Sequential products for effects

Dominique Duval
LJK, University of Grenoble, France
joint work with *Jean-Guillaume Dumas* and *Jean-Claude Reynaud*

4th ACCAT Workshop — York — March 22., 2009

Outline

Motivation

In a categorical semantics for a programming language,
the construction of terms is interpreted
by composition and products.

When the language has side-effects, this has to be adapted.
One major issue is that the value of a term $f(t_1, \ldots, t_n)$
may depend on the order of evaluation of its arguments.

The aim of this talk is to present a new framework
and to compare it to existing ones.

Categorical semantics

Language	Category
sort	object
operation:	morphism:
$f : X_1, \ldots, X_n \to Y$	$f : X_1 \times \cdots \times X_n \to Y$
term construction:	composition and tuple:
$f(t_1, \ldots, t_n)$	$f \circ \langle t_1, \ldots, t_n \rangle$

$$\begin{tikzcd}
 & Y_1 & \\
X \arrow[ur, "t_1"] \arrow[r, "\langle t_1, t_2 \rangle"] \arrow[dr, "t_2"'] & Y_1 \times Y_2 \arrow[u] \arrow[d] \arrow[r, "f"] \arrow[ul, phantom, "=", near start] \arrow[dl, phantom, "=", near start] & Z \\
 & Y_2 &
\end{tikzcd}$$

The product functor

Binary products on C define a functor $\boxed{\times : C^2 \to C}$:

- On objects: $X_1 \times X_2$,
  with projections $p_i : X_1 \times X_2 \to X_i$.
- On morphims: $f_1 \times f_2 : X_1 \times X_2 \to Y_1 \times Y_2$,
  defined as $f_1 \times f_2 = \langle f_1 \circ p_1, f_2 \circ p_2 \rangle$,
  i.e., characterized by:

$$\begin{tikzcd}
X_1 \arrow[r, "f_1"] \arrow[dr, phantom, "="] & Y_1 \\
X_1 \times X_2 \arrow[u] \arrow[d] \arrow[r, "f_1 \times f_2"] \arrow[dr, phantom, "="] & Y_1 \times Y_2 \arrow[u] \arrow[d] \\
X_2 \arrow[r, "f_2"] & Y_2
\end{tikzcd}$$

Computational effects

Without effects, an operation symbol $f : X \to Y$
stands for a (total) function $f : X \to Y$.

With effects, an operation symbol $f : X \to Y$
stands for "something else", e.g.:

- Partiality: a partial function $f : X \rightharpoonup Y$,
- State: a function $f : S \times X \to S \times Y$
- Non-determinism: a function $f : X \to \mathcal{L}(Y)$
- and so on...

What about term construction?
I.e., what about composition and products?

Frameworks for effects

Several frameworks, quite “similar” [Haskell]:

- Strong monads [Moggi’89]
- Premonoidal categories [Power&Robinson’97]
- Arrows [Hughes’00]

Our framework is more “restricted” and more “homogeneous”:

- Cartesian effect categories [Dumas&Duval&Reynaud’07,’09].

Homogeneity

K is a category, C is a wide subcategory of K:

$$C \subseteqq K$$

Freyd-category:

C	K
cartesian $\Downarrow$ monoidal	premonoidal

Cartesian effect category:

C	K
cartesian $\Downarrow$ monoidal	"sequential cartesian" $\Downarrow$ premonoidal

Our result, in short

The universal property for the product $f \times v$:

$$\begin{tikzcd}
X_1 \arrow[r, "f"] & Y_1 \\
X_1 \times X_2 \arrow[u] \arrow[d] \arrow[r, "f \times v"] \arrow[u, phantom, shift left=8ex, "="] \arrow[d, phantom, shift left=8ex, "="] & Y_1 \times Y_2 \arrow[u] \arrow[d] \\
X_2 \arrow[r, "v"] & Y_2
\end{tikzcd}$$

has to be "decorated":

$$\begin{tikzcd}
X_1 \arrow[r, "f"] & Y_1 \\
X_1 \times X_2 \arrow[u, squiggly] \arrow[d, squiggly] \arrow[r, "f \rtimes v"] \arrow[u, phantom, shift left=8ex, "="] \arrow[d, phantom, shift left=8ex, "\geq"] & Y_1 \times X_2 \arrow[u, squiggly] \arrow[d, squiggly] \\
X_2 \arrow[r, squiggly, "v"] & X_2
\end{tikzcd}$$

The aim of this talk is to explain what this means.

Example: partiality

f is partial, $v = \mathrm{id}$ is total,
$\leq$ is the usual order on partial functions.

Let $f \rtimes \mathrm{id}$ be such that:

$$\begin{tikzcd}
X_1 \arrow[rr, "f"] & & Y_1 \\
X_1 \times X_2 \arrow[u, squiggly] \arrow[d, squiggly] \arrow[rr, "f \rtimes \mathrm{id}"] \arrow[urr, phantom, "="] & & Y_1 \times Y_2 \arrow[u, squiggly] \arrow[d, squiggly] \\
X_2 \arrow[rr, squiggly, "\mathrm{id}"] \arrow[urr, phantom, "\geq"] & & Y_2
\end{tikzcd}$$

then $f \rtimes \mathrm{id}$ is the partial function:

$$\begin{cases} \mathcal{D}(f \rtimes \mathrm{id}) = \{(x_1, x_2) \mid x_1 \in \mathcal{D}(f)\} \text{ and} \\ \forall (x_1, x_2) \in \mathcal{D}(f \rtimes \mathrm{id})\,,\ (f \rtimes \mathrm{id})(x_1, x_2) = (f(x_1), x_2) \end{cases}$$

Two questions

- What is an effect?
  $\longrightarrow$ effect categories.

- What is a sequential product?
  $\longrightarrow$ cartesian effect categories.

Outline

Pure vs. general morphisms

K is a category, C is a wide subcategory of K:

$$C \subseteq K$$

- (General) morphisms $f : X \to Y$ in K,
- pure morphisms $v : X \rightsquigarrow Y$ in C.

Example. $\mathcal{S}et \subseteq \mathcal{P}art$

a morphism $f : X \to Y$ is a partial function,
a pure morphism $v : X \rightsquigarrow Y$ is a total function.

Effects

The effect of $f : X \to Y$ should provide a measure of the "distance" from f to pure functions.

Let 1 be a terminal object in C:

$$\text{for all } X \text{ there is a unique } \langle\,\rangle_X : X \rightsquigarrow 1$$

The effect of $f : X \to Y$ is $\langle\,\rangle_Y \circ f : X \to 1$.

$f : X \to Y$ is effect-free if $\langle\,\rangle_Y \circ f = \langle\,\rangle_X$.
Hence, every pure morphism is effect-free.

Example. $\mathcal{S}et \subseteq \mathcal{P}art$

$1 = \{*\}$ (a singleton).
The effect of f is $\langle\,\rangle \circ f : X \to \{*\}$, such that $\mathcal{D}(\langle\,\rangle \circ f) = \mathcal{D}(f)$.

Same-effect equivalence

Let $\boxed{\approx}$ be the relation between morphisms such that
for all $f : X \to Y$ and $f' : X \to Y'$,
$f \approx f'$ if and only if f and f' have the same effect, i.e.

$$f \approx f' \iff \langle\,\rangle \circ f = \langle\,\rangle \circ f'$$

Example. $\mathcal{S}et \subseteq \mathcal{P}art$

$$f \approx f' \iff \mathcal{D}(f) = \mathcal{D}(f')\ .$$

Symmetric up-to-effects consistency

Let $\boxed{\smile}$ be a relation between parallel morphisms that satisfies:

- reflexivity, symmetry,
- substitution: $g \leq g' \implies g \circ f \leq g' \circ f$
- pure replacement: $f \leq f' \implies w \circ f \leq w \circ f'$ when w is pure.
- complementarity wrt $\approx$: for all $f, f' : X \to Y$,

$$f \approx f' \text{ and } f \smile f' \implies f = f'$$

Example. $\mathcal{S}et \subseteqq \mathcal{P}art$

$$f \smile f' \iff f = f' \text{ on } \mathcal{D}(f) \cap \mathcal{D}(f') \,.$$

Transitive up-to-effects consistency

Let $\boxed{\leq}$ be a relation between parallel morphisms that satisfies:

- reflexivity, transitivity,
- substitution, pure replacement,
- complementarity wrt $\approx$: for all $f, f', f'' : X \to Y$,

$$f \approx f' \text{ and } f \leq f'' \text{ and } f' \leq f'' \implies f = f'$$

Let $f \smile f' \iff \exists f''\ ,\ f \leq f'' \text{ and } f' \leq f''$.
Then $\smile$ is a symmetric up-to-effects consistency.

Example. $\mathcal{S}et \subseteq \mathcal{P}art$

$$f \leq f' \iff \mathcal{D}(f) \subseteq \mathcal{D}(f') \text{ and } f = f' \text{ on } \mathcal{D}(f)$$

Hence, three relations

- Same-effect equivalence $\boxed{f \approx f'}$:

$$f \approx f' \iff \langle\,\rangle \circ f = \langle\,\rangle \circ f'$$

- Symmetric up-to-effects consistency $\boxed{f \smile f'}$:

$$f \approx f' \text{ and } f \smile f' \implies f = f'$$

- Transitive up-to-effects consistency $\boxed{f \leq f'}$:

$$f \smile f' \iff \exists f''\ f \leq f'' \text{ and } f' \leq f''$$

Example. $\mathcal{S}et \subseteq \mathcal{P}art$

$$\begin{cases} f \approx f' & \iff & \mathcal{D}(f) = \mathcal{D}(f') \\ f \smile f' & \iff & f = f' \text{ on } \mathcal{D}(f) \cap \mathcal{D}(f') \\ f \leq f' & \iff & \mathcal{D}(f) \subseteq \mathcal{D}(f') \text{ and } f = f' \text{ on } \mathcal{D}(f) \end{cases}$$

Effect categories

Definition.
An effect category is $C \subseteq K$ with
a transitive up-to-effects consistency $\boxed{\leq}$,
i.e., a relation between parallel morphisms that satisfies:

- reflexivity, transitivity,
- substitution, pure replacement,
- equality on pure morphisms.

Example. $\mathcal{S}et \subseteq \mathcal{P}art$

$$f \leq f' \iff \mathcal{D}(f) \subseteq \mathcal{D}(f') \text{ and } f = f' \text{ on } \mathcal{D}(f)$$

Outline

Semi-pure products

Let $C \subseteq K$ with $\leq$ be an effect category,
with a binary product $\times$ on C.

Definition.
The left semi-pure product $v_1 \ltimes f_2$
and the right semi-pure product $f_1 \rtimes v_2$
are characterized by:

$$
\begin{tikzcd}
X_1 \arrow[rr, rightsquigarrow, "v_1"] & & Y_1 \\
& \geq & \\
X_1 \times X_2 \arrow[uu, rightsquigarrow] \arrow[dd, rightsquigarrow] \arrow[rr, "v_1 \ltimes f_2"] & & Y_1 \times Y_2 \arrow[uu, rightsquigarrow] \arrow[dd, rightsquigarrow] \\
& = & \\
X_2 \arrow[rr, "f_2"] & & Y_2
\end{tikzcd}
\qquad
\begin{tikzcd}
X_1 \arrow[rr, "f_1"] & & Y_1 \\
& = & \\
X_1 \times X_2 \arrow[uu, rightsquigarrow] \arrow[dd, rightsquigarrow] \arrow[rr, "f_1 \rtimes v_2"] & & Y_1 \times Y_2 \arrow[uu, rightsquigarrow] \arrow[dd, rightsquigarrow] \\
& \geq & \\
X_2 \arrow[rr, rightsquigarrow, "v_2"] & & Y_2
\end{tikzcd}
$$

Sequential products

Definition.
The left sequential product $f_1 \ltimes f_2$ is defined as:

$$f_1 \ltimes f_2 = (\mathrm{id}_{Y_1} \ltimes f_2) \circ (f_1 \rtimes \mathrm{id}_{X_2}) \qquad \text{“}\mathit{first}\ f_1,\ \mathit{then}\ f_2\text{”}$$

$$\begin{tikzcd}
X_1 \arrow[r, "f_1"] \arrow[dr, phantom, "="] & Y_1 \arrow[r, rightsquigarrow, "\mathrm{id}"] \arrow[dr, phantom, "\geq"] & Y_1 \\
X_1 \times X_2 \arrow[u, rightsquigarrow] \arrow[d, rightsquigarrow] \arrow[r, "f_1 \rtimes \mathrm{id}"] \arrow[dr, phantom, "\geq"] & Y_1 \times X_2 \arrow[u, rightsquigarrow] \arrow[d, rightsquigarrow] \arrow[r, "\mathrm{id} \ltimes f_2"] \arrow[dr, phantom, "="] & Y_1 \times Y_2 \arrow[u, rightsquigarrow] \arrow[d, rightsquigarrow] \\
X_2 \arrow[r, rightsquigarrow, "\mathrm{id}"] & X_2 \arrow[r, "f_2"] & Y_2
\end{tikzcd}$$

and symmetrically for the right sequential product $f_1 \rtimes f_2$:

$$f_1 \rtimes f_2 = (f_1 \rtimes \mathrm{id}_{Y_2}) \circ (\mathrm{id}_{X_1} \ltimes f_2) \qquad \text{“}\mathit{first}\ f_2,\ \mathit{then}\ f_1\text{”}$$

Example: partiality

Then $f_1 \ltimes f_2 = f_1 \rtimes f_2$: every function is central.

$$\begin{cases} \mathcal{D}(f_1 \ltimes f_2) = \{(x_1, x_2) \mid x_1 \in \mathcal{D}(f_1) \,\wedge\, x_2 \in \mathcal{D}(f_2)\} \\ \text{and } \forall (x_1, x_2) \in \mathcal{D}(f_1 \ltimes f_2)\,, \\ \qquad\qquad (f_1 \ltimes f_2)(x_1, x_2) = (f(x_1), f_2(x_2)) \end{cases}$$

Sequential products, directly

Theorem. The left and right sequential products can be defined directly, in a mutually recursive way, by another "decorated" version of the product property:

$$
\begin{tikzcd}
X_1 \arrow[rr, "f_1"] \arrow[drr, phantom, "\geq"] & & Y_1 \\
X_1 \times X_2 \arrow[u, squiggly] \arrow[rr, "f_1 \ltimes f_2"] \arrow[dr, bend right, "f_1 \rtimes \mathrm{id}"'] & & Y_1 \times Y_2 \arrow[u, squiggly] \arrow[dd, squiggly] \\
& Y_1 \times X_2 \arrow[d, squiggly] \arrow[ur, phantom, "="] & \\
& X_2 \arrow[r, "f_2"] & Y_2
\end{tikzcd}
$$

and symmetrically...

Cartesian effect categories

Definition.

A cartesian effect category is

- an effect category $C \subseteq K$ with $\leq$
- with a binary product $\times$ on C
- and with sequential products $\ltimes$, $\rtimes$.

Theorem.

A cartesian effect category is a Freyd-category

Example: state

S: a fixed set of states (or stores).

$$S \xleftarrow{\ \sigma\ } S \times X \xrightarrow{\ \pi\ } X$$

Objects of C and K: sets
Morphism $f : X \to Y$ in K: function $[f] : S \times X \to S \times Y$
Pure morphism $v : X \rightsquigarrow Y$ in C: $[v] = \mathrm{id}_S \times v_0$

$$\begin{cases} f \approx f' & \Longleftrightarrow & \sigma \circ [f] = \sigma \circ [f'] \\ f \smile f' & \Longleftrightarrow & \pi \circ [f] = \pi \circ [f'] \\ f \leq f' & \Longleftrightarrow & \pi \circ [f] = \pi \circ [f'] \end{cases}$$

$$\forall x_1, x_2, s \ , \ [f_1 \ltimes f_2](s, x_1, x_2) = (s_2, y_1, y_2)$$

where $[f_1](s, x_1) = (s_1, y_1)$ and $[f_2](s_1, x_2) = (s_2, y_2)$.

Example: non-determinism

Cf. the monad of lists $\mathcal{L}(-)$.

Objects of C and K: sets
Morphism $f : X \to Y$ in K: function $[f] : X \to \mathcal{L}(Y)$
Pure morphism $v : X \rightsquigarrow Y$ in C: $[v]$ of length 1.

For all $f : X \to Y$ in K and $k \in \mathbb{N}$,
let $f^{\langle k \rangle} : X \to Y$ in K be the k-th "stutter":

$$[f^{\langle k \rangle}](x) = (y_1^k, \ldots, y_n^k) \text{ where } [f](x) = (y_1, \ldots, y_n)$$

$$\begin{cases} f \approx f' & \iff & \ell(f) = \ell(f') \\ f \smile f' & \iff & f = () \text{ or } f' = () \text{ or } \exists n, n' \in \mathbb{N},\ f^{\langle n \rangle} = f'^{\langle n' \rangle} \\ f \leq f' & \iff & \exists k \in \mathbb{N},\ f = f'^{\langle k \rangle} \end{cases}$$

$\forall x_1, x_2\ ,\ [f_1 \ltimes f_2](x_1, x_2) =$
$$(\langle y_{1,1}, y_{2,1} \rangle, \ldots, \langle y_{1,1}, y_{2,n_2} \rangle, \ldots, \langle y_{1,n_1}, y_{2,1} \rangle, \ldots, \langle y_{1,n_1}, y_{2,n_2} \rangle)$$
where $[f_1](x_1) = (y_{1,1}, \ldots, y_{1,n_1})$ and $[f_2](x_2) = (y_{2,1}, \ldots, y_{2,n_2})$

Conclusion

In this talk:

a new approach is provided for the major issue of dealing with multivariate operations when there are effects.

Future work:

more examples, look at the issue of combining effects.

Some references

- J.-G. Dumas, D. Duval, J.-C. Reynaud. Sequential products in effect categories. arXiv, 2007.
- J.-G. Dumas, D. Duval, J.-C. Reynaud. Cartesian effect categories are Freyd-categories. arXiv, 2009.
- Haskell. All About Monads. A comprehensive guide to the theory and practice of monadic programming in Haskell. `www.haskell.org/all_about_monads/`
- J. Hughes. Generalising monads to arrows. Science of Computer Programming 37 (1–3), p. 67–111, 2000.
- E. Moggi. Computational lambda-calculus and monads. Logic In Computer Science (LICS), IEEE Press, p. 14-23, 1989.
- J. Power, E. Robinson. Premonoidal Categories and Notions of Computation. Mathematical Structures in Computer Science 7 (5), p. 453–468, 1997.