A Decision Algorithm for Hájek's Basic Logic

Simone Bova

joint work with Franco Montagna

{bova,montagna}@unisi.it

University of Siena, Italy

MANYVAL'06, Gargnano (Italy)

March 20-22, 2006

Contribution

The tautology problem of Basic Logic is exponential-time decidable through a semantic algorithm [BHMV02, MPT03].

We describe a decision algorithm based on a different formalization of the problem.

The algorithm runs in $2^{O(n)}$ worst-case time.

Outline

Outline

- **Introduction**
  - A Complete Semantics for **BL**
  - **BL** Decidability
- **A Decision Algorithm for BL**
  - The Semantic Calculus RWBL
  - The Algorithm CHECKBL
  - Correctness and Complexity
- **Conclusion**
  - RWBL Pros/Cons

A Complete Semantics for BL

$L = (\bot, \top, \odot, \to)$. $p_0, p_1, \ldots$ variables. $A, B, C \in L$.

Definition [H98]. The **BL** Hilbert system has the axioms:

(A1)	$(A \to B) \to ((B \to C) \to (A \to C))$
(A2)	$(A \odot B) \to A$
(A3)	$(A \odot B) \to (B \odot A)$
(A4)	$(A \odot (A \to B)) \to (B \odot (B \to A))$
(A5a)	$(A \to (B \to C)) \to ((A \odot B) \to C)$
(A5b)	$((A \odot B) \to C) \to (A \to (B \to C))$
(A6)	$((A \to B) \to C) \to (((B \to A) \to C) \to C)$
(A7)	$\bot \to A$

and the *Modus Ponens* inference rule.

Definition. BL-TAUT $= \{\langle A \rangle : \mathbf{BL} \vdash_{HBL} A\} \subseteq \{0,1\}^*$.

A Complete Semantics for BL

For all $x \in \mathbf{R}$, $\lfloor x \rfloor$ is the integer part of x. $\lfloor \infty \rfloor = \infty$.

Definition [MPT03]. $(\omega)[0,1] = ([0,\infty], *, \Rightarrow_*, 0, \infty)$, where:

$$x * y = \begin{cases} x & \text{if } \lfloor x \rfloor < \lfloor y \rfloor \\ y & \text{if } \lfloor x \rfloor > \lfloor y \rfloor \\ x + y - \lfloor x \rfloor - 1 & \text{if } \lfloor x \rfloor = \lfloor y \rfloor < \infty \text{ and } 1 \leq x - \lfloor x \rfloor + y - \lfloor y \rfloor \\ \lfloor x \rfloor & \text{if } \lfloor x \rfloor = \lfloor y \rfloor < \infty \text{ and } 1 > x - \lfloor x \rfloor + y - \lfloor y \rfloor \\ \infty & \text{if } x = y = \infty \end{cases}$$

$$x \Rightarrow_* y = \begin{cases} y & \text{if } \lfloor y \rfloor < \lfloor x \rfloor \\ \lfloor x \rfloor + 1 - x + y & \text{if } \lfloor x \rfloor = \lfloor y \rfloor \text{ and } y < x \\ \infty & \text{if } x \leq y \end{cases}$$

A Complete Semantics for BL

Definition [MPT03]. A valuation of L into $(\omega)[0,1]$ is a map v such that:

- $v(\bot) = 0$, $v(\top) = \infty$ and $v(p_i) \in [0, \infty]$, $i \in \mathbf{N}$;
- $v(A \odot B) = v(A) * v(B)$ and $v(A \to B) = v(A) \Rightarrow_* v(B)$.

Theorem [MPT03]. **BL** $\vdash_{HBL} A$ iff $(\forall v)v(A) = \infty$, *i.e.*, $(\omega)[0,1]$ is a *complete* semantics for **BL**.

Write $A^v \equiv v(A)$, $A^i \equiv \lfloor v(A) \rfloor$, $A^d \equiv A^v - A^i$. $\top^v \equiv \top^i$. $size(A)$ is the *circuit complexity* of A.

BL Decidability

Theorem [$\sim$MPT03]. BL-TAUT $\in$ **EXP**.

Input: $A \in L$, $size(A) > 0$.
Question: $(\forall v) A^v = \top^v$?
Answer: *Divide-and-conquer* approach:

- **Divide**: Choose a *pivot* formula in the question and *reduce* the question to *simpler* subquestions, applying the definition by cases of v to the pivot.
- **Conquer**: Answer the subquestions:
  - *recursively*, if they are *reducible*;
  - *easily*, if they are *irreducible*.
- **Combine**: Answer "Yes" iff the anwer to *all* the subquestions is "Yes".

BL Decidability

Example (trivial). $size(A) = 0$, $A = p_i$:

Q: $(\forall v) p_i^v = \top^v$?

A: "No".

Example (hard). $size(A) > 0$, $A = (B \to C)$:

Q: $(\forall v)(B \to C)^v = \top^v$?

A: "Yes" iff the anwer to *all* the subquestions:

Q$_1$: $(\forall v)(C^v = \top^v \Leftarrow C^i < B^i)$?

Q$_2$: $(\forall v)(1 - B^d + C^v = \top^v \Leftarrow (B^i = C^i \wedge C^d < B^d))$?

Q$_3$: $(\forall v)(\top^v = \top^v \Leftarrow B^v \leq C^v)$?

is "Yes".

BL Decidability

Example (hard). $size(A) > 0$, $A = (B \odot C)$:

$$\textbf{Q}: \ (\forall v)\,\overbrace{(B \odot C)^v = \top^v}^{\sim literal}?$$

A: "Yes" iff the anwer to *all* the subquestions:

$\textbf{Q}_1$: $(\forall v)(B^v = \top^v \Leftarrow B^i < C^i)$?

$\textbf{Q}_2$: $(\forall v)(C^v = \top^v \Leftarrow C^i < B^i)$?

$\textbf{Q}_3$: $(\forall v)(B^d + C^v - 1 = \top^v \Leftarrow (B^i = C^i < \top^v \wedge 1 \leq B^d + C^d))$?

$\textbf{Q}_4$: $(\forall v)(B^i = \top^v \Leftarrow (B^i = C^i < \top^v \wedge B^d + C^d < 1))$?

$\textbf{Q}_5$: $(\forall v)\,\underbrace{(\top^v = \top^v \Leftarrow (B^v = \top^v \wedge C^v = \top^v)}_{\sim clause}$?

is "Yes".

Issue: Generalize to clause matrices.

Outline

The Semantic Calculus RWBL

Goal: Formalize *efficiently* the algorithm sketched above.

Idea (vague): Simplify the subquestions *as much as possible*.

Issues:

- **Subgoal 1**: Formalize the problem.
- **Subgoal 2**: Formalize the divide step and the recursive case of the conquer step.
- **Subgoal 3**: Formalize the easy case of the conquer step.

The Semantic Calculus RWBL

Subgoal 1: Find a set of *predicates* to formalize the questions.

Informal questions are clauses, *e.g.*:

$$1 - B^d + C^v = \top^v \vee B^i \neq C^i \vee B^d \leq C^d.$$

Informal literals are relations over *pairs* of reals, *e.g.*:

$$B^d \leq C^d,$$

and *sums of tuples* of reals, *e.g.*:

$$1 - B^d + C^v = \top^v.$$

The Semantic Calculus RWBL

The relations *chosen* are ($z \in \mathbf{Z}$):

- $B \ll_v C$ iff $B^i < C^i$
- $B \prec_v C$ iff $B^i = C^i \wedge B^d < C^d$
- $B \preccurlyeq_v C$ iff $B^i = C^i \wedge B^d \leq C^d$
- $A_1, \ldots, A_n \prec_{v,z} B_1, \ldots, B_m$ iff $A^i_1 = \cdots = B^i_m < \infty$ and

$$\sum_{i=1}^{n}(A_i^d - 1) < z + \sum_{i=1}^{m}(B_i^d - 1)$$

- $A_1, \ldots, A_n \preccurlyeq_{v,z} B_1, \ldots, B_m$ iff $A^i_1 = \cdots = B^i_m < \infty$ and

$$\sum_{i=1}^{n}(A_i^d - 1) \leq z + \sum_{i=1}^{m}(B_i^d - 1)$$

The Semantic Calculus RWBL

$\emptyset$ contributes 0.

Example.

- $A^v = \top^v$ becomes $\top \preccurlyeq_v A$.
- $\top^v = \top^v \vee C^v < B^v$ becomes $\top \preccurlyeq_v \top \vee C \prec_v B \vee C \ll_v B$.
- $B^d + C^d < 1$ becomes $B, C \prec_{v,-1} \emptyset$.
- $1 \leq B^d + C^d$ becomes $\emptyset \preccurlyeq_{v,1} B, C$.

A question is *reducible* if it has at least one formula of complexity > 0, and *irreducible* otherwise.

Example. $\top \preccurlyeq_v p_i$ is irreducible, $\top \preccurlyeq_v p_i \odot p_j$ is reducible.

The Semantic Calculus RWBL

Subgoal 2. Find a set of *rules* to simplify recursively a formalized question (*minimizing* the recursions).

Example. By the interpretation of $\odot$:

- if $B^i = C^i$, then $(B \odot C)^i = B^i = C^i$;
- if $B^i = C^i < \infty$ and $1 \leq B^d + C^d$, then $(B \odot C)^d - 1 = (B^d - 1) + (C^d - 1)$.

Let, *e.g.*, $B^i = C^i < \infty$ and $1 \leq B^d + C^d$.

- $B \odot C \ll_v A$ if and only if $B \ll_v A$
- $B \odot C \lhd_v A$ if and only if $B, C \lhd_{v,0} A$
- $\Gamma, B \odot C \lhd_{z,v} \Delta$ if and only if $\Gamma, B, C \lhd_{z,v} \Delta$

where $\lhd \in \{\prec, \preccurlyeq\}$.

The Semantic Calculus RWBL

Idea (definite): Generate subquestions with no occurrences of the *pivot*.

The *ReWriting Basic Logic calculus* has two *rewriting rules*:

$$(B \odot C)\frac{\mathbf{Q}_{\odot,1} \quad \mathbf{Q}_{\odot,2} \quad \mathbf{Q}_{\odot,3} \quad \mathbf{Q}_{\odot,4} \quad \mathbf{Q}_{\odot,5}}{\mathbf{Q}}$$

$$(B \rightarrow C)\frac{\mathbf{Q}_{\rightarrow,1} \quad \mathbf{Q}_{\rightarrow,2} \quad \mathbf{Q}_{\rightarrow,3}}{\mathbf{Q}}$$

where $\mathbf{Q}$ is the clause matrix of a reducible question, $(B \circ C)$ is the pivot and each $\mathbf{Q}_{\circ,j}$ is a clause matrix, $\circ \in \{\odot, \rightarrow\}$.

Remark: The divide step and recursive case of the conquer step are settled.

The Semantic Calculus RWBL

The rules meet logical and complexity requirements.

Claim 1. The rewriting rules are *sound* and *invertible*.

Proof (sketch). The rewriting rules satisfy:

- $(\forall v)(\mathbf{Q}_{\circ,1} \wedge \cdots \wedge \mathbf{Q}_{\circ,k_\circ}) \Rightarrow (\forall v)\mathbf{Q}$, and
- $(\forall v)\mathbf{Q} \Rightarrow (\forall v)(\mathbf{Q}_{\circ,1} \wedge \cdots \wedge \mathbf{Q}_{\circ,k_\circ})$

where $\circ \in \{\odot, \rightarrow\}$, $k_\odot = 5$, $k_\rightarrow = 3$. $\square$

Claim 2. The rewriting rules eliminate the pivot.

Proof (sketch). The pivot can be eliminated exploiting the consequences of the interpretation of $\odot$ and $\rightarrow$ while deriving the subquestions of a question. $\square$

The Semantic Calculus RWBL

Subgoal 3. Find an irreducible questions *checker*.

Claim 3. $\textsc{CheckAx}(\langle \mathbf{Q} \rangle) = 1$ iff $\mathbf{Q}$ is irreducible and the answer to the question $(\forall v)\mathbf{Q}$ is "Yes".

Proof (sketch). If $\mathbf{Q}$ is reducible, output 0. Otherwise, the negation $\neg\mathbf{Q}$ is a conjunction of atomic literals and, by the interpretation of the language, there exists a *linear program* P such that P is *feasible* iff $(\exists v)\neg\mathbf{Q}$. So, output 1 iff P is *un*feasible. $\square$

Remark: The easy case of the conquer step is settled.

The Algorithm CHECKBL

Input: $\langle A \rangle \in \{0,1\}^*$.
Output: 1 iff **BL** $\vdash_{HBL} A$.

CHECKBL($\langle A \rangle$)

- $A = A_1 >_c \cdots >_c A_n$, $size(A_i) > 0$;
- initialize a *labelled tree* T_A with root $A^v = \top^v$;
- for $i = 1, \ldots, n$, extend the *leaves* of T_A applying the rewriting rule with pivot A_i, until all the leaves are irreducible or the loop terminates;
- after $m \leq n$ steps, T_A has leaves $\{\mathbf{Q}_1, \ldots, \mathbf{Q}_k\}$;
- output 1 iff the checker outputs 1 for all the $\mathbf{Q}_j$'s.

Correctness and Complexity

Theorem [BM]. $\langle A \rangle \in$ BL-TAUT iff $\textsc{CheckBL}(\langle A \rangle) = 1$.

Proof (sketch). By Claim 1, the rewriting procedure

$$\{\mathbf{Q}\} = T_0 \xrightarrow{A_1} T_1 \xrightarrow{A_2} \ldots \xrightarrow{A_{m-1}} T_m = \{\mathbf{Q}_1, \ldots, \mathbf{Q}_k\},$$

satisfies $(\forall v)\mathbf{Q}$ iff $(\forall v)\mathbf{Q}_i$ for all $i = 1, \ldots, k$.
By Claim 2, $\mathbf{Q}_1, \ldots, \mathbf{Q}_k$ are irreducible, since the pivot A_i of the ith rewriting does not occur in the external nodes of T_{i+1}.
By Claim 3, the output is 1 iff $(\forall v)\mathbf{Q}$. $\square$

Correctness and Complexity

Theorem [BM]. BL-TAUT $\in$ **DTIME(**$2^{O(n)}$**)**.

Proof (sketch). In the worst case, T_A has $5^{n+1} - 1$ nodes. Each node can be encoded in $O(n^2)$ space. So, T_A can be encoded in $2^{O(n)}$ space.
The construction is feasible in time polynomial in the size of T_A, and each leaf can be checked in time polynomial in n and the *size* of the corresponding linear program [Y91]. □

Outline

RWBL Pros/Cons

Pro: the *size* of the proof trees is improved:

	[MPT03]	RWBL
height $\leq$	n^3	n
width $\leq$	$2^{O(n^3)}$	$2^{O(n)}$

Pro: suitable for *automatic proof search*.

Con: *slower* than Boolean logic propositional proof systems.

Con: *strongly coupled* with linear programming.

References

[A04] S. Aguzzoli.
Uniform description of calculi for all t-norm logics.

[BM] S. Bova, F. Montagna. Manuscript.
Proof Search in Hájek's Basic Logic.

[BHMV02] M. Baaz, P. Hájek, F. Montagna, H. Veith.
Complexity of t-tautologies.

[H98] P. Hájek.
Metamathematics of Fuzzy Logic.

[CFM04] A. Ciabattoni, C. G. Fermüller, G. Metcalfe.
Uniform rules and dialogue games for fuzzy logics.

[MPT03] F. Montagna, G. M. Pinna, E. B. P. Tiezzi.
A tableau calculus for Hájek's logic *BL*.

[Y91] Y. Ye.
An $O(n^3 L)$ potential reduction algorithm for linear programming.