) — это один из наиболее популярных способов эффективного представления булевых функций для таких задач, в которых удаётся символично представить основные структуры данных решения

Существует несколько широко применяющихся русских вариантов названия этой структуры данных:

- ▶ Вместо «двоичная» иногда пишут «бинарная»
- ▶ Вместо «решающая диаграмма» иногда пишут «разрешающая диаграмма» или «диаграмма решений»

# Вступление

Символьные представления нередко основываются на BDD или аналогичных структурах, и поэтому считается, что для более полного понимания тонкостей эффективной работы с символьными представлениями следует иметь общие знания об устройстве BDD

В эти знания входит *как минимум* то,

- ▶ как устроены BDD (синтаксис) и какие булевы функции ими реализуются (семантика)
- ▶ как строить BDD по другим представлениям (например, формулам)
- ▶ какие операции можно выполнять над BDD и как устроены соответствующие алгоритмы

# BDD: синтаксис

Двоичная решающая диаграмма над упорядоченным набором переменных  $x_1, \dots, x_n$  — это конечный ациклический ориентированный граф, устроенный так

Вершины BDD обычно называют **узлами**

Особо выделены два **терминальных узла**: 0 и 1 — а остальные узлы называются **внутренними**

Каждый внутренний узел  $v$  помечен одной из переменных ( $var(v)$ )

Из каждого внутреннего узла исходят ровно две дуги, одна помечена символом 0 ( $\xrightarrow{0}$ ), другая — символом 1 ( $\xrightarrow{1}$ )

Из терминального узла не исходит ни одной дуги

Узел, в который из  $v$  ведёт дуга  $\xrightarrow{b}$ , будем обозначать записью  $v[b]$

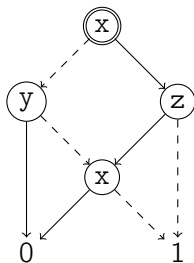
Узлы  $v[0]$  и  $v[1]$  будем называть соответственно **младшим потомком** ( $low(v)$ ) и **старшим потомком** ( $high(v)$ ) узла  $v$

Дуги  $\xrightarrow{1}$  и  $\xrightarrow{0}$  также будем изображать соответственно как сплошную и пунктирную стрелки без помечающих чисел

Один (и только один) из узлов BDD объявлен **корнем**

# BDD: синтаксис

## Пример



Корень BDD изображён двойным контуром

Для узла  $y$  верно следующее:

- ▶  $high(v) = v[1] = 0$
- ▶  $var(low(v)) = var(v[0]) = x$

## BDD: семантика

Каждому узлу  $v$  BDD  $\mathcal{D}$  над переменными  $x_1, \dots, x_n$  сопоставим  $n$ -местную булеву формулу  $\Phi_v^{\mathcal{D}}$ :

- ▶  $\Phi_0^{\mathcal{D}} = 0$
- ▶  $\Phi_1^{\mathcal{D}} = 1$
- ▶ Для внутреннего узла  $v$ :  $\Phi_v^{\mathcal{D}} = \neg \text{var}(v) \ \& \ \Phi_{\text{low}(v)}^{\mathcal{D}} \vee \text{var}(v) \ \& \ \Phi_{\text{high}(v)}^{\mathcal{D}}$

В узле  $v$  BDD  $\mathcal{D}$  над переменными  $x_1, \dots, x_n$  **реализуется**  $n$ -местная булева функция  $f_v^{\mathcal{D}}$  — это функция, **реализуемая формулой**  $\Phi_v^{\mathcal{D}}$

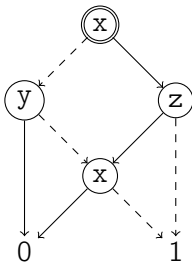
BDD  $\mathcal{D}$  **реализует булеву функцию**  $f^{\mathcal{D}}$ , реализуемую в её корне

Альтернативный способ определения значения  $f^{\mathcal{D}}(\alpha_1, \dots, \alpha_n)$  (значения  $\mathcal{D}$  на **оценке переменных**  $\xi = [x_1/\alpha_1, \dots, x_n/\alpha_n]$ ):

- ▶ Обойдём BDD как граф, начав в корне
- ▶ Если текущий узел  $v$  — внутренний, то следующим обходится узел  $v[\xi(\text{var}(v))]$
- ▶ Если обход достиг узла 0 или 1, то этот достигнутый узел объявляется значением  $f^{\mathcal{D}}(\alpha_1, \dots, \alpha_n)$

# BDD: семантика

Пример диаграммы  $\mathcal{D}$ :



Пронумеруем внутренние узлы так: верхний, левый, правый, нижний —  $v_0, v_1, v_2, v_3$

$$\Phi_{v_3}^{\mathcal{D}} = \neg x \& 1 \vee x \& 0 \sim \neg x$$

$$\Phi_{v_2}^{\mathcal{D}} = \neg z \& 1 \vee z \& \Phi_{v_3}^{\mathcal{D}} \sim \neg z \vee \neg x$$

$$\Phi_{v_1}^{\mathcal{D}} = \neg y \& \Phi_{v_3}^{\mathcal{D}} \vee y \& 0 \sim \neg y \& \neg x$$

$$\Phi_{v_0}^{\mathcal{D}} = \neg x \& \Phi_{v_1}^{\mathcal{D}} \vee x \& \Phi_{v_2}^{\mathcal{D}} \sim \neg x \& \neg y \vee x \& \neg z$$

То есть диаграммой  $\mathcal{D}$  реализуется та же функция, что и формулой  $\Phi_{v_0}^{\mathcal{D}}$

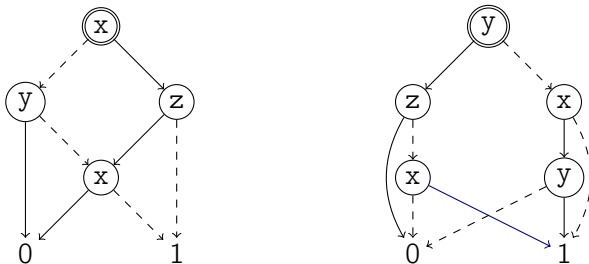
Это подтверждает и обход на оценке  $[x/1, y/0, z/1]$ :  $\textcircled{x} \xrightarrow{1} \textcircled{z} \xrightarrow{1} \textcircled{x} \xrightarrow{1} 0$

# BDD: семантика

BDD  $\mathcal{D}_1$ ,  $\mathcal{D}_2$  (а также BDD  $\mathcal{D}$  и формула  $\varphi$ ) над переменными  $x_1, \dots, x_n$  называются **эквивалентными** ( $\mathcal{D}_1 \sim \mathcal{D}_2$ ;  $\mathcal{D} \sim \varphi$ ), если ими реализуются одинаковые функции над этим набором переменных

Существенная часть применения BDD — это проверка их эквивалентности, и способ такой проверки неочевиден: одна и та же функция может быть реализована существенно разными по структуре диаграммами

**Например**, такие две BDD эквивалентны:



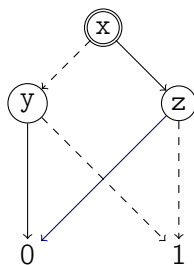
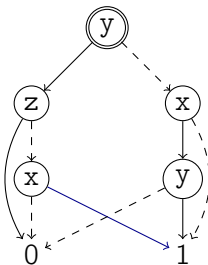
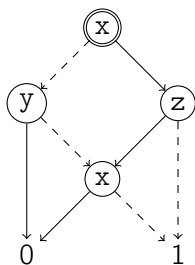


# OBDD

Один из факторов, влияющих на трудность проверки эквивалентности BDD — это возможность записывать переменные в узлах «хаотично»

BDD  $\mathcal{D}$  над набором переменных  $x_1, \dots, x_n$  называется **упорядоченной** (Ordered BDD; **OBDD**), если для любой дуги  $v \rightarrow w$ , ведущей во внутренний узел, верно  $var(v) < var(w)$  для естественного порядка  $<$  переменных:  $x_i < x_j \Leftrightarrow i < j$

**Например**, среди изображённых ниже эквивалентных BDD над набором переменных  $x, y, z$  только самая правая упорядочена (является OBDD)

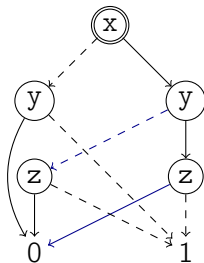
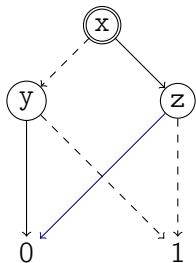


# ROBDD

Для практических целей хотелось бы иметь **каноническое** представление булевых функций решающими диаграммами: единственное, и при этом такое, с которым было бы достаточно удобно работать (строить и преобразовывать)

OBDD не могут быть использованы в этом качестве: эквивалентные OBDD могут иметь заметно разную структуру

**Например**, следующие две OBDD над  $x, y, z$  эквивалентны, но имеют различное число узлов и хотя и не очень сильно, но всё же заметно различный внешний вид

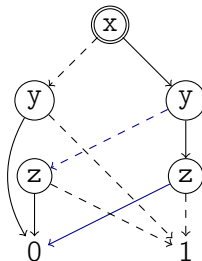
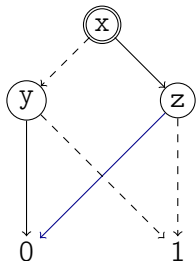


# ROBDD

OBDD называется **приведённой** (или **сокращённой**, или **редуцированной**; Reduced OBDD; **ROBDD**), если для неё выполнены два условия:

1. Все внутренние узлы достижимы из корня
2. В любой паре различных узлов реализуются различные функции

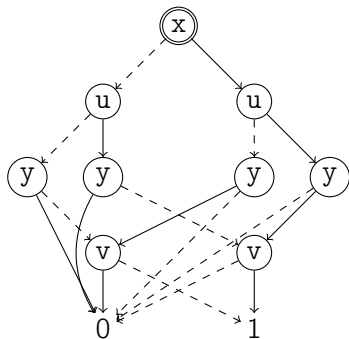
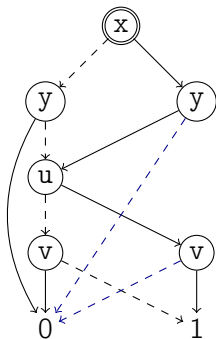
**Например**, среди изображённых ниже OBDD над  $x, y, z$  левая является приведённой, а правая — нет



# ROBDD

При использовании ROBDD очень важную роль играет выбор порядка переменных, над которой строится BDD

**Например**, ROBDD, эквивалентная формуле  $(x \leftrightarrow y) \& (u \leftrightarrow v)$  для порядка  $x < y < u < v$  имеет заметно меньше узлов, чем для порядка  $x < u < y < v$ :



# ROBDD

Так как BDD обсуждаются только как вспомогательный инструмент, основные их свойства и алгоритмы будут приводиться без обоснования

**Утверждение.** Если ROBDD  $D_1$  и  $D_2$  над общим набором переменных эквивалентны, то они изоморфны как размеченные графы

Изоморфизм ROBDD проверить несложно:

- ▶ В изоморфизм  $\phi$  (как двуместное отношение на вершинах) входит пара корней диаграмм
- ▶ Если  $(v, w) \in \phi$ , то и  $(low(v), low(w)), (high(v), high(w)) \in \phi$
- ▶ Используя два предыдущих пункта, можно вычислить множество всех пар узлов, которые обязаны входить в изоморфизм  $\phi$
- ▶ Диаграммы изоморфны  $\Leftrightarrow$  в каждой паре  $(v, w) \in \phi$ 
  - ▶ либо оба узла внутренние и  $var(v) = var(w)$ ,
  - ▶ либо оба узла терминальные и  $v = w$

Чтобы ROBDD были «полезным» представлением булевых функций, кроме простоты проверки равенства реализуемых функций необходимо уметь эффективно строить и преобразовывать диаграммы

# ROBDD: приведение OBDD

**Дано:** OBDD  $\mathcal{D}$  над  $x_1, \dots, x_n$

**Требуется** построить ROBDD  $\mathcal{D}^*$  над  $x_1, \dots, x_n$ , эквивалентную  $\mathcal{D}$

**Алгоритм** устроен несложно — пока это возможно, выполнять следующие три преобразования:

1. Если в диаграмме есть внутренний узел  $v$ , отличный от корня и не имеющий ни одной входящей дуги, то удалить  $v$  и исходящие дуги
2. Если в диаграмме есть внутренний узел  $v$ , такой что  $low(v) = high(v)$ , то удалить  $v$  и перенаправить все дуги, входившие в  $v$ , в  $low(v)$ ; если  $v$  был корнем, то объявить  $low(v)$  корнем
3. Если в диаграмме есть различные внутренние узлы  $v, w$ , такие что  $var(v) = var(w)$ ,  $low(v) = low(w)$ ,  $high(v) = high(w)$  и  $w$  не корень, то удалить узел  $w$  и перенаправить все дуги, входившие в  $w$ , в  $v$

# ROBDD: простейшие диаграммы

$\mathcal{D}^\varphi$  — так обозначим ROBDD, эквивалентную формуле  $\varphi$

ROBDD для простейших функций устроены очень просто

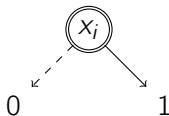
$\mathcal{D}^0$ :



$\mathcal{D}^1$ :



$\mathcal{D}^{x_i}$ :



# ROBDD: отрицание

**Дано:** ROBDD  $\mathcal{D}^\varphi$  над  $x_1, \dots, x_n$

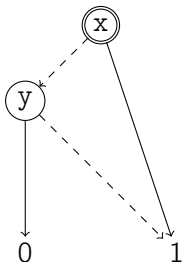
**Требуется** построить ROBDD  $\neg \mathcal{D}^\varphi = \mathcal{D}^{\neg\varphi}$  над  $x_1, \dots, x_n$

**Алгоритм** устроен несложно:

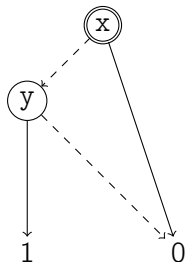
1. Поменять местами терминальные узлы

**Пример**

$\mathcal{D}^{x \vee \neg y}$ :



$\neg \mathcal{D}^{x \vee \neg y}$ :





## ROBDD: подстановка константы

$\varphi[x/e]$  — формула, получающаяся из  $\varphi$  подстановкой выражения  $e$  на место каждого вхождения переменной  $x$

**Дано:** ROBDD  $\mathcal{D}^\varphi$  над  $x_1, \dots, x_n$ , значение  $b \in \{0, 1\}$

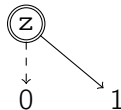
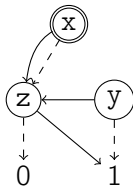
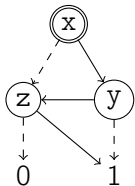
**Требуется** построить ROBDD  $\mathcal{D}^\varphi[x/b] = \mathcal{D}^{\varphi[x/b]}$  над  $x_1, \dots, x_n$

**Алгоритм** и тут устроен несложно:

1. Для каждого узла  $v$ , такого что  $\text{var}(v) = x$ , перенаправить входящие дуги в узел  $v[b]$
2. Привести получившуюся OBDD

**Пример** ( $\varphi = z \& (\neg x \vee y) \vee x \& \neg y$ )

$\mathcal{D}^\varphi$ :      OBDD  $\mathcal{D} \sim \varphi[y/1]$ :       $\mathcal{D}^\varphi[y/1]$ :



# ROBDD: применение двуместной операции

**Дано:** ROBDD  $\mathcal{D}^\varphi$ ,  $\mathcal{D}^\psi$  над  $x_1, \dots, x_n$ , двуместная булева операция  $\circ$

**Требуется** построить ROBDD  $\mathcal{D}^\varphi \circ \mathcal{D}^\psi = \mathcal{D}^{\varphi \circ \psi}$  над  $x_1, \dots, x_n$

**Алгоритм** устроен сложнее всех предыдущих, но тоже достаточно просто:

- ▶ Если  $n = 0$ , то обе формулы  $\varphi$ ,  $\psi$  — это константы 0, 1, и результат — простейшая диаграмма  $\mathcal{D}^{\varphi \circ \psi}$
- ▶ Иначе:
  - ▶ Рекурсивно построить диаграммы  $\mathcal{D}_0 = \mathcal{D}^\varphi[x_1/0] \circ \mathcal{D}^\psi[x_1/0]$  и  $\mathcal{D}_1 = \mathcal{D}^\varphi[x_1/1] \circ \mathcal{D}^\psi[x_1/1]$  над  $x_2, \dots, x_n$
  - ▶ Объединить  $\mathcal{D}_0$  и  $\mathcal{D}_1$ , считая все внутренние узлы попарно различными, добавить узел  $x_1$ , объявить его корнем и направить из него дуги  $\xrightarrow{0}$ ,  $\xrightarrow{1}$  в бывшие корни диаграмм  $\mathcal{D}_0$ ,  $\mathcal{D}_1$  соответственно
  - ▶ Привести полученную OBDD

А почему это работает?

# ROBDD: построение по формуле

Формулу  $\varphi$  над  $\neg$  и двуместными операциями можно трактовать как схему применения операций к функциям, реализуемым простейшими формулами  $0, 1, x_i$

Диаграмму  $\mathcal{D}^\varphi$  можно получить как результат применения операций согласно той же схеме к соответствующим простейшим диаграммам  $\mathcal{D}^0, \mathcal{D}^1, \mathcal{D}^{x_i}$

**Например,**  $\mathcal{D}^x \& \neg y \vee z = \mathcal{D}^x \& \neg \mathcal{D}^y \vee \mathcal{D}^z$

# ROBDD: производные операции

На практике к ROBDD применяются и другие удобные «производные» операции:

- ▶ Для формулы  $\varphi$ :  $\exists x \varphi = \varphi[x/0] \vee \varphi[x/1]$   
Для диаграммы:  $\exists x \mathcal{D}^\varphi = \mathcal{D}^{\exists x \varphi}$
- ▶ Для формулы  $\varphi$ :  $\forall x \varphi = \varphi[x/0] \& \varphi[x/1]$   
Для диаграммы:  $\forall x \mathcal{D}^\varphi = \mathcal{D}^{\forall x \varphi}$
- ▶ Для формулы  $\varphi$ :  $relprod(\varphi, \psi, y_1, \dots, y_k) = \exists y_1 \dots \exists y_k (\varphi \& \psi)$   
Для диаграммы:  $relprod(\mathcal{D}^\varphi, \mathcal{D}^\psi, y_1, \dots, y_k) = \mathcal{D}^{relprod(\varphi, \psi, y_1, \dots, y_k)}$ 
  - ▶ Пусть формулой  $\varphi$  задаётся двуместное отношение  $R_1$  над комплектами переменных  $\tilde{x}$  и  $\tilde{y}$ , а формулой  $\psi$  — одноместное отношение  $R_2$  над комплектом  $\tilde{y}$  или двуместное над  $\tilde{y}$  и  $\tilde{z}$   
Тогда  $relprod(\varphi, \psi, \tilde{y})$  задаёт отношение
    - ▶  $\exists \tilde{y}(R_1(\tilde{x}, \tilde{y}) \& R_2(\tilde{y}))$ : множество всех  $\tilde{x}$ , входящих в отношение  $R_1$  с хотя бы одним  $\tilde{y}$  из  $R_2$  — или
    - ▶  $\exists \tilde{y}(R_1(\tilde{x}, \tilde{y}) \& R_2(\tilde{y}, \tilde{z}))$ : множество всех пар  $(\tilde{x}, \tilde{z})$ , соединяющихся посредством  $R_1$  и  $R_2$  через хотя бы один  $\tilde{y}$

## ROBDD: производные операции

Задача **QBF** (выполнимости квантифицированных булевых формул): для заданной произвольной формулы вида  $Q_1x_1 \dots Q_nx_n(\varphi)$ , где  $Q_i \in \{\exists, \forall\}$  и  $\varphi$  — КНФ над  $x_1, \dots, x_n$ , проверить соотношение  $Q_1x_1 \dots Q_nx_n(\varphi) \not\approx 0$ . Эта задача **весьма трудна**: является PSPACE-полной.

Но её можно решить при помощи малого числа операций над BDD:

- ▶ Построить ROBDD по заданной формуле (с кванторами)
- ▶ Проверить изоморфизм полученной BDD и  $\mathcal{D}^0$

Сложность работы с ROBDD «скрыта» в возрастании их размера при применении операций:

- ▶ При применении одной операции размер возрастает несильно (*полиномиально с низкой степенью*)
- ▶ При многократном применении операций размер может возрастать весьма существенно

Но хотя использование ROBDD и неэффективно в теории, оно всё же считается эффективным на практике: при должном выборе порядка переменных нередко получаются достаточно компактные диаграммы