A Labelled Sequent Calculus for BBI: Proof Theory and Proof Search

Zhé Hóu, Alwen Tiu, and Rajeev Goré

September 18, 2013

Motivation: reasoning about resources

Syntax of some resource logic L:

$$F := p \mid \top \mid \bot \mid F \wedge F \mid F \to F \mid \top^* \mid F * F \mid F \mathrel{-\!\!*} F$$

Two flavours of connectives.

Difference between $\wedge$ and $*$: reasoning about memory addresses

- m satisfies p and m satisfies $q \Leftrightarrow m$ satisfies $p \wedge q$.
- m_1 satisfies p and m_2 (disjoint from m_1) satisfies $q \Leftrightarrow$ the combination of m_1 and m_2 satisfies $p * q$.

Decide $\Gamma \vdash_L \varphi$ in a Hoare logic extended with these connectives.

Boolean BI

Syntax:

$$F := p \mid \top \mid \bot \mid F \wedge F \mid F \to F \mid \top^* \mid F * F \mid F -\!\!* F$$

Semantics:

- A monoid $(M, \circ, \epsilon)$: a set M, binary operator $\circ$, unit ϵ.
- A ternary relation $a, b \triangleright c$ iff $c \in a \circ b$.

$m \Vdash \top^*$ iff $m = \epsilon$ $\quad$ $m \Vdash P$ iff $P \in Var$ and $m \in v(P)$
$m \Vdash \bot$ iff never $\quad$ $m \Vdash A \vee B$ iff $m \Vdash A$ or $m \Vdash B$
$m \Vdash \top$ iff always $\quad$ $m \Vdash A \wedge B$ iff $m \Vdash A$ and $m \Vdash B$
$m \Vdash \neg A$ iff $m \nVdash A$ $\quad$ $m \Vdash A \to B$ iff $m \nVdash A$ or $m \Vdash B$
$m \Vdash A * B$ iff $\exists a, b.(a, b \triangleright m$ and $a \Vdash A$ and $b \Vdash B)$
$m \Vdash A -\!\!* B$ iff $\forall a, b.((m, a \triangleright b$ and $a \Vdash A)$ implies $b \Vdash B)$

Three flavours of BBI

Non-deterministic monoid $(M, \circ, \epsilon)$: given some $a, b \in M$, the composition $a \circ b$ may yield multiple results.

Partial deterministic monoid $(M, \circ, \epsilon)$: $\forall a, b \in M$ the composition $a \circ b$ either results in a singleton, or is undefined.

Total deterministic: the composition of any two elements must be defined as a singleton.

Our goal: give a simple proof system for BBI_{ND} that can be extended easily to *these semantics and other properties*, and consider proof search for automated reasoning.

Previous work

- 2013, Park, Seo & Park: automated reasoning using nested sequent calculus for BBI_{ND}, *but* they do not consider other semantics and properties.

- 2012, Larchey-Wendling & Galmiche: tableaux for BBI_{PD}, *but* they do not consider proof search and implementation, nor syntactic cut-elimination.

- 2010, Brotherston: display proof theory framework for BI family logics (eps. BBI_{ND}), that enjoys cut-elimination, *but* not proof search friendly.

LS_{BBI}: some sample rules

$$\frac{\Gamma[\epsilon/w] \vdash \Delta[\epsilon/w]}{\Gamma; w : \top^* \vdash \Delta}\ \top^* L \qquad \frac{}{\Gamma \vdash \epsilon : \top^*; \Delta}\ \top^* R$$

$$\frac{(x, y \rhd z); \Gamma; x : A; y : B \vdash \Delta}{\Gamma; z : A * B \vdash \Delta}\ *L$$

$$\frac{(x, y \rhd z); \Gamma \vdash x : A; z : A * B; \Delta \qquad (x, y \rhd z); \Gamma \vdash y : B; z : A * B; \Delta}{(x, y \rhd z); \Gamma \vdash z : A * B; \Delta}\ *R$$

$$\frac{(y, x \rhd z); (x, y \rhd z); \Gamma \vdash \Delta}{(x, y \rhd z); \Gamma \vdash \Delta}\ E \qquad \frac{(u, w \rhd z); (y, v \rhd w); (x, y \rhd z); (u, v \rhd x); \Gamma \vdash \Delta}{(x, y \rhd z); (u, v \rhd x); \Gamma \vdash \Delta}\ A$$

$$\frac{(x, \epsilon \rhd x); \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\ U \qquad \frac{(\epsilon, w \rhd w); \Gamma[w/w'] \vdash \Delta[w/w']}{(\epsilon, w' \rhd w); \Gamma \vdash \Delta}\ Eq_1$$

Soundness with respect to the non-deterministic semantics of BBI

Completeness with respect to the Hilbert system for BBI

Cut-elimination in the traditional way

Example derivation

$$\frac{a0 : (a * b) * c \vdash a0 : a * (b * c)}{\vdash a0 : (a * b) * c \to a * (b * c)} \to R$$

Example derivation

$$\frac{(x, y \triangleright z); \Gamma; x : A; y : B \vdash \Delta}{\Gamma; z : A * B \vdash \Delta} *L$$

$$\frac{\dfrac{(a1, a2 \triangleright a0); a1 : a * b; a2 : c \vdash a0 : a * (b * c)}{a0 : (a * b) * c \vdash a0 : a * (b * c)} *L}{\vdash a0 : (a * b) * c \to a * (b * c)} \to R$$

Example derivation

$$\frac{(u, w \triangleright z); (y, v \triangleright w); (x, y \triangleright z); (u, v \triangleright x); \Gamma \vdash \Delta}{(x, y \triangleright z); (u, v \triangleright x); \Gamma \vdash \Delta}\, A$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(a3, a5 \triangleright a0); (a2, a4 \triangleright a5); \cdots \vdash a0 : a * (b * c)}{(a1, a2 \triangleright a0); (a3, a4 \triangleright a1); a2 : c; a3 : a; a4 : b \vdash a0 : a * (b * c)}\, A}{(a1, a2 \triangleright a0); a1 : a * b; a2 : c \vdash a0 : a * (b * c)}\, *L}{a0 : (a * b) * c \vdash a0 : a * (b * c)}\, *L}{\vdash a0 : (a * b) * c \to a * (b * c)}\, \to R$$

Example derivation

$$\frac{(y, x \triangleright z); (x, y \triangleright z); \Gamma \vdash \Delta}{(x, y \triangleright z); \Gamma \vdash \Delta}\, E$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(a4, a2 \triangleright a5); (a3, a5 \triangleright a0); \cdots \vdash a0 : a * (b * c)}{(a3, a5 \triangleright a0); (a2, a4 \triangleright a5); \cdots \vdash a0 : a * (b * c)}\, E}{(a1, a2 \triangleright a0); (a3, a4 \triangleright a1); a2 : c; a3 : a; a4 : b \vdash a0 : a * (b * c)}\, A}{(a1, a2 \triangleright a0); a1 : a * b; a2 : c \vdash a0 : a * (b * c)}\, *L}{a0 : (a * b) * c \vdash a0 : a * (b * c)}\, *L}{\vdash a0 : (a * b) * c \rightarrow a * (b * c)}\, \rightarrow R$$

Example derivation

$$\dfrac{(x, y \triangleright z); \Gamma \vdash x : A; z : A * B; \Delta \qquad (x, y \triangleright z); \Gamma \vdash y : B; z : A * B; \Delta}{(x, y \triangleright z); \Gamma \vdash z : A * B; \Delta}\, {*R}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{a3 : a; \cdots \vdash a3 : a \qquad a2 : c; a3 : a; a4 : b \vdash a5 : b * c}{(a4, a2 \triangleright a5); (a3, a5 \triangleright a0); \cdots \vdash a0 : a * (b * c)}\, {*R}}{(a3, a5 \triangleright a0); (a2, a4 \triangleright a5); \cdots \vdash a0 : a * (b * c)}\, E}{(a1, a2 \triangleright a0); (a3, a4 \triangleright a1); a2 : c; a3 : a; a4 : b \vdash a0 : a * (b * c)}\, A}{(a1, a2 \triangleright a0); a1 : a * b; a2 : c \vdash a0 : a * (b * c)}\, {*L}}{a0 : (a * b) * c \vdash a0 : a * (b * c)}\, {*L}}{\vdash a0 : (a * b) * c \to a * (b * c)}\, {\to R}$$

Example derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a3 : a; \cdots \vdash a3 : a}\,id \quad \dfrac{\dfrac{}{\cdots ; a4 : b \vdash a4 : b}\,id \quad \dfrac{}{a2 : c; \cdots \vdash a2 : c}\,id}{a2 : c; a3 : a; a4 : b \vdash a5 : b * c}\,{*R}}{(a4, a2 \triangleright a5); (a3, a5 \triangleright a0); \cdots \vdash a0 : a * (b * c)}\,{*R}}{(a3, a5 \triangleright a0); (a2, a4 \triangleright a5); \cdots \vdash a0 : a * (b * c)}\,E}{(a1, a2 \triangleright a0); (a3, a4 \triangleright a1); a2 : c; a3 : a; a4 : b \vdash a0 : a * (b * c)}\,A}{(a1, a2 \triangleright a0); a1 : a * b; a2 : c \vdash a0 : a * (b * c)}\,{*L}}{a0 : (a * b) * c \vdash a0 : a * (b * c)}\,{*L}}{\vdash a0 : (a * b) * c \to a * (b * c)}\,{\to R}$$

Example derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a3 : a; \cdots \vdash a3 : a}\,id \quad \dfrac{\dfrac{}{\cdots ; a4 : b \vdash a4 : b}\,id \quad \dfrac{}{a2 : c; \cdots \vdash a2 : c}\,id}{a2 : c; a3 : a; a4 : b \vdash a5 : b * c}\,{*R}}{(a4, a2 \triangleright a5); (a3, a5 \triangleright a0); \cdots \vdash a0 : a * (b * c)}\,{*R}}{(a3, a5 \triangleright a0); (a2, a4 \triangleright a5); \cdots \vdash a0 : a * (b * c)}\,E}{(a1, a2 \triangleright a0); (a3, a4 \triangleright a1); a2 : c; a3 : a; a4 : b \vdash a0 : a * (b * c)}\,A}{(a1, a2 \triangleright a0); a1 : a * b; a2 : c \vdash a0 : a * (b * c)}\,{*L}}{a0 : (a * b) * c \vdash a0 : a * (b * c)}\,{*L}}{\vdash a0 : (a * b) * c \to a * (b * c)}\,{\to R}$$

Problems with proof search:

- Not sure when to apply which rule
- Not sure which relational atoms to use for structural rules
- Not sure which relational atom to use for $*R$ and $-\!\!* L$

Proof search: localising structural rules

(1) Apply structural rules to unify labels.

(2) Try to close the branch by applying zero-premise rules.

(3) Apply invertible logical rules as much as possible.

(4) Apply structural rules to generate necessary relational atoms.

(5) Apply "non-invertible" logical rules ($*R$, $-\!\!* L$) using existing relational atoms.

> *i.e., we only need to apply structural rules exactly before applying* $*R$, $-\!\!* L$, $\top^* R$, *or id.*

But...

- Not sure which relational atoms to use for structural rules
- Not sure which relational atom to use for $*R$ and $-\!\!* L$

Proof search: $FVLS_{BBI}$

use id, $\top^* R$ to guide structural rules and $*R$, $-\!\!* L$

$$\frac{}{\mathcal{G}||\Gamma; \mathbf{w}_1 : P \vdash \mathbf{w}_2 : P; \Delta}\ id \qquad \mathcal{G} \vdash^?_R (\mathbf{w}_1 = \mathbf{w}_2)$$

$$\frac{}{\mathcal{G}||\Gamma \vdash \mathbf{w} : \top^*; \Delta}\ \top^* R \qquad \mathcal{G} \vdash^?_R (\mathbf{w} = \epsilon)$$

$$\frac{\mathcal{G}||\Gamma \vdash \mathbf{x} : A; \mathbf{w} : A * B; \Delta \qquad \mathcal{G}||\Gamma \vdash \mathbf{y} : B; \mathbf{w} : A * B; \Delta}{\mathcal{G}||\Gamma \vdash \mathbf{w} : A * B; \Delta}\ *R \qquad \mathcal{G} \vdash^?_R (\mathbf{x}, \mathbf{y} \triangleright \mathbf{w})$$

$$\frac{\mathcal{G}||\Gamma; \mathbf{w} : A -\!\!* B \vdash \mathbf{x} : A; \Delta \qquad \mathcal{G}||\Gamma; \mathbf{w} : A -\!\!* B; \mathbf{z} : B \vdash \Delta}{\mathcal{G}||\Gamma; \mathbf{w} : A -\!\!* B \vdash \Delta}\ -\!\!* L \qquad \mathcal{G} \vdash^?_R (\mathbf{x}, \mathbf{w} \triangleright \mathbf{y})$$

To guarantee **soundness**, every relational atom must be accumulated along the branch upwards.

$FVLS_{BBI}$ is sound and complete w.r.t. LS_{BBI}.

Constraint solving in $FVLS_{BBI}$

Derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a3 : a; \cdots \vdash \mathbf{x}5 : a}\, id_1 \quad \dfrac{\dfrac{}{\cdots ; a4 : b \vdash \mathbf{x}7 : b}\, id_2 \quad \dfrac{}{a2 : c; \cdots \vdash \mathbf{x}8 : c}\, id_3}{a2 : c; a3 : a; a4 : b \vdash \mathbf{x}6 : b * c}\, *R_2}{(a3, a4 \rhd a1); (a1, a2 \rhd a0); a2 : c; a3 : a; a4 : b \vdash a0 : a * (b * c)}\, *R_1}{(a1, a2 \rhd a0); a1 : a * b; a2 : c \vdash a0 : a * (b * c)}\, *L_2}{a0 : (a * b) * c \vdash a0 : a * (b * c)}\, *L_1}{\vdash a0 : (a * b) * c \rightarrow a * (b * c)}\, \rightarrow R$$

Constraints:

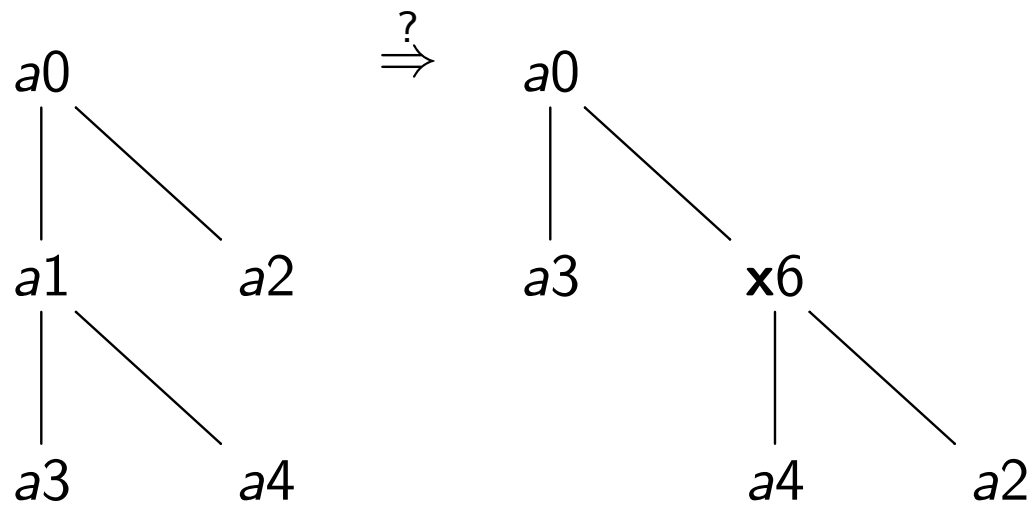
$$\begin{aligned}
id_3&: \quad (a1, a2 \rhd a0); (a3, a4 \rhd a1) \vdash^?_R (a2 = \mathbf{x}8)\\
id_2&: \quad (a1, a2 \rhd a0); (a3, a4 \rhd a1) \vdash^?_R (a4 = \mathbf{x}7)\\
*R_2&: \quad (a1, a2 \rhd a0); (a3, a4 \rhd a1) \vdash^?_R (\mathbf{x}7, \mathbf{x}8 \rhd \mathbf{x}6)\\
id_1&: \quad (a1, a2 \rhd a0); (a3, a4 \rhd a1) \vdash^?_R (a3 = \mathbf{x}5)\\
*R_1&: \quad (a1, a2 \rhd a0); (a3, a4 \rhd a1) \vdash^?_R (\mathbf{x}5, \mathbf{x}6 \rhd a0).
\end{aligned}$$

Constraint solving

Constrains simplified:

$$(a1, a2 \triangleright a0); (a3, a4 \triangleright a1) \vdash^{?}_{R} (a4, a2 \triangleright \mathbf{x}6)$$
$$(a1, a2 \triangleright a0); (a3, a4 \triangleright a1) \vdash^{?}_{R} (a3, \mathbf{x}6 \triangleright a0)$$

Visualised:



Can be solved by applying E (commutativity), A (associativity)!

Experimental results

Formula	BBeye (opt)	Naive (Vamp)	$FVLS_{BBI}$ Heuristic
$(a \mathrel{-\!\!*} b) \wedge (\top * (\top^* \wedge a)) \rightarrow b$	d(2) 0	0.003	0.001
$(\top^* \mathrel{-\!\!*} \neg(\neg a * \top^*)) \rightarrow a$	d(2) 0	0.003	0.000
$\neg((a \mathrel{-\!\!*} \neg(a * b)) \wedge ((\neg a \mathrel{-\!\!*} \neg b) \wedge b))$	d(2) 0	0.004	0.001
$\top^* \rightarrow ((a \mathrel{-\!\!*} (b \mathrel{-\!\!*} c)) \mathrel{-\!\!*} ((a * b) \mathrel{-\!\!*} c))$	d(2) 0.015	0.017	0.001
$\top^* \rightarrow ((a * (b * c)) \mathrel{-\!\!*} ((a * b) * c))$	d(2) 0.036	0.006	0.000
$\top^* \rightarrow ((a * ((b \mathrel{-\!\!*} e) * c)) \mathrel{-\!\!*} ((a * (b \mathrel{-\!\!*} e)) * c))$	d(2) 0.07	0.019	0.001
$\neg((a \mathrel{-\!\!*} \neg(\neg(d \mathrel{-\!\!*} \neg(a * (c * b))) * a)) \wedge c * (d \wedge (a * b)))$	d(2) 0.036	0.037	0.001
$\neg((c * (d * e)) \wedge B)$ where	d(2) 0.016	0.075	0.039
$B := ((a \mathrel{-\!\!*} \neg(\neg(b \mathrel{-\!\!*} \neg(d * (e * c))) * a)) * (b \wedge (a * \top)))$			
$\neg(C * (d \wedge (a * (b * e))))$ where	d(3) 96.639	0.089	0.038
$C := ((a \mathrel{-\!\!*} \neg(\neg(d \mathrel{-\!\!*} \neg((c * e) * (b * a))) * a)) \wedge c)$			
$(a * (b * (c * d))) \rightarrow (d * (c * (b * a)))$	d(2) 0.009	0.048	0.001
$(a * (b * (c * d))) \rightarrow (d * (b * (c * a)))$	d(3) 0.03	0.07	0.001
$(a * (b * (c * (d * e)))) \rightarrow (e * (d * (a * (b * c))))$	d(3) 1.625	1.912	0.001
$(a * (b * (c * (d * e)))) \rightarrow (e * (b * (a * (c * d))))$	d(4) 20.829	0.333	0.001
$\top^* \rightarrow (a * ((b \mathrel{-\!\!*} e) * (c * d)) \mathrel{-\!\!*} ((a * d) * (c * (b \mathrel{-\!\!*} e))))$	d(3) 6.258	0.152	0.007

BBeye: Park et al.'s prover.

Naive: Naive translation into first order logic, then use Vampire.

$FVLS_{BBI}$: Heuristic based free-variable prover.

Modularity

Partial deterministic and total deterministic

Partial deterministic monoid $(M, \circ, \epsilon)$: $\forall a, b \in M$ the composition $a \circ b$ either results in a singleton, or is undefined.

$$\frac{(a, b \triangleright c); \Gamma[c/d] \vdash \Delta[c/d]}{(a, b \triangleright c); (a, b \triangleright d); \Gamma \vdash \Delta} \; P$$

Total deterministic: the composition of any two elements must be defined as a singleton.

$$\frac{(a, b \triangleright c); \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \; T$$

c is fresh, a, b occur in the conclusion.

Modularity

Cancellativity and indivisible unit in separation logic models

Cancellativity: if $a \circ b$ defined, and $a \circ b = a \circ d$, then $b = d$.

$$\frac{(a, b \rhd c); \Gamma[b/d] \vdash \Delta[b/d]}{(a, b \rhd c); (a, d \rhd c); \Gamma \vdash \Delta}\ C$$

Indivisible unit: the unit ϵ cannot be divided into non-unit elements.

$$\frac{(\epsilon, \epsilon \rhd \epsilon); \Gamma[\epsilon/a][\epsilon/b] \vdash \Delta[\epsilon/a][\epsilon/b]}{(a, b \rhd \epsilon); \Gamma \vdash \Delta}\ IU$$

A problem

- The formula $\neg(\top^* \wedge A \wedge (B * \neg(C \mathrel{-\!\!*} (\top^* \to A))))$ is valid in the non-deterministic semantics, and provable using LS_{BBI}.
- But we cannot find a proof in the display calculus for BBI nor the nested sequent calculus for BBI.
  - We ran this formula on park's prover for a week, no result...

Future work

- A complete and terminating strategy for solving constraints
- Extend to applications
- Completeness issues w.r.t. other semantics & properties
- Locate the undecidability