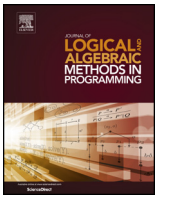
Journal of Logical and Algebraic Methods in Programming ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp



Abstract categorical semantics for resourceful functional reactive programming ☆

Wolfgang Jeltsch

Institute of Cybernetics at Tallinn University of Technology, Akadeemia tee 21, 12618 Tallinn, Estonia

ARTICLE INFO

Article history:
Received 22 May 2014
Received in revised form 30 June 2016
Accepted 1 July 2016
Available online xxxx

Keywords:
Functional reactive programming
Temporal logic
Linear logic
Logic of bunched implications
Categorical semantics

ABSTRACT

Functional reactive programming (FRP) makes it possible to express temporal aspects in a declarative way. Traditional approaches to FRP cannot handle objects like widgets in a graphical user interface or files in a file system. Therefore, programmers have to resort to ordinary methods of effectful programming when working with objects. In this paper, we devise a variant of FRP with support for objects, called "resourceful FRP", and develop an abstract categorical semantics for this FRP variant.

1. Introduction

Functional reactive programming (FRP) makes it possible to express the temporal aspects of software systems in a declarative way. Traditional approaches to FRP can deal with the purely functional parts of a program, but cannot handle objects like widgets in a graphical user interface (GUI) or files in a file system. Therefore, programmers have to resort to ordinary methods of effectful programming when working with objects. This is a particular problem when the set of existing objects changes throughout the run of a program. This happens, for example, in GUIs whose structure evolves over time, with windows being opened and closed, and widgets appearing and disappearing. In such a situation, it is not possible to implement the temporal behavior of objects and their interaction with pure computations in a declarative style.

In this paper, we extend FRP's declarative method for handling temporal phenomena to objects and develop a categorical semantics for the resulting FRP dialect. Our semantics is based on abstract process categories [7, APCs], which are categorical models of ordinary FRP, and on categorical models of the logic of bunched implications (BI) by O'Hearn and Pym [11, Section 3]. We call our new categorical models generalized abstract process categories (GAPCs) and show that they indeed generalize APCs. GAPCs can express instantaneous changes to the objects of a system as well as changes that happen gradually over time.

We first describe ordinary FRP in Section 2. Afterwards, we develop a declarative approach to object handling in Section 3 and integrate it with FRP in Section 4. In Sections 5 through 7, we describe abstract categorical semantics for the three programming paradigms presented in Sections 2 through 4, culminating in the definition of GAPCs. We present theorems about the relationship between APCs and GAPCs in Section 8. Finally, we discuss related work in Section 9 and give conclusions and an outlook on further work in Section 10.

☆ This article is the full version of an extended abstract presented at the 25th Nordic Workshop on Programming Theory (NWPT '13) in Tallinn, Estonia.

E-mail address: wolfgang@cs.ioc.ee.

http://dx.doi.org/10.1016/j.jlamp.2016.07.001
2352-2208/

Please cite this article in press as: W. Jeltsch, Abstract categorical semantics for resourceful functional reactive programming, J. Log. Algebraic Methods Program. (2016), http://dx.doi.org/10.1016/j.jlamp.2016.07.001

2. Functional reactive programming

Functional reactive programming (FRP) enables programmers to describe the temporal behavior of a program in a declarative style. It uses a linear notion of time; so times form a totally ordered set $(T, \leq)$. The FRP dialect we use in this paper puts no additional constraints on time scales. In particular, it is compatible with both continuous and discrete time.

The basic building blocks for behavioral descriptions in FRP are signals and events:

- A signal corresponds to a function that maps the elements of a time interval to values. For example, the behavior of an audio channel in a multimedia application can be described by a signal that maps each time during which the channel exists to the sound intensity that the channel carries at this time.
- An event corresponds to a pair of a time and a value. We say that an event corresponding to a pair (t, x) occurs at time t and carries the value x. For example, a key press can be described by an event that occurs when the key is pressed and carries the code of the key.

In Subsection 2.1, we develop a restricted FRP dialect that uses signals of a specific kind and events as the fundamental time-related values [5]. We extend this to our full FRP dialect in Subsection 2.2.

2.1. Basic FRP

The type system of our basic FRP dialect is based on the type constructors 1, $\times$, $\to$, 0, and $+$ from simply-typed λ-calculus. We add two new type constructors $\Box'$ and $\Diamond'$ to this set:

- An inhabitant of a type $\Box'\tau$ is a signal that maps each time $t' > t$ for some $t \in T$ to a value of type τ. We call t the start time of the signal.
- An inhabitant of a type $\Diamond'\tau$ is an event that carries a value of type τ.

We want to be able to constrain start times and occurrence times statically. Therefore, we introduce the concept of time-dependent type inhabitance, which says that it depends on the time whether a particular value inhabits a particular type. We write $x : \tau \ @ \ t$ to say that the value x inhabits the type τ at time t.[1] We establish the following rules:

- An inhabitant of a type $\Box'\tau$ at a time t is a signal that maps each time $t' > t$ to a value $x : \tau \ @ \ t'$.
- An inhabitant of a type $\Diamond'\tau$ at a time t is an event that occurs at a time $t' > t$ and carries a value $x : \tau \ @ \ t'$.

So a signal $s : \Box'\tau \ @ \ t$ is guaranteed to have t as its start time, and an event $e : \Diamond'\tau \ @ \ t$ is guaranteed to occur after t.

Note how the above rules fix inhabitance times of signal values and values carried by events. This leads to meaningful static constraints when nesting the type constructors $\Box'$ and $\Diamond'$. Following are some examples:

- The value of a signal $s : \Box'\Box'\tau$ at a time t is a signal that starts at t.
- An event $e : \Diamond'\Diamond'\tau$ that occurs at a time t carries an event that occurs after t.
- An event $e : \Diamond'\Box'\tau$ that occurs at a time t carries a signal that starts at t.

The traditional type constructors 1, $\times$, $\to$, 0, and $+$ naturally extend to time-dependent type inhabitance. For example, an inhabitant of a type $\tau_1 \times \tau_2$ at a time t is a pair (x, y) with $x : \tau_1 \ @ \ t$ and $y : \tau_2 \ @ \ t$, and an inhabitant of a type $\tau_1 + \tau_2$ at a time t is either $\iota_1(x)$ with $x : \tau_1 \ @ \ t$ or $\iota_2(y)$ with $y : \tau_2 \ @ \ t$.

A signal $s : \Box'\tau \ @ \ t$ does not assign a value to its inhabitance time t, and an event $e : \Diamond'\tau \ @ \ t$ cannot occur at its inhabitance time t. However, we can derive type constructors $\Box$ and $\Diamond$ that incorporate inhabitance times:

$$\Box\tau = \tau \times \Box'\tau \qquad \Diamond\tau = \tau + \Diamond'\tau \tag{1}$$

An inhabitant of $\Box\tau$ at a time t is a pair (x, s) with $x : \tau \ @ \ t$ and $s : \Box'\tau \ @ \ t$. We interpret such a pair as the signal that maps the inhabitance time t to x and otherwise maps times to values like s does. An inhabitant of $\Diamond\tau$ at a time t is either $\iota_1(x)$ with $x : \tau \ @ \ t$ or $\iota_2(e)$ with $e : \Diamond'\tau \ @ \ t$. We interpret $\iota_1(x)$ as the event that occurs at the inhabitance time t and carries x, and $\iota_2(e)$ as the event e, which occurs after the inhabitance time t.

A program in our FRP dialect is essentially an expression that describes the desired temporal behavior of the reactive system using signals and events. To allow programmers to build such expressions, we have to offer operations that work with inhabitants of $\Box'$ and $\Diamond'$. We do not discuss these operations here. In Subsection 2.2, we present our full FRP dialect, including its FRP-specific core operations.

The basic FRP dialect described in this subsection corresponds to an intuitionistic temporal logic via a Curry–Howard isomorphism. In particular, $\Box$ and $\Diamond$ correspond to "always" and "eventually" modalities, $\Box'$ and $\Diamond'$ correspond to future-

[1] We continue to write $x : \tau$ in cases where we do not care about the time of inhabitance; so we use $x : \tau$ as a synonym for $\exists t \in T \,.\, x : \tau \ @ \ t$.

Please cite this article in press as: W. Jeltsch, Abstract categorical semantics for resourceful functional reactive programming, J. Log. Algebraic Methods Program. (2016), http://dx.doi.org/10.1016/j.jlamp.2016.07.001