Theoretical Computer Science 637 (2016) 30–58

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A logic of separating modalities

Jean-René Courtault [a], Didier Galmiche [a], David Pym [b,*]

[a] LORIA, Université de Lorraine, Campus Scientifique, BP 239, 54506 Vandoeuvre-lès-Nancy Cedex, France
[b] University College London, Gower Street, London WC1E 6BT, UK

ARTICLE INFO

Article history:
Received 29 April 2015
Received in revised form 16 February 2016
Accepted 30 April 2016
Available online 6 May 2016
Communicated by A. Avron

Keywords:
Bunched logic
Separation logic
Modal logic
Resource semantics
Tableaux
Concurrency

ABSTRACT

We present a logic of separating modalities, LSM, that is based on Boolean BI. LSM's modalities, which generalize those of S4, combine, within a quite general relational semantics, BI's resource semantics with modal accessibility. We provide a range of examples illustrating their use for modelling. We give a proof system based on a labelled tableaux calculus with countermodel extraction, establishing its soundness and completeness with respect to the semantics.

1. Introduction

The concept of resource is important in many fields of enquiry — including, among others, computer science, economics, and security. In recent years, mathematical work in logic has begun to analyse the concept of resource in quite systematic and quite useful ways, with computer science providing a rich source of motivations and examples.

One impetus for this work was provided by the so-called resource interpretation of Girard's Linear Logic [19], in which the number of occurrences of a propositional formula in a sequent is counted and in which the exponentials are used to provide countably infinitely many copies of propositional formulæ. An alternative approach — inspired, on the one hand, by a long semantic history in relevant logic (e.g., [34,11]) and, on the other, by work in the semantics of type theories — is exemplified by O'Hearn and Pym's Logic of Bunched Implications (BI) [30,26,33,16,17]. In BI, the concept of resource resides in an interpretation of BI's semantics: this approach, and its developments, is known as resource semantics.

Conceptually, resource semantics begins with a simple axiomatization of resource. Starting with a given homogeneous set of resource elements — for example, bags of fruit, units of currency, or computer memory — we expect the following properties:

- to be able to combine two units of the given type of resource to form a new unit of that type of resource;
- to be able to compare (using either a simple equality or an ordering) two units of a given type of resource;
- that combination and comparison should be appropriately compatible.

* Corresponding author.
E-mail addresses: jean-rene.courtault@loria.fr (J.-R. Courtault), didier.galmiche@loria.fr (D. Galmiche), d.pym@ucl.ac.uk (D. Pym).

http://dx.doi.org/10.1016/j.tcs.2016.04.040

This basic axiomatization has proved remarkably robust, supporting, for example, a good deal of work in Separation Logic and its precursors and developments, [11,22,28,35,27], and a vast subsequent literature.

Mathematically, this basic set-up is captured by a pre-ordered monoid of resources, defined as follows: $\mathbf{R} = (R, \sqsubseteq, \bullet, e)$, where R is a set of resource elements, $\sqsubseteq$ is a pre-order (writing $=$ for $\sqsubseteq \cap \sqsupseteq$) and $\bullet$ is a monoidal composition with unit e, subject to the functoriality coherence condition that if $r = s$ and $r' = s'$, then $r \bullet r' = s \bullet s'$ [30,26,33,17].

The semantics of (Boolean) BI is given using a satisfaction relation between resources and propositional formulæ, with cases such as

$$r \models \phi_1 \wedge \phi_2 \text{ iff } r \models \phi_1 \text{ and } r \models \phi_2,$$

that give the usual (additive) classical connectives, and cases such as

$$r \models \phi_1 * \phi_2 \text{ iff there exist } r_1 \text{ and } r_2 \text{ such that } r_1 \bullet r_2 = r \text{ and } r_1 \models \phi_1 \text{ and } r_2 \models \phi_2$$

and

$$r \models \phi \mathrel{-\!\!*} \psi \text{ iff for all } s \text{ such that } s \models \phi, \; r \bullet s \models \psi$$

that give the multiplicative, or separating, connectives.

In terms of resource semantics, the additive conjunction ($\wedge$) is simply interpreted as specifying that the conjuncts must share the available resources whereas in the case of multiplicative conjunction ($*$) the available resources must be divided between the two conjuncts. Similarly, in the multiplicative implication ($-\!\!*$), the resources required to support the implicational formula must be combined with those required to support the 'input' formula in order to obtain, by implication, the resources required to support the 'output' formula.

We can also work with intuitionistic BI, with its intuitionistic additives, as in [30,33,16,17], by considering a monoid of resources that carries not merely an equality but a pre-order, allowing intuitionistic implication to be defined in the usual way and leading to the multiplicative conjunction

$$r \models \phi_1 * \phi_2 \text{ iff there exist } r_1 \text{ and } r_2 \text{ such that } r_1 \bullet r_2 \sqsubseteq r \text{ and } r_1 \models \phi_1 \text{ and } r_2 \models \phi_2$$

In this case, the functoriality condition is that if $r \sqsubseteq s$ and $r' \sqsubseteq s'$, then $r \bullet r' \sqsubseteq s \bullet s'$.

The dynamics of systems is a central concern in computer science. Many models and logics have been proposed in order to capture system behaviours and reason about their properties. In particular, modal logics based on S4 or S5 and their intuitionistic variants [2,36] and temporal logics such as LTL [31] or CTL [12]. The interest in such logics derives from their ability to express properties such as *invariance* (is a property satisfied in all reachable states of the system?) and *reachability* (is it possible to reach a state satisfying a property?).

Modal extensions of BI have been proposed in order to introduce dynamics into resource semantics. One of them, called MBI [6,4,5], is a logic in which resources and processes co-evolve according to an operational semantics based on judgements of the form $R, E \xrightarrow{a} R', E'$, meaning that the process E evolves by performing an action a relative to available resources R so as to become the process E' with available resources R'. This logic captures the manipulation of resources through the dynamic of a system, but is not able to express properties relative to quantified actions (e.g., properties deriving from performing *any* action). MBI's purely logical theory remains relatively undeveloped. Nevertheless, the use of these ideas as a basis for a rigorously resource-based modelling tool has been described in [7,5].

Another modal extension of BI, called DBI, introduces a simple notion of dynamic resource in which properties of resources can change or be modified during the iteration of the system [8]. The modalities of DBI ($\Diamond$ and $\Box$) allow the expression of properties of resources at any reachable state. Moreover, there exists a sound and complete calculus with a countermodel extraction method for this logic. DBI is not able to capture resource manipulations by a system: its models capture systems that modify properties of resources, but not systems that produce and consume resources.

In this paper, we present a modal logic of resources — LSM, for 'Logic of Separating Modalities' — that is based on Boolean BI's resource semantics. The logic extends S4. The basic idea is to work with two-dimensional worlds (w, r) that correspond to the purely modal and purely resource components of the semantics. The key development derives from their combination to define resource-modalities $\Diamond_r$ and $\Box_r$ in which 'modal truth' is offset by 'resource truth'. These modalities generalize their counterparts in S4 ($\Diamond$ and $\Box$). In Section 2, we introduce the language and the semantics of LSM, using a quite general relational formulation. In Section 3, we illustrate the expressiveness of its modalities thorough a range of core examples from computer systems. Then, in Section 4, we develop an extended example, showing that LSM provides useful tools for reasoning about a rich model of concurrent computation: in particular,