



HAL
open science

Solving equations in pure double Boolean algebras

Philippe Balbiani

► **To cite this version:**

Philippe Balbiani. Solving equations in pure double Boolean algebras. 30th International Workshop on Unification (UNIF 2016), Jun 2016, Porto, Portugal. pp.13-18. hal-03176437

HAL Id: hal-03176437

<https://hal.science/hal-03176437>

Submitted on 24 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Solving equations in pure double Boolean algebras

Philippe Balbiani

Institut de recherche en informatique de Toulouse

1 Introduction

Herrmann *et al.* [7] have generalized lattices of concepts to algebras of semiconcepts. Operations between semiconcepts give rise to pure double Boolean algebras [9, 10]. Such algebraic structures can be seen as the union of two Boolean algebras, the intersection of which being a lattice of concepts. In this paper, we study word problems and unification problems in several classes of pure double Boolean algebras.

2 Semiconcepts

Formal contexts are structures of the form $\mathbb{K} = (G, M, \Delta)$ where G is a nonempty set (with typical member noted g), M is a nonempty set (with typical member noted m) and Δ is a binary relation between G and M . The elements of G are called *objects*, the elements of M are called *attributes* and the intended meaning of $g\Delta m$ is: *object g possesses attribute m* . For all $X \subseteq G$ and for all $Y \subseteq M$, let $X^\triangleright = \{m \in M: \text{for all } g \in G, \text{ if } g \in X \text{ then } g\Delta m\}$ and $Y^\triangleleft = \{g \in G: \text{for all } m \in M, \text{ if } m \in Y \text{ then } g\Delta m\}$. The claims in the next lemma follow directly from the definition of the maps $\triangleright: X \mapsto X^\triangleright$ and $\triangleleft: Y \mapsto Y^\triangleleft$. See [3, 6] for details.

Lemma 1. *Let $X_1, X_2, X \subseteq G$ and $Y_1, Y_2, Y \subseteq M$.*

1. *The following conditions are equivalent: (a) $X \subseteq Y^\triangleleft$, (b) $X^\triangleright \supseteq Y$.*
2. *If $X_1 \subseteq X_2$ then $X_1^\triangleright \supseteq X_2^\triangleright$ and if $Y_1 \supseteq Y_2$ then $Y_1^\triangleleft \subseteq Y_2^\triangleleft$.*
3. *$X \subseteq X^{\triangleright\triangleleft}$ and $Y^{\triangleleft\triangleright} \supseteq Y$.*
4. *If there exists $Y' \subseteq M$ such that $X = Y'^\triangleleft$ then $X = X^{\triangleright\triangleleft}$ and if there exists $X' \subseteq G$ such that $X'^\triangleright = Y$ then $Y^{\triangleleft\triangleright} = Y$.*

Given $X \subseteq G$ and $Y \subseteq M$, the pairs (X, X^\triangleright) and (Y^\triangleleft, Y) are called *semiconcepts of \mathbb{K}* . More precisely, pairs of the form (X, X^\triangleright) are called *left semiconcepts of \mathbb{K}* and pairs of the form (Y^\triangleleft, Y) are called *right semiconcepts of \mathbb{K}* . Remark that (\emptyset, M) and (G, \emptyset) are semiconcepts of \mathbb{K} . Let $\underline{\mathcal{H}}(\mathbb{K}) = (\mathcal{H}(\mathbb{K}), 0_l, 0_r, 1_l, 1_r, \sim_l, \sim_r, \sqcup_l, \sqcup_r, \sqcap_l, \sqcap_r)$ be the algebraic structure of type $(0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2)$ defined by

- $\mathcal{H}(\mathbb{K})$ is the set of all semiconcepts of \mathbb{K} ,
- $0_l = (\emptyset, M)$,
- $0_r = (M^\triangleleft, M)$,
- $1_l = (G, G^\triangleright)$,
- $1_r = (G, \emptyset)$,
- $\sim_l(X, Y) = (G \setminus X, (G \setminus X)^\triangleright)$,

- $\sim_r (X, Y) = ((M \setminus Y)^{\triangleleft}, M \setminus Y)$,
- $(X_1, Y_1) \sqcup_l (X_2, Y_2) = (X_1 \cup X_2, (X_1 \cup X_2)^{\triangleright})$,
- $(X_1, Y_1) \sqcup_r (X_2, Y_2) = ((Y_1 \cap Y_2)^{\triangleleft}, Y_1 \cap Y_2)$,
- $(X_1, Y_1) \sqcap_l (X_2, Y_2) = (X_1 \cap X_2, (X_1 \cap X_2)^{\triangleright})$,
- $(X_1, Y_1) \sqcap_r (X_2, Y_2) = ((Y_1 \cup Y_2)^{\triangleleft}, Y_1 \cup Y_2)$.

Remark that if G, M are finite then $\mathcal{H}(\mathbb{K})$ is finite too and moreover, $|\mathcal{H}(\mathbb{K})| \leq 2^{|G|} + 2^{|M|}$. The set $\mathcal{H}(\mathbb{K})$ can be ordered by the binary relation \sqsubseteq defined by

- $(X_1, Y_1) \sqsubseteq (X_2, Y_2)$ iff $X_1 \subseteq X_2$ and $Y_1 \supseteq Y_2$.

Before formally introducing pure double Boolean algebras, we prove lemmas which will put the above definitions into perspective.

Lemma 2. *Let $(X_1, Y_1), (X_2, Y_2) \in \mathcal{H}(\mathbb{K})$.*

1. *The following conditions are equivalent: (a) $(X_1, Y_1) \sqsubseteq (X_2, Y_2)$, (b) $(X_1, Y_1) \sqcap_l (X_2, Y_2) = (X_1, Y_1) \sqcap_l (X_1, Y_1)$ and $(X_1, Y_1) \sqcup_r (X_2, Y_2) = (X_2, Y_2) \sqcup_r (X_2, Y_2)$.*
2. *If (X_1, Y_1) is a left semiconcept then $(X_1, Y_1) \sqsubseteq (X_2, Y_2)$ iff $(X_1, Y_1) \sqcap_l (X_2, Y_2) = (X_1, Y_1)$.*
3. *If (X_2, Y_2) is a right semiconcept then $(X_1, Y_1) \sqsubseteq (X_2, Y_2)$ iff $(X_1, Y_1) \sqcup_r (X_2, Y_2) = (X_2, Y_2)$.*

Lemma 3. *The binary relation \sqsubseteq is reflexive, antisymmetric and transitive on $\mathcal{H}(\mathbb{K})$.*

We shall say that an object g is *sparse* if $\Delta(g) \neq M$ and an attribute m is *sparse* if $\Delta^{-1}(m) \neq G$. \mathbb{K} is said to be *sparse* if for all $g \in G$, g is sparse and for all $m \in M$, m is sparse. We shall say that a couple (g, g') of objects is a *cover* if $\Delta(g) \cup \Delta(g') = M$ and a couple (m, m') of attributes is a *cover* if $\Delta^{-1}(m) \cup \Delta^{-1}(m') = G$. A sparse \mathbb{K} is said to be *covered* if for all $g, g', g'' \in G$, if (g', g'') is a cover then (g, g') is a cover or (g, g'') is a cover and for all $m, m', m'' \in M$, if (m', m'') is a cover then (m, m') is a cover or (m, m'') is a cover.

3 Pure double Boolean algebras

Let $\mathcal{A} = (A, 0_l, 0_r, 1_l, 1_r, \sim_l, \sim_r, \sqcup_l, \sqcup_r, \sqcap_l, \sqcap_r)$ be an algebraic structure of type $(0, 0, 1, 1, 2, 2, 2, 2)$. For all $a \in A$, let $\star_l a = \sim_r \sim_l a$ and $\star_r a = \sim_l \sim_r a$. \mathcal{A} is said to be *concrete* iff there exists a formal context $\mathbb{K} = (G, M, \Delta)$ and an injective homomorphism from \mathcal{A} to $\underline{\mathcal{H}}(\mathbb{K})$. We shall say that \mathcal{A} is a *pure double Boolean algebra* if for all $a, b, c \in A$, it satisfies the conditions 1–13, 16–28, 31 and 32 in Fig. 1. Now, we can relate pure double Boolean algebras and concrete structures.

Proposition 1 ([7]). *The following conditions are equivalent:*

1. *\mathcal{A} is concrete.*
2. *\mathcal{A} is a pure double Boolean algebra.*

\mathcal{A} is said to be *s-concrete* iff there exists a sparse formal context $\mathbb{K} = (G, M, \Delta)$ and an injective homomorphism from \mathcal{A} to $\underline{\mathcal{H}}(\mathbb{K})$. We shall say that a pure double Boolean algebra \mathcal{A} is *sparse* if it satisfies the conditions 14 and 29 in Fig. 1.

Proposition 2. *The following conditions are equivalent:*

1. \mathcal{A} is *s-concrete*.
2. \mathcal{A} is a *sparse pure double Boolean algebra*.

Proof. Simple variant of the proof of Proposition 1.

\mathcal{A} is said to be *c-concrete* iff there exists a covered sparse formal context $\mathbb{K} = (G, M, \Delta)$ and an injective homomorphism from \mathcal{A} to $\underline{\mathcal{H}}(\mathbb{K})$. We shall say that a sparse pure double Boolean algebra \mathcal{A} is *covered* if it satisfies the conditions 15 and 30 in Fig. 1.

Proposition 3. *The following conditions are equivalent:*

1. \mathcal{A} is *c-concrete*.
2. \mathcal{A} is a *covered sparse pure double Boolean algebra*.

Proof. Simple variant of the proof of Proposition 1.

- | | |
|---|--|
| <ol style="list-style-type: none"> 1. $a \sqcap_l (b \sqcap_l c) = (a \sqcap_l b) \sqcap_l c$, 2. $a \sqcap_l b = b \sqcap_l a$, 3. $\sim_l (a \sqcap_l a) = \sim_l a$, 4. $a \sqcap_l (b \sqcap_l b) = a \sqcap_l b$, 5. $a \sqcap_l (b \sqcup_l c) = (a \sqcap_l b) \sqcup_l (a \sqcap_l c)$, 6. $a \sqcap_l (a \sqcup_l b) = a \sqcap_l a$, 7. $a \sqcap_l (a \sqcup_r b) = a \sqcap_l a$, 8. $\sim_l (\sim_l a \sqcap_l \sim_l b) = a \sqcup_l b$, 9. $\sim_l 0_l = 1_l$, 10. $\sim_l 1_r = 0_l$, 11. $1_r \sqcap_l 1_r = 1_l$, 12. $a \sqcap_l \sim_l a = 0_l$, 13. $\sim_l \sim_l (a \sqcap_l b) = a \sqcap_l b$, 14. $0_l = 0_r$, 15. $\star_l \star_l a \sqcap_l \star_l a = \star_l a \sqcap_l 1_l$, | <ol style="list-style-type: none"> 16. $a \sqcup_r (b \sqcup_r c) = (a \sqcup_r b) \sqcup_r c$, 17. $a \sqcup_r b = b \sqcup_r a$, 18. $\sim_r (a \sqcup_r a) = \sim_r a$, 19. $a \sqcup_r (b \sqcup_r b) = a \sqcup_r b$, 20. $a \sqcup_r (b \sqcap_r c) = (a \sqcup_r b) \sqcap_r (a \sqcup_r c)$, 21. $a \sqcup_r (a \sqcap_r b) = a \sqcup_r a$, 22. $a \sqcup_r (a \sqcap_l b) = a \sqcup_r a$, 23. $\sim_r (\sim_r a \sqcup_r \sim_r b) = a \sqcap_r b$, 24. $\sim_r 1_r = 0_r$, 25. $\sim_r 0_l = 1_r$, 26. $0_l \sqcup_r 0_l = 0_r$, 27. $a \sqcup_r \sim_r a = 1_r$, 28. $\sim_r \sim_r (a \sqcup_r b) = a \sqcup_r b$, 29. $1_r = 1_l$, 30. $\star_r \star_r a \sqcup_r \star_r a = \star_r a \sqcup_r 0_r$, |
|---|--|
31. $(a \sqcap_l a) \sqcup_r (a \sqcap_l a) = (a \sqcup_r a) \sqcap_l (a \sqcup_r a)$,
 32. $a \sqcap_l a = a$ or $a \sqcup_r a = a$.

Fig. 1.

4 A first-order signature

Let Ω be the first-order signature consisting of the following function symbols together with their arities: \perp_l (0), \perp_r (0), \top_l (0), \top_r (0), \neg_l (1), \neg_r (1), \vee_l (2), \vee_r (2), \wedge_l (2) and \wedge_r (2). Let VAR be a countable set of variables (with typical members noted x, y , etc). The set $\mathcal{T}(\Omega, VAR)$ of all *terms over Ω and VAR* (with typical members noted s, t , etc) is inductively defined as usual. We write $s(x_1, \dots, x_n)$ to denote a term whose

variables form a subset of $\{x_1, \dots, x_n\}$. The result of the replacement of x_1, \dots, x_n in their places in s with terms t_1, \dots, t_n will be noted $s(t_1, \dots, t_n)$. A *substitution* is a function σ assigning to each variable x a term $\sigma(x)$. We shall say that a substitution σ is ground if for all variables x , $\sigma(x)$ is a variable-free term. For all terms $s(x_1, \dots, x_n)$ let $\sigma(s)$ be $s(\sigma(x_1), \dots, \sigma(x_n))$. The *composition* $\sigma \circ \tau$ of the substitutions σ and τ assigns to each variable x the term $\tau(\sigma(x))$.

5 Word problems

Let \mathcal{C} be a class of pure double Boolean algebras. Now, for the *word problem* in \mathcal{C} : given terms s, t , decide whether $\mathcal{C} \models s = t$.

- Proposition 4.** 1. *The word problem in the class of all pure double Boolean algebras is PSPACE-complete.*
 2. *The word problem in the class of all sparse pure double Boolean algebras is PSPACE-complete.*
 3. *The word problem in the class of all covered sparse pure double Boolean algebras is NP-complete.*

Proof. (1) See the proofs of Propositions 45 and 51 in [2].

(2) and (3) Simple variants of the proof of Propositions 45 and 51 in [2].

6 Unification problems

Let \mathcal{C} be a class of pure double Boolean algebras. Now, for the *unification problem* in \mathcal{C} : given a finite set $\Sigma = \{(s_1, t_1), \dots, (s_n, t_n)\}$ of pairs of terms, decide whether there exists a substitution σ such that $\mathcal{C} \models \sigma(s_1) = \sigma(t_1), \dots, \mathcal{C} \models \sigma(s_n) = \sigma(t_n)$. In that case, the substitution σ is called *unifier* of Σ . Remark that if a finite set of pairs of terms possesses a unifier then it possesses a ground unifier. This follows from the fact that for all unifiers σ of a finite set Σ of pairs of terms and for all ground substitutions τ , $\sigma \circ \tau$ is a ground unifier of Σ . There are two main questions about the unification problem. Firstly, there is the question of its computability.

- Proposition 5.** 1. *The unification problem in the class of all sparse pure double Boolean algebras is NP-complete.*
 2. *The unification problem in the class of all covered sparse pure double Boolean algebras is NP-complete.*

Proof. Firstly, remark that, in any class of sparse pure double Boolean algebras, every variable-free term is equal either to 0_l , or to 1_r . Secondly, remark that, in any class of sparse pure double Boolean algebras, the word problem is in P when restricted to variable-free terms. Hence, in any class of sparse pure double Boolean algebras, the unification problem is in NP . As for its NP -hardness, it follows from a reduction of the satisfiability problem for Boolean formulas.

Secondly, there is the question of its type. See [1, 4, 5, 8] for details about unification types.

- Proposition 6.** 1. *The unification type in the class of all pure double Boolean algebras is nullary.*
 2. *The unification type in the class of all covered sparse pure double Boolean algebras is unitary.*

Proof. (1) Adapting the line of reasoning suggested by Jeřábek [8] in the case of modal logic K , we prove that $\{(\neg_r \neg_l x \vee_l x, \neg_r \neg_l x \vee_l 0_l)\}$ has no minimal complete set of unifiers in the class of all pure double Boolean algebras.

(2) Adapting the line of reasoning suggested by Baader and Ghilardi [1] or Dzik [4, 5] in the case of modal logic $S5$, we prove that every finite set of pairs of terms has a most general unifiers in the class of all covered sparse pure double Boolean algebras.

7 Conclusion

The decidability of the unification problem in the class of all pure double Boolean algebras and the unification type in the class of all sparse pure double Boolean algebras are still open.

Acknowledgements

Special acknowledgement is heartily granted to the organizers and the referees of UNIF 2016 for the feedback we have obtained from them and the colleagues of the *Institut de recherche en informatique de Toulouse* who made several helpful comments for improving the correctness and the readability of this article.

References

1. Baader, F., Ghilardi, S.: *Unification in modal and description logics*. Logic Journal of the IGPL **19** (2011) 705–730.
2. Balbiani, P.: *Deciding the word problem in pure double Boolean algebras*. Journal of Applied Logic **10** (2012) 260–273.
3. Davey, B., Priestley, H.: *Introduction to Lattices and Order*. Cambridge University Press (2002).
4. Dzik, W.: *Unitary unification of S5 modal logics and its extensions*. Bulletin of the Section of Logic **32** (2003) 19–26.
5. Dzik, W.: *Unification Types in Logic*. Wydawnictwo Uniwersytetu Śląskiego (2007).
6. Ganter, B., Wille, R.: *Formal Concept Analysis: Mathematical Foundations*. Springer (1999).
7. Herrmann, C., Luksch, P., Skorsky, M., Wille, R.: *Algebras of semiconcepts and double Boolean algebras*. Technische Universität Darmstadt (2000).
8. Jeřábek, E.: *Blending margins: the modal logic K has nullary unification type*. Journal of Logic and Computation **25** (2015) 1231–1240.
9. Vormbrock, B.: *A solution of the word problem for free double Boolean algebras*. In Kuznetsov, S., Schmidt, S. (editors): *Formal Concept Analysis*. Springer (2007) 240–270.
10. Wille, R.: *Boolean concept logic*. In Ganter, B., Mineau, G. (editors): *Conceptual Structures: Logical, Linguistic, and Computational Issues*. Springer (2000) 317–331.