RIT Department of Computer Science
MSc Thesis Pre-Proposal:
Design and Implementation of a Push-Based Signal-Function FRP System

Edward Amsden

June 5, 2012

1 Problem Description

The signal function family of FRP semantics provides distinct advantages over the alternative signal/behavior family. These advantages include explicit inputs and the elimination of large classes of space and time leaks. However, there does not yet exist an implementation that provides ”push-based” semantics (i.e. evaluation on event occurrences, rather than polling for events) for signal functions.

Most of the work in Functional Reactive Programming centers around the construction and evaluation of FRP programs, and does not concern itself with the interface between FRP programs and the rest of the world. In fact, most literature on FRP ignores completely the problem of evaluation interface for FRP programs.

I intend to show that a monadic evaluation interface and signal vector typing enable a useful and efficient push-based FRP system.

2 Importance of Research

Functional reactive programming enables a denotative method of defining reactive systems. The efficient evaluation of such systems is still an open problem. Signal function style FRP semantics have several advantages over classic FRP semantics, including the elimination of a large class of space and time leaks, as well as explicit input. However, current implementations use “pull” based evaluation, leading to undesirable consequences: repeating unnecessary work in evaluating event non-occurrences, forcing input to be synchronous with output, and introducing sampling latency into event occurrences.

A signal function FRP system which uses “push”-based evaluation would eliminate these negatives, permitting more efficient implementation with only the latency of processing, rather than the sampling interval. It would also permit the decoupling of the input and output of the signal function network.

3 Related Work

FRP is a well-explored field. There is a wealth of research on the best semantics and most efficient implementation of FRP systems, though almost no discussion or exploration of the evaluation interface (the means of providing input and sampling output). I previously surveyed the literature regarding FRP for independent study credit [1]. The proposed research would provide an efficient implementation of an elegant family of FRP semantics, and a compositional and intuitive means of providing input to and using output from such systems.

Signal function FRP or Arrowized FRP (AFRP) is one of two main families of semantics for Functional Reactive Programming [4]. It was recently extended with additional typechecked properties using the notion of signal vectors [5].

Push based FRP has been demonstrated for classic FRP semantics [2], but it has not yet been explored in the context of signal function semantics.

4 Methodology

I will explore a semantic model of signal function FRP which considers the signal function network as state and events as mutators of the network. The network can of course evaluate events as well.

The exploration of this model will lead to the design of an implementation which permits the separation of evaluation of continuous values and discrete events. The implementation will hold the current signal function network as state in the monadic computation described using the evaluation monad. Event occurrences provided as input or produced within the network will mutate the network state.

The evaluation monad will provide monadic actions for "pushing" an event occurrence on an input to the signal function network, stepping time and evaluating continous values, and sampling continuous output values. The evaluation monad is a key component of the design. It permits much finer control over FRP evaluation that in possible with existing models and implementations, and enables the push-based evaluation of the signal function network by threading state.

Signal function networks will be described by a GADT using HLists [3] and phantom types (Event α and Signal α) to describe signal vectors.

Possibilities to be explored for implementation include "callbacks" to be activated on output event occurrences, or on time steps for continuous output, and a means of addressing multiple inputs or outputs to a signal function network.

The implementation will be compared against existing pull-based signal function FRP systems.

5 Potential Outcomes

I expect to produce an implementation of signal function semantics that utilizes push-based evaluation. I expect that this implementation will include both constructs for producing a signal function network and a monadic interface for evaluating a signal function network. The constructs will include routing combinators, switching combinators (which will introduce information denoting where mutations to the network by events will take place), and stateful signal functions such as numeric integration.

I expect to explore type-safe means of permitting multiple inputs and outputs.

References

[1] AMSDEN, E. A survey of functional reactive programming. `http://www.cs.rit.edu/~eca7215/frp-independent-study/Survey.pdf`, 2011.

[2] ELLIOTT, C. Push-pull functional reactive programming. In *Haskell Symposium* (2009).

[3] KISELYOV, O., LÄMMEL, R., AND SCHUPKE, K. Strongly typed heterogeneous collections. In *Proceedings of the 2004 ACM SIGPLAN workshop on Haskell* (New York, NY, USA, 2004), Haskell '04, ACM, pp. 96–107.

[4] NILSSON, H., COURTNEY, A., AND PETERSON, J. Functional reactive programming, continued. In *Proceedings of the 2002 ACM SIGPLAN workshop on Haskell* (New York, NY, USA, 2002), Haskell '02, ACM, pp. 51–64.

[5] SCHULTHORPE, N. *Towards Safe and Efficient Functional Reactive Programming.* PhD thesis, University of Nottingham, UK, 2011.