

Кулик Борис Александрович

ПРОГРАММА ДЛЯ МОДЕЛИРОВАНИЯ И АНАЛИЗА ЕСТЕСТВЕННЫХ РАССУЖДЕНИЙ

ВВЕДЕНИЕ

В процессе работы над своей книгой [1], идею которой мне подсказал учитель математики В.И. Рыжик, у меня возникла необходимость быстро и точно просчитывать многочисленные приведенные в ней примеры рассуждений, анализ которых на бумаге занимал довольно много времени. А поскольку среди компьютерных реализаций методов решения логических задач мне не удалось найти метода, о котором рассказывается в этой книге, то пришлось самому заняться составлением такой программы. Работа оказалась нелегкой, но ее результаты окупилась сторицей - появилась возможность не только значительно ускорить процесс анализа рассуждений, но и найти новые закономерности общей математической модели рассуждений, которые существенно дополнили книгу.

Данная программа предназначена для моделирования и анализа содержательных рассуждений (в рамках расширенной полисиллогистики). Реализованный в ней метод позволяет не только проверить выводимость определенного суждения из произвольно заданной системы посылок, но и решать важные для практических приложений задачи, которые в традиционной полисиллогистике либо решаются с большими трудностями, либо не решаются вообще.

При работе над составлением программы основное внимание я уделял алгоритмике метода и не особенно заботился о пользовательском интерфейсе. Теоретические аспекты используемого в программе метода моделирования и анализа рассуждений изложены в книге, и здесь о них будет сказано немного. Приемы работы с програм-

мой и возможности метода мы рассмотрим на нескольких примерах.

ОСОБЕННОСТИ РАБОТЫ ПРОГРАММЫ

1) Программа представлена в варианте для DOS с упрощенным экранным интерфейсом в текстовом режиме;

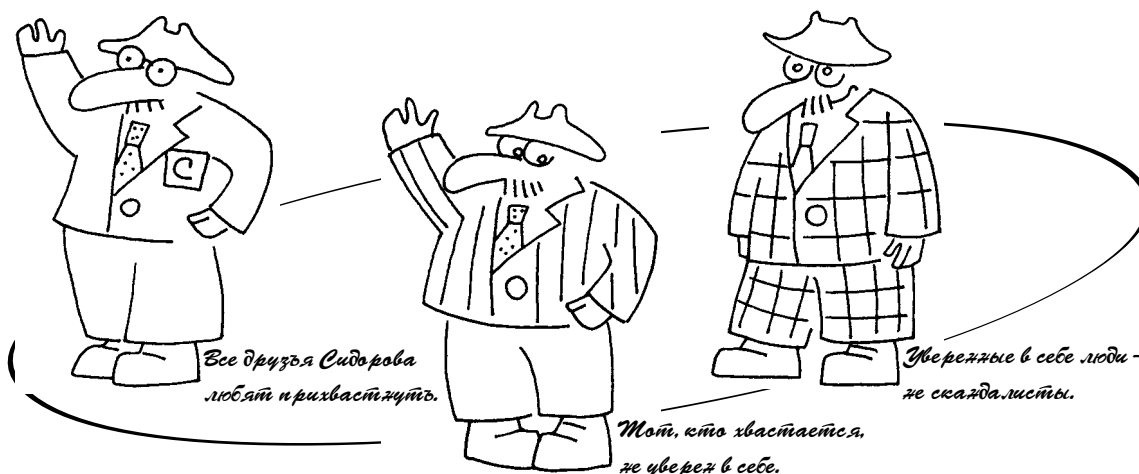
2) Программа записывает в дисковую память и считывает с нее систему исходных посылок. Протокол работы программы выводится на экран. По желанию пользователя он может быть выведен на печатающее устройство или записан в текстовом файле.

3) Все исходные задачи (с расширением LOG) и протоколы сеансов работы с задачами (с расширением TXT) записываются в подкаталоге TASKS.

4) Программа не предназначена для работы с “натуральными” терминами и понятиями. Прежде чем начинать работать с программой, необходимо термины и понятия анализируемой системы рассуждений закодировать числами. Правила “кодирования” изложены ниже.

5) Программа позволяет работать с системами, в которых содержится не более 100 “позитивных” терминов - это не так уж и мало, если учесть, что в составе любого из известных и до сих пор еще не объясненных окончательно парадоксов содержится не более 10 терминов. Данная программа позволяет произвести анализ любого из известных парадоксов.

6) В каталоге TASKS содержится несколько модельных задач с числом “позитивных” терминов от 5 до 24-х. В меню предусматривается возможность загрузки любой из этих задач. Можно составить



также свою задачу и записать ее в каталог с выбранным именем.

7) Программа составлена в среде Турбо Паскаль 6.0.

Для инсталляции программы достаточно переписать ее в любой каталог любого IBM-совместимого компьютера вместе с подкаталогом TASKS.

ПРИМЕРЫ НОРМАЛЬНЫХ СИСТЕМ

Пример 1.

Предположим, что нам нужно проанализировать следующие послышки:

- 1) Все друзья Сидорова любят хвастнуть.
- 2) Тот, кто хвастается, не уверен в себе.
- 3) Уверенные в себе люди - не скандалисты.

Прежде, чем вводить эти послышки в машину, надо немного поработать с карандашом и бумагой. Во-первых, нужно определить “возможный мир”. В данном случае очевидно, что речь идет о каком-то множестве людей.

Во-вторых, выделим сформулированные в послышках составные части этого “возможного мира” и введем для них соответствующие обозначения. В нашем примере это можно сделать сравнительно легко:

- друзья Сидорова (A1),
- все, кто хвастается (A2),
- все уверенные в себе (A3),
- потенциальные скандалисты (A4).

Здесь выделены только “позитивные” термины, то есть термины без

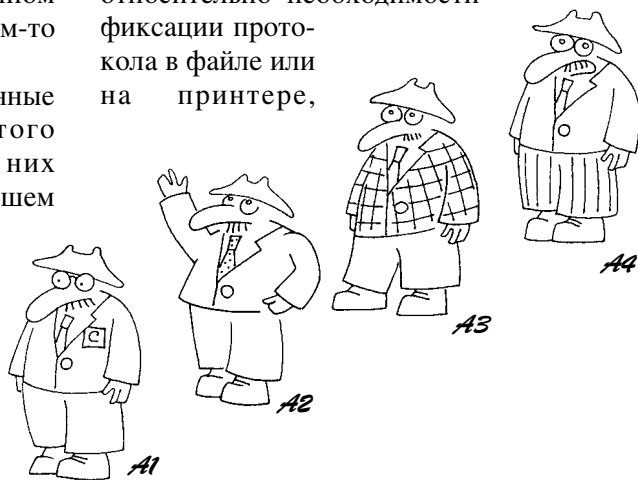
отрицания. Само собой разумеется, что для каждого “позитивного” термина должна присутствовать его антитеза, в частности:

- не друзья Сидорова (-A1),
- все, кто не хвастается (-A2),
- все неуверенные в себе (-A3),
- не скандалисты (-A4).

Для простоты мы будем использовать знак “минус” (-) для обозначения отрицания. Кроме того, чтобы не вводить лишних символов, мы, вместо указанных обозначений терминов, будем вводить просто числа с отрицанием или без него: термин A2 будет у нас обозначен как 2, а термин с отрицанием -A4 - просто как отрицательное число (-4). Тогда наши исходные послышки приобретут следующий вид:

- 1) 1 → 2;
- 2) 2 → -3;
- 3) 3 → -4.

Теперь можно запустить программу. Ответив на несколько вопросов относительно необходимости фиксации протокола в файле или на принтере,





перейдем к
ответу на
основные вопросы.

Нам будет предложено выбрать задачу из базы данных или составить ее самому. Выберем последнее.

Следующий вопрос: Сколько “позитивных” терминов необходимо для посылок? Тут все ясно. Отвечаем: 4.

Сколько посылок Вы собираетесь вводить?

Тут тоже все ясно. Отвечаем: 3.

Далее следует такой диалог, в котором нам необходимо только ввести соответствующие положительные или отрицательные числа:

ПОСЫЛКА N 1:

ЛЕВАЯ часть: 1

ПРАВАЯ часть: 2

ПОСЫЛКА N 2:

ЛЕВАЯ часть: 2

ПРАВАЯ часть: -3

ПОСЫЛКА N 3:

ЛЕВАЯ часть: 3

ПРАВАЯ часть: -4.

Мы не будем подробно рассматривать эту часть диалога. Отметим только, что наши посылки могут иметь и более сложный вид. Например, такой: 2,-3 → 1,-4,5. Тогда эту посылку можно ввести так:

ЛЕВАЯ часть: 2,-3

ПРАВАЯ часть: 1,-4,5

При этом надо правильно расставлять знаки и использовать в качестве разделителя между “терминами” только запятые.

Перейдем к нашей задаче. После ввода



1 → {-3}



-2 → {-1}



3 → {-1,-2}



4 → {-3}

посылок мы увидим на экране следующий ответ:

ИСХОДНЫЕ ПОСЫЛКИ:

1 → {2}

2 → {-3}

3 → {-4}

Система не содержит коллизий.

СПИСОК ПОЛУЧЕННЫХ

СЛЕДСТВИЙ:

1 → {-3}

-2 → {-1}

3 → {-1,-2}

4 → {-3}

О коллизиях мы поговорим позже. А сейчас попробуем расшифровать следствия:

1 → {-3}: Все друзья Сидорова не уверены в себе.

-2 → {-1}: Все, кто не хвастается, не друзья Сидорова.

3 → {-1,-2}: Все уверенные в себе не относятся к друзьям Сидорова и не хвастаются.

4 → {-3}: Все скандалисты не уверены в себе.

Можно считать, что система, приняв исходные посылки в качестве аксиом, вывела из этих “аксиом” все возможные “теоремы”. Но программа на этом не успокаивается и задает нам новые вопросы:

Будем проверять полноту системы (Y/N)?

А почему бы не проверить? Наждем “Y” и получим ответ:

Система НЕПОЛНА. В нее можно ввести по отдельности следующее множество элементарных предложений:

1 → {4,-4}

-1 → {-4}

2 → {4,-4}

-2 → {-4}

4 → {1,-1,2,-2}

-4 → {-1,-2}

Вот тебе раз! А мы-то думали, что выжали

из этой простенькой задачи все, что можно!

Сначала по поводу термина “элементарные предложения”. Это предложения, в котором в левой и правой части содержится по одному термину. Так что в первой строке выведено два элементарных предложения $1 \rightarrow 4$ и $1 \rightarrow -4$. А в пятой строке их содержится 4 штуки.

Эти “дополнительные” элементарные предложения обладают одной особенностью: если мы любое из них введем в нашу систему рассуждений в виде дополнительной посылки, то от этого в системе не произойдет никаких неприятностей. Но если мы введем сразу несколько предложений, то такие неприятности могут произойти. Разговор об этих “неприятностях” (точнее, коллизиях) мы немного отложим, а сейчас пойдем дальше.

Рассмотрим первую пару “дополнительных” предложений, то есть переведем их на нормальный язык. Получим: 1) “**Все друзья Сидорова скандалисты**” и 2) “**Все друзья Сидорова не скандалисты**”. Возможно, что какое-то из этих утверждений соответствует действительности. Если Сидоров утверждает, что верно второе предложение, то мы можем в новом сеансе работы с программой ввести это предложение в нашу систему в виде новой посылки.

Но Сидоров может усомниться и в том и в другом предложении. Он может сказать, например: “Кажется, только некоторые из моих друзей не скандалисты”. Оказывается, и в этом случае можно внести больше ясности в систему. Для этого введем новый термин А5 - “некоторые люди” и отобразим возможное высказывание Сидорова в виде “машинного” предложения $5 \rightarrow \{1, -4\}$.

Возьмем это на заметку, а пока что продолжим работу с программой. Следующий

вопрос к нам: “Хотите вывести совместно все посылки и следствия?”. Если ответим “У”, то получим следующий ответ:

СТ-ЗАМЫКАНИЕ СИСТЕМЫ:

- $1 \rightarrow \{2, -3\}$
- $-1 \rightarrow \{\}$
- $2 \rightarrow \{-3\}$
- $-2 \rightarrow \{-1\}$
- $3 \rightarrow \{-1, -2, -4\}$

- $-3 \rightarrow \{\}$
- $4 \rightarrow \{-3\}$
- $-4 \rightarrow \{\}$

Здесь собраны вместе и исходные посылки и следствия. Все это названо “СТ-замыканием”. Происхождение названия нам сейчас знать необязательно. Просто запомним его. Важно лишь

знать, что эти строчки являются одним из способов математического описания ориентированных графов. Чтобы отобразить этот граф на рисунке, надо произвольно нанести на листе бумаги точки, в каждой из которых будет какой-то термин или его отрицание, обозначить эти точки и соединить их линиями со стрелками в соответствии с нашим СТ-замыканием. Но мы для большей ясности сначала построим граф исходных посылок (схема 1). Если дополним этот граф следствиями, то получим граф

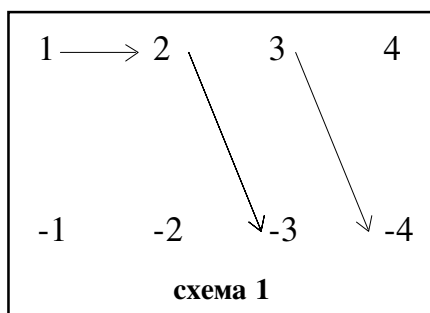


схема 1

СТ-замыкания нашей системы (схема 2).

Такие изображения, когда число терминов в системе невелико, могут послужить нам для структурного анализа полученной системы. Но при большом числе терминов это весьма затруднительно. Поэтому в программе предлагается воспользо-

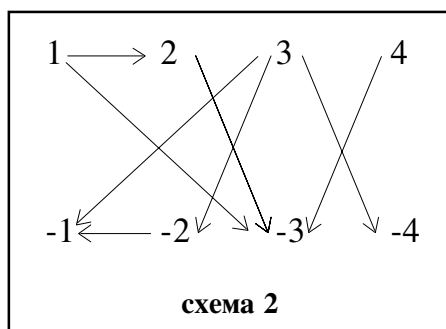


схема 2

ваться более простым изображением системы, которая называется “диаграммой Хассе”. В диаграмме Хассе убраны многие лишние стрелки, которые можно легко получить из оставшихся. Образно говоря, диаграмма Хассе - это “каркас” системы. Этот “каркас” мы получаем в виде следующего сообщения:

ДИАГРАММА ХАССЕ СИСТЕМЫ:

- 1 \rightarrow {2}
- 1 \rightarrow {}
- 2 \rightarrow {-3}
- 2 \rightarrow {-1}
- 3 \rightarrow {-2,-4}
- 3 \rightarrow {}
- 4 \rightarrow {-3}
- 4 \rightarrow {}

Для удобства мы переставим при отображении диаграммы Хассе некоторые точки для того, чтобы эта диаграмма предстала перед нашим взором в виде непересекающихся цепочек, а все цепочки были направлены в одну сторону (схема 3).

И, наконец, последнюю информацию, которая характеризует нашу систему, мы получим из следующего сообщения:

МИНИМАЛЬНОЕ МНОЖЕСТВО АКСИОМ СИСТЕМЫ:

(из каждой пары посылок нужно выбрать любую одну)

- 1) 1 \rightarrow 2; -2 \rightarrow -1
- 2) 2 \rightarrow -3; 3 \rightarrow -2
- 3) 3 \rightarrow -4; 4 \rightarrow -3

Здесь мы увидим, что число посылок в нашей системе не уменьшилось. Если мы выберем из каждой строки первые предложения, то получим в точности те посылки, которые и были нами введены. Но вполне возможны случаи, когда вначале вводятся “лишние” посылки, которые могут быть получены как следствия остальных. Так что этой возможностью программы пренебрегать не следует.

Пример 2.

Вспомним утверждение Сидорова о том, что все его друзья не скандалисты и дополним этим утверждением систему из примера 1. Тогда получим следующую систему исходных посылок:

- 1 \rightarrow {2,-4}

- 2 \rightarrow {-3}
- 3 \rightarrow {-4}.

Мы опустим все этапы анализа этой системы. В новой системе нет коллизий, и это вполне закономерно, так как мы ввели в нее новое предложение из множества “дополнительных” предложений. При проигрывании этой системы на машине стоит

обратить внимание на то, что, во-первых, число “дополнительных” предложений существенно уменьшилось, а диаграмма Хассе системы изменилась (схема 4).

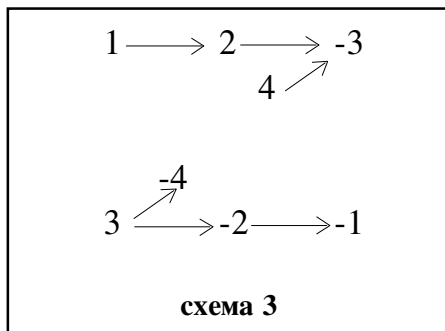


схема 3

ПРИМЕРЫ СИСТЕМ С ПРОТИВОРЕЧИЕМ

Вполне возможно, что найдется скептик, который задаст следующие

вопросы: “А на каком основании мы получаем всю эту информацию о системе? Почему мы получаем именно эти следствия, а не другие?” Вкратце ответ такой: все исходные посылки и получаемые следствия мы записываем в виде включений множеств. Оказывается, в таком виде можно представить многие предложения естественного и научного языка. К таким предложениям относятся, в частности:

- 1) все типы категорических суждений Аристотелевской силлогистики;
- 2) предложения, выраженные в форме определений терминов;
- 3) условные предложения, выраженные в формах “Если X, то Y” или “событие Y есть необходимое условие существования события X”.

Из этого далеко неполного перечня ясно, что область применения этой модели рассуждений весьма широка. С ее помощью можно выразить не только соотношения между точными математическими или техническими терминами, но и соотношения между понятиями, смысл которых, если их рассматривать изолированно, характеризуется большей или

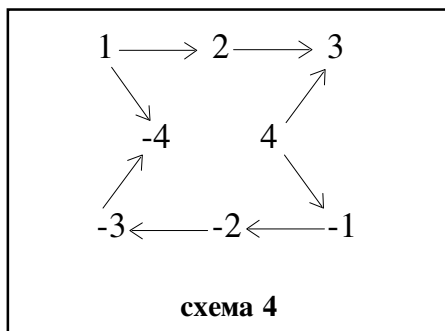


схема 4

меньшей степенью неопределенности. Мы можем, например, не знать всех друзей Сидорова или однозначного определения скандалиста, но в предложении, которое сводится к соотношению включения между множествами, появляется точное математическое соотношение, которое мы просто обязаны использовать с максимальной отдачей.

Проблема лишь в том, чтобы выявить все аналитические возможности этой модели. Для решения этой проблемы необходимо выбрать подходящий математический аппарат. В качестве такого математического аппарата была выбрана строгая и точная математическая система, которая носит название “алгебра множеств”. Это не совсем то, что известно под названием “теория множеств”, но о различии и сходстве этих математических систем мы здесь говорить не будем. О них кое-что сказано в книге [1]. Сейчас готовится к публикации цикл работ, в которых эта проблема освещается более подробно.

Такой выбор математического аппарата однозначно определяет правила вывода и методы анализа системы, которые непосредственно следуют из законов алгебры множеств. Заодно в этой системе используются аналитические и алгоритмические возможности теории графов [2].

Не вдаваясь в теоретические подробности, рассмотрим, какие правила вывода в ней используются. Этим правил всего три.

Первое правило - **двойного отрицания** (дополнения): для любого термина X справедливо $\neg(\neg X) = X$.

Второе правило - **контрапозиции**: предложение $X \rightarrow Y$ равносильно предложению $\neg Y \rightarrow \neg X$. Например, если в системе есть предложение $A1 \rightarrow \neg A2$, то в этой же системе должно быть и предложение $\neg(\neg A2) \rightarrow A1$ или, если использовать первое правило, $A2 \rightarrow A1$.

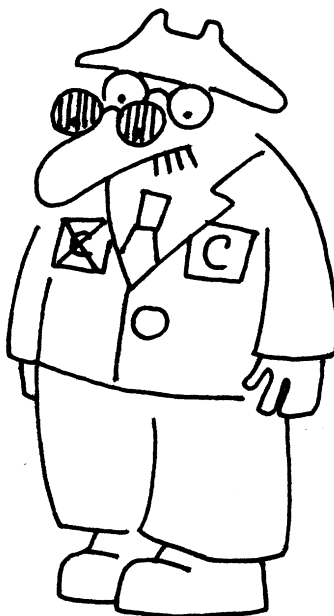
Третье правило - **транзитивности**: если

в системе имеются предложения $X \rightarrow Y$ и $Y \rightarrow Z$, то предложение $X \rightarrow Z$ также должно быть в этой системе.

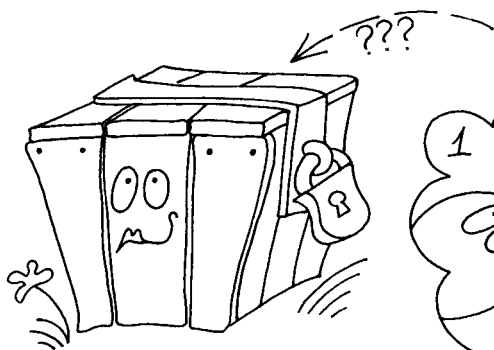
Эти правила аналогичны числовым соотношениям, если стрелку понимать как отношение “меньше”, а знак “-” понимать как просто “минус”. Но это всего лишь аналогия. Например, для множеств вполне возможна истинность предложения $A1 \rightarrow \neg A2$ ($A1$ включено в дополнение $A2$). Для чисел же положительное число не может быть меньше отрицательного.

Расставив стрелки между терминами или их дополнениями, мы получаем в виде посылок точную математическую систему. Но мы не всегда можем сразу же правильно сформулировать эти посылки или дать точное и бесспорное определение некоторому термину. И это вполне закономерно - часто нам приходится иметь дело с системами, о которых нам мало что известно. Мы можем даже ошибиться в своих определениях и при этом получить вполне “приличную” систему без всяких видимых “неприятностей”. Но в некоторых случаях можно все же с помощью программы получить вполне убедительные доказательства ошибочности наших посылок и попытаться привести их к более совершенному виду.

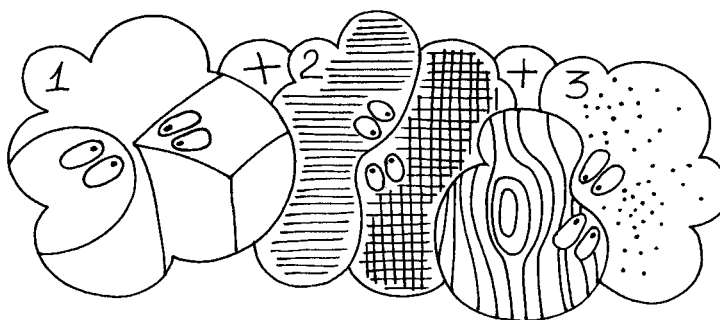
Первое, что мы можем сделать, это проверить адекватность исходной системы посылок. Возможны случаи, когда исходные посылки кажутся вполне правдоподобными, но когда мы перейдем к сопоставлению полученных следствий с реальностью, может оказаться так, что некоторые следствия противоречат этой реальности. Многочисленные примеры такого анализа традиционных взглядов о происхождении и развитии этносов можно найти в прекрасных написанных книгах Л.Н.Гумилева [3 и др.]. Заодно, читая их, можно значительно обогатить свое знание



Все друзья Сидорова не его друзья!



Пусть в закрытом ящике содержится неизвестное число предметов.



Известно, что они различаются по форме, по цвету и по материалу.

и понимание истории. Но мы рассмотрим этот аспект проверки на более простых примерах.

Пример 3.

Предположим, что наш предыдущий пример посмотрел Мешалкин, у которого брат собирается учиться на психолога, и обратил наше внимание на то, что третье предложение не научно: более правильным будет предложение “Все, не склонные скандалить, уверены в себе”. В “машинном” варианте это означает, что нам нужно предложение 3 \rightarrow {-4} заменить на обратное ему -4 \rightarrow {3}. С учетом этого введем в машину следующие посылки:

1 \rightarrow {2,-4}
2 \rightarrow {-3}
-4 \rightarrow {3}.

В ответ получим следующее сообщение: “Система противоречива! Противоречивые пары: 1 \rightarrow -1”

Получается, что множество друзей Сидорова включено в множество тех, кто не является его друзьями. Другими словами: все друзья Сидорова не его друзья. Можно поступить просто и считать, что у Сидорова нет друзей. Тогда весь наш “возможный мир” (или универсум) представлен теми, кто не относится к друзьям Сидорова. Тем самым мы должны исключить из системы всякое упоминание о друзьях Сидорова, то есть в нашем случае изъять из обращения первую посылку и заодно термины A1 и -A1.

Но можно попытаться откорректировать эту систему. Правда, в чем тут дело, пока не совсем понятно: то ли Сидоров плохо знает своих друзей, то ли Мешалкин, несмотря на свой солидный научный багаж, в чем-то все же неправ.

Можно попытаться сделать еще одну проверку этой системы: заменим категоричное высказывание о друзьях Сидорова 1 \rightarrow -4 на более осторожное 5 \rightarrow {1,-4} (некоторые друзья Сидорова не скандалисты). Оказывается, что и такая замена также не позволяет получить “нормальную” систему. Введем в машину посылки (не забывайте при этом, что число исходных “позитивных” терминов здесь будет уже не 4, а 5):

1 \rightarrow {2}
5 \rightarrow {1,-4}
2 \rightarrow {-3}
-4 \rightarrow {3}

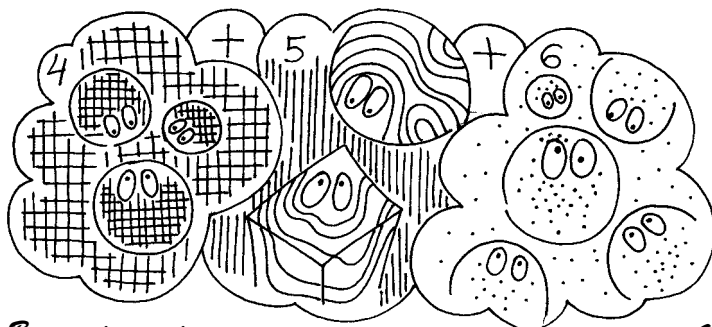
и получим сообщение о том, что система противоречива. Так что скорее всего здесь виноват Мешалкин.

В последних примерах мы столкнулись с первой коллизией в рассуждениях - с противоречием. В этом случае система “настоятельно требует”, чтобы мы убрали из нее некоторые множества или события как несуществующие. Но если мы уверены в существовании этих множеств или событий, то можно попытаться их “спасти”, произведя некоторые коррективы в исходных посылках. Как раз такой анализ (правда, в менее формализованном виде) довольно часто встречается в упомянутых выше книгах Л.Н.Гумилева. Перейдем теперь к знакомству с другой коллизией.

ПРИМЕРЫ СИСТЕМ С ЦИКЛОМ

Пример 4

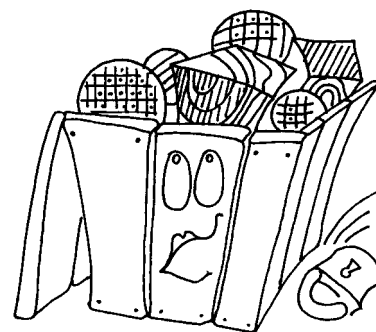
Пусть в закрытом ящике содержится неизвестное число предметов. Известно, что они различаются по форме (шар или кубик), по цвету (белый или красный) и по



Все шары - красные.

Все деревянные предметы - белые.

Все пластмассовые предметы - шары.



Ответ: в ящике могут находиться красные пластмассовые шары и белые деревянные кубики.

материалу (дерево или пластмасса). Известны также, следующие соотношения между признаками:

- 1) все шары красного цвета;
- 2) все деревянные предметы белого цвета;
- 3) все пластмассовые предметы шары.

Спрашивается, что нового можно узнать о предметах в ящике на основе этих сведений?

Построим модель. Тут ясно, что “возможный мир” - это предметы в ящике. Поскольку признаки взаимоисключающие, то можно ввести следующие обозначения: A1 - шары, A2 - предметы белого цвета, A3 - пластмассовые предметы. Тогда -A1 - кубики, -A2 - предметы красного цвета, -A3 - предметы из дерева. “Машинный” вариант посылок тогда будет таким:

1 —> {-2}

-3 —> {2}

3 —> {1}

Теперь, если мы запустим программу и введем эти посылки, то получим следующий ответ:

В системе есть следующие множества равных терминов:

Множество 1: {1,-2,3}

Множество 2: {-1,2,-3}.

В данном случае это означает, что в ящике может находиться лишь два класса предметов: красные пластмассовые шары и белые деревянные кубики. А получился такой вывод потому что на одном из этапов построения следствий в системе появились циклы

1 —> -2 —> 3 —> 1 и -1 —> 2 —> -3 —> -1.

Если мы стрелку понимаем как отноше-

ние включения множеств, то наличие цикла говорит о том, что все множества, входящие в цикл, должны быть равны друг другу.

Спрашивается, почему ситуацию с циклом мы назвали коллизией - ведь в данном случае эта ситуация пошла только на пользу: распознаны эквивалентные множества? Но ведь и первая коллизия тоже может использоваться не только для критики, но для распознавания: если мы уверены в исходных посылах, а система утверждает, что некоторые множества нашей системы не существуют, то, может быть, в некоторых случаях более конструктивно согласиться с этим, чем изменять правдоподобные посылки.

Критический смысл коллизии цикла можно понять так: мы строим сложные умозаключения, предполагая, что все используемые в ней термины различны, а система говорит, что мы получили лишь то, что “масло масляное” и ничего больше. То есть тем самым мы распознали ситуацию, которая в логике называется “круг в доказательстве” и считается логической ошибкой.

Теперь становится более понятным смысл “дополнительных” предложений при проверке неполноты системы: при вводе любого “дополнительного” предложения в системе невозможны ни коллизия противоречия, ни коллизия цикла. Стоит отметить, что образцовые логические системы рассуждений характеризуются наименьшей мерой неполноты. С другой стороны, “рыхлые” системы с большим количеством “дырок” свидетельствуют о незавершенности логического анализа этой системы.

ЗАКЛЮЧЕНИЕ

Представленная в программе система моделирования и анализа рассуждений намного перекрывает силлогистику Аристотеля и к тому же в отличие от нее имеет строгое математическое обоснование.

При сравнении ее с системами логического вывода на основе исчисления высказываний и предикатов оказывается, что эта система более проста по исполнению. В математической логике для того, чтобы осуществить логический вывод, необходимо представить множество исходных посылок P_1, P_2, \dots, P_n и предполагаемое следствие G . Тогда вывод производится как поиск ответа на вопрос “Является ли G следствием этих посылок?”.

В предлагаемой системе предусматривается более простая постановка задачи логического вывода: нам не нужно предъявлять системе непонятно откуда взятое предполагаемое следствие - все возможные следствия система находит сама. И заодно позволяет легко проверить совместимость исходных посылок. И хотя структуры этой системы менее универсальны, чем структуры математической логики, она вполне подходит для решения многих и порой довольно сложных задач, связанных с детальным анализом естественных рассуждений, и при решении этих задач может успешно конкурировать с системами логического вывода, предусмотренными в математической логике.

Литература.

1. Кулик Б.А. Логические основы здравого смысла/под ред. Д.А.Поспелова. - СПб. Политехника, 1997.
2. Кристофидес Н. Теория графов. Алгоритмический подход. - М.: Мир, 1978.
3. Гумилев Л.Н. Этногенез и биосфера Земли. Л.: Гидрометеоиздат, 1990.

ОТ РЕДАКЦИИ

Редакция журнала объявляет конкурс на лучшую модель рассуждений, которую можно анализировать с помощью программы Б.А.Кулика. Тема рассуждений может быть самая разная - от соотношений между объектами математики до модели фантастической ситуации. Модель должна содержать не менее шести “позитивных” терминов и в окончательном, полученном после анализа виде быть полной или почти полной, то есть иметь при проверке неполноты малое число “дополнительных” предложений. Лучшие модели задач будут опубликованы в журнале. Программа находится на дискете, прилагаемой к журналу в каталоге LOGIC.

*Кулик Борис Александрович,
кандидат технических наук,
сотрудник Института проблем
машинovedения РАН.*

НАШИ АВТОРЫ