

УДК 004.384:004.272:004.414.2

**ВЫЯВЛЕНИЕ ИЗОМОРФНЫХ ВХОЖДЕНИЙ R-ВЫРАЖЕНИЙ
ПРИ ПОСТРОЕНИИ МНОЖЕСТВА СЕЧЕНИЙ ПАРАЛЛЕЛЬНЫХ
АЛГОРИТМОВ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ**

Э.И. Ватутин, И.В. Зотов, В.С. Титов

В статье предложен метод выявления изоморфных вхождений R-выражений при построении множества сечений в ходе декомпозиции параллельных алгоритмов логического управления. Метод обеспечивает проверку изоморфизма за время порядка $O(n^3)$ и базируется на сформулированных в статье требованиях корректности R-выражений, а также ряде специфичных для них особенностей, лемм и теорем.

**CHECKING UP ISOMORPHIC INCLUSIONS OF R-EXPRESSIONS
IN THE CONSTRUCTION OF A SET OF SECTIONS FOR PARALLEL
LOGIC CONTROL ALGORITHMS**

E.I. Vatutin, I.V. Zotov, V.S. Titov

Введение

Реализация параллельных управляющих алгоритмов в базисе логических мультиконтроллеров (ЛМК) требует их декомпозиции на множество частных алгоритмов ограниченной сложности [1]. Получение оптимального набора частных алгоритмов (разбиения) представляет собой сложную комбинаторную задачу, качество решения которой напрямую влияет на аппаратную сложность ЛМК и определяет, в конечном счете, оперативность управления. Один из наиболее эффективных путей ее решения дает развиваемый авторами параллельно-последовательный метод декомпозиции [2-7]. Как показано в [8], он позволяет формировать наиболее близкие к оптимальным разбиения с учетом основных структурных и технологических ограничений базиса ЛМК.

Один из важнейших этапов параллельно-последовательной декомпозиции – построение множества сечений, покрывающего все вершины исходного алгоритма. Формирование сечений осуществляется путем выполнения трудоемких операций подстановки над множеством описывающих алгоритм R-выражений. Как показывают исследования,

упрощение и ускорение этих операций возможно путем их сведения к действиям над деревьями, в частности, к проверке изоморфизма.

В данной статье предложен метод определения изоморфных вхождений R -выражений. Метод основан на ряде специфических свойств R -выражений, не присущих графам или деревьям общего вида, что обеспечивает проверку изоморфизма (либо его отсутствия) за полиномиальное время. (Как известно [9], для решения задачи распознавания изоморфизма графов общего вида до сих пор не только не придумано эффективного универсального алгоритма, но и не доказана ее принадлежность к классу P или NP [10].)

Постановка задачи

Размещаемый в ЛМК алгоритм логического управления задается в виде системы R -выражений Ξ , получаемой, например, по его граф-схеме [1,3-5]. Каждое выражение системы описывает непосредственный переход между определенными множествами вершин алгоритма. С использованием системы R -выражений Ξ сперва производится нахождение базового сечения алгоритма Ω_{\max} , а затем – построение множества смежных сечений $\{\Omega_1, \Omega_2, \dots, \Omega_{\max}, \dots, \Omega_M\}$, покрывающих множество вершин алгоритма управления и используемых для организации сокращенного перебора вершин и формирования частных алгоритмов [1-5]. На рис. 1 приведен фрагмент граф-схемы алгоритма управления и показано его покрытие множеством смежных сечений $\{\Omega_{i-1}^u, \Omega_i, \Omega_{i+1}^d\}$.

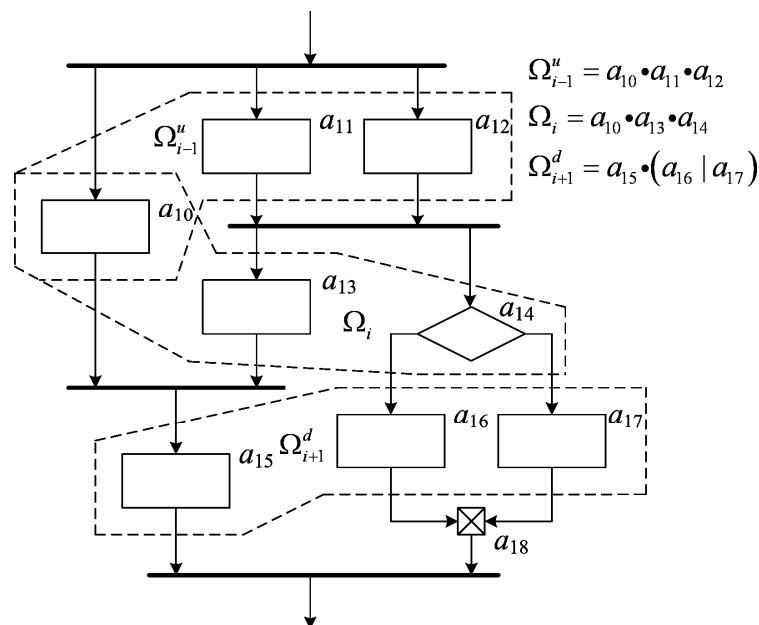


Рис. 1. Фрагмент параллельного алгоритма управления и его покрытие множеством смежных сечений

Сечение включает только попарно параллельные и/или альтернативные вершины алгоритма и максимально по включению [1]. В общем случае оно описывается рекуррентным определением

$$R = (a_i) \vee (R_{j_1} \bullet R_{j_2} \bullet \dots \bullet R_{j_k}) \vee (R_{k_1} | R_{k_2} | \dots | R_{k_p}). \quad (1)$$

Другими словами, оно может быть представлено вершиной алгоритма либо совокупностью конструктивных подмножеств R_{j_h} , R_{k_g} , объединенных отношением параллельности « \bullet » или альтернативы « $|$ ».

Каждое выражение S_i системы Ξ представляется в виде $R_1^i \rightarrow R_2^i$, где R_1^i и R_2^i – конструктивные подмножества (субсечения) сечений алгоритма. Для фрагмента алгоритма, изображенного на рис. 1, система выражений Ξ выглядит следующим образом:

$$\left\{ \begin{array}{l} x \rightarrow a_{10} \bullet a_{11} \bullet a_{12}; \\ a_{10} \bullet a_{11} \bullet a_{12} \rightarrow a_{10} \bullet a_{13} \bullet a_{14}; \\ a_{14} \rightarrow a_{16} | a_{17}; \\ a_{10} \bullet a_{13} \rightarrow a_{15}; \\ a_{16} | a_{17} \rightarrow a_{18}; \\ a_{15} \bullet a_{18} \rightarrow y. \end{array} \right. \quad (2)$$

Здесь x и y – не изображенные на рис. 1 множества вершин, располагающиеся соответственно до и после приведенного фрагмента алгоритма.

При практической реализации операций над R -выражениями удобной формой описания, отражающей конструктивный характер и иерархию вложенности конструктивных подмножеств, является дерево, которое, в свою очередь, может быть преобразовано в табличный вид [3]. В подобном представлении R -выражения листья дерева соответствуют вершинам алгоритма, а узлы определяют тип отношения между вершинами [1]. Каждый элемент дерева представляется набором полей, характеристики которых сведены в табл. 1.

Табл. 1. Описание полей узлов в табличном представлении дерева

Поле	Присутствие в дереве		Размер, бит	Назначение
	в узлах	в наборах листьев		
Тип узла (ТУ)	+	–	1	Кодирование параллельного («•») или альтернативного («») узла дерева
Множество вершин (МВ)	–	+	L_{\max}	Множество вершин (двоичный вектор), присутствующих в наборе листьев
Ссылка на предка (СП)	+	+	$\lceil \log_2 N_{T_{\max}} \rceil$	Номер узла-предка в двоичном коде
Тип соответствия (ТС)	+	+	2	Полное («11»), частичное («10») или отсутствующее («0*») соответствие изоморфному эквиваленту для элемента дерева
Номер соответствия (НС)	+	+	$\lceil \log_2 N_{T_{\max}} \rceil$	Номер изоморфного эквивалента элемента дерева

Здесь $N_{T_{\max}}$ – максимальное количество узлов в дереве, L_{\max} – максимальное количество листьев в наборе (фактически, число вершин в алгоритме управления).

Алгоритмы обработки R -выражений, оперирующие их табличным представлением в виде совокупности полей ТУ, МВ, СП, позволяет обойтись лишь одним направлением обхода дерева (снизу вверх, т.е. от листьев к корню), что позволяет исключить из рассмотрения поле ссылок на узлы-потомки (поля ТС и НС используются только в рассматриваемой операции подстановки и не используются другими алгоритмами обработки R -выражений [3, 11]). В то же время при программной реализации [6, 7] используются оба направления (обход дерева сверху вниз, т.е. от корня к листьям, применяется, например, при рекуррентном сравнении деревьев). Все узлы дерева разделяются на две группы, хранимые отдельно и кодируемые различными подмножествами полей. В *первую группу* входят параллельные и альтернативные узлы (далее просто узлы), во *вторую* – наборы листьев дерева. Подобное разделение, с одной стороны, обусловлено спецификой операций, выполняемых над узлами различных типов, а с другой, как уже было отмечено выше, позволяет обойтись без поля с множеством ссылок на потомков узлов. Это, в свою очередь, ведет к существенной экономии используемой памяти (поле со ссылками на потомков имеет значительно больший размер (N_R бит) по сравнению с остальными полями элементов дерева). Совместное хранение листьев дерева в виде наборов листьев (битовых векторов) обеспечивает уменьшение затрат памяти (хранится весь набор, а не его компоненты по отдель-

ности) за счет уменьшения дублирования полей (например, СП). С другой стороны повышается быстродействие алгоритмов обработки, так как компоненты битовых векторов обрабатываются параллельно во времени, а отдельные элементы дерева – последовательно.

Совокупность полей N_T узлов, где N_T – текущее количество узлов в дереве, и N_L наборов листьев, где N_L – текущее количество наборов листьев, образуют табличное представление дерева. На рис. 2 дан пример R -выражения, показано его представление в виде дерева и таблицы.

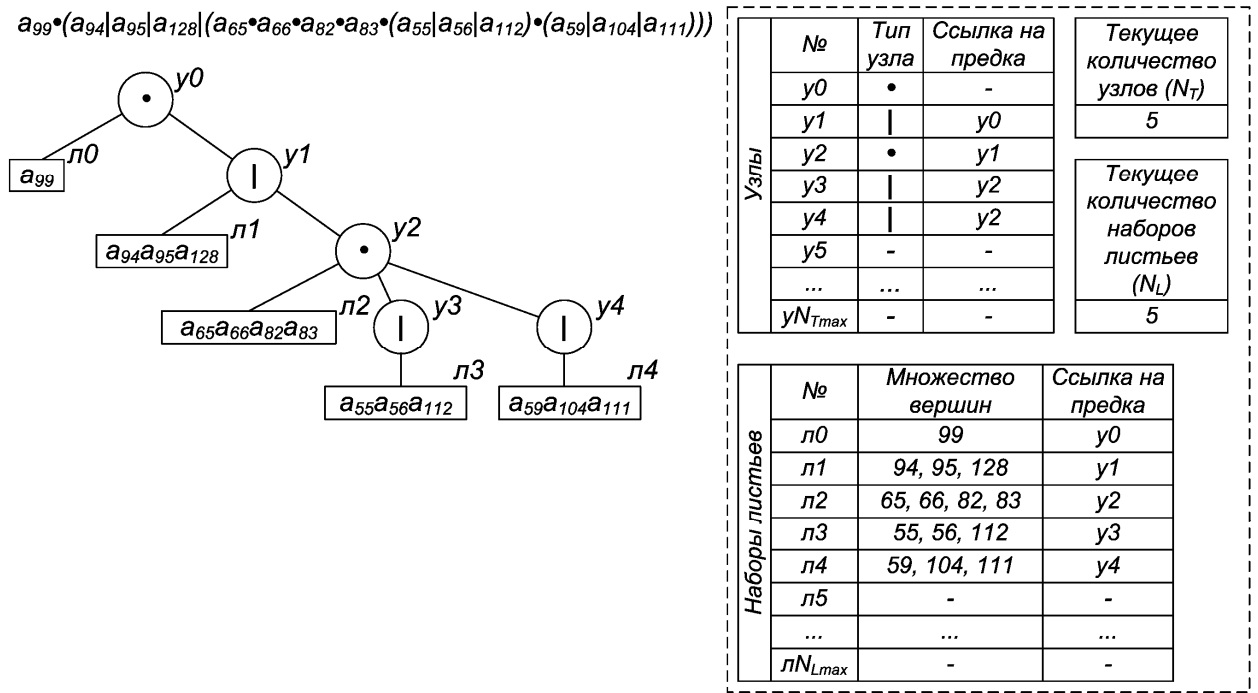


Рис. 2. Пример R -выражения, соответствующее ему дерево и его табличное представление

Процесс преобразования системы выражений Ξ сводится к последовательному применению операций u - и d -подстановок [1,3-5]. Возможные типы подстановок [3, 4], возникающие в процессе преобразования, приведены ниже.

$$\begin{cases} a_1 \rightarrow a_2 \\ a_2 \rightarrow a_3 \end{cases} \Rightarrow a_1 \rightarrow a_3, \quad (3)$$

$$\begin{cases} a_1 \rightarrow a_2 \\ a_2 \rightarrow a_3 \bullet a_4 \end{cases} \Rightarrow a_1 \rightarrow a_3 \bullet a_4, \quad (4)$$

$$\begin{cases} a_1 \rightarrow a_2 \bullet a_3 \\ a_2 \bullet a_3 \rightarrow a_4 | a_5 \end{cases} \Rightarrow a_1 \rightarrow a_4 | a_5, \quad (5)$$

$$\begin{cases} a_1 \rightarrow a_2 | a_3 \\ a_2 \rightarrow a_4 \end{cases} \Rightarrow a_1 \rightarrow a_4 | a_3, \quad (6)$$

$$\begin{cases} a_1 \rightarrow a_2 | a_3 \\ a_2 \rightarrow a_4 \cdot a_5 \end{cases} \Rightarrow a_1 \rightarrow (a_4 \cdot a_5) | a_3, \quad (7)$$

$$\begin{cases} a_0 \rightarrow a_1 | a_2 | (a_3 \cdot a_4) \\ a_2 | (a_3 \cdot a_4) \rightarrow a_5 \cdot a_6 \cdot a_7 \end{cases} \Rightarrow a_0 \rightarrow a_1 | (a_5 \cdot a_6 \cdot a_7), \quad (8)$$

$$\begin{cases} a_1 \rightarrow a_2 \cdot a_3 \\ a_2 \cdot a_3 \rightarrow a_4 \end{cases} \Rightarrow a_1 \rightarrow a_4, \quad (9)$$

$$\begin{cases} a_1 \rightarrow a_2 | a_3 | a_4 \\ a_2 | a_3 \rightarrow a_5 \end{cases} \Rightarrow a_1 \rightarrow a_4 | a_5. \quad (10)$$

Принятое разделение на типы подстановок объясняется с позиции того, что (лист или дерево) ищется в качестве изоморфного вхождения, какой тип соответствия (полное или частичное) при этом наблюдается и что (лист или дерево) подставляется взамен удаляемых элементов. Все остальные подстановки, возникающие в процессе преобразования системы выражений Ξ , сводятся к одному из представленных выше типов.

Одним из действий, выполняемым в процессе подстановки, является проверка того, что R -выражение A является конструктивным подмножеством R -выражения B (определяемым через отношение нестрогого включения $A[\subseteq]B$ [1]), т.е. подмножеством элементов R -выражения, заключенным в единые скобки. Например, если $A_1 = a_3 | a_6 | (a_4 \cdot a_5)$, $A_2 = a_1 \cdot a_2 \cdot a_7$ и $B = a_1 \cdot a_2 \cdot (a_3 | (a_4 \cdot a_5) | a_6)$, то $A_1[\subseteq]B$, а $A_2[\not\subseteq]B$.

С учетом принятого представления R -выражений в виде деревьев задача выяснения отношения нестрогого включения сводится к задаче поиска изоморфного вхождения подграфа в граф [9]. Однако существует ряд отличий, одно из которых продемонстрировано на рис. 3.

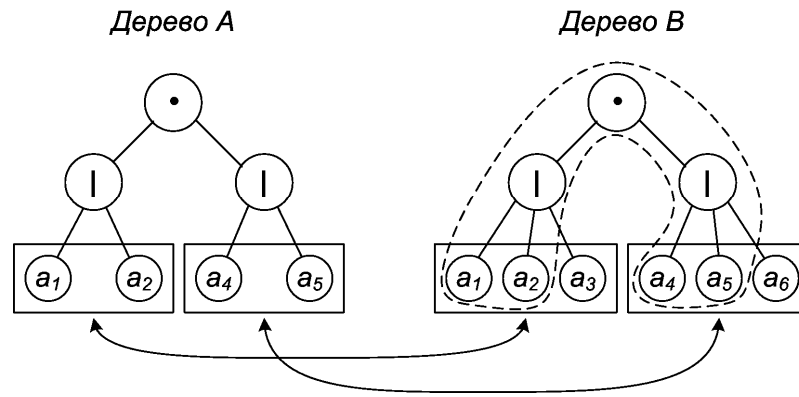


Рис. 3. Иллюстрация, поясняющая различия в «классическом» и r -изоморфизме (стрелками показано отношение неполной эквивалентности между наборами листьев, изоморфное поддерево в составе дерева B обозначено штриховкой, однако $A[\not\subseteq]B$)

На рисунке представлен пример пары деревьев, соответствующих R -выражениям, причем в составе дерева B есть поддерево, изоморфное дереву A . Однако при этом выполнить ни одну из подстановок (3)-(10) не представляется возможным, то есть $A[\not\subseteq]B$ (подобная ситуация идентифицируется согласно приведенному ниже условию 2). Рассмотренный пример приведен исключительно с целью иллюстрации различий в классической постановке задачи изоморфизма и изоморфизме R -выражений (именуемом здесь и далее r -изоморфизмом), при обработке реальных алгоритмов управления указанная на рис. 3 ситуация не может возникнуть, т.к. на самом деле дерево A соответствует некорректному сечению, которое не максимально по включению [1].

Особые свойства R -выражений

R -выражения, синтезированные для корректного алгоритма управления, обладают рядом особых свойств, позволяющих искать единственно возможное r -изоморфное вхождение (в случае его наличия) за полиномиальное время. Прежде чем приступить к рассмотрению особых свойств R -выражений, необходимо сформулировать требования к их корректности. R -выражение будем считать корректно заданным в табличной форме в том случае, если соблюдаются следующие условия:

1. Корень дерева хранится в позиции с номером 0. Значение поля СП корня равно «11...1», что при попытке обращения к нему обеспечивает обращение в несуществующему элементу дерева и выдачу нулевых значений «00...0» полей.
2. Для каждого узла дерева его потомки хранятся в позициях с номерами, превосходящими номер позиции самого узла (при этом порядок хранения потомков не важен).

3. Все узлы и наборы листьев дерева хранятся в смежных позициях (без «пропусков»), причем значение текущего количества узлов указывает на номер позиции последнего узла, а значение текущего количества листьев – на номер позиции последнего листа.

4. Каждому узлу дерева соответствует не более одного дочернего набора листьев.

5. Если дерево представлено единственным набором листьев (без узлов), то в составе этого набора может быть всего один элемент (одна вершина алгоритма управления).

6. В составе корректного дерева не может быть совпадения типа узла у любых предка и потомка. В случае наличия подобной пары узлов требуется выполнение операции раскрытия скобок $a_1 | (a_2 \bullet (a_3 \bullet a_4 \bullet a_5)) \Rightarrow a_1 | (a_2 \bullet a_3 \bullet a_4 \bullet a_5)$ [3], после выполнения которой дерево считается корректным.

Требование 1 защищает от возможных ложных срабатываний в ситуации, когда изоморфный эквивалент рассматриваемого узла является корнем дерева и у него по определению отсутствует предок. Требование 2 гарантирует, что при рассмотрении узлов дерева в его табличном представлении в порядке от узлов с большим номером позиции к узлам с меньшим номером будет обеспечен обход дерева по ярусам в направлении снизу-вверх. Требование 3, с одной стороны, гарантирует экономное использование позиций табличного представления дерева (в операциях подстановок максимально эффективно используется свободная часть таблицы), количество которых ограничено значениями $N_{T_{\max}}$ для узлов и $N_{L_{\max}}$ для наборов листьев ($N_{L_{\max}}$ – максимальное количество наборов листьев в дереве), а с другой – защищает от ложных срабатываний при попытке обращения к неиспользуемым (пустым) позициям, которые в общем случае могут содержать произвольные двоичные значения. Требование 4 обеспечивает группировку листьев дерева с общим предком в один набор, что упрощает алгоритмы их обработки и повышает быстродействие. Требование 5 продиктовано невозможностью присутствия в наборе двух и более вершин без установления отношения параллельности или альтернативы [1] между ними (отношение между вершинами в таком случае определяется типом узла – предка набора листьев). Несоблюдение требования 6 может привести к невозможности нахождения множества сечений для некоторых корректных алгоритмов управления ввиду невозможности выполнять ни одну из подстановок.

Введем понятие полной и частичной эквивалентности наборов листьев. Под полной эквивалентностью наборов листьев L_1 и L_2 будем понимать их полное поэлементное совпадение: $L_1 = L_2$. Под частичной эквивалентностью наборов листьев будем понимать ситуацию, когда $(L_1 \subset L_2) \vee (L_2 \subset L_1)$. При употреблении понятия эквивалентности на-

боров листьев, обозначаемого $L_1[\sim]L_2$, будем подразумевать под ним наличие полной или частичное эквивалентности: $L_1[\sim]L_2 \Leftrightarrow (L_1 = L_2) \vee (L_1 \subset L_2) \vee (L_2 \subset L_1)$, причем $L_1[\sim]L_2 \not\Leftrightarrow L_1 \cap L_2 \neq \emptyset$.

На основании определения понятия r -изоморфизма можно сформулировать первое необходимое условие отсутствия r -изоморфизма у рассматриваемой пары деревьев.

Необходимое условие 1 отсутствия r -изоморфизма. Если в составе дерева A найдется набор листьев L_i^A , не находящийся в отношении эквивалентности ни с одним набором листьев дерева B , то дерево A не является r -изоморфным дереву B : $\exists L_i^A \in A, \forall L_j^B \in B, L_i^A[\neq]L_j^B \rightarrow A[\not\cong]B$. Следует из определения изоморфизма (каждой вершине дерева A должен соответствовать изоморфный эквивалент в виде вершины в составе дерева B).

При анализе приведенных выше подстановок различного типа (3)-(10) можно подметить следующую особенность.

Особенность 1. Если дерево A является r -изоморфным поддеревом дерева B , то в составе наборов листьев деревьев A и B не может быть более одной пары наборов листьев L_i^A и L_j^B с частичным совпадением $L_i^A \subset L_j^B$. При этом набор листьев L_i^A может быть потомком только корня дерева A .

На основании особенности 1 можно сформулировать второе необходимое условие отсутствия r -изоморфизма у рассматриваемой пары деревьев.

Необходимое условие 2 отсутствия r -изоморфизма. Если в составе деревьев A и B присутствует более одной пары наборов листьев с частичным соответствием, рассматриваемые деревья не r -изоморфны.

На основании приведенных выше требования корректности можно сформулировать ряд следствий.

Следствие 1 (из требования корректности № 4). Невозможно найти такие два набора листьев L_i^A и L_j^A , где L_i^A – i -й набор листьев дерева A , предком которых являлся бы один и тот же узел. В противном случае будет нарушено требование корректности № 4.

Следствие 2 (из требования корректности № 6). При движении по узлам любой ветви дерева тип узлов строго чередуется. В противном случае будет нарушено требование корректности № 6.

Сформулируем и докажем также ряд лемм, упрощающих формулировку приведенных ниже теорем.

Лемма 1 (об ортогональности наборов листьев в составе дерева). В пределах дерева все наборы листьев ортогональны: $L_i^A \cap L_j^A = \emptyset, \forall i \neq j$.

Доказательство леммы проведем методом от противного. Допустим, что $\exists a_k : (a_k \in L_i^A) \wedge (a_k \in L_j^A), i \neq j$, где a_k – некоторая вершина алгоритма. Каждый набор листьев фактически представляет собой подмножество вершин различных, не вложенных друг в друга фрагментов алгоритма управления. С учетом принятого допущения получается, что вершина a_k принадлежит одновременно двум различным фрагментам алгоритма, чего в составе корректного алгоритма не может быть. Полученное противоречие доказывает лемму.

Лемма 2 (о совпадении типа узлов-предков для вершин в составе различных наборов листьев). Любые две вершины, входящие в состав одного набора листьев L_i^X дерева X , могут одновременно войти в состав набора листьев L_j^Y дерева Y только в том случае, если тип узла предков наборов листьев L_i^X и L_j^Y совпадает.

Доказательство леммы проведем методом от противного. Допустим что найдутся такие деревья X и Y , в состав которых входят наборы листьев L_i^X и L_j^Y , причем тип узла их предков не совпадает. Без потери общности предположим, что тип узла – предка набора листьев L_i^X равен «•», а предка набора листьев L_j^Y – «|». Т.к. тип узла-предка определяется типом фрагмента (параллельный или альтернативный) алгоритма управления, то для вершин $a_{k_1} \in L_i^X$ и $a_{k_2} \in L_j^Y$ имеет место отношение параллельности [1]: $a_{k_1} \omega a_{k_2}$. С дру-

гой стороны, если $a_{k_1} \in L_j^Y$ и $a_{k_2} \in L_j^Y$, то имеет место отношение альтернативы [1]: $a_{k_1} \Psi a_{k_2}$. То есть рассматриваемая пара вершин одновременно и параллельны, и альтернативны друг другу, чего не может быть ввиду нарушения ограничения на ортогональность бинарных отношений между вершинами, сформулированного в [1]. Полученное противоречие доказывает лемму.

Следствие. У полностью или частично эквивалентных наборов листьев не может быть предков разного типа.

На основании приведенных выше лемм сформулируем и докажем ряд теорем.

Теорема 1 (о единственности r -изоморфной пары наборов листьев). Набору листьев L_i^A может соответствовать не более одного полностью или частично эквивалентного набора листьев $L_j^B: L_i^A [\sim] L_j^B: \exists L_k^B, L_i^A [\sim] L_k^B, k \neq j$.

Доказательство теоремы произведем методом от противного. Предположим возможность существования пары наборов листьев L_j^B и $L_k^B, j \neq k$, таких что $L_i^A [\sim] L_j^B$ и $L_i^A [\sim] L_k^B$. В таком случае найдется хотя бы одна вершина $a_l \in L_i^A$, такая что $a_l \in L_j^B$ и $a_l \in L_k^B$. То есть $a_l \in L_j^B \cap L_k^B \neq \emptyset$, что противоречит лемме 1. Полученное утверждение доказывает теорему.

Теорема 2 (о единственности r -изоморфной пары поддеревьев). В составе деревьев A и B не может быть более одной пары совпадающих поддеревьев.

Доказательство теоремы произведем методом от противного. Предположим, что в составе дерева $A = \langle S_T^A, S_L^A, E^A \rangle$, образованного $S_T^A = \{T_1, T_2, \dots, T_{N_T}\}$ узлами, $S_L^A = \{L_1^A, L_2^A, \dots, L_{N_L}^A\}$ наборами листьев и связями E^A между ними, имеется поддерево $C = \langle S_T^C, S_L^C, E^C \rangle, S_T^C \subseteq S_T^A, S_L^C \subseteq S_L^A, E^C \subseteq E^A$, которому соответствует пара различных изоморфных поддеревьев $\tilde{C}_1 = \langle S_T^{\tilde{C}_1}, S_L^{\tilde{C}_1}, E^{\tilde{C}_1} \rangle$ и $\tilde{C}_2 = \langle S_T^{\tilde{C}_2}, S_L^{\tilde{C}_2}, E^{\tilde{C}_2} \rangle$ в составе дерева $B = \langle S_T^B, S_L^B, E^B \rangle, S_T^{\tilde{C}_1} \subseteq S_T^B, S_T^{\tilde{C}_2} \subseteq S_T^B, S_L^{\tilde{C}_1} \subseteq S_L^B, S_L^{\tilde{C}_2} \subseteq S_L^B, E^{\tilde{C}_1} \subseteq E^B, E^{\tilde{C}_2} \subseteq E^B: C[\sim]\tilde{C}_1$ и $C[\sim]\tilde{C}_2, \tilde{C}_1 \neq \tilde{C}_2$. Это означает, что в составе наборов листьев

поддерева C найдется хотя бы один набор листьев L_i^C , такой что $L_i^C [\sim] L_{j_1}^{\tilde{C}_1}$ и $L_i^C [\sim] L_{j_2}^{\tilde{C}_2}$, $j_1 \neq j_2$, причем $L_{j_1}^{\tilde{C}_1} \in S_L^B$ и $L_{j_2}^{\tilde{C}_2} \in S_L^B$. Исходя из определения бинарного отношения эквивалентности наборов листьев, в составе набора листьев L_i^C найдется хотя бы одна вершина $a_k \in L_i^C$, одновременно входящая в состав наборов листьев $L_{j_1}^{\tilde{C}_1}$ и $L_{j_2}^{\tilde{C}_2}$: $a_k \in L_{j_1}^{\tilde{C}_1}$ и $a_k \in L_{j_2}^{\tilde{C}_2}$. Другими словами, $a_k \in L_{j_1}^{\tilde{C}_1} \cap L_{j_2}^{\tilde{C}_2} \neq \emptyset$, то есть в составе дерева B присутствует неортогональная пара наборов листьев, что противоречит теореме 1. Полученное противоречие доказывает теорему.

Следствие. Дереву A может быть изоморфно не более одного поддерева в составе дерева B .

Метод выявления изоморфных вхождений R -выражений

Приведенные и доказанные теоремы позволяют сформулировать метод проверки отношения нестрогого включения для заданной пары R -выражений A и B и отыскания подстановки изоморфизма в случае $A[\subseteq]B$. Его основные идеи отражены ниже.

На основании теоремы 1 несложно показать, что если имеет место отношение $A[\subseteq]B$, то каждому из наборов листьев L_i^A можно найти единственный эквивалентный набор листьев L_j^B , такой что $L_i^A [\sim] L_j^B$ (невозможности нахождения такого набора листьев L_j^B , согласно условию 1, хотя бы для одного набора листьев L_i^A говорит о том, что $A[\not\subseteq]B$). Таким образом, необходимо проведение попарного сопоставления наборов листьев деревьев A и B с целью установления соответствия в виде подстановки изоморфизма

$$\begin{bmatrix} L_1^A & L_2^A & \cdots & L_n^A \\ L_{i_1}^B & L_{i_2}^B & \cdots & L_{i_n}^B \end{bmatrix}$$
. С учетом принятого табличного представления деревьев искомую

подстановку удобно представлять в виде номеров позиций наборов листьев в табличном представлении. Для этого потребуется порядка $O(N_{L_{\max}}^2 L_{\max})$ операций при последовательной программной реализации действий.

Далее (в случае невыполнения необходимых условий 1 и 2 отсутствия r -изоморфизма) необходимо осуществить достройку снизу-вверх дерева A одновременно с его единственным (согласно теореме 2) изоморфным эквивалентом в составе дерева B на

основании анализа множества дуг E^A дерева A . Одновременно с этим возможно выясне-

ние подстановки изоморфизма $\begin{bmatrix} T_1^A & T_2^A & \dots & T_m^A \\ T_{j_1}^B & T_{j_2}^B & \dots & T_{j_m}^B \end{bmatrix}$ (а также корректировка значений

типа соответствия узлов, используемая в дальнейшем при удалении и вставке поддерева в рамках выполнения операции подстановки [3]). При этом, согласно следствию 1, не может быть коллизий, связанных с наличием ссылок нескольких наборов листьев на один и тот же узел.

Интересной особенностью операции проверки r -изоморфизма является отсутствие проверки на совпадение типа узлов при достройке дерева вверх: типы узлов в любой ветви дерева A и ее изоморфной ветви дерева B совпадают у узла-предка наборов листьев (лемма 2) и далее чередуются вплоть до корня (следствие 2). Процесс достройки дерева вверх требует порядка $O(N_{T_{\max}})$ операций при программной реализации.

Выводы

Таким образом, выяснение изоморфизма R -выражений при последовательной программной реализации предложенного метода возможно за время порядка $O(N_{L_{\max}}^2 L_{\max} + N_{T_{\max}}) \approx O(L_{\max}^3)$, полиномиально зависящее от размерности исходных данных. Универсальный переборный алгоритм [3] потребовал бы для решения данной задачи порядка $O((N_R!)^2) \approx O((L_{\max}!)^2)$ операций, где $N_R = N_{L_{\max}} + N_{T_{\max}}$. Аналогичная асимптотика свойственна и алгоритмам наподобие [12, 13], основанным на выявлении предположительно-изоморфных пар вершин с последующим рекуррентным сравнением подмножеств вершин. Несмотря на относительно узкую область применения разработанного метода, возможно его расширение на ряд схожих задач (например, на выделение общих подвыражений в арифметических выражениях, выполняемое многими оптимизирующими компиляторами).

Библиографический список

1. Организация и синтез микропрограммных мультимикроконтроллеров / И.В. Зотов [и др.]. — Курск: КурскГТУ, 1999. — 368 с.
2. Зотов И.В., Колосков В.А., Титов В.С. Выбор оптимальных разбиений алгоритмов при проектировании микроконтроллерных сетей // Автоматика и вычислительная техника. — 1997. № 5. — С. 51–62.

3. Ватутин Э.И., Зотов И.В. Поиск базового сечения в задаче разбиения параллельных алгоритмов // Рус. деп. в ВИНТИ 24.11.03 № 2036-B2003. — Курск, 2003. — 30 с.
4. Ватутин Э.И., Зотов И.В., Титов В.С. Построение множества сечений в задаче оптимального разбиения параллельных управляющих алгоритмов // Известия ТулГУ. Вычислительная техника. Информационные технологии. Системы управления. — Тула: ТулГУ, 2003. Т. 1. Вып. 2. — С. 70–77.
5. Ватутин Э.И., Зотов И.В. Метод формирования субоптимальных разбиений параллельных управляющих алгоритмов // Параллельные вычисления и задачи управления (РАСО'04). — М.: ИПУ РАН, 2004. — С. 884–917.
6. Ватутин Э.И., Зотов И.В. Программная система для построения разбиений параллельных управляющих алгоритмов // Идентификация систем и задачи управления (SICPRO'06). — М.: ИПУ РАН, 2006. — С. 2239–2250.
7. Ватутин Э.И., Зотов И.В. Визуальная среда синтеза разбиений параллельных алгоритмов логического управления // Свидетельство об официальной регистрации программы для ЭВМ № 2007613222 от 30.07.07.
8. Ватутин Э.И., Волобуев С.В., Зотов И.В. Комплексная сравнительная оценка методов выбора разбиений при проектировании логических мультиконтроллеров // Идентификация систем и задачи управления (SICPRO'08). — М.: ИПУ РАН, 2008. — С. 1917–1940.
9. Зыков А.А. Основы теории графов / А.А. Зыков. — М.: «Наука», 1987. — 381 с.
10. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений, 2-е изд.: Пер. с англ. — М.: «Вильямс», 2002. — 528 с.
11. Ватутин Э.И., Зотов И.В. Аппаратная модель для определения минимального числа блоков при декомпозиции параллельных алгоритмов логического управления // Известия вузов. Приборостроение. — 2008. Т. 51, № 2. — С. 39–43.
12. А. с. СССР 596951, МКИ³ G06F15/20. Устройство для определения изоморфизма графов / В.М. Курейчик, В.А. Калашников, А.Г. Королев (СССР). — № 2323377/18–24; заявл. 16.02.76; опубл. 05.03.78; Бюл. № 9. — 5 с.
13. А. с. СССР 732879, МКИ³ G06F15/20. Устройство для определения изоморфизма ориентированных графов / А.Г. Королев, В.М. Курейчик, В.А. Калашников (СССР). — № 2540026/18–24; заявл. 04.11.77; опубл. 05.05.80; Бюл. № 17. — 5 с.