Available online at www.sciencedirect.com
SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 125 (2005) 117–135
www.elsevier.com/locate/entcs

Resource Graphs and Countermodels in Resource Logics

Didier Galmiche and Daniel Méry [1]

LORIA UMR 7503 - Université Henri Poincaré
Campus Scientifique, BP 239
54506 Vandœuvre-les-Nancy, France

Abstract

In this abstract we emphasize the role of a semantic structure called resource graph in order to study the provability in some resource-sensitive logics, like the Bunched Implications Logic (BI) or the Non-commutative Logic (NL). Such a semantic structure is appropriate for capturing the particular interactions between different kinds of connectives (additives and multiplicatives in BI, commutatives and non-commutatives in NL) that occur during proof-search and is also well-suited for providing countermodels in case of non-provability. We illustrate the key points with a tableau method with labels and constraints for BI and then present tools, namely BILL and CheckBI, which are respectively dedicated to countermodel generation and verification in this logic.

Keywords: resources, proof-search, semantics, labels, countermodels.

1 Introduction

Over the past few years there has been an increasing amount of interest for resource-sensitive logical systems. The notion of *resource* is a basic one in many fields, including in computer science. The location, ownership, access to and, indeed, consumption of, resources are central concerns in the design of systems, such as networks, and in the design of programs, which access memory and manipulate data structures like pointers. Among so-called resource logics, we can mention Linear Logic (LL) [11] with its resource consumption interpretation, and Bunched Implications logic (BI) [15,16] with its

[1] Email: galmiche@loria.fr, dmery@loria.fr

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2004.06.069

resource sharing interpretation but also order-aware (non-commutative) logic (NL) [1]. As specification logics, they can represent features as interaction, resource distribution and mobility, non-determinism, sequentiality or coordination of entities. For instance, BI has been recently used as an assertion language for mutable data structures [12] and in this context it is important to verify pre- or post-conditions expressed in this logic, mainly to discover non-theorems and if possible to provide explanation about this non-provability by generating readable and usable countermodels.

For the above mentioned resource logics, proof search is not trivial mainly because of the management of context splitting and bunches in the related sequent calculi. Moreover, the design of semantic-based methods is difficult because the semantics of such logics (like Grothendieck topological semantics for BI [16]), even if they are complete, are not always manageable in the context of proving or disproving formulae. Known methods, like tableaux or connections, dedicated to classical, intuitionistic or linear logics by using prefixes [13] cannot be easily extended to other resource logics. Therefore, in order to deal with the particular interactions occuring between connectives, for instance additive and multiplicative connectives in BI or commutative and non-commutative connectives in NL, our proposal is to start with a standard proof-search method (tableau or connection-based) and to define, for each logic, specific *labels* and (label) *constraints* that allow us to capture the interactions at a semantic level. It leads to the design of new calculi with labelled signed formulæ and constraints from which we define a new characterization of provability from standard notions, such as complementarity and closure conditions, extended with specific conditions about constraint satisfaction with respect to a particular set of constraints. This set is built during the proof-search process (tableau expansions or connection search) and can be easily represented as a graph, called *dependency graph*. It arises as the central syntactico-semantic structure from which the provability in some resource logics can be studied and allows us to generate countermodels, for instance, in Grothendieck topological semantics that is complete for BI. Another interesting point is to consider such a structure, with an appropriate valuation attached to some nodes, directly as a countermodel.

The relationships between semantics and syntax (labels and constraints) used to defined labelled calculi can be studied in both directions. For instance, in the case of BI without $\bot$, the labels and constraints directly reflect the elementary Kripke semantics of the logic [8] and thus the relationships between semantics and dependency graphs is clearly identified. In the case of BI (with $\bot$), the labels and constraints do not reflect the initial Grothendieck topological semantics, but considering the dependency or *resource graph*