Logic for Processes

Luís Soares Barbosa

HASLab - INESC TEC
Universidade do Minho
Braga, Portugal

May 2019

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 $p \in$ PROP and $m \in$ 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 $\mathsf{md}\,\phi$ 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

- $\mathfrak{F} = \langle W, \{R_m\}_{m \in \mathrm{MOD}} \rangle$
  is a Kripke frame, ie, a non empty set W and a family of binary relations over W, one for each modality symbol $m \in \mathrm{MOD}$.
  Elements of W are called points, states, worlds or simply vertices in the directed graphs corresponding to the modality symbols.
- $V : \mathrm{PROP} \longrightarrow \mathcal{P}(W)$ 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 $wR_m v$ and $\mathfrak{M}, v \models \phi$
$\mathfrak{M}, w \models [m]\phi$	iff	for all $v \in W$ st $wR_m v$ and $\mathfrak{M}, v \models \phi$

The language

Safistaction

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 $K \subseteq Act$ of actions generates a modality corresponding to transitions labelled by an element of K

Assuming the underlying LTS $\mathfrak{F} = \langle \mathbb{P}, \{p \xrightarrow{K} p' \mid K \subseteq Act\} \rangle$ 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

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

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

Process logic: typical properties

- 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

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: $E \models \phi$

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

Example

$$\begin{aligned} Sem &\triangleq get.put.Sem \\ P_i &\triangleq \overline{get}.c_i.\overline{put}.P_i \\ S &\triangleq (Sem \mid (|_{i \in I}\ P_i)) \backslash \{get, put\} \end{aligned}$$

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

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

  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 permmited to S is τ: $\models [-\tau]$false.

Example

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

- 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: $S \models [\tau][c_j](\langle - \rangle \text{true} \wedge [-\tau]\text{false})$

Exercise

Verify:

$$
\begin{aligned}
\neg\langle a\rangle\phi &= [a]\neg\phi \\
\neg[a]\phi &= \langle a\rangle\neg\phi \\
\langle a\rangle\mathsf{false} &= \mathsf{false} \\
[a]\mathsf{true} &= \mathsf{true} \\
\langle a\rangle(\phi\vee\psi) &= \langle a\rangle\phi\vee\langle a\rangle\psi \\
[a](\phi\wedge\psi) &= [a]\phi\wedge[a]\psi \\
\langle a\rangle\phi\wedge[a]\psi &\Rightarrow \langle a\rangle(\phi\wedge\psi)
\end{aligned}
$$

A denotational semantics

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

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

$$\|\mathsf{true}\| = \mathbb{P}$$
$$\|\mathsf{false}\| = \emptyset$$
$$\|\phi_1 \wedge \phi_2\| = \|\phi_1\| \cap \|\phi_2\|$$
$$\|\phi_1 \vee \phi_2\| = \|\phi_1\| \cup \|\phi_2\|$$

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

A denotational semantics

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

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

$$\|\mathsf{true}\| = \mathbb{P}$$
$$\|\mathsf{false}\| = \emptyset$$
$$\|\phi_1 \wedge \phi_2\| = \|\phi_1\| \cap \|\phi_2\|$$
$$\|\phi_1 \vee \phi_2\| = \|\phi_1\| \cup \|\phi_2\|$$

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

$\|[K]\|$ 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:

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

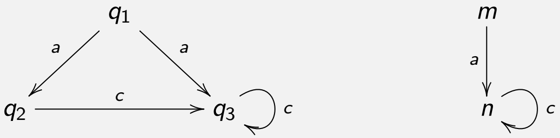
$$\|\langle K\rangle\|(X) = \{F \in \mathbb{P} \mid \exists_{F' \in X, a \in K} \,.\, F \xrightarrow{a} F'\}$$

Note

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

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

Example



$$
\begin{aligned}
\|\langle a\rangle\|\{q_2, n\} &= \{q_1, m\} \\
\|[a]\|\{q_2, n\} &= \{q_2, q_3, m, n\}
\end{aligned}
$$

A denotational semantics

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

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

because

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

A denotational semantics

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

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

because

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

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}:

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

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

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

... 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,

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

Examples

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

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

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

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

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

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 $A_{i+1} \triangleq a.A_i$
- $A' \triangleq A + \underline{fix}\ (X = a.X)$

$$\neg(A \sim A') \quad \text{but} \quad A \simeq A'$$

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 $A_{i+1} \triangleq a.A_i$
- $A' \triangleq A + \underline{fix}\ (X = a.X)$

$$\neg(A \sim A') \quad \text{but} \quad A \simeq A'$$

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 $A_{i+1} \triangleq a.A_i$
- $A' \triangleq A + \underline{fix}\ (X = a.X)$

$$\neg(A \sim A') \quad \text{but} \quad A \simeq A'$$

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 $\{F \mid E \xrightarrow{a} F\}$ 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.

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]

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

for image-finite processes.

proof

$\Rightarrow$: by induction of the formula structure

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

Is Hennessy-Milner logic expressive enough?

Is Hennessy-Milner logic expressive enough?

- It cannot detect deadlock in an arbitrary process
- or general safety: all reachable states verify ϕ
- or general liveness: there is a reachable states which verifies ϕ
- ...

... essentially because

formulas in cannot see deeper than their modal depth

Is Hennessy-Milner logic expressive enough?

Example

$$\phi \;=\; \text{a taxi eventually returns to its Central}$$

$$\phi \;=\; \langle reg\rangle\mathsf{true} \vee \langle -\rangle\langle reg\rangle\mathsf{true} \vee \langle -\rangle\langle -\rangle\langle reg\rangle\mathsf{true} \vee \langle -\rangle\langle -\rangle\langle -\rangle\langle reg\rangle\mathsf{true} \vee \ldots$$

Revisiting Hennessy-Milner logic

Adding regular expressions

ie, with regular expressions within modalities

$$\rho \ ::= \epsilon \mid \alpha \mid \rho.\rho \mid \rho + \rho \mid \rho^* \mid \rho^+$$

where

- α is an action formula and ϵ is the empty word
- concatenation $\rho.\rho$, choice $\rho + \rho$ and closures ρ^* and ρ^+

Laws

$$\begin{aligned}
\langle \rho_1 + \rho_2 \rangle \phi &= \langle \rho_1 \rangle \phi \vee \langle \rho_2 \rangle \phi \\
[\rho_1 + \rho_2]\phi &= [\rho_1]\phi \wedge [\rho_2]\phi \\
\langle \rho_1 . \rho_2 \rangle \phi &= \langle \rho_1 \rangle \langle \rho_2 \rangle \phi \\
[\rho_1 . \rho_2]\phi &= [\rho_1][\rho_2]\phi
\end{aligned}$$

Revisiting Hennessy-Milner logic

Examples of properties

- $\langle\epsilon\rangle\phi \;=\; [\epsilon]\phi \;=\; \phi$
- $\langle a.a.b\rangle\phi \;=\; \langle a\rangle\langle a\rangle\langle b\rangle\phi$
- $\langle a.b + g.d\rangle\phi$

Safety

- $[-^*]\phi$
- it is impossible to do two consecutive enter actions without a leave action in between:
  $[-^*.enter. - leave^*.enter]\text{false}$
- absence of deadlock:
  $[-^*]\langle -\rangle\text{true}$

Revisiting Hennessy-Milner logic

Examples of properties

Liveness

- $\langle -^* \rangle \phi$
- after sending a message, it can eventually be received: $[send]\langle -^*.receive\rangle\mathsf{true}$
- after a send a receive is possible as long as an exception does not happen: $[send.-excp^*]\langle -^*.receive\rangle\mathsf{true}$

The general case: Modal μ-calculus

Intuition

- look at modal formulas as set-theoretic combinators
- introduce mechanisms to specify their fixed points
- introduced as a generalisation of Hennessy-Milner logic for processes to capture enduring properties.

References

- Original reference: *Results on the propositional μ-calculus*, D. Kozen, 1983.
- Introductory text: *Modal and temporal logics for processes*, C. Stirling, 1996

The modal μ-calculus

- modalities with regular expressions are not enough in general
- ... but correspond to a subset of the modal μ-calculus [Kozen83]

Add explicit minimal/maximal fixed point operators to Hennessy-Milner logic

$$\phi \ ::= \ X \mid \mathsf{true} \mid \mathsf{false} \mid \neg\phi \mid \phi\wedge\phi \mid \phi\vee\phi \mid \phi\rightarrow\phi \mid \langle a\rangle\phi \mid [a]\phi \mid \mu X \,.\, \phi \mid \nu X \,.\, \phi$$

The modal μ-calculus

The modal μ-calculus (intuition)

- $\mu X \,.\, \phi$ is valid for all those states in the smallest set X that satisfies the equation $X = \phi$ (finite paths, liveness)
- $\nu X \,.\, \phi$ is valid for the states in the largest set X that satisfies the equation $X = \phi$ (infinite paths, safety)

Warning

In order to be sure that a fixed point exists, X must occur positively in the formula, ie preceded by an even number of negations.

Temporal properties as limits

Example

$$A \triangleq \sum_{i \geq 0} A_i \quad \text{with} \quad A_0 \triangleq \mathbf{0} \text{ e } A_{i+1} \triangleq a.A_i$$

$$A' \triangleq A + D \quad \text{with} \quad D \triangleq a.D$$

- $A \nsim A'$
- but there is no modal formula to distinguish A from A'
- notice $A' \models \langle a \rangle^{i+1}$true which A_i fails
- a distinguishing formula would require infinite conjunction
- what we want to express is the possibility of doing a in the long run

Temporal properties as limits

idea: introduce recursion in formulas

$$X \triangleq \langle a \rangle X$$

meaning?

- the recursive formula is interpreted as a fixed point of function

$$\| \langle a \rangle \|$$

  in $\mathcal{P}\mathbb{P}$

- i.e., the solutions, $S \subseteq \mathbb{P}$ such that of

$$S = \| \langle a \rangle \| (S)$$

- how do we solve this equation?

Solving equations ...

over natural numbers

$$\begin{aligned} x &= 3x && \text{one solution } (x = 0) \\ x &= 1 + x && \text{no solutions} \\ x &= 1x && \text{many solutions (every natural } x\text{)} \end{aligned}$$

over sets of integers

$$\begin{aligned} x &= \{22\} \cap x && \text{one solution } (x = \{22\}) \\ x &= \mathbb{N} \setminus x && \text{no solutions} \\ x &= \{22\} \cup x && \text{many solutions (every } x \text{ st } \{22\} \subseteq x\text{)} \end{aligned}$$

Solving equations ...

In general, for a monotonic function f, i.e.

$$X \subseteq Y \;\Rightarrow\; f\,X \subseteq f\,Y$$

Knaster-Tarski Theorem [1928]

A monotonic function f in a complete lattice has a

- unique maximal fixed point:

$$\nu_f \;=\; \bigcup\{X \in \mathcal{P}\mathbb{P} \mid X \subseteq f\,X\}$$

- unique minimal fixed point:

$$\mu_f \;=\; \bigcap\{X \in \mathcal{P}\mathbb{P} \mid f\,X \subseteq X\}$$

- moreover the space of its solutions forms a complete lattice

Back to the example ...

$S \in \mathcal{P}\mathbb{P}$ is a pre-fixed point of $\|\langle a\rangle\|$
iff

$$\|\langle a\rangle\|(S) \ \subseteq \ S$$

Recalling,

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

the set of sets of processes we are interested in is

$$\boxed{\begin{aligned} \text{Pre} \ &= \ \{S \subseteq \mathbb{P} \mid \{E \in \mathbb{P} \mid \exists_{E' \in S} \, . \, E \xrightarrow{a} E'\} \subseteq S\} \\ &= \ \{S \subseteq \mathbb{P} \mid \forall_{Z \in \mathbb{P}} \, . \, (Z \in \{E \in \mathbb{P} \mid \exists_{E' \in S} \, . \, E \xrightarrow{a} E'\} \Rightarrow Z \in S)\} \\ &= \ \{S \subseteq \mathbb{P} \mid \forall_{E \in \mathbb{P}} \, . \, ((\exists_{E' \in S} \, . \, E \xrightarrow{a} E') \Rightarrow E \in S)\} \end{aligned}}$$

which can be characterized by predicate

$$\text{(PRE)} \qquad (\exists_{E' \in S} \, . \, E \xrightarrow{a} E') \Rightarrow E \in S \qquad \text{(for all } E \in \mathbb{P}\text{)}$$

Back to the example ...

The set of pre-fixed points of

$$\|\langle a\rangle\|$$

is

$$\begin{aligned} \mathrm{Pre} &= \{S \subseteq \mathbb{P} \mid \|\langle a\rangle\|(S) \subseteq S\} \\ &= \{S \subseteq \mathbb{P} \mid \forall_{E \in \mathbb{P}} \,.\, ((\exists_{E' \in S} \,.\, E \xrightarrow{a} E') \Rightarrow E \in S)\} \end{aligned}$$

- Clearly, $\{A \triangleq a.A\} \in \mathrm{Pre}$
- but $\emptyset \in \mathrm{Pre}$ as well

Therefore, its least solution is

$$\bigcap \mathrm{Pre} = \emptyset$$

Conclusion: taking the meaning of $X = \langle a\rangle X$ as the least solution of the equation leads us to equate it to false

... but there is another possibility ...

$S \in \mathcal{P}\mathbb{P}$ is a post-fixed point of

$$\|\langle a \rangle\|$$

iff

$$S \;\subseteq\; \|\langle a \rangle\|(S)$$

leading to the following set of post-fixed points

$$\begin{aligned}
\text{Post} &= \{S \subseteq \mathbb{P} \mid S \subseteq \{E \in \mathbb{P} \mid \exists_{E' \in S} \,.\, E \xrightarrow{a} E'\}\} \\
&= \{S \subseteq \mathbb{P} \mid \forall_{Z \in \mathbb{P}} \,.\, (Z \in S \Rightarrow Z \in \{E \in \mathbb{P} \mid \exists_{E' \in S} \,.\, E \xrightarrow{a} E'\})\} \\
&= \{S \subseteq \mathbb{P} \mid \forall_{E \in \mathbb{P}} \,.\, (E \in S \Rightarrow \exists_{E' \in S} \,.\, E \xrightarrow{a} E')\}
\end{aligned}$$

(POST) If $E \in S$ then $E \xrightarrow{a} E'$ for some $E' \in S$ (for all $E \in P$)

- i.e., if $E \in S$ it can perform a and this ability is maintained in its continuation

... but there is another possibility ...

- i.e., if $E \in S$ it can perform a and this ability is maintained in its continuation
- the greatest subset of $\mathbb{P}$ verifying this condition is the set of processes with at least an infinite computation

Conclusion: taking the meaning of $X = \langle a \rangle X$ as the greatest solution of the equation characterizes the property occurrence of a is possible

The general case

- The meaning (i.e., set of processes) of a formula $X \triangleq \phi X$ where X occurs free in ϕ
- is a solution of equation

$$X = f(X) \qquad \text{with} \quad f(S) = \|\{S/X\}\phi\|$$

in $\mathcal{P}\mathbb{P}$, where $\|.\|$ is extended to formulae with variables by $\|X\| = X$

The general case

The Knaster-Tarski theorem gives precise characterizations of the

- smallest solution: the intersection of all S such that

$$\text{(PRE)} \qquad \text{If} \quad E \in f(S) \quad \text{then} \quad E \in S$$

  to be denoted by

$$\mu X \,.\, \phi$$

- greatest solution: the union of all S such that

$$\text{(POST)} \qquad \text{If} \quad E \in S \quad \text{then} \quad E \in f(S)$$

  to be denoted by

$$\nu X \,.\, \phi$$

In the previous example:

$$\nu X \,.\, \langle a \rangle \text{true} \qquad\qquad \mu X \,.\, \langle a \rangle \text{true}$$

The general case

The Knaster-Tarski theorem gives precise characterizations of the

- smallest solution: the intersection of all S such that

$$\text{(PRE)} \qquad \text{If} \quad E \in f(S) \quad \text{then} \quad E \in S$$

  to be denoted by

$$\mu X \,.\, \phi$$

- greatest solution: the union of all S such that

$$\text{(POST)} \qquad \text{If} \quad E \in S \quad \text{then} \quad E \in f(S)$$

  to be denoted by

$$\nu X \,.\, \phi$$

In the previous example:

$$\nu X \,.\, \langle a \rangle \mathsf{true} \qquad\qquad \mu X \,.\, \langle a \rangle \mathsf{true}$$

The modal μ-calculus: syntax

... Hennessy-Milner + recursion (i.e. fixed points):

$$\phi \ ::= \ X \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \langle K \rangle \phi \mid [K]\phi \mid \mu X \,.\, \phi \mid \nu X \,.\, \phi$$

where $K \subseteq Act$ and X is a set of propositional variables

- Note that

$$\text{true} \overset{\text{abv}}{=} \nu X \,.\, X \qquad \text{and} \qquad \text{false} \overset{\text{abv}}{=} \mu X \,.\, X$$

The modal μ-calculus: denotational semantics

- Presence of variables requires models parametric on valuations:

$$V : X \to \mathcal{P}\mathbb{P}$$

- Then,

$$\begin{aligned} \|X\|_V =& V(X) \\ \|\phi_1 \wedge \phi_2\|_V =& \|\phi_1\|_V \cap \|\phi_2\|_V \\ \|\phi_1 \vee \phi_2\|_V =& \|\phi_1\|_V \cup \|\phi_2\|_V \\ \|[K]\phi\|_V =& \|[K]\|(\|\phi\|_V) \\ \|\langle K\rangle\phi\|_V =& \|\langle K\rangle\|(\|\phi\|_V) \end{aligned}$$

- and add

$$\begin{aligned} \|\nu X\,.\,\phi\|_V =& \bigcup\{S \in \mathbb{P} \mid S \subseteq \|\{S/X\}\phi\|_V\} \\ \|\mu X\,.\,\phi\|_V =& \bigcap\{S \in \mathbb{P} \mid \|\{S/X\}\phi\|_V \subseteq S\} \end{aligned}$$

Notes

where

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

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

Modal μ-calculus

Intuition

- look at modal formulas as set-theoretic combinators
- introduce mechanisms to specify their fixed points
- introduced as a generalisation of Hennessy-Milner logic for processes to capture enduring properties.

References

- Original reference: *Results on the propositional μ-calculus*, D. Kozen, 1983.
- Introductory text: *Modal and temporal logics for processes*, C. Stirling, 1996

Notes

The modal μ-calculus [Kozen, 1983] is

- decidable
- strictly more expressive than PDL and CTL*

Moreover

- The correspondence theorem of the induced temporal logic with bisimilarity is kept

Example 1: $X \triangleq \phi \vee \langle a \rangle X$

Look for fixed points of

$$f(X) \triangleq \|\phi\| \cup \|\langle a \rangle\|(X)$$

Example 1: $X \triangleq \phi \vee \langle a \rangle X$

$$
\begin{aligned}
\text{(PRE)}\ & \text{If} \quad E \in f(X) \quad \text{then} \quad E \in X \\
\equiv\ & \text{If} \quad E \in (\|\phi\| \cup \|\langle a \rangle\|(X)) \quad \text{then} \quad E \in X \\
\equiv\ & \text{If} \quad E \in \{F \mid F \models \phi\} \cup \{F \in \mathbb{P} \mid \exists_{F' \in X} \,.\, F \xrightarrow{a} F'\} \\
& \qquad\quad \text{then} \quad E \in X \\
\equiv\ & \text{if} \quad E \models \phi \vee \exists_{E' \in X} \,.\, E \xrightarrow{a} E' \quad \text{then} \quad E \in X
\end{aligned}
$$

The smallest set of processes verifying this condition is composed of processes with at least a computation along which a can occur until ϕ holds. Taking its intersection, we end up with processes in which ϕ holds in a finite number of steps.

Example 1: $X \triangleq \phi \vee \langle a \rangle X$

$$
\begin{array}{llllll}
\text{(POST)} & \text{If} & E \in X & \text{then} & E \in f(X) \\
\equiv & \text{If} & E \in X & \text{then} & E \in (\|\phi\| \cup \|\langle a \rangle\|(X)) \\
\equiv & \text{If} & E \in X & \text{then} & E \in \{F \mid F \models \phi\} \cup \{F \in X \mid \exists_{F' \in X} \,.\, F \xrightarrow{a} F'\} \\
\equiv & \text{If} & E \in X & \text{then} & E \models \phi \vee \exists_{E' \in X} \,.\, E \xrightarrow{a} E'
\end{array}
$$

The greatest fixed point also includes processes which keep the possibility of doing a without ever reaching a state where ϕ holds.

Example 1: $X \triangleq \phi \vee \langle a \rangle X$

- strong until:

$$\mu X \,.\, \phi \vee \langle a \rangle X$$

- weak until

$$\nu X \,.\, \phi \vee \langle a \rangle X$$

Relevant particular cases:

- ϕ holds after internal activity:

$$\mu X \,.\, \phi \vee \langle \tau \rangle X$$

- ϕ holds in a finite number of steps

$$\mu X \,.\, \phi \vee \langle - \rangle X$$

Example 2: $X \triangleq \phi \wedge \langle a \rangle X$

(PRE) If $E \models \phi \wedge \exists_{E' \in X} . E \xrightarrow{a} E'$ then $E \in X$

implies that

$$\mu X . \phi \wedge \langle a \rangle X \Leftrightarrow \text{false}$$

(POST) If $E \in X$ then $E \models \phi \wedge \exists_{E' \in X} . E \xrightarrow{a} E'$

implies that

$$\nu X . \phi \wedge \langle a \rangle X$$

denote all processes which verify ϕ and have an infinite computation

Example 2: $X \triangleq \phi \wedge \langle a \rangle X$

Variant:

- ϕ holds along a finite or infinite *a*-computation:

$$\nu X \,.\, \phi \wedge (\langle a \rangle X \vee [a]\mathsf{false})$$

In general:

- weak safety:

$$\nu X \,.\, \phi \wedge (\langle K \rangle X \vee [K]\mathsf{false})$$

- weak safety, for $K = Act$:

$$\nu X \,.\, \phi \wedge (\langle - \rangle X \vee [-]\mathsf{false})$$

Example 3: $X \triangleq [-]X$

$$
\begin{array}{llllll}
\text{(POST)} & \text{If} & E \in X & \text{then} & E \in \|[-]\|(X) \\
\equiv & \text{If} & E \in X & \text{then} & (\text{if } E \xrightarrow{x} E' \text{ and } x \in Act \text{ then } E' \in X)
\end{array}
$$

implies $\nu X \,.\, [-]X \Leftrightarrow \text{true}$

$$
\text{(PRE)} \quad \text{If} \quad (\text{if } E \xrightarrow{x} E' \text{ and } x \in Act \text{ then } E' \in X) \quad \text{then} \quad E \in X
$$

implies $\mu X \,.\, [-]X$ represent finite processes (why?)

Safety and liveness

- weak liveness:

$$\mu X \,.\, \phi \vee \langle - \rangle X$$

- strong safety

$$\nu X \,.\, \psi \wedge [-] X$$

making $\psi = \neg\phi$ both properties are dual:

- there is at least a computation reaching a state s such that $s \models \phi$
- all states s reached along all computations maintain ϕ, ie, $s \models \neg\phi$

Safety and liveness

Qualifiers weak and strong refer to a quatification over computations

- weak liveness:

$$\mu X \,.\, \phi \vee \langle - \rangle X$$

(corresponds to Ctl formula `E F` ϕ)

- strong safety

$$\nu X \,.\, \psi \wedge [-] X$$

(corresponds to Ctl formula `A G` ψ)

cf, liner time vs branching time

Duality

$$\neg(\mu X \,.\, \phi) = \nu X \,.\, \neg\phi$$
$$\neg(\nu X \,.\, \phi) = \mu X \,.\, \neg\phi$$

Example:

- divergence:

$$\nu X \,.\, \langle\tau\rangle X$$

- convergence (= all non observable behaviour is finite)

$$\neg(\nu X \,.\, \langle\tau\rangle X) \;=\; \mu X \,.\, \neg(\langle\tau\rangle X) \;=\; \mu X \,.\, [\tau] X$$

Safety and liveness

- weak safety:

$$\nu X . \phi \wedge (\langle - \rangle X \vee [-]\mathsf{false})$$

(there is a computation along which ϕ holds)

- strong liveness

$$\mu X . \neg\phi \vee ([-]X \wedge \langle - \rangle \mathsf{true})$$

(a state where the complement of ϕ holds can be finitely reached)

Conditional properties

ϕ_1 =
After collecting a passenger (*icr*), the taxi drops him at destination (*fcr*)
Second part of ϕ_1 is strong liveness:

$$\mu X \,.\, [-fcr]X \wedge \langle - \rangle \mathrm{true}$$

holding only after *icr*.
Is it enough to write:

$$[icr](\mu X \,.\, [-fcr]X \wedge \langle - \rangle \mathrm{true})$$

?
what we want does not depend on the initial state: it is liveness embedded into strong safety:

$$\nu Y \,.\, [icr](\mu X \,.\, [-fcr]X \wedge \langle - \rangle \mathrm{true}) \;\wedge\; [-]Y$$

Conditional properties

ϕ_1 =
After collecting a passenger (*icr*), the taxi drops him at destination (*fcr*)
Second part of ϕ_1 is strong liveness:

$$\mu X \,.\, [-fcr]X \wedge \langle - \rangle \mathsf{true}$$

holding only after *icr*.
Is it enough to write:

$$[icr](\mu X \,.\, [-fcr]X \wedge \langle - \rangle \mathsf{true})$$

?
what we want does not depend on the initial state: it is liveness embedded into strong safety:

$$\nu Y \,.\, [icr](\mu X \,.\, [-fcr]X \wedge \langle - \rangle \mathsf{true}) \;\wedge\; [-]Y$$

Conditional properties

The previous example is conditional liveness but one can also have

- conditional safety:

$$\nu Y \,.\, (\neg\phi \vee (\phi \wedge \nu X \,.\, \psi \wedge [-]X)) \wedge [-]Y$$

(whenever ϕ holds, ψ cannot cease to hold)

Cyclic properties

ϕ = every second action is *out*
is expressed by

$$\nu X \,.\, [-]([-out]\mathsf{false} \wedge [-]X)$$

ϕ = *out* follows *in*, but other actions can occur in between

$$\nu X \,.\, [out]\mathsf{false} \wedge [in](\mu Y \,.\, [in]\mathsf{false} \wedge [out]X \wedge [-out]Y) \wedge [-in]X$$

Note that the use of least fixed points imposes that the amount of computation between *in* and *out* is finite

Cyclic properties

ϕ = a state in which *in* can occur, can be reached an infinite number of times

$$\nu X \,.\, \mu Y \,.\, (\langle in \rangle \mathsf{true} \vee \langle - \rangle Y) \;\wedge\; ([-]X \;\wedge\; \langle - \rangle \mathsf{true})$$

ϕ = *in* occurs an infinite number of times

$$\nu X \,.\, \mu Y \,.\, [-in]Y \wedge [-]X \wedge \langle - \rangle \mathsf{true}$$

ϕ = *in* occurs an finite number of times

$$\mu X \,.\, \nu Y \,.\, [-in]Y \wedge [in]X$$

μ-calculus in mCRL2

The verification problem

- Given a specification of the system's behaviour is in mCRL2
- and the system's requirements are specified as properties in a temporal logic,
- a model checking algorithm decides whether the property holds for the model: the property can be verified or refuted;
- sometimes, witnesses or counter examples can be provided

Which logic?

μ-calculus with data, time and regular expressions

Example: The dining philosophers problem

Formulas to verify Demo

- No deadlock (every philosopher holds a left fork and waits for a right fork (or vice versa):

```
[true*]<true>true
```

- No starvation (a philosopher cannot acquire 2 forks):

```
forall p:Phil.  [true*.!eat(p)*] <!eat(p)*.eat(p)>true
```

- A philosopher can only eat for a finite consecutive amount of time:

```
forall p:Phil.  nu X. mu Y. [eat(p)]Y && [!eat(p)]X
```

- there is no starvation: for all reachable states it should be possible to eventually perform an `eat(p)` for each possible value of `p:Phil`.

```
[true*](forall p:Phil.  mu Y. ([!eat(p)]Y && <true>true))
```