# Proofs and countermodels in BI's pointer logic
# - abstract -

D. Galmiche and J. Gobillot and D. Méry

LORIA - Université Henri Poincaré
54506 Vandœuvre-lès-Nancy Cedex, France

## 1 Abstract

The logic of Bunched Implications (BI) is a resource-aware logic in which additive ($\wedge$,$\rightarrow$) and multiplicative ($*, -\!\!*$) connectives cohabit [7,8]. It appears as a central logic of resource (de)composition as illustrated by separation logics [6,9] and also by spatial logics for distributed and mobile computations [1]. Separation logics, which generally derive from BI logic, have been developed for reasoning about mutable data structures through extensions of Hoare logic and its related assertion language [5].

The question of defining an usable proof theory for such separation logics naturally arises [9] and therefore our aim is to study this question firstly for BI's pointer logic (PL) [5]. For that, we start from our recent results on a proof theory for propositional BI logic in which labels and constraints on labels are essential in order to capture the semantic interactions between connectives [3]. Our labelled tableaux method leads to important results like the decidability of propositional BI and the finite model property w.r.t. the Grothendieck topological semantics [4]. In case of non-provability, it builds countermodels from an open branch and a dependency graph generated in parallel with the tableaux rule expansions. It is clear that this method can be used to verify pre- and post-conditions, expressed in BI logic, that are generated during local reasoning and program verification for pointer's programs. But here, we aim to define a semantic method for proving or disproving PL's formulae and thus to provide an alternative to the model-checking approach with a procedure that allows to generate countermodels and consequently a way to analyze failures in the verification of some properties.

PL's formulae are $\phi ::= \alpha \mid false \mid emp \mid \phi \rightarrow \phi \mid \phi * \phi \mid \phi -\!\!* \phi \mid \exists x.\phi$ where $emp$ is the empty heap and $false$ is the falsity. As we are in a Boolean context, we can define other formulae as usual, for instance $\neg\phi \equiv \phi \rightarrow false$ and $\phi \vee \psi \equiv (\neg\phi) \rightarrow \psi$. A key point here is that the atomic formulae include the points-to relation ($E \mapsto E_1, E_2$) and an equality relation $E = E'$. Moreover, it explicitly deals with heaps ($h \in H$) and stacks ($s \in S$) and the semantics of assertions is given by a forcing relation of the form $s, h \models P$ which asserts that $P$ is true of stack $s$ and heap $h$. Concepts, models and results are detailed in [5]. Then, the challenge is to propose a proof-search method for PL that captures the properties of the pointer model and deals with the specificities of the points-to relation.

First, we apply the same methodology than for proof-search in BI logic, that consists in capturing the semantics of PL through rules with labels and constraints and then defining specific closure conditions for the corresponding tableaux method [4]. We begin our study with a fragment of PL ($\text{PL}^{\mapsto}$) including only atoms of the form ($E \mapsto E_1, E_2$) and propose a labelled tableau method for this fragment in which labels correspond to heaps and constraints capture the semantical conditions on these structures. In addition to the standard closure conditions for the BI connectives, we provide new closure conditions in order to deal with the points-to relation. Moreover, we prove the

correctness of the method for $\mathrm{PL}^{\mapsto}$ and its completeness through the construction of countermodels from an open branch and the dependency graph. From these results, we extend the calculus and the method in order to deal with atomic predicate of the form $E = E'$ and also with the existential quantification $\exists x.\phi$ for which we can use a standard technique that eliminates such a quantification by instantiation of the variables with constants according to the sign (or polarity) of formulae [2].

There exists an intuitionistic version of BI's pointer logic [5] and we can observe that its underlying BI logic is not Boolean BI but Affine BI, that is intuitionistic BI with weakening for the multiplicative conjunction $*$. We can extend our method in order to take into account the weakening through the labels and constraints management. It leads us to the same rules than for classical PL but with some new closure conditions. Moreover, we can prove the same above results for this intuitionistic variant of PL. As both versions (classical and intuitionistic) of PL are very interesting for specifying properties on pointers, for instance w.r.t. sharing, it is important to define usable and efficient methods that build proofs or countermodels for these logics. A next step will be to consider our proof-theoretical approach, and thus to provide such proof calculi, for separation logics in general [6,9].

## References


1. L. Cardelli and A.D. Gordon. Anytime, anywhere - modal logics for mobile ambients. In *27th ACM Symposium on Principles of Programming Languages, POPL 2000*, pages 1–13, Boston, USA, 2000.
2. M. Fitting. *First-Order Logic and Automated Theorem Proving.* Texts and Monographs in Computer Science. Springer Verlag, 1990.
3. D. Galmiche and D. Méry. Semantic labelled tableaux for propositional BI without bottom. *Journal of Logic and Computation*, 13(5):707–753, 2003.
4. D. Galmiche, D. Méry, and D. Pym. Resource Tableaux (extended abstract). In *16th Int. Workshop on Computer Science Logic, CSL 2002, LNCS 2471*, pages 183–199, September 2002. Edinburgh, Scotland.
5. S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. In *28th ACM Symposium on Principles of Programming Languages, POPL 2001*, pages 14–26, London, UK, 2001.
6. P. O'Hearn, J. Reynolds, and H. Yang. Local reasoning about programs that alter data structures. In *15th Int. Workshop on Computer Science Logic, CSL 2001, LNCS 2142*, pages 1–19, Paris, France, 2001.
7. P.W. O'Hearn and D. Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.
8. D.J. Pym. *The Semantics and Proof Theory of the Logic of Bunched Implications*, volume 26 of *Applied Logic Series.* Kluwer Academic Publishers, 2002.
9. J. Reynolds. Separation logic: A logic for shared mutable data structures. In *IEEE Symposium on Logic in Computer Science*, pages 55–74, Copenhagen, Danemark, July 2002.