

Anytime, Anywhere

Logics of Time and Space

Luca Cardelli
Andy Gordon

Microsoft Research

MSRC, July 23, 1999

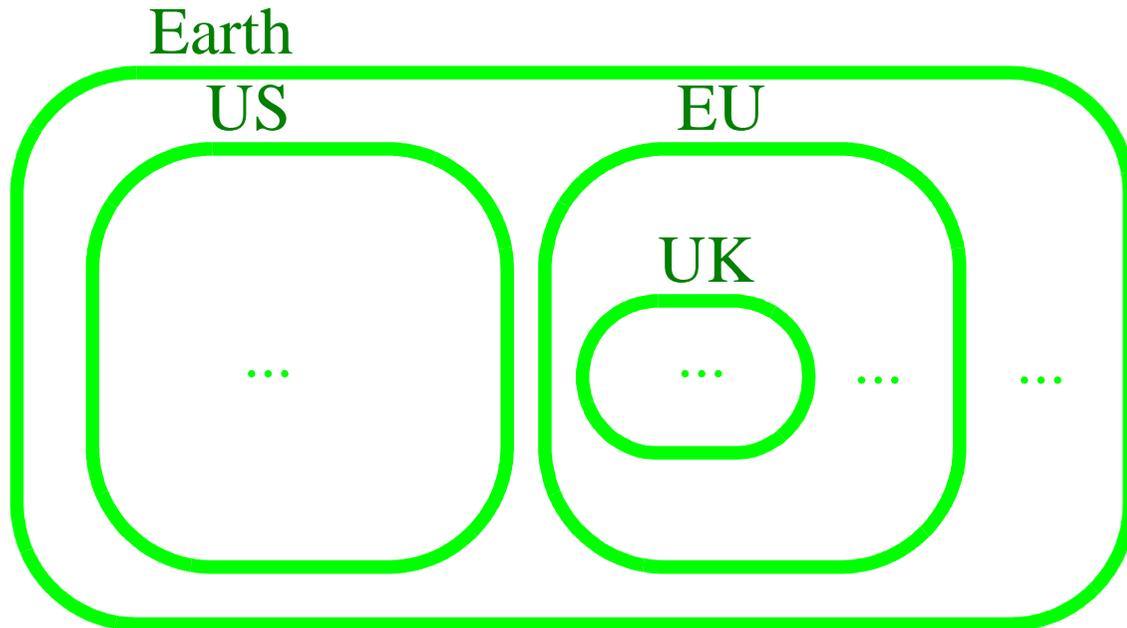
Introduction

- We have been looking for ways to express properties of mobile computations, E.g.:
 - "Here today, gone tomorrow."
 - "Eventually the agent crosses the firewall."
 - "Every agent carries a suitcase."
 - "Somewhere there is a virus."
 - "There is always at most one ambient called n here."
- Approach: devise a logic that can talk about *space* as well as time.

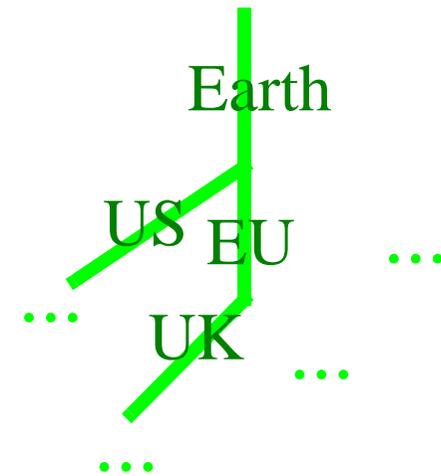
Representations of "Space"

- Postulate: space is tree-structured.

Geographical maps



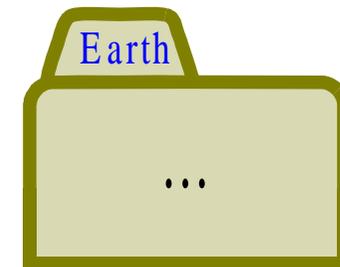
Edge-labeled trees



Spatial expressions

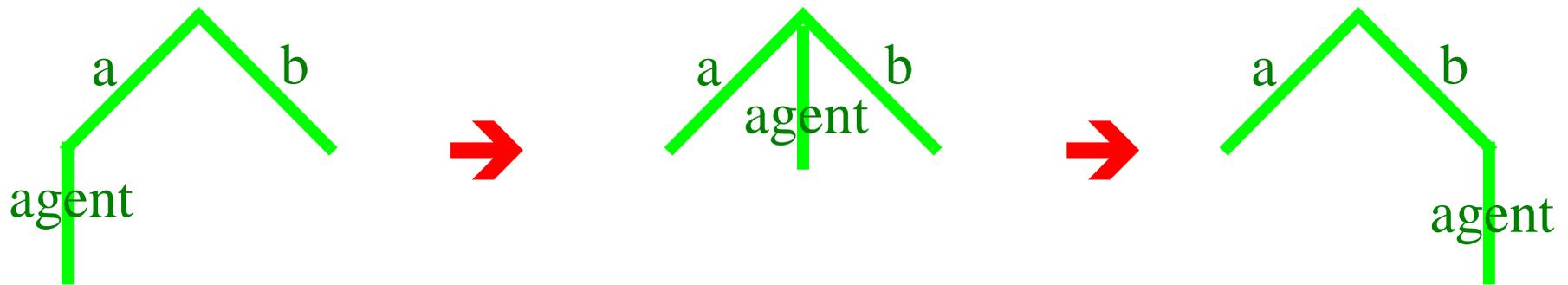
Earth[US[...] | EU[UK[...] | ...] ...]

Folders



Mobility

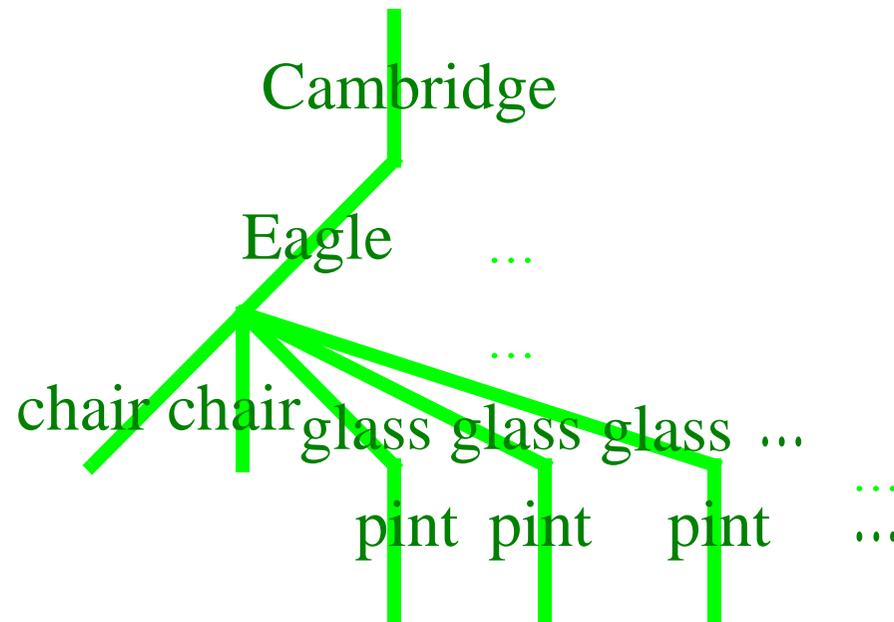
- Then, *mobility* is change of spatial structures over time:



$a[\text{agent}[]] \mid b[] \rightarrow a[] \mid \text{agent}[] \mid b[] \rightarrow a[] \mid b[\text{agent}[]]$

Spatial Trees

- Our basic model of space is going to be *finite-depth edge-labeled unordered trees*; for short: *spatial trees*.
- One subtlety: unbounded resources are represented by infinite branching:



Spatial Expressions

- We use *spatial expressions* to describe spatial trees. These are nested expressions with ! for unbounded replication.

Cambridge[Eagle[chair[] | chair[] | !glass[pint[]]] | ...]

Cambridge[!ParkingSpace[] | ...] (not!)

- Two spatial expressions are equivalent when they describe the same spatial tree.

– Ex.:

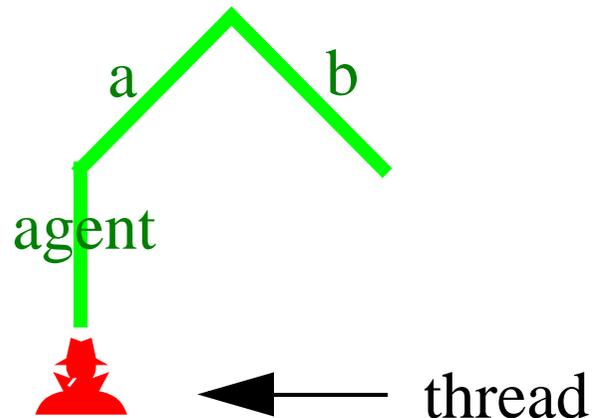
$$a[] | b[] \equiv b[] | a[]$$

$$a[] | !a[] \equiv !a[]$$

- This is not totally trivial (because of !), but we have a complete axiomatization of such equivalence.

Ambient Expressions

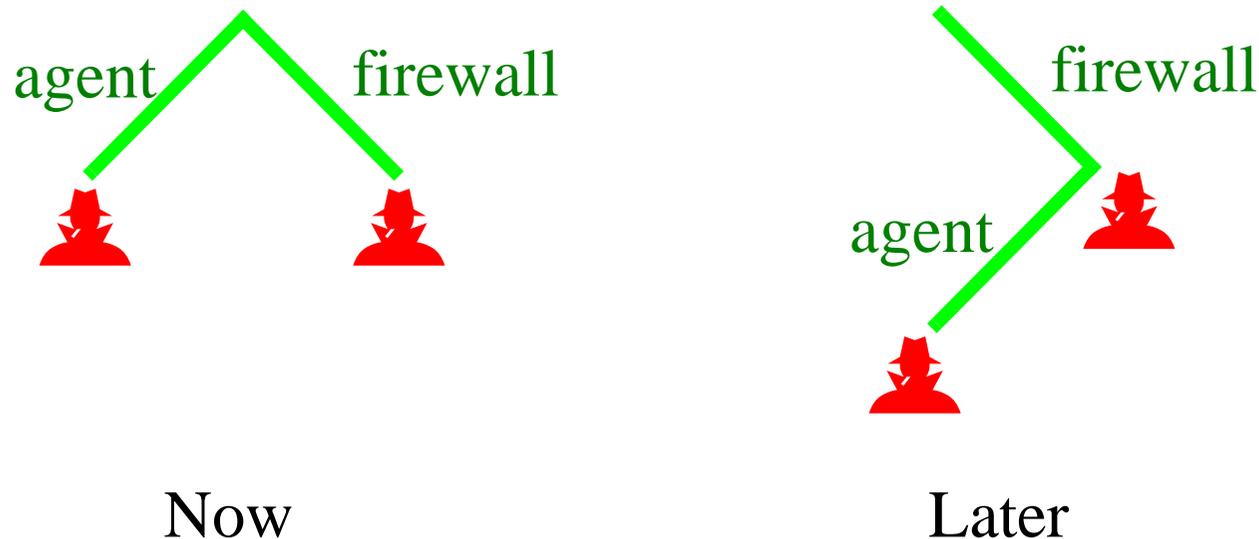
- Spatial expressions/trees are a subset of *ambient expressions/trees*, where we can represent not only the spatial aspects, but also the dynamic aspects of mobile computation.



- We will not get into the details of full ambient expressions in this talk. Just remember:
 - An ambient tree is a spatial tree with, possibly, threads at each node that can locally change the shape of the tree.

Properties of Mobile Computation

- These often have the form:
 - *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.



Modal Logics

- In standard logic, assertions are either **true** or **false**.
- In a *modal* logic, the truth of an assertion is relative to a *state*.
 - In epistemic logic: a *knowledge state*.
 - In temporal logic: an *execution state*.
 - In our logic: a *space-time state*, relative to the *current place* and the *current time*.
- Here is a **formula** talking about a tree (not a **tree** itself):

Cambridge[Eagle[chair[] | ..] | ..]

Right now in Cambridge there is a pub called the Eagle, and inside the Eagle there is at least an empty chair.

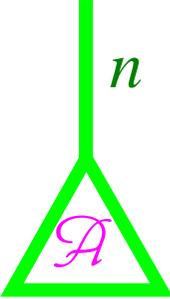
This may be true or false depending on the time of day (happy hour?) and location (Cambridge England or Mass.?).

Basic Modalities

- $\mathbf{0}$: *here now* there is absolutely nothing ($n[]$ abbreviates $n[\mathbf{0}]$):

$\mathbf{0}$ satisfied by *(void)* i.e. by $\mathbf{0}$

- $n[\mathcal{A}]$: *here now* there is exactly one place called n , whose contents satisfy (*there now*) the formula \mathcal{A} :

$n[\mathcal{A}]$ satisfied by  i.e. by $n[P]$
if P sat. \mathcal{A}

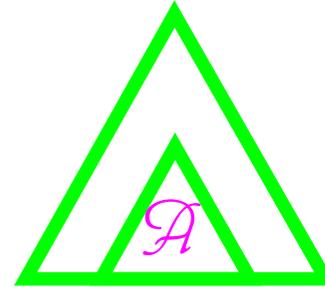
- $\mathcal{A} | \mathcal{B}$: *here now* there are exactly two things next to each other, one satisfying (*there now*) \mathcal{A} and one satisfying (*there now*) \mathcal{B} :

$\mathcal{A} | \mathcal{B}$ satisfied by  i.e. by $P | Q$
if P sat. \mathcal{A}
and Q sat. \mathcal{B}

-
- $\spadesuit A$: *somewhere now*, there is a place satisfying (*there now*) A :

$\spadesuit A$

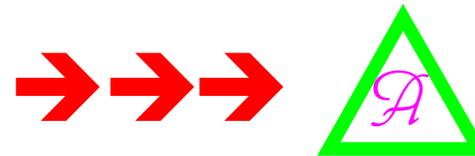
satisfied by



- $\diamond A$: *here sometime*, there is a thing satisfying (*here then*) A :

$\diamond A$

satisfied by



Derived Modalities

- *Everywhere* \mathcal{A} :

$$\Box \mathcal{A} \triangleq \neg \Diamond \neg \mathcal{A}$$

What is true *everywhere*? Not much, unless qualified:

$$\Box (\mathcal{A} \Rightarrow \mathcal{B})$$

everywhere \mathcal{A} is true, \mathcal{B} is true as well

- *Always* \mathcal{A} :

$$\Box \mathcal{A} \triangleq \neg \Diamond \neg \mathcal{A}$$

What will *always* be there?

$$\Box \text{Pisa}[\text{LeaningTower}[\dots] \mid \dots]$$

Other Logical Connectives

- Anything (including void)

T (Anything satisfies it.) A.k.a.: ..

- Normal implication

$$\mathcal{A} \Rightarrow \mathcal{B}$$

if \mathcal{A} is true here now, then \mathcal{B} is true here now

$$\text{Borders}[..] \Rightarrow \text{Borders}[\text{Starbucks}[..] | ..]$$

If there is a Borders bookstore, there is a Starbucks inside.

$$(\text{NonSmoker}[..] | ..) \Rightarrow (\text{NonSmoker}[..] | \text{Smoker}[..] | ..)$$

If there is a non-smoker, there is nearby a smoker.

Spatial Implications

- Parallel implication

$$\mathcal{A} \mid \Rightarrow \mathcal{B} \triangleq \neg(\mathcal{A} \mid \neg \mathcal{B})$$

It is not possible to split the current location in such a way that one part satisfies \mathcal{A} and the other does not satisfy \mathcal{B} .

In other words, every way we split the current location, if one part satisfies \mathcal{A} , then the other part must satisfy \mathcal{B} .

$$\square \text{ Bath} [\spadesuit (\text{NonSmoker}[\dots] \mid \Rightarrow \text{Smoker}[\dots] \mid \dots)]$$

It is always the case that at the Bath, anywhere there is a non-smoker there is, nearby, a smoker.

-
- Nested implication

$$n[\Rightarrow \mathcal{A}] \triangleq \neg n[\neg \mathcal{A}]$$

It is not possible that the contents of an n location do not satisfy \mathcal{A} .

In other words, if there is an n location, its contents satisfy \mathcal{A} .

$$\square \text{US}[\forall \text{Borders}[\Rightarrow \text{Starbucks}[\dots] \mid \dots]]$$

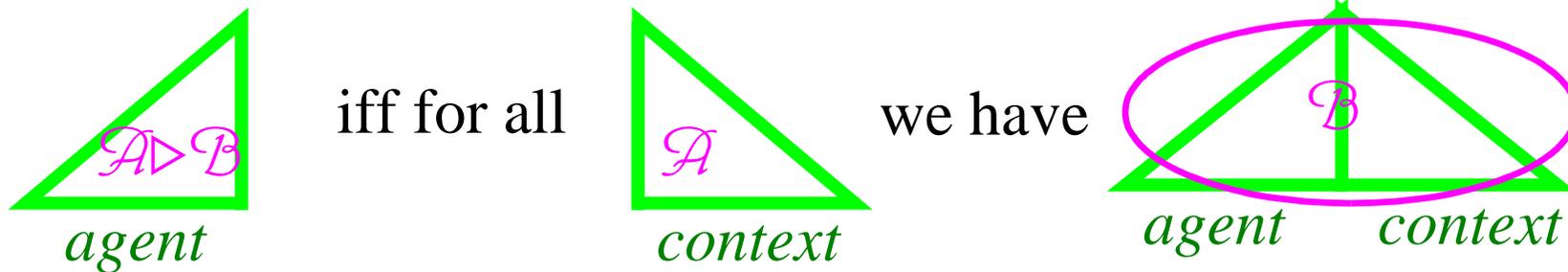
Everywhere in the US, there is always a Starbucks inside a Borders.

Context/System Specs

- Pre/post-condition specs for sequential programs:
 - If the context satisfies \mathcal{A} *before* the execution of the program, the system satisfies \mathcal{B} *after* the execution of the program.
- Context/system specs for concurrent programs:
 - If the context satisfies \mathcal{A} *on its own*, the program running *in parallel* with the context satisfies \mathcal{B} . (program + context = system)
- Context/system specs for mobile agents:
 - (Parallel context.) When the agent is *near* something satisfying \mathcal{A} ; the system satisfies \mathcal{B} ,
 - (Nested context.) When the agent is *inside* a location n ; the system satisfies \mathcal{C} .
 - Mixed parallel/nested contexts: combine the above.

Security Connectives

- $A \triangleright B$: even when the agent is in presence of any context (e.g.: "attacker") bound to satisfy A , the system satisfies B .



- Example (from two logically contradictory points of view):

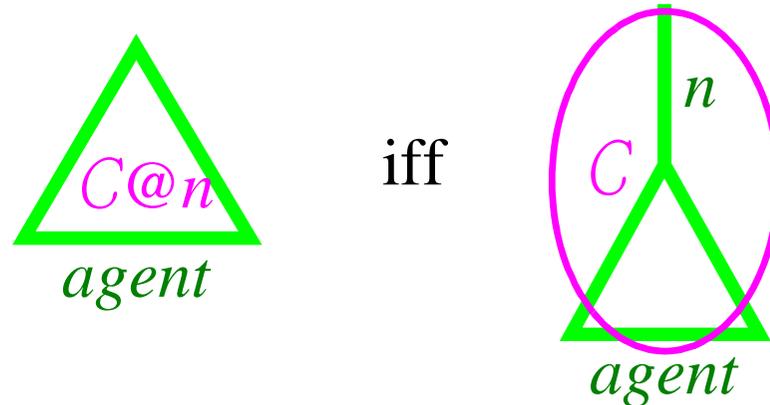
bait[...] sat. fish[..] \triangleright \diamond fish[bait[..] | ..]

fish[...] sat. bait[..] \triangleright \square (fish[..] | bait[..])

Bait wants to catch fish. Fish wants to avoid bait.

- A logical duality: $(A | B) \Rightarrow C$ iff $A \Rightarrow (B \triangleright C)$

- $C@n$: even when the agent is ("thrown") in a location n , the system satisfies C .



- Example: one would hope that $\text{fish}[\dots]$ satisfies:

$(\Box \text{tank}[\text{fish}[\dots] \mid \dots]) @ \text{tank}$

A fish will survive in a tank.

- A logical duality: $n[\mathcal{A}] \Rightarrow C$ iff $\mathcal{A} \Rightarrow C@n$.

Thief!

A *shopper* is likely to pull out a wallet. A *thief* is likely to grab it.

Shopper \triangleq

Person[Wallet[£] | ..] \wedge

\diamond (Person[NoWallet] | Wallet[£])

NoWallet \triangleq \neg (Wallet[£] | ..)

Thief \triangleq Wallet[£] \triangleright \diamond *NoWallet*

By simple logical deductions involving the laws of \triangleright and \diamond :

Shopper | *Thief* \Rightarrow

(Person[Wallet[£] | ..] | *Thief*) \wedge

\diamond (Person[NoWallet] | *NoWallet*)

Logical Formulas

$\mathcal{A}, \mathcal{B} : \Phi ::=$

\mathbf{T}	true
$\neg \mathcal{A}$	negation
$\mathcal{A} \vee \mathcal{B}$	disjunction
$\mathbf{0}$	void
$\eta[\mathcal{A}]$	location
$\mathcal{A} \mathcal{B}$	composition
$\diamond \mathcal{A}$	sometime modality (temporal)
$\diamondsuit \mathcal{A}$	somewhere modality (spatial)
$\mathcal{A} @ \eta$	location adjunct
$\mathcal{A} \triangleright \mathcal{B}$	composition adjunct
$\forall x. \mathcal{A}$	universal quantification over names

where η is a name n or a (quantifiable) variable x .

Conservation of Space

- Space cannot be instantaneously destroyed:

$$n[] \Rightarrow 0$$

This is not valid (no tree can satisfy the lhs and rhs at once).

- Space cannot be instantaneously created:

$$0 \Rightarrow n[]$$

This is not valid either.

- Technically, we have a *logical linearity* property: the same "amount of space" must be found on lhs and rhs of a simple implication.

Applications

- Model checking of spatial formulas
 - We have an algorithm for deciding the satisfaction relation for !-free processes and \triangleright -free formulas.
- Screening of mobile applets
 - By typechecking
 - By modelchecking
 - By proofchecking
- Spatial databases?

Connection: Semi-Structured Data

- Advanced research on databases is now focusing on *semi-structured data*, which, by a total coincidence, are edge-labeled trees or graphs. (E.g.: XML trees.)
- So, we have an unexpected connection:
 - With slight modifications, our spatial logic can be seen as a *query-language* for semi-structured data.
 - The ambient calculus can be seen as a *computational model* over semi-structured data. (E.g. for database updates.)
 - Type systems for the ambient calculus can be seen as *weak schemas* for semi-structured data (which are largely unexplored).
 - We have just begun discussing the connections with Giorgio Ghelli.

Conclusions

- The novel aspects of our logic lie in its treatment of *space* (spatial structures) and of the evolution of space over time (*mobility*).
- Security connectives emerge as natural adjuncts of spatial connectives.
- These ideas can be applied to any language that embodies a distinction between geometric and dynamic operators.
- The logic is based on strong computational intuitions. From a purely logical point of view, it has unusual formal properties.