

ОРДЕНА ЛЕНИНА
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ им. М.В. КЕЛДЫША
РОССИЙСКОЙ АКАДЕМИИ НАУК

Корухова Л. С., Любимский Э. З., Малышко В. В.

ПРОГРАММНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ
АССОЦИАТИВНОГО ПЛАНИРОВАНИЯ

Москва
2002

Аннотация

Работа посвящена исследованию средств программирования для решения задач планирования целенаправленной деятельности. Описана реализация ассоциативного планирования в решателе геометрических задач. Предложен алгоритм поиска *близких ситуаций*, основанный на компиляции описаний стереотипных ситуаций в сеть специального вида. Рассмотрены критерии отбора стереотипов для получения эффективного плана. Приведены решения интеллектуальных многошаговых задач, демонстрирующие метод ассоциативного планирования на основе стереотипов и близких ситуаций.

Ключевые слова и фразы:

планирование решений, стереотип, решатель геометрических задач, граф решения, ситуация, ассоциативное планирование.

L. S. Korukhova, E. Z. Lubimsky, V. V. Malyshko

Programming tools of associative planning realization

Preprint, Inst. Appl. Math., the Russian Academy of Science

Abstract

This paper concerns research on new programming tools for solving some planning problems of goal-oriented activity. By example of geometric problem solver we describe the realization of the associative planning method for problem solving. We propose the algorithm of resembling situations searching, which is based on compiling of stereotype situation's descriptions to the special net. The criteria of stereotypes' selection for effective planning are discussed. The paper contains solutions of intellectual multi-step problems as examples of associative planning method's application.

Key words and phases:

problem solving, stereotype, geometric problem solver, solution graph, situation, associative planning.

Введение¹

Настоящая работа является продолжением работ (см. [Корухова и Любимский, 1994], [Вакин, Корухова и др., 1997], [Корухова и др., 2000]), посвященных исследованию эффективных методов планирования для решения интеллектуальных задач. При разработке таких методов принципиальным является то, что процесс планирования должен основываться на тех приемах и методах, которые использует человек, решая сложную задачу. В работе рассматривается проблема реализации ассоциативного планирования.

Ассоциативное мышление и рассуждение по аналогии применяются человеком при составлении планов и решении задач наряду с логическим анализом и эвристическим поиском. Внимание разработчиков ранних систем искусственного интеллекта (ИИ) было сосредоточено на последних двух методах решения сложных задач. Многие системы базировались именно на этих методах, ставших классическими методами ИИ. С течением времени из-за недостатков, присущих логике и эвристическому поиску, часть исследователей занялась методами, основанными на аналогии. Привлекательность этих методов состоит в том, что при решении задач используется прежний опыт системы, а также система шире использует знания, содержащиеся в ее базе.

Основным направлением в этой области является исследование *рассуждений на основе прецедентов (case-based reasoning)*. Этот метод рассуждений используется для автоматического планирования в различных предметных областях (доказательстве теорем, юриспруденции, поиске маршрутов по карте, программном управлении заводским станком). Суть метода состоит в том, что система хранит базу *прецедентов* (полученных ранее решений задач), приступая к работе над новой задачей, система просматривает базу в поисках решения похожей задачи и, обнаружив таковое, пытается адаптировать к новым условиям и применить для получения ответа. В случае успеха удается достаточно быстро получить результат, поскольку модификация готового решения зачастую оказывается проще, нежели его поиск «с нуля».

В качестве примера рассмотрим использующую прецеденты систему CL^AM , которая осуществляет автоматическое доказательство теорем

¹ Работа выполнена на кафедре системного программирования факультета ВМиК МГУ им. М. В. Ломоносова при поддержке Российского фонда фундаментальных исследований (проект № 99-01-00306).

(см. [Melis and Whittle, 1997]). Прецедентами в системе являются планы доказательства теорем. Для построения плана целевой теоремы подыскивается подходящий прецедент (например, теорема $\text{sum}(x)+\text{sum}(y)=\text{sum}(x+y)$ может быть доказана аналогично доказательству теоремы $x \leftrightarrow (y \leftrightarrow z) = (x \leftrightarrow y) \leftrightarrow z$), затем строится отображение исходной теоремы в целевую. На основании этого отображения осуществляется преобразование прецедента в план для целевой теоремы. При этом он может быть переформулирован, в доказательство могут быть добавлены новые шаги.

Процедура планирования системы CL^AM представляет собой последовательность шагов:

1. Найти в базе прецедентов теорему, похожую на целевую (похожесть определяется по виду формул теоремы).
2. Построить отображение теоремы-прецедента в целевую.
3. Расширить отображение, сопоставляя правила исходного доказательства и правила для доказательства целевой теоремы.
4. Принять решение о переформулировании доказательства-прецедента (на основании образцов из отображения).
5. Применить для целевой теоремы шаги доказательства, следуя исходному плану.

Рассуждения на основе прецедентов не всегда успешны. Для ряда задач система CL^AM строит планы с изъянами – участками планов, которые невозможно уточнить полностью при помощи аналогии. Изъяны в планах могут быть устранены с помощью обычной техники доказательства (с привлечением дедуктивного метода доказательства).

Основные проблемы при реализации систем, основанных на прецедентах:

- сопоставление (поиск подходящих прецедентов в базе);
- адаптация прецедентов к условиям задачи;
- пополнение базы прецедентов;
- привлечение других методов планирования в случае неудачи рассуждений по аналогии.

В нашей работе предлагается метод планирования на основе аналогии – ассоциативное планирование, и рассматривается его реализация в решателе геометрических задач (РГЗ). Метод имеет некоторое сходство с рассуждениями на основе прецедентов, но ряд отличий между ними позволяет говорить о нем, как о новом методе планирования решений.

В первой главе дается описание идеи метода, вводятся основные понятия. Во второй главе описана структура и реализация сопоставителя,

обеспечивающего полное и частичное сопоставление образцов. Реализация ассоциативного уточнения цели описана в третьей главе. Завершает работу описание процесса решения геометрической задачи при использовании метода ассоциативного планирования.

1. Ассоциативное планирование с использованием близких ситуаций

Мы рассматриваем систему планирования, основанную на *стереотипах*. *Стереотипы* – это непроцедурные конструкции для представления стереотипных приемов решения задач. Эффект применения стереотипов в системе похож на применение правил в системах продукций, но проблемно-ориентированная активация и назначение стереотипа существенно отличают его от правил продукционной системы. Используя термин *стереотип*, мы исходим из следующей модели планирования человеком своей деятельности. Человек, решая сложную задачу, анализирует сложившуюся ситуацию и предпринимает стереотипные действия, которые у него с данной ситуацией ассоциируются. Применение действий приводит к изменению состояния задачи. В новых условиях применяются другие стереотипные приемы решения. Процесс применения стереотипов является циклическим и довольно быстро (в сравнении с полным перебором вариантов) приводит к искомому решению, либо к убеждению, что такого решения нет.

Представление для стереотипного приема решения состоит из двух частей: ситуации и действия стереотипа. Ситуация стереотипа – это совокупность условий, накладываемых на состояние предметной области. Описание ситуации содержит переменные, оно является образцом, который сопоставляется с текущим состоянием предметной области. Действие стереотипа – это последовательность элементарных действий, которые можно применить, если выполняются условия ситуации. Мы исходим из того, что состояние предметной области описывается совокупностью фактов и хранится в базе знаний системы, а элементарными действиями являются запись фактов в базу, удаление фактов из базы, добавление новых целей и связей между целями. Таким образом, стереотип может представить знания о правилах вывода фактов и правилах разбиения целей на подцели.

Метод ассоциативного планирования сочетает в себе редукцию целей и рассуждения на основе прецедентов. Ассоциативное планирование предполагает итеративное разбиение целей на подцели и построение графа

целей (который можно рассматривать как альтернативное представление И/ИЛИ-графа). Разбиение на подцели производится с привлечением аналогии. Идея метода такова: если при уточнении цели нельзя применить стереотип из-за нарушенных условий в его ситуации, то следует адаптировать стереотип к текущему состоянию задачи, и затем применить его действие.

1.1. Близкая ситуация

Базовым понятием ассоциативного планирования является *близкая* (частично-распознанная) ситуация. Ситуация стереотипа является близкой, если при ее сопоставлении с текущим состоянием задачи нарушается часть ее условий, причем только те условия, для которых есть потенциальная возможность вывода фактов, недостающих для полного сопоставления ситуации. Если ситуация стереотипа является близкой, то имеет смысл модифицировать стереотип в соответствии с текущим состоянием задачи. Адаптация стереотипа состоит в том, что для каждого нарушенного условия в действие стереотипа добавляется оператор, записывающий в граф цель, по достижении которой условие будет выполнено (например, если нарушено условие «длина стороны треугольника известна», то следует добавить оператор «запиши цель „найди длину стороны треугольника“»). В других случаях (когда ситуация не является близкой) попытка использования стереотипа при построении плана будет безуспешной, так как недостающие факты вывести невозможно, и этот путь решения ведет в тупик. Таким образом, *близость* ситуации показывает, имеет ли стереотип отношение к решению текущей задачи.

В РГЗ, как и в любой системе логического вывода можно выделить класс фактов, которые могут добавляться в ходе решения, и класс фактов, которые не могут быть получены путем вывода. В решателе, не реализовано добавление в ходе решения в базу задачи новых объектов, поскольку это может привести к неоправданному увеличению перебора. Новые отношения, отражающие конфигурацию объектов, тоже не могут возникать в ходе решения. Факты других типов, такие как значения величин геометрических объектов и проблемные отношения между объектами (например, равенство отрезков или углов и т. п.) составляют класс потенциально выводимых фактов в РГЗ. Следовательно, у *близких* ситуаций могут быть не выполнены при сопоставлении только условия на значения величин и условия на проблемные отношения. Заметим, что при расширении класса выводимых фактов должен соответственно меняться класс близких ситуаций. Например, если в решателе будут реализованы дополнительные построения, то отсутствие

в текущем состоянии задачи геометрического объекта не будет означать, что ситуация не является близкой, поскольку недостающий факт будет относиться к выводимым.

Рассмотрим пример. Пусть в базе знаний есть такое описание стереотипа²:

Имя стереотипа: *треугольник_и_окружность_4*

Комментарий: *Сторона треугольника является диаметром описанной окружности, если она равна диаметру окружности.*

Ситуация: *fsit242*

Описание цели: *[\$диаметр окружности\$, [\$s1\$, \$.o\$]]*

Действие: *a([\$доказательство\$, 42, [\$s\$, \$.s1\$, \$.o\$, \$.tr\$]], 0)*
(записать факт, используя доказательство 42)

Трудность: *0*

Описание ситуации, в которой применимо действие стереотипа:

Ситуация: *fsit242*

Объекты: *obj(\$.o\$, \$окружность\$), obj(\$.s\$, \$отрезок\$),*
obj(\$.tr\$, \$треугольник\$)

Элементы: *elem(\$.s1\$, \$.tr\$, \$сторона\$)*

Условия на значения элементов: {в данной ситуации отсутствуют}

Отношения конфигурации: {в данной ситуации отсутствуют}

Проблемные отношения: *prob(\$равны отрезки\$, [\$s\$, \$.s1\$]),*
prob(\$диаметр окружности\$, [\$s\$, \$.o\$]),
prob(\$описана около треугольника\$, [\$tr\$, \$.o\$])

Допустим также, что состояние предметной области описано следующим набором фактов: объект ABC – треугольник, объект O1 – окружность, объект MN – отрезок, O1 описана около ABC, равны отрезки AB и MN. При сопоставлении ситуации fsit242 не выполнено условие, что MN является диаметром O1. Недостающий факт имеет тип проблемного отношения и относится к классу потенциально выводимых в РГЗ. Следовательно, ситуация fsit242 является близкой со строкой сопоставления <.tr-ABC, .s1-AB, .s-MN, .o-O1>. Заметим, что ситуация fsit242 также является близкой со строкой сопоставления <.tr-ABC, .s1-AC, .s-MN, .o-O1>. Недостающих фактов при таком сопоставлении будет два: равны отрезки AC и MN,

² Замечания по синтаксису описаний: знак \$ используется для отметки начала и конца строк; имена, начинающиеся с точки, – это имена переменных.

отрезок MN является диаметром окружности O1. Еще один вариант сопоставления, при котором рассматриваемая ситуация является близкой – $\langle .tr-ABC, .s1-BC, .s-MN, .o-O1 \rangle$.

В зависимости от текущей цели рационально рассматривать применение стереотипа только с одним из трех возможных означиваний. Если стоит цель „доказать, что АВ является диаметром окружности O1“, то только первый вариант сопоставления ситуации fsit242 может считаться подходящим для решения. Только в этом случае стереотип может быть адаптирован для его использования при планировании. Его действия могут быть расширены добавлением еще одного действия – «записать цель „доказать, что MN является диаметром окружности O1“», после чего адаптированный стереотип можно применять для редукции цели.

При ассоциативном планировании каждый используемый стереотип можно рассматривать как своего рода прецедент. Его действие нельзя применить в первоначальном виде, но после адаптации к текущему состоянию предметной области стереотип можно использовать при поиске плана решения. Общим у стереотипа и прецедента является то, что в отличие от правил, которые выбираются на основе *точного* сопоставления antecedента и данных из рабочей памяти, прецеденты и стереотипы выбираются на основе *частичного* сопоставления. Отметим сразу отличия нашего подхода от рассуждений на основе прецедентов:

1. Выше было отмечено, что стереотипы – это образцы, содержащие переменные, тогда как прецеденты являются конкретными решениями, в которых фигурируют константы. Это обстоятельство не позволяет использовать при ассоциативном планировании методы поиска прецедентов (такие как метод *разделяемой сети свойств* [Kolodner, 1993]).
2. Применение стереотипов при ассоциативном планировании организовано в виде итерационного цикла – последовательности шагов, приводящих к решению. Прецедент обычно является приближенным вариантом полного решения, и план строится на основе только одного прецедента.

1.2. Критерии отбора стереотипов для получения эффективного плана

Обычные алгоритмы сопоставления с образцом неприменимы для поиска близких ситуаций, поскольку предназначены для полного, а не частичного сопоставления. Такие алгоритмы пригодны только для нахождения полностью распознанных ситуаций. Для решателя разработан алгоритм поиска близких ситуаций, подробное изложение которого содержится во второй главе работы. Входными данными алгоритма являются факты из базы задачи решателя.

Результатом его работы является список четверок вида: (имя стереотипа, ситуация, строка сопоставления, набор недостающих фактов). На основании этих данных производится адаптация стереотипов и применение их действий, но предварительно производится отбор среди стереотипов с близкими ситуациями. Цель этого отбора – получение эффективного плана.

Критерии отбора стереотипов следующие:

- условие «полезности»;
- условие «доступности»;
- комбинированный критерий.

Условие «полезности» заключается в том, что трудность действия стереотипа, не должна превышать установленного предела. Выполнение данного условия гарантирует, что действие рассматриваемого стереотипа не слишком сложно, то есть применение данного действия не повлечет излишнего усложнения плана.

Условие «доступности» накладывает ограничения, связанные с адаптацией стереотипа. При адаптации стереотипа его действие усложняется – добавляются операторы, связанные с недостающими фактами. Возникает *добавленная* трудность стереотипа, которую можно охарактеризовать суммарной трудностью целей, необходимых для вывода недостающих фактов. «Доступность» стереотипа означает, что вывод недостающих фактов представляется достаточно простым, другими словами, трудность действий, добавленных при адаптации стереотипа, невелика. При проверке данного условия учитывается тип недостающих фактов, поскольку для вывода фактов разных типов ставятся цели различной трудности (как правило, найти площадь треугольника труднее, чем величину его стороны). Например, «доступным» может оказаться стереотип, адаптация которого требует вывода нескольких несложных фактов, а стереотип с одним недостающим «трудным» фактом может быть отвергнут.

Комбинированный критерий позволяет совместно оценить «полезность» и «доступность» стереотипа. При отборе стереотипа по данному критерию осуществляется проверка: лежит ли значение функции взвешивания в априорно заданных пределах. Функция взвешивания представляет собой функцию от двух переменных – основной трудности стереотипа и его *добавленной* трудности. С помощью комбинированного критерия исключаются из рассмотрения стереотипы, для которых значения обеих характеристик близки к предельным. В настоящее время в РГЗ используется следующая

функция взвешивания: $f(x, y) = x + y$ (где x – основная трудность, а y – добавленная трудность).

Конкретные значения пределов трудности, используемые в критериях, устанавливаются априорно – независимо от текущей цели и текущего состояния предметной области. Они определяются классом решаемых задач и являются параметрами, с помощью которых можно управлять работой РГЗ, сужая или расширяя рассмотрение близких ситуаций.

Можно провести некоторые параллели с продукционными системами, в которых отбор применяемых продукций является основным способом управления эффективностью их работы. В таких системах каждый раз запускается не более одной активной продукции, а случае, когда одновременно есть несколько активных продукций, производит отбор лучшей – той, которую выгоднее всего запустить³. Для ряда продукционных систем использование специальных способов отбора продукций позволило повысить их эффективность. Например, одним из таких способов является правило МЕА, предложенное разработчиками экспертной системы R1/XCON [McDermott, 1982].

2. Сопоставитель ситуаций

Распознавание ситуаций стереотипов осуществляется в решателе сопоставителем ситуаций. В основу его реализации положен Rete-алгоритм, использованный в продукционной системе R1/XCON (см. [Forgy, 1982]). Этот алгоритм, как и еще один алгоритм сопоставления с образцом – Treat-алгоритм, предложенный Миранкером для продукционных систем (см. [Miranker and Lofaso, 1991]), осуществляет полное сопоставление. Оба они не пригодны для нахождения близких ситуаций стереотипов. Нахождение неполных означивающих наборов фактов для ситуаций в них не реализовано, поскольку при их создании не ставилось такой задачи. Тем не менее, данные, получаемые при работе алгоритмов полного сопоставления, можно использовать для поиска близких ситуаций. Предлагаемый далее алгоритм поиска близких ситуаций использует данные, накопленные в ходе распознавания ситуаций стереотипов РГЗ. Поэтому начнем рассмотрение с того, как осуществляется распознавание ситуаций.

³ Существуют также параллельные реализации продукционных систем, в которых продукции запускаются параллельно без предварительного отбора.

2.1. Идея Rete-алгоритма

Распознавание ситуаций происходит каждый раз при записи новых фактов в базу задачи. Факты (объекты, отношения между объектами, величины объектов) подаются на вход Rete-алгоритма. Его задачей является определение изменений во множестве распознанных ситуаций, вызванных поступившими фактами. Во избежание неэффективного рассмотрения всех ситуаций, они организуются в сеть – разновидность классификационной сети (сеть строится перед сеансом работы решателя). В узлах сети происходит обработка поступивших фактов, во время которой проверяются условия из описания ситуаций. В зависимости от результата проверок факты задерживаются в узлах или передаются дальше по сети. Если в результате обработки какие-либо факты оказываются в конечных узлах сети, это означает, что распознаны ситуации, связанные с данными конечными узлами.

В ходе распознавания интерпретатор сети проверяет различные условия для фактов базы задачи. Их можно разделить на два класса: внутренние условия (проверяются для одного факта) и внешние (определяют взаимосвязь между несколькими фактами).

При прохождении факта через сеть он сначала попадает в корневую вершину, которая передает его своим преемникам – одноходовым узлам. Это узлы, осуществляющие проверку внутренних условий. Если условие выполнено для поступившего факта, то он передается дальше. Среди преемников одноходовых узлов могут быть двухходовые узлы, с которыми связаны внешние условия. В таких узлах факты объединяются, а затем проверяется внешнее условие, в зависимости от выполнения которого объединенный факт будет пропущен дальше или остановлен в данном узле. С каждым входом двухходового узла связан список-память, в котором хранятся копии фактов, пришедших на данный вход. Это необходимо для сопоставления с фактами, которые позднее придут на другой вход узла. Пропуск факта через сеть заканчивается, если не будет выполнена проверка в каком-либо узле, или если факт попадет в конечной (терминальный) узел. Конечной узел завершает последовательность узлов для некоторой ситуации. Факт, попадающий на его вход, сигнализирует об успешном распознавании ситуации и содержит означивание переменных из описания ситуации – строку сопоставления.

Для эффективной работы Rete-алгоритма необходимо, чтобы выполнялись следующие условия:

- Множество ситуаций должно быть статично.

- Анализируемые факты не должны содержать переменных.
- Множество фактов должно изменяться достаточно медленно.

Все эти условия соблюдены в РГЗ, так как набор ситуаций не меняется в ходе решения задачи, и множество записываемых фактов невелико. Это позволило использовать Rete-алгоритм как основу при реализации сопоставителя ситуаций решателя.

2.2. Структура используемой в решателе Rete-сети

При модернизации Rete-алгоритма для нужд РГЗ основные изменения были внесены в структуру Rete-сети. В решателе факты, приходящие на вход сопоставителя ситуаций делятся на два класса – факты-объекты и факты-отношения. Факт-объект поступает при анализе текущего контекста, в случае создания нового фрейма-экземпляра какого-либо геометрического объекта или записи нового значения слота в ранее порожденный фрейм. Факт-отношение обрабатывается сопоставителем ситуаций при записи нового проблемного отношения в базу задачи. Каждый факт в базе задачи имеет опору – пустую в случае безусловно истинного факта или список условных фактов, при условии истинности которых факт считается истинным. Опора факта учитывается при его обработке в сети решателя, что позволяет учитывать текущий контекст при распознавании ситуаций.

Как и в Rete-сети, в сети решателя каждый узел проверяет некоторое свойство факта, и если оно выполнено в момент проверки, то передает факт своим преемникам, иначе факт отбрасывается. Узлы сети решателя можно разделить на пять классов:

- 1) начальный узел – не имеет входов, выполняет служебные функции, в сети присутствует один такой узел;
- 2) одноходовые проверяющие узлы – имеют один вход, один или более выходов, проверяют некоторое свойство факта;
- 3) одноходовые запоминающие узлы – имеют один вход, один и более выходов, запоминают попавшие на вход факты и без проверок передают дальше по сети;
- 4) двухходовые узлы – имеют два входа (правый и левый), один или более выходов, объединяют факты, пришедшие на один вход, с фактами из другого входа, выполняют сложные проверки над объединенными фактами;

5) терминальные узлы – имеют один вход и ни одного выхода, запоминают строки сопоставления ситуаций.

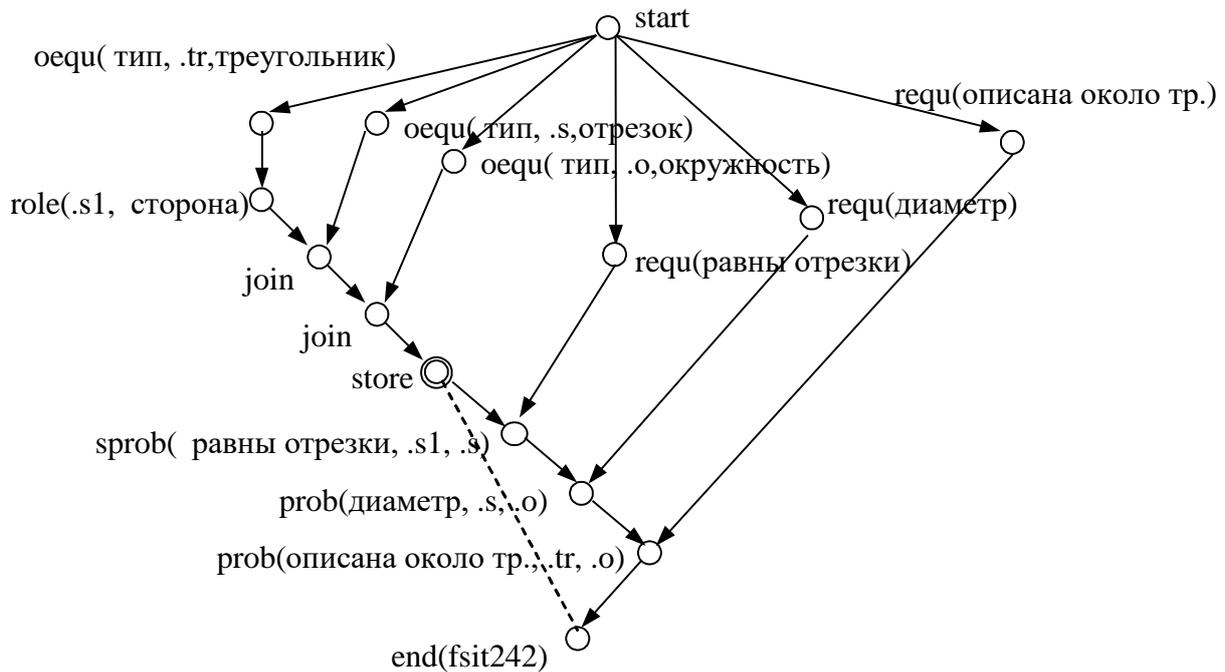


Рис.1. Фрагмент Rete-сети, соответствующий ситуации fsit242

Узлу, как правило, соответствует условие из описания ситуации, исключение составляют служебные (начальный и терминальный) узлы и одноходовые запоминающие узлы, введенные специально для реализации поиска близких ситуаций.

Рассмотрим структуру сети на примере, ранее использованной ситуации fsit242 (описание см. на странице 7). Соответствующий данному описанию фрагмент сети изображен на рис.1.

Условиям ситуации соответствуют десять узлов:

- 1) проверка факта-объекта на тип „треугольник“;
- 2) проверка факта-объекта на тип „отрезок“;
- 3) проверка факта-объекта на тип „окружность“;
- 4) проверка факта-отношения на тип „равны отрезки“;
- 5) проверка факта-отношения на тип „диаметр“;
- 6) проверка факта-отношения на тип „описана около треугольника“;
- 7) получение имени элемента объекта по роли „сторона“;
- 8) проверка проблемного отношения „равны отрезки“;
- 9) проверка проблемного отношения „диаметр“;
- 10) проверка проблемного отношения “описана около треугольника”.

Кроме того ситуации соответствует один терминальный узел, один одновходовый запоминающий узел для проверки, является ли близкой ситуация fsit242, и два объединяющих узла join.

В сети могут быть узлы 18 типов. С каждым типом узлов связана определенная функция, выполняемая при интерпретации сети сопоставителем ситуаций.

Узел типа start:

1. Начальный узел сети (start). Присутствует в сети в единственном экземпляре. Определяет тип информации, поступающий на вход сети (объект или отношение) и в зависимости от результата записывает факт в одну из частей сети - для объектов или для отношений.

Одновходовые проверочные узлы:

2. Проверка совпадения типа объекта с указанным (oequ). Является начальным узлом для части сети, предназначенной для объектов. Узел проверяет совпадение типа факта-объекта с заданным значением. В случае успеха передает объект своим преемникам. Например, изображенные на рис. 1 вершины типа oequ проверяют типы геометрических объектов.

3. Проверка совпадения типа отношения с указанным (requ). Узлы этого типа являются начальными для части сети, предназначенной для отношений. В узле осуществляется сравнение типа записываемого факта-отношения с указанным типом (например, типом «диаметр», см. рис.1).

4, 5. Получение имени элемента по имени слота или по роли (slot и role). Узел осуществляет сопоставление элемента, заданного именем слота (ролью), с реальным именем элемента, определенным в задаче. При этом полученное соответствие абстрактного имени элемента из описания ситуации с реальным именем добавляется к факту как часть строки сопоставления. Иногда возможны несколько сопоставлений (например, в треугольнике можно получить три различных имени стороны или угла). Каждое из них передается дальше по сети независимо от других, как новый факт. В рассматриваемом примере узел типа role служит для получения имени стороны треугольника (см. рис.1).

6, 7. Проверка отношения конфигурации между элементами одного или нескольких геометрических объектов (econf и conf). По реальным именам элементов определяется, выполнено отношение или нет. Примером отношения конфигурации может служить отношение «противолежат» между стороной и углом треугольника. В нашем примере узлы данных типов отсутствуют.

8, 9. Проверка известности (или неизвестности) значения элемента (valknown и valnknown). Узлы данного типа проверяют известность слота геометрического объекта. Если имя слота не указано, то по умолчанию у объекта проверяется слот «величина» (value). В узлах типа valknown, если значение слота известно, то опора обрабатываемого факта объединяется с опорой проверяемого слота (поскольку его значение может быть известно условно), и факт с объединенной опорой пропускается дальше по сети. Если требуется, чтобы значение не было известно, опора слота объединяется с отрицанием опоры пришедшего факта. Факт с новой опорой передается потомкам узла. Узлы типа valnknown пропускают факт дальше и при неизвестном значении слота, но в этом случае меняют его опоры. На рис.1 узлов типа valknown нет, так как в описании ситуации fsit242 нет соответствующих условий.

10, 11. Проверка равенства (или неравенства) значения слота заданной константе (valequ и valnoequ). Узлы данного типа проверяют совпадение значения слота геометрического объекта со значением, записанным в узле. Преобразование опоры факта осуществляется аналогично вышеописанным преобразованиям в узлах типа valknown/valnknown с поправкой на проверяемое условие.

Одновходовые запоминающие узлы:

12. Узлы типа store. С каждым из них связан список-память, в который заносятся все приходящие на вход узла факты. После запоминания факт без проверок и изменений передается потомкам узла. (Здесь описана функция узла только при распознавании ситуаций. Обработка фактов в узлах такого типа при поиске близких ситуаций будет описана ниже.)

Двухходовые вершины:

13. Объединение фактов без проверок (join). Узел служит для получения всех различных парных объединений всех приходящих в нее фактов. Никаких проверок в узле не выполняется. Для запоминания фактов, как и все двухходовые узлы, он имеет две памяти – левую для левого входа и правую для правого. Пришедший факт объединяется с фактами из памяти противоположной вершины, опоры которых включены в текущий контекст. Полученные объединенные факты пропускаются дальше по сети. На рис.1 узлов данного типа два.

14, 15. Проверка наличия (отсутствия) проблемного отношения в базе задачи (prob или prob). Узел проверяет проблемное отношение между объектами или элементами объектов. Он также имеет две памяти

для запоминания фактов. Входы узла не являются симметричными. На левый вход приходит факт-объект (возможно, несколько объединенных фактов-объектов), а на правый – факт-отношение. Для пришедшего отношения проверяется наличие в левой памяти факта-объекта, у которого в строке сопоставления аргументы отношения имеют такое же означивание, что и в факте-отношении. Опора каждого такого факта (отрицание опоры в узлах типа `prob`) объединяется с опорой отношения, после чего факт передается потомкам узла. В случае, когда в узел пришел факт-объект, производится поиск соответствующих отношений в правой памяти. Если в узле типа `prob` не найдено ни одного соответствия между объектами и отношениями, факт-объект пропускается дальше по сети без изменения опоры. В узлах типа `prob` такие факты отвергаются, как не прошедшие проверку. На рис.1 узлы данного типа проверяют проблемные отношения, «диаметр» и «описана около треугольника».

16, 17. Проверка наличия (отсутствия) симметричного отношения (`sprob` или `nsprob`). Этот тип узлов отличается от предыдущего тем, что предназначен для проверки симметричных проблемных отношений. В узлах учитывается то, что порядок аргументов симметричного отношения значения не имеет, в отличие от обычных проблемных отношений. Обработка фактов в узлах данного типа производится аналогично обработке в узлах типа `prob`/`prob`, но с поправкой на указанное обстоятельство. Узел данного типа в сети для ситуации `fsit242` проверяет симметричное проблемное отношение «равны отрезки».

Узел типа end:

18. Терминальный узел сети (`end`). Узлы данного типа являются заключительными узлами фрагментов сети, соответствующих описаниям ситуаций из базы знаний. Они запоминают все пришедшие к ним объединенные факты в своей памяти (вместе со строкой сопоставления). Если факт приходит с ложной в текущем контексте опорой, то он не запоминается, более того из памяти удаляются все тождественные ему факты, имеющие другие опоры.

Результатом обработки фактов в узлах сети решателя является множество ситуаций распознанных в текущем контексте. Для поиска близких ситуаций сопоставитель решателя в том виде, котором он описан выше, не пригоден. Неполные означивающие наборы фактов для близких ситуаций будут остановлены в промежуточных узлах сети, поскольку по определению часть условий близких ситуаций не выполнена. Тем не менее, при интерпретации

сети решателя в памяти узлов запоминаются данные, которые полезны при поиске близких ситуаций. Использование этих данных при поиске близких ситуаций описано ниже.

2.3. Использование Rete-сети при поиске близких ситуаций

При ассоциативном планировании стереотип должен быть пригодным для уточнения текущей цели. Поэтому первым этапом поиска близких стереотипов является определение множества стереотипов, уточняющих цели того же типа, что и текущая. Абстрактное описание уточняемой стереотипом цели сравнивается с текущей целью, и если цель подходит по типу, то стереотип считается пригодным и с ним связывается частичная строка сопоставления абстрактных объектов из описания стереотипа с реальными объектами из текущей уточняемой цели.

На втором этапе поиска происходит проверка является ли близкой ситуация стереотипа, то есть, существует ли для нее неполный означивающий набор фактов, который можно достроить для полного распознавания ситуации. Если набор существует (а их может быть несколько, как показано в 1.1.), то определяются недостающие факты (для каждого из наборов). Этот этап занимает большую часть времени, поскольку требует проверки условий из описания ситуации. Существенно сократить число проверок позволяет использование данных, накопленных сопоставителем ситуаций.



Рис. 2. Схема расположения узлов в Rete-сети.

В каждом фрагменте Rete-сети, соответствующем ситуации, узлы осуществляющие проверки фактов, вывод которых невозможен при работе решателя, расположены выше узлов проверяющих условия, которые можно удовлетворить выводом новых фактов (см. Рис. 2). Во время работы сопоставителя ситуаций на выход верхней части Rete-сети попадают

все строки сопоставления для данной ситуации, возможные в данном состоянии базы задачи. Они запоминаются в памяти узла store, чтобы затем быть использованными при поиске близких ситуаций. Это полные строки сопоставления, поскольку узлов, в которых происходит означивание в нижней части Rete-сети нет. Некоторые из этих строк проходят нижнюю часть сети и попадают в терминальный узел (узел end) – они соответствуют распознанным ситуациям. Эти строки запоминаются в памяти терминального узла и в дальнейшем выдаются в ответ на запросы к сопоставителю ситуаций. Остальные строки сопоставления соответствуют близким ситуациям, поскольку для каждой из них невыполненные условия в описании ситуации можно удовлетворить добавлением новых фактов (условия на значения слотов объектов и проблемные отношения). То есть, каждая из строк соответствует неполному означивающему набору фактов, который можно достроить. Остается определить недостающие факты, чтобы можно было применить найденную близкую ситуацию при планировании.

Узел store специально предназначен для поиска близких ситуаций. Это одновходовый узел с памятью, в которой запоминаются все входящие строки сопоставления. При обычной работе сопоставителя, функция этого узла ограничивается только запоминанием и передачей строк сопоставления потомкам узла. В режиме поиска близких ситуаций строки сопоставления из памяти узла store (те из них, которые включают в себя частичные строки, полученные при сопоставлении текущей цели с целью из описания стереотипа, и которые не содержатся в памяти узла end) передаются для проверок потомкам узла store. При этом проход фактов по сети отличается от прохода в режиме распознавания ситуаций. Попадая в какой-либо узел, строка сопоставления обрабатывается в соответствии с типом данного узла, но если проверка, связанная с узлом неуспешна, то строка передается потомкам узла с пометкой о невыполненном условии. Если же проверка успешна, строка передается дальше без пометок. Таким образом, дойдя до узла end, каждая строка сопоставления для близкой ситуации будет иметь одну или более пометок о невыполненных условиях, по которым будет построены недостающие факты.

Перечень шагов алгоритма поиска близких стереотипов:

1. Положить пустым множество близких стереотипов S.
2. Для всех стереотипов из базы знаний повторять шаги 3-12.
3. Если тип уточняемой стереотипом цели совпадает с типом текущей цели, то запомнить частичную строку сопоставления объектов из описания

уточняемой стереотипом цели с реальными объектами из текущей цели, иначе перейти к обработке другого стереотипа (шаг 2).

4. Для ситуации стереотипа найти соответствующие узлы Rete-сети типа store и end.
5. Для всех строк сопоставления из памяти узла store повторять шаги 6-11.
6. Если строка сопоставления содержит частичную строку, запомненную на шаге 3, и отсутствует в памяти вершины end, то передать строку узлу, который является потомком вершины store, иначе перейти к обработке другой строки (шаг 5).
7. Если тип узла-потомка \neq end, то выполнить проверку, соответствующую узлу, иначе перейти на шаг 10.
8. Если проверка неуспешна, то сделать пометку о невыполненном условии.
9. Передать строку потомку текущего узла и перейти на шаг 7.
10. Если строка сопоставления имеет пометки о невыполненных условиях, то стереотип считать близким, построить набор недостающих фактов по пометкам строки, занести четверку (имя стереотипа, ситуация, строка сопоставления, список недостающих фактов) в множество S.
11. Если есть еще строки в памяти узла store, перейти на шаг 5.
12. Если в базе знаний остались нерассмотренные стереотипы, то перейти на шаг 1.

Запись алгоритма на псевдокоде:

```

Set:=  $\emptyset$  {Set – множество близких стереотипов}
FOR Ster:= Ster1 TO SterN DO {Ster1, ..., SterN – все стереотипы в базе знаний}
  IF тип цели Ster совпадает с типом текущей цели THEN
    Match1:= частичная строка сопоставления объектов из описания уточняемой
    стереотипом цели с реальными объектами из текущей цели
    FOR Str:= Str1 TO StrM DO
      {Str1, ..., StrM – строки сопоставления из памяти узла Store ситуации стереотипа
      Ster, которых нет в памяти узла End}
      IF Match1 является подстрокой Str THEN
        Node := узел-потомок узла Store
        WHILE тип Node  $\neq$  End DO
          Выполнить проверку в узле Node
          IF проверка неуспешна THEN
            приписать к Str пометку о невыполненном условии
          END IF
          Node := узел-потомок Node
        END WHILE
      END IF
    END FOR
  END IF
END FOR
  
```

построить список недостающих фактов Facts по пометкам Str

Set:= Set + (Ster, ситуация Ster, Str, Facts)

END IF

END FOR

END IF

END FOR

Приведенный алгоритм поиска близких стереотипов можно сравнить с другим алгоритмом, в котором поиск ситуаций близких стереотипов происходит одновременно с сопоставлением ситуаций. Во втором алгоритме не требуется специальных запоминающих узлов, а факты, обрабатываемые в сети решателя, проходят от узла к узлу с пометками о выполнении или невыполнении проверяемых в узлах условий. Попадая в концевую вершину, факты, не имеющие пометок о неудачных проверках, сигнализируют о распознавании ситуаций, а факты с пометками о нарушенных условиях – о близких ситуациях.

Недостатком сопоставителя ситуаций, выполняющего одновременно полное распознавание и поиск близких ситуаций, является его меньшая эффективность по сравнению с алгоритмом, который был рассмотрен ранее. Во-первых, значительно увеличивается время распознавания ситуаций, поскольку факты не задерживаются в промежуточных узлах сети, тогда как именно отбрасывание фактов, не прошедших проверок, обеспечивает сравнительно быстрое распознавание ситуаций в Rete-алгоритме (оценки эффективности Rete-алгоритма приведены в работе [Forgy, 1982]). Во вторых, лишь малую часть найденных по второму алгоритму близких ситуаций удастся использовать для уточнения текущей цели, так как не все стереотипы, связанные с ними, будут пригодны для уточнения текущей цели. Это означает, что большая часть времени работы сопоставителя ситуаций будет затрачена впустую.

Необходимо заметить, что алгоритм поиска близких ситуаций является достаточно универсальным. Его можно применять в тех предметных областях, которые отвечают требованиям, необходимым для применения Rete-алгоритма.

3. Построение уточнения цели при ассоциативном планировании

Уточнение текущей цели, получаемое при ассоциативном планировании, представляет собой подграф графа целей, имеющий параллельно-последовательную структуру – такую же что и структура уточнения,

получаемого методом обратной цепочки рассуждений (будем далее называть такое уточнение дедуктивным). Разница между этими двумя видами уточнений заключается в способе их построения.

Дедуктивное уточнение состоит из параллельно объединенных цепочек целей, добавление каждой из которых предписано действиями стереотипа, уточняющего текущую цель. Процесс построения ассоциативного уточнения плана (уточнения на основе близких ситуаций) сложнее, чем построение дедуктивного уточнения. Ассоциативное уточнение тоже состоит из параллельно объединенных цепочек целей, каждая из которых построена на основе стереотипа с близкой ситуацией. Такие цепочки состояются из двух частей – подцепочки целей для вывода недостающих фактов и целей, добавляемых в действие стереотипа. Подцепочка целей для вывода недостающих фактов строится следующим образом:

- 1) для каждого недостающего факта конструируется цель в соответствии с его типом, реальными объектами и их слотами, указанными в факте;
- 2) сконструированные цели объединяются последовательно без указания их порядка, так как все они должны лежать в графе на одном пути, а порядок их достижения не важен.

На первом шаге построения подцепочки используются знания о соответствии типов выводимых фактов типам целей, явно представленные в базе знаний РГЗ конструкциями следующего вида:

Комментарий: утверждение о равенстве значения слота объекта заданной величине

Содержание цели: ["equal", Объект, Слот, Величина]

Факт: val("equal", Объект, Слот, Величина)

Тип цели: "equal"

Трудность цели: 1

Приведенная конструкция представляет знания о том, что при отсутствии в текущем состоянии задачи фактов вида, указанного в поле «факт», следует добавлять в граф планирования цели типа "equal". Достижение данной цели, то есть, успешное доказательство равенства, повлечет добавление в базу задачи факта равенства значения слота указанной величине.

Полученная подцепочка целей для вывода недостающих фактов последовательно объединяется с цепочкой целей, запись которой предписана действиями стереотипа. При их объединении важен порядок, поскольку только после вывода недостающих фактов действия стереотипа могут быть выполнены.

Алгоритм построения ассоциативного уточнения плана таков:

- 1) Стереотипы с близкими ситуациями оцениваются согласно критерию «полезности».
 - 2) Для каждого стереотипа, признанного «полезным», строится цепочка целей для вывода недостающих фактов. Остальные стереотипы в построении уточнения не используются.
 - 3) Для каждого «полезного» стереотипа вычисляется трудность построенной цепочки и проверяется условие «доступности».
 - 4) «Доступные» стереотипы проходят проверку комбинированным критерием.
 - 5) Для каждого стереотипа, прошедшего все проверки, строится часть уточнения, т.е. цепочка, полученная на шаге 2, последовательно объединяется с цепочкой целей из действий стереотипа, связанного с рассматриваемой ситуацией.
- б) Все полученные на шаге 5 части уточнения параллельно объединяются.

При вставке полученного уточнения в граф планирования из него удаляются цепочки, приводящие к появлению в графе так называемого «порочного круга», если таковые входят в состав уточнения. «Порочным кругом» называется состояние графа, при котором одна и та же подцель два раза встречается на каком-либо из допустимых путей в графе. Недопущение возникновения в графе «порочного круга» осуществляется средствами поддержки работы с графом.

Одним из свойств ассоциативного уточнения является то, что любая из его параллельно объединенных цепочек целей содержит недостигнутые цели (цели для вывода недостающих фактов). Этим свойством ассоциативное уточнение отличается от дедуктивного, в котором могут быть цепочки из элементарных целей, не требующих уточнения и считающихся достигнутыми сразу после занесения в граф планирования. Вставка ассоциативного уточнения в граф не может привести к окончанию решения, поскольку на всяком появившемся новом пути в графе будут лежать недостигнутые цели. Следовательно, ассоциативный метод должен использоваться при решении задач в сочетании с другими методами планирования решений.

Преимущества использования ассоциативного планирования состоят в том, что когда нет распознанных ситуаций, а решение еще не окончено, система может продолжить планирование ассоциативным методом (если будет найден подходящий стереотип с близкой ситуацией), а также в том, что система может вести более широкие рассуждения, нежели те, что предусмотрены набором функциональных стереотипов. Для решения задачи

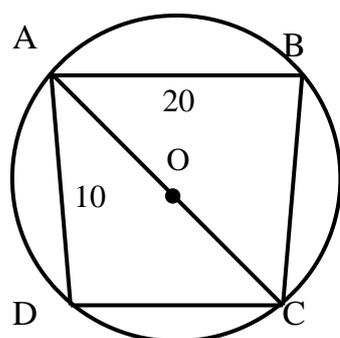
могут быть использованы стереотипы, которые при применении традиционных методов (прямой цепочки рассуждений и обратной цепочки рассуждений) неприменимы.

Заканчивая рассмотрение метода ассоциативного планирования, приведем описание его алгоритма. Метод итеративный, алгоритм представляет собой последовательность шагов повторяющихся до тех пор, пока не будет получен план, или не возникнет тупик в решении. Действия, совершаемые на каждом шаге таковы:

- 1) Среди неуточненных и недостигнутых целей в графе целей выбирается текущая цель (ТЦ) для уточнения.
- 2) Формируется активный контекст ТЦ.
- 3) Определяется набор стереотипов пригодных для уточнения текущей цели, ситуации которых считаются близкими. Стереотипы из набора оцениваются в соответствии с критериями «полезности», «доступности» и комбинированным критерием. С каждым стереотипом, удовлетворяющим всем критериям, связывается цепочка целей для вывода недостающих фактов.
- 4) Если подходящих стереотипов нет, то ТЦ заносится в «черный список», и осуществляется переход на следующий шаг планирования.
- 5) По набору стереотипов, отобранных ранее, строится ассоциативное уточнение ТЦ. Уточнение заносится в граф.
- 6) Осуществляется переход на следующий шаг планирования.

4. Примеры решений геометрических задач

Рассмотрим несколько примеров решения задач. Для поиска решений будет использоваться, так называемая, стратегия объединенного уточнения, при которой совместно применяются ассоциативное планирование и обратная цепочка рассуждений.



Задача №1

Дано: ABCD – трапеция с основаниями AB и CD. Окружность O описана около ABCD. AC – диаметр. AB=20 см, AD=10 см.

Найти: площадь трапеции ABCD.

Рис. 3. Чертеж условий задачи №1

На первом шаге планирования текущей целью выбирается цель, поставленная в задаче. Объединенное уточнение, построенное для нее, состоит из двух цепочек целей: найти площадь треугольника ABC , найти площадь треугольника ACD и вычислить S_{ABCD} как сумму найденных площадей; доказать, что $ABCD$ – прямоугольник и вычислить S_{ABCD} как произведение $AB \cdot AD$.

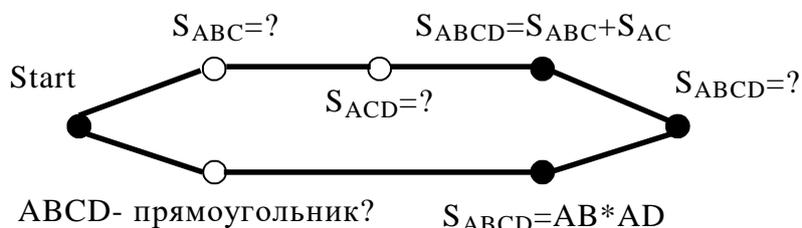


Рис. 4. Состояние графа после первого шага планирования

Вторая альтернатива в уточнении получена на основе близкой ситуации. Вид графа планирования после вставки уточнения изображен на рис. 4. Такое же уточнение могло быть получено и при использовании обратной цепочки рассуждений, но для этого необходимо добавление в базу знаний решателя дополнительного стереотипа. Ассоциативное планирование, основанное на аналогии между ситуациями функциональных стереотипов, позволяет в данном случае обойтись без расширения базы знаний.

На втором шаге текущей выбирается цель «доказать, что $ABCD$ прямоугольник», поскольку она лежит на более дешевом пути. Эта цель уточняется цепочкой из четырех целей: доказать что угол $abc=90^\circ$, доказать что угол $bcd=90^\circ$, доказать что угол $adc=90^\circ$, доказать что угол $bad=90^\circ$ и записать, что $ABCD$ является прямоугольником по определению.

При дальнейшем планировании происходит уточнение целей на нижнем пути (этот путь является более дешевым). За четыре шага производится доказательство того, что все углы $ABCD$ являются прямыми. По окончании шестого шага планирования на нижнем пути в графе не остается недостигнутых вершин, и решение заканчивается. Объяснение полученного решения имеет вид:

Угол abc - прямой, так как противолежащая ему в треугольнике ABC сторона AC является диаметром описанной вокруг ABC окружности O . Угол $bcd=180-abc=90$, так как они внутренние односторонние при основаниях AB и CD трапеции $ABCD$. Угол adc - прямой, так как противолежащая ему в треугольнике ACD сторона AC является диаметром описанной вокруг ACD окружности O . Угол $bad=180-adc=90$, так как они внутренние односторонние при основаниях CD и AB трапеции $ABCD$. Четырехугольник $ABCD$ –

прямоугольник, так как все углы (abc , bcd , adc , bad) равны 90 . Площадь прямоугольника $ABCD=AB*AD=20*10=200$.

Данный метод решения имеет ряд преимуществ по сравнению с методом прямого вывода. Во-первых, все цели появляющиеся при планировании имеют непосредственное отношение к поставленной задаче, и не производится вывода лишних фактов. Во-вторых, сравнительно немного времени тратится на счет. Это происходит, поскольку при ассоциативном планировании рассуждений число фактов, анализируемых распознавателем ситуаций, значительно меньше, а на работу распознавателя в РГЗ приходится основные временные затраты. Преимущества стратегии объединенного уточнения перед методом обратной цепочки рассуждений связаны с использованием ассоциативного планирования, ведь именно благодаря использованию ассоциаций в самом начале решения появляется путь, который в последствии ведет к достижению поставленной цели.

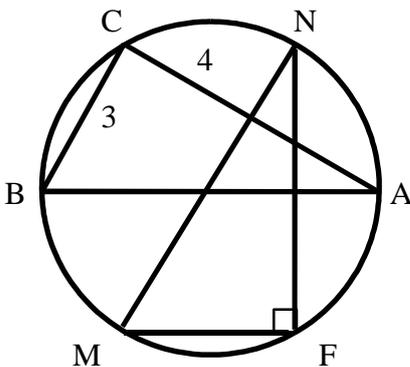


Рис. 5. Чертеж условий задачи №2

Задача №2

Дано: окружность O описана около треугольников ABC и MNF , отрезки AB и MN равны, угол mfn равен 90° , угол fmp равен 45° , величина отрезка BC равна 3 , величина отрезка AC равна 4 .

Найти: площадь треугольника ABC .

На первом шаге решения текущей целью выбирается цель, поставленная в задаче – «найти площадь ABC ». Для нее строится уточнение: найти AB и вычислить площадь по формуле Герона или найти угол acb и вычислить площадь ABC по формуле $S_{ABC}=\frac{1}{2}*AC*BC*\sin(acb)$. После вставки уточнения граф планирования имеет структуру, изображенную на Рис. 6.

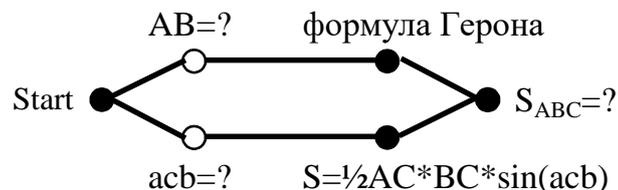


Рис. 6. Граф целей после 1-го шага

Следующей целью для уточнения выбирается цель «найти величину AB ». Объединенное уточнение, построенное для нее на втором шаге, состоит из трех альтернатив: найти acb и вычислить AB по теореме косинусов; или найти MN и записать значение AB , поскольку $AB=MN$; или найти acb , найти bac и

вычислить AB по теореме косинусов. Структура графа планирования изображена на рисунке 7. Фактически на первых двух шагах ассоциативное планирование не вносит в ход решения никакого вклада, так как предлагаемые им пути решения совпадают с теми, которые найдены методом обратной цепочки.

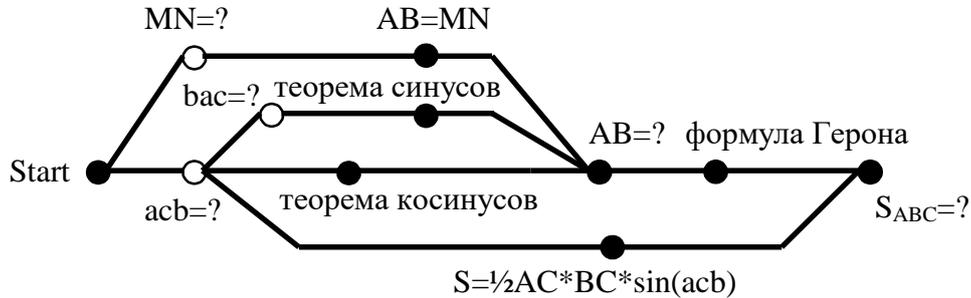


Рис. 7. Граф планирования после 2-го шага

Ситуация меняется на третьем шаге решения. Текущей выбирается цель «найти величину acb », и для нее строится уточнение: найти bac , найти abc и вычислить acb по теореме о сумме углов треугольника; или доказать, что AB является диаметром окружности O и записать, что acb – прямой угол, поскольку опирается на диаметр. При этом вторая альтернатива в уточнении построена на основе стереотипа с близкой ситуацией. Вид графа планирования после вставки уточнения изображен на рисунке 8.

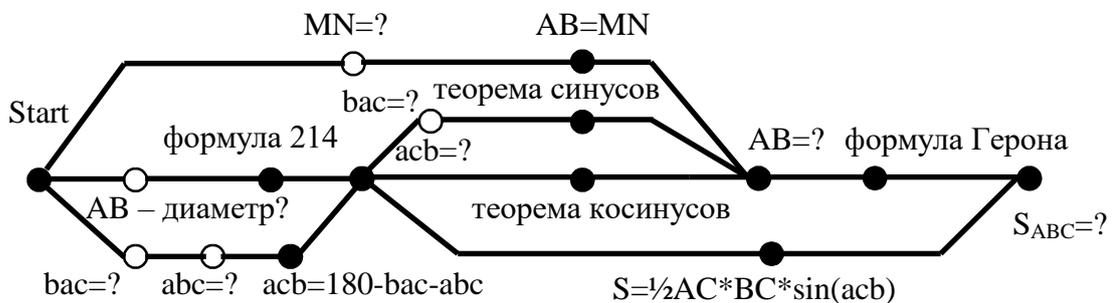


Рис. 8. Граф планирования после 3-го шага решения

Текущей целью на четвертом шаге решения выбирается цель «найти величину MN », так как она лежит на самом дешевом пути. Для нее строится уточнение: найти величину FN , найти величину FM и вычислить MN по теореме косинусов; или найти FN и вычислить MN по теореме синусов. Поскольку цель «найти величину FN » лежит на двух путях в графе, на пятом шаге она выбирается текущей. Уточнение, построенное для этой цели, имеет вид: найти величину угла fnm и вычислить FN по теореме косинусов. При построении уточнения считается условно достигнутой целью «найти величину FM », входящая в контекст текущей цели.

На шестом шаге решения для уточнения выбирается цель «доказать, что АВ является диаметром О», поскольку она лежит на одном из дешевых путей и является более трудной, чем остальные недостигнутые цели. С привлечением ассоциативного планирования строится уточнение цели: доказать, что MN является диаметром и записать, что АВ диаметр, так как он является хордой окружности О, равной ее диаметру. На седьмом шаге цель «доказать, что MN является диаметром О» получает тривиальное уточнение: применить доказательство 215 (MN является диаметром, так как MN гипотенуза прямоугольного треугольника, около которого описана окружность О). В графе планирования появляется путь-решение, которому соответствует такое объяснение: *Отрезок MN – диаметр окружности О, описанной около прямоугольного треугольника MNF, так как он является гипотенузой MNF. Сторона АВ треугольника ABC является диаметром описанной около него окружности О, так как АВ=MN – диаметру О. Угол acb – прямой, так как противоположная ему в треугольнике ABC сторона АВ является диаметром описанной около ABC окружности О. Площадь треугольника ABC вычисляется как $S = 0.5 * BC * AC * \sin(acb) = 6$.*

Решение задачи получено за семь шагов, при этом два шага затрачены на уточнение подцелей, лежащих в стороне от решающего пути. Тем не менее, при применении стратегии объединенного уточнения все факты, добавляемые в базу задачи, связаны с искомой целью, вывод лишних фактов не производится, меньше временные затраты на сопоставление ситуаций.

Подведем итог рассмотренным примерам решения задач. Практика показывает, что использование ассоциативного планирования позволяет повышать эффективность поиска решений некоторых задач. Реализация в решателе этого нового метода планирования, дает значительные преимущества по сравнению с системами, ориентированными на использование традиционных методов решения.

Литература

- [Вакин, Корухова и др., 1997] Вакин В. В., Корухова Л. С., Любимский Э. З., Малышко В. В. *Ассоциативные методы планирования решений сложных задач.* – М.: Препринт Института прикладной математики им. М. В. Келдыша РАН, № 81, 1997 г.
- [Корухова и Любимский, 1994] Корухова Л. С., Любимский Э. З. *О процедурности и непроцедурности в задачах планирования целенаправленной деятельности.* – М.: Известия Академии Наук. Серия: «Техническая кибернетика», № 5, стр. 72-78, , 1994.
- [Корухова и др., 2000] Корухова Л. С., Любимский Э. З., Малышко В. В. *Реализация стратегий планирования на основе управляющих стереотипов.* – М.: Препринт Института прикладной математики им. М. В. Келдыша РАН, № 13, 2000 г.
- [Forgy, 1982] Forgy C. L. *RETE: A fast algorithm for the many pattern / many object pattern match problem.* Artificial Intelligence, Vol. 19, pp. 17-37, 1982.
- [Kolodner, 1993] Kolodner J. L. *Case-Based Reasoning.* Los Altos, CA: Morgan Kaufmann. 1993.
- [McDermott, 1982] McDermott J. *R1: a rule-based configurer of computer systems.* Artificial Intelligence, Vol. 19, pp. 39-88. 1982.
- [Melis and Whittle, 1997] Melis E., Whittle J. *External Analogy in Inductive Theorem Proving.* Proceedings of the 21st German Annual Conference on Artificial Intelligence (KI-97). Albert-Ludwigs University. 1997.
- [Melis, 1998] Melis E. *Proof Planning with Multiple Strategies.* Proceedings of the 14th Annual Conference of the Canadian Association for Distant Education (CADE-98). 1998.
- [Miranker and Lofaso, 1991] Miranker D.P., Lofaso B.J. The organization and performance of a TREAT-based production system compiler. - The IEEE Transactions on Knowledge and Data Engineering. Vol. 3, No. 1, pp. 3-9 March 1991.

Содержание

ВВЕДЕНИЕ	3
1. АССОЦИАТИВНОЕ ПЛАНИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ БЛИЗКИХ СИТУАЦИЙ .	5
1.1. Близкая ситуация.....	6
1.2. Критерии отбора стереотипов для получения эффективного плана	8
2. СОПОСТАВИТЕЛЬ СИТУАЦИЙ	10
2.1. Идея Rete-алгоритма	11
2.2. Структура используемой в решателе Rete-сети.....	12
2.3. Использование Rete-сети при поиске близких ситуаций.....	17
3. ПОСТРОЕНИЕ УТОЧНЕНИЯ ЦЕЛИ ПРИ АССОЦИАТИВНОМ ПЛАНИРОВАНИИ	20
4. ПРИМЕРЫ РЕШЕНИЙ ГЕОМЕТРИЧЕСКИХ ЗАДАЧ	23
ЛИТЕРАТУРА.....	28