Interacção e Concorrência 2016/17
Bloco de acetatos 8

Alexandre Madeira
(based on Luís S. Barbosa 2014/15 course Slides)
HASLab INESC TEC, DI UMINHO

April 26, 2017

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?

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)

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)

Signatures

Signatures are pairs $(\mathrm{PROP}, \mathrm{MOD})$ where PROP and MOD are sets of **propositional** symbols and **modality** symbols.

The language

Signatures

Signatures are pairs $(\mathrm{PROP}, \mathrm{MOD})$ where PROP and MOD are sets of **propositional** symbols and **modality** symbols.

Formulas

$$\phi \ ::= \ p \mid \mathbf{tt} \mid \mathbf{ff} \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \to \phi_2 \mid \langle m \rangle \phi \mid [m] \phi$$

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

The language

Signatures

Signatures are pairs $(\text{PROP}, \text{MOD})$ where PROP and MOD are sets of **propositional** symbols and **modality** symbols.

Formulas

$$\phi \ ::= \ p \mid \mathbf{tt} \mid \mathbf{ff} \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \to \phi_2 \mid \langle m \rangle \, \phi \mid [m] \, \phi$$

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

Disjunction ($\vee$) and equivalence ($\leftrightarrow$) are defined by abbreviation.

Notes

- if there is only one modality in the signature (i.e., MOD is a singleton), write simply $\diamond\phi$ and

The language

Notes

- if there is only one modality in the signature (i.e., MOD is a singleton), write simply $\diamond\phi$ and
- the language has some redundancy: in particular modal connectives are **dual** (as quantifiers are in first-order logic):

  $[m]\,\phi$ is equivalent to $\neg\langle m\rangle\,\neg\phi$

The language

Semantics

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

- $\mathcal{F} = \langle W, \{R_m\}_{m \in \mathrm{MOD}} \rangle$ is a Kripke frame, ie,
  - W is a a non empty set (of **states** or **worlds**)
  - $\{R_m\}_{m \in \mathrm{MOD}}$ is a family of binary relations $R_m \subseteq W \times W$, for each modality symbol $m \in \mathrm{MOD}$.
- $V : \mathrm{PROP} \to \mathcal{P}(W)$ is a valuation.

The language

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

$\mathcal{M}, w \models \mathbf{tt}$		
$\mathcal{M}, w \not\models \mathbf{ff}$		
$\mathcal{M}, w \models p$	iff	$w \in V(p)$
$\mathcal{M}, w \models \neg\phi$	iff	$\mathcal{M}, w \not\models \phi$
$\mathcal{M}, w \models \phi_1 \wedge \phi_2$	iff	$\mathcal{M}, w \models \phi_1$ and $\mathcal{M}, w \models \phi_2$
$\mathcal{M}, w \models \phi_1 \to \phi_2$	iff	$\mathcal{M}, w \not\models \phi_1$ or $\mathcal{M}, w \models \phi_2$
$\mathcal{M}, w \models \langle m \rangle \phi$	iff	there exists $v \in W$ st $wR_m v$ and $\mathcal{M}, v \models \phi$
$\mathcal{M}, w \models [m]\phi$	iff	for all $v \in W$ st $wR_m v$ and $\mathcal{M}, v \models \phi$

The language

Safistaction

A formula ϕ is

- **satisfiable in a model** $\mathcal{M}$ if it is satisfied at some point of $\mathcal{M}$
- **globally satisfied** in $\mathcal{M}$ ($\mathcal{M} \models \phi$) if it is satisfied at all points in $\mathcal{M}$
- **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 $\mathcal{M}$ and all points w, if $\mathcal{M}, w \models \Gamma$ then $\mathcal{M}, w \models \phi$

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

$\mathcal{M}, w \models \phi \,\mathcal{U}\, \psi$ iff

there exists $v \in W$ such that $(w, v) \in R$ and $\mathcal{M}, v \models \psi$, and for all $u \in W$ such that $(w, u) \in R$ and $(u, v) \in R$ one has $\mathcal{M}, u \models \phi$

Examples: Temporal logics with $\mathcal{U}$ and $\mathcal{S}$

$\mathcal{M}, w \models \phi \mathcal{U} \psi$ iff

there exists $v \in W$ such that $(w, v) \in R$ and $\mathcal{M}, v \models \psi$, and for all $u \in W$ such that $(w, u) \in R$ and $(u, v) \in R$ one has $\mathcal{M}, u \models \phi$

$\mathcal{M}, w \models \phi \mathcal{S} \psi$ iff

there exists a $v \in W$ such that $(v, w) \in R$ and $\mathcal{M}, v \models \psi$ and, for all u such that $(v, u) \in R$ and $(u, w) \in R$ one has $\mathcal{M}, u \models \phi$

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

Examples

Process logic (**Hennessy-Milner logic**)

- $\mathsf{PROP} = \emptyset$

Examples

Process logic (**Hennessy-Milner logic**)

- $\mathrm{PROP} = \emptyset$
- $W = \mathbb{P}$ is a set of states, typically process terms, in a labelled transition system

Examples

Process logic (**Hennessy-Milner logic**)

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

Process logic (**Hennessy-Milner logic**)

- 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 $\mathcal{F} = \langle \mathbb{P}, \{p \xrightarrow{K} p' \mid K \subseteq Act\}\rangle$ as the modal frame, satisfaction is abbreviated as

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

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

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*

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*
    - $\phi_0 = \langle rec, alo \rangle\, \mathbf{tt}$

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*
    - $\phi_0 = \langle rec, alo \rangle \, \mathbf{tt}$
- $\phi_1 =$ *This applies only to cars already on service*

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*
  - $\phi_0 = \langle rec, alo \rangle\, \mathbf{tt}$
- $\phi_1 =$ *This applies only to cars already on service*
  - $\phi_1 = [onservice]\, \langle rec, alo \rangle\, \mathbf{tt}$ or
    $\phi_1 = [onservice]\, \phi_0$

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*
  - $\phi_0 = \langle rec, alo \rangle\, \mathbf{tt}$
- $\phi_1 =$ *This applies only to cars already on service*
  - $\phi_1 = [onservice]\, \langle rec, alo \rangle\, \mathbf{tt}$ or
    $\phi_1 = [onservice]\, \phi_0$
- $\phi_2 =$ *If a car is allocated to a service, it must first collect the passenger and then plan the route*

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*
  - $\phi_0 = \langle rec, alo \rangle \, \mathbf{tt}$
- $\phi_1 =$ *This applies only to cars already on service*
  - $\phi_1 = [onservice] \langle rec, alo \rangle \, \mathbf{tt}$ or
    $\phi_1 = [onservice] \, \phi_0$
- $\phi_2 =$ *If a car is allocated to a service, it must first collect the passenger and then plan the route*
  - $\phi_2 = [alo] \langle rec \rangle \langle plan \rangle \, \mathbf{tt}$

Process logic: The taxi network example

- $\phi_3 =$ *On detecting an emergence the taxi becomes inactive*

Process logic: The taxi network example

- $\phi_3 =$ *On detecting an emergence the taxi becomes inactive*
  - $\phi_3 = \; [sos]\,[-]\,\mathbf{ff}$

Process logic: The taxi network example

- $\phi_3 =$ *On detecting an emergence the taxi becomes inactive*
  - $\phi_3 = [sos]\,[-]\,\mathbf{ff}$
- $\phi_4 =$ *A car on service is not inactive*

Process logic: The taxi network example

- $\phi_3 =$ *On detecting an emergence the taxi becomes inactive*
  - $\phi_3 = [sos]\,[-]\,\mathbf{ff}$
- $\phi_4 =$ *A car on service is not inactive*
  - $\phi_4 = [onservice]\,\langle - \rangle\,\mathbf{tt}$

Process logic: typical properties

- **inevitability of** a: $\langle - \rangle\, \mathbf{tt} \wedge [-a]\, \mathbf{ff}$

Process logic: typical properties

- **inevitability of a**: $\langle - \rangle\, \mathbf{tt} \wedge [-a]\, \mathbf{ff}$
- **progress**: $\langle - \rangle\, \mathbf{tt}$

Process logic: typical properties

- **inevitability of** a: $\langle - \rangle\, \mathbf{tt} \wedge [-a]\, \mathbf{ff}$
- **progress**: $\langle - \rangle\, \mathbf{tt}$
- **deadlock or termination**: $[-]\, \mathbf{ff}$
- satisfaction decided by unfolding the definition of $\models$: no need to compute the **transition graph**

Hennessy-Milner logic

... propositional logic with action modalities

Syntax

$$\phi \ ::= \ \mathbf{tt} \ | \ \mathbf{ff} \ | \ \phi_1 \wedge \phi_2 \ | \ \phi_1 \vee \phi_2 \ | \ \langle K \rangle \, \phi \ | \ [K] \, \phi$$

Semantics: $E \models \phi$

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

Example

$$Sem =^{df} get.put.Sem$$
$$P_i =^{df} \overline{get}.c_i.\overline{put}.P_i$$
$$S =^{df} (Sem \mid (|_{i \in I} P_i)) \setminus \{get, put\}$$

- $Sem \models \langle get \rangle \mathbf{tt}$

Example

$$Sem =^{df} get.put.Sem$$
$$P_i =^{df} \overline{get}.c_i.\overline{put}.P_i$$
$$S =^{df} (Sem \mid (|_{i \in I}\ P_i)) \setminus \{get, put\}$$

- $Sem \models \langle get \rangle \mathbf{tt}$ holds because

$$\exists_{F \in \{Sem' \mid Sem \xrightarrow{get} Sem'\}} . F \models \mathbf{tt}$$

with $F = put.Sem$.

Example

$$Sem =^{df} get.put.Sem$$
$$P_i =^{df} \overline{get}.c_i.\overline{put}.P_i$$
$$S =^{df} (Sem \mid (|_{i\in I} P_i)) \setminus \{get, put\}$$

- $Sem \models \langle get\rangle\,\mathbf{tt}$ holds because

$$\exists_{F\in\{Sem' \mid Sem \xrightarrow{get} Sem'\}} \,.\, F \models \mathbf{tt}$$

  with $F = put.Sem$.

- $Sem \models [put]\,\mathbf{ff}$

Example

$$Sem =^{df} get.put.Sem$$
$$P_i =^{df} \overline{get}.c_i.\overline{put}.P_i$$
$$S =^{df} (Sem \mid (|_{i \in I}\ P_i)) \setminus \{get, put\}$$

- $Sem \models \langle get \rangle \mathbf{tt}$ holds because

  $$\exists_{F \in \{Sem' \mid Sem \xrightarrow{get} Sem'\}} \,.\, F \models \mathbf{tt}$$

  with $F = put.Sem$.

- $Sem \models [put]\,\mathbf{ff}$ also holds, because
  $T = \{Sem' \mid Sem \xrightarrow{put} Sem'\} = \emptyset$.
  Hence $\forall_{F \in T} \,.\, F \models \mathbf{ff}$ becomes trivially true.

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

Example

$$Sem =^{df} get.put.Sem$$
$$P_i =^{df} \overline{get}.c_i.\overline{put}.P_i$$
$$S =^{df} (Sem \mid (|_{i \in I} P_i)) \setminus \{get, put\}$$

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

Exercise

Verify:

$$\begin{aligned}
\neg\langle a\rangle\,\phi &= [a]\,\neg\phi \\
\neg[a]\,\phi &= \langle a\rangle\,\neg\phi \\
\langle a\rangle\,\mathbf{ff} &= \mathbf{ff} \\
[a]\,\mathbf{tt} &= \mathbf{tt} \\
\langle a\rangle\,(\phi\vee\psi) &= \langle a\rangle\,\phi\vee\langle a\rangle\,\psi \\
[a]\,(\phi\wedge\psi) &= [a]\,\phi\wedge[a]\,\psi \\
\langle a\rangle\,\phi\wedge[a]\,\psi &\Rightarrow \langle a\rangle\,(\phi\wedge\psi)
\end{aligned}$$

Exercise

Formalise each of the following properties:

1. The occurrence of a and b is impossible.
2. The occurrence of a followed by b is impossible.
3. Only the occurrence of a is possible.
4. Once a occurred, b or c may occur.
5. After a occurred followed by b, c may occur.
6. Once a occurred, b or c may occur but not both.
7. a cannot occur before b.
8. There is only an initial transition labelled by a.

Exercise

Consider the following processes and enumerate for each of them the properties they verify:

1. $E_1 =^{df} a.b.\mathbf{0}$
2. $E_2 =^{df} a.c.\mathbf{0}$
3. $E =^{df} E_1 + E_2$
4. $F =^{df} a.(b.\mathbf{0} + c.\mathbf{0})$
5. $G =^{df} E + F$

Exercise

Specify a LTS such that the following modal properties hold simultaneously in its initial state:

- $\langle a \rangle \langle b \rangle \langle c \rangle \mathbf{tt} \wedge \langle c \rangle \mathbf{tt}$
- $\langle a \rangle \langle b \rangle ([a]\,\mathbf{ff} \wedge [c]\,\mathbf{ff} \wedge [b]\,\mathbf{ff})$
- $\langle a \rangle \langle b \rangle (\langle a \rangle \mathbf{tt} \wedge [c]\,\mathbf{ff})$

Exercise

Consider the following specification of a CNC program:

$$\begin{aligned} \mathit{Start} &=^{df} \mathit{fw}.\mathit{Go} + \mathit{stop}.\mathbf{0} \\ \mathit{Go} &=^{df} \mathit{fw}.\mathit{bk}.\mathit{bk}.\mathit{Start} + \mathit{right}.\mathit{left}.\mathit{bk}.\mathit{Start} \end{aligned}$$

Formalise the following properties:

1. After *fw* another *fw* is immediately possible
2. After *fw* followed by *right*, *left* is possible but *bk* is not.
3. Action *fw* is the only one initially possible
4. The third action of process *Start* is not *fw*.

Hennessy-Milner with regular modalities

As in mCRL2, we can enrich modalities with regular expressions of modal symbols:

$$\alpha := K \mid K \cup K \mid K \cap K$$

for $K \subseteq A$.

Hennessy-Milner with regular modalities

As in mCRL2, we can enrich modalities with regular expressions of modal symbols:

$$\alpha := K \mid K \cup K \mid K \cap K$$

for $K \subseteq A$. As above we represent

- the set A with $-$
- the set $A \setminus \{a\}$ with $-a$

Regular modalities

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

Hennessy-Milner with regular modalities

interpretation of regular modalities

- $\langle R_1 + R_2 \rangle true = \langle R_1 \rangle true \vee \langle R_2 \rangle true$
  $[R_1 + R_2] true = [R_1] true \vee [R_2] true$
- $\langle R_1 . R_2 \rangle true = \langle R_1 \rangle \langle R_2 \rangle true$
  $[R_1 . R_2] true = [R_1][R_2] true$

Hennessy-Milner with regular modalities

- As long as no *error* happens, a deadlock will not occur.

$$[(-\mathit{error})^*]\langle - \rangle \mathbf{tt}$$

Hennessy-Milner with regular modalities

- As long as no *error* happens, a deadlock will not occur.

$$[(-error)^*]\langle - \rangle \mathbf{tt}$$

- Whenever an a can happen in any reachable state, a b action can subsequently be done unless a c happens cancelling the need to do the b.

$$[-^*.a]\langle -^*.(b \cup c)\rangle \mathbf{tt}$$

Hennessy-Milner with regular modalities

- As long as no *error* happens, a deadlock will not occur.

$$[(-error)^*]\langle - \rangle \mathbf{tt}$$

- Whenever an a can happen in any reachable state, a b action can subsequently be done unless a c happens cancelling the need to do the b.

$$[-^*.a]\langle -^*.(b \cup c)\rangle \mathbf{tt}$$

- Whenever an a action happens, it must always be possible to do a b after that, although doing the b can infinitely be postponed.

$$[-^*.a.(-b)^*]\langle -^*.b\rangle \mathbf{tt}$$

A denotational semantics

$\boxed{\textbf{Idea}\text{: associate to each formula } \phi \text{ the set of processes that makes it true}}$

A denotational semantics

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

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

A denotational semantics

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

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

$$\begin{aligned} \|\mathbf{tt}\| &= \mathbb{P} \\ \|\mathbf{ff}\| &= \emptyset \\ \|\phi_1 \wedge \phi_2\| &= \|\phi_1\| \cap \|\phi_2\| \\ \|\phi_1 \vee \phi_2\| &= \|\phi_1\| \cup \|\phi_2\| \end{aligned}$$

A denotational semantics

$\boxed{\textbf{Idea}\text{: associate to each formula } \phi \text{ the set of processes that makes it true}}$

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

$$\begin{aligned}
\|\mathbf{tt}\| &= \mathbb{P} \\
\|\mathbf{ff}\| &= \emptyset \\
\|\phi_1 \wedge \phi_2\| &= \|\phi_1\| \cap \|\phi_2\| \\
\|\phi_1 \vee \phi_2\| &= \|\phi_1\| \cup \|\phi_2\|
\end{aligned}$$

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

$\|[K]\|$ 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:

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

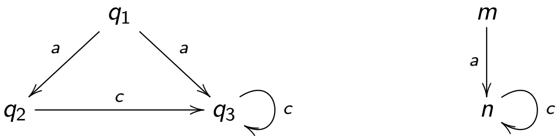
$$\|\langle K\rangle\|(X) \;=\; \{F \in \mathbb{P} \mid \exists_{F' \in X, a \in K} \,.\, F \xrightarrow{a} F'\}$$

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: $\mathbf{0} \models [-]\,\mathbf{ff}$

A denotational semantics

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

Example: $\mathbf{0} \models [-]\,\mathbf{ff}$

because

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

A denotational semantics

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

Example: $?? \models \langle - \rangle \mathbf{tt}$

A denotational semantics

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

Example: $?? \models \langle - \rangle\, \mathbf{tt}$

because

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

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:

$$\begin{aligned} \mathbf{tt}^c &= \mathbf{ff} \quad \mathbf{ff}^c = \mathbf{tt} \\ (\phi_1 \wedge \phi_2)^c &= \phi_1^c \vee \phi_2^c \\ (\phi_1 \vee \phi_2)^c &= \phi_1^c \wedge \phi_2^c \\ (\langle a \rangle \, \phi)^c &= [a] \, \phi^c \end{aligned}$$

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

Exercise

Compute

1. $\|\langle a \rangle \langle - \rangle \mathbf{tt}\|$
2. $\|[a] \langle - \rangle \mathbf{tt} \wedge [b] [-] \mathbf{ff}\|$
3. $\|[a] \langle - \rangle \mathbf{tt} \vee [b] [-] \mathbf{ff}\|$

Modal Equivalence

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

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

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 $\Gamma \;=\; \{\langle x_1 \rangle \, \langle x_2 \rangle \ldots \langle x_n \rangle \, \mathbf{tt} \mid x_i \in Act\}$

Modal Equivalence

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

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

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

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

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

- $A =^{df} \sum_{i \geq 0} A_i$, where $A_0 =^{df} \mathbf{0}$ and $A_{i+1} =^{df} a.A_i$
- $A' =^{df} A + K,\ K = a.K$

$$A \not\sim A' \quad \text{but} \quad A \simeq A'$$

Modal Equivalence

Theorem [Hennessy-Milner, 1985]

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

for image-finite processes.

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]

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

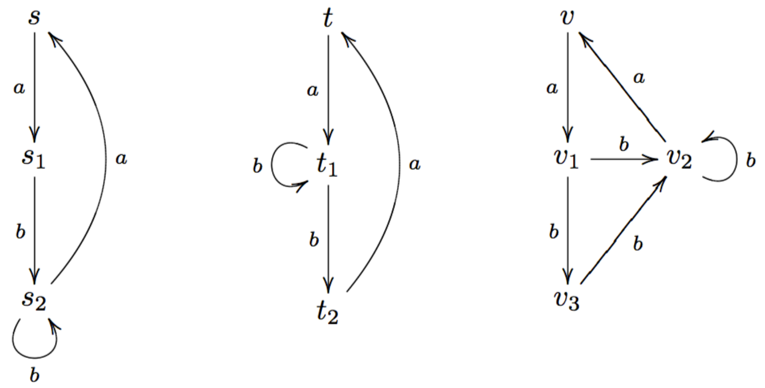
for image-finite processes.

proof

$\Rightarrow$: by induction of the formula structure

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

Exercise



Show that states s, t and v are not bisimilar and determine the modal properties which distinguish between them.

Exercise

Consider processes $E =^{df} a.(b.\mathbf{0} + c.\mathbf{0})$ e $F =^{df} a.b.\mathbf{0} + a.c.\mathbf{0}$.
Propose a formula ϕ in $\mathcal{M}$ valid in E but false in F.

Exercise

Let E be a process. A formula ϕ is said to be *characteristic* of E iff

$$\forall_{F \in \mathbb{P}} \, . \, F \models \phi \;\; sse \;\; F \sim E$$

Note that a process verifies the characteristic formula of E iff it is strongly bisimilar to E.
Determine the *characteristic* formula of process $x.0$.

Exercise

Consider processes below and write down a formula in $\mathcal{M}$ valid in R but not in S.

$$E =^{df} b.c.\mathbf{0} + b.d.\mathbf{0} \quad (1)$$

$$F =^{df} E + b.(c.\mathbf{0} + d.\mathbf{0}) \quad (2)$$

$$R =^{df} a.E + a.F \quad (3)$$

$$S =^{df} a.F \quad (4)$$

Exercise

In general, parallel composite in process algebra fails to be idempotent.

1. Making $E =^{df} a.b.E$, formalise a property in $\mathcal{M}$ to distinguish between E and $E \mid E$.
2. In some cases idempotency holds. Build a bissimulation to witness equivalence $E \sim E \mid E$ when E is $E =^{df} \sum_{x \in K} x.E$, for any $K \subseteq Act - \{\tau\}$. Would this remain true for Act?