Equations and Quasiequations of Commutative Bounded GBL-Algebras are PSPACE-Complete

Simone Bova

Vanderbilt University (Nashville TN, USA)

joint work with Franco Montagna

BLAST 2011
University of Kansas (Lawrence KS, USA)
June 1-5, 2011

Outline

Outline

Motivation

Commutative Bounded GBL-Algebras
Equations and Quasiequations

Background

(Strong) Finite Model Property
Finite Representation

Contribution

PSPACE-Hardness
PSPACE-Containment

Open

Commutative Bounded GBL-Algebras | Definition

$\mathbf{A} = (A, \wedge, \vee, \cdot, \backslash, \top, \bot)$ algebra of type $(2, 2, 2, 2, 0, 0)$.

Definition (Commutative Bounded GBL-Algebras, [JT02])

$\mathbf{A}$ is a *commutative bounded (cb) residuated lattice* if:

1. $(A, \wedge, \vee, \top, \bot)$ is a bounded lattice;
2. $(A, \cdot, \top)$ is a commutative monoid; *
3. $x \cdot z \leq y$ iff $z \leq x \backslash y$ holds identically (*residuation*).

A cb residuated lattice $\mathbf{A}$ is a *(cb) GBL-algebra*, $\mathbf{A} \in \mathcal{CBGBL}$, if:

4. $x \wedge y = x \cdot (x \backslash y)$ holds identically (*divisibility*).

---

*The property that the identity is the top is called *integrality*.

Commutative Bounded GBL-Algebras | Logic

Examples (Algebraic Semantics of Propositional Logics)

1. *Heyting* algebras, algebraic semantics of intuitionistic logic, are *idempotent* GBL-algebras, $x \cdot x = x = x \wedge x$.
2. *BL-algebras*, algebraic semantics of fuzzy logic [H98], are *prelinear* GBL-algebras, $x \backslash y \vee y \backslash x = \top$.

Thus, GBL-algebras form the algebraic semantics
of an (interesting) common fragment of intuitionistic logic and fuzzy logic
(a *many-valued intuitionistic logic*, or a *constructive fuzzy logic*).

Equations and Quasiequations

t, s GBL-terms. For all $\mathbf{A} \in \mathcal{CBGBL}$, $\mathbf{A} \models t = s$ iff $\mathbf{A} \models t \backslash s \wedge s \backslash t = \top$.

Definition (Equational and Quasiequational Theories of $\mathcal{CBGBL}$)

$\mathsf{H} = \{(\{s_1, \ldots, s_k\}, t) \mid \forall \mathbf{A} \in \mathcal{CBGBL}, \mathbf{A} \models s_1 = \top \wedge \cdots \wedge s_k = \top \rightarrow t = \top\}$.
$\mathsf{E} = \{(S, t) \in \mathsf{H} \mid S = \{\top\}\} \subseteq \mathsf{H}$.

† Noncommutative GBL-quasiequations are undecidable [JM09]. Decidability of noncommutative GBL-equations is open.

Equations and Quasiequations

t, s GBL-terms. For all $\mathbf{A} \in \mathcal{CBGBL}$, $\mathbf{A} \models t = s$ iff $\mathbf{A} \models t \backslash s \wedge s \backslash t = \top$.

Definition (Equational and Quasiequational Theories of $\mathcal{CBGBL}$)

$\mathsf{H} = \{(\{s_1, \ldots, s_k\}, t) \mid \forall \mathbf{A} \in \mathcal{CBGBL}, \mathbf{A} \models s_1 = \top \wedge \cdots \wedge s_k = \top \rightarrow t = \top\}$.
$\mathsf{E} = \{(S, t) \in \mathsf{H} \mid S = \{\top\}\} \subseteq \mathsf{H}$.

Fact

H *(thus,* E*) is decidable [JM09] via strong finite model property.* †

† Noncommutative GBL-quasiequations are undecidable [JM09]. Decidability of noncommutative GBL-equations is open.

Equations and Quasiequations

t, s GBL-terms. For all $\mathbf{A} \in \mathcal{CBGBL}$, $\mathbf{A} \models t = s$ iff $\mathbf{A} \models t\backslash s \wedge s\backslash t = \top$.

Definition (Equational and Quasiequational Theories of $\mathcal{CBGBL}$)

$\mathsf{H} = \{(\{s_1, \dots, s_k\}, t) \mid \forall \mathbf{A} \in \mathcal{CBGBL}, \mathbf{A} \models s_1 = \top \wedge \cdots \wedge s_k = \top \rightarrow t = \top\}$.
$\mathsf{E} = \{(S, t) \in \mathsf{H} \mid S = \{\top\}\} \subseteq \mathsf{H}$.

Fact

H *(thus,* E*) is decidable [JM09] via strong finite model property.* †

Question

Computational complexity of E *and* H*?*

---

† Noncommutative GBL-quasiequations are undecidable [JM09]. Decidability of noncommutative GBL-equations is open.

Equations and Quasiequations

t, s GBL-terms. For all $\mathbf{A} \in \mathcal{CBGBL}$, $\mathbf{A} \models t = s$ iff $\mathbf{A} \models t \backslash s \wedge s \backslash t = \top$.

Definition (Equational and Quasiequational Theories of $\mathcal{CBGBL}$)

$\mathsf{H} = \{(\{s_1, \ldots, s_k\}, t) \mid \forall \mathbf{A} \in \mathcal{CBGBL}, \mathbf{A} \models s_1 = \top \wedge \cdots \wedge s_k = \top \rightarrow t = \top\}$.
$\mathsf{E} = \{(S, t) \in \mathsf{H} \mid S = \{\top\}\} \subseteq \mathsf{H}$.

Fact

H *(thus,* E*) is decidable [JM09] via strong finite model property.* †

Question

Computational complexity of E *and* H*?*

Remark

Both theories are PSPACE-complete for Heyting algebras [S03],
coNP-complete for BL-algebras [BHMV01].

† Noncommutative GBL-quasiequations are undecidable [JM09]. Decidability of noncommutative GBL-equations is open.

Outline

Commutative GBL-Algebras | Finite Model Property

Definition (Countermodel)

Q GBL-quasiequation over $\{y_1, \ldots, y_l\}$. Q *fails* in $\mathcal{CBGBL}$ iff
Q has a *countermodel*, ie, exist $\mathbf{A} \in \mathcal{CBGBL}$, $\mathbf{h} \in A^{\{y_1,\ldots,y_l\}}$ st $\mathbf{A}, \mathbf{h} \not\models Q$.

Commutative GBL-Algebras | Finite Model Property

Definition (Countermodel)

Q GBL-quasiequation over $\{y_1, \ldots, y_l\}$. Q *fails* in $\mathcal{CBGBL}$ iff
Q has a *countermodel*, ie, exist $\mathbf{A} \in \mathcal{CBGBL}$, $\mathbf{h} \in A^{\{y_1,\ldots,y_l\}}$ st $\mathbf{A}, \mathbf{h} \not\models Q$.

Definition (Finite GBL-Algebras)

$\mathbf{FCGBL} = \{\mathbf{A} \mid \mathbf{A} \text{ finite in } \mathcal{CBGBL}\}$.

Theorem (Strong Finite Model Property, [JM09])

Q fails in $\mathcal{CBGBL}$ iff Q fails in **FCGBL**.

Proof (Sketch).

$\mathcal{CBGBL}$ is generated as a quasivariety by finite members [JM09, Theorem 5.2]. □

Finite Commutative GBL-Algebras | Representation

Proposition (Divisibility implies Distributivity)

$\mathbf{A} \in \mathcal{CBGBL}$ *has a distributive bounded lattice reduct.*

Proof.

$(x \wedge y) \vee (x \wedge z) \leq x \wedge (y \vee z)$ *and*

$$
\begin{aligned}
x \wedge (y \vee z) &= (y \vee z)((y \vee z)\backslash x), && \text{by } v \wedge w = w \wedge v = w(w\backslash v),\\
&= y((y \vee z)\backslash x) \vee z((y \vee z)\backslash x), && \text{by } (v \vee w)u = vu \vee wu,\\
&= y(y\backslash x \wedge z\backslash x) \vee z(y\backslash x \wedge z\backslash x), && \text{by } (v \vee w)\backslash u = v\backslash u \wedge w\backslash u,\\
&\leq y(y\backslash x) \vee z(z\backslash x), && \text{by } v \leq w \text{ implies } uv \leq uw,\\
&= (x \wedge y) \vee (x \wedge z), && \text{by } v \wedge w = v(v\backslash w).
\end{aligned}
$$

□

Finite Commutative GBL-Algebras | Representation

Proposition (Divisibility implies Distributivity)

$\mathbf{A} \in \mathcal{CBGBL}$ *has a distributive bounded lattice reduct.*

Proof.

$(x \wedge y) \vee (x \wedge z) \leq x \wedge (y \vee z)$ *and*

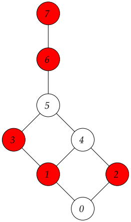
$$
\begin{aligned}
x \wedge (y \vee z) &= (y \vee z)((y \vee z)\backslash x), && \text{by } v \wedge w = w \wedge v = w(w\backslash v),\\
&= y((y \vee z)\backslash x) \vee z((y \vee z)\backslash x), && \text{by } (v \vee w)u = vu \vee wu,\\
&= y(y\backslash x \wedge z\backslash x) \vee z(y\backslash x \wedge z\backslash x), && \text{by } (v \vee w)\backslash u = v\backslash u \wedge w\backslash u,\\
&\leq y(y\backslash x) \vee z(z\backslash x), && \text{by } v \leq w \text{ implies } uv \leq uw,\\
&= (x \wedge y) \vee (x \wedge z), && \text{by } v \wedge w = v(v\backslash w).
\end{aligned}
$$

□

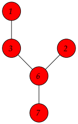
Idea

Adapt Birkhoff representation of finite distributive lattices by finite posets to finite commutative bounded GBL-algebras.

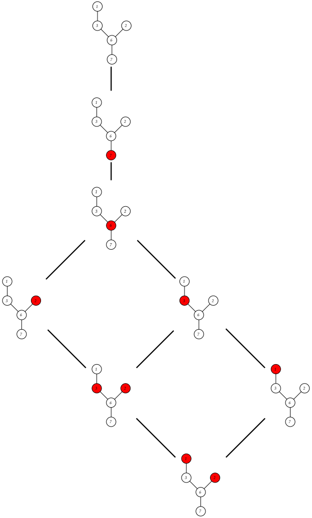
Finite Distributive Lattices | Birkhoff Representation



$\mathbf{L} \in \mathbf{FBDL}$



$J(\mathbf{L}) \in \mathbf{FP}$



$D(J(\mathbf{L})) = \mathbf{L}$

Finite Commutative GBL-Algebras | Representation

Definition (Finite $\mathbb{N}$-Labelled Posets)

$\mathbf{FNP} = \{(P, \leq_P, l_P) \mid (P, \leq_P) \text{ finite poset}, l_P \colon P \to \mathbb{N}\}$.

Notation

$I(A) = \{z \in A \mid z^2 = z\} = \{z \in A \mid z \textit{ idempotent}\}$.

Finite Commutative GBL-Algebras | Representation

Definition (Finite $\mathbb{N}$-Labelled Posets)

$\mathbf{FNP} = \{(P, \leq_P, l_P) \mid (P, \leq_P)$ finite poset, $l_P \colon P \to \mathbb{N}\}$.

Notation

$I(A) = \{z \in A \mid z^2 = z\} = \{z \in A \mid z$ *idempotent*$\}$.

Definition (Map J)

$J \colon \mathbf{FCGBL} \to \mathbf{FNP}$ such that, for all $\mathbf{A} \in \mathbf{FCGBL}$,

$$J(\mathbf{A}) = (P, \leq_P, l_P),$$

where $P = \{x \in I(A) \mid x$ join irreducible in $\mathbf{A}\}$, $x \leq_P y$ iff $y \leq x$ in $\mathbf{A}$, and

$$l_P(x) = |\{y \mid \bigvee_{x > w \in I(A)} w < y \leq x\}|.$$

Finite Commutative GBL-Algebras | Algebra to Poset via J

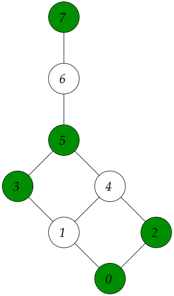
$\mathbf{A} = (\{0, \dots, 7\}, \wedge, \vee,$

xy	01234567
0	00000000
1	00010111
2	00202222
3	01031333
4	00212444
5	01234555
6	01234556
7	01234567

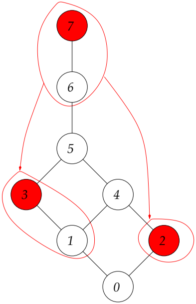
,

$x\backslash y = \bigvee\{z \mid xz \leq y\}$	01234567
0	74321000
1	77343111
2	74724222
3	77377333
4	77777444
5	77777755
6	77777776
7	77777777

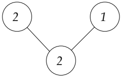
$, 7, 0)$,



$\mathbf{A} \in \mathbf{FCGBL}$



computing $J(\mathbf{A}) \dots$



$J(\mathbf{A}) \in \mathbf{FNP}$

Finite Commutative GBL-Algebras | Representation

Definition (Map D, Poset Product, [JM09])

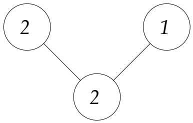
$D\colon \mathbf{FNP} \to \mathbf{FCGBL}$ such that, for all $\mathbf{P} = (P, \leq_P, l_P) \in \mathbf{FNP}$,

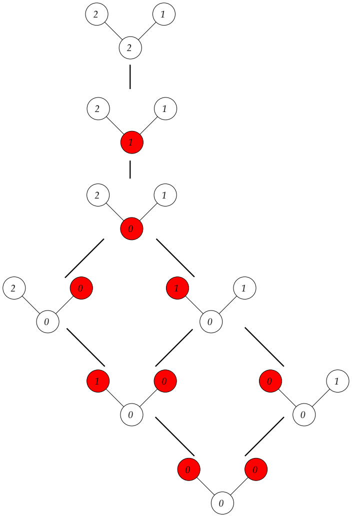
$$D((P, \leq_P, l_P)) = \bigotimes_{x \in \mathbf{P}} [l_P(x)] = (\prod_{x \in \mathbf{P}} [l_P(x)], \wedge, \vee, \cdot, \backslash, \top, \bot),$$

the *(finite) poset product (over* $\mathbf{P}$*)*, where:

1. $[l_P(x)] = (\{0, 1, \ldots, l_P(x)\}, \wedge_x, \vee_x, \cdot_x, \backslash_x, \top_x, \bot_x)$, where:
   - 1.1 $\wedge_x = \min, \vee_x = \max, \top_x = l_P(x), \bot_x = 0$;
   - 1.2 $n \cdot_x m = \max\{n + m - l_P(x), 0\}$;
   - 1.3 $n \backslash_x m = \min\{m + l_P(x) - n, l_P(x)\}$;
2. $\prod_{x \in \mathbf{P}} [l_P(x)] = \{h \in \prod_{x \in P} [l_P(x)] \mid h(x) = \bot_x$ or $h(y) = \top_y$ for all $x <_P y\}$;
3. $(f \circ g)(x) = f(x) \circ_x g(x)$ for all $x \in P$ and $\circ \in \{\wedge, \vee, \cdot\}$;
4. $(f \backslash g)(x) = f(x) \backslash_x g(x)$ if $f(y) \leq_y g(y)$ for all $x <_P y$, and $\bot_x$ otherwise;
5. $\top(x) = \top_x$ and $\bot(x) = \bot_x$ for all $x \in P$.

Finite Commutative GBL-Algebras | Poset to Algebra via D





$J(\mathbf{A}) = (P, \leq_P, l_P) \in \mathbf{FNP}$

$D(J(\mathbf{A})) = \bigotimes_{x \in \mathbf{P}} [l_P(x)] = \mathbf{A} \in \mathbf{FCGBL}$

Finite Commutative GBL-Algebras | Representation

Theorem (Finite Representation, [JM09])

$D(J(\mathbf{A})) = \mathbf{A}$ *for all* $\mathbf{A} \in \mathbf{FCGBL}$.

Finite Commutative GBL-Algebras | Representation

Theorem (Finite Representation, [JM09])

$D(J(\mathbf{A})) = \mathbf{A}$ *for all* $\mathbf{A} \in \mathbf{FCGBL}$.

Examples

Finite Heyting algebras correspond to $\{(P, \leq_P, l_P) \in \mathbf{FNP} \mid l_P\colon P \to \{1\}\}$.
Finite BL-algebras correspond to $\{(P, \leq_P, l_P) \in \mathbf{FNP} \mid (P, \leq_P^{dual})$ forest$\}$.

Finite Commutative GBL-Algebras | Representation

Theorem (Finite Representation, [JM09])

$D(J(\mathbf{A})) = \mathbf{A}$ *for all* $\mathbf{A} \in \mathbf{FCGBL}$.

Examples

Finite Heyting algebras correspond to $\{(P, \leq_P, l_P) \in \mathbf{FNP} \mid l_P \colon P \to \{1\}\}$.
Finite BL-algebras correspond to $\{(P, \leq_P, l_P) \in \mathbf{FNP} \mid (P, \leq_P^{dual})$ forest$\}$.

Corollary

Q fails in $\mathcal{CBGBL}$ iff Q fails in a finite poset product $\bigotimes_{x \in \mathbf{P}}[l_P(x)]$.

Proof (Sketch).

By the representation theorem, every finite GBL-algebra is isomorphic to some finite poset product $\bigotimes_{x \in \mathbf{P}}[l_P(x)]$ *[JM09, Theorem 6.5].* □

Outline

Computational Complexity | PSPACE

$L \subseteq \{0,1\}^*$ decision problem. $x \in \{0,1\}^n$ has size n.

Computational Complexity | PSPACE

$L \subseteq \{0,1\}^*$ decision problem. $x \in \{0,1\}^n$ has size n.

Definition (Karp Reduction)

$L' \leq_m^p L$ if there is a Karp reduction $K\colon \{0,1\}^* \to \{0,1\}^*$ from L' to L, ie, an algorithm K using $\leq n^c$ time (n size, c constant) st $x \in L'$ iff $K(x) \in L$.

Computational Complexity | PSPACE

$L \subseteq \{0,1\}^*$ decision problem. $x \in \{0,1\}^n$ has size n.

Definition (Karp Reduction)

$L' \leq_m^p L$ if there is a Karp reduction $K\colon \{0,1\}^* \to \{0,1\}^*$ from L' to L, ie, an algorithm K using $\leq n^c$ time (n size, c constant) st $x \in L'$ iff $K(x) \in L$.

Definition (PSPACE-Complete)

$L \in$ PSPACE iff L has decision algorithm using $\leq n^c$ space (n size, c constant).
L is PSPACE-hard if $L' \leq_m^p L$ for all $L' \in$ PSPACE.
L is PSPACE-complete if $L \in$ PSPACE and L is PSPACE-hard.

Computational Complexity | PSPACE

$L \subseteq \{0,1\}^*$ decision problem. $x \in \{0,1\}^n$ has size n.

Definition (Karp Reduction)

$L' \leq_m^p L$ if there is a Karp reduction $K\colon \{0,1\}^* \to \{0,1\}^*$ from L' to L, ie, an algorithm K using $\leq n^c$ time (n size, c constant) st $x \in L'$ iff $K(x) \in L$.

Definition (PSPACE-Complete)

$L \in$ PSPACE iff L has decision algorithm using $\leq n^c$ space (n size, c constant).
L is PSPACE-hard if $L' \leq_m^p L$ for all $L' \in$ PSPACE.
L is PSPACE-complete if $L \in$ PSPACE and L is PSPACE-hard.

Definition (QBF)

Let $A = Q_l y_l \cdots Q_1 y_1 B$ be a sentence where $Q_i \in \{\forall, \exists\}$ and
$B = D_1 \vee \cdots \vee D_k$ Boolean DNF over variables $\{y_1, \ldots, y_l\}$.
Then, $A \in$ QBF iff $\mathbf{2} \models A$.

Main Result

Theorem

Both E *and* H *are PSPACE-complete.*

‡ Adaptation of [S03] to the nonidempotent case. Conjectured in [BM09].

Main Result

Theorem
Both E *and* H *are PSPACE-complete.*

Proof.
As $\mathsf{E} \subseteq \mathsf{H}$*, it is sufficient to show the following two facts.*

Lemma
E *is PSPACE-hard (GBL-equations are PSPACE-hard).* ‡

Lemma ([BM09])
H *is in PSPACE (GBL-quasiequations are in PSPACE).*

□

‡ Adaptation of [S03] to the nonidempotent case. Conjectured in [BM09].

Commutative GBL-Equations are PSPACE-Hard

Notation

t GBL-term. $\bar{t} = t\backslash\bot$, $t^2 = t \cdot t$, $2t = ((t\backslash\bot) \cdot (t\backslash\bot))\backslash\bot$.

Definition (Reduction K)

For all sentences $A = Q_l y_l \cdots Q_1 y_1 B$ st $Q_i \in \{\forall, \exists\}$ and $B = \bigvee_{j=1,\ldots,m} D_j$ is a Boolean DNF, define $K(A) = t_l(y_1, \ldots, y_l, y_{1+l}, \ldots, y_{2l})$ inductively by:

Commutative GBL-Equations are PSPACE-Hard

Notation

t GBL-term. $\bar{t} = t\backslash\bot$, $t^2 = t \cdot t$, $2t = ((t\backslash\bot) \cdot (t\backslash\bot))\backslash\bot$.

Definition (Reduction K)

For all sentences $A = Q_l y_l \cdots Q_1 y_1 B$ st $Q_i \in \{\forall, \exists\}$ and $B = \bigvee_{j=1,\ldots,m} D_j$ is a Boolean DNF, define $K(A) = t_l(y_1, \ldots, y_l, y_{1+l}, \ldots, y_{2l})$ inductively by:

$$t_0 = \bigvee_{j=1,\ldots,m} D_j[y_k/2y_k, \neg y_k/2\bar{y}_k \mid k = 1, \ldots, l];$$

$$t_i = \begin{cases} (t_{i-1}\backslash y_{i+l})\backslash(y_i^2\backslash y_{i+l} \vee \bar{y}_i{}^2\backslash y_{i+l}), & \text{if } Q_i = \exists; \\ (y_i^2 \vee \bar{y}_i{}^2)\backslash t_{i-1}, & \text{if } Q_i = \forall. \end{cases}$$

Commutative GBL-Equations are PSPACE-Hard

Notation

t GBL-term. $\bar{t} = t\backslash\bot$, $t^2 = t \cdot t$, $2t = ((t\backslash\bot)\cdot(t\backslash\bot))\backslash\bot$.

Definition (Reduction K)

For all sentences $A = Q_l y_l \cdots Q_1 y_1 B$ st $Q_i \in \{\forall, \exists\}$ and $B = \bigvee_{j=1,\ldots,m} D_j$ is a Boolean DNF, define $K(A) = t_l(y_1, \ldots, y_l, y_{1+l}, \ldots, y_{2l})$ inductively by:

$$t_0 = \bigvee_{j=1,\ldots,m} D_j[y_k/2y_k, \neg y_k/2\bar{y}_k \mid k = 1, \ldots, l];$$

$$t_i = \begin{cases} (t_{i-1}\backslash y_{i+l})\backslash(y_i^2\backslash y_{i+l} \vee \bar{y}_i{}^2\backslash y_{i+l}), & \text{if } Q_i = \exists; \\ (y_i^2 \vee \bar{y}_i{}^2)\backslash t_{i-1}, & \text{if } Q_i = \forall. \end{cases}$$

Lemma

E *is PSPACE-hard.*

Proof (Sketch).

$K(A)$ is logspace computable in the size of A. A (nontrivial) induction on $k = 0, 1, \ldots, l$ shows that $\mathbf{2} \not\models A$ iff $K(A)$ fails over a finite poset product iff $K(A) \notin$ E. Thus, QBF $\leq_m^p$ E via K, but QBF is PSPACE-hard [Pap94]. □

Example

$A = \exists y_2 \forall y_1((\neg y_1 \wedge y_2) \vee (y_1 \wedge \neg y_2)).$

Example

$A = \exists y_2 \forall y_1 ((\neg y_1 \wedge y_2) \vee (y_1 \wedge \neg y_2))$.

Inductive computation of $K(A) = t_2(y_1, y_2, y_3, y_4)$:

Example

$A = \exists y_2 \forall y_1 ((\neg y_1 \wedge y_2) \vee (y_1 \wedge \neg y_2))$.

Inductive computation of $K(A) = t_2(y_1, y_2, y_3, y_4)$:

$t_0 = \ (2\bar{y_1} \wedge 2y_2) \vee (2y_1 \wedge 2\bar{y_2})$,

Example

$A = \exists y_2 \forall y_1((\neg y_1 \wedge y_2) \vee (y_1 \wedge \neg y_2))$.

Inductive computation of $K(A) = t_2(y_1, y_2, y_3, y_4)$:

$t_0 = \ (2\bar{y_1} \wedge 2y_2) \vee (2y_1 \wedge 2\bar{y_2}),$

$t_1 = \ (y_1^2 \vee \bar{y_1}^2) \backslash t_0,$

Example

$A = \exists y_2 \forall y_1((\neg y_1 \wedge y_2) \vee (y_1 \wedge \neg y_2))$.

Inductive computation of $K(A) = t_2(y_1, y_2, y_3, y_4)$:

$t_0 = \ (2\bar{y_1} \wedge 2y_2) \vee (2y_1 \wedge 2\bar{y_2})$,

$t_1 = \ (y_1^2 \vee \bar{y_1}^2) \backslash t_0$,

$t_2 = \ (t_1 \backslash y_4) \backslash (y_2^2 \backslash y_4 \vee \bar{y_2}^2 \backslash y_4)$

Example

$A = \exists y_2 \forall y_1((\neg y_1 \wedge y_2) \vee (y_1 \wedge \neg y_2))$.

Inductive computation of $K(A) = t_2(y_1, y_2, y_3, y_4)$:

$$
\begin{aligned}
t_0 =&\ (2\bar{y_1} \wedge 2y_2) \vee (2y_1 \wedge 2\bar{y_2}),\\
t_1 =&\ (y_1^2 \vee \bar{y_1}^2)\backslash t_0,\\
t_2 =&\ (t_1\backslash y_4)\backslash(y_2^2\backslash y_4 \vee \bar{y_2}^2\backslash y_4)\\
=&\ (((y_1^2 \vee \bar{y_1}^2)\backslash((2\bar{y_1} \wedge 2y_2) \vee (2y_1 \wedge 2\bar{y_2})))\backslash y_4)\backslash(y_2^2\backslash y_4 \vee \bar{y_2}^2\backslash y_4).
\end{aligned}
$$

Example

$A = \exists y_2 \forall y_1 ((\neg y_1 \wedge y_2) \vee (y_1 \wedge \neg y_2))$.

Inductive computation of $K(A) = t_2(y_1, y_2, y_3, y_4)$:

$$
\begin{aligned}
t_0 &= (2\bar{y_1} \wedge 2y_2) \vee (2y_1 \wedge 2\bar{y_2}), \\
t_1 &= (y_1^2 \vee \bar{y_1}^2)\backslash t_0, \\
t_2 &= (t_1\backslash y_4)\backslash(y_2^2\backslash y_4 \vee \bar{y_2}^2\backslash y_4) \\
&= (((y_1^2 \vee \bar{y_1}^2)\backslash((2\bar{y_1} \wedge 2y_2) \vee (2y_1 \wedge 2\bar{y_2})))\backslash y_4)\backslash(y_2^2\backslash y_4 \vee \bar{y_2}^2\backslash y_4).
\end{aligned}
$$

$\mathbf{2} \not\models A \ldots$

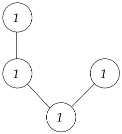
Example

$A = \exists y_2 \forall y_1((\neg y_1 \wedge y_2) \vee (y_1 \wedge \neg y_2)).$

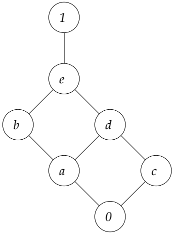
Inductive computation of $K(A) = t_2(y_1, y_2, y_3, y_4)$:

$$
\begin{aligned}
t_0 &= (2\bar{y_1} \wedge 2y_2) \vee (2y_1 \wedge 2\bar{y_2}),\\
t_1 &= (y_1^2 \vee \bar{y_1}^2)\backslash t_0,\\
t_2 &= (t_1 \backslash y_4)\backslash(y_2^2\backslash y_4 \vee \bar{y_2}^2\backslash y_4)\\
&= (((y_1^2 \vee \bar{y_1}^2)\backslash((2\bar{y_1} \wedge 2y_2) \vee (2y_1 \wedge 2\bar{y_2})))\backslash y_4)\backslash(y_2^2\backslash y_4 \vee \bar{y_2}^2\backslash y_4).
\end{aligned}
$$

$\mathbf{2} \not\models A \ldots$ the lemma yields a finite countermodel $\mathbf{A}$ to $K(A)$ (take $y_1 = a, y_2 = b, y_4 = 0$):



$(T, \leq_T, 1)$



$\mathbf{A} = D((T, \leq_T, 1))$

Tight Tree Embedding Lemma

Theorem (Tight Tree Embedding, [BM09])

Let Q be a GBL-quasiequation of size n. Then, Q fails in $\mathcal{CBGBL}$ iff Q fails in a poset product $\bigotimes_{x \in \mathbf{P}}[l_P(x)]$ over a finite rooted tree $(P, \leq_P)$ such that:

1. $|P| \in \exp(\text{poly}(n))$;
2. $\max\{|S| \mid S \text{ chain in } P\} \in \text{poly}(n)$;
3. $l_P(x) \in \exp(\text{poly}(n))$ *for all* $x \in P$.

Proof (Sketch).

[BM09, Lemma 2] Every finite countermodel to Q embeds into some finite poset product $\bigotimes_{x \in \mathbf{P}}[l_P(x)]$ where $\mathbf{P}$ is satisfies conditions (1)-(3). (1)-(2) obtained combinatorially, (3) obtained geometrically along the lines of [M87]. □

Commutative GBL-Quasiequations are in PSPACE

Lemma

H *is in PSPACE.*

Proof (Sketch).

[BM09, Lemma 4] We describe a nondeterministic polynomial space algorithm that decides the complement of H*. But coNPSPACE = PSPACE [Pap94].*
Let Q be a GBL-quasiequation. The idea of the algorithm is to search exhaustively the space of countermodels (poset products) satisfying conditions (1)-(3) in the tight embedding theorem wrt Q. (1)-(3) allow to implement a terminating search in polyspace. □

Pseudocode

FINDCOUNTERMODEL($Q = (\{s_1, \ldots, s_k\}, t)$)
guess $\mathbf{h}(v_1) = (h_1(v_1), \ldots, h_l(v_1)) \in [l_P(v_1)]^l$ ▶ $y_1, \ldots, y_l$ variables in Q
$H \leftarrow () + \mathbf{h}(v_1)$
guess $\mathbf{i}(v_1) \in \{0,1\}^m$ ▶ $r_1, \ldots, r_m$ subterms of form $r_{i_1} \backslash r_{i_2}$ in Q, r_i evaluated pointwise at v_0 iff $\mathbf{i}(v_1)_i = 1$
$I \leftarrow () + \mathbf{i}(v_1)$
if not($t < \top = s_1 = \cdots = s_k$ at v_1 wrt $\mathbf{h}(v_1), \mathbf{i}(v_1)$)
  output 0 ▶ countermodel not found
guess $B = |P|$
$b \leftarrow 2, j \leftarrow 1$
while $b \leq B$
  if($j = 1$ and $\{i \mid \mathbf{i}(v_j)_i = 0\} = \emptyset$)
    output 1 ▶ countermodel found
  else if($j > 1$ and $\{i \mid \mathbf{i}(v_j)_i = 0\} = \emptyset$)
    $j \leftarrow j - 1, H \leftarrow H - \mathbf{h}(v_j)$ ▶ backtrack
  else if($\{i \mid \mathbf{i}(v_j)_i = 0\} = \emptyset$)
    $j \leftarrow j + 1, b \leftarrow b + 1$ ▶ iterate
    guess $\mathbf{h}(v_j) = (h_1(v_j), \ldots, h_l(v_j)) \in [l_P(v_j)]^l$
    $H \leftarrow H + \mathbf{h}(v_j)$
    guess $\mathbf{i}(v_j) \in \{0,1\}^m$
    $I \leftarrow I + \mathbf{i}(v_j)$
    if($\mathbf{h}(v_j)$ sound wrt $\mathbf{h}(v_{j-1}), \mathbf{i}(v_j) > \mathbf{i}(v_{j-1})$, and $u_{i_1} \leq u_{i_2}$ at v_j wrt $\mathbf{h}(v_j), \mathbf{i}(v_j)$ for all i st $\mathbf{i}(v_{j-1})_i = 1$)
      $\mathbf{i}(v_k)_i \leftarrow 1$ for all $k < j$ and i st $\mathbf{i}(v_j)_i = 1, \mathbf{i}(v_k)_i = 0$
    else output 0 ▶ countermodel not found
endwhile
output 0 ▶ countermodel not found

Outline

Open Problems

1. Hardness of unbounded commutative case (easy).
2. Decidability of noncommutative GBL-equations (difficult).

References

M. Baaz, P. Hájek, F. Montagna, and H. Veith.
Complexity of t-Tautologies.
Annals of Pure and Applied Logic, 113(1-3):3–11, 2001.

S. Bova and F. Montagna.
The Consequence Relation in the Logic of Commutative GBL-Algebras is PSPACE-Complete.
Theoretical Computer Science, 410:1143–1158, 2009.

P. Jipsen and F. Montagna.
The Blok-Ferreirim Theorem for Normal GBL-Algebras and its Application.
Algebra Universalis, 60:381–404, 2009.

P. Hájek.
Metamathematics of Fuzzy Logic.
Kluwer, 1998.

P. Jipsen and C. Tsinakis.
A Survey on Residuated Lattices.
In: J. Martínez (Editor), *Ordered Algebraic Structures*, Kluwer, 2002, pp. 19–56.

D. Mundici.
Satisfiability in Many-Valued Sentential Logic is NP-Complete.
Theoretical Computer Science, 52(3):145–153, 1987.

C. H. Papadimitriou.
Computational Complexity.
Addison-Wesley, 1994.

V. Švejdar.
On the Polynomial-Space Completeness of Intuitionistic Propositional Logic.
Archive for Mathematical Logic, 42(7):711–716, 2003.

Thank you!