Anytime, Anywhere: Modal Logics for Mobile Ambients

Luca Cardelli
Andy Gordon

Microsoft Research

Orientation: Ambients

An ambient is a named, bounded place where computation happens; it is both a unit of mobility and a security perimeter.

A capability represents a right to move into or out of an ambient, or to dissolve its boundary.

Ambient security rests on the controlled distribution of capabilities; the right to enter an ambient does not imply the right to exit it.

We are investigating a process calculus based on ambients, with the goal of developing a flexible, secure, and typeful model for mobile computation.

Mobile Ambients: a packet from a to b

$$\overbrace{a[\underbrace{m[\mathit{out}\ a.\mathit{in}\ b.\langle M\rangle]}_{a \to b : M}]}^{\text{Machine } a} \mid \overbrace{b[\underbrace{\mathit{open}\ m.(x).P}_{\text{receive } x;\ P}]}^{\text{Machine } b}$$

- Ambients may model both machines and packets
- Ambients are mobile: $m[\cdots]$ moves out of a and into b
- You need capability *out* a to exit a; capability *in* b to enter b; and capability *open* m to dissolve m
- There is an ether local to each ambient for message exchange

Ambient Behaviour, By Example

There are four basic reduction rules in the calculus:

$$
\begin{aligned}
& a[m[\mathit{out}\ a.\mathit{in}\ b.\langle M\rangle]] \mid b[\mathit{open}\ m.(x).P] \\
\rightarrow\ & a[] \mid m[\mathit{in}\ b.\langle M\rangle] \mid b[\mathit{open}\ m.(x).P] \\
\rightarrow\ & a[] \mid b[m[\langle M\rangle] \mid \mathit{open}\ m.(x).P] \\
\rightarrow\ & a[] \mid b[\langle M\rangle \mid (x).P] \\
\rightarrow\ & a[] \mid b[P\{x \leftarrow M\}]
\end{aligned}
$$

The Restriction-Free Ambient Calculus

Expressions and Processes:

$M, N ::=$	expressions	$P, Q, R ::=$	processes
n	name	$\mathbf{0}$	inactivity
in M	can enter M	$P \mid Q$	composition
out M	can exit M	$!P$	replication
open M	can open M	$M[P]$	ambient
ϵ	null	$M.P$	action
$M.M'$	path	$(n).P$	input
		$\langle M \rangle$	output

Reduction:

$$n[P \mid \textit{in}\ m.Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R]$$

$$m[n[P \mid \textit{out}\ m.Q] \mid R] \rightarrow n[P \mid Q] \mid m[R]$$

$$n[P] \mid \textit{open}\ n.Q \rightarrow P \mid Q$$

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

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

$$P \rightarrow Q \Rightarrow n[P] \rightarrow n[Q]$$

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

A Vexing Problem

A recurring issue is how to state behavioural properties of ambients.

We have tools for establishing equational properties.

But many properties cannot easily be formulated as equations.

For example, type systems for ambients guarantee certain properties, such as that some ambients are immobile, some are persistent.

It's hard to write down equations for immobility or persistence!

Our solution: invent a modal logic for ambients.

A Modal Logic for Mobile Ambients

Modal Formulas

In a modal logic, the truth of a formula is relative to a state.

In our case, formulas such as the following are true relative to an ambient calculus process:

(1) “there’s an ambient p here”

(2) “somewhere there’s an ambient p”

(3) “if (2) then forever (2)”

(4) “somewhere there’s an ambient p with a child q”

(5) “if (4) then forever (4)”

A Logic of Spacetime

To talk about time, we use standard temporal operators.

To talk about space, we introduce spatial operators:

Process	Formula	
$\mathbf{0}$	$\mathbf{0}$	(there is nothing here)
$n[P]$	$n[\mathcal{A}]$	(there is one thing here)
$P \mid Q$	$\mathcal{A} \mid \mathcal{B}$	(there are two things here)

Syntax of the Logic

Logical Formulas:

η	a name n or a variable x
$\mathcal{A}, \mathcal{B} ::=$	formula
$\quad \mathbf{T}$	$\quad$ true
$\quad \neg\mathcal{A}$	$\quad$ negation
$\quad \mathcal{A} \vee \mathcal{B}$	$\quad$ disjunction
$\quad \mathbf{0}$	$\quad$ inaction
$\quad \eta[\mathcal{A}]$	$\quad$ ambient match
$\quad \mathcal{A} \mid \mathcal{B}$	$\quad$ composition match
$\quad \forall x.\mathcal{A}$	$\quad$ universal quantification
$\quad \Diamond\mathcal{A}$	$\quad$ sometime modality (temporal)
$\quad \diamondsuit\mathcal{A}$	$\quad$ somewhere modality (spatial)
$\quad \mathcal{A}@\eta$	$\quad$ location adjunct
$\quad \mathcal{A} \triangleright \mathcal{B}$	$\quad$ composition adjunct

Semantics of the Logic

Satisfaction: $\mathrm{P} \models \mathcal{A}$ (where $\mathcal{A}$ has no free variables)

$$
\begin{array}{lcl}
\mathrm{P} \models \mathbf{T} & & \\
\mathrm{P} \models \neg\mathcal{A} & \triangleq & \neg(\mathrm{P} \models \mathcal{A}) \\
\mathrm{P} \models \mathcal{A} \vee \mathcal{B} & \triangleq & \mathrm{P} \models \mathcal{A} \vee \mathrm{P} \models \mathcal{B} \\
\mathrm{P} \models \mathbf{0} & \triangleq & \mathrm{P} \equiv \mathbf{0} \\
\mathrm{P} \models \mathrm{n}[\mathcal{A}] & \triangleq & \exists \mathrm{P}'.\mathrm{P} \equiv \mathrm{n}[\mathrm{P}'] \wedge \mathrm{P}' \models \mathcal{A} \\
\mathrm{P} \models \mathcal{A} \mid \mathcal{B} & \triangleq & \exists \mathrm{P}', \mathrm{P}''.\mathrm{P} \equiv \mathrm{P}' \mid \mathrm{P}'' \wedge \mathrm{P}' \models \mathcal{A} \wedge \mathrm{P}'' \models \mathcal{B} \\
\mathrm{P} \models \forall \mathrm{x}.\mathcal{A} & \triangleq & \forall \mathrm{m}.\mathrm{P} \models \mathcal{A}\{\mathrm{x} \leftarrow \mathrm{m}\} \\
\mathrm{P} \models \Diamond\mathcal{A} & \triangleq & \exists \mathrm{P}'.\mathrm{P} \rightarrow^* \mathrm{P}' \wedge \mathrm{P}' \models \mathcal{A} \\
\mathrm{P} \models \diamond\mathcal{A} & \triangleq & \exists \mathrm{P}'.\mathrm{P} \downarrow^* \mathrm{P}' \wedge \mathrm{P}' \models \mathcal{A} \\
\mathrm{P} \models \mathcal{A}@\mathrm{n} & \triangleq & \mathrm{n}[\mathrm{P}] \models \mathcal{A} \\
\mathrm{P} \models \mathcal{A} \triangleright \mathcal{B} & \triangleq & \forall \mathrm{P}'.\mathrm{P}' \models \mathcal{A} \Rightarrow \mathrm{P} \mid \mathrm{P}' \models \mathcal{B}
\end{array}
$$

$\mathrm{P} \downarrow \mathrm{P}'$ iff $\exists \mathrm{n}, \mathrm{P}''.\mathrm{P} \equiv \mathrm{n}[\mathrm{P}'] \mid \mathrm{P}''$ $\quad\quad$ $\downarrow^*$ is the reflexive, transitive closure of $\downarrow$

Congruence Preserves Satisfaction

Structural congruence $\mathrm{P} \equiv \mathrm{Q}$ equates processes we would never wish to tell apart.

For example:

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

The logic is a way of writing down predicates that are automatically preserved by structural congruence:

If $\mathrm{P} \models \mathcal{A}$ and $\mathrm{P} \equiv \mathrm{Q}$ then $\mathrm{Q} \models \mathcal{A}$.

The Characteristic Equivalence

Any modal logic induces an equivalence on states:

Let $\mathrm{P} \bigtriangleup \mathrm{Q} \stackrel{\Delta}{=} \forall \text{closed } \mathcal{A}.\mathrm{P} \models \mathcal{A} \Leftrightarrow \mathrm{Q} \models \mathcal{A}$

If $\mathrm{P} \equiv \mathrm{Q}$ then $\mathrm{P} \bigtriangleup \mathrm{Q}$. On the other hand, $\epsilon[\langle \mathrm{m} \rangle] \bigtriangleup \epsilon[\langle \mathrm{n} \rangle]$ but $\epsilon[\langle \mathrm{m} \rangle] \not\equiv \epsilon[\langle \mathrm{n} \rangle]$. So we have the strict inclusion $\Delta \subset \equiv$.

If we added formulas $\{\mathrm{Q}\}$ such that $\mathrm{P} \models \{\mathrm{Q}\} \Leftrightarrow \mathrm{P} \equiv \mathrm{Q}$, we would have the identity $\Delta = \equiv$.

Finally, the characteristic equivalence is finer than most behavioural equivalences. For instance, it does not satisfy $!\mathrm{n}[] \mid !\mathrm{n}[] \bigtriangleup !\mathrm{n}[]$.

(Consider the formula: *Inf* $\mathrm{n} \mid$ *Inf* n where *Inf* $\mathrm{n} \stackrel{\Delta}{=}$ **T** $\triangleright \Box$*an* n.)

Location

$n[] \models n[\mathbf{T}]$

$n[] \mid \mathbf{0} \models n[\mathbf{T}]$, because $n[] \mid \mathbf{0} \equiv n[]$

$n[m[]] \models n[m[\mathbf{T}]]$

$\neg\mathbf{0} \models n[\mathbf{T}]$

$\neg n[] \models m[\mathbf{T}]$, if $m \neq n$

Composition

$n[] \mid m[] \models m[\mathbf{T}] \mid n[\mathbf{T}]$, because $n[] \mid m[] \equiv m[] \mid n[]$

$n[] \models n[\mathbf{T}] \mid \mathbf{T}$, because $n[] \equiv n[] \mid \mathbf{0}$

$!n[] \models n[\mathbf{T}] \mid \mathbf{T}$, because $!n[] \equiv n[] \mid !n[]$

Inaction

$\mathrm{m}[] \mid \mathrm{n}[] \models \neg\mathbf{0} \mid \neg\mathbf{0}$

$\mathrm{n}[] \models \neg(\neg\mathbf{0} \mid \neg\mathbf{0})$ (read: $\mathrm{n}[]$ is single-threaded)

Spatial Modality

$\mathrm{n}[\mathrm{m}[]] \models \diamond\mathrm{m}[\mathbf{T}]$

$\neg\mathrm{n}[\mathrm{m}[] \mid \mathrm{m}[]] \models \diamond\mathrm{m}[\mathbf{T}]$

$\mathrm{m}[\mathrm{n}[\mathrm{p}[\mathrm{q}[]]]] \models \diamond(\mathrm{p}\ \textit{parents}\ \mathrm{q})$,

where $\mathrm{p}\ \textit{parents}\ \mathrm{q} \stackrel{\Delta}{=} \mathrm{p}[\mathrm{q}[\mathbf{T}] \mid \mathbf{T}] \mid \mathbf{T}$

Temporal Modality

$\mathrm{n}[\mathrm{m}[]] \mid \mathit{open}\,\mathrm{n} \models \Diamond \mathrm{m}[\mathbf{T}]$

$\mathrm{n}[\mathrm{n}[]] \mid \mathit{open}\,\mathrm{n} \models \Box \mathit{an}\,\mathrm{n}$, where $\mathit{an}\,\mathrm{n} \stackrel{\Delta}{=} \mathrm{n}[\mathbf{T}] \mid \mathbf{T}$

Location Adjunct

$\mathrm{n}[] \models \mathrm{m}[\mathrm{n}[\mathbf{T}]]@\mathrm{m}$

$\mathrm{n}[\mathit{out}\,\mathrm{m}] \models (\Diamond \mathit{an}\,\mathrm{n})@\mathrm{m}$

Composition Adjunct

$\mathrm{n}[] \models \mathrm{m}[\mathbf{T}] \triangleright (\mathrm{n}[\mathbf{T}] \mid \mathrm{m}[\mathbf{T}])$

$\mathit{open}\,\mathrm{n}.\mathrm{m}[] \models (\Box \mathrm{n}[\mathbf{T}]) \triangleright (\Diamond \mathrm{m}[\mathbf{T}])$

Derived Connectives:

$\mathbf{F}$	$\triangleq \neg\mathbf{T}$	false
$\mathcal{A} \wedge \mathcal{B}$	$\triangleq \neg(\neg\mathcal{A} \vee \neg\mathcal{B})$	conjunction
$\mathcal{A} \Rightarrow \mathcal{B}$	$\triangleq \neg\mathcal{A} \vee \mathcal{B}$	implication
$\mathcal{A} \parallel \mathcal{B}$	$\triangleq \neg(\neg\mathcal{A} \mid \neg\mathcal{B})$	decomposition
$\mathcal{A}^*$	$\triangleq \mathcal{A} \parallel \mathbf{F}$	every component satisfies $\mathcal{A}$
$\mathcal{A}^+$	$\triangleq \mathcal{A} \mid \mathbf{T}$	some component satisfies $\mathcal{A}$
$\exists x.\mathcal{A}$	$\triangleq \neg\forall x.\neg\mathcal{A}$	existential quantification
$\Box\mathcal{A}$	$\triangleq \neg\Diamond\neg\mathcal{A}$	everytime modality (temporal)
$\blacksquare\mathcal{A}$	$\triangleq \neg\diamond\neg\mathcal{A}$	everywhere modality (spatial)
$\mathcal{A}^{\mathbf{F}}$	$\triangleq \mathcal{A} \triangleright \mathbf{F}$	$\mathcal{A}$ is unsatisfiable
$\mathcal{A} \ltimes \mathcal{B}$	$\triangleq \neg(\mathcal{A} \triangleright \neg\mathcal{B})$	some new $\mathcal{A}$ component makes $\mathcal{B}$ hold.

Semantic Reasoning:

A Decidable Sublogic

A Model-Checking Algorithm

We consider only replication-free processes, and $\triangleright$-free formulas.

For replication-free processes P, and $\triangleright$-free formulas $\mathcal{A}$,
the truth of $\mathrm{P} \models \mathcal{A}$ is decidable.

Propositional logic is easy:

Checking propositional logic:

$$\mathit{Check}(\mathrm{P}, \mathbf{T}) = \mathbf{T}$$
$$\mathit{Check}(\mathrm{P}, \neg\mathcal{A}) = \neg\mathit{Check}(\mathrm{P}, \mathcal{A})$$
$$\mathit{Check}(\mathrm{P}, \mathcal{A} \vee \mathcal{B}) = \mathit{Check}(\mathrm{P}, \mathcal{A}) \vee \mathit{Check}(\mathrm{P}, \mathcal{B})$$

Normal Forms

Let a **prime process**, π, be one of

$$M[P],\ H.P,\ (x).P,\ \text{or}\ \langle M \rangle$$

for $H \in \{\mathit{in}\ N, \mathit{out}\ N, \mathit{open}\ N, n\}$.

Let a **normal form** be a process of the form

$$\pi_1 \mid \cdots \mid \pi_k \mid \mathbf{0}$$

with $k \geq 0$, written as $\prod_{i \in 1..k} \pi_i$.

There is a terminating procedure *Norm* such that for all replication-free P, if $\mathit{Norm}(P) = [\pi_1, \ldots, \pi_k]$ then $P \equiv \prod_{i \in 1..k} \pi_i$.

Checking inaction:

$$\mathit{Check}(\mathrm{P}, \mathbf{0}) = \begin{cases} \mathbf{T} & \text{if } \mathit{Norm}(\mathrm{P}) = [] \\ \mathbf{F} & \text{otherwise} \end{cases}$$

We rely on the fact that $\prod_{i \in 1..k} \pi_i \equiv \mathbf{0}$ iff $k = 0$.

Checking ambient matching and its adjunct:

$$\mathit{Check}(P, n[\mathcal{A}]) = \begin{cases} \mathit{Check}(Q, \mathcal{A}) & \text{if } \mathit{Norm}(P) = [n[Q]] \text{ for some } Q \\ \mathbf{F} & \text{otherwise} \end{cases}$$

$$\mathit{Check}(P, \mathcal{A}@n) = \mathit{Check}(n[P], \mathcal{A})$$

We rely on the fact that $\prod_{i \in 1..k} \pi_i \equiv n[P]$ iff $k = 1$ and $\exists Q$ such that $\pi_1 = n[Q]$ and $P \equiv Q$.

Checking composition matching:

$$\mathit{Check}(P, \mathcal{A} \mid \mathcal{B}) = \text{let } [\pi_1, \ldots, \pi_k] = \mathit{Norm}(P) \text{ in}$$

$$\begin{cases} \mathbf{T} & \text{if } \exists I, J.I \cup J = 1..k \wedge I \cap J = \emptyset \wedge \\ & \qquad \mathit{Check}(\prod_{i \in I} \pi_i, \mathcal{A}) \wedge \\ & \qquad \mathit{Check}(\prod_{j \in J} \pi_j, \mathcal{B}) \\ \mathbf{F} & \text{otherwise} \end{cases}$$

We rely on the fact that $\prod_{i \in 1..k} \pi_i \equiv P \mid Q$ iff $\exists I, J$ such that $I \cap J = \emptyset$, $I \cup J = 1..k$, $\prod_{i \in I} \pi_i \equiv P$, and $\prod_{j \in J} \pi_j \equiv Q$.

Checking universal quantification:

$$\mathit{Check}(P, \forall x.\mathcal{A}) = \text{let } \{m_1, \ldots, m_k\} = \mathit{fn}(P) \cup \mathit{fn}(\mathcal{A}) \text{ in}$$

$$\text{let } m_0 \notin \{m_1, \ldots, m_k\} \text{ be some fresh name in}$$

$$\begin{cases} \mathbf{T} & \text{if } \mathit{Check}(P, \mathcal{A}\{x \leftarrow m_i\}) \text{ for all } i \in 0..k \\ \mathbf{F} & \text{otherwise} \end{cases}$$

We rely on the fact that if $\{m_1, \ldots, m_k\} = \mathit{fn}(P) \cup \mathit{fn}(\mathcal{A})$, and $m_0 \notin \{m_1, \ldots, m_k\}$, then

$$P \models \forall x.\mathcal{A} \Leftrightarrow \forall i \in 0..k.P \models \mathcal{A}\{x \leftarrow m_i\}$$

Checking the temporal modality:

$$\mathit{Check}(\mathrm{P}, \Diamond\mathcal{A}) = \text{let } [\mathrm{P}_1, \ldots, \mathrm{P}_k] = \mathit{Reachable}(\mathrm{P}) \text{ in } \begin{cases} \mathbf{T} & \text{if } \mathit{Check}(\mathrm{P}_i, \mathcal{A}) \text{ for some } i \in 1..k \\ \mathbf{F} & \text{otherwise} \end{cases}$$

We rely on a terminating procedure *Reachable* such that if $[\mathrm{P}_1, \ldots, \mathrm{P}_k] = \mathit{Reachable}(\mathrm{P})$.

(1) For all $i \in 1..k$, $\mathrm{P} \rightarrow^* \mathrm{P}_i$.

(2) If $\mathrm{P} \rightarrow^* \mathrm{Q}$ then $\mathrm{Q} \equiv \mathrm{P}_i$ for some $i \in 1..k$.

Problem: cannot define *Reachable* directly from $\mathrm{P} \rightarrow \mathrm{Q}$.

Solution: use a labelled transition system.

Checking the spatial modality:

$$\mathit{Check}(P, \diamondsuit\mathcal{A}) = \text{let } [P_1, \ldots, P_k] = \mathit{SubLocations}(P) \text{ in}$$

$$\begin{cases} \mathbf{T} & \text{if } \mathit{Check}(P_i, \mathcal{A}) \text{ for some } i \in 1..k \\ \mathbf{F} & \text{otherwise} \end{cases}$$

We rely on a terminating procedure *SubLocations* such that if $[P_1, \ldots, P_k] = \mathit{SubLocations}(P)$.

(1) For all $i \in 1..k$, $P \downarrow^* P_i$.

(2) If $P \downarrow^* Q$ then $Q \equiv P_i$ for some $i \in 1..k$.

Problem: cannot define *Sublocations* directly from $P \downarrow Q$.

Solution: go via normal forms.

Syntactic Reasoning:

A Sequent Calculus

Validity

For closed formulas, validity is given by:

Validity and Satisfiability:

vld $\mathcal{A}$	$\stackrel{\Delta}{=} \forall \mathrm{P}.\mathrm{P} \models \mathcal{A}$	$\mathcal{A}$ (closed) is valid
sat $\mathcal{A}$	$\stackrel{\Delta}{=} \exists \mathrm{P}.\mathrm{P} \models \mathcal{A}$	$\mathcal{A}$ (closed) is satisfiable

We have a sequent calculus for deriving the validity of formulas, based on the following definitions:

Sequents and Inference Rules:

$\mathcal{A} \vdash \mathcal{B}$	$\stackrel{\Delta}{=}$ **vld**$(\mathcal{A} \Rightarrow \mathcal{B})$	sequent
$\mathcal{A}_1 \vdash \mathcal{B}_1 ; \cdots ; \mathcal{A}_k \vdash \mathcal{B}_k \} \mathcal{A} \vdash \mathcal{B}$		
	$\stackrel{\Delta}{=} \mathcal{A}_1 \vdash \mathcal{B}_1 \wedge \cdots \wedge \mathcal{A}_k \vdash \mathcal{B}_k \Rightarrow \mathcal{A} \vdash \mathcal{B}$	inference rule

Highlights of our Sequent Calculus

- Propositional logic
- Predicate logic
- Location logic
- S4 (but not S5) axioms for time and space modalities
- Linear logic

Location Logic

Monoid laws:

($|$ **0**) $\quad \} \mathcal{A} \mid \mathbf{0} \dashv\vdash \mathcal{A}$

(X $|$) $\quad \} \mathcal{A} \mid \mathcal{B} \dashv\vdash \mathcal{B} \mid \mathcal{A}$

(A $|$) $\quad \} \mathcal{A} \mid (\mathcal{B} \mid \mathcal{C}) \dashv\vdash (\mathcal{A} \mid \mathcal{B}) \mid \mathcal{C}$

Adjunctions:

($\mathrm{n}[]$ @) $\quad \mathrm{n}[\mathcal{A}] \vdash \mathcal{B} \quad \{ \} \mathcal{A} \vdash \mathrm{B}@\mathrm{n}$

($|$ $\triangleright$) $\quad \mathcal{A} \mid \mathcal{C} \vdash \mathcal{B} \quad \{ \} \mathcal{A} \vdash \mathcal{C} \triangleright \mathcal{B}$

Time and Space Modalities

They are S4 modalities:

($\Diamond$)	$\wr\ \Diamond\mathcal{A} \dashv\vdash \neg\Box\neg\mathcal{A}$	($\diamondsuit$)	$\wr\ \diamondsuit\mathcal{A} \dashv\vdash \neg\blacksquare\neg\mathcal{A}$
($\Box$ K)	$\Box(\mathcal{A} \Rightarrow \mathcal{B}) \vdash \Box\mathcal{A} \Rightarrow \Box\mathcal{B}$	($\blacksquare$ K)	$\blacksquare(\mathcal{A} \Rightarrow \mathcal{B}) \vdash \blacksquare\mathcal{A} \Rightarrow \blacksquare\mathcal{B}$
($\Box$ T)	$\Box\mathcal{A} \vdash \mathcal{A}$	($\blacksquare$ T)	$\blacksquare\mathcal{A} \vdash \mathcal{A}$
($\Box$ 4)	$\Box\mathcal{A} \vdash \Box\Box\mathcal{A}$	($\blacksquare$ 4)	$\blacksquare\mathcal{A} \vdash \blacksquare\blacksquare\mathcal{A}$
($\Box\vdash$)	$\mathcal{A} \vdash \mathcal{B} \wr\ \Box\mathcal{A} \vdash \Box\mathcal{B}$	($\blacksquare\vdash$)	$\mathcal{A} \vdash \mathcal{B} \wr\ \blacksquare\mathcal{A} \vdash \blacksquare\mathcal{B}$

But they are not S5:

$$\neg\mathbf{vld}\ \Diamond\mathcal{A} \Rightarrow \Box\Diamond\mathcal{A} \qquad\qquad \neg\mathbf{vld}\ \diamondsuit\mathcal{A} \Rightarrow \blacksquare\diamondsuit\mathcal{A}$$

Time and space are not quite symmetric:

($\Diamond\ \mathfrak{n}[]$)	$\wr\ \mathfrak{n}[\Diamond\mathcal{A}] \vdash \Diamond\mathfrak{n}[\mathcal{A}]$	($\diamondsuit\ \mathfrak{n}[]$)	$\wr\ \mathfrak{n}[\diamondsuit\mathcal{A}] \vdash \diamondsuit\mathcal{A}$
($\Diamond\ \vert$)	$\wr\ \Diamond\mathcal{A} \mid \Diamond\mathcal{B} \vdash \Diamond(\mathcal{A} \mid \mathcal{B})$	($\diamondsuit\ \vert$)	$\wr\ \diamondsuit\mathcal{A} \mid \mathcal{B} \vdash \diamondsuit(\mathcal{A} \mid \mathbf{T})$

Connections with Linear Logic

Like many logics, ours can interpret intuitionistic linear logic:

$$\mathbf{1}_{\mathsf{IL}} \triangleq \mathbf{0} \qquad \mathcal{A} \oplus \mathcal{B} \triangleq \mathcal{A} \vee \mathcal{B} \qquad !\mathcal{A} \triangleq \mathbf{0} \wedge (\neg(\mathbf{0} \Rightarrow \mathcal{A})) \triangleright \mathbf{F}$$

$$\bot_{\mathsf{IL}} \triangleq \mathbf{F} \qquad \mathcal{A} \,\&\, \mathcal{B} \triangleq \mathcal{A} \wedge \mathcal{B} \qquad \text{(so that } \mathrm{P} \models\!! \mathcal{A} \Leftrightarrow$$

$$\top_{\mathsf{IL}} \triangleq \mathbf{T} \qquad \mathcal{A} \otimes \mathcal{B} \triangleq \mathcal{A} \mid \mathcal{B} \qquad \mathrm{P} \equiv \mathbf{0} \wedge \mathbf{0} \models \mathcal{A})$$

$$\mathbf{0}_{\mathsf{IL}} \triangleq \mathbf{F} \qquad \mathcal{A} \multimap \mathcal{B} \triangleq \mathcal{A} \triangleright \mathcal{B}$$

These definitions correspond to the standard quantale model of IL.

$\mathcal{A}_1, \ldots, \mathcal{A}_n \vdash_{\mathsf{IL}} \mathcal{B}$ is valid in IL iff $\mathbf{vld}(\mathcal{A}_1 \mid \cdots \mid \mathcal{A}_n \vdash \mathcal{B})$.

If $\mathcal{A}_1, \ldots, \mathcal{A}_n \vdash_{\mathsf{IL}} \mathcal{B}$ then $\mathcal{A}_1 \mid \cdots \mid \mathcal{A}_n \vdash \mathcal{B}$.

So some, but not all, of our spatial operators are already familiar!

Application:
Logical Properties of Type Systems

Application: Expressing Persistence

In work with Ghelli, we invented a type system that statically tracks whether each ambient is locked or not.

The logic solves the problem of stating properties of well-typed processes independently of the typing rules.

If P is well-typed given that n is a name for a locked ambient, then

$$\mathrm{P} \models \Box(\diamond \mathit{an}\, \mathrm{n} \Rightarrow \Box \diamond \mathit{an}\, \mathrm{n})$$

(Unfortunately, the current logic cannot express that the number of ambients named n never decreases, nor can it talk about a unique ambient named n.)

Application: Expressing Immobility

Another type system additionally tracks whether an ambient is immobile or not.

If P is well-typed given that p is a name for a locked ambient, and q is a name for a locked, immobile ambient, then

$$\mathrm{P} \models \Box(\diamondsuit(\mathrm{p}\ \mathit{parents}\ \mathrm{q}) \Rightarrow \Box\diamondsuit(\mathrm{p}\ \mathit{parents}\ \mathrm{q}))$$

where $\mathrm{p}\ \mathit{parents}\ \mathrm{q} \stackrel{\Delta}{=} \mathrm{p}[\mathrm{q}[\mathbf{T}] \mid \mathbf{T}] \mid \mathbf{T}$.

Related Work

A plethora of modal logics for process calculi was inaugurated by Hennessy-Milner logic for CCS.

We have not found much closely related work on spatial properties of processes.

Operators akin to $\mathcal{A} \mid \mathcal{B}$ can be found in the systems of Holmström (1988) and Damm (1989), and in linear logic.

There is a long logical tradition of $\mathcal{A} \wedge \mathcal{B}$ specifying a system assembled from components satisfying $\mathcal{A}$ and $\mathcal{B}$.

Future Work

Recursive formulas would allow us to say more things, such as: "there is a unique ambient named n":

$$\begin{aligned}
\mathit{an}\, n &\triangleq n[\mathbf{T}] \mid \mathbf{T} \\
\mathit{no}\, n &\triangleq \neg \diamondsuit \mathit{an}\, n \\
\eta = \mu &\triangleq \eta[\mathbf{T}]@\mu \\
\mathit{unique}\, n &\triangleq \mu X.\mathit{no}\, n \mid (n[\mathit{no}\, n] \vee \exists x.x[X] \wedge \neg(x = n))
\end{aligned}$$

Other tasks: restore restriction, model check processes with finite-state recursion.

Contributions

The novel aspects of our logic lie in its treatment of space:

- New logical primitives
- New rules of inference
- New model checking techniques

The first application is expressing properties of type systems.