Theoretical Computer Science 604 (2015) 2–29

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Generic weakest precondition semantics from monads enriched with order ☆

Ichiro Hasuo

Department of Computer Science, The University of Tokyo, Hongo 7-3-1, Tokyo 113-8656, Japan

ARTICLE INFO

Article history:
Received 15 September 2014
Received in revised form 21 March 2015
Accepted 30 March 2015
Available online 3 April 2015

Keywords:
Program verification
Program logic
Precondition calculus
Nondeterminism
Game
Effect
Algebra
Coalgebra
Categorical model
Monad

ABSTRACT

We devise a generic framework where a weakest precondition semantics, in the form of indexed posets, is derived from a monad whose Kleisli category is enriched by posets. It is inspired by Jacobs' recent identification of a categorical structure that is common in various predicate transformers, but adds generality in the following aspects: (1) different notions of modality (such as "may" vs. "must") are captured by Eilenberg–Moore algebras; (2) nested alternating branching—like in games and in probabilistic systems with nondeterministic environments—is modularly modeled by a monad on the Eilenberg–Moore category of another.

1. Introduction

Among various styles of program semantics, the one by *predicate transformers* [2] is arguably the most intuitive. Its presentation is inherently logical, representing a program's behaviors by what properties (or *predicates*) hold before and after its execution. Predicate transformer semantics therefore form a basis of *program verification*, where specifications are given in the form of pre- and post-conditions [3]. It has also been used for *refinement* of specifications into programs (see e.g. [4]). Its success has driven extensions of the original nondeterministic framework, e.g. to the probabilistic one [5,6] and to the setting with both nondeterministic and probabilistic branching [7].

A categorical picture More recently, Jacobs in his series of papers [8–10] has pushed forward a categorical view on predicate transformers. It starts with a monad T that models a notion of branching. Then a program—henceforth called a *(branching) computation*—is a Kleisli arrow $X \to TY$; and the weakest precondition semantics is given as a contravariant functor $\mathbb{P}^{\mathcal{K}\ell} \colon \mathcal{K}\ell(T)^{\mathrm{op}} \to \mathbb{A}$, from the Kleisli category to the category $\mathbb{A}$ of suitable ordered algebras.

☆ An earlier version of this paper [1] has been presented at the Twelfth International Workshop on Coalgebraic Methods in Computer Science (CMCS 2014), 5–6 April 2014, Grenoble, France.

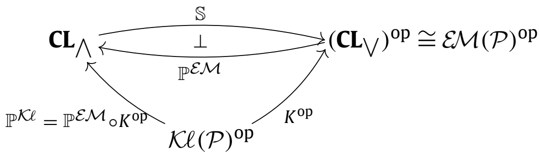
E-mail address: ichiro@is.s.u-tokyo.ac.jp.
URL: http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro.

http://dx.doi.org/10.1016/j.tcs.2015.03.047
0304-3975/

For example, in the basic nondeterministic setting, T is the powerset monad $\mathcal{P}$ on **Sets** and $\mathbb{A}$ is the category $\mathbf{CL}_{\bigwedge}$ of complete lattices and $\bigwedge$-preserving maps. The weakest precondition functor $\mathbb{P}^{\mathcal{K}\ell} : \mathcal{K}\ell(T)^{\mathrm{op}} \to \mathbf{CL}_{\bigwedge}$ then carries a function $f : X \to \mathcal{P}Y$ to

$$\mathrm{wpre}(f) \;:\; \mathcal{P}Y \longrightarrow \mathcal{P}X \;, \qquad Q \longmapsto \{x \in X \mid f(x) \subseteq Q\} \;. \tag{1}$$

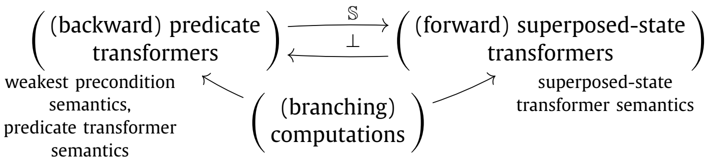
Moreover it can be seen that: 1) the functor $\mathbb{P}^{\mathcal{K}\ell}$ factors through the comparison functor $K : \mathcal{K}\ell(\mathcal{P}) \to \mathcal{EM}(\mathcal{P})$ to the Eilenberg–Moore category $\mathcal{EM}(\mathcal{P})$; and 2) the extended functor $\mathbb{P}^{\mathcal{EM}}$ has a dual adjoint $\mathbb{S}$. The situation is as follows.



(2)

Here the functor K carries $f : X \to \mathcal{P}Y$ to $f^{\dagger} : \mathcal{P}X \to \mathcal{P}Y$, $P \mapsto \bigcup_{x \in P} f(x)$. We shall call this mapping $f \mapsto f^{\dagger}$ a *superposed-state transformer semantics*—it can be understood as the *strongest postcondition semantics* in this specific instance of $T = \mathcal{P}$, but not necessarily in other instances. See Remark 2.11.

Therefore the picture (2)—understood as the one below—identifies a general categorical structure that underlies predicate transformer semantics. The dual adjunction here (which is in fact an isomorphism in the specific instance of (2)) indicates a “duality” between (backward) predicate transformers and (forward) superposed-state transformers.



(3)

Jacobs has identified other instances of (3) for: discrete probabilistic branching [8]; quantum logic [8]; and continuous probabilistic branching [9].[1] See [10] for an overview and also for additional instances. In all these instances the notion of *effect module*—originally from the study of quantum probability [11]—plays an essential role as algebras of “quantitative logics.”

Towards generic weakest precondition semantics In [8–10] the picture (3) is presented through examples, and its categorical axiomatics—that encompass many different instances of the picture—have not been pursued as a main goal.[2] Finding such axiomatics is the current paper's aim. In doing so, moreover, we acquire additional generality in two aspects: *different modalities* and *nested alternating branching*.

To motivate the first aspect of generality, observe that the weakest precondition semantics in (1) is the *must* semantics. The *may* variant looks as interesting; it would carry a postcondition $Q \subseteq Y$ to $\{x \in X \mid f(x) \cap Q \neq \emptyset\}$. The difference between the two semantics is much like the one between the modal operators $\Box$ and $\Diamond$.

On the second aspect, situations are abound in computer science where a computation involves two heterogeneous layers of branching. Typically these layers correspond to two distinct *players* with conflicting interests. Examples are *games*, a two-player version of automata which are essential tools in various topics including model-checking; and *probabilistic systems* where it is common to include nondeterministic branching too for modeling the environment's choices. Further details will be discussed later in Section 4.

Predicates and modalities from monads In this paper we present two categorical setups that are inspired by [12–14]—specifically by their use of $T1$ as a domain of *truth values* or *quantities*.

The first “one-player” setup is when we have only one layer of branching. Much like in [8–10] we start from a monad T. Assuming that T is *order-enriched*—in the sense that its Kleisli category $\mathcal{K}\ell(T)$ is **Posets**-enriched—we observe that:

- a natural notion of *truth value* arises from an object $T\Omega$ (where the object Ω is typically the terminal one 1);
- and a modality (like “may” and “must”) corresponds to a choice of an Eilenberg–Moore algebra $\tau : T(T\Omega) \to T\Omega$.

The required data set (T, Ω, τ) shall be called a *predicate transformer situation*. We prove that it induces a *weakest precondition semantics* functor $\mathcal{K}\ell(T)^{\mathrm{op}} \to$ **Posets**, and that it factors through $K : \mathcal{K}\ell(T) \to \mathcal{EM}(T)$, much like in (2). The general setup addresses common instances like the original nondeterministic one [2] and the probabilistic predicate transformers

[1] Different terminologies are used in [8] to describe the picture (3). See Remark 1.1.

[2] An exception is a unified treatment of branching weighted by a semiring R; see e.g. [10, §3]. This, however, does not generalize to the probabilistic branching as it is.