Termination of mobile processes

R. Demangeon[1]

(under the direction of D. Hirschkoff[1] & D. Sangiorgi[2])

TIC 2010 - Bangalore

[1]ENS Lyon

[2]Universita di Bologna

Objectives

Termination

- A system $(S, \rightarrow)$ terminates if every *reduction starting from S is finite*.
- Undecidable property for interesting languages.
- Key property for proving soundness, lock-freedom, complexity bounds.
- Well-studied in rewriting theory and proof theory.

Objectives

Termination

- A system $(S, \to)$ terminates if every *reduction starting from S is finite*.
- Undecidable property for interesting languages.
- Key property for proving soundness, lock-freedom, complexity bounds.
- Well-studied in rewriting theory and proof theory.

Concurrency

- Termination of concurrent system less studied.
- Challenging: systems with dynamically evolving topology.
- Framework of π-calculi.

π as a framework

π-calculus

- Names/Channels a, b, x.
- Inactive process $\mathbf{0}$.

π as a framework

π-calculus

- Names/Channels a, b, x.
- Inactive process $\mathbf{0}$.
- Input $a(x).P$.
- Output $\overline{a}\langle v\rangle.P$.
- Parallel composition $P_1 \mid P_2$ (associativity / commutativity).

π as a framework

π-calculus

- Names/Channels a, b, x.
- Inactive process $\mathbf{0}$.
- Input $a(x).P$.
- Output $\overline{a}\langle v\rangle.P$.
- Parallel composition $P_1 \mid P_2$ (associativity / commutativity).
- Communication $a(x).P \mid \overline{a}\langle v\rangle.Q \mid R \rightarrow P\{v/x\} \mid Q \mid R$.

π as a framework

π-calculus

- Names/Channels a, b, x.
- Inactive process $\mathbf{0}$.
- Input $a(x).P$.
- Output $\overline{a}\langle v\rangle.P$.
- Parallel composition $P_1 \mid P_2$ (associativity / commutativity).
- Communication $a(x).P \mid \overline{a}\langle v\rangle.Q \mid R \rightarrow P\{v/x\} \mid Q \mid R$.

Replication

- Terminating system: if $P \rightarrow P'$, $\texttt{prefix}(P') < \texttt{prefix}(P)$.

π as a framework

π-calculus

- Names/Channels a, b, x.
- Inactive process $\mathbf{0}$.
- Input $a(x).P$.
- Output $\overline{a}\langle v \rangle.P$.
- Parallel composition $P_1 \mid P_2$ (associativity / commutativity).
- Communication $a(x).P \mid \overline{a}\langle v \rangle.Q \mid R \rightarrow P\{v/x\} \mid Q \mid R$.

Replication

- Terminating system: if $P \rightarrow P'$, $\texttt{prefix}(P') < \texttt{prefix}(P)$.
- Replicated input: $!a(x).P$.
- $!a(x).P \mid \overline{a}\langle v \rangle.Q \mid R \rightarrow !a(x).P \mid P\{v/x\} \mid Q \mid R$.

Divergence

- $D_1 = !a(x).\overline{a}\langle x\rangle \mid \overline{a}\langle v\rangle$.

Divergence

- $D_1 = !a(x).\overline{a}\langle x\rangle \mid \overline{a}\langle v\rangle$. We have $D_1 \rightarrow D_1$.

Divergence

- $D_1 = !a(x).\overline{a}\langle x\rangle \mid \overline{a}\langle v\rangle$. We have $D_1 \rightarrow D_1$.
- $D_2 = !a(x).\overline{b}\langle x\rangle \mid !b(y).\overline{a}\langle y\rangle \mid \overline{a}\langle v\rangle$.

Divergence

- $D_1 = !a(x).\overline{a}\langle x\rangle \mid \overline{a}\langle v\rangle$. We have $D_1 \rightarrow D_1$.
- $D_2 = !a(x).\overline{b}\langle x\rangle \mid !b(y).\overline{a}\langle y\rangle \mid \overline{a}\langle v\rangle$.We have $D_2 \rightarrow\rightarrow D_2$.

Divergence

- $D_1 = !a(x).\overline{a}\langle x\rangle \mid \overline{a}\langle v\rangle$. We have $D_1 \rightarrow D_1$.
- $D_2 = !a(x).\overline{b}\langle x\rangle \mid !b(y).\overline{a}\langle y\rangle \mid \overline{a}\langle v\rangle$.We have $D_2 \rightarrow\rightarrow D_2$.
- $D_3 = c(y).!a(x).\overline{y}\langle x\rangle \mid \overline{a}\langle v\rangle \mid \overline{c}\langle a\rangle$.

Divergence

- $D_1 = !a(x).\overline{a}\langle x\rangle \mid \overline{a}\langle v\rangle$. We have $D_1 \rightarrow D_1$.
- $D_2 = !a(x).\overline{b}\langle x\rangle \mid !b(y).\overline{a}\langle y\rangle \mid \overline{a}\langle v\rangle$.We have $D_2 \rightarrow\rightarrow D_2$.
- $D_3 = c(y).!a(x).\overline{y}\langle x\rangle \mid \overline{a}\langle v\rangle \mid \overline{c}\langle a\rangle$.We have $D_3 \rightarrow D_1$.

Weight-based type systems

The idea

- Assigning level to names $a^k(x)$, $\overline{b}^l\langle v\rangle$.
- Controlling the replication: typing $!a^k(x).P$ only if every $\overline{c}^l\langle w\rangle$ in P is s.t. $k > l$.

Weight-based type systems

The idea

- Assigning level to names $a^k(x)$, $\overline{b}^l\langle v\rangle$.
- Controlling the replication: typing $!a^k(x).P$ only if every $\overline{c}^l\langle w\rangle$ in P is s.t. $k > l$.
- At each reduction step, a multiset measure decreases: trading an output k for outputs $(l_i)_i$, $l < k$.

Weight-based type systems

The idea

- Assigning level to names $a^k(x)$, $\overline{b}^l\langle v\rangle$.
- Controlling the replication: typing $!a^k(x).P$ only if every $\overline{c}^l\langle w\rangle$ in P is s.t. $k > l$.
- At each reduction step, a multiset measure decreases: trading an output k for outputs $(l_i)_i$, $l < k$.

Typing rule for replication

$$\frac{\vdash_\Gamma P : n \qquad \Gamma(a) = \sharp^k\, T \qquad \Gamma(x) = T \qquad k > n}{\vdash_\Gamma !a(x).P : 0}$$

A terminating example

$$c(x).!x(y).(\overline{b}\langle v_1\rangle \mid \overline{b}\langle v_2\rangle)$$

$$\mid\ !b(z).(\overline{c}\langle w_1\rangle \mid \overline{c}\langle w_2\rangle \mid \overline{c}\langle w_3\rangle)$$

$$\mid \overline{b}\langle v_3\rangle \mid \overline{c}\langle a\rangle \mid \overline{a}\langle v\rangle$$

A terminating example

$$c(x).!x(y).(\overline{b}\langle v_1\rangle \mid \overline{b}\langle v_2\rangle)$$

$$\mid\ !b(z).(\overline{c}\langle w_1\rangle \mid \overline{c}\langle w_2\rangle \mid \overline{c}\langle w_3\rangle)$$

$$\mid \overline{b}\langle v_3\rangle \mid \overline{c}\langle a\rangle \mid \overline{a}\langle v\rangle$$

- Typed with $c : 1; b : 2; a, x : 3$ as $3 > 2$ and $2 > 1$.
- Measure of possible reduction $\{2, 1, 3\} \rightarrow \{2, 3\} \rightarrow \{2, 2, 2\} \rightarrow 1, 1, 1, 2, 2 \rightarrow\rightarrow 1, 1, 1, 1, 1, 1, 1, 1, 1$

Ruling out the diverging examples

- $D_1 = !a(x).\overline{a}\langle x\rangle \mid \overline{a}\langle v\rangle$. If $a : n$ we get $n > n$.

Ruling out the diverging examples

- $D_1 = !a(x).\overline{a}\langle x\rangle \mid \overline{a}\langle v\rangle$. If $a : n$ we get $n > n$.
- $D_2 = !a(x).\overline{b}\langle x\rangle \mid !b(y).\overline{a}\langle y\rangle \mid \overline{a}\langle v\rangle$. If $a : n$ and $b : m$ then we get $n > m > n$.

Ruling out the diverging examples

- $D_1 = !a(x).\overline{a}\langle x\rangle \mid \overline{a}\langle v\rangle$. If $a : n$ we get $n > n$.
- $D_2 = !a(x).\overline{b}\langle x\rangle \mid !b(y).\overline{a}\langle y\rangle \mid \overline{a}\langle v\rangle$. If $a : n$ and $b : m$ then we get $n > m > n$.
- $D_3 = c(y).!a(x).\overline{y}\langle x\rangle \mid \overline{a}\langle v\rangle \mid \overline{c}\langle a\rangle$. Types ensures y and a have same type n, so we get $n > n$.

Increasing Expressiveness

More complex weight-based systems

- Handling sequence comparison $!a(x).b(y).P$.
- Allowing comparisons between names of same type: typechecking inductive structures.
- Using weight-based methods for termination in higher-order calculi.
- Using static/dynamic analysis.

Other methods

- Logical relations.
- Combining the two methods.