Formalised Inductive Reasoning in the Logic of Bunched Implications

James Brotherston
Dept. of Computing, Imperial College London

SAS-14, 22–24 August 2007
Kongens Lyngby, Denmark

Overview

- the logic of bunched implications, BI, offers a convenient means of expressing properties of programs that access and modify some shared resource;
- separation logic is obtained by taking a model of BI in which the resources are heaps;
- program analysis based on separation logic, such as shape analysis, typically relies on inductively defined predicates to describe heap properties;
- inductive theorem proving based upon BI thus plays a key role in many such analyses.

Our contributions

- we extend BI with a general framework for inductive definitions;
- we give two proof systems in sequent calculus style for two different inductive reasoning techniques in the extended logic, $\mathrm{BI}_{\mathrm{ID}}$:
  1. explicit *rule induction* over definitions;
  2. *cyclic proof* embodying a notion of proof by infinite descent for inductively defined relations.
- we argue that cyclic proof has potential advantages over the standard approach to induction.

The logic of bunched implications (BI)

- our structures M contain a notion of resource, given by a partial commutative monoid $\langle R, \circ, e \rangle$;
- BI has the usual first-order connectives plus the new atomic formula I and binary connectives $*$ and $-\!\!*$;
- satisfaction of a formula F is given by the relation $M, r \models_\rho F$, where $r \in R$ is the "current resource state":

$$
\begin{array}{rcl}
M, r \models_\rho I & \Leftrightarrow & r = e \\
M, r \models_\rho Q\mathbf{t} & \Leftrightarrow & Q^M(r, \rho(\mathbf{t})) \\
M, r \models_\rho F_1 * F_2 & \Leftrightarrow & r = r_1 \circ r_2 \text{ and } M, r_1 \models_\rho F_1 \\
 & & \quad \text{and } M, r_2 \models_\rho F_2 \text{ for some } r_1, r_2 \in R \\
M, r \models_\rho F_1 -\!\!* F_2 & \Leftrightarrow & M, r' \models_\rho F_1 \text{ and } r' \circ r \text{ defined} \\
 & & \quad \text{implies } M, r' \circ r \models_\rho F_2 \text{ for all } r' \in R
\end{array}
$$

BI *with inductive definitions* (BI_{ID})

- two types of predicate symbol: ordinary $Q_1, Q_2, \ldots$ and inductive $P_1, \ldots, P_n$;
- our inductive definitions are given by a finite set Φ of productions which are rules of the form:

$$\frac{C(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \quad i \in \{1, \ldots, n\}$$

$$\begin{aligned} C(\mathbf{x}) ::= \ & \hat{F}(\mathbf{x}) \mid C(\mathbf{x}) \wedge C(\mathbf{x}) \mid C(\mathbf{x}) * C(\mathbf{x}) \\ & \mid \hat{F}(\mathbf{x}) \rightarrow C(\mathbf{x}) \mid \hat{F}(\mathbf{x}) \mathrel{-\!\!*} C(\mathbf{x}) \mid \forall x C(\mathbf{x}) \end{aligned}$$

where $\hat{F}(\mathbf{x})$ is any formula of BI not containing inductive predicates;

Standard models of BI_{ID}

- A set Φ of productions determines an n-ary monotone operator, φ_Φ;
- from the monotone operator φ_Φ we construct a sequence $(\varphi_\Phi^\alpha)_{\alpha \geq 0}$ of approximants by iteratively applying φ_Φ to $(\emptyset, \ldots, \emptyset)$;
- standard result: $\bigcup_\alpha \varphi_\Phi^\alpha$ is the least prefixed point of φ_Φ.

Definition

M is a standard model if we have $(P_1^M, \ldots, P_n^M) = \bigcup_\alpha \varphi_\Phi^\alpha$.

Example: inductive definitions

$$\frac{\top}{N0} \quad \frac{Nx}{Nsx}$$

$$\varphi_{\Phi_N}(X) = \{(r, 0^M) \mid r \in R\} \cup \{(r, s^M d) \mid (r, d) \in X\}$$

(Intuitively, the predicate N represents the property of being a natural number.)

$$\frac{I}{\mathtt{ls}\, x\, x} \qquad \frac{x \mapsto x' * \mathtt{ls}\, x'\, y}{\mathtt{ls}\, x\, y}$$

where $\mapsto$ is an ordinary predicate. (In separation logic, `ls` is a predicate representing (possibly cyclic) list segments.)

$$\begin{array}{rcl} \varphi_{\Phi_{\mathtt{ls}}}(X) & = & \{(e, (d, d)) \mid d \in D\} \\ & \cup & \{(r_1 \circ r_2, (d, d')) \mid (r_1, (d, d'')) \in \mapsto^M \\ & & \text{ and } (r_2, (d'', d')) \in X\} \end{array}$$

Sequent calculus rules for BI

We write sequents $\Gamma \vdash F$ where F is a formula and Γ is a bunch:

$$\Gamma ::= F \mid \Gamma ; \Gamma \mid \Gamma , \Gamma$$

where ; is equivalent to $\wedge$ and , is equivalent to $*$. The rules for the multiplicative connectives $*$ and $-\!\!*$ are:

$$\frac{\Delta \vdash F_1 \quad \Gamma(F_2) \vdash F}{\Gamma(\Delta, F_1 -\!\!* F_2) \vdash F} \,(-\!\!*\mathrm{L}) \qquad \frac{\Gamma(F_1, F_2) \vdash F}{\Gamma(F_1 * F_2) \vdash F} \,(*\mathrm{L})$$

$$\frac{\Gamma, F_1 \vdash F_2}{\Gamma \vdash F_1 -\!\!* F_2} \,(-\!\!*\mathrm{R}) \qquad \frac{\Gamma \vdash F_1 \quad \Delta \vdash F_2}{\Gamma, \Delta \vdash F_1 * F_2} \,(*\mathrm{R})$$

LBI_{ID}: a sequent calculus for induction in BI_{ID}

Extend sequent calculus for BI by adding introduction rules for inductively defined predicates. The right-introduction rules are simple unfolding rules, e.g. for `ls`:

$$\frac{\Gamma \vdash I}{\Gamma \vdash \mathtt{ls}\, t\, t}\ (\mathtt{ls}R_1) \qquad \frac{\Gamma \vdash t_1 \mapsto t * \mathtt{ls}\, t\, t_2}{\Gamma \vdash \mathtt{ls}\, t_1\, t_2}\ (\mathtt{ls}R_2)$$

The left-introduction rules embody rule induction over definitions, e.g. for `ls`:

$$\frac{\Delta; I \vdash Hxx \quad \Delta; x \mapsto x' * Hx'y \vdash Hxy \quad \Gamma(\Delta; Htu) \vdash F}{\Gamma(\Delta; \mathtt{ls}\, t\, u) \vdash F}\ (\text{Ind } \mathtt{ls})$$

where H is the induction hypothesis associated with `ls` and x, x', y are fresh.
(NB. mutual definitions give rise to mutual induction rules.)

A sample LBI_{ID} proof

We want to prove $\mathtt{ls}\, t_1\, t_2 * \mathtt{ls}\, t_2\, t_3 \vdash \mathtt{ls}\, t_1\, t_3$. After $(*\mathrm{L})$, apply the induction rule (Ind $\mathtt{ls}$) to $\mathtt{ls}\, t_1\, t_2$ with induction variables z_1, z_2 and induction hypothesis $\mathtt{ls}\, z_2\, t_3 \mathrel{-\!\!*} \mathtt{ls}\, z_1\, t_3$:

$$\dfrac{I \vdash \mathtt{ls}\, x\, t_3 \mathrel{-\!\!*} \mathtt{ls}\, x\, t_3 \quad x \mapsto x' * (\mathtt{ls}\, y\, t_3 \mathrel{-\!\!*} \mathtt{ls}\, x'\, t_3) \vdash \mathtt{ls}\, y\, t_3 \mathrel{-\!\!*} \mathtt{ls}\, x\, t_3 \quad \mathtt{ls}\, t_2\, t_3 \mathrel{-\!\!*} \mathtt{ls}\, t_1\, t_3, \mathtt{ls}\, t_2\, t_3 \vdash \mathtt{ls}\, t_1\, t_3}{\mathtt{ls}\, t_1\, t_2, \mathtt{ls}\, t_2\, t_3 \vdash \mathtt{ls}\, t_1\, t_3}\ (\text{Ind } \mathtt{l}$$

Only the second premise (induction step case) is non-trivial:

$$\dfrac{\dfrac{\dfrac{}{\mathtt{ls}\, y\, t_3 \vdash \mathtt{ls}\, y\, t_3}\ (\mathrm{Id}) \quad \dfrac{\dfrac{\dfrac{}{x \mapsto x' \vdash x \mapsto x'}\ (\mathrm{Id}) \quad \dfrac{}{\mathtt{ls}\, x'\, t_3 \vdash \mathtt{ls}\, x'\, t_3}\ (\mathrm{Id})}{x \mapsto x', \mathtt{ls}\, x'\, t_3 \vdash x \mapsto x' * \mathtt{ls}\, x'\, t_3}\ (*\mathrm{R})}{x \mapsto x', \mathtt{ls}\, x'\, t_3 \vdash \mathtt{ls}\, x\, t_3}\ (\mathtt{ls}R_2)}{\dfrac{x \mapsto x', (\mathtt{ls}\, y\, t_3 \mathrel{-\!\!*} \mathtt{ls}\, x'\, t_3), \mathtt{ls}\, y\, t_3 \vdash \mathtt{ls}\, x\, t_3}{\dfrac{x \mapsto x' * (\mathtt{ls}\, y\, t_3 \mathrel{-\!\!*} \mathtt{ls}\, x'\, t_3), \mathtt{ls}\, y\, t_3 \vdash \mathtt{ls}\, x\, t_3}{x \mapsto x' * (\mathtt{ls}\, y\, t_3 \mathrel{-\!\!*} \mathtt{ls}\, x'\, t_3) \vdash \mathtt{ls}\, y\, t_3 \mathrel{-\!\!*} \mathtt{ls}\, x\, t_3}\ (\mathrel{-\!\!*}\mathrm{R})}\ (*\mathrm{L})}\ (\mathrel{-\!\!*}\mathrm{L})$$

$CLBI^{\omega}_{ID}$: *a cyclic proof system for* BI_{ID}

- Rules are as for $\mathrm{LBI_{ID}}$ except the induction rules are replaced by weaker case-split rules, e.g. for `ls`:

$$\frac{\Gamma(t_1 = t_2; I) \vdash F \qquad \Gamma(t_1 \mapsto x, \mathtt{ls}\, x\, t_2) \vdash F}{\Gamma(\mathtt{ls}\, t_1\, t_2) \vdash F}\;(\text{Case } \mathtt{ls})$$

  where x is fresh.

- pre-proofs are finite derivation trees in which every *bud* (node to which no proof rule is applied) is assigned a *companion* (an identically labelled interior node);
- by identifying buds with their companions, pre-proofs can be understood as cyclic graphs.

Traces

$$\dfrac{\dfrac{(\dagger)\quad F \vdash G}{F; F \vdash G}\text{(Weak)}}{(\dagger)\quad F \vdash G}\text{(ContrL)}$$

- for soundness we need to impose some global condition on $\text{CLBI}^{\omega}_{\text{ID}}$ pre-proofs;
- a trace following a path in an $\text{CLBI}^{\omega}_{\text{ID}}$ pre-proof follows a formula occurring on the left of the sequents on the path;
- the trace progresses when the formula is an inductive predicate which is unfolded using its case-split rule;
- see Defn. 4.5 in the paper for a full definition!

Definition

An $\text{CLBI}^{\omega}_{\text{ID}}$ pre-proof $\mathcal{P}$ is a proof if for every infinite path in $\mathcal{P}$ there is a trace following some tail of the path that progresses infinitely often.

A sample $CLBI^{\omega}_{ID}$ proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\texttt{ls}\,x\,y \vdash \texttt{ls}\,x\,y}\,(\text{Id})}{I, \texttt{ls}\,x\,y \vdash \texttt{ls}\,x\,y}\,(\equiv)}{(x'=x; I), \texttt{ls}\,x'\,y \vdash \texttt{ls}\,x\,y}\,(=\text{L}) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{x\mapsto z \vdash x \mapsto z}\,(\text{Id}) \quad \dfrac{(\dagger)\ \texttt{ls}\,x\,x', \texttt{ls}\,x'\,y \vdash \texttt{ls}\,x\,y}{\texttt{ls}\,z\,x', \texttt{ls}\,x'\,y\vdash \texttt{ls}\,z\,y}\,(\text{Subst})}{x\mapsto z, \texttt{ls}\,z\,x', \texttt{ls}\,x'\,y \vdash x\mapsto z * \texttt{ls}\,z\,y}\,(*\text{R})}{x\mapsto z, \texttt{ls}\,z\,x', \texttt{ls}\,x'\,y \vdash \texttt{ls}\,x\,y}\,(\texttt{ls}R_2)}{x\mapsto z * \texttt{ls}\,z\,x', \texttt{ls}\,x'\,y \vdash \texttt{ls}\,x\,y}\,(*\text{L})}{(\dagger)\ \texttt{ls}\,x\,x', \texttt{ls}\,x'\,y \vdash \texttt{ls}\,x\,y}\,(\text{Case }\texttt{ls})}{\texttt{ls}\,x\,x' * \texttt{ls}\,x'\,y \vdash \texttt{ls}\,x\,y}\,(*\text{L})$$

A progressing trace following the cycle given by ($\dagger$) is highlighted. One can build an infinitely progressing trace on the only infinite path by concatenating copies of this trace. So this pre-proof is a proof.

LBI_{ID} *versus* $CLBI^{\omega}_{ID}$

Proposition

It is decidable whether a $CLBI^{\omega}_{ID}$ *pre-proof is a proof.*

Proposition

Both LBI_{ID} *and* $CLBI^{\omega}_{ID}$ *are sound: any provable sequent is true in all standard models.*

- some cyclic proofs seem to avoid the need for generalisation in inductive proof;
- for first-order logic with inductive definitions, cyclic proof subsumes proof by induction, with the equivalence of the two styles conjectured but not proven;
- our current work with Calcagno and Bornat develops a cyclic proof system employing separation logic to prove termination of imperative programs.

Further reading

- J. Brotherston, C. Calcagno and R. Bornat.
  Cyclic proofs of program termination in separation logic.
  Submitted; available from the first author's homepage.
- J. Brotherston and A. Simpson.
  Complete sequent calculi for induction and infinite descent.
  In *Proceedings of LICS 2007*.
- J. Brotherston.
  Sequent calculus proof systems for inductive definitions.
  PhD thesis, University of Edinburgh, November 2006.
- J. Brotherston.
  Cyclic proofs for first-order logic with inductive definitions.
  In *Proceedings of TABLEAUX 2005*.