

ГРАФЫ КЛАССОВ И АЛГОРИТМ РАСПОЗНАВАНИЯ ИЗОМОРФНОГО ВЛОЖЕНИЯ АЛГОРИТМИЧЕСКИХ СЕТЕЙ

THE CLASS NTNS AND THE ALGORITHM OF ISOMORPHOUS INVESTMENT OF THE ALGORITHMIC NETWORKS

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.

This article is devoted to the definition of isomorphs 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: модель, изоморфизм, алгоритмическая сеть, граф.
Key words: model, iomorphisms, algorithmic network, graph.

В ПРЕДЫДУЩЕЙ статье [1, с. 5–9] рассмотрены основные идеи алгоритма распознавания алгоритмических сетей, было введено понятие графа классов вершин алгоритмических сетей.

Введем формальные определения.

Повторим коротко основные определения для алгоритмических сетей. Алгоритмическая сеть определяется как ориентированный нагруженный граф без петель при вершинах $G(V, X)$, в котором дуги обозначают модельные переменные $x_i \in X$, $i = 1, n$, а вершины $v_j \in V$ — функциональные соотношения (операторы), $f_j \in F$, $j = 1, m$, связывающие модельные значения переменных на интервале времени, соответствующем Δt [2].

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

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

$$AC ::= \langle P, Q, X, F, P \rightarrow F, Q \rightarrow X \rangle,$$

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 , то есть индекс оператора можно считать индексом вершины; $Q \rightarrow X$ — отображение Q на X .

Оператор $f_i \in F$ описывается как $f_i := \langle X_{\text{ин}}, f^i, X_{\text{аут}} \rangle$, где $X_{\text{ин}}$, $X_{\text{аут}}$ — множество входных и выходных переменных оператора, f^i — символ операции или функции. Для некоммутативных операторов в $X_{\text{ин}}$, $X_{\text{аут}}$ должен быть задан частичный порядок для их элементов. Так, для операции

вычитания в X_{in} необходимо отметить уменьшаемое, порядок следования остальных переменных несуществен. 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 контуры 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.

Для АС определены отношения равенства и изоморфности, введены основные операции преобразования [2]. 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 $in(f_i)$ и $out(f_i)$. 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 введенные для АС операции пересечения, на которых основаны алгоритмы установления равенства и вложения [2];

2) переменные моделей относятся к разным тезаурусам. В данном случае для поиска полной сети может быть использован алгоритм, описанный в [3], 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.

Используем следующее допущение, разбиение множеств вершин каждой из АС на непересекающиеся классы и подклассы, между элементами которых возможно сопоставление имен переменных, должно не увеличивать число возможных вариантов сопоставления имен переменных сравниваемых сетей. Увеличение числа классов и подклассов увеличивает вероятность снижения числа вариантов. Рассматриваем только связные АС (имеющие одну компоненту связности).

Проведем предварительное преобразование анализируемых АС. Разорвем контуры 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 хотя 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 разобьются на классы. Каждая вершина в АС характеризуется собственными $|in(f_i)|$ (мощность множества входных переменных оператора), $|out(f_i)|$ (мощность множества выходных переменных оператора) и f^i (символ операции с индексом), соответствующим ей уровню вычислимости и списком вершин АС, связанных с ней или по входу, или по выходу. Каждая вершина из такого списка также характеризуется собственными $|in(f_i)|$, $|out(f_i)|$ и f^i , соответствующим ей шагом алгоритма. Все вершины, имеющие одинаковые перечисленные характеристики, попа-

дают 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 $\sim \subseteq AC_1$), если и только если между их множеством $F2$ и некоторым подмножеством $F1$ можно определить взаимно-однозначное отображение (изоморфизм), при условии что поставленные в соответствие элементы множества $F2$ и подмножества $F1$ 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.

На изоморфность имеет смысл проверять только сети, имеющие одинаковое число классов вершин, и если соответствующие друг другу классы равномощны.

Рассмотрим коротко процесс поиска фрагмента AC_1 , который может быть эквивалентен AC_2 . Прежде всего необходимо, чтобы число уровней вычислимости было 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.

Введем понятие графа связей классов вершин АС. Граф связей классов — это ориентированный нагруженный граф, который определяется:

$$ГКАС ::= \langle KP, G, FN, KP \rightarrow FN, F \rightarrow FN \rangle$$

где KP — множество вершин графа, соответствующих выделенным классам; G — множество дуг графа, дуга между классами означает, что хотя бы один элемент одного класса по выходу связан со входом хотя бы одного элемента в другом классе, дуги не нагружены; FN — множество наборов параметров, определяющих классы; $KP \rightarrow FN$ — изоморфное отображение KP на FN ; $F \rightarrow FN$ — сюръективное отображение множества операторов сопоставленным вершинам АС на множество наборов параметров. Правила операций над графами классов принимаем такие же, как над обычными графами. Два ГКАС равны, когда равны все элементы определения ГКАС.

Теперь основная идея рассматриваемого алгоритма. Строим граф связей между классами AC_2 . Аналогично строится граф связей классов для AC_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.

Рассмотрим признаки, по которым выделяются классы вершин АС, например:

- тип оператора в вершине;
- уровень вычислимости вершины;
- мощность множества входных переменных вершины;
- мощность множества выходных переменных вершины;
- количество вершин других уровней вычислимости, с которыми она связана для каждой выходной переменной вершины.

Любую вершину АС, имеющую несколько выходов, можно представить в виде фрагмента АС, в котором в каждой вершине есть только один выход [2], поэтому в дальнейшем пока рассма-

триваем только такие АС. Таким образом, класс, к которому принадлежит вершина, будет определяться следующими параметрами:

- тип оператора в вершине;
- уровень вычислимости вершины;
- мощность множества входных переменных вершины;
- количество вершин других уровней вычислимости, с которыми связана выходная переменная вершины.

Данные параметры обеспечивают очень подробную факторизацию множества вершин АС. В самом деле, по определению АС вершине может быть приписан только один тип оператора, то есть деление по типу операторов задает непересекающиеся классы. Деление внутри класса вершин с одинаковым типом оператора по уровню вычислимости также дает непересекающиеся подклассы. В соответствии с вышеприведенным алгоритмом определения уровней вычислимости вершина может принадлежать только одному уровню. Деление внутри этого подкласса по мощности множества входных переменных также дает непересекающиеся подклассы, по определению у вершины АС не может быть несколько входных множеств. Количество вершин других уровней, связанных с выходом вершины АС, также дает непересекающиеся множества. То есть введенные параметры обеспечивают разделение множества вершин АС на непересекающиеся классы.

Каждая вершина АС имеет свой уникальный номер i . Таким образом, i -я вершина АС в классе будет характеризоваться:

- 1) типом оператора (f^i);
- 2) номером уровня вычислимости (N);
- 3) мощностью множества входных переменных ($|\text{in}(f_i)|$);
- 4) количеством вершин, с которыми вершина связана по своей выходной переменной (k_i).

Рассмотрим свойства ГКАС.

Утверждение 1.

Для каждой АС существует только один ГКАС.

Доказательство.

Предположим, что для какой-либо АС существуют два различных ГКАС. Это означает, что существует хотя бы одна вершина АС с сопоставленным ей оператором, принадлежащая одновременно к разным классам, но тогда из определения ГКАС $F \rightarrow FN$ не будет являться отображением, что исключает подобный вариант.

Следствие утверждения 1.

ГКАС подграфа АС является подграфом ГКАС исходной сети или равен ему.

Доказательство.

Поскольку ГКАС исходной АС и ее подграфа основан на одном и том же множестве FN , то его множество наборов параметров либо будет совпадать с множеством исходной, либо будет подмножеством его. Если множество наборов подграфа АС есть подмножество исходной, то получаем вариант с подграфом ГКАС исходной АС. Так как каждому классу, соответствующему какому-либо набору параметров, соответствует несколько вершин, то возможна ситуация, когда все вершины подграфа АС входят во все классы соответствующим вершинам исходной АС, то есть ГКАС подграфа будет иметь те же вершины, что ГКАС исходной АС. Аналогично и для множества дуг, тогда получаем вариант равного ГКАС, если множество дуг не совпадает, то вариант подграфа.

Можно сформулировать следующие утверждения, которые позволят упростить поиск изоморфного вложения.

1. *Утверждение 2.*

Если АС изоморфны, то их графы классов равны.

Доказательство.

В самом деле, если AC изоморфны, то существует переобозначение переменных одной из AC , при котором она становится равной второй AC . Следовательно, $ГКАС$ у нее будет равный $ГКАС$ второй AC , так как по утверждению 1 AC имеет только один $ГКАС$.

2. Следствие утверждения 2.

Если AC равны, то их графы классов равны.

Данное следствие есть частный случай утверждения 2, когда не требуется переобозначения.

Следующие утверждения относятся уже к изоморфному вложению, поэтому из набора параметров исключаем уровень вычислимости.

3. Утверждение 3.

Если AC изоморфно вложима в другую AC , то ее граф классов подграф графа классов большей AC или равен ему.

Доказательство.

Если AC изоморфно вложима во вторую AC , значит, существует подграф второй AC , изоморфный рассматриваемой. Тогда, проведя переобозначение у вкладываемой, получаем AC , равную подграфу исходной AC , далее используем следствие утверждения 1 и получаем требуемый результат.

4. Следствие утверждения 3.

Если AC есть подграф другой AC , то ее граф классов подграф графа классов большей AC или равен ему.

Данное следствие есть частный случай утверждения 3, когда не требуется переобозначения.

Рассмотрим теперь укрупненно процедуру установления изоморфного вложения, AC_1 исходная, AC_2 вкладываемая.

1. Начало.

2. Построить $ГКАС$ для AC_1 и AC_2 (без учета уровня вычислимости).

3. Выполнить $ГКАС_1 \cap ГКАС_2$.

4. Если $ГКАС_1 \cap ГКАС_2 \neq ГКАС_2$, то конец — изоморфного вложения нет.

5. В AC_2 выбираем самый длинный путь. (Все начала входных дуг AC считаем исходящей из фиктивной вершины I , все концы выходных дуг AC считаем входящими в фиктивную вершину S , длина каждой дуги единица.)

6. Ищем в AC_1 последовательность вершин с типами операторов, совпадающими с вершинами максимального пути в AC_2 , с мощностями множеств входных переменных числом вершин, связанных с выходами не меньшими, чем в пути. Если таких путей нет, то конец — изоморфного вложения нет.

7. Производим переобозначения переменных в вершинах найденного в AC_2 пути (подграфа AC_2) в соответствии вершинами AC_1 .

8. Проверяем связи вершин полученного подграфа AC_2 . Если тип оператора совпадает, число входных переменных и связей по выходу не более соответствующих вершин соответствует аналогичным вершинам в AC_1 , то включаем их в формируемый подграф и производим переобозначения переменных, нет к 11.

9. Все связанные с рассматриваемым подграфом вершины просмотрены, нет к 8.

10. Подграф изоморфен AC_2 ? Да — конец, определено изоморфное вложение, нет к 8.

11. Запоминаем исходную последовательность в AC_1 как запрещенную, к 6.

Данный алгоритм носит обобщенный характер и ссылается на другие алгоритмы. Рассмотрим их укрупненно.

Алгоритм построения $ГКАС$.

1. Сортировка вершин AC по классам, имеющим одинаковый уровень вычислимости, тип оператора, мощность множества входных переменных и число вершин, связанных с выходом вершины. Это множество вершин $ГКАС$.

2. Просмотр пар полученных классов, если в них есть вершины, связанные дугой, то вершины ГКАС связываются дугой.

3. Все пары классов просмотрены — ГКАС построен.

В АС при определении уровней вычислимости разрываются связи по операторам задержки, то есть АС становится бесконтурной, поэтому ГКАС также бесконтурный.

Пересечение ГКАС — это обычная операция пересечения графов. Равенство — это равенство для нагруженного графа. Поиск максимального пути может быть реализован, например, методом редукции индексов.

Рассмотренный алгоритм может быть использован для определения изоморфности АС, только в пункте 4 нужно вставить определение равенства ГКАС и использовать параметр уровня вычислимости.

Данный алгоритм позволяет его использовать для поиска фрагментов в базе моделей независимо от того, к какой предметной области модели принадлежат.

Ключевые слова

Алгоритмическая модель (АМ) — формализованное описание сценария предметного специалиста для моделируемого процесса.

Алгоритмическая сеть (АС) — конечный ориентированный нагруженный граф, вершинам которого сопоставлены операторы, а дугам — переменные, связываемые операторами.

Изоморфизм — сходство двух или более объектов по форме или строению.

Абстрагирование — это мысленное выделение, вычленение некоторых элементов конкретного множества и отвлечение их от прочих элементов данного множества.

Алгоритм — точный набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное время.

Список литературы

1. Плотников С. Н. Распознавание изоморфного вложения алгоритмических сетей / С. Н. Плотников // Вестник ВГУ. Сер.: Системный анализ и информационные технологии. — Воронеж, 2014. — № 2.
2. Иванищев В. В. Введение в теорию алгоритмических сетей / В. В. Иванищев, В. Е. Марлей. — СПб.: СПбГТУ, 2000. — 180 с.
3. Васильченко Д. С. Распознавание изоморфизма алгоритмических сетей при моделировании транспортных процессов / Д. С. Васильченко // Журнал Университета водных коммуникаций. — СПб., 2010. — Вып. 8.