

ОРГАНИЗАЦИЯ ВЫВОДА РЕШЕНИЙ В БАЗЕ ЗНАНИЙ ИНСТРУМЕНТАЛЬНОЙ ОБОЛОЧКИ ДЛЯ СОЗДАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ САПР

Григорьев А.В.
Кафедра ПМиИ, ДонГТУ
grigorie@r5.dgtu.donetsk.ua

Abstact

Grigoriev A.V. Decisions inference organization in knowledges base of instrumental envelope for creation intellectual CAD. Consider the knowledges conception peculiarities in instrumental envelope. Describes a decisions inference organization method.

Введение

В работе [1] был определен метод организация пространства поиска решений в специализированной инструментальной оболочке для создания интеллектуальных САПР - мета-эвристической оболочки (МЭО). Ранее в работе [2] был определен ряд принципов построения МЭО.

К главным особенностям МЭО относится:

- 1) Использование семиотической модели [3] при представлении концептуальной модели предметной области МЭО;
- 2) Ориентации применяемой концептуальной модели ПрОб (КМ ПрОб) на физическую семантику ПрОб;
- 3) Ориентация при обучении базы знаний комплекса на ограниченное количество имеющихся в наличии апробированных на практике моделей – прототипов сложных объектов данной предметной области;
- 4) Задание признаков объектов (генотипа) как идентификаторов подмножества структурных элементов, явно определяющих его структуру (т.е. фенотип);
- 5) Грамматический подход к представлению описаний моделей.

Ранее были изложены основные принципы представления знаний в МЭО. Целью данной работы является общее изложение организации вывода в базе знаний МЭО.

Решение данной задачи позволит сформировать более полное теоретическое обоснование для построения специализированной инструментальной оболочки создания интеллектуальных САПР.

1 Краткое изложение метода представления знаний в МЭО

В соответствии с работой [1] кратко изложим метод организации представления знаний в МЭО. Принятая в МЭО КМ ПрОб предполагает наличие явно введенной внутренней границы блока (включающей как подмножество и внешнюю границу блока), что позволяет рассматривать внутреннюю среду блока как замкнутый одноуровневый набор связей.

Предлагаемый путь представления знаний основан на выполнении теоретико-множественных операций (ТМО) - объединение, пересечение, дополнение и разность - над имеющимися прототипами как совокупностями "обобщенных" связей внутренней среды. Т.о. обучение, т.е. формирование базы знаний, выполняется автоматически, без участия пользователя.

Рассмотрим пример. Пусть имеется некоторое множество прототипов, входящих в тип блоков $A: A'=(P_1 \vee P_2 \vee P_3)$. На рис. 1 показана обобщенная схема типа A , построенная в рамках предлагаемой концептуальной модели.

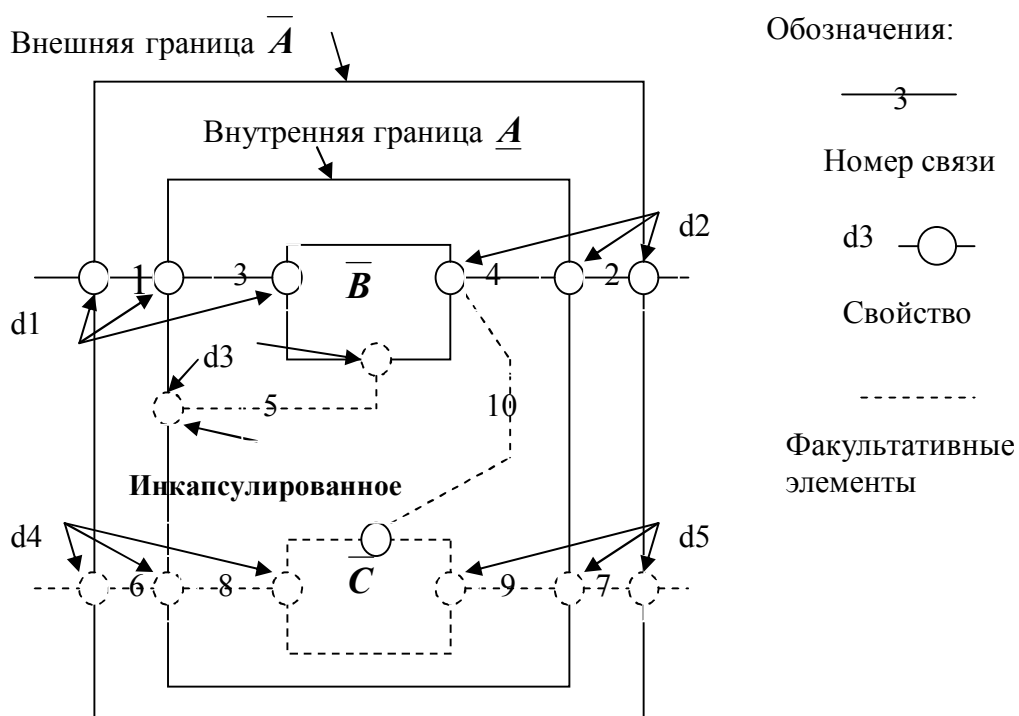


Рисунок 1 - Обобщенная схема типа A

Результат выполнения теоретико-множественных операций над совокупностями "обобщенных" связей (далее - просто связей), образующими данные прототипы, составляет ряд подмножеств $@j$, определяющих некоторые части внутренней среды прототипов. При этом:

$$P_1=@1\&@2; P_2=@1\&@2\&@3; P_3=@2\&@3;$$

$$@1=5; @2=1\&2\&3\&4; @3=6\&7\&8\&9\&10.$$

Тут 1,2... - сквозные номера оригинальных связей, определенные в пределах всего типа блоков. Преобразуем множество прототипов А к форме И-ИЛИ-дерева:

$$A'=(P_1 \vee P_2 \vee P_3) = @2 \& N1; N1=@1 \vee @3 \vee N2; N2=@1\&@3.$$

На рис. 2 изображено полученное И/ИЛИ дерево. В скобках показаны номера прототипов, входящих в данную вершину, числами заданы номера связей, стрелками показан порядок декомпозиции узлов.

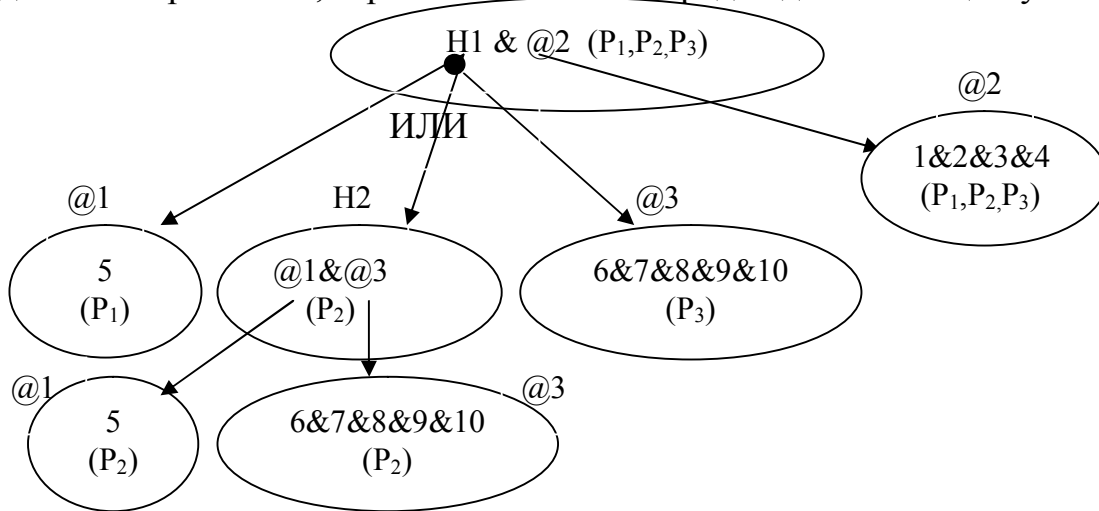


Рисунок 2 - Форма И-ИЛИ-дерева

Номера прототипов в вершинах неявно задают продукционные зависимости для ИЛИ-узлов, используемые при выводе с целью синтеза (выбора) необходимых прототипов в САПР. Семантика зависимостей такова: "если в данном ИЛИ-синтерме (узле) удалить вариант (признак), включающий прототип К, то прототип К должен быть удален во всех прочих узлах". С учетом названных зависимостей А может быть определен как идентификатор модуля знаний о структуре внутренней среды объектов типа А.

Детально механизм вывода описан в [2]. Кратко суть вывода в таких условиях может быть охарактеризована следующими положениями:

- множество обликов возможных систем, составляющих возможные технические задания – это множество комбинаций альтернатив всех ИЛИ-узлов;
- синтез решения, т.е. задание ТЗ производится в диалогом режиме;
- для оптимизации процесса вывода используется специальный критерий выбора следующего ИЛИ-узла – вопроса, обеспечивающий сокращение длительности процесса задания ТЗ и снижения сложности вопросов, задаваемых пользователю.

Общий подход к синтезу гипотез предполагает, что автоматически формируется декартово произведению всех составляющих всех ИЛИ-синтермов, входящих в И-ИЛИ-дерево.

Описанный выше порядок вывода может быть охарактеризован как монотонный вывод без предварительных допущений [4], на базе одной модели мира.

Недостатком данного изложения является отсутствие:

1) Возможности прямой передачи экспертом знаний о технологии проектирования объектов данного типа как совокупности эвристических правил, т.е. возможности прямого ввода экспертом в базу знаний множества продукций такого типа, т.е. продукций, работающих с грамматиками;

2) Описания общего порядка организации вывода на множестве модулей знаний.

Разрешим данные проблемы.

2. Определение общего подхода к построению базы знаний

Считаем, что база знаний ориентирована на решение задач конструирования.

Все средства организации вывода адаптируются к специфике модели САПР решения типичных задач проектирования и отражают модифицированную модель САПР, заданную ранее в теории сложности САУ [2].

Общий подход к организации базы знаний как множеству эвристических правил - продукций, работающих с грамматиками, базируется на организации немонотонного вывода, т.е. вывода, допускающего отрицание фактов, ранее определенных как истинные [4,5]. Релаксация вывода, т.е. переход базы знаний в состояние, когда все условия соблюдены [5], достигается предлагаемым ниже порядком выполнения продукций, обеспечивающим сходимость процесса вывода. Цель вывода, т.е. релаксации - соотнести техническое задание (ТЗ) со структурой искомого решения.

База знаний, как совокупность продукций строится экспертным путем и задает "поверхностные" знания [5]. Т.е. продукции представляют собой эвристические правила, связывающие абстрактные категории данных (вариант из текущего состояния модели структуры объекта как возможное техническое задание, вводимое пользователем) с абстрактными категориями решений (добавить в схему фрагмент, удалить фрагмент).

"Глубинные знания", т.е. знания о поведении и свойствах объекта, заданного структурой, выражаются через совокупность функций базовых структурных блоков и проявляются только после синтеза структуры уже в

ходе моделирования. До этого "глубинные знания" определяются только через признаки структуры объекта.

По-прежнему в качестве основы базы знаний рассматривается И-ИЛИ-дерево, описывающее возможные комбинации групп связей. Но прототипы, лежащие в основе БЗ, *явно не рассматриваются*, и номера прототипов явно не входят в альтернативы синтермов. Неявные зависимости - продукции, составляющие основу вывода в ранее рассмотренной модели представления знаний, задаются в этом случае уже как *явные* продукции. Цель таких *явных* продукций, в данном случае, это *неявно* задать возможные допустимые прототипы через разрешенные комбинации частей.

Данные продукции связывают по некоторому алгоритму номера альтернатив, входящие в ИЛИ-синтермы, составляющие посылку продукции, с номерами альтернатив, входящих в ИЛИ-синтермы, составляющие вывод продукции.

Типичная форма зависимости посылки и вывода:

ЕСЛИ данный вариант *существует* в посылке,

ТО *существует* и соответствующий вариант в посылке.

Возможные варианты:

- "существует" - "не существует";
- "не существует" - "существует";
- "не существует" - "не существует".

ИЛИ-синтермы, входящие в посылку или вывод продукций, задают фрагмент или группу фрагментов описания модели, входящие в базу знаний.

Посылка, как правило, явно привязана к текущему состоянию модели, формируемой в процессе синтеза. Однако возможным считается режим, при котором, косвенно подразумевая наличие посылки в текущем состоянии модели, тем не менее, посылка привязана только к диалогу с пользователем, задающем ТЗ.

Т.о., продукция есть совокупность ряда "простых" продукций связывающих зависимость множеств альтернатив посылки и вывода, что позволяет условно отнести такие продукции к макропродукциям.

Отличные исходные допущения, задаваемые как контексты - "среды обитания", задающие гипотетические модели мира [6], могут формироваться на основе продукций, требующих первоначально наличия в куче некоторого набора вариантов структуры в соответствии с выбором предметной области, задаваемой пользователем как фрагмент ТЗ.

2.1. Модуль знаний

Основой представления знаний есть модуль знаний - т.е. локальная БЗД о структурном блоке. Модуль включает знания о:

- внешней среде блока;
- внешней границе блока;
- связи внешней и внутренней границы блока;
- связи внутренней границы блока и внутренней среды.

"Типичный" процесс синтеза структуры блока по модулю знаний состоит в построении его структуры, исходя из:

- заданной внешней среды;
- дополнительной информации об особенностях самого блока.

Этапы синтеза по модулю:

- формируется внутренняя среда блока-модуля как список связей;
- связи, с помощью статической базы данных, где хранятся именованные структуры (границы, среды блоков), формируют список блоков, входов-выходов и список их агрегаций (массивов);
- если структурный блок базовый, то по составу входов-выходов блока, по типу блока и его назначению (принадлежности к агрегации, т.е. массиву) формируется вариант внутренней функции блока.

2.2. Метод представления знаний в модуле знаний

База знаний это:

1) И-ИЛИ-дерево, т.е. упорядоченное множество синтермов по "И" и по "ИЛИ", где все синтермы имеют адрес в И-ИЛИ-дереве;

2) Множество семантических двунаправленных зависимостей между ИЛИ-синтермами, описывающими их совместимость между собой, т.е. правил вывода или продукций.

Примечание 1.

Имеется фиксированный порядок просмотра или нумерации синтермов по ИЛИ в порождающей грамматике как при подсчете вариантов структур в порождающей грамматике так и при формировании вопросов к пользователю (проектировщику), т.е. "спрашивании" у пользователя. Допускается любой возможный порядок: сверху-вниз и справа-налево или любой другой.

2.3. Куча

Наименование "куча" по отношению к элементам базы знаний взято по аналогии к известному примеру конструирования требуемой башни из ряда деталей, сложенных в кучу [].

В нашем случае база знаний - это:

1) "Полная" куча, т.е. список всех синтермов по ИЛИ, т.е. альтернатив в их исходном, полном виде, взятых из исходного И-ИЛИ-дерева, при том, что:

- все альтернативы имеют адреса, по которым можно найти их положение в И-ИЛИ-дереве;
- по их абсолютным адресам в И-ИЛИ-дереве формируется шкала альтернатив;
- И-синтермы не входят в кучу;
- куча формируется при начале вывода как копия всех ИЛИ-синтермов;
- 2) Множество продукций;
- 3) "Пустая" куча, т.е. список имен ИЛИ-синтермов, у которых нет альтернатив; предназначена для заполнения конкретными решениями при синтезе, т.е. при работе множества продукций.

Примечание 2.

Имена пустых ИЛИ-синтермов могут порождаться самими продукциями. В этом случае "пустая" куча первоначально действительно пуста.

2.4. Возможные формы продукций

Продукции могут иметь различные значения четырех параметров, определяющих механизм управления их срабатыванием. Эти параметры задают отношения между различными составляющими продукций и системы управления выводом, т.е. - посылки, вывода и кучи. Возможные отношения:

- 1) посылки с "кучей" - проверяющие и не проверяющие состояние кучи ("опускающие голову" и "не опускающие");
- 2) посылки с пользователем - спрашивающие и неспрашивающие;
- 3) вывода с "кучей" - удаляющие и прибавляющие;
- 4) посылки и вывода - однонаправленные и двунаправленные.

Допустимы любые формы продукций, которые можно получить, исходя из декартового произведения значений атрибутов:

- (оп, неоп)*(спр, неспр)*(удал, приб)*(одно, дву).

Т.е. возможно 16 типов продукций.

Тип продукции является ее динамической характеристикой и определяется в зависимости от текущего режима вывода.

Потенциально все продукции:

- проверяющие (опускающие);
- спрашивающие;
- двунаправленные.

Характеристика продукций "прибавляющие" или "удаляющие" определяется в зависимости от операции, выполняемой выводом продукции в куче.

Задача выбора типа продукций может ставиться следующим образом. *Дано:*

- информация о проектируемом объекте у пользователя, т.е. ТЗ, само ТЗ может быть не полным или противоречивым, или же не иметь соответствия в БЗД;

- предпочтительный режим работы для пользователя (диалоговый, пакетный), требующий применения тех или иных критериев управления выводом.

Определить:

- тип продукции.

3. Общий порядок вывода на множестве модулей знаний

3.1. Порядок вывода на множестве модулей знаний

Деление процесса вывода:

- дерево модулей знаний;
- отдельный модуль знаний;
- этап, как часть модуля, как определение в описании структуры блока группы отношений некоторого типа (т.е. таблица);
- шаг, как часть этапа.

3.2. Вывод по дереву модулей

Вывод по дереву модулей знаний:

- при синтезе имеет смысл рассосредоточить процесс вывода по отдельным блокам-модулям знаний;
- выбор модулей может осуществляться в соответствии с их порядком следования в дереве модулей, где они структурированы по включению друг в друга; движение по дереву модулей может осуществляться, например, по принципу слева-направо и сверху-вниз.

3.3. Процесс вывода над кучей в пределах модуля знаний

Собственно процесс вывода прототипа - это упорядоченный процесс прохода по синтермам ИЛИ, целью которого является задание ТЗ пользователем и проверка введенного ТЗ на семантическую и синтаксическую корректность.

Порядок выбора синтермов для просмотра и отработки и связанных с ними продукций определяется критериями управления выводом.

Имеется два варианта работы с кучей:

- 1) Сначала ТЗ, затем вывод;
- 2) Совмещение ввода ТЗ и процесса вывода.

Рассмотрим их отдельно.

3.3.1. Сначала ТЗ, затем вывод

Этап задания пользователем ТЗ предшествует процессу вывода, т.е. проверки семантической корректности. При этом пользователь формирует ТЗ как ответы на все вопросы, касающиеся особенностей объекта:

◆ *заданные явно* (предполагает проход по всем ИЛИ-синтермам в "полной" куче); назовем этот режим задания ТЗ диалогово-пакетным; он подразумевает работу системы вывода в режиме использования всех продукций "неопускающий" - "спрашивающий";

◆ *незаданные, но подразумевающиеся возможными* (т.е. в заранее известной грамматике пользователь задал известные альтернативы по известным же синтермам по ИЛИ); назовем этот режим задания ТЗ пакетным; подразумевает работу системы вывода в режиме "неопускающий" - "неспрашивающий".

В случае, если пользователь задал ТЗ как совокупность ответов в диалогово-пакетном режиме, то с интаксическая правильность ответов гарантируется самой системой вывода.

В случае, если пользователь задал ТЗ в пакетном режиме, то необходим предварительный этап проверки синтаксической корректности ТЗ, цель которого - выяснение вопросов:

- есть ли такие ИЛИ-синтермы – вопросы в И-ИЛИ-дереве;
- есть ли в них такие термы - ответы.

В случае синтаксической верности ТЗ проводится собственно вывод, т.е. проверка семантической верности ТЗ. "Пустая" куча модифицируется в процессе вывода путем отработки продукций и редукции альтернатив, не соответствующих заданным в продукциях отношениям между альтернативами. В этом случае подразумевается работа системы вывода в режиме "опускающий"-"неспрашивающий".

3.3.2. Совмещение ввода ТЗ и процесса вывода

Задание ТЗ проводится постепенно, с проведением промежуточной проверки семантической корректности полученных от пользователя частных ответов, т.е.:

- в соответствии с критериями вывода производится выбор некоторого ИЛИ-синтерма из кучи; к этому времени по причине отработки продукций, связанных с выбором и сужением ряда предыдущих ИЛИ-синтермов, число альтернатив в данном ИЛИ-синтерме сужено;
- для оставшихся альтернатив в данном ИЛИ-синтерме организуется диалог с пользователем для уточнения ТЗ в связи с возможными вариантами данной альтернативы;

- после получения ответа запускается отработка всех продукций, прямо или косвенно связанных с данным синтермом; таким образом сужаем ряд прочих альтернатив;

Управление выводом в этом случае задается следующим рядом мета-продукций, специфичных для данного случая:

1) ЕСЛИ после отработки всех продукций выполнилось одно из условий

((все альтернативы –ИЛИ-синтермы имеют один вариант, т.е. отсутствует неопределенность) И (пользователю больше нечего добавить, т.е. все ТЗ уже задано))

ИЛИ

(пользователь дал ответ "больше ничего не знаю", т.е. ТЗ исчерпалось еще до конца процесса редукции неопределенности по ИЛИ в БЗД)

ТО достигнут "благоприятный" результат вывода, т.е. получено одно или ряд решений, соответствующих поставленному ТЗ.

2) ЕСЛИ после отработки всех продукций выполнилось одно из условий

((одна или несколько альтернатив не имеют ни одного варианта, т.е. ТЗ задает решение, отсутствующее в БЗД) И (пользователь не закончил ввод ТЗ))

ИЛИ

(пользователь дал ответ "такого варианта в предложенном наборе нет") И (ТЗ не исчерпалось до конца)

ТО достигнут "неблагоприятный" результат вывода, т.е. не получено ни одно решение, соответствующее поставленному ТЗ;

3) ЕСЛИ ни одно из предыдущих условий не выполнилось

ТО идем на начало алгоритма, т.е. на выбор ИЛИ-синтерма.

3.4. Управление переходами между отдельными этапами

Предполагает такие случаи переходов:

- сработало условие С, что можно выполнять продукции данного этапа (т.е. этап синтеза типов подблоков, синтеза массивов типов подблоков и т.д. по связанным таблицам в БД);

- отработали все продукции данного этапа;

- сработал запрет на выполнение продукции данного этапа в дальнейшем (C=false) и разрешение срабатывание продукции следующего этапа.

3.5. Управление выводом продукции разных типов на текущем этапе

1) Крупные шаги:

- сначала все прибавляющие продукции, работают в любом порядке до тех пор, пока полный их цикл ничего не изменит в куче;
- затем работают удаляющие продукции, в цикле, в произвольном порядке, но один раз.

2) Мелкие шаги:

- среди прибавляющих продукции сначала идут "не опускающие голову", а затем "опускающие голову";
- в режиме спрашивания управлять спрашиванием с помощью критериев оптимизации процесса вывода с точек зрения "экономии ресурсов машины" или "экономии времени человека".

Предлагаемый порядок призван обеспечить *убедительность* вывода [4], т.е. отсутствие зацикливаний в цепочке вывода.

Заключение

В данной работе получены следующие результаты:

- изложено более полное изложение предлагаемого метода представления знаний в специализированной оболочке для построения интеллектуальных САПР;
- сформулированы положения о общем порядке организации вывода, ориентированном на предлагаемый метод представления знаний.

Таким образом, поставленная задача выполнена. Полученные результаты дают более полное представление о методах организации базы знаний в МЭО.

В качестве перспективных направления исследований, не раскрытых в данной статье, можно назвать:

1) Задание конкретных алгоритмов задания множества семантических зависимостей над ИЛИ-синтермами в дереве при определении множества имеющих смысл "производных" прототипов;

2) Детальное определение критериев оптимизации процесса вывода с точек зрения "экономии ресурсов машины" или "экономии времени человека";

3) Определение конкретных формы организации диалога с пользователем при задании ТЗ.

Литература

1. *Григорьев А.В.*. Организация пространства поиска решений в специализированной оболочке для создания интеллектуальных САПР. Вісник ТРТУ-ДонГТУ. Матеріали 2-го міжд. н.-т. семінара "Практика и перспективы институционального партнерства". Донецк, ДонГТУ, 2001, N1, С.57-68.
2. *Григорьев А.В.* Семиотическая модель базы знаний САПР. Научные труды Донецкого государственного технического университета Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 10: Донецк, ДонГТУ, 1999. — С. 30-37.
3. *Поспелов Д.А.* Ситуационное управление: теория и практика. М.: Наука, 1986. - 288 с.
4. *Джексон П.* Введение в экспертные системы. М.: ИД Вильямс, 2001.
5. *Doule J.* A truth maintenance system. 1979. Artificial Intelligence, N12, p.231-272.
6. *De Kleer J.* An assumption based TMS. 1986. Artificial Intelligence, N28, p.127-162.

Дата надходження до редколегії: 20.12.2003 р.