Lax Logical Relations

Gordon Plotkin¹,*; John Power¹,**; Donald Sannella¹,***; and Robert Tennent¹,²,†

¹ Laboratory for Foundations of Computer Science
University of Edinburgh, Edinburgh, U.K. EH9 3JZ
{gdp,ajp,dts}@dcs.ed.ac.uk
² Department of Computing and Information Science
Queen’s University, Kingston, Canada K7L 3N6
rdt@cs.queensu.ca

Abstract. Lax logical relations are a categorical generalisation of logical relations; though they preserve product types, they need not preserve exponential types. But, like logical relations, they are preserved by the meanings of all lambda-calculus terms. We show that lax logical relations coincide with the correspondences of Schoett, the algebraic relations of Mitchell and the pre-logical relations of Honsell and Sannella on Henkin models, but also generalise naturally to models in cartesian closed categories and to richer languages.

1 Introduction

Logical relations and various generalisations are used extensively in the study of typed lambda calculi, and have many applications, including

- characterising lambda definability [Pl73, Pl80, JT93, Al95];
- relating denotational semantic definitions [Re74, MS76];
- characterising parametric polymorphism [Re83];
- modelling abstract interpretation [Ab90];
- verifying data representations [Mi91];
- defining fully abstract semantics [OR95]; and
- modelling local state in higher-order languages [OT95, St96].

The two key properties of logical relations are

1. the so-called Basic Lemma: a logical relation is preserved by the meaning of every lambda term; and
2. inductive definition: the type-indexed family of relations is determined by the base-type components.

* This author was supported by EPSRC grants GR/J84205 and GR/M56333.
** This author was supported by EPSRC grants GR/J84205 and GR/M56333, and by a grant from the British Council.
*** This author was supported by EPSRC grant GR/K63795.
† This author was supported by a grant from the Natural Sciences and Engineering Research Council of Canada.
It has long been known that there are type-indexed families of conventional relations that satisfy the Basic Lemma but are not determined inductively in a straightforward way. Schoett \cite{Sc87} uses families of relations that are preserved by algebraic operations to treat behavioural inclusion and equivalence of algebraic data types; he terms them “correspondences,” but they have also been called “simulations” \cite{Mi71} and “weak homomorphisms” \cite{Gi68}. Furthermore, Schoett conjectures (pages 280–81) that the Basic Lemma will hold when appropriate correspondences are used between models of lambda calculi, and that such relations compose. Mitchell \cite[Sect. 3.6.2]{Mi90} terms them “algebraic relations,” attributing the suggestion to Gordon Plotkin\footnote{Plotkin recalls that the suggestion was made to him by Eugenio Moggi in a conversation.} and Samson Abramsky, independently, and asserts that the Basic Lemma is easily proved and (binary) algebraic relations compose. But Mitchell concludes that, because logical relations are easily constructed by induction on types, they “seem to be the important special case for proving properties of typed lambda calculi.”

Recently, Honsell and Sannella \cite{HS99} have shown that such relation families, which they term “pre-logical relations,” are both the largest class of conventional relations on Henkin models that satisfy the Basic Lemma, and the smallest class that both includes logical relations and is closed under composition. They give a number of examples and applications, and study their closure properties.

We briefly sketch two of their applications.

- The composite of (binary) logical relations need not be logical. It is an easy exercise to construct a counter-example; see, for instance, \cite{HS99}. But the composite of binary pre-logical relations is a pre-logical relation.
- Mitchell \cite{Mi91} showed that the use of logical relations to verify data representations in typed lambda calculi is complete, provided that all of the primitive functions are first-order. In \cite{HS99}, this is strengthened to allow for higher-order primitives by generalising to pre-logical relations. Honsell, Longley et al. \cite{HL1+} give an example in which a pre-logical relation is used to justify the correctness of a data representation that cannot be justified using a conventional logical relation.

In this work, we give a categorical characterisation of algebraic relations (simulations, correspondences) between Henkin models of typed lambda calculi. The key advantage of this characterisation is its generality. By using it, one can immediately generalise from Henkin models to models in categories very different from $Set$, and to languages very different from the simply typed lambda calculus, for example to languages with co-products or tensor products, or to imperative languages without higher-order constructs.

The paper is organised as follows. In Sect. 2 we recall the definition of logical relation and a category theoretic formulation. In Sect. 3 we give our categorical notion of lax logical relation, proving a Basic Lemma, with a converse. In Sect. 4 we explain the relationship with pre-logical relations and in Sect. 5 give another syntax-based characterisation. In Sect. 6 we consider models in cartesian closed categories. In Sect. 7 we generalise our analysis to richer languages.