Квантовые ветвящиеся программы — новая парадигма модели квантовых алгоритмов

Аблаев Ф.М.

Казанский Федеральный Университет, Институт информатики АН РТ

"Перспективные компьютерные системы:
устройства, методы и концепции"
Таруса, 2-4 марта 2011

Физико-Математическая группа

Физическая группа

Математическая группа

Физико-Математическая группа

Физическая группа

- Моисеев С.
- Андрианов С.
- Губайдуллин Ф.
- Арсланов Н.

Математическая группа

Физико-Математическая группа

Физическая группа

- Моисеев С.
- Андрианов С.
- Губайдуллин Ф.
- Арсланов Н.

Математическая группа

- Аблаев Ф.
- Васильев А.
- Гайнутдинова А.
- Хасьянов А.

Физико-Математическая группа

Физическая группа

- Моисеев С.
- Андрианов С.
- Губайдуллин Ф.
- Арсланов Н.

Математическая группа

- Аблаев Ф.
- Васильев А.
- Гайнутдинова А.
- Хасьянов А.

Аспиранты

- Гараев Р.
- Хадиев К.
- Абрамский М.
- Зиатдинов М

Модели алгоритмов

- В тридцатых годах прошлого столетия было формализовано понятие алгоритма. Наиболее известна формализация Алана Тьюринга — МАШИНА ТЬЮРИНГА
  В настоящее время известны и используются различные формализации моделей алгоритмов, которые отличаются

Модели алгоритмов

- В тридцатых годах прошлого столетия было формализовано понятие алгоритма. Наиболее известна формализация Алана Тьюринга — МАШИНА ТЬЮРИНГА
  В настоящее время известны и используются различные формализации моделей алгоритмов, которые отличаются

- элементарными операциями

Модели алгоритмов

- В тридцатых годах прошлого столетия было формализовано понятие алгоритма. Наиболее известна формализация Алана Тьюринга — МАШИНА ТЬЮРИНГА
  В настоящее время известны и используются различные формализации моделей алгоритмов, которые отличаются
- элементарными операциями
- правилами их комбинаций друг с другом

Схемы из функциональных элементов

Схемы из функциональных элементов наиболее распространенная модель алгоритмов, ориентированных на реализации булевых функций.

$$f(x_1, \ldots, x_n)$$

$$f : \{0,1\}^n \to \{0,1\}$$

Классические схемы из функциональных элементов. Стандартный базис

One bit: $f(x) = \neg x$,

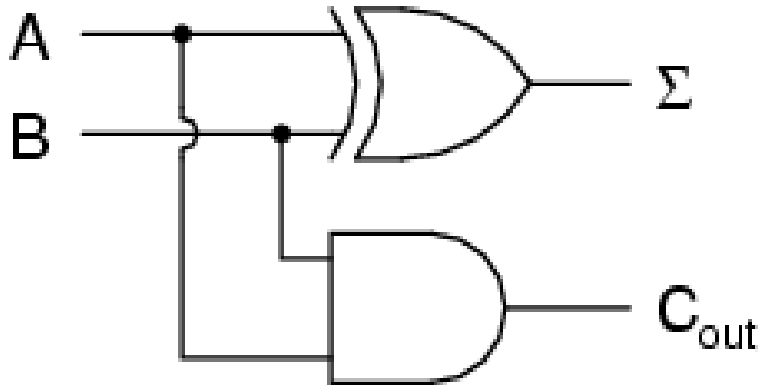
Two bit: $\bigvee(x, y) = x \vee y$,

x	y	$\bigvee(x, y)$
0	0	0
0	1	1
1	0	1
1	1	1

$\bigwedge(x, y) = x \wedge y$,

x	y	$\bigwedge(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1

Классические схемы из функциональных элементов. Стандартный базис



“Классические” Классы сложности

- P — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности (с полиномиальным числом базисных элементов).

“Классические” Классы сложности

- P — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности (с полиномиальным числом базисных элементов).
- NP — класс функций, вычислимых недетерминированными схемами из функциональных элементов полиномиальной сложности.

“Классические” Классы сложности

- P — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности (с полиномиальным числом базисных элементов).
- NP — класс функций, вычислимых недетерминированными схемами из функциональных элементов полиномиальной сложности.
- $$P \subseteq NP, \quad P \neq NP\,?$$

“Классические” Классы сложности

- P — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности (с полиномиальным числом базисных элементов).
- NP — класс функций, вычислимых недетерминированными схемами из функциональных элементов полиномиальной сложности.
- $$P \subseteq NP, \quad P \neq NP\,?$$
- NC^1 — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности логарифмической глубины

“Классические” Классы сложности

- P — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности (с полиномиальным числом базисных элементов).
- NP — класс функций, вычислимых недетерминированными схемами из функциональных элементов полиномиальной сложности.
- $$P \subseteq NP, \quad P \neq NP\,?$$
- NC^1 — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности логарифмической глубины
- $$NC^1 \subseteq P \subseteq NP, \quad NC^1 \neq NP\,?$$

Квантовые схемы из функциональных элементов. Стандартный базис

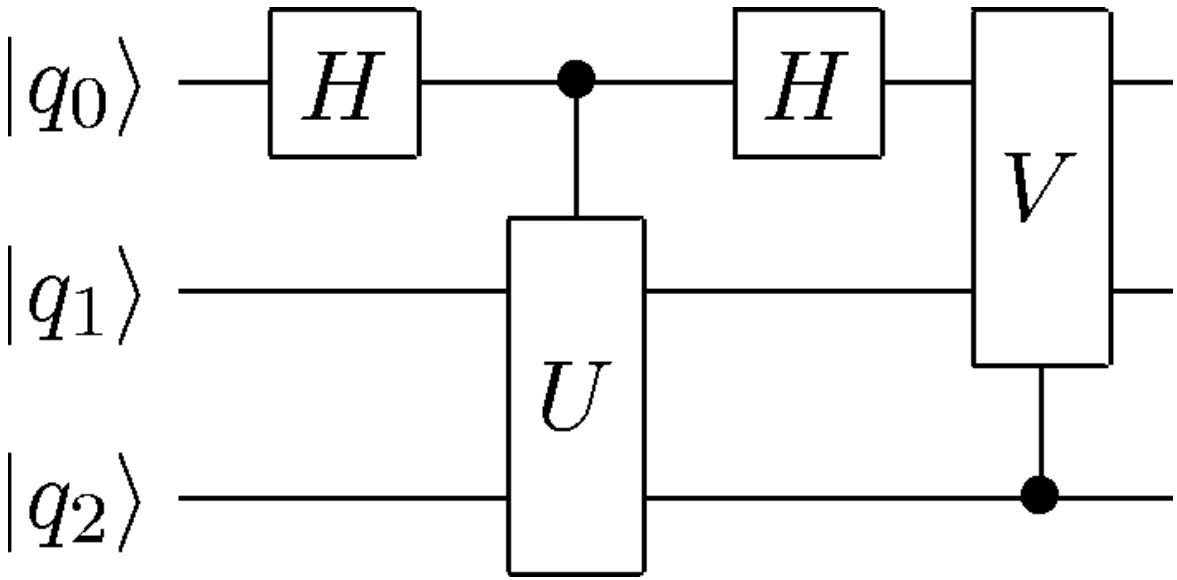
One qubit:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}; \; S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}; \; T = \pi/8 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix};$$

Two qubit:

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Квантовые схемы из функциональных элементов. Стандартный базис



“Квантовые” Классы сложности

- *BQP* — класс функций, вычислимых квантовыми схемами из функциональных элементов полиномиальной сложности (с полиномиальным числом базисных элементов).

“Квантовые” Классы сложности

- *BQP* — класс функций, вычислимых квантовыми схемами из функциональных элементов полиномиальной сложности (с полиномиальным числом базисных элементов).
- Факторизация числа — в классе *BQP* (P. Shor).

“Квантовые” Классы сложности

- BQP — класс функций, вычислимых квантовыми схемами из функциональных элементов полиномиальной сложности (с полиномиальным числом базисных элементов).
- Факторизация числа — в классе BQP (P. Shor).
- $P \subseteq BQP$, $NP \subseteq BQP$???

Детерминированные Ветвящиеся Программы

- "Программистский" подход к описанию алгоритмов (для вычисления булевых функций):

Детерминированные Ветвящиеся Программы

- “Программистский” подход к описанию алгоритмов (для вычисления булевых функций):
- Следующая операция является полной:

Детерминированные Ветвящиеся Программы

- "Программистский" подход к описанию алгоритмов (для вычисления булевых функций):
- Следующая операция является полной:
  - $\mathcal{O}$: if $x_i = 1$ then go to $\mathcal{O}1$ else go to $\mathcal{O}2$

Детерминированные Ветвящиеся Программы

- "Программистский" подход к описанию алгоритмов (для вычисления булевых функций):
- Следующая операция является полной:
  - $\mathcal{O}$: if $x_i = 1$ then go to $\mathcal{O}1$ else go to $\mathcal{O}2$
- Она является основой для модели Ветвящихся Программ.

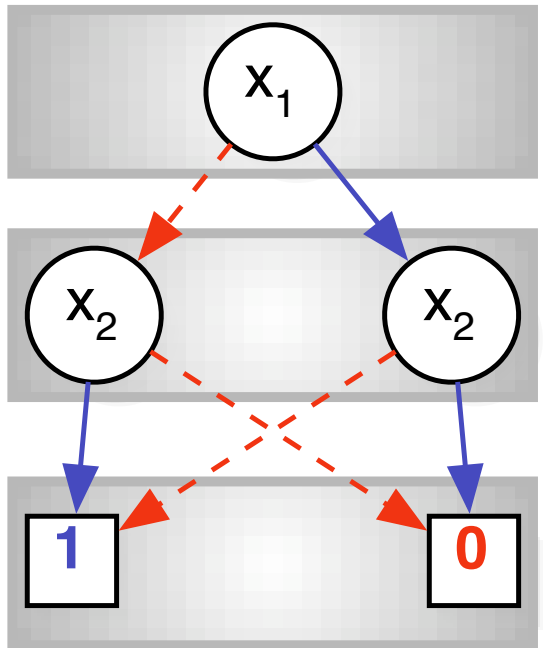
Детерминированные Ветвящиеся Программы

- "Программистский" подход к описанию алгоритмов (для вычисления булевых функций):
- Следующая операция является полной:
  - $\mathcal{O}$: if $x_i = 1$ then go to $\mathcal{O}1$ else go to $\mathcal{O}2$
- Она является основой для модели Ветвящихся Программ.

Для произвольной булевой функции можно построить программу, состоящую из операций такого типа.

Детерминированные Ветвящиеся Программы

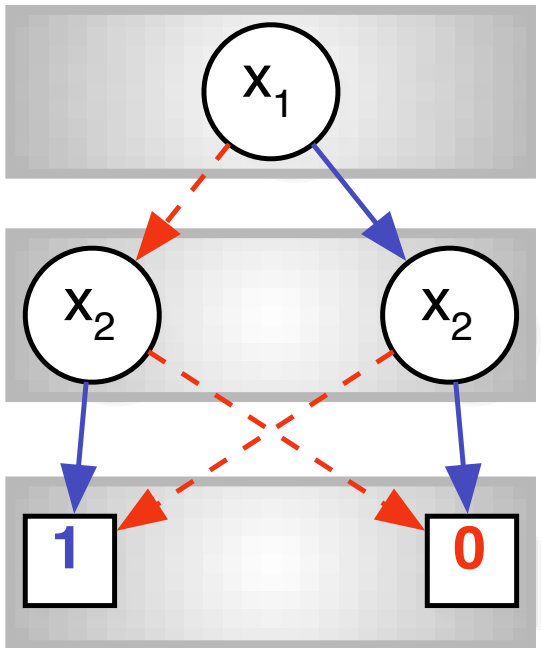
Ветвящаяся программа – это направленный ациклический граф:



Детерминированные Ветвящиеся Программы

Ветвящаяся программа – это направленный ациклический граф:

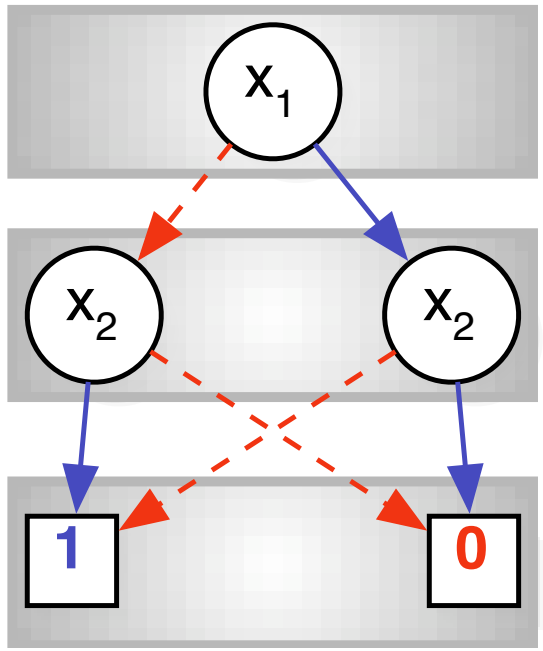
1. Каждая внутренняя вершина помечена переменной x_i, имеет два исходящих ребра, и соответствует операции:



Детерминированные Ветвящиеся Программы

Ветвящаяся программа – это направленный ациклический граф:

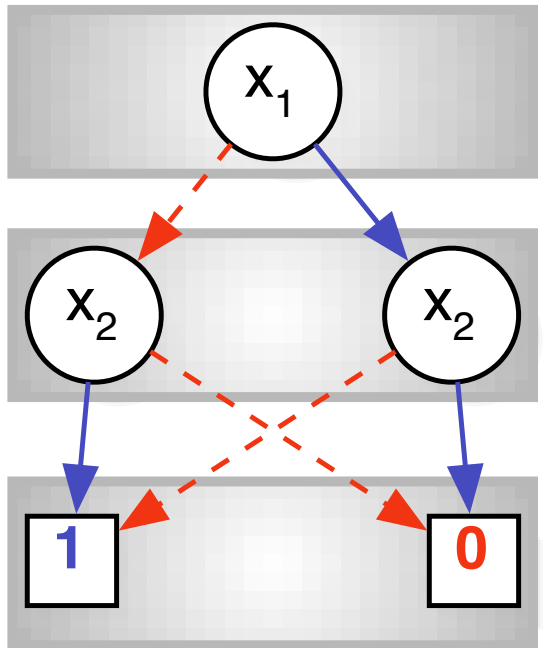
1. Каждая внутренняя вершина помечена переменной x_i, имеет два исходящих ребра, и соответствует операции:
   - $\mathcal{O}$: if $x_i = 1$ then go to $\mathcal{O}1$ else go to $\mathcal{O}2$



Детерминированные Ветвящиеся Программы

Ветвящаяся программа – это направленный ациклический граф:

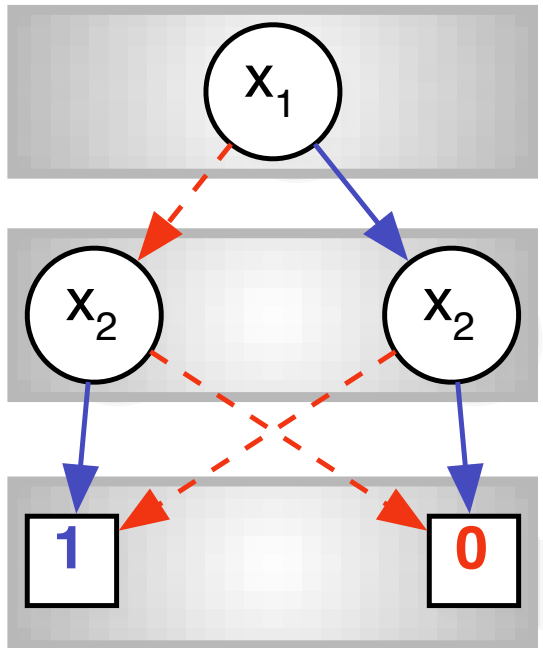
1. Каждая внутренняя вершина помечена переменной x_i, имеет два исходящих ребра, и соответствует операции:
   - $\mathcal{O}$: if $x_i = 1$ then go to $\mathcal{O}1$ else go to $\mathcal{O}2$
2. Выделена начальная вершина и подмножество *Accept* конечных вершин.



Детерминированные Ветвящиеся Программы

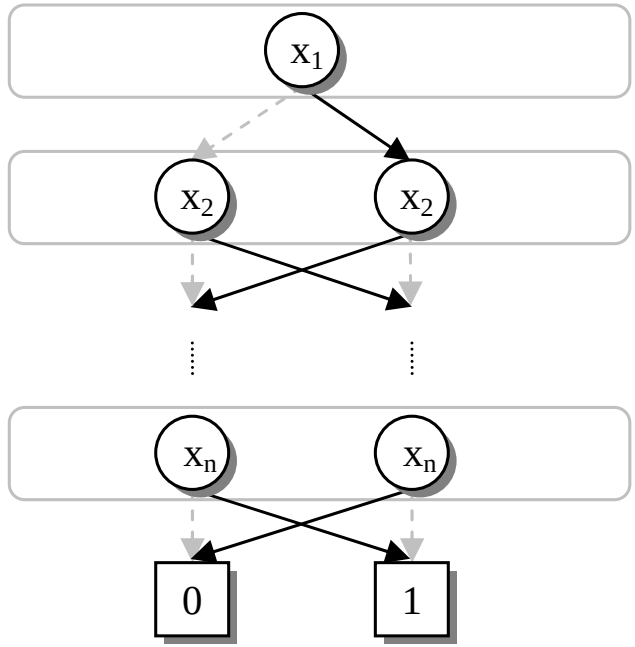
Ветвящаяся программа – это направленный ациклический граф:

1. Каждая внутренняя вершина помечена переменной x_i, имеет два исходящих ребра, и соответствует операции:
   - $\mathcal{O}$: if $x_i = 1$ then go to $\mathcal{O}1$ else go to $\mathcal{O}2$
2. Выделена начальная вершина и подмножество *Accept* конечных вершин.
3. Входной набор принимается $\Longleftrightarrow$ он порождает цепочку переходов, приводящих из начальной вершины в конечную из множества *Accept*.



Забывающие Ветвящиеся Программы

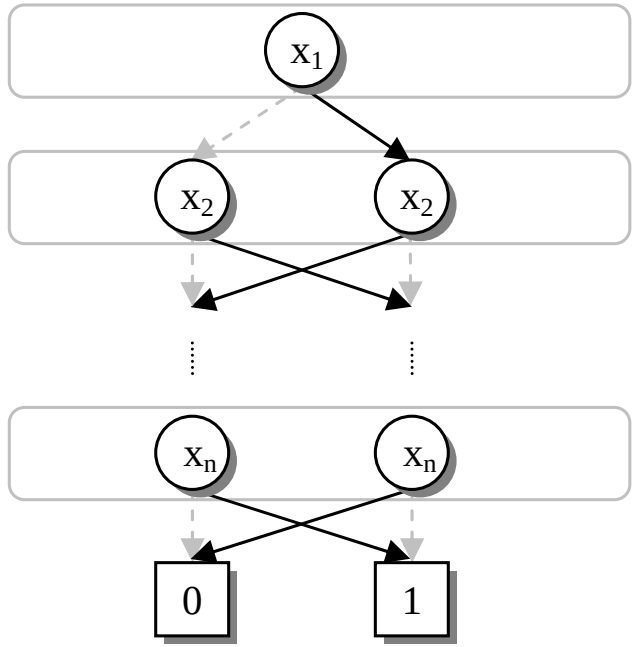
Ветвящаяся программа — забывающая, если:



Забывающие Ветвящиеся Программы

Ветвящаяся программа — забывающая, если:

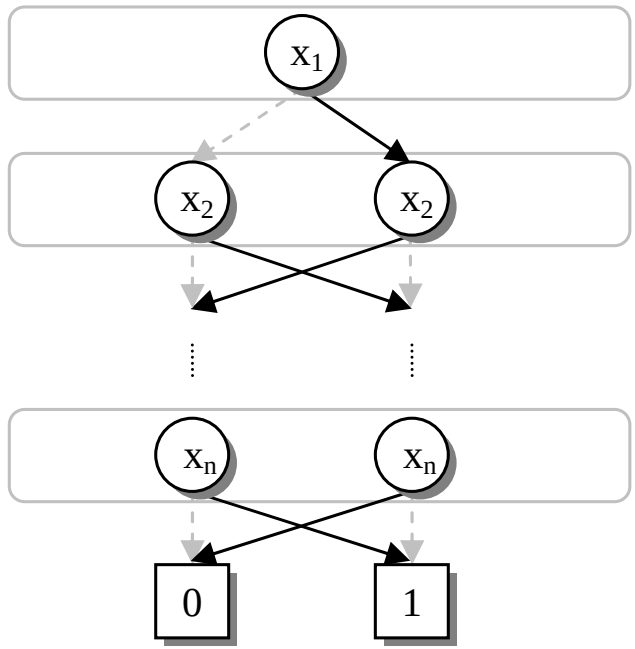
1. вершины могут быть разбиты на уровни $V_1, \ldots, V_\ell$ и уровень $V_{\ell+1}$ такие, что



Забывающие Ветвящиеся Программы

Ветвящаяся программа — забывающая, если:

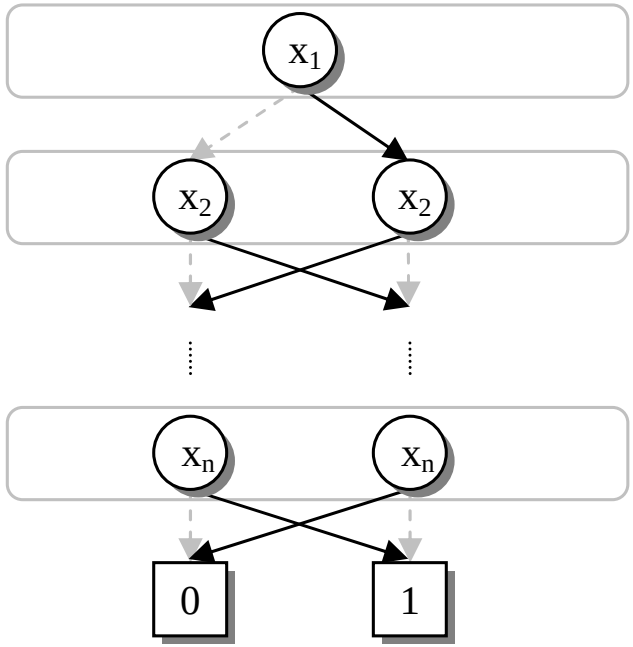
1. вершины могут быть разбиты на уровни $V_1, \ldots, V_\ell$ и уровень $V_{\ell+1}$ такие, что
2. вершины уровня $V_{\ell+1}$ являются конечными вершинами,



Забывающие Ветвящиеся Программы

Ветвящаяся программа — забывающая, если:

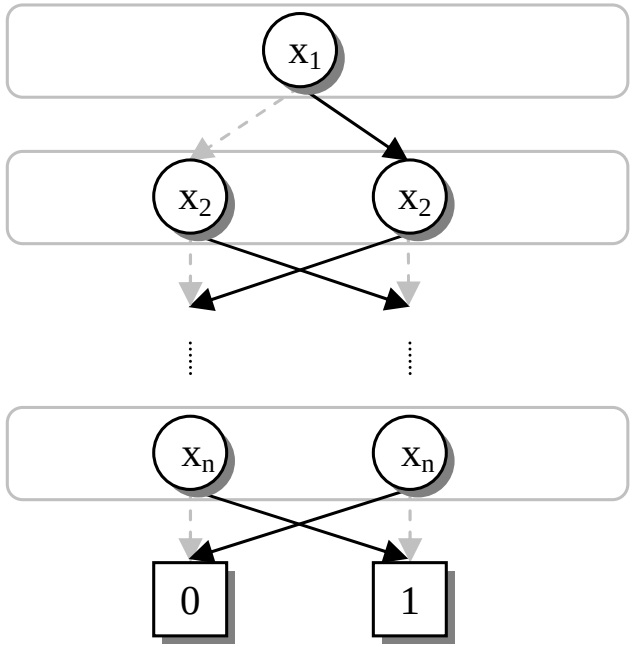
1. вершины могут быть разбиты на уровни $V_1, \ldots, V_\ell$ и уровень $V_{\ell+1}$ такие, что
2. вершины уровня $V_{\ell+1}$ являются конечными вершинами,
3. исходящие ребра вершин на уровне V_j при $j \leq \ell$ ведут только к вершинам на уровне V_{j+1},



Забывающие Ветвящиеся Программы

Ветвящаяся программа — забывающая, если:

1. вершины могут быть разбиты на уровни $V_1, \ldots, V_\ell$ и уровень $V_{\ell+1}$ такие, что
2. вершины уровня $V_{\ell+1}$ являются конечными вершинами,
3. исходящие ребра вершин на уровне V_j при $j \leq \ell$ ведут только к вершинам на уровне V_{j+1},
4. все вершины одного уровня V_j помечены одной и той же переменной x_{i_j}.



Меры сложности

Определение

Сложностью $Size(P)$ ветвящейся программы P называется число ее внутренних вершин (число команд).

Меры сложности

Определение

Сложностью $Size(P)$ ветвящейся программы P называется число ее внутренних вершин (число команд).

Определение

Длиной $Length(P)$ ветвящейся программы P называется длина максимального пути от начальной вершины к конечной (число шагов вычисления — время вычисления).

Меры сложности

Определение

Сложностью $Size(P)$ ветвящейся программы P называется число ее внутренних вершин (число команд).

Определение

Длиной $Length(P)$ ветвящейся программы P называется длина максимального пути от начальной вершины к конечной (число шагов вычисления — время вычисления).

Определение

Шириной $Width(P)$ ветвящейся программы P называется максимальное число вершин на уровне.

Схемы из функциональных элементов и ветвящиеся программы

- NC^1 — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности логарифмической глубины

Схемы из функциональных элементов и ветвящиеся программы

- NC^1 — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности логарифмической глубины
- $$NC^1 \subseteq P \subseteq NP, \quad NC^1 \neq NP\,?$$

Схемы из функциональных элементов и ветвящиеся программы

- NC^1 — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности логарифмической глубины
- $$NC^1 \subseteq P \subseteq NP, \quad NC^1 \neq NP\,?$$
- $P - BP_{const}$ – класс функций, вычислимых ветвящимися программами полиномиальной сложности и константной ширины.

Схемы из функциональных элементов и ветвящиеся программы

- NC^1 — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности логарифмической глубины
- $$NC^1 \subseteq P \subseteq NP, \quad NC^1 \neq NP\,?$$
- $P-BP_{const}$ – класс функций, вычислимых ветвящимися программами полиномиальной сложности и константной ширины.

1989

$$P-BP_{const}=NC^1.$$

Схемы из функциональных элементов и ветвящиеся программы

- NC^1 — класс функций, вычислимых схемами из функциональных элементов полиномиальной сложности логарифмической глубины
- $$NC^1 \subseteq P \subseteq NP, \quad NC^1 \neq NP\,?$$
- $P-BP_{const}$ – класс функций, вычислимых ветвящимися программами полиномиальной сложности и константной ширины.

1989

$$P-BP_{const}=NC^1.$$

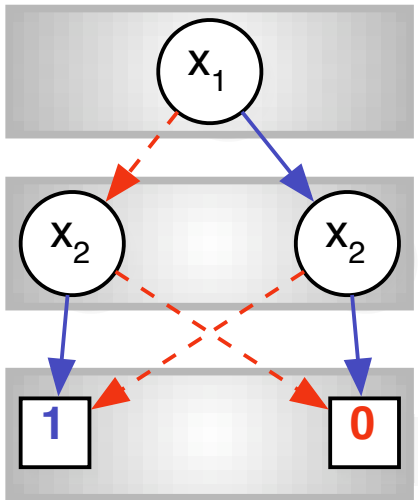
$$NC^1 \subseteq P \subseteq NP, \qquad P = NP?, \quad NC^1 = NP?$$

OBDD

Упорядоченные бинарные диаграммы решений

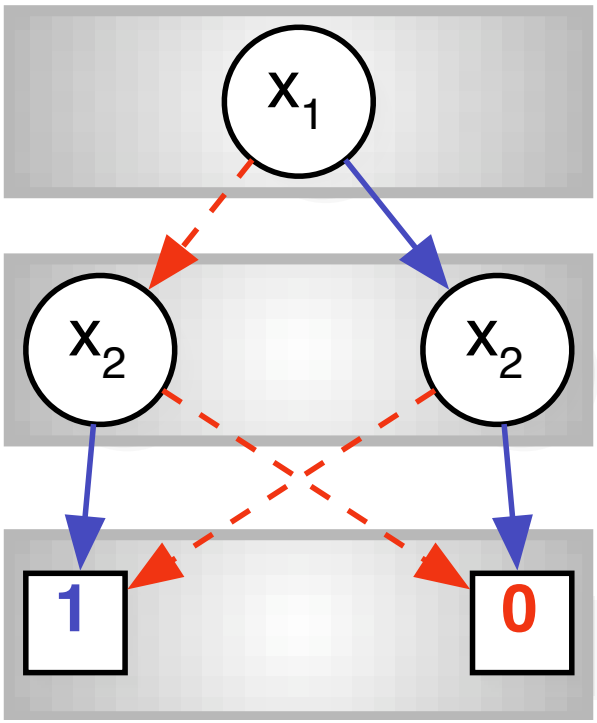
Определение OBDD

Ordered Binary Decision Diagram – это ветвящаяся программа, в которой на каждом пути от начальной вершины до конечной все переменные встречаются не более одного раза, причем согласно некоторому фиксированному порядку.



Пример OBDD

Ветвящаяся программа P



Уровень 1

Уровень 2

Уровень 3

$$Size(P) = 3$$
$$Length(P) = 2$$
$$Width(P) = 2$$

Квантовые ветвящиеся программы

- M.Nakanishi, K.Hamaguchi, T.Kashiwabara (2000)

Квантовые ветвящиеся программы

- M.Nakanishi, K.Hamaguchi, T.Kashiwabara (2000)
- Ф. Аблаев, А. Гайнутдинова, M.Karpinski (2001)

Квантовые ветвящиеся программы

- M.Nakanishi, K.Hamaguchi, T.Kashiwabara (2000)
- Ф. Аблаев, А. Гайнутдинова, M.Karpinski (2001)
- M.Sauerhoff, D.Sieling, T. Hofmeister (2003)

Квантовые ветвящиеся программы

- M.Nakanishi, K.Hamaguchi, T.Kashiwabara (2000)
- Ф. Аблаев, А. Гайнутдинова, M.Karpinski (2001)
- M.Sauerhoff, D.Sieling, T. Hofmeister (2003)
- А. Хасьянов (2004)

Квантовые ветвящиеся программы

- M.Nakanishi, K.Hamaguchi, T.Kashiwabara (2000)
- Ф. Аблаев, А. Гайнутдинова, M.Karpinski (2001)
- M.Sauerhoff, D.Sieling, T. Hofmeister (2003)
- А. Хасьянов (2004)
- C.Moore, C.Pollett (2003, 2005)

Квантовые ветвящиеся программы

- M.Nakanishi, K.Hamaguchi, T.Kashiwabara (2000)
- Ф. Аблаев, А. Гайнутдинова, M.Karpinski (2001)
- M.Sauerhoff, D.Sieling, T. Hofmeister (2003)
- А. Хасьянов (2004)
- C.Moore, C.Pollett (2003, 2005)
- А. Васильев (2009, 2010, 2011)

Квантовые ветвящиеся программы

Алгебраическое представление

Квантовая ветвящаяся программа Q над $\mathcal{H}^d$ – это тройка

$$Q = \left(|\psi_0\rangle, T, Accept \right) \quad :$$

Квантовые ветвящиеся программы

Алгебраическое представление

Квантовая ветвящаяся программа Q над $\mathcal{H}^d$ – это тройка

$$Q = \left(|\psi_0\rangle, T, Accept \right) \quad :$$

- $|\psi\rangle \in \mathcal{H}^d$ — состояния (нормированные вектора из $\mathcal{H}^d$), $|\psi_0\rangle \in \mathcal{H}^d$ — начальное состояние;

Квантовые ветвящиеся программы

Алгебраическое представление

Квантовая ветвящаяся программа Q над $\mathcal{H}^d$ – это тройка

$$Q = \left(|\psi_0\rangle, T, Accept \right) \quad :$$

- $|\psi\rangle \in \mathcal{H}^d$ — состояния (нормированные вектора из $\mathcal{H}^d$), $|\psi_0\rangle \in \mathcal{H}^d$ — начальное состояние;
- $T = (T_1, \ldots, T_\ell)$ есть последовательность инструкций:

Квантовые ветвящиеся программы

Алгебраическое представление

Квантовая ветвящаяся программа Q над $\mathcal{H}^d$ – это тройка

$$Q = \Big(|\psi_0\rangle , T, Accept \Big) \quad :$$

- $|\psi\rangle \in \mathcal{H}^d$ — состояния (нормированные вектора из $\mathcal{H}^d$), $|\psi_0\rangle \in \mathcal{H}^d$ — начальное состояние;
- $T = (T_1, \ldots, T_\ell)$ есть последовательность инструкций:
- $T_j = \langle x_{i_j}, U_j(0), U_j(1) \rangle$ определяется переменной x_{i_j}, $U_j(0)$ и $U_j(1)$ – унитарные преобразования в $\mathcal{H}^d$;

Квантовые ветвящиеся программы

Алгебраическое представление

Квантовая ветвящаяся программа Q над $\mathcal{H}^d$ – это тройка

$$Q = \left(|\psi_0\rangle, T, Accept \right) \quad :$$

- $|\psi\rangle \in \mathcal{H}^d$ — состояния (нормированные вектора из $\mathcal{H}^d$), $|\psi_0\rangle \in \mathcal{H}^d$ — начальное состояние;
- $T = (T_1, \ldots, T_\ell)$ есть последовательность инструкций:
- $T_j = \langle x_{i_j}, U_j(0), U_j(1) \rangle$ определяется переменной x_{i_j}, $U_j(0)$ и $U_j(1)$ – унитарные преобразования в $\mathcal{H}^d$;
- j-ый шаг: тестируется переменная x_{i_j} и выполняется переход в состояние $|\psi'\rangle = U_j(\sigma_{i_j}) |\psi\rangle$;

Квантовые ветвящиеся программы

Алгебраическое представление

Квантовая ветвящаяся программа Q над $\mathcal{H}^d$ – это тройка

$$Q = \Big(|\psi_0\rangle, T, \mathit{Accept} \Big) \quad :$$

- $|\psi\rangle \in \mathcal{H}^d$ — состояния (нормированные вектора из $\mathcal{H}^d$), $|\psi_0\rangle \in \mathcal{H}^d$ — начальное состояние;
- $T = (T_1, \ldots, T_\ell)$ есть последовательность инструкций:
- $T_j = \langle x_{i_j}, U_j(0), U_j(1) \rangle$ определяется переменной x_{i_j}, $U_j(0)$ и $U_j(1)$ – унитарные преобразования в $\mathcal{H}^d$;
- j-ый шаг: тестируется переменная x_{i_j} и выполняется переход в состояние $|\psi'\rangle = U_j(\sigma_{i_j})\,|\psi\rangle$;
- $\mathit{Accept} \subset \{1, \ldots, d\}$ задает принимающее множество

$$|\psi_0\rangle \xrightarrow{U_1(\sigma_{i_1})} \ldots |\psi\rangle \xrightarrow{U_j(\sigma_{i_j})} |\psi'\rangle \ldots \xrightarrow{U_\ell(\sigma_{i_\ell})} |\psi(\sigma)\rangle \longrightarrow \begin{cases} \mathit{Accept} \\ \\ \mathit{Reject} \end{cases}$$

Вычисление с ограниченной ошибкой

Пусть $|\psi(\sigma)\rangle = (\alpha_1, \ldots, \alpha_d)$ – конечное состояние вычислений на наборе σ.

- $Pr_{accept}(\sigma) = \sum_{i \in Accept} |\alpha_i|^2$
- $Pr_{reject}(\sigma) = \sum_{i \notin Accept} |\alpha_i|^2$

Вычисление с ограниченной ошибкой

Вычисление с ограниченной ошибкой

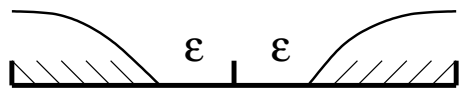
Пусть $|\psi(\sigma)\rangle = (\alpha_1, \ldots, \alpha_d)$ – конечное состояние вычислений на наборе σ.

- $Pr_{accept}(\sigma) = \sum_{i \in Accept} |\alpha_i|^2$
- $Pr_{reject}(\sigma) = \sum_{i \notin Accept} |\alpha_i|^2$

Вычисление с ограниченной ошибкой

- Квантовая ветвящаяся программа Q вычисляет булеву функцию f с ограниченной ошибкой, если существует $\epsilon \in (0, 1/2)$ такое, что для любого входного набора σ вероятность ошибки не превосходит $1/2 - \epsilon$, т.е.

$$Pr[Q(\sigma) \neq f(\sigma)] \leq \frac{1}{2} - \epsilon.$$



Сравнительные сложностные характеристики классических и квантовых ветвящихся программ

- Ширина $width(Q)$ квантовой ветвящейся программы Q определяется как размерность пространства состояний $\mathcal{H}^d$.

Сравнительные сложностные характеристики классических и квантовых ветвящихся программ

- Ширина $width(Q)$ квантовой ветвящейся программы Q определяется как размерность пространства состояний $\mathcal{H}^d$.

Теорема (Аблаев, Гайнутдинова, Карпински)

Пусть P - минимальная детерминированная OBDD, вычисляющая функцию f. Тогда для квантовой OBDD Q, вычисляющей ту же функцию с ограниченной ошибкой ϵ, справедливо

$$width(Q) \geq c(\epsilon)(\log width(P))$$

- А. Ф. Гайнутдинова (2000-2004) Алгебраическое представление квантовых ветвящихся программ.
  Функция MOD_m эффективно вычислима квантовой OBDD

	OBDD	QOBDD
MOD_m	$\Omega(m)$	$O(\log m)$

- А. Ф. Гайнутдинова (2000-2004) Алгебраическое представление квантовых ветвящихся программ.
  Функция MOD_m эффективно вычислима квантовой OBDD

	OBDD	QOBDD
MOD_m	$\Omega(m)$	$O(\log m)$

- А.Ф. Хасьянов (2002-2005) Алгебраическое представление квантовых ветвящихся программ. Семейство функций эффективно вычислимые квантовыми OBDD

- А. Ф. Гайнутдинова (2000-2004) Алгебраическое представление квантовых ветвящихся программ.
  Функция MOD_m эффективно вычислима квантовой OBDD

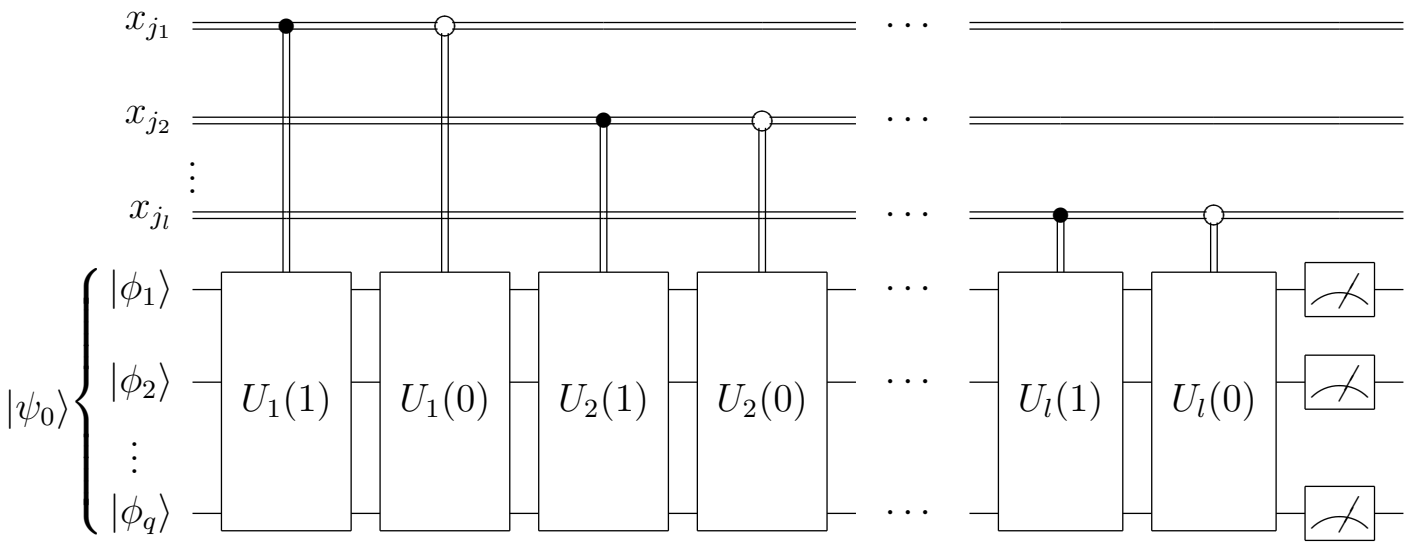
	OBDD	QOBDD
MOD_m	$\Omega(m)$	$O(\log m)$

- А.Ф. Хасьянов (2002-2005) Алгебраическое представление квантовых ветвящихся программ. Семейство функций эффективно вычислимые квантовыми OBDD
- А. В. Васильев (2005-2008) Схемное представление, метод отпечатков. Функции, представимые линейными полиномами — эффективно вычислимы в квантовых OBDD.

Квантовые ветвящиеся программы

Схемное представление

Квантовая ВП — квантовая схема из функциональных элементов, дополненная возможностью считывать классические биты в качестве контролирующих для унитарных операторов.



QOBDD – переменные читаются ровно 1 раз.

Сложность квантовых ВП

- Для квантовой ветвящейся программы в схемном представлении явным образом проявляется еще одна мера сложности – число кубит q.

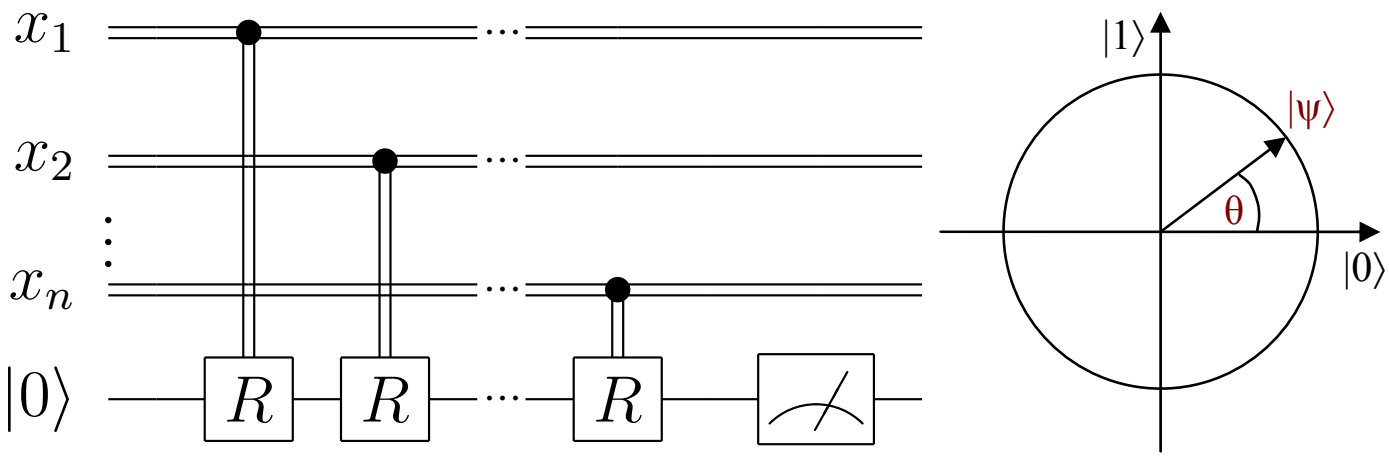
Сложность квантовых ВП

- Для квантовой ветвящейся программы в схемном представлении явным образом проявляется еще одна мера сложности – число кубит q.
- Для реализации квантовой ветвящейся программы ширины d потребуется $\log d$ кубит

Квантовые ветвящиеся программы

Пример

$$MOD_m(x_1, \ldots, x_n) = 1 \iff \sum_{i=1}^{n} x_i = 0 \bmod m, \text{ где } m \geq 2.$$



R – оператор поворота на угол $\theta = \pi/m$

Результаты последних исследований

Разработан метод построения оптимальных по числу кубит квантовых OBDD для индивидуальных функций на основе:

- представления булевых функций характеристическими полиномами
- метода отпечатков (fingerprinting)

	OBDD	QOBDD
MOD_m	$\Omega(\log m)$	$O(\log \log m)$
MOD'_m	$\Omega(\log m)$	$O(\log \log m)$
EQ_n	$\Omega(n)$	$O(\log n)$
$Palindrome_n$	$\Omega(n)$	$O(\log n)$
$PERM_n$	$\Omega(n)$	$O(\log n)$

Представление булевых функций характеристическими полиномами

$f : \{0,1\}^n \to \{0,1\}$ – булева функция от n переменных.

Представление булевых функций характеристическими полиномами

$f : \{0,1\}^n \to \{0,1\}$ – булева функция от n переменных.

Полином $g_f(x_1, \ldots, x_n)$ над $\mathbb{Z}_m$ называется характеристическим полиномом для функции f, если:

- $g_f(\sigma) = 0$ для всех $\sigma \in f^{-1}(1)$
- $g_f(\sigma) \neq 0$ для всех $\sigma \in f^{-1}(0)$

Функции с линейными полиномами

Теорема

Если g является линейным характеристическим полиномом над $\mathbb{Z}_m$ для функции f,то для любого $\epsilon \in (0,1)$ существует $O(\log\log m)$-кубитная квантовая OBDD, вычисляющая функцию f с односторонней ошибкой ϵ.

Примеры

MOD_m	$\sum_{i=1}^{n} x_i$	$\mathbb{Z}_m$	$O(\log\log m)$
MOD'_m	$\sum_{i=1}^{n} x_i 2^{i-1}$	$\mathbb{Z}_m$	$O(\log\log m)$
EQ_n	$\sum_{i=1}^{n} x_i 2^{i-1} - \sum_{i=1}^{n} y_i 2^{i-1}$	$\mathbb{Z}_{2^n}$	$O(\log n)$
$Palindrome_n$	$\sum_{i=1}^{\lfloor n/2 \rfloor} x_i 2^{i-1} - \sum_{i=\lceil n/2 \rceil}^{n} x_i 2^{n-i}$	$\mathbb{Z}_{2^{\lfloor n/2 \rfloor}}$	$O(\log n)$
$PERM_n$	$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}\left((n+1)^{i-1} + (n+1)^{n+j-1}\right) - \sum_{i=1}^{2n}(n+1)^{i-1}$	$\mathbb{Z}_{(n+1)^{2n}}$	$O(\log n)$

Классы функций, вычислимые ветвящимися программами ширины 2 и полиномиальной длины:

DBP_2 — детерминированными перестановочными,
$SyntSBP_2$— дважды стохастическими
$SyntQBP_2$ — квантовыми.

Классы функций, вычислимые ветвящимися программами ширины 2 и полиномиальной длины:

DBP_2 — детерминированными перестановочными,
$SyntSBP_2$— дважды стохастическими
$SyntQBP_2$ — квантовыми.

- $DBP_2 \subsetneq SyntSBP_2 \subsetneq SyntQBP_2 = NC^1 \subseteq QBP_2$
  (F. Ablayev, C. Moore, C. Pollett, 2005)

Классы функций, вычислимые ветвящимися программами ширины 2 и полиномиальной длины:

DBP_2 — детерминированными перестановочными,
$SyntSBP_2$— дважды стохастическими
$SyntQBP_2$ — квантовыми.

- $DBP_2 \subsetneq SyntSBP_2 \subsetneq SyntQBP_2 = NC^1 \subseteq QBP_2$
  (F. Ablayev, C. Moore, C. Pollett, 2005)
- Открытая проблема: $SyntQBP_2 = QBP_2$?

Классы функций, вычислимые ветвящимися программами ширины 2 и полиномиальной длины:

DBP_2 — детерминированными перестановочными,
$SyntSBP_2$— дважды стохастическими
$SyntQBP_2$ — квантовыми.

- $DBP_2 \subsetneq SyntSBP_2 \subsetneq SyntQBP_2 = NC^1 \subseteq QBP_2$
  (F. Ablayev, C. Moore, C. Pollett, 2005)
- Открытая проблема: $SyntQBP_2 = QBP_2$?
- (Barrington 1985) $NC^1 = DBP_5 = DBP_{const}$.

Классы функций, вычислимые ветвящимися программами ширины 2 и полиномиальной длины:

DBP_2 — детерминированными перестановочными,
$SyntSBP_2$— дважды стохастическими
$SyntQBP_2$ — квантовыми.

- $DBP_2 \subsetneq SyntSBP_2 \subsetneq SyntQBP_2 = NC^1 \subseteq QBP_2$
  (F. Ablayev, C. Moore, C. Pollett, 2005)
- Открытая проблема: $SyntQBP_2 = QBP_2$?
- (Barrington 1985) $NC^1 = DBP_5 = DBP_{const}$.
- А.Васильев (2007-2008) показал, что функции из $SyntSBP_2$ представимы квантовыми $poly - OBDD_2$