Введение в моделирование и верификацию аппаратных и программных систем

Лекция 7: Двоичные разрешающие диаграммы (BDD). Символьная проверка моделей

Борис Юрьевич Конев

konev@liverpool.ac.uk

Liverpool University

Октябрь-Ноябрь 2007

План

В прошлый раз

- Системы переходов как модели вычислений
- Логический формализм для описания свойств
  - LTL
  - CTL
- Алгоритм проверки CTL моделей с явным представлением
- NuSMV

Сегодня

- Структуры данных и алгоритмы для эффективной реализации

Система переходов

$S = (Q, T, q_0, L)$

- Q — конечное множество состояний
- $T \subseteq Q \times Q$ — множество переходов
- q_0 — начальное состояние
- $L : Q \to 2^P rop$ — пометка пропозициональными переменными

Проверка CTL моделей

- По данной системе переходов $S = (Q, T, q_0, L)$ и CTL формуле φ проверить, верно ли, что $S \models \varphi$.
- Мы представили динамический алгоритм, который помечает состояния системы переходов S формулами таким образом, что состояние p помечается формулой ψ тогда и только тогда, когда $(S, p) \models \psi$.

Проблема: Множества состояний может быть чрезвычайно велико $> 10^{20}$

Идея

- При явном представлении, требуется хотя бы один объект для представления одного состояния системы переходов
  - Для представления n состояний n объектов
  - “Унарная запись”
- Чтобы записать номер состояния, требуется $\log n$ бит

Представим системы переходов с помощью булевых функций

Представление множеств булевыми функциями

- Пусть Σ – некоторое множество; $|\Sigma| \leqslant 2^n$
- Представим элементы Σ как вектора из n бит
  - Пример: $V = \{\text{ДД} : 1..31, \text{ММ} : 1..12, \text{ГГГГ} : -4000..4000\}$
    $n = \log 31 + \log 12 + \log 8001 = 5 + 4 + 13 = 22$
- Представим $S \subseteq \Sigma$ как функцию $\{0,1\}^n \to \{0,1\}$ (характеристическую функцию S)

$$f_S(s) = \begin{cases} 1, & \text{если } s \in S \\ 0, & \text{если } s \notin S \end{cases}$$

  - Пример: $D \subseteq V$ реальные даты

$$f_D(\underbrace{00001}_{1}\ \underbrace{0001}_{\text{январь}}\ \underbrace{0}_{+}\ \underbrace{011111000110}_{1980}) = 1$$

$$f_D(\underbrace{11111}_{31}\ \underbrace{0010}_{\text{февраль}}\ \underbrace{0}_{+}\ \underbrace{011111000110}_{1980}) = 0$$

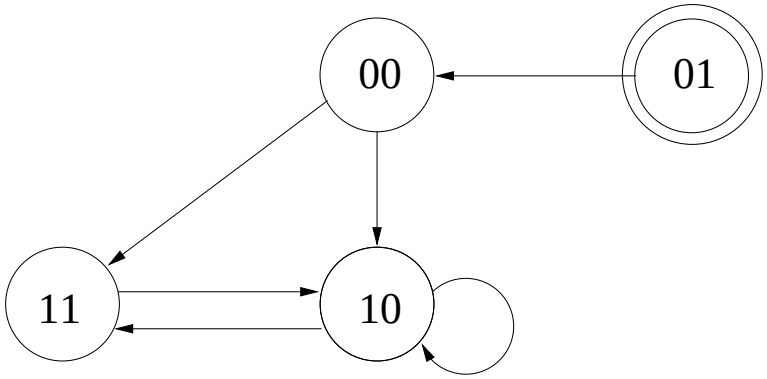
Символьное представление систем переходов

Дана система переходов (Q, T, q_0, L)
Пусть $n = \log Q$, $m = \log L$

- Q представим как $f_Q : \{0,1\}^n \to \{0,1\}$
- $T \subseteq Q \times Q$ представим как $f_T : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$
- q_0 представим как вектор из n бит
- $L : Q \to 2^{Prop}$ представим как $M : Prop \to 2^Q$
  $f_M : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}$

Пример

$q_0 = 01$
$T = \{(01, 00),\ (00, 11),\ (00, 10),$
$(11, 10),\ (10, 11), (10, 10)\}$



$$f_T = (\neg x_0 \wedge x_1 \wedge \neg x_0' \wedge \neg x_1') \vee (\neg x_0 \wedge \neg x_1 \wedge x_0' \wedge x_1') \wedge \ldots$$

Вычисление Pre и Post

В динамическом алгоритме нам были необходимы

- $\mathrm{Pre}(P) = \{q \in Q \mid \exists p \in P, \exists e \in E : (q, e, p) \in T\}$
- $\mathrm{Post}(P) = \{q \in Q \mid \exists p \in P, \exists e \in E : (p, e, q) \in T\}$

Допустим, $P \subseteq Q$ и T представлены характеристическими функциями:

- $f_P(x_1, \ldots, x_n) : \{0,1\}^n \to \{0,1\}$
- $f_T(x_1, \ldots, x_n, x'_1, \ldots, x'_n) : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$

Тогда

$$f_{\mathrm{Pre}(P)}(x_1, \ldots, x_n) = \exists x'_1, \ldots, x'_n (f_P(x'_1, \ldots, x'_n) \wedge f_T(x_1, \ldots, x_n, x'_1, \ldots, x'_n))$$

и

$$f_{\mathrm{Post}(P)}(x'_1, \ldots, x'_n) = \exists x_1, \ldots, x_n (f_P(x_1, \ldots, x_n) \wedge f_T(x_1, \ldots, x_n, x'_1, \ldots, x'_n))$$

Требования к представлению булевых функций

- Компактность
- Быстрое вычисление $f_1 \wedge f_2$, $f_1 \vee f_2$, $\neg f_1$
- Быстрое вычисление $g(x) = \exists y f(x, y)$

Не бывает

На практике, двоичные разрешающие диаграммы (binary decision diagrams, BDD) эффективно реализуют эти операции

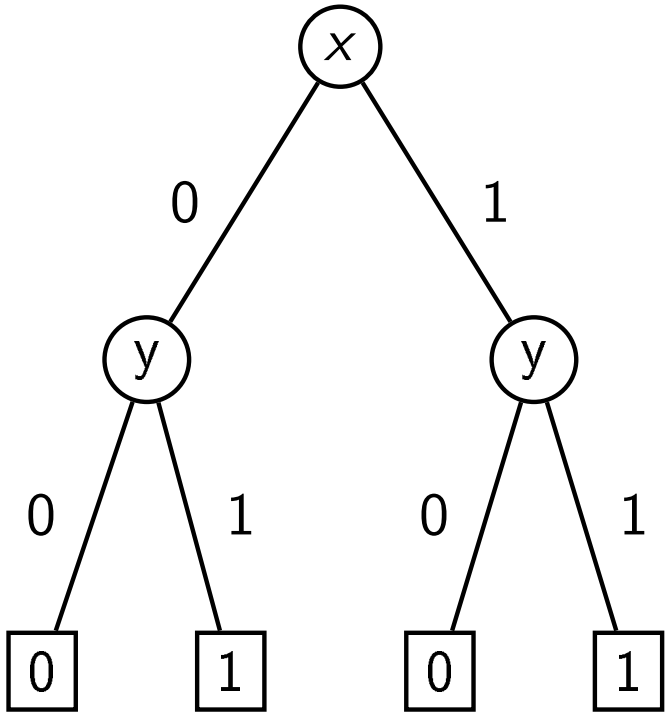
BDD

Двоичная разрешающая диаграмма это ациклический ориентированный граф с корнем, содержащий вершины двух типов

- Терминальные вершины — вершины выходной степени 0, помеченные $\boxed{0}$ или $\boxed{1}$
- Нетерминальные вершины, помеченные пропозициональными переменными, имеющие две исходящие дуги, помеченные 0 и 1 (или low и high)

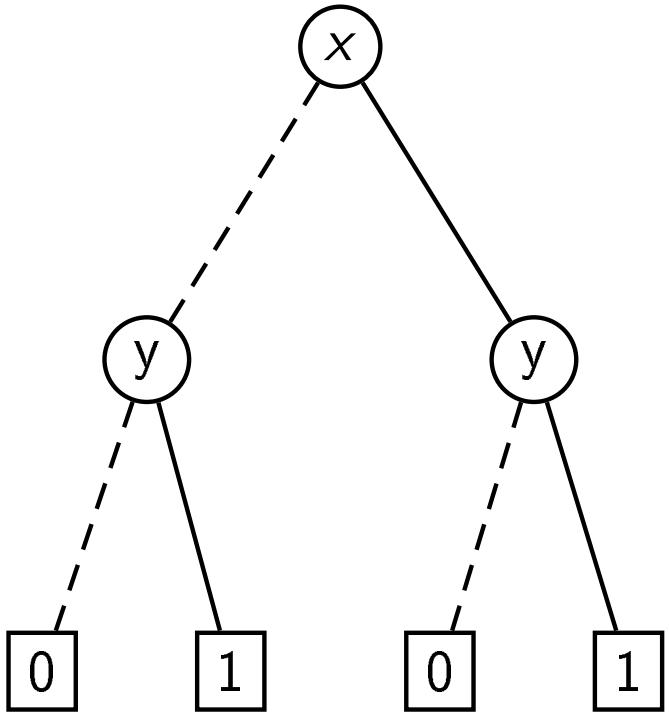
Пример

$$f(x, y) = (\neg x \wedge y) \vee (x \wedge y)$$



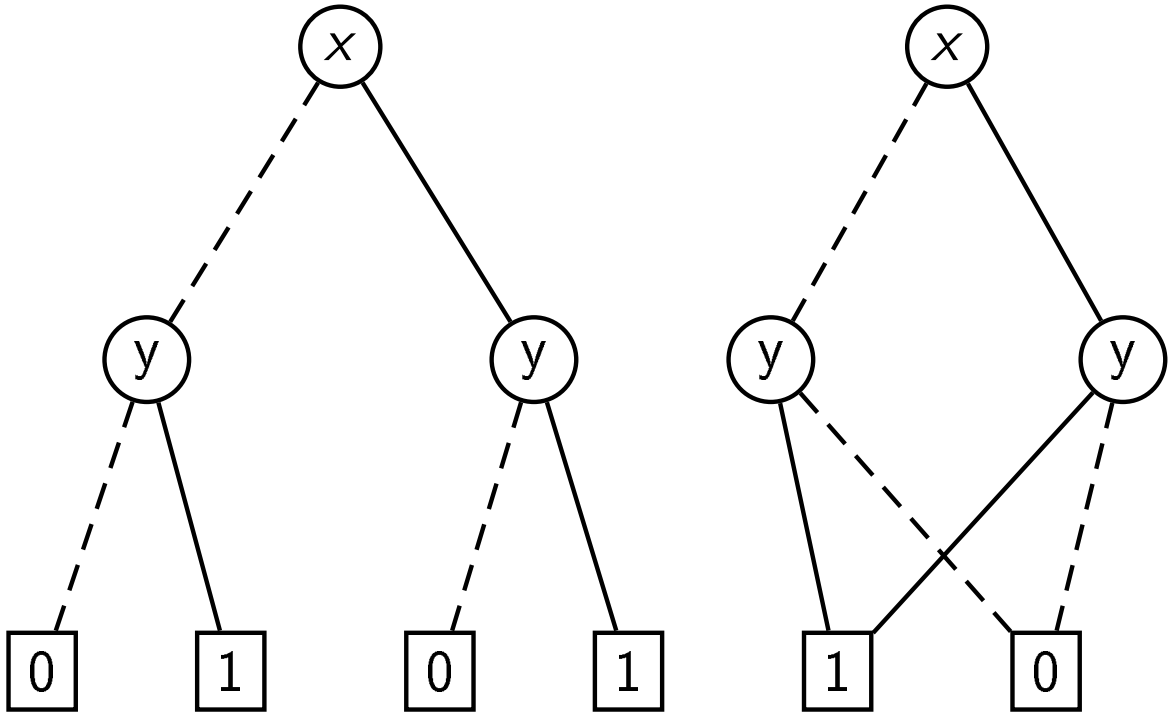
Пример

$$f(x, y) = (\neg x \wedge y) \vee (x \wedge y)$$



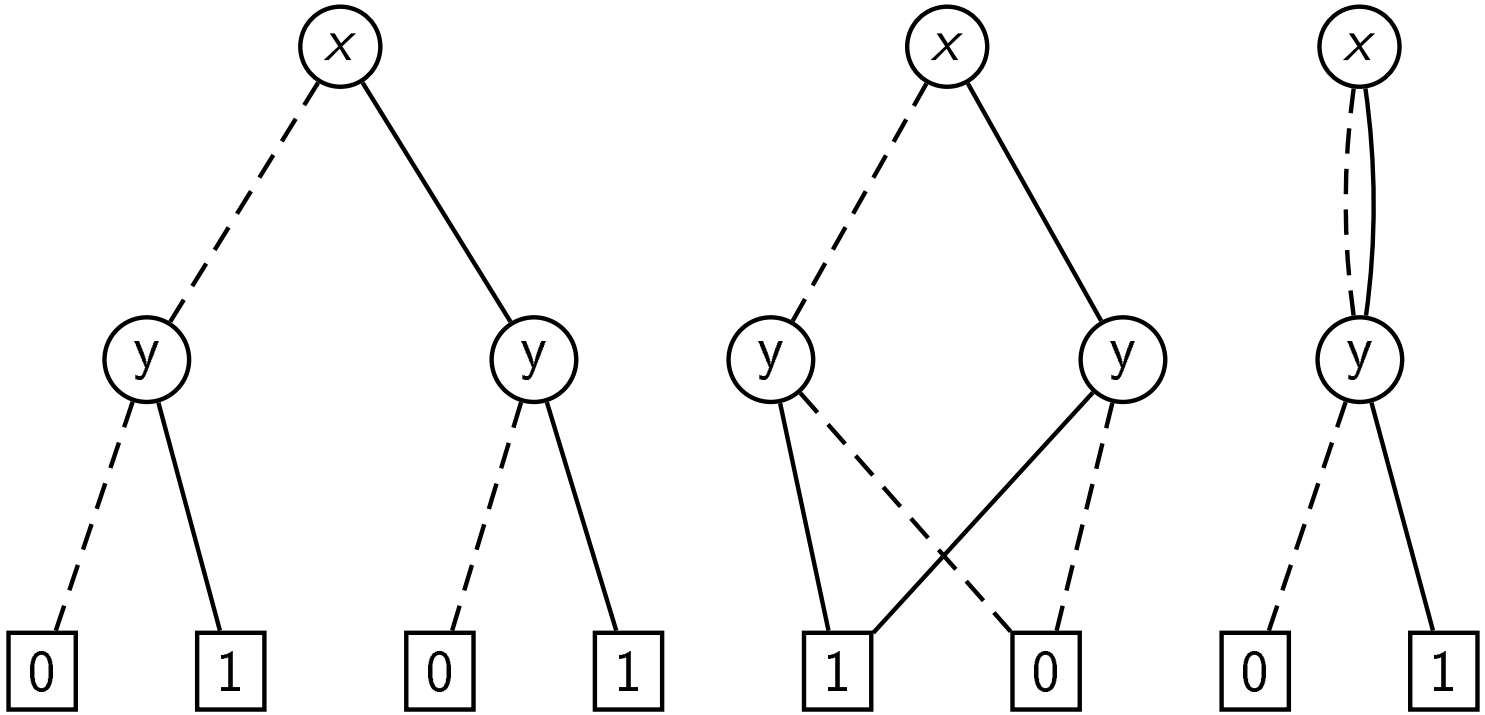
Пример

$f(x, y) = (\neg x \wedge y) \vee (x \wedge y)$



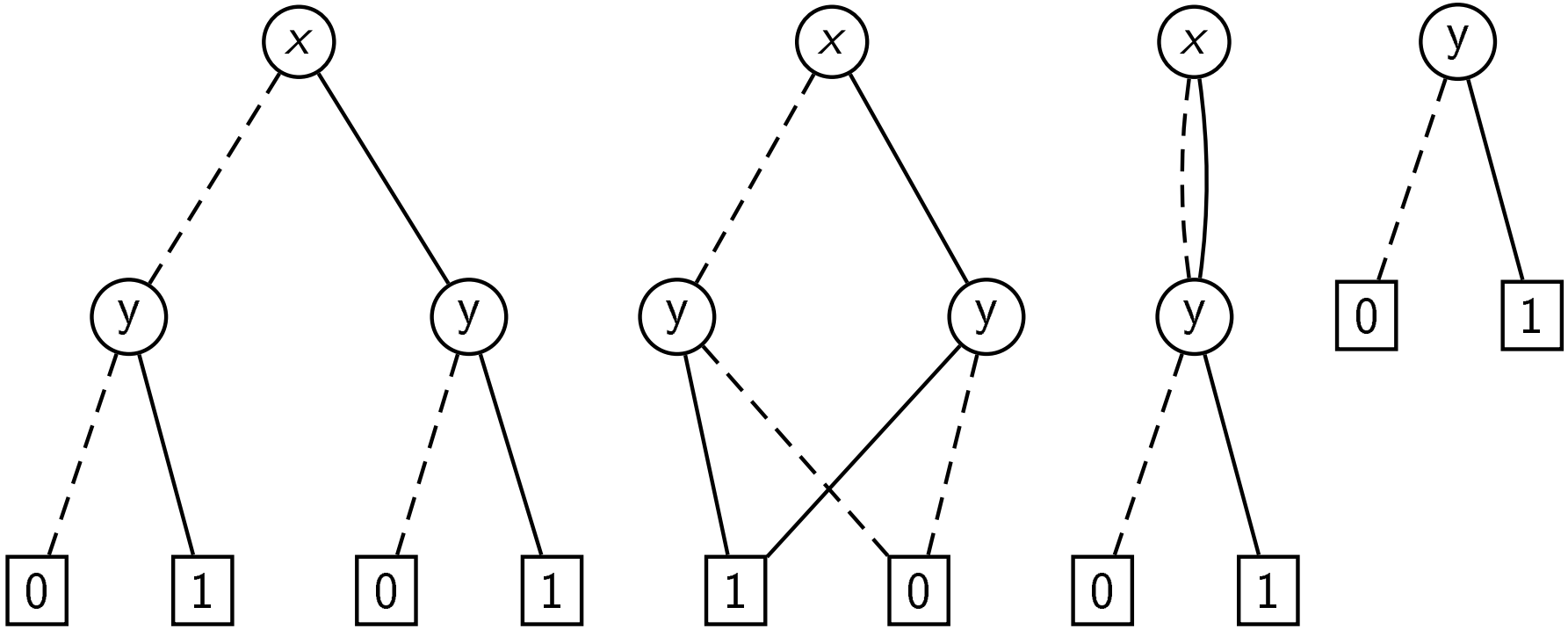
Пример

$f(x, y) = (\neg x \wedge y) \vee (x \wedge y)$



Пример

$f(x, y) = (\neg x \wedge y) \vee (x \wedge y)$



Reduce

Повторять, пока возможно

- Удалить повторные вхождения терминальных вершин и перенаправить дуги
- Удалить повторяющиеся нетерминальные вершины: Если для нетерминальных вершин u и v мы имеем
  $var(u) = var(v)$, $low(u) = low(v)$, $high(u) = high(v)$,
  удалить одну из них и перенаправить входящие дуги на другую вершину
- Удалить избыточные проверки: если для нетерминальной вершины v мы имеем $low(v) = high(v)$, удалить v и перенаправить все входящие дуги на $low(v)$.

Сложность: $O(|B| \cdot \log |B|)$

Упорядоченные бинарные разрешающие диаграммы

Упорядоченные бинарные разрешающие диаграммы (Ordered Binary Decision Diagrams, OBDD)

- Ни одна переменная не повторяется ни на одном пути
- Переменные встречаются в одном и том же порядке на любом пути от корня к терминальной вершине
  - Например, $x_3 < x_1 < x_{n-1} < \cdots < x_2$

Теорема (Бриан)

Для любой булевой функции $f(x_1, \ldots, x_n) : \{0,1\}^n \to \{0,1\}$ и любого порядка на переменных $x_1, \ldots, x_n$ существует единственная редуцированная OBDD, представляющая f.

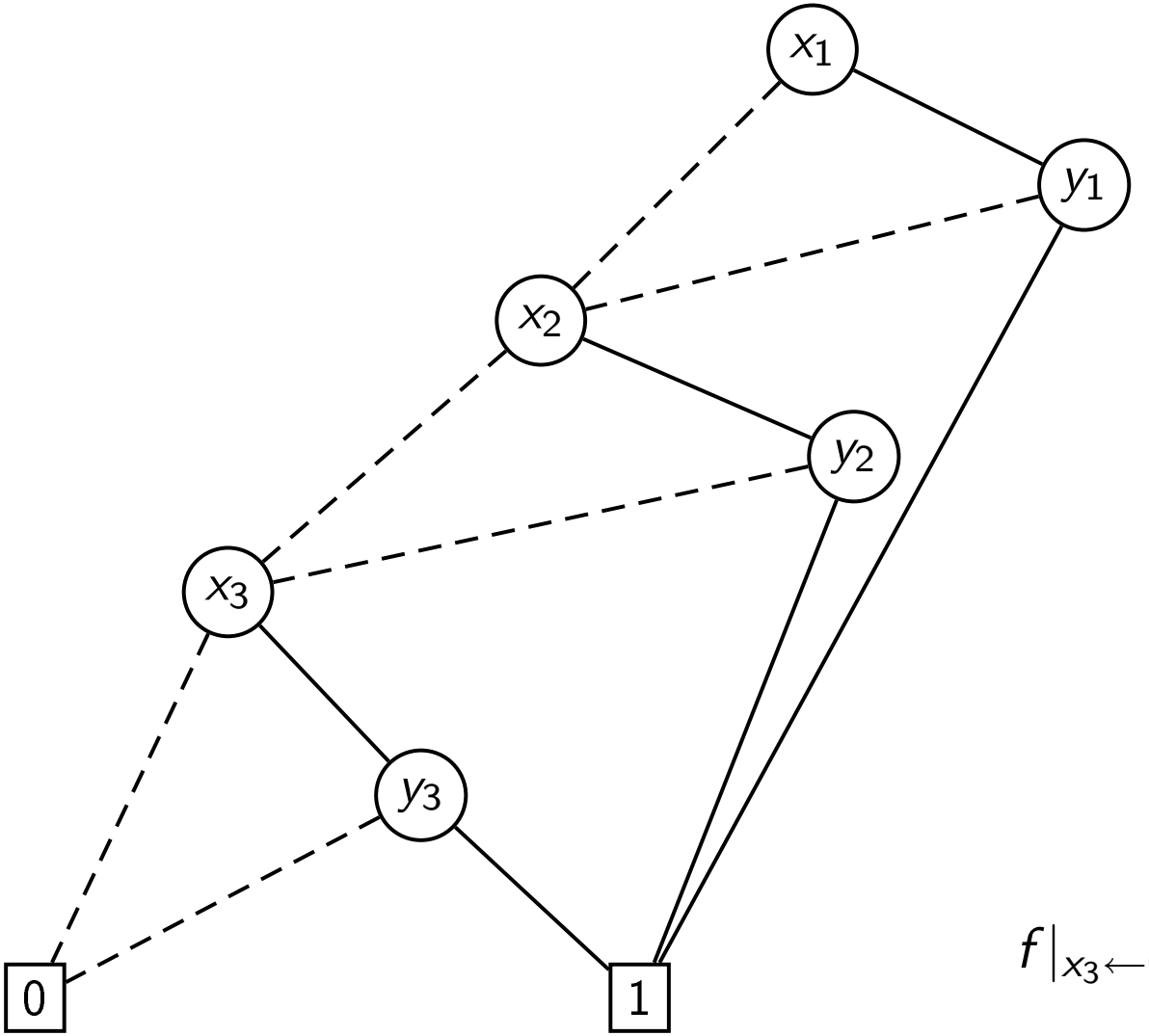
Операции над OBDD: ограничение (Restrict)

$f|_{x_i \leftarrow b}(x_1, \ldots, x_n) = f(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots x_n)$

Алгоритм:

- Если $x_i = var(root)$, взять low-подграф, если $b = 0$, или $high$-подграф, если $b = 1$
- Иначе для каждой OBDD вершины v т.ч. существует дуга $v \rightarrow w$, где $var(w) = x_i$ перенаправить дугу на $low(w)$, если $b = 0$, или $high(w)$, если $b = 1$,
- Применить Reduce к полученной OBDD

Пример



$f|_{x_3 \leftarrow 0} = ?$ $\qquad$ $f|_{x_3 \leftarrow 1} = ?$

Операции над BDD

- Даны: OBDD для f_1 и f_2.
- Пусть $\star = \wedge, \vee, \rightarrow \ldots$
- Рекурсивная функция $Apply(root(f_1), root(f_2), \star)$ строит OBDD для $f_1 \star f_2$

- Используем разложение Шеннона:
  Для любой булевой функции f верно

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

(см. также Буль, 1854)

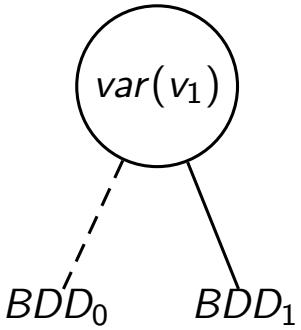
  В частности,

$$f_1 \star f_2 = (\neg x \wedge (f_1|_{x \leftarrow 0} \star f_2|_{x \leftarrow 0})) \vee (x \wedge (f_1|_{x \leftarrow 1} \star f_2|_{x \leftarrow 1}))$$

$Apply(v_1, v_2, \star)$

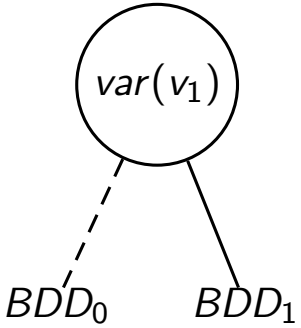
- Если и v_1 и v_2 — терминальные вершины, вернуть $\boxed{v_1 \star v_2}$
- Если $var(v_1) = var(v_2)$, построить

  $BDD_0 = Apply(low(v_1), low(v_2), \star)$
  $BDD_1 = Apply(high(v_1), high(v_2), \star)$
  вернуть Reduce примененный к



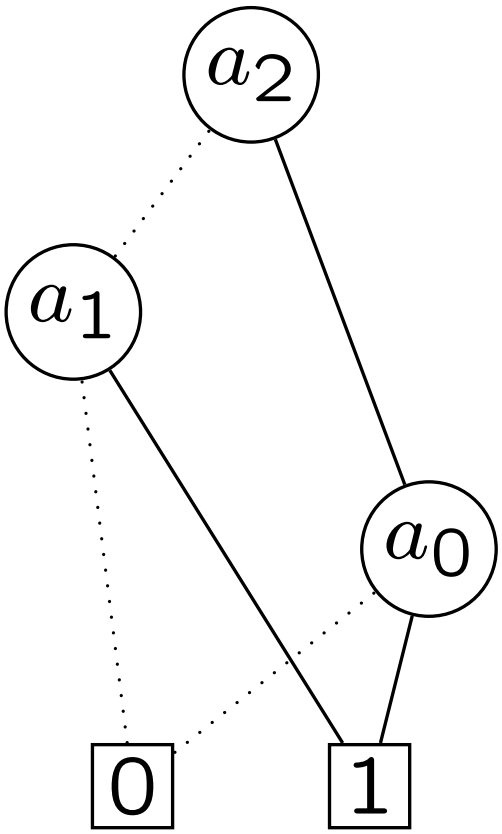
- Если $var(v_1) < var(v_2)$, то f_2 не зависит от $var(v_1)$
  Построить

  $BDD_0 = Apply(low(v_1), v_2, \star)$
  $BDD_1 = Apply(high(v_1), v_2, \star)$
  вернуть Reduce примененный к

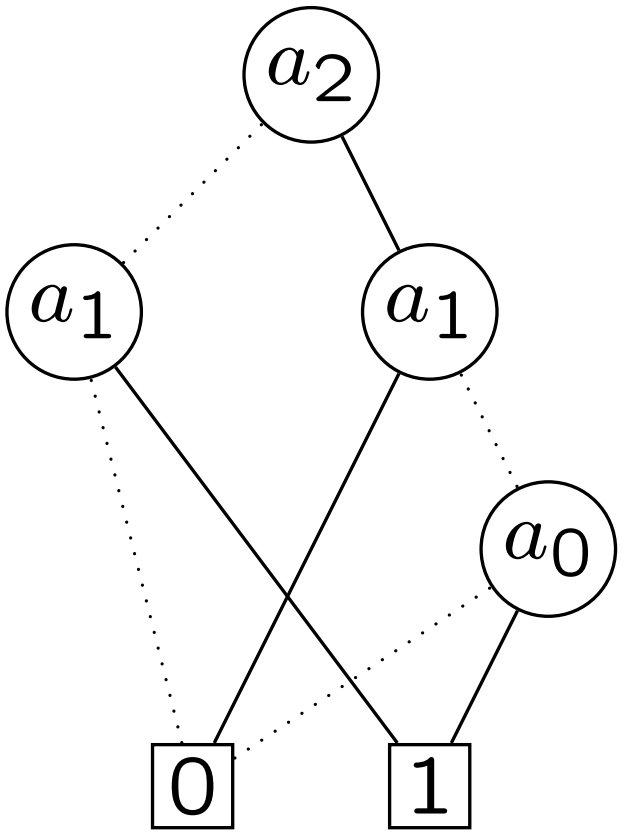


- Случай $var(v_1) > var(v_2)$ аналогичен

Пример



f



g

Применение отрицания

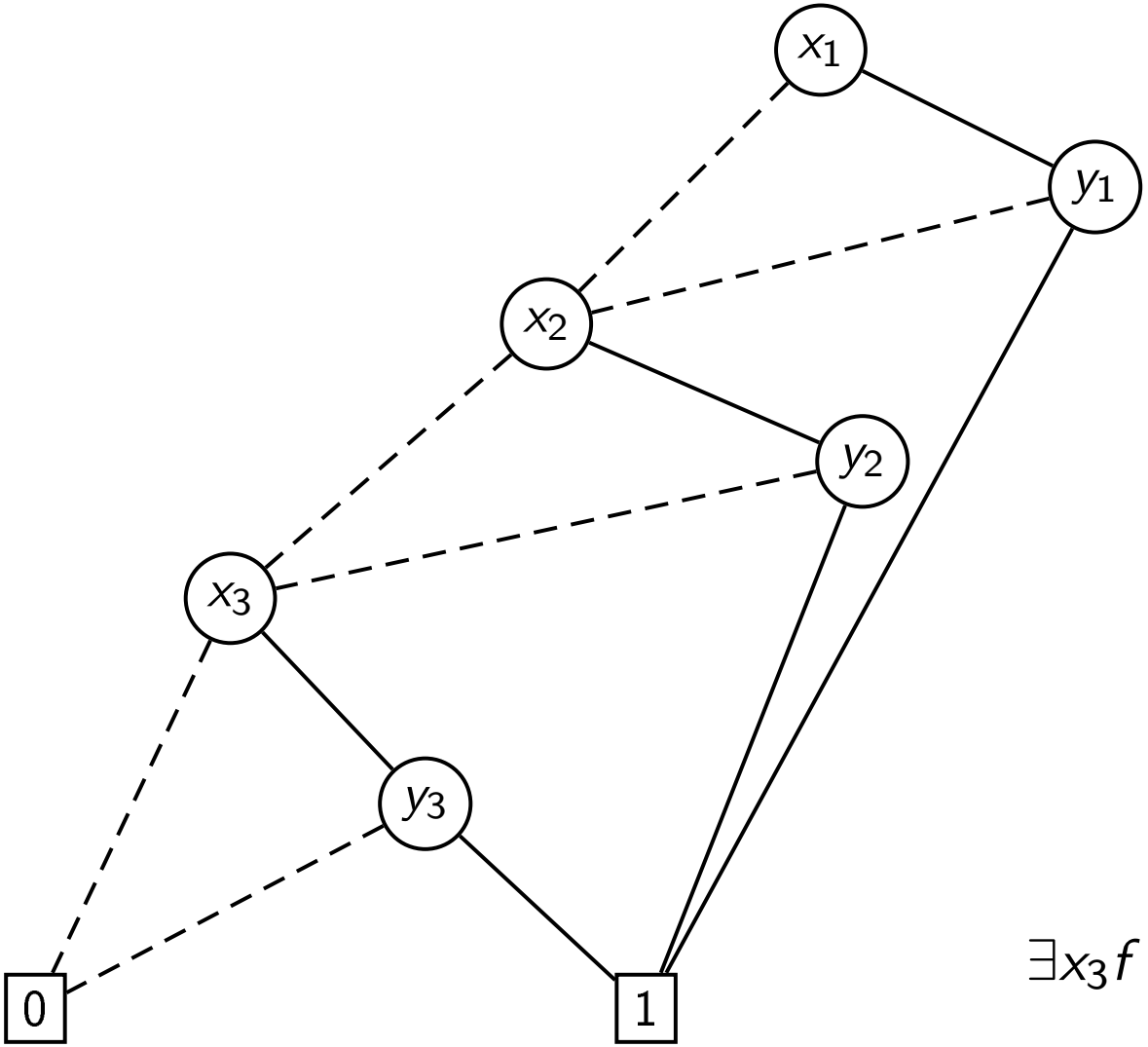
Хотя OBDD для отрицания B может быть построена применением функции *Apply*, есть гораздо более эффективный метод: поменять местами $\boxed{1}$ и $\boxed{0}$

Квантификация

$$\exists x_i f(x_1, \ldots, x_n) = f|_{x_i \leftarrow 0} \vee f|_{x_i \leftarrow 1}$$

$$\forall x_i f(x_1, \ldots, x_n) = f|_{x_i \leftarrow 0} \wedge f|_{x_i \leftarrow 1}$$

Пример



$\exists x_3 f = ?$ $\qquad$ $\exists x_2 \exists x_3 f = ?$

Сложность

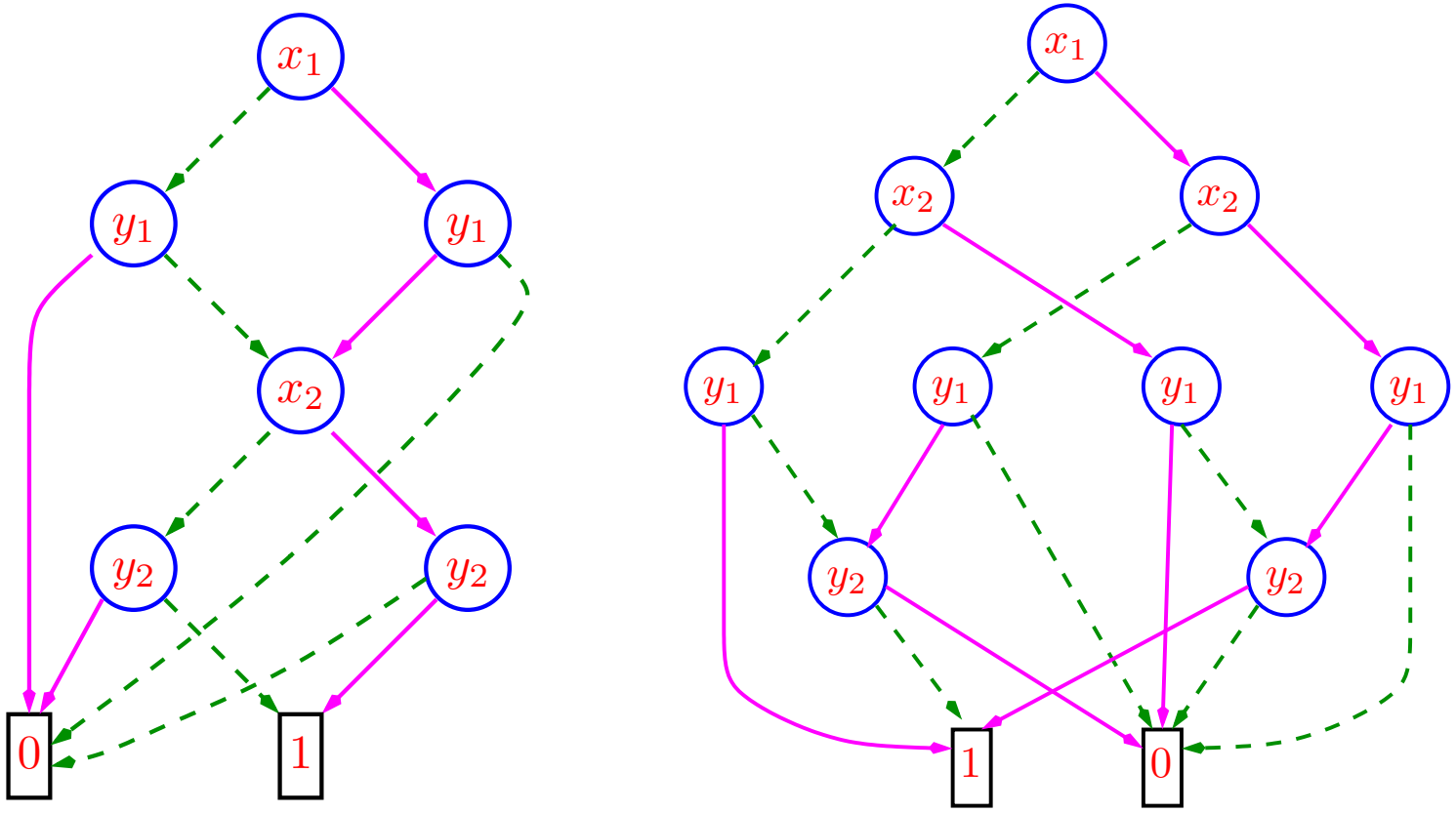
Алгоритм	Вход	Выход	Сложность
Reduce	B reduced BDD		$O(\|B\| \cdot \log \|B\|)$
Restrict	B	$B\|_{x \leftarrow 0}$ или $B\|_{x \leftarrow 1}$	$O(\|B\| \cdot \log \|B\|)$
Apply*	B_1, B_2	$B_1 \star B_2$	$O(\|B\| \cdot \|B_2\|)$
Exists*	B	$\exists x B$	$O(\|B\|^2)$

* с кэшированием промежуточных результатов

Заметим, что сложность зависит от размера исходной BDD...

А насколько компактны сами BDD?

Зависит от порядка



Существуют булевы формулы для которых наименьшая OBDD имеет экспоненциальный размер

Комментарий

- Здесь лекция может кончится. В таком случае, продолжение будет рассказано завтра.
- Если же останется время, мы продолжим.

Символьное представление систем переходов при помощи OBDD

Представим систему переходов (Q, T, q_0, L) при помощи OBDD
Пусть $n = \log Q$, $m = \log L$

- Q представим как $B_{f_Q} : \{0,1\}^n \to \{0,1\}$
- $T \subseteq Q \times Q$ представим как $B_{f_T} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$
- q_0 представим как вектор из n бит
- $L : Q \to 2^{Prop}$ представим как $B_M : Prop \to 2^Q$
  $B_M : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}$

Символьный динамический алгоритм

- Алгоритм работает по индукции по построению формулы начиная с наиболее вложенных подформул.
- Для каждой подформулы ψ, алгоритм строит OBDD B_ψ т.ч. $B_\psi(x_1, \ldots, x_n) = 1$ тогда и только тогда, когда битовый вектор $x_1, \ldots, x_n$ представляет элемент q такой что $(S, q) \models \psi$.
- Рассмотрим случаи в зависимости от формы ψ
  - $\psi = \textbf{false}$: $B_{\textbf{false}} = \boxed{0}$
  - $\psi = p$: Пусть $z_1, \ldots, z_m$ представляет p. $B_p = B_M | x_1 \leftarrow z_1, \ldots, x_m \leftarrow z_m$
  - $\psi = \psi_1 \wedge \psi_2$: $B_\psi = Apply(B_{\psi_1}, B_{\psi_2}, \wedge)$
  - $\psi = \neg\psi_1$: $B_{\neg\psi_1} = Not(B_{\psi_1})$
  - $\psi = \textbf{EX}\psi_1$: $B_{\textbf{EX}\psi} = \exists x'_1, \ldots, x'_n (B_{f_T}(x_1, \ldots, x_n, x'_1, \ldots, x'_n) \wedge B_{\psi_1}(x'_1, \ldots, x'_n))$

Символьный динамический алгоритм: $\mathbf{AF}\psi_1$

$\psi = \mathbf{AF}\psi_1$

$B_{\mathbf{AF}\psi_1} = B' = B_{\psi_1}$

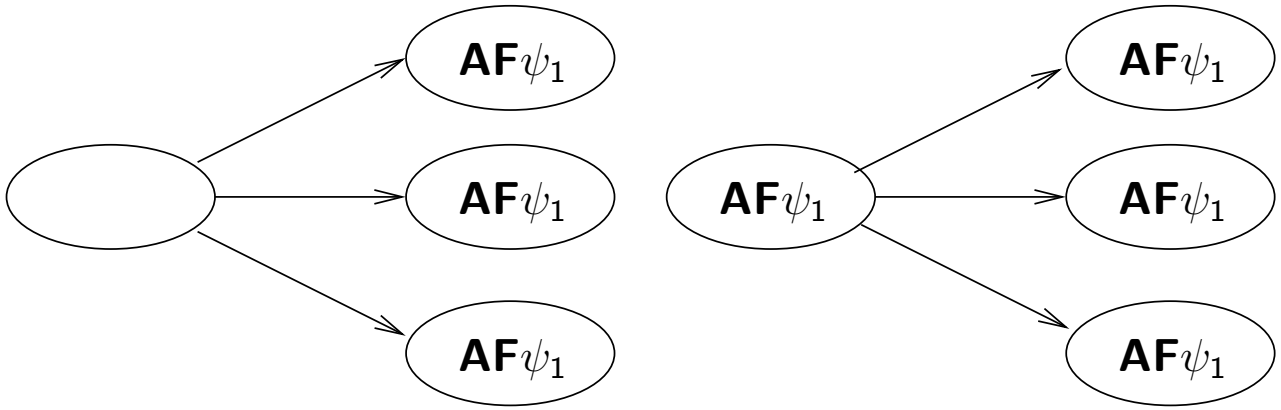
repeat

$B' = B_{\mathbf{AF}\psi_1}$

$B_{\mathbf{AF}\psi_1} = B' \vee$

$\forall x'_1, \ldots, x'_n (B_{f_T}(x_1, \ldots, x_n, x'_1, \ldots, x'_n) \rightarrow B'(x'_1, \ldots, x'_n))$

until $B_{\mathbf{AF}\psi_1} \neq B'$



Символьный динамический алгоритм: $\mathbf{E}\psi_1\mathbf{U}\psi_2$

$\psi = \mathbf{E}\psi_1\mathbf{U}\psi_2$

$B_{\mathbf{E}\psi_1\mathbf{U}\psi_2} = B' = B_{\psi_2}$

repeat

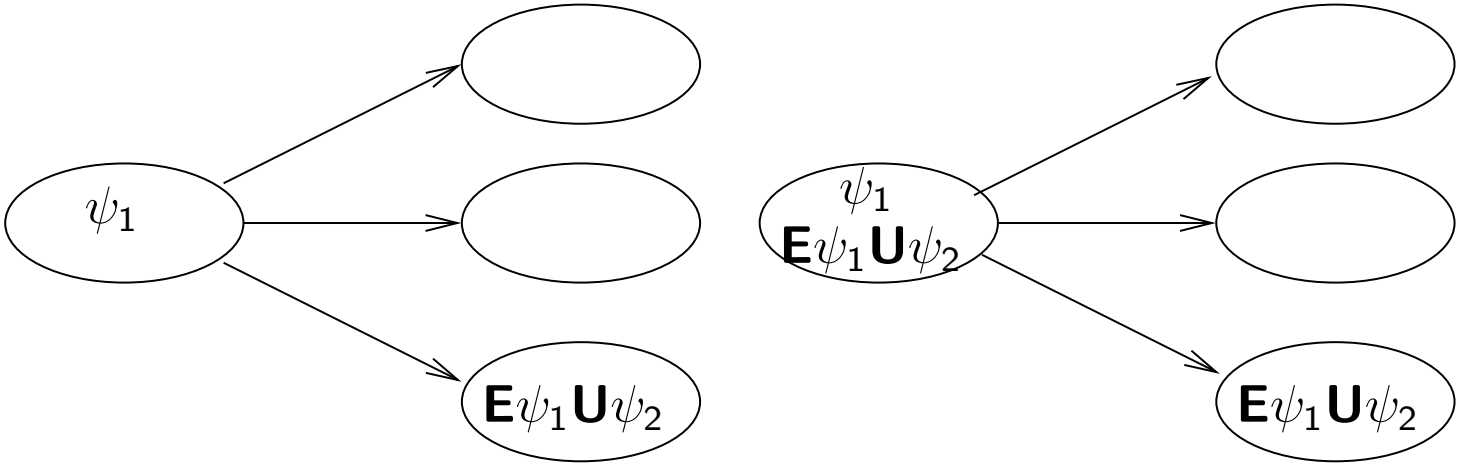
$B' = B_{\mathbf{E}\psi_1\mathbf{U}\psi_2}$

$B_{\mathbf{E}\psi_1\mathbf{U}\psi_2} = B' \vee$

$\exists x'_1, \ldots, x'_n (B_{f_T}(x_1, \ldots, x_n, x'_1, \ldots, x'_n) \wedge$

$B_{\psi_1}(x_1, \ldots, x_n) \wedge B'(x'_1, \ldots, x'_n))$

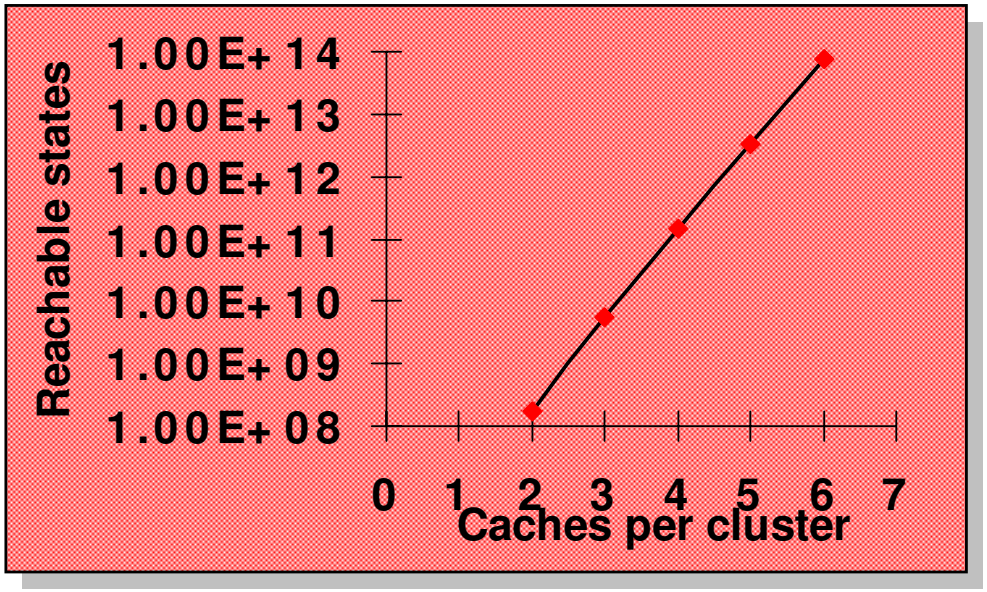
until $B_{\mathbf{AF}\psi_1} \neq B'$



На практике

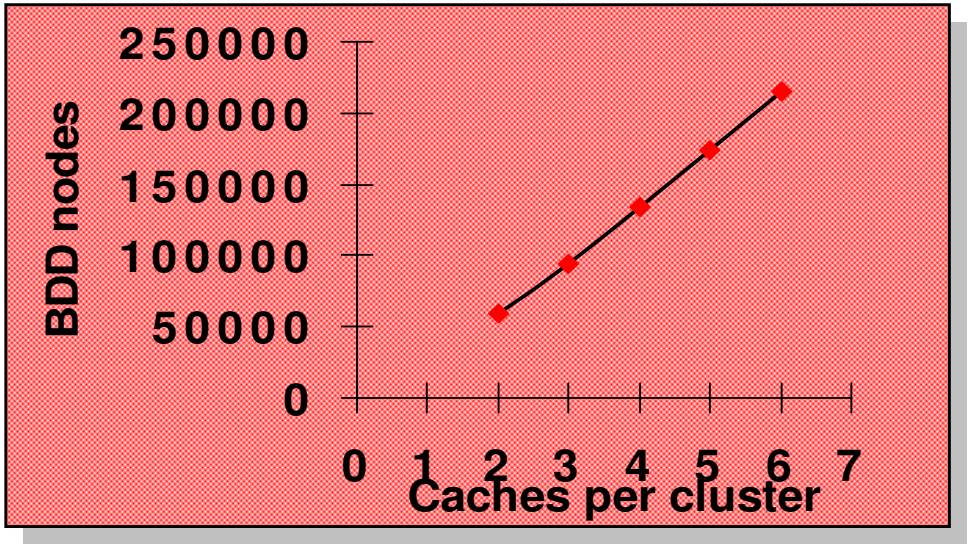
Проверка Encore Gigamax Cache System

- Мультипроцессор с распределенной памятью



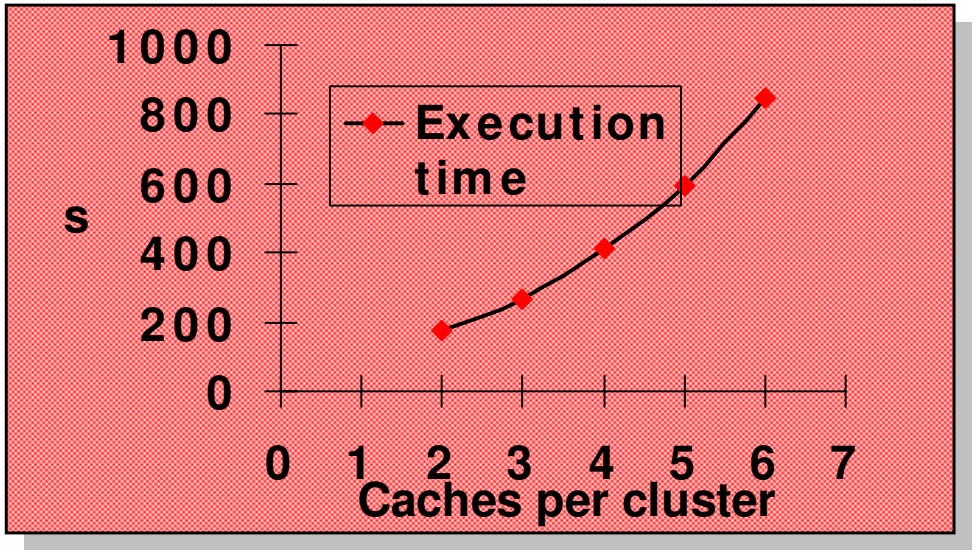
- Экспоненциальный рост числа состояний

Размер BDD



- Рост размера линеен

Производительность BDD



- Рост времени квадратичен