Equivalence checking vs Model Checking

Let $I$ be the implementation of a system, using the syntax of MPA/mCRL2.

**Equivalence checking**
Determine whether $I \equiv S$
- $\equiv$ is some behavioural equivalence (e.g., $\sim$);
- $S$ is a specification, also in MPA/mCRL2, of the intended behaviour of the system

**Model checking**
Determine whether $I \models F$
- $\models$ is a satisfaction relation
- $F$ specifies a desirable property of the system, expressed in some suitable logic
Properties of Reactive Systems

What kind of properties would we want to verify of reactive systems?

**Modal properties**—What can happen now? (possibility, necessity)
- can drink coffee now
- after a coin is inserted, a coffee can be obtained
- both tea and coffee can be obtained

**Temporal properties**—behaviour in time
- never drinks any alcohol
  (safety property: “something bad never happens”)
- every message will eventually be received
  (likeness property: “something good eventually happens”)

Hennessy-Milner Logic (syntax)

The set $\mathcal{M}$ of Hennessy-Milner formulae is defined by:

$$F, G ::= \mathbf{tt} \mid \mathbf{ff} \mid F \land G \mid F \lor G \mid \langle \alpha \rangle F \mid [\alpha]F,$$

where $\alpha$ ranges over Act.

Abbreviations for $A = \{\alpha_1, \ldots, \alpha_n\} (n \in \mathbb{N})$:
- $\langle A \rangle F ::= \langle \alpha_1 \rangle F \lor \cdots \lor \langle \alpha_n \rangle F$
- $[A]F ::= [\alpha_1]F \land \cdots \land [\alpha_n]F$

(In particular, $\langle \emptyset \rangle F = \mathbf{ff}$ and $[\emptyset]F = \mathbf{tt}$.)
Intuitive interpretation

A formula $F$ can be true or false in a particular state of a given LTS:

- $tt$ is true in every state;
- $ff$ is false in every state;
- $F \land G$ is true in a state $s$ if, and only if, $F$ and $G$ are both true in $s$;
- $F \lor G$ is true in a state $s$ if, and only if, (at least) one of $F$ or $G$ is true in $s$;
- $\langle \alpha \rangle F$ is true in a state $s$ if, and only if, there exists a state $s'$ such that $s \xrightarrow{\alpha} s'$ and $F$ is true in $s'$; and
- $[\alpha]F$ is true in a state $s$ if, and only if, for all states $s'$ such that $s \xrightarrow{\alpha} s'$ it holds that $F$ is true in $s'$.

The formula $\langle a \rangle tt$ expresses that an $a$-transition is possible. Intuitively, it is true in state $s$, but not in states $s_1$ and $s_2$.

The formula $\langle a \rangle tt \lor \langle b \rangle tt$ expresses that an $a$-transition or a $b$-transition is possible. Intuitively, it is true in states $s$ and $s_1$, but not in state $s_2$.

The formula $[c](\langle a \rangle tt \land \langle b \rangle tt)$ expresses that every $c$-transition leads to a state that has both an outgoing $a$-transition and an outgoing $b$-transition. Intuitively, it is true in state $s$. Is it also true in state $s_1$? And what about state $s_2$?

The formula $[b]ff$ expresses that a $b$-transition always leads to a state in which $ff$ is true. Intuitively, this formula is not true in states $s$ and $s_1$. Is it true in $s_2$?
Properties of Reactive Systems

What kind of properties would we want to verify of reactive systems?

Modal properties—What can happen now? (possibility, necessity)
- can drink coffee now ............................................................... ⟨coffee⟩tt
- after a coin is inserted, a coffee can be obtained ....................... [coin]⟨coffee⟩tt
- both tea and coffee can be obtained ....................................... ⟨coffee⟩tt ∧ ⟨tea⟩tt
  or ⟨coffee⟩⟨tea⟩tt ∨ ⟨tea⟩⟨coffee⟩tt, or ...

Temporal properties—behaviour in time
- never drinks any alcohol ...................................................... cannot yet be expressed
  (safety property: “something bad never happens”)
- every message will eventually be received .............................. cannot yet be expressed
  (liveness property: “something good eventually happens”)
(To be able to express temporal properties, we shall later add recursion to Hennessy-Milner Logic.)

Hennessy-Milner Logic (denotational semantics)

Let (S, Act, →) be an LTS.

For F ∈ M, we define [F] ⊆ S with recursion on the structure of F as follows:

- [tt] = S;
- [F ∧ G] = [F] ∩ [G];
- [[α]F] = ⟨·α·⟩[F];
- [ff] = ∅;
- [F ∨ G] = [F] ∪ [G];
- [[[α]F] = [·α·][F];

where ⟨·α·⟩ : 2^S → 2^S and [·α·] : 2^S → 2^S are defined by

⟨·α·⟩S' = {p ∈ S | ∃p'. p ⊢ p' and p' ∈ S'} ,
[·α·]S' = {p ∈ S | ∀p'. p ⊢ p' implies p' ∈ S'} .

2^S denotes the powerset of S, i.e., the set of all subsets of S
Let $S = \{s, s_1, s_2\}$. Then

$$\llbracket \langle a \rangle \rrbracket \top = \langle t \cdot \rangle \top = \langle \cdot \rangle S$$

$$= \{ p \in S \mid \exists p'. p \xrightarrow{a} p' \text{ and } p \in S \} = \{s\}$$

$$\llbracket \langle a \rangle \rrbracket \top \lor \llbracket \langle b \rangle \rrbracket \top = \llbracket \langle a \rangle \rrbracket \top \cup \llbracket \langle b \rangle \rrbracket \top = \langle \cdot \rangle S \cup \langle \cdot \rangle \{s\}$$

$$= \{ p \in S \mid \exists p'. p \xrightarrow{a} p' \text{ and } p \in S \}$$

$$\cup \{ p \in S \mid \exists p'. p \xrightarrow{b} p' \text{ and } p \in S \}$$

$$= \{s\} \cup \{s, s_1\} = \{s, s_1\}$$
Satisfaction

Let $(S, \text{Act}, \rightarrow)$ be an LTS.

We say that $p \in S$ satisfies Hennessy-Milner formula $F \in \mathcal{M}$ (notation: $p \models F$) if $p \in [F]$.

(If $p$ does not satisfy $F$, then we write $p \not\models F$.)

Then the following properties hold for all $p \in S$, for all $F, G \in \mathcal{M}$ and for all $\alpha \in \text{Act}$:

- $p \models tt$;
- $p \nmodels ff$;
- $p \models F \land G$ if, and only if, $p \models F$ and $p \models G$;
- $p \models F \lor G$ if, and only if, $p \models F$ or $p \models G$;
- $p \models \langle \alpha \rangle F$ if, and only if, there exists $p' \in S$ such that $p \xrightarrow{\alpha} p'$ and $p' \models F$; and
- $p \models [\alpha] F$ if, and only if, for all $p' \in S$ such that $p \xrightarrow{\alpha} p'$ it holds that $p' \models F$.

(Proofs of these properties are straightforward.)

Negation?

Negation has (purposely) been omitted from the syntax, but it can be expressed.

For every formula $F \in \mathcal{M}$, let $F^c \in \mathcal{M}$ be recursively defined as follows:

- $tt^c = ff$;
- $(F \land G)^c = F^c \lor G^c$;
- $(\langle \alpha \rangle F)^c = [\alpha] F^c$;
- $ff^c = tt$;
- $(F \lor G)^c = F^c \land G^c$;
- $([\alpha] F)^c = \langle \alpha \rangle F^c$.

Let $(S, \text{Act}, \{ \xrightarrow{\alpha} \mid \alpha \in \text{Act} \})$.

Then, for every $F \in \mathcal{M}$, we have $[F^c] = S \setminus [F]$.

Hence, for all $p \in S$, $p \models F^c$ if, and only if, $p \nmodels F$. 

/ department of mathematics and computer science
Hennessy-Milner Logic and Strong Bisimilarity

**Image finite**
Let \((S, \text{Act}, \rightarrow)\) be an LTS.
A state \(p \in S\) is image-finite if \(\{p' \mid p \xrightarrow{\alpha} p'\}\) is finite for every \(\alpha \in \text{Act}\).
The LTS is image-finite if so is each of its states.

**Hennessy-Milner Theorem**
Let \((S, \text{Act}, \rightarrow)\) be an image-finite LTS.
Then, for all \(p, q \in S\), we have that \(p \sim q\) if, and only if, \(p\) and \(q\) satisfy the same Hennessy-Milner logic formulae.

**Distinguishing formula**
If two states \(p\) and \(q\) are states in an image-finite LTS and \(p \not\sim q\), then there exists a so-called distinguishing formula, i.e., a Hennessy-Milner logic formula \(F\) such that \(p \models F\) and \(q \not\models F\).

Note that the LTS is image-finite.

By the Hennessy-Milner, to prove that \(s \not\sim t\), it suffices to find a distinguishing Hennessy-Milner formula for \(s\) and \(t\), i.e., a formula \(F\) such that \(s \models F\) and \(t \not\models F\).

We can take \(F = \langle a \rangle[c] \mathbf{ff}\). On the one hand, since \(s \not\xrightarrow{a} s_1\) and \(s_1 \not\xrightarrow{c}\), we have that \(s \models F\). On the other hand, since \(t_1 \xrightarrow{a} t_2\) and \(t \not\xrightarrow{a} t_1\) is the only transition from \(t\) labelled \(a\), we have that \(t \not\models F\).