Bisimulation and Logic
Lecture 3

Colin Stirling

Laboratory for Foundations of Computer Science (LFCS)
School of Informatics
Edinburgh University

Summer School on Model Checking
Ziyu Hotel, Beijing
Oct 11–16 2010

Modal logic and bisimulation

- Behavioural equivalence between concurrent processes (Park, Hennessy + Milner)
- Model theory of modal logic (van Benthem)

Modal characterisation of bisimulation and some model theory

Modal (Hennessy-Milner) logic: syntax

$$\Phi ::= \texttt{tt} \mid \texttt{ff} \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [a]\Phi \mid \langle a \rangle \Phi$$

A formula can be

Modal (Hennessy-Milner) logic: syntax

$$\Phi ::= \texttt{tt} \mid \texttt{ff} \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [a]\Phi \mid \langle a \rangle \Phi$$

A formula can be

- the constant true formula $\texttt{tt}$
- the constant false formula $\texttt{ff}$,

Modal (Hennessy-Milner) logic: syntax

$$\Phi ::= \mathtt{tt} \mid \mathtt{ff} \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [a]\Phi \mid \langle a \rangle \Phi$$

A formula can be

- the constant true formula $\mathtt{tt}$
- the constant false formula $\mathtt{ff}$,
- a conjunction of formulas $\Phi_1 \wedge \Phi_2$
- a disjunction of formulas $\Phi_1 \vee \Phi_2$,

Modal (Hennessy-Milner) logic: syntax

$$\Phi ::= \texttt{tt} \mid \texttt{ff} \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [a]\Phi \mid \langle a \rangle \Phi$$

A formula can be

- the constant true formula $\texttt{tt}$
- the constant false formula $\texttt{ff}$,
- a conjunction of formulas $\Phi_1 \wedge \Phi_2$
- a disjunction of formulas $\Phi_1 \vee \Phi_2$,
- a formula $[a]\Phi$, read as "box *a* Φ", or "for all *a*-derivatives Φ,"

Modal (Hennessy-Milner) logic: syntax

$$\Phi ::= \texttt{tt} \mid \texttt{ff} \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [a]\Phi \mid \langle a \rangle \Phi$$

A formula can be

- the constant true formula $\texttt{tt}$
- the constant false formula $\texttt{ff}$,
- a conjunction of formulas $\Phi_1 \wedge \Phi_2$
- a disjunction of formulas $\Phi_1 \vee \Phi_2$,
- a formula $[a]\Phi$, read as "box a Φ", or "for all a-derivatives Φ,"
- a formula $\langle a \rangle \Phi$, read as "diamond a Φ", or "for some a-derivative Φ."

Modal (Hennessy-Milner) logic: semantics

We define when a process E satisfies a formula Φ. Either E satisfies Φ, denoted by $E \models \Phi$, or it doesn't, denoted by $E \not\models \Phi$.

- $E \models \mathtt{tt} \qquad E \not\models \mathtt{ff}$

Modal (Hennessy-Milner) logic: semantics

We define when a process E satisfies a formula Φ. Either E satisfies Φ, denoted by $E \models \Phi$, or it doesn't, denoted by $E \not\models \Phi$.

- $E \models \texttt{tt}$ $\quad$ $E \not\models \texttt{ff}$
- $E \models \Phi \wedge \Psi$ iff $E \models \Phi$ and $E \models \Psi$
- $E \models \Phi \vee \Psi$ iff $E \models \Phi$ or $E \models \Psi$

Modal (Hennessy-Milner) logic: semantics

We define when a process E satisfies a formula Φ. Either E satisfies Φ, denoted by $E \models \Phi$, or it doesn't, denoted by $E \not\models \Phi$.

- $E \models \texttt{tt} \qquad E \not\models \texttt{ff}$
- $E \models \Phi \wedge \Psi$ iff $E \models \Phi$ and $E \models \Psi$
- $E \models \Phi \vee \Psi$ iff $E \models \Phi$ or $E \models \Psi$
- $E \models [a]\Phi$ iff $\forall F.$ if $E \xrightarrow{a} F$ then $F \models \Phi$

Modal (Hennessy-Milner) logic: semantics

We define when a process E satisfies a formula Φ. Either E satisfies Φ, denoted by $E \models \Phi$, or it doesn't, denoted by $E \not\models \Phi$.

- $E \models \texttt{tt} \qquad E \not\models \texttt{ff}$
- $E \models \Phi \wedge \Psi$ iff $E \models \Phi$ and $E \models \Psi$
- $E \models \Phi \vee \Psi$ iff $E \models \Phi$ or $E \models \Psi$
- $E \models [a]\Phi$ iff $\forall F.$ if $E \xrightarrow{a} F$ then $F \models \Phi$
- $E \models \langle a \rangle \Phi$ iff $\exists F.\ E \xrightarrow{a} F$ and $F \models \Phi$

Examples

- $E \models \langle \mathtt{tick} \rangle \mathtt{tt}$
  E can do a $\mathtt{tick}$

Examples

- $E \models \langle \texttt{tick} \rangle \texttt{tt}$
  E can do a `tick`
- $E \models \langle \texttt{tick} \rangle \langle \texttt{tock} \rangle \texttt{tt}$
  E can do a `tick` and then a `tock`

Examples

- $E \models \langle \texttt{tick} \rangle \texttt{tt}$
  E can do a `tick`
- $E \models \langle \texttt{tick} \rangle \langle \texttt{tock} \rangle \texttt{tt}$
  E can do a `tick` and then a `tock`
- $E \models [\texttt{tick}] \texttt{ff}$
  E cannot do a `tick`

Examples

- $E \models \langle \texttt{tick} \rangle \texttt{tt}$
  E can do a `tick`
- $E \models \langle \texttt{tick} \rangle \langle \texttt{tock} \rangle \texttt{tt}$
  E can do a `tick` and then a `tock`
- $E \models [\texttt{tick}] \texttt{ff}$
  E cannot do a `tick`
- $E \models \langle \texttt{tick} \rangle \texttt{ff}$
  This is equivalent to `ff`!

Examples

- $E \models \langle \mathtt{tick} \rangle \mathtt{tt}$
  E can do a `tick`
- $E \models \langle \mathtt{tick} \rangle \langle \mathtt{tock} \rangle \mathtt{tt}$
  E can do a `tick` and then a `tock`
- $E \models [\mathtt{tick}] \mathtt{ff}$
  E cannot do a `tick`
- $E \models \langle \mathtt{tick} \rangle \mathtt{ff}$
  This is equivalent to `ff`!
- $E \models [\mathtt{tick}] \mathtt{tt}$
  This is equivalent to `true`!

Checking satisfaction

$$\mathtt{Cl} \stackrel{\text{def}}{=} \mathtt{tick}.\mathtt{Cl}$$

Does $\mathtt{Cl}$ have the property: $[\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$?

Checking satisfaction

$$\mathtt{Cl} \stackrel{\text{def}}{=} \mathtt{tick}.\mathtt{Cl}$$

Does $\mathtt{Cl}$ have the property: $[\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$?

- $\mathtt{Cl} \models [\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$

Checking satisfaction

$$\mathtt{Cl} \stackrel{\mathrm{def}}{=} \mathtt{tick}.\mathtt{Cl}$$

Does $\mathtt{Cl}$ have the property: $[\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$?

- $\mathtt{Cl} \models [\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$
- iff $\forall F$ if $\mathtt{Cl} \xrightarrow{\mathtt{tick}} F$ then $F \models \langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff}$

Checking satisfaction

$$\mathtt{Cl} \stackrel{\mathrm{def}}{=} \mathtt{tick}.\mathtt{Cl}$$

Does $\mathtt{Cl}$ have the property: $[\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$?

- $\mathtt{Cl} \models [\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$
- iff $\forall F$ if $\mathtt{Cl} \xrightarrow{\mathtt{tick}} F$ then $F \models \langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff}$
- iff $\mathtt{Cl} \models \langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff}$

Checking satisfaction

$$\texttt{Cl} \stackrel{\text{def}}{=} \texttt{tick.Cl}$$

Does $\texttt{Cl}$ have the property: $[\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff})$?

- $\texttt{Cl} \models [\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff})$
- iff $\forall F$ if $\texttt{Cl} \xrightarrow{\texttt{tick}} F$ then $F \models \langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt}$ and $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$

Checking satisfaction

$$\texttt{Cl} \stackrel{\text{def}}{=} \texttt{tick}.\texttt{Cl}$$

Does $\texttt{Cl}$ have the property: $[\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff})$?

- $\texttt{Cl} \models [\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff})$
- iff $\forall F$ if $\texttt{Cl} \xrightarrow{\texttt{tick}} F$ then $F \models \langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt}$ and $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$
- iff $\exists F.\ \texttt{Cl} \xrightarrow{\texttt{tick}} F$ and $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$

Checking satisfaction

$$\texttt{Cl} \stackrel{\text{def}}{=} \texttt{tick.Cl}$$

Does $\texttt{Cl}$ have the property: $[\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff})$?

- $\texttt{Cl} \models [\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff})$
- iff $\forall F$ if $\texttt{Cl} \xrightarrow{\texttt{tick}} F$ then $F \models \langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt}$ and $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$
- iff $\exists F.\ \texttt{Cl} \xrightarrow{\texttt{tick}} F$ and $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$

Checking satisfaction

$$\mathtt{Cl} \stackrel{\text{def}}{=} \mathtt{tick}.\mathtt{Cl}$$

Does $\mathtt{Cl}$ have the property: $[\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$?

- $\mathtt{Cl} \models [\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$
- iff $\forall F$ if $\mathtt{Cl} \xrightarrow{\mathtt{tick}} F$ then $F \models \langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff}$
- iff $\mathtt{Cl} \models \langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff}$
- iff $\mathtt{Cl} \models \langle\mathtt{tick}\rangle\mathtt{tt}$ and $\mathtt{Cl} \models [\mathtt{tock}]\mathtt{ff}$
- iff $\exists F.\ \mathtt{Cl} \xrightarrow{\mathtt{tick}} F$ and $\mathtt{Cl} \models [\mathtt{tock}]\mathtt{ff}$
- iff $\mathtt{Cl} \models [\mathtt{tock}]\mathtt{ff}$

Checking satisfaction

$$\mathtt{Cl} \stackrel{\text{def}}{=} \mathtt{tick}.\mathtt{Cl}$$

Does $\mathtt{Cl}$ have the property: $[\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$?

- $\mathtt{Cl} \models [\mathtt{tick}](\langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff})$
- iff $\forall F$ if $\mathtt{Cl} \xrightarrow{\mathtt{tick}} F$ then $F \models \langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff}$
- iff $\mathtt{Cl} \models \langle\mathtt{tick}\rangle\mathtt{tt} \wedge [\mathtt{tock}]\mathtt{ff}$
- iff $\mathtt{Cl} \models \langle\mathtt{tick}\rangle\mathtt{tt}$ and $\mathtt{Cl} \models [\mathtt{tock}]\mathtt{ff}$
- iff $\exists F.\ \mathtt{Cl} \xrightarrow{\mathtt{tick}} F$ and $\mathtt{Cl} \models [\mathtt{tock}]\mathtt{ff}$
- iff $\mathtt{Cl} \models [\mathtt{tock}]\mathtt{ff}$
- iff $\{E : \mathtt{Cl} \xrightarrow{\mathtt{tock}} E\} = \emptyset$

Checking satisfaction

$$\texttt{Cl} \stackrel{\text{def}}{=} \texttt{tick}.\texttt{Cl}$$

Does $\texttt{Cl}$ have the property: $[\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff})$?

- $\texttt{Cl} \models [\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff})$
- iff $\forall F$ if $\texttt{Cl} \xrightarrow{\texttt{tick}} F$ then $F \models \langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt} \wedge [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt}$ and $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$
- iff $\exists F.\ \texttt{Cl} \xrightarrow{\texttt{tick}} F$ and $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$
- iff $\{E \,:\, \texttt{Cl} \xrightarrow{\texttt{tock}} E\} = \emptyset$
- iff $\emptyset = \emptyset$

Negation

Modal logic can be extended with a negation operator $\neg$ having the semantics: $E \models \neg\Phi \ \text{ iff } \ E \not\models \Phi$

Negation

Modal logic can be extended with a negation operator $\neg$ having the semantics: $E \models \neg\Phi \;\text{ iff }\; E \not\models \Phi$

Negation is redundant in the following sense: For every formula Φ of HML there is a formula Φ^c such that for every process E

$$E \models \Phi^c \;\text{ iff }\; E \not\models \Phi$$

Negation

Modal logic can be extended with a negation operator $\neg$ having the semantics: $E \models \neg\Phi \text{ iff } E \not\models \Phi$

Negation is redundant in the following sense: For every formula Φ of HML there is a formula Φ^c such that for every process E

$$E \models \Phi^c \quad \text{iff} \quad E \not\models \Phi$$

Φ^c is inductively defined as follows:

$$\begin{array}{rcl}
\mathtt{tt}^c & = & \mathtt{ff} \\
\mathtt{ff}^c & = & \mathtt{tt} \\
(\Phi_1 \wedge \Phi_2)^c & = & \Phi_1^c \vee \Phi_2^c \\
(\Phi_1 \vee \Phi_2)^c & = & \Phi_1^c \wedge \Phi_2^c \\
([a]\Phi)^c & = & \langle a \rangle \Phi^c \\
(\langle a \rangle \Phi)^c & = & [a]\Phi^c
\end{array}$$

Proposition: For every process F and HML-formula Φ:

$$F \models \Phi^c \text{ iff } F \not\models \Phi \, .$$

Proposition: For every process F and HML-formula Φ:

$$F \models \Phi^c \text{ iff } F \not\models \Phi \,.$$

Proof: By induction on the structure of Φ

Proposition: For every process F and HML-formula Φ:

$$F \models \Phi^c \text{ iff } F \not\models \Phi \,.$$

Proof: By induction on the structure of Φ

Basis: $\Phi = \texttt{tt}$ and $\Phi = \texttt{ff}$. Trivial.

Proposition: For every process F and HML-formula Φ:

$$F \models \Phi^c \text{ iff } F \not\models \Phi \,.$$

Proof: By induction on the structure of Φ
Basis: $\Phi = \mathtt{tt}$ and $\Phi = \mathtt{ff}$. Trivial.
Induction step:

Proposition: For every process F and HML-formula Φ:

$$F \models \Phi^c \text{ iff } F \not\models \Phi\,.$$

Proof: By induction on the structure of Φ

Basis: $\Phi = \mathtt{tt}$ and $\Phi = \mathtt{ff}$. Trivial.

Induction step:

Case $\Phi = \Phi_1 \wedge \Phi_2$

$$\begin{array}{lll}
 & F \models (\Phi_1 \wedge \Phi_2)^c & \\
\text{iff} & F \models \Phi_1^c \vee \Phi_2^c & \\
\text{iff} & F \models \Phi_1^c \ \text{ or } \ F \models \Phi_2^c & \text{(by clause for } \vee) \\
\text{iff} & F \not\models \Phi_1 \ \text{ or } \ F \not\models \Phi_2 & \text{(by i.h.)} \\
\text{iff} & F \not\models \Phi_1 \wedge \Phi_2 & \text{(by clause for } \wedge).
\end{array}$$

Case $\Phi = [a]\Phi_1$.

$$
\begin{array}{lll}
 & F \models ([a]\Phi_1)^c & \\
\text{iff} & F \models \langle a \rangle \Phi_1^c & \\
\text{iff} & \exists G.\, F \xrightarrow{a} G \text{ and } G \models \Phi_1^c & \\
\text{iff} & \exists G.\, F \xrightarrow{a} G \text{ and } G \not\models \Phi_1 & \text{(by i.h.)} \\
\text{iff} & F \not\models [a]\Phi_1 &
\end{array}
$$

Bisimilarity and Hennessy-Milner Logic I

- Let $E \equiv_M F$ if E and F satisfy exactly the same formulas of modal logic.

Bisimilarity and Hennessy-Milner Logic I

- Let $E \equiv_M F$ if E and F satisfy exactly the same formulas of modal logic.
- Theorem: If $E \sim F$ then $E \equiv_M F$.

Bisimilarity and Hennessy-Milner Logic I

- Let $E \equiv_M F$ if E and F satisfy exactly the same formulas of modal logic.
- Theorem: If $E \sim F$ then $E \equiv_M F$.
- Proof: By induction on modal formulas Φ.
  For any G and H, if $G \sim H$, then $G \models \Phi$ iff $H \models \Phi$.

Bisimilarity and Hennessy-Milner Logic I

- Let $E \equiv_M F$ if E and F satisfy exactly the same formulas of modal logic.
- Theorem: If $E \sim F$ then $E \equiv_M F$.
- Proof: By induction on modal formulas Φ.
  For any G and H, if $G \sim H$, then $G \models \Phi$ iff $H \models \Phi$.
- Basis: $\Phi = \mathtt{tt}$ or $\Phi = \mathtt{ff}$. Clear.

Bisimilarity and Hennessy-Milner Logic I

- Let $E \equiv_M F$ if E and F satisfy exactly the same formulas of modal logic.
- Theorem: If $E \sim F$ then $E \equiv_M F$.
- Proof: By induction on modal formulas Φ.
  For any G and H, if $G \sim H$, then $G \models \Phi$ iff $H \models \Phi$.
- Basis: $\Phi = \mathtt{tt}$ or $\Phi = \mathtt{ff}$. Clear.
- Step: We consider only the case $\Phi = [a]\Psi$. By symmetry, it suffices to show that $G \models [a]\Psi$ implies $H \models [a]\Psi$.
  Assume $G \models [a]\Psi$. For any G' such that $G \xrightarrow{a} G'$, it follows that $G' \models \Psi$.
  Let $H \xrightarrow{a} H'$. Since $G \sim H$, there is a G' such that $G \xrightarrow{a} G'$ and $G' \sim H'$. By the induction hypothesis $H' \models \Psi$, and therefore $H \models \Phi$.

Bisimilarity and Hennessy-Milner Logic II

- E is immediately image-finite if, for each $a \in \mathsf{A}$, the set $\{F : E \xrightarrow{a} F\}$ is finite.

Bisimilarity and Hennessy-Milner Logic II

- E is immediately image-finite if, for each $a \in \mathsf{A}$, the set $\{F : E \xrightarrow{a} F\}$ is finite.
- E is image-finite if all processes reachable from it are immediately image-finite.

Bisimilarity and Hennessy-Milner Logic III

- Theorem: If E, F image-finite and $E \equiv_{\mathrm{M}} F$, then $E \sim F$.

Bisimilarity and Hennessy-Milner Logic III

- Theorem: If E, F image-finite and $E \equiv_{\mathrm{M}} F$, then $E \sim F$.
- Proof: the following relation is a bisimulation.
  $\{(E, F) \,:\, E \equiv_{\mathrm{M}} F \text{ and } E, F \text{ are image-finite}\}$

Bisimilarity and Hennessy-Milner Logic III

- Theorem: If E, F image-finite and $E \equiv_{\mathrm{M}} F$, then $E \sim F$.
- Proof: the following relation is a bisimulation.
  $\{(E, F) : E \equiv_{\mathrm{M}} F \text{ and } E, F \text{ are image-finite}\}$
- Assume $G \equiv_{\mathrm{M}} H$ and $G \xrightarrow{a} G'$
  Need to show $H \xrightarrow{a} H_i$ and $G' \equiv_{\mathrm{M}} H_i$
- Because $G \models \langle a \rangle \mathtt{tt}$ and $G \equiv_{\mathrm{M}} H$, $H \models \langle a \rangle \mathtt{tt}$
  So $\{H' : H \xrightarrow{a} H'\} = \{H_1, \ldots, H_n\}$ is non-empty and finite by image-finiteness.

Bisimilarity and Hennessy-Milner Logic III

- Theorem: If E, F image-finite and $E \equiv_{\mathrm{M}} F$, then $E \sim F$.
- Proof: the following relation is a bisimulation.
  $\{(E, F) : E \equiv_{\mathrm{M}} F \text{ and } E, F \text{ are image-finite}\}$
- Assume $G \equiv_{\mathrm{M}} H$ and $G \xrightarrow{a} G'$
  Need to show $H \xrightarrow{a} H_i$ and $G' \equiv_{\mathrm{M}} H_i$
- Because $G \models \langle a \rangle \mathtt{tt}$ and $G \equiv_{\mathrm{M}} H$, $H \models \langle a \rangle \mathtt{tt}$
  So $\{H' : H \xrightarrow{a} H'\} = \{H_1, \ldots, H_n\}$ is non-empty and finite by image-finiteness.
- If $G' \not\equiv_{\mathrm{M}} H_i$ for each $i : 1 \leq i \leq n$, there are formulas $\Phi_1, \ldots, \Phi_n$ such that $G' \models \Phi_i$ and $H_i \not\models \Phi_i$.
  (Here we use the fact that M is closed under complement.)

Bisimilarity and Hennessy-Milner Logic III

- Theorem: If E, F image-finite and $E \equiv_{\mathrm{M}} F$, then $E \sim F$.
- Proof: the following relation is a bisimulation.
  $\{(E, F) : E \equiv_{\mathrm{M}} F$ and E, F are image-finite$\}$
- Assume $G \equiv_{\mathrm{M}} H$ and $G \xrightarrow{a} G'$
  Need to show $H \xrightarrow{a} H_i$ and $G' \equiv_{\mathrm{M}} H_i$
- Because $G \models \langle a \rangle \mathtt{tt}$ and $G \equiv_{\mathrm{M}} H$, $H \models \langle a \rangle \mathtt{tt}$
  So $\{H' : H \xrightarrow{a} H'\} = \{H_1, \ldots, H_n\}$ is non-empty and finite by image-finiteness.
- If $G' \not\equiv_{\mathrm{M}} H_i$ for each $i : 1 \leq i \leq n$, there are formulas $\Phi_1, \ldots, \Phi_n$ such that $G' \models \Phi_i$ and $H_i \not\models \Phi_i$.
  (Here we use the fact that M is closed under complement.)
- Let $\Psi = \Phi_1 \wedge \ldots \wedge \Phi_n$.
  $G \models \langle a \rangle \Psi$ but $H \not\models \langle a \rangle \Psi$ because each H_i fails to have property Ψ. Contradicts $G \equiv_{\mathrm{HM}} H$.

Bisimilarity and Hennessy-Milner Logic III

- Theorem: If E, F image-finite and $E \equiv_{\mathrm{M}} F$, then $E \sim F$.
- Proof: the following relation is a bisimulation.
  $\{(E, F) : E \equiv_{\mathrm{M}} F$ and E, F are image-finite$\}$
- Assume $G \equiv_{\mathrm{M}} H$ and $G \xrightarrow{a} G'$
  Need to show $H \xrightarrow{a} H_i$ and $G' \equiv_{\mathrm{M}} H_i$
- Because $G \models \langle a \rangle \mathtt{tt}$ and $G \equiv_{\mathrm{M}} H$, $H \models \langle a \rangle \mathtt{tt}$
  So $\{H' : H \xrightarrow{a} H'\} = \{H_1, \ldots, H_n\}$ is non-empty and finite by image-finiteness.
- If $G' \not\equiv_{\mathrm{M}} H_i$ for each $i : 1 \leq i \leq n$, there are formulas $\Phi_1, \ldots, \Phi_n$ such that $G' \models \Phi_i$ and $H_i \not\models \Phi_i$.
  (Here we use the fact that M is closed under complement.)
- Let $\Psi = \Phi_1 \wedge \ldots \wedge \Phi_n$.
  $G \models \langle a \rangle \Psi$ but $H \not\models \langle a \rangle \Psi$ because each H_i fails to have property Ψ. Contradicts $G \equiv_{\mathrm{HM}} H$.
- Case $H \xrightarrow{a} H'$ is symmetric.

Modal characterisation of bisimulation

Given by the previous two results:

- Theorem: If $E \sim F$ then $E \equiv_M F$

Modal characterisation of bisimulation

Given by the previous two results:

- Theorem: If $E \sim F$ then $E \equiv_M F$
- Theorem: If E, F image-finite and $E \equiv_{\mathrm{M}} F$, then $E \sim F$

Modal characterisation of bisimulation

Given by the previous two results:

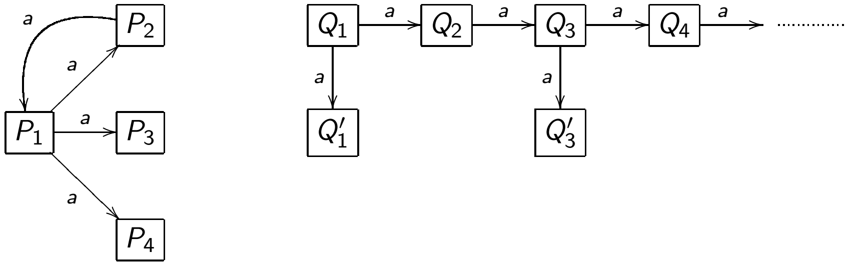
- Theorem: If $E \sim F$ then $E \equiv_M F$
- Theorem: If E, F image-finite and $E \equiv_{\mathrm{M}} F$, then $E \sim F$
- Alternative perspective: properties
- Let $\|\phi\| = \{E \mid E \models \phi\}$
  (May restrict to particular transition system)

Modal characterisation of bisimulation

Given by the previous two results:

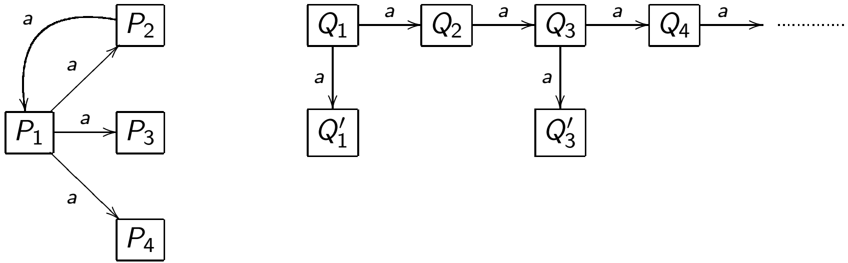
- Theorem: If $E \sim F$ then $E \equiv_M F$
- Theorem: If E, F image-finite and $E \equiv_{\mathrm{M}} F$, then $E \sim F$
- Alternative perspective: properties
- Let $\|\phi\| = \{E \mid E \models \phi\}$
  (May restrict to particular transition system)
- First theorem equivalent to properties expressed by modal formulas are bisimulation invariant: if $E \in \|\phi\|$ and $E \sim F$ then $F \in \|\phi\|$

Bisimulation invariance



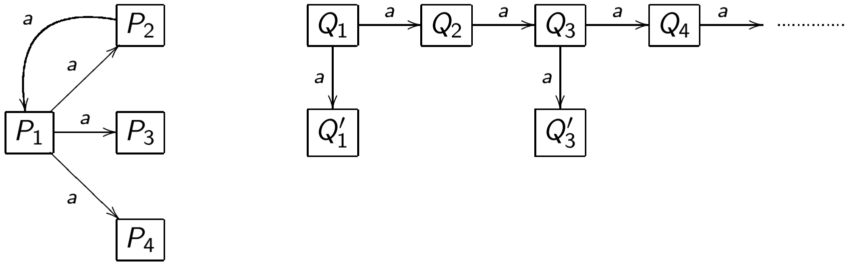
- Many kinds of properties not bisimulation invariant
- $P_1 \sim Q_1$

Bisimulation invariance



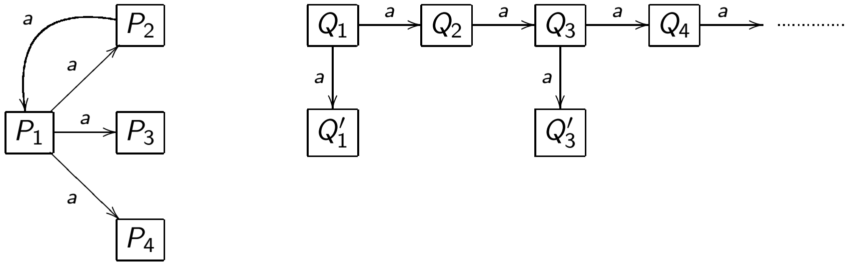
- Many kinds of properties not bisimulation invariant
- $P_1 \sim Q_1$
- But P_1 unlike Q_1
  - has 3 a-transitions

Bisimulation invariance



- Many kinds of properties not bisimulation invariant
- $P_1 \sim Q_1$
- But P_1 unlike Q_1
  - has 3 a-transitions
  - is finite-state

Bisimulation invariance



- Many kinds of properties not bisimulation invariant
- $P_1 \sim Q_1$
- But P_1 unlike Q_1
  - has 3 a-transitions
  - is finite-state
  - has a sequence of transitions that is eventually cyclic

First order logic (FOL)

$$\phi \ ::= xE_ay \mid x = y \mid \neg \phi \mid \phi_1 \vee \phi_2 \mid \exists x.\phi$$

- $x, y \in Var$ (variables); E_a is binary transition relation for each action a

First order logic (FOL)

$$\phi \ ::= xE_ay \ | \ x = y \ | \ \neg\phi \ | \ \phi_1 \vee \phi_2 \ | \ \exists x.\phi$$

- $x, y \in Var$ (variables); E_a is binary transition relation for each action a
- formulas are interpreted over transition systems

First order logic (FOL)

$$\phi \ ::= xE_ay \ \mid \ x = y \ \mid \ \neg\,\phi \ \mid \ \phi_1 \vee \phi_2 \ \mid \ \exists x.\phi$$

- $x, y \in Var$ (variables); E_a is binary transition relation for each action a
- formulas are interpreted over transition systems
- Valuation $\sigma : Var \rightarrow Pr$ (Pr are the processes)

First order logic (FOL)

$$\phi \ ::= xE_ay \ | \ x = y \ | \ \neg \phi \ | \ \phi_1 \vee \phi_2 \ | \ \exists x.\phi$$

- $x, y \in Var$ (variables); E_a is binary transition relation for each action a
- formulas are interpreted over transition systems
- Valuation $\sigma : Var \rightarrow Pr$ (Pr are the processes)
- $\sigma\{P_1/x_1, \ldots, P_n/x_n\}$ is the valuation that is the same as σ except that its value for x_i is P_i, $1 \leq i \leq n$ (where each x_i is distinct).

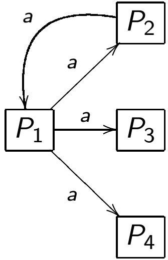
Semantics

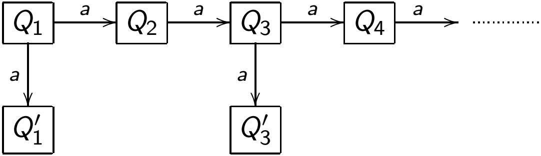
Inductively define when FOL formula ϕ is true on an LTS with respect to a valuation σ as $\sigma \models \phi$

$$
\begin{array}{lll}
\sigma \models xE_a y & \text{iff} & \sigma(x) \xrightarrow{a} \sigma(y) \\
\sigma \models x = y & \text{iff} & \sigma(x) = \sigma(y) \\
\sigma \models \neg\phi & \text{iff} & \sigma \not\models \phi \\
\sigma \models \phi_1 \vee \phi_2 & \text{iff} & \sigma \models \phi_1 \text{ or } \sigma \models \phi_2 \\
\sigma \models \exists x.\phi & \text{iff} & \sigma\{P/x\} \models \phi \text{ for some } P \in Pr
\end{array}
$$

The universal quantifier, $\forall x.\phi = \neg\exists\neg\phi$
$\sigma \models \forall x.\phi$ iff $\sigma\{P/x\} \models \phi$ for all $P \in Pr$.

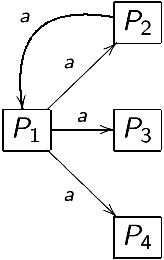
Example

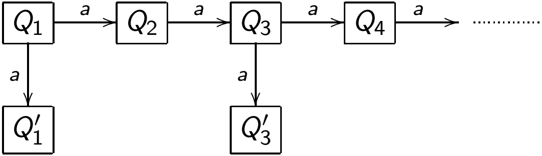




- Assume $\sigma(x_1) = P_1$ and $\sigma(x_2) = Q_1$

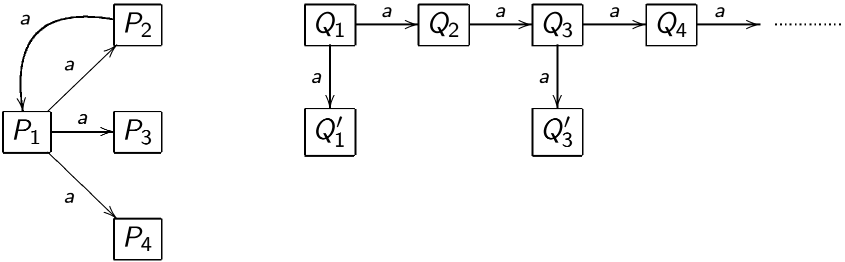
Example





- Assume $\sigma(x_1) = P_1$ and $\sigma(x_2) = Q_1$
- $\sigma \models \exists x.\exists y.\exists z.(x_1 E_a x \wedge x_1 E_a y \wedge x_1 E_a z \wedge x \neq y \wedge x \neq z \wedge y \neq z)$

Example



- Assume $\sigma(x_1) = P_1$ and $\sigma(x_2) = Q_1$
- $\sigma \models \exists x.\exists y.\exists z.(x_1 E_a x \wedge x_1 E_a y \wedge x_1 E_a z \wedge x \neq y \wedge x \neq z \wedge y \neq z)$
- $\sigma \models \forall y.\forall z.(x_2 E_a y \wedge y E_a z \to z \neq x_2)$

Translating modal logic into FOL

The FOL translation of modal formula ϕ relative to variable x is $T_x(\phi)$ which is defined inductively

$$
\begin{array}{lcl}
T_x(\texttt{tt}) & = & x = x \\
T_x(\texttt{ff}) & = & \neg(x = x) \\
T_x(\phi_1 \wedge \phi_2) & = & T_x(\phi_1) \wedge T_x(\phi_2) \\
T_x(\phi_1 \vee \phi_2) & = & T_x(\phi_1) \vee T_x(\phi_2) \\
T_x([a]\phi) & = & \forall y.\neg(xE_a y) \vee T_y(\phi) \\
T_x(\langle a \rangle \phi) & = & \exists y.xE_a y \wedge T_y(\phi)
\end{array}
$$

Translating modal logic into FOL

The FOL translation of modal formula ϕ relative to variable x is $T_x(\phi)$ which is defined inductively

$$
\begin{array}{lcl}
T_x(\mathtt{tt}) & = & x = x \\
T_x(\mathtt{ff}) & = & \neg(x = x) \\
T_x(\phi_1 \wedge \phi_2) & = & T_x(\phi_1) \wedge T_x(\phi_2) \\
T_x(\phi_1 \vee \phi_2) & = & T_x(\phi_1) \vee T_x(\phi_2) \\
T_x([a]\phi) & = & \forall y. \neg(x E_a y) \vee T_y(\phi) \\
T_x(\langle a \rangle \phi) & = & \exists y. x E_a y \wedge T_y(\phi)
\end{array}
$$

Theorem $P \models \phi$ iff $\sigma\{P/x\} \models T_x(\phi)$
Theorem Any first-order formula $T_x(\phi)$ is bisimulation invariant

Translating modal logic into FOL

The FOL translation of modal formula ϕ relative to variable x is $T_x(\phi)$ which is defined inductively

$$
\begin{array}{lcl}
T_x(\mathtt{tt}) & = & x = x \\
T_x(\mathtt{ff}) & = & \neg(x = x) \\
T_x(\phi_1 \wedge \phi_2) & = & T_x(\phi_1) \wedge T_x(\phi_2) \\
T_x(\phi_1 \vee \phi_2) & = & T_x(\phi_1) \vee T_x(\phi_2) \\
T_x([a]\phi) & = & \forall y.\neg(xE_a y) \vee T_y(\phi) \\
T_x(\langle a\rangle\phi) & = & \exists y.xE_a y \wedge T_y(\phi)
\end{array}
$$

Theorem $P \models \phi$ iff $\sigma\{P/x\} \models T_x(\phi)$

Theorem Any first-order formula $T_x(\phi)$ is bisimulation invariant

A FOL formula $\phi(x)$ is equivalent to modal $\phi' \in M$ provided that for any LTS and for any state P, $\sigma\{P/x\} \models \phi$ iff $P \models \phi'$

Van Benthem's theorem

Theorem A FOL formula $\phi(x)$ is equivalent to a modal formula iff $\phi(x)$ is bisimulation invariant.
Proof

Van Benthem's theorem

Theorem A FOL formula $\phi(x)$ is equivalent to a modal formula iff $\phi(x)$ is bisimulation invariant.

Proof If $\phi(x)$ is equivalent to a modal formula ϕ' then $\{P \mid \sigma\{P/x\} \models \phi\} = \|\phi'\|$ which is bisimulation invariant

Van Benthem's theorem

Theorem A FOL formula $\phi(x)$ is equivalent to a modal formula iff $\phi(x)$ is bisimulation invariant.

Proof If $\phi(x)$ is equivalent to a modal formula ϕ' then $\{P \mid \sigma\{P/x\} \models \phi\} = \|\phi'\|$ which is bisimulation invariant

For the converse property, assume that $\phi(x)$ is bisimulation invariant.

Let $\Phi = \{T_x(\psi) \mid \psi \in M$ and $\{\phi(x)\} \models T_x(\psi)\}$

We show that $\Phi \models \phi(x)$ and, therefore, by the compactness theorem, $\phi(x)$ is equivalent to a modal formula ψ' such that $T_x(\psi') \in \Phi$.

Van Benthem's theorem

Theorem A FOL formula $\phi(x)$ is equivalent to a modal formula iff $\phi(x)$ is bisimulation invariant.
Proof If $\phi(x)$ is equivalent to a modal formula ϕ' then $\{P \mid \sigma\{P/x\} \models \phi\} = \|\phi'\|$ which is bisimulation invariant
For the converse property, assume that $\phi(x)$ is bisimulation invariant.
Let $\Phi = \{T_x(\psi) \mid \psi \in M \text{ and } \{\phi(x)\} \models T_x(\psi)\}$
We show that $\Phi \models \phi(x)$ and, therefore, by the compactness theorem, $\phi(x)$ is equivalent to a modal formula ψ' such that $T_x(\psi') \in \Phi$.
Assume $\sigma\{P/x\} \models \psi$ for all $\psi \in \Phi$. We show $\sigma\{P/x\} \models \phi$. We choose a P with the Hennessy-Milner property (that is, if $P' \equiv_M P$ then $P' \sim P$)

Proof Continued

Let $\Psi = \{\, T_x(\psi) \mid P \models \psi \}$.
First, $\Phi \subseteq \Psi$.
Next, $\Psi \cup \{\phi\}$ is satisfiable
Therefore, for some Q, $\sigma\{Q/x\} \models \psi$ for all $\psi \in \Psi$ and $\sigma\{Q/x\} \models \phi$.
However, $Q \sim P$ and because ϕ is bisimulation invariant, $\sigma\{P/x\} \models \phi$ as required.

Alternative Proof

Uses ω-unravelling;
Given a LTS there is a way of unfolding $P \in Pr$ and all its reachable processes into a tree rooted at P which is called *unravelling*.
Theorem If $P \sim Q$. then the ω-unravellings of P and Q are isomorphic