On Hierarchical Communication Topologies in the $\pi$-calculus

Emanuele D’Osualdo$^1$ Luke Ong$^2$

$^1$TU Kaiserslautern
$^2$University of Oxford

CPDS 2016
The Problem

Goal: Automated analysis of concurrent systems.

Challenges:

- Unbounded process creation + message passing
- Dynamic reconfiguration of communication topology
- Turing completeness
The Client/Server example

\[ S[s] := !s(x).(\nu d. \overline{x}\langle d\rangle) \]

\[ C[s, m] := \overline{s}\langle m\rangle \parallel m(x).C[s, m] \]

\[ E[s] := !\tau.(\nu m.C[s, m]) \]

\[ \nu s.(S[s] \parallel E[s]) \]
Property of interest: mailboxes are bounded by 1

Typical abstractions ignore the topology: the property cannot be proven

Alternatively we can prove the property using suitable inductive invariants
An inductive invariant
- The picture represents a set of configurations: each bubble can be cloned any number of times.
- The invariant contains the initial configuration: instantiate once the outer bubble and zero times each inner bubble.
- The invariant is closed under reductions.
- The invariant satisfies the property: there is at most one message in each mailbox.

**Problem:** such invariants do not always exist for arbitrary $\pi$-terms.

**Solution:** there is a fragment of $\pi$-calculus for which such invariants always exist.
If the simple paths of the reachable terms have bounded length, the initial term is **Depth Bounded**

If a system is Depth Bounded then some semantic properties are decidable

- termination
- coverability

One of the most expressive fragments of $\pi$-calculus to date

*On Boundedness in Depth in the $\pi$-calculus*

R. Meyer, 2008
Depth boundedness is undecidable!

And checking if a term is bounded in depth by a given $k$ has \emph{non primitive recursive} complexity
Depth boundedness is **undecidable**!

And checking if a term is bounded in depth by a given $k$ has *non primitive recursive* complexity.

We need more structure: Hierarchical systems.
Depth boundedness is **undecidable**!

And checking if a term is bounded in depth by a given $k$ has *non primitive recursive* complexity.

We need more structure: Hierarchical systems.

Key contribution: a type system to check/infer if a system is hierarchical.
Example Hierarchy
Example Hierarchy
Example Hierarchy
Example Hierarchy

server
mailb
data
Example Hierarchy
\[ S[s] := !s(x).(\nu(d : data).\overline{x}(d)) \]

\[ C[s, m] := \overline{s}(m) \parallel m(x).C[s, m] \]

\[ E[s] := !\tau.(\nu(m : mailb).C[s, m]) \]

\[ \nu(s : server).((S[s] \parallel E[s]) \parallel C[s, m]) \]
$S[s] := !s(x). (\nu(d : \text{data}). \overline{x}(d))$

$C[s, m] := \overline{s}(m) \parallel m(x). C[s, m]$

$E[s] := !\tau. (\nu(m : \text{mailb[data]}). C[s, m])$

$\nu(s : \text{server[mailb[data]]}). (S[s] \parallel E[s])$
$\forall a.\forall b.\forall c. (P[a] \parallel Q[a, b] \parallel R[c, a])$
$\nu_a.\nu_b.\nu_c. (P[a] \parallel Q[a, b] \parallel R[c, a]) \equiv \nu_b.\nu_a. (P[a] \parallel Q[a, b] \parallel \nu_c R[c, a])$
$\nu_a.\nu b.\nu c. (P[a] \parallel Q[a, b] \parallel R[c, a]) \equiv \nu a. (P[a] \parallel \nu b. Q[a, b] \parallel \nu c. R[c, a])$
$\nu(a : t_1).\nu(b : t_2).\nu(c : t_2).(P[a] \parallel Q[a,b] \parallel R[c,a])$
$\nu(a : t_1).\nu(b : t_2).\nu(c : t_2). (P[a] \parallel Q[a, b] \parallel R[c, a])$

\[ \mathcal{T} \]

\[ \begin{array}{c}
\text{t}_1 \\
\text{t}_2 \\
\end{array} \]
\( \nu(a : t_1) . \nu(b : t_2) . \nu(c : t_2) . (P[a] \parallel Q[a, b] \parallel R[c, a]) \)
\( \nu(a: t_1) \cdot \nu(b: t_2) \cdot \nu(c: t_2). (P[a] \parallel Q[a, b] \parallel R[c, a]) \)
$\nu(a:t_1).\nu(b:t_2).\nu(c:t_2).(P[a] \parallel Q[a,b] \parallel R[c,a])$ is $\mathcal{T}$-shaped because at least one of its presentations respects $\mathcal{T}$
The Client/Server example is $\mathcal{T}$-shaped.
The Client/Server example is $\mathcal{T}$-shaped

Every reachable term is $\mathcal{T}$-shaped

(but note that the communication topology is not a tree)
Definition

$P$ is hierarchical

\[ \text{iff} \]

\[ \exists \mathcal{T} \text{ finite} . \forall Q. P \rightarrow^* Q \implies Q \text{ is } \mathcal{T}-\text{shaped} \]

(Hierarchical = $\mathcal{T}$-shapedness is invariant)
There are terms for which $\mathcal{T}$-shapedness is not an invariant, for any finite $\mathcal{T}$: if the term is not depth-bounded, one can reach forests of unbounded height.
Proposition

Hierarchical $\implies$ Depth-bounded
Proposition

Hierarchical $\implies$ Depth-bounded

Problem: Being hierarchical is still an undecidable property.
Proposition

Hierarchical $\implies$ Depth-bounded

**Problem:** Being hierarchical is still an **undecidable** property.

**Solution:** But now we have more structure, which we exploit to design a type system such that

$$\text{If } \Gamma \vdash_{\mathcal{T}} P \text{ then }$$

$P$ is $\mathcal{T}$-shaped and $P \rightarrow Q \implies Q$ is $\mathcal{T}$-shaped
Standard reductions

\[ S = \overline{a} \langle b \rangle . S' \quad R = a(x). R' \]

\[ \nu a.((\nu b. S) \parallel R) \equiv \nu a.\nu b.(S \parallel R) \rightarrow \nu a.\nu b.(S' \parallel R'[b/x]) \]
$T$-shaped reductions

\[
S = \overline{a}\langle b \rangle.S' \quad R = a(x).(R'_{\text{mig}} \parallel R'_{\neg \text{mig}})
\]

\[
\nu a.((\nu b.S) \parallel R) \quad \rightarrow \quad \nu a.((\nu b.(S' \parallel R'_{\text{mig}}[b/x])) \parallel R'_{\neg \text{mig}})
\]
\( \mathcal{T} \)-shaped reductions are special

\[
\overline{a}\langle b \rangle \quad \rightarrow \quad a(x).P
\]
\[\begin{align*}
\text{IN} & \quad a : t_a[\tau_x] \in \Gamma \quad \Gamma, x : \tau_x \vdash \nu X. \prod_{i \in I} A_i \\
& \quad \text{base}(\tau_x) <_{\tau} t_a \lor (\forall i \in I. \text{Mig}_a(x).P(i) \implies \text{base}(\Gamma(\text{fn}(A_i) \setminus \{a\})) <_{\tau} t_a) \\
& \quad \Gamma \vdash_{\tau} a(x).\nu X. \prod_{i \in I} A_i
\end{align*}\]

\[\begin{align*}
\text{PAR} & \quad \forall i \in I. \forall x : \tau_x \in X. x <_P i \implies \text{base}(\Gamma(\text{fn}(A_i))) <_{\tau} \text{base}(\tau_x) \\
& \quad \Gamma \vdash_{\tau} \nu X. \prod_{i \in I} A_i
\end{align*}\]

\[\begin{align*}
\text{CHOICE} & \quad \forall i \in I. \Gamma \vdash_{\tau} \pi_i.P_i \\
& \quad \Gamma \vdash_{\tau} \sum_{i \in I} \pi_i.P_i\]

\[\begin{align*}
\text{REPL} & \quad \Gamma \vdash_{\tau} A \\
& \quad \Gamma \vdash_{\tau} \pi_i.P_i \implies \Gamma \vdash_{\tau} \pi_i.P_i \\
& \quad \Gamma \vdash_{\tau} \sum_{i \in I} \nu X. \prod_{i \in I} A_i \\
& \quad \Gamma \vdash_{\tau} \nu X. \prod_{i \in I} A_i
\end{align*}\]

\[\begin{align*}
\text{TAU} & \quad \Gamma \vdash_{\tau} P \\
& \quad \Gamma \vdash_{\tau} \text{tau}.P \\
& \quad \Gamma \vdash_{\tau} \text{tau}.P
\end{align*}\]

\[\begin{align*}
\text{OUT} & \quad a : t_a[\tau_b] \in \Gamma \quad \Gamma \vdash_{\tau} Q \\
& \quad \Gamma \vdash_{\tau} \overline{a}(b).Q
\end{align*}\]
Subject reduction

If $\Gamma \vdash \tau P$ and $P \rightarrow Q$, then $\Gamma \vdash \tau Q$
Subject reduction

If $\Gamma \vdash_T P$ and $P \rightarrow Q$, then $\Gamma \vdash_T Q$

Theorem

If $\Gamma \vdash_T P$ and $P$ is $T$-shaped $\implies P$ is hierarchical
Subject reduction

If $\Gamma \vdash_{\mathcal{T}} P$ and $P \rightarrow Q$, then $\Gamma \vdash_{\mathcal{T}} Q$

Theorem

If $\Gamma \vdash_{\mathcal{T}} P$ and $P$ is $\mathcal{T}$-shaped $\implies P$ is hierarchical

When $\Gamma \vdash_{\mathcal{T}} P$ and $P$ is $\mathcal{T}$-shaped, we say $P$ is typably hierarchical.
The type system:

- **type checking**: decidable in \( P \)
- **type inference**: decidable in \( NP \)
- first type system inferring topological properties

Implementation available at
[github.com/bordaigorl/jamesbound](https://github.com/bordaigorl/jamesbound)
Expressivity

$\pi$-calculus
Expressivity

Decidable coverability

\( \pi \)-calculus

Depth-Bounded
Expressivity

- Decidable coverability
- Undecidable membership

π-calculus

Depth-Bounded

Hierarchical
Expressivity

Decidable coverability

Undecidable membership

π-calculus

Depth-Bounded

Hierarchical

Terminating
Expressivity

Decidable coverability

Decidable membership

Undecidable membership

π-calculus

Depth-Bounded
Hierarchical
Typably Hier.
Terminating
Expressivity

- Decidable coverability
- Decidable membership
- Undecidable reachability

π-calculus

Depth-Bounded
Hierarchical
Typably Hier.

PN
Terminating

Undecidable membership
Expressivity

- Decidable coverability
- Decidable membership
- Undecidable reachability
- Undecidable membership

π-calculus

Depth-Bounded
Hierarchical
Typably Hier.
CCS¹
PN
Terminating
Contributions:

- Definition of *hierarchical system*
- A *type system* for hierarchical systems
- Hierarchical systems are expressive but have *decidable coverability & termination*

Future work:

- use typing failures to do smart abstractions
- make the type system more precise
- applications to
  - protocol verification
  - concurrent heap manipulating programs verification
Thank you!

@bordaigorl
emanueledosualdo.com
Appendix
Verification of Depth Bounded systems

Coverability

Init
Verification of Depth Bounded systems

Coverability

Init

Query

$\exists \leq$
Verification of Depth Bounded systems

Coverability

Init

Query
Verification of Depth Bounded systems

**Coverability**  Decidable for depth bounded systems via WSTS

\[ \exists \sum_{c, b} \land wqo \]
**π-calculus**

Syntax:

\[ \mathcal{P} \ni P, Q ::= 0 \mid \nu x. P \mid P_1 \parallel P_2 \mid M \mid !M \]

process

\[ M ::= M + M \mid \pi. P \]

choice

\[ \pi ::= a(x) \mid \overline{a}(b) \mid \tau \]

prefix

Normal form:

\[ \mathcal{P}_{nf} \ni N ::= \nu x_1. \cdots \nu x_n.(A_1 \parallel \cdots \parallel A_m) \]

\[ A ::= \pi_1.N_1 + \cdots + \pi_n.N_n \mid !(\pi_1.N_1 + \cdots + \pi_n.N_n) \]
The nesting of restrictions of a term is given by the function

\[
\text{nest}_\nu(M) := \text{nest}_\nu(!M) := \text{nest}_\nu(0) := 0
\]
\[
\text{nest}_\nu(\nu x.P) := 1 + \text{nest}_\nu(P)
\]
\[
\text{nest}_\nu(P \parallel Q) := \max(\text{nest}_\nu(P), \text{nest}_\nu(Q)).
\]

The depth of a term is defined as the minimal nesting of restrictions in its congruence class:

\[
\text{depth}(P) := \min \{ \text{nest}_\nu(Q) \mid P \equiv Q \}
\]

A term \( P \) is depth-bounded if there exists \( k \in \mathbb{N} \) such that for each \( Q \in \text{Reach}(P) \), \( \text{depth}(Q) \leq k \).
Soundness argument

\[ S = \overline{a}\langle b \rangle . S' \]
\[ R = a(x). (R'_{\text{mig}} \parallel R'_{\lnot\text{mig}}) \]

Diagram:
- Node \( n_a \)
- Node \( n_b \)
- Node \( n_S \)
- Node \( n_R \)
\[ S = \overline{a\langle b \rangle}.S' \]

\[ R = a(x).(R'_{\text{mig}} \parallel R'_{\neg \text{mig}}) \]

- \( \oplus \in S' \)
- \( \triangledown \in R'_{\text{mig}} \)
- \( \blacktriangle \in R'_{\neg \text{mig}} \)
Soundness argument

\[ S = \bar{a}\langle b \rangle.S' \quad R = a(x).( R'_{\text{mig}} \parallel R'_{\neg\text{mig}} ) \]

\[ n_a \quad n_b \quad pS \quad pR \]

\[ \in S' \quad \in R'_{\text{mig}} \quad \in R'_{\neg\text{mig}} \]
\[ S = \overline{a(b)}.S' \quad R = a(x). ( R'_{\text{mig}} \parallel R'_{\neg \text{mig}} ) \]