

Introduction to modal logic

José Proença

HASLab - INESC TEC
Universidade do Minho
Braga, Portugal

April, 2016

A logic

A language

i.e. a collection of well-formed expressions to which meaning can be assigned.

A semantics

describing how language expressions are interpreted as statements about something.

A deductive system

i.e. a collection of rules to derive in a purely syntactic way facts and relationships among semantic objects described in the language.

Note

- a purely syntactic approach (up to the 1940's; the **sacred form**)
- a model theoretic approach (A. Tarski legacy)

Semantic reasoning: models

- sentences
- models & satisfaction: $\mathfrak{M} \models \phi$
- validity: $\models \phi$ (ϕ is satisfied in every possible structure)
- logical consequence: $\Phi \models \phi$ (ϕ is satisfied in every model of Φ)
- theory: $Th \Phi$ (set of logical consequences of a set of sentences Φ)

Syntactic reasoning: deductive systems

Deductive systems

- sequents
 - Hilbert systems
 - natural deduction
 - tableaux systems
 - resolution
 - ...
-
- derivation and proof
 - deductive consequence: $\Phi \vdash \phi$
 - theorem: $\vdash \phi$

Soundness & completeness

- A deductive system \vdash is **sound** wrt a semantics \models if for all sentences ϕ

$$\vdash \phi \implies \models \phi$$

(every theorem is valid)

- ... **complete** ...

$$\models \phi \implies \vdash \phi$$

(every valid sentence is a theorem)

Consistency & refutability

For logics with **negation** and a **conjunction** operator

- A sentence ϕ is **refutable** if $\neg\phi$ is a theorem (i.e. $\vdash \neg\phi$)
- A set of sentences Φ is **refutable** if some finite conjunction of elements in Φ is refutable
- ϕ or Φ is **consistent** if it is not refutable.

Examples

$$\mathfrak{M} \models \phi$$

- Propositional logic (logic of **uninterpreted assertions**; models are **truth assignments**)
- Equational logic (formalises **equational** reasoning; models are **algebras**)
- First-order logic (logic of **predicates** and **quantification** over structures; models are **relational structures**)
- Modal logics
- ...

Modal logic (from P. Blackburn, 2007)

*Over the years modal logic has been applied in many different ways. It has been used as a tool for reasoning about **time**, **beliefs**, **computational systems**, **necessity** and **possibility**, and much else besides.*

*These applications, though diverse, have something important in common: the key ideas they employ (flows of time, relations between epistemic alternatives, transitions between computational states, networks of possible worlds) can all be represented as **simple graph-like structures**.*

Modal logics are

- tools to talk about relational, or graph-like structures.
- fragments of classical ones, with restricted forms of quantification ...
- ... which tend to be **decidable** and described in a pointfree notations.

The language

Syntax

$$\phi ::= p \mid \text{true} \mid \text{false} \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid \langle m \rangle \phi \mid [m] \phi$$

where $p \in \text{PROP}$ and $m \in \text{MOD}$

Disjunction (\vee) and equivalence (\leftrightarrow) are defined by abbreviation. The **signature** of the basic modal language is determined by sets **PROP** of **propositional** symbols (typically assumed to be denumerably infinite) and **MOD** of **modality** symbols.

The language

Notes

- if there is only one modality in the signature (i.e., MOD is a singleton), write simply $\Diamond\phi$ and $\Box\phi$
- the language has some redundancy: in particular modal connectives are **dual** (as quantifiers are in first-order logic): $[m]\phi$ is equivalent to $\neg\langle m\rangle\neg\phi$
- define **modal depth** in a formula ϕ , denoted by $\text{md } \phi$ as the maximum level of nesting of modalities in ϕ

Example

Models as LTSs over Act.

$\text{MOD} = \mathbb{P}\text{Act}$ – sets of actions.

\langle

$\rangle\phi$ can be read as “after observing a **or** b , ϕ must hold.”

$[$

$]\phi$ can be read as “after observing a **and** b , ϕ must hold.”

Semantics

Semantics

A **model** for the language is a pair $\mathfrak{M} = \langle \mathfrak{F}, V \rangle$, where

- $\mathfrak{F} = \langle W, \{R_m\}_{m \in \text{MOD}} \rangle$
is a **Kripke frame**, ie, a non empty set W and a family of **binary relations** (called *accessibility relations*) over W , one for each modality symbol $m \in \text{MOD}$. Elements of W are called **points**, **states**, **worlds** or simply **vertices** in directed graphs.
- $V : \text{PROP} \longrightarrow \mathcal{P}(W)$ is a **valuation**.

When $\text{MOD} = 1$

- $\Diamond\phi$ and $\Box\phi$ instead of $\langle \cdot \rangle \phi$ and $[\cdot] \phi$
- $\mathfrak{F} = \langle W, R \rangle$ instead of $\mathfrak{F} = \langle W, \{R_m\}_{m \in \text{MOD}} \rangle$

Semantics

Satisfaction: for a model \mathfrak{M} and a point w

$\mathfrak{M}, w \models \text{true}$

$\mathfrak{M}, w \not\models \text{false}$

$\mathfrak{M}, w \models p$ iff $w \in V(p)$

$\mathfrak{M}, w \models \neg\phi$ iff $\mathfrak{M}, w \not\models \phi$

$\mathfrak{M}, w \models \phi_1 \wedge \phi_2$ iff $\mathfrak{M}, w \models \phi_1$ and $\mathfrak{M}, w \models \phi_2$

$\mathfrak{M}, w \models \phi_1 \rightarrow \phi_2$ iff $\mathfrak{M}, w \not\models \phi_1$ or $\mathfrak{M}, w \models \phi_2$

$\mathfrak{M}, w \models \langle m \rangle \phi$ iff there exists $v \in W$ st $vR_m w$ and $\mathfrak{M}, v \models \phi$

$\mathfrak{M}, w \models [m] \phi$ iff for all $v \in W$ st $vR_m w$ and $\mathfrak{M}, v \models \phi$

Semantics

Satisfaction

A formula ϕ is

- **satisfiable in a model** \mathfrak{M} if it is satisfied at some point of \mathfrak{M}
- **globally satisfied** in \mathfrak{M} ($\mathfrak{M} \models \phi$) if it is satisfied at all points in \mathfrak{M}
- **valid** ($\models \phi$) if it is globally satisfied in all models
- **a semantic consequence** of a set of formulas Γ ($\Gamma \models \phi$) if for all models \mathfrak{M} and all points w , if $\mathfrak{M}, w \models \Gamma$ then $\mathfrak{M}, w \models \phi$

Proof system **K**

Minimal modal logic

- all formulas with the form of a **propositional tautology** (including formulas which contain modalities but are truth-functionally tautologous)
- all instances of the axiom schema:

$$\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$$

- two proof rules:

if $\vdash \phi$ and $\vdash \phi \rightarrow \psi$ then $\vdash \psi$ (**modus ponens**)

if $\vdash \phi$ then $\vdash \Box\phi$ (**generalization**)

Variants

Normal modal logics are **axiomatic extensions to K**

- different applications of modal logic typically validate different modal axioms;
- a normal modal logic is identified with the set of formulas it generates; it is said to be **consistent** if it does not contain all formulas. This identification immediately induces a lattice structure on the set of all such logics.

Variants

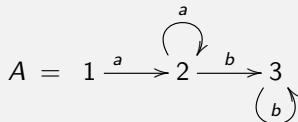
Modal axioms reflect **properties of accessibility relations**:

- **transitive** frames: $\Box\phi \rightarrow \Box\Box\phi$
- **simple** frames: $\Diamond\phi \rightarrow \Box\phi$
- frames consisting of **isolated reflexive points**: $\phi \leftrightarrow \Box\phi$
- frames consisting of **isolated irreflexive points**: $\Box\text{false}$

But there are classes of frames which are not modally definable, eg, **connected**, **irreflexive**, **containing a isolated irreflexive point**

Examples I

An automaton



- two modalities $\langle a \rangle$ and $\langle b \rangle$ to explore the corresponding classes of transitions
- note that

$$1 \models \langle a \rangle \cdots \langle a \rangle \langle b \rangle \cdots \langle b \rangle t$$

where t is a proposition valid only at the (terminal) state 3.

- all modal formulas of this form correspond to the strings accepted by the automaton, i.e. in language $\mathcal{L} = \{a^m b^n \mid m, n > 0\}$

Examples II

$(P, <)$ a strict partial order with infimum 0

- $P, x \models \Box \text{false}$ if x is a maximal element of P
- $P, 0 \models \Diamond \Box \text{false}$ iff ...
- $P, 0 \models \Box \Diamond \Box \text{false}$ iff ...

Examples III

Process logic (Hennessy-Milner logic)

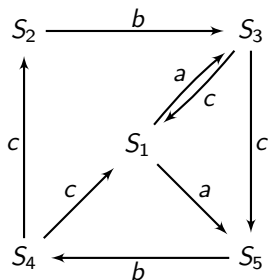
- $\text{PROP} = \emptyset$
- $W = \mathbb{P}$ is a set of states, typically process terms, in a labelled transition system
- each subset $K \subseteq \text{Act}$ of actions generates a modality corresponding to transitions labelled by an element of K

Assuming the underlying LTS $\mathfrak{F} = \langle \mathbb{P}, \{p \xrightarrow{K} p' \mid K \subseteq \text{Act}\} \rangle$ as the modal frame, satisfaction is abbreviated as

$$p \models \langle K \rangle \phi \quad \text{iff} \quad \exists_{q \in \{p' \mid p \xrightarrow{a} p' \wedge a \in K\}} \cdot q \models \phi$$

$$p \models [K] \phi \quad \text{iff} \quad \forall_{q \in \{p' \mid p \xrightarrow{a} p' \wedge a \in K\}} \cdot q \models \phi$$

Examples III



Prove:

- ① $S_2 \models [a] (\langle b \rangle tt \wedge \langle c \rangle tt)$
- ② $S_1 \not\models [a] (\langle b \rangle tt \wedge \langle c \rangle tt)$
- ③ $S_2 \models [b] [c] (\langle a \rangle tt \vee \langle b \rangle tt)$
- ④ $S_1 \models [b] [c] (\langle a \rangle tt \vee \langle b \rangle tt)$

Examples IV

Temporal logic

- $\langle T, < \rangle$ where T is a set of time points (instants, execution states, ...) and $<$ is the **earlier than** relation on T .
- Thus, $\Box\varphi$ (respectively, $\Diamond\varphi$) means that φ holds in all (respectively, some) time points.

Examples IV

$\langle T, < \rangle$

The structure of time is a **strict partial order**
(i.e., a transitive and asymmetric relation)

For any such structure, a new modality, \bigcirc , can be defined based on the **cover** relation \triangleleft for $<$ (i.e., $x \triangleleft y$ if (1) every $x < y$ and (2) there is no z such that $x < z < y$). Thus,

$$t \models \bigcirc \phi \quad \text{iff} \quad \forall t' \in \{p' \mid t \triangleleft t'\} . t' \models \phi$$

$$t \models \Box \phi \quad \text{iff} \quad \forall t' \in \{p' \mid t < t'\} . t' \models \phi$$

$$t \models \Diamond \phi \quad \text{iff} \quad \exists t' \in \{p' \mid t < t'\} . t' \models \phi$$

Examples IV

... but typical structures, however, are

Linear time structures

- **linear**: $\langle \forall x, y : x, y \in T : x = y \wedge x < y \wedge y < x \rangle$.
- **discrete**: for each $t \in T$, i) if there is a $u > t$ there is a first such u ;
ii) if there is a $u < t$ there is a last such u .
- **dense**: if for all $t, x \in T$, if $x < t$ there is a $v \in T$ such that $x < v < t$.
- **Dedekind complete**: if for all $S \subseteq T$ non-empty and bounded above, there is a least upper bound in T .
- **continuous**: if it is both dense and Dedekind complete

Examples V

Epistemic logic (J. Hintikka, 1962)

- W is a set of agents
- $\alpha \models i$ means i is the current knowledge of agent i
- $\alpha \models \Box j$ means the agent knows that j (in the sense that at each alternative epistemic situation information j is known)
- $\alpha \models \Diamond j$ means the agent knows that knowledge j is consistent with what the agent knows (is an epistemically acceptable alternative)

The first order connection

From modal logic

$$\phi ::= p \mid \text{true} \mid \text{false} \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid \langle m \rangle \phi \mid [m] \phi$$

To first order logic

$$\phi ::= Px \mid \text{true} \mid \text{false} \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \rightarrow \phi_2 \mid \langle \exists x :: \phi \rangle \mid \langle \forall y :: \phi \rangle$$

The first order connection

Boxes and diamonds are essentially a **macro notation** to encode quantification over accessible states in a point free way.

The standard translation

... to first-order logic **expands** these macros:

$$ST_x(p) = P_x$$

$$ST_x(\text{true}) = \text{true}$$

$$ST_x(\text{false}) = \text{false}$$

$$ST_x(\neg\phi) = \neg ST_x(\phi)$$

$$ST_x(\phi_1 \wedge \phi_2) = ST_x(\phi_1) \wedge ST_x(\phi_2)$$

$$ST_x(\phi_1 \rightarrow \phi_2) = ST_x(\phi_1) \rightarrow ST_x(\phi_2)$$

$$ST_x(\langle m \rangle \phi) = \langle \exists y :: (yR_mx \wedge ST_y(\phi)) \rangle$$

$$ST_x([m] \phi) = \langle \forall y :: (yR_mx \rightarrow ST_y(\phi)) \rangle$$

The first order connection

Lemma

For any ϕ , \mathfrak{M} and point w in \mathfrak{M} ,

$$\mathfrak{M}, w \models \phi \quad \text{iff} \quad \mathfrak{M} \models ST_x(\phi)[x \leftarrow w]$$

Note

Note how the (unique) free variable x in ST_x mirrors in first-order the internal perspective: **assigning a value to x corresponds to evaluating the modal formula at a certain state.**

The first order connection

The standard translation provides a **bridge** between modal logic and classical logic which makes possible to **transfer** results from one side to the other. For example,

Compactness

If Φ is a set of basic modal formulas and every finite subset of Φ is satisfiable, then Φ itself is satisfiable.

Löwenheim-Skolem

If Φ is a set of basic modal formulas satisfiable in at least one infinite model, then it is satisfiable in models of every infinite cardinality.

Summing up

- Propositional modal languages are syntactically simple languages that offer a **pointfree** notation for talking about **relational structures**
- They do this from the **inside**, using the modal operators to look for information at accessible states
- Regarded as a tool for talking about models, any basic modal language can be seen as **a fragment of first-order language**
- The **standard translation** systematically maps modal formulas to first-order formulas (in one free variable) and makes the quantification over accessible states explicit

Exercise

Express the following properties in Process Logic

- inevitability of a :
- progress:
- deadlock or termination:

“ $-$ ” stands for *Act*, and “ $-x$ ” abbreviates $Act - \{x\}$

Exercise

Express the following properties in Process Logic

- inevitability of a : $\langle - \rangle \text{ true} \wedge [-a] \text{ false}$
- progress:
- deadlock or termination:

“ $-$ ” stands for Act , and “ $-x$ ” abbreviates $Act - \{x\}$

Exercise

Express the following properties in Process Logic

- inevitability of a : $\langle - \rangle \text{true} \wedge [-a] \text{false}$
- progress: $\langle - \rangle \text{true}$
- deadlock or termination:

“ $-$ ” stands for Act , and “ $-x$ ” abbreviates $Act - \{x\}$

Exercise

Express the following properties in Process Logic

- inevitability of a : $\langle - \rangle \text{true} \wedge [-a] \text{false}$
- progress: $\langle - \rangle \text{true}$
- deadlock or termination: $[-] \text{false}$
- what about

$\langle - \rangle \text{false}$ and $[-] \text{true}$?

“ $-$ ” stands for *Act*, and “ $-x$ ” abbreviates $\text{Act} - \{x\}$

Exercise

Express the following properties in Process Logic

- $\phi_0 =$ *In a taxi network, a car can collect a passenger or be allocated by the Central to a pending service*
- $\phi_1 =$ *This applies only to cars already on service*
- $\phi_2 =$ *If a car is allocated to a service, it must first collect the passenger and then plan the route*
- $\phi_3 =$ *On detecting an emergence the taxi becomes inactive*
- $\phi_4 =$ *A car on service is not inactive*

Exercise

Process logic: The taxi network example

- $\phi_0 = \langle \text{rec}, \text{alo} \rangle \text{ true}$
- $\phi_1 = [\text{onservice}] \langle \text{rec}, \text{alo} \rangle \text{ true}$ or
 $\phi_1 = [\text{onservice}] \phi_0$
- $\phi_2 = [\text{alo}] \langle \text{rec} \rangle \langle \text{plan} \rangle \text{ true}$
- $\phi_3 = [\text{sos}] [-] \text{ false}$
- $\phi_4 = [\text{onservice}] \langle - \rangle \text{ true}$

Exercise

Standard translation to FOL

- Explain how propositional symbols and modalities are translated to first-order logic?
- In what sense can modal logic be regarded as a **pointfree** version of a FOL fragment?
- Compute $ST_x(p \Rightarrow \langle m \rangle p)$

Bisimulation

Definition

Given two models $\mathfrak{M} = \langle \langle W, R \rangle, V \rangle$ and $\mathfrak{M}' = \langle \langle W', R' \rangle, V' \rangle$, a **bisimulation** is a non-empty binary relation $S \subseteq W \times W'$ st whenever wSw' one has that

- points w and w' satisfy the same propositional symbols
- if vRw , then there is a point v' in \mathfrak{M}' st $v'Rw'$ and vSv' (zig)
- if $v'R'w'$, then there is a point v in \mathfrak{M} st vRw and vSv' (zag)

Bisimulation

Definition

- Bisimulations can be used to **expand** or **contract** models (cf via tree unraveling and contraction)
- Bisimulation vs model constructions (**disjoint union**, **generated submodels** and **bounded morphisms**)

Note

Note the relation to the notion of bisimulation in transition systems, independently discovered by Park (1982) in Computer Science.

Invariance and definability

Lemma (invariance: bisimulation implies modal equivalence)

Given two models $\mathfrak{M} = \langle \langle W, R \rangle, V \rangle$ and $\mathfrak{M}' = \langle \langle W', R' \rangle, V' \rangle$, and a **bisimulation** $S \subseteq W \times W'$, if two points w, w' are related by S , i.e. wSw' , then w, w' satisfy the same basic modal formulas.

Applications

- to prove bisimulation failures
- to show the undefinability of some structural notions, e.g.
irreflexivity is modally undefinable
- to show that typical model constructions are satisfaction preserving
- ...

Frame definability

- A modal formula is valid on a frame if it is true under every valuation at every world (i.e., it cannot be refuted)
- The class of frames defined by a modal formula ϕ are those where ϕ is valid.
- Example: $\Diamond\Diamond p \rightarrow \Diamond p$ defines transitivity:
 $\mathfrak{F} = \langle W, R \rangle$ is transitive iff for all V and w ,
 $\langle \mathfrak{F}, V \rangle, w \models \Diamond\Diamond p \rightarrow \Diamond p$

Exercise

Exercise: other properties

- Transitivity: $\Diamond\Diamond p \rightarrow \Diamond p$
- Reflexivity:
- Symmetry:
- Confluence:
- Irreflexibility:

Exercise

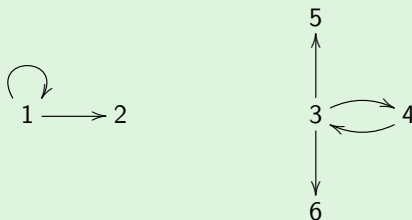
Exercise: other properties

- Transitivity: $\Diamond\Diamond p \rightarrow \Diamond p$
- Reflexivity: $p \rightarrow \Diamond p$
- Symmetry: $p \rightarrow \Box\Diamond p$
- Confluence: $\Diamond\Box p \rightarrow \Box\Diamond p$
- Irreflexibility: **Not possible**

Exercise

Bisimilarity and modal equivalence

- Show that **irreflexivity** is modally undefinable.
(i.e., no formula that characterises a irreflexive system)
- Consider the following transition systems:



Give a modal formula that can be satisfied at point 1 but not at 3.

Invariance and definability

To prove the converse of the invariance lemma requires passing to an **infinitary** modal language with arbitrary (countable) conjunctions and disjunctions. Alternatively, and more usefully, it can be shown for **finite** models:

Lemma (modal equivalence implies bisimulation)

If two points w, w' from two finite models $\mathfrak{M} = \langle \langle W, R \rangle, V \rangle$ and $\mathfrak{M}' = \langle \langle W', R' \rangle, V' \rangle$ satisfy the same modal formulas, then there is a bisimulation $S \subseteq W \times W'$ such that wSw' .

Invariance and definability

Note

- The result can be **weakened** to **image-finite** models.
- Combining this result with the invariance lemma one gets the so-called **modal equivalence theorem** stating that, for image-finite models, bisimilarity and modal equivalence coincide. The result is also known as the **Hennessy-Milner theorem** who first proved it for process logics.

Exercise

- Give an example of modally equivalent states in different Kripke structure which fail to be bisimilar.

Invariance and definability

Lemma (modal logic vs first-order)

The following are equivalent for all first-order formulas $\phi(x)$ in one free variable x :

- 1 $\phi(x)$ is invariant for bisimulation.
- 2 $\phi(x)$ is equivalent to the standard translation of a basic modal formula.

Therefore:

the basic modal language corresponds to the fragment of their first-order correspondence language that is invariant for bisimulation

Invariance and definability

- the basic modal language (interpreted over the class of all models) is computationally better behaved than the corresponding first-order language (interpreted over the same models)
- ... but clearly less expressive

	model checking	satisfiability
ML	PTIME	PSPACE-complete
FOL	PSPACE-complete	undecidable

What are the trade-offs? Can this better computational behaviour be lifted to more expressive modal logics?

mCRL2 - modal logic

todo

Richer modal logics

can be obtained in different ways, e.g.

- axiomatic extensions
- introducing more complex satisfaction relations
- support novel semantic capabilities
- ...

Examples

- richer temporal logics
- hybrid logic
- modal μ -calculus

Temporal logics with \mathcal{U} and \mathcal{S}

Until and Since

$\mathfrak{M}, w \models \phi \mathcal{U} \psi$	iff	there exists $v \in W$ st vRw and $\mathfrak{M}, v \models \psi$, and for all u st uRw and vRu , one has $\mathfrak{M}, u \models \phi$
$\mathfrak{M}, w \models \phi \mathcal{S} \psi$	iff	there exists $v \in W$ st wRv and $\mathfrak{M}, v \models \psi$, and for all u st uRv and wRu , one has $\mathfrak{M}, u \models \phi$

- note the $\exists\forall$ qualification pattern: these operators are neither diamonds nor boxes.
- more expressive — e.g. helpful to express **guarantee** properties, e.g. **some event will happen, and a certain condition will hold until then**

Exercise

Temporal logics

- Show that \mathcal{U} is modally undefinable.

Hint Consider the following transition structures and formula $\text{false} \mathcal{U} \text{true}$:



- Would this be the case if we restrict ourselves to transitive, irreflexive models?

Linear temporal logic (LTL)

$$\phi := \text{true} \mid p \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \bigcirc\phi \mid \phi_1 \mathcal{U} \phi_2 \mid \Diamond\phi \mid \Box\phi$$

mutual exclusion	$\Box(\neg c_1 \vee \neg c_2)$
liveness	$\Box\Diamond c_1 \wedge \Box\Diamond c_2$
starvation freedom	$(\Box\Diamond w_1 \rightarrow \Box\Diamond c_1) \wedge (\Box\Diamond w_1 \rightarrow \Box\Diamond c_1)$
progress	$\Box(w_1 \rightarrow \Diamond c_1)$
weak fairness	$\Diamond\Box w_1 \rightarrow \Box\Diamond c_1$
eventually forever	$\Diamond\Box w_1$

- First temporal logic to reason about reactive systems [Pnueli, 1977]
- Formulas are interpreted over **execution paths**
- Express **linear-time properties**

Computational tree logic (CTL, CTL*)

state formulas to express properties of a state:

$$\Phi := \text{true} \mid \Phi \wedge \Phi \mid \neg\Phi \mid \exists\psi \mid \forall\psi$$

path formulas to express properties of a path:

$$\psi := \bigcirc\Phi \mid \Phi \mathcal{U} \Psi \mid \Diamond\phi \mid \Box\phi$$

mutual exclusion	$\forall \Box (\neg c_1 \vee \neg c_2)$
liveness	$\forall \Box \forall \Diamond c_1 \wedge \forall \Box \forall \Diamond c_2$
order	$\forall \Box (c_1 \vee \forall \bigcirc c_2)$

- Branching time structure encode transitive, irreflexive but not necessarily linear flows of time
- flows are **trees**: past linear; branching future

Hybrid logic

Motivation

Add the possibility of **naming** points and reason about their **identity**

Compare:

$$\Diamond(r \wedge p) \wedge \Diamond(r \wedge q) \rightarrow \Diamond(p \wedge q)$$

with

$$\Diamond(i \wedge p) \wedge \Diamond(i \wedge q) \rightarrow \Diamond(p \wedge q)$$

for $i \in \text{NOM}$ (a **nominal**)

Syntax

$$\phi ::= \dots \mid p \mid \langle m \rangle \phi \mid [m] \phi \mid i \mid @_i \phi$$

where $p \in \text{PROP}$ and $m \in \text{MOD}$ and $i \in \text{NOM}$

Hybrid logic

Nominals i

- Are special propositional symbols that hold exactly on one state (the state they **name**)
- In a model the **valuation** V is extended from

$$V : \text{PROP} \longrightarrow \mathcal{P}(W)$$

to

$$V : \text{PROP} \longrightarrow \mathcal{P}(W) \quad \text{and} \quad V : \text{NOM} \longrightarrow W$$

where NOM is the set of nominals in the model

- Satisfaction:

$$\mathfrak{M}, w \models i \qquad \text{iff } w = V(i)$$

Hybrid logic

The $@_i$ operator

$\mathfrak{M}, w \models @_i \phi$ iff $\mathfrak{M}, u \models \phi$ and u is the state denoted by i

Standard translation to first-order

$$\begin{aligned} ST_x(i) &= (x = i) \\ ST_x(@_i \phi) &= ST_i(\phi)(x = i) \end{aligned}$$

i.e., hybrid logic corresponds to a first-order language enriched with constants and equality.

Hybrid logic

Increased frame definability

- irreflexivity: $i \rightarrow \neg \Diamond i$
- asymmetry: $i \rightarrow \neg \Diamond \Diamond i$
- antisymmetry: $i \rightarrow \Box (\Diamond i \rightarrow i)$
- trichotomy: $@_j \Diamond i \vee @_{i_j} \vee @_i \Diamond j$

Bisimulation with nominals

Definition

Given two models $\mathfrak{M} = \langle \langle W, R \rangle, V \rangle$ and $\mathfrak{M}' = \langle \langle W', R' \rangle, V' \rangle$, a **bisimulation** is a non-empty binary relation $S \subseteq W \times W'$ st whenever wSw' one has that

- points w and w' satisfy the same propositional symbols **and nominals**
- if vRw , then there is a point v' in \mathfrak{M}' st $v'Rw'$ and vSv' (**zig**)
- if $v'Rw'$, then there is a point v in \mathfrak{M} st vRw and vSv' (**zag**)
- $V(i) R V'(i)$ for all nominal i (**name consistency**)

An **invariance** theorem and its **dual** (for image finite models) can also be proved

Hybrid logic

Summing up

- basic hybrid logic is a simple notation for capturing the bisimulation-invariant fragment of first-order logic with constants and equality, i.e., a mechanism for equality reasoning in propositional modal logic.
- comes cheap: up to a polynomial, the complexity of the resulting decision problem is no worse than for the basic modal language

Hybrid logic

Applications to architectural design

- layout of coordination circuits (e.g. in Reo)
- reconfigurable architectures (parametric on a specification logic)
- hierarchical architectures (e.g. UML statecharts)
- ...

[recent research at HASLab: projects Dali and Nasoni]

Applications to architectural design

Structural reasoning over Reo circuits

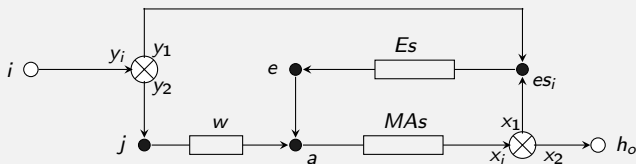
$$\phi ::= p \mid i \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid [K]\phi \mid \llbracket K \rrbracket\phi \mid @_i\phi$$

- modalities are indexed by regular expressions over channel **types**;
- $\langle K \rangle$ and $[K]$ (resp., $\llbracket K \rrbracket$ and $\llbracket K \rrbracket$) express properties of **outgoing** (resp., **incoming**) connections from the node in which they are evaluated.

[Nuno Oliveira PhD thesis (MAP-i, 2015)]

Applications to architectural design

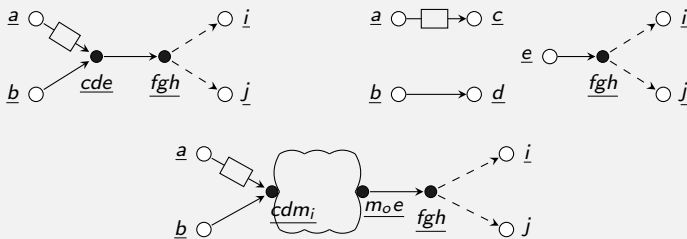
Structural reasoning over Reo circuits



- 1 $\phi_1 \triangleq @_{t_o} \langle -^* \rangle \text{true} \wedge [-^*] [-MAs] \text{false}$
(there is a path from triage input port (t_o) to a MAs edge)
- 2 $\phi_2 \triangleq [-] \text{false} \rightarrow [-^*] h_o$
(all paths from input ports, lead to the billing service (h_o) port)

Applications to architectural design

Reconfiguration of Reo circuits



Invariant $\Phi = \langle \text{sync} \rangle (\langle - \rangle \text{true} \wedge [-\text{lossy}] \text{false})$ is displaced along a reconfiguration:

$$@_{\underline{cde}} \Phi \rightsquigarrow @_{\underline{m_o e}} \Phi$$

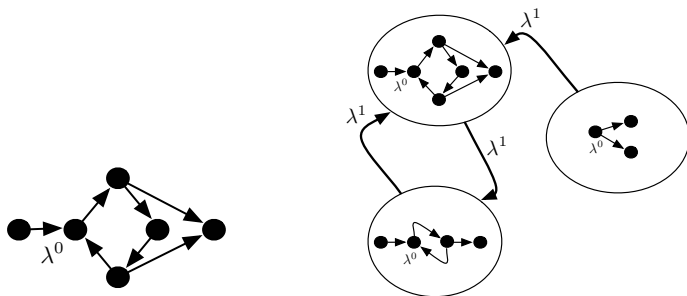
Applications to architectural design

Specifying reconfigurable architectures

- Reconfigurable architectures are represented as **structured transition systems** whose
- states are endowed with **local** specifications and
- the **global** transition structure models system's evolution through possible configurations.
- The hybrid language is developed **on top** whichever logic is taken for the local configurations (e.g., equational, first-order, fuzzy, etc.) — by **hybridisation**.

[Alexandre Madeira PhD thesis (MAP-i, 2013)]

Applications to architectural design



- \mathcal{H} : pure hybrid formulas
- \mathcal{H}^2 : hierarchical structures, e.g.

$$@_{j^1} k^0 \wedge^1 [\lambda^1](\rho_1, \dots, \rho_n)$$

Applications to architectural design

Hierarchical architectures

- Hierarchical architectures are represented as **hierarchical transition systems** whose states are transition systems themselves
- and (intrusive) transitions between designated states in different local transition systems at different levels of abstraction are allowed.
- Hybrid logic captures this principle which is inherent to well known design formalisms such as statecharts and UML.

