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

PhD Defence

Maxime Gamboni

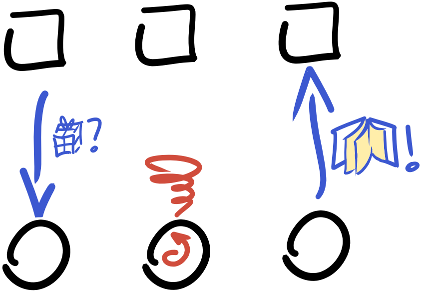
Instituto de Telecomunicações, Instituto Superior Técnico, Portugal

December 17th, 2010

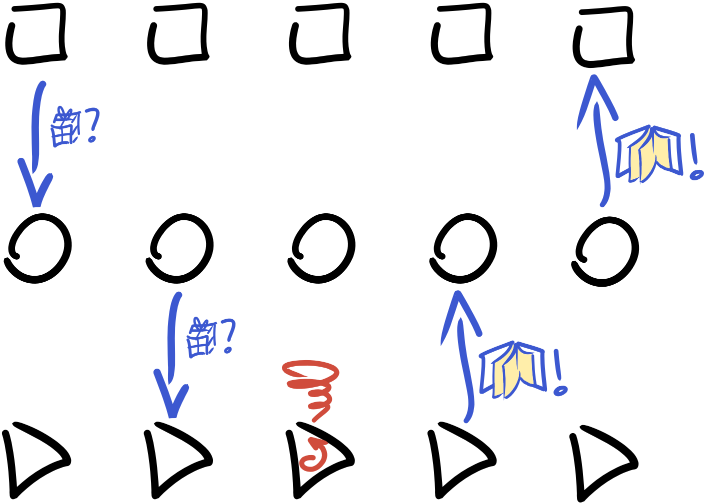
Plan

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

Context: Request & Answer



Context: Proxy



Statical vs Dynamical Analysis

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

Definition (Model Checking)

Finding Properties by simulating execution

Definition (Statical Analysis)

Finding Properties without running the program

Statical vs Dynamical Analysis

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

Definition (Model Checking)

Finding Properties by simulating execution

Definition (Statical Analysis)

Finding Properties without running the program

Statical vs Dynamical Analysis

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

Definition (Model Checking)

Finding Properties by simulating execution

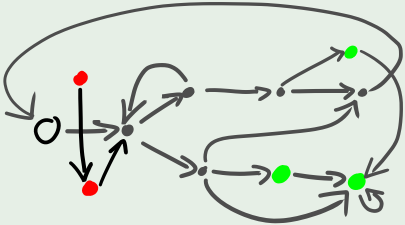
Definition (Statical Analysis)

Finding Properties without running the program

Model Checking

Finding/Verifying properties by simulating execution

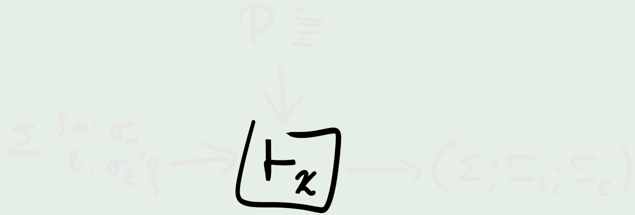
Model Checking



Type Systems

Finding/Verifying properties without running the program

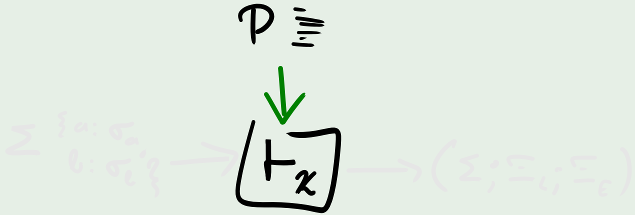
My Type Inference System



Type Systems

Finding/Verifying properties without running the program

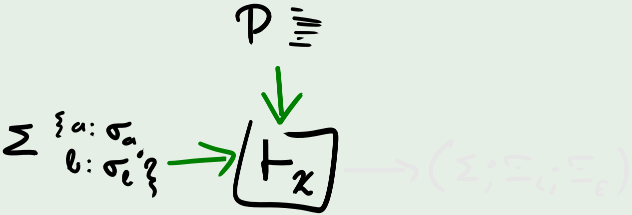
My Type Inference System



Type Systems

Finding/Verifying properties without running the program

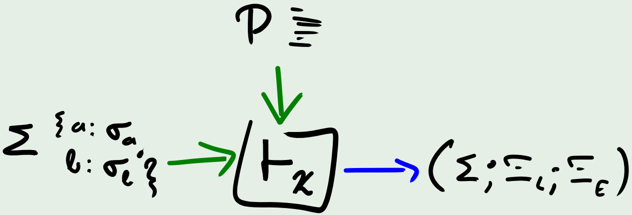
My Type Inference System



Type Systems

Finding/Verifying properties without running the program

My Type Inference System



Behavioural Properties

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

Examples

- Activeness (Receptiveness)
- Isolation

Behavioural Properties

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

Examples

- Activeness (Receptiveness)
- Isolation

Behavioural Properties

Statically Proving Behavioural Properties in the π-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"

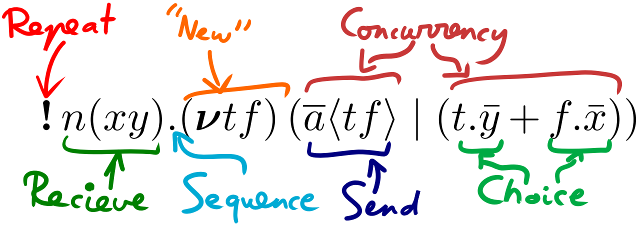
The π-calculus

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

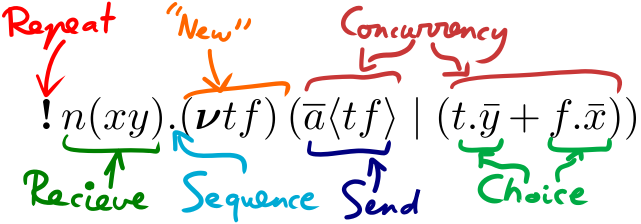
The π-calculus

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

$$!\,n(xy).(\boldsymbol{\nu} t f)\,(\overline{a}\langle t f\rangle \mid (t.\bar{y} + f.\bar{x}))$$



The π-calculus



$$!\,n(xy).(\boldsymbol{\nu} t f)\,(\overline{a}\langle t f\rangle \mid (t.\bar{y} + f.\bar{x}))$$

Example

$$\bigcirc(qr).\overline{\nabla}\langle qr'\rangle.r'(a).\overline{r}\langle a\rangle$$

Dependency Analysis

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

Definition (Dependency $A \lhd B$)

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

○ *is* **I***solated if* ▽ *is* **I***solated*

$$(\bigcirc_{\mathbf{I}}) \lhd (\triangledown_{\mathbf{I}})$$

Dependency Analysis

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

Definition (Dependency $A \lhd B$)

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

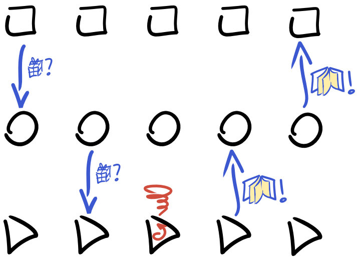
$\bigcirc$ *is* **I***solated if* $\bigtriangledown$ *is* **I***solated*

$$(\bigcirc_{\mathbf{I}}) \lhd (\bigtriangledown_{\mathbf{I}})$$

Dependency Analysis

Definition (Dependency $A \lhd B$)

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



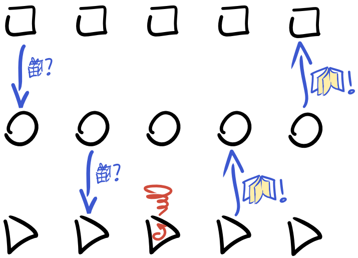
○ *is* **I***solated if* ▽ *is* **I***solated*

$$(\bigcirc_{\mathbf{I}}) \lhd (\triangledown_{\mathbf{I}})$$

Dependency Analysis

Definition (Dependency $A \lhd B$)

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



$\bigcirc$ *is* **I***solated if* $\triangledown$ *is* **I***solated*

$$(\bigcirc_{\mathbf{I}}) \lhd (\triangledown_{\mathbf{I}})$$

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

Type Language		Process Behaviour
Selection & Branching	$A \vee B$, $p + q$	Choice
Activeness	$p_{\mathbf{A}}$	Liveness
Determinism, Isolation, ...	$p_{\mathbf{D}}$, $p_{\mathbf{I}}$, ...	Safety
Dependencies	$A \lhd B$	Causality

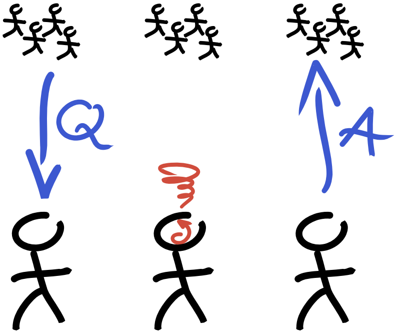
Generic Type System:

- Decidable
- Constructs Logical Formulæ
- Sound
- Compositional

Conclusion

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

Questions



Supplementary Material

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

Types & Multiplicities

Behavioural Statements Δ, Ξ, ...

$$\Delta \quad ::= \quad \Delta \vee \Delta \;\mid\; \Delta + \Delta \;\mid\; \Delta \wedge \Delta \;\mid\; \Delta \lhd \Delta \;\mid\; p_k \;\mid\; \bot \;\mid\; \top \;\mid\; p^m$$

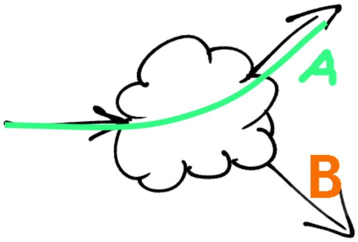
Multiplicities

$$m \quad ::= \quad 0 \;\mid\; 1 \;\mid\; \omega \;\mid\; \star$$

Choice

Definition (Selection $A \vee B$)

I will either behave like A or like B



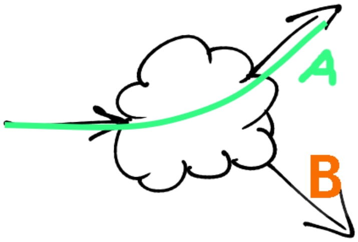
Definition (Branching $A + B$)

You can make me do A or B

Choice

Definition (Selection $A \vee B$)

I will either behave like A or like B



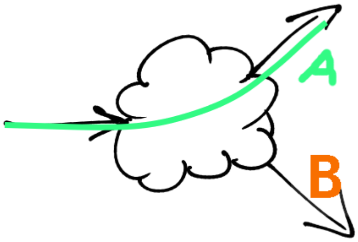
Definition (Branching $A + B$)

You can make me do A or B

Choice

Definition (Selection $A \vee B$)

I will either behave like A or like B



Definition (Branching $A + B$)

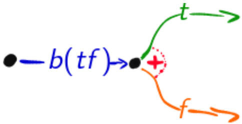
You can make me do A or B

Choice Examples (I)

- Data Encodings

$$b := \text{True} \stackrel{\text{def}}{=} !\, b(tf).\overline{t}$$

$$b := \text{False} \stackrel{\text{def}}{=} !\, b(tf).\overline{f}$$

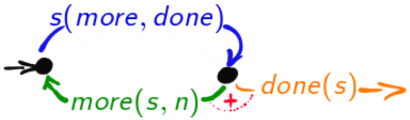


$$\text{If } b \text{ Then } P \text{ Else } Q \stackrel{\text{def}}{=} \overline{b}(\nu tf).(t.P+f.Q)$$

Choice Examples (II)

- Client-Server Conversations

$$\overline{prod}(\boldsymbol{\nu} s).s(more, done).\overline{more}(\boldsymbol{\nu} s, 2).$$
$$s(more, done).\overline{more}(\boldsymbol{\nu} s, 5).$$
$$s(more, done).\overline{done}(\boldsymbol{\nu} s).s(x).\overline{print}\langle x\rangle$$



$$!\,prod(s).\overline{p_0}\langle 1, s\rangle \quad | \quad !\,p_0(t, s).\overline{s}(\boldsymbol{\nu} more, done).$$
$$\big(more(s, n).\overline{p_0}\langle t \times n, s\rangle + done(s).\overline{s}\langle r\rangle\big)$$

Algebra

Spatial Operators

Parallel Composition $\Gamma_1 \odot \Gamma_2$, Restriction $(\boldsymbol{\nu} x)\,\Gamma$, ...

Logical Operators

Equivalence $\cong$, Weakening $\preceq$, Reduction $\hookrightarrow$, ...

Dynamic Operator

Transition Operator $\Gamma \xrightarrow{\mu} (\Gamma \wr \mu)$.

Semantics (Universal)

Definition (Universal Semantics)

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

Semantics (Existential)

(Abbreviated) Existential Semantics

A typed process $(\Gamma; P)$ is *correct* ("$\Gamma \models P$"), if $\exists$ a strategy f s.t. For any sequence

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

Then $\exists$ a resource p_k and $n \geqslant 0$ such that:

1. $\forall i : (p_k \lhd \mathrm{dep}_{\mathcal{K}}(\mu_i)) \preceq \Gamma'_i$
2. $\exists \varepsilon : (p_k \lhd \varepsilon) \preceq \Gamma_n$ and $\mathrm{good}_k(p \lhd \varepsilon, (\Gamma_n; P_n))$.

Type System (Universal)

$$\frac{\forall i : \Gamma_i \vdash_{\mathcal{K}} P_i}{\Gamma_1 \odot \Gamma_2 \vdash_{\mathcal{K}} P_1 \mid P_2} \text{ (U-PAR)} \qquad \frac{\Gamma \vdash_{\mathcal{K}} P \qquad \Gamma(x) = \sigma}{(\boldsymbol{\nu} x)\, \Gamma \vdash_{\mathcal{K}} (\boldsymbol{\nu} x : \sigma)\, P} \text{ (U-RES)}$$

$$\frac{\begin{array}{c}\forall i : (\Sigma_i; \Xi_{\mathrm{L}i} \blacktriangleleft \Xi_{\mathrm{E}i}) \vdash_{\mathcal{K}} G_i.P_i \\ \Xi_{\mathrm{E}} \preceq \bigwedge_i \Xi_{\mathrm{E}i}\end{array}}{(\bigwedge_i \Sigma_i; \bigwedge_{k \in \mathcal{K}} \mathsf{sum}_k(\{p_i\}_i, \Xi_{\mathrm{E}}) \wedge \bigvee_i \Xi_{\mathrm{L}i} \blacktriangleleft \Xi_{\mathrm{E}}) \vdash_{\mathcal{K}} \sum_i G_i.P_i} \text{ (U-SUM)}$$

$$\frac{\Gamma \vdash_{\mathcal{K}} P \quad \mathsf{sub}(G) = p \quad \mathsf{obj}(G) = \tilde{x}}{\begin{array}{r}\left(p : \sigma; \blacktriangleleft p^m \wedge \bar{p}^{m'}\right) \odot \\ \left(; p^{\#(G)} \blacktriangleleft\right) \odot \\ !_{\text{if } \#(G) = \omega}\, (\boldsymbol{\nu} \mathsf{bn}(G)) \Big(\Gamma \odot \\ \overline{\sigma}[\tilde{x}] \odot \\ (; \bigwedge_{k \in \mathcal{K}} \mathsf{prop}_k(\sigma, G, m, m') \blacktriangleleft) \Big) \vdash_{\mathcal{K}} G.P\end{array}} \text{ (U-PRE)}$$

Type System (Existential)

$$\frac{\forall i : \Gamma_i \vdash_{\mathcal{K}} P_i}{\Gamma_1 \odot \Gamma_2 \vdash_{\mathcal{K}} P_1 \mid P_2}\ (\text{E-PAR}) \qquad \frac{\Gamma \vdash_{\mathcal{K}} P \qquad \Gamma(x) = \sigma}{(\boldsymbol{\nu} x)\,\Gamma \vdash_{\mathcal{K}} (\boldsymbol{\nu} x : \sigma)\, P}\ (\text{E-RES})$$

$$\frac{\begin{array}{c}\forall i : (\Sigma_i; \Xi_{\mathrm{L}i} \blacktriangleleft \Xi_{\mathrm{E}i}) \vdash_{\mathcal{K}} G_i.P_i \\ \Xi_{\mathrm{E}} \preceq \bigwedge_i \Xi_{\mathrm{E}i}\end{array}}{(\bigwedge_i \Sigma_i; \bigwedge_{k \in \mathcal{K}} \mathsf{sum}_k(\{p_i\}_i, \Xi_{\mathrm{E}}) \wedge \bigvee_i \Xi_{\mathrm{L}i} \blacktriangleleft \Xi_{\mathrm{E}}) \vdash_{\mathcal{K}} \sum_i G_i.P_i}\ (\text{E-SUM})$$

$$\frac{\Gamma \vdash_{\mathcal{K}} P \quad \mathsf{sub}(G) = p \quad \mathsf{obj}(G) = \tilde{x}}{\begin{array}{r}\left(p : \sigma; \blacktriangleleft p^{m} \wedge \bar{p}^{m'}\right) \odot \\ \left(; p^{\#(G)} \blacktriangleleft\right) \odot \\ !_{\text{if } \#(G) = \omega}\ (\boldsymbol{\nu}\mathsf{bn}(G)) \Big(\Gamma \lhd \color{red}{\mathsf{dep}_{\mathcal{K}}(G)} \odot \\ \bar{\sigma}[\tilde{x}] \lhd \color{red}{(\mathsf{dep}_{\mathcal{K}}(G) \wedge \bar{p}_{\mathbf{R}})} \odot \\ (; \bigwedge_{k \in \mathcal{K}} \mathsf{prop}_k(\sigma, G, m, m') \blacktriangleleft) \Big) \quad \vdash_{\mathcal{K}} G.P\end{array}}\ (\text{E-PRE})$$

Properties

- **A** — Activeness

$$\text{prop}_{\mathbf{A}}(G, \sigma, m, m') = \begin{cases} \text{sub}(G)_{\mathbf{A}} & \text{if } \#(G) = \omega \text{ or } m' \neq \star \\ \top & \text{otherwise} \end{cases}$$

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

Properties

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

$$\mathrm{prop}_{\mathbf{R}}(\sigma, G, m, m') = \mathrm{sub}(G)_{\mathbf{R}} \lhd \begin{cases} \sigma[\mathrm{obj}(G)] & \text{if } G \text{ is an input} \\ \overline{\sigma}[\mathrm{obj}(G)] & \text{if } G \text{ is an output} \end{cases}$$

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

Properties

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

$$\varphi_{\mathbf{D}}(\sigma, G, m, m') \stackrel{\text{def}}{=} \begin{cases} \perp & \text{if } \star \in \{m, m'\} \text{ and } \omega \notin \{m, m'\} \\ \overline{\text{sub}(G)_{\mathbf{D}}} & \text{otherwise} \end{cases}$$

$$\varphi_{\mathbf{D}}(\{p_i\}_i, \Xi) \stackrel{\text{def}}{=} \begin{cases} \perp & \text{if } \Xi \text{ has concurrent environment } p_i \\ \top & \text{otherwise} \end{cases}$$

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

Properties

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

$$\varphi_{\mathbf{I}}(\sigma, G, m, m') = \overline{\mathsf{sub}(G)}_{\mathbf{I}}$$

- **df** — Lock-Freedom
- **N** — Non-Reachability
- ϖ — Termination

Properties

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

$$\mathsf{prop}_{\mathbf{df}}(G, \sigma, m, m') = \mathsf{proc}_{\mathbf{df}} \lhd \overline{\mathsf{sub}(G)}_{\mathbf{A}}$$

- **N** — Non-Reachability
- ϖ — Termination

Properties

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

$$\mathsf{prop}_{\mathbf{N}}(G, \sigma, m, m') \stackrel{\text{def}}{=} \mathsf{sub}(G)_{\mathbf{N}} \lhd \bot$$

- ϖ — Termination

Properties

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

$$\mathsf{prop}_{\mathbf{N}}(G, \sigma, m, m') \stackrel{\mathrm{def}}{=} \mathsf{sub}(G)_{\mathbf{N}} \lhd \perp \wedge \tau_{\mathbf{N}} \lhd \overline{\mathsf{sub}(G)}_{\mathbf{N}}$$

Universal Soundness

- Based on transition sequences?
  Semantic Predicates aren't transition based!
- Based on contextual semantics?
  "$\Delta_1 \lhd \Delta_2 \models P$ 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?
  "$\Delta_1 \lhd \Delta_2 \models P$ 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?
  "$\Delta_1 \lhd \Delta_2 \models P$ 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?
  "$\Delta_1 \lhd \Delta_2 \models P$ 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?
  "$\Delta_1 \lhd \Delta_2 \models P$ 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?
  "$\Delta_1 \lhd \Delta_2 \models P$ 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

$$
\begin{aligned}
\rho &::= \pi\delta \quad | \quad \mathfrak{l} \quad | \quad \cdots \\
\delta &::= \div\ \rho \quad | \quad [\,s\,] \\
\pi &::= (\mathfrak{l}|\rho) \quad | \quad (\mathfrak{l}|\bullet) \quad | \quad (\bullet|\rho) \\
s &::= p_1 + p_2 + p_3 \ldots
\end{aligned}
$$

$\mathfrak{l}$: Guard reference

$\bullet$: Environment

$(\mathfrak{l}|\rho)$: Make $\mathfrak{l}$ and ρ communicate.

Future Work

- Generic Universal Soundness Proof
- Recursivity and **B**ounded Channels.
- Channel Type Reconstruction.
- Software Implementation.

Link to Appendices