

Modal logic for concurrent processes

Luís Soares Barbosa



Universidade do Minho



UNITED NATIONS
UNIVERSITY

UNU-EGOV

Interaction & Concurrency Course Unit (Lcc)

Universidade do Minho, 23.III.2018

Motivation

System's correctness wrt a specification

- equivalence checking (between two designs), through \sim and $=$
- unsuitable to check properties such as

can the system perform action α followed by β ?

which are best answered by exploring the process state space

Which logic?

- **Modal logic** over transition systems
- The **Hennessy-Milner logic** (offered in mCRL2)
- The **modal μ -calculus** (offered in mCRL2)

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

The language

Semantics

A **model** for the language is a pair $\mathfrak{M} = \langle \mathbb{F}, V \rangle$, 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 over W , one for each modality symbol $m \in \text{MOD}$.
Elements of W are called **points**, **states**, **worlds** or simply **vertices** in the directed graphs corresponding to the modality symbols.
- $V : \text{PROP} \longrightarrow \mathcal{P}(W)$ is a **valuation**.

The language

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 wR_mv and $\mathfrak{M}, v \models \phi$

$\mathfrak{M}, w \models [m]\phi$

iff for all $v \in W$ st wR_mv and $\mathfrak{M}, v \models \phi$

The language

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

Examples

Temporal logic

- W is a set of instants
- there is a unique modality corresponding to the **transitive closure of the next-time relation**
- **origin**: Arthur Prior, an attempt to *deal with temporal information from the inside, capturing the situated nature of our experience and the context-dependent way we talk about it*

Examples

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

$$\begin{array}{ll}
 p \models \langle K \rangle \phi & \text{iff} \quad \exists_{q \in \{p' \mid p \xrightarrow{a} p' \wedge a \in K\}} \cdot q \models \phi \\
 p \models [K] \phi & \text{iff} \quad \forall_{q \in \{p' \mid p \xrightarrow{a} p' \wedge a \in K\}} \cdot q \models \phi
 \end{array}$$

Examples

Process logic: The taxi network example

- $\phi_0 =$ *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*

Examples

Process logic: The taxi network example

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

Process logic: typical properties

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

$\langle - \rangle false$ and $[-] true$?

- satisfaction decided by unfolding the definition of \models : no need to compute the transition graph

Hennessy-Milner logic

... propositional logic with **action** modalities

Syntax

$$\phi ::= \text{true} \mid \text{false} \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \langle K \rangle \phi \mid [K] \phi$$

Semantics: $E \models \phi$

$$E \models \text{true}$$

$$E \not\models \text{false}$$

$$E \models \phi_1 \wedge \phi_2 \quad \text{iff} \quad E \models \phi_1 \quad \wedge \quad E \models \phi_2$$

$$E \models \phi_1 \vee \phi_2 \quad \text{iff} \quad E \models \phi_1 \quad \vee \quad E \models \phi_2$$

$$E \models \langle K \rangle \phi \quad \text{iff} \quad \exists_{F \in \{E' \mid E \xrightarrow{a} E' \wedge a \in K\}} . F \models \phi$$

$$E \models [K] \phi \quad \text{iff} \quad \forall_{F \in \{E' \mid E \xrightarrow{a} E' \wedge a \in K\}} . F \models \phi$$

Example

$$Sem \hat{=} get.put.Sem$$

$$P_i \hat{=} \overline{get}.c_i.\overline{put}.P_i$$

$$S \hat{=} (Sem \mid (\mid_{i \in I} P_i)) \setminus_{\{get, put\}}$$

- $Sem \models \langle get \rangle true$ holds because

$$\exists F \in \{Sem' \mid Sem \xrightarrow{get} Sem'\} . F \models true$$

with $F = put.Sem$.

- However, $Sem \models [put] false$ also holds, because

$$T = \{Sem' \mid Sem \xrightarrow{put} Sem'\} = \emptyset.$$

Hence $\forall F \in T . F \models false$ becomes trivially true.

- The only action initially permited to S is τ : $\models [-\tau] false$.

Example

$$Sem \hat{=} get.put.Sem$$

$$P_i \hat{=} \overline{get}.c_i.\overline{put}.P_i$$

$$S \hat{=} (Sem \mid (\mid_{i \in I} P_i)) \setminus_{\{get, put\}}$$

- Afterwards, S can engage in any of the critical events c_1, c_2, \dots, c_i :
 $[\tau] \langle c_1, c_2, \dots, c_i \rangle true$
- After the semaphore initial synchronization and the occurrence of c_j in P_j , a new synchronization becomes inevitable:
 $S \models [\tau][c_j](\langle - \rangle true \wedge [-\tau] false)$

Exercise

Verify:

$$\neg \langle a \rangle \phi = [a] \neg \phi$$

$$\neg [a] \phi = \langle a \rangle \neg \phi$$

$$\langle a \rangle \text{false} = \text{false}$$

$$[a] \text{true} = \text{true}$$

$$\langle a \rangle (\phi \vee \psi) = \langle a \rangle \phi \vee \langle a \rangle \psi$$

$$[a] (\phi \wedge \psi) = [a] \phi \wedge [a] \psi$$

$$\langle a \rangle \phi \wedge [a] \psi \Rightarrow \langle a \rangle (\phi \wedge \psi)$$

A denotational semantics

Idea: associate to each formula ϕ the **set** of processes that makes it true

ϕ vs $|\phi| = \{E \in \mathbb{P} \mid E \models \phi\}$

$$|true| = \mathbb{P}$$

$$|false| = \emptyset$$

$$|\phi_1 \wedge \phi_2| = |\phi_1| \cap |\phi_2|$$

$$|\phi_1 \vee \phi_2| = |\phi_1| \cup |\phi_2|$$

$$|[K]\phi| = |[K]|(|\phi|)$$

$$|\langle K \rangle \phi| = |\langle K \rangle|(|\phi|)$$

A denotational semantics

Idea: associate to each formula ϕ the **set** of processes that makes it true

ϕ vs $|\phi| = \{E \in \mathbb{P} \mid E \models \phi\}$

$$|\text{true}| = \mathbb{P}$$

$$|\text{false}| = \emptyset$$

$$|\phi_1 \wedge \phi_2| = |\phi_1| \cap |\phi_2|$$

$$|\phi_1 \vee \phi_2| = |\phi_1| \cup |\phi_2|$$

$$|[K]\phi| = |[K]|(|\phi|)$$

$$|\langle K \rangle \phi| = |\langle K \rangle|(|\phi|)$$

$[[K]]$ and $|\langle K \rangle|$

Just as \wedge corresponds to \cap and \vee to \cup , modal logic combinators correspond to **unary functions** on sets of processes:

$$[[K]](X) = \{F \in \mathbb{P} \mid \text{if } F \xrightarrow{a} F' \wedge a \in K \text{ then } F' \in X\}$$

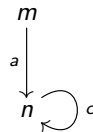
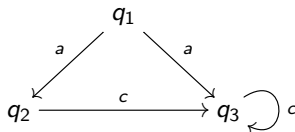
$$|\langle K \rangle|(X) = \{F \in \mathbb{P} \mid \exists_{F' \in X, a \in K} . F \xrightarrow{a} F'\}$$

Note

These combinators perform a **reduction to the previous state** indexed by actions in K

$||[K]||$ and $|\langle K \rangle|$

Example



$$|\langle a \rangle| \{q_2, n\} = \{q_1, m\}$$

$$|[a]| \{q_2, n\} = \{q_2, q_3, m, n\}$$

A denotational semantics

$$E \models \phi \text{ iff } E \in |\phi|$$

Example: $0 \models [-]false$

because

$$\begin{aligned} |[-]false| &= |[-]|(|false|) \\ &= |[-]|(\emptyset) \\ &= \{F \in \mathbb{P} \mid \text{if } F \xrightarrow{x} F' \wedge x \in Act \text{ then } F' \in \emptyset\} \\ &= \{0\} \end{aligned}$$

A denotational semantics

$$E \models \phi \text{ iff } E \in |\phi|$$

Example: $?? \models \langle - \rangle true$

because

$$\begin{aligned} |\langle - \rangle true| &= |\langle - \rangle|(|true|) \\ &= |\langle - \rangle|(\mathbb{P}) \\ &= \{F \in \mathbb{P} \mid \exists_{F' \in \mathbb{P}, a \in K} . F \xrightarrow{a} F'\} \\ &= \mathbb{P} \setminus \{0\} \end{aligned}$$

A denotational semantics

Complement

Any property ϕ divides \mathbb{P} into two disjoint sets:

$$|\phi| \text{ and } \mathbb{P} - |\phi|$$

The **characteristic formula** of the complement of $|\phi|$ is ϕ^c :

$$|\phi^c| = \mathbb{P} - |\phi|$$

where ϕ^c is defined inductively on the formulae structure:

$$true^c = false \quad false^c = true$$

$$(\phi_1 \wedge \phi_2)^c = \phi_1^c \vee \phi_2^c$$

$$(\phi_1 \vee \phi_2)^c = \phi_1^c \wedge \phi_2^c$$

$$(\langle a \rangle \phi)^c = [a] \phi^c$$

... but **negation** is not explicitly introduced in the logic.

Modal Equivalence

For each (finite or infinite) set Γ of formulae,

$$E \equiv_{\Gamma} F \iff \forall \phi \in \Gamma. E \models \phi \iff F \models \phi$$

Examples

$$a.b.0 + a.c.0 \equiv_{\Gamma} a.(b.0 + c.0)$$

for $\Gamma = \{\langle x_1 \rangle \langle x_2 \rangle \dots \langle x_n \rangle \text{true} \mid x_i \in \text{Act}\}$

(what about \equiv_{Γ} for $\Gamma = \{\langle x_1 \rangle \langle x_2 \rangle \langle x_3 \rangle \dots \langle x_n \rangle [-] \text{false} \mid x_i \in \text{Act}\} ?$)

Modal Equivalence

For each (finite or infinite) set Γ of formulae,

$$E \equiv_{\Gamma} F \iff \forall \phi \in \Gamma. E \models \phi \iff F \models \phi$$

Examples

$$a.b.0 + a.c.0 \equiv_{\Gamma} a.(b.0 + c.0)$$

for $\Gamma = \{\langle x_1 \rangle \langle x_2 \rangle \dots \langle x_n \rangle \text{true} \mid x_i \in \text{Act}\}$

(what about \equiv_{Γ} for $\Gamma = \{\langle x_1 \rangle \langle x_2 \rangle \langle x_3 \rangle \dots \langle x_n \rangle [-] \text{false} \mid x_i \in \text{Act}\} \text{ ?}$)

Modal Equivalence

For each (finite or infinite) set Γ of formulae,

$$E \equiv_{\Gamma} F \iff \forall \phi \in \Gamma. E \models \phi \iff F \models \phi$$

Examples

$$a.b.0 + a.c.0 \equiv_{\Gamma} a.(b.0 + c.0)$$

for $\Gamma = \{\langle x_1 \rangle \langle x_2 \rangle \dots \langle x_n \rangle \text{true} \mid x_i \in \text{Act}\}$

(what about \equiv_{Γ} for $\Gamma = \{\langle x_1 \rangle \langle x_2 \rangle \langle x_3 \rangle \dots \langle x_n \rangle [-] \text{false} \mid x_i \in \text{Act}\}$?)

Modal Equivalence

For each (finite or infinite) set Γ of formulae,

$$E \models F \iff E \models_{\Gamma} F \text{ for every set } \Gamma \text{ of well-formed formulae}$$

Lemma

$$E \sim F \Rightarrow EF$$

Note

the converse of this lemma does not hold, e.g. let

- $A \hat{=} \sum_{i \geq 0} A_i$, where $A_0 \hat{=} 0$ and $A_{i+1} \hat{=} a.A_i$
- $A' \hat{=} A + \underline{fix}(X = a.X)$

$$A \sim A' \text{ but } A \not\models A'$$

Modal Equivalence

For each (finite or infinite) set Γ of formulae,

$$E \models F \iff E \models_{\Gamma} F \text{ for every set } \Gamma \text{ of well-formed formulae}$$

Lemma

$$E \sim F \Rightarrow EF$$

Note

the converse of this lemma does not hold, e.g. let

- $A \hat{=} \sum_{i \geq 0} A_i$, where $A_0 \hat{=} 0$ and $A_{i+1} \hat{=} a.A_i$
- $A' \hat{=} A + \underline{\text{fix}} (X = a.X)$

$$A \not\sim A' \text{ but } A \hat{=} A'$$

Modal Equivalence

For each (finite or infinite) set Γ of formulae,

$$E \models F \iff E \models F \text{ for every set } \Gamma \text{ of well-formed formulae}$$

Lemma

$$E \sim F \Rightarrow EF$$

Note

the converse of this lemma does not hold, e.g. let

- $A \hat{=} \sum_{i \geq 0} A_i$, where $A_0 \hat{=} 0$ and $A_{i+1} \hat{=} a.A_i$
- $A' \hat{=} A + \underline{fix}(X = a.X)$

$$A \not\sim A' \text{ but } A \hat{=} A'$$

Modal Equivalence

Theorem [Hennessy-Milner, 1985]

$$E \sim F \Leftrightarrow EF$$

for **image-finite** processes.

Image-finite processes

E is **image-finite** iff $\{F \mid E \xrightarrow{a} F\}$ is **finite** for every action $a \in Act$

Modal Equivalence

Theorem [Hennessy-Milner, 1985]

$$E \sim F \Leftrightarrow EF$$

for **image-finite** processes.

Image-finite processes

E is **image-finite** iff $\{F \mid E \xrightarrow{a} F\}$ is **finite** for every action $a \in Act$

Modal Equivalence

Theorem [Hennessy-Milner, 1985]

$$E \sim F \Leftrightarrow EF$$

for image-finite processes.

proof

\Rightarrow : by induction of the formula structure

\Leftarrow : show that \sim is itself a bisimulation, by contradiction