Focusing Prover for Bunched Implications

Sabrina CHANTRELLE

Logic provides a good way to certifiably check the correctness in many systems. More specifically, the logic of Bunched Implications, **BI**, is a substructural logic, which provides a base for reasoning about resources. It serves as a foundation for reasoning about low-level program operations and constructs such as memory allocation and deallocation as well as mutable shared data-structures.

In this project we explore a proof theory with the aim of developing an automated theorem prover for the propositional fragment of **BI** so that the machine can find and check proofs too lenghty to be written out by hand. To achieve a practical implementation, we proceed in three steps to limit existing non-determinism: First, we develop a sequent caclulus for **BI** where structural rules such as weakening and contraction are admissible. Second, we classify and prove certain proof rules to be invertible. As a consequence, these proof rules can be applied immediately during proof search without losing completeness. Once application of invertible proof rules has terminated, we can focus on non-invertible proof rules. Finally, we propose implementation techniques to handle the tree-like context by a constraint structure.

Based on these theoretical ideas, we have implemented a prototypical propositional theorem prover for **BI**. This is one of the few existing provers for **BI** and is a first step towards providing an automated reasoning tool for **BI** which allows us to reason about correctness and safety properties of programs that manipulate shared mutable data-structures.