Communication and Concurrency
Lecture 5

Colin Stirling (cps)

School of Informatics

8th October 2012

Modal (Hennessy-Milner) logic: syntax

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

A formula can be

Modal (Hennessy-Milner) logic: syntax

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

A formula can be

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

Modal (Hennessy-Milner) logic: syntax

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

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

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

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 $[K]\Phi$, where K is any set of actions, read as "box K Φ", or "for all K-derivatives Φ,"

Modal (Hennessy-Milner) logic: syntax

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

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 $[K]\Phi$, where K is any set of actions, read as "box K Φ", or "for all K-derivatives Φ,"

- a formula $\langle K \rangle \Phi$, where K is any set of actions, read as "diamond K Φ", or "for some K-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 $E \not\models \Phi$.

- $E \models \texttt{tt}$ $\quad$ $E \not\models \texttt{ff}$

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 $E \models \Psi$
- $E \models [K]\Phi$ iff $\forall F \in \{E' : E \xrightarrow{a} E' \text{ and } a \in K\}.\ F \models \Phi$

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 $E \models \Psi$
- $E \models [K]\Phi$ iff $\forall F \in \{E' : E \xrightarrow{a} E' \text{ and } a \in K\}.\ F \models \Phi$
- $E \models \langle K \rangle\Phi$ iff $\exists F \in \{E' : E \xrightarrow{a} E' \text{ and } a \in K\}.\ F \models \Phi$

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 $E \models \Psi$
- $E \models [K]\Phi$ iff $\forall F \in \{E' : E \xrightarrow{a} E' \text{ and } a \in K\}.\ F \models \Phi$
- $E \models \langle K \rangle\Phi$ iff $\exists F \in \{E' : E \xrightarrow{a} E' \text{ and } a \in K\}.\ F \models \Phi$
- A process E has the property $[K]\Phi$ if every process which E evolves to after carrying out any action in K has the property 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 $E \not\models \Phi$.

- $E \models \texttt{tt} \qquad E \not\models \texttt{ff}$
- $E \models \Phi \wedge \Psi$ iff $E \models \Phi$ and $E \models \Psi$
- $E \models \Phi \vee \Psi$ iff $E \models \Phi$ or $E \models \Psi$
- $E \models [K]\Phi$ iff $\forall F \in \{E' : E \xrightarrow{a} E' \text{ and } a \in K\}.\ F \models \Phi$
- $E \models \langle K \rangle\Phi$ iff $\exists F \in \{E' : E \xrightarrow{a} E' \text{ and } a \in K\}.\ F \models \Phi$
- A process E has the property $[K]\Phi$ if every process which E evolves to after carrying out any action in K has the property Φ
- A process E satisfies $\langle K \rangle\Phi$ if E can become a process that satisfies Φ by carrying out an action in K

Examples

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

Examples

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

Examples

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

Examples

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

Examples

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

Examples

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

Checking satisfaction

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

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

Checking satisfaction

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

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

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

Checking satisfaction

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

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

- $\texttt{Cl} \models [\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \land [\texttt{tock}]\texttt{ff})$
- iff $\forall F \in \{E : \texttt{Cl} \xrightarrow{\texttt{tick}} E\}.\ F \models \langle\texttt{tick}\rangle\texttt{tt} \land [\texttt{tock}]\texttt{ff}$

Checking satisfaction

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

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

- $\texttt{Cl} \models [\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \land [\texttt{tock}]\texttt{ff})$
- iff $\forall F \in \{E : \texttt{Cl} \xrightarrow{\texttt{tick}} E\}.\ F \models \langle\texttt{tick}\rangle\texttt{tt} \land [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt} \land [\texttt{tock}]\texttt{ff}$

Checking satisfaction

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

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

- $\texttt{Cl} \models [\texttt{tick}](\langle\texttt{tick}\rangle\texttt{tt} \land [\texttt{tock}]\texttt{ff})$
- iff $\forall F \in \{E : \texttt{Cl} \xrightarrow{\texttt{tick}} E\}.\ F \models \langle\texttt{tick}\rangle\texttt{tt} \land [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt} \land [\texttt{tock}]\texttt{ff}$
- iff $\texttt{Cl} \models \langle\texttt{tick}\rangle\texttt{tt}$ and $\texttt{Cl} \models [\texttt{tock}]\texttt{ff}$

Checking satisfaction

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

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

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

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

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

Syntactic sugar for sets of actions

Let A be a universal set of actions including τ.
We write

- $a_1, \ldots, a_n$ for $\{a_1, \ldots, a_n\}$

Syntactic sugar for sets of actions

Let A be a universal set of actions including τ.
We write

- $a_1, \ldots, a_n$ for $\{a_1, \ldots, a_n\}$
- $-$ for the set A

Syntactic sugar for sets of actions

Let A be a universal set of actions including τ.
We write

- $a_1, \ldots, a_n$ for $\{a_1, \ldots, a_n\}$
- $-$ for the set A
- $-K$ for the set $A - K$

Syntactic sugar for sets of actions

Let A be a universal set of actions including τ.
We write

- $a_1, \ldots, a_n$ for $\{a_1, \ldots, a_n\}$
- $-$ for the set A
- $-K$ for the set $A - K$
- $-a_1, \ldots, a_n$ for $A - \{a_1, \ldots, a_n\}$

More examples

- $E \models [-]\mathtt{ff}$

More examples

- $E \models [-]\mathtt{ff}$
- E is deadlocked, i.e., it cannot execute any action

More examples

- $E \models [-]\mathtt{ff}$
- E is deadlocked, i.e., it cannot execute any action
- $E \models \langle - \rangle \mathtt{tt}$

More examples

- $E \models [-]\mathtt{ff}$
- E is deadlocked, i.e., it cannot execute any action
- $E \models \langle - \rangle \mathtt{tt}$
- E can execute some action

More examples

- $E \models [-]\mathtt{ff}$
- E is deadlocked, i.e., it cannot execute any action
- $E \models \langle - \rangle \mathtt{tt}$
- E can execute some action
- $E \models \langle - \rangle \mathtt{tt} \wedge [-a]\mathtt{ff}$

More examples

- $E \models [-]\mathtt{ff}$
- E is deadlocked, i.e., it cannot execute any action
- $E \models \langle - \rangle \mathtt{tt}$
- E can execute some action
- $E \models \langle - \rangle \mathtt{tt} \wedge [-a]\mathtt{ff}$
- a must happen next; something can happen, and nothing but a can happen

More examples

- $E \models [-]\mathtt{ff}$
- E is deadlocked, i.e., it cannot execute any action
- $E \models \langle - \rangle \mathtt{tt}$
- E can execute some action
- $E \models \langle - \rangle \mathtt{tt} \wedge [-a]\mathtt{ff}$
- a must happen next; something can happen, and nothing but a can happen
- $E \models \langle - \rangle \mathtt{tt} \wedge [-]\Phi$

More examples

- $E \models [-]\mathtt{ff}$
- E is deadlocked, i.e., it cannot execute any action
- $E \models \langle - \rangle \mathtt{tt}$
- E can execute some action
- $E \models \langle - \rangle \mathtt{tt} \wedge [-a]\mathtt{ff}$
- a must happen next; something can happen, and nothing but a can happen
- $E \models \langle - \rangle \mathtt{tt} \wedge [-]\Phi$
- Φ holds after one step

More examples

- $E \models [-]\mathtt{ff}$
- E is deadlocked, i.e., it cannot execute any action
- $E \models \langle - \rangle \mathtt{tt}$
- E can execute some action
- $E \models \langle - \rangle \mathtt{tt} \wedge [-a]\mathtt{ff}$
- a must happen next; something can happen, and nothing but a can happen
- $E \models \langle - \rangle \mathtt{tt} \wedge [-]\Phi$
- Φ holds after one step
- $E \models \langle - \rangle \mathtt{tt} \wedge [-](\langle - \rangle \mathtt{tt} \wedge [-](\langle - \rangle \mathtt{tt} \wedge [-a]\mathtt{ff}))$

Exercise

Process	Formula	Y/N
$\mathrm{a.0 + a.b.0}$	$\langle \mathrm{a} \rangle \langle \mathrm{b} \rangle \mathtt{tt}$	
	$\langle \mathrm{a} \rangle [\mathrm{b}] \mathtt{ff}$	
	$[\mathrm{a}] \langle \mathrm{b} \rangle \mathtt{tt}$	
	$[\mathrm{a}][\mathrm{b}] \mathtt{ff}$	
$(\mathrm{a.0} \mid \bar{\mathrm{a}}.0)$	$\langle \mathrm{a} \rangle \mathtt{tt}$	
	$\langle \tau \rangle \mathtt{tt}$	
	$\langle \mathrm{a} \rangle \langle \tau \rangle \mathtt{tt}$	
$(\mathrm{a.0} \mid \bar{\mathrm{a}}.0)\backslash \mathrm{a}$	$\langle \mathrm{a} \rangle \mathtt{tt}$	
	$\langle \tau \rangle \mathtt{tt}$	
	$\langle \mathrm{a} \rangle \langle \tau \rangle \mathtt{tt}$	

Negation

HML can be extended with a negation operator $\neg$ having the semantics: $E \models \neg\Phi$ iff $E \not\models \Phi$

Negation

HML can be extended with a negation operator $\neg$ having the semantics: $E \models \neg\Phi$ 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

HML can be extended with a negation operator $\neg$ having the semantics: $E \models \neg\Phi$ 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

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

Φ^c is inductively defined as follows:

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

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: $\Phi = \mathtt{tt}$ and $\Phi = \mathtt{ff}$. 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 $\Phi = [K]\Phi_1$.

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

Realisability, validity, equivalence

- A formula is satisfiable (or realisable) if some process satisfies it.

Realisability, validity, equivalence

- A formula is satisfiable (or realisable) if some process satisfies it.
- A formula is unsatisfiable if no process satisfies it.

Realisability, validity, equivalence

- A formula is satisfiable (or realisable) if some process satisfies it.
- A formula is unsatisfiable if no process satisfies it.
- A formula is valid if all processes satisfy it.

Realisability, validity, equivalence

- A formula is satisfiable (or realisable) if some process satisfies it.
- A formula is unsatisfiable if no process satisfies it.
- A formula is valid if all processes satisfy it.
- Two formulas are equivalent if they are satisfied by exactly the same processes

Exercise

Are the following statements true?

				Y/N
If	Φ valid	then	Φ satisfiable	
If	Φ satisfiable	then	Φ^c unsatisfiable	
If	Φ valid	then	Φ^c unsatisfiable	
If	Φ unsatisfiable	then	Φ^c valid	

Exercise

Let $\longrightarrow$ be the implies connective whose definition is

$$\Phi \longrightarrow \Psi \stackrel{\mathrm{def}}{=} \Phi^c \vee \Psi.$$

Are the following statements true?

		Y/N
If $(\Phi \longrightarrow \Psi)$ valid and Φ valid	then Ψ valid	
If $(\Phi \longrightarrow \Psi)$ satisfiable and Φ satisfiable	then Ψ satisfiable	
If $(\Phi \longrightarrow \Psi)$ valid and Φ satisfiable	then Ψ satisfiable	

Exercise

Which of the following are valid, V, unsatisfiable, U, or neither, *N*?

	V	U	N
$\Phi \longrightarrow \neg\Phi$			
$\neg\Phi \longrightarrow \Phi$			
$\Phi \longrightarrow (\Psi \longrightarrow \Phi)$			
$\Phi \longrightarrow (\Phi \longrightarrow \Psi)$			
$\langle a\rangle\mathtt{tt} \wedge [a]\mathtt{ff}$			
$\langle a\rangle[b](\langle a\rangle\mathtt{tt} \wedge [a]\mathtt{ff})$			
$\langle a\rangle[b](\langle a\rangle\mathtt{tt} \wedge [a]\mathtt{ff}) \wedge [-]\langle b\rangle\mathtt{tt}$			
$\langle a\rangle[b](\langle a\rangle\mathtt{tt} \wedge [a]\mathtt{ff}) \wedge [-]\langle -\rangle\mathtt{tt}$			
$\langle a\rangle(\Phi \vee \Psi) \longrightarrow (\langle a\rangle\Phi \vee \langle a\rangle\Psi)$			
$(\langle a\rangle\Phi \wedge \langle a\rangle\Psi) \longrightarrow \langle a\rangle(\Phi \wedge \Psi)$			
$[a](\Phi \longrightarrow \Psi) \longrightarrow ([a]\Phi \longrightarrow [a]\Psi)$			