

УДК 519.714.5

СХОДЯЩИЙСЯ АЛГОРИТМ АНАЛИЗА ИЕРАРХИИ ДИНАМИЧЕСКИ РАЗВИВАЮЩИХСЯ СИСТЕМ

Голдобин А. А., Запорожский национальный технический университет, Запорожье

*Рассмотрены проблемы разработки алгоритмов
автоматизированного анализа открытых, динамически развивающихся
систем.*

В последнее время имеет место тенденция сокращения этапов жизненного цикла изделий вычислительной техники [1]. Например, средний срок разработки нового изделия в базе программируемых логических интегральных схем (ПЛИС) уплотнен до 2-3 месяцев. Приоритет исследования иерархии вновь создаваемых и находящихся в эксплуатации систем быстро растет, и акцент соответствующей дискуссии переместился из сферы проблемно ориентированного обоснования фундаментальных идей теории ациклических графов [2] в область практической реализации применительно к перспективным информационным системам и процессам [3].

Актуальность автоматизированного анализа иерархии развивающихся систем обусловлена в равной мере широким и частично определенным спектром системообразующих приложений на протяжении цикла жизни, от эскизного проекта системы до вывода ее из эксплуатации.

Постановка проблемы. Дана система, образованная на множестве узлов $U = \{a_i\}$, $i = 0..u$. Между узлами системы установлены каналы однородного подчинения, заданные парами (a_s, a_d) , где a_s – узел-источник, a_d – узел-получатель. Совокупность всех каналов подчинения образует таблицу $T = \{(a_s, a_d)_j\}$, $a_s \in U, a_d \in U, j = 0..t$. Задан узел a_0 , являющийся корнем дерева иерархии и определено непустое множество финальных узлов $R, R \subset U$. Необходимо выполнить сортировку элементов таблицы каналов T таким образом, чтобы возможно было установить внешний вид дерева иерархии системы.

Решение проблемы

Установление иерархии является корректным, если индексированная матрица смежности IM иерархического дерева содержит маркированные узлы в ячейках $IM(i, j)$ с индексами i и j , для которых выполняется условие

$$i > j. \quad (1)$$

Здесь j – индекс столбца, $j = 0..u-1$; i – индекс строки, $i = 0..u-1$.

Сформулируем общие указания относительно порядка формального исследования иерархии.

Шаг 1. Составляется совмещенная матрица смежности SM . Всем корням иерархии

системы ставится в соответствие индекс 0 – крайний младший индекс матрицы SM.

Шаг 2. Осуществляется заполнение строк матрицы смежности в соответствии с правилом маркировки.

Шаг 3. Используя изоморфные перестановки, осуществляется перемещение листьев иерархии в область крайних старших индексов. Формируется конфигурация матрицы смежности KSM.

Шаг 4. Применяя изоморфные перестановки, относительно индексов матрицы смежности, сортируются транзитные узлы таким образом, чтобы для индексов строк и столбцов выполнялось условие (1). В результате образуется верхняя треугольная матрица, соответствующая текущей конфигурации.

Шаг 5. Решение существует, если на графе определен хотя бы один финальный узел. На основе треугольной матрицы SM строится граф иерархии системы.

Синтез изображения иерархического графа выполняется построчно, начиная со строки, имеющей индекс 0 и далее в порядке возрастания индекса. Основной принцип такой: так как для индексов маркированных ячеек верхней треугольной матрицы всегда выполняется условие (1), то узлы, которым соответствуют индексы столбцов маркированных на строке ячеек, должны размещаться на уровнях иерархии, ниже текущего.

Алгоритм 1. Сортировка транзитных узлов матрицы смежности.

1. Построение таблицы каналов. Начиная с индекса актуальной строки y матрицы смежности, осуществляется поиск маркированных ячеек. Индекс строки i маркированной ячейки копируется в элемент s объединения T , индекс столбца j – в элемент d . Пары, где $s = y$ автоматически размещаются первыми. Неиспользованные пары (s, d) заполняются NULL.
2. Составление и сортировка списка наследников. Из всех пар элементов таблицы T , где $s = y$, последовательно копируются элементы d в список наследников Q . Элементы списка Q сортируются в порядке возрастания. Неиспользованные элементы списка наследников заполняются NULL.
3. Составление списка запрещенных индексов. В список X заносятся индексы в диапазоне от 0 до y . Неиспользованные элементы списка запрещенных индексов заполняются NULL.
4. Составление списка финальных индексов. В список R заносятся элементы списка индексов D , такие, что $d_i \geq v$. Здесь v – первый индекс поля финальных узлов матрицы смежности. Неиспользованные элементы списка финальных индексов заполняются NULL.
5. Составление и сортировка списка позиций перестановки. В список S помещаются индексы столбцов j немаркированных ячеек, расположенных в актуальной строке, причем $j \notin X$. Элементы списка S сортируются в порядке возрастания. Неиспользованные элементы списка позиций заполняются NULL.
6. Анализ списка наследников. Начало исполнения двух вложенных циклов, где внешний цикл предназначен для сканирования непустых элементов списка наследников Q (q – индекс актуального наследника), а внутренний цикл – для

сканирования элементов списка позиций C (c – индекс актуальной позиции перестановки). Два варианта условия выхода из цикла:

$$c < q \ \& \ c \notin X \ \& \ c \notin R \ \& \ q \notin X \ \& \ q \notin R; \quad (2)$$

$$c > q \ \& \ c \notin X \ \& \ c \notin R \ \& \ q \in X \ \& \ q \notin R. \quad (3)$$

Условие (2) обеспечивает формирование изоморфной перестановки на подмножестве элементов верхнего треугольника матрицы смежности. Условие (3) позволяет выполнить перестановку на всем диапазоне индексов столбцов.

7. Определение типа и направления перестановки. Конец исполнения циклов сканирования списков C и Q . Если проверка условий (2) и (3) на всем диапазоне сканирования дает ЛОЖЬ, то индекс актуальной строки u увеличивается на единицу и если $u < v$, выполняется переход на п.1.
8. Перестановка столбцов матрицы смежности. Осуществляется взаимный обмен значениями столбцов, имеющих индексы c и q . В протокол индексирования P записывается текстовая строка вида « $c := q$ ».
9. Перестановка строк матрицы смежности. Осуществляется взаимный обмен значениями строк, имеющих индексы c и q . В протокол индексирования P записывается текстовая строка вида « $q := c$ ».
10. В качестве символа NULL используется максимальное значение, возможное в физическом диапазоне беззнакового типа данных `int`. Например, если слово данных целочисленного типа имеет длину два байта, то `NULL = FFFFh = 65535`.
11. Для маркированных ячеек корректно составленной верхней треугольной матрицы смежности должно выполняться условие (1). Поэтому после сортировки транзитных узлов необходимо выполнить проверку индексов маркированных элементов матрицы. Если хотя бы одно сравнение индексов строк i и индексов столбцов j дает в результате ЛОЖЬ, исполнение Алгоритма 1 следует повторить.
12. Для защиты от бесконечного выполнения рефлексивных перестановок, которые не приводят к сортировке транзитных узлов, необходимо перед выполнением пп.8 и 9 проверять содержание крайних текстовых строк протокола индексирования P .

Алгоритм 2. Сортировка финальных узлов матрицы смежности.

1. Вычисление первого индекса поля финальных узлов. $v := u - h$. Здесь v – первый индекс поля финальных узлов матрицы смежности; u – общее количество узлов; h – количество финальных узлов.
2. Удаление списка запрещенных индексов. Элементы списка X устанавливаются в NULL.
3. Составление и сортировка списка финальных индексов. Путем взаимного сопоставления содержимого списков U , K и D , заполняется список финальных индексов R . Здесь U – список узлов графа, K – список финальных узлов, D – список индексов. Элементы списка R сортируются в порядке возрастания.

4. Составление и сортировка списка позиций перестановки. В список C помещаются индексы строк i , содержащих маркированные ячейки, если $i \geq v$. Здесь v – первый индекс поля финальных узлов. Элементы списка C сортируются в порядке возрастания. Неиспользованные элементы списка позиций заполняются NULL.
5. Анализ списка финальных индексов. Начало исполнения двух вложенных циклов, где внешний цикл предназначен для сканирования элементов списка финальных индексов R (q – индекс актуального наследника), а внутренний цикл – для сканирования непустых элементов списка позиций C (c – индекс актуальной позиции перестановки). Условие выхода из цикла:

$$v < q \ \& \ c \notin X \ \& \ c \notin R \ \& \ q \notin X \ \& \ q \notin R; \quad (4)$$

Условие (4) обеспечивает формирование изоморфной перестановки, перемещающей финальный узел в соответствующее поле матрицы смежности.

Перестановка столбцов матрицы смежности. Выполняется, если срабатывает условие (4). Осуществляется взаимный обмен значениями столбцов, имеющих индексы c и q . В протокол индексирования P записывается текстовая строка вида « $c := q$ ».

Перестановка строк матрицы смежности. Выполняется, если условие (4) дает ИСТИНА. Осуществляется взаимный обмен значениями строк, имеющих индексы c и q . В протокол индексирования P записывается текстовая строка вида « $q := c$ ».

Определение типа и направления перестановки. Конец исполнения циклов сканирования списков C и R . Осуществляется переход на выполнение Алгоритма 1.

Результаты и рекомендации

Рекомендации, касающиеся вопроса практического исполнения изоморфных перестановок разного типа.

1. При подготовке к выполнению «бесполезных» перестановок $Zp(c \leftrightarrow q)$, приоритет сканирования списка наследников Q выше приоритета проверки списка позиций C .
2. При подготовке к выполнению «полезных» перестановок $Sp(c \leftrightarrow q)$, приоритет сканирования списка позиций C выше приоритета проверки списка наследников Q .
3. При подготовке к выполнению «нормирующих» перестановок $Np(c \leftrightarrow q)$, приоритет сканирования списка позиций C выше приоритета проверки списка финальных индексов R .

Формальный анализ иерархии предполагает соблюдение принципа постоянного документирования. Любое изменение в порядок соответствия узлов графа индексам поля IM совмещенной матрицы смежности должно фиксироваться в протоколе индексирования P . Такой подход дает возможность анализировать иерархические системы произвольной сложности без потерь данных.

На рис. 1 показан пример формирования графа иерархии.

