Statically Proving Behavioural Properties in the $\pi$-calculus via Dependency Analysis

PhD Defence

Maxime Gamboni

Instituto de Telecomunicações, Instituto Superior Técnico, Portugal

December 17th, 2010
Plan

- Statically Proving
- Behavioural Properties
- in the $\pi$-calculus
- via Dependency Analysis
Context: Request & Answer
Context: Proxy
Statically vs Dynamical Analysis

**Statically Proving** Behavioural Properties in the $\pi$-calculus via Dependency Analysis

**Definition (Model Checking)**
Finding Properties by simulating execution

**Definition (Statically Analysis)**
Finding Properties without running the program
Statically vs Dynamical Analysis

**Statically Proving** Behavioural Properties in the $\pi$-calculus via Dependency Analysis

**Definition (Model Checking)**
Finding Properties by simulating execution

**Definition (Statical Analysis)**
Finding Properties without running the program
Statically vs Dynamical Analysis

**Statically Proving** Behavioural Properties in the $\pi$-calculus via Dependency Analysis

**Definition (Model Checking)**
Finding Properties by simulating execution

**Definition (Statical Analysis)**
Finding Properties without running the program
Model Checking

Finding/Verifying properties by simulating execution
Type Systems

Finding/Verifying properties without running the program

My Type Inference System

\[ \Sigma \vdash \alpha : \sigma \rightarrow \Sigma ; \Xi \vdash \chi \rightarrow (\Sigma ; \Xi) \]
Type Systems

Finding/Verifying properties without running the program

My Type Inference System

\[
\begin{array}{c}
\Sigma \vdash \sigma_0 : \sigma_c \\
\downarrow
\end{array}
\Rightarrow
\left( \Sigma ; \equiv_{\psi} : \equiv_c \right)
\]
Type Systems

Finding/Verifying properties without running the program

My Type Inference System

\( \Sigma ; \xi_1 : \sigma_1 ; \xi_2 : \sigma_2 \rightarrow \Gamma \chi \rightarrow (\Sigma ; \xi_1 ; \xi_2) \)
Type Systems

Finding/Verifying properties without running the program

My Type Inference System

$$\Sigma \triangleright \chi \rightarrow (\Sigma; \Xi; \Xi)$$
Statically Proving **Behavioural Properties** in the $\pi$-calculus via Dependency Analysis

**Examples**
- Activeness (Receptiveness)
- Isolation
Behavioural Properties

Statically Proving **Behavioural Properties** in the $\pi$-calculus via Dependency Analysis

**Examples**

- Activeness (Receptiveness)
- Isolation
Behavioural Properties

Statically Proving Behavioural Properties in the $\pi$-calculus via Dependency Analysis

Examples

- Activeness (Receptiveness)
- Isolation
Behavioural Properties: Existential vs Universal

**Definition (Existential Property)**
Available *somewhere*. Good things happen *eventually*.

E.g. “Activeness”

**Definition (Universal Property)**
Available *everywhere*. Good things happen *constantly*.

E.g. “Isolation”
Behavioural Properties: Existential vs Universal

**Definition (Existential Property)**
Available *somewhere*. Good things happen *eventually*.

E.g. “Activeness”

**Definition (Universal Property)**
Available *everywhere*. Good things happen *constantly*.

E.g. “Isolation”
The π-calculus

Statically Proving Behavioural Properties in the π-calculus via Dependency Analysis

\[
\begin{align*}
&\text{Repeat} \quad \text{“New”} \\
&\ !n(xy) \cdot (\nu tf) (\overline{a} \langle tf \rangle \mid (t.y + f.x))
\end{align*}
\]

Repeat, "New", Concurrency, Receive, Sequence, Send, Choice
The $\pi$-calculus

Statically Proving Behavioural Properties in the $\pi$-calculus via Dependency Analysis

\[ \! n(xy) . (\nu t f) (\bar{a}\langle tf \rangle | (t.\bar{y} + f.\bar{x})) \]
The $\pi$-calculus

Example

$\bigcirc (qr) . \overline{\nabla} (qr') . r'(a) . \overline{r}(a)$
Dependency Analysis

Statically Proving Behavioural Properties in the $\pi$-calculus via Dependency Analysis

**Definition (Dependency $A \triangleleft B$)**

If you give me $B$, I’ll give you $A$.

$\bigcirc$ is isolated if $\nabla$ is isolated

$(\bigcirc_1) \triangleleft (\nabla_1)$
Dependency Analysis

Statically Proving Behavioural Properties in the $\pi$-calculus via Dependency Analysis

**Definition (Dependency $A \triangleleft B$)**

If you give me $B$, I’ll give you $A$.

- $\bigcirc$ is isolated if $\nabla$ is isolated

$(\bigcirc_1) \triangleleft (\nabla_1)$
Dependency Analysis

Definition (Dependency $A \preceq B$)

If you give me $B$, I'll give you $A$.

○ is isolated if ▼ is isolated

$(\bigcirc_1) \preceq (\nabla_1)$
Dependency Analysis

**Definition (Dependency \( A \bowtie B \))**

If you give me \( B \), I’ll give you \( A \).

\[ \bigcirc \text{ is isolated if } \triangledown \text{ is isolated} \]

\[ (\bigcirc_1) \bowtie (\triangledown_1) \]
Generic Type System

- Not specific to a property

Instantiation:
- Write semantic goals
- Rules parametrised by elementary rules
Generic Type System

- Not specific to a property

Instantiation:
- Write *semantic goals*
- Rules *parametrised by elementary rules*
Generic Type System

- Not specific to a property

Instantiation:

- Write *semantic goals*

- Rules *parametrised by elementary rules*
Generic Type System

- Not specific to a property

Instantiation:

- Write *semantic goals*
- Rules *parametrised by elementary rules*
## Contributions

<table>
<thead>
<tr>
<th>Type Language</th>
<th>Process Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection &amp; Branching</td>
<td>Choice</td>
</tr>
<tr>
<td>Activeness</td>
<td>Liveness</td>
</tr>
<tr>
<td>Determinism, Isolation, ...</td>
<td>Safety</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Causality</td>
</tr>
</tbody>
</table>

### Generic Type System:
- Decidable
- Constructs Logical Formulae
- Sound
- Compositional
Conclusion

“Statically Proving Behavioural Properties in the $\pi$-calculus via Dependency Analysis”
Questions
Supplementary Material

- Types & Multiplicities
- Choice
- Algebra
- Semantics
- Type Systems
- Properties
- Soundness
- Future Work
Types & Multiplicities

**Behavioural Statements $\Delta$, $\Xi$, ...**

- $\Delta ::= \ldots$
- $\Delta \lor \Delta$
- $\Delta + \Delta$
- $\Delta \land \Delta$
- $\Delta \triangleleft \Delta$
- $p_k$
- $\bot$
- $\top$
- $p^m$

**Multiplicities**

- $m ::= 0 | 1 | \omega | \star$
Choice

Definition (Selection $A \lor B$)
I will either behave like $A$ or like $B$

Definition (Branching $A + B$)
You can make me do $A$ or $B$
Choice

Definition (Selection $A \lor B$)
I will either behave like $A$ or like $B$

Definition (Branching $A + B$)
You can make me do $A$ or $B$
Choice

**Definition (Selection \( A \lor B \))**

I will either behave like \( A \) or like \( B \)

**Definition (Branching \( A + B \))**

You can make me do \( A \) or \( B \)
Choice Examples (I)

- Data Encodings

\[ b := \text{True} \quad \overset{\text{def}}{=} \quad !b(tf).\overline{t} \]

\[ b := \text{False} \quad \overset{\text{def}}{=} \quad !b(tf).\overline{f} \]

\[ \text{If } b \text{ Then } P \text{ Else } Q \quad \overset{\text{def}}{=} \quad \overline{b}(\nu tf).(t.P + f.Q) \]
Choice Examples (II)

- Client-Server Conversations

\[
\prod(v_s).s(more, done).\text{more}(v_s, 2).
\]
\[
s(more, done).\text{more}(v_s, 5).
\]
\[
s(more, done).\text{done}(v_s).s(x).\text{print}\langle x \rangle
\]

\[
!\prod(s).\overline{p_0}\langle 1, s \rangle \mid !p_0(t, s).\overline{s}(v\more, done).
\]
\[
(more(s, n).\overline{p_0}\langle t \times n, s \rangle + \text{done}(s).\overline{s}\langle r \rangle)
\]
Algebra

**Spatial Operators**
- Parallel Composition $\Gamma_1 \odot \Gamma_2$
- Restriction $(\nu x)\Gamma$

**Logical Operators**
- Equivalence $\cong$
- Weakening $\trianglelefteq$
- Reduction $\leftrightarrow$

**Dynamic Operator**
- Transition Operator $\Gamma \xrightarrow{\mu} (\Gamma \triangledown \mu)$
Definition (Universal Semantics)

A \( (\Gamma; P) \) typed process is correct wrt. universal semantics (\( \Gamma \models_{\mathcal{U}} P \)) if, for all transition sequences (\( \Gamma; P \xrightarrow{\tilde{\mu}} (\Gamma'; P') \)), the local component of \( \Gamma' \) being \( \bigvee_{i \in I} p_i k_i \triangleleft \varepsilon_i \): for all \( i \in I \) with \( k_i \in \mathcal{U}, \text{good}_{k_i}(p_i \triangleleft \varepsilon_i, (\Gamma'; P')) \) holds.
Semantics (Existential)

(abbreviated) Existential Semantics

A typed process \((\Gamma; P)\) is correct \(\text{("}\Gamma \models P\text{"})\), if \(\exists\) a strategy \(f\) s.t.

For any sequence

\((\Gamma; P) = (\Gamma_0; P_0) \cdot \cdot \cdot \xrightarrow{\tilde{\mu}_i} (\Gamma'_i; P'_i) \xrightarrow{f} (\Gamma_{i+1}; P_{i+1}) \cdot \cdot \cdot\), let (for all \(i\)) \(\mu_i\) be the label of \((\Gamma'_i; P'_i) \xrightarrow{f} (\Gamma_{i+1}; P_{i+1})\).

Then \(\exists\) a resource \(p_k\) and \(n \geq 0\) such that:

1. \(\forall i : (p_k \triangleleft \text{dep}_K(\mu_i)) \leq \Gamma'_i\)

2. \(\exists \varepsilon : (p_k \triangleleft \varepsilon) \leq \Gamma_n\) and \(\text{good}_k(p \triangleleft \varepsilon, (\Gamma_n; P_n))\).
Type System (Universal)

\[
\forall i : \Gamma_i \vdash \kappa P_i \quad \Rightarrow \quad \Gamma_1 \otimes \Gamma_2 \vdash \kappa P_1 | P_2 \quad (U-PAR)
\]

\[
\Gamma \vdash \kappa P \quad \Gamma(x) = \sigma \quad \Rightarrow \quad (\nu x) \Gamma \vdash \kappa (\nu x : \sigma) P \quad (U-RES)
\]

\[
\forall i : (\sum_i; \Xi_{Li} \downarrow \Xi_{Ei}) \vdash \kappa G_i P_i \quad \Xi_E \leq \bigwedge_i \Xi_{Ei} \quad \Rightarrow \quad \bigwedge_i \Xi_{Li} \downarrow \Xi_E \quad \vdash \kappa \sum_i G_i P_i \quad (U-SUM)
\]

\[
\Gamma \vdash \kappa P \quad \text{sub}(G) = p \quad \text{obj}(G) = \tilde{x} \quad \Rightarrow \quad \left( p : \sigma; \blacktriangleleft \; \sigma^m \wedge \bar{p}^{m'} \right) \quad \circ \quad \left( ; p\#(G) \blacktriangleleft \right) \quad \circ \quad !\text{if} \; \#(G) = \omega \left( \nu \text{bn}(G) \right) \left( \Gamma \quad \circ \quad \sigma[\tilde{x}] \quad \circ \quad \bigwedge_{k \in \kappa} \text{prop}_k(\sigma, G, m, m') \blacktriangleleft \right) \quad \vdash \kappa G.P \quad (U-PRE)
\]
Type System (Existential)

\[ \forall i : \Gamma_i \vdash \kappa \ P_i \quad (\text{E-PAR}) \]
\[ \Gamma \vdash \kappa \ P \quad (\nu x) \Gamma \vdash \kappa \ (\nu x : \sigma) \ P \quad (\text{E-RES}) \]

\[ \forall i : (\Sigma_i ; \Xi_{Li} \triangleright \Xi_{Ei}) \vdash \kappa \ G_i.P_i \]
\[ \Xi_E \leq \bigwedge_i \Xi_{Ei} \quad (\text{E-SUM}) \]

\[ \Gamma \vdash \kappa \ P \quad \text{sub}(G) = p \quad \text{obj}(G) = \bar{x} \quad (\text{E-PRE}) \]
\[ \left( p : \sigma ; \triangleright p^m \land \bar{p}^{m'} \right) \quad \circ \]
\[ \left( ; p^\#(G) \triangleright \right) \quad \circ \]
\[ \text{! if } \#(G) = \omega \left( \nu \text{bn}(G) \right) \left( \Gamma \triangleright \text{dep}_\kappa (G) \right) \quad \circ \]
\[ \bar{\sigma}[ar{x}] \triangleright (\text{dep}_\kappa (G) \land \bar{p}_R) \quad \circ \]
\[ ( ; \bigwedge_{k \in \kappa} \text{prop}_k(\sigma, G, m, m') \triangleright ) \quad \vdash \kappa \ G.P \]
Properties

- **A** — Activeness
  
  \[
  \text{prop}_A(G, \sigma, m, m') = \begin{cases} 
  \text{sub}(G)_A & \text{if } \#(G) = \omega \text{ or } m' \neq \star \\
  \top & \text{otherwise}
  \end{cases}
  \]

- **R** — Responsiveness
- **D** — Determinism (Functionality)
- **I** — Isolation
- **df** — Lock-Freedom
- **N** — Non-Reachability
- **ϖ** — Termination
Properties

- **A** — Activeness
- **R** — Responsiveness

\[
\text{prop}_R(\sigma, G, m, m') = \text{sub}(G)_{\not\triangleleft} \left\{ \begin{array}{ll}
\sigma[\text{obj}(G)] & \text{if } G \text{ is an input} \\
\overline{\sigma}[\text{obj}(G)] & \text{if } G \text{ is an output}
\end{array} \right.
\]

- **D** — Determinism (Functionality)
- **I** — Isolation
- **df** — Lock-Freedom
- **N** — Non-Reachability
- **ϖ** — Termination
Properties

- **A** — Activeness
- **R** — Responsiveness
- **D** — Determinism (Functionality)

\[ \varphi_D(\sigma, G, m, m') \overset{\text{def}}{=} \begin{cases} \bot & \text{if } \star \in \{m, m'\} \text{ and } \omega \notin \{m, m'\} \\ \text{sub}(G)_D & \text{otherwise} \end{cases} \]

\[ \varphi_D(\{p_i\}_i, \Xi) \overset{\text{def}}{=} \begin{cases} \bot & \text{if } \Xi \text{ has concurrent environment } p_i \\ \top & \text{otherwise} \end{cases} \]

- **I** — Isolation
- **df** — Lock-Freedom
- **N** — Non-Reachability
- **ϖ** — Termination
Properties

- **A** — Activeness
- **R** — Responsiveness
- **D** — Determinism (Functionality)
- **I** — Isolation

\[ \varphi_I(\sigma, G, m, m') = \text{sub}(G)_I \]

- **df** — Lock-Freedom
- **N** — Non-Reachability
- **\(\varpi\)** — Termination
Properties

- **A** — Activeness
- **R** — Responsiveness
- **D** — Determinism (Functionality)
- **I** — Isolation
- **df** — Lock-Freedom

\[ \text{prop}_{\text{df}}(G, \sigma, m, m') = \text{proc}_{\text{df}} < \text{sub}(G) \]

- **N** — Non-Reachability
- **\(\omega\)** — Termination
Properties

- A — Activeness
- R — Responsiveness
- D — Determinism (Functionality)
- I — Isolation
- df — Lock-Freedom
- N — Non-Reachability

\[
\text{prop}_N(G, \sigma, m, m') \overset{\text{def}}{=} \text{sub}(G)_N \triangleleft \perp
\]

- \( \upomega \) — Termination
Properties

- **A** — Activeness
- **R** — Responsiveness
- **D** — Determinism (Functionality)
- **I** — Isolation
- **df** — Lock-Freedom
- **N** — Non-Reachability
- **ϖ** — Termination

\[ \text{prop}_N(G, \sigma, m, m') \overset{\text{def}}{=} \text{sub}(G)_N \not\!\!\not\!\!\not\!\!\not\!\!\not\downarrow \land \tau_N \not\!\!\not\!\!\not\!\!\not\!\!\not\not\!\!\not\!\!\not\downarrow \text{sub}(G)_N \]
Universal Soundness

- Based on transition sequences?
  Semantic Predicates aren’t transition based!

- Based on contextual semantics?
  “$\Delta_1 \vdash \Delta_2 \models P$ if $\forall Q$ s.t. $\Delta_2 \vdash Q$: $\Delta_1 \models P \mid Q$.”
  The definition is circular!

- Implicit definition?
  “The set of correct typed processes is the largest that satisfies the above”
  There are many solutions!

- Stricter implicit definition?
  “The set of correct typed processes is the intersection of all those that satisfy the above”
  The intersection is empty!

- To be continued . . .
Universal Soundness

- Based on transition sequences?
  Semantic Predicates aren’t transition based!

- Based on contextual semantics?
  “Δ₁ ≤ Δ₂ |⇒ P if ∀Q s.t. Δ₂ ⊨ Q: Δ₁ |⇒ P | Q.”
  The definition is circular!

- Implicit definition?
  “The set of correct typed processes is the largest that satisfies the above”
  There are many solutions!

- Stricter implicit definition?
  “The set of correct typed processes is the intersection of all those that satisfy the above”
  The intersection is empty!

- To be continued . . .
Universal Soundness

- Based on transition sequences?
  Semantic Predicates aren’t transition based!
- Based on contextual semantics?
  “Δ₁ ⊨ Δ₂ |≡ P if ∀ Q s.t. Δ₂ ⊨ Q: Δ₁ ⊨ P | Q.”
  The definition is circular!
- Implicit definition?
  “The set of correct typed processes is the largest that satisfies the above”
  There are many solutions!
- Stricter implicit definition?
  “The set of correct typed processes is the intersection of all those that satisfy the above”
  The intersection is empty!
- To be continued . . .
Universal Soundness

- Based on transition sequences?
  Semantic Predicates aren’t transition based!

- Based on contextual semantics?
  “$\Delta_1 \ll \Delta_2 \models P$ if $\forall Q$ s.t. $\Delta_2 \vdash Q: \Delta_1 \models P \mid Q$.”
  The definition is circular!

- Implicit definition?
  “The set of correct typed processes is the largest that satisfies the above”
  There are many solutions!

- Stricter implicit definition?
  “The set of correct typed processes is the intersection of all those that satisfy the above”
  The intersection is empty!

- To be continued . . .
Universal Soundness

- Based on transition sequences?
  Semantic Predicates aren’t transition based!

- Based on contextual semantics?
  \[ \Delta_1 \prec \Delta_2 \models P \text{ if } \forall Q \text{ s.t. } \Delta_2 \vdash Q : \Delta_1 \models P | Q. \]
  The definition is circular!

- Implicit definition?
  “The set of correct typed processes is the largest that satisfies the above”
  There are many solutions!

- Stricter implicit definition?
  “The set of correct typed processes is the intersection of all those that satisfy the above”
  The intersection is empty!

- To be continued . . .
Universal Soundness

- Based on transition sequences?
  Semantic Predicates aren’t transition based!

- Based on contextual semantics?
  “$\Delta_1 \triangleleft \Delta_2 \models P$ if $\forall Q$ s.t. $\Delta_2 \vdash Q$: $\Delta_1 \models P \mid Q$.”
  The definition is circular!

- Implicit definition?
  “The set of correct typed processes is the largest that satisfies the above”
  There are many solutions!

- Stricter implicit definition?
  “The set of correct typed processes is the intersection of all those that satisfy the above”
  The intersection is empty!

- To be continued . . .
Existential Soundness

**Structural Liveness Strategies**

\[
\rho ::= \pi \delta \mid l \mid \ldots
\]
\[
\delta ::= \div \rho \mid [s]
\]
\[
\pi ::= (l|\rho) \mid (l|\bullet) \mid (\bullet|\rho)
\]
\[
s ::= p_1 + p_2 + p_3 \ldots
\]

\(l\): Guard reference

\(\bullet\): Environment

\((l|\rho)\): Make \(l\) and \(\rho\) communicate.
Future Work

- Generic Universal Soundness Proof
- Recursivity and Bounded Channels.
- Channel Type Reconstruction.
- Software Implementation.
<table>
<thead>
<tr>
<th>Statically</th>
<th>Behavioural</th>
<th>$\pi$-calculus</th>
<th>Dependency</th>
</tr>
</thead>
</table>

- Link to Appendices