

Раздел III. Искусственный интеллект и нечеткие системы

УДК 004.832.34

В.Н. Вагин, Д.С. Зарецкий

РЕШЕНИЕ ЗАДАЧ ДИАГНОСТИКИ С ИСПОЛЬЗОВАНИЕМ СИСТЕМ ПОДДЕРЖКИ ИСТИННОСТИ*

Работа посвящена изучению возможностей применения системы поддержки истинности, основанной на предположениях (ATMS) в решении задачи диагностики. В статье описана, как теоретическая сторона ATMS и системы поддержки логического вывода, так и возможность ее практического применения, а также приведен пример использования ATMS как составляющей части системы диагностики логических схем. Кроме этого, выявлены преимущества использования ATMS в системах диагностики, а также означены основные проблемы, решение которых может повысить эффективность полученной системы.

Системы поддержки истинности; задача диагностики; логический вывод.

V.N. Vagin, D.S. Zaretsky

TRUTH MAINTENANCE SYSTEMS IN DIAGNOSTICS PROBLEMS

The paper is devoted to studying application possibilities of an assumption-based truth maintenance system (ATMS) at solving diagnostics problems. The theory of ATMS as a system of supporting logical reasoning and its practical applications are given. The example of using ATMS as a part of a diagnostic system of logical circuits is viewed. The main advantages and basic problems of ATMS application for enhancing the efficiency of diagnostic systems are considered.

Truth maintenance systems; diagnostics problem; logical reasoning.

Введение. Технической диагностикой является отрасль научно-технических знаний, сущность которой составляет теория, методы и средства обнаружения и поиска дефектов объектов технической природы. Под дефектом обычно понимается любое несоответствие свойств объекта заданным, требуемым или ожидаемым его свойствам. В свою очередь обнаружение дефекта – есть установление факта его наличия или отсутствия в объекте. Диагностирование, целью которого является определение места и при необходимости причины и вида дефекта объекта называется поиском дефекта [1-2].

С момента своего появления и до наших дней суть задачи диагностики не изменилась. Однако расширение возможностей вычислительной техники и развитие методов искусственного интеллекта привели к активному развитию задач автоматической диагностики сложных объектов, систем и комплексов.

В ходе развития автоматических систем диагностирования был разработан метод, основанный на модели устройства. Тезисно его суть заключается в моделировании работы диагностируемого устройства с помощью продукционных правил, поиска различий в поведении между моделью устройства и его оригиналом и построении необходимых выводов на основе этих различий [3].

* Работа выполнена при поддержке РФФИ (грант № 08-07-00212а).

Этот метод позволил решить такие проблемы его предшественников (в частности метода, использующего в своей основе знания эксперта) как:

- ◆ сложность извлечения знаний эксперта;
- ◆ неоднозначность формализации знаний.

Однако неразрешенными все же оставались следующие проблемы:

- ◆ несовершенство классических систем логического вывода (требование полноты, состоятельности, непротиворечивости исходных данных, высокая размерность вычислений);
- ◆ неочевидность теоретических расчетов системы и, как следствие, недоверие пользователя к результатам ее работы.

Одним из путей решения этих проблем является использование так называемых систем поддержки логического вывода. Одной из таких систем является ATMS (the Assumption-based Truth Maintenance System – система поддержки истинности, основанная на предположениях), позволяющей в ходе работы делать предположения об исправности или неисправности компонент диагностируемого устройства и, за счет поддержки непротиворечивости системы, выявлять противоречивые множества сделанных предположений, и как следствие, неисправности компонент системы [3-5].

Такой подход имеет ряд преимуществ по сравнению с использованием классического вывода:

- ◆ простота и понятность вывода;
- ◆ простота архитектуры – возможность разбить систему рассуждения на системы логического вывода и поддержки истинности;
- ◆ гарантия непротиворечивости вывода;
- ◆ сужение комбинаторного пространства логического вывода.

Ниже рассмотрим принципы работы диагностирующей системы, использующей ATMS на примере задачи поиска дефекта.

Постановка задачи. Опишем решаемую задачу следующим образом:

1. Пусть имеется модель диагностируемой системы в виде продукционных правил.
2. Одна или несколько компонент системы работают некорректно (результаты работы не соответствуют модели).
3. Имеются некоторые начальные данные (характеристики работы компонент) о состоянии системы.
4. Имеется возможность получать дополнительные сведения (состояние компонент) о системе.
5. Требуется определить неисправные компоненты, затратив на это наименьшее число снятий показаний.

Как уже упоминалось, система рассуждения диагностирующей системы разбивается на две части [3]:

1. Система логического вывода.
2. Система поддержки истинности.

Первая система отвечает за вывод новых утверждений о состоянии диагностируемого устройства из исходных данных. Для этого используются стандартные алгоритмы логического вывода на основе продукционных правил. Это могут быть точные, но громоздкие алгоритмы полного перебора или более изящные алгоритмы с использованием эвристик.

Однако какой бы алгоритм ни использовался, вполне естественно желание оптимизировать его работу, в первую очередь за счет сокращения пространства поиска, с сохранением полноты исходного алгоритма. Это и является основной задачей системы поддержки истинности.

Система поддержки истинности, основанная на предположениях, (ATMS) позволяет на любом этапе проведения логического вывода делать предположения об истинности некоторого множества утверждений. На основе этих предположений система логического вывода способна производить новые утверждения, истинность которых, вообще говоря, неочевидна. Однако за счет сохранения информации о предположениях, поддерживающих эти утверждения, а также анализа возникающих противоречий, ATMS имеет возможность делать вывод о неверности (противоречивости) того или иного подмножества сделанных предположений, что равносильно истинности или ложности соответствующих утверждений.

Остановимся подробнее на работе этой системы.

Концептуальная модель ATMS. Попробуем охарактеризовать работу ATMS следующим образом:

- ◆ задано множество правил хорновского вида.
- ◆ заданы значения (определена заведомая истинность) некоторых утверждений, являющихся посылками правил.

Определение 1. Правилор хорновского вида (хорновским правилор) является правилор вида: $a_1 \& a_2 \& \dots \& a_n \rightarrow b$, где $a_1 \dots a_n$ называются посылками, а b – результатом (заклучением).

Определение 2. Фактом называется утверждение, чья истинность точно определена.

Определение 3. Предположением называется утверждение с неопределенной истинностью, однако временно принятое истинным.

Сделаем предположения об истинности нескольких утверждений, истинность которых еще не определена.

Основным материалом для обработки системы поддержки истинности являются метки утверждения.

Определение 4. Меткой утверждения является множество предположений, на основе которых это утверждение выведено (множество поддержки утверждения).

Для меток утверждений верно следующее [4,6]:

- ◆ метка заведомо истинных утверждений всегда пуста (записывается как {});
- ◆ утверждение, являющееся предположением, поддерживается самим собой ({<имя предположения>});
- ◆ метка утверждения может состоять из нескольких множеств;
- ◆ в случае, если утверждение не поддерживается ни одним из предположений, его метка неопределенна ({ \emptyset });
- ◆ противоречивые множества записываются как поддержка утверждения-противоречия \perp (утверждения Nogoood).

Замечание 1. Утверждение Nogoood, выделяется для удобства обработки противоречивых множеств предположений. Все операции, производимые над обычными утверждениями аналогично производятся и над утверждением Nogoood.

Определим тогда условия применимости правил:

1. Посылка условно истинна, если ее множество поддержки определено.
2. Правилор применимо, если все его посылки условно истинны.
3. Поддержкой результата применимого правилор является объединение множеств поддержки его посылок.
4. Если в ходе логического вывода получены результаты, противоречащие заведомо истинным фактам, то поддержка этих результатов является противоречивой.

Опишем правила минимизации меток:

Rule 1. Любое множество, содержащее противоречивое множество – противоречиво.

Rule 2. Противоречивое множество не может быть поддержкой утверждения. Если поддержка метки содержит противоречивое множество, это множество может быть удалено.

Rule 3. Поддержка утверждения не должна быть избыточной. Если метка противоречивого утверждения содержит два множества A и B , причем, $A \in B$, тогда множество B может быть удалено из метки.

Замечание 2. Рассмотрим пример, иллюстрирующий ход минимизации меток.

Пусть в ходе рассуждения стало известно, что некоторый факт S поддерживается множествами $V=\{A1,A2,A3\}$ и $C = \{A2,A3,A4\}$, причем, известно, что множество $D = \{A3,A4, A5\}$ – противоречиво.

От системы поддержки истинности приходит сообщение о противоречивости множества $E=\{A3,A4\}$, а также о том, что утверждение S поддерживается множеством $F=\{A1,A2\}$. Таким образом, после добавления полученных сведений, состояние системы принимает следующий вид:

$$S \{ \{A1,A2,A3\}, \{A1,A2\}, \{A2,A3,A4\} \}, \\ \perp \{ \{A3,A4,A5\}, \{A3,A4\} \}$$

Видно, что множество $E=\{A3,A4\}$ содержится во множествах $C = \{A2,A3,A4\}$, $D = \{A3,A4, A5\}$. Значит, множества C и D могут быть удалены из поддержки соответствующих меток (**Rule3**). Более того, множество $F=\{A1,A2\}$ полностью входит во множество $V=\{A1,A2,A3\}$, а значит, множество V также может быть удалено из метки, поддерживающей утверждение S (**Rule2**). В итоге, после применения правил минимизации, состояние системы будет выглядеть следующим образом:

$$S \{ \{A1,A2\} \}, \\ \perp \{ \{A3,A4\} \}$$

Подобное изменение состояний системы характерно для работы ATMS и говорит о том, что с каждым добавлением уточняющей информации, с одной стороны, уточняется множество поддержки утверждений, с другой – сужается противоречивое подмножество предположений.

Алгоритм ATMS. Алгоритм ATMS представляет собой итерационное добавление утверждений о диагностируемой системе, поиск противоречивых множеств поддержки, добавление и обновление меток и анализ следующего места снятия показаний (предложение определения истинности наиболее подходящего утверждения, если такое возможно).

Формально это описание выглядит следующим образом:

В качестве входных данных можно читать наличие связи с рассуждающей системой (системой логического вывода).

atms:

Начало:

пока не найдена причина противоречия
[найти наиболее подходящее место для снятия показания];
получить от рассуждающей системы новое *утверждение*;
добавить новое утверждение (**add_sentence**(*утверждение*, {}));
выдать причину противоречия.

Конец.

Примечание 1. Критерий причины противоречия выбирается в зависимости от задачи, точным критерием является мощность противоречивого множества: если поддержка утверждения \perp содержит множество из одного элемента, то это предположение и является причиной противоречия.

Примечание 2. Нахождение наиболее подходящего места для снятия показаний является отдельной и далеко нетривиальной задачей. Точных методов для общего случая не существует. Эвристические же методы калибруются, в первую очередь, под специфику отдельных задач [2].

Процедура добавления нового утверждения имеет следующий вид:

Входными параметрами функции являются добавляемое утверждение, полученное в процессе рассуждения, а также его метка (множество поддержки).

Как в случае с процедурой **atms**, процедуре **add_sentence** необходима связь с системой логического вывода.

add_sentence(утверждение, метка):

Начало:

проверить, есть ли противоречие:

если да –

добавить множество поддержки вызывающих противоречие утверждений в качестве поддержки противоречивого утверждения;

добавить утверждение в базу знаний;

применить правила минимизации меток;

найти все применимые правила (получить их результаты от рассуждающей системы);

для результата каждого применимого правила выполнить процедуру добавления утверждения **add_sentence**(утверждение, метка);

Конец.

Примечание 3. Проверка на наличие противоречия заключается в следующем: содержится ли одинаковое множество поддержки в метках взаимоисключающих утверждений. Простейшим примером является добавление заведомо истинного утверждения ($\{\}$): если уже существует обратное ему утверждение, то метка, поддерживающая его, является противоречивой.

Примечание 4. Процедура минимизации меток заключается в последовательном применении правил минимизации.

Примечание 5. Поиск применимых правил занимается блок логического вывода: это может быть простой переборной процедурой или содержать эвристические элементы.

Реализация ATMS: Основные задачи для решения. Из рассмотренного примера можно выявить следующие положительные черты алгоритма ATMS:

- 1) наглядность работы – все выводы сделанные системой полностью соответствуют логике человека, пытающегося решить проблему диагностики «от противного», принимающего работоспособность элементов схемы в качестве посылки и принимающего факты о состоянии системы до возникновения противоречия;
- 2) нет необходимости следить за непротиворечивостью фактов – поддержка непротиворечивости входит в состав алгоритма ATMS;
- 3) сужение пространства логического вывода – при отсутствии предположений об исправности элементов пришлось бы вводить дополнительные правила, учитывающие этот вариант, что увеличило бы количество правил в два раза, что в значительной мере увеличило бы объем необходимых рассуждений.

Однако очевидными являются и некоторые проблемы, требующие решения для использования ATMS:

1. Трансляция схемы в систему логических правил в общем случае.
2. Поддержка непротиворечивости системы правил. Действительно, хотя ATMS и гарантирует непротиворечивость фактов, выведенных системой, однако, на правила это не распространяется. Это приводит к необходимости накладывать ограничения на использование некоторых правил в логическом выводе.
3. Контроль «зацикливания» логического вывода. Дело в том, что наряду с прямым дедуктивным выводом (от посылок к следствию) при выводе на модели устройства необходимо использовать обратный, абдуктивный (от следствия к посылкам) вывод. Такое совмещение может вызвать появление логических колец и, соответственно, к отсутствию решения, что также необходимо контролировать.
4. Выбор места снятия показаний. На самом деле добавление каждого нового факта для ATMS абсолютно равнозначно, более того, требование, чтобы исходными данными были значения на входах и выходах схемы, абсолютно необязательно. В связи с этим возникает вопрос: как выбрать следующее место снятия показаний так, чтобы результат был получен как можно раньше.

Рассмотрим подробнее некоторые из этих проблем.

Трансляция схемы в систему логических правил. Несмотря на то, что, казалось бы, перевести логическую схему в систему логических правил – дело не самой высокой сложности, однако, в случае использования схем высокой размерности весьма полезным было бы вывести некоторый механизм, а лучше алгоритм для их автоматической обработки.

На самом деле, полная определенность результата работы каждого логического элемента схемы от входных данных позволяет достаточно легко и безболезненно перевести схему в систему логических правил, более того, существует алгоритм, позволяющий унифицировать этот процесс, используя логические функции, соответствующие элементу.

Можно утверждать, что для любого элемента, построенного на основе произвольной булевой логической функции, существует система логических правил, описывающая его работу. Действительно, известно, что любая булева функция может быть описана соответствующей таблицей истинности. Пусть, существует элемент, построенный на основе функции $F = f(in_1, \dots, in_n)$. Построим тогда для него следующую систему правил:

Для любой строки A_i таблицы построим правило следующего вида:

$$\langle in_1 = A_{i_1} \rangle \& \dots \& \langle in_n = A_{i_n} \rangle \rightarrow \langle F = f_i \rangle,$$

где $\langle in_1 \rangle \dots \langle in_n \rangle$ обозначают входы элемента, F – его выход, а $A_{i_1} \dots A_{i_n}$, f_i – значения соответствующего кортежа таблицы истинности.

Замечание 5. Можно легко показать, используя уникальность каждой строки таблицы истинности, что подобное преобразование является однозначным, а полученная система правил полностью соответствует исходной функции.

Таким образом, последовательно рассматривая каждый элемент схемы и используя в качестве названий переменных имена элементов, соединенных с соответствующим входом, а в качестве значений – суперпозицию соответствующих функций, мы можем применить алгоритм, описанный выше, и получить необходимую систему правил, которая будет соответствовать каждому из элементов схемы, а значит, полностью описывать ее работу.

Замечание 6. Подобный результат верен также для функции логики любой размерности (не только булевой), задающейся с помощью таблицы истинности.

Таким образом, мы получаем следующий алгоритм для трансляции любой логической схемы в систему правил, необходимой для работы ATMS:

- 1) сгенерировать таблицу истинности функции, на основе которой был построен элемент;
- 2) для каждой из строк таблицы истинности построить соответствующее правило;
- 3) переобозначить входы элементов, как имена соответствующих элементов схемы, соединенных с данным элементом.

Поддержка непротиворечивости системы правил. Важной проблемой при реализации ATMS является проблема контроля непротиворечивости системы логических правил.

Определение 5. Система правил называется дедуктивно непротиворечивой тогда и только тогда, когда нельзя из одних и тех же посылок вывести при помощи дедукции одновременно истинность и ложность какого-либо утверждения.

Иными словами, система не должна содержать пар правил вида:

Правило 1: $\langle in_1=a_1 \rangle \& \dots \& \langle in_n=a_n \rangle \& \langle out \rangle \rightarrow \langle out=0 \rangle$

Правило 2: $\langle in_1=a_1 \rangle \& \dots \& \langle in_n=a_n \rangle \& \langle out \rangle \rightarrow \langle out=1 \rangle$

для $\forall a_1, \dots, a_n \in \{0,1\}$

Определение 6. Система правил называется абдуктивно непротиворечивой тогда и только тогда, когда нельзя из одних и тех же заключения и достаточного числа посылок вывести при помощи абдукции одновременно истинность и ложность какого-либо утверждения.

То есть система не должна содержать пар правил вида:

Правило 1: $\langle in_1=a_1 \rangle \& \dots \& \langle in_i=0 \rangle \& \dots \& \langle in_n=a_n \rangle \& \langle out \rangle \rightarrow \langle out=b \rangle$

Правило 2: $\langle in_1=a_1 \rangle \& \dots \& \langle in_i=1 \rangle \& \dots \& \langle in_n=a_n \rangle \& \langle out \rangle \rightarrow \langle out=b \rangle$

для $\forall a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n, b \in \{0,1\}, i = 1..n$

Замечание 7. Под достаточным числом посылок понимается та часть множества посылок, на основании достоверности которой и результата можно судить о правдоподобности оставшейся части посылок. Обычно достаточным считается $n-1$ посылок, где n – общее число посылок в правиле.

Рассмотрим, чем грозит нарушение непротиворечивости при использовании алгоритма ATMS.

Пусть в системе правил существуют два правила:

Правило 1: $\langle in_1=0 \rangle \& \langle in_2=0 \rangle \& \langle out \rangle \rightarrow \langle out=0 \rangle$,

Правило 2: $\langle in_1=0 \rangle \& \langle in_2=0 \rangle \& \langle out \rangle \rightarrow \langle out=1 \rangle$,

где $\langle out \rangle$ – требование исправности элемента out

Пусть в базе знаний системы хранятся факты: $\langle in_1=0 \rangle \{ \}$, $\langle in_2=0 \rangle \{ \}$

Задается предположение об исправности элемента out .

Тогда из этих фактов при помощи дедуктивного логического вывода выведутся новые факты: $\langle out=0 \rangle \{ out \}$ и $\langle out=1 \rangle \{ out \}$.

Теперь какое бы показание не было бы снято с элемента out оно всегда вступит в противоречие с одним из полученных фактов, а следовательно множество предположений $\{ out \}$ будет признано противоречивым, а значит, элемент out – неисправным, а, вообще говоря, это не так.

Иными словами, контроль непротиворечивости становится ключевой проблемой при проектировании системы логических правил.

Утверждение 1. Система правил, построенная согласно алгоритму, данному ранее, является дедуктивно непротиворечивой.

Доказательство. Пусть это не так, то есть существуют два правила вида:

Правило 1: $\langle a_1 \rangle \dots \langle a_n \rangle \mid \langle 0 \rangle$

Правило 2: $\langle a_1 \rangle \dots \langle a_n \rangle \mid \langle 1 \rangle$

Следовательно, они построены на основе двух разных строк таблицы истинности вида: то есть, $f(a_1, \dots, a_n) = 0 \neq 1 = f(a_1, \dots, a_n)$ - противоречие. А это и значит, что система правил непротиворечива.

Иными словами, при генерации системы правил, соответствующих логической схеме, беспокоиться о дедуктивной непротиворечивости не нужно.

Утверждение 2. Система правил, построенная согласно алгоритму, данному в ранее, может быть абдуктивно противоречивой.

Доказательство. Рассмотрим пример – система правил, построенная на основе элемента “И”.

Элемент “И” построен на основе функции $F=f(x,y)=x \& y$.

Построим систему правил:

Правило 1: $\langle x=0 \rangle \& \langle y=0 \rangle \& \langle \text{И} \rangle \rightarrow \langle F=0 \rangle$

Правило 2: $\langle x=0 \rangle \& \langle y=1 \rangle \& \langle \text{И} \rangle \rightarrow \langle F=0 \rangle$

Правило 3: $\langle x=1 \rangle \& \langle y=0 \rangle \& \langle \text{И} \rangle \rightarrow \langle F=0 \rangle$

Правило 4: $\langle x=1 \rangle \& \langle y=1 \rangle \& \langle \text{И} \rangle \rightarrow \langle F=1 \rangle$

Видно, что пары правил (1,2), (1,3) являются абдуктивно противоречивыми, а это и значит, что вся система правил абдуктивно противоречива.

Единственным решением этой проблемы на данный момент видится только запрет на абдуктивный вывод для подобного рода правил. Действительно, интуитивно, когда мы видим на одном из входов и выходе элемента “И” нули, мы ничего не можем сказать о сигнале, поданном на другой выход.

Заключение. В данной статье был теоретически описан механизм решения задачи диагностики при помощи системы поддержки истинности, основанной на предположениях. Также были рассмотрены некоторые проблемы, возникающие при реализации ATMS на автоматических системах, и подходы к решению этих проблем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Reiter R.A.* Theory of Diagnosis from First Principles // Artificial Intelligence, 1987. – Vol. 32. – P. 97-130.
2. *Вагин В.Н., Оськин П.В.* Эвристические и вероятностные методы снятия эффективных показаний в системах диагностики // Известия РАН. Теория и системы управления. – 2006. – № 4. – С. 78-93.
3. *Подиновский В.В., Констанденко О.С.* Исследование метода последовательных уступок // Методы анализа и реконструкции сложных систем. Тезисы доклада. – Рига: Зинатне, 1972. – С. 181.
4. *Forbus K.D., de Kleer J.* Building Problem Solver // A Bradford Book. Cambridge, Massachusetts, London, England: The MIT Press, 1993.
5. *De Kleer J.* An Assumption-based TMS // Artificial Intelligence. – 1986. – Vol. 28. – P. 127-162.
6. *Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В.* Достоверный и правдоподобный вывод в интеллектуальных системах. Монография / Под ред. В.Н. Вагина и Д.А. Поспелова. – М.: 2-ое изд. исправл. и дополн. – Физматлит, 2008. – 714 с.
7. *Teow-Hin Ngair, Gregory Provan.* Focusing ATMS Problem-solving: a formal approach // Computer and Information Science Department University of Pennsylvania Philadelphia, 1992.

Вагин Вадим Николаевич

Московский Энергетический Институт (Технический Университет).

E-mail: vagin@appmat.ru.

143400, г. Красногорск, Московская область, ул. Ленина, 47, корп. 3, кв. 137.

Тел.: +79152399103.

Кафедра прикладной математики; профессор.

Зарецкий Дмитрий Сергеевич

Московский Энергетический Институт (Технический Университет).

E-mail: zardim@yandex.ru.

143992, г. Балашиха, Московская область, мкр. Заря, ул. Садовая, 8, корп. 3, кв. 10.

Тел.: +79165788560.

Кафедра прикладной математики; студент.

Vagin Vadim Nikolaevich

Moscow Power Engineering Institute (Technical University).

E-mail: vagin@appmat.ru.

47/3 ap. 137, Lenina Street, Krasnogorsk, Moscow Region, 143400, Russia.

Phone: +79152399103.

The Department of Applied Mathematics; Professor.

Zaretsky Dmitry Sergeevich

Moscow Power Engineering Institute (Technical University).

E-mail: zardim@yandex.ru.

8/3, ap. 10, Sadovaya Street, building estate Zarya, Balashiha, Moscow Region, 143992, Russia.

Phone: +79165788560.

The Department of Applied Mathematics; Student.

УДК 519.816.

С.М. Ковалев, С.А. Кулькин, В.А. Терновой

**ФОРМИРОВАНИЕ ТЕМПОРАЛЬНЫХ СЕТЕВЫХ МОДЕЛЕЙ НА ОСНОВЕ
ГРАФОВЫХ ПОКРЫТИЙ***

Рассматривается подход к формированию минимальных темпоральных пирамидальных сетей (ТМПС) в базах данных временных рядов. ТМПС является сетевой моделью, в структуре которой объединены образы всех сценариев, предшествующих вхождению символа q во временной ряд (ВР) S . Изложены алгоритмы формирования ТМПС, использующие ряд количественных характеристик, отражающих потенциальную полезность применения той или иной группы темпоральных отношений в качестве обобщающих или детализирующих признаков.

Растущая пирамидальная сеть; временной ряд; темпоральные правила; база данных; сетевая модель.

S.M. Kovalev, S.A. Kulkin, V.A. Ternovoy

**FORMATION TEMPORAL NETWORK MODELS ON THE BASIS OF GRAPH
COVERINGS**

In article the approach to formation minimal temporal pyramidal networks (TMPN) in databases of time numbers is considered. TMPN is network model in which structure images of all scenarios previous occurrence of a symbol q in a time number (TN) S are united. The algorithms of formation TMPN using a number of quantitative characteristics, reflecting potential utility of application of this or that group temporal relations as generalising or detailing signs are stated.

Growing pyramidal network; time number; temporal rules; Database; Network model.

Эффективным инструментом формирования обобщенных описаний данных, имеющих темпоральные особенности и представляемые в виде временных рядов (ВР) $S = (s(t_1), s(t_2), \dots, s(t_n))$, являются темпоральные сетевые модели (ТМПС) [3].

* Работа выполнена при поддержке РФФИ: (гранты № 10-01-00058, № 10-07-00158).