Anytime, Anywhere
Modal Logics for Mobile Ambients

Luca Cardelli
Andy Gordon
Microsoft Research

POPL 2000
Introduction

- We want to describe mobile behaviors. The *ambient calculus* provides an operational model, where spatial structures (agents, networks, etc.) are represented by nested locations.

- We also want to specify mobile behaviors. To this end, we devise an *ambient logic* that can talk about spatial structures.

**Processes**

- $0$ (void)
- $n[P]$ (location)
- $P \mid Q$ (composition)

**Formulas**

- $0$ (there is nothing here)
- $n[A]$ (there is one thing here)
- $A \mid B$ (there are two things here)

**Trees**

- (void)
- (location)
- (composition)
Spatial Structures

- Our basic model of space is going to be finite-depth edge-labeled unordered trees; for short: spatial trees, represented by a syntax of spatial expressions. Unbounded resources are represented by infinite branching:

```
Cambridge
  Eagle
  chair chair glass glass glass ...
  pint pint pint pint ...
```

`Cambridge[Eagle[chair[0] | chair[0] | !glass[pint[0]]]] | ...)`
Spatial expressions/trees are a subset of ambient expressions/trees, which can represent both the spatial and the dynamic aspects of mobile computation.

An ambient tree is a spatial tree with, possibly, threads at each node that can locally change the shape of the tree.

\[ a[c[\text{out a. in b. P}]] \mid b[0] \]
Mobility

- *Mobility* is change of spatial structures over time.

\[a[Q \parallel c[\text{out } a. \text{ in } b. \ P]] \parallel b[R] \]
**Mobility**

- **Mobility** is change of spatial structures over time.

\[
a[Q] \quad \mid \quad c[\text{in } b. \ P] \quad \mid \quad b[R]
\]
Mobility

- **Mobility** is change of spatial structures over time.
# Restriction-Free Ambient Calculus

\[
P \in \Pi ::= \begin{align*}
P & \mid 0 \quad \text{inactivity} \\
\ & \mid P \mid P' \quad \text{parallel} \\
\ & \mid !P \quad \text{replication} \\
\ & \mid M[P] \quad \text{ambient} \\
\ & \mid M.P \quad \text{exercise a capability} \\
\ & \mid (n).P \quad \text{input locally, bind to } n \\
\ & \mid \langle M \rangle \quad \text{output locally (async)}
\end{align*}
\]

\[
M ::= \begin{align*}
\ & \mid n \quad \text{name} \\
\ & \mid in M \quad \text{entry capability} \\
\ & \mid out M \quad \text{exit capability} \\
\ & \mid open M \quad \text{open capability} \\
\ & \mid \epsilon \quad \text{empty path} \\
\ & \mid M.M' \quad \text{composite path}
\end{align*}
\]

\[
\begin{align*}
n[] & \triangleq n[0] \\
M & \triangleq M.0 \quad (\text{where appropriate})
\end{align*}
\]
Reduction Semantics

• A structural congruence relation $P \equiv Q$:
  • On spatial expressions, $P \equiv Q$ iff $P$ and $Q$ denote the same tree.
  • On full ambient expressions, $P \equiv Q$ if in addition the respective threads are “trivially equivalent”.
  • Prominent in the definition of the logic.

• A reduction relation $P \rightarrow^* Q$:
  • Defining the meaning of mobility and communication actions.
  • Closed up to structural congruence:
    \[ P \equiv P', \quad P' \rightarrow^* Q', \quad Q' \equiv Q \quad \Rightarrow \quad P \rightarrow^* Q \]
Space-Time Modalities

• In a modal logic, the truth of a formula is relative to a state (called a *world*).

• In our case, the truth of a space-time modal formula is relative to the *here and now* of a process.
  • The formula $n[0]$ is read:
    
    there is here and now an empty location called $n$
  • The operator $n[A]$ is a single step in space (akin to the temporal next), which allows us talk about that place one step down into $n$.
  • Other modal operators can be used to talk about undetermined times (in the future) and undetermined places (in the location tree).
Logical Formulas

\[ \forall x.A \]

Formulas

- \( T \) (true)
- \( \neg A \) (negation)
- \( A \lor A' \) (disjunction)
- \( 0 \) (void)
- \( \eta[A] \) (location)
- \( A \mid A' \) (composition)
- \( \Diamond A \) (somewhere modality)
- \( \Box A \) (sometime modality)
- \( A @ \eta \) (location adjunct)
- \( A \rightarrow A' \) (composition adjunct)
- \( \forall x. A \) (universal quantification over names)

(\( \eta \) is a name or a variable)
Satisfaction Relation

\[ P \models T \]
\[ P \models \neg A \quad \triangleq \quad \neg P \models A \]
\[ P \models A \lor B \quad \triangleq \quad P \models A \lor P \models B \]
\[ P \models 0 \quad \triangleq \quad P \equiv 0 \]
\[ P \models n[A] \quad \triangleq \quad \exists P' \in \Pi. \quad P \equiv n[P'] \land P' \models A \]
\[ P \models A \triangledown B \quad \triangleq \quad \exists P', P'' \in \Pi. \quad P \equiv P' \rightarrow P'' \land P' \models A \land P'' \models B \]
\[ P \models \bigotimes A \quad \triangleq \quad \exists P' \in \Pi. \quad P \downarrow * P' \land P' \models A \]
\[ P \models \bigcirc A \quad \triangleq \quad \exists P' \in \Pi. \quad P \rightarrow * P' \land P' \models A \]
\[ P \models A@n \quad \triangleq \quad n[P] \models A \]
\[ P \models A @ B \quad \triangleq \quad \forall P' \in \Pi. \quad P' \models A \Rightarrow P \downarrow P' \models B \]
\[ P \models \forall x. A \quad \triangleq \quad \forall m \in \Lambda. \quad P \models A \{x \leftarrow m\} \]

\[ P \downarrow P' \quad \text{iff} \quad \exists n, P''. \quad P \equiv n[P'] \downarrow P'' \]
\[ \downarrow \ast \quad \text{is the reflexive and transitive closure of} \quad \downarrow \]
Satisfaction Relation for Trees

- \[ P \vdash 0 \]

\[ \vdash n[A] \quad \text{if} \quad P \vdash A \]

\[ P \quad Q \quad \vdash A \mid B \quad \text{if} \quad P \vdash A \quad \text{and} \quad Q \vdash B \]

\[ P \quad \vdash \ast[A] \quad \text{if} \quad Q \vdash A \]

\[ P \quad \vdash \diamond A \quad \text{if} \quad P \rightarrow^* Q \quad \text{and} \quad Q \vdash A \]
Basic Fact: satisfaction is invariant under structural congruence:

\[ P \models A \land n \quad \text{if} \quad n \models A \]

\[ P \models A \rightarrow B \quad \text{if for all} \quad Q \models A \quad \text{we have} \quad P \land Q \models B \]

- I.e.: \( \{ P \in \Pi \mid P \models A \} \) is closed under \( \equiv \).

Hence, formulas describe only congruence-invariant properties.
### Some Derived Connectives

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>≡ ¬T</td>
<td>false</td>
</tr>
<tr>
<td>A ∧ B</td>
<td>≡ ¬(¬A ∨ ¬B)</td>
<td>conjunction</td>
</tr>
<tr>
<td>A ⇒ B</td>
<td>≡ ¬A ∨ B</td>
<td>implication</td>
</tr>
<tr>
<td>□ A</td>
<td>≡ ¬◊¬A</td>
<td>everytime modality</td>
</tr>
<tr>
<td>◊ A</td>
<td>≡ ¬¬◊¬A</td>
<td>everywhere modality</td>
</tr>
<tr>
<td>∃ x. A</td>
<td>≡ ¬∀x.¬A</td>
<td>existential quantification</td>
</tr>
<tr>
<td>A ∽ B</td>
<td>≡ ¬(B ⊳ ¬A)</td>
<td>fusion</td>
</tr>
<tr>
<td>A ⊳ B</td>
<td>≡ ¬(A</td>
<td>¬B)</td>
</tr>
<tr>
<td>A</td>
<td>≡ ¬(¬A</td>
<td>¬B)</td>
</tr>
<tr>
<td>A</td>
<td>≡ A ─ F</td>
<td>every component satisfies A</td>
</tr>
<tr>
<td>A</td>
<td>≡ A</td>
<td>T</td>
</tr>
<tr>
<td>A</td>
<td>≡ A</td>
<td>□ F</td>
</tr>
</tbody>
</table>
Claims

• The satisfaction relation looks natural (to us):
  • The definitions of $0$, $n[A]$, and $A \parallel B$ seem inevitable, once we accept that formulas should be able to talk about the tree structure of locations (up to $\equiv$).
  • The connectives $A@n$ and $A\triangleright B$ have security motivations.
  • The modalities $\Diamond A$ and $\Diamond A$ talk about process evolution and structure in an undetermined way (good for specs).
  • The fragment $T, \neg A, A \lor B, \forall x.A$, is classical: why not?

• The logic is induced by the satisfaction relation.
  • We did not have any preconceptions about what kind of logic this ought to be. We didn’t invent this logic, we discovered it!
From Satisfaction to (Propositional) Logic

- Propositional validity
  \[ \text{vld } \mathcal{A} \equiv \forall P \in \Pi. P \vdash \mathcal{A} \]
  \( \mathcal{A} \) (closed) is valid

- Sequents
  \[ \mathcal{A} \vdash B \equiv \forall P \in \Pi. P \vdash \mathcal{A} \Rightarrow P \vdash B \]

- Rules
  \[ \mathcal{A}_1 \vdash B_1; \ldots; \mathcal{A}_n \vdash B_n \{ \{ \} \} \mathcal{A} \vdash B \equiv (n \geq 0) \]
  \[ \mathcal{A}_1 \vdash B_1 \land \ldots \land \mathcal{A}_n \vdash B_n \Rightarrow \mathcal{A} \vdash B \]
  (N.B.: all the rules shown later are validated accordingly.)

- Conventions:
  - \( \vdash \) means \( \vdash \) in both directions
  - \{ \} means \{ \} in both directions
Logical Adjunctions

- This is a logic with multiple logical adjunctions (3 of them!):

  \( \land / \Rightarrow \) (classical)
  
  \[ \mathcal{A} \land C \vdash B \iff \mathcal{A} \vdash C \Rightarrow B \]

- 1 / \( \triangleright \) (linear, \( \otimes / \leftarrow \))

  \[ \mathcal{A} \triangleright C \vdash B \iff \mathcal{A} \vdash C \triangleright B \]

- \( n[-] / \sim \bowtie n \)

  \[ n[\mathcal{A}] \vdash B \iff \mathcal{A} \vdash B \bowtie n \]

- Which one should be taken as the logical adjunction for sequents? I.e., what should "","" mean in a sequent?
“Neutral” Sequent

Our logic is formulated with single-premise, single-conclusion sequents. We don’t pre-judge “,”.

• By taking $\land$ on the left and $\lor$ on the right of $\vdash$ as structural operators, we can derive all the standard rules of sequent and natural deduction systems with multiple premises/conclusions.

• By taking $|$ on the left of $\vdash$ as a structural operator, we can derive all the rules of intuitionistic linear logic (by appropriate mappings of the ILL connectives).

• By taking nestings of $\land$ and $|$ on the left of $\vdash$ as structural “bunches”, we obtain a bunched logic, with its two associated implications, $\Rightarrow$ and $\triangleright$.

• This is convenient. We do not know much, however, about the meta-theory of this presentation style.
Rules: Propositional Calculus

(A-L) $\mathcal{A} \land (\mathcal{C} \land \mathcal{D}) \vdash B \quad \{ \} \quad (\mathcal{A} \land \mathcal{C}) \land \mathcal{D} \vdash B$

(A-R) $\mathcal{A} \vdash (\mathcal{C} \lor \mathcal{D}) \lor B \quad \{ \} \quad \mathcal{A} \vdash \mathcal{C} \lor (\mathcal{D} \lor B)$

(X-L) $\mathcal{A} \land \mathcal{C} \vdash B \quad \{ \} \quad \mathcal{C} \land \mathcal{A} \vdash B$

(X-R) $\mathcal{A} \vdash \mathcal{C} \lor B \quad \{ \} \quad \mathcal{A} \vdash \mathcal{B} \lor \mathcal{C}$

(C-L) $\mathcal{A} \land \mathcal{A} \vdash B \quad \{ \} \quad \mathcal{A} \vdash B$

(C-R) $\mathcal{A} \vdash \mathcal{B} \lor \mathcal{B} \quad \{ \} \quad \mathcal{A} \vdash \mathcal{B}$

(W-L) $\mathcal{A} \vdash B \quad \{ \} \quad \mathcal{A} \land \mathcal{C} \vdash B$

(W-R) $\mathcal{A} \vdash B \quad \{ \} \quad \mathcal{A} \vdash \mathcal{C} \lor B$

(Id) $\{ \} \quad \mathcal{A} \vdash \mathcal{A}$

(Cut) $\mathcal{A} \vdash \mathcal{C} \lor B; \mathcal{A} \land \mathcal{C} \vdash B' \quad \{ \} \quad \mathcal{A} \land \mathcal{A} \vdash B \lor B'$

(T) $\mathcal{A} \land \mathcal{T} \vdash B \quad \{ \} \quad \mathcal{A} \vdash B$

(F) $\mathcal{A} \vdash \mathcal{F} \lor \mathcal{B} \quad \{ \} \quad \mathcal{A} \vdash \mathcal{B}$

(\neg-L) $\mathcal{A} \vdash \mathcal{C} \lor \mathcal{B} \quad \{ \} \quad \mathcal{A} \land \neg \mathcal{C} \vdash \mathcal{B}$

(\neg-R) $\mathcal{A} \land \mathcal{C} \vdash \mathcal{B} \quad \{ \} \quad \mathcal{A} \vdash \neg \mathcal{C} \lor \mathcal{B}$
Rules: Composition

(\| 0) \{ \mathcal{A} \| 0 \rightarrow \mathcal{A} \} \quad 0 \text{ is nothing}

(\| \neg 0) \{ \mathcal{A} \| \neg 0 \rightarrow \neg 0 \} \quad \text{if a part is non-} 0, \text{ so is the whole}

(A \|) \{ \mathcal{A} \| (B \| C) \rightarrow (\mathcal{A} \| B) \| C \} \quad \text{\| associativity}

(X \|) \{ \mathcal{A} \| B \rightarrow B \| \mathcal{A} \} \quad \text{\| commutativity}

(\| \vdash) \{ \mathcal{A}' \vdash B'; \mathcal{A}'' \vdash B'' \} \quad \mathcal{A}' \| \mathcal{A}'' \vdash B' \| B'' \} \quad \text{\| congruence}

(\| \lor) \{ (\mathcal{A} \lor B) \| C \rightarrow \mathcal{A} \| C \lor B \| C \} \quad \text{\|\lor distribution}

(\| \land) \{ \mathcal{A}' \| \mathcal{A}'' \vdash \mathcal{A}' \| B' \lor \mathcal{A}'' \lor \neg B' \| \neg B'' \} \quad \text{decomposition}

(\| \rightarrow) \{ \mathcal{A} \| C \rightarrow B \} \{ \{ \mathcal{A} \rightarrow C \rightarrow B \} \quad \text{\|\rightarrow adjunction}

(\triangleright F \rightarrow) \{ \mathcal{A}^F \rightarrow \mathcal{A}^\neg \} \quad \text{if } \mathcal{A} \text{ is unsatisfiable then } \mathcal{A} \text{ is false}

(\neg \triangleright F) \{ \mathcal{A}^F \rightarrow \mathcal{A}^{FF} \} \quad \text{if } \mathcal{A} \text{ is satisfiable then } \mathcal{A}^F \text{ is unsatisfiable}

\text{where } \mathcal{A}^\neg \triangleq \neg \mathcal{A} \text{ and } \mathcal{A}^F \triangleq \mathcal{A} \triangleright F
The Decomposition Operator

• Consider the De Morgan dual of $|$:

$A \parallel B \triangleq \neg (\neg A \lor \neg B)$

$P \models - \iff \forall P', P'' \in \Pi. P \equiv P' \mid P'' \Rightarrow$

$P' \models A \lor P'' \models B$

$A^\forall \triangleq A \parallel \bot$

$P \models - \iff \forall P', P'' \in \Pi. P \equiv P' \mid P'' \Rightarrow P' \models A$

$A^\exists \triangleq A \parallel \top$

$P \models - \iff \exists P', P'' \in \Pi. P \equiv P' \mid P'' \land P' \models A$

$A \parallel B$ for every partition, one piece satisfies $A$
or the other piece satisfies $B$

$A^\forall \iff \neg ((\neg A)^\exists)$ every component satisfies $A$

$A^\exists \iff \neg ((\neg A)^\forall)$ some component satisfies $A$

Examples:

$(p[T] \Rightarrow p[q[T]^\exists])^\forall$ every $p$ has a $q$ child

$(p[T] \Rightarrow p[q[T] \mid (\neg q[T])^\forall])^\forall$ every $p$ has a unique $q$ child
The Decomposition Axiom

\[(1 \| ) \quad \{(A' | A'') \vdash (A' | B'') \lor (B' | A'') \lor (\neg B' | \neg B'')\]

- **Alternative formulations and special cases:**
  
  \[\{(A' | A'') \land (B' \| B'') \vdash (A' | B'') \lor (B' | A'')\]

  “If \(P\) has a partition into pieces that satisfy \(A'\) and \(A''\), and every partition has one piece that satisfies \(B'\) or the other that satisfies \(B''\), then either \(P\) has a partition into pieces that satisfy \(A'\) and \(B''\), or it has a partition into pieces that satisfy \(B'\) and \(A''\).”

\[\neg(A | B) \vdash (A | T) \Rightarrow (T | \neg B)\]

“If \(P\) has no partition into pieces that satisfy \(A\) and \(B\), but \(P\) has a piece that satisfies \(A\), then \(P\) has a piece that does not satisfy \(B\).”

\[\neg(T | B) \vdash T | \neg B\]

\[\neg(A | B) \vdash (\neg A | T) \lor (T | \neg B)\]
The Composition Adjunct

\[(\triangleright) \quad \mathcal{A} \triangleright \mathcal{C} \vdash \mathcal{B} \quad \text{iff} \quad \mathcal{A} \vdash \mathcal{C} \triangleright \mathcal{B} \]

“Assume that every process that has a partition into pieces that satisfy \( \mathcal{A} \) and \( \mathcal{C} \), also satisfies \( \mathcal{B} \). Then, every process that satisfies \( \mathcal{A} \), together with any process that satisfies \( \mathcal{C} \), satisfies \( \mathcal{B} \). (And vice versa.)” (c.f. \( \neg \triangleright \mathcal{R} \))

- **Interpretations of \( \mathcal{A} \triangleright \mathcal{B} \):**
  - \( P \) provides \( \mathcal{B} \) in any context that provides \( \mathcal{A} \)
  - \( P \) ensures \( \mathcal{B} \) under any attack that ensures \( \mathcal{A} \)

That is, \( P \vdash \mathcal{A} \triangleright \mathcal{B} \) is a context-system spec (a concurrent version of a pre-post spec).

Moreover \( \mathcal{A} \triangleright \mathcal{B} \) is, in a precise sense, linear implication: the context that satisfies \( \mathcal{A} \) is used exactly once in the system that satisfies \( \mathcal{B} \).
Some Derived Rules

\{(A \triangleright B) | A \vdash B\}

“If \(P\) provides \(B\) in any context that provides \(A\), and \(Q\) provides \(A\), then \(P\) and \(Q\) together provide \(B\).”

- Proof: \(A \triangleright B \vdash A \triangleright B \{ (A \triangleright B) | A \vdash B \) by (Id), (|>)

\[D \vdash A; \ B \vdash C \{ D | (A \triangleright B) \vdash C \] (c.f. (\(\neg\) L))

“If anything that satisfies \(D\) satisfies \(A\), and anything that satisfies \(B\) satisfies \(C\), then: anything that has a partition into a piece satisfying \(D\) (and hence \(A\)), and another piece satisfying \(B\) in a context that satisfies \(A\), it satisfies \((B\) and hence\() C\).”

- Proof:
  \[D \vdash A; \ A \triangleright B \vdash A \triangleright B \{ D | A \triangleright B \vdash A | A \triangleright B \] assumption, (Id), (|\(\leftarrow\))
  \[A | A \triangleright B \vdash B\] above
  \[B \vdash C\] assumption
More Derived Rules

\{ A \vdash T \mid A \} 
\{ F \mid A \vdash F \} 
\{ 0 \vdash \neg(\neg 0 \mid \neg 0) \} 
\{ A \mid B \land 0 \vdash A \} 

\{ A \vdash A ; B \vdash B' \} 
\{ A \rightarrow B \vdash A \rightarrow B' \} 
\{ A \rightarrow B \mid B \rightarrow C \vdash A \rightarrow C \} 
\{ (A \mid B) \rightarrow C \vdash A \rightarrow (B \rightarrow C) \} 
\{ A \rightarrow (B \rightarrow C) \vdash B \rightarrow (A \rightarrow C) \} 

\{ T \vdash T \rightarrow T \} 
\{ T \vdash F \rightarrow A \} 
\{ T \rightarrow A \vdash A \}

you can always add more pieces (if they are 0)

if a piece is absurd, so is the whole

0 is single-threaded

you can split 0 (but you get 0). Proof uses ( || )

\( \rightarrow \) is contravariant on the left

\( \rightarrow \) is transitive

\( \rightarrow \) curry/uncurry

contexts commute

truth can withstand any attack

anything goes if you can find an absurd partner

if A resists any attack, then it holds
### Rules: Location

- **Locations exist**
  - $(n[] \dashv 0)$  \Rightarrow  $n[\overline{A}] \vdash \neg 0$
  - $(n[] \dashv \neg 1)$  \Rightarrow  $n[\overline{A}] \vdash \neg (\neg 0 \mid \neg 0)$

- **Are not decomposable**
  - $(n[] \vdash)$  $\overline{A} \vdash B$  \Rightarrow  $n[\overline{A}] \vdash n[B]$
  - $(n[] \land)$  $\Rightarrow  n[\overline{A}] \land n[C] \vdash n[\overline{A} \land C]$
  - $(n[] \lor)$  $\Rightarrow  n[C \lor B] \vdash n[C] \lor n[B]$

- **$n[]$ congruence**
  - $(n[] @)$  $n[\overline{A}] \vdash B$  \Rightarrow  $\overline{A} \vdash B@n$

- **$n[]$-$\land$ distribution**
  - $(\neg @)$  $\overline{A}@n \dashv \neg ((\neg A)@n)$

- **$n[]$-$\lor$ distribution**
  - $\overline{A}@n$ is self-dual
### Rules: Time and Space Modalities

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(◊)</td>
<td>◊A ⊨ ¬□¬A</td>
</tr>
<tr>
<td>(□K)</td>
<td>□(A⇒B) ⊨ □A⇒□B</td>
</tr>
<tr>
<td>(□T)</td>
<td>□A ⊨ A</td>
</tr>
<tr>
<td>(□4)</td>
<td>□A ⊨ □□A</td>
</tr>
<tr>
<td>(□T)</td>
<td>T ⊨ □T</td>
</tr>
<tr>
<td>(□⊥)</td>
<td>A ⊨ B ⊨ □A⇒□B</td>
</tr>
<tr>
<td>(◊n[])</td>
<td>n[◊A] ⊨ ◊n[A]</td>
</tr>
<tr>
<td>(◊⊥)</td>
<td>A ⊨ B ⊨ □A⇒□B</td>
</tr>
<tr>
<td>(◊◊)</td>
<td>◊◊A ⊨ ◊◊A</td>
</tr>
</tbody>
</table>

S4, but not S5:  \( \neg \vDash \) ◊A ⊨ □◊A  \( \neg \vDash \) ◊◊A ⊨ □◊◊A

(◊◊): if somewhere sometime A, then sometime somewhere A
Some Derived Rules

$A \vdash B \} A@n \vdash B@n$  \hspace{1cm} \@ congruence

\{ $n[A@n] \vdash A$
\{ $A \vdash n[A]@n$

\{ $n[\neg A] \vdash \neg n[A]$
\{ $\neg n[A] \vdash \neg n[T] \lor n[\neg A]$
Examples

- \( an\ n \triangleq n[T] \mid T \) there is now an \( n \) here
- \( no\ n \triangleq \neg an\ n \) there is now no \( n \) here
- \( one\ n \triangleq n[T] \mid no\ n \) there is now exactly one \( n \) here
- \( \forall \triangleq \neg (\neg \forall \mid T) \) everybody here satisfies \( \forall \)
- \( (n[T] \Rightarrow n[\forall]) \forall \) every \( n \) here satisfies \( \forall \)
- \( \Box ((n[T] \Rightarrow n[\forall]) \forall) \) every \( n \) everywhere satisfies \( \forall \)
Ex: Immovable Object vs. Irresistible Force

\[ Im \cong T \triangleright \Box (\text{obj}[0] \mid T) \]
\[ Ir \cong T \triangleright \Box \Diamond \neg (\text{obj}[0] \mid T) \]

\[ Im \mid Ir = (T \triangleright \Box (\text{obj}[0] \mid T)) \mid Ir \]
\[ \vdash \Box (\text{obj}[0] \mid T) \]
\[ \vdash \Diamond \text{P}(\text{obj}[0] \mid T) \]

\[ Im \mid Ir = Im \mid (T \triangleright \Box \Diamond \neg (\text{obj}[0] \mid T)) \]
\[ \vdash \Box \neg \text{P}(\text{obj}[0] \mid T) \]
\[ \vdash \neg \Diamond \text{P}(\text{obj}[0] \mid T) \]

Hence: \[ Im \mid Ir \vdash F \]
Model Checking

- If $P$ is $!$-free and $\mathcal{A}$ is $\triangleright$-free, then $P \models \mathcal{A}$ is decidable.

- This provides a way of mechanically checking (certain) assertions about (certain) mobile processes.

- Potential application: checking (the bytecode of) mobile agents against the internal mobility policies of receiving sites. (I.e.: conferring more flexibility than just sandboxing the agent.)
Connections with Intuitionistic Linear Logic

- Weakening and contraction are not valid rules: principle of conservation of space.

- Semantic connection: sets of processes closed under $\equiv$ and ordered by inclusion form a quantale (a model of ILL).

- Multiplicative intuitionistic linear logic (MILL) can be faithfully embedded in our logic:

$$
\begin{align*}
1_{\text{MILL}} & \triangleq 0 \\
A \otimes_{\text{MILL}} B & \triangleq A \mid B \\
A \rightarrow_{\text{MILL}} B & \triangleq A \triangleright B
\end{align*}
$$

MILL rules and our rules are interderivable ("our rules" means the rules involving only $0$, $\mid$, $\triangleright$, plus a derivable cut rule for $\mid$).
• Full intuitionistic linear logic (ILL) can be embedded:

| 1_{\text{ILL}} | \equiv | 0         | \quad A \oplus B | \equiv | A \lor B  |
|-------------|---------|-------------|-----------------|---------|
| \bot_{\text{ILL}} | \equiv | F         | \quad A & B   | \equiv | A \land B |
| \top_{\text{ILL}}  | \equiv | T         | \quad A \otimes B | \equiv | A \mid B  |
| 0_{\text{ILL}}    | \equiv | F         | \quad A \rightarrow B | \equiv | A \triangleright B |
| !A            | \equiv | 0 \land (0 \Rightarrow A)^\neg F |

• The rules of ILL can be logically derived from these definitions. (E.g.: the proof of \(!A \vdash !A \otimes !A\) uses the decomposition axiom.)

• So, \(A_1, \ldots, A_n \vdash_{\text{ILL}} B\) implies \(A_1 \mid \ldots \mid A_n \vdash B\).

• Some discrepancies: \(\bot_{\text{ILL}} = 0_{\text{ILL}}\); the additives distribute; \(!A\) is not “replication”; \(!A \rightarrow B\) is not so interesting; \(A^\bot / A^0\) is unusually interesting.
Connection with Relevant Logic

• (Noted after the fact [O’Hearn, Pym].) The definition of the satisfaction relation is very similar to Urquhart’s semantics of relevant logic. In particular $\mathcal{A} \mid \mathcal{B}$ is defined just like intensional conjunction, and $\mathcal{A} \supset \mathcal{B}$ is defined just like relevant implication in that semantics.

• Except:
  • We do not have contraction. This does not make sense in process calculi, because $P \mid P \neq P$. Urquhart semantics without contraction does not seem to have been studied.
  • We use an equivalence $\equiv$, instead of a Kripke-style partial order $\varnothing$ as in Urquhart’s general case. (We may have a need for a partial order in more sophisticated versions of our logic.)
Connections with Bunched Logic

• Peter O’Hearn and David Pym study *bunched logics*, where sequents have two structural combinators, instead of the standard single “,” combinator (usually meaning $\land$ or $\otimes$ on the left) found in most presentations of logic. Thus, sequents are *bunches* of formulas, instead of lists of formulas. Correspondingly, there are two implications that arise as the adjuncts of the two structural combinators.

• The situation is very similar to our combinators $|$ and $\land$, which can combine to irreducible bunches of formulas in sequents, and to our two implications $\Rightarrow$ and $\triangleright$. However, we have a classical and a linear implication, while bunched logics have so far had an intuitionistic and a linear implication.
Conclusions

• The novel aspects of our logic lie in its explicit treatment of space and of the evolution of space over time (mobility). The logic has a linear flavor in the sense that space cannot be instantly created or deleted, although it can be transformed over time.

• These ideas can be applied to any process calculus that embodies a distinction between topological and dynamic operators.

• Our logical rules arise from a particular model. This approach makes the logic very concrete, but raises questions of logical completeness, which are being investigated.

• We are now working on generalizing the logic to the full ambient calculus (including restriction), in order to talk about properties of hidden/secret locations.
Ambient Calculus: Example

The packet \textit{msg} moves from \textit{a} to \textit{b}, mediated by the capabilities \textit{out a} (to exit \textit{a}), \textit{in b} (to enter \textit{b}), and \textit{open msg} (to open the \textit{msg} envelope).

\[
\begin{align*}
\text{location } a & \quad \text{location } b \\
\text{send } M \text{ from } a \text{ to } b & \quad \text{receive } n; \text{ do } P \\
\textit{a[msg[\langle M \rangle \mid \text{out a. in b. } P]]} & \quad \textit{b[open msg. (n). P]}
\end{align*}
\]
Reduction

- Four basic reductions plus propagation, rearrangement (composition with structural congruence), and transitivity.

\[
\begin{align*}
\text{Red In} & : n[\text{in } m. \, P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R] \\
\text{Red Out} & : m[n[\text{out } m. \, P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R] \\
\text{Red Open} & : \text{open } m. \, P \mid m[Q] \rightarrow P \mid Q \\
\text{Red Comm} & : (n).P \mid \langle M \rangle \rightarrow P\{n \leftarrow M\}
\end{align*}
\]

\[
\begin{align*}
P \rightarrow Q & \Rightarrow n[P] \rightarrow n[Q] \quad \text{(Red Amb)} \\
P \rightarrow Q & \Rightarrow P \mid R \rightarrow Q \mid R \quad \text{(Red Par)} \\
P' \equiv P, \, P \rightarrow Q, \, Q \equiv Q' & \Rightarrow P' \rightarrow Q' \quad \text{(Red } \equiv \text{)}
\end{align*}
\]

\[\rightarrow^* \text{ is the reflexive-transitive closure of } \rightarrow\]
Structural Congruence

- Routine definition, but used heavily in the logic and semantics.

\[
P \equiv P
\]

\[
P \equiv Q \Rightarrow Q \equiv P
\] (Struct Refl)

\[
P \equiv Q, Q \equiv R \Rightarrow P \equiv R
\] (Struct Symm)

\[
P \equiv Q \Rightarrow P \parallel R \equiv Q \parallel R
\] (Struct Trans)

\[
P \equiv Q \Rightarrow !P \equiv !Q
\] (Struct Par)

\[
P \equiv Q \Rightarrow M[P] \equiv M[Q]
\] (Struct Repl)

\[
P \equiv Q \Rightarrow M.P \equiv M.Q
\] (Struct Amb)

\[
P \equiv Q \Rightarrow (n).P \equiv (n).Q
\] (Struct Action)

\[
\varepsilon.P \equiv P
\] (Struct Input)

\[
(M.M').P \equiv M.M'.P
\] (Struct .)
\begin{align*}
    P \parallel Q & \equiv Q \parallel P \\
    (P \parallel Q) \parallel R & \equiv P \parallel (Q \parallel R) \\
    P \parallel 0 & \equiv P \\
    ! (P \parallel Q) & \equiv !P \parallel !Q \\
    !0 & \equiv 0 \\
    !P & \equiv P \parallel !P \\
    !P & \equiv !!P
\end{align*}

(Struct Par Comm) \\
(Struct Par Assoc) \\
(Struct Par Zero) \\
(Struct Repl Par) \\
(Struct Repl Zero) \\
(Struct Repl Copy) \\
(Struct Repl Repl)

- These axioms (particularly the ones for !) are sound and complete with respect to equality of spatial trees: edge-labeled finite-depth unordered trees, with infinite-branching but finitely many distinct labels under each node.