УЧЕНЫЕ ЗАПИСКИ КАЗАНСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
Том 151, кн. 2 Физико-математические науки *2009*

УДК 519.71

КВАНТОВОЕ И КЛАССИЧЕСКОЕ МОДЕЛИРОВАНИЕ ВЕТВЯЩИХСЯ ПРОГРАММ

А.Ф. Гайнутдинова

Аннотация

В статье рассматривается модель для вычисления булевых функций – ветвящиеся программы (ВР – branching programs). Изучаются классические вероятностные ВР и две модели квантовых ВР – один раз измеряющие и много раз измеряющие ВР, соответственно использующие единственное измерение в конце вычислений и использующие измерения после каждого вычислительного шага.

В статье представлены три различных метода моделирования ВР: метод вероятностного моделирования квантовых ВР, и два различных метода квантового моделирования вероятностных ВР. Доказывается сложность методов, приводится их сравнительный анализ. Как следствие доказанных теорем приводятся соотношения классов сложности, определяемых для модели ветвящихся программ.

Ключевые слова: ветвящиеся программы, сложность вычислений, квантовое и классическое моделирование.

Введение

Идея о возможном использовании эффектов квантовой механики в вычислениях была высказана в 80-х годах двадцатого столетия (Ю.И. Манин [1], Р. Фейнман [2]). С тех пор область квантовых вычислений активно развивается. Были определены и активно исследуются квантовые аналоги классических вычислительных моделей, таких, как автоматы, схемы, ветвящиеся программы и т. д. По аналогии с классическими классами сложности, на основе квантовых вычислительных моделей были определены сложностные классы и показаны основные соотношения между ними. Наиболее актуальной исследовательской задачей является поиск проблем, сложных для классических вычислений (или для которых неизвестны эффективные классические алгоритмы), но которые эффективно решаются в квантовых моделях. К сожалению, на данный момент таких примеров найдено немного, одними из самых известных являются полиномиальный алгоритм Шора факторизации чисел и алгоритм Гровера поиска в неупорядоченной базе данных. Не менее важной задачей является исследование сложностных соотношений между классическими вычислительными моделями и их квантовыми аналогами. В данной работе рассматривается известная модель для вычисления булевых функций – ветвящиеся программы. Известно, что логарифм сложности ветвящейся программы соответствует объему памяти машины Тьюринга, а максимальная длина вычислительного пути – времени вычисления. Квантовый аналог классической ВР был впервые определен в работе [3]. Данная модель определялась как последовательность унитарных эволюций квантовой системы с заключительным измерением как процедурой извлечения результата вычислений. В настоящей работе мы будем называть эту модель один раз измеряющей квантовой ВР. Известная модель перестановочных бинарных программ, рассматриваемая в работе [4],

является частным случаем такой модели. Напомним, что класс функций, вычислимых перестановочными ВР полиномиальной сложности, в точности совпадает с классом NC_1 функций, представимых схемами из функциональных элементов логарифмической глубины полиномиальной сложности [4]. Несколько иные модели квантовой ВР были определены в работах [5, 6].

В работе [7] был приведен пример функции, для которой квантовые ветвящиеся программы могут быть экспоненциально экономнее как детерминированных, так и вероятностных «стабильных» ветвящихся программ. Под «стабильными» понимаются ВР, у которых преобразования, соответствующие одному и тому же значению входной переменной, не зависят от номера уровня, на котором они применяются.

В работе [8] была определа модель много раз измеряющей квантовой ВР. В этой модели каждый шаг вычислений состоит из унитарного преобразования и последующего промежуточного измерения. По окончании вычислений производится финальное измерение как процедура извлечения результата вычислений. В работе [8] были представлены два различных метода моделирования: метод классического моделирования квантовых один раз измеряющих ветвящихся программ и метод квантового моделирования классических вероятностных ветвящихся программ с использованием многократного измерения. В настоящей работе мы предлагаем новый метод квантового моделирования классических вероятностных ветвящихся программ с использованием однократного измерения. В работе проводится сравнительный анализ данных методов моделирования.

1. Определения

Детерминированная ветвящаяся программа (ВР – Branching Program) над множеством переменных $X = \{x_1, \ldots, x_n\}$ – это ориентированный ациклический граф с конечными вершинами, помеченными нулем и единицей (будем называть их отвергающими и принимающими вершинами, соответственно). Каждая внутренняя вершина помечена булевой переменной $x \in X$, и имеет два исходящих ребра, помеченных нулем и единицей соответственно. ВР представляет булеву функцию $f : \{0,1\}^n \to \{0,1\}$ следующим образом. Вычисление значения $f(\sigma)$ для входного набора $\sigma \in \{0,1\}^n$ стартует из выделенной начальной вершины. Для каждой внутренней вершины, помеченной переменной x_j, осуществляется переход из этой вершины либо по 0-ребру, либо по 1-ребру в соответствии со значением σ_j, которое принимает переменная x_j во входном наборе. Значение функции f для входа σ – это значение достигнутой конечной вершины.

Сложность $\mathrm{Size}(P)$ ветвящейся программы P – это количество ее внутренних вершин.

Длина $\mathrm{Length}(P)$ ветвящейся программы P – это длина ее самого длинного пути из начальной вершины в конечную.

Длина ВР очевидным образом оценивает время, требуемое для вычисления функции f в худшем случае. Сложность ВР оценивает память, затрачиваемую в процессе вычисления.

Ветвящаяся программа называется *уровневой*, если ее вершины могут быть разбиты на уровни $0, 1, \ldots$ таким образом, что для каждого i рёбра из вершин уровня i ведут только в вершины уровня $(i+1)$.

Известно, что каждая ВР P может быть преобразована в уровневую ВР P', вычисляющую ту же самую функцию. При этом длина ВР не изменится, а сложность возрастет не более чем в квадрат [9].

Ширина $\mathrm{Width}(P)$ уровневой ВР P – это максимум числа вершин на уровне, взятый по всем уровням программы P.

Уровневая ВР P называется *забывающей*, если во всех вершинах одного уровня P считывается одна и та же переменная.

Известно [4], что уровневая ВР может быть преобразована в забывающую путем полиномиального увеличения длины и удвоения ширины.

OBDD (Ordered Binary Decision Diagram) – это ветвящаяся программа с ограничением, что на каждом пути из начальной вершины в конечную все переменные считываются не более одного раза в одном и том же порядке.

Вероятностная ветвящаяся программа (PBP – Probabilistic Branching Program) была впервые определена в работе Ф.М. Аблаева, М. Карпинского [10] как естественное обобщение детерминированной ВР. Вероятностная ветвящаяся программа – это программа, в которой каждая внутренняя вершина имеет выходную степень, не меньшую 2. При этом из каждой внутренней вершины выходит два типа ребер – помеченные 0 и 1. Каждому ребру e приписана вероятность $p(e)$, $p(e)$– рациональное число, $(0 \leq p(e) \leq 1)$, таким образом, что для каждой вершины сумма вероятностей всех ребер, исходящих из этой вершины, помеченных нулем (единицей), равна 1. Вычисление на входном наборе $\sigma \in \{0,1\}^n$ осуществляется следующим образом. На каждом шаге, начиная с выделенной начальной вершины, PBP P считывает значение переменной, приписанной вершине, и в зависимости от значения считанной переменной переходит в следующие вершины либо по 0-ребрам, либо по 1-ребрам с вероятностями, приписанными соответствующим ребрам. Вероятность $Pr^P_{\text{acc}}(\sigma)$ принятия PBP P входа σ – это вероятность того, что вычисление на входе σ приведет в конечную вершину, помеченную единицей.

Если ВР P – уровневая ВР, то без ограничения общности можем считать, что каждый уровень ВР содержит одинаковое число вершин, перенумерованных как $1, \ldots, d$ (будем называть их состояниями). Тогда в процессе вычисления макросостояние ВР может быть описано при помощи вектора ψ распределения вероятностей состояний на уровне, и шаг работы ВР состоит в преобразовании вектора ψ.

1.1. Квантовая система. Пусть $\mathcal{H}^d$ – d-мерное гильбертово пространство с нормой $||.||_2$. Пусть QS – квантовая физическая система с d устойчивыми состояниями $\{1, \ldots, d\}$. *Чистое состояние* QS описывается вектором $|\psi\rangle$ единичной длины в гильбертовом пространстве $\mathcal{H}^d$ в базисе $\{|1\rangle, \ldots, |d\rangle\}$ (будем называть его стандартным базисом), где $|i\rangle$ – вектор-столбец, i-я компонента которого равна единице, остальные равны нулю, то есть

$$|\psi\rangle = \sum_{i=1}^{d} z_i |i\rangle,$$

или кратко $|\psi\rangle = (z_1, \ldots, z_d)$.

Всюду далее будем называть $|\psi\rangle$ *конфигурацией*. Комплексное число z_i называется амплитудой базисного состояния $|i\rangle$ в конфигурации $|\psi\rangle$.

Эволюция квантовой системы. *Эволюция квантовой системы* (изменение состояния QS за определенный промежуток времени) задается унитарным оператором и описывается следующим образом. Если $|\psi\rangle$ – конфигурация системы QS на текущем шаге, то на следующем шаге конфигурация QS будет $|\psi'\rangle$, где $|\psi'\rangle = U|\psi\rangle$ и U – $(d \times d)$-унитарная матрица.

Измерение квантовой системы. Извлечение информации о QS из конфигурации $|\psi\rangle$ называется *измерением* и задается оператором проекции.

В работе мы будем использовать два типа измерения: *промежуточное* и *финальное*. Промежуточное измерение производится на некотором этапе вычислительного процесса, а финальное измерение – по окончании вычисления. Будем задавать их следующим образом (см., например, книгу [11, глава 1.6]).

Промежуточное измерение. Пусть $|\psi\rangle$ – текущая конфигурация и пусть $\mathcal{H}^d = W_1 \oplus \cdots \oplus W_k$ – ортогональная декомпозиция пространства $\mathcal{H}^d$. Измерение $\mathcal{O} = \{W_1, \ldots, W_k\}$ относительно подпространств $W_1, \ldots, W_k$ состоит в следующем.

1. Выбирается одно из подпространств $\{W_1, \ldots, W_k\}$. Вероятность выбора подпространства W_i равна $||P_{W_i}(|\psi\rangle)||_2^2$.

2. После выбора подпространства состояние $|\psi_{W_i}\rangle = P_{W_i}(|\psi\rangle)$ нормализуется, то есть конфигурация $|\psi'\rangle$ после измерения становится равной

$$|\psi'\rangle = \frac{P_{W_i}(|\psi\rangle)}{||P_{W_i}(|\psi\rangle)||_2}.$$

Вся информация, не принадлежащая $|\psi_{W_i}\rangle$, теряется.

3. Как результат измерения $\mathcal{O}$ мы получаем некоторое значение μ, которое называется *исходом измерения*. Всюду в работе μ есть информация о том, какое из подпространств $W_1, \ldots, W_k$ было выбрано в результате измерения, то есть $\mu = i$, где $i \in \{1, \ldots, k\}$.

Промежуточное измерение $\mathcal{O}$ производится на некотором этапе вычисления, и дальнейший процесс вычисления зависит от исхода μ этого промежуточного измерения.

Финальное измерение. В этом случае вектор текущей конфигурации проецируется на одно двух подпространств W_{acc}, W_{rej}: $W_{\text{acc}} \oplus W_{\text{rej}} = \mathcal{H}^d$, $W_{\text{acc}} \perp W_{\text{rej}}$. Будем называть W_{acc} (W_{rej}) подпространством принимающих (отвергающих) состояний. Оператор проекции на подпространство W_{acc} будем задавать при помощи матрицы проекции M_{acc} следующим образом. Пусть $|\psi\rangle$ – конфигурация и пусть подпространство $W_{\text{acc}} \subseteq \mathcal{H}^d$ принимающих состояний задается системой $\{|e_i\rangle\}_{i=1}^r$ ортонормированных векторов. *Матрица проекции* на подпространство принимающих состояний M_{acc} – это $(d \times d)$-матрица, которая содержит r ненулевых строк $M_{\text{acc}}[i_1]$, …, $M_{\text{acc}}[i_r]$, определяемых следующим образом: $M_{\text{acc}}[i] = (\alpha_1, \ldots, \alpha_d)$, где $|e_i\rangle = \alpha_1|1\rangle + \cdots + \alpha_d|d\rangle$, то есть $M_{\text{acc}}[i]$ – это представление базисного вектора $|e_i\rangle$ в стандартном базисе $\{|1\rangle, \ldots, |d\rangle\}$.

Для конфигурации $|\psi\rangle$ вероятность выбора подпространства принимающих состояний равна

$$p_{\text{acc}} = ||M_{\text{acc}}|\psi\rangle||_2^2.$$

1.2. Определение один раз измеряющей квантовой ветвящейся программы. Модель один раз измеряющей квантовой ветвящейся программы (measure-once QBP), определенная как обобщение уровневой забывающей BP, была впервые введена в работе [3]. В этой модели измерение текущей конфигурации производится один раз по окончании вычислительного процесса.

Определение 1. Один раз измеряющая квантовая BP ширины d и длины l ((d,l)- MO-QBP) есть

$$P = \big(|\psi_0\rangle, T, M_{\text{acc}}\big),$$

где $|\psi_0\rangle$ – начальная конфигурация P, $||\,|\psi_0\rangle||_2 = 1$, $T = \{\langle j_i,\ U_i(0),\ U_i(1)\rangle\}_{i=1}^l$ – последовательность d-мерных квантовых преобразований квантовой системы QS с d устойчивыми состояниями, где $U_i(0)$, $U_i(1)$ – унитарные $(d \times d)$-матрицы, M_{acc} – $(d \times d)$-матрица проекции на пространство принимающих состояний.

Процесс вычисления на входном наборе $\sigma = \sigma_1, \ldots, \sigma_n \in \{0,1\}^n$ осуществляется следующим образом.

• Вычисление начинается из начального вектора $|\psi_0\rangle$.

• На i-м шаге вычисления $(i = 1, \ldots, l)$ P считывает значение входной переменной $x_{j_i} = \sigma_{j_i}$ и преобразует текущий вектор конфигурации $|\psi\rangle$ в $|\psi'\rangle = U_i(\sigma_{j_i})|\psi_0\rangle$.

• После последнего l-го шага P производит измерение финального вектора конфигурации $|\psi_{\text{final}}\rangle = U_l(\sigma_{i_l}) \cdots U_1(\sigma_{i_1})|\psi_0\rangle$, задаваемое матрицей проекции M_{acc} на подпространство принимающих состояний.

• Вероятность принятия входного набора σ определяется как

$$Pr_{\text{acc}}(\sigma) = ||M_{\text{acc}}|\psi_{\text{final}}\rangle||_2^2.$$

1.3. Определение много раз измеряющей квантовой ветвящейся программы. В отличие от предыдущей модели, в модели много раз измеряющей квантовой ВР (measure-many QBP) измерение текущей конфигурации производится после каждого вычислительного шага, и выбор дальнейшего преобразования зависит от исхода измерения. Применение операции измерения в ходе вычисления нарушает обратимость процесса вычисления.

Пусть $\mathcal{H}^d = W_1 \oplus \cdots \oplus W_k$, $k \leq d$ – ортогональная декомпозиция пространства $\mathcal{H}^d$ на подпространства $W_1, \ldots, W_k$.

Определение 2. Много раз измеряющая квантовая ветвящаяся программа ширины d и длины l ((d,l)-MM-QBP) определяется как

$$P = \big(|\psi_0\rangle, R, M_{acc}\big),$$

где $|\psi_0\rangle$ – начальная конфигурация, R – последовательность (длины l) d-мерных квантовых преобразований квантовой системы QS с d-устойчивыми состояниями, определенная следующим образом:

$$R = \{\langle j_i, U_i^1(0), \ldots, U_i^k(0), U_i^1(1), \ldots, U_i^k(1)\rangle\}_{i=1}^l.$$

Здесь $U_i^j(0)$ и $U_i^j(1)$, $i = 1, \ldots, l$, $j = 1, \ldots, k$ – унитарные $(d \times d)$-матрицы, M_{acc} – матрица проекции на пространство принимающих состояний.

P осуществляет вычисление на входном наборе $\sigma = \sigma_1 \ldots \sigma_n \in \{0,1\}^n$ следующим образом.

• Вычисление начинается из начальной конфигурации $|\psi_0\rangle$.

• Каждый шаг j $(j = 1, \ldots, l)$ работы программы P состоит из двух частей:

Измерение: Производится промежуточное измерение $\mathcal{O} = \{W_1, \ldots, W_k\}$ текущей конфигурации $|\psi\rangle$ относительно подпространств $W_1, \ldots, W_k$. Результат измерения – вероятностный. Вероятность выбора подпространства W_i равна $||P_{W_i}(|\psi\rangle)||_2^2$. В качестве исхода измерения $\mathcal{O}$ на j-м шаге программа P выдает значение $\mu_j \in \{1, \ldots, k\}$. Конфигурация после измерения становится равной

$$|\psi'\rangle = \frac{P_{W_{\mu_j}}(|\psi\rangle)}{||P_{W_{\mu_j}}(|\psi\rangle)||_2};$$

Преобразование: P применяет к текущей конфигурации $|\psi'\rangle$ унитарное преобразование $U_j^{\mu_j}(\sigma_{i_j})$, определяемое исходом измерения μ_j на j-м шаге и входным символом $x_{i_j} = \sigma_{i_j}$.

• После l-го (последнего) шага P производит финальное измерение текущей конфигурации, задаваемое матрицей проекции M_{acc}.

1.4. Представление функций. Ветвящаяся программа P (квантовая или вероятностная) вычисляет функцию f с *неизолированной ошибкой* (в англоязычной литературе – unbounded error), если для любого $\sigma \in f^{-1}(1)$ вероятность принятия программой P входа σ удовлетворяет неравенству $Pr^P_{\text{acc}}(\sigma) > 1/2$ и для любого $\sigma' \in f^{-1}(0)$ справедливо неравенство $Pr^P_{\text{acc}}(\sigma') < 1/2$. В этом случае будем также говорить, что P вычисляет функцию f с неизолированной точкой сечения $1/2$;

Ветвящаяся программа P (квантовая или вероятностная) вычисляет функцию f с *изолированной ошибкой* (в англоязычной литературе – bounded error), если существует $\varepsilon \in (0, 1/2]$ такое, что для любого $\sigma \in f^{-1}(1)$ вероятность принятия программой P входа σ справедливо неравенство $Pr^P_{\text{acc}}(\sigma) \geq 1/2 + \varepsilon$ и для любого $\sigma' \in f^{-1}(0)$ вероятность принятия программой P входа σ' удовлетворяет неравенству $Pr^P_{\text{acc}}(\sigma') \leq 1/2 - \varepsilon$. При этом будем также говорить, что программа P вычисляет f с ε-изолированной точкой сечения $1/2$.

1.5. Классы сложности. Определим классы сложности на основе классических и квантовых один раз и много раз измеряющих ветвящихся программ.

- BPP-BP, PP-BP – классы функций, вычислимых с изолированной и неизолированной ошибками соответственно вероятностными ветвящимися программами полиномиальной сложности.
- BQP-BP_{MO} (BQP-BP_{MM}), $PrQP$-BP_{MO} ($PrQP$-BP_{MM}) – классы функций, вычислимых с изолированной и неизолированной ошибками соответственно один раз измеряющими (много раз измеряющими) квантовыми бинарными программами полиномиальной сложности.
- BPP-$OBDD$, PP-$OBDD$ – классы функций, вычислимых с ограниченной и неограниченной ошибками соответственно вероятностными $OBDD$ полиномиальной сложности.
- BQP-$OBDD_{MO}$ (BQP-$OBDD_{MM}$), $PrQP$-$OBDD_{MO}$ ($PrQP$- $OBDD_{MM}$) – классы функций, вычислимых с ограниченной и неограниченной ошибками соответственно много раз измеряющими квантовыми $OBDD$ полиномиальной сложности.

2. Классическое моделирование

Теорема 1 (классическое моделирование [8]). *Пусть f вычислима с неограниченной (ограниченной) ошибкой один раз измеряющей квантовой ветвящейся программой QBP Q. Тогда существует вероятностная ветвящаяся программа PBP P, которая вычисляет f с неограниченной ошибкой и при этом*

$$\text{Width}(P) = 4\text{Width}^2(Q) + 3, \quad \text{Length}(P) = \text{Length}(Q).$$

Идея доказательства теоремы состоит в следующем. Один раз измеряющая квантовая и вероятностная BP рассматриваются как частный случай вычислительной модели общего вида – линейной ветвящейся программы (LBP). Модель линейной ветвящейся программы ширины d и длины l устроена следующим образом. На каждом шаге вычисления состояние такой LBP P описывается вектором d-мерного линейного пространства над полем K. Вычисления на входном наборе $\sigma = \sigma_1, \ldots, \sigma_n$ начинаются из начального вектора μ_0. На i-м шаге вычисления $(i = 1, \ldots, l)$ P считывает значение входной переменной $x_{j_i} = \sigma_{j_i}$ и преобразует текущий вектор состояния μ, используя линейное преобразование $M_i(0)$ $(M_i(1))$,

если $x_{j_i} = 0$ ($x_{j_i} = 1$). После последнего l-го шага текущий вектор μ проецируется на подпространство принимающих состояний, в результате чего получим финальный вектор μ_{final}. Вероятность принятия входного слова $x = x_1, \ldots, x_n$ определяется как норма вектора μ_{final}. Используя такое определение линейной ВР, имеем следующее.

• Вероятностная ВР – это LBP над полем вещественных чисел. Линейное пространство состояний использует норму $\|\cdot\|_1$. Вектора μ состояний программы – вещественные вектора, при этом всегда выполняется равенство $\|\mu\|_1 = 1$ (за исключением финального вектора μ_{final}). Матрицы преобразования являются матрицами, стохастическими по столбцам (сохраняющими $\|\cdot\|_1$ преобразуемого вектора).

• Квантовая ВР – это LBP над полем комплексных чисел. Линейное пространство состояний использует норму $\|\cdot\|_2$. Вектора μ состояний программы – комплекснозначные вектора, при этом всегда выполняется равенство $\|\mu\|_2 = 1$ (за исключением финального вектора μ_{final}). Матрицы преобразования являются комплекснозначными унитарными матрицами (сохраняющими $\|\cdot\|_2$ преобразуемого вектора).

Процесс моделирования заключается в преодолении указанных различий над данными двумя разновидностями LBP шаг за шагом. При этом длина ветвящейся программы не увеличивается, ширина возрастает квадратично. Однако может быть потеряно свойство изолированности ошибки, то есть если QBP вычисляла f с ограниченной ошибкой, то построенная PBP может вычислять f с неограниченной ошибкой.

Полное доказательство теоремы 1 приведено в [8].

Следствием из теоремы 1 являются следующие соотношения классов сложности:

$$BQP\text{-}BP_{MO} \subseteq PP\text{-}BP,$$

$$PrQP\text{-}BP_{MO} \subseteq PP\text{-}BP,$$

$$BQP\text{-}OBDD_{MO} \subseteq PP\text{-}OBDD,$$

$$PrQP\text{-}OBDD_{MO} \subseteq PP\text{-}OBDD$$

3. Квантовое моделирование

Согласно законам квантовой механики преобразования изолированной квантовой системы описываются унитарными матрицами. Одним из основных свойств унитарных преобразований является их обратимость. Единственным необратимым преобразованием такой квантовой системы является измерение (извлечение результата вычислений). Преобразования, применяемые в ходе классических вычислений, в общем случае не являются обратимыми. При этом результат необратимого преобразования не содержит достаточно информации для восстановления исходных аргументов. Применение измерения в процессе вычисления нарушает обратимость процесса вычисления. Данный способ используется в первом методе квантового моделирования – моделировании с использованием многократного измерения. В $MM\text{-}QBP$ после каждого вычислительного шага производится измерение текущей конфигурации, и выбор дальнейшего преобразования зависит от исхода измерения.

Другой способ сделать необратимые преобразования обратимыми – хранить всю траекторию вычисления. Однако это приводит к экспоненциальному увеличению сложности. Второй предлагаемый нами метод квантового моделирования использует сохранение промежуточных состояний не на всей траектории вычисления, а только на некоторых отдельных шагах. Сохранение состояний вычислений на некоторых шагах позволяет восстановить всю историю вычисления.

Отметим также, что если в первом из представленных ниже методов квантовость используется только для моделирования классической вероятности, то во втором существенную роль играет квантовый параллелизм. Другими словами, если в вероятностной ВР вероятностным образом происходит выбор одной из траекторий вычислений, то в квантовой ВР все вычислительные траектории выполняются одновременно, каждая с соответствующей амплитудой.

3.1. Квантовое моделирование с многократным измерением.

Теорема 2 (квантовое моделирование с использованием многократного измерения [8]). *Пусть функция f вычислима вероятностной забывающей ветвящейся программой (d,l)-PBP P. Тогда она вычислима много раз измеряющей квантовой ветвящейся программой (d,l)-MM-QBP Q такой, что*

$$Pr^Q_{\text{acc}}(\sigma) = Pr^P_{\text{acc}}(\sigma) \quad \textit{для всех} \;\; \sigma \in \{0,1\}^n.$$

Идея доказательства состоит в следующем. При построении квантовой ВР используется квантовое моделирование классической вероятности.

Если p – классическая вероятностная переменная, принимающая значение 0 и 1 с вероятностью $1/2$, то ее поведение может быть промоделировано квантовым кубитом q, приготовленным в состоянии $|q\rangle = (1/\sqrt{2}, 1/\sqrt{2})^T$. Измеряя данное состояние по отношению к стандартному базису $\{|0\rangle, |1\rangle\}$, как результат измерения мы получаем значение кубита равным 0 или 1 с вероятностью $1/2$. Аналогично, если p – классическая вероятностная переменная, принимающая значение i $(i = 1, \ldots, d)$ с вероятностью p_i, ее поведение моделируется квантовой системой из $\log d$[1] кубитов следующим образом:

$$|\psi\rangle = (1, 0, \ldots, 0) \xrightarrow{U} (\sqrt{p_1}, \ldots, \sqrt{p_d}),$$

Измеряя полученное состояние по отношения к стандартному базису $\{|1\rangle, \ldots, |d\rangle\}$, мы получаем значение i с вероятностью p_i.

Метод квантового моделирования с использованием многократного измерения использует измерение текущей квантовой суперпозиции по отношению к стандартному базису на каждом шаге вычисления. После каждого измерения мы имеем распределение вероятностей нахождения квантовой ВР в базисных состояниях, соответствующих классическим состояниям вероятностной ВР. Унитарные преобразования, применяемые на каждом шаге вычисления, зависят от значения считанной входной переменной, и от результата предыдущего измерения.

На основе данного метода квантового моделирования по вероятностной ВР, вычисляющей функцию f с некоторой вероятностью, строится квантовая ВР, использующая измерение текущей суперпозиции на каждом шаге. При этом полученная квантовая много раз измеряющая ВР будет вычислять функцию f с той же вероятностью и будет иметь ту же длину и ширину, что и исходная вероятностная ВР, например, если исходная ВР была один раз читающей ВР, то построенная ВР будет один раз читающей ВР. Полное доказательство теоремы 2 приведено в [8].

Следствием из теоремы 2 являются следующие соотношения классов сложности:

$$BPP\text{-}BP \subseteq BQP\text{-}BP_{MM},$$

$$PP\text{-}BP \subseteq PrQP\text{-}BP_{MM},$$

$$BPP\text{-}OBDD \subseteq BQP\text{-}OBDD_{MM},$$

$$PP\text{-}OBDD \subseteq PrQP\text{-}OBDD_{MM}.$$

[1] Все логарифмы в данной работе берутся по основанию 2.

3.2. Квантовое моделирование с однократным измерением. Метод квантового моделирования с многократным измерением использует измерение текущей квантовой суперпозиции для нарушения обратимости вычислений. При этом на каждом шаге мы полностью забываем историю прошедших вычислений. В данном пункте мы опишем метод квантового моделирования вероятностной ВР с использованием однократного измерения по окончании вычислений.

Приведем некоторые обозначение и определения, которые понадобятся нам в этом пункте.

Через I_n будем обозначать единичную матрицу размера $n \times n$. Пусть $a = (a_1, \ldots, a_n)$, $b = (a_1, \ldots, b_m)$. Тензорное (правое кронекерово) произведение векторов a и b – это вектор размерности nm, определяемый следующим образом: $a \otimes b = (a_1b_1, a_1, b_2, \ldots, a_1b_m, \ldots, a_nb_m)$. Тензорное произведение k векторов a будем обозначать $a^{\otimes k}$.

Тензорное произведение матриц

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \quad \text{и} \quad B = \begin{pmatrix} b_{11} & \cdots & b_{1k} \\ \vdots & \ddots & \vdots \\ b_{l1} & \cdots & b_{lk} \end{pmatrix}$$

есть $(ml \times nk)$-матрица

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}.$$

Определение 3. Вероятностную ветвящуюся программу P будем называть ветвящейся программой *нормальной формы*, если P имеет следующий вид:

- P может быть разбита на две части: вероятностную, содержащую только вероятностные вершины, и детерминированную, содержащую только детерминированные вершины;
- вероятностная часть программы P находится в верхней части программы и представляет собой дерево с корнем – начальной вершиной P;

Теорема 3 [12]. *Пусть $f : \{0,1\}^n \to \{0,1\}$ – булева функция над множеством переменных $X = \{x_1, \ldots, x_n\}$, P – вероятностная ВР, вычисляющая функцию f с ε-изолированной точкой сечения $1/2$ $(0 < \varepsilon < 1/2)$. Тогда для любого $\delta : 0 < \delta < \varepsilon$ существует вероятностная ВР P' такая, что:*

- *P' вычисляет функцию f с $(\varepsilon - \delta)$-изолированной точкой сечения $1/2$;*
- *P' является вероятностной ВР нормальной формы;*
- *$S(P') = O(n/\delta^2 S(P))$.*

Теорема 4 (квантовое моделирование с использованием однократного измерения). *Пусть функция f вычислима с ограниченной (неограниченной) ошибкой вероятностной ВР P. Тогда она вычислима с ограниченной (неограниченной) ошибкой один раз измеряющей квантовой ВР Q и при этом*

$$\mathrm{Length}(Q) = O(\mathrm{Length}(P)^k),$$

$$\mathrm{Width}(Q) = O(\mathrm{Width}(P)^{c \log(\mathrm{Length}(P))}),$$

где k, c – некоторые константы.

Доказательство. Пусть P – вероятностная ВР, вычисляющая функцию f с ограниченной ошибкой. Процесс построения квантовой ВР Q, вычисляющей f, состоит из следующих этапов: определение начальной конфигурации программы Q, определение последовательности преобразований и определение множества финальных состояний (задающих подпространство принимающих состояний) программы Q.

Первый этап моделирования – определение начальной конфигурации. По РВР P строим РВР P' нормальной формы. При этом сложность программы возрастает по порядку не более чем в n раз. Согласно теореме 3 P' вычисляет f с ограниченной ошибкой.

Пусть $\mu = (p_1, \ldots, p_d)$ – вероятностное распределение вершин программы P' после считывания всех вероятностных переменных. Обозначим $|a\rangle = (\sqrt{p_1}, \ldots \sqrt{p_d})^T$.

Второй этап моделирования – определение последовательности преобразований. На данном этапе моделирования рассматриваем детерминированную часть программы P'. Обозначим ее через D. Без ограничения общности полагаем, что D является уровневой, забывающей и все уровни имеют одинаковое количество d вершин, перенумерованных $1, \ldots, d$. Пусть l – длина программы D, $T_D = \{\langle j_i, M_i(0),\ M_i(1)\rangle\}_{i=1}^{l}$ – последовательность преобразований детерминированной программы D, $M_i(0),\ M_i(1)$ – булевы $(d \times d)$-матрицы переходов, соответствующие i-му уровню программы D, каждый столбец которых содержит ровно одну единицу. А именно, в матрицах $M_i(\sigma)$, $(i = 1, \ldots, l,\ \sigma \in \{0, 1\}$ единица стоит на пересечении s-го столбца и s'-й строки, если в ВР D есть ребро из вершины s i-го уровня в вершину s' $(i+1)$-го уровня, помеченное σ.

Если программа D является обратимой, то преобразования на каждом уровне реализуют некоторую перестановку на d вершинах. В этом случае $M_i(0),\ M_i(1)$, $i = 1, \ldots, l$, – перестановочные матрицы, которые являются унитарными. В этом случае последовательность преобразований квантовой программы Q определим следующим образом:

$$T_Q = \{\langle j_i, M_i(0), M_i(1)\rangle\}_{i=1}^{l}.$$

Начальную конфигурацию программы Q определим как $|\psi_0\rangle = |a\rangle$.

Если программа D не является обратимой, переходим к следующей стадии моделирования.

На данной стадии моделирования используется подход, предложенный в [13]. В основе этого подхода используется следующая игра. Пусть у нас имеется $k+1$ камень, один из которых постоянно лежит на позиции 0, а остальные в начальный момент времени находятся в куче. На каждом шаге мы можем либо положить камень на позицию i, либо снять его с позиции i. Но при этом и положить, и снять камень возможно, только если на $i-1$ позиции уже находится другой камень. Цель игры – положить камень как можно дальше от позиции 0.

Пусть d_k – наиболее удаленная позиция от позиции 0, на которую мы можем положить камень, t_k – число шагов, необходимых для этого. В [13] был предложен следующий рекурсивный алгоритм $P(k)$:

1) если $k = 0$, то $d_k = 0$ и ничего делать не надо;

2) если $k > 0$, то запускаем алгоритм $P(k-1)$, чтобы положить камень в позицию d_{k-1}, кладем камень в позицию $d_{k-1} + 1$, собираем $k-1$ камень в кучу, запуская алгоритм, обратный к $P(k-1)$, и выполняем $P(k-1)$ с использованием собранных камней, используя в качестве начальной позиции позицию $d_{k-1} + 1$.

Индукцией несложно показывается, что для данной стратегии $d_k = 2^k - 1$, $t_k = (3^k - 1)/2$. Показывается также, что данная стратегия неулучшаема [13].

Аналогично [13] будем строить последовательность преобразований программы Q согласно этой стратегии. Квантовая ВР Q устроена следующим образом. Каждый уровень ВР Q содержит d^k вершин, где k мы определим позднее. Вершины перенумерованы таким образом, что каждая вершина имеет k индексов $i_1, \ldots, i_k$, где $1 \le i_j \le d$.

Начальная конфигурация – $|\psi_0\rangle = |a\rangle \otimes |e\rangle^{\otimes(k-1)}$, где $|e\rangle = (1, 0, \ldots, 0)^T$ – d-мерный вектор. Будем говорить, что конфигурация программы Q состоит из k регистров $|r_1\rangle, \ldots, |r_k\rangle$, где каждый регистр содержит d состояний $|1\rangle, \ldots, |d\rangle$, представленных с соответствующей амплитудой. Квантовая ВР Q моделирует программу D с использованием данных k регистров согласно описанной выше стратегии игры. При этом регистры играют роль камней, а уровни программы D соответствуют позициям, на которые можем класть камни. Если i-й регистр хранит состояние программы D на некотором уровне, то это соответствует тому, что i-й камень покрывает соответствующую позицию. Если регистр находится в состоянии $|e\rangle$, то это значит, что камень лежит в куче. Таким образом, начальная конфигурация $|\psi_0\rangle$ соответствует начальному состоянию игры – один камень лежит в нулевой позиции, остальные $k-1$ камень находятся в куче.

Последовательность преобразований $T_Q = \{\langle j_i,\ U_i(0),\ U_i(1)\rangle\}_{i=1}^m$ определяется последовательностью преобразований T_D и стратегией игры. Квантовая ВР Q моделирует работу программы D, выполняя преобразования каждого уровня программы D в соответствии со стратегией следующим образом.

Пусть на шаге j в соответствии со стратегией игры мы кладем камень в позицию n. При этом в позиции $n-1$ уже лежит другой камень. Это означает, что Q вычисляет состояние ВР D на уровне n из состояния на уровне $n-1$. При этом для размещения нового состояния Q использует некоторый пустой регистр $|r_t\rangle$ (находящийся в состоянии $|e\rangle$), что соответствует тому, что мы используем t-й камень, чтобы покрыть позицию n, а некоторый регистр $|r_s\rangle$ при этом хранит состояние программы D на $(n-1)$-м уровне. Для простоты изложения предположим, что $|r_s\rangle$ и $|r_t\rangle$ – два соседних регистра, то есть $t = s+1$. Преобразование на j-м шаге задействует только регистры $|r_s\rangle$ и $|r_t\rangle$ и тождественно на остальных $k-2$ регистрах. То есть $U_j(\sigma) = I_d \otimes \cdots \otimes I_d \otimes U_j^{st}(\sigma) \otimes I_d \otimes \cdots \otimes I_d$.

Матрица $U_j^{st}(\sigma)$ – это $d^2 \times d^2$ блочно-диагональная матрица, где на диагонали стоят $d \times d$-матрицы $B_1(\sigma), \ldots, B_d(\sigma)$. Матрицы $B_v(\sigma),\ v = 1, \ldots, d$, определяются преобразованием $M_{n-1}(\sigma)$ ветвящейся программы D. А именно, для каждого $B_v(\sigma),\ v = 1, \ldots, d$, – это перестановочные матрицы, переставляющие состояния 1 и v', если в программе D существует помеченное σ ребро, ведущее из вершины v $(n-1)$-го уровня в вершину v' n-го уровня. Перестановка на остальных состояниях тождественна. Таким образом, общее состояние регистров $|r_s\rangle$ и $|r_t\rangle$ после преобразования становится $|1\rangle B_1(\sigma)|r_t\rangle + |d\rangle B_d(\sigma)|r_t\rangle$. Для каждой вершины v $(n-1)$-го уровня результат преобразования размещается в своем участке памяти. Если $|r_s\rangle$ и $|r_t\rangle$ не являются двумя соседними регистрами, то преобразование выполняется аналогично и осуществляет перестановку соответствующих состояний.

Пусть на шаге j в соответствии со стратегией игры мы снимаем камень с позиции n. При этом в позиции $n-1$ также должен лежать камень. Это означает, что мы выполняем преобразование, обратное тому, которое мы выполняли при вычислении состояния на уровне n. Матрица преобразования в этом случае совпадает с матрицей прямого преобразования в соответствии с определением этой перестановочной матрицы.

Определение множества финальных состояний. Пусть F – множество финальных состояний ВР D, $|r_t\rangle$ – регистр, в котором размещено состояние последнего l-го уровня программы D. К множеству финальных состояний QBP Q отнесем

все состояния $|i_1 \dots i_d\rangle$, у которых $i_t \in F$.

По построению полученная QBP Q представляет функцию f с той же вероятностью, что и исходная PBP P. Оценим сложность построенной QBP Q в соответствии с первым и вторым этапами моделирования. На первом этапе сложность увеличивается по порядку не более чем в n раз. Оценим изменения сложности на втором этапе. Поскольку согласно стратегии игры $\mathrm{Length}(D) = d_k$, то число регистров $k = \lceil \log(\mathrm{Length}(D) + 1) \rceil$. Так как длина построенной QBP – это число шагов в игре, то, используя полученное значение k, имеем: $\mathrm{Length}(Q) = O(\mathrm{Length}(D)^{\log 3})$. Ширина построенной программы $\mathrm{Width}(Q) = \mathrm{Width}(D)^k = \mathrm{Width}(D)^{\lceil \log(\mathrm{Length}(D)+1) \rceil}$. Следовательно,

$$\mathrm{Length}(Q) = O(\mathrm{Length}(P)^k),$$

$$\mathrm{Width}(Q) = O(\mathrm{Width}(P)^{c \log(\mathrm{Length}(P))}),$$

где k, c – некоторые константы. □

4. Сравнительный анализ методов

Таким образом, в работе рассмотрены три различных метода моделирования: классическое моделирование квантовой BP и два различных метода квантового моделирования вероятностной BP. При этом при классическом моделировании квантовой BP сложность увеличивается полиномиально: длина программы не изменяется, ширина изменяется квадратично. Однако «платой» за незначительное увеличение сложности является потеря важного свойства BP – свойства изолированности точки сечения: если исходная квантовая BP вычисляла функцию f с большой вероятностью правильного результата, то в построенной классической BP вероятность ошибки сильно возрастает.

При квантовом моделировании вероятностной BP основной трудностью, которую приходится преодолевать, является то, что квантовые преобразования, которые реализуются посредством унитарной эволюции, в силу своей природы являются обратимыми. Классические преобразования в общем случае являются необратимыми. Два предлагаемых метода моделирования по-разному решают данную проблему. Первый метод моделирования использует измерение на каждом вычислительном шаге с целью нарушить обратимость вычислений, поскольку из возможных преобразований изолированной квантовой системы только измерение является необратимым преобразованием. При этом в построенной квантовой BP квантовость используется только для моделирования вероятности. Использование промежуточного измерения как способа выполнения необратимых преобразования не изменяет сложность программы.

Второй предлагаемый метод использует единственное измерение по окончании вычислений. При этом основными моментами моделирования являются следующие. Во-первых, вероятностный выбор заменяется на одновременное параллельное выполнение, то есть если в вероятностной BP один вычислительный путь выполняется с некоторой вероятностью, то в построенной BP с использованием квантового параллелизма запускаются одновременно все вычислительные траектории с амплитудами, квадраты которых равны соответствующим вероятностям. Второй момент связан с обратимым выполнением необратимых преобразований. Чтобы обратимым образом выполнять необратимые преобразования, мы вынуждены сохранять исходные аргументы каждой операции. При этом, чтобы память не возрастала очень сильно, в процессе вычисления мы освобождаем участки памяти для повторного использования, запуская некоторые шаги вычисления в обратную сторону. В результате такого моделирования длина программы возрастает

полиномиально, ширина возрастает как $\mathrm{Width}(P)^{c\log(\mathrm{Length}(P))}$, то есть при таком моделировании мы не остаемся в классе BP полиномиальной сложности. Однако следует отметить, что важной сложностной характеристикой квантовых BP является число k кубит, используемых для вычислений. Для данной модели BP $k = \log(\mathrm{Width}(Q))$ и значение k при моделировании возрастает следующим образом. Если для представления текущего состояния BPB P использовалось k бит, то построенная QBP Q использует $O(k\log(\mathrm{Length}(P))$ кубит. В частности, для BP полиномиальной сложности значение k возрастет по порядку не более чем в квадрат.

Работа выполнена при финансовой поддержке фонда «Научный потенциал» (“Human capital foundation”).

Summary

A.F. Gainutdinova. Quantum and Classical Simulation of Quantum Branching Programs.

The paper views a model of branching programs (BP). A model of classical probabilistic BP is considered along with two different models of quantum BP (once-measuring and many time measuring quantum BP). Three different methods of simulation of BPs are presented: method of classical simulation of quantum BPs and two different methods of quantum simulation of probabilistic BPs. Complexity of methods is proved. Comparative analysis of methods is presented.

Key words: branching programs, computation complexity, quantum and classical simulation.

Литература

1. *Манин Ю.И.* Вычислимое и невычислимое. – М.: Сов. радио, 1980. – 128 с.
2. *Feynman R.* Simulating physics with computers // Int. J. Theor. Phys. – 1982. – V. 21, No 6,7. – P. 467–488.
3. *Ablayev F., Gainutdinova A., Karpinski M.* On Computational Power of Quantum Branching Programs // Proc. of the 13th Int. Symposium, Fundamentals of computation theory (FCT 2001, Riga, Latvia). LNCS. – 2001. – V. 2138. – P. 59–70.
4. *Баррингтон Д.* Ветвящиеся программы ограниченной ширины, имеющие полиномиальную сложность, распознают в точности языки из NC^1 // Киберн. сб. – М.: Мир. 1991. – Вып. 28. – С. 94–113.
5. *Nakanishi M., Hamaguchi K., Kashiwabara T.* Ordered quantum branching programs are more powerful than ordered probabilistic branching programs under a bounded-width restriction // Proc. of the 6th Annual Int. Conf. on Computing and Combinatorics, COCOON'2000. LNCS. – Springer-Verlag, 2000. – V. 1858. – P. 467–476.
6. *Sauerhoff M., Sieling D.* Quantum branching programs and space-bounded nonuniform quantum complexity // Theor. Comput. Sci. – 2005. – V. 334. – P. 177–225.
7. *Гайнутдинова А.Ф.* О сравнительной сложности квантовых и классических бинарных программ // Дискр. матем. – 2002. – Т. 14, Вып. 3. – С. 109–121.
8. *Гайнутдинова А.Ф.* Моделировании квантовых и классических бинарных программ // Дискр. анализ и исслед. операций. Сер. 1. – 2006. – Т. 13, № 1. – С. 45–64.
9. *Borodin A., Fischer M., Kirkpatrick D., Lynch N., Tompa M.* A time-space tradeoff for sorting on non-oblivious machines // Proc. 20th Annual Symposium on Foundations of Computer Science. – 1979. – P. 319–327.

10. *Ablayev F., Karpinski M.* On the power of randomized branching programs // Proc. 28th ICALP (1996). LNCS. – Springer, 1996. – V. 1099. – P. 348–356.

11. *Gruska J.* Quantum computing. – McGraw-Hill Publishing Company, 1999. – 419 p.

12. *Sauerhoff M.* Complexity theoretical results for randomized branching programs: Ph.D. Dissertation. – 1998. – URL: http://ls2- www.cs.uni-dortmund.de/∼sauerhof/publications.shtml.

13. *Spalek R.* Space complexity of quantum computation: Dissertation. – 2002. – URL: http://eccc.hpi-web.de/eccc-local/ECCC-These/spalek.html.

Поступила в редакцию
27.02.09

Гайнутдинова Аида Фаритовна – кандидат физико-математических наук, ассистент кафедры теоретической кибернетики Казанского государственного университета.

E-mail: *aida@ksu.ru*