

Раздел VII. Высокопроизводительные вычислительные алгоритмы

УДК 681.3.06:530.145.001.57

О.К. Евсеев, С.М. Гушанский, В.Ф. Гузик

МОДЕЛИРОВАНИЕ КВАНТОВЫХ ВЫЧИСЛЕНИЙ НА ОСНОВЕ QUIDD-ГРАФОВ

Создание алгоритмов квантовых вычислений требует возможности проверить результаты их работы. Математические модели доступны любому разработчику, но их эффективность не позволяет реализовать вычисления с объёмом регистра более 12 бит. В данной статье приводится структура и алгоритмы разработанной 2-уровневой модели, обладающей повышенной эффективностью. Предложены усовершенствованные алгоритмы перестановки q -бит и синтеза QuIDD-графа путём исключения бит-индекса элементов исходной матрицы. Определено направление оптимизации методики QuIDD.

Квантовые вычисления; моделирование; q -бит; QuIDD-граф; матрицы; вектор состояния; тензорное произведение; оператор; эффективность.

O.K. Evseev, S.M. Gushansky, V.F. Guzik

QUIDD-BASED QUANTUM COMPUTER MODELING

Creating algorithms for quantum computing requires the ability to verify their results through the use of physical or mathematical models of quantum computer. Mathematical models are available to any developer, but their efficiency is too low to simulate the register with volume more than 12 bits. This article describes the structure and algorithms of developed 2-level model of increased efficiency. The enhanced algorithms for permutation of q -bits and QuIDD-graph synthesis by removing the bits of index of original matrix elements are proposed. The direction for further optimization for QuIDD techniques are defined.

Quantum computation; simulation; q -bit; QuIDD graph; matrix; state vector; tensor product; operator; efficiency.

В настоящее время универсальные квантовые вычислители существуют лишь в теории, однако их физические прототипы, не обладающие достаточной мощностью, уже построены и активно совершенствуются. Разработанные физические устройства являются моделями с объёмом регистра не более 10–20 [1,2] квантовых бит (q -бит) и ограниченной функциональностью. Такая разрядность означает их фактическую пригодность для задач, оперирующих числами, не превышающими 1048576. Вместе с тем, стоимость подобных моделей такова, что их могут позволить себе лишь некоторые научно-исследовательские институты, занимающиеся совершенствованием возможностей квантовых вычислительных устройств [1].

Параллельно с созданием квантовой вычислительной аппаратуры ведётся анализ и разработка вычислительных алгоритмов, позволяющих эффективно решать задачи, принципиально слишком сложные для классической вычислительной техники. Для этих целей служат программные математические модели, которые значительно доступнее аппаратных.

Процесс моделирования строится на основании уравнении Шредингера для нерелятивистского случая эволюции квантовых систем:

$$i\hbar \frac{\partial \Psi}{\partial t} = H\Psi, \quad (1)$$

где Ψ – волновая функция состояния системы, H – некоторый эрмитов оператор, однозначно определяющий изменение состояния квантовой системы. Волновая функция для ансамбля из n q-битов в общем виде записывается с помощью формулы:

$$\Psi_n = \sum_{k=0}^{2^n-1} \lambda_k |k\rangle; \quad \sum_{k=0}^{2^n-1} |\lambda_k|^2 = 1. \quad (2)$$

Здесь k – номер возможного состояния системы, а λ_k – амплитуда, задающая вероятность такого состояния. Полная вероятность всех состояний системы равна единице.

Моделирование квантовых вычислений заключается в задании начальных состояний системы и формировании последовательности управляющих операторов H , реализующих квантовый вычислительный алгоритм. Технически процесс моделирования заключается в получении необходимых операторов и умножении их на волновую функцию квантовой системы. Фактически, волновая функция Ψ_n

выражается вектором из 2^n комплексных чисел. Оператор, реализующий вычисление или шаг вычисления, в этом случае будет выражаться матрицей размерности $2^n \times 2^n$. Очевидно, что наблюдаемый экспоненциальный рост сложности представления системы сильно ограничивает возможности модели: к примеру, одна матрица комплексных чисел двойной точности для размерности 12 q-бит будет занимать около 1 Гб и обрабатываться несколько десятков секунд.

Кроме того, добавление одного q-бита к системе ведёт к четырёхкратному увеличению потребления памяти. Этот факт значительно затрудняет моделирование систем большей размерности.

Таким образом, при использовании прямой реализации матричного математического аппарата, на классических ЭВМ с объёмом оперативной памяти около 4 Гб и производительностью порядка 160 млрд. операций в секунду (2×2200 МГц), фактически, возможно моделировать квантовые вычислительные системы размерности не более 12 q-бит.

В ходе научно-исследовательской работы над оценкой эффективности и реализуемости квантовых алгоритмов [3] было принято решение о построении математической модели квантового вычислителя, способной работать более чем с 12 q-битами, и превосходящей по эффективности матричные модели [4].

Для повышения эффективности модели может быть использована архитектура, содержащая 2 уровня, которые основываются на 2 разных реализациях матричного математического аппарата. Первый уровень оперирует массивами комплексных чисел и формирует входные данные для работы второго, графового уровня. Графовый уровень использует QuIDD-структуры [5], требующие значительно меньших объёмов памяти. На этом уровне протекает весь процесс моделирования квантового алгоритма. Выходные данные снова возвращаются на первый уровень в исходное представление, что позволяет экономить вычислительные ресурсы и иметь входные и выходные данные в удобном для анализа представлении.

Матричный математический аппарат первого уровня модели был дополнен безоператорным алгоритмом, позволяющим эффективно решить проблему перестановки q -бит квантового регистра (рис. 1).

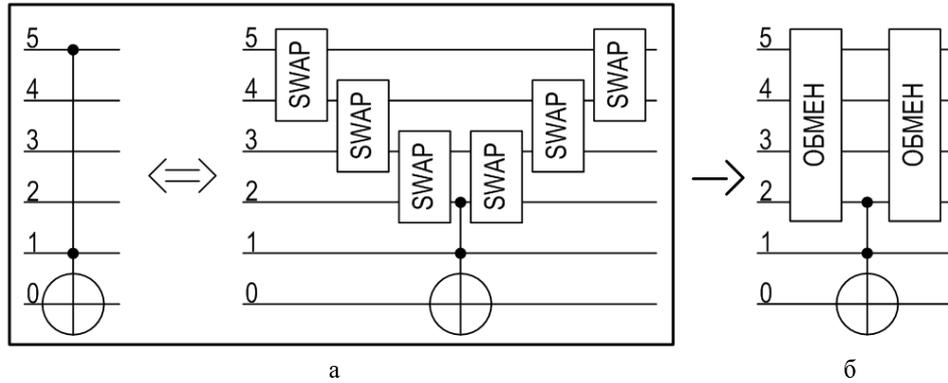


Рис. 1. Перестановка q -бит: а – использование SWAP для перестановки 3 q -бит в схеме с CCN; б – разработанный безоператорный алгоритм обмена состояний q -бит

Представленный оператор воздействует на 0-ый q -бит под управлением 1-го и 5-го. В матричном представлении получить такой оператор удастся только с помощью каскада операторов SWAP до CCN (рис. 1) и такого же каскада после него. Естественно, каждый оператор SWAP должен быть соразмерен квантовому регистру, а общее количество операторов соответствует удвоенной разнице между индексами обмениваемых q -битов. Следовательно, возникают вычислительные затраты, не относящиеся непосредственно к алгоритму квантового вычисления порядка

$$t_{SWAP} = O(2^{2N} \cdot \Delta i), \quad (3)$$

где N – количество q -битов в регистре, Δi – разность индексов q -бит, чьи состояния должны быть подвергнуты обмену.

Разработанный для решения данной проблемы алгоритм основывается на следующем утверждении:

Теорема: перестановка j -го и k -го q -битов квантового регистра приведёт к такой перестановке компонентов вектора состояния этого регистра, что состояния вида $|\dots X_j \dots X_k \dots\rangle$, у которых значения j -го и k -го битов отличаются, будут заменены на компоненты с вида $|\dots X_k \dots X_j \dots\rangle$. При этом количество необходимых перестановок компонентов вектора состояния составит $2^{N/2}$.

$$|\dots X_j \dots X_k \dots\rangle \xrightarrow{j \neq k} |\dots X_k \dots X_j \dots\rangle. \quad (4)$$

Например, для регистра из 4-х q -бит, состояния которого задаются в виде $|X_3 X_2 X_1 X_0\rangle$, потребуется 4 перестановки. Для перестановки 3-го и 0-го q -бит должны быть заменены следующие состояния:

$$\begin{aligned}
 |0001\rangle &\xrightarrow{x_3=0, x_0=1} |1000\rangle \\
 |0011\rangle &\xrightarrow{x_3=0, x_0=1} |1010\rangle \\
 |0101\rangle &\xrightarrow{x_3=0, x_0=1} |1100\rangle \\
 |0111\rangle &\xrightarrow{x_3=0, x_0=1} |1110\rangle
 \end{aligned} \tag{5}$$

Доказательство:

- а) количество амплитуд не изменяется при перестановке;
- б) значения амплитуд не меняются при перестановке.

Модель квантового регистра строится посредством суперпозиции моделей квантовых бит через тензорное произведение. Так, модель регистра из 3 q-бит представляет собой вектор-столбец:

$$|\Psi\rangle = \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} |0\rangle |1\rangle \otimes \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} |0\rangle |1\rangle \otimes \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} |0\rangle |1\rangle = \begin{pmatrix} a_2 \begin{pmatrix} a_1 \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \\ b_1 \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \end{pmatrix} \\ b_2 \begin{pmatrix} a_1 \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \\ b_1 \begin{pmatrix} a_0 \\ b_0 \end{pmatrix} \end{pmatrix} \end{pmatrix} = \begin{pmatrix} a_2 a_1 a_0 |000\rangle \\ a_2 a_1 b_0 |001\rangle \\ a_2 b_1 a_0 |010\rangle \\ a_2 b_1 b_0 |011\rangle \\ b_2 a_1 a_0 |100\rangle \\ b_2 a_1 b_0 |101\rangle \\ b_2 b_1 a_0 |110\rangle \\ b_2 b_1 b_0 |111\rangle \end{pmatrix} \cdot \tag{6}$$

При суперпозиции будет получено 2^N произведений вида $\{a_2, b_2\} \cdot \{a_1, b_1\} \cdot \{a_0, b_0\}$, в которые амплитуды каждого q-бита входят только 1 раз. Последовательность вхождения q-битов в регистр не изменит ни количество амплитуд регистра, ни сами амплитуды (следует из определения тензорного произведения), так как произведения формируются из одних и тех же сомножителей. Следовательно, для перестановки q-битов достаточно произвести перестановку компонентов вектора состояния.

в) порядок перестановки $|\dots X_j \dots X_k \dots\rangle \xrightarrow{j \neq k} |\dots X_k \dots X_j \dots\rangle$

Порядок вхождения векторов состояния q-битов в тензорное произведение определяет последовательность расположения q-битов в регистре. При порядке вхождения q-битов $|\dots X_k \dots X_j \dots\rangle$ вместо $|\dots X_j \dots X_k \dots\rangle$ амплитуды регистра будут формироваться из тех же сомножителей, и, вследствие коммутативности умножения, амплитуды, чьи индексы после перестановки q-бит не изменятся, будут занимать те же места, что и до перестановки. Амплитуды регистра, чьи индексы после перестановки изменятся (т.к., $a_j \cdot b_k \neq a_k \cdot b_j$), будут взаимно переставлены:

$$\begin{pmatrix} a_2 a_1 a_0 \\ a_2 a_1 b_0 \\ a_2 b_1 a_0 \\ a_2 b_1 b_0 \\ b_2 a_1 a_0 \\ b_2 a_1 b_0 \\ b_2 b_1 a_0 \\ b_2 b_1 b_0 \end{pmatrix} \begin{matrix} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{matrix} \xrightarrow{x_1=1, x_0=0} \begin{pmatrix} a_2 a_1 a_0 \\ a_2 b_1 a_0 \\ a_2 a_1 b_0 \\ a_2 b_1 b_0 \\ b_2 a_1 a_0 \\ b_2 b_1 a_0 \\ b_2 a_1 b_0 \\ b_2 b_1 b_0 \end{pmatrix} \begin{matrix} |000\rangle \\ |001\rangle \\ |010\rangle \\ |011\rangle \\ |100\rangle \\ |101\rangle \\ |110\rangle \\ |111\rangle \end{matrix}. \quad (7)$$

г) количество перестановок составит $2^{N/2}$.

Количество элементов вектора состояний составляет 2^N . В этом случае ровно половина ($2^{N/2}$) индексов элементов содержит конкретный разряд равным 0 и половина – 1. Количество индексов с разными значениями разрядов j и k , очевидно, также составляет $2^{N/2}$.

Исходя из приведённых выше рассуждений, производится анализ индексов элементов вектора состояния квантового регистра на предмет изменения значений при перестановке в нём битов j и k . Если значение индекса после перестановки изменится, в векторе состояния производится перестановка элементов, соответствующих значению индекса до и после перестановки его битов j и k .

Такой алгоритм не использует матричные операторы и требует вычислительных затрат порядка

$$t = O(2^{N/2}); t_{SWAP} = O(2^{2N} \cdot \Delta i). \quad (6)$$

Сложность операции перестановки снижается в $2^{N+N/2}$ раз.

В основу графового уровня модели положено представление квантового вычисления с помощью диаграмм QuIDD [5,6]. Здесь матрицы преобразуются в графы определённой структуры, а вычислительные операции над ними заменяются операцией порождения графов *Apply(граф 1, граф 2, операция)* [5,6].

Главное преимущество QuIDD-графов перед матрицами состоит в том, что все одинаковые элементы матриц в графе представляются с помощью единственной вершины. Благодаря этому, количество хранимых в памяти числовых значений существенным образом сокращается, и становится возможной значительная экономия вычислительных ресурсов.

Граф QuIDD имеет древовидную структуру. Терминальные вершины имеют только входящие дуги и (рис. 2,а) представляют уникальные числовые значения, содержащиеся в матрице исходного оператора. Промежуточные вершины имеют 1 входящую и 2 исходящих дуги (рис. 2,б, в) и служат для задания местоположения (местоположений) уникальных числовых значений в матрице исходного оператора.

Каждый путь от корня до терминала однозначно определяют местоположение (местоположения) числового значения из терминальной вершины, которой заканчивается путь, в исходной матрице.

Квадратной матрице, у которой все элементы уникальны, соответствует полный QuIDD-граф.

Полный QuIDD-граф (рис. 3) содержит вдвое больше вершин, чем элементов в исходной матрице, но для типичной матрицы квантового оператора, содержащей порядка 2–4 уникальных элементов, граф сокращается.

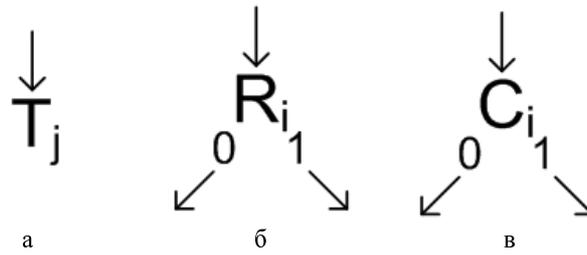


Рис. 2. Типы вершин QuIDD-графа: а – терминальная вершина графа QuIDD, содержащая j-й элемент набора уникальных элементов исходной матрицы; б – промежуточная вершина, задающая i разряд индекса строки матрицы; в – промежуточная вершина, задающая i разряд индекса столбца матрицы

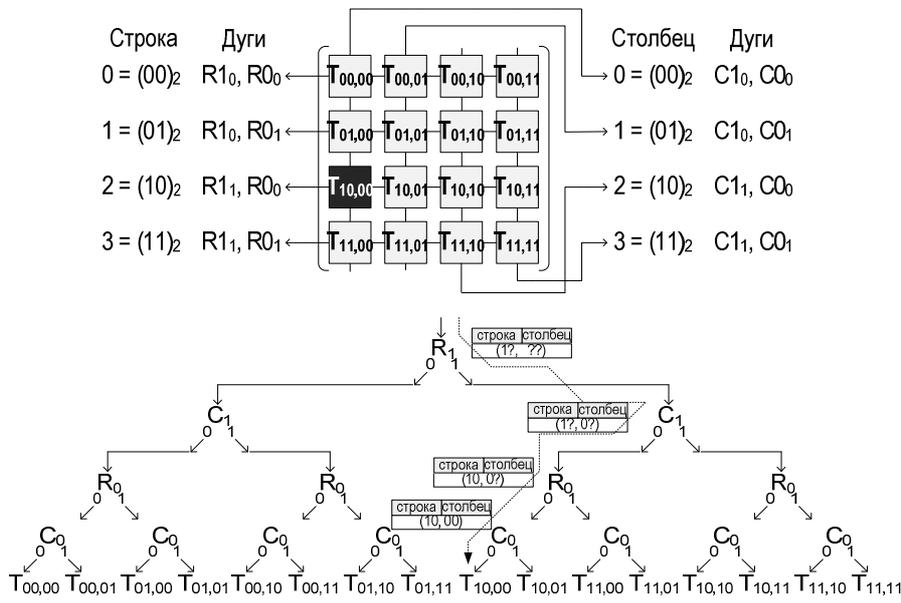


Рис. 3. Общий вид матрицы и графа оператора на 2 q-бита

Путь от корня к любому из терминалов соответствует побитовой «сборке» индексов строки и столбца, задающих местоположение в исходной матрице числа, содержащегося в терминале, находящемся в конце пути. Так для элемента с координатами (строка 10, столбец 00), по графу матрицы нужно пройти путь: $(1?, ??) \rightarrow (1?, 0?) \rightarrow (10, 0?) \rightarrow (10, 00)$ т.е., $R_1^1 \rightarrow C_1^0 \rightarrow R_0^0 \rightarrow C_0^0 \rightarrow T_{10,00}$. Здесь «?» – двоичный разряд индекса, чьё значение ещё не определено, R_1^1 – вершина, соответствующая 1-му разряду индекса строки, R_1^1 – дуга вершины R_1^1 , задающая, что соответствующий вершине двоичный разряд равен единице.

Малое количество уникальных числовых значений в матрицах квантовых операторов, а также наличие правил редукции QuIDD-графов (рис. 4) позволяют существенно сократить количество элементов графа. Например, для QuIDD-представления оператора тождественного преобразования 12 q-битов (единичная матрица 4096×4096) граф будет иметь 38 вершин.

Для редукции QuIDD-графов [5] применяются приведенные ниже правила.

1. Если обе дуги вершины А входят в общую вершину В, то вершина А должна быть заменена на вершину В. Вершина А исключается.
2. Гомоморфные подграфы исключаются.

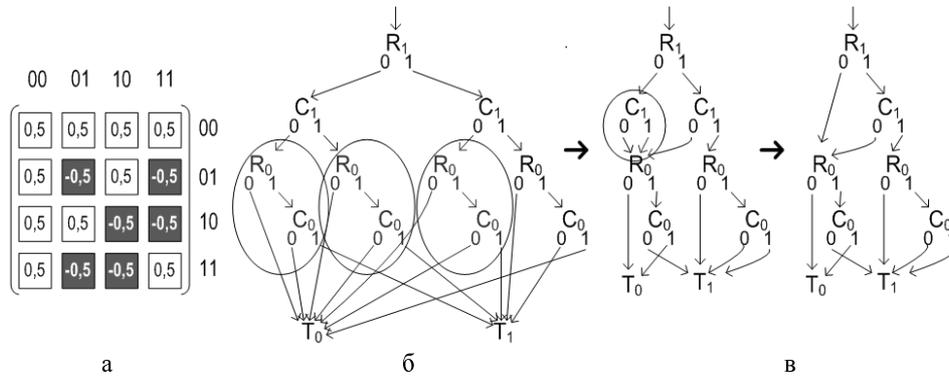


Рис. 4. Редукция QuIDD-графа оператора Уолша-Адамара: а – матрица оператора; б – редукция по правилу 2; в – редукция по правилу 1

Сравнение загрузки памяти для матриц и редуцированных QuIDD-графов (рис. 5) показывает экспоненциальное преимущество QuIDD-графов.

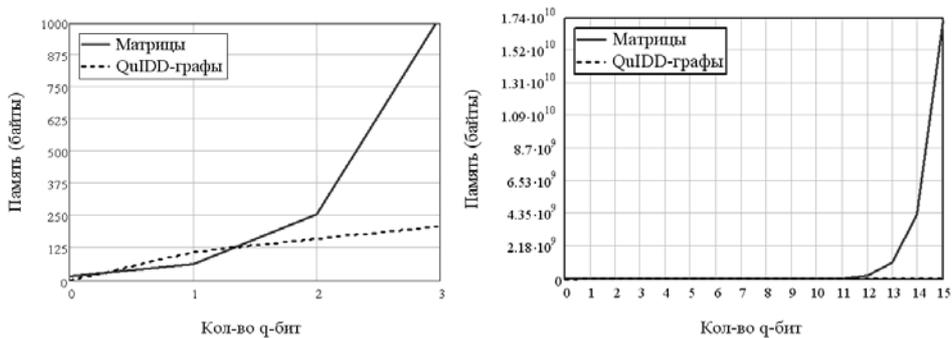


Рис. 5. Потребление памяти уровнями модели

В ходе разработки алгоритма синтеза QuIDD-графов была обнаружена повторяемость структуры графов, начиная с верхнего уровня, заканчивая уровнем терминальных вершин. Данное наблюдение позволило создать метод рекурсивного построения графа посредством поразрядного исключения битов индекса, начиная со старших разрядов.

В соответствии с разработанным методом, на 1-м этапе вся матрица разбивается на 4 подматрицы равного размера. Эти подматрицы условно принимаются за элементы исходной матрицы. Так исходная матрица условно принимает размерность 2×2 , точнее размерность 2×2 матрицы размерностью $N/2$.

Индексы элементов полученных подматриц будут короче на старший (N-1)-ый двоичный разряд. Здесь N – количество разрядов индекса элемента исходной (большой) матрицы, оно же и количество q-бит в квантовом регистре, к которому может быть применён оператор, представленный этой матрицей.

Далее строится QuIDD-граф полученной матрицы 2×2 , количество промежуточных вершин которого составит 2. Однако эти вершины будут пронумерованы не с нуля (R_0 и C_0), как было бы в случае с реальной матрицей 2×2 , а с номера старшего разряда индекса элементов исходной матрицы, т.е., с $N - 1$: R_{N-1} и C_{N-1} .

В терминальные вершины конструируемого графа включаются только уникальные значения элементов исходной матрицы, что позволяет разом отбросить повторяющиеся подматрицы размерностью $2^{N/4} \times 2^{N/4}$, т.е. в четверть исходной.

Далее к построенному графу применяются правила редукции, что позволяет его ещё более упростить.

Переход к следующему шагу рекурсии производится, если условно расположенные в терминальных вершинах графа подматрицы не являются одноэлементными.

На следующем шаге на месте каждой терминальной вершины аналогичным образом строится QuIDD-граф, содержащий вершины R_{N-2} и C_{N-2} .

Итерации будут продолжаться, пока от матрицы, находящейся в каждой терминальной вершине, не останется единственный элемент, являющийся результирующим терминалом графа. В результате, на 1-м шаге строится редуцированный «ярус» графа, соответствующий индексам $N - 1$, на 2-м – $N - 2$, на последнем – 0. Соответственно, на 1-м шаге возможно исключение дублирующихся блоков размерностью в четверть исходной матрицы, на 2-м шаге – размерностью в восьмую часть исходной матрицы и т.д.

Для эффективного моделирования квантовых вычислений мало иметь способ сжатого представления матриц, кроме него требуется эффективный способ их перемножения с двумя частными случаями: умножение матрицы на матрицу и умножение матрицы на вектор состояния. В теории QuIDD-графов эта операция реализуется с помощью процедуры $Apply(A, B, операция)$ [5].

В качестве аргументов она принимает пару графов и матричную операцию над ними. Процедура порождает QuIDD-граф $C = A операция B$, представляющий результат данной операции.

Синтез результирующего графа производится рекурсивно, начиная с корневых вершин графов A и B. На каждом шаге производится сравнение текущей вершины a графа A и b графа B, согласно выражению (4):

$$R_0 < C_0 < R_1 < C_1 < \dots \ll T. \tag{7}$$

Пока обе сравниваемые вершины не являются терминальными, в результирующий граф C добавляется вершина a , дуги которой формируются согласно правилам [6], представленным на рис. 6. Когда обе сравниваемые вершины являются терминальными, в граф C добавляется вершина $c = a операция b$. К результирующему графу C добавляется вершина a .

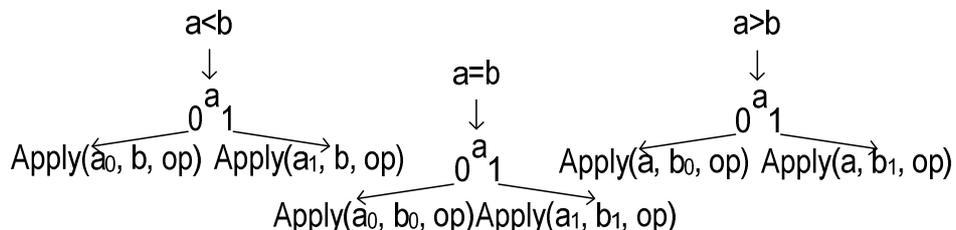


Рис. 6. Правила порождения графа через Apply

Если проанализировать общий случай получающихся таким образом графов, то по их матричным эквивалентам видно, что операция применяется к матрицам поэлементно, что показывает формула 8.

$$Apply\left(\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}, \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix}, " \cdot " \right) = \begin{pmatrix} a_{00} \cdot b_{00} & a_{01} \cdot b_{01} \\ a_{10} \cdot b_{10} & a_{11} \cdot b_{11} \end{pmatrix}. \quad (8)$$

Для реализации перемножения матриц потребуются серии последовательных вызовов $Apply(A, B, " \cdot ")$ и перестановок терминальных вершин графа, результаты которых должны быть просуммированы посредством вызовов $Apply(A, B, " + ")$. Здесь « \cdot » и « $+$ » соответственно означают перемножение и суммирование матриц. Если количество вершин графа велико, выигрыш производительности по сравнению с матричным аппаратом может оказаться не столь существенным, как выигрыш в экономии памяти, что признаётся в работе [7].

Рассмотренные выше методы работы с QuIDD-графами можно сформулировать в виде псевдокода алгоритмов.

Ниже представлен алгоритм синтеза QuIDD-графа для вектора состояния квантового регистра из N q -бит (входные данные: вектор комплексных чисел из 2^N элементов).

1. Последовательный перебор элементов вектора с номерами $i = 0 \dots 2^N - 1$.
 - 1.1. Проверить, находится ли i -е числовое значение из вектора состояния среди терминальных вершин графа. Да – запомнить эту терминальную вершину, нет – добавить терминальную вершину с таким значением.
 - 1.2. Преобразовать i в двоичный код.
 - 1.3. Пройти по графу в соответствии с двоичным кодом, добавляя недостающие вершины (Например, для $i = 12 = (1100)_2$: $R_3^1 \rightarrow R_2^1 \rightarrow R_1^0 \rightarrow R_0^0 \rightarrow T$).
 - 1.4. Добавить дугу из последней вершины добавленного пути к терминальной вершине из шага 1.1.
2. Исключить вершины, обе выходные дуги которых сходятся в одной и той же вершине следующего уровня (правило 2).
3. Исключить гомоморфные подграфы (правило 1).

Ниже представлен алгоритм синтеза QuIDD-графа для матрицы оператора, действующего на регистр из N q -бит (входные данные: условно-терминальная вершина с матрицей комплексных чисел $2^N \times 2^N$ и дуга, входящая в условно-терминальную вершину).

1. Если размерность матрицы 1×1 – конец рекурсии; иначе – пункт 2.
2. Выделить старший бит в числах, определяющих размерности матрицы. Разбить матрицу на 4 подматрицы, в соответствии с его значениями:
 - ◆ включить в 0 подматрицу элементы с $M_{00\dots 0, 00\dots 0}$ по $M_{01\dots 1, 01\dots 1}$;
 - ◆ включить в 1 подматрицу элементы с $M_{00\dots 0, 10\dots 0}$ по $M_{01\dots 1, 11\dots 1}$;
 - ◆ включить во 2 подматрицу элементы с $M_{10\dots 0, 00\dots 0}$ по $M_{11\dots 1, 01\dots 1}$;
 - ◆ включить в 3 подматрицу элементы с $M_{10\dots 0, 10\dots 0}$ по $M_{11\dots 1, 11\dots 1}$.

3. Сформировать полный граф для матрицы 2×2 (рис. 8,б)), чьи терминальные вершины – неповторяющиеся подматрицы 2.1 – 2.4).
4. Редукция графа по правилам 1 и 2.
5. Соединить корень полученного графа с дугой из набора входных данных. Удалить исходную матрицу.
6. Для всех терминальных вершин полученного графа – рекурсивный вызов алгоритма.

Алгоритм получения QuIDD-графа, равного тензорному произведению операторов, представленных QuIDD-графами (входные данные: 2 QuIDD-графа), необходим для работы модели в режиме, когда матрицы требуются только для построения графов базовых операторов размерностью в несколько q-битов; графы всех операторов, реализующих шаги вычисления, строятся из базовых операторов. Такой подход приводит к значительной экономии памяти. Сам алгоритм, воспроизводящий тензорное произведение графов [5], представлен далее.

1. К индексу всех нетерминальных вершин графа множимо прибавить индекс корневой вершины графа множителя, увеличенный на 1.
2. Последовательный обход всех терминальных вершин графа множимо:
 - 2.1. Вершина содержит значение «0» – вершина остаётся без изменений.
 - 2.2. Иначе – заменить вершину графом множителя; числа в терминальных вершинах графа множителя умножить на число из замещённой вершины графа множимо.

Следующий рекурсивный алгоритм поэлементного применения бинарной операции к матрицам, представленным QuIDD-графами (входные данные: операция, 2 вершины QuIDD-графов операндов дуга, к которой должна добавляться вершина результата), может использоваться для организации перемножения матриц и векторов, если добавить к нему переупорядочение элементов операндов.

1. Обе вершины операндов являются терминальными – выполнение операции над их числовыми значениями. Текущая ветвь графа построена – конец рекурсии.
2. Сравнение пары вершин по правилу из формулы 7.
3. Создание вершины результата по правилу из рис. 6, соединение с дугой.
4. Выбор 2-х пар вершин для следующего шага построения результирующего графа – по тому же правилу.
5. Рекурсивный вызов алгоритма для каждой пары вершин и соответствующей дуги вершины, добавленной в пункте 3.

Недостатком базовой реализации QuIDD Apply является возможность применения операции умножения только совместно с переупорядочением вершин графа. Фактически, это означает, что нужно держать в памяти несколько графов одной и той же матрицы с разным порядком следования терминальных вершин, а также промежуточные графы, представляющие частичные суммы для формирования элементов результирующей матрицы. Данный факт существенно повышает затраты времени оперативной памяти на работу модели.

Последующая научная работа в этой области будет направлена на разработку алгоритмов специализированного перемножения матриц, не использующего Apply, что позволит повысить эффективность моделирования, а, как следствие – и максимальный объём моделируемого квантового регистра. Кроме того, возможна дальнейшая оптимизация алгоритмов построения и редукции QuIDD-графов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Netanel H.L, Terry R. A photonic cluster state machine gun // Cornell University Library URL: <http://arxiv.org/abs/0810.2587> (дата обращения 28.03.2011).

2. *D-Wave* // The quantum computer company URL: <http://www.dwavesys.com/> (дата обращения 28.03.2011).
3. *Гузик В.Ф., Гушанский С.М., Евсеев О.К.* Использование параллелизма квантовых вычислений // Материалы Международной научно-технической конференции «Суперкомпьютерные технологии: разработка, программирование, применение». – Таганрог: Изд-во ТТИ ЮФУ, 2010. – Т. 1. – С. 209-212.
4. *Гузик В.Ф., Гушанский С.М., Евсеев О.К.* Реализация модели квантового вычислителя // Материалы X научно-практического семинара г. Донецк 4-7 мая 2009 г. – Донецк: Изд-во ДонНТУ, 2009. – Т. 1. – С. 224-234.
5. *Viamontes G.F., Markov I.L., Hayes J.P.* Graph-based simulation of quantum computation in the density matrix representation // *Quantum Information Processing*, Springer Netherlands ISSN 1570-0755. – 2003.
6. *Viamontes G.F.* Efficient Quantum Circuit Simulation // A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy - Computer Science and Engineering – 230 p. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.3074&rep=rep1&type=pdf> (дата обращения 17.03.2011).
7. *Bahar R.I., Frohm E.A., Gaona C.M.* Algebraic decision diagrams and their applications // ICCAD '93, Santa Clara, CA, USA – November 07-11, 1993 IEEE Computer Society Press. – Los Alamitos, CA, USA, 1993. – P. 188-191.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

Гузик Вячеслав Филиппович

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: gvf@tsure.ru.

347900, г. Таганрог ул. Чехова, 49 кв. 40.

Тел.: 88634371656.

Заведующий кафедрой вычислительной техники; д.т.н.; профессор.

Гушанский Сергей Михайлович

E-mail: kron@pbox.ttn.ru.

347900, г. Таганрог ул. Морозова, 11, кв. 27.

К.т.н.; доцент.

Евсеев Олег Константинович

E-mail: Aleg.a33@rambler.ru.

347900, г. Таганрог ул. Р. Люксембург, 240, кв. 38.

Тел.: +79281721487.

Аспирант.

Guzik Vyacheslav Filippovich

Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: gvf@tsure.ru.

49, Chekhov Street, Apt. 40, Taganrog, 347900, Russia.

Phone: +78634371656.

Head of the Department of Computer Engineering; Dr. of Eng. Sc.; Professor.

Gushansky Sergei Mikhailovich

E-mail: kron@pbox.ttn.ru.

11, Morozov Street, Apt. 27, Taganrog, 347900, Russia.

Cand. of Eng. Sc.; Associate Professor.

Evsseev Oleg Konstantinovich

E-mail: Aleg.a33@rambler.ru.

240, R. Luxemburg Street, Apt., 38, Taganrog, 347900, Russia.

Phone: +79281721487.

Postgraduate Student.