

УДК 681.3.07

К ВОПРОСУ АВТОМАТИЗИРОВАННОЙ ОБРАБОТКИ ЛОГИЧЕСКОЙ ИНФОРМАЦИИ

С.Т. Кузнецов, Н.В. Столярчук, Т.Ф. Шмелёва

Государственная летная академия Украины

Предложен способ анализа и формализации задач решение которых традиционно считается прерогативой разумной деятельности человека. Автоматизация этой деятельности – одна из основных задач искусственного интеллекта. Весьма актуальной здесь является проблема создания информационного интерфейса человек-техническое устройство. На практике всегда нелегко поместить логическую обработку информации в «прокрустово ложе» компьютерных операций. Нагромождение информационных объектов бывает таковым, что быстродействие электроники вступает в противоречие со сложностью формулировки задач на доступном машине языке.

The way of analyses and formalization of tasks, which are traditionally considered to be solved by a human sense activity, is proposed. The problem of creating an information human-machine interface is sharply actual in this situation. Overlaying of information objects could be so that quick-operation of electronics may conflict with a complicated formulation of tasks on a language which is clear for a machine.

В данной статье рассмотрено применение алгоритма поиска решения интеллектуальных задач путем исключения противоречий. Алгоритм реализован в виде специальной компьютерной программы ViloR (виртуальный логический решатель – см. листинг 1).

На основании ограниченного, но функционально полного набора операций алгебры высказываний, как оказалось, можно простым и естественным образом подготовить информационный запрос на решение задачи, после чего компьютерная программа решение находит автоматически. В определенной степени неожиданным оказалось то, что довольно просто переборный алгоритм для решения невычислительных задач реализуется посредством применения стандартных операций, имеющих практически во всех широко используемых языках программирования.

Суть вопроса поясним на трех конкретных задачах.

Задача о расписании (взята из [1]).

Рассмотрим упрощенный учебный план, где неделя включает всего три учебных дня – понедельник, среду и пятницу, причем каждый день содержит не более трех пар учебных часов. В течении недели учащиеся должны иметь три пары учебных часов по математике, две – по физике и по одной – по химии, истории и физкультуре. При этом:

1) Математик настаивает, чтобы его часы никогда не были последними и по крайней мере два раза – первыми;

2) Физик желает, чтобы его часы также не были последними; по крайней мере один раз он хочет иметь первую пару часов; в среду он должен быть

свободен первые два часа, а в пятницу, напротив того, может работать лишь первые два часа;

3) Историк может преподавать лишь в понедельник в течение первых четырех часов или в среду в течение третьего и четвертого часов; кроме того, он не желает, чтобы его занятия непосредственно предшествовали физкультуре;

4) Химик настаивает, чтобы его занятия происходили не в пятницу и не в те дни, когда учащиеся занимаются физикой;

5) Занятия по физкультуре проводятся на стадионе, и поэтому естественно требовать, чтобы они были последними в свой день; кроме того, физкультурник в пятницу занят на другой работе;

6) Естественно требовать, чтобы в течение каждого учебного дня у учащихся было не больше двух часов занятий по одному и тому же предмету;

7) Свободные от занятий два часа в рамках учебной недели из $3 \times 3 = 9$ пар часов (из числа которых заняты лишь $3 + 2 + 1 + 1 + 1 = 8$ пар часов) должны приходиться на последнюю пару часов в пятницу или первую пару часов в понедельник.

Как можно составить расписание с соблюдением всех поставленных условий?

Решение легко находится путем простых и понятных рассуждений, если задаться целью последовательно отсеивать все невозможные варианты, оставляя возможные.

Будем рассуждать по пунктам.

1. Две первые пары заняты математикой, а одна физикой. Следовательно, свободные часы приходятся на последнюю пару в пятницу.

2. В пятницу могут работать только физик и математик, причем физика может быть только на первой паре. Для математики остается вторая.

3. Расписание на пятницу готово. Кроме того становится ясно, что на первых парах, как в понедельник, так и в среду должна быть математика.

4. Вторые пары как в понедельник, так и в среду остаются для физики и истории, так как физик так и историк не должны работать на третьих парах (физик не желает, а историк не может).

5. После истории не должна быть физкультура, а после физики не должна быть химия. Следовательно, физкультура следует за физикой, а химия за историей.

Таким образом, существуют только два способа составления расписания, удовлетворяющие всем поставленным требованиям:

1) понедельник: математика, история, химия; среда: математика, физика, физкультура; пятница: физика, математика, свободные часы;

2) понедельник: математика, физика, физкультура; среда: математика, история, химия; пятница: физика, математика, свободные часы.

Автор задачи приходит к тому же выводу путем виртуозного применения аппарата математической логики.

Вторая, рассматриваемая нами задача «Задача об утреннем купании» (взята тоже из [1]).

Предположим, что в санатории на берегу моря отдыхают отец O , мать M , сын C и две дочери D_1 и D_2 . До завтрака члены семьи часто купаются в море, причем известно, что если отец утром отправляется купаться, то с ним обязательно идут мать и сын; если сын идет купаться, то его сестра D_1 отправляется вместе с ним; вторая дочь D_2 купается тогда и только тогда, когда купается мать, и каждое утро купается по крайней мере один из родителей. Если в воскресенье утром купалась в море лишь одна из дочерей, то кто из членов семьи в это утро ходил на море?

Для нас данная задача интересна тем, что мы можем весьма просто показать, как без формального владения алгеброй логики, без умения программировать вообще и без применения ПРОЛОГ-подобного языка в частности можно компьютеру «растолковать» условие и он найдет решение.

Вот как это выглядит. Заметим, что в задаче пять величин: O , M , C , D_1 и D_2 каждая из которых может принимать одно из двух значений – «1» - купается; «0» - не купается.

Теперь внимательно прочитывая условие задачи записываем его последовательно в виде противоречий.

1. \overline{OM} - не может быть, чтобы отец купался, а мать – нет. 2. \overline{OC} - не может быть, чтобы отец купался, а сын – нет. Далее: 3. $\overline{CD_1}$; 4. $\overline{MD_2}$; 5. $\overline{MD_2}$; 6. \overline{OM} ; 7. $\overline{D_1D_2}$; 8. $\overline{D_1D_2}$.

Порядок записи условий не имеет значения. Однако для ввода в компьютер данных нужно произвести еще две весьма несложные операции:

1. Задать порядок следования величины при помощи двоичных весовых коэффициентов: $O - 1$, $M - 2$, $C - 4$, $D_1 - 8$, $D_2 - 16$.

2. Представить противоречие в арифметической форме:

1. - 1,2; 2. - 1,4; 3. - 4,8; 4. - 2,16; 5. - 16,2; 6. - 0,3; 7. - 0,24; 8. - 24,0.

Таким образом, каждое противоречие имеет довольно простой формат – пара целых неотрицательных чисел, разделенных запятой. До запятой сумма двоичных «весов» для «единиц»; после – «для нулей». Если что-то отсутствует – пишем «0». После введения данных в компьютер получаем единственное решение:

$$1 - 0 1 0 0 1$$

То есть в воскресное утро купались мать и вторая дочь.

Могли бы мы эту задачу решить так же как и первую? Автор задачи И.М.Яглом [1] отмечает, что решение можно заменить «... «неформальными рассуждениями» (т.е. рассуждениями, опирающимися не на строго сформулированные законы логики, а на «здравый смысл») такого рода: «Если бы отец в воскресенье утром пошел купаться, то с ним пошли бы мать

и сын; но вместе с сыном пошла бы и первая дочь, а вместе с матерью – вторая дочь; а так как в воскресенье из двух дочерей на море была лишь одна, то отец не мог пойти купаться» и т.д. Но нетрудно понять, что рассуждения такого рода на самом деле также опираются на строгие законы алгебры высказываний и пресловутый «здравый смысл» как раз и заключается в точном следовании этим законам».

Тут мы сделаем довольно сильное, возможно спорное утверждение: *в нашем случае для получения решения задачи нам практически не понадобилось рассуждать*. На самом деле компьютер отсеял все противоречивые варианты, подобно тому, как решето Эратосфена отсеивает все составные числа.

И наконец, приведем задачу из [2]. Вернее самую интересную её часть. Речь идет о назначении на четыре вида работ четырех из восьми кандидатов по критерию минимальных затрат (или максимальной выгоды). Это задача о назначении – широко известная задача целочисленного линейного программирования.

Задача становится более трудной, а её математическая модель перестает быть в чисто виде линейной, если ввести дополнительные ограничения по совместимости работников, учитывая, например, их пожелания.

Пусть в нашей задаче *A* согласен работать с кем угодно. *B* не будет работать, если не будет *C*. *C* не хочет работать с *E*. *E* согласен работать с кем угодно. *D* не будет работать без *H*. *F* не будет без *G* работать с *D*, а без *D* не будет работать с *C*. *G* не будет работать, если будут *B* и *C* оба, а кроме того, он не желает работать ни с *A* ни с *E*. Чтобы *H* дал согласие работать, надо взять либо *B*, либо *F*. Кроме того *H* не будет работать с *C*, если не будет *G* и не желает работать ни с *A* ни с *E*. Считается, что хотя работники и будут работать на различных работах, они составляют один слаженный коллектив и, в первую очередь, должны неукоснительно соблюдаться требования совместимости.

Становится понятным, что прежде всего нужно отобрать все бесконфликтные четверки, потом для каждой такой четверки найти лучшее решение, решая каждый раз классическую задачу о назначениях. После этого из всех решений можно будет выбрать оптимальное по критерию максимума-минимума баллов.

Запись противоречий не содержит принципиальных трудностей. После ввода в компьютер противоречий и сообщения численности коллектива (четыре человека) получаем единственное решение:

$$1 - 0 0 0 1 0 1 1 1$$

Таким образом, единственно подходящая четверка

D, F, G и H

* * *

В настоящей статье мы стремились показать, что «виртуальный логический решатель» способен на гораздо большее, чем, например, «логическое пианино» У.С.Джеванса (1835-1882) и ему подобные устройства, включая их компьютерные эмуляторы. Это с одной стороны. С другой стороны, мы надеемся, что простота и естественность ввода данных вызовет больший интерес, чем специальные (например, ПРОЛОГ-подобные) языки программирования. Большим преимуществом на наш взгляд является то, что умение программировать, здесь, не требуется. Это мы говорим потому, что существует много специальных программ, каждая из которых решает конкретную логическую задачу. В нашем же случае речь идет о целом классе задач решаемых по единому алгоритму при помощи одной специальной программы.

Листинг 1 – Программа ViLoR (Виртуальный логический решатель).
Язык программирования - GW Basic.

```
5   REM VILOR      avtor Kuznetsov
10  DIM T1(200), T0(200), T(14)
20  FOR I=0 TO 14:T(I)=2^I:NEXT I
30  CLS:K=0:FOR I=0 TO 200:T1(I)=0:T0(I)=0:NEXT I: INPUT "LOGPEREM";L
40  IF L<2 OR L>15 OR L<>INT(L) THEN 30ELSE N=2^L-1
50  FOR I=1 TO L:PRINT I;"-";T(I-1):NEXT I
60  PRINT "LOGPROTIVORECHIJA."
70  M=0
80  M=M+1
90  PRINT M;"+";
100 INPUT T1 (M), T0 (M)
110 IF T1 (M)>0 OR T0 (M)>0 THEN 80 ELSE M=M-1
120 F=0
130 INPUT "UPRAVLENIE (69-OBYCHNO/N-AKTPEREM/25-REDACT/33-
PROSMOTR)";U
140 IF U>L AND U<>69 AND U<>25 AND U<>77 AND U<>777 AND
U<>33 THEN 130
150 IF U=77 THEN 30 ELSE IF U=777THEN 420
160 IF U<>25THEN 210
170 INPUT "NIZ, IZ1, IZ0"; NIZ, IZ1, IZ0: IF NIZ> M THEN M=NIZ
180 IF NIZ=0 AND IZ1=0 AND IZ0=0 THEN 130
190 IF (NIZ>0 AND IZ1=0 AND IZ0=0) OR (NIZ=0 AND (IZ1>0 OR
IZ0>0)) THEN PRINT "VNIMANIE!": GOTO 170
200 T1 (NIZ)= IZ1 : T0 (NIZ)= IZ0 : GOTO 170
```

```
210 IF U <> 33 THEN 230
220 INPUT "FLAG (0/1/2)="; F: IF F <> 0 AND F <> 1 AND F <> 2 THEN 220
    ELSE 130
230 FOR I=0 TO N
240 IF I=0 THEN A=0: ED=0: GOTO 280
250 T=((I XOR (I-1))+1)/2
260 B=A XOR T; IF B > A THEN ED+1 ELSE ED=ED-1
270 A=B
280 IF U <> 69 AND ED <> U THEN 400
290 FL=0
300 FOR P=1 TO M
310 IF (A XOR T1(P))+T1(P) <> A OR (A XOR T0(P))-T0(P) <> A THEN
    330
320 FL=P: P=M
330 NEXT P
340 IF FL > 0 AND F <> 2 THEN 400 ELSE K=K+1: PRINT K; "-";
350 FOR J=0 TO L-1
360 Q=A XOR T(J): IF Q < A THEN PRINT 1; ELSE PRINT 0;
370 NEXT J: IF F <> 2 THEN PRINT " ED="; ELSE PRINT " (;FL;)"
380 IF F=0 THEN 400
390 R=INKEYS: IF R=" " THEN 390
400 NEXT I: IF K=0 THEN PRINT "PUSTO!!!"
410 BEEP: K=0: GOTO 130
420 PRINT "K O N E C !": END
Ok
```

БИБЛІОГРАФІЯ

1. Яглом И. М.. Алгебры Буля. В кн. О некоторых вопросах современной математики и кибернетики. Из-во «Просвещение». Москва. 1965г.
2. Бондарь О.П., Кузнецов С.Т., Столярчук Н.В.. Задача о назначениях, критерии совместимости работников и справедливости назначений. В научно-теретическом журнале «Искусственный интеллект». 4' 2005.

*Кіровоградський державний педагогічний
університет ім. В.Винниченка*

Надійшло 14 березня 2006 р.