

УДК 62.001.57

Зав. кафедрой В.Е. Марлей

(Государственный университет морского и речного флота) кафедра вычислительных систем и информатики. тел. (812) 748-96-35

E-mail: vmarley@yandex.ru

старший преподаватель С.Н. Плотников

(Воронежский филиал ГМУРФ) кафедра информационных систем и технологий

E-mail: 279622@mail.ru

Head. of department V.E. Marlei

(State University of sea and river fleet) Department of computing and informatics.

phone (812) 748-96-35

E-mail: vmarley@yandex.ru

senior lecturer S.N. Plotnikov

(Voronezh branch of SUSRF) Department of information systems and technologies

E-mail: 279622@mail.ru

Алгоритм распознавания изоморфного вложения алгоритмических сетей

The algorithm of isomorphous investment of the algorithmic networks

Реферат. Проблема снижения барьера между пользователем и ЭВМ появилась сразу же, как появилась ЭВМ, и остается актуальной и в настоящее время. Эта проблема формулируется как задача разработки дружественного интерфейса. Одним из путей решения проблемы является использование графического представления, которое может реально уменьшить барьер между человеком и ЭВМ. В этом ряду стоит и формализм алгоритмических сетей, предназначенный для описания алгоритмических моделей, предложенный В.В.Иванищевым около 30 лет назад. Под алгоритмической моделью понимается формализованное описание сценария предметного специалиста для моделируемого процесса, структура которого сопоставима со структурой причинно-следственных и временных зависимостей между явлениями моделируемого процесса, вместе со всей информацией, необходимой для ее программной реализации. Статья посвящена определению изоморфного вложения алгоритмических сетей и описанию необходимых преобразований для реализации, используя принцип деления вершин на классы. Также в этой статье представлен и подробно описан новый подход и алгоритм распознавания изоморфности алгоритмических сетей и аспекты его применения при поиске в базах моделей.

Summary. The problem of reducing the barrier between the user and the computer appeared immediately, as there were computers, and remains relevant in the present time. This problem is formulated as the task of developing a user-friendly interface. One way of solving this problem is the use of graphical representation, which can really reduce the barrier between human and computer. In this series stands and algorithmic formalism networks intended to describe algorithmic models proposed by V.V. Iaanishchev about 30 years ago. Under algorithmic model refers to a formalized description of the scenario subject specialist for the simulated process, the structure of which is comparable with the structure of the causal and temporal relationships between phenomena simulated process, together with all information necessary for its software implementation. This article is devoted to the definition of isomorphous investment of the algorithmic networks and description of conversions for implementation, using principle to division vertices into classes. Also in this article presented and described approach end algorithm to identification of isomorphism algorithmic networks in detail and its using for base of algorithmic models.

Ключевые слова: алгоритмическая модель, алгоритмическая сеть, изоморфизм, абстрагирование, алгоритм.

Key words: algorithmic model, algorithmic network, isomorphism, abstraction, algorithm.

Проблема снижения барьера между пользователем и ЭВМ появилась сразу же, как появилась ЭВМ, и остается актуальной и в настоящее время. Эта проблема формулируется как задача разработки дружественного интерфейса. Одним из путей решения проблемы является использование графического представления, которое может реально уменьшить барьер между человеком и ЭВМ. В этом ряду стоит и формализм алгоритмических сетей, предназначенный для описания алгоритмических моделей,

предложенный В.В.Иванищевым около 30 лет назад [1]. Под алгоритмической моделью понимается формализованное описание сценария предметного специалиста для моделируемого процесса, структура которого сопоставима со структурой причинно-следственных и временных зависимостей между явлениями моделируемого процесса, вместе со всей информацией, необходимой для ее программной реализации.

© Марлей В.Е., Плотников С.Н., 2014

Разработка теории алгоритмических сетей, методов представления моделей на основе данного формализма и построение систем автоматизации моделирования выполнялись в Санкт-Петербургском институте информатики и автоматизации РАН [3]. Алгоритмические сети использовались для создания ряда моделей, которые эксплуатировались в различных организациях (например, Госплан РСФСР) и используются и в настоящее время. Однако язык представления требует средств поддержки автоматизированного построения моделей. Актуальным моментом автоматизации построения моделей является возможность использования фрагментов ранее созданных моделей, использование баз данных моделей.

Прежде чем перейти к описанию методов поиска фрагментов в базах моделей введем их формальные определения [1].

Алгоритмическая сеть (АС) определяется как ориентированный гиперграф без петель при вершинах $G(V, X)$, в котором дуги обозначают модельные переменные $x_i \in X, i = \overline{1, n}$, а вершины $v_j \in V$ – функциональные соотношения (операторы), $f_j \in F, j = \overline{1, m}$, связывающие модельные значения переменных на интервале времени, соответствующем Δt [3].

Структура переменных сети $X = X_{вх} \cup X_{вых} \cup X_{вн}$, где $X_{вх}, X_{вых}, X_{вн}$ – соответственно входные, выходные и внутренние переменные, причем, $X_{вх} \cap X_{вых} = \emptyset, X_{вх} \cap X_{вн} = \emptyset, X_{вых} \cap X_{вн} = \emptyset$. Учитывая, что все вершины АС наблюдаемы, можно задать $X = X_{вх} \cup X_{выч}, X_{выч}$ – вычисляемые переменные АС, $X_{выч} = X_{вых} \cup X_{вн}$, где $X_{выч}$ – вычисляемые переменные.

Формально алгоритмическая сеть (АС) может быть определена следующим образом:

$$AC ::= \langle P, Q, X, F, P \rightarrow F, Q \rightarrow X \rangle, \quad (1)$$

где P – множество вершин сети; Q – множество дуг сети; X – множество переменных, при этом $X = \cup X_i$; X_i – множество переменных i -й вершины; I – множество всех индексов вершин; F – множество операторов; $P \rightarrow F$ – изоморфное отображение P на F , т.е. индекс оператора можно считать индексом вершины; $Q \rightarrow X$ – отображение Q на X .

Оператор $f_i \in F$ описывается как:

$$f_i ::= \langle X_{iin}, f^i, X_{iout} \rangle, \quad (2)$$

где X_{iin}, X_{iout} – множество входных и выходных переменных оператора, f^i – символ операции или функции. Для некоммутативных операторов в X_{iin}, X_{iout} должен быть задан частичный порядок для их элементов. Так, для операции

вычитания в X_{iin} необходимо отметить уменьшаемое, порядок следования остальных переменных несущественен. В общем случае f^i может быть именем некоторого программного модуля или другой АС. Операторы могут быть определены над действительными числами, логическими переменными, скалярами, векторами, матрицами, информационными структурами и др. В состав операторов АС включается оператор задержки, который задает исходное состояние моделируемому процессу и описывает переход модели на следующий шаг или определяет задержку появления некоторой переменной в части АС. В АС допускаются контура, только если они включают в себя оператор задержки. АС будем называть канонической, если все операторы задержки определяют рекуррентные соотношения относительно одной и той же величины и с одинаковым шагом ее изменения. Иными словами, в канонической АС все операторы задержки срабатывают одновременно и обеспечивают формирование одного внешнего цикла счета операторов сети. В дальнейшем рассматриваем именно такие АС.

Алгоритмическая модель (АМ) формально определяется:

$$AM ::= \langle IM, OPO, IO, OP, AC, T, MD, X \rightarrow T, F \rightarrow T \rangle, \quad (3)$$

где IM – идентификатор модели; OPO – описание предметной области; IO – описание объекта, для которого разработана модель; OP – описание процесса, для которого разработана модель; AC – алгоритмическая сеть (описание структуры модели); T – множество терминов предметной области, используемых в модели, $T \subseteq TPO$, где TPO – множество всех терминов предметной области; MD – множество вариантов законов изменения значений переменных модели; $X \rightarrow T$ – отображение множества переменных АС модели в множество терминов предметной области, используемых в модели, возможно $X' \rightarrow T$, где $X' \subseteq X$; $F \rightarrow T$ – отображение множества операторов АС модели в множество терминов предметной области, возможно $F' \rightarrow T$, где $F' \subseteq F$; $X \rightarrow T \cap F \rightarrow T = \emptyset$;

если модель не пуста, то всегда не пусты IM и AC.

Для АС определены отношения равенства и изоморфности. Две АС равны ($AC_1 = AC_2$), если и только если их множества F совпадают ($F_1 = F_2$). АС изоморфны ($AC_1 \sim AC_2$), если и только если между их множествами F можно определить взаимно-однозначное отображение ($F_1 \sim F_2$), при условии что поставленные в соответствие элементы множеств F_1, F_2 имеют совпадающие символы операций f^i и

одинаковые мощности множеств $\text{in}(f_i)$ и $\text{out}(f_i)$. То есть изоморфные АС отличаются друг от друга только обозначениями переменных. Соответственно изоморфное вложение для АМ будем рассматривать как установление изоморфизма одной АС фрагменту другой АС.

Для поиска фрагментов возможны два случая:

1. Все переменные моделей относятся к одному тезаурусу, тогда для поиска фрагментов АС достаточно использовать веденные для АС операции пересечения, на которой основаны алгоритмы установления равенства и вложения [1].

2. Переменные моделей относятся к разным тезаурусам. В данном случае для поиска полной сети может быть использован алгоритм, описанный в [2], но для поиска фрагмента, данный алгоритм не подходит. Необходимо разработать новый алгоритм.

Алгоритм установления изоморфного вложения требует установить соответствие между вершинами двух АС и носит переборный характер. Каждая вершина характеризуется своим индексом, множеством переменных связанной с ней X_i , символом оператора, сопоставленного вершине f^i , который у соответствующих друг другу вершин сравниваемых АС всегда один и тот же. То есть, при установлении соответствия вершин требуется определить соответствие переменных вершин имеющих одинаковые f^i и одинаковое число входных и выходных переменных.

Если сгенерировать все возможные варианты переобозначения переменных одной модели по образцу обозначения другой, а потом их все перебрать и проанализировать, то если сети изоморфны или изоморфно вложимы, всегда можно будет найти хотя бы один такой вариант. Эта идея положена в основу алгоритмов установления изоморфизма и изоморфного вложения графов и конечных автоматов.

Используем следующее допущение: разбиение множеств вершин каждого из графов на непересекающиеся классы и подклассы, между элементами которых возможно сопоставление имён переменных, должно, как минимум, не увеличивать, а зачастую уменьшать число возможных вариантов сопоставления имён переменных сравниваемых сетей; увеличение числа классов и подклассов увеличивает вероятность снижения числа вариантов, рассматриваются только связанные АС (имеющие только одну компоненту связности).

Проведём предварительное преобразование анализируемых АС. Разорвем контура на входах операторов задержек и выполним

для обеих АС их упорядочение по уровням вычислимости. То есть первый уровень будет состоять только из тех вершин, которые могут быть вычислены только на основании входных переменных, второй из тех вершин, в которых хоть одна входная переменная вычисляется в первом уровне (но не в других уровнях) и т.д., выходы операторов задержки рассматриваем как входные переменные. В результате в каждой АС множества вершин разобьётся на классы. Каждая вершина в АС характеризуется собственными $|\text{in}(f_i)|$ (множество входных переменных оператора), $|\text{out}(f_i)|$ (множество выходных переменных оператора) и f^i (символ операции с индексом), соответствующим ей уровнем вычислимости и списком вершин АС, связанных с ней или по входу или по выходу. Каждая вершина из такого списка также характеризуется собственными $|\text{in}(f_i)|$, $|\text{out}(f_i)|$ и f^i , соответствующим ей шагом алгоритма. Все вершины, имеющие одинаковые перечисленные характеристики, попадают в один класс. Таким образом, имеем достаточно подробное деление на классы, которое в подавляющем большинстве практических случаев обеспечивает наличие одного или нескольких классов вершин состоящих из одного элемента, что существенно сокращает число возможных сопоставлений, а в ряде случаев однозначно определяет возможность переобозначения переменных в одной из АС. Реализация алгоритма разбиения на классы требует многократных просмотров множества F , но, в силу своей однонаправленности и однозначности определения принадлежности вершины к какому-либо классу, будет иметь полиномиальную оценку верхней границы сложности.

АС изоморфно вложимы (обозначим это $\sim \subseteq \text{АС}_1$), если и только если между их множеством F_2 и некоторым подмножеством F_1 можно определить взаимно-однозначное отображение (изоморфизм), при условии что поставленные в соответствие элементы множества F_2 и подмножества F_1 имеют совпадающие символы операций f^i и одинаковые мощности множеств $|\text{in}(f_i)|$, $|\text{out}(f_i)|$. То есть изоморфные АС отличаются друг от друга только обозначениями переменных.

Алгоритм распознавания изоморфности АС построен на основе аналогичных алгоритмов распознавания строгой эквивалентности алгоритмов. На изоморфность имеет смысл проверять только сети, имеющие одинаковое число классов вершин, и если соответствующие друг другу классы равномощны. Алгоритм организует просмотр всех возможных вариантов переобозначения и заключается в по-

очерёдном просмотре вершин одной АС и выделенных классов в другой и выбор для каждой вершины одной АС вершины другой АС, принадлежащей к тому же классу, переобозначению переменных выбранной вершины, с учётом ранее сделанных переобозначений. Если возникли противоречия с ранее сделанными переобозначениями, то формируется другой вариант преобразования для анализируемой вершины. Если таких элементов в классе нет, то запоминается вариант переобозначения для всех ранее переобозначенных вершин как недопустимый и возвращаются к началу просмотра и т.д. Если удалось найти хотя бы один вариант переобозначения, то сети изоморфны. Если варианта переобозначения не найдено, то сети не изоморфны. Предпочтительно начинать анализ с классов, имеющих наименьшее число элементов. Данный алгоритм подробно рассмотрен в [2]. Однако он не решает проблему нахождения подмножества F_2 , с которым должен устанавливаться изоморфизм.

Рассмотрим теперь процесс поиска фрагмента AC_1 , который может быть эквивалентен AC_2 . Прежде всего необходимо, чтобы число уровней вычислимости в AC_1 было не менее чем в AC_2 , далее, чтобы мощности и число всех выделенных подклассов AC_1 в каждом уровне были не менее чем в AC_2 . После данных проверок начинается собственно работа алгоритма распознавания изоморфного вложения. Данный алгоритм отличается от алгоритма, имеет в своей основе идею несколько отличную от идеи, изложенной в предыдущей статье [3]. Прежде всего строится граф связей между классами AC_2 . Связь между классами означает, что хотя бы один элемент класса связан с хотя бы одним элементом другого класса, дуги между классами не имеют наименований. Аналогично строится граф связей классов для AC_1 . Правила операций над графами классов принимаем такие же, как над АС [1]. Далее проводим операцию

пересечения графов классов. Если получаем результат, что пересечение полностью включает граф классов AC_2 , то возможен дальнейший поиск изоморфного вложения.

Далее выделяется подграф графа классов AC_1 , соответствующий графу классов AC_2 . В нем проверяется, что мощность соответствующих классов не менее мощности соответствующих классов в графе классов AC_2 , далее ищется связный подграф, который проверяется по ранее разработанному алгоритму [2] на изоморфность с AC_2 . Если связного подграфа нет, то изоморфного вложения нет. Возможно, что таких подграфов будет несколько.

Таким образом, можно сформулировать следующие утверждения:

1. Утверждение 1.

Если АС изоморфны, то их графы классов равны.

2. Следствие утверждения 1.

Если АС равны, то их графы классов равны.

3. Утверждение 2.

Если АС изоморфно вложима в другую АС, то ее граф классов есть подграф графа классов большей АС.

4. Следствие утверждения 2.

Если АС есть подграф другой АС, то ее граф классов есть подграф графа классов большей АС.

Доказательство данных утверждений и соответствующий им алгоритм будет приведено в следующей статье.

Данный подход соответствует определению изоморфного вложения АС и обеспечивает окончание работы соответствующего алгоритма в конечное время, а большое количество классов вершин позволяет сократить время поиска и распознавания, что позволяет его использовать для поиска фрагментов в базе моделей независимо от того к какой предметной области модели принадлежат.

ЛИТЕРАТУРА

1 Иванищев В.В., Марлей В.Е. Введение в теорию алгоритмических сетей. СПб: СПбГТУ, 2000, 180 с.

2 Васильченко Д.С. Распознавание изоморфизма алгоритмических сетей при моделировании транспортных процессов // Журнал университета водных коммуникаций. 2010. № 8.

3 Плотников С.Н. Распознавание изоморфного вложения алгоритмических сетей // Вестник ВГУ. 2014. № 2. С. 5-9.

REFERENCES

1 Ivanishchev V.V., Marlei V.E. Vvedenie v teoriyu algoritmicheskikh setei [Introduction to the theory of algorithmic networks]. Saint Petersburg, SPbGTU, 2000. 180 p. (In Russ.).

2 Vasil'chenko D.S. Detection of isomorphism algorithmic networks for modeling transport processes. *Zhurnal universiteta vodnykh kommunikatsii*. [Journal of University of water communications], 2010, no. 8. (In Russ.).

3 Plotnikov S.N. Recognition isomorphic attachments algorithmic networks. *Vestnik VGU*. [Bulletin of VSU], 2014, no. 2, pp. 5-9. (In Russ.).