An Abstract Categorical Semantics for Functional Reactive Programming with Processes

Wolfgang Jeltsch

TTÜ Küberneetika Instituut
Tallinn, Estonia

8th Workshop on
Programming Languages Meets Program Verification

21 January 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

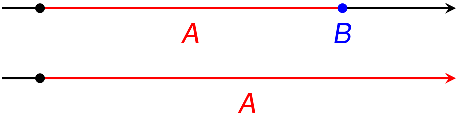
Processes

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

$A \rhd'' B$:

- weak process types allow for nonterminating processes:

$A \blacktriangleright'' B$:



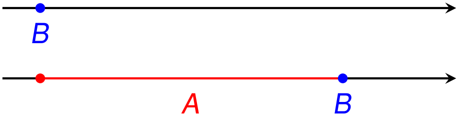
Processes that deal with the present

- processes that start immediately:

$A \rhd' B$:

- processes that may terminate immediately:

$A \rhd B$:



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

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

- weak variants $\blacktriangleright'$ and $\blacktriangleright$ can be defined analogously

Topic of this talk

- axiomatically defined categorical semantics for FRP with processes
- road to this semantics:
  1. categorical models of intuitionistic S4
     (Kobayashi 1997; Bierman and de Paiva 2000)
  2. temporal categories for FRP with behaviors and events
     (Jeltsch 2012)
  3. abstract process categories for FRP with processes
     (this talk)

Basic structure

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

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

- unified functor for modeling process type constructors:

$$- \triangleright''_{-} - : \mathbf{2} \times \mathcal{C} \times \mathcal{C} \to \mathcal{C}$$

- traditional functors by specialization:

$$\begin{aligned} \triangleright'' &= \triangleright''_0 \\ \blacktriangleright'' &= \triangleright''_1 \end{aligned}$$

- weakening as mapping:

$$\frac{w : 0 \to 1}{\triangleright''_w : \triangleright'' \to \blacktriangleright''}$$

Process expansion

- generates a continuous part of shorter and shorter suffixes
- three kinds of structures:
  - comonads:

$$\pi_1 : A \rhd'_W B \to A$$

$$\theta'_{A,W,B} : A \rhd'_W B \to \left(A \rhd'_W B\right) \rhd'_W B$$

  - ideal comonads:

$$\theta''_{A,W,B} : A \rhd''_W B \to \left(A \rhd'_W B\right) \rhd''_W B$$

  - “real comonads:”

$$\theta_{A,W,B} : A \rhd_W B \to \left(A \rhd'_W B\right) \rhd_W B$$

- derivation:

ideal comonads → comonads → “real comonads”

Process joining

- concatenates a continuous part with a follow-up process
- three kinds of structures:
  - monads:

$$\iota_1 : B \to A \rhd_W B$$
$$\vartheta_{A,W,B} : A \rhd_W (A \rhd_W B) \to A \rhd_W B$$

  - ideal monads:

$$\vartheta'_{A,W,B} : A \rhd'_W (A \rhd_W B) \to A \rhd'_W B$$

  - "gorgeous monads:"

$$\vartheta''_{A,W,B} : A \rhd''_W (A \rhd_W B) \to A \rhd''_W B$$

- derivation:

"gorgeous monads" → ideal monads → monads

Process merging and the canonical nonterminating process

- merging of two processes:

$$A_1 \rhd''_{W_1} B_1 \times A_2 \rhd''_{W_2} B_2$$
$$\downarrow$$
$$(A_1 \times A_2) \rhd''_{W_1 \times W_2} ((A_1, W_1, B_1) \odot (A_2, W_2, B_2))$$

- definition of $\odot$:

$$(A_1, W_1, B_1) \odot (A_2, W_2, B_2)$$
$$=$$
$$\left(B_1 \times B_2\right) + \left(B_1 \times A_2 \rhd'_{W_2} B_2\right) + \left(A_1 \rhd'_{W_1} B_1 \times B_2\right)$$

- $W_1 \times W_2$ is minimum of W_1 and W_2
- nullary version constructs the canonical nonterminating process:

$$1 \rightarrow 1 \rhd''_1 0$$

Process merging and the canonical nonterminating process

- merging of two processes:

$$A_1 \rhd''_{W_1} B_1 \times A_2 \rhd''_{W_2} B_2$$
$$\downarrow$$
$$(A_1 \times A_2) \rhd''_{W_1 \times W_2} ((A_1, W_1, B_1) \odot (A_2, W_2, B_2))$$

- definition of $\odot$:

$$(A_1, W_1, B_1) \odot (A_2, W_2, B_2)$$
$$=$$
$$\left(B_1 \times B_2\right) + \left(B_1 \times A_2 \rhd'_{W_2} B_2\right) + \left(A_1 \rhd'_{W_1} B_1 \times B_2\right)$$

- $W_1 \times W_2$ is minimum of W_1 and W_2
- nullary version constructs the canonical nonterminating process:

$$1 \to 1 \rhd''_1 0$$

Process merging and the canonical nonterminating process

- merging of two processes:

$$A_1 \rhd''_{W_1} B_1 \times A_2 \rhd''_{W_2} B_2$$
$$\downarrow$$
$$(A_1 \times A_2) \rhd''_{W_1 \times W_2} ((A_1, W_1, B_1) \odot (A_2, W_2, B_2))$$

- definition of $\odot$:

$$(A_1, W_1, B_1) \odot (A_2, W_2, B_2)$$
$$=$$
$$\left(B_1 \times B_2\right) + \left(B_1 \times A_2 \rhd'_{W_2} B_2\right) + \left(A_1 \rhd'_{W_1} B_1 \times B_2\right)$$

- $W_1 \times W_2$ is minimum of W_1 and W_2
- nullary version constructs the canonical nonterminating process:

$$1 \rightarrow 1 \rhd''_1 0$$

Process merging and the canonical nonterminating process

- merging of two processes:

$$A_1 \rhd''_{W_1} B_1 \times A_2 \rhd''_{W_2} B_2$$
$$\downarrow$$
$$(A_1 \times A_2) \rhd''_{W_1 \times W_2} ((A_1, W_1, B_1) \odot (A_2, W_2, B_2))$$

- definition of $\odot$:

$$(A_1, W_1, B_1) \odot (A_2, W_2, B_2)$$
$$=$$
$$\left(B_1 \times B_2\right) + \left(B_1 \times A_2 \rhd'_{W_2} B_2\right) + \left(A_1 \rhd'_{W_1} B_1 \times B_2\right)$$

- $W_1 \times W_2$ is minimum of W_1 and W_2
- nullary version constructs the canonical nonterminating process:

$$1 \rightarrow 1 \rhd''_1 0$$

Process merging and the canonical nonterminating process

- merging of two processes:

$$A_1 \rhd''_{W_1} B_1 \times A_2 \rhd''_{W_2} B_2$$
$$\downarrow$$
$$(A_1 \times A_2) \rhd''_{W_1 \times W_2} ((A_1, W_1, B_1) \odot (A_2, W_2, B_2))$$

- definition of $\odot$:

$$(A_1, W_1, B_1) \odot (A_2, W_2, B_2)$$
$$=$$
$$\left(B_1 \times B_2\right) + \left(B_1 \times A_2 \rhd'_{W_2} B_2\right) + \left(A_1 \rhd'_{W_1} B_1 \times B_2\right)$$

- $W_1 \times W_2$ is minimum of W_1 and W_2
- nullary version constructs the canonical nonterminating process:

$$1 \to 1 \rhd''_1 0$$

Conclusions and further work

- developed abstract process categories (APCs):
  - categorical semantics for FRP with processes
  - axiomatically defined
  - generalize temporal categories (Jeltsch 2012)
- further work:
  - recursion and corecursion on processes
  - mutable state in FRP
  - FRP implementation with API inspired by (extended versions of) APCs

References

Satoshi Kobayashi.
Monad as Modality.
Theoretical Computer Science, 175 (1):29–74, 1997.

Gavin Bierman and Valeria de Paiva.
On an Intuitionistic Modal Logic.
Studia Logica, 65 (3):383–416, 2000.

Wolfgang Jeltsch.
Towards a Common Categorical Semantics for Linear-Time Temporal Logic and Functional Reactive Programming.
Electronic Notes in Theoretical Computer Science, 286:229–242, 2012.