Streaming Functions in riff

Glyn Normington

January 17, 2019

This document provides a formal model of streaming functions in the riff FaaS project (`https://projectriff.io`).

Contents

1 Introduction

This document provides a formal model of streaming functions in the riff project.

2 Overview of this document

Event streaming is fraught with ambiguous terminology. For example, the word stream has multiple, overlapping connotations. This document attempts to model streaming functions precisely. The model is abstract in the sense that it does not capture all facets of streaming functions, but those it does capture, it captures precisely and unambiguously.

The Z specification language is used to capture the model, but sufficient English text is also provided that readers who do not know Z should be able to understand the model. The appendix contains a summary of the Z notation. For more information about Z, please consult the Z Manual (`https://www.cs.umd.edu/ mvz/handouts/z-manual.pdf`). The model was type checked using `fuzz` (`https://bitbucket.org/Spivey/fuzz`).

3 Fundamentals

Streams consist of messages, but for our current purposes, the details of a message are irrelevant. Similarly, errors are modelled but their details are not.

$$[Message, Error]$$

4 Reactive Streams

A reactive stream consist of an ordered series of zero or more messages optionally followed by a completion signal or an error.

$$\begin{aligned} RStream ::= \; & Next\langle\langle Message \times RStream\rangle\rangle \mid \\ & Complete \mid \\ & Failed\langle\langle Error\rangle\rangle \end{aligned}$$

5 Streaming Functions

A streaming function takes a reactive stream as input and produces another reactive stream as output.

$$StreamingFun == RStream \to RStream$$

Streaming functions may be chained together to produce another streaming function. The input stream of the chain is input to the first function in the chain. The output stream of the first function is the input stream of the second function and so forth. The output stream of the chain is the output stream of the last function in the chain.

$$\begin{array}{|l} Chain : \mathrm{seq}_1\, StreamingFun \to StreamingFun \\ \hline \forall f : StreamingFun;\ fs : \mathrm{seq}_1\, StreamingFun \bullet \\ \quad Chain(\langle f \rangle) = f \ \land \\ \quad Chain(\langle f \rangle \frown fs) = Chain(fs) \circ f \end{array}$$

A Z Notation

Numbers:

$\mathbb{N}$	Natural numbers {0,1,...}

Propositional logic and the schema calculus:

$\ldots \land \ldots$	And	$\langle\!\langle \ldots \rangle\!\rangle$	Free type injection
$\ldots \lor \ldots$	Or	$[\ldots]$	Given sets
$\ldots \Rightarrow \ldots$	Implies	$', ?, !_{0 \ldots 9}$	Schema decorations
$\forall .. \mid .. \bullet ..$	For all	$\ldots \vdash \ldots$	theorem
$\exists .. \mid .. \bullet ..$	There exists	$\theta \ldots$	Binding formation
$\ldots \setminus \ldots$	Hiding	$\lambda \ldots$	Function definition
$\ldots \mathrel{\widehat{=}} \ldots$	Schema definition	$\mu \ldots$	Mu-expression
$\ldots == \ldots$	Abbreviation	$\Delta \ldots$	State change
$\ldots ::= \ldots \mid \ldots$	Free type definition	$\Xi \ldots$	Invariant state change

Sets and sequences:

$\{\ldots\}$	Set	$\ldots \setminus \ldots$	Set difference
$\{.. \mid .. \bullet ..\}$	Set comprehension	$\bigcup \ldots$	Distributed union
$\mathbb{P} \ldots$	Set of subsets of	$\# \ldots$	Cardinality
$\varnothing$	Empty set	$\ldots \subseteq \ldots$	Subset
$\ldots \times \ldots$	Cartesian product	$\ldots \subset \ldots$	Proper subset
$\ldots \in \ldots$	Set membership	$\ldots$ partition $\ldots$	Set partition
$\ldots \notin \ldots$	Set non-membership	seq	Sequences
$\ldots \cup \ldots$	Union	$\langle \ldots \rangle$	Sequence
$\ldots \cap \ldots$	Intersection	disjoint $\ldots$	Disjoint sequence of sets

Functions and relations:

$\ldots \leftrightarrow \ldots$	Relation	$\ldots^*$	Reflexive-transitive
$\ldots \nrightarrow \ldots$	Partial function		closure
$\ldots \rightarrow \ldots$	Total function	$\ldots (\!\mid \ldots \mid\!)$	Relational image
$\ldots \rightarrowtail \ldots$	Partial injection	$\ldots \oplus \ldots$	Functional overriding
$\ldots \rightarrowtail \ldots$	Injection	$\ldots \lhd \ldots$	Domain restriction
$\mathrm{dom} \ldots$	Domain	$\ldots \rhd \ldots$	Range restriction
$\mathrm{ran} \ldots$	Range	$\ldots \lhd \ldots$	Domain subtraction
$\ldots \mapsto \ldots$	maplet	$\ldots \rhd \ldots$	Range subtraction
$\ldots^\sim$	Relational inverse		

Axiomatic descriptions:

Declarations

Predicates

Schema definitions:

SchemaName

Declaration

Predicates