Support Ordered Resolution

Bruce Spencer and Joseph D. Horton

Faculty of Computer Science, University of New Brunswick
P.O. Box 4400, Fredericton, New Brunswick, Canada E3B 5A3
bspencer@unb.ca, jdh@unb.ca
http://www.cs.unb.ca

Abstract. In a binary tree representation of a binary resolution proof, rotating some tree edge reorders two adjacent resolution steps. When rotation is not permitted to disturb factoring, and thus does not change the size of the tree, it is invertible and defines an equivalence relation on proof trees. When one resolution step is performed later than another after every sequence of such rotations, we say that resolution supports the other.

For a given ordering on atoms, or on atom occurrences, a support ordered proof orders its resolution steps so that the atoms are resolved consistently with the given order without violating the support relation between nodes. Any proof, including the smallest proof tree, can be converted to a support ordered proof by rotations. For a total order, the support ordered proof is unique. The support ordered proof is also a rank/activity proof where atom occurrences are ranked in the given order.

Procedures intermediate between literal ordered resolution and support ordered resolution are considered. One of these, 1-weak support ordered resolution, allows to resolve on a non-maximal literal only if it is immediately followed by both a factoring and a resolution on some greater literal.

In a constrained experiment where literal ordered resolution solves only six of 408 TPTP problems with difficulty between 0.11 and 0.56, 1-weak support ordered resolution solves 75.

1 Introduction

Automated theorem provers, in their search for a proof, must balance the deductive power of a calculus, telling what can be derived from a given point in the search, with restriction strategies, telling which deductions are to be avoided. Clearly the restriction strategy must not remove all of the choices that eventually lead to a proof, at least not without the user’s being aware of its incompleteness. But even so, the restriction strategy may remove all shortest proofs, leading to another undesirable effect: the theorem prover takes longer to find a longer proof. An ideal restriction strategy would reduce the space to one richly populated with only short proofs, be simple to implement and quick to check. This is an unrealistic ideal. In this paper we give a reduction strategy that is quick to check, simple to implement, admits smallest proofs trees, and is almost as
restrictive as literal ordered resolution, a widely used and successful restriction. Literal ordered resolution was used in the CADE-16 System Competition [9] by at least Bliksem, Gandalf, HOTTER, SPASS and Vampire.

In our setting we represent proofs as binary trees, labeled by clauses according to Robinson’s resolution method [6]. A node is labeled both by a clause, referring to the conclusion drawn at this point by the resolution, and if it is not a leaf, by the atom that was resolved upon to give this conclusion. Our measure of size is the number of nodes in this tree. Often theorem provers build sequences of formulae where each deductively follows from previous ones. This sequence represents a traversal of a directed acyclic graph (dag) that underlies the tree. The size of an underlying dag is a more natural measure of proof size than the size of the tree. But often if one dag is smaller than another then the tree expanded from the first dag is smaller than the tree expanded from the other.

Support ordered resolution depends on the notion of support between two resolution steps, first defined in [4] and defined for proof trees in [7]. When compared with the literal ordered resolution, used by many theorem provers and explored in [3], support ordered resolution is less restrictive, in that it admits a very specific additional resolution step. On the other hand the support ordered restriction does not increase the size of the smallest tree, unlike literal ordered resolution which may restrict all smallest trees, and in some cases admit only exponentially larger trees and dags, as shown in Example 5 below.

Rotating some tree edge reorders two adjacent resolution steps. When rotation is not permitted to disturb factoring, and thus does not change the size of the tree, it is invertible and defines an equivalence relation on proof trees. For a given total order on atoms, there is a single support ordered tree in each rotation equivalence class. Since the equivalence classes typically contain an exponential number of trees, support ordered resolution substantially reduces the search space.

It is interesting to find a restriction of a resolution calculus that admits a smallest deduction (tree) while substantially reducing the search space, as support ordered resolution does. It is also interesting in that it brings together two apparently different restrictions, the rank/activity restriction [5] and literal ordered resolution. A given ordering on atoms can be used to set the ranks of literals in each clause, and then the rank/activity proof is the support ordered proof.

Viewed as a generalization of other restrictions, we can identify a number of other special cases of support ordered resolution. These suggest themselves as candidates for experiments. One set of these experiments has been done for the special case called 1-weak support ordered resolution, or 1-wso. This proof format depends on a restriction that can be quickly checked on partially closed binary resolution trees. Recall that the literal ordered restriction allows a resolution only on a maximal atom in each clause; 1-wso allows maximal resolutions and also allows a resolution on some non-maximal atom but then requires an immediate merge on a greater atom from different parents followed by a resolution step on