Spatial and Behavioural types: safety, liveness and decidability

Lucia Acciai and Michele Boreale

Dipartimento di Sistemi e Informatica
Università degli Studi di Firenze

Lisbon, April 19–21, 2011

Outline

1. Introduction
2. Processes, types and formulae
3. The local and the global systems
4. Decidability
5. Conclusion

Outline

Logics and Types

Need to control the usage of (new) names in pi-calculus

Spatial Logic: suitable to

- analyze properties of systems
- describe the spatial structure of processes
- reason on distribution and concurrency

Behavioral types: combines static analisys and model checking

- abstract (the behavior of) processes
- simplify the analysis of concurrent message-passing processes
- properties are checked against types
- E.g. in [Igarashi,Kobayashi'01]
  - processes = pi-calculus, types = CCS
  - (global) invariant safety properties are considered

Our approach

Introduce a type system where

- processes and types share the same “shallow” spatial structure
- each block of declared names is annotated with a SL formula
- type safety: restricted processes are guaranteed to satisfy precise properties on bound names

Benefits

- properties not limited to safety invariants
- compositionality: only relevant names are considered when checking properties

Our approach

Introduce a type system where

- processes and types share the same “shallow” spatial structure
- each block of declared names is annotated with a SL formula
- type safety: restricted processes are guaranteed to satisfy precise properties on bound names

Benefits

- properties not limited to safety invariants
- compositionality: only relevant names are considered when checking properties

Outline

Processes

Pi-calculus with replicated input and guarded summation:

Prefixes	α	$::=$	$a(\tilde{b})$	Input
		$\vert$	$\overline{a}\langle\tilde{b}\rangle$	Output
		$\vert$	τ	Silent prefix
Processes	P	$::=$	$\sum_{i\in I}\alpha_i.P_i$	Guarded summation
		$\vert$	$P \vert P$	Parallel composition
		$\vert$	$(\nu\tilde{b})P$	Restriction
		$\vert$	$!a(\tilde{b}).P$	Replicated input

Types

CCS with replicated input and guarded summation:

Prefixes $\mu ::= a \mid \overline{a} \mid \tau$

Process types	$\mathsf{T} ::= \sum_i \mu_i.\mathsf{T}_i$	Guarded summation
	$\mid \; \mathsf{T} \mid \mathsf{T}$	Parallel composition
	$\mid \; (\nu \tilde{a})\mathsf{T}$	Restriction
	$\mid \; !a.\mathsf{T}$	Replicated input

Channel types $\mathsf{t} ::= (\tilde{x} : \tilde{\mathsf{t}})\mathsf{T}$

Shallow Logic (SL): examples of formulae

shallow = input and output barbs are not followed by a continuation

Race freedom:

$$NoRace(a) \triangleq \Box^* \quad \neg \mathrm{H}^*(\overline{a}|\overline{a})$$

Unique receptiveness:

$$UniRec(a) \triangleq \Box^* \quad \big(a \wedge \neg \mathrm{H}^*(a|a)\big)$$

Responsiveness:

$$Resp(a) \triangleq \Box^*_{-a} \; \Diamond^* \langle a \rangle$$

Deadlock freedom:

$$DeadFree(a) \triangleq \Box^* \Big[\; \big(\overline{a} \rightarrow \mathrm{H}^*(\overline{a}|\Diamond^* \, a)\big) \wedge \big(a \rightarrow \mathrm{H}^*(a|\Diamond^* \, \overline{a})\big) \Big]$$

Well-annotated processes

$$P ::= \cdots \;\Big|\; (\nu \tilde{a} : \tilde{t}; \phi)P \quad \text{with} \quad \mathrm{fn}(\phi) \subseteq \tilde{a}$$

with ϕ a shallow logic formula

Definition (well-annotated processes)

A process $P \in \mathcal{P}$ is *well-annotated* if whenever $P \equiv (\tilde{\nu}\tilde{b})(\nu \tilde{a} : \phi)Q$ then $Q \models \phi$.

Remark: a "weakening" property of SL

Lemma

In Shallow Logic $\forall B$ *with* $\mathrm{fn}(B) = \emptyset$: $\quad A \models \phi \Leftrightarrow A|B \models \phi$

Necessary for soundness of scope extrusion

$$(\nu\tilde{a} : \phi)P \,|\, Q \equiv (\nu\tilde{a} : \phi)(P \,|\, Q) \;\text{ if }\; \tilde{a} \notin Q$$

In (Caires and Cardelli's) Spatial Logic this does not hold. E.g.

- $\neg(\neg\mathbf{0} \,|\, \neg\mathbf{0})$
- $\Diamond\mathbf{T}$

Remark: a "weakening" property of SL

Lemma

In Shallow Logic $\forall B$ *with* $\mathrm{fn}(B) = \emptyset$*:* $A \models \phi \Leftrightarrow A|B \models \phi$

Necessary for soundness of scope extrusion

$$(\nu \tilde{a} : \phi)P \mid Q \equiv (\nu \tilde{a} : \phi)(P \mid Q) \ \text{ if } \ \tilde{a} \notin Q$$

In (Caires and Cardelli's) Spatial Logic this does not hold. E.g.

- $\neg(\neg \mathbf{0} \mid \neg \mathbf{0})$
- $\Diamond \mathbf{T}$

Outline

A "Local" Type System

Judgments: $\Gamma \vdash_{\mathrm{L}} P : \mathrm{T}$

Key rule: (T-RES): $$\frac{\Gamma, \tilde{a} : \tilde{\mathrm{t}} \vdash P : \mathrm{T} \quad \mathrm{T} \downarrow_{\tilde{a}} \models \phi}{\Gamma \vdash (\nu \tilde{a} : \tilde{\mathrm{t}}; \phi) P : (\nu \tilde{a} : \tilde{\mathrm{t}}) \mathrm{T}}$$

Local: in (T-RES) only the part of T depending on the restricted names, $\mathrm{T} \downarrow_{\tilde{x}}$, is taken into account - the rest is hidden

Example: $\left(a.\overline{b}.\overline{a} \,|\, (\nu c)(b.c \,|\, \overline{d} \,|\, \overline{c})\right) \downarrow_a = a.\tau.\overline{a} \,|\, (\nu c)(\tau.c \,|\, \tau \,|\, \overline{c})$

A "Local" Type System

Judgments: $\Gamma \vdash_{\mathrm{L}} P : \mathrm{T}$

Key rule: (T-RES): $\dfrac{\Gamma, \tilde{a} : \tilde{\mathrm{t}} \vdash P : \mathrm{T} \quad \mathrm{T} \downarrow_{\tilde{a}} \models \phi}{\Gamma \vdash (\nu \tilde{a} : \tilde{\mathrm{t}}; \phi) P : (\nu \tilde{a} : \tilde{\mathrm{t}}) \mathrm{T}}$

Local: in (T-RES) only the part of T depending on the restricted names, $\mathrm{T} \downarrow_{\tilde{x}}$, is taken into account - the rest is hidden

Example: $\big(a.\overline{b}.\overline{a} \,|\, (\nu c)(b.c \,|\, \overline{d} \,|\, \overline{c})\big) \downarrow_a = \; a.\tau.\overline{a} \,|\, (\nu c)(\tau.c \,|\, \tau \,|\, \overline{c})$

A "Local" Type System

Judgments: $\Gamma \vdash_{\mathrm{L}} P : \mathsf{T}$

Key rule: (T-RES): $$\frac{\Gamma, \tilde{a} : \tilde{\mathrm{t}} \vdash P : \mathsf{T} \quad \mathsf{T}\downarrow_{\tilde{a}} \models \phi}{\Gamma \vdash (\nu \tilde{a} : \tilde{\mathrm{t}}; \phi) P : (\nu \tilde{a} : \tilde{\mathrm{t}})\mathsf{T}}$$

Local: in (T-RES) only the part of T depending on the restricted names, $\mathsf{T}\downarrow_{\tilde{x}}$, is taken into account - the rest is hidden

Example: $\left(a.\overline{b}.\overline{a} \,|\, (\nu c)(b.c \,|\, \overline{d} \,|\, \overline{c})\right) \downarrow_a = \; a.\tau.\overline{a} \,|\, (\nu c)(\tau.c \,|\, \tau \,|\, \overline{c})$

relevant names = newly created names

Definitions and Results

Definition (negative formulae)

In a negative formula each $\langle -\tilde{x} \rangle^*$ is under an **odd** number of $\neg$

Note: no limitations on other modalities!

Theorem (run-time soundness)

Suppose that $\Gamma \vdash_{\mathrm{L}} P : \mathrm{T}$ *and that* P *is decorated with negative formulae of the form* $\Box^* \phi$. *Then* $P \rightarrow^* P'$ *implies that* P' *is well-annotated.*

Race Freedom and Unique Receptiveness ***are*** *negative*

Definitions and Results

Definition (negative formulae)

In a negative formula each $\langle -\tilde{x} \rangle^*$ is under an **odd** number of $\neg$

Note: no limitations on other modalities!

Theorem (run-time soundness)

Suppose that $\Gamma \vdash_{\mathrm{L}} P : \mathrm{T}$ *and that* P *is decorated with negative formulae of the form* $\Box^* \phi$. *Then* $P \rightarrow^* P'$ *implies that* P' *is well-annotated.*

Race Freedom and Unique Receptiveness ***are*** *negative*

A “Global” Type System: motivations

Type soundness does not hold for non-negative formulae like *Resp*(a) and *DeadFree*(a)

E.g.:

$$R = (\nu a; \mathit{Resp}(a))(c.a|\overline{a})$$

is well-typed for suitable Γ. Indeed

$$\Gamma, a \vdash_{\mathrm{L}} c.a|\overline{a} : c.a|\overline{a}$$

and

$$(c.a|\overline{a}) \downarrow_a = \tau.a|\overline{a} \models \mathit{Resp}(a)$$

but

$$c.a|\overline{a} \not\models \mathit{Resp}(a)$$

Problem: *Resp* on *a* also **depends** on a “global” name *c*

A “Global” Type System: motivations

Type soundness does not hold for non-negative formulae like $\mathit{Resp}(a)$ and $\mathit{DeadFree}(a)$

E.g.:

$$R = (\nu a; \mathit{Resp}(a))(c.a|\overline{a})$$

is well-typed for suitable Γ. Indeed

$$\Gamma, a \vdash_{\mathrm{L}} c.a|\overline{a} : c.a|\overline{a}$$

and

$$(c.a|\overline{a}) \downarrow_a = \tau.a|\overline{a} \models \mathit{Resp}(a)$$

but

$$c.a|\overline{a} \not\models \mathit{Resp}(a)$$

Problem: *Resp* on *a* also **depends** on a “global” name *c*

A "Global" Type System: motivations

Type soundness does not hold for non-negative formulae like $\mathit{Resp}(a)$ and $\mathit{DeadFree}(a)$

E.g.:

$$R = (\nu a; \mathit{Resp}(a))(c.a|\overline{a})$$

is well-typed for suitable Γ. Indeed

$$\Gamma, a \vdash_{\mathrm{L}} c.a|\overline{a} : c.a|\overline{a}$$

and

$$(c.a|\overline{a}) \downarrow_a = \tau.a|\overline{a} \models \mathit{Resp}(a)$$

but

$$c.a|\overline{a} \not\models \mathit{Resp}(a)$$

Problem: *Resp* on *a* also **depends** on a "global" name *c*

A "Global" Type System

Main change:

$\downarrow_{\tilde{x}}$ **replaced by** $\Downarrow_{\tilde{x}}$

where T $\Downarrow_{\tilde{x}}$ keeps the names in $\tilde{x}$ **and the causes of** $\tilde{x}$ in T

(plus some bookkeeping on names)

E.g.:

$$(c.a|\overline{a}) \Downarrow_a = c.a|\overline{a} \not\models Resp(a)$$

relevant names = new names + causally related free names

A “Global” Type System

Main change:

$\downarrow_{\tilde{x}}$ **replaced by** $\Downarrow_{\tilde{x}}$

where T $\Downarrow_{\tilde{x}}$ keeps the names in $\tilde{x}$ **and the causes of** $\tilde{x}$ in T

(plus some bookkeeping on names)

E.g.:

$$(c.a|\overline{a})\Downarrow_a = c.a|\overline{a} \not\models Resp(a)$$

relevant names = new names + causally related free names

A "Global" Type System

Main change:

$\downarrow_{\tilde{x}}$ **replaced by** $\Downarrow_{\tilde{x}}$

where T $\Downarrow_{\tilde{x}}$ keeps the names in $\tilde{x}$ **and the causes of** $\tilde{x}$ in T

(plus some bookkeeping on names)

E.g.:

$$(c.a|\overline{a})\Downarrow_a = c.a|\overline{a} \not\models Resp(a)$$

relevant names = new names + causally related free names

Definitions and Results

Consider ϕ of the form

1. either $\Box^*\psi$ with negation not occurring underneath any $\langle -\tilde{y}\rangle^*$ in ψ
2. or $\Box^*_{-\tilde{y}}\Diamond^*\psi'$, with negation not occurring in ψ'.

Theorem (run-time soundness)

Suppose that $\Gamma \vdash_{\mathrm{G}} P : \mathrm{T}$ and that P is decorated with formulae of the form (1) or (2) above. Then $P \rightarrow^* P'$ implies that P' is well-annotated.

*Responsiveness and Deadlock Freedom **are** of the form (2) and (1) respectively*

Definitions and Results

Consider ϕ of the form

1. either $\Box^* \psi$ with negation not occurring underneath any $\langle -\tilde{y} \rangle^*$ in ψ
2. or $\Box^*_{-\tilde{y}} \Diamond^* \psi'$, with negation not occurring in ψ'.

Theorem (run-time soundness)

Suppose that $\Gamma \vdash_{\mathrm{G}} P : \mathrm{T}$ and that P is decorated with formulae of the form (1) or (2) above. Then $P \rightarrow^* P'$ implies that P' is well-annotated.

*Responsiveness and Deadlock Freedom **are** of the form (2) and (1) respectively*

Outline

Decidability of the type system

The type system is decidable provided that:

1. $\equiv$ is decidable
2. $\models$ is decidable

Decidability of the type system

The type system is decidable provided that:

1. $\equiv$ is decidable
2. $\models$ is decidable

1) $\equiv$ is decidable

From [Engelfriet & Gelsema 2004]

Decidability of the type system

The type system is decidable provided that:

1. $\equiv$ is decidable
2. $\models$ is decidable

1) $\equiv$ is decidable

From [Engelfriet & Gelsema 2004]

2) $\models$ is decidable (?)

The idea is to extend the approach in [BGZ04] for the decidability of weak barbs on CCS to handle SL

WSTS techniques for deciding "$\models$"

Given a (decidable) preorder $\leq$ on types in $\mathcal{T}$

Theorem ([Finkel and Schnoebelen'01])

*Under **certain conditions** for each $I \subseteq \mathcal{T}$ it is possible to compute a **finite** X such that*

$$\uparrow X = \mathit{Pred}^*(I) \qquad \textit{(\textbf{finite basis} of } \mathit{Pred}^*(I)\textit{)}$$

Since $[\![\Diamond^*\phi]\!] = \mathit{Pred}^*([\![\phi]\!])$, to check $T \models \Diamond^*\phi$

1. set $I = [\![\phi]\!]$ above
2. check if $\exists S \in X$ s.t. $S \leq T$

$$\mathit{Pred}(s) = \{s' \mid s' \to s\} \qquad \mathit{Pred}^*(s) = \{s' \mid s' \to^* s\}$$

Conditions [Finkel and Schnoebelen'01]

1. $\mathcal{T}$ forms a **WSTS** w.r.t. (a decidable) $\leq$
2. $\forall T \in \mathcal{T}$ it is possible to compute a **finite** Y s.t.

$$\uparrow Y = \uparrow Pred(\uparrow T) \qquad \text{(effective pred-basis)}$$

3. $\forall I (= [\![\phi]\!])$ it is possible to compute a **finite** Z s.t.

$$\uparrow Z = I (= [\![\phi]\!]) \qquad \text{(finite basis)}$$

Our task:

Find a preorder satisfying the three conditions above

Our approach:

Viewing types as forests and defining a preorder similar to Kruskal's tree-preorder

Conditions [Finkel and Schnoebelen'01]

1. $\mathcal{T}$ forms a **WSTS** w.r.t. (a decidable) $\leq$
2. $\forall T \in \mathcal{T}$ it is possible to compute a **finite** Y s.t.

$$\uparrow Y = \uparrow Pred(\uparrow T) \qquad \text{(\textbf{effective pred-basis})}$$

3. $\forall I (= [\![\phi]\!])$ it is possible to compute a **finite** Z s.t.

$$\uparrow Z = I (= [\![\phi]\!]) \qquad \text{(\textbf{finite basis})}$$

Our task:

Find a preorder satisfying the three conditions above

Our approach:

Viewing types as forests and defining a preorder similar to Kruskal's tree-preorder

Preliminary definition

Fix an initial type T_0

Definition ($\mathcal{F}$)

$\mathcal{F} \triangleq$ the set of all terms:

- containing only subterms and restrictions of T_0
- having nesting depth smaller than T_0's

E.g. $\mathrm{T}_0 = (\nu a)\big(a.b|\overline{a}.\overline{b}\big)$: $\begin{cases} (\nu a)\big(a.b\,|\,\overline{b}\,|\,a.b\big) & \in \mathcal{F} \\ (\nu a)(\nu a)(a.b) & \notin \mathcal{F} \end{cases}$

WSTS I: types as forests

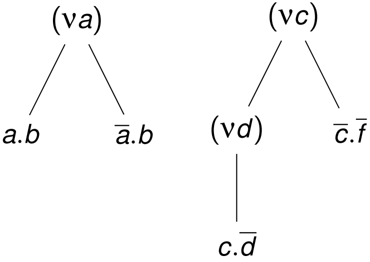
1. Make types a WSTS

We consider types as forests where:

internal nodes = restrictions

leaves = prefix-guarded terms

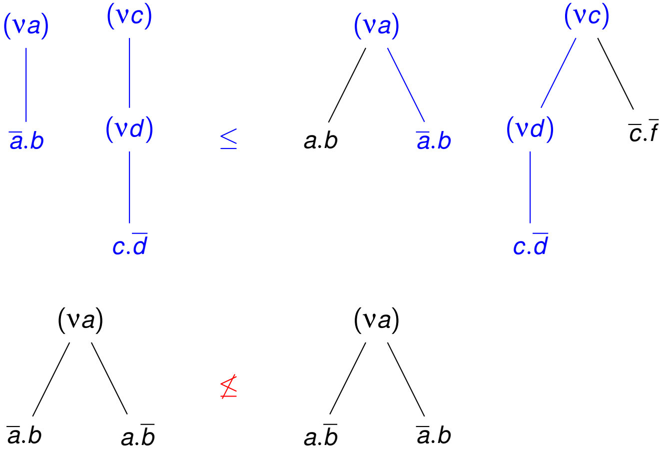
E.g. $T = (\nu a)\big(a.b|\overline{a}.b\big) \,|\, (\nu c)\big((\nu d)c.\overline{d} \,|\, \overline{c}.\overline{f}\big)$



WSTS II: decidable $\leq$

1. Make types a WSTS

Defining the preorder $\leq$ = **rooted tree embedding**



WSTS III: $\langle \mathcal{F}, \rightarrow, \leq \rangle$ is a WSTS

Theorem

(i) $\leq$ is a well-quasi order over $\mathcal{F}$ and (ii) $\langle \mathcal{F}, \rightarrow, \leq \rangle$ is a WSTS

Proof idea: (i) by induction on the nesting depth of restrictions of terms in $\mathcal{F}$ and by using the Higman's lemma. The base case (height $= 0$) relies on finiteness of guarded subterms in T_0. The inductive step relies on the fact that each forest can be decomposed into a finite number of subforests with smaller height

(ii) $\langle \mathcal{F}, \rightarrow, \leq \rangle$ is a finitely branching transition system and $\leq$ is easily proved to be a computable simulation relation in $\mathcal{F}$

WSTS III: $\langle \mathcal{F}, \rightarrow, \leq \rangle$ is a WSTS

Theorem

(i) $\leq$ is a well-quasi order over $\mathcal{F}$ and (ii) $\langle \mathcal{F}, \rightarrow, \leq \rangle$ is a WSTS

Proof idea: (i) by induction on the nesting depth of restrictions of terms in $\mathcal{F}$ and by using the Higman's lemma. The base case (height $= 0$) relies on finiteness of guarded subterms in T_0. The inductive step relies on the fact that each forest can be decomposed into a finite number of subforests with smaller height

(ii) $\langle \mathcal{F}, \rightarrow, \leq \rangle$ is a finitely branching transition system and $\leq$ is easily proved to be a computable simulation relation in $\mathcal{F}$

TRUST ME

WSTS III: $\langle \mathcal{F}, \rightarrow, \leq \rangle$ is a WSTS

Theorem

(i) $\leq$ is a well-quasi order over $\mathcal{F}$ and (ii) $\langle \mathcal{F}, \rightarrow, \leq \rangle$ is a WSTS

Proof idea: (i) by induction on the nesting depth of restrictions of terms in $\mathcal{F}$ and by using the Higman's lemma. The base case (height $= 0$) relies on finiteness of guarded subterms in T_0. The inductive step relies on the fact that each forest can be decomposed into a finite number of subforests with smaller height

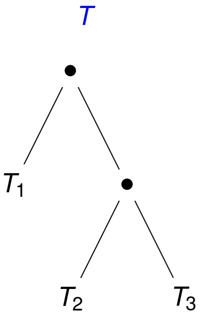
(ii) $\langle \mathcal{F}, \rightarrow, \leq \rangle$ is a finitely branching transition system and $\leq$ is easily proved to be a computable simulation relation in $\mathcal{F}$

TRUST ME

NB: in CCS reductions cannot increase the nesting depth, on the contrary in pi-calculus $\quad (\nu b)\overline{a}\langle b\rangle \mid (\nu c)a(x).\overline{x}.c \;\rightarrow\; (\nu b)(\nu c)\overline{b}.c$

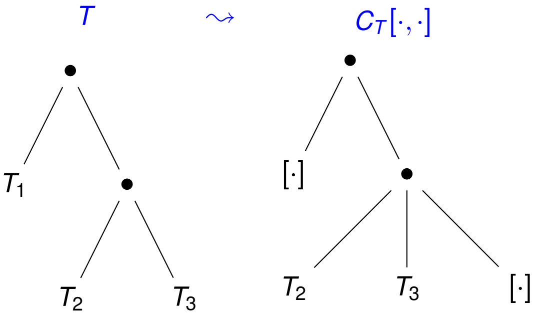
Effective Pred-basis: pb(T)

❷ $\forall T \in \mathcal{T}$ it is possible to compute a **finite** Y s.t. $\uparrow Y = \uparrow Pred(\uparrow T)$



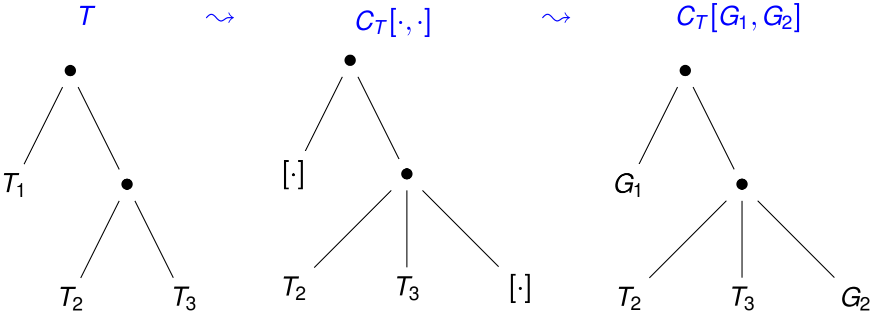
Effective Pred-basis: pb(T)

❷ $\forall T \in \mathcal{T}$ it is possible to compute a **finite** Y s.t. $\uparrow Y = \uparrow Pred(\uparrow T)$



Effective Pred-basis: pb(T)

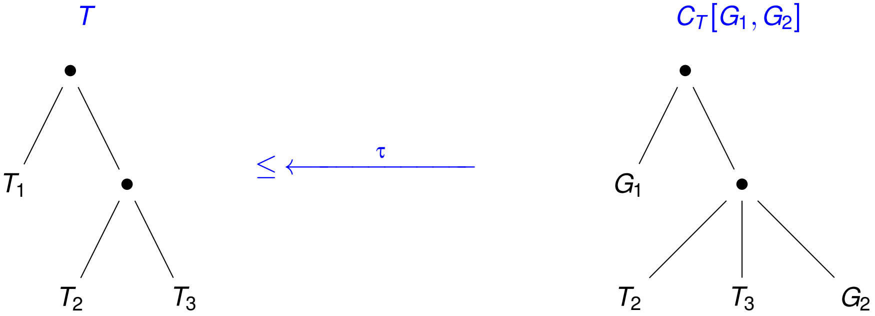
❷ $\forall T \in \mathcal{T}$ it is possible to compute a **finite** Y s.t. $\uparrow Y = \uparrow Pred(\uparrow T)$



G_1, G_2 = prefix-guarded processes (leaves)

Effective Pred-basis: pb(T)

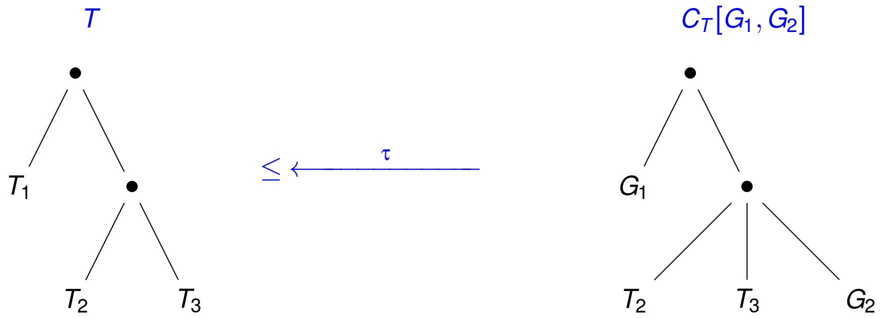
❷ $\forall T \in \mathcal{T}$ it is possible to compute a **finite** Y s.t. $\uparrow Y = \uparrow Pred(\uparrow T)$



G_1, G_2 = prefix-guarded processes (leaves)

Effective Pred-basis: pb(T)

❷ $\forall T \in \mathcal{T}$ it is possible to compute a **finite** Y s.t. $\uparrow Y = \uparrow Pred(\uparrow T)$



Theorem

$\forall T \in \mathcal{T}$: *pb*(T) *is effective and* $\uparrow pb(T) = \uparrow Pred(\uparrow T)$

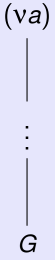
Finite-basis: $\uparrow fb(\phi) = [\![\phi]\!] \cap \mathcal{F}$

3. $\forall I (= [\![\phi]\!])$ it is possible to compute a **finite** Z s.t. $\uparrow Z = I (= [\![\phi]\!])$

(G = prefix-guarded process (leaf) – D = context of parallel and restrictions)

Definition ($fb(\phi)$)

$$fb(a) \stackrel{\triangle}{=} \{ D[G] \in \mathcal{F} \mid G \searrow_a \}$$



Finite-basis: $\uparrow fb(\phi) = [\![\phi]\!] \cap \mathcal{F}$

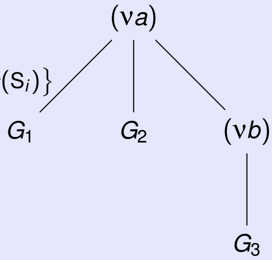
3. $\forall I (= [\![\phi]\!])$ it is possible to compute a **finite** Z s.t. $\uparrow Z = I (= [\![\phi]\!])$

(G = prefix-guarded process (leaf) – D = context of parallel and restrictions)

Definition ($fb(\phi)$)

$$fb(a) \triangleq \{D[G] \in \mathcal{F} \mid G \searrow_a\}$$

$$fb(\mathrm{H}^*(\phi_1|\phi_2)) \triangleq \bigcup_{\mathsf{S}_i \in fb(\phi_i)} \{D[\tilde{G}_1, \tilde{G}_2] \in \mathcal{F} \mid \tilde{G}_i = leaves(\mathsf{S}_i)\}$$



Finite-basis: $\uparrow fb(\phi) = [\![\phi]\!] \cap \mathcal{F}$

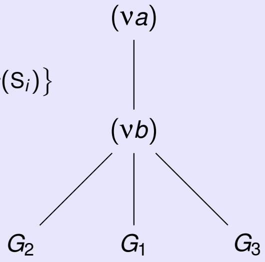
3. $\forall I (= [\![\phi]\!])$ it is possible to compute a **finite** Z s.t. $\uparrow Z = I (= [\![\phi]\!])$

(G = prefix-guarded process (leaf) – D = context of parallel and restrictions)

Definition ($fb(\phi)$)

$$fb(a) \stackrel{\triangle}{=} \{D[G] \in \mathcal{F} \mid G \searrow_a\}$$

$$fb(\mathrm{H}^*(\phi_1|\phi_2)) \stackrel{\triangle}{=} \bigcup_{\mathsf{S}_i \in fb(\phi_i)} \{D[\tilde{G}_1, \tilde{G}_2] \in \mathcal{F} \mid \tilde{G}_i = leaves(\mathsf{S}_i)\}$$



Finite-basis: $\uparrow fb(\phi) = [\![\phi]\!] \cap \mathcal{F}$

3. $\forall I (= [\![\phi]\!])$ it is possible to compute a **finite** Z s.t. $\uparrow Z = I (= [\![\phi]\!])$

(G = prefix-guarded process (leaf) – D = context of parallel and restrictions)

Definition ($fb(\phi)$)

$$fb(a) \triangleq \{D[G] \in \mathcal{F} \mid G \searrow_a\}$$

$$fb(\mathrm{H}^*(\phi_1|\phi_2)) \triangleq \bigcup_{\mathrm{S}_i \in fb(\phi_i)} \{D[\tilde{G}_1, \tilde{G}_2] \in \mathcal{F} \mid \tilde{G}_i = leaves(\mathrm{S}_i)\}$$

$$fb(\phi_1 \vee \phi_2) \triangleq fb(\phi_1) \cup fb(\phi_2)$$

Finite-basis: $\uparrow fb(\phi) = [\![\phi]\!] \cap \mathcal{F}$

3. $\forall I (= [\![\phi]\!])$ it is possible to compute a **finite** Z s.t. $\uparrow Z = I (= [\![\phi]\!])$

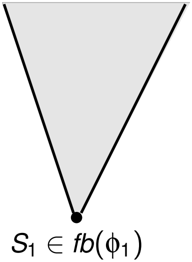
(G = prefix-guarded process (leaf) – D = context of parallel and restrictions)

Definition ($fb(\phi)$)

$$
\begin{aligned}
fb(a) &\stackrel{\triangle}{=} \{D[G] \in \mathcal{F} \mid G \searrow_a\} \\
fb(\mathrm{H}^*(\phi_1|\phi_2)) &\stackrel{\triangle}{=} \bigcup_{\mathrm{S}_i \in fb(\phi_i)} \{D[\tilde{G}_1, \tilde{G}_2] \in \mathcal{F} \mid \tilde{G}_i = leaves(\mathrm{S}_i)\} \\
fb(\phi_1 \vee \phi_2) &\stackrel{\triangle}{=} fb(\phi_1) \cup fb(\phi_2) \\
fb(\Diamond^*\phi) &\stackrel{\triangle}{=} X \quad \text{s.t.} \quad \uparrow X = Pred^*(fb(\phi)) \\
&\dots
\end{aligned}
$$

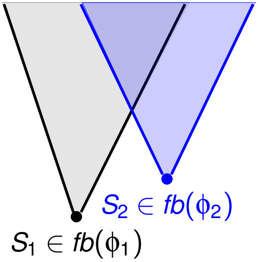
What about fb($\phi_1 \wedge \phi_2$)?

Idea:



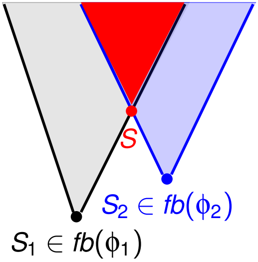
What about fb($\phi_1 \wedge \phi_2$)?

Idea:



What about fb($\phi_1 \wedge \phi_2$)?

Idea:



S = "least common multiple" of S_1 and S_2

E.g. $S_1 = a|b$, $S_2 = b|c \Longrightarrow S = a|b|c$

Main results

Definition (monotone, anti-monotone and plain formulae)

- ϕ is **monotone** if it does not contain occurrences of $\neg$
- **anti-monotone** if it is of the form $\neg\psi$, with ψ monotone
- ϕ is **plain** if it does not contain $\Diamond^*$ underneath H^*

Theorem (decidability on types and processes)

For any ϕ plain and (anti-)monotone

1. *$fb(\phi)$ is a computable finite basis for $[\![\phi]\!] \cap \mathcal{F}$*
2. $\mathsf{T} \models \phi$ *is decidable for any* T
3. *$P \models \phi$ is decidable for any P well-typed*

Examples of decidable formulae

Never two concurrent outputs on a:

$$\mathit{NoRace}(a) \triangleq \neg \Diamond^{*} \mathrm{H}^{*}(\overline{a} \mid \overline{a})$$

Communication on a never occurs more than once:

$$\mathit{Linear}(a) \triangleq \neg \Diamond^{*} \langle a \rangle \Diamond^{*} \langle a \rangle$$

Resource a never acquired in presence of the lock l:

$$\mathit{Lock}(a, l) \triangleq \neg \Diamond^{*} \mathrm{H}^{*}(l \mid \langle a \rangle)$$

Outline

Further and related works

Further:

- Decidability: relax some constraints? Difficult:
  Known result: $\Diamond^*(a \wedge \neg b)$ is undecidable [Zavattaro'09]
- Quantitative behavioural types? Ongoing work

Related:

Behavioural types: Acciai and Boreale'08; Chaki et al.'02; Igarashi and Kobayashi'01;

Decidability results in CCS: Valencia et al.'09; Busi et al.'04

Spatial logics: Caires'04

Undecidability results: Kobayashi and Suto 2007

Type system

$$\text{(T-INP)}\ \frac{\Gamma \vdash a : (\tilde{x} : \tilde{\mathrm{t}})\mathrm{T} \quad \mathrm{fn}(\tilde{\mathrm{t}}) \cup \mathrm{fn}(\mathrm{T}) \setminus \tilde{x} = \mathfrak{a}, \quad \Gamma, \tilde{x} : \tilde{\mathrm{t}} \vdash P : \mathrm{T}|\mathrm{T}' \quad \tilde{x} \notin \mathrm{fn}(\mathrm{T}')}{\Gamma \vdash a(\tilde{x}).P : a^{\mathfrak{a}}.\mathrm{T}'}$$

$$\text{(T-OUT)}\ \frac{\Gamma \vdash a : (\tilde{x} : \tilde{\mathrm{t}})\mathrm{T} \quad \Gamma \vdash \tilde{b} : \tilde{\mathrm{t}} \quad \Gamma vdash P : \mathrm{S}}{\Gamma \vdash \overline{a}\langle \tilde{b} \rangle.P : \overline{a}.(\mathrm{T}[\tilde{b}/\tilde{x}] \,|\, \mathrm{S})}$$

$$\text{(T-RES)}\ \frac{\Gamma, a : \mathrm{t} \vdash P : \mathrm{T} \quad \mathfrak{a} = \mathrm{fn}(\mathrm{t})}{\Gamma \vdash (\nu a : \mathrm{t})P : (\nu a^{\mathfrak{a}})\mathrm{T}} \qquad \text{(T-PAR)}\ \frac{\Gamma \vdash P : \mathrm{T} \quad \Gamma \vdash Q : \mathrm{S}}{\Gamma \vdash P|Q : \mathrm{T}|\mathrm{S}}$$

$$\text{(T-SUM)}\ \frac{|I| \neq 1 \quad \forall i \in I : \ \Gamma \vdash \alpha_i.P_i : \mu_i.\mathrm{T}_i}{\Gamma \vdash \sum_{i\in} \alpha_i.P_i : \sum_{i \in I} \mu_i.\mathrm{T}_i} \qquad \text{(T-REP)}\ \frac{\Gamma \vdash a(\tilde{x}).P : a^{\mathfrak{a}}.\mathrm{T}}{\Gamma \vdash !a(\tilde{x}).P : !a^{\mathfrak{a}}.\mathrm{T}}$$

$$\text{(T-EQ)}\ \frac{\Gamma \vdash P : \mathrm{T} \quad \mathrm{T} \equiv \mathrm{S}}{\Gamma \vdash P : \mathrm{S}} \qquad \text{(T-TAU)}\ \frac{\Gamma \vdash P : \mathrm{T}}{\Gamma \vdash \tau.P : \tau.\mathrm{T}}$$

Example: Unique Receptiveness (a liveness property)

$\Rightarrow$ Local Type System

$$UniRec(a) \stackrel{\triangle}{=} \Box^*\big(a \wedge \neg \mathrm{H}^*(a|a)\big)$$

$$P = (\nu a, b, c\,;\, UniRec(a))Q$$

$$Q = \big(\,(\overline{c}\langle a\rangle \mid a + b(x).x) \mid c(y).\overline{b}\langle y\rangle\big)$$

is well-typed. Indeed, for a suitable Γ:

$$\Gamma, a, b, c \vdash_{\mathrm{L}} Q : \mathrm{T} \stackrel{\triangle}{=} \overline{c}.\overline{b}.a \mid a + b \mid c$$

with

$$\mathrm{T}\downarrow_{a,b,c} = \mathrm{T} \models UniRec(a)$$

hence well-typed by (T-RES)

Example: Unique Receptiveness (a liveness property)

$\Rightarrow$ Local Type System

$$UniRec(a) \triangleq \Box^*\big(a \wedge \neg \mathrm{H}^*(a|a)\big)$$

$$P = (\nu a, b, c\, ; UniRec(a))Q$$

$$Q = \big((\overline{c}\langle a\rangle \mid a + b(x).x) \mid c(y).\overline{b}\langle y\rangle\big)$$

is well-typed. Indeed, for a suitable Γ:

$$\Gamma, a, b, c \vdash_{\mathrm{L}} Q : \mathsf{T} \triangleq \overline{c}.\overline{b}.a \mid a + b \mid c$$

with

$$\mathsf{T}\downarrow_{a,b,c} = \mathsf{T} \models UniRec(a)$$

hence well-typed by (T-RES)

Example: Unique Receptiveness (a liveness property)

⇒ Local Type System

$$UniRec(a) \triangleq \Box^{*}\big(a \wedge \neg \mathrm{H}^{*}(a|a)\big)$$

$$P = (\nu a, b, c\,;\, UniRec(a))Q$$

$$Q = \big(\,(\overline{c}\langle a\rangle \mid a + b(x).x) \mid c(y).\overline{b}\langle y\rangle\big)$$

is well-typed. Indeed, for a suitable Γ:

$$\Gamma, a, b, c \vdash_{\mathrm{L}} Q : \mathsf{T} \triangleq \overline{c}.\overline{b}.a \mid a + b \mid c$$

with

$$\mathsf{T}\downarrow_{a,b,c} = \mathsf{T} \models UniRec(a)$$

hence well-typed by (T-RES)

Example: Responsiveness

$\Rightarrow$ Global Type System

$$Resp(a) \triangleq \Box^{*}_{-a}\Diamond^{*}\langle a\rangle$$

$$P = (\nu a : Resp(a))(\overline{c}\langle a\rangle)|Q$$

$$Q = !c(x).(\overline{x}|x)|\overline{c}\langle b\rangle$$

is well-typed. Indeed, for a suitable Γ:

$$\Gamma \vdash_{\mathrm{G}} \overline{c}\langle a\rangle|Q : \overline{c}.(\overline{a}|a)|!c|\overline{c}.(\overline{b}|b) \triangleq \mathrm{T}$$

and

$$\mathrm{T}\Downarrow_a = \overline{c}.(\overline{a}|a)|!c|\overline{c}.(\tau|\tau) \models Resp(a)$$

hence well-typed by (T-RES)

Example: Responsiveness

$\Rightarrow$ Global Type System

$$Resp(a) \triangleq \Box^*_{-a} \Diamond^* \langle a \rangle$$

$$P = (\nu a : Resp(a))(\overline{c}\langle a \rangle) | Q$$

$$Q = !c(x).(\overline{x}|x) | \overline{c}\langle b \rangle$$

is well-typed. Indeed, for a suitable Γ:

$$\Gamma \vdash_G \overline{c}\langle a \rangle | Q : \overline{c}.(\overline{a}|a) | !c | \overline{c}.(\overline{b}|b) \triangleq T$$

and

$$T \Downarrow_a = \overline{c}.(\overline{a}|a) | !c | \overline{c}.(\tau|\tau) \models Resp(a)$$

hence well-typed by (T-RES)

Example: Responsiveness

$\Rightarrow$ Global Type System

$$Resp(a) \triangleq \Box^*_{-a} \Diamond^* \langle a \rangle$$

$$P = (\nu a : Resp(a))(\overline{c}\langle a \rangle)|Q$$

$$Q = !c(x).(\overline{x}|x)|\overline{c}\langle b \rangle$$

is well-typed. Indeed, for a suitable Γ:

$$\Gamma \vdash_{\mathrm{G}} \overline{c}\langle a \rangle|Q : \overline{c}.(\overline{a}|a)|!c|\overline{c}.(\overline{b}|b) \triangleq \mathsf{T}$$

and

$$\mathsf{T} \Downarrow_a = \overline{c}.(\overline{a}|a)|!c|\overline{c}.(\tau|\tau) \models Resp(a)$$

hence well-typed by (T-RES)

Shallow Logic (SL)

$\phi ::= \mathbf{T}$	$[\![\mathbf{T}]\!] = \mathcal{U}$
$\mid \neg\phi$	$[\![\neg\phi]\!] = \mathcal{U} \setminus [\![\phi]\!]$
$\mid \phi \vee \phi$	$[\![\phi_1 \vee \phi_2]\!] = [\![\phi_1]\!] \cup [\![\phi_2]\!]$
$\mid \phi \wedge \phi$	$[\![\phi_1 \wedge \phi_2]\!] = [\![\phi_1]\!] \cap [\![\phi_2]\!]$
$\mid a$	$[\![a]\!] = \{A \mid A \searrow_a\}$
$\mid \bar{a}$	$[\![\bar{a}]\!] = \{A \mid A \searrow_{\bar{a}}\}$
$\mid \phi \mid \phi$	$[\![\phi_1 \mid \phi_2]\!] = \{A \mid \exists A_1, A_2 : A \equiv A_1 \mid A_2,\ A_1 \in [\![\phi_1]\!],\ A_2 \in [\![\phi_2]\!]\}$
$\mid \mathrm{H}^*\phi$	$[\![\mathrm{H}^*\phi]\!] = \{A \mid \exists \tilde{a}, B : A \equiv (\tilde{\nu}\tilde{a})B,\ \tilde{a}\#\phi,\ B \in [\![\phi]\!]\}$
$\mid \langle a \rangle \phi$	$[\![\langle a \rangle \phi]\!] = \{A \mid \exists B : A \xrightarrow{\langle a \rangle} B,\ B \in [\![\phi]\!]\}$
$\mid \langle \tilde{a} \rangle^* \phi$	$[\![\langle \tilde{a} \rangle^* \phi]\!] = \{A \mid \exists \sigma, B : A \xrightarrow{\sigma} B,\ \mathcal{N} \setminus \tilde{a}\#\sigma,\ B \in [\![\phi]\!]\}$
$\mid \langle -\tilde{a} \rangle^* \phi$	$[\![\langle -\tilde{a} \rangle^* \phi]\!] = \{A \mid \exists \sigma, B : A \xrightarrow{\sigma} B,\ \tilde{a}\#\sigma,\ B \in [\![\phi]\!]\}$

Shallow Logic (SL)

$$
\begin{array}{lll}
\phi ::= \mathbf{T} & & [\![\mathbf{T}]\!] = \mathcal{U} \\
\mid \ \neg\phi & & [\![\neg\phi]\!] = \mathcal{U} \setminus [\![\phi]\!] \\
\mid \ \phi \vee \phi & & [\![\phi_1 \vee \phi_2]\!] = [\![\phi_1]\!] \cup [\![\phi_2]\!] \\
\mid \ \phi \wedge \phi & & [\![\phi_1 \wedge \phi_2]\!] = [\![\phi_1]\!] \cap [\![\phi_2]\!] \\
\mid \ a & & [\![a]\!] = \{A \mid A \searrow_a\} \\
\mid \ \bar{a} & & [\![\bar{a}]\!] = \{A \mid A \searrow_{\bar{a}}\} \\
\mid \ \phi|\phi & & [\![\phi_1|\phi_2]\!] = \{A \mid \exists A_1, A_2 : A \equiv A_1|A_2,\ A_1 \in [\![\phi_1]\!],\ A_2 \in [\![\phi_2]\!]\} \\
\mid \ \mathrm{H}^*\phi & & [\![\mathrm{H}^*\phi]\!] = \{A \mid \exists \tilde{a}, B : A \equiv (\tilde{\nu}\tilde{a})B,\ \tilde{a}\#\phi,\ B \in [\![\phi]\!]\} \\
\mid \ \langle a \rangle\phi & & [\![\langle a \rangle\phi]\!] = \{A \mid \exists B : A \xrightarrow{\langle a \rangle} B,\ B \in [\![\phi]\!]\} \\
\mid \ \langle \tilde{a} \rangle^*\phi & & [\![\langle \tilde{a} \rangle^*\phi]\!] = \{A \mid \exists \sigma, B : A \xrightarrow{\sigma} B,\ \mathcal{N} \setminus \tilde{a}\#\sigma,\ B \in [\![\phi]\!]\} \\
\mid \ \langle -\tilde{a} \rangle^*\phi & & [\![\langle -\tilde{a} \rangle^*\phi]\!] = \{A \mid \exists \sigma, B : A \xrightarrow{\sigma} B,\ \tilde{a}\#\sigma,\ B \in [\![\phi]\!]\}
\end{array}
$$