A Spatial Logic for Concurrency

(CONCUR, Brno, August 2002)

Luís Caires
Departamento de Informática, FCT/UNL, Lisboa

Luca Cardelli
Microsoft Research Cambridge

Spatial Properties

- Distributed Systems
  - Systems where behavior is *spatially* distributed
  - Processes behave in time and move in space (mobility)
  - Space: a structured set of places (multiset, tree, graph ?)
- Spatial Properties
  - *The truth value of a formula depends on its location*
    - location dependent access to resources
  - Spatial properties are not invariant under *bisimulation*
    - we want to observe the internal structure of the system
  - Spatial properties are not invariant under reduction
    - the structure of space may change in time
  - But a spatial property may define a structural invariant
    - E.g., connectivity, unique handling of names
    - Spatial logics always offer a degree of intensionality

Spatial Operators

- Process operators are traditionally seen as mappings from behaviors into behaviors (*cf.*, denotational semantics)
- Some basic operators have a natural spatial meaning
  - E.g., $P \mid Q$, $(\nu n)P$
  - These usually correspond to the **static** operators of process calculi
  - Actors model, Chemical semantics, **structural** congruence
- Spatial Operators
  - Spatial operators assemble systems from subsystems
  - Some "new" operators are deliberately spatial (e.g., $n[P]$, $P \| Q$)
  - Spatial operators may or may not induce proper behavior
- Spatial properties we focus on
  - Decomposition into subsystems (parallel components)
  - Local resources (restricted names)

Spatial Operators

Processes		Formulas	
$\mathbf{0}$	*void*	$\mathbf{0}$	
$P \mid Q$	*composition*	$\mathsf{A} \mid \mathsf{B}$	
$(\nu n)P$	*restriction*	$\mathsf{H}x.\mathsf{A}$	*Hidden name quantification*
		$n \circledR \mathsf{A}$	*Revelation*
		$\mathsf{H}x.\mathsf{A} \triangleq И x.x \circledR \mathsf{A}$	
$n\langle m \rangle$	*message*	$n\langle m \rangle$	

The sound way to refer to a secret name is by using a fresh identity: the secret name cannot clash with any known name.

Process Model: Asynchronous π-Calculus (Aπ)

$n,m,p \in \mathrm{N}$ Names

$P,Q \in \mathrm{P}$ ::= Processes

$(\nu n)P$	*restriction*
$\mathbf{0}$	*void*
$P \mid Q$	*composition*
$!P$	*replication*
$n\langle m\rangle$	*message*
$n(m).P$	*input*

Reduction:

$m\langle n\rangle \mid m(p).P \rightarrow P\{p \leftarrow n\}$

$P \rightarrow Q \Rightarrow (\nu n)P \rightarrow (\nu n)Q$

$P \rightarrow Q \Rightarrow P \mid R \rightarrow Q \mid R$

$P' \equiv P, P \rightarrow Q, Q \equiv Q' \Rightarrow P' \rightarrow Q'$

Structural Congruence:

$P \equiv P$

$P \equiv Q \Rightarrow Q \equiv P$

$P \equiv Q, Q \equiv R \Rightarrow P \equiv R$

$P \equiv Q \Rightarrow (\nu n)P \equiv (\nu n)Q$

$P \equiv Q \Rightarrow P \mid R \equiv Q \mid R$

$P \equiv Q \Rightarrow !P \equiv !Q$

$P \equiv Q \Rightarrow m(n).P \equiv m(n).Q$

$P \mid 0 \equiv P$

$P \mid Q \equiv Q \mid P$

$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$

$(\nu n)P \equiv (\nu m)P\{n \leftarrow m\}$ if $m \notin fn(P)$

$(\nu n)0 \equiv 0$

$(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$

$(\nu n)(P \mid Q) \equiv P \mid (\nu n)Q$ if $n \notin fn(P)$

$(\nu n)(m(p).P) \equiv m(p).(\nu n)P$ if $p \neq m$, $p \neq n$

Formulas

A,B ∈ Φ ::= Formulas

x,y ∈ V Variables; η,μ ∈ N ∪V

X,Y ∈ X Propositional variables

F	*False*
A ∧ B	*Conjunction*
0	*Void*
A | B	*Composition*
A ▷ B	*Guarantee*
η ® A	*Revelation*
A ⊘ η	*Hiding*
η⟨μ⟩	*Message*
◇A	*Next Step*
∀x.A	*Universal Name Quantification*
Иx.A	*Fresh Name Quantification*
∀X.A	*Second-order Universal Quantification*
X	*Propositional Variable*

Some Simple Examples

- Somewhere:
  $\mathsf{A} \mid \mathbf{T}$
- Prime:
  $\mathbf{1} \triangleq \neg(\neg\mathbf{0} \mid \neg\mathbf{0}) \wedge \neg\mathbf{0}$
- Input [*cf.* Sangiorgi]:
  $n(x)\mathsf{A} \triangleq \forall x.\ n\langle x\rangle \triangleright \Diamond A$
- Free Name:
  $©n \triangleq \neg\ n®\mathbf{T}$
- Nonce Generator:
  $(\nu n)pub\langle n\rangle \models \mathsf{H}x.pub\langle x\rangle$
- Somewhere A:
  $☝\mathsf{A} \triangleq \nu X.(\mathsf{A} \mid \mathbf{T}) \wedge \mathsf{H}x.\ X$
- Unique handling:
  $\forall x.\ \neg ☝ (\exists y.\langle\ x(y)\mathbf{T} \mid \exists y\ x(y)\mathbf{T} \mid \mathbf{T})$

Satisfaction and Validity

- The *denotation* of a formula A is a *set of processes* $[\![A]\!]$

P satisfies A	*if and only if*	$P \in [\![A]\!]$

- A *simple* sequent $A \vdash B$ is *valid* if all processes that satisfy A also satisfy B

$A \vdash B$ is valid	*if and only if*	$[\![A]\!] \subseteq [\![B]\!]$

- Some simple valid sequents:
  - $\neg\mathbf{0} \mid \mathbf{T} \vdash \neg\mathbf{0}$
  - $\mathbf{0} \wedge (A \mid B) \vdash A \wedge B$

- Remarks**:**
  - Satisfaction should be invariant under $\equiv$.
  - To interpret $\text{И}x.A$ we need to express a notion of name freshness w.r.t. (possibly infinite) sets of processes.

Property Sets and Freshness

Support. The set of names relevant for any property expressible in our logic is always finite (*cf*. the set of free names of formulas).

A *transposition* τ is a pair $\{n \leftrightarrow m\}$ of names. A transposition $\{n \leftrightarrow m\}$ acts on process P ($\tau \bullet P$) by swaping in P all occurrences of n and m.

Transposition of a Set of Processes. $\tau \bullet \psi \triangleq \{ \tau \bullet P \,\|\, P \in \psi \}$

Support of a Set of Processes. A *support* of a set of processes ψ is a set of names N such that for all $n, m \notin N$ we have $\{n \leftrightarrow m\} \bullet \psi = \psi$.

Pset. A ***Pset*** ($\psi \in \mathbf{P}$) is a finitely supported, $\equiv$ -closed set of processes. Every Pset ψ has a (finite) least support, denoted by ***supp***(ψ).

Our semantics assigns to each formula a Pset

$$[\![\_]\!] : \Phi \rightarrow \mathbf{P}$$

Semantic Freshness. A name n is *fresh* w.r.t. a Pset ψ if $n \notin$ ***supp***(ψ)

Semantics

$$[\![\mathbf{F}]\!]_v \triangleq \emptyset$$

$$[\![\mathsf{A} \wedge \mathsf{B}]\!]_v \triangleq [\![\mathsf{A}]\!]_v \cap [\![\mathsf{B}]\!]_v$$

$$[\![\mathsf{A} \Rightarrow \mathsf{B}]\!]_v \triangleq \{P \mid P \in [\![\mathsf{A}]\!]_v \Rightarrow P \in [\![\mathsf{B}]\!]_v\}$$

$$[\![\mathbf{0}]\!]_v \triangleq \{P \mid P \equiv \mathbf{0} \}$$

$$[\![\mathsf{A} \mid \mathsf{B}]\!]_v \triangleq \{P \mid \exists Q.\ \exists R.\ P \equiv Q \mid R \wedge Q \in [\![\mathsf{A}]\!]_v \wedge R \in [\![\mathsf{B}]\!]_v\}$$

$$[\![\mathsf{A} \triangleright \mathsf{B}]\!]_v \triangleq \{P \mid \forall Q.\ Q \in [\![\mathsf{A}]\!]_v \Rightarrow P \mid Q \in [\![\mathsf{B}]\!]_v\}$$

$$[\![n \circledR \mathsf{A}]\!]_v \triangleq \{P \mid \exists P'.\ P \equiv (\nu n)P' \wedge P' \in [\![\mathsf{A}]\!]_v\}$$

$$[\![\mathsf{A} \oslash n]\!]_v \triangleq \{P \mid (\nu n)P \in [\![\mathsf{A}]\!]_v\}$$

$$[\![n\langle m\rangle]\!]_v \triangleq \{P \mid P \equiv n\langle m\rangle\}$$

$$[\![\Diamond \mathsf{A}]\!]_v \triangleq \{P \mid \exists P'.\ P \rightarrow P' \wedge P' \in [\![\mathsf{A}]\!]_v\}$$

$$[\![\forall x.\mathsf{A}]\!]_v \triangleq \bigcap n \in \mathrm{N}.\ [\![\mathsf{A}\{x \leftarrow n\}]\!]_v$$

$$[\![X]\!]_v \triangleq v(X)$$

$$[\![\forall X.\mathsf{A}]\!]_v \triangleq \bigcap \psi \in \mathbf{P}.\ [\![\mathsf{A}]\!]_{v[X \leftarrow \psi]}$$

$$P \models_v \mathsf{A} \triangleq P \in [\![\mathsf{A}]\!]_v$$

for name-closed A

Freshness and Hiding

- The fresh quantifier $\text{И}x.\mathsf{A}$ is defined such that a process P satisfies $\text{И}x.\mathsf{A}$ if and only if P satisfies $\mathsf{A}\{x\leftarrow n\}$ for some name n fresh in P and in A.

$$P \vDash_v \text{И}x.\mathsf{A} \;\text{ iff }\; \exists n\in\mathsf{N}.\ n\notin fn^v(P,\mathsf{A}) \wedge P \vDash_v \mathsf{A}\{x\leftarrow n\}$$

$$P \vDash_v \text{И}x.\mathsf{A} \;\text{ iff }\; \forall n\in\mathsf{N}.\ n\notin fn^v(P,\mathsf{A}) \Rightarrow P \vDash_v \mathsf{A}\{x\leftarrow n\}$$ [Gabbay-Pitts]

(this means that *a fresh name is as good as any other*)

- The hiding quantifier $\mathsf{H}x.\mathsf{A}$ is defined such that a process P satisfies $\mathsf{H}x.\mathsf{A}$ if and only if $P \equiv (\nu n)Q$ and Q satisfies $\mathsf{A}\{x\leftarrow n\}$ for some name n fresh in A.

$$P \vDash_v \mathsf{H}x.\mathsf{A} \text{ iff } \exists n\in\mathsf{N}.\ n\notin fn^v(\mathsf{A}) \wedge P \equiv (\nu n)Q \wedge Q \vDash_v \mathsf{A}\{x\leftarrow n\}$$

- One can then define $\mathsf{H}x.\mathsf{A} \triangleq \text{И}x.x®\mathsf{A}$:
- A main use for $\mathsf{H}x.\mathsf{A}$: expressing properties of secrets

$\mathsf{H}x.(\ ©x \wedge \mathsf{A})$ $\qquad\qquad$ $\neg\, \mathsf{H}x.\ (\ pub\langle x\rangle \mid \mathbf{T}\ )$

A Simple Protocol

$\text{Client} \triangleq \mathbf{H}x.(\text{Proto}(x) \mid \text{Request}(x))$

$\text{Server} \triangleq \nu Y.\, \text{И}x.\ \text{Proto}(x) \triangleright \Diamond\, (\text{Handler}(x) \mid Y\,)$

$\text{Proto}(x) \triangleq pub\langle x\rangle$

- By unfolding we have:

  $\text{Server} \dashv\vdash \text{И}x.\ \text{Proto}(x) \triangleright \Diamond\, (\text{Handler}(x) \mid \text{Server}\,)$

- We can then show:

  $\text{Server} \mid \text{Client} \vdash \Diamond\, (\text{Server} \mid \mathbf{H}x.(\text{Handler}(x) \mid \text{Request}(x)))$

- Guarantee is granted just for fresh nonces, e.g., we may have

  $\forall x.\ (\text{Server} \wedge \copyright x) \Rightarrow (\text{Proto}(x) \triangleright \Diamond\, \text{Server}\,)$

A Proof System

- We define a (modal) labeled sequent calculus where *labels* denote π-calculus processes and *accessibility* is reduction

$$\langle S \rangle \;\; u_1 : \mathsf{A}_1, \ldots, u_n : \mathsf{A}_n \vdash v_1 : \mathsf{B}_1, \ldots, v_m : \mathsf{B}_m$$

  - A_i, B_j are (nameless) formulas.
  - u_i v_j, labels are *indexes*, elements of
    - The *term π-algebra* $\mathbf{P} = \langle \mathcal{N}, \mathcal{I}, \mathbf{0}, |, \nu, \leftrightarrow_{\mathcal{N}}, \leftrightarrow_{\mathcal{I}} \rangle$ over process variables $\mathcal{X}$, where $\mathcal{N}$ are name terms and $\mathcal{I}$ are process terms
  - S is a finite set of *constraints*, describing the "current world"
  - Constraints:
    - *Equations* $u = v$ between indexes (to handle spatial structure)
    - *Distinctions* $n \# m$ (to handle freshness)
    - *Reductions* $u \rightarrow v$ (to handle dynamics)

A Proof System

- Propositional Rules, e.g.,

(∧ L)

$$\frac{\langle S\rangle\ \Gamma, u : \mathsf{A}, u : \mathsf{B} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \mathsf{A} \wedge \mathsf{B} \vdash \Delta}$$

(∧ R)

$$\frac{\langle S\rangle\ \Gamma \vdash u : \mathsf{A}, \Delta \quad \langle S\rangle\ \Gamma \vdash u : \mathsf{B}, \Delta}{\langle S\rangle\ \Gamma \vdash u : \mathsf{A} \wedge \mathsf{B}, \Delta}$$

- Spatial Rules, e.g.,

(| L) *𝒳, 𝒴 not free in the conclusion*

$$\frac{\langle S, u \doteq \mathcal{X} | \mathcal{Y}\rangle\ \Gamma, \mathcal{X} : \mathsf{A}, \mathcal{Y} : \mathsf{B} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \mathsf{A} \mid \mathsf{B} \vdash \Delta}$$

(| R)

$$\frac{\langle S\rangle\ \Gamma \vdash v : \mathsf{A}, \Delta \quad \langle S\rangle\ \Gamma \vdash t : \mathsf{B}, \Delta \quad u \doteq_S v|t}{\langle S\rangle\ \Gamma \vdash u : \mathsf{A} \mid \mathsf{B}, \Delta}$$

- World Rules, e.g.,

$$\frac{\langle S, u \doteq \mathbf{0}\rangle\ \Gamma \vdash \Delta \quad u|v \doteq_S \mathbf{0}}{\langle S\rangle\ \Gamma \vdash \Delta}$$

- Freshness Rules, e.g.,

(!) *𝒴, x not free in the conclusion*

$$\frac{\langle S, \boldsymbol{x} \# \boldsymbol{N}, u \doteq (\nu \boldsymbol{x})\mathcal{Y}\rangle\ \Gamma \vdash \Delta}{\langle S\rangle\ \Gamma \vdash \Delta}$$

A Simple Proof

(0 R)

$$\frac{u \doteq_S \mathbf{0}}{\langle S \rangle\ \Gamma \vdash u : \mathbf{0}, \Delta}$$

(| R)

$$\frac{\langle S \rangle\ \Gamma \vdash v : \mathsf{A}, \Delta \quad \langle S \rangle\ \Gamma \vdash t : \mathsf{B}, \Delta \quad u \doteq_S v | t}{\langle S \rangle\ \Gamma \vdash u : \mathsf{A} \mid \mathsf{B}, \Delta}$$

(0 L)

$$\frac{\langle S, u \doteq \mathbf{0} \rangle\ \Gamma \vdash \Delta}{\langle S \rangle\ \Gamma, u : \mathbf{0} \vdash \Delta}$$

(| L) *$\mathcal{X}, \mathcal{Y}$ not free in the conclusion*

$$\frac{\langle S, u \doteq \mathcal{X} | \mathcal{Y} \rangle\ \Gamma, \mathcal{X} : \mathsf{A}, \mathcal{Y} : \mathsf{B} \vdash \Delta}{\langle S \rangle\ \Gamma, u : \mathsf{A} \mid \mathsf{B} \vdash \Delta}$$

5	$\langle \mathcal{Z} \doteq \mathcal{X} | \mathcal{Y}, \mathcal{Z} \doteq 0, \mathcal{X} \doteq 0 \rangle\ \mathcal{X} : \mathsf{A}, \mathcal{Y} : \mathsf{B} \vdash \mathcal{Z} : \mathsf{A}$	**(Id) since $z = x$**
4	$\langle \mathcal{Z} \doteq \mathcal{X} | \mathcal{Y}, \mathcal{Z} \doteq 0 \rangle\ \mathcal{X} : \mathsf{A}, \mathcal{Y} : \mathsf{B} \vdash \mathcal{Z} : \mathsf{A}$	**5, (S | 0) since $x \mid y = 0$**
3	$\langle \mathcal{Z} \doteq \mathcal{X} | \mathcal{Y} \rangle\ \mathcal{X} : \mathsf{A}, \mathcal{Y} : \mathsf{B}, \mathcal{Z} : \mathbf{0} \vdash \mathcal{Z} : \mathsf{A}$	**4, (0 L)**
2	$\langle\ \rangle\ \mathcal{Z} : \mathsf{A} \mid \mathsf{B}, \mathcal{Z} : \mathbf{0} \vdash \mathcal{Z} : \mathsf{A}$	**3, (| L)**
1	$\langle\ \rangle\ \mathcal{Z} : (\mathsf{A} \mid \mathsf{B}) \wedge \mathbf{0} \vdash \mathcal{Z} : \mathsf{A}$	**2, ($\wedge$ L)**

An Example with Freshness

4 $\langle\ \mathcal{Z} \doteq (\nu x)\mathcal{X}, \mathcal{X} \doteq (\nu x)\mathcal{Y}, x \# \mathsf{A} \rangle\ \ \mathcal{X} : \mathsf{A}, \mathcal{Y} : \mathbf{T} \vdash \mathcal{Z} : \mathsf{A}$ **(Id) since** $\mathcal{Z} \doteq \mathcal{X} \doteq (\nu x)\mathcal{Y}$

3 $\langle\ \mathcal{Z} \doteq (\nu x)\mathcal{X}, \mathcal{X} \doteq (\nu x)\mathcal{Y}, x \# \mathsf{A} \rangle\ \ \mathcal{X} : \mathsf{A}, \mathcal{Y} : \mathbf{T} \vdash \mathcal{Z} : \text{И}x.\mathsf{A}$ **4, (И R)**

2 $\langle\ \mathcal{Z} \doteq (\nu x)\mathcal{X}, x \# \mathsf{A}\ \rangle\ \ \mathcal{X} : \mathsf{A}, \mathcal{X} : x ® \mathbf{T} \vdash \mathcal{Z} : \text{И}x.\mathsf{A}$ **3, (® L)**

1 $\langle\ \rangle\ \ \mathcal{Z} : \text{И}x.\ x ® (\mathsf{A} \wedge x ® \mathbf{T}) \vdash \mathcal{Z} : \text{И}x.\mathsf{A}$ **2, (∧ L) (И L)**

0 $\langle\ \rangle\ \ \mathcal{Z} : \mathbf{H}x.\ (\mathsf{A} \wedge x ® \mathbf{T}) \vdash \mathcal{Z} : \text{И}x.\mathsf{A}$

(И R)

$$\frac{\langle S \rangle\ \ \Gamma \vdash u : \mathsf{A}\{x \leftarrow n\}, \Delta \quad u \doteq_S (\nu n)t \quad n \#_S \mathsf{A}}{\langle S \rangle\ \ \Gamma \vdash u : \text{И}x.\mathsf{A}, \Delta}$$

(И L)

$$\frac{\langle S \rangle\ \ \Gamma, u : \mathsf{A}\{x \leftarrow n\} \vdash \Delta \quad u \doteq_S (\nu n)t \quad n \#_S \mathsf{A}}{\langle S \rangle\ \ \Gamma, u : \text{И}x.\mathsf{A} \vdash \Delta}$$

Concluding Remarks

- We defined a modal logic for describing the spatial structure and the behaviour of concurrent systems:
  - Semantics of freshness and recursion (Part I)
  - Proof theory (cut-free proof system) (Part II)
- Key Idea: *modal logics for structured process worlds*
  - *Structural congruence* expresses laws of *spatial structure*
  - *Reduction* expresses laws of *dynamic behaviour*
  - We seek logics to capture both dimensions of concurrent systems
- Spatial logics are very expressive
  - Can talk about fine details of process structure [Sangiorgi01]
  - A degree of intensionality seems needed to describe:
    - spatial distribution
    - resource dependent behaviour