

Mechanizing Complemented Lattices Within Mizar Type System

Adam Grabowski¹

Received: 1 November 2013 / Accepted: 26 May 2015 / Published online: 18 June 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract Recently some longstanding open lattice theory problems were solved with the help of automated theorem provers. The question which may be posed is how to cope with such results to improve their presentation for human without loss of machine-readability, not only at the proof level, which should be rather straightforward, but also at the stage of rebuilding appropriate data structure. We describe the framework extending already existed in the Mizar library for Boolean algebras to cover more general cases of lattice with complements. The efficiency of this approach was tested e.g. on short axiom systems for Boolean algebras based on negation and disjunction. We also proved Nachbin theorem for spectra of distributive lattices.

Keywords Formalization of mathematics · Mizar · Complemented lattices

1 Introduction

Lattices are important structures which can be applied in many mathematical theories. It is not surprising that they are also present in repositories of automatically verified mathematical knowledge. As forty years of MIZAR [25] is connected with the forthcoming 25th anniversary of the MIZAR Mathematical Library (MML), we describe our enhancement of the data structure already developed for lattices in MML. The article where lattices were introduced formally [27] is numbered 26 in the list of all MIZAR articles [21]; only incidence structures were introduced earlier, before the rich hierarchy of group, fields, rings, and vector spaces.

The author was heavily involved with lattices in MIZAR; either during the formalization of *Compendium of Continuous Lattices* (CCL) [3], or certifying in MIZAR some well-known

✉ Adam Grabowski
adam@mizar.org

¹ Institute of Informatics, University of Białystok, Ciołkowskiego 1M, Białystok 15-245, Poland

problems of axiomatizations of Boolean algebras – as, e.g. Robbins' problem, various axiom systems in terms of the Sheffer stroke or disjunction and negation etc., authoring 14 MIZAR articles devoted to the lattice theory. We choose some problems solved with the help of automated deduction systems. As a test for flexibility and effectiveness of the framework for lattice theory coded in MML we translate some of the examples of such output (which is essentially hard to follow by human) into more human-readable format without loss of machine readability. We describe selected issues and methods concerning the mechanization of certain class of lattices, namely complemented ones, including Boolean algebras (BA).

The problem of development of a framework to present a machine proof is not new. Large algebraic hierarchy has been built especially to prove (also in Coq) the Fundamental Theorem of Algebra [6]. The main difference comparing to our approach is that we didn't construct the whole framework from scratch. A huge library of formalized mathematical knowledge with an open architecture [1] may become not so convenient to work with if we have to integrate approaches developed in many styles by different authors.

The structure of this paper is as follows. In Section 2 the mathematical and historical background is given, the next section describes the structures we used. Fourth section contains basic notions for Boolean algebras which were available in MML. Section 5 deals with main mechanisms and issues which were of major importance during our work with adopting other proofs into MIZAR. Some statistics showing position of the MIZAR proof among other machine proofs is shown. In the sixth section we describe experiences where complementation operator is not explicitly given (Nachbin's theorem) while in the last sections we draw some concluding remarks and present future work.

We cite definitions straight from the corresponding *abstract* files which are generated from articles and which contain only items exported to the library, so correctness conditions are not given (as well as justification of theorems, schemes, and clusters). Published MIZAR articles cited in this paper are available in *Formalized Mathematics* where their MML identifier is given.

2 Background

In 1933 Huntington [14, 15] presented the following basis for Boolean algebras:

$$\begin{aligned} a + b &= b + a, \\ (a + b) + c &= a + (b + c), \\ (a' + b')' + (a' + b)' &= a; \end{aligned}$$

where a, b, c are arbitrary elements of an abstract algebra $(L, +, ')$.

In this *set of independent postulates for the algebra of logic* the last one is called *Huntington equation*. Robbins proposed another axiom as a replacement for Huntington equation, namely:

$$((a + b)' + (a + b')')' = a.$$

The hypothesis that commutativity and associativity together with *Robbins equation* is sufficient to characterize Boolean algebras remained open for over 60 years. In 1992 Winker gave a partial solution for the problem by proposing several other conditions which imply Huntington axiom. One of them, named *Winker's second condition*, played a crucial role in

the solution of Robbins problem. The proof that all Robbins algebras satisfy also Winker condition given by McCune with the help of automated theorem prover EQP/OTTER (October 1996) started a series of papers devoted to methods of proving/analysing that fact in different systems (Mathematica [5], ILF [4]). Shortly thereafter the problem has been also formalized in MIZAR language. However, we decided not to include the article to the MML as we proved a partial result – that algebras satisfying Robbins equation satisfy also Winker’s second condition. The aim of our work was quite different: “another attempt to translate EQP/OTTER proof into more human-readable form” [17]. In our opinion, the state of MIZAR database at that time (1997, MIZAR version 5.3.07, MML version 3.5.544, presently the number of Mizar articles in MML approaches 1240, twice as much as in 1997) posed us far from the acceptable way of formalization. Numerous revisions of MML urged us to revise also our proof to keep it correct with respect to MML and enabled us to find a way of more convenient construction of the proof. Later on, in [9] we formalized a full proof of the fact that all Robbins algebras are Boolean and, as far as we know, it was the first mechanically checked proof of Robbins conjecture which took into account Huntington’s [15] paper as well. Furthermore, extensive use of notions and theorems for lattice theory already defined in the MIZAR repository suggested their possible improvements, improving also lattice-theoretic representation of other areas of mathematics [7].

3 Hierarchy of Structures

According to the classical definition, Boolean algebra is a 6-tuple of the form $\langle B, \vee, \wedge, ', 0, 1 \rangle$, where \vee, \wedge are binary operations in B , $'$ is a function from B into B and $0, 1$ are two distinct elements of B ; the axiomatics concerned with it is well known. The structures in MIZAR are not tuples but rather partial functions on selectors which is more appropriate for implementing inheritance mechanisms. One of the first approaches was to introduce a new MIZAR structure as an ancestor of `GroupStr` and `doubleLoopStr` as defined in `VECTSP_1`. The latter one has been used widely e.g. while developing theory of polynomials in MIZAR, especially formalizing Fundamental Theorem of Algebra. This structure was also used in our very first attempt to formalize Robbins algebras.

The backbone structure for almost all MML structures is `1-sorted` and all considered structures in the paper are descendants of it. Some of the objects have only one prefix (an ancestor) – in that case additional fields (*selectors*) are given. The number of multiprefixed structures in MML is relatively high (45 out of total 145). Essentially, structures play a role of signatures for domains. The definition of a structure provides only the typing data for operators (selectors) and this information is enriched by the use of adjectives which will be explained in the next section in more detail.

Our motivation was to be as close as possible to Huntington’s original paper where the underlying structure was $\langle B, \vee, ' \rangle$. Moreover, intensive work with formalization of CCL (nearly 60 MIZAR articles and numerous revisions of MML) made lattice theory developed in MIZAR more suitable to work with. In fact, the only two definitions of attribute “Boolean” in MML were introduced just for lattices. The first one was for posets, and the other for algebras with two binary operations. The first approach has its full exposure in [3]. Inheritance mechanisms implemented in MIZAR verifier allow for applying the attributes in all descendant structures.

We cite as an example the definition of a structure with signature (2,1) which may be treated as orthosemilattice. It is prefixed by the sup-semilattice and structure with complementation.

```

definition
  struct (\/-SemiLattStr, ComplStr) ComplLattStr
    (# carrier -> set,
      L_join -> BinOp of the carrier,
      Compl -> UnOp of the carrier #);
end;

```

The `ComplLattStr` has been introduced to keep defined notions and theorems at a satisfactory level of generality. It has two antecedents: upper semilattice – from which `L_join` is inherited, and a structure with complement operation which brings the field `Compl`. The `carrier` is inherited from both structures. The `OrthoLattStr` ensures that all the attributes (with “Boolean” at the top) for `LattStr` can be used. Obviously, full correspondence between newly introduced Robbins and previously defined in MML Boolean algebras can be shown for lattices, not for semilattices.

4 Boolean Algebras in MIZAR

In this section we shall describe briefly the formal apparatus for lattices introduced in MML [27] and its correspondence to axiom systems proposed by Huntington (the first set of independent postulates for the algebra of logic [15]). The lattice theory in MIZAR is built on `LattStr` as a fundamental structure mode. The mode `Lattice` is defined as `Lattice-like` non empty `LattStr`, where the single adjective `Lattice-like` is a shorthand for six attributes: concerning the supremum (join-commutative, join-associative) and the infimum (meet-commutative, meet-associative) as well as the both operations (join-absorbing and meet-absorbing) – and the names are self-explanatory enough. It could be noticed that many theorems and even definitions are often stated with all six attributes although the proof effectively uses only two–three of them. We have software detecting assumptions and proof steps which are not used but we still lack mechanisms to find automatically more general required type or conditions under which the proposition remains true.

Obviously, the problem of keeping notions at the possibly high level of generality is particularly important in articles which form the foundations of a given field of mathematics. It can be explained better using the definitions taken from [27]; the existence of the upper bound may be defined for semilattices:

```

definition let IT be non empty \/-SemiLattStr;
  attr IT is upper-bounded means
:: LATTICES: def 14
  ex c being Element of IT st
    for a being Element of IT holds c "\/" a = c & a "\/" c = c;
end;

```

The above adjective is used in the assumption of the following definition:

```

definition
  let L be non empty \/-SemiLattStr;
  assume L is upper-bounded;
  func Top L -> Element of L means
:: LATTICES: def 17
  for a being Element of L holds it "\/" a = it & a "\/" it = it;
end;

```

The definition of the upper bound for arbitrary upper semilattice structure is *permissive*: if the semilattice L is not necessarily upper-bounded, all we can say about $\text{Top } L$ is that is an element of L , but we do not know which one. If the semilattice L is upper-bounded according to the above definition, i.e. if a suitable element of the carrier of L exists, it plays a role of the upper bound for L . Obviously, to use the definiens of $\text{Top } L$ one has to prove that L is upper-bounded. Due to permissiveness of the bounds in the definition of the predicate `is_a_complement_of` the attribute bounded for one of the variables can be dropped.

```

definition
  let L be non empty LattStr, a, b be Element of L;
  pred a is_a_complement_of b means      :: LATTICES:def 18
    a "\/" b = Top L & b "\/" a = Top L &
    a "/" b = Bottom L & b "/" a = Bottom L;
end;

```

Collecting all previous notions together, we can properly define lattices with the complementation operator:

```

definition let IT be non empty LattStr;
  attr IT is complemented means      :: LATTICES:def 19
    for b being Element of IT ex a being Element of IT st a is_a_complement_of b;
end;

```

Of course, both definitions don't claim the uniqueness of the complement. MIZAR functors `Top` and `Bottom` have been defined previously for bounded lattices. Then, to use the attribute `complemented` one had to prove that the underlying object has all these properties which was obviously a big disadvantage. We added two conjuncts to the first definiens to get rid of attribute `join-commutative` for L . The notation $a \text{ "\/" } b$ is a shorthand for (the L -join of L) . $[a, b]$, i.e. the application of a structure selector to an ordered pair of elements of the carrier, dually is $a \text{ "/" } b$.

What is worth noticing at this point is that the classical axiomatics (see e.g. [13]) consists of 8 axioms and the idempotency laws can be derived from the other axioms (namely absorption). For any element a of a structure in which both absorption laws hold $(a \wedge (a \vee a)) \vee a$ is equal to a . These two axioms may be introduced as attributes and the appropriate collection of adjectives (called a *cluster* in MIZAR jargon) can be automatically enriched by these two additional properties. Huntington's definition of a Boolean algebra agrees with that of [27], with one exception: according to the definition in [27], the carrier of a lattice may be a singleton. This contradicts with Huntington's *the first set of postulates for the algebra of logic*, where we have as an axiom:

VI. *There are at least two distinct elements in the class K .*

"Boolean" is an abbreviation for "bounded complemented distributive" Lattice. A few explanations are due to the definition of an adjective "bounded". On the one hand, a bounded lattice can be seen as a structure with additional 0 and 1 . This leads to introducing three additional structures to make inheritance implemented for MIZAR structures working. On the other hand, lower and upper bounds can be defined separately, not as selectors in underlying structures, but using adjectives. The latter approach has been chosen in the MIZAR library. In a private conversation with Andrzej Trybulec we agreed that having separate fields for the top and the bottom was a really unfortunate idea; similarly as with unity or zero for 1-sorted structures. Most of inheritance issues works better if the existence of the appropriate object (as the bottom element of the lattice) is guaranteed by the attribute. Some problems with equational characterization can occur, but it's not that dangerous.

5 Lattices with Complementation Operator

We could define new MIZAR modes, e.g. `BooleanAlgebra` (which satisfies commutativity, associativity and Huntington equation) and `RobbinsAlgebra` with commutativity, associativity and Robbins axiom. However, introducing individual axioms by adjectives is more elegant and provide easier generalizations. We work mainly with seven attributes for `OrthoLattStr`, that is `join-associative`, `join-commutative`, `Boolean`, `Huntington`, `Robbins`, `deMorgan`, and `well-complemented`. While the first three were described in the previous section, let us cite two others, which are defined for `ComplLattStr` and are fundamental for our formalization:

```
definition let L be non empty ComplLattStr;
  attr L is Robbins means      :: ROBBINS1:def 5
    for x, y being Element of L holds ((x + y)' + (x + y')')' = x;
  attr L is Huntington means  :: ROBBINS1:def 6
    for x, y being Element of L holds (x' + y')' + (x' + y)' = x;
end;
```

It is clear that to have the notion of Robbins algebra according to the literature, one has to define it as `Robbins join-commutative join-associative non empty ComplLattStr`. Because the adjective `Boolean` cannot be used with these attributes (we have only one binary operation defined), we have to conclude our formalization of this part of the proof with the following conditional cluster registration:

```
registration
  cluster Robbins -> Huntington
    for join-associative join-commutative non empty ComplLattStr;
end;
```

Conditional registrations [10] significantly enhance our MIZAR formalization. To make use of the whole formal apparatus from [27], we introduced two adjectives of rather technical nature: if we read `OrthoLattStr` as $(L, \vee, \wedge, ')$, then attribute `deMorgan` means that \wedge is properly defined in terms of \vee and $'$. In MIZAR text:

```
definition let L be non empty OrthoLattStr;
  attr L is de_Morgan means
    for x, y being Element of L holds x /\ y = (x' /\ y')';
end;
```

Taking into account that in de Morgan algebras the complement operation is involutive, we can use either the above or the dual form $((x \vee y)' = x' \wedge y')$. If L is of the type `non empty OrthoLattStr`, all we can say about a' for an arbitrary element a of L is that a' is again an element of L . In other words, $'$ is an unary operation on the carrier of L . The second attribute guarantees that $'$ has properties of a complement as defined in [27].

```
definition let L be non empty OrthoLattStr;
  attr L is well-complemented means
    for a being Element of L holds a' is_a_complement_of a;
end;
```

One of the most important questions which arose during our work was how to establish a connection between $\langle L, \vee, ' \rangle$ and $\langle L, \vee, \wedge, ' \rangle$ (or rather $\langle L, \vee, \wedge \rangle$). We wanted to inherit the carrier, the join and the complement operation in underlying structures and add one more binary operation of meet which is defined in terms of \vee and $'$. We decided to define

a MIZAR functor which transforms a semilattice into the corresponding lattice structure (CLatt). The main theorem stating that all Robbins algebras are Boolean is expressed as a conditional cluster registration:

```
registration
  cluster Robbins de_Morgan -> Boolean for preOrthoLattice;
  cluster Boolean -> Robbins for well-complemented preOrthoLattice;
end;
```

where `preOrthoLattice` is `OrthoLattStr` satisfying all six lattice axioms.

As a test of the theory development level we tried to code some further results of McCune et al. [18] obtained with a help of PROVER9. The 2-basis discovered by Meredith [18] characterizes also the class of Boolean algebras. The axioms, which in MIZAR article ROBBINS2 we named (MD₁) and (MD₂), are as follows:

$$(x' + y)' + x = x$$

$$(x' + y)' + (z + y) = y + (z + x)$$

Although the original proof (written without computer assistance) of equivalence of this system and BAs due to Meredith is available, we used PROVER9 to produce proofs of equivalence (MD₁) and (MD₂) with Huntington's set of equations. One of the aims was to test the efficiency of adjectives and clusters introduced during formalization of the Robbins problem. The final equivalence of the axiom sets is given by the following two registrations of clusters:

```
registration
  cluster satisfying_MD_1 satisfying_MD_2 de_Morgan -> Boolean
    for preOrthoLattice;
  cluster Boolean -> satisfying_MD_1 satisfying_MD_2
    for well-complemented preOrthoLattice;
end;
```

The authors of [18] show that the equation (DN₁)

$$\left(((x + y)' + z)' + (x + (z' + (z + u)'))' \right)' = z$$

is a 1-basis for Boolean algebras in terms of disjunction and negation. The proof shows direct equivalence of systems based on (DN₁) and Robbins axioms.

We had essentially no serious problems during this work. We don't concentrate here on the length of the proofs, but essentially they are as long as corresponding PROVER9 proof objects. The special influence on the quality and smoothness of MIZAR-ing had the observation that if we split the blocks of cluster registrations into parts in order to be accessible as soon as it was proved, it is more effective and the proofs are less complicated. As an example, introducing as soon as it was possible the following:

```
registration
  cluster satisfying_DN_1 -> join-commutative for non empty ComplLattStr;
end;
```

we made references for the definition of the attribute "join-commutative" unnecessary since it became obvious for the MIZAR checker. It results in many lemmas, some of them of only technical character, but their importance can be automatically measured as we described in [12].

6 Nachbin Theorems for Distributive Lattices

Nachbin results deal basically with the correspondence between prime and maximal ideals in the class of distributive lattices. But originally, if we dig through the proof, it uses bottom element, which suggests that additionally we have the assumption of boundedness. Partial explanation of such situation is given in Balbes, Dwinger [2], where we can read often about “distributive lattices” where 0 and 1 are mentioned later. They claim that if the bottom is not an element of the structure, it can be easily attached. In fact, the so-called Jaśkowski star-addition operator which adds to arbitrary lattice an external bound is also formalized in MML (LATSUM_1). In MIZAR, the problem is that if the lattice doesn’t have the adjective bounded, the functor `Bottom L` is meaningless (you cannot use it) as we already described the issues of permissiveness before.

We proved the following variant of the above (in an original spelling from [13]):

Theorem 123 *Let L be a bounded distributive lattice with $0 \neq 1$. Then L is a boolean lattice iff $\text{Spec } L$ is unordered.*

where $\text{Spec } L$ (the spectrum of L) denotes the set of all prime ideals of L , ordered by set-theoretic inclusion; *unordered* means that the spectrum is just an antichain (arbitrary elements of this set are not comparable w.r.t. \subseteq). Note that the notion of an ideal varies in the literature; especially when primality is taken into account, as ideals of L usually are proper subsets of the carrier of L . But once in MML ideals can be arbitrary subsets (also equal to the whole carrier), it should be explicitly stated in the definition of a spectrum that we deal only with proper ideals.

```
definition let L be Lattice;
  func Spectrum L -> Subset-Family of L equals    :: LATTICEA:def 10
    { I where I is Ideal of L : I is prime proper };
end;
```

In an arbitrary distributive lattice all maximal ideals are prime [13]; if the reverse implication is true, it makes the lattice Boolean, so for Boolean lattices the following equivalence holds (in terms of spectra):

```
theorem :: LATTICEA:34
  for L being distributive bounded Lattice holds
    L is Boolean iff Spectrum L is unordered;
```

As in Boolean algebras the notions of prime filters, maximal filters, and ultrafilters coincide (similarly the notions of ideals but this is obvious due to the duality), it is not very surprising that in the current state of MML the properties of all these subsets were not fully explored in a general setting. Our initial idea was to formalize certain class of lattices with complementation operator which are more general than Boolean ones (including Stone and de Morgan algebras; also Nelson algebras where there are two negations) without repeating actually a part of the proving work in every case. Additionally, we noticed that some parts of the library can be unified even now: as the important example of a Stone lattice is constructed when proving Stone’s representation theorem, why not to simplify things? We discovered that the following representation theorems were already formalized within MML: for Boolean, Heyting, and modular lattices, for finite distributive lattices (by Birkhoff), and essentially there is no version for distributive lattices. We constructed the more general version of Stone mapping which eventually be moved before all aforementioned articles in the process of MML revision [11] and will cover them all as now we have the situation where crucial parts of the proofs were repeated. The path was actually opened

by [8] – our pretty fresh submission to the MML. We proved there e.g. the Prime Ideal Theorem not as Boolean Prime Ideal, but in the more general setting; we have shown also that if the notion of maximal and prime filters coincide in the lattice, it is Boolean.

The situation we meet here is essentially distinct than in the previous section: we don't have explicit complementation operator given in the signature. But where are complements there as we deal with only distributive bounded lattices? How can one construct the complementation? The question is actually answered in the proof; we extensively used newly introduced definition of the set of pseudocomplements (as these elements can be multiple in arbitrarily chosen lattice):

```
definition let L be Lattice; let a be Element of L;
  func PseudoComplements a -> Subset of L equals  :: LATTICEA:def 8
    { x where x is Element of L : a /\ x = Bottom L };
end;
```

If this ideal is maximal, its maximal element plays a role of the unique complement of a in L and as this construction works for arbitrary element of L , it makes it Boolean [8].

7 Conclusions

We presented the uniform algebraic framework developed in the MIZAR Mathematical Library. The mechanism of structures and cluster of adjectives has been tested e.g. during the solution of the Robbins problem: the MIZAR article [9] stems from at least three papers covering different steps of the proof that all Robbins algebras are Boolean: in the first one [15] proofs are done by hand, the second [17] presents proofs as an output straight from EQP/OTTER theorem prover, in the last one [4] automatically derived proofs have been simplified by introducing some auxiliary functions and concepts. There is no guarantee that all theorems are at the most general level and any of the assumptions cannot be weakened – even in original work of Huntington [15] one of the axioms was dependent on the others and the corrected proof was published later [14].

One of the major problems is that all the lemmas or definitions should be stated for the minimal collection of conditions under which they are true. Having to reuse theorems not fulfilling this condition is not easy. This is why MIZAR articles which form a basis of MML – like [27] had to be revised to enable more convenient formalization. MIZAR still lacks the tools which significantly restructure proofs (although there is some initial work available [23, 24]), which effects in low readability of some articles. Implementing fully functional interface between MIZAR, provers like PROVER9, and other proof assistants [19] (which can be a powerful tool like a Sledgehammer in case of Isabelle/HOL [22], as massive work was already done by Urban [26]) could be a quite promising step in the direction of making proofs in MML shorter, more readable, and easier to understand. We hope that further improvement of inheritance mechanisms for algebraic structures as well as type modifiers with more “atomic” attributes offered by MIZAR will allow to make it work.

8 Related and Future Work

The formalization of lattice theory is an example of theory exploration; as it is important in the interpretation of many areas of mathematics, it is also present in other repositories of mathematical knowledge. A bunch of axiom systems, although with no hierarchy in the sense as in MIZAR is available from McCune's page (although he passed away, the page

is still there). Also in Isabelle/HOL [22] the set of corresponding theories is available – “Lattices and orders in Isabelle/HOL” essentially covers LATTICES and LATTICE3 with the exception of Knaster-Tarski theorem which is formalized in separate KNASTER article in MIZAR; lattices are partial orders there, and this approach is one of the two available in MML.

The translation from PROVER9 into MIZAR source is not fully automated as yet; we used e.g. Lisp script written by Urban transforming proof objects into MIZAR formalism. We have to work out by hand the framework which we plan to use (build appropriate environment declaration [20], define proper structures or apply used ones, choose and introduce notations, reserve variables and so on). The automatization may serve well to resolve the type dependencies for a given object [7]. Although our submission [8] is pretty well self-containing and can be considered closed work, we still see some potential in it: some work was a kind of generalization effort done mainly with automatic tools offered with the distribution, however many revisions have to be made [11].

As the MIZAR system doesn’t offer smooth exploration of counterexamples, especially in case of lattices, where the visualization is really important, we will try to implement at least a kind of automatic translation of the output given by MACE (Models and Counterexamples) system. Presently, we implement the suggestion of a referee of our article [8] by constructing the bunch of useful examples of lattices where maximal, prime filters, and ultrafilters will be enumerated. As the MIZAR checker gets stronger [16], we hope to obtain more results automatically.

Acknowledgments Thanks go to Professor Andrzej Trybulec who passed away before this paper was completed, for his continuous support as the Chief for all these years of cooperation, giving the motivation, valuable hints and thoughts (including those of not fully scientific character) and making the MIZAR system moving forward.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Alama, J., Kohlhase, M., Mamane, L., Naumowicz, A., Rudnicki, P., Urban, J.: Licensing the Mizar Mathematical Library. In: Davenport, J.H., Farmer, W.M., et al. (eds.) Proc. of MKM 2011, LNCS, vol. 6824, pp. 149–163. Springer, Berlin, Heidelberg (2011)
2. Balbes, R., Dwinger, P.: Distributive Lattices. University of Missouri Press (1975)
3. Bancerek, G.: Development of the theory of continuous lattices in Mizar. In: Kerber, M., Kohlhase, M. (eds.) Symbolic Computation and Automatic Reasoning. The Calculemus-2000 Symposium, A K Peters, pp. 65–80 (2000)
4. Dahn, B.I.: Robbins algebras are Boolean: A revision of McCune’s computer-generated solution of Robbins problem. *J. Algebra* **208**, 526–532 (1998)
5. Fitelson, B.: Using Mathematica 3.0 to understand the computer proof of the Robbins conjecture. *Math. Ed. Res.* **7**(1) (1998)
6. Geuvers, H., Pollack, R., Wiedijk, F., Zwanenburg, J.: A constructive algebraic hierarchy in Coq. *J. Symb. Comput.* **34**(4), 271–286 (2002)
7. Grabowski, A.: Automated discovery of properties of rough sets. *Fundamenta Informaticae* **128**(1–2), 65–79 (2013)
8. Grabowski, A.: Prime filters and ideals in distributive lattices, MML Id: LATTICEA. *Formalized Mathematics* **21**(3), 213–221 (2013)

9. Grabowski, A.: Robbins algebras vs. Boolean algebras, MML Id: ROBBINS1. *Formalized Mathematics* **9**(4), 681–690 (2001)
10. Grabowski, A., Korniłowicz, A., Naumowicz, A.: Mizar in a nutshell. *J. Formalized Reason.* **3**(2), 153–245 (2010)
11. Grabowski, A., Schwarzweller, Ch.: Revisions as an essential tool to maintain mathematical repositories. In: *Proc. of the 14th symposium on Towards Mechanized Mathematical Assistants, MKM 2007, Hagenberg, Austria, LNCS, vol. 4573, pp. 235–249, Springer (2007)*
12. Grabowski, A., Schwarzweller, Ch.: Towards automatically categorizing mathematical knowledge. In: *Federated Conference on Computer Science and Information Systems – FedCSIS 2012, Wroclaw, Poland, 9–12 September 2012, Proceedings, pp. 63–68 (2012)*
13. Grätzer, G.: *Lattice Theory: Foundation*. Birkhäuser (2011)
14. Huntington, E.V.: Boolean algebra: a correction. *Trans. AMS* **35**(2), 557–558 (1933)
15. Huntington, E.V.: New sets of independent postulates for the algebra of logic, with special reference to Whitehead and Russell’s *Principia Mathematica*. *Trans. AMS* **35**, 274–304 (1933)
16. Korniłowicz, A.: On rewriting rules in Mizar. *J. Autom. Reason.* **50**(2), 203–210 (2013)
17. McCune, W.: Solution of the Robbins problem. *J. Autom. Reason.* **19**, 263–276 (1997)
18. McCune, W., Veroff, R., Fitelson, B., Harris, K., Feist, A., Wos, L.: Short single axioms for Boolean algebra. *J. Autom. Reason.* **29**(1), 1–16 (2002)
19. Naumowicz, A.: Interfacing external CA systems for Grobner bases computation in Mizar proof checking. *Int. J. Comput. Math.* **87**(1), 1–11 (2010)
20. Naumowicz, A., Byliński, C.: Improving Mizar texts with properties and requirements. In: *Asperti, A., Bancerek, G., Trybulec, A. (eds.) Proc. of Third International Conference on Mathematical Knowledge Management, LNCS, vol. 3119, pp. 290–301. Springer (2004)*
21. Naumowicz, A., Korniłowicz, A.: A brief overview of Mizar. In: *Berghofer, S., Nipkow, T., Urban, C., Wenzel, M. (eds.) Proceedings of TPHOLs’09, LNCS, vol. 5674, pp. 67–72. Springer, Berlin, Heidelberg (2009)*
22. Nipkow, T., Wenzel, M., Paulson, L.C.: *Isabelle/HOL: a proof assistant for higher order logic*. Springer, Berlin (2002)
23. Pąk, K.: Improving legibility of natural deduction proofs is not trivial. *Logical Methods Comput. Sci.* **10**(3), 1–30 (2014)
24. Pąk, K.: Methods of lemma extraction in natural deduction proofs. *J. Autom. Reason.* **50**(2), 217–228 (2013)
25. Trybulec, A., Korniłowicz, A., Naumowicz, A., Kuperberg, K.: Formal mathematics for mathematicians. *J. Autom. Reason.* **50**(2), 119–121 (2013)
26. Urban, J., Rudnicki, P., Sutcliffe, G.: ATP and presentation service for Mizar formalizations. *J. Autom. Reason.* **50**(2), 229–241 (2013)
27. Żukowski, S.: Introduction to lattice theory, MML Id: LATTICES. *Formalized Mathematics* **1**(1), 215–222 (1990)