Type Systems for the Termination of Mobile Processes

Romain Demangeon

ENS Lyon

(Joint work with D.Hirschkoff, ENS Lyon and D.Sangiorgi, University of Bologna)

An overview of recent results

Starting point

- *Ensuring termination by typability* [06] (Deng, Sangiorgi)

Our countributions

- *On the complexity of termination inference for processes* [TGC'07] (Demangeon, Hirschkoff, Kobayashi, Sangiorgi).
- *Static and dynamic typing for the termination of mobile processes* [TCS'08] (Demangeon, Hirschkoff, Sangiorgi).
- *Termination in higher-order concurrent calculi* [FSEN'09] (Demangeon, Hirschkoff, Sangiorgi).

Termination for mobile processes

Termination

- Existence of multiple techniques for sequential programs (rewriting theory, proof theory).
- Termination of mobile systems is challenging. Structures evolve at run-time.

The Formalism

π-calculus: studying termination from different points of view within a unique framework:

- First-order: termination and name-passing.
- Higher-order: termination and process-passing.

Syntax and semantics

- Names (channels): $a, b, c, v, x, \dots$
- Syntax: $P ::= \mathbf{0} \mid \overline{a}\langle v\rangle.P \mid a(x).P \mid !a(x).P \mid (P \mid P)$
- Reductions:

$$\frac{}{a(x).P \mid \overline{a}\langle v\rangle.Q \rightarrow P[v/x] \mid Q}$$

$$\frac{}{!a(x).P \mid \overline{a}\langle v\rangle.Q \rightarrow !a(x).P \mid P[v/x] \mid Q} \qquad \frac{P \rightarrow P'}{(P \mid Q) \rightarrow (P' \mid Q)}$$

Syntax and semantics

- Names (channels): $a, b, c, v, x, \dots$
- Syntax: $P ::= \mathbf{0} \mid \overline{a}\langle v\rangle.P \mid a(x).P \mid !a(x).P \mid (P \mid P)$
- Reductions:

$$\frac{}{a(x).P \mid \overline{a}\langle v\rangle.Q \to P[v/x] \mid Q}$$

$$\frac{}{!a(x).P \mid \overline{a}\langle v\rangle.Q \to !a(x).P \mid P[v/x] \mid Q} \qquad \frac{P \to P'}{(P \mid Q) \to (P' \mid Q)}$$

Replication: persistence, source of divergence

$$P_1 = \overline{a}\langle v\rangle \mid !a(x).\overline{a}\langle x\rangle \qquad P_2 = \overline{a}\langle v\rangle \mid !a(x).\overline{b}\langle x\rangle \mid !b(y).\overline{a}\langle y\rangle$$

$$P_3 = \overline{c}\langle a\rangle \mid \overline{a}\langle b\rangle \mid c(y).!y(x).\overline{a}\langle x\rangle$$

Syntax and semantics

- Names (channels): $a, b, c, v, x, \ldots$
- Syntax: $P ::= \mathbf{0} \mid \overline{a}\langle v\rangle.P \mid a(x).P \mid !a(x).P \mid (P \mid P)$
- Reductions:

$$\frac{}{a(x).P \mid \overline{a}\langle v\rangle.Q \to P[v/x] \mid Q}$$

$$\frac{}{!a(x).P \mid \overline{a}\langle v\rangle.Q \to !a(x).P \mid P[v/x] \mid Q} \qquad \frac{P \to P'}{(P \mid Q) \to (P' \mid Q)}$$

Replication: persistence, source of divergence

$$P_1 = \overline{a}\langle v\rangle \mid !a(x).\overline{a}\langle x\rangle \qquad P_2 = \overline{a}\langle v\rangle \mid !a(x).\overline{b}\langle x\rangle \mid !b(y).\overline{a}\langle y\rangle$$

$$P_3 = \overline{c}\langle a\rangle \mid \overline{a}\langle b\rangle \mid c(y).!y(x).\overline{a}\langle x\rangle$$

S1: the original type system

Principles

- Assigning *levels* (integers) to channels.
- Assigning levels to processes: maximum level of a name used in output (not appearing under a replication).

$$Q_1 = !a(x).(\overline{b}\langle x\rangle \mid \overline{b}\langle x\rangle \mid \overline{c}\langle x\rangle) \mid !b(y).\overline{c}\langle y\rangle \mid \overline{a}\langle v\rangle \mid \overline{c}\langle v\rangle$$

- Typing $!a^n(\widetilde{x}).P$ with $P : m$ when $n > m$.
- Firing a replicated input: trading an output for several smaller outputs.

S1: the original type system

Principles

- Assigning *levels* (integers) to channels.
- Assigning levels to processes: maximum level of a name used in output (not appearing under a replication).

$$Q_1 = !a^3(x).(\overline{b}^2\langle x\rangle \mid \overline{b}^2\langle x\rangle \mid \overline{c}^1\langle x\rangle) \mid !b^2(y).\overline{c}^1\langle y\rangle \mid \overline{a}^3\langle v\rangle \mid \overline{c}^1\langle v\rangle$$

- Typing $!a^n(\widetilde{x}).P$ with $P : m$ when $n > m$.
- Firing a replicated input: trading an output for several smaller outputs.

The System S1

Typing rule for replicated input

$$\frac{\Gamma, \widetilde{x} : \widetilde{T} \vdash P : m \qquad m < n}{\Gamma \vdash !a^n(\widetilde{x}).P : 0} \quad \text{(S1)}$$

Giving level 0 to replicated process

- Guarded by a replication.
- $!a(x).(\overline{b} \mid !x(y).P)$: outputs in P do not count for the replication on a.

Soundness

- If P is S1-typable, P terminates
- *(Proof)* the multiset of the levels of every outputs (not under a replication) is decreasing for the multiset ordering.

- $Q_1 = !a^3(x).(\overline{b}^2\langle x\rangle \mid \overline{b}^2\langle x\rangle \mid \overline{c}^1\langle x\rangle) \mid !b^2(y).\overline{c}^1\langle y\rangle \mid \overline{a}^3\langle v\rangle \mid \overline{c}^1\langle v\rangle$
  $[1,0,1] \rightarrow [0,2,2] \rightarrow [0,1,3] \rightarrow [0,0,4]$.
- $P_1 = \overline{a}^n\langle v\rangle \mid !a^n(x).\overline{a}^n\langle x\rangle$ not S1-typable as it forces $n > n$.
- Similarly $P_2 = \overline{a}^n\langle v\rangle \mid !a^n(x).\overline{b}^m\langle x\rangle \mid !b^m(y).\overline{a}^n\langle y\rangle$ gives $n > m > n$.

Soundness

- If P is S1-typable, P terminates
- *(Proof)* the multiset of the levels of every outputs (not under a replication) is decreasing for the multiset ordering.

- $Q_1 =$
  $!a^3(x).(\overline{b}^2\langle x\rangle \mid \overline{b}^2\langle x\rangle \mid \overline{c}^1\langle x\rangle) \mid !b^2(y).\overline{c}^1\langle y\rangle \mid \overline{a}^3\langle v\rangle \mid \overline{c}^1\langle v\rangle$
  $[1, 0, 1] \rightarrow [0, 2, 2] \rightarrow [0, 1, 3] \rightarrow [0, 0, 4]$.
- $P_1 = \overline{a}^n\langle v\rangle \mid !a^n(x).\overline{a}^n\langle x\rangle$ not S1-typable as it forces $n > n$.
- Similarly $P_2 = \overline{a}^n\langle v\rangle \mid !a^n(x).\overline{b}^m\langle x\rangle \mid !b^m(y).\overline{a}^n\langle y\rangle$ gives $n > m > n$.
- In $P_3 = \overline{c}^k\langle a\rangle \mid \overline{a}^n\langle b\rangle \mid c^k(y).!y^m(x).\overline{a}^n\langle x\rangle$, the type of c forces $n = m$.

Soundness

- If P is S1-typable, P terminates
- *(Proof)* the multiset of the levels of every outputs (not under a replication) is decreasing for the multiset ordering.

- $Q_1 =$
  $!a^3(x).(\overline{b}^2\langle x\rangle \mid \overline{b}^2\langle x\rangle \mid \overline{c}^1\langle x\rangle) \mid !b^2(y).\overline{c}^1\langle y\rangle \mid \overline{a}^3\langle v\rangle \mid \overline{c}^1\langle v\rangle$
  $[1,0,1] \rightarrow [0,2,2] \rightarrow [0,1,3] \rightarrow [0,0,4]$.
- $P_1 = \overline{a}^n\langle v\rangle \mid !a^n(x).\overline{a}^n\langle x\rangle$ not S1-typable as it forces $n > n$.
- Similarly $P_2 = \overline{a}^n\langle v\rangle \mid !a^n(x).\overline{b}^m\langle x\rangle \mid !b^m(y).\overline{a}^n\langle y\rangle$ gives $n > m > n$.
- In $P_3 = \overline{c}^k\langle a\rangle \mid \overline{a}^n\langle b\rangle \mid c^k(y).!y^m(x).\overline{a}^n\langle x\rangle$, the type of c forces $n = m$.

Type inference

- Inference for S1 is polynomial.
- Graph where nodes are names and edges are constraints $\Rightarrow$ topological sort gives a solution if there is one.

Length of reductions

- P typable process of size N : less than N^{N+1} reductions from P.
- $S_n = \overline{a_1} \mid !a_1.(\overline{a_2} \mid \overline{a_2}) \mid !a_2.(\overline{a_3} \mid \overline{a_3} \mid \overline{a_3}) \mid \ldots$.
  - Size of S_n: $\Theta(n^2)$
  - S_n can perform $\Theta(!n)$ reductions.

Type inference

- Inference for S1 is polynomial.
- Graph where nodes are names and edges are constraints $\Rightarrow$ topological sort gives a solution if there is one.

Length of reductions

- P typable process of size N : less than N^{N+1} reductions from P.
- $S_n = \overline{a_1} \mid !a_1.(\overline{a_2} \mid \overline{a_2}) \mid !a_2.(\overline{a_3} \mid \overline{a_3} \mid \overline{a_3}) \mid \ldots.$
  - Size of S_n: $\Theta(n^2)$
  - S_n can perform $\Theta(!n)$ reductions.

S2: a more expressive system

Principles

- No recursion is allowed in S1. $!a(x).P$ typable $\Rightarrow \overline{a} \notin P$.
- Recursions:
  - sometimes desirable (modelling recursive functions)
  - sometimes innocuous like in $!a(x).b(y).\overline{a}\langle x\rangle$.
- S2: more expressive system.
  - Input sequences are taken as a whole. $!a_1(\widetilde{x_1}).\ldots.a_k(\widetilde{x_k}).P$
  - Comparing multisets of levels

$Q_2 = !a(x)^n.b(y)^m.\overline{a}^n\langle x\rangle \mid !a^n(z).\overline{b}^m\langle z\rangle \mid \overline{a}^n\langle v_1\rangle \mid \overline{b}^m\langle v_2\rangle$

- Ruled out by S1: $n > n$.
- Can be type-checked in S2 with $n = 2$ and $m = 1$. The first replication is typed as $\{1, 2\} >_{mul} \{2\}$.

S2: a more expressive system

Principles

- No recursion is allowed in S1. $!a(x).P$ typable $\Rightarrow \overline{a} \notin P$.
- Recursions:
  - sometimes desirable (modelling recursive functions)
  - sometimes innocuous like in $!a(x).b(y).\overline{a}\langle x\rangle$.
- S2: more expressive system.
  - Input sequences are taken as a whole. $!a_1(\widetilde{x_1}).\ldots.a_k(\widetilde{x_k}).P$
  - Comparing multisets of levels

$$Q_2 = !a(x)^n.b(y)^m.\overline{a}^n\langle x\rangle \mid !a^n(z).\overline{b}^m\langle z\rangle \mid \overline{a}^n\langle v_1\rangle \mid \overline{b}^m\langle v_2\rangle$$

- Ruled out by S1: $n > n$.
- Can be type-checked in S2 with $n = 2$ and $m = 1$. The first replication is typed as $\{1,2\} >_{mul} \{2\}$.

Typing rule for replicated inputs

$$\frac{\Gamma, \widetilde{x_1} : \widetilde{T_1}, \ldots, \widetilde{x_k} : \widetilde{T_k} \vdash P : M \qquad \{n_1, \ldots, n_k\} >_{mul} M}{\Gamma \vdash !a_1^{n_1}(\widetilde{x_1}). \ldots .a_k^{n_k}(\widetilde{x_k}).P : \emptyset} \quad \text{(S2)}$$

Complexity

- S1-typable $\Rightarrow$ S2-typable.
- Inference for S2 is NP-complete.
- Polynomial if we use algebrical operations on levels in the type system.

S3: A system using partial orders

Principles

- A system allowing replications with no decreasing in weight.
- A well-founded partial order between names of the same level ensures termination.

Typing complex structures

$!p(a,b).a(r).(\overline{b}\langle r\rangle \mid \overline{p}\langle a,b\rangle)$ (a and b play the same role)

- Models the behavior of a list structure.
- Typed with a low level for p, a higher level for a, b and $a \succ b$.
- Type of p: first argument dominates the second one.
- S4: extension of S3 in which we can type tree structures $!p(a,b,c).a(r).(\overline{b}\langle r\rangle \mid \overline{c}\langle r\rangle \mid \overline{p}\langle a,b,c\rangle)$. $\Rightarrow$ the weight increases.

HOPi as a framework for higher order mobility

HOPi

- Messages exchanged are processes.
- $a(X).P_1 \mid \overline{a}\langle Q\rangle.P_2 \rightarrow P_1[Q/X] \mid P_2$.
- Replication is no longer present ...

Another form of recursion

- ... however, divergences arise.
- $Q_0 = a(X).(\overline{a}\langle X\rangle \mid X)$ and $P_0 = Q_0 \mid \overline{a}\langle Q_0\rangle$. ($P_0$ diverges.)

A weight-based type system for higher order π

Typing rule for output

$$\frac{\Gamma \vdash P : k \qquad \Gamma \vdash Q : m \qquad k < n}{\Gamma \vdash \overline{a}^n\langle P\rangle.Q : \max(m, n)} \quad \text{(S4)}$$

Principles

- Forbidding self-emissions: for $\overline{a}\langle P\rangle$ to be typable, $\overline{a} \notin P$.
- Use of levels: for $\overline{a}^n\langle P\rangle$ with $P : m$, we must have $n > m$.
- *Soundness*: typable processes terminate.

Examples

Ruling out P_0

$Q_0 = a(X).(\overline{a}\langle X\rangle \mid X)$ and $P_0 = Q_0 \mid \overline{a}\langle Q_0\rangle$

- P_0 contains $\overline{a}^n\langle Q_0\rangle$ and Q_0 contains an output on a^n.
- To type P_0, we must have $n > n$.

A typable process

$Q_3 = a(X).(X \mid X) \mid \overline{a}\langle \overline{b}\langle \mathbf{0}\rangle\rangle \mid b(Y).Y$

Examples

Ruling out P_0

$Q_0 = a(X).(\overline{a}\langle X\rangle \mid X)$ and $P_0 = Q_0 \mid \overline{a}\langle Q_0\rangle$

- P_0 contains $\overline{a}^n\langle Q_0\rangle$ and Q_0 contains an output on a^n.
- To type P_0, we must have $n > n$.

A typable process

$Q_3 = a^2(X).(X \mid X) \mid \overline{a}^2\langle \overline{b}^1\langle \mathbf{0}\rangle\rangle \mid b^1(Y).Y$
$[1,0] \rightarrow [0,2] \rightarrow [0,1]$

Relation with S1

- S4 is to HOPi what S1 is to π.
- Polynomial inference: like in S1.
- Standard encoding of HOPi in π [SangiorgiWalker01]: nearly every typable HOPi process is encoded into a S1-typable process.

Further extensions

Exploring termination in richer higher-order calculi:

- A calculus where messages are functions of any order.
- A calculus where localized computation and passivation are allowed.

Conclusion

Future Works

- Develop extensions of the system for higher-order message passing.
- Analyse other approaches to termination (logical relations).
- Explore the use of polymorphism.
- Study termination in calculi composed of a functional part (termination proved with logical relations) and an imperative part (termination proved with a multiset decreasing). (π with 'functional names', λ+ref (Boudol [Concur07]), λ+objects).

Annex 1

Reduction from 3SAT

- Initial problem: clauses $(C_j)_j$ of 3 litterals l_j^1, l_j^2, l_j^3, propositional variables $V = v_1, \ldots v_p$.
- Solution: mapping $val : V \rightarrow \{\mathbf{True}, \mathbf{False}\}$ s.t. for every clause C_i at least one element of $\{val(l_j^1), val(l_j^2), val(l_j^3)\}$ is **True**.
- Creation fo a name τ. Having a level $\geq niv(\tau) =$ "Is valued to **True**".
- Creation of two names x_i, x_i' for each prop.var. v_i and adding the module $!x_i.x_i'.\overline{\tau} \mid !\tau.\tau.(\overline{x_i} \mid \overline{x_i'})$.
- For every clause $C_j = l_j^1, l_j^2, l_j^3$, if $l_j^i = v_k$ we use x_k if $l_j^i = \neg v_k$ we use x_k'. We add the module $!\tau.\tau.(\overline{l_j^1} \mid \overline{l_j^2} \mid \overline{l_j^3}) \mid !l_j^1.l_j^2.l_j^3.\overline{\tau}$.
- Equivalence between "the initial problem has a solution" and "the parallel composition of the module is typable".

Annex 2

Code of the *symbol table*

$!p(a, b, s, e).a(v, r)$ **if** $s = v$ **then**
    $\overline{r}\langle e\rangle \mid \overline{p}\langle a, b, s, e\rangle$
else
    if $b \neq nil$ **then**
        $\overline{b}\langle v, r\rangle \mid \overline{p}\langle a, b, s, e\rangle$
    else
        $(\nu c, f)(\overline{r}\langle f\rangle \mid \overline{p}\langle a, c, s, e\rangle \mid \overline{p}\langle c, nil, v, f\rangle)$

Initialization

$\overline{p}\langle root, nil, dummy1, dummy2\rangle$

Requests

$\overline{root}\langle$"Existence"$, r_1\rangle \mid \overline{root}\langle$"Essence"$, r_2\rangle \mid \overline{root}\langle$"Existence"$, r_2\rangle$