Master Internship Proposal - 2025/2026

Non-aggregative BI Models and Calculi

D.Galmiche, D. Larchey-Wendling, D. Méry
LORIA, CNRS, Nancy
email: galmiche@loria.fr, larchey@loria.fr, dmery@loria.fr

1 Context

Nowadays computer systems are highly complex. Most of them are intrinsically distributed, dynamically and reconfigurable (Big Data, Internet of Things, Cloud Computing) and concurrent at both the physical level (CPU cores) and the logical level (threads). This level of complexity raises important technical problems. For instance, how to ensure that a system consisting of a huge multitude of heterogeneous components, built by different developers, provides the expected services? How to ensure that the components do not block waiting one for another (deadlock), or that one component has to wait indefinitely (denial of service)? How to check that the update of a component's software does not leak confidential information to third parties?
Understanding complex systems is only possible due to their *modularity* and the components of a modular system can be viewed, at a higher level of abstraction, as *resources*, that can be either static or dynamic (a piece of data or a process), physical or logical (CPUs or threads), having simple atomic (a memory cell) or more elaborated structure (such as a linked list or a pipeline). The main operation on a resource domain is *composition*, formally understood as the product of a partial monoid. The resource domains having received most attention so far have *aggregative* composition operations, i.e. composition operations that behave like unions of disjoint sets, as in Separation and Bunched Logics [6, 10]. Some of the algebraic (equational) properties of aggregative compositions, i.e. compositions which obey the principle that *the whole is the sum of its parts*, have already been considered in existing works [3, 8]. However, because the aggregative nature of a composition operator cannot be fully understood in terms of its equational properties alone, we need to introduce more structural aspects like *preordering* the resources or measuring their *size*. For instance, an expected property of an aggregative operator is that it should keep its operands as discernable parts of its result. It should also make the whole composite *bigger* than any of its parts. However, many natural resource compositions do not follow those principles.

For an intuitive example, consider a *merging* composition about dual resources that annihilate each other (like particles and anti-particles in physics). Merging a resource (particle) a with its dual resource (anti-particle) b, we get the empty resource e (intangible amount of energy). Obviously, the size of the composition $a * b$ is not bigger than the size of a or the size of b, and only looking at e, one cannot tell if it is the result of $a * b$, or of the merging of another dual pair c and d.

2 Subject

BI is a resource logic that combines additive and multiplicative connectives [9]. In the standard version the additive connectives are interpreted as connectives of intuitionisitc logic (IL) and the multiplicative connectives are interpreted as connectives of Multiplicative Intuitionnistic Linear Logic (MILL). In other words, BI logic is a combination of IL and MILL, and a conservative extension of each of these logics. BI and its variants are the logical kernel of others resource logics: separation logics, tree logics, context logics, ambiants... There exist several proof systems for BI, mainly a purely syntactic sequent calculus, called LBI [9], and a labelled tableaux systems, called TBI [5].

The first goal of the internship is to investigate more specific classes of BI models in which resource composition satisfy additional properties, like cancellativity ($x * y = x * z \Longrightarrow y = z$) or local contraction ($x * x \leq x$ for resources x in a strict subset of the resource monoid), and ask whether those models are complete w.r.t. BI and if not, for which variant of BI (as a logic) could they be complete.

The second goal of the internship is to extend and adapt existing calculi for BI to the new classes of non-aggregative BI models derived in the first goal and try to identify decidable classes of such models (for example, via the termination of the corresponding tableau calculi).

Implementing such calculi is also in the scope of the internship.

References

[1] Gabrielle Anderson and David Pym. A calculus and logic of bunched resources and processes. *Theoretical Computer Science*, 614:63–96, February 2016.

[2] Stephen Brookes and Peter W. O'Hearn. Concurrent separation logic. *ACM SIGLOG News*, 3(3):47–65, August 2016.

[3] James Brotherston and Jules Villard. Parametric Completeness for Separation Theories. In *Principles of Programmong Languages, POPL*, pages 453–464. ACM, 2014.

[4] Jean-René Courtault, Didier Galmiche, and David Pym. A logic of separating modalities. *Theoretical Computer Science*, 637:30–58, July 2016.

[5] Didier Galmiche and Daniel Méry. Semantic labelled tableaux for propositional BI without bottom. *Journal of Logic and Computation*, 13(5):707–753, 2003.

[6] Didier Galmiche, Daniel Méry, and David Pym. The semantics of BI and Resource Tableaux. *Mathematical Structures in Computer Science*, 15(6):1033–1088, 2005.

[7] Jonas Braband Jensen and Lars Birkedal. Fictional separation logic. In Helmut Seidl, editor, *Programming Languages and Systems*, pages 377–396, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[8] Dominique Larchey-Wendling and Didier Galmiche. Looking at separation algebras with boolean bi-eyes. In Josep Díaz, Ivan Lanese, and Davide Sangiorgi, editors, *TCS 2014*, volume 8705 of *LNCS*, pages 326–340. Springer, 2014.

[9] Peter-W O'Hearn and David Pym. The Logic of Bunched Implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

[10] Peter-W O'Hearn, John C. Reynolds, and Hongseok Yang. Local reasoning about programs that alter data structures. In *Proceedings of the 15th International Workshop on Computer Science Logic*, CSL '01, pages 1–19, 2001.