Nominal Types for a Nominal Calculus

Silvano Dal_Zilio
Laboratoire d'Informatique Fondamentale de Marseille (CNRS) & INRIA, projet MIMOSA

Types for Global Computing
ENS Paris - January 2003

A Brief History of Type Systems for π

- The basic type system for the polyadic π-calculus is Milner's sorting system.
  - Names are partitioned into a collection of sorts: S, T, …
  - a sorting function maps sorts onto sequences: $S \mapsto [S_1, \ldots, S_n]$ meaning that a channel of sort S can only carry tuples of n names with sorts $S_1, \ldots, S_n$
- Pierce and Sangiorgi have popularized a structural variant of sorts.
  - Channels have type: $Chan^{ann}\ [T_1, \ldots, T_n]$
  - Well-typed processes have a silent type: $E \vdash P : \diamond$
  - Variants are obtained by adding annotations: ann; recursive types: $\mu\mathcal{X}.T$; …

A Nominal Type System

- In our system, channels have type G, where G is a group name (following Ghelli terminology.)

- Fresh group names, like channel names, can be created dynamically and they have a dynamic scope.

$$(\nu G{:}T)P \quad \text{and} \quad (\nu x{:}G)P$$

A Nominal Type System

- A type, T, is a sequence of groups names (with possibly some annotations). We may also add a type for unfettered channels

$$T ::= [G_1, \ldots, G_n]\backslash ann \quad | \quad Un$$

- **Extension**: We may add polymorphism / subtyping (in the OO style) using "inheritance":

$$(\nu G \leq H)P$$

A Type System with Groups

- A group is an example of a pure name (in the sense of Needham) at the level of types. We can use them to represent:
  - kinds, such as Fun, Chan, … for function and continuation channels ;
  - tags: High, Low, … for control flow analysis ; ...
- Generation of fresh, unguessable group names models type generativity, a phenomenon observed e.g. with:
  - datatype in ML ;
  - runST in Haskell ;
  - letregion, in MLkit ; ...

Expressions and Processes

x, y, p, q	name: variable, channel
$P, Q, R ::=$	process
$\quad x(y_1, ..., y_n).P$	input
$\quad \overline{x}\langle y_1, \ldots, y_n \rangle$	output: asynchronous
$\quad (\nu G{:}T)P$	new-group: group restriction
$\quad (\nu x{:}G)P$	new-name: name restriction
$\quad P \mid Q$	composition
$\quad !P$	replication
$\quad \mathbf{0}$	inactivity
$E ::=$	type environment
$\quad \varnothing$	empty environment
$\quad E, x : G$	type of channel
$\quad E, G{:}T$	group definition

Structural Congruence: P ≡ Q

$$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$$

$$!P \equiv P \mid !P$$

$$\ldots$$

$$E_1 \equiv E_2 \Rightarrow (\nu E_1)P \equiv (\nu E_2)P$$

$$G \neq H \Rightarrow (\nu G{:}T)(\nu x{:}H)P \equiv (\nu x{:}H)(\nu G{:}T)P$$

$$G \notin fg(P) \Rightarrow (\nu G{:}T)(P \mid Q) \equiv P \mid (\nu G{:}T)Q$$

Dynamic Semantics

- Groups do not interfere with reductions and cannot be passed as values in a communication. Groups allow to express confinement properties (non-interference) of channels in the untyped calculus.

Property:

- If $P \to Q$ then $erase(P) \to erase(Q)$.
- If $erase(P) \to R$, then there is a typed process Q such that $P \to Q$ and $R \equiv erase(Q)$.

Static Semantics: a Type and Effect System

- In this talk, a type takes the form

  $[G_1, \ldots, G_n] \backslash \{H_1, \ldots, H_k\}$

- A channel in group G may only exchange n-uples of names in the groups $G_1, \ldots, G_n$. The set of groups $H_1, \ldots, H_k$ is called the hidden effect of the channel.

- The main judgment is:

  $E \vdash P : \{G_1, \ldots, G_n\}$

  meaning that P uses names according to their types and that all external/public reads and writes are on channels in groups $G_1, \ldots, G_n$.

(Proc Un Input)
$$\frac{E \vdash x : Un = Un \quad E, y_1{:}Un, \ldots, y_n{:}Un \vdash P : \mathbf{G}}{E \vdash x(y_1, \ldots, y_n).P : \{Un\} \cup \mathbf{G}}$$

(Proc Un Output)
$$\frac{E \vdash x : Un = Un \quad E \vdash y_1 : Un \quad \cdots \quad E \vdash y_n : Un}{E \vdash \overline{x}\langle y_1, \ldots, y_n\rangle : \{Un\}}$$

$$\frac{E \vdash P : \mathbf{G} \quad E \vdash Q : \mathbf{H}}{E \vdash P \mid Q : \mathbf{G} \cup \mathbf{H}} \qquad \frac{E \vdash P : \mathbf{H}}{E \vdash !P : \mathbf{H}} \qquad \frac{E \vdash \diamond}{E \vdash \mathbf{0} : \varnothing}$$

$$\frac{E, G{:}T \vdash P : \mathbf{H} \quad G \neq Un}{E \vdash (\nu G{:}T)P : \mathbf{H} - \{G\}} \qquad \frac{E, x{:}T \vdash P : \mathbf{H}}{E \vdash (\nu x{:}T)P : \mathbf{H}}$$

$$\frac{E \vdash P : \mathbf{G} \quad \mathbf{G} \subseteq \mathbf{H} \subseteq dom(E)}{E \vdash P : \mathbf{H}}$$

(Proc Un Input)
$$\frac{E \vdash x : Un = Un \quad E, y_1{:}Un, \ldots, y_n{:}Un \vdash P : \mathbf{G}}{E \vdash x(y_1, \ldots, y_n).P : \{Un\} \cup \mathbf{G}}$$

(Proc Un Output)
$$\frac{E \vdash x : Un = Un \quad E \vdash y_1 : Un \quad \cdots \quad E \vdash y_n : Un}{E \vdash \overline{x}\langle y_1, \ldots, y_n\rangle : \{Un\}}$$

$$\frac{E \vdash P : \mathbf{G} \quad E \vdash Q : \mathbf{H}}{E \vdash P \mid Q : \mathbf{G} \cup \mathbf{H}} \qquad \frac{E \vdash P : \mathbf{H}}{E \vdash !P : \mathbf{H}} \qquad \frac{E \vdash \diamond}{E \vdash \mathbf{0} : \varnothing}$$

$$\frac{E, G{:}T \vdash P : \mathbf{H} \quad G \neq Un}{E \vdash (\nu G{:}T)P : \mathbf{H} - \{G\}} \qquad \frac{E, x{:}T \vdash P : \mathbf{H}}{E \vdash (\nu x{:}T)P : \mathbf{H}}$$

$$\frac{E \vdash P : \mathbf{G} \quad \mathbf{G} \subseteq \mathbf{H} \subseteq dom(E)}{E \vdash P : \mathbf{H}}$$

(Proc Un Input)
$$\frac{E \vdash x : Un = Un \quad E, y_1{:}Un, \ldots, y_n{:}Un \vdash P : \mathbf{G}}{E \vdash x(y_1, \ldots, y_n).P : \{Un\} \cup \mathbf{G}}$$

(Proc Un Output)
$$\frac{E \vdash x : Un = Un \quad E \vdash y_1 : Un \quad \cdots \quad E \vdash y_n : Un}{E \vdash \overline{x}\langle y_1, \ldots, y_n\rangle : \{Un\}}$$

$$\frac{E \vdash P : \mathbf{G} \quad E \vdash Q : \mathbf{H}}{E \vdash P \mid Q : \mathbf{G} \cup \mathbf{H}} \qquad \frac{E \vdash P : \mathbf{H}}{E \vdash !P : \mathbf{H}} \qquad \frac{E \vdash \diamond}{E \vdash \mathbf{0} : \varnothing}$$

$$\frac{E, G{:}T \vdash P : \mathbf{H} \quad G \neq Un}{E \vdash (\nu G{:}T)P : \mathbf{H} - \{G\}} \qquad \frac{E, x{:}T \vdash P : \mathbf{H}}{E \vdash (\nu x{:}T)P : \mathbf{H}}$$

$$\frac{E \vdash P : \mathbf{G} \quad \mathbf{G} \subseteq \mathbf{H} \subseteq dom(E)}{E \vdash P : \mathbf{H}}$$

(Proc Un Input)
$$\frac{E \vdash x : Un = Un \quad E, y_1{:}Un, \ldots, y_n{:}Un \vdash P : \mathbf{G}}{E \vdash x(y_1, \ldots, y_n).P : \{Un\} \cup \mathbf{G}}$$

(Proc Un Output)
$$\frac{E \vdash x : Un = Un \quad E \vdash y_1 : Un \quad \cdots \quad E \vdash y_n : Un}{E \vdash \overline{x}\langle y_1, \ldots, y_n\rangle : \{Un\}}$$

$$\frac{E \vdash P : \mathbf{G} \quad E \vdash Q : \mathbf{H}}{E \vdash P \mid Q : \mathbf{G} \cup \mathbf{H}} \qquad \frac{E \vdash P : \mathbf{H}}{E \vdash !P : \mathbf{H}} \qquad \frac{E \vdash \diamond}{E \vdash \mathbf{0} : \varnothing}$$

$$\frac{E, G{:}T \vdash P : \mathbf{H} \quad G \neq Un}{E \vdash (\nu G{:}T)P : \mathbf{H} - \{G\}} \qquad \frac{E, x{:}T \vdash P : \mathbf{H}}{E \vdash (\nu x{:}T)P : \mathbf{H}}$$

$$\frac{E \vdash P : \mathbf{G} \quad \mathbf{G} \subseteq \mathbf{H} \subseteq dom(E)}{E \vdash P : \mathbf{H}}$$

(Proc Un Input)
$$\frac{E \vdash x : Un = Un \quad E, y_1{:}Un, \ldots, y_n{:}Un \vdash P : \mathbf{G}}{E \vdash x(y_1, \ldots, y_n).P : \{Un\} \cup \mathbf{G}}$$

(Proc Un Output)
$$\frac{E \vdash x : Un = Un \quad E \vdash y_1 : Un \quad \cdots \quad E \vdash y_n : Un}{E \vdash \overline{x}\langle y_1, \ldots, y_n\rangle : \{Un\}}$$

$$\frac{E \vdash P : \mathbf{G} \quad E \vdash Q : \mathbf{H}}{E \vdash P \mid Q : \mathbf{G} \cup \mathbf{H}} \qquad \frac{E \vdash P : \mathbf{H}}{E \vdash !P : \mathbf{H}} \qquad \frac{E \vdash \diamond}{E \vdash \mathbf{0} : \varnothing}$$

$$\frac{E, G{:}T \vdash P : \mathbf{H} \quad G \neq Un}{E \vdash (\nu G{:}T)P : \mathbf{H} - \{G\}} \qquad \frac{E, x{:}T \vdash P : \mathbf{H}}{E \vdash (\nu x{:}T)P : \mathbf{H}}$$

$$\frac{E \vdash P : \mathbf{G} \quad \mathbf{G} \subseteq \mathbf{H} \subseteq dom(E)}{E \vdash P : \mathbf{H}}$$

(Proc Un Input)
$$\frac{E \vdash x : Un = Un \quad E, y_1{:}Un, \ldots, y_n{:}Un \vdash P : \mathbf{G}}{E \vdash x(y_1, \ldots, y_n).P : \{Un\} \cup \mathbf{G}}$$

(Proc Un Output)
$$\frac{E \vdash x : Un = Un \quad E \vdash y_1 : Un \quad \cdots \quad E \vdash y_n : Un}{E \vdash \overline{x}\langle y_1, \ldots, y_n\rangle : \{Un\}}$$

$$\frac{E \vdash P : \mathbf{G} \quad E \vdash Q : \mathbf{H}}{E \vdash P \mid Q : \mathbf{G} \cup \mathbf{H}} \qquad \frac{E \vdash P : \mathbf{H}}{E \vdash !P : \mathbf{H}} \qquad \frac{E \vdash \diamond}{E \vdash \mathbf{0} : \varnothing}$$

$$\frac{E, G{:}T \vdash P : \mathbf{H} \quad G \neq Un}{E \vdash (\nu G{:}T)P : \mathbf{H} - \{G\}} \qquad \frac{E, x{:}T \vdash P : \mathbf{H}}{E \vdash (\nu x{:}T)P : \mathbf{H}}$$

$$\frac{E \vdash P : \mathbf{G} \quad \mathbf{G} \subseteq \mathbf{H} \subseteq dom(E)}{E \vdash P : \mathbf{H}}$$

Good Environments and Processes

$E \vdash \diamond$	good environment
$E \vdash x : G = T$	x in group G has type T
$E \vdash P : \mathbf{H}$	good process P with effect $\mathbf{H}$

$$\frac{}{Un{:}Un \vdash \diamond} \qquad \frac{E \vdash \diamond \quad G \in dom(E) \quad x \notin dom(E)}{E, x{:}G \vdash \diamond}$$

$$\frac{E \vdash \diamond \quad \{G_1, \ldots, G_n\} \cup \mathbf{H} \subseteq dom(E) \cup \{G\} \quad G \notin dom(E)}{E, G{:}[G_1, \ldots, G_n] \backslash \mathbf{H} \vdash \diamond}$$

$$\frac{E \vdash \diamond \quad E(x) = G \quad E(G) = T}{E \vdash x : G = T}$$

(Proc Input)

$$\frac{E \vdash x : G = [G_1, \ldots, G_n]\backslash\mathbf{H} \qquad E, y_1{:}G_1, \ldots, y_n{:}G_n \vdash P : \mathbf{G}}{E \vdash x(y_1, \ldots, y_n).P : \{G\} \cup (\mathbf{G} - \mathbf{H})}$$

(Proc Output)

$$\frac{E \vdash x : G = [G_1, \ldots, G_n]\backslash\mathbf{H} \qquad E \vdash y_1 : G_1 \qquad \cdots \qquad E \vdash y_n : G_n}{E \vdash \overline{x}\langle y_1, \ldots, y_n\rangle : \{G\} \cup \mathbf{H}}$$

$$E \vdash x(y_1, \ldots, y_n).P \mid \overline{x}\langle y_1, \ldots, y_n\rangle : \{G\} \cup \mathbf{G}$$

(Proc Un Input)

$$\frac{E \vdash x : Un = Un \quad E, y_1{:}Un, \ldots, y_n{:}Un \vdash P : \mathbf{G}}{E \vdash x(y_1, \ldots, y_n).P : \{Un\} \cup \mathbf{G}}$$

(Proc Un Output)

$$\frac{E \vdash x : Un = Un \quad E \vdash y_1 : Un \quad \cdots \quad E \vdash y_n : Un}{E \vdash \overline{x}\langle y_1, \ldots, y_n\rangle : \{Un\}}$$

$$\frac{E \vdash P : \mathbf{G} \quad E \vdash Q : \mathbf{H}}{E \vdash P \mid Q : \mathbf{G} \cup \mathbf{H}} \qquad \frac{E \vdash P : \mathbf{H}}{E \vdash !P : \mathbf{H}} \qquad \frac{E \vdash \diamond}{E \vdash \mathbf{0} : \varnothing}$$

$$\frac{E, G{:}T \vdash P : \mathbf{H} \quad G \neq Un}{E \vdash (\nu G{:}T)P : \mathbf{H} - \{G\}} \qquad \frac{E, x{:}T \vdash P : \mathbf{H}}{E \vdash (\nu x{:}T)P : \mathbf{H}}$$

$$\frac{E \vdash P : \mathbf{G} \quad \mathbf{G} \subseteq \mathbf{H} \subseteq dom(E)}{E \vdash P : \mathbf{H}}$$

Results

Theorem (Subject Reduction) If $E \vdash P : \mathbf{H}$ and $P \rightarrow Q$ then $E \vdash Q : \mathbf{H}$.

Theorem (Sort Soundness) If $E \vdash P$ in the simple structural sort system, then there is a typed process Q such that $E \vdash Q : \mathbf{H}$ and $erase(Q) = P$.

Theorem (Effect Soundness) If $E \vdash P : \mathbf{H}$ and $P \downarrow x$ or $P \downarrow \overline{x}$ then there is a type T such that $E \vdash x : G = T$ and $G \in \mathbf{H}$.

A New Garbage Collection Law

Let $\approx$ denotes barbed congruence in the untyped π-calculus. Define $P \approx Q$ if $erase(P) \approx erase(Q)$.

Theorem (Confinement) Assume $E, G{:}T, E' \vdash P : \mathbf{H}_1$ and $E, G{:}T, E' \vdash R : \{\mathbf{H}_2\}$ where $\mathbf{H}_1 \cap \mathbf{H}_2 = \varnothing$ then:

$$(\nu G{:}T, E')(P \mid R) \approx (\nu G{:}T, E')P$$

Proof We use a (typed) translation to an untyped π-calculus equipped with a type system for *untyped/untrusted* channels, as in Abadi and Gordon's spi-calculus.

Garbage Collection of Inactive Processes

- We can use our theorem to prove a series of “garbage collection laws” for the untyped calculus.

- Since:

$$G{:}[\,]\backslash\emptyset\ ,\ x{:}G \vdash x().R : \{G\}$$

$$G{:}[\,]\backslash\emptyset\ ,\ x{:}G \vdash 0 : \emptyset$$

we have that :

$$(\nu x)\ (0 \mid x().R) \approx 0$$

Garbage Collection of Cycles and Threads

Let $E = G_1:[G_2], G_2:[], x : G_1, y : G_2$

$$(\nu x,y)(x\langle y\rangle \mid y().x(z).R) \approx 0$$

Assume $E' = G:[], x:G$ and $E, E' \vdash P : \mathbf{H}$ with $G \notin \mathbf{H}$:

$$(\nu x)(P \mid !x().(x\langle\ \rangle \mid R) \mid x\langle\ \rangle) \approx (\nu x)P$$

Conclusion

- The notion of group has many interesting applications. For instance, the type π-calculus presented here has been used to reason about automatic memory management.
  - **Region Analysis and a π-calculus with Groups.** Dal-Zilio, Gordon, MFCS 2000, LNCS vol. 1893.
- A π-calculus with groups, but without effects, has been used to reason about secrecy. We capture more examples.
  - **Secrecy and Group Creation.** Cardelli, Ghelli, Gordon, CONCUR 2000, LNCS vol. 1877.

Conclusion

- Groups can be easily integrated to other nominal calculi. In fact, group-based type systems have been first developed for the ambient calculus to control the mobility of ambients allowing static checking of access rights.
  - **Ambient Groups and Mobility Types.** Cardelli, Ghelli, Gordon, IFIP TCS2000, LNCS vol. 1872.