An Application of the Theorem Prover SBR3 to Many-valued Logic

Siva Anantharaman
LIFO, Dépt. Math-Info.
Université d'Orléans
45067 ORLEANS Cedex 02 (Fr.)
e-mail: siva@univ-orleans.fr

Maria Paola Bonacina*
Department of Computer Science
SUNY at Stony Brook
Stony Brook NY 11794-4400 USA
e-mail: bonacina@sbcs.sunysb.edu

Abstract

In this paper we present the theorem prover *SBR3* for equational logic and its application in the many-valued logic of Lukasiewicz. We give a new equational axiomatization of many-valued logic and we prove by SBR3 that it is equivalent to the classical equational presentation of such logic given by Wajsberg. We feel that our equational axiomatization of Wajsberg algebras is more suited for automated reasoning than the classical one. Indeed, it has allowed us to obtain a fast mechanical proof of the so called "fifth Lukasiewicz conjecture", which is regarded as a challenge problem for theorem provers. We present many-valued logic in Section 1, the mechanical proofs by SBR3 in Section 2 and SBR3 itself in Section 3.

1 Many-valued logic

Many-valued propositional logic was first introduced by Jan Lukasiewicz in the 1920's. All the following results about early work on many-valued logic are reported in [TL-56].

The original definition of many-valued logic is purely semantical. No axioms and no inference rules are given. Lukasiewicz defines first a model and then the logic is defined as the set of all sentences in propositional calculus which are true in that model. More precisely, the n-valued logic L_n is defined as the set of all sentences satisfied by the structure

$$\mathcal{L}_n =< \{\frac{k}{n-1} | 0 \leq k \leq n-1\}, g, f >$$

where $A_n = \{\frac{k}{n-1}|0 \leq k \leq n-1\}$ is the domain, $g : A_n \to A_n$ is the unary function $g(x) = 1 - x$ and $f : A_n \times A_n \to A_n$ is the binary function $f(x, y) = min(1 - x + y, 1)$.

L_1 is the set of all legal propositional sentences, L_2 is classical two-valued propositional logic with model

$$\mathcal{L}_2 =< \{0, 1\}, g, f >$$

where the functions g and f are classical negation and implication. L_3 is three-valued logic, the first one introduced by Lukasiewicz.

*Research supported in part by NSF grants INT-8715231, CCR-8805734 and CCR-8901322. Has been also supported by Dottorato di Ricerca in Informatica, Universitá degli Studi di Milano.

As n increases, the domain A_n grows:

$$A_1 = \{1\}, \quad A_2 = \{0, 1\}, \quad A_3 = \{0, \tfrac{1}{2}, 1\},$$

$$A_4 = \{0, \tfrac{1}{3}, \tfrac{2}{3}, 1\}, \quad A_5 = \{0, \tfrac{1}{4}, \tfrac{2}{4}, \tfrac{3}{4}, 1\} \ldots$$

The limit of this sequence is the set Q_0 of all rational numbers in the interval $[0, 1]$, which is the domain of the many valued logic

$$\mathcal{L}_{\aleph_0} =< \{\frac{k}{l} | 0 \leq k \leq l\}, g, f >$$

As n increases, the set L_n shrinks and $L_{\aleph_0}$ is the smallest such set, i.e. the intersection of all the L_n. It has been proved by Lindenbaum that the domain $\mathcal{L}_{\aleph_0}$ can be any arbitrary set of numbers $\{x | 0 \leq x \leq 1\}$ closed with respect to g and f.

Lukasiewicz conjectured that the following axioms are an axiomatization for $L_{\aleph_0}$:

(1) $p \Rightarrow (q \Rightarrow p)$

(2) $(p \Rightarrow q) \Rightarrow ((q \Rightarrow r) \Rightarrow (p \Rightarrow r))$

(3) $((p \Rightarrow q) \Rightarrow q) \Rightarrow ((q \Rightarrow p) \Rightarrow p)$

(4) $((p \Rightarrow q) \Rightarrow (q \Rightarrow p)) \Rightarrow (q \Rightarrow p)$

(5) $(not(p) \Rightarrow not(q)) \Rightarrow (q \Rightarrow p)$

where *not* and $\Rightarrow$ are interpreted as g and f in the model $\mathcal{L}_{\aleph_0}$. We will write *not*, $\Rightarrow$ and *true* rather than g, f and 1 whenever we are working on the axiomatization rather than on the model. The fourth axiom was later proved to be dependent on the others.

The *Wajsberg algebras* are a class of algebras related to Lukasiewicz logic [FRT-84]. The following is the equational axiomatization of Wajsberg algebras as reported in [FRT-84], which we denote by $\mathcal{W}$:

(1) $true \Rightarrow x == x$

(2) $(x \Rightarrow y) \Rightarrow ((y \Rightarrow z) \Rightarrow (x \Rightarrow z)) == true$

(3) $((x \Rightarrow y) \Rightarrow y == (y \Rightarrow x) \Rightarrow x$

(4) $(not(x) \Rightarrow not(y)) \Rightarrow (y \Rightarrow x) == true$

Our interest in many-valued logic has been originally motivated by the so called *Fifth Lukasiewicz conjecture*, brought to our attention by [Mu]. The problem consists in proving from $\mathcal{W}$ that

$$(x \Rightarrow y) \vee (y \Rightarrow x) == true,$$

where the operator $\vee$ is defined by:

$$x \vee y == (x \Rightarrow y) \Rightarrow y.$$

This problem was originally given by Lukasiewicz as a conjecture [TL-56] and proved several years later [RR-58, MC-58].

We succeeded in obtaining a mechanical proof of this conjecture as reported in [AB-90]. During the effort of proving the Fifth Lukasiewicz conjecture we have obtained by our prover SBR3 a new equational axiomatization of many-valued logic, which we present in this paper.

2 A new set of axioms for Wajsberg algebras

We start from the observation that the connective $\vee$, which occurs in the Fifth Lukasiewicz conjecture, is not classical disjunction. The classical definition of disjunction in terms of implication and negation is

$$or(x,y) = not(x) \Rightarrow y.$$

Since in the model $\mathcal{L}_{\aleph_0}$ the operator $\Rightarrow$ is interpreted as $min(1-x+y,1)$, such connective *or* is interpreted as $min(1-(1-x)+y,1) = min(x+y,1)$. On the other hand, according to the above definition of the operator $\vee$, $x \vee y$ is interpreted as $max(x,y)$:

$$min(1-min(1-x+y,1)+y,1) = \begin{cases} min(1-1+y,1) = y\ if\ y \geq x \\ min(1-1+x-y+y,1) = x\ if\ x \geq y \end{cases}$$

Thus $\vee$ and *or* are different connectives, except when the domain of interpretation is $\{0,1\}$, i.e. when the logic is 2-valued.

Our first bunch of experiments consists in adding to the axioms $\mathcal{W}$, the two definitions

$$or(x,y) = not(x) \Rightarrow y$$

$$and(x,y) \Rightarrow z = (x \Rightarrow y) \Rightarrow z$$

where the second one is an implicit definition of classical conjunction[1].

Our first goal is to prove by *SBR3* that *or* and *and* are *associative and commutative* (AC). The proof has been obtained in three steps:

Lemma 1. Prove that $not(x) \Rightarrow not(y) == y \Rightarrow x$ in the theory $\mathcal{W}$.

The proof of Lemma 1 also generates the following important result: $not(not(x)) == x$.

Lemma 2. Prove that the connective *and* is commutative, that is $x \Rightarrow (y \Rightarrow z) == y \Rightarrow (x \Rightarrow z)$ follows from $\mathcal{W}$ and Lemma 1.

Lemma 3. Prove that the connective *or* is AC from $\mathcal{W}$ and Lemmas 1 and 2.

This proof also produces the equation: $and(x,y) = not(or(not(x), not(y)))$, which defines *and* in terms of *or* and shows that *and* is also AC.

Since $not(not(x)) == x$, we may write henceforth $x \Rightarrow y = or(not(x), y)$. Thus *and* and *or* can replace $\Rightarrow$ as basic connective of many-valued logic. This has been important to our purposes, since the possibility to express implication in terms of other operators has been fundamental to achieve a proof of the Fifth Lukasiewicz conjecture.

Our second goal is to obtain an axiomatization of many-valued logic in terms of *and* and *exclusive or*. Therefore, we define

$$x * y = and(x,y) == not(or(not(x), not(y))) \text{ and}$$

$$x + y == or(and(x, not(y)), and(not(x), y)).$$

[1] The reason for introducing conjunction in this way is in the details of our proof of the Fifth Lukasiewicz conjecture [AB-90].

If we add to $\mathcal{W}$ these two definitions, we can prove by SBR3 the following set of theorems, which we denote by $\mathcal{W}'$:

$$not(x) == x + 1$$
$$x + 0 == x$$
$$x + x == 0$$
$$x * 1 == x$$
$$x * 0 == 0$$
$$(1 + x) * x == 0$$
$$x + (1 + y) == (x + 1) + y$$
$$((1 + x) * y) + 1) * y == ((1 + y) * x) + 1) * x$$

where $*$ is AC, while $+$ is *commutative* only. Furthermore, the prover generates the definition of *or* in terms of $+$:

$$or(x, y) = 1 + ((1 + x) * (1 + y)).$$

Inversely, if we start with $\mathcal{W}'$ as axiomatization and we add the definition:

$$(x \Rightarrow y) = 1 + (x * (1 + y)) \ ,$$

we obtain by SBR3 all the equations of $\mathcal{W}$ as theorems. Therefore, we have proved the following

Theorem The sets $\mathcal{W}$ and $\mathcal{W}'$ are equivalent axiomatizations.

Our axiomatization $\mathcal{W}'$ for many-valued logic is partly resemblant of the system of axioms for the Boolean ring given by J. Hsiang. However, there are three substantial differences: our $+$ is only commutative, whereas it is AC in the Boolean case. Distributivity of $*$ on $+$ does *not* hold in many-valued logic, whereas it does in the Boolean ring. Finally, $*$ is *not* idempotent.

The experience gathered so far with SBR3 suggests that the axiomatization $\mathcal{W}'$ is more suited for automated equational reasoning than the original $\mathcal{W}$. For instance, we have observed that if we start with $\mathcal{W}'$, we obtain a proof of the Fifth Lukasiewicz conjecture which is faster than the one reported in [AB-90].

3 The equational theorem prover SBR3

We conclude with an overview of our prover. *SBR*3 is a term rewriting based theorem prover for equational logic. It is the latest offspring of the *Reve* family. The *Reve* family is mostly concerned with completing a set of equations into a canonical system. Although well-worthy in its own right, such systems are not ideal for the purpose of proving a specific equational theorem.

In 1986, Mzali and Hsiang started to develop a new system, *SbReve*1 [AM-88], based on *Reve*2.4. The goal of *SbReve*1 was to modify the completion process in order to efficiently prove a single theorem of an equational theory rather than generating a canonical system. *SbReve*1 was overhauled into *SbReve*2 [AHM-89] and finally into *SBR*3 by Anantharaman.

The entire *SbReve* family provers employ the overall methodology of *simplification-first*. Critical pairs are never generated as long as there is still room for simplification.

Even if the superposition procedure is invoked, critical pairs are generated one at a time, and the simplification process is re-started as soon as a divergent critical pair is generated. This is the most significant design difference between the *SbReve* and *Reve* families.

The major difference between $SBR3$ and its predecessors is the incorporation of much more sophisticated search strategies, which we will describe in the following subsections.

Finally, $SBR3$ is making further progress. First, the approach adopted in *SBR3* has recently been extended to Horn theories with equality [AA-91]. Second, *SBR3*, which is currently available in CLU, for SUN-3/VAX, is being rewritten in C/C++. A first version for Horn theories, in C/C++ and for all SUNs is expected by fall 1991.

3.1 The Inference Mechanisms of SBR3

SBR3 takes as inputs an equational theory E and an equation $s = t$ and tries to prove refutationally that $s = t$ is a theorem of E.

In addition to the theory and the equation, the user should also provide an ordering to order the terms. Usually the ordering should be a *complete simplification ordering* (a simplification ordering which is total on ground terms). In $SBR3$ the user has the choice of assigning a precedence among the operators in the theory and choose an ordering from a list implemented in the system. However, SBR3 will not check the totality for the user, so that this constraint is relaxed. This relaxation is very important for two reasons. First, no practical cso compatible with AC is known. If the totality requirement were enforced, it would be an obstacle to the application of the C/AC-UKB procedures [AM-88], which are the backbones of *SBR3*. Second, the lack of totality on ground terms may help in getting shorter proof, by the *Order-Saturation strategy* [AH-90, AA-90].

The inference mechanism of *SBR3* includes inference rules for the *cancellation* axioms [HRS-87], for *functional subsumption* and simplification, that is an array of powerful rules to replace large equations by smaller ones.

3.2 The Search Mechanisms of SBR3

The simplification-first search strategy coupled with cancellation controls the growth of the number and size of rules to some extent, but more clever means are quite often needed.

The first problem is to find a shorter path to a solution. UKB, being complete, guarantees the existence of a proof through simplification and superposition should there be one. It does not, however, guarantee to provide a *short* proof. Suppose the prover can look at several different inequalities and tries to find a contradiction simultaneously[2], then conceivably one can find a proof faster. On the other hand, one should also keep in mind not to inundate the search space with irrelevant inequalities. *SBR3* provides a facility for working on a reasonable number of inequalities. This is the above mentioned *Ordered-Saturation strategy*, which has proved indispensable, together with the Cancellation-Laws, to prove some of the more difficult problems we succeeded to prove [AH-90].

Another challenge is to eliminate redundant critical pairs, especially in AC-completion, due to the potentially astronomical number of AC-unifiers. The critical pair criteria defined for this purpose are not sufficiently effective, since all of them are designed not to destroy the confluence property of any given two terms. In refutational theorem proving, on the

[2] The basic UKB only looks at one.

other hand, we are only interested in the confluence of the two terms of the target theorem. Therefore a critical pair can be deleted or suspended as long as it does not destroy the confluence of the intended terms.

In *SBR3*, we attack this probelm by employing *measures*. A measure estimates the likelihood of whether a critical pair may contribute to an eventual proof of the intended theorem. Such measures are defined syntactically on the structure of terms, based for instance on the number of occurrences of a specific operator. Critical pairs are ordered according to the measure which decides the next equation to be chosen to perform superposition. Certain measures even allow us to delete critical pairs if they are deemed irrelevant for producing a proof. This search strategy is called *filtration-sorted strategy*. All the details can be found in [AA-90]. Several such strategies are implemented in *SBR3* and they have played a decisive role in our experiments so far, including the above application to many-valued logic.

References

[AA-90] S.Anantharaman, N. Andrianarivelo, Heuristic Criteria in Refutational Theorem Proving, *Research-Report, LRI-Orsay(Fr.), no. 541, 1990, Extended-Abstract in Proc. DISCO'90, LNCS. no. 429*

[AA-91] S.Anantharaman, N. Andrianarivelo, A Complete Semi-Decision Procedure for Horn Theories, *Research Report, LIFO - LRI(Fr.), (to appear in Feb.-March 1991)*

[AB-90] S.Anantharaman, M.P.Bonacina, Automated Proofs in Lukasiewicz Logic, *Technical Report, Department of Computer Science, SUNY at Stony Brook, November 1989*

[AH-90] S.Anantharaman, J.Hsiang, Automated Proofs of the Moufang Identities in Alternative Rings, *Vol. 1, J. Automated Reasoning, 1990*

[AHM-89] S.Anantharaman, J.Hsiang, J.Mzali, SbReve2: A term rewriting laboratory with (AC-)Unfailing Completion, *RTA, Springer-Verlag LNCS Vol. 355, pp533–537, 1989*

[AM-88] S.Anantharaman, J.Mzali, Unfailing Completion Modulo a set of Equations, *Research Report, no. 470, LRI-Orsay (Fr.), 1989*

[FRT-84] J.M.Font, A.J.Rodriguez, A.Torrens, Wajsberg algebras, *Stochastica, Vol. 8, No. 1, pp5-31, 1984*

[HRS-87] J. Hsiang, M. Rusinowitch, K. Sakai, Complete set of inference rules for the cancellation laws, *IJCAI 87, Milan, Italy, 1987*

[MC-58] C.A.Meredith, C.C.Chang, , *Trans. Amer. Math. Soc. 87, 13, pp54,55,56, 1958*

[Mu] D.Mundici, Personal communication.

[RR-58] A.Rose, C.C.Rosser, , *Trans. Amer. Math. Soc. 87, 13, pp1–53, 1958*

[TL-56] A.Tarski, J.Lukasiewicz, Investigations into the sentential calculus, *Chapter 4th in A.Tarski, Logic, Semantics, Meta-mathematics, pp38-56, Clarendon Press, Oxford, 1956*