Symbolic Deduction in Mathematical Databases based on Properties

Christoph Schwarzweller

Wilhelm-Schickard-Institute for Computer Science
University of Tübingen
Sand 13, D-72076 Tübingen
schwarzw@informatik.uni-tuebingen.de

Abstract. We claim that mathematical databases should be more than a collection of domains with associated theorems; in particular theorems should be stated as general as possible, that is independent of domains. A database then should be able to check whether such a general theorem holds in a particular domain. To this end we use a properties-based representation for both theorems and domains, and present a deduction calculus that using additional rules about the problem domain allows to perform such a theorem check.

1 Introduction

Numerous mathematical theorems have been proven with various mechanized reasoning systems. Among them are, for example, the proof of Robbin's conjecture in Otter [McC97], a proof of the Jordan curve theorem in Mizar [RT99], and a proof of the Chinese Remainder theorem in RRL [ZH93]. However, mathematical databases allowing to reuse such theorems in a general sense can hardly be found. The reason is that mechanized reasoning systems rely on rather involved logics and proof languages, so that proofs cannot be translated from one system to another easily. Thus the only possibility is often to prove such a theorem again or to include it as an axiom. Furthermore, the domain used to prove a theorem often includes unnecessary restrictions, for instance the above mentioned Chinese remainder theorem is proven for the integers and not for rings (or even more general domains). Hence even inside a reasoning system it is sometimes a non-trivial task to reuse an already proven theorem because the current domain does not fit to the one that has been used for the proof.

In this paper we present an approach to mathematical databases focussing on the reuse of theorems in various domains. We have proposed to organize mathematical databases by decoupling proving theorems and reusing them in other domains [Sch01]. To this end, theorems are decorated with sets of properties describing conditions under which a theorem holds. Thus by using as less properties as possible theorems are given in a general setting. These properties-based theorems then allow for checking their validity in particular domains by just checking whether the domain fulfills the properties connected with the theorem. Here, we present a calculus to perform this check in a Prolog-style manner. We think of such a checker as being part of a mathematical database.

2 Representation of Theorems and Domains

In this section we describe the representation of theorems and domains underlying our database approach. The key idea is to separate the content of a theorem from the properties necessary to prove a theorem correct [Sch01]. The content $\mathit{Cont}(T)$ of a theorem T states the proposition the theorem is about. It can be compared to a first-order formula. However, the domain and the operations necessary to express $\mathit{Cont}(T)$ are given separately in a signature $\mathit{Sig}(T)$. This allows to distinguish between the proposition of the theorem and conditions under which it holds. This is further elaborated in the third component of a theorem T. Here, a set of properties $\mathit{Prop}(T)$ is given. The intended meaning is that using these properties $\mathit{Cont}(T)$ can be proven correct. To enable easier deduction we represent properties by predicate symbols. Thereby, the arity of these symbols corresponds to the carriers and operations necessary to formulate the property as a first order formula. Summarized we consider a theorem T as a triple

$$T \;=\; (\mathit{Sig}(T), \mathit{Cont}(T), \mathit{Prop}(T))$$

where the statements of $\mathit{Cont}(T)$ and $\mathit{Prop}(T)$ fit to the given signature $\mathit{Sig}(T)$. The other way round $\mathit{Sig}(T)$ should not include more than necessary for the statements given in $\mathit{Cont}(T)$ and $\mathit{Prop}(T)$. Note, that we do not use a formal definition of properties in the sense of first-order logic here. We assume that the meaning of a property is indicated by its name, that is by the chosen predicate symbol. Consider, for example, the following theorem T.

Let R be a (commutative) ring. Then $\{0\}$ is an ideal in R.

Then we get in our notation

$$\mathit{Cont}(T) \;=\; \{0\} \textit{ is an ideal in } R.$$

It is a straightforward task to expand the right-hand phrase "$\{0\}$ *is an ideal in* R" into a first-order formula. More importantly, the signature necessary to formulate this proposition is

$$Sig(T) \;=\; (R, +, *, 0),$$

that is the symbol 1 usually part of a ring signature is not included. Furthermore, in order to prove that $\{0\}$ is an ideal in R it is only necessary that $+$ is associative, provides a right zero as well as right inverses and that $+$ and $*$ are distributive. So we get

$$\begin{aligned} Prop(T) = \{&\text{associative}(R,+),\ \text{right-zero}(R,+,0),\\ &\text{right-inverse}(R,+,0),\ \text{distributive}(R,+,*)\}, \end{aligned}$$

that is the properties connected with T are much weaker than the properties of a ring required in the original version of the theorem. Note that the arguments $R, +, *$ and 0 can be interpreted as variable symbols since they represent arbitrary carriers and operations respectively. This will play an important role later for the calculus.

Domains D can be represented in a similar way. They also consist of a signature $Sig(D)$ giving carriers and operations of the domain and a set $Prop(D)$ containing properties the domain fulfills, thus

$$D \;=\; (Sig(D), Prop(D)).$$

This works for both abstract domains such as rings or fields and concrete domains. For example, the ring of integers $\mathbb{Z}$ would look like

$$\begin{aligned} Sig(\mathbb{Z}) \;&\supseteq\; (\mathbb{Z}, +_{\mathbb{Z}}, *_{\mathbb{Z}}, 0_{\mathbb{Z}}, 1_{\mathbb{Z}})\\ Prop(\mathbb{Z}) \;&\supseteq\; \{\text{associative}(\mathbb{Z}, +_{\mathbb{Z}}), \text{distributive}(\mathbb{Z}, +_{\mathbb{Z}}, *_{\mathbb{Z}}),\\ &\qquad \text{commutative}(\mathbb{Z}, +_{\mathbb{Z}}), \text{commutative}(\mathbb{Z}, *_{\mathbb{Z}}),\\ &\qquad \text{Euclidean}(\mathbb{Z}, +_{\mathbb{Z}}, *_{\mathbb{Z}}, 0_{\mathbb{Z}})\} \end{aligned}$$

where $\mathbb{Z}, +_{\mathbb{Z}}, *_{\mathbb{Z}}, 0_{\mathbb{Z}}$ and $1_{\mathbb{Z}}$ are now constant symbols. Thus the approach allows for the description of both domains and theorems with regard to properties of domains and properties ensuring the correctness of theorems. This gives rise to a straightforward criterion whether a theorem T holds in a domain D where D may be both an abstract or a concrete domain. It has only to be checked whether D provides both the necessary signature and the properties connected with T, thus

$$Cont(T) \text{ holds in } D \;:\Longleftrightarrow\; Sig_D(T) \subseteq Sig(D) \wedge Prop_D(T) \subseteq Prop(D).$$

The notations $Sig_D(T)$ and $Prop_D(T)$ resp. mean that the variable symbols occurring in the theorem T, actually in $Sig(T)$, are replaced by the corresponding symbols of the domain D. Note that the failure of this test not necessarily implies that a theorem does not hold in a domain, the check is relative to the properties stated about the theorem and the domain. Reasoning is not necessary here, just checking whether certain properties, that is predicates connected with domains and theorems, are present. Nevertheless the deduced results are correct provided that the meaning of the properties was defined properly. Note, that this method also allows for straightforward error messages by collecting the properties of T not included in the ones of D.

3 A Calculus for Deducing Properties

The distinction between the content of a theorem and properties under which this content can be proven allows for checking whether theorems hold in a domain by comparing sets of properties with respect to inclusion. However, this setting is too restricted. For example, if a theorem T requires the property left-distributive, and a domain D obeys the property distributive, it is not desirable that left-distributive has to be added to the domain's properties. The mathematical database should rather be able to conclude that T holds in D, although the sets of properties involved are not related by inclusion.

In the following we replace the subset relation between two sets P_1 and P_2 of properties by a relation $P_1 \Longrightarrow P_2$ with the meaning that every domain D that fulfills the properties of P_1 also fulfills the ones in P_2, or more formally

$$\models P_1 \Longrightarrow P_2 \quad :\Longleftrightarrow \quad \forall D : D \models P_1 \text{ implies } D \models P_2$$

where $\models$ on the right-hand side is the model operator well-known from first-order logic. Thus the content of a properties-based theorem $T = (Sig(T), Cont(T), Prop(T))$ holds in a domain $D = (Sig(D), Prop(D))$ if both $Sig_D(T) \subseteq Sig(D)$ and $\models Prop(D) \Longrightarrow Prop_D(T)$ are valid. Note that $\models P_1 \Longrightarrow P_2$ corresponds to the usual semantic implication. However, in a mathematical database as proposed in the last section the formulas occurring in P_1 and P_2 are given by a set of predicates only, whose arguments are either variables or constants.

Obviously, an implication $P_1 \Longrightarrow P_2$ cannot be checked in this generality, in particular if we represent properties by predicate symbols without giving the definition of properties as a first-order formula. Therefore we incorporate a set of rules L describing basic relations between sets of properties. For example, the following rule

$$\begin{array}{l}\{\text{distributive}(R,+,*)\} \longrightarrow \\ \{\text{left-distributive}(R,+,*),\ \text{right-distributive}(R,+,*)\}\end{array} \qquad (\mathbf{A})$$

states that structures $(R,+,*)$ that are distributive are also both left- and right-distributive. In this way a mathematical database is provided with additional knowledge about the problem domain. The deduction of $\models P_1 \Longrightarrow P_2$ is then performed relative to such a set of rules.

The calculus has two axioms. The first mirrors the fact, that an implication $P_1 \Longrightarrow P_2$ trivially holds, if $P_2 \subseteq P_1$. The second axiom allows to incorporate the external rules: if $l \longrightarrow r \in L$, then $\sigma(l)$ implies $\sigma(r)$ where σ is an arbitrary substitution compatible with the signature. Further on, there are a rule allowing to combine different implications P_2 and P_3 both made from P_1 and a rule for concatenating implications $P_1 \Longrightarrow P_2$ and $P_2 \Longrightarrow P_3$.

$$\frac{P_2 \ \subseteq \ P_1}{\vdash \ P_1 \Longrightarrow P_2} \qquad (\mathbf{AX1})$$

$$\frac{l \ \longrightarrow \ r \ \in L}{\vdash \ \sigma(l) \Longrightarrow \sigma(r)} \qquad (\mathbf{AX2})$$

$$\frac{\vdash \ P_1 \Longrightarrow P_2, \ \ \vdash \ P_1 \Longrightarrow P_3}{\vdash \ P_1 \Longrightarrow P_2 \cup P_3} \qquad (\mathbf{R1})$$

$$\frac{\vdash \ P_1 \Longrightarrow P_2, \ \ \vdash \ P_2 \Longrightarrow P_3}{\vdash \ P_1 \Longrightarrow P_3} \qquad (\mathbf{R2})$$

Provided that the rules in L are correct, that is if from $l \longrightarrow r \in L$ indeed follows $\models \sigma(l) \Longrightarrow \sigma(r)$ for the substitutions σ used in a deduction, it is straightforward to see that the calculus is correct. In other words, we have that (relative to L) $\vdash P_1 \Longrightarrow P_2$ implies $\models P_1 \Longrightarrow P_2$. However, if no deduction sequence is found this does not necessarily mean that $\models P_1 \Longrightarrow P_2$ is not valid. The reason is that the calculus checks for implications with respect to the rule set L only. In other words, if L does not contain enough knowledge about the problem domain, the deduction of an implication may fail, although this implication is true.

The calculus can be extended with some straightforward derived rules, among them

$$\frac{\vdash\ P_1 \Longrightarrow P_2 \cup P_3}{\vdash\ P_1 \Longrightarrow P_2} \qquad (\mathbf{L1})$$

$$\frac{\vdash\ P_1 \Longrightarrow P_2,\ \ P_1\ \subseteq\ P_3}{\vdash\ P_3 \Longrightarrow P_2} \qquad (\mathbf{L2})$$

These rules can be easily proven correct in the sense that their consequences can be deduced from their premises in the original calculus. To see how the calculus works let us deduce the following implication.

$$\begin{aligned}\vdash\ &\{\text{associative}(\mathbb{Z},+_{\mathbb{Z}}),\ \text{distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}})\}\ \Longrightarrow\\ &\{\text{associative}(\mathbb{Z},+_{\mathbb{Z}}),\\ &\ \text{left-distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}}),\ \text{right-distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}})\}\end{aligned}$$

To do so, we assume that the distributivity rule **A** from above is present in the set of rules L. Then, using Lemma **L2**, we get the following deduction sequence. Note that the actual definition of the properties involved has no influence on the deduction, that is the deduction is purely symbolic.

$$\begin{aligned}
(1) \vdash\ &\{\text{associative}(\mathbb{Z},+_{\mathbb{Z}}),\ \text{distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}})\}\ \Longrightarrow\\
&\{\text{associative}(\mathbb{Z},+_{\mathbb{Z}})\}\\
&\ \text{by } \mathbf{AX1}\\
(2) \vdash\ &\{\text{distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}})\}\ \Longrightarrow\\
&\{\text{left-distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}}),\ \text{right-distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}})\}\\
&\ \text{by } \mathbf{AX2} \text{ with } \mathbf{A} \text{ and } \sigma(R)=\mathbb{Z},\ \sigma(+)=+_{\mathbb{Z}},\ \sigma(*)=*_{\mathbb{Z}}\\
(3) \vdash\ &\{\text{associative}(\mathbb{Z},+_{\mathbb{Z}}),\ \text{distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}})\}\ \Longrightarrow\\
&\{\text{left-distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}}),\ \text{right-distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}})\}\\
&\ \text{by } \mathbf{L2}(2)\\
(4) \vdash\ &\{\text{associative}(\mathbb{Z},+_{\mathbb{Z}},\ \text{distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}})\}\ \Longrightarrow\\
&\{\text{associative}(\mathbb{Z},+_{\mathbb{Z}}),\\
&\ \text{left-distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}}),\ \text{right-distributive}(\mathbb{Z},+_{\mathbb{Z}},*_{\mathbb{Z}})\}\\
&\ \text{by } \mathbf{R1}(1,3)
\end{aligned}$$

Thus a theorem T requiring for instance the properties associative$(R,+)$, left-distributive$(R,+,*)$ and right-distributive$(R,+,*)$ holds in particular for $\mathbb{Z}$. Though for $\mathbb{Z}$ only the property distributive$(R,+,*)$ has been stated, this can be checked by a mathematical database using the calculus. Note that, by just taking the identity substitution for σ, the above sequence can be easily transformed in a deduction sequence using $R,+$ and $*$ instead of $\mathbb{Z},+_{\mathbb{Z}}$ and $*_{\mathbb{Z}}$, that is general lemmas can be shown.

Finding a deduction sequence for an implication $P_1 \Longrightarrow P_2$ requires some amount of guessing in which way the set on the left-hand side of an implication has to be extended, that is guessing which property should be additionally considered in order to combine already deduced implications. This can be seen, for example, in step (3) of the deduction above where using Lemma **L2** the set on the left-hand side is extended from $\{\text{distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z})\}$ to $\{\text{associative}(\mathbb{Z}, +_\mathbb{Z}),\ \text{distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z})\}$; any other extension would have been a correct application of **L2**, too. Fortunately, this problem can be avoided using backward propagation. The idea is, given an implication $P_1 \Longrightarrow P_2$, to successively remove properties from P_2 that are implied by the ones from P_1. We use the following three rules.

(B1) Replace $\vdash P_1 \Longrightarrow P_2$ by $\vdash P_1 \Longrightarrow P_2 \backslash (P_1 \cap P_2)$.
(B2) Replace $\vdash P_1 \Longrightarrow P_2$ by $\vdash P_1 \Longrightarrow (P_2 \backslash (\sigma(r) \cap P_2)) \cup \sigma(l)$ if there are a rule $l \longrightarrow r \in L$ and a substitution σ with $\sigma(r) \cap P_2 \neq \emptyset$.
(B3) Accept $\vdash P_1 \Longrightarrow \emptyset$.

Thus an implication $P_1 \Longrightarrow P_2$ is accepted, if it can be transformed into an implication of the form $P_1 \Longrightarrow \emptyset$. The rules are correct with respect to the above calculus in the sense that every deduction starting with $P_1 \Longrightarrow P_2$ and ending with $P_1 \Longrightarrow \emptyset$ using **B1** - **B3** can be translated into a correct sequence of the original calculus. For the example from above we get

$\vdash \{\text{associative}(\mathbb{Z}, +_\mathbb{Z}),\ \text{distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z})\} \Longrightarrow$
$\{\text{associative}(\mathbb{Z}, +_\mathbb{Z}),$
$\text{left-distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z}),\ \text{right-distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z})\}$
$\vdash \{\text{associative}(\mathbb{Z}, +_\mathbb{Z}),\ \text{distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z})\} \Longrightarrow$
$\{\text{left-distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z}),\ \text{right-distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z})\}$
by **B1**
$\vdash \{\text{associative}(\mathbb{Z}, +_\mathbb{Z}),\ \text{distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z})\} \Longrightarrow$
$\{\text{distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z})\}$
by **B2** with **A** and $\sigma(R) = \mathbb{Z},\ \sigma(+) = +_\mathbb{Z},\ \sigma(*) = *_\mathbb{Z}$
$\vdash \{\text{associative}(\mathbb{Z}, +_\mathbb{Z}),\ \text{distributive}(\mathbb{Z}, +_\mathbb{Z}, *_\mathbb{Z})\} \Longrightarrow \emptyset$
by **B1**

which is accepted by **B3**. Note, that the only choice throughout the deduction consists of determining which rule of L should be applied. Also the left-hand side P_1 of the goal does not change throughout the whole deduction, so that keeping track of the changes occurring in P_2 is sufficient. Finally it may be worth mentioning that for rules $l \longrightarrow r$ with a right-hand side r consisting of one property only, **B2** can be simplified to

(B2') Replace $\vdash P_1 \Longrightarrow P_2$ by $\vdash P_1 \Longrightarrow (P_2 \backslash \sigma(r)) \cup \sigma(l)$ if there are a rule $l \longrightarrow r \in L$ and a substitution σ with $\sigma(r) \in P_2$.

Thus no intersection has to be computed in this case. Note, that this kind of rules can be easily obtained by splitting up given rules as for example the distributivity rule from above into the following two ones.

$$\begin{aligned}\{\text{distributive}(R,+,*)\} &\longrightarrow \{\text{left-distributive}(R,+,*)\}\\ \{\text{distributive}(R,+,*)\} &\longrightarrow \{\text{right-distributive}(R,+,*)\}\end{aligned}$$

However, transforming all the rules of L this way would heavily increase the number of steps in a deduction and only detailed experiments will show which method is to prefer.

4 Conclusion

The calculus presented provides mathematical databases with additional knowledge: it allows to infer implications of sets of properties with respect to a given set of basic rules. In the database both theorems and domains are represented based on properties so that the implication of sets of properties is sufficient to check whether a theorem holds in a particular domain. Thus theorems stated in this general manner can be checked for validity in special domains easily.

Stating theorems with respect to properties rather than domains may at first glance be somewhat unfamiliar. However, if working in a particular domain only, the familiar representation of theorems can be easily regained. For example, a theory of rings can be constructed as follows. First—if this has not already been done—the database has to be extended by a domain $R = (Sig(R), Prop(R))$ with the appropriate signature and properties defining rings. Then all theorems T with $Sig_R(T) \subseteq Sig(R)$ and $\models Prop(R) \Longrightarrow Prop_R(T)$ can be extracted from the database, in this way building a new database for rings.

Mechanized reasoning systems can be incorporated the following way. Theorems and properties included in the library as well as rules used for deduction can be proven with a mechanized reasoning system (provided that first-order formulae has been attached to the properties' predicate symbols). Note, that this indeed is a realization of proving theorems and properties of domains on the one side, and storing and reusing knowledge in a mathematical database on the other side. Mechanized reasoning systems allowing to formulate and prove properties-based theorems are for example Mizar [RT99], Imps [Far93], and Theorema [BJK97].

The properties-based approach has applications in other areas, for instance in the area of generic programming. Here, generic algorithms obey type parameters which are instantiated later to get a running instance of the algorithm. Both the feasibility and the correctness of such an instance depends on whether the operations instantiated fulfill certain properties which is usually not checked. Using an properties-based approach, that is providing generic algorithms and possible instantiations with properties required resp. fulfilled, this can be done in terms of the calculus presented.

References

[BJK97] B. Buchberger, T. Jebelean, F. Kriftner, M. Marin, E. Tomuta, and D. Vasaru, A Survey on the Theorema Project, in: Proceedings of ISSAC'97 (International Symposium on Symbolic and Algebraic Computation), ed. W. Küchlin, ACM Press, 1997, pp.384-391.

[Far93] W. M. Farmer, J. D. Guttman, and F. J. Thayer, IMPS: An Interactive Mathematical Proof System, Journal of Automated Reasoning, 11 (1993) 213-248.

[McC97] W. McCune, *Solution of the Robbins Problem*; in: Journal of Automated Reasoning (19), p. 263-276, 1997.

[RT99] Piotr Rudnicki and Andrzej Trybulec, *On Equivalents of Well-foundedness. An Experiment in Mizar.* in: Journal of Automated Reasoning, 23:197–234, 1999.

[Sch01] C. Schwarzweller, *Designing Mathematical Libraries based on Minimal Requirements for Theorems.* in: Proceedings of the First International Workshop on Mathematical Knowledge Management (MKM2001), 2001.

[ZH93] H. Zhang and X. Hua, *Proving the Chinese Remainder Theorem by the Cover Set Induction*; in: D. Kapur (ed.), Proceedings of the 1992 International Conference of Automated Deduction, LNAI 607, Springer-Verlag, p. 431-445, 1993.