П. В. Фокин

ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ БУЛЕВЫХ МНОГОЧЛЕНОВ В ВИДЕ ZDD-ДИАГРАММ

В статье предлагается альтернативная точка зрения на использование ZDD-диаграмм, заключающаяся в реализации операций сложения и умножения на переменную, а не в операциях, связанных объединением и пересечением множеств булевых мономов.

Бинарные диаграммы решений (Binary Decision Diagram, BDD) являются удобным инструментом представления и оперирования булевыми функциями и широко используются в различных областях, например для формальной верификации программных и аппаратных систем.

Определение 1. *Бинарные диаграммы решений (BDD) – направленный ациклический граф с двумя терминальными узлами $\{0, 1\}$, которые соответствуют значениям представляемой булевой функции. Выходная степень терминальных вершин равна 0. Все остальные вершины имеют выходную степень 2 и называются узлами решений. Одна вершина имеет входную степень, равную 0, эта вершина является корнем. Ребра, выходящие из узлов решений (high/low или then/else), соответствуют значению 0(ребро else) или 1(ребро then) для соответствующей переменной.*

Последовательность узлов, начинающихся с вершины и заканчивающаяся терминальным узлом, называется путём. В случае, когда порядок переменных для всех путей остаётся постоянным, такую диаграмму будем называть упорядоченной (OBDD). Упорядоченная BDD называется сокращенной или редуцированной (ROBDD), если она не содержит повторяющихся фрагментов.

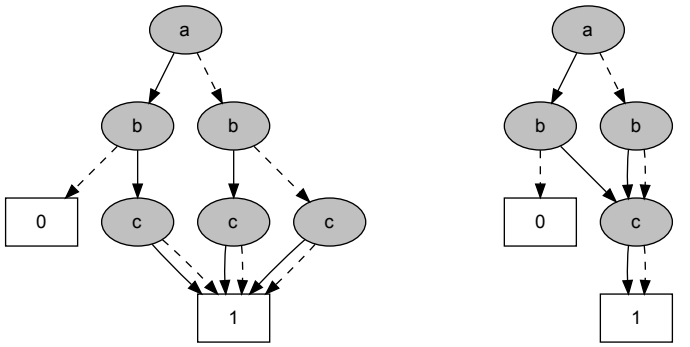
Определение 2. *Пусть z будет упорядоченной бинарной диаграммой решений и не содержит одинаковых поддиаграмм. Тогда z будет называться zero-suppressed binary decision diagram (ZDD), если из неё исключены те узлы, then-ребра которых заканчиваются в терминальной вершине, соответствующей 0.*

ZDD были введены Shin-ichi Minato в 1993 [1]. В книге Кнута [2] достаточно подробно рассмотрены многие свойства таких диаграмм.

Рассмотрим в качестве примера рекурсивное представление для многочлена $p = abc + ab + bc + b + c + 1$ с порядком переменных $a \succ b \succ c$

$$p = a(b(c + 1)) + b(c + 1) + c + 1\,. \qquad (1)$$

Графически его можно представить в виде диаграммы (рис. 1, *а*), где непрерывная линия соответствует умножению, пунктирная — сложению, а в узлах хранятся переменные или 1 и 0.



а *б*

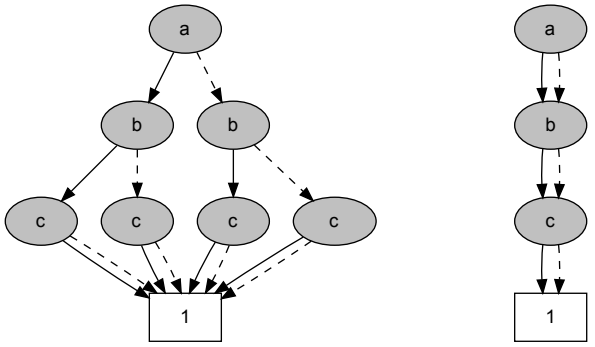
Рис. 1

В формуле (1) можно заметить повторы, используя которые, многочлен можно переписать следующим образом:

$$p = ay + y + x, \quad \text{где} \quad y = bx,\ x = c + 1\,. \tag{2}$$

На рис. 1, *б* диаграмма соответствует ZDD.

На рис. 2, *а* диаграмма соответствует обычному представлению, а на рис. 2, *б* ZDD, для многочлена $(a+1)(b+1)(c+1)$, который содержит все возможные булевские мономы для трех переменных. Из диаграммы на рис. 2, *б* видно, что для представления многочлена от всех возможных булевских мономов от n переменных всего n узлов.



а *б*

Рис. 2

Пакет **PolyBoRi** [3, 4] применяет это представление для многочленов при построении базисов Грёбнера, используя внешнюю библиотеку **CUDD** [5], которая характеризуется следующими свойствами:

• библиотека **CUDD** создана для работы с разными типами представлений BDD, ADD и ZDD[5];

• в **CUDD**, а значит, и в **PolyBori** используется общий кэш для всех поддеревьев всех используемых многочленов.

Использование сторонней библиотеки **CUDD** не очень эффективно. Нами построена реализация операций сложения и умножения многочленов, а также умножение на переменную на языке Python. Ее главные отличия от библиотеки **CUDD**:

• кэш является локальным для каждого многочлена, что позволяет сократить время поиска возможного поддерева используя номер переменной, а также эффективно реализовать сборку мусора;

• встроена специализированная операция умножение на переменную;

• для оптимизации поиска старшего монома для упорядочений по полной степени в узлах предусмотрено хранение наибольшей полной степени монома, содержащегося в этом поддереве.

Выполнена экспериментальныя оценка использования памяти для некоторых многочленов, встречающихся при работе с HFE(Hidden Fields Equations)[6]. Для проверенных многочленов с числом переменных, не превышающих 80, структуры данных ZDD показали более устйчивое использование памяти, чем с использованием списков или рекурсивного представления, особенно при умножении на переменную.

В настоящие время разработана реализация ZDD представления многочленов на C++ для встраивания в пакет построения булевских базисов Грёбнера [7].

Работа выполнена при финансовой поддержке РФФИ (проект 10-01-00200-а) и гранта Президента РФ(проект НШ-3810.2010.2).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Shin-ichi Minato.J.* Zero-suppressed bdds for set manipulation in combinatorial problems // Proc. of the 30th international Design Automation Conference, DAC '93. New York, USA, 1993. ACM. P. 272–277.

2. *Knuth D.E.* Addison-Wesley Professional, // The Art of Computer Programming. Vol. 4, Fascicle 1: Bitwise Tricks & Techniques; Binary Decision Diagrams/ 12th ed., March. 2009.

3. *Brickenstein M., Dreyer A.* PolyBori: A framework for gröbner-basis computations with boolean polynomials // Journal of Symbolic Computation, Effective Methods in Algebraic Geometry. 2009. Vol. 44, № 9. P. 1326–1345.

4. *Brickenstein M., Dreyer A., Greuel G.-M., Wedler M., Wienand O.* New developments in the theory of gröbner bases and applications to formal verification // Journal of Pure and Applied Algebra. 2009. Vol. 213, № 8. P. 1612–1635. Theoretical Effectivity and Practical Effectivity of Gröbner Bases.

5. *Somenzi F.* URL: http://vlsi.colorado.edu/fabio/ (дата обращения : 25.05.2011) Cudd: Cu decision diagram package release.

6. *Fokin P. V., Blinkov Yu. A..* ZDD diagrams as appropriate data structures in construction of Boolean Gröbner bases by involutive algorithms // Polynomial Computer Algebra. 2011. P. 22–24 .

7. *Gerdt V. P., Zinin M. V., Blinkov Yu. A.* On computation of boolean involutive bases // Program. Comput. Softw. March. 2010.

УДК 519.53, 519.713

Е. В. Хворостухина

ОБ ЭЛЕМЕНТАРНЫХ СВОЙСТВАХ УНИВЕРСАЛЬНЫХ ГИПЕРГРАФИЧЕСКИХ АВТОМАТОВ

В настоящей статье рассматриваются гиперграфические автоматы без выходных сигналов, т.е. автоматы, у которых множества состояний наделены дополнительной алгебраической структурой гиперграфа.

Исследуются взаимосвязи элементарных свойств универсальных гиперграфических автоматов с элементарными свойствами полугрупп их входных сигналов.

Следуя [1], *гиперграфом* называется система вида $H = (X, L)$, где X – непустое множество вершин гиперграфа и L – семейство некоторых подмножеств множества X, называемых *ребрами гиперграфа.* Вершины гиперграфа, принадлежащие некоторому его ребру, называются *смежными.*

Гиперграф $H = (X, L)$ называется *эффективным*, если любая его вершина принадлежит некоторому его ребру.

Пусть p – некоторое натуральное число. Гиперграф H будем называть *гиперграфом с p-определимыми ребрами*, если в каждом его ребре гиперграфа найдется по крайней мере $p+1$ вершина и, с другой стороны, любые p вершин этого гиперграфа содержатся не более чем в одном ребре. То есть в таком гиперграфе каждое ребро однозначно определяется любыми своими p вершинами.

Например, эффективный гиперграф с 1-определимыми ребрами – это гиперграф, ребра которого образуют нетривиальное разбиение множества вершин без одноэлементных классов. Кроме того, если рассмотреть плоскость как гиперграф, вершинами которого являются точки этих