Solving a Problem in Relevance Logic with an Automated Theorem Prover

Hans-Jürgen Ohlbach
Institut für Informatik I
University of Kaiserslautern
Postfach 3049
D-675 Kaiserslautern

Graham Wrightson
Victoria University
Dep. of Computer Science
Private Bag
Wellington, New Zealand

Abstract

A new challenging problem for automated theorem provers (ATP) is presented. It is from the field of relevance logic and is known as "converse of contraction". We firstly give some background information about relevance logic and the problem itself and then discuss a proof for the theorem which has been found by the Markgraf Karl Refutation Procedure (MKRP), a resolution based theorem prover, under development at the Universities of Karlsruhe and Kaiserslautern.

For solving this problem we implemented a new method to control the application of "generator clauses" like Exy \rightarrow Bf(x)f(y). An uncontrolled application of such clauses produces arbitrarily deeply nested and often useless terms f(f(f(\ldots which in general cannot be avoided with a global term depth limit because deeply nested terms of more heterogenous structure may occur in the proof.

I. 0-Order Relevance Logic

I.1 Introduction

Relevance logic was apparently first treated by Ackermann and Church and has been intensively refined and developed mainly by Anderson and Belnap [AB75]. The main motivation was to avoid certain paradoxes of implication which are present in classical formal logic. For example, ex falso quod libet, which has been shown to lead to all sorts of unpleasant results, is a valid formula in classical logic. The cause of the matter seems to lie in the definition of implication, which leads to other counterintuitive properties as well. For example the two true sentences "Grass is green" and "Two plus two equals four" lead to the true sentence "Grass is green implies two plus two equals four". Yet another feature of classical logic which does seem to be adequate for many applications lies in negation: by not denying a statement it cannot be concluded that the statement is affirmed.

I.2 Syntax of 0-order Relevance Logic RL

We closely follow [RM72] in presenting the syntax and semantics of RL. RL is built up syntactically in the usual way from a denumerably infinite set S of sentential parameters, the unary connective ¬ (called relevant negation), the usual truthfunctional connectives & and v, the binary connective \rightarrow (called relevant implication).

The axiom schemata of RL are:

A1. A \rightarrow A
A2. A \rightarrow ((A \rightarrow B) \rightarrow B)
Section 3.

(A + B) + ((B + C) + (A + C))

(A + (A + B)) + (A + B)

A & B + A

A & B + B

(A + B) & (A + C) + (A + B & C)

A + A v B

B + A v B

(A + C) & (B + C) + (A v B + C)

A & (B v C) + A & B v A & C

(A + -B) + (B + -A)

--A + A

and the deduction rule schemata are:

R1. From A + B and A, infer B
R2. From A and B infer A & B

I.3 Semantics of RL

A relevant model structure (rms) is a quadruple (0,K,R,*), where K is a set of objects called set-ups, 0 ∈ K, R is a ternary relation on K, and * is a unary operation on K, satisfying the postulates p1–p6 below.

A binary relation < and a quaternary relation R^2 on K are defined as abbreviations:

For all a,b,c,d in K

d1. a < b := R0ab

d2. R^2abcd := ∃ x (Rabx and Rxcd and x in K)

The postulates for all a,b,c,d in K are

p1. R0aa
p2. Rgaa
p3. R^2abcd => R^2acbd
p4. R^20abc => Rabc
p5. Rabc => Rca*b*

The definition d2 is generalized in the following way:

d3. R^nabc...def := ∃ xR^nabc...x and R^{n-r}x...def

and x in K and 1 < r < n-1

From the postulates the following lemmas can easily be derived:

L1. R^nabc...def <=> R^{n+1}0abc...def
L2. All R^n-relations, which can be formed by a permutation of the first n+1 set-ups in the relation R^nabc...def, can be shown to follow from R^nabc...def
L3. Each relation R^n implies a relation R^{n+1} by replacement of any set-up a in R^n by aa.