Spatial Logics

Luca Cardelli

Microsoft Research

Bertinoro 2005-04-26

Reflecting joint work with Luís Caires, Andrew D. Gordon.

Spatial Logic

Informal statements:

- Distribution: *Where* are things happening?
- Security: *Where* are things kept, and who can get there?
- Privacy: *Where* are things known, and where are they leaked?

We need a new way of reasoning (i.e. a new logic):

- Classical logic: *Whether* something is true.
- Intuitionistic logic: *How* something is true.
- Temporal logic: *When* something is true.
- *Spatial* logic: *Where* something is true.

Why logic?

- Essentially as a foundation for future type/analysis systems.
- The technical sequent calculus presentation is actually very similar to type systems judgments.

Motivation

We have plenty of logics for *sequential* (i.e. deterministic) computation.

We want logics for *concurrent* computation (Ex.: Hennessy-Milner).

We want logics for *distributed* computation.

- Spatial arrangements of processes are explicit.
- Formulas are modal in time and space.
- The spatial intuition is strong for process calculi with locations.
- But we are now applying it to a standard π-calculus.

We are *not* doing Curry-Howard.

- Because spatial properties are not meant to be preserved by reduction (because of mobility).
- A formula is not realized by a proof tree/computation; it is realized by a *world* (at a particular place and time).

Aim: Describing Distributed Systems

Distributed Systems

- Concurrent systems that are *spatially* distributed.
- And have well-defined subsystems that hold secrets (administrative domains).

Spatial Operators and Spatial Properties

- Are common to all process calculi (e.g., $P \mid Q$).
- Are prominent in calculi with locations (e.g., $n[P]$).
- Spatial properties are finer that popular equivalences such as (temporal) *bisimulation*. (*Cf.* space-time bisimulation.)

We want formal tools to talk about spatial properties.

- So we can precisely describe modern distributed systems.

Spatial Properties: Identifiable Subsystems

A system is often composed of identifiable subsystems.

- "A message is sent from Alice to Bob."
- "The protocol is split between two participants."
- "The virus attacks the server."

Such partitions of a system are (obviously) spatial properties. They correspond to a spatial arrangement of processes in different places.

- Process calculi are *very* good at expressing such arrangements operationally (*c.f.*, chemical semantics, structural congruence).
- To the point that a process is often used as a specification of another process. (We consider this as an anomaly!)
- We want something equally good at the specification, or logical, level.

Spatial Properties: Restricted Resources

A system often restricts the use of certain resources to certain subsystems.

- "A <u>shared private</u> key *n* is established between two processes."
- "A <u>fresh</u> nonce *n* is generated locally and transmitted."
- "The applet runs in a <u>secret</u> sandbox."

Something is *hidden/secret/private* if it is present only in a limited subsystem. So these are spatial properties too.

- If something is secret, by assumption it cannot be known. Still, we want to talk about it in specifications.
- We can talk about a secret name only by using a *fresh* name for it (we cannot assume the secret name matches any known name).
- So freshness will be an important concept. Logics of freshness are very new.

Spatial-style Protocol Specification

Right now, we have a spatial configuration, and later, we have another spatial configuration.

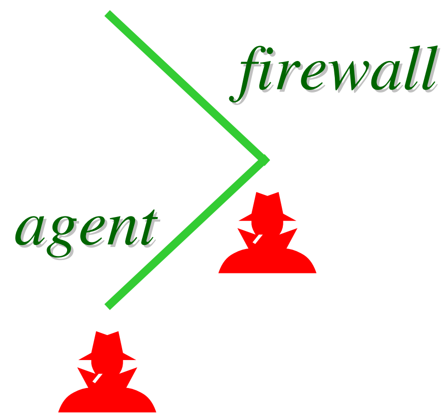
E.g.: Right now, the agent is outside the firewall, …

$(\mathit{agent}[\mathbf{T}] \mid \mathit{firewall}[\mathbf{T}] \mid \mathbf{T})$

Spatial-style Protocol Specification

Right now, we have a spatial configuration, and later, we have another spatial configuration.

E.g.: Right now, the agent is outside the firewall, and later (after running an authentication protocol), the agent is inside the firewall.

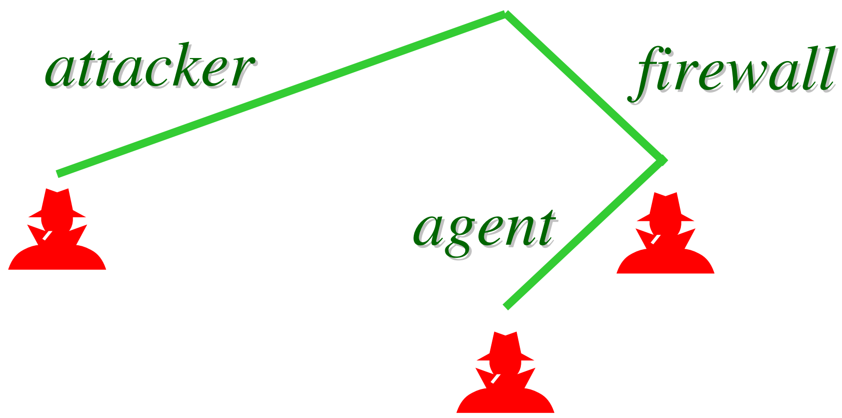


$$(\mathit{agent}[\mathbf{T}] \mid \mathit{firewall}[\mathbf{T}] \mid \mathbf{T}) \wedge \Diamond(\mathit{firewall}[\mathit{agent}[\mathbf{T}] \mid \mathbf{T}] \mid \mathbf{T})$$

Spatial-style Protocol Specification

Right now, we have a spatial configuration, and later, we have another spatial configuration.

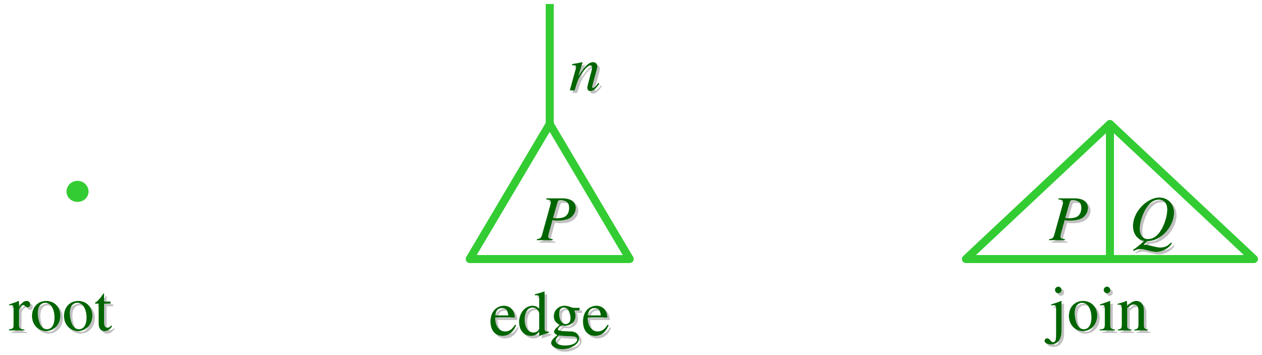
E.g.: Right now, the agent is outside the firewall, and later (after running an authentication protocol), the agent is inside the firewall. And this works in presence of any (reasonable) attacker.



$$\mathit{Attack} \triangleright ((\mathit{agent}[\mathbf{T}] \mid \mathit{firewall}[\mathbf{T}] \mid \mathbf{T}) \wedge \Diamond(\mathit{firewall}[\mathit{agent}[\mathbf{T}] \mid \mathbf{T}] \mid \mathbf{T}))$$

Trees and their Descriptions

Trees



Syntax for Trees (P,Q)

$\mathbf{0}$	root
$n[P]$	edge
$P \mid Q$	join

$P \equiv Q$ iff they represent the same tree.
It is the congruence induced by:

$$P_1 \mid P_2 \equiv P_2 \mid P_1$$
$$P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3$$
$$P \mid \mathbf{0} \equiv P$$

Basic Descriptions ($\mathcal{A}$,$\mathcal{B}$)

$\mathbf{0}$	there is only a root
$n[\mathcal{A}]$	there is an edge n to a subtree
$\mathcal{A} \mid \mathcal{B}$	there are two joined trees
$\mathbf{T}$	there is anything

Formulas and Satisfaction Relation

$P \vDash \mathbf{F}$		*never*	$(\mathbf{T} \triangleq \mathbf{F} \Rightarrow \mathbf{F})$
$P \vDash \mathcal{A} \wedge \mathcal{B}$	$\triangleq$	$P \vDash \mathcal{A} \wedge P \vDash \mathcal{B}$	
$P \vDash \mathcal{A} \Rightarrow \mathcal{B}$	$\triangleq$	$P \vDash \mathcal{A} \Rightarrow P \vDash \mathcal{B}$	
$P \vDash \mathbf{0}$	$\triangleq$	$P \equiv \mathbf{0}$	
$P \vDash \mathcal{A} \mid \mathcal{B}$	$\triangleq$	$\exists P', P'' \in \Pi.\ P \equiv P' \mid P'' \wedge P' \vDash \mathcal{A} \wedge P'' \vDash \mathcal{B}$	
$P \vDash \mathcal{A} \triangleright \mathcal{B}$	$\triangleq$	$\forall P' \in \Pi.\ P' \vDash \mathcal{A} \Rightarrow P \mid P' \vDash \mathcal{B}$	
$P \vDash n[\mathcal{A}]$	$\triangleq$	$\exists P' \in \Pi.\ P \equiv n[P'] \wedge P' \vDash \mathcal{A}$	
$P \vDash \mathcal{A}@n$	$\triangleq$	$n[P] \vDash \mathcal{A}$	

Basic fact: if $P \vDash \mathcal{A}$ and $P \equiv Q$, then $Q \vDash \mathcal{A}$

Model:

- The collection of those sets of P's that are closed under $\equiv$. (I.e., in this simple case, the collection of all sets of trees.)
- A boolean algebra ($\mathbf{F}$ $\wedge$ $\Rightarrow$), a quantale ($\mid$ $\triangleright$), and more.
- With some interesting interactions: $\mathcal{A} \triangleright \mathbf{F}$ = "$\mathcal{A}$ unsatisfiable"

Examples

"Vertical" implications about nesting

"Business Policy"

Borders[
  Starbucks[…] |
  Books[…] |
  Records[…]
]

Borders[**T**] $\Rightarrow$
Borders[*Starbucks*[**T**] | *Books*[**T**] | **T**]

If it's a Borders,
then it must contain a
Starbucks (and some books)

"Horizontal" implications about proximity

"Social Policy"

Smoker[…] /
NonSmoker[…] /
Smoker[…]

(*NonSmoker*[**T**] | **T**) $\Rightarrow$
(*Smoker*[**T**] | **T**)

If there is a NonSmoker,
then there must be a Smoker
nearby

What makes a room bad for a nonsmoker?

$? \vDash \mathit{NonSmoker}[\mathbf{T}] \triangleright \mathit{Pub}$

$\mathit{Pub} \triangleq (\mathit{NonSmoker}[\mathbf{T}] \mid \mathbf{T}) \Rightarrow (\mathit{Smoker}[\mathbf{T}] \mid \mathbf{T})$

Answer: $? = \mathit{Smoker}[\ldots]$

What makes a Borders legal?

$? \vDash \mathit{OkBorders}@\mathit{Borders}$

$\mathit{OkBorders} \triangleq \mathit{Borders}[\mathbf{T}] \Rightarrow \mathit{Borders}[\mathit{Starbucks}[\mathbf{T}] \mid \mathit{Books}[\mathbf{T}] \mid \mathbf{T}]$

Answer: $? = \mathit{Starbucks}[\ldots] \mid \mathit{Books}[\ldots]$

Or illegal:

$? \vDash (\neg \mathit{OkBorders})@\mathit{Borders}$

Answer: $? = \mathit{Books}[\ldots]$

Ground Propositional Spatial Logic (for Trees)

$$\ldots\ t_i : \mathcal{A}_i \ldots\ \vdash\ \ldots\ u_j : \mathcal{B}_j \ldots$$

Identity, Cut, and Contraction

(Id)

$$\frac{t \equiv u}{\Gamma, t : \mathcal{A} \vdash u : \mathcal{A}, \Delta}$$

(Cut)

$$\frac{\Gamma \vdash t : \mathcal{A}, \Delta \quad \Gamma, t : \mathcal{A} \vdash \Delta}{\Gamma \vdash \Delta}$$

(C L)

$$\frac{\Gamma, t : \mathcal{A}, t : \mathcal{A} \vdash \Delta}{\Gamma, t : \mathcal{A} \vdash \Delta}$$

(C R)

$$\frac{\Gamma \vdash t : \mathcal{A}, t : \mathcal{A}, \Delta}{\Gamma \vdash t : \mathcal{A}, \Delta}$$

Propositional Connectives

(F L)

$$\frac{}{\Gamma, t : \mathbf{F} \vdash \Delta}$$

(F R)

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash t : \mathbf{F}, \Delta}$$

(∧ L)

$$\frac{\Gamma, t : \mathcal{A}, t : \mathcal{B} \vdash \Delta}{\Gamma, t : \mathcal{A} \wedge \mathcal{B} \vdash \Delta}$$

(∧ R)

$$\frac{\Gamma \vdash t : \mathcal{A}, \Delta \quad \Gamma \vdash t : \mathcal{B}, \Delta}{\Gamma \vdash t : \mathcal{A} \wedge \mathcal{B}, \Delta}$$

(⇒ L)

$$\frac{\Gamma \vdash t : \mathcal{A}, \Delta \quad \Gamma, t : \mathcal{B} \vdash \Delta}{\Gamma, t : \mathcal{A} \Rightarrow \mathcal{B} \vdash \Delta}$$

(⇒ R)

$$\frac{\Gamma, t : \mathcal{A} \vdash t : \mathcal{B}, \Delta}{\Gamma \vdash t : \mathcal{A} \Rightarrow \mathcal{B}, \Delta}$$

Spatial Connectives

(0 L)
$$\frac{t \not\equiv 0}{\Gamma, t : \mathbf{0} \vdash \Delta}$$

(0 R)
$$\frac{t \equiv 0}{\Gamma \vdash t : \mathbf{0}, \Delta}$$

(| L)
$$\frac{\forall u,v \therefore u|v \equiv t.\quad \Gamma, u : \mathcal{A}, v : \mathcal{B} \vdash \Delta}{\Gamma, t : \mathcal{A} \mid \mathcal{B} \vdash \Delta}$$

(| R)
$$\frac{\exists u,v \therefore u|v \equiv t.\quad \Gamma \vdash u : \mathcal{A}, \Delta \quad \Gamma \vdash v : \mathcal{B}, \Delta}{\Gamma \vdash t : \mathcal{A} \mid \mathcal{B}, \Delta}$$

(▷ L)
$$\frac{\exists u.\quad \Gamma \vdash u : \mathcal{A}, \Delta \quad \Gamma, t|u : \mathcal{B} \vdash \Delta}{\Gamma, t : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta}$$

(▷ R)
$$\frac{\forall u.\quad \Gamma, u : \mathcal{A} \vdash t|u : \mathcal{B}, \Delta}{\Gamma \vdash t : \mathcal{A} \triangleright \mathcal{B}, \Delta}$$

(n[] L)
$$\frac{\forall u \therefore n[u] \equiv t.\quad \Gamma, u : \mathcal{A} \vdash \Delta}{\Gamma, t : n[\mathcal{A}] \vdash \Delta}$$

(n[] R)
$$\frac{\exists u \therefore n[u] \equiv t.\quad \Gamma \vdash u : \mathcal{A}, \Delta}{\Gamma \vdash t : n[\mathcal{A}], \Delta}$$

(@n L)
$$\frac{\Gamma, n[t] : \mathcal{A} \vdash \Delta}{\Gamma, t : \mathcal{A}@n \vdash \Delta}$$

(@n R)
$$\frac{\Gamma \vdash n[t] : \mathcal{A}, \Delta}{\Gamma \vdash t : \mathcal{A}@n, \Delta}$$

Calcagno-Cardelli-Gordon:

***Deciding Validity in a Spatial Logic for Trees*.**

N.B.: neither t nor $\mathcal{A}$ contain variables. Then:

- $t \vDash \mathcal{A}$ is decidable.
- Validity is expressible in the logic, so it is also decidable whether $\mathcal{A}$ is valid (i.e.: whether $0 \vDash (\mathcal{A} \Rightarrow \mathbf{F}) \triangleright \mathbf{F}$).
- There is a finitary version of the proof system.
- There is a complete decision procedure for $\Gamma \vdash \Delta$.

New Logics for Concurrency

In the process of making spatial sense of $n[\mathcal{A}]$, we also had to make spatial sense of $\mathcal{A} \mid \mathcal{B}$. The latter is, in fact, the harder part. So, in retrospect, it makes sense to consider it on its own.

An outcome is spatial logics for CCS/CSP-like process calculi. Basic idea: take a Hennessy-Milner modal logic and add an $\mathcal{A} \mid \mathcal{B}$ operator. ([Dam] Very hard to reconcile with bisimulation.)

One can go further and investigate spatial logics for restriction, with a *hiding quantifier* $\mathsf{H}x.\mathcal{A}$ (e.g. for π-calculus). This is essential for security/privacy specifications.
([Caires] Very hard to reconcile with bisimulation.)

We can make all that work smoothly by taking a very *intensional* point of view. The logical formulas are not *up-to-bisimulation*: they are *up-to-structural-congruence*. This requires a pretty drastic change in point of view.

Caires-Cardelli: ***A Spatial Logic for Concurrency (Part I,II).*** TACS'01, CONCUR'02.

New Type Systems for "Web Data"

Idea: use spatial logic formulas as types, describing the structure of tree-shaped data in a rich and flexible way (c.f. XDuce). Use function types over those data types to type data transformers:

$$\mathrm{Starbucks}[\mathrm{Smoker}[\mathbf{T}] \mid \mathbf{T}] \;\rightarrow\; \mathrm{Starbucks}[\neg(\mathrm{Smoker}[\mathbf{T}] \mid \mathbf{T})]$$

It is possible to extend that idea by using $\mathrm{H}x.\mathcal{A}$ to type hidden/private information:

*Cardelli-Gardner-Ghelli: **Manipulating Trees with Hidden Labels.***

Spatial Logic for π-calculus

We do this kind of thing for a whole asynchronous π-calculus.

This gets considerably more complex, but allows us to the write one-line specifications of spatial properties such as:

The protocol ensures that there is a private name shared between two distinct parts of the system, and nowhere else.

Adding locations (e.g. switching to ambient calculus) is quite easy.

The general methodology seems very flexible.

A Motivating Example

$Client \triangleq Hx.\,(Protocol(x) \mid Request(x))$

A *Client* generates a secret x and then engages in a $Protocol(x)$ (e.g. simply $pub\langle x\rangle$) in order to perform a request $Request(x)$ (e.g. some communication on x) which is uniquely associated with the secret x.

$Server \triangleq Иx.(Protocol(x) \triangleright \Diamond(Handler(x) \mid Server))$

A (recursive) *Server*, in presence of an instance of *Protocol* for a fresh x, produces a $Handler(x)$ uniquely associated with the secret x, and is ready again as a *Server*.

$Client \mid Server \;\Rightarrow\; \Diamond(Server \mid Hx.\,(Request(x) \mid Handler(x)))$

When a client interacts with a server, the result is eventually again a server, together with a <u>private</u> handler for the client request.

We can show this implication in the logic, without looking at any implementation of *Client* and *Server*.

Note the subtle distinction between having/creating a secret (Hx) and obtaining/using a fresh secret ($Иx$).

Typical Spatial Formulas

Processes		*Formulas*	
$\mathbf{0}$	(void)	$\mathbf{0}$	(nothing here)
$P \mid Q$	(composition)	$\mathcal{A} \mid \mathcal{B}$	(two things here)
$n[P]$	(composition)	$n[\mathcal{A}]$	(one thing here)
$(\nu n)P$	(restriction)	$n \circledR \mathcal{A}$	(hidden thing here)
$n\langle m\rangle$	(message)	$n\langle m\rangle$	(a message here)

Modal Logic Revisited

(Alex Simpson's Thesis)

(Id)

$$\frac{}{\langle S\rangle\, \Gamma, x : \mathcal{A} \vdash x : \mathcal{A}, \Delta}$$

Finite graph
$S = \{x_i \rightarrow y_i\}$

x enjoys $\mathcal{A}$

(Cut)

$$\frac{\langle S\rangle\, \Gamma \vdash x : \mathcal{A}, \Delta \qquad \langle S\rangle\, \Gamma, x : \mathcal{A} \vdash \Delta}{\langle S\rangle\, \Gamma \vdash \Delta}$$

+ contraction

(∧ L)

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

(∧ R)

$$\frac{\langle S\rangle\, \Gamma \vdash x : \mathcal{A}, \Delta \qquad \langle S\rangle\, \Gamma \vdash x : \mathcal{B}, \Delta}{\langle S\rangle\, \Gamma \vdash x : \mathcal{A} \wedge \mathcal{B}, \Delta}$$

(⇒ L)

$$\frac{\langle S\rangle\, \Gamma \vdash x : \mathcal{A}, \Delta \qquad \langle S\rangle\, \Gamma, x : \mathcal{B} \vdash \Delta}{\langle S\rangle\, \Gamma, x : \mathcal{A} \Rightarrow \mathcal{B} \vdash \Delta}$$

(⇒ R)

$$\frac{\langle S\rangle\, \Gamma, x : \mathcal{A} \vdash x : \mathcal{B}, \Delta}{\langle S\rangle\, \Gamma \vdash x : \mathcal{A} \Rightarrow \mathcal{B}, \Delta}$$

(F L)

$$\frac{}{\langle S\rangle\, \Gamma, x : \mathbf{F} \vdash \Delta}$$

(F R)

$$\frac{\langle S\rangle\, \Gamma \vdash \Delta}{\langle S\rangle\, \Gamma \vdash x : \mathbf{F}, \Delta}$$

x reduces to y

(◇ L) *y not in the conclusion*

$$\frac{\langle S, x \rightarrow y\rangle\, \Gamma, y : \mathcal{A} \vdash \Delta}{\langle S\rangle\, \Gamma, x : \diamond\mathcal{A} \vdash \Delta}$$

$\diamond\mathcal{A}$: someone I reduce to enjoys $\mathcal{A}$

(◇ R)

$$\frac{\langle S\rangle\, \Gamma \vdash y : \mathcal{A}, \Delta \qquad x \rightarrow_S y}{\langle S\rangle\, \Gamma \vdash x : \diamond\mathcal{A}, \Delta}$$

(□ L)

$$\frac{\langle S\rangle\, \Gamma, y : \mathcal{A} \vdash \Delta \qquad x \rightarrow_S y}{\langle S\rangle\, \Gamma, x : \Box\mathcal{A} \vdash \Delta}$$

$\Box\mathcal{A}$: everyone I reduce to enjoys $\mathcal{A}$

(□ R) *y not in the conclusion*

$$\frac{\langle S, x \rightarrow y\rangle\, \Gamma \vdash y : \mathcal{A}, \Delta}{\langle S\rangle\, \Gamma \vdash x : \Box\mathcal{A}, \Delta}$$

x reduces to arbitrary y

Modal Variations

That is minimal modal logic.

Additional knowledge about the visibility relation (e.g. transitivity) can be added without modifying the rules for logical connectives.

Additional knowledge is embedded in "world" rules for S. E.g.:

Additional Visibility Structure:

(S → refl)

$$\frac{\langle S,\ x{\to}x\rangle\ \Gamma \vdash \Delta}{\langle S\rangle\ \Gamma \vdash \Delta}$$

If $\to$ is by assumption reflexive, we can discard a superfluous assumption that $x{\to}x$

(S → trans)

$$\frac{\langle S,\ x{\to}z\rangle\ \Gamma \vdash \Delta \qquad x\to_S y \qquad y \to_S z}{\langle S\rangle\ \Gamma \vdash \Delta}$$

If $\to$ is by assumption transitive, and can already derive in S that $x{\to}y$ and $y{\to}z$, then we can discard a superfluous assumption that $x{\to}z$

3	$\langle x{\to}x\rangle\ x : \mathcal{A} \vdash x : \mathcal{A}$	(Id)
2	$\langle x{\to}x\rangle\ x : \Box\mathcal{A} \vdash x : \mathcal{A}$	3, (□ L)
1	$\langle\rangle\ x : \Box\mathcal{A} \vdash x : \mathcal{A}$	2, (S → refl)

5	$\langle x{\to}y, y{\to}z, x{\to}z\rangle\ z : \mathcal{A} \vdash z : \mathcal{A}$	(Id)
4	$\langle x{\to}y, y{\to}z, x{\to}z\rangle\ x : \Box\mathcal{A} \vdash z : \mathcal{A}$	5, (□ L)
3	$\langle x{\to}y, y{\to}z\rangle\ x : \Box\mathcal{A} \vdash z : \mathcal{A}$	4, (S → trans)
2	$\langle x{\to}y\rangle\ x : \Box\mathcal{A} \vdash y : \Box\mathcal{A}$	3, (□ R)
1	$\langle\rangle\ x : \Box\mathcal{A} \vdash x : \Box\Box\mathcal{A}$	2, (□ R)

Many-World Sequents for Spatial Logics

$$\langle S \rangle\ \Gamma \vdash \Delta$$

Validity: if all the constraints S_k and all the assumptions Γ_i are satisfied, then one of the conclusions Δ_j is satisfied

(Spatial) equivalence constraints (denote structural congruence)

Indexes (denote processes, i.e. "worlds")

$$\langle \ldots u' \doteq v' \ldots u'' \rightarrow^{a} v'' \ldots \rangle \ldots u : \mathcal{A} \ldots \vdash \ldots v : \mathcal{B} \ldots$$

(Temporal) reduction constraints (denote process reduction)

Formulas (denote properties)

What's going on

Ex.: $\langle x \rightarrow y \rangle\ x : \Box\mathcal{A} \vdash y : \Box\mathcal{A}$

This is a bit strange because we embed a piece of the semantics (the worlds) into the sequents. However it is done abstractly ("x").

It is natural in the sense that sequents looks very much like a type/ND system: there are terms and their "types" $x : \mathcal{A}$.

Unlike a type sytem, the terms on the left of $\vdash$ are not just unrestricted variables. We need the $\langle S \rangle$ part to express constraints on how these terms relate to each other.

Within a single sequent, we can talk about properties of different worlds. This give us lots of freedom and orthogonality in proofs.

Despite the $x : \mathcal{A}$ look, we are not doing Curry-Howard. The terms do not encode proof trees: in standard modal logics, the terms are just variables with no structure. (But we will use structured terms.)

Basic Process Calculus

Intended Model

$P,Q \in \Pi ::=$ Processes

$\mathbf{0}$	void
$P \mid Q$	composition
$n\langle m\rangle$	output ($n,m \in \Lambda$)
$n(m).P$	input

$P \mid \mathbf{0} \equiv P$
$P \mid Q \equiv Q \mid P$
$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$

$P \equiv P$
$P \equiv Q \Rightarrow Q \equiv P$
$P \equiv R \wedge R \equiv Q \Rightarrow P \equiv Q$
$P \equiv Q \Rightarrow P \mid R \equiv Q \mid R$
$P \equiv Q \Rightarrow n(m).P \equiv n(m).Q$

$n\langle m\rangle \mid n(r).P \rightarrow P\{r \leftarrow m\}$
$P \rightarrow Q \;\Rightarrow\; P|R \rightarrow Q|R$
$P \equiv P' \wedge P' \rightarrow Q' \wedge Q' \equiv Q \;\Rightarrow\; P \rightarrow Q$

Labeled transitions:

$P \rightarrow^{\tau} Q \quad \triangleq \quad P \rightarrow Q$
$P \rightarrow^{n\langle m\rangle} Q \quad \triangleq \quad P \equiv n\langle m\rangle \mid Q$ $\qquad x\langle y\rangle^* \triangleq x(y)$
$P \rightarrow^{n(m)} Q \quad \triangleq \quad P \equiv n(p).P' \mid P'' \;\wedge\; Q \equiv P'\{p \leftarrow m\} \mid P''$ $\qquad x(y)^* \triangleq x\langle y\rangle$

Inversion lemmas:

$P|Q \equiv \mathbf{0} \Rightarrow P \equiv \mathbf{0}$

$U|V \equiv T|S \Rightarrow \exists\, x,y,z,w$ s.t. $U \equiv x|y$, $V \equiv z|w$, $T \equiv x|z$, $S \equiv y|w$

$\mathbf{0} \rightarrow^{a} P$ never

$a \neq \tau \wedge U|V \rightarrow^{a} T \Rightarrow \exists\, x,y$ s.t.
$(T = x|V \wedge U \rightarrow^{a} x)$
$\vee\, (T = U|y \wedge V \rightarrow^{a} y)$

$U \rightarrow^{\tau} V \Rightarrow \exists\, x,y,x',y',a$ s.t.
$U = x|y \wedge x \rightarrow^{a} x'$
$\wedge\, y \rightarrow^{a^*} y' \wedge x'|y' = V$

Minimal Process Logic

$\mathcal{A}, \mathcal{B} \in \Phi$ ::=	Formulas		
$\mathbf{F}$	false		
$\mathcal{A} \wedge \mathcal{B}$	conjunction	$\mathcal{A} \Rightarrow \mathcal{B}$	implication
$\mathbf{0}$	void		
$\mathcal{A} \mid \mathcal{B}$	composition	$\mathcal{A} \triangleright \mathcal{B}$	guarantee
$a » \mathcal{A}$	after a	$\mathcal{A} « a$	before a
$\forall x.\mathcal{A}$	universal name quantifier		
$\forall X.\mathcal{A}$	propositional quantifier		
X	propositional variables		
a ::=	Actions ($a \in \mathcal{A}ct$, $x,y \in \mathcal{V}$)		
τ	silent		
$x\langle y \rangle$	output		
$x(y)$	input		

$$» \mathcal{A} \triangleq \tau » \mathcal{A}$$

$$\mathcal{A} « \triangleq \mathcal{A} « \tau$$

Things one can say

Single-threaded (or void):

$\neg(\neg\mathbf{0} \mid \neg\mathbf{0})$ $\qquad$ $(\neg\mathcal{A} \triangleq \mathcal{A}\Rightarrow\mathbf{F})$

Somewhere $\mathcal{A}$ holds:

$\mathcal{A} \mid \mathbf{T}$ $\qquad$ $(\mathbf{T} \triangleq \neg\mathbf{F})$

Output: outputs a message m on n (and is/does nothing else):

$n\langle m\rangle$ $\qquad$ $(n\langle m\rangle \triangleq n\langle m\rangle »\mathbf{0})$

In presence of a message m on n, sends a message n on m and stops:

$n\langle m\rangle \triangleright »m\langle n\rangle$

Fixed input: inputs m on n and then satisfies $\mathcal{A}$:

$n(m)»\mathcal{A}$

Parametric input: inputs some x on n and then satisfies:

$n(x).\mathcal{A} \quad \triangleq \quad \forall x.\ n(x)»\mathcal{A}$

Satisfaction

Intended Model

$\mathbf{P} \triangleq \{S \subseteq \Pi \mid P \in S \wedge P \equiv Q \Rightarrow Q \in S\}$ the *properties*

$P \vDash_\sigma \mathbf{F}$	never
$P \vDash_\sigma \mathcal{A} \wedge \mathcal{B}$	iff $P \vDash_\sigma \mathcal{A} \wedge P \vDash_\sigma \mathcal{B}$
$P \vDash_\sigma \mathcal{A} \Rightarrow \mathcal{B}$	iff $P \vDash_\sigma \mathcal{A} \Rightarrow P \vDash_\sigma \mathcal{B}$
$P \vDash_\sigma \mathbf{0}$	iff $P \equiv 0$
$P \vDash_\sigma \mathcal{A} \mid \mathcal{B}$	iff $\exists P', P'' \in \Pi.\ P \equiv P' \mid P'' \wedge P' \vDash_\sigma \mathcal{A} \wedge P'' \vDash_\sigma \mathcal{B}$
$P \vDash_\sigma \mathcal{A} \triangleright \mathcal{B}$	iff $\forall Q \in \Pi.\ Q \vDash_\sigma \mathcal{A} \Rightarrow P \mid Q \vDash_\sigma \mathcal{B}$
$P \vDash_\sigma a \gg \mathcal{A}$	iff $\exists P' \in \Pi.\ P \rightarrow^{\sigma_a} P' \wedge P' \vDash_\sigma \mathcal{A}$
$P \vDash_\sigma \mathcal{A} \ll a$	iff $\forall P' \in \Pi.\ P' \rightarrow^{\sigma_a} P \Rightarrow P' \vDash_\sigma \mathcal{A}$
$P \vDash_\sigma \forall x.\mathcal{A}$	iff $\forall n \in \Lambda.\ P \vDash_{\sigma\{x \leftarrow n\}} \mathcal{A}$
$P \vDash_\sigma \forall X.\mathcal{A}$	iff $\forall S \in \mathbf{P}.\ P \vDash_{\sigma\{X \leftarrow S\}} \mathcal{A}$
$P \vDash_\sigma X$	iff $P \in \sigma(X)$

Closed formulas denote properties:

$$\forall \mathcal{A} \in \Phi.\ \forall P, Q \in \Pi.\ \{P \mid P \vDash \mathcal{A}\} \in \mathbf{P}$$

N.B.: $\mathbf{P}$ is a commutative quantale and a boolean algebra.

Rules

General pattern:

- *Left rules*, *right rules*. Operate mainly on the $\Gamma \vdash \Delta$ part.
  When operating on constraints $\langle S \rangle$:
  Going up: One adds, the other checks constraints.
  Going down: One removes, the other assumes constraints.
  They form cut elimination pairs.
- *World rules (optional)*. Operate on the $\langle S \rangle$ part only.
  Embody inversion lemmas.
  Going up: add deducible constraints.
  Going down: remove redundant constaints.
  Commute easily with cuts.

$(\mathcal{A} \mid \mathcal{B}) \wedge \mathbf{0} \vdash \mathcal{A} \wedge \mathcal{B}$

6.2 $\langle S,\ u \doteq \mathcal{X} \mid \mathcal{Y},\ u \doteq \mathbf{0},\ \mathcal{X} \doteq \mathbf{0} \rangle\ \Gamma,\ \mathcal{X} : \mathcal{A},\ \mathcal{Y} : \mathcal{B} \vdash u : \mathcal{A},\ \Delta$ (Id) since $u = \mathcal{X}$

5.2 $\langle S,\ u \doteq \mathcal{X} \mid \mathcal{Y},\ u \doteq \mathbf{0} \rangle\ \Gamma,\ \mathcal{X} : \mathcal{A},\ \mathcal{Y} : \mathcal{B} \vdash u : \mathcal{A},\ \Delta$ 6.2, (S | **0**) since $\mathcal{X} \mid \mathcal{Y} = \mathbf{0}$

4.2 $\langle S,\ u \doteq \mathcal{X} \mid \mathcal{Y} \rangle\ \Gamma,\ \mathcal{X} : \mathcal{A},\ \mathcal{Y} : \mathcal{B},\ u : \mathbf{0} \vdash u : \mathcal{A},\ \Delta$ 5.2, (**0** L)

3.2 $\langle S \rangle\ \Gamma,\ u : (\mathcal{A} \mid \mathcal{B}),\ u : \mathbf{0} \vdash u : \mathcal{A},\ \Delta$ 4.2, (| L)

2.2 $\langle S \rangle\ \Gamma,\ u : (\mathcal{A} \mid \mathcal{B}) \wedge \mathbf{0} \vdash u : \mathcal{A},\ \Delta$ 3.2, ($\wedge$ L)

...

2.1 $\langle S \rangle\ \Gamma,\ u : (\mathcal{A} \mid \mathcal{B}) \wedge \mathbf{0} \vdash u : \mathcal{B},\ \Delta$ Similarly

1 $\langle S \rangle\ \Gamma,\ u : (\mathcal{A} \mid \mathcal{B}) \wedge \mathbf{0} \vdash u : \mathcal{A} \wedge \mathcal{B},\ \Delta$ 2.1, 2.2, ($\wedge$ R)

Ex: Immovable Object vs. Irresistible Force

$$Im \triangleq \mathbf{T} \rhd \Box(obj\langle\rangle \mid \mathbf{T})$$

$$Ir \triangleq \mathbf{T} \rhd \Box\Diamond\neg(obj\langle\rangle \mid \mathbf{T})$$

$Im \mid Ir \vdash (\mathbf{T} \rhd \Box(obj\langle\rangle \mid \mathbf{T})) \mid \mathbf{T}$	$\mathcal{A} \vdash \mathbf{T}$
$\vdash \Box(obj\langle\rangle \mid \mathbf{T})$	$(\mathcal{A} \rhd \mathcal{B}) \mid \mathcal{A} \vdash \mathcal{B}$
$\vdash \Diamond\Box(obj\langle\rangle \mid \mathbf{T})$	$\mathcal{A} \vdash \Diamond\mathcal{A}$
$Im \mid Ir \vdash \mathbf{T} \mid (\mathbf{T} \rhd \Box\Diamond\neg(obj\langle\rangle \mid \mathbf{T}))$	$\mathcal{A} \vdash \mathbf{T}$
$\vdash \Box\Diamond\neg(obj\langle\rangle \mid \mathbf{T})$	$\Diamond\neg\mathcal{A} \vdash \neg\Box\mathcal{A}$
$\vdash \neg\Diamond\Box(obj\langle\rangle \mid \mathbf{T})$	$\Box\neg\mathcal{A} \vdash \neg\Diamond\mathcal{A}$
Hence: $Im \mid Ir \vdash \mathbf{F}$	$\mathcal{A} \wedge \neg\mathcal{A} \vdash \mathbf{F}$