Expressing Causality in Categorical Models of Functional Reactive Programming

Wolfgang Jeltsch

TTÜ Küberneetika Instituut
Tallinn, Estonia

Joint Estonian–Latvian Theory Days at Medzābaki

29 September 2012

Functional reactive programming

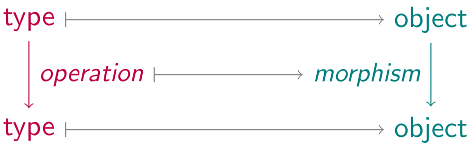
- extension of functional programming
- supports description of temporal behavior
- two key concepts:
  - computing with time-varying values and events
  - time-dependent type membership
- new type constructors:

  □ time-varying values

  ◇ events

Categorical models of simply typed programming calculus

- models are cartesian closed categories with coproducts (CCCCs)
- use of basic category structure:



- use of CCCC structure:

product type	$\tau_1 \times \tau_2$	$\longmapsto$	$A \times B$	product
sum type	$\tau_1 + \tau_2$	$\longmapsto$	$A + B$	coproduct
function type	$\tau_1 \to \tau_2$	$\longmapsto$	B^A	exponential
unit type	1	$\longmapsto$	1	terminal object
empty type	0	$\longmapsto$	0	initial object

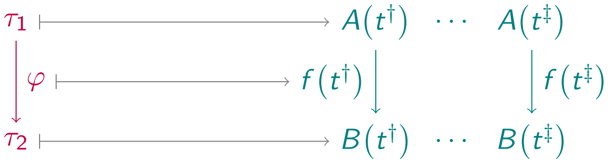
Categorical models of FRP

- ingredients:

  totally ordered set $(T, \leqslant)$ time scale

  CCCC $\mathcal{B}$ simple types and functions

- product category $\mathcal{B}^T$ models FRP types and operations with indices denoting inhabitation times:



Meanings of FRP type constructors

- general picture:

$$\begin{array}{rcl} \text{simple type constructors} & \longmapsto & \text{CCCC structure of } \mathcal{B}^T \\ \text{type constructors } \Box \text{ and } \Diamond & \longmapsto & \text{functors } \Box \text{ and } \Diamond \end{array}$$

- CCCC structure of $\mathcal{B}^T$ from CCCC structure of $\mathcal{B}$ with operations working pointwise
- functors $\Box$ and $\Diamond$ defined as follows:

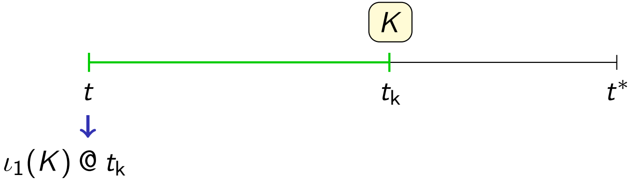
$$(\Box A)(t) = \prod_{t' \geqslant t} A(t')$$

$$(\Diamond A)(t) = \coprod_{t' \geqslant t} A(t')$$

An example program component

- looks for the next key press up to a certain timeout
- emits a value of type $\Diamond(\mathsf{Key} + 1)$ when it starts:

Case 1 key press before timeout:



Case 2 no key press before timeout:

A noncausal operation

- hypothetical polymorphic operation d from $\Diamond(\tau_1 + \tau_2)$ to $\Diamond\tau_1 + \Diamond\tau_2$:

$$\iota_1(x) @ t' \mapsto \iota_1(x @ t')$$
$$\iota_2(y) @ t' \mapsto \iota_2(y @ t')$$

- applying d to the output of the key press listener gives value of type $\Diamond\mathrm{Key} + \Diamond 1$:

  key press before timeout $\iota_1(K @ t_\mathrm{k})$
  no key press before timeout $\iota_2(\mathrm{tt} @ t^*)$

- tells us immediately if the user will press a key before the timeout
- so d cannot exist

Semantics allow for noncausal operations

- polymorphic operations from $\Diamond(\tau_1 + \tau_2)$ to $\Diamond\tau_1 + \Diamond\tau_2$ modeled by natural transformations τ with

$$\tau_{A,B} : \Diamond(A + B) \to \Diamond A + \Diamond B$$

- there is such a τ (which is even an isomorphism):

$$\coprod_{t' \geqslant t} \Big(A(t') + B(t') \Big) \cong \coprod_{t' \geqslant t} A(t') + \coprod_{t' \geqslant t} B(t')$$

- reason:

  semantics do not deal with time-dependent knowledge about values

Knowledge-aware semantics

- replace category $\mathcal{B}^T$ by category $\mathcal{B}^I$ where

$$I = \{(t, t_o) \in T \times T \mid t \leqslant t_o\}$$

- dealing with knowledge at t_o:

knowledge type $\longmapsto A(t, t_o)$

↓ *knowledge transformation* $\longmapsto f(t, t_o)$ ↓

knowledge type $\longmapsto B(t, t_o)$

- functors $\Box$ and $\Diamond$ defined as follows:

$$(\Box A)(t, t_o) = \prod_{t' \in [t, t_o]} A(t', t_o)$$

$$(\Diamond A)(t, t_o) = \coprod_{t' \in [t, t_o]} A(t', t_o) + 1$$

Compatibility of knowledge transformations

- knowledge transformations may be incompatible
- extend set I to category $\mathcal{I}$ by adding morphisms

$$(t, t_o, t_o') : (t, t_o') \to (t, t_o)$$

for $t \leqslant t_o \leqslant t_o'$

- replace product category $\mathcal{B}^I$ by functor category $\mathcal{B}^{\mathcal{I}}$
- objects $A(t, t_o, t_o')$ model knowledge reduction
- morphisms of $\mathcal{B}^{\mathcal{I}}$ are natural transformations
- means that knowledge transformations are compatible:

$$\begin{array}{ccc}
A(t, t_o) & \xleftarrow{\quad A(t, t_o, t_o') \quad} & A(t, t_o') \\
{\scriptstyle f_{(t,t_o)}} \downarrow & & \downarrow {\scriptstyle f_{(t,t_o')}} \\
B(t, t_o) & \xleftarrow[\quad B(t, t_o, t_o') \quad]{} & B(t, t_o')
\end{array}$$

Upper bounds for occurrence times

- definition of functor $\Diamond$ allows for never occurring events:

$$(\Diamond A)(t, t_o) = \coprod_{t' \in [t, t_o]} A(t', t_o) + 1$$

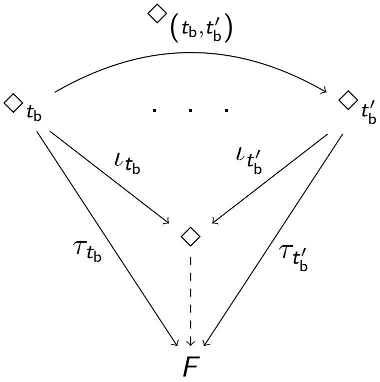
- introduction of new functor $\Diamond_{\_} : \mathcal{T} \to \left(\mathcal{B}^{\mathcal{I}}\right)^{\mathcal{B}^{\mathcal{I}}}$ where $\mathcal{T}$ is the category of $(T, \leqslant)$
- $\Diamond_{t_b}$ models an event type constructor with upper bound t_b for occurrence times:

$$(\Diamond_{t_b} A)(t, t_o) = \begin{cases} 0 & \text{if } t_b < t \\ \coprod_{t' \in [t, t_b]} A(t', t_o) & \text{if } t \leqslant t_b \leqslant t_o \\ \coprod_{t' \in [t, t_o]} A(t', t_o) + 1 & \text{it } t_o < t_b \end{cases}$$

- $\Diamond_{(t_b, t'_b)}$ models type conversion

Meaning of event type constructor

- type constructor $\diamond$ is the least upper bound of all $\diamond_{t_b}$-constructors
- functor $\diamond$ must be a colimit of the functor $\diamond_{\_}$:



The shape of the $\Diamond$-functor

Theorem

If $(T, \leqslant)$ has a maximum t_{max}, then $\Diamond \cong \Diamond_{t_{\mathrm{max}}}$.

Theorem

If $(T, \leqslant)$ has no maximum, $\Diamond$ models an event type constructor that allows for never occurring events.

Causality ensured

Theorem

There are categorical models that do not contain any natural transformation τ with

$$\tau_{A,B} : \Diamond(A + B) \to \Diamond A + \Diamond B \quad .$$

Conclusions

- categorical models of FRP that express causality by reflecting time-dependancy of knowledge
- liveness not expressed under certain conditions
- ultimate goal is an axiomatic semantics with the following properties:
  - expresses causality
  - expresses liveness constraint of $\Diamond$
  - covers the categorical semantics of this talk as a special case
  - models process type constructors, which are a generalization of $\Box$ and $\Diamond$