Categorical Semantics for Functional Reactive Programming with Temporal Recursion and Corecursion

Wolfgang Jeltsch

TTÜ Küberneetika Instituut
Tallinn, Estonia

5th Workshop on
Mathematically Structured Functional Programming

12 April 2014

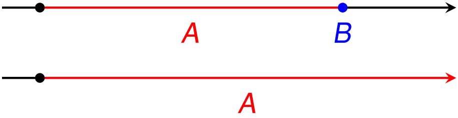
Functional Reactive Programming (FRP)

- programming paradigm for treating temporal aspects in a declarative fashion
- two key features:
  - time-dependent type membership
  - temporal type constructors
- Curry–Howard correspondence to temporal logic:
  - time-dependent trueness
  - temporal operators
- time:
  - linear
  - not necessarily discrete

Processes

- process consists of a continuous part and optionally a terminal event:

$A \rhd'' B$:



- different process types with different termination guarantees:
  - nontermination possible
  - termination guaranteed
  - termination guaranteed with upper bound on termination time

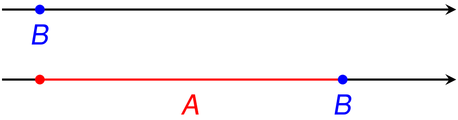
Processes that deal with the present

- processes that start immediately:

$A \rhd' B$:

- processes that may terminate immediately:

$A \rhd B$:



- $\rhd'$ and $\rhd$ definable in terms of $\rhd''$:

$$A \rhd' B = A \times A \rhd'' B \qquad A \rhd B = B + A \rhd' B$$

Abstract process categories (APCs)

- cartesian closed category $\mathcal{C}$ with coproducts
- functors that model process type constructors:

$$\triangleright'' : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$$

- natural transformations that model FRP operations:
  - ideal monads
  - ideal comonads
  - further structure (not in this talk)

Ideal monads

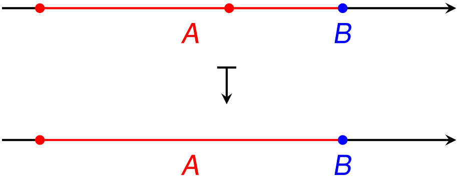
- each $A \rhd' -$ is an ideal monad:

$$\mu'_B : A \rhd' (A \rhd B) \to A \rhd' B$$

- concatenation of a continuous part with a follow-up process:

$A \rhd' (A \rhd B)$:

$A \rhd' B$:



Ideal comonads

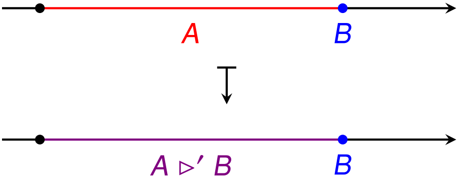
- each $- \rhd'' B$ is an ideal comonad:

$$\delta'_A : A \rhd'' B \to (A \rhd' B) \rhd'' B$$

- generation of a continuous part of shorter and shorter suffixes:

$A \rhd'' B$:

$(A \rhd' B) \rhd'' B$:

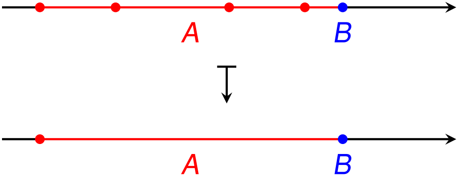


Iteration of ideal multiplication and comultiplication

- iterated concatenation via induction:

$\mu C\ .\ A \rhd' (B + C)$:

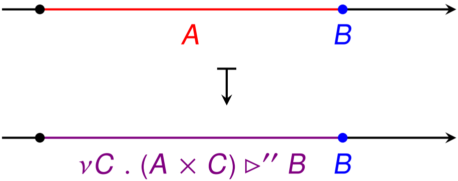
$A \rhd' B$:



- iterated suffix generation via coinduction:

$A \rhd'' B$:

$\nu C\ .\ (A \times C) \rhd'' B$:

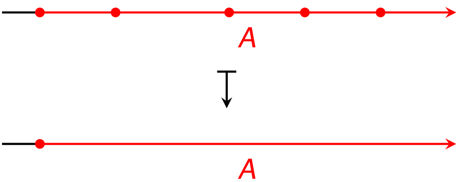


Wanted: Stronger variants of these iterations

- sequence of continuous parts may be infinite:

$\nu C \,.\, A \rhd' (B + C)$:

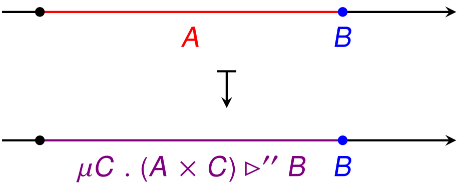
$A \rhd' B$:



- nesting depth must be finite:

$A \rhd'' B$:

$\mu C \,.\, (A \times C) \rhd'' B$:



Solution: Extending the ideal monad and comonad structure

- each $A \rhd' -$ is a completely iterative monad:

$$\frac{f : C \to A \rhd' (B + C)}{f^\infty : C \to A \rhd' B}$$

- each $- \rhd'' B$ is a recursive comonad:

$$\frac{f : (A \times C) \rhd'' B \to C}{f^* : A \rhd'' B \to C}$$

Are these extensions reasonable?

- check whether there are nontrivial instances of APCs that have the additional structure
- concrete process categories (CPCs) are instances of APCs
- do they have the required additional structure?

Concrete process categories

- make times explicit:
  - time scale can be any totally ordered set
- express causality of operations:
  - the prefix of a result that ends at a time t can only depend on the prefix of the argument that ends at t
  - operations expressed as families of prefix transformations, one for each t
- process types with simple termination guarantee cannot be modeled:
  - termination is a liveness property
  - only safety properties can be expressed, because only prefixes are considered
- the following process types can be modeled:

$\rhd_\infty$ nontermination possible

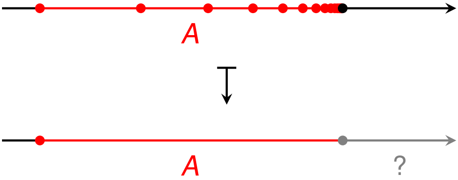
$\rhd_{t_b}$ termination at or before t_b guaranteed

A constraint on time scales

- infinitely many concatenations can be problematic:

$\nu C \,.\, A \rhd' (B + C)$:

$A \rhd' B$:



- analogous problem for suffix generation
- solution is to disallow “pathological” time scales:
  - every ascending sequence of times must be unbounded
  - Achilles catches up with the Tortoise
  - certain “interesting” time scales still allowed:

$$\{\, z + 1/n \mid z \in \mathbb{Z} \wedge n \in \mathbb{N} \setminus \{0\} \,\}$$

Compatibility with different termination (non)guarantees

- completely iterative monad:
  - $\triangleright_\infty$ infinitely many concatenations are no problem
  - $\triangleright_{t_b}$ only finitely many concatenations can occur, since all subprocesses terminate at or before t_b
- recursive comonad:
  - $\triangleright_{t_b}$ nesting depth is finite, since given process terminates
  - $\triangleright_\infty$ nesting depth is finite, since only finite prefixes of processes are considered

Categorical Semantics for Functional Reactive Programming with Temporal Recursion and Corecursion

Wolfgang Jeltsch

TTÜ Küberneetika Instituut
Tallinn, Estonia

5th Workshop on
Mathematically Structured Functional Programming

12 April 2014