Journal of Automated Reasoning **2** (1986) 191–216.

Problem Corner:

Seventy-Five Problems for Testing Automatic Theorem Provers

FRANCIS JEFFRY PELLETIER
Department of Philosophy, University of Alberta, Edmonton, Alberta, Canada T6G 2E5.

(Received: 30 August 1985)

The purpose of this note is to provide a graduated selection of problems for use in testing an automatic theorem proving (ATP) system. Included in these problems were the ones I have used in testing my own ATP system (Pelletier 1982, and more recent updates to the system). Some of the problems, especially some of the more difficult ones described at the end of this note, are due to discussions with Len Schubert (Computing, Univ. Alberta), Alasdair Urquhart (Philosophy, Univ. Toronto), and Charles Morgan (Philosophy, Univ. Victoria).

People who have tried to compile lists of problems for ATPs in the past (for example, the Association for Automated Reasoning) have discovered that the production of such a list is difficult because (a) what is 'easy' for one system might not be for another, (b) researchers are understandably shy about saying what problems their ATP might have, unless they know that it is so difficult that *any* ATP will have trouble with it, (c) especially with problems at the 'easy end' of the scale, researchers are prone to think that *any* ATP system can prove it and so there is no call to write them up, (d) the goals of 'natural' system ATPs and resolution-based ATPs are different: the former tries to produce a 'natural' proof usually without prior conversion to clause form. This means that some problems, especially at the 'easy end' of the scale, will be trivial for resolution systems but difficult for 'natural' systems. On the other hand, proponents of 'natural' systems think that at the 'difficult end' of the scale, there will be problems within the grasp of the 'natural' systems which are beyond the reach of resolution systems. It is therefore difficult to even give a graduated scale of problems. All this has led to publication of *difficult* problems, but not to publication of lists of problems suitable for *developing* an ATP. The *locus classicus* of problems is McCharen *et al.* (1976) which contains a wide range of problems all in clause form. More recently, the *Journal of Automated Reasoning* has instituted its 'Problem Corner'. A notable item herein is Lusk and Overbeek (1985). All of us involved in ATP wish to encourage others – graduate students, for example – to look into the field. But where is such a person to start? It is with such neophyte

ATPers in mind that the following list is offered. None of these problems will be the sort whose solution is, of itself, of any mathematical or logical interest. Such 'open problems' are regularly published in the *Newsletter* of the Association for Automated Reasoning. Most (but not all) of my problems can be found in elementary logic textbooks – but they have been chosen either because logic students find them difficult, or because previous ATP systems have reported difficulties in establishing them, or because they have some interesting connection with other areas of mathematics (such as set theory). The judgements of difficulty are my own, and are based primarily on my years of teaching elementary logic (so the judgements reflect how difficult beginning students find the problems). The scales are from 1 (easiest) to 10, and are relativized to the kind of problem involved. Thus; a '9' in the propositional logic section might in fact be easier than a '4' in the monadic logic section. I give problems in both 'natural' and clause form. The negated-conclusion clause form has eliminated tautologous clauses (but see the remark below in the 'Acknowledgement' section). As mentioned, many of the 'easier' problems – especially in propositional calculus – are trivial in resolution systems. Still, 'natural' systems might find them diverting. In either case, however, I would think that any ATP ought to produce an explicit proof, detailing what preceding formulas/clauses were invoked and (in the case of 'natural' systems) what rules of inference were used. When comparing two ATPs for efficiency (say by measuring CPU time on identical machines) it is of course mandatory that the systems should include the cost of conversion to clause form since it is only very rarely that problems 'in the real world' are presented in clause form.

A Word on Notation

$\rightarrow$: if-then
$\neg$: not
$+$: or
&: and
$\leftrightarrow$: if and only if
A: for all
E: there exists
$p, q, r, s \ldots$ (perhaps with subscripts): propositional (sentence) letters; that is, 0-place predicate letters.
$F, G, \ldots P, Q, \ldots$ (perhaps with subscripts): predicate letters; context is used to determine the adicity.
$x, y, z, w, \ldots$ (perhaps with subscripts): (individual) variables.
$a, b, c, \ldots$ (perhaps with subscripts): individual constants; that is, 0-place function symbols.
$f, g, h \ldots$ (perhaps with subscripts): function symbols; context is used to determine the adicity.

Precedence

$\neg$, A, E highest
&, + medium
$\rightarrow$, $\leftrightarrow$ lowest

Associativity

& and + are allowed to take arbitrarily many conjuncts/disjuncts. For the most part, the notation is in common use and should provide no difficulties.

Propositional Logic

What follows here are some propositional logic theorems and arguments. They are given both in 'natural' form and in negated-conclusion clause form. Some are of historical interest (mostly having to do with the logical theorist), while others illustrate various 'tricks'.

	'Natural' form	*Negated-conclusion clause form*

1. (2pts) A biconditional version of the 'most difficult' theorem proved by the original Logic Theorist (Newell *et al.* 1957)

$(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$	$\neg p + q$ $\neg q + p$ $\neg q$ p

2. (2pts) A biconditional version of the 'most difficult' theorem proved by the new logic theorist (Newell and Simon 1972)

$\neg\neg p \leftrightarrow p$	p $\neg p$

3. (1pt) The 'hardest' theorem proved by a breadth-first logic theorist (Siklóssy *et al.* 1973)

$\neg(p \rightarrow q) \rightarrow (q \rightarrow p)$	p $\neg q$ q $\neg p$

4. (2pts) Judged by Siklóssy *et al.* (1973) to be 'hardest' of first 52 theorems of Whitehead and Russell (1910)

$(\neg p \rightarrow q) \leftrightarrow (\neg q \rightarrow p)$	$p + q$ $\neg q + \neg p$ $\neg q$ $\neg p$

5. (4pts) Judged by Siklóssy *et al.* to be 'hardest' of first 67 theorems of Whitehead and Russell (1910)

$((p + q) \rightarrow (p + r)) \rightarrow (p + (q \rightarrow r))$

$\neg q + p + r$

$\neg p$
q
$\neg r$

6. (2pts) The Law of Excluded Middle: can be quite difficult for 'natural' systems

$(p + \neg p)$

$\neg p$
p

7. (3pts) Expanded Law of Excluded Middle. The strategies of the original Logic Theorist cannot prove this

$(p + \neg\neg\neg p)$

$\neg p$
p

8. (5pts) Pierce's Law. Unprovable by Logic Theorist, and tricky for 'natural' systems.

$((p \rightarrow q) \rightarrow p) \rightarrow p$

p
$\neg p$

9. (6pts) A problem not solvable by unit resolution nor by 'pure' input resolution.

$[(p + q) \,\&\, (\neg p + q) \,\&\, (p + \neg q)] \rightarrow \neg(\neg p + \neg q)$

$p + q$

$\neg p + q$
$p + \neg q$
$\neg p + \neg q$

10. (4pts) A reasonably simple problem with premises, designed to see whether 'natural' systems correctly manipulate premises.

$q \rightarrow r$
$r \rightarrow (p \,\&\, q)$

$\neg q + r$
$\neg r + p$
$\neg r + q$

$\underline{p \rightarrow (q + r)}$
$p \leftrightarrow q$

$\neg p + q + r$
$p + q$

11. (1pt) A simple problem designed to see whether 'natural' systems can do it efficiently (or whether they incorrectly try to prove the $\rightarrow$ each way)

$(p \leftrightarrow p)$

p
$\neg p$

12. (7pts) The 'hardest' propositional problem found in Kalish and Montague (1964), according to Pelletier (1982)

$[(p \leftrightarrow q) \leftrightarrow r] \leftrightarrow [p \leftrightarrow (q \leftrightarrow r)]$

$p + q + r$
$\neg q + \neg p + r$
$\neg r + \neg p + q$
$\neg r + \neg q + p$
$p + \neg q + r$
$p + \neg r + q$
$q + r + \neg p$
$\neg r + \neg q + \neg p$

Distribution Laws can be very tricky for 'natural' systems. (They are *assumed* by resolution systems in the conversion to clause form). The next few problems list some

13. (5pts)

$[p + (q \,\&\, r)] \leftrightarrow [(p + q) \,\&\, (p + r)]$

$p + q$
$p + r$
$\neg p$
$\neg q + \neg r$

14. (6pts)

$(p \leftrightarrow q) \leftrightarrow ((q + \neg p) \,\&\, (\neg q + p))$

$\neg p + q$
$\neg q + p$
$\neg q + \neg p$
$p + q$

15. (5pts)

$(p \rightarrow q) \leftrightarrow (\neg q + q)$

$\neg p + q$
p
$\neg q$

16. (4pts) A surprising theorem of propositional logic

$(p \rightarrow q) + (q \rightarrow p)$

p
$\neg q$
q
$\neg p$

17. (6pts) A problem which appears not to be provable by Bledsoe *et al.* (1972). (For details of why not, see Pelletier (1982), p. 135f).

$((p \,\&\, (q \rightarrow r)) \rightarrow s) \leftrightarrow ((\neg p + q + s) \,\&\, (\neg p + \neg r + s))$

$\neg p + q + s$

$\neg p + \neg r + s$
p
$\neg q + r$
$\neg s$

Monadic Predicate Logic

Problems in the monadic predicate logic are not much more difficult than those in the propositional logic. All that's required is a method of handling quantifiers correctly (by finding appropriate instances or substitutions). The problems in this section are designed to test whether this happens.

18. (1pt)

$(\mathrm{E}y)(\mathrm{A}x)(Fy \rightarrow Fx)$

Fx
$\neg Ff(x)$

19. (3pts)

$(\mathrm{E}x)(\mathrm{A}y)(\mathrm{A}z)((Py \rightarrow Qz) \rightarrow (Px \rightarrow Qx))$

$\neg Pf(x) + Qg(x)$
Px
$\neg Qx$

20. (4pts)

$[(\mathrm{A}x)(\mathrm{A}y)(\mathrm{E}z)(\mathrm{A}w)((Px \,\&\, Qy) \rightarrow (Rz \,\&\, Sw)) \rightarrow ((\mathrm{E}x)(\mathrm{E}y)(Px \,\&\, Qy) \rightarrow (\mathrm{E}z)\, Rz)]$

$\neg Py + \neg Qz + Rf(y, z)$
$\neg Py + \neg Qz + Sx$
Pa
Qb
$\neg Rw$

21. (5pts) A moderately tricky problem, especially for 'natural' systems with 'strong' restrictions on variables generated from existential quantifiers.

$(\mathrm{E}x)(p \rightarrow Fx)$
$\underline{(\mathrm{E}x)(Fx \rightarrow p)}$
$(\mathrm{E}x)(p \leftrightarrow Fx)$

$\neg p + Fa$
$\neg Fb + p$
$p + Fx$
$\neg Fx + \neg p$

Some problems having to do with 'confinement' of quantifiers. These are often trivial in reslution systems because they are assumed in conversion to clause form.

22. (3pts)

$(\mathrm{A}x)(p \leftrightarrow Fx) \rightarrow (p \leftrightarrow (\mathrm{A}x)\, Fx)$

$p + \neg Fx$
$Fx + \neg p$

$Fy + p$
$\neg p + \neg Fa$
$Fy + \neg Fa$

23. (4pts)

$(\mathrm{A}x)(p + Fx) \leftrightarrow (p + (\mathrm{A}x)\, Fx)$

$p + Fx + Fy$
$p + Fx + Fb$
$\neg p$
$\neg Fa + p + Fy$
$\neg Fa + \neg Fb$

The following are some more tedious monadic logic problems from Kalish and Montague (1964).

24. (6pts)

$\neg(\mathrm{E}x)(Sx \,\&\, Qx)$
$(\mathrm{A}x)(Px \rightarrow (Qx + Rx))$
$\neg(\mathrm{E}x)Px \rightarrow (\mathrm{E}x)Qx$
$(\mathrm{A}x)(Qx + Rx \rightarrow Sx)$
$(\mathrm{E}x)(Px \,\&\, Rx)$

$\neg Sx + \neg Qx$
$\neg Px + Qx + Rx$
$Pa + Qb$
$\neg Qx + Sx$
$\neg Rx + Sx$
$\neg Px + \neg Rx$

25. (7pts)

$(\mathrm{E}x)\, Px$
$(\mathrm{A}x)(Fx \rightarrow (\neg Gx \,\&\, Rx))$
$(\mathrm{A}x)(Px \rightarrow (Gx \,\&\, Fx))$

$[(\mathrm{A}x)(Px \rightarrow Qx) +$
$(\mathrm{E}x)(Px \,\&\, Rx)]$
$(\mathrm{E}x)(Qx \,\&\, Px)$

Pa
$\neg Fx + \neg Gx + \neg Rx$
$\neg Px + Fx$
$\neg Px + Gx$
$\neg Px + Qx + Pb$

$\neg Px + Qx + Rb$
$\neg Qx + \neg Px$

26. (7pts)

$(Ex)Px \leftrightarrow (\mathrm{E}x)\, Qx$
$(\mathrm{A}x)(\mathrm{A}y)(Px \,\&\, Qy \rightarrow (Rx \leftrightarrow Sy))$
$[(\mathrm{A}x)(Px \rightarrow Rx) \leftrightarrow$
$(\mathrm{A}x)(Qx \rightarrow Sx)]$

$\neg Px + Qa$
$\neg Qx + Pb$
$\neg Px + \neg Qy + \neg Rx + Sy$

$\neg Px + \neg Qy + \neg Sy + Rx$
$\neg Px + Rx + \neg Qx + Sx$
$\neg Px + Rx + Pc$
$\neg Px + Rx + \neg Rc$
$Qd + \neg Qx + Sx$
$Qd + Pc$

$Qd + \neg Rc$
$\neg Sd + \neg Qx + Sx$
$\neg Sd + Pc$
$\neg Sd + \neg Rc$

27. (6pts)

$(\mathrm{E}x)(Fx \,\&\, \neg Gx)$
$(\mathrm{A}x)(Fx \rightarrow Hx)$
$(\mathrm{A}x)(Jx \,\&\, Ix \rightarrow Fx)$
$\underline{[(\mathrm{E}x)(Hx \,\&\, \neg Gx) \rightarrow (Ax)(Ix \rightarrow \neg Hx)]}$
$(\mathrm{A}x)(Jx \rightarrow \neg Ix)$

Fa
$\neg Ga$
$\neg Fx + Hx$
$\neg Jx + \neg Ix + Fx$

$\neg Hx + Gx + \neg Iy \neg Hy$
Jb
Ib

28. (8pts)

$[(\mathrm{A}x)Px \rightarrow (\mathrm{A}x)Qx]$
$[(\mathrm{A}x)(Qx + Rx) \rightarrow (\mathrm{E}x)(Qx \,\&\, Sx)]$
$\underline{[(\mathrm{E}x)Sx \rightarrow (\mathrm{A}x)(Fx \rightarrow Gx)]}$
$(\mathrm{A}x)(Px \,\&\, Fx \rightarrow Gx)$

$\neg Pa + Qx$
$\neg Qb + Qc$

$\neg Qb + Sc$
$\neg Rb + Qc$
$\neg Rb + Sc$
$\neg Sx + \neg Fy + Gy$
Pd
Fd
$\neg Gd$

29. (7pts)

$\underline{(\mathrm{E}x)Fx \,\&\, (\mathrm{E}x)Gx}$
$[(\mathrm{A}x)(Fx \rightarrow Hx) \,\&\, (\mathrm{A}x)(Gx \rightarrow Jx)] \leftrightarrow [(\mathrm{A}x)(\mathrm{A}y)(Fx \,\&\, Gy \rightarrow Hx \,\&\, Jy)]$

Fa
Gb
$\neg Fx + Hx + \neg Fy + \neg Gz + Hy$
$\neg Fx + Hx + \neg Fy + \neg Gz + Jz$
$\neg Fx + Hx + Fe + Gf$
$\neg Fx + Hx + Fe + \neg Jf$
$\neg Fx + Hx + \neg He + Gf$
$\neg Fx + Hx + \neg He + \neg Jf$
$\neg Gx + Jx + \neg Fy + \neg Gz + Hy$
$\neg Gx + Jx + \neg Fy + \neg Gz + Jz$
$\neg Gx + Jx + Fe + Gj$
$\neg Gx + Jx + Fe + \neg Jf$
$\neg Gx + Jx + \neg He + Gf$
$\neg Gx + Jx + \neg He + \neg Jf$
$Fc + \neg Fx + \neg Gy + Hx$

$Fc + \neg Fx + \neg Gy + Jy$
$Fc + Fe + Gf$
$Fc + Fe + \neg Jf$
$Fc + \neg He + Gf$
$Fc + \neg He + \neg Jf$
$Gd + \neg Fx + \neg Gy + Hx$
$Gd + \neg Fx + \neg Gy + Jy$
$Gd + Fe + Gf$
$Gd + Fe + \neg Jf$
$Gd + \neg He + Gf$
$Gd + \neg He + \neg Jf$
$\neg Hc + \neg Jd + \neg Fx + \neg Gy + Hx$
$\neg Hc + \neg Jd + \neg Fx + \neg Gy + Jy$
$\neg Hc + \neg Jd + Fe + Gf$
$\neg Hc + \neg Jd + Fe + \neg Jf$
$\neg Hc + Jd + \neg He + Gf$
$\neg \mathrm{H}c + \neg Jd + \neg He + \neg Jf$

30. (6pts)

$(\mathrm{A}x)(Fx + Gx \rightarrow \neg Hx)$
$\underline{(\mathrm{A}x)((Gx \rightarrow \neg Ix) \rightarrow Fx \;\&\; Hx)}$
$(\mathrm{A}x)Ix$

$\neg Fx + \neg Hx$
$Gx + Fx$
$\neg Gx + \neg Hx$
$Gx + Hx$
$Ix + Fx$
$Ix + Hx$
$\neg Ia$

31. (5pts)

$\neg(\mathrm{E}x)(Fx \;\&\; (Gx + Hx))$
$(\mathrm{E}x)(Ix \;\&\; Fx)$
$\underline{(\mathrm{A}x)(\neg Hx \rightarrow Jx)}$
$(\mathrm{E}x)(Ix \;\&\; Jx)$

$\neg Fx + \neg Gx$
$\neg Fx + \neg Hx$
Ia
Fa
$Hx + Jx$
$\neg Ix + \neg Jx$

32. (6pts)

$(\mathrm{A}x)(Fx \;\&\; (Gx + Hx) \rightarrow Ix)$
$(\mathrm{A}x)(Ix \;\&\; Hx \rightarrow Jx)$
$\underline{(\mathrm{A}x)(Kx \rightarrow Hx)}$
$(\mathrm{A}x)(Fx \;\&\; Kx \rightarrow Jx)$

$\neg Fx + \neg Gx + Ix$
$\neg Fx + \neg Hx + Ix$
$\neg Ix + \neg Hx + Jx$
$\neg Kx + Hx$
Fa
Ka
$\neg Ja$

33. (4pts) This is a monadic predicate logic formulation of problem (17) above.

$$(\mathrm{A}x)(Pa \,\&\, (Px \rightarrow Pb) \rightarrow Pc) \leftrightarrow (\mathrm{A}x)((\neg Pa + (Px + Pc)) \,\&\, (\neg Pa + (\neg Pb + Pc)))$$

$$\neg Pa + Px + Pc + Py$$
$$\neg Pc$$
$$\neg Pa + Px + Pc + \neg Pd + Pb$$
$$\neg Pa + \neg Pb + Pc$$
$$Pa$$
$$\neg Pe + Pb + \neg Pa + Px + Pc$$
$$\neg Pe + Pb + \neg Pd$$

34. (10pts) Andrew's Challenge (cf. de Champeaux (1979)). The problem is logically simple, but its size makes it difficult. (The conversion to clause form is left as an exercise for the reader, about 1600 clauses.)

$$[(\mathrm{E}x)(\mathrm{A}y)(Px \leftrightarrow Py) \leftrightarrow ((\mathrm{E}x)Qx \leftrightarrow (\mathrm{A}y)\,Py)] \leftrightarrow [(\mathrm{E}x)(\mathrm{A}y)(Qx \leftrightarrow Qy) \leftrightarrow ((\mathrm{E}x)Px \leftrightarrow (\mathrm{A}y)Py)]$$

Full Predicate Logic (without Identity and Functions)

Once again we start with some problems to determine whether the quantifiers are being handled properly.

35. (2pts)

$$(\mathrm{E}x)(\mathrm{E}y)(Pxy \rightarrow (\mathrm{A}x)(\mathrm{A}y)Pxy)$$

$$Pxy$$
$$\neg Pf(x, y)g(x, y)$$

36. (3pts)

$$\frac{(\mathrm{A}x)(\mathrm{E}y)Fxy \quad (\mathrm{A}x)(\mathrm{E}y)Gxy \quad \underline{(\mathrm{A}x)(\mathrm{A}y)(Fxy + Gxy \rightarrow (\mathrm{A}z)(Fyz + Gyz \rightarrow Hxz))}}{(\mathrm{A}x)(\mathrm{E}y)Hxy}$$

$$Fxf(x)$$
$$Gxg(x)$$
$$\neg Fxy + \neg Fyz + Hxz$$
$$\neg Fxy + \neg Gyz + Hxz$$
$$\neg Gxy + \neg Fyz + Hxz$$
$$\neg Gxy + \neg Gyz + Hxz$$
$$\neg Hax$$

37. (3pts)

$$\frac{(\mathrm{A}z)(\mathrm{E}w)(\mathrm{A}x)(\mathrm{E}y)[(Pxz \rightarrow Pyw) \,\&\, Pyz \,\&\, (Pyw \rightarrow (\mathrm{E}u)Quw)] \quad (\mathrm{A}x)(\mathrm{A}z)[\neg Pxz \rightarrow (\mathrm{E}y)Qyz] \quad \underline{(\mathrm{E}x)(\mathrm{E}y)Qxy \rightarrow (\mathrm{A}x)Rxx}}{(\mathrm{A}x)(\mathrm{E}y)Rxy}$$

$$\neg Pyx + Pf(x, y)g(x)$$
$$Pf(x, y), x$$
$$\neg Pf(x, y), g(x) + Qh(x, y), g(x)$$
$$Px, y + Qi(x, y), x$$
$$\neg Qx, y + Rz, z$$
$$\neg Ra, x$$

38. (4pts) Here is a full predicate logic version of problems (17) and (33). Conversion to clause form left as an exercise.

$$\{(\mathrm{A}x)[Pa \,\&\, (Px \rightarrow (\mathrm{E}y)(Py \,\&\, Rxy)) \rightarrow (\mathrm{E}z)(\mathrm{E}w)(Pz \,\&\, Rxw \,\&\, Rwz)] \leftrightarrow$$
$$(\mathrm{A}x)[(\neg Pa + Px + (\mathrm{E}z)(\mathrm{E}w)Pz \,\&\, Rxw \,\&\, Rwz)) \,\&$$
$$(\neg Pa + \neg(\mathrm{E}y)(Py \,\&\, Rxy) + (\mathrm{E}z)(\mathrm{E}w)(Pz \,\&\, Rxw \,\&\, Rwz))]\}$$

Some problems in set theory can be represented in first order logic using the predicate 'F' to stand for 'is and element of'. Here are some reasonably simple ones.

39. (3pts) Russell's paradox: there is no 'Russell set' (a set which contains exactly those sets which are not members of themselves)

$$\neg(\mathrm{E}x)(\mathrm{A}y)(Fyx \leftrightarrow \neg Fyy)$$

$$\neg Fxa + \neg Fxx$$
$$Fxx + Fxa$$

40. (5pts) If there were an 'anti-Russell set' (a set which contains exactly those sets which *are* members of themselves), then not every set has a complement.

$$(\mathrm{E}y)(\mathrm{A}x)(Fxy \leftrightarrow Fxx) \rightarrow$$
$$\neg(\mathrm{A}x)(\mathrm{E}y)(\mathrm{A}z)(Fxy \leftrightarrow \neg Fzx)$$

$$\neg Fxa + Fxx$$
$$\neg Fxx + Fxa$$
$$\neg Fx, f(x) + \neg Fyx$$
$$Fyx + Fx, f(x)$$

41. (6pts) The 'restricted comprehension axiom' says: given a set z, there is a set all of whose members are drawn from z and which satisfy some property. If there were a universal set then the Russell set could be formed, using this axiom. So given the appropriate instance of this axiom, there is no universal set.

$$\underline{(\mathrm{A}z)(\mathrm{E}y)(\mathrm{A}x)(Fxy \leftrightarrow (Fxz \,\&\, \neg Fxx))}$$
$$\neg(\mathrm{E}z)(\mathrm{A}x)Fxz$$

$$\neg Fx, f(y) + Fxy$$
$$\neg Fx, f(y) + \neg Fxx$$
$$\neg Fxy + Fxx + Fx, f(y)$$
$$Fxa$$

42. (6pts) A set is 'circular' if it is a member of another set which in turn is a member of the original. Intuitively all sets are non-circular. Prove there is no set of noncircular sets.

$$\neg(\mathrm{E}y)(\mathrm{A}x)(Fxy \leftrightarrow \neg(\mathrm{E}z)(Fxz \,\&\, Fzx))$$

$$\neg Fxa + Fxy + \neg Fyx$$
$$Fxf(x) + Fxa$$
$$Ff(x)x + Fxa$$

43. (5pts) de Champeaux (1979). Define set equality ('Q') as having exactly the same members. Prove set equality is symmetric.

$$\underline{(\mathrm{A}x)(\mathrm{A}y)(Qxy \leftrightarrow (\mathrm{A}z)(Fzx \leftrightarrow Fzy))}$$
$$(\mathrm{A}x)(\mathrm{A}y)(Qxy \leftrightarrow Qyx)$$

$$\neg Qxy + \neg Fzx + Fzy$$
$$\neg Qxy + \neg Fzy + Fzx$$
$$Ff(x, y), x + Ff(x, y), y + Qxy$$
$$\neg Ff(x, y), y + \neg Ff(x, y), x + Qxy$$

$Qab + Qba$
$\neg Qba + \neg Qab$

Here are some problems taken from Kalish & Montague (1964)

44. (3pts)

$(\mathrm{A}x)[Fx \rightarrow (\mathrm{E}y)(Gy \,\&\, Hxy) \,\&\, (\mathrm{E}y)(Gy \,\&\, \neg Hxy)]$
$\underline{(\mathrm{E}x)[Jx \,\&\, (\mathrm{A}y)[Gy \rightarrow Hxy]]}$
$(\mathrm{E}x)(Jx \,\&\, \neg Fx)$

$\neg Fx + Gf(x)$
$\neg Fx + Hx, f(x)$
$\neg Fx + Gg(x)$
$\neg Fx + \neg Hx, g(x)$
Ja
$\neg Gx + Hax$
$\neg Jx + Fx$

45. (5pts)

$(\mathrm{A}x)(Fx \,\&\, (\mathrm{A}y)[Gy \,\&\, Hxy \rightarrow Jxy] \rightarrow (\mathrm{A}y)(Gy \,\&\, Hxy \rightarrow Ky))$
$\neg(\mathrm{E}y)(Ly \,\&\, Ky)$
$\underline{(\mathrm{E}x)[Fx \,\&\, (\mathrm{A}y)(Hxy \rightarrow Ly) \,\&\, (\mathrm{A}y)(Gy \,\&\, Hxy \rightarrow Jxy)]}$
$(\mathrm{E}x)(Fx \,\&\, \neg(\mathrm{E}y)(Gy \,\&\, Hxy))$

$\neg Fx + Gf(x) + \neg Gy + \neg Hxy + Ky$
$\neg Fx + Hx, f(x) + \neg Gy + \neg Hxy + Ky$
$\neg Fx + \neg Jx, f(x) + \neg Gy + \neg Hxy + Ky$
$\neg Lx + \neg Kx$
Fa
$\neg Hax + Lx$
$\neg Gx + \neg Hax + Jax$
$\neg Fx + Gg(x)$
$\neg Fx + Hx, g(x)$

46. (6pts)

$(\mathrm{A}x)(Fx \,\&\, (\mathrm{A}y)[Fy \,\&\, Hyx \rightarrow Gy] \rightarrow Gx)$
$\{(\mathrm{E}x)(Fx \,\&\, \neg Gx) \rightarrow (\mathrm{E}x)(Fx \,\&\, \neg Gx \,\&\, (\mathrm{A}y)(Fy \,\&\, \neg Gy \rightarrow Jxy)\}$
$\underline{(\mathrm{A}x)(\mathrm{A}y)(Fx \,\&\, Fy \,\&\, Hxy \rightarrow \neg Jyx)}$
$(\mathrm{A}x)(Fx \rightarrow Gx)$

$\neg Fx + Ff(x) + Gx$
$\neg Fx + Hf(x) + Gx$
$\neg Fx + \neg Gf(x) + Gx$
$\neg Fx + Gx + Fa$
$\neg Fx + Gx + \neg Ga$
$\neg Fx + Gx + \neg Fy + Gy + Jay$
$\neg Fx + \neg Fy + \neg Hxy + \neg Jyx$
Fb
$\neg Gb$

A problem which has gotten some considerable play in the literature is 'Schubert's Steamroller', after Len Schubert. (See Pelletier (1982), Walther (1985), McCune (1985), Stickel (1986)). The problem presented in English is this:

47. (10pts) Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants. Therefore there is an animal that likes to eat a grain-eating animal.

The previously mentioned authors symbolize the problem (especially the conclusion) differently (see Stickel (1986) for details). In its original form it is as follows: Let

P_0: a is an animal
P_1: a is a wolf
P_2: a is a fox
P_3: a is a bird
P_4: a is a caterpillar
P_5: a is a snail
Q_0: a is a plant
Q_1: a is a grain
S: a is much smaller than b
R: a likes to eat b

$(\mathrm{A}x)(P_1x \rightarrow P_0x)$ & $(\mathrm{E}x)P_1x$
$(\mathrm{A}x)(P_2x \rightarrow P_0x)$ & $(\mathrm{E}x)P_2x$
$(\mathrm{A}x)(P_3x \rightarrow P_0x)$ & $(\mathrm{E}x)P_3x$
$(\mathrm{A}x)(P_4x \rightarrow P_0x)$ & $(\mathrm{E}x)P_4x$
$(\mathrm{A}x)(P_5x \rightarrow P_0x)$ & $(\mathrm{E}x)P_5x$
$(\mathrm{E}x)Q_1x$ & $(\mathrm{A}x)(Q_1x \rightarrow Q_0x)$
$(\mathrm{A}x)(P_0x \rightarrow [(\mathrm{A}y)(Q_0y \rightarrow Rxy) + (\mathrm{A}y)((P_0y \,\&\, Syx \,\&\, (\mathrm{E}z)(Q_0z \,\&\, Ryz)) \rightarrow Rxy)])$
$(\mathrm{A}x)(\mathrm{A}y)((P_3y \,\&\, (P_5x + P_4x)) \rightarrow Sxy)$
$(\mathrm{A}x)(\mathrm{A}y)((P_3x \,\&\, P_2y) \rightarrow Sxy)$
$(\mathrm{A}x)(\mathrm{A}y)((P_2x \,\&\, P_1y) \rightarrow Sxy)$
$(\mathrm{A}x)(\mathrm{A}y)[(P_1x \,\&\, (P_2y + Q_1y)) \rightarrow \neg Rxy]$
$(\mathrm{A}x)(\mathrm{A}y)((P_3x \,\&\, P_4y) \rightarrow Rxy)$
$(\mathrm{A}x)(\mathrm{A}y)((P_3x \,\&\, P_5y) \rightarrow \neg Rxy)$
$(\mathrm{A}x)((P_4x + P_5x) \rightarrow (\mathrm{E}y)(Q_0y \,\&\, Rxy))$

$(\mathrm{E}x)(\mathrm{E}y)(P_0x \,\&\, P_0y \,\&\, (\mathrm{E}z)(Q_1z \,\&\, Ryz \,\&\, Rxy))$

In negated conclusion clause form, the problem becomes:

P_1a	$\neg P_0x + \neg Q_0 + \neg P_0z + \neg Szx + \neg Qw + \neg Rzw + Rxy + Rxz$
P_2b	$\neg P_4x + \neg P_3y + Sxy$
P_3c	$\neg P_5x + \neg P_3y + Sxy$
P_4d	$\neg P_3x + \neg P_2y + Sxy$
P_5e	$\neg P_2x + \neg P_1y + Sxy$
Q_1f	$\neg P_3x + \neg P_4y + Rxy$
$\neg P_1x + P_0x$	$\neg P_4 + Q_0i(x)$
$\neg P_2x + P_0x$	$\neg P_4x + Rx, i(x)$
$\neg P_3x + P_0x$	$\neg P_5 + Q_0j(x)$
$\neg P_4x + P_0x$	$\neg P_5x + Rx, j(x)$
$\neg P_5x + P_0x$	$\neg P_1x + \neg P_2y + \neg Rxy$
$\neg Q_1x + Q_0x$	$\neg P_1x + \neg Q_1y + \neg Rxy$
$\neg P_3 + \neg P_5y + \neg Rxy$	$\neg P_0x + \neg P_0y + \neg P_1z + \neg Ryz + Rxy$

Full Predicate Logic with Identity (without Functions)

48. (3pts) 'A problem to test identity ATPs' – Piotr Rudnicki (Computing, Univ. Alberta).

$a = b + c = d$	$a = b + c = d$
$\underline{a = c + b = d}$	$a = c + b = d$
$a = d + b = c$	$a \neq d$
	$b \neq c$

Here are some problems which straightforwardly test the identity components of ATPs without putting much strain on the rest of the system.

49. (5pts)

$(\mathrm{E}x)(\mathrm{E}y)(\mathrm{A}z)(z = x + z = y)$	$x = c + x = d$
$Pa \,\&\, Pb$	Pa
$\underline{a \neq b}$	Pb
$(\mathrm{A}x)Px$	$a \neq b$
	$\neg Pe$

50. (4pts)

$(\mathrm{A}x)[Fax + (\mathrm{A}y)Fxy] \rightarrow (\mathrm{E}x)(\mathrm{A}y)Fxy$	$Fax + Fxy$
	$\neg Fxf(x)$

51. (5pts)

$\underline{(\mathrm{E}z)(\mathrm{E}w)(\mathrm{A}x)(\mathrm{A}y)[Fxy \leftrightarrow (x = z \,\&\, y = w)]}$	$\neg Fxy + x = a$

$(\mathrm{E}z)(\mathrm{A}x)[(\mathrm{E}w)(\mathrm{A}y)(Fxy \leftrightarrow y = w) \leftrightarrow x = z]$

$\neg Fxy + y = b$

$x \neq a + y \neq b + Fxy$
$\neg Ff(x), y + y = g(x) + f(x) = x$
$\neg Ff(x), y + y = g(x) + Ff(x), h(x, z) + h(x, z) = z$
$\neg Ff(x), y + y = g(x) + h(x, z) \neq z + \neg Ff(x), h, (x, z)$
$y \neq g(x) + Ff(x), y + Ff(x), h(x, z) + h(x, z) = z$
$y \neq g(x) + Ff(x), y + f(x) = x$
$y \neq g(x) + Ff(x), y + h(x, z) \neq z + \neg Ff(x), h(h, z)$
$f(x) \neq x + Ff(x), h(x, z) + h(x, z) = z$
$f(x) \neq x + h(x, z) \neq z + \neg Ff(x), h(x, z)$

52. (5pts)

$\underline{(\mathrm{E}z)(\mathrm{E}w)(\mathrm{A}x)(\mathrm{A}y)[Fxy \leftrightarrow (x = z \,\&\, y = w)]}$
$(\mathrm{E}w)(\mathrm{A}y)[(\mathrm{E}z)(\mathrm{A}x)(Fxy \leftrightarrow x = z) \leftrightarrow y = w]$

$\neg Fxy + x = y$

$\neg Fxy + y = b$

$x \neq y + y \neq b + Fxy$
$\neg Fy, f(x) + y \neq g(x) + f(x) = x$
$\neg Fy, f(x) + y = g(x) + Fh(x, z)f(x) + \neg Fh(x, z), f(x)$
$y \neq g(x) + Fy, f(x) + f(x) = x$
$y \neq g(x) + Fy, f(x) + Fh(x, z), f(x) + h(x, z) = z$
$y \neq g(x) + Fy, f(x) + h(x, z) \neq z + \neg Fh(x, z), f(x)$
$f(x) \neq x + Fh(x, z), f(x) + h(x, z) = z$
$f(x) \neq x + h(x, z) \neq z + \neg Fh(x, z)f(x)$

53. (7pts) [Test your converter-to-clause-form: about 146 clauses.]

$\underline{(\mathrm{E}x)(\mathrm{E}y)[x \neq y \,\&\, (\mathrm{A}z)(z = x + z = y)]}$

$\{(\mathrm{E}z)(\mathrm{A}x)[(\mathrm{E}w)(\mathrm{A}y)(Fxy \leftrightarrow$
$y = w) \leftrightarrow x = z] \leftrightarrow$
$(\mathrm{E}w)(\mathrm{A}y)[(\mathrm{E}z)(\mathrm{A}x)(Fxy \leftrightarrow$
$x = z) \leftrightarrow y = w]\}$

54. (9pts) Montague's (1955) paradox of grounded classes.

$$\frac{(\mathrm{A}y)(\mathrm{E}z)(\mathrm{A}x)(Fxz \leftrightarrow x = y)}{\neg(\mathrm{E}w)(\mathrm{A}x)[Fxw \leftrightarrow (\mathrm{A}u)\{Fxu \rightarrow (\mathrm{E}y)(Fyu \,\&\, \neg(\mathrm{E}z)(Fxu \,\&\, Fzy))\}]}$$

$\neg Fx, f(y) + x = y$
$x \neq y + Fx, f(y)$
$\neg Fxa + \neg Fxy + Fg(x, y), y$
$\neg Fxa + \neg Fxy + \neg Fz, g(x, y)$
$Fx, h(x) + Fxa$
$\neg Fx, h(y) + Fi(y, x), x + Fya$

55. (8pts) The following problem was given by Len Schubert. For ATPs with an 'answer extraction' mechanism, the conclusion might replaced with the query 'who killed Aunt Agatha?'

English: Someone who lives in Dreadsbury Mansion killed Aunt Agatha. Agatha, the butler, and Charles live in Dreadsbury Mansion, and are the only people who live therein. A killer always hates his victim, and is never richer than his victim. Charles hates no one that Aunt Agatha hates. Agatha hates everyone except the butler. The butler hates everyone not richer than Aunt Agatha. The butler hates everyone Agatha hates. No one hates everyone. Agatha is not the butler. Therefore: Agatha killed herself.

$$\frac{\begin{array}{l}(\mathrm{E}x)(Lx \,\&\, Kxa)\\ La \,\&\, Lb \,\&\, Lc\\ (\mathrm{A}x)(Lx \rightarrow x = a + x = b + x = c)\\ (\mathrm{A}y)(\mathrm{A}x)(Kxy \rightarrow Hxy)\\ (\mathrm{A}x)(\mathrm{A}y)(Kxy \rightarrow \neg Rxy)\\ (\mathrm{A}x)(Hax \rightarrow \neg Hcx)\\ (\mathrm{A}x)(x \neq b \rightarrow Hax)\\ (\mathrm{A}x)(\neg Rxa \rightarrow Hbx)\\ (\mathrm{A}x)(Hax \rightarrow Hbz)\\ (\mathrm{A}x)(\mathrm{E}y) \neg Hxy\\ a \neq b\end{array}}{Kaa}$$

Ld
Kda
La
Lb
Lc
$\neg Lx + x = a + x = b + x = c$
$\neg Kxy + Hxy$
$\neg Kxy + \neg Rxy$
$\neg Hax + \neg Hcx$
$x = b + Hax$
$Rxa + Hbx$
$\neg Hax + Hbx$
$\neg Hx, f(x)$
$a \neq b$
$\neg Kaa$

The Full Predicate Logic with Identity and Arbitrary Functions

56. (4pts)

$(\mathrm{A}x)((\mathrm{E}y)[Fy \,\&\, x = f(y)] \rightarrow Fx) \leftrightarrow$

$(\mathrm{A}x)[Fx \rightarrow Ff(x))]$

$\neg Fx + y \neq f(y) + Fy +$
$\quad \neg Fz + Ff(z)$
$\neg Fx + y \neq f(y) + Fy + Fa$
$\neg Fx + y \neq f(y) +$
$\quad Fy + b = f(b)$
$\neg Fx + y \neq f(y) + f(y) + \neg Fb$
$Fc + \neg Fx + Ff(x)$
$Fc + Fa$
$Fc + b = f(b)$
$Fc + \neg Fb$
$\neg Ff(c) + \neg Fx + Ff(x)$
$\neg Ff(c) + Fa$
$\neg Ff(c) + b = f(b)$
$\neg Ff(c) + \neg Fb$

57. (2pts)

$Ff(a, b), f(b, c)$
$Ff(b, c), f(a, c)$
$\underline{(\mathrm{A}x)(\mathrm{A}y)(\mathrm{A}z)[Fxy \,\&\, Fyz \rightarrow Fxz]}$
$Ff(a, b), f(a, c)$

$Ff(a, b), f(b, c)$
$Ff(b, c), f(a, c)$
$\neg Fxy + \neg Fyz + Fxz$
$\neg Ff(a, b), f(a, c)$

58. (3pts)

$\underline{(\mathrm{A}x)(\mathrm{A}y)f(x) = g(y)}$
$(\mathrm{A}x)(\mathrm{A}y)f(f(x)) = f(g(y))$

$f(x) = g(y)$
$f(f(a)) \neq f(g(b))$

59. (3pts)

$\underline{(\mathrm{A}x)(Fx \leftrightarrow \neg Ff(x))}$
$(\mathrm{E}x)(Fx \,\&\, \neg Ff(x))$

$\neg Fx + \neg Ff(x)$
$Ff(x) + F(x)$
$\neg F(x) + Ff(x)$

60. (4pts)

$(\mathrm{A}x)[Fx, f(x) \leftrightarrow (\mathrm{E}y)[(\mathrm{A}z)(Fzy \rightarrow$
$\quad Fz, f(x)) \,\&\, Fxy]$

$Fa, f(a) + \neg Fby + Fy, f(a)$

$Fa, f(a) + Fa, b$
$Fg(x), x + \neg Fax + \neg Fyb +$
$\quad Fy, f(a)$
$Fg(x), x + \neg Fax + Fab$
$Fg(x), x + \neg Fax + \neg Fa, f(a)$
$\neg Fg(x), f(a) + \neg Fax +$
$\quad \neg Fby + Fy, f(a)$
$\neg Fg(x), f(a) + \neg Fax + Fab$
$\neg Fg(x), f(a) + \neg Fax +$
$\quad \neg Fa, f(a)$

Having warmed up with some straightforward examples, let's turn our attention to some more difficult ones.

61. (6pts)

$\underline{(\mathrm{A}x)(\mathrm{A}y)(\mathrm{A}z)f(x, f(y, z)) = f(f(x, y), z)}$	$f(x, f(y, z)) = f(f(x, y)\, z)$
$(\mathrm{A}x)(\mathrm{A}y)(\mathrm{A}z)(\mathrm{A}w)$ $f(x, f(y, f(z, w))) = f(f(f(x, y), z), w$	$f(\mathrm{a}, f(b, f(c, d))) \neq f(f(f(a, b), c), d)$

62. (5pts) Here is the original formulation in Bledsoe *et al.* (1972), p. 59 of the problem mentioned in (17), (33), and (38).

$[Fa \,\&\, (\mathrm{A}x)(Fx \rightarrow Ff(x))] \leftrightarrow$	$\neg Fa + Fb + Ff(f(x)) + Fy + Ff(f(y))$
$(\mathrm{A}x)[[\neg Fa + Fx + Ff(f(x))] \,\&$	$\neg Fa + Fb + Ff(f(x)) + \neg Ff(y) + Ff(f(y))$
$[\neg Fa + \neg Ff(x) + Ff(f(x))]]$	$\neg Fa + Fb + Ff(f(x)) + \neg Fz + Ff(y)$
	$\neg Fa + Fb + Ff(f(x)) + \neg Ff(f(d))$
	$\neg Fa + \neg Ff(b) + Ff(f(x)) + Fy + Ff(f(y))$
	$\neg Fa + \neg Ff(b) + Ff(f(x)) + \neg Ff(y) + Ff(f(y))$
	$\neg Fa + \neg Ff(b) + Ff(f(x)) + \neg Fz + Ff(z)$
	$\neg Fa + \neg Ff(b) + Ff(f(x)) + \neg Ff(f(d))$
	Fa
	$\neg Fc + Ff(c) + \neg Fa + Fy + Ff(f(y))$
	$\neg Fc + Ff(c) + \neg Fa + \neg Ff(y) + Ff(f(y))$
	$\neg Fc + Ff(c) + \neg Fz + Ff(z)$
	$\neg Fc + Ff(c) + \neg Ff(f(d))$
	$\neg Ff(f(c)) + \neg Fa + Fy + Ff(f(y))$
	$\neg Ff(f(c)) + \neg Fa + \neg Ff(y) + Ff(f(y))$
	$\neg Ff(f(c)) + \neg Fz + Ff(z)$
	$\neg Ff(f(c)) + \neg Ff(f(d))$

Here are three group theory problems, published I think, by Larry Wos (but I cannot locate where). Consider

(a) $(\mathrm{A}x)(\mathrm{A}y)(\mathrm{A}z)f(f(x, y), z) = f(x, f(y, z))$ $\quad f(f(x, y), z) = f(x, f(y, z))$

(b) $(\mathrm{A}x)f(a, x) = x$ $\quad f(a, x) = x$

(c) $(\mathrm{A}x)(\mathrm{E}y)f(y, x) = a$ $\quad f(g(x), x) = a$

63. (6pts) Show that (a), (b), (c) entail

$(\mathrm{A}x)(\mathrm{A}y)(\mathrm{A}z)[f(x, y) = f(z, y) \rightarrow x = z]$ $\quad f(b, c) = f(d, c)$

$b \neq d$

64. (6pts) Show that (a), (b), (c) entail

$(\mathrm{A}x)(\mathrm{A}y)[f(y, x) = a \rightarrow f(x, y) = a]$ $\quad f(c, b) = a$

$f(b, c) \neq a$

65. (8pts) Show that (a) and (b) entail

$[(\mathrm{A}x)f(x, x) = a \rightarrow (\mathrm{A}x)(\mathrm{A}y)f(x, y) = f(y, x)]$ $\quad f(x, x) = a$

$f(b, c) \neq f(c, b)$

Charles Morgan (see AAR *Newsletter* #3) has suggested a method of constructing difficult function-problems out of easy propositional logic ones. To do this, take a propositional logic theorem, eg $(\neg\neg P \rightarrow P)$, treat the propositional letters as objects which can be quantified over, encode the sentence operators ($\neg$ and $\rightarrow$, here) as functions, and treat 'is a theorem' as a (monadic) predicate 'T'. Thus, this theorem would become $(\mathrm{A}x)Ti(n(n(x)), x)$. Do this for every axiom of the logic under consideration, and treat the results as premises for every argument. The rules of inference of the logic are treated in this manner by saying that if the premise of the rule has 'T' applied to it then the conclusion of the argument has 'T' applied to it. So for the rule modus ponens: if P and $(P \rightarrow Q)$ are derivable, then so is Q, we would convert this to $(\mathrm{A}x)(\mathrm{A}y)(Ti(x, y) \ \& \ Tx \rightarrow Ty)$, and treat it as a premise of every argument. That this can generate quite difficult problems out of very simple ones is demonstrated by the following examples of Morgan's. The initial propositional logic has three axioms plus *modus ponens*:

(a) $(\mathrm{A}x)(\mathrm{A}y)T(i(x, i(y, x)))$

(b) $(\mathrm{A}x)(\mathrm{A}y)(\mathrm{A}z)T(i(i(x, i(y, z)), i(i(x, y), i(x, z))))$

(c) $(\mathrm{A}x)(\mathrm{A}y)T(i(i(n(x), n(y)), i(y, x)))$

(d) $(\mathrm{A}x)(\mathrm{A}y)(T(i(x, y)) \ \& \ T(x) \rightarrow T(y))$

66. (7pts) From (a)–(d) prove

$(\mathrm{A}x)T(i(x, n(n(x))))$

67. (7pts) From (a)–(d) prove

$(\mathrm{A}x)T(i(n(n(x)), x)$

68. (8pts) Replace (c) with $(Ax)(Ay)T(i(i(y, x), i(n(x), n(y))))$ and prove $(Ax)T(i(x, n(n(x))))$

Morgan's method is completely generalizable. The examples above used as an 'object language' only $\rightarrow$ and $\neg$, but we could have added on other connectives together with appropriate definitions or axioms. For example, we might add on $\leftrightarrow$ together with one or more of the definitions of $\leftrightarrow$ in terms of $\rightarrow$ and $\neg$ and perhaps with new axioms for $\leftrightarrow$ and perhaps with new rules of inference for $\leftrightarrow$. For instance, we might add on

$(Ax)(Ay)(T(i(x, y)) \& T(i(y, x)) \leftrightarrow T(e(x, y)))$
and/or $(Ax)(Ay)T(e(e(x, y), n(i(i(x, y), n(i(y, x))))))$
and/or $(Ax)(Ay)e(x, y) = n(i(i(x, y), n(i(y, x))))$
and/or $(Ax)(Ay)(T(e(x, y)) \& T(x) \rightarrow T(y))$
and/or $(Ax)(Ay)e(x, y) = e(n(x), n(y))$
and/or $(Ax)(Ay)e(x, y) = e(y, x)$
and/or $(Ax)(Ay)n(e(x, y)) = e(x, n(y))$

and the like. With sufficient of these, you can try to prove (the translations of) arguments using $\rightarrow, \neg, \leftrightarrow$. They are *very* difficult. But the method can be generalized even further and extended to other logics which include propositional logic as a part. Let us suppose we have a characterization of the propositional logic using Morgan's method. That is, suppose we have translated a complete set of propositional axioms and rules of inference, such as the ones mentioned before Problem 66, into the function-notation just mentioned. Now suppose we wish to consider modal propositional logics. These logics introduce one new propositional operator, L (logical necessity). The modal system T can be described as

(a) propositional logic axioms
(b) Modus Ponens
(c) $L(p \rightarrow q) \rightarrow (Lp \rightarrow Lq)$
(d) $Lp \rightarrow p$
(e) if p is a theorem, then Lp is a theorem.

Morgan's method as described shows how to get the translation of (a) and (b). Thus the function-notation version of system T would be arrived at by introducing a new function symbols for L, say 'b_1', and an axiomatization of system T would be

(a) (the translations of the propositional logic axioms)
(b) (the translation of Modus Ponens)
(c) $(Ax)(Ay)T(i(b_1(i(x, y)), i(b_1(x), b_1(y))))$
(d) $(Ax)T(i(b_1(x), x))$
(e) $(Ax)(T(x) \rightarrow T(b_1(x)))$

69. (9pts) Using the (a)–(e) just given as premises, translate simple theorems of the modal system T and prove them. For example:

$(Ax)T(i(b_1(x), n(b_1(n(x)))))$

70. (open problem) The example in (69) used the modal system T, and 'b_1' was the function corresponding to the L of system T. Now let's consider the modal system K, and let's represent its L by 'b_2'. The axioms and rules of inference for K are translated as

(a) (translations of the propositional logic axioms)
(b) (translation of Modus 'Ponens)
(f) $(Ax)(Ay)T(i(b_2(i(x, y)), i(b_2(x), b_2(y))))$
(g) $(Ax)(T(x) \rightarrow T(b_2(x)))$

In Pelletier (1985) I posed the question of whether there might not be a way to 'define' the b_1 function in terms of the other functions (including b_2), and a way to 'define' the b_2 function in terms of the other functions (including b_1), so that if X was a theorem which mentioned 'b_1' but not 'b_2', then the result of replacing 'b_1' (and its argument) by its 'definition' would result in a theorem. [And conversely for a theorem Y which mentioned 'b_2' but not 'b_1'.] These 'definitions' were also to have the following feature: let f_1 be the 'definition' of 'b_1' in terms of 'b_2' and f_2 be the 'definition' of 'b_2' in terms of 'b_1'. Then

(h) $(Ax)T(e(x, f_1(f_2(x))))$
(i) $(Ax)T(e(x, f_2(f_1(x))))$

were also to be true. That is, the result of 'translating' any sentence purely of one of the sublanguages into the other sublanguage can be 'translated' back into the first sublanguage and the result will be provably equivalent to the original sentence. Therefore, given (a)–(i) the problem is to first determine *whether* there are such f's and second find out what they are.

Some Problems for Studying the Computational Complexity of ATP's

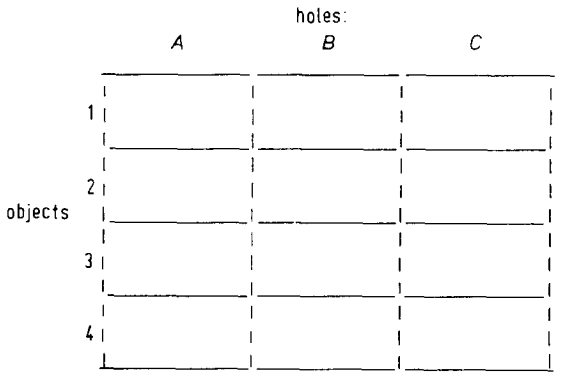
The difficulty in constructing problems for studying the complexity of the proof system of an ATP is to describe a set of problems whose complexity can independently be characterized in terms of some metric which can be varied and which does not introduce any 'side effects' into the resulting proofs. Various attempts to state such a set of problems have usually focussed on (a) number of clauses, (b) number of symbols, (c) number of distinct symbols. It is extremely difficult to guarantee in advance that the increase in proof size which is observed when (say) the number of clauses is increased is due solely to the increase in number of clauses, as opposed to being also influenced by some hidden increase in 'number of tricks required' (say). The following problem-types are designed to give a measure of complexity which does not involve any other types of difficulty as the problems get more complex.

71. (U-problems, after Alasdair Urquhart).
Consider the following problems (the sub-problem of conversion to clause form is left to the reader).

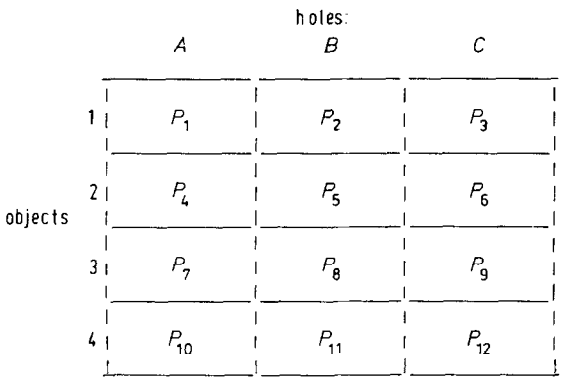
U_1: $(P_1 \leftrightarrow P_1)$
U_2: $(P_1 \leftrightarrow (P_2 \leftrightarrow (P_1 \leftrightarrow P_2)))$
U_3: $(P_1 \leftrightarrow (P_2 \leftrightarrow (P_3 \leftrightarrow (P_1 \leftrightarrow (P_2 \leftrightarrow P_3)))))$
.
.
.
U_n: $(P_1 \leftrightarrow (P_2 \leftrightarrow (P_3 \leftrightarrow \ldots . \leftrightarrow (P_n \leftrightarrow (P_1 \leftrightarrow (P_2 \ldots . \leftrightarrow P_n) \ldots .)$

The number of distinct sentence letters in U_n is n. the number of occurrences of sentence letters is $2n$. The number of embedded $\leftrightarrow$'s is $(2n\text{-}1)$. The number of clauses goes up dramatically as U_n increases, but I don't think it shows that the *problems* are dramatically more difficult as we go from U_2 to U_3, say. Rather, it's that the awkward clause form representation comes to the fore most dramatically with embedded biconditionals. On all other measure of increase of complexity from U_1 to U_n, one should say that the problems increase linearly in difficulty. So, given that the U-series of problems increases linearly in difficulty, compare the increase of your ATP along the dimensions of CPU time, size of proof, number of program statements executed, etc. [Alasdair Urquhart informs me that the proof size of *any* resolution system increases exponentially with increase in n].

72. (Pigeonhole problems – cf. Cook and Reckhow 1979).
Suppose there are n holes and $(n + 1)$ objects to put in the holes. Every object is in a hole and no hole contains more than one object. Pictorially, we can represent this (for the 4 object, 3 hole problem) as:



Each cell (i, j) says that the ith object is in the jth hole. For each cell use a different sentence letter $(P_1, P_2, \ldots)$. for the 4-object, 3 hole problem we might assign the letters in this fashion:



P_8 for example says that object 3 is in hole B. Let us now state the problem: 'Each object is in a hole' becomes (for this example)

(a) $P_1 + P_2 + P_3$
(b) $P_4 + P_5 + P_6$
(c) $P_7 + P_8 + P_9$
(d) $P_{10} + P_{11} + P_{12}$

'No hole has more than one object in it' becomes

(e) $\neg P_1 + \neg P_4$
(f) $\neg P_1 + \neg P_7$
(g) $\neg P_1 + \neg P_{10}$
(h) $\neg P_4 + \neg P_7$
(i) $\neg P_4 + \neg P_{10}$
(j) $\neg P_7 + \neg P_{10}$

} for hole A

(k) $\neg P_2 + \neg P_5$
(l) $\neg P_2 + \neg P_8$
(m) $\neg P_2 + \neg P_{11}$
(n) $\neg P_5 + \neg P_8$
(o) $\neg P_5 + \neg P_{11}$
(p) $\neg P_8 + \neg P_{11}$

} for hole B

(q) $\neg P_3 + \neg P_6$
(r) $\neg P_3 + \neg P_9$
(s) $\neg P_3 + \neg P_{12}$
(t) $\neg P_6 + \neg P_9$
(u) $\neg P_6 + \neg P_{12}$
(v) $\neg P_9 + \neg P_{12}$

} for hole C

The set of clauses (a)–(v) are inconsistent, as will any set be which is generated this way when the number of holes is less than the number of objects. Picking the number of objects to be *one* greater than the number of holes will yield the

'hardest' problem (for that number of holes), so we will just consider the 'n-hole problems', assuming that the number of objects is $(n + 1)$. It will be noticed that, for an n-holes problem, the number of distinct sentence letters is $(n^2 + n)$ and the number of clauses is

$$\frac{n^3 + n^2}{2} + (n + 1)$$

Thus the number of sentence letters increases quadradically and the number of clauses increases cubically. In any case, it seems that the 'difficulty' of the n-hole problems increases polynomially with n. See if your ATP can emulate that.

73. (Predicate logic pigeon hole problems). Problem (72) represented the n-hole problems by distinct sentence letters. This problem can also be represented by selecting a predicate 'O' meaning that x is an object, and a predicate 'H' meaning that x is a hole. Let Ixy be a 2-place relation saying that (object) x is in hole y. The 3-hole problem then becomes

(a) $(\mathrm{E}x)(\mathrm{E}y)(\mathrm{E}z)(\mathrm{E}w)[Ox \,\&\, Oy \,\&\, Oz \,\&\, Ow \,\&\, x \neq y \,\&\, x \neq z \,\&\, x \neq w \,\&\, y \neq z \,\&\, y \neq w \,\&\, z \neq w]$

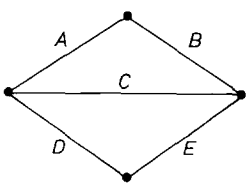
(b) $(\mathrm{E}x)(\mathrm{E}y)(\mathrm{E}z)[Hx \,\&\, Hy \,\&\, Hz \,\&\, x \neq y \,\&\, x \neq z \,\&\, y \neq z \,\&\, (Aw)(Hw \rightarrow w = x + w = y + w = z)]$

(c) $(\mathrm{A}x)(Ox \rightarrow (\mathrm{E}y)(Hy \,\&\, Ixy))$

(d) $(\mathrm{A}x(Hy \rightarrow (\mathrm{A}y)(\mathrm{A}z)(Oy \,\&\, Oz \,\&\, Iyx \,\&\, Izx \rightarrow y = z))$

These four formulas (a)–(d) are inconsistent, as are any generated in this manner. Test how your ATP increases in effort spent as the number of holes increases.

74. (Arbitrary graph problems. Due to Tseitin (1968), see Galil (1977) for an expository version. See also Urquhart (unpublished a, b). My thanks to Urquhart for explaining them to me.) Consider a graph with the edges labelled. For example



Assign 0 or 1 arbitrarily to nodes of the graph. For each node of the graph, we associate a set of clauses as follows:

(1) every label of an edge emanating from that node will occur in each clause (of the set of clauses generated from that node)

(2) if the node is assigned 0, then the number of negated literals in each of the generated clauses is to be odd. Generate all such clauses for that node.

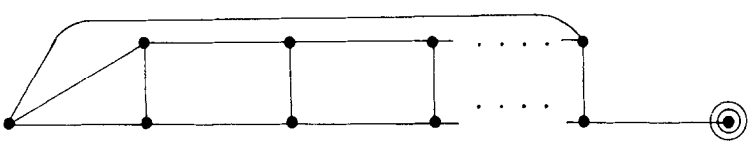
(3) if the node is assigned 1, then the number of negated literals in each of the generated clauses is to be even. Generate all such clauses for that node.

Tseitin's result is this: the sum (mod 2) of the 0's and 1's assigned to the nodes of the graph equals 1 if and only if the set of all generated clauses is inconsistent. For example, if we assign the node at the top of the above graph a 1 and all others 0, then the set of all generated clauses will be inconsistent. The clauses generated are:

(a) $A + B$	from top node, which was assigned 1, so the number of negated literals is even in each clause
(b) $\neg A + \neg B$	
(c) $A + C + \neg D$	from left node, which was assigned 0, so the number of negated literals is odd in each clause (generate all possible such clauses)
(d) $A + \neg C + D$	
(e) $\neg A + C + D$	
(f) $\neg A + \neg C + \neg D$	
(g) $B + C + \neg E$	(which was assigned 0)
(h) $B + \neg C + E$	
(i) $\neg B + C + E$	
(j) $\neg B + \neg C + \neg E$	
(k) $D + \neg E$	generated from bottom node
(l) $\neg D + E$	

Clauses (a)–(l) are inconsistent: prove that with your ATP. Now, we can increase the complexity of the graph in a number of different ways. Pick a 'natural' way and see how your ATP's proof increases CPU or a number of resolvants generated, etc.

75. According to Alasdair Urquhart, the *U*-problems of problem (71) can be graphically represented by the method of problem (74). The relevant graph is



where the double-circled node is assigned 1 and all others are assigned 0. Have your ATP prove this. (Number of vertical lines is number of distinct sentence letters).

Acknowledgements

I am grateful to a number of people. The system described in Pelletier (1982), that was used to verify which problems in this paper were harder than which others, was jointly developed by myself and Dan Wilson (now of Mirias Research of Edmonton). Further assistance on this system (1983–85) was aided by Gilles Chartrand and Randy Kopach. Len Schubert has a constant source of helpful critique, as well as the author of problems (47) and (55). David Sharp (Philosophy, University of Alberta) also

suggested which problems from Thomason (1972) students found difficult. Some of the these problems occur intermingled in the ones given under 'identity'. Charles Morgan explained to me his method (which was described (66)), and pointed out that it could be used to solve the problems of Pelletier (1985) – see problem (70) above. Finally, and especially, I offer my thanks to Alasdair Urquhart for numerous discussions about ATP – both its theory and specific problems. It was he who interested me in ways to test the proof complexity of various ATP systems. The problems (71)–(75) are directly due to him. I am also grateful to him for showing me Urquart (unpublished a, b), and telling me about Tseitin (1968). Thanks also to Sven Hurum (Computing, University of Alberta) for allowing me use of his 'convert to clause form' program. He warns that it is not perfect in its elimination of subsumed clauses. (But it sure beats doing it by hand!)

I gratefully acknowledge the assistance of the Canadian National Science and Engineering Research Council grant A5525 in the research involved in my ATP studies.

References

Bledsoe, W., Boyer, R., and Henneman, W. (1972) 'Computer proofs of limit theorems', *Artificial Intelligence* **3**, 27–60.

Cook, S. and Reckhow, R. (1979) 'The relative efficiency of propositional proof systems', *J. Symbolic Logic* **44**, 36–50.

De Champeaux, D. (1979) 'Sub-problem finder and instance checker: two cooperating preprocessors for theorem provers' *IJCAI* **6**, 191–196.

Galil, Z. (1977) 'On resolution with clauses of bounded size', *SIAM J. Computing* **6**, 444–459.

Kalish, D. & Montague, R. (1964) *Logic: Techniques of Formal Reasoning*, World, Harcourt & Brace,

Lusk, E. and Overbeek, R. (1985) 'Reasoning about equality', *J. Automated Reasoning* **1**, 209–228.

McCharen, R., Overbeek, R., & Wos, L. (1976) 'Problems and experiments for and with automated theorem-proving programs', *IEEE Trans. Computers* **C-25**(8), 773–782.

McCune, W. (1985) 'Schubert's Steamroller Problem with linked UR-resolution' *Assoc. Automated Reasoning Newsletter* **4**, 4–6.

Montague, R. (1955) 'On the paradox of grounded classes' *J. Symbolic Logic* **20**, 140.

Newell, A., Shaw, J., and Simon, H. (1957) 'Empirical explorations with the logic theory machine: a case study in heuristics'. Reprinted in E. Feigenbaum and J. Feldman (eds.) *Computers and Thought* (McGraw-Hill, N.Y.) pp. 279–293 (1963).

Newell, A. and Simon, H. (1972) *Human Problem Solving*, (Prentice-Hall, Englewood Cliffs).

Pelletier, F. J. (1982) 'Completely nonclausal, completely heuristically driven automatic theorem proving', Technical Report TR82-7, (Dept. of Computing Science, Edmonton, Alberta).

Pelletier, F. J. (1985) 'Six problems in translational equivalence', *Logique et Analyse* **108**, 423–434.

Siklóssy, L., Rich, A., and Marinov, V. (1973) 'Breadth first search: some surprising results', *Artificial Intelligence* **4**, 1–27.

Stickel, M. (1986) 'Schubert's steamroller problem: formulations and solutions', *J. Automated Reasoning* **2**, 89–104.

Thomason, R. (1972) *Symbolic Logic* (Prentice-Hall, Englewood Cliffs).

Tseitin, G. S. (1968) 'On the complexity of derivation in propositional calculus', reprinted in J. Siekmann and G. Wrightson (eds.) *Automation of Reasoning* (Springer-Verlag, Berlin).

Urquhart, A. (unpublished a) 'The complexity of Genzen systems for propositional logic'.

Urquhart, A. (unpublished b) 'Hard examples for resolution'.

Walther, C. (1985) 'A mechanical solution of Schubert's steamroller by many-sorted resolution', *Artificial Intelligence* **26**, 217–224.

Whitehead, A. and Russell, B. (1910) *Principia Mathematica*, Vol. I. (Cambridge UP, Cambridge).