Tableaux Modulo Theories using Superdeduction

An Application to the Verification of B Proof Rules with the Zenon Automated Theorem Prover

Mélanie Jacquel[1], Karim Berkani[1], David Delahaye[2], and Catherine Dubois[3]

[1] Siemens IC-MOL, Châtillon, France,
Melanie.Jacquel@siemens.com
Karim.Berkani@siemens.com
[2] CEDRIC/CNAM, Paris, France,
David.Delahaye@cnam.fr
[3] INRIA/CEDRIC/ENSIIE, Évry, France,
dubois@ensiie.fr

Abstract. We propose a method which allows us to develop tableaux modulo theories using the principles of superdeduction, among which the theory is used to enrich the deduction system with new deduction rules. This method is presented in the framework of the Zenon automated theorem prover, and is applied to the set theory of the B method. This allows us to provide another prover to Atelier B, which can be used to verify B proof rules in particular. We also propose some benchmarks, in which this prover is able to automatically verify a part of the rules coming from the database maintained by Siemens IC-MOL.

Keywords: Tableaux, Superdeduction, Zenon, Set Theory, B Method, Proof Rules, Verification.

1 Introduction

In this paper, we propose to integrate superdeduction [3] (a variant of deduction modulo) into the tableau method in order to reason modulo theories (see also [4] for a similar approach). This integration is motivated by an experiment which is managed by Siemens IC-MOL regarding the verification of B proof rules [5]. The B method [1], or B for short, allows engineers to develop correct by design software with high guarantees of confidence. A significant use of B by Siemens IC-MOL has concerned the control system of the driverless metro line 14 in Paris. B is a formal method based on set theory and theorem proving, and which relies on a refinement-based development process. The Atelier B environment is a platform that supports B and offers both automated and interactive provers. To ensure the global correctness of formalized applications, the user must discharge proof obligations. These proof obligations may be proved automatically, but otherwise, they have to be handled manually either by using the interactive prover, or by adding new proof rules that the automated prover can exploit. These new proof rules can be seen as axioms and must be verified by other means.

In [5], we develop an approach based on the use of the Zenon automated theorem prover [2], which relies on classical first order logic with equality and applies the tableau method. In this context, the choice of Zenon is strongly influenced by its ability of producing checkable proof traces under the form of Coq proofs in particular. The method used in this approach consists in first normalizing the formulas to be proved, in order to obtain first order logic formulas containing only the membership set operator, and then calling Zenon on these new formulas. This experiment gives satisfactory results in the sense that it can prove a significant part of the rules coming from the database maintained by Siemens IC-MOL. However, this approach is not complete and suffers from efficiency issues due to the preliminary normalization. To deal with these problems, the idea developed in this paper is to integrate the B set theory into the Zenon proof search method by means of superdeduction rules. This integration can be concretely achieved thanks to the extension mechanism offered by Zenon, which allows us to extend its core of deductive rules to match specific requirements.

The paper is organized as follows: in Section 2, we present the computation of superdeduction rules from axioms in the framework of the tableau method used by Zenon; we then introduce, in Section 3, the superdeduction rules corresponding to the B set theory; finally, in Section 4, we describe the corresponding implementation and provide some benchmarks concerning the verification of B proof rules coming from the database maintained by Siemens IC-MOL.

2 From Axioms to Superdeduction Rules

Reasoning modulo a theory in a tableau method using superdeduction requires to generate new deduction rules from some axioms of the theory. The axioms which can be considered for superdeduction are of the form $\forall \bar{x}\ (P \Leftrightarrow \varphi)$, where P is atomic. This specific form of axiom allows us to introduce an orientation of the axiom from P to φ, and we introduce the notion of proposition rewrite rule (this notion appears in [3], from which we borrow the following notation and definition). The notation $R : P \rightarrow \varphi$ is a proposition rewrite rule and denotes the axiom $\forall \bar{x}\ (P \Leftrightarrow \varphi)$, where R is the name of the rule, P an atomic proposition, φ a proposition, and $\bar{x}$ the free variables of P and φ.

As said in the introduction, one of our main objectives is to develop a proof search procedure for the set theory of the B method using the Zenon automated theorem prover [2]. In the following, we will thus consider the tableau method used by Zenon as the framework in which superdeduction rules will be generated from proposition rewrite rules.

The proof search rules of Zenon are described in detail in [2] and summarized in Figure 1 (for the sake of simplification, we have omitted the relational, unfolding, and extension rules), where ϵ is Hilbert's operator, capital letters are used for metavariables, and R_r and R_s are respectively reflexive and symmetric relations. As hinted by the use of Hilbert's operator, the δ-rules are handled by means of ϵ-terms rather than using Skolemization. What we call here metavariables are often named free variables in the tableau-related literature; they are

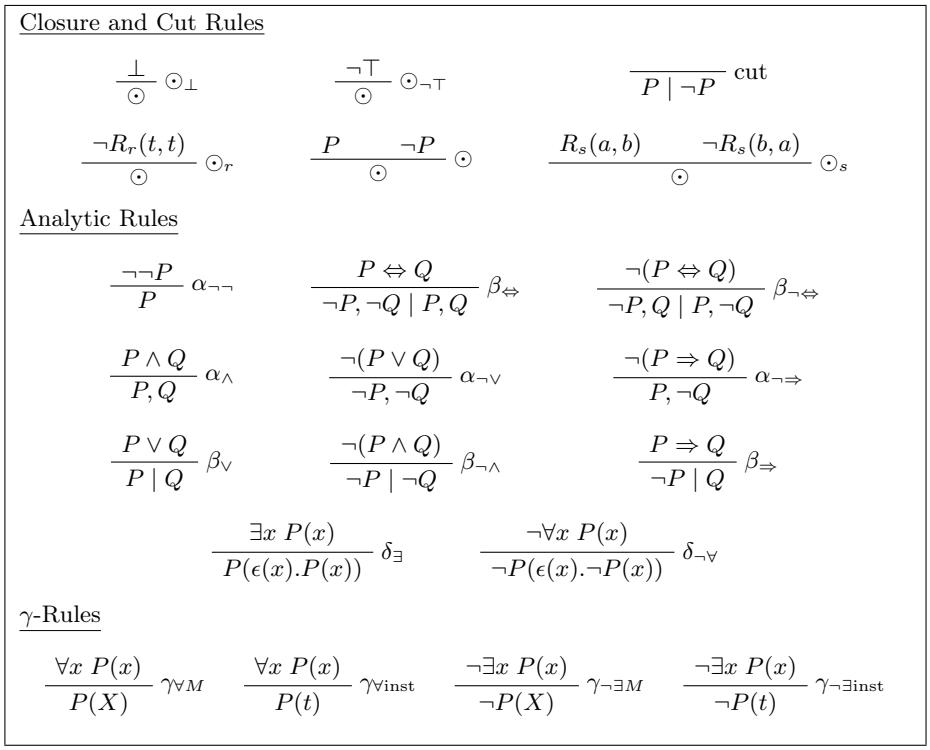

Fig. 1. Proof Search Rules of Zenon

not used as variables as they are never substituted. The proof search rules are applied with the normal tableau method: starting from the negation of the goal, apply the rules in a top-down fashion to build a tree. When all branches are closed, the tree is closed, and this closed tree is a proof of the goal.

Let us now describe how the computation of superdeduction rules for Zenon is performed from a given proposition rewrite rule.

Definition 1 (Computation of Superdeduction Rules). *Let $\mathcal{S}$ be a set of rules composed by the subset of the proof search rules of Zenon formed of the closure rules, the analytic rules, as well as the $\gamma_{\forall M}$ and $\gamma_{\neg\exists M}$ rules. Given a proposition rewrite rule $R : P \to \varphi$, two superdeduction rules (a positive one R and a negative one $\neg R$) are generated.*

To get the positive rule R (resp. the negative rule $\neg R$), initialize the procedure with the formula φ (resp. $\neg\varphi$). Next, apply the rules of $\mathcal{S}$ until there is no open leaf anymore on which they can be applied. Then, collect the premises and the conclusion, and replace φ by P (resp. $\neg\varphi$ by $\neg P$) to obtain the positive rule R (resp. the negative rule $\neg R$).

If the rule R (resp. $\neg R$) involves metavariables, an instantiation rule R_{inst} (resp. $\neg R_{\mathrm{inst}}$) is added, where one or several metavariables can be instantiated.

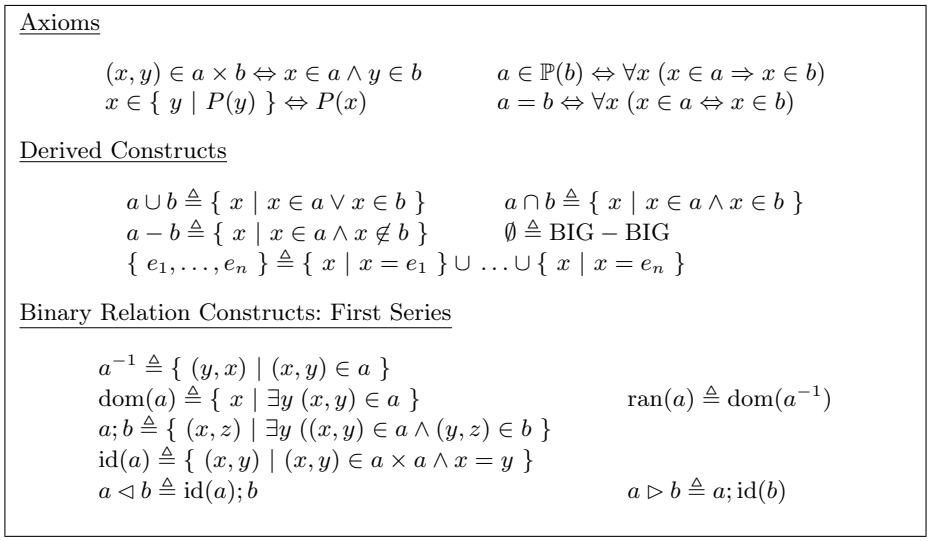

Fig. 2. Axioms and Constructs of the B Set Theory

3 Superdeduction Rules for the B Set Theory

The B method [1] is based on a typed set theory, which consists of six axiom schemes defining the basic operators and the extensional equality. The other operators ($\cup$, $\cap$, etc.) are defined using the previous basic ones. Figure 2 gathers a part of the axioms and constructs of the B set theory, where BIG is an infinite set. In this figure, we only consider the four first axioms of the B set theory, as we do not need the two remaining axioms in the rules that we want to verify (see Section 4). Due to space restrictions, we only present the main constructs, even though we can deal with other constructs (like functions) in our superdeduction system. Compared to [1], all type information has been removed from the axioms and constructs thanks to the modularity between the type and proof systems.

To generate the superdeduction rules corresponding to the axioms and constructs defined in Figure 2, we use the algorithm described in Definition 1 of Section 2, and we must therefore identify the proposition rewrite rules. On the one hand, the axioms are of the form $P_i \Leftrightarrow Q_i$, and the associated proposition rewrite rules are $R_i : P_i \rightarrow Q_i$. On the other hand, the constructs are expressed by the definitions $E_i \triangleq F_i$, where E_i and F_i are expressions, and the corresponding proposition rewrite rules are $R_i : x \in E_i \rightarrow x \in F_i$. The superdeduction rules are then generated as described in Figure 3 (except the instantiation rules associated with rules involving metavariables, due to space restrictions). The computation of these superdeduction rules goes further than the one proposed in Section 2, since given a proposition rewrite rule $R : P \rightarrow Q$, we apply to Q not only all the rules considered by Definition 1, but also the new generated superdeduction rules (except the rules for the extensional equality, in order to benefit from the dedicated rules of Zenon for equality) whenever applicable.

Rules for Axioms

$$\frac{(x,y) \in a \times b}{x \in a, y \in b}\ \times \qquad \frac{a \in \mathbb{P}(b)}{X \notin a \mid X \in b}\ \mathbb{P} \qquad \frac{x \in \{\ y \mid P(y)\ \}}{P(x)}\ \{|\}$$

$$\frac{(x,y) \notin a \times b}{x \notin a \mid y \notin b}\ \neg\times \qquad \frac{a \notin \mathbb{P}(b)}{\epsilon_x \in a, \epsilon_x \notin b}\ \neg\mathbb{P} \quad \text{with } \epsilon_x = \epsilon(x).\neg(x \in a \Rightarrow x \in b) \qquad \frac{x \notin \{\ y \mid P(y)\ \}}{\neg P(x)}\ \neg\{|\}$$

$$\frac{a = b}{X \notin a, X \notin b \mid X \in a, X \in b}\ = \qquad \frac{a \neq b}{\epsilon_x \notin a, \epsilon_x \in b \mid \epsilon_x \in a, \epsilon_x \notin b}\ \neq \quad \text{with } \epsilon_x = \epsilon(x).\neg(x \in a \Leftrightarrow x \in b)$$

Rules for Derived Constructs

$$\frac{x \in a \cup b}{x \in a \mid x \in b}\ \cup \qquad \frac{x \in a \cap b}{x \in a, x \in b}\ \cap \qquad \frac{x \in a - b}{x \in a, x \notin b}\ -$$

$$\frac{x \notin a \cup b}{x \notin a, x \notin b}\ \neg\cup \qquad \frac{x \notin a \cap b}{x \notin a \mid x \notin b}\ \neg\cap \qquad \frac{x \notin a - b}{x \notin a \mid x \in b}\ \neg-$$

$$\frac{x \in \{\ e_1, \ldots, e_n\ \}}{x = e_1 \mid \ldots \mid x = e_1}\ \{\} \qquad \frac{x \notin \{\ e_1, \ldots, e_n\ \}}{x \neq e_1, \ldots, x \neq e_n}\ \neg\{\} \qquad \frac{x \in \emptyset}{\odot}\ \emptyset$$

Rules for Binary Relation Constructs: First Series

$$\frac{(x,y) \in a^{-1}}{(y,x) \in a}\ a^{-1} \qquad \frac{x \in \mathrm{dom}(a)}{(x,\epsilon_y) \in a}\ \mathrm{dom} \quad \text{with } \epsilon_y = \epsilon(y).((x,y) \in a) \qquad \frac{y \in \mathrm{ran}(a)}{(\epsilon_x, y) \in a}\ \mathrm{ran} \quad \text{with } \epsilon_x = \epsilon(x).((x,y) \in a)$$

$$\frac{(x,y) \notin a^{-1}}{(y,x) \notin a}\ \neg a^{-1} \qquad \frac{x \notin \mathrm{dom}(a)}{(x,Y) \notin a}\ \neg\mathrm{dom} \qquad \frac{y \notin \mathrm{ran}(a)}{(X,y) \notin a}\ \neg\mathrm{ran}$$

$$\frac{(x,z) \in a;b}{(x,\epsilon_y) \in a, (\epsilon_y, z) \in b}\ ; \quad \text{with } \epsilon_y = \epsilon(y).((x,y) \in a \wedge (y,z) \in b) \qquad \frac{(x,z) \notin a;b}{(x,Y) \notin a \mid (Y,z) \notin b}\ \neg;$$

$$\frac{(x,y) \in \mathrm{id}(a)}{x = y, x \in a, y \in a}\ \mathrm{id} \qquad \frac{(x,y) \in a \lhd b}{(x,y) \in b, x \in a}\ \lhd \qquad \frac{(x,y) \in a \rhd b}{(x,y) \in a, y \in b}\ \rhd$$

$$\frac{(x,y) \notin \mathrm{id}(a)}{x \neq y \mid x \notin a \mid y \notin a}\ \neg\mathrm{id} \qquad \frac{(x,y) \notin a \lhd b}{(x,y) \notin b \mid x \notin a}\ \neg\lhd \qquad \frac{(x,y) \notin a \rhd b}{(x,y) \notin a \mid y \notin b}\ \neg\rhd$$

Fig. 3. Superdeduction Rules for the B Set Theory

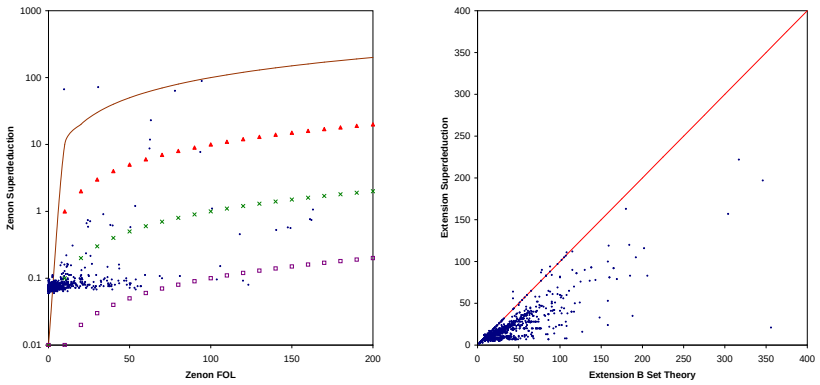

Fig. 4. Proof Time and Proof Size Comparative Benchmarks

4 Implementation and Benchmarks

The extension of Zenon for the B set theory described in Section 3 has been implemented thanks to the ability of Zenon to extend its core of deductive rules. The motivation of this extension is to verify B proof rules of Atelier B, and in particular rules coming from the database maintained by Siemens IC-MOL. Regarding benchmarks, we consider a selection of rules of this database consisting of well-typed and well-defined rules, which involve the B set constructs handled by our extension, i.e. all the constructs of the B-Book [1] until the override construct. This represents 1,397 rules (over a total of 5,281 rules), and we propose two benchmarks whose results are gathered in Figure 4.

The first benchmark aims to compare our extension of Zenon with the approach described in [5], where the set formulas must be preliminarily normalized (in order to obtain first order logic formulas containing only the membership set operator) before calling Zenon. Over the 1,397 selected rules, our extension proves 1,340 rules (96%), while our initial approach proves 1,145 rules (82%). The left-hand side graph of Figure 4 presents a comparison of both approaches in terms of proof time (run on an Intel Core i5-2500K 3.30GHz/12GB computer) for a subset of the 1,397 selected rules, where both approaches succeed in finding a proof (the time measures include the compilation of Coq proofs generated by Zenon), i.e. for 1,145 rules. In this figure, a point represents the result for a rule, and the x/y-axes respectively correspond to the approach with pre-normalization of the formulas and to our extension using superdeduction. On average, the superdeduction proofs are obtained 67 times faster (the best ratio is 1,540).

We propose a second benchmark whose purpose is to compare our extension of Zenon using superdeduction with another extension of Zenon for the B set theory, where the proposition rewrite rules are not computed into superdeduction rules, but just unfolded/folded (like in Prawitz's approach). The comparison consists in computing the number of proof nodes of each proof generated by

Zenon. We consider a subset of 1,340 rules, for which both extensions succeed in finding a proof. The results are summarized by the right-hand side graph of Figure 4, where a point represents the result for a rule, and where the x/y-axes respectively correspond to the extension without and with superdeduction. As can be seen, the major part of proofs in the latter are on average 1.6 times shorter than the former proofs (the best ratio is 6.25).

5 Conclusion

We have proposed a method which allows us to develop tableaux modulo theories using superdeduction. This method has been presented in the framework of the Zenon automated theorem prover, and applied to the set theory of the B method. This has allowed us to provide another prover to Atelier B, which can be used to verify B proof rules automatically. We have also proposed some benchmarks using rules coming from the database maintained by Siemens IC-MOL. These benchmarks have emphasized significant speed-ups both in terms of proof time and proof size compared to previous and alternative approaches.

As future work, we first aim to generalize our approach of superdeduction for Zenon and provide a generator of superdeduction rules from proposition rewrite rules. This will allow us to generate automatically a superdeduction prover from a theory, provided that a part of the axioms of this theory can be turned into proposition rewrite rules. We also plan to extend our implementation realized for verifying B proof rules in order to deal with a larger set of rules of the database maintained by Siemens IC-MOL. Finally, we intend to study some properties of this system for the B set theory, such as consistency and completeness.

Acknowledgement. Many thanks to G. Burel and O. Hermant for their detailed comments on this paper, to G. Dowek for seminal discussions of this work, and to D. Doligez for his help in the integration of superdeduction into Zenon.

References

1. J.-R. Abrial. *The B-Book, Assigning Programs to Meanings.* Cambridge University Press, Cambridge (UK), 1996. ISBN 0521496195.
2. R. Bonichon, D. Delahaye, and D. Doligez. Zenon: An Extensible Automated Theorem Prover Producing Checkable Proofs. In *Logic for Programming Artificial Intelligence and Reasoning (LPAR)*, volume 4790 of *LNCS/LNAI*, pages 151–165, Yerevan (Armenia), Oct. 2007. Springer.
3. P. Brauner, C. Houtmann, and C. Kirchner. Principles of Superdeduction. In *Logic in Computer Science (LICS)*, pages 41–50, Wrocław (Poland), July 2007. IEEE Computer Society Press.
4. C. Houtmann. Axiom Directed Focusing. In *Types for Proofs and Programs (TYPES)*, volume 5497 of *LNCS*, pages 169–185, Torino (Italy), Mar. 2008. Springer.
5. M. Jacquel, K. Berkani, D. Delahaye, and C. Dubois. Verifying B Proof Rules using Deep Embedding and Automated Theorem Proving. In *Software Engineering and Formal Methods (SEFM)*, volume 7041 of *LNCS*, pages 253–268, Montevideo (Uruguay), Nov. 2011. Springer.