BEHAVIORAL SEPARATION TYPES AT WORK

Luís Caires
(with João C. Seco, Filipe Militão, Luís Lourenço)
Universidade Nova de Lisboa
CITI@DI

IFIP meeting 2013 Lisboa

CENTER FOR INFORMATICS AND INFORMATION TECHNOLOGIES

BEHAVIORAL TYPES

- Several kinds of behavioral types
  - io-types
  - session types
  - usage types
  - contract types
  - ...
- Behavioral Separation Types: "***The Type Discipline of Behavioral Separation***" (Caires & Seco) **POPL 2013**

  Behavioral Separation: a general principle for disciplining interference in HO imperative concurrent programs

  Goal: "true" type safety for modern mainstream programming

PROGRAMMING LANGUAGE

e, f	$::=$	x	*(Variable)*
	$\mid$	$\lambda x.e$	*(Abstraction)*
	$\mid$	$e_1 e_2$	*(Application)*
	$\mid$	$\textsf{\textbf{let}}\ x = e_1\ \textsf{\textbf{in}}\ e_2$	*(Definition)*
	$\mid$	$\textsf{\textbf{var}}\ a\ \textsf{\textbf{in}}\ e$	*(Heap variable decl)*
	$\mid$	$a := v$	*(Assignment)*
	$\mid$	a	*(Dereference)*
	$\mid$	$[l_1 = e_1, \ldots]$	*(Tupling)*
	$\mid$	$e.l$	*(Selection)*
	$\mid$	$l(e)$	*(Variant)*
	$\mid$	$\textsf{\textbf{case}}\ e\ \textsf{\textbf{of}}\ l_i(x_i) \rightarrow e_i$	*(Conditional)*
	$\mid$	$\textsf{\textbf{rec}}(X)e$	*(Recursion)*
	$\mid$	X	*(Recursion variable)*
	$\mid$	$\textsf{\textbf{fork}}\ e$	*(New thread)*
	$\mid$	$\textsf{\textbf{wait}}\ e$	*(Wait)*
	$\mid$	$\textsf{\textbf{sync}}(a)e$	*(Synchronized block)*

KEY IDEAS

- traditionally, types seen as **state/structural properties**
- we move to types as **usage behaviors/protocols**
- Take stock on separation logics and behavioral types
- **focus**: separation of **usage protocols** for (stateful) values
- "structural" operators (basic usages) + "sequential" separation (traces) + "parallel" separation (aliasing / sharing)
  - "global" type assertions (talk about many values at once)
  - parallel and sequential frame principles (enable local reasoning both in the space and in the time dimensions)

BEHAVIORAL SEPARATION TYPES

$$
\begin{array}{rclrcl}
T, U & ::= & \mathtt{0} & \textit{(stop)} & \mid & T \mapsto V \quad \textit{(function)} \\
 & \mid & T\,\mathbf{;}\,U & \textit{(sequential)} & \mid & T \,|\, U \quad \textit{(parallel)} \\
 & \mid & T \,\&\, U & \textit{(intersection)} & \mid & l{:}T \quad \textit{(qualification)} \\
 & \mid & \bigoplus_{l \in I} l{:}T_l & \textit{(sum)} & \mid & !T \quad \textit{(shared)} \\
 & \mid & \circ T & \textit{(isolated)} & \mid & \tau(T) \quad \textit{(thread)} \\
 & \mid & \mathtt{rec}(X)T & \textit{(recursion)} & \mid & X \quad \textit{(recursion var)}
\end{array}
$$

KEY ALGEBRAIC STRUCTURE

- symmetric monoidal closed

$$(T, 0, (- \mid -), \mapsto)$$

- concurrent Kleene algebra

$$(T, (- \,\&\, -), (- \mid -), (- \,;\, -), 0)$$

- monoidal co-monads

$\circ(-)$ isolated

$!(-)$ shared

SEQUENTIAL AND PARALLEL TYPES

$U\,;(V\,;T) <:> (U\,;V)\,;T \quad U\,;0 <:> U \quad 0\,;U <:> U$

$U\,|(V\,|\,T) <:> (U\,|\,V)\,|\,T \quad U\,|\,V <:> V\,|\,U \quad U\,|\,0 <:> U$

$(A\,;C)\,|\,(B\,;D) <: (A\,|\,B)\;;(C\,|\,D)$

SHARED TYPE

$$!U <: U$$

$$!U <: !!U$$

$$0 <: !0$$

$$!U \mid !V <: !(U \mid V)$$

$$!U <: 0$$

$$!U <: !U \mid !U$$

ISOLATED TYPE

$$0 <: \circ 0$$

$$\circ U \mid \circ V <: \circ(U \mid V)$$

$$\circ U <: U$$

$$\circ U <: \circ\circ U$$

$$\circ U <: 0$$

$$!\circ U <: \circ ! U$$

$$(\circ U \mid V) ; T <: \circ U \mid (V ; T)$$

REMARKS

- **serialization** derivable from the exchange law

  $U \mid V <: V \,; U$

- isolation and the **postponing** law

  $(\circ U) \,; V <: (\circ U) \mid V \qquad\qquad (\circ U) \,; T <: T \,; (\circ U)$

- **pure** types (behave as "usual" types)

  let $T = !\circ U$. Then $T <:> T \mid T$ and $T <:> \circ T$ and $T <:> !T$

  include "usual" basic types, such as nat, bool, etc.

  (embedding into **pure** types $\eta : X \to !\circ X$)

BEHAVIORAL SEPARATION TYPES

$$
\begin{array}{lclrclr}
T, U & ::= & \mathtt{0} & (\textit{stop}) & \mid & T \mapsto V & (\textit{function}) \\
 & \mid & T \,\mathtt{;}\, U & (\textit{sequential}) & \mid & T \mid U & (\textit{parallel}) \\
 & \mid & T \,\&\, U & (\textit{intersection}) & \mid & l{:}T & (\textit{qualification}) \\
 & \mid & \bigoplus_{l \in I} l{:}T_l & (\textit{sum}) & \mid & !T & (\textit{shared}) \\
 & \mid & \circ T & (\textit{isolated}) & \mid & \tau(T) & (\textit{thread}) \\
 & \mid & \mathtt{rec}(X)T & (\textit{recursion}) & \mid & X & (\textit{recursion var})
\end{array}
$$