Logics for processes (I)

Luís S. Barbosa

DI-CCTC
Universidade do Minho
Braga, Portugal

8 April, 2011

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

Motivation

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*

Motivation

The taxi network example

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

Notes

- Modalities: $\langle K\rangle\phi$, $[L]\psi$ for $K, L \subset Act$
- Valuations in non modal logics are based on valuations V : Variables $\longrightarrow \mathbf{2}$: propositions are true or false depending on the unique referential provided by V
- Valuations in a modal logic also depends on the current state of computation: V : Variables $\times\ \mathbb{P} \longrightarrow \mathbf{2}$ or, equivalently, , V : Variables $\longrightarrow \mathcal{P}\mathbb{P}$: each variable is associated to the set of processes in which its value is fixed as true
- In our case, models for such a logic are defined over the universe of processes $\mathbb{P}$ (*i.e.*, terms of our process language) equipped with relations $\{\xrightarrow{x}\ | x \in Act\}$ defined by the operational semantics of the language.
- ... but the topic modal logics has a longer story and a broad spectrum of applications ...

The language

Syntax

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

The language

Semantics: $E \models \phi$

$$
\begin{array}{lll}
E \models \text{true} & & \\
E \not\models \text{false} & & \\
E \models \phi_1 \wedge \phi_2 & \text{iff} & E \models \phi_1 \;\wedge\; E \models \phi_2 \\
E \models \phi_1 \vee \phi_2 & \text{iff} & E \models \phi_1 \;\vee\; E \models \phi_2 \\
E \models \langle K \rangle \phi & \text{iff} & \exists_{F \in \{E' \mid E \xrightarrow{a} E' \,\wedge\, a \in K\}} \,.\, F \models \phi \\
E \models [K]\phi & \text{iff} & \forall_{F \in \{E' \mid E \xrightarrow{a} E' \,\wedge\, a \in K\}} \,.\, F \models \phi
\end{array}
$$

Example

$$Sem \triangleq get.put.Sem$$
$$P_i \triangleq \overline{get}.c_i.\overline{put}.P_i$$
$$S \triangleq \text{new}\,\{get, put\}\ (Sem \mid (|_{i \in I}\ P_i))$$

- $Sem \models \langle get \rangle \text{true}$ holds because

  $$\exists_{F \in \{Sem' \mid Sem \overset{get}{\rightarrow} Sem'\}} \,.\, F \models \text{true}$$

  with $F = put.Sem$.

- However, $Sem \models [put]\text{false}$ also holds, because
  $T = \{Sem' \mid Sem \overset{put}{\rightarrow} Sem'\} = \emptyset$.
  Hence $\forall_{F \in T} \,.\, F \models \text{false}$ becomes trivially true.

- The only action initially permmited to S is τ: $\models [-\tau]\text{false}$.

Example

$$Sem \triangleq get.put.Sem$$
$$P_i \triangleq \overline{get}.c_i.\overline{put}.P_i$$
$$S \triangleq \text{new}\,\{get, put\}\ (Sem \mid (|_{i \in I}\ P_i))$$

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

Notes

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

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

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

A denotational semantics

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

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

$$\begin{aligned} \|\mathrm{true}\| &= \mathbb{P} \\ \|\mathrm{false}\| &= \emptyset \\ \|\phi_1 \wedge \phi_2\| &= \|\phi_1\| \cap \|\phi_2\| \\ \|\phi_1 \vee \phi_2\| &= \|\phi_1\| \cup \|\phi_2\| \end{aligned}$$

$$\begin{aligned} \|[K]\phi\| &= \|[K]\|(\|\phi\|) \\ \|\langle K\rangle\phi\| &= \|\langle K\rangle\|(\|\phi\|) \end{aligned}$$

$\|[K]\|$ 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:

$$\|[K]\| = \lambda_{X\subseteq\mathbb{P}} \,.\, \{F \in \mathbb{P} \mid \text{if } F \xrightarrow{a} F' \wedge a \in K \text{ then } F' \in X\}$$

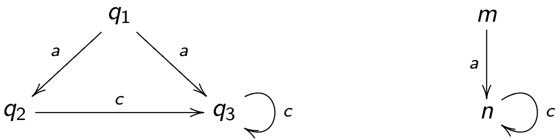
$$\|\langle K\rangle\| = \lambda_{X\subseteq\mathbb{P}} \,.\, \{F \in \mathbb{P} \mid \exists_{F'\in X, a\in K} \,.\, F \xrightarrow{a} F'\}$$

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

$$\boxed{E \models \phi \ \text{ iif } \ E \in \|\phi\|}$$

Example: $\mathbf{0} \models [-]\text{false}$

because

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

A denotational semantics

$$\boxed{E \models \phi \text{ iif } E \in \|\phi\|}$$

Example: ?? $\models \langle - \rangle \mathsf{true}$

because

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

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:

$$\begin{gathered} \text{true}^c = \text{false} \qquad \text{false}^c = \text{true} \\ (\phi_1 \wedge \phi_2)^c = \phi_1^c \vee \phi_2^c \\ (\phi_1 \vee \phi_2)^c = \phi_1^c \wedge \phi_2^c \\ (\langle a \rangle \phi)^c = [a]\phi^c \end{gathered}$$

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

Modal Equivalence

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

$$E \simeq_\Gamma F \quad \Leftrightarrow \quad \forall_{\phi \in \Gamma} \,.\, E \models \phi \Leftrightarrow F \models \phi$$

Examples

$$a.b.\mathbf{0} + a.c.\mathbf{0} \;\simeq_\Gamma\; a.(b.\mathbf{0} + c.\mathbf{0})$$

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

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

Modal Equivalence

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

$$E \simeq F \quad \Leftrightarrow \quad E \simeq_\Gamma F \text{for every set } \Gamma \text{ of well-formed formulae}$$

Lemma

$$E \sim F \quad \Rightarrow \quad E \simeq F$$

Note

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

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

$$A \nsim A' \quad \text{but} \quad A \simeq A'$$

Modal Equivalence

Theorem [Hennessy-Milner, 1985]

$$E \sim F \quad \Leftrightarrow \quad E \simeq F$$

for image-finite processes.

Image-finite processes

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

Modal Equivalence

Theorem [Hennessy-Milner, 1985]

$$E \sim F \quad \Leftrightarrow \quad E \simeq F$$

for image-finite processes.

proof

$\Rightarrow$: by induction of the formula structure

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