Information and Computation 208 (2010)474–499

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

Adjunct elimination in Context Logic for trees

Cristiano Calcagno*, Thomas Dinsdale-Young, Philippa Gardner

Department of Computing, Imperial College, London, UK

ARTICLE INFO

Article history:
Received 5 November 2007
Revised 18 February 2009
Available online 5 December 2009

Keywords:
Context Logic
Adjunct elimination
Ehrenfeucht–Fraïssé games

ABSTRACT

We study adjunct-elimination results for Context Logic applied to trees, following previous results by Lozes for Separation Logic and Ambient Logic. In fact, it is not possible to prove such elimination results for the original single-holed formulation of Context Logic. Instead, we prove our results for multi-holed Context Logic.

1. Introduction

Separation Logic [1–3] and Ambient Logic [4] are related theories for reasoning, respectively, about heap update and static trees. Inspired by this work, Calcagno et al. invented Context Logic [5] for reasoning about structured data. In particular, they used Context Logic applied to trees to reason locally about tree update, following the reasoning style of Separation Logic for reasoning locally about heap update. Such local reasoning is not possible using Ambient Logic [6].

All these logics extend the standard propositional connectives with a structural (separating) composition for reasoning about disjoint subdata and the corresponding structural adjoint(s) for expressing properties such as weakest pre-conditions and safety conditions. For Separation Logic and Ambient Logic, Lozes [7] and then Dawar et al. [8] showed that the structural adjoints provide no additional expressive power on closed formulae. This result is interesting, as the adjunct connectives introduce quantification over potentially infinite sets whereas the structural composition only requires quantification over finite substructures. Following this work, Calcagno et al. proved adjunct elimination for Context Logic applied to sequences, and showed the correspondence with the $*$-free regular languages [9,6]. We expected an analogous result for Context Logic applied to trees, but instead found a counterexample (first reported in Dinsdale-Young's Masters thesis [10]).

Context Logic was originally introduced to establish local Hoare reasoning about tree update. For this application, it was enough to work with single-holed contexts, although we always understood that there were other forms of contexts requiring study. In Section 2, we present our counterexample to adjunct elimination for single-holed Context Logic. The key point is that, whereas structural composition reasons about trees by splitting them into contexts and trees, contexts cannot be split. One possible solution is simply to extend Context Logic with context composition and its corresponding adjoints. We do not know if adjunct elimination holds for this extension. We do know that current proof techniques cannot be immediately adapted. Instead, we prove an adjunct-elimination result for *multi-holed* Context Logic applied to trees, which provides a more general approach for splitting contexts.

* Corresponding author.
Email addresses: ccris@doc.ic.ac.uk (C. Calcagno), td202@doc.ic.ac.uk (T. Dinsdale-Young), pg@doc.ic.ac.uk (P. Gardner).

0890-5401/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2009.02.013

Our adjunct-elimination result uses a technique based on Ehrenfeucht–Fraïssé games, which was first used to prove adjunct elimination for Ambient Logic in [8]. For Context Logic, this technique naturally requires multi-holed contexts. To illustrate this, consider the tree $t = c_1(t_1)$ which denotes the application of context c_1 to tree t_1. The structural composition move in a game will split t into $c_2(t_2)$, leading to a case analysis relating c_1 and t_1 with c_2 and t_2 involving multi-holed contexts. For example, when t_2 is a subtree of c_1, this case is simply expressed using a two-holed context $d(\_, \_)$ with $d(t_2, \_) = c_1$ and $d(\_, t_1) = c_2$. Using multi-holed Context Logic, we are thus able to provide an adjunct-elimination result which conforms with the analogous results for Separation Logic and Ambient Logic.

We first published this adjunct-elimination result in the conference APLAS 2007 [11], although it does not contain most of the proofs. This journal paper provides the proofs, gives a more detailed account of adjunct elimination in the single-holed case (Section 2), where one adjoint can be removed and the other cannot, and provides a fuller account of multi-holed Context Logic (Section 3). We believe multi-holed Context Logic introduced here will play an important role in our future development of Context Logic since, although analysing multi-holed contexts was not necessary for our preliminary work on tree update, they do seem to be fundamental for other applications such as reasoning about concurrent tree update.

2. Single-holed Context Logic for trees

In order to motivate our use of multi-holed Context Logic, we shall first summarise single-holed Context Logic for trees (CL^{s}_{Tree}) [5] and the known facts concerning adjunct elimination.

2.1. The tree model

We begin by defining the tree model which consists of finite, ordered, unranked trees and tree contexts. Throughout the paper, the nodes of trees are labelled from an infinite set of atoms, the set of *node labels* Σ, ranged over by u, v, w.[1]

In the literature, a distinction is often drawn between structures with a single root node, which are called 'trees', and structures with any number of roots, called 'forests'. Results in Context Logic, including those presented here, do not generally rely on this distinction, and so we use the term 'trees' to refer to structures with any number of root nodes.

Definition 1 (*Trees and tree contexts*)**.** The set of trees $\mathcal{T}$, ranged over by a, b, and the set of (single-holed) tree contexts $\mathcal{C}^{s}$, ranged over by c, d, are defined as

$$a, b ::= \quad \varepsilon \;\big|\; u[a] \;\big|\; a_1 \mid a_2 \qquad (u \in \Sigma)$$

$$c, d ::= \quad \_ \;\big|\; u[c] \;\big|\; a \mid c \;\big|\; c \mid a \qquad (u \in \Sigma)$$

modulo structural equivalences given by the '|' operators being mutually associative and having identity ε (the empty tree). The notation u is used to abbreviate $u[\varepsilon]$.

Definition 2 (*Context application*)**.** Context application is a function, $ap : \mathcal{C}^{s} \times \mathcal{T} \to \mathcal{T}$, defined inductively over the structure of contexts by

$$\begin{aligned} ap(\_, b) &= b \\ ap(u[c], b) &= u[ap(c, b)] \\ ap(a \mid c, b) &= a \mid ap(c, b) \\ ap(c \mid a, b) &= ap(c, b) \mid a. \end{aligned}$$

The notation $c(a)$ is used to abbreviate $ap(c, a)$. Note that _ is the left identity of ap.

2.2. Single-holed Context Logic

Context Logic [5] was introduced by Calcagno et al. to reason about structured data (for example, trees), in contrast with Bunched Logic of O'Hearn and Pym [12] which reasons about unstructured resource (for example, heaps). Using Context Logic, it is possible to provide local Hoare reasoning about tree update, following O'Hearn, Reynolds and Yang's work on local Hoare reasoning about heap update [1–3]. The key observation in [5] was that local data update typically identifies the portion of data to be replaced, removes it, and inserts new data *in the same place*. Context Logic was therefore introduced to reason about both data and this place of insertion (contexts).

We now define single-holed Context Logic applied to trees, denoted CL^{s}_{Tree}. Our definition follows a similar pattern to the definitions of Separation Logic and Ambient Logic. It extends the propositional connectives of classical logic with general *structural* connectives for analysing the structure of single-holed contexts, and *specific* connectives for analysing the particular model under consideration (in this case, trees and tree contexts).

[1] We assume that the elements of Σ are distinct from all other constants introduced in this paper.