Basic Logic, SMT solvers and finitely generated varieties of GBL-algebras

Peter Jipsen

Chapman University

Extended Abstract

Basic Logic was introduced by Petr Hájek to provide a unified approach to fuzzy logics, and judging by its rapid adoption in the research community, it has enjoyed considerable success in this regard. One of the reasons is that while it is a very general logic, it has elegant semantics with respect to the real unit interval, which allow for practical applications and use of standard software tools. Here we show how to use these semantics to encode propositional Basic Logic into the Satisfiability Modulo Theories (SMT) framework, based on an interpretation of Lukasiewicz logic, Gödel logic and product logic into SMT [4]. Ultimately these ideas go back to Mundici's result [13] that satisfiablity for Lukasiewicz logic is NP-complete, and Hähnle's translation from Lukasiewicz logic to integer linear programming [8, 9]. In the current setting the translation to SMT is very simple, and since there are several efficient SMT-solvers available, this is probably one of the most effective and flexible ways of implementing a decision procedure for propositional basic logic.

Basic logic algebras (or BL-algebras for short) are bounded integral commutative residuated lattices that satisfy divisibility and prelinearity. The latter property implies that subdirectly irreducible BL-algebras are linearly ordered. Generalized BL-algebras (or GBL-algebras) are just divisible residuated lattices, but still retain many of the properties of BL-algebras. For example they have distributive lattice reducts, the fusion operation distributes over the meet operation, and in the n-potent case they are integral and commutative [10]. The subdirectly irreducible GBL-algebras are no longer linearly ordered, but in the finite case they have a well-understood structure theory based on the so-called poset product construction [10, 11].

In [3] it is shown that the category of finite BL-algebras is dually equivalent to a category of finite rooted forests, labeled by positive integers. This result is extended here to finite GBL-algebras, with rooted forests replaced by posets. Since subdirectly irreducible GBL-algebras correspond to labeled rooted posets under this duality, it becomes feasible to calculate the HS-poset of subdirectly irreducible GBL-algebras up to a fixed size, giving a view of the lattice of finitely generated GBL varieties. The duality also implies that GBL-algebras with n join-irreducibles are in one-one correspondence with preorders on n-elements, which simplifies the enumeration of finite GBL-algebras.

The Kripke models of the modal logic S4 are determined by preorders, so the duality can be extended to the category of finite closure algebras (the algebraic models of S4). Since the larger category of all modal algebras has a well-developed duality theory using descriptive frames, these results may be useful for obtaining a duality for all n-potent commutative GBL-algebras.

1 Deciding propositional Basic Logic with SMT-solvers

Boolean satisfiability solvers (SAT-solvers) are programs that take a classical propositional formula (often restricted to conjunctive normal form) as input and search for an assignment of

truth values to the variables such that the formula is true, or report that no such assignment exists. Satisfiability modulo theories solvers (SMT-solvers) are generalizations of SAT-solvers that take as input a formula of typed first-order logic with equality (perhaps restricted to be quantifier-free), and determine if there is an assignment into a specific model (such as $\mathbb{R}$ or $\mathbb{Z}$) under which the formula is true. The "modulo theories" in the name of SMT-solvers refers to the theory of the model in which satisfiability is tested. E.g. a formula such as $0 < x + y < 10 \;\&\; x + x - y - y = 1$ would be satisfiable in $\mathbb{R}$ but not in $\mathbb{Z}$.

Applying SMT-solvers to decide propositional formulas in Lukasiewicz logic or Gödel logic is quite straight forward, as shown in [4]. We take an algebraic view, and implement decision procedures for prelinear Heyting algebras, abelian lattice-ordered groups, MV-algebras and BL-algebras. Recall that $(A, \wedge, \vee, \to, 1, 0)$ is a *Heyting algebra* if $(A, \wedge, \vee, 1, 0)$ is a bounded distributive lattice and $x \wedge y \leq z \iff y \leq x \to z$ for all $x, y, z \in A$. It is *prelinear* if the identity $(x \to y) \vee (y \to x) = 1$ holds, in which case the subdirectly irreducible models are linearly ordered. Prelinear Heyting algebras are the algebraic semantics of Gödel logic, and a propositional formula φ of Gödel logic is a tautology precisely when the equation $\varphi = 1$ is an identity of prelinear Heyting algebras. The same correspondence holds for Lukasiewicz logic and MV-algebras.

An *abelian lattice-ordered group* is of the form $(A, \wedge, \vee, +, -, 0)$ where $(A, \wedge, \vee)$ is a (necessarily distributive) lattice, $(A, +, -, 0)$ is an abelian group and $+$ is order-preserving in both arguments. The variety of abelian lattice-ordered groups is generated by the model $(\mathbb{Z}, \min, \max, +, -, 0)$ as well as by $(\mathbb{R}, \min, \max, +, -, 0)$.

A MV-algebra is given by $(A, \wedge, \vee, \cdot, \neg, 1, 0)$ where $(A, \wedge, \vee)$ is a lattice, $(A, \cdot, 1)$ is a commutative monoid, $\cdot$ is order-preserving in both arguments, $\neg\neg x = x$, $0 = \neg 1$ and $x \cdot y \leq z \iff y \leq \neg x \vee z$. This definition of MV-algebras emphasizes that they are residuated lattices, though they are often defined equationally using the the dual operation $x \oplus y = \neg(\neg x \cdot \neg y)$.

The input for SMT-solvers is usually written in a standard language called SMT-LIB2. The input for deciding MV-identities is given below and can be used with a variety of solvers, such as Z3, SMTinterpol, opensmt, etc. For the algebraic operations we use standard LaTeX names for the symbols. Any semicolon and all following characters up to the end of each line are optional comments. The SMT-LIB2 language has a syntax similar to LISP, so expressions are lists of tokens separated by spaces and enclosed in parentheses. The first token is usually a command or function name, and the remaining tokens are inputs for the function. E.g. (ite (< x y) x y) is the if-then-else function applied to a boolean test and producing (in this case) the smaller of the two values as output. The full syntax is defined at `www.smtlib.org`.

```
; Deciding MV (in)equations in SMT
(set-logic QF_LRA)
(define-fun wedge ((x Real) (y Real)) Real (ite (> x y) y x))   ; x ∧ y
(define-fun vee ((x Real) (y Real)) Real (ite (> x y) x y))     ; x ∨ y
(define-fun oplus ((x Real) (y Real)) Real (wedge (+ x y) 1))   ; x ⊕ y
(define-fun cdot ((x Real) (y Real)) Real (vee (- (+ x y) 1) 0)) ; x · y
(define-fun neg ((x Real)) Real (- 1 x))                        ; ¬x = 1 − x
(define-fun to ((x Real) (y Real)) Real (wedge 1 (- (+ 1 y) x))) ; x → y
(define-fun leftrightarrow ((x Real) (y Real)) Real (wedge (to x y) (to y x)))
(declare-const x Real) (assert (<= 0 x)) (assert (<= x 1))
(declare-const y Real) (assert (<= 0 y)) (assert (<= y 1))
(assert (< (leftrightarrow (vee x y) (to (to x y) y)) 1)) ; (in)equation to be tested
; test if ((x ∨ y) ↔ ((x → y) → y)) < 1 is satisfiable
(check-sat)
```

We briefly describe the SMT-LIB2 code listed above. The first line of the code is a descriptive comment and the second line selects *quantifier-free linear real arithmetic* (QF_LRA) as the theory used by the SMT-solver. The next 7 lines define the MV-operations on the unit interval by $x \wedge y = \min(x, y)$, $x \vee y = \max(x, y)$, $x \oplus y = (x + y) \wedge 1$, $x \cdot y = (x + y - 1) \vee 0$, $\neg x = 1 - x$, $x \to y = (1 - x + y) \wedge 1$, and $x \leftrightarrow y = (x \to y) \wedge (y \to x)$. The lines that start with "declare-const" define two real variables x, y and restrict their values to the interval $[0, 1]$. The third last line asserts the formula that is to be checked, followed by a comment showing the formula in standard notation. The last line asks the SMT-solver to check if the formula $\varphi < 1$ is satisfiable, in which case the formula φ is not a tautology. To test if an equation $s = t$ is an identity, one would check the formula $s \leftrightarrow t$, adding more "declare-const" lines if the formula contains more than two variables.

Checking equations in prelinear Heyting algebras is a matter of deleting the definitions for oplus and cdot, and replacing the next two lines by:

(define-fun neg ((x Real)) Real (ite (= x 0) 1 0)); $\neg x$

(define-fun to ((x Real) (y Real)) Real (ite (<= x y) 1 y)); $x \to y$

An abelian ℓ-group inequation $s \leq t$ is coded directly using the operations $+, -, 0$ of the logic QF_LRA, and the SMT-solver is asked to check if $s > t$ is satisfiable. The assertions that restrict variables to the unit interval have to be removed in this case. For equations $s = t$ one checks if $s > t$ or $t > s$ is satisfiable, i.e., (assert (or (< s t) (< t s))) in SMT-LIB2 syntax. A similar approach can be used to check (in)equations in the negative cone of $\mathbb{R}$ by defining $x \cdot y = (x + y) \wedge 0$. By using a translation with an extra variable z as in [7] one can also check (in)equations in the negative cone of $\mathbb{R}$ with a new bottom element, which is equivalent to checking propositional formulas in *product logic*. This is an improvement over the suggestion in [4] to use full real arithmetic for product logic, since implementations of *linear* real arithmetic in SMT-solvers are currently more efficient.

The decision procedure for propositional basic logic with an SMT-solver uses the following result of [2] (see also [1]).

Theorem 1. *Let $A_n = \bigoplus_{i=0}^{n}[0, 1]$ be the ordinal sum of $n + 1$ unit-interval MV-algebras, and let $\mathcal{V}_n$ be the variety generated by all n-generated BL-algebras. Then $\mathcal{V}_n = HSP(A_n)$, hence an n-variable BL-identity holds in A_n if and only if it holds in all BL-algebras.*

By constructing the algebra A_n of the above result within the SMT language, one obtains an effective means of checking n-variable BL-identities. The universe for A_n is taken to be the interval $[0, n + 1]$. The definition of fusion and implication are

$$x \cdot y = \begin{cases} \max(x + y - 1 - \lfloor y \rfloor, \lfloor x \rfloor) & \text{if } \lfloor x \rfloor = \lfloor y \rfloor \\ \min(x, y) & \text{otherwise} \end{cases}$$

$$x \to y = \begin{cases} n + 1 & \text{if } x \leq y \\ y & \text{if } \lfloor y \rfloor < \lfloor x \rfloor \\ \min(1 + y - x + \lfloor x \rfloor, 1 + \lfloor y \rfloor) & \text{otherwise} \end{cases}$$

A straightforward SMT-LIB2 implementation of these operations uses $n+1$ cases, so the formula does become long even for small values of n. Below we give the implementations for $n = 1$ and $n = 2$, which can be used to check 1-variable and 2-variable BL-identities.

$n = 1$:

(define-fun cdot ((x Real) (y Real)) Real (ite (and (< x 1) (< y 1)) (vee (- (+ x y) 1) 0) (ite (and (<= 1 x) (<= 1 y)) (vee (- (+ x y) 2) 1) (wedge x y))))

(define-fun to ((x Real) (y Real)) Real (ite (<= x y) 2 (ite (and (<= 1 x) (< y 1)) y (wedge 1 (- (+ 1 y) x)))))

$n = 2$:

(define-fun cdot ((x Real) (y Real)) Real (ite (and (< x 1) (< y 1)) (vee (- (+ x y) 1) 0) (ite (and (<= 1 x) (< x 2) (<= 1 y) (< y 2)) (vee (- (+ x y) 2) 1) (ite (and (<= 2 x) (<= 2 y)) (vee (- (+ x y) 3) 2) (wedge x y)))))

(define-fun to ((x Real) (y Real)) Real (ite (<= x y) 3 (ite (and (< x 1) (< y 1)) (+ (- 1 x) y) (ite (and (<= 1 x) (< x 2) (<= 1 y) (< y 2)) (+ (- 2 x) y) (ite (and (<= 2 x) (<= 2 y)) (+ (- 3 x) y) y)))))

A Python program has been written that takes as input a BL-(in)equation in LaTeX using standard infix notation and generates the SMT-LIB2 file for this identity. Calling a SMT-solver with this file determines if the (in)equation is valid in all BL-algebras, or produces a counterexample in a suitable ordinal sum of standard MV-algebras.

2 Finitely generated varieties of BL-algebras

It is well known that subdirectly irreducible BL-algebras are linearly ordered (this is more generally true for commutative prelinear residuated lattices). In the finite case this means that they are simply n-element chains. Every finite subdirectly irreducible BL-algebra A is an ordinal sum of finite MV-chains, hence the structure of A is completely determined by the idempotent elements in the chain. The top and bottom of the chain are always idempotent, so if A has n elements then there are 2^{n-2} choices for the idempotent elements, and therefore 2^{n-2} nonisomorphic subdirectly irreducible BL-algebras. We will denote each of these algebras by $B_{a_1a_2\dots a_m}$ where $a_1, a_2, \dots, a_m$ is a list of positive integers, m is the number of join-irreducible idempotent elements and a_i is one greater than the number of non-idempotent elements between the ith idempotent element and the $(i+1)$th idempotent element in the chain, counting from the bottom. Note that with this definition we have $B_{a_1a_2\dots a_m} = MV_{a_1} \oplus MV_{a_2} \oplus \cdots \oplus MV_{a_m}$ (where MV_n is the $n+1$-element MV-chain and $\oplus$ is the adjoined ordinal sum). Hence $B_{11\dots1}$ (with n 1's in the subscript) is the n-element linear Heyting algebra, $B_n = MV_n$, B_1 is the 2-element Boolean algebra, and we use B_0 to denote the trivial algebra. In particular, any finite BL-chain is determined by a unique finite sequence of positive integers. The length of the chain is always $a_1 + a_2 + \cdots + a_m + 1$.

A variety $\mathcal{V}$ of algebras of finite similarity type is said to be finitely generated if $\mathcal{V} = HSP(\mathcal{K})$ for some finite set $\mathcal{K}$ of finite algebras. If, in addition, $\mathcal{V}$ is congruence distributive then by Jónsson's lemma the subdirectly irreducible members of $\mathcal{V}$ are all contained in $HS(\mathcal{K})$, hence there are only finitely many such members. In particular, for two finite subdirectly irreducible algebras A, B of the same type, $HSP(A) \subseteq HSP(B)$ if and only if $A \in HS(B)$, and we write $A \leq_{HS} B$ in case the latter relation holds. Since $HSHS = HS$, this relation is a partial order on isomorphism classes of finite subdirectly irreducible algebras. Any variety is determined by it subdirectly irreducible members, hence the lattice of finitely generated subvarieties is isomorphic to the lattice of finite downsets of this partial order.

Lemma 2. *Let A, B be finite subdirectly irreducible GBL-algebras. Every subalgebra of B is subdirectly irreducible, and if A is a homomorphic image of B then A is a subalgebra of B. Hence $A \in HS(B)$ if and only if $A \in S(B)$.*

The preceding result simplifies calculating the $\leq_{HS}$ partial order relation between subdirectly irreducible GBL-algebras. Komori gave a complete description of the lattice of subvarieties of MV-algebras, showing that it is countable and that the $\leq_{HS}$ poset of finite subdi-

rectly irreducible MV-algebras is isomorphic to the divisibility lattice $\mathbb{D} = (\mathbb{N} \setminus \{0\}, |)$, with $MV_m \leq_{HS} MV_n$ if and only if $m|n$. Here we describe the $\leq_{HS}$ poset for finite s.i. BL-algebras. As observed previously, these algebras are chains determined by finite sequences of positive integers.

Theorem 3. *The $\leq_{HS}$ poset of finite s.i. BL-algebras is isomorphic to $\mathbb{D}^* = \bigcup_{n=0}^{\infty} \mathbb{D}^n$ with the order on $\mathbb{D}^*$ extending the pointwise divisibility order on each component by $(a_1, \ldots, a_m) \leq (b_1, \ldots, b_n)$ if and only if there exists an injection $f : \{1, \ldots, m\} \to \{1, \ldots, n\}$ such that $f(1) = 1$ and $a_i | b_{f(i)}$ for all $i \in \{1, \ldots, m\}$. The order relation $(a_1, \ldots, a_m) \leq (b_1, \ldots, b_n)$ is a covering relation if and only if either*

- *$m = n$ and $(b_1, \ldots, b_n) = (a_1, \ldots, a_{i-1}, pa_i, a_{i+1}, \ldots, a_n)$ for some prime p and a unique $i \leq n$, or*
- *$m + 1 = n$ and $(b_1, \ldots, b_n) = (a_1, \ldots, a_{i-1}, 1, a_i, \ldots, a_m)$ for some $i \in \{2, \ldots, n\}$.*

A schematic diagram of the HS-poset is shown in the Appendix.

3 Mapping finite GBL-algebras to finite closure algebras

Recall that finite GBL-algebras are poset products of finite *Wajsberg chains* (= 0-free reducts of MV-chains) [10]. The poset product does not dependent on the divisibility law, hence we first consider the more general setting of poset products of bounded integral simple residuated chains.

Recall that an algebra is *simple* if it only has two congruence relations. An element c in a monoid is *central* if it commutes with every element of the monoid, and it is *idempotent* if $cc = c$. For any negative central idempotent c in a residuated lattice, the principal filter $\uparrow c$ is a normal filter and hence determines a congruence of the residuated lattice. Let $\mathcal{C}$ be the class of all bounded simple residuated chains, where we denote the bounds by $0, 1$ and assume that 1 is the monoid identity. By simplicity $0, 1$ are the only central idempotents of each member of $\mathcal{C}$.

Let $\mathbf{P}$ be a poset. The (dual) poset product of a family $\{C_i : i \in P\} \subseteq \mathcal{C}$ is defined on a subset of the cartesian product by

$$\prod_{\mathbf{P}} C_i = \{f \in \prod_{i \in P} C_i : \forall i > j \in P \, (f(i) = 0 \text{ or } f(j) = 1)\}.$$

The operations $\wedge, \vee, \cdot$ are defined pointwise and the bounds are the constant functions $\mathbf{0}, \mathbf{1}$. The residuals are given by

$$(f \backslash g)(i) = \begin{cases} f(i) \backslash g(i) & \text{if } f(j) \leq g(j) \text{ for all } j < i \\ 0 & \text{otherwise} \end{cases}$$

$$(g / f)(i) = \begin{cases} g(i) / f(i) & \text{if } f(j) \leq g(j) \text{ for all } j < i \\ 0 & \text{otherwise.} \end{cases}$$

It is shown in [10] that a (dual) poset product (called poset sum in that paper) of bounded integral residuated lattices is again a bounded residuated lattice, and if the factors are divisible, so is the poset product. Hence a poset product of GBL-algebras is also a GBL-algebra. Since we are assuming that the factors are simple, the only central idempotents in the resulting poset product will be the functions with range $0, 1$. Moreover, every finite GBL-algebra is (isomorphic to) a poset product of Wajsberg chains [10]. These linearly ordered factors are completely

determined by their cardinality, so a finite GBL-algebra is determined by a finite preorder $\sqsubseteq$, where the blocks of the equivalence relation $\sqsubseteq \cap \sqsupseteq$ contain the elements of each Wajsberg chain. Homomorphisms between finite GBL-algebras correspond to certain p-morphisms between the preorders, hence the HS-poset of finite subdirectly irreducible GBL-algebras can be characterized in this way.

A preorder also determines a Kripke model of the modal logic S4, so every finite GBL-algebra can be mapped to a finite closure algebra. Moreover, the homomorphisms between finite GBL-algebras are homomorphisms between the corresponding closure algebras (however the converse does not hold).

Theorem 4. *There is a functor, faithful on objects, from the category of finite GBL-algebras to the category of finite closure algebras and hence also to its dual category of preorders with p-morphisms.*

Current research is aiming to extend this functor to larger categories, such as the category of complete perfect n-potent GBL-algebras.

References

[1] S. Aguzzoli and S. Bova, *The free n-generated BL-algebra*, Annals of Pure and Applied Logic 161 (2010) 1144–1170

[2] P. Agliano and F. Montagna, *Varieties of BL-algebras I: general properties*, Journal of Pure and Applied Algebra, 181 (2003), 105–129.

[3] S. Aguzzoli, S. Bova and V. Marra, *Applications of Finite Duality to Locally Finite Varieties of BL-Algebras*, in Proceedings of the 2009 International Symposium on Logical Foundations of Computer Science, S. Artemov and A. Nerode (Eds.), LNCS 5407, 1–15, 2009.

[4] C. Ansótegui, M. Bofill, F. Manyà and M. Villaret, *Building automated theorem provers for infinitely-valued logics with satisfyability modulo theory solvers*, in Proceedings, IEEE 42nd International Symposium on Multiple-Valued Logic. ISMVL 2012, 25–30.

[5] A. Di Nola and A. Lettieri, *Finite BL-algebras*, Discrete Mathematics **269** (2003), 93–112.

[6] N. Galatos, P. Jipsen, T. Kowalski and H. Ono, "Residuated lattices: an algebraic glimpse at substructural logics", Studies in Logic and the Foundations of Mathematics, Vol. 151, Elsevier, 2007.

[7] N. Galatos and C. Tsinakis, *Generalized MV-algebras*, Journal of Algebra, 283 (2005), no. 1, 254–291.

[8] R. Hähnle, *Many-valued logic and mixed integer programming*, Annals of Mathematics and Articial Intelligence, 12(3,4) (1994), 231–264.

[9] R. Hähnle, *Proof theory of many-valued logic linear optimization logic design*, Soft Computing, 1(3) (1997), 107–119.

[10] P. Jipsen and F. Montagna, *The Blok-Ferreirim theorem for normal GBL-algebras and its applications*, Algebra Universalis **60** (2009), 381–404.

[11] P. Jipsen and F. Montagna, *Embedding theorems for classes of GBL-algebras*, Journal of Pure and Applied Algebra, 214 (2010), 1559–1575

[12] W. McCune, Prover9 and Mace4, `www.cs.unm.edu/~mccune/Prover9`, 2005-2010.

[13] D. Mundici, *Satisfiability in many-valued sentential logic is NP-complete*, Theoretical Computer Science, 52 (1987), 145–153.

[14] P. Wojciechowski, *Embeddings of totally ordered MV-algebras of bounded cardinality*, Fundamenta Mathathematicae 203 (2009), 57–63.

4 Appendix

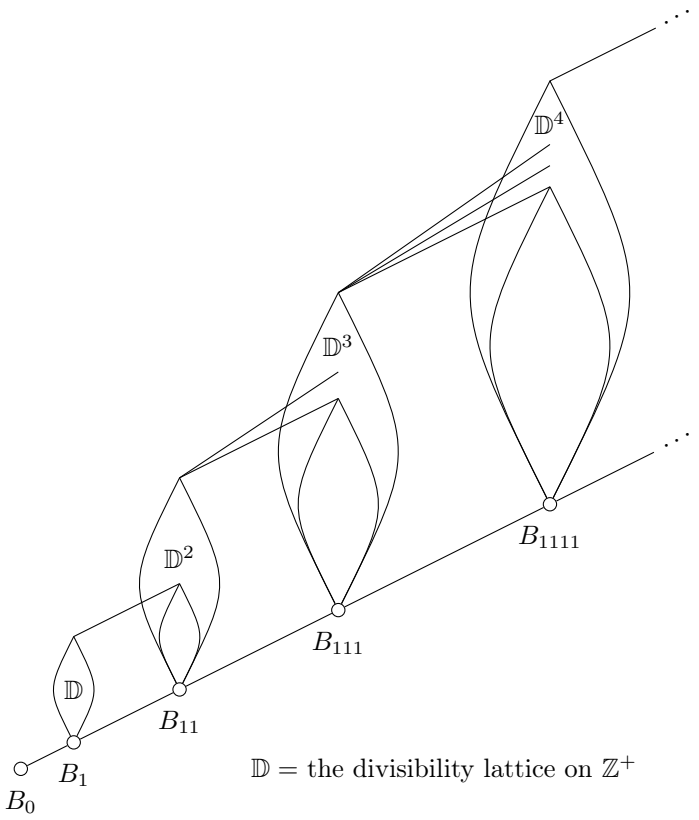

Figure 1: HS-poset of subdirectly irreducible BL-algebras