Modeling & Verification

Hennessy-Milner Logic

Max Tschaikowski (tschaikowski@cs.aau.dk)
Slides Courtesy of Giorgio Bacci

in the last Lecture

- Properties of Strong Bisimilarity (review)
- Example: Buffer implementation in CCS
- Weak Bisimilarity (Properties & Game characterisation)
- Tool: Concurrency Workbench Aalborg Edition (CAAL)

in this Lecture

- Model Checking (idea & motivations)
- Hennessy-Milner Logic (syntax & semantics)
- Correspondence with Strong Bisimilarity
- example in CAAL

Verifying Correctness

Equivalence Checking Approach

Impl $\equiv$ **Spec**

- $\equiv$ is an abstract equivalence, e.g. $\sim$ or $\approx$
- **Spec** & **Impl** often expressed in the same language
- **Spec** provides the full specification of the behaviour

Model Checking Approach

Impl $\models$ **Property**

- $\models$ is the *satisfaction* relation
- **Property** is often expressed via a logic
- **Property** is a specific feature of the behaviour

Example of Properties

$$CS \stackrel{\text{def}}{=} \overline{\text{pub}}.\overline{\text{coin}}.(\text{coffee}.CS + \text{tea}.CS)$$

- is not willing to drink tea, now
- is willing to drink both coffee and tea, now
- is willing to drink tea but not coffee, now
- never drinks alcoholic beverages
- always produces a publication after drinking coffee

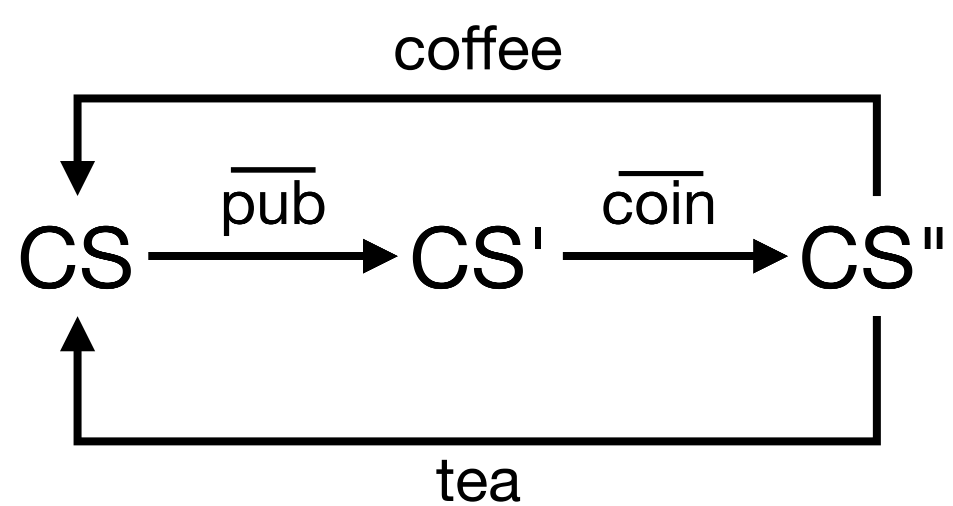
We need a logical language to formally express
the above properties of reactive systems

Modal Logic

We need special logical connectives able to relate the current with the next states of a process

Modal connectives (possibility vs necessity)

- **can** drink a coffee now
- **cannot** drink coffee now
- **always** can produce a publication **after** drinking coffee



Hennessy-Milner Logic

syntax & semantics

Syntax of HML

false | disjunction | box

$$\phi ::= tt \mid ff \mid \phi \wedge \phi \mid \phi \vee \phi \mid \langle a \rangle \phi \mid [a]\phi$$

true | conjunction | diamond

We denote by $\mathcal{M}$ the set of all HML formulas

Intuition

- **tt** all processes satisfy this property
- **ff** no processes satisfy this property
- $\wedge$ and $\vee$, usual Boolean connectives (AND and OR)
- $\langle \mathbf{a} \rangle \boldsymbol{\phi}$ there is at least one a-successor that satisfies ϕ
- $\mathbf{[a]}\boldsymbol{\phi}$ all a-successors satisfy ϕ

Semantics of HML

Let (Proc, Act, $\{\xrightarrow{\alpha} \mid \alpha \in Act\}$) be an LTS

Definition (Satisfiability Relation $\models \subseteq Proc \times \mathcal{M}$)

- $p \models tt$ always (i.e., for all $p \in Proc$)
- $p \models ff$ never (i.e., for no $p \in Proc$)
- $p \models \phi \wedge \psi$ iff $p \models \phi$ and $p \models \psi$
- $p \models \phi \vee \psi$ iff $p \models \phi$ or $p \models \psi$
- $p \models \langle a \rangle \phi$ iff $\exists p' \in Proc$ such that $p \xrightarrow{a} p'$ and $p' \models \phi$
- $p \models [a]\phi$ iff $\forall p' \in Proc$ such that $p \xrightarrow{a} p'$ then $p' \models \phi$

We write $p \not\models \phi$ whenever p does not satisfy ϕ

Examples

How can we formally express the properties seen before?

Informal Description	**HML**
can drink a coffee now	$\langle coffee \rangle tt$
cannot drink coffee now	**??**
always can produce a publication **after** drinking coffee	$[coffee]\langle \overline{pub} \rangle tt$

Expressing Negation

For every formula ϕ we define the formula ϕ^c as follows:

Definition (Complement)

$$tt^c = ff$$

$$ff^c = tt$$

$$(\phi \wedge \psi)^c = \phi^c \vee \psi^c$$

$$(\phi \vee \psi)^c = \phi^c \wedge \psi^c$$

$$(\langle a \rangle \phi)^c = [a]\phi^c$$

$$([a]\phi)^c = \langle a \rangle \phi^c$$

De Morgan's laws

note the switching of the modalities

"**cannot** drink coffee now" can be expressed as

$(\langle coffee \rangle tt)^c =$ **[coffee]ff**

Theorem

For any $p \in Proc$ any ϕ HML formula, $p \not\models \phi$ iff $p \models \phi^c$

Denotational Semantics

For a formula ϕ, let $\langle\!\langle\phi\rangle\!\rangle \subseteq \mathrm{Proc}$ contain all states that satisfy ϕ

Definition (Denotation $\langle\!\langle - \rangle\!\rangle : \mathcal{M} \longrightarrow 2^{\mathrm{Proc}}$)

$$\langle\!\langle tt \rangle\!\rangle = \mathrm{Proc}$$

$$\langle\!\langle ff \rangle\!\rangle = \varnothing$$

$$\langle\!\langle \phi \wedge \psi \rangle\!\rangle = \langle\!\langle \phi \rangle\!\rangle \cap \langle\!\langle \psi \rangle\!\rangle$$

$$\langle\!\langle \phi \vee \psi \rangle\!\rangle = \langle\!\langle \phi \rangle\!\rangle \cup \langle\!\langle \psi \rangle\!\rangle$$

$$\langle\!\langle \langle a \rangle \phi \rangle\!\rangle = \langle \cdot a \cdot \rangle \langle\!\langle \phi \rangle\!\rangle$$

$$\langle\!\langle [a] \phi \rangle\!\rangle = [\cdot a \cdot] \langle\!\langle \phi \rangle\!\rangle$$

where for all $S \subseteq \mathrm{Proc}$ we define

$$\langle \cdot a \cdot \rangle S = \{\, p \in \mathrm{Proc} \mid \exists p'.\ p \xrightarrow{a} p' \text{ and } p' \in S \,\}$$

$$[\cdot a \cdot] S = \{\, p \in \mathrm{Proc} \mid \forall p'.\ p \xrightarrow{a} p' \text{ implies } p' \in S \,\}$$

Equivalence of Semantics

Theorem

Let (Proc, Act, $\{\xrightarrow{\alpha} \mid \alpha \in \text{Act}\}$) be an LTS, $p \in \text{Proc}$ and ϕ a formula of Hennessy-Milner Logic, then

$$p \models \phi \quad \text{iff} \quad p \in \langle\!\langle \phi \rangle\!\rangle$$

Proof: by induction on the structure of the formula ϕ (see Exercise 5.6 in the textbook) break?

Relation between HML & Strong Bisimilarity

Hennessy-Milner Theorem

There is a fruitful connection between the apparently unrelated concepts of strong bisimilarity and HML

Theorem

Let (Proc, Act, $\{\xrightarrow{\alpha} \mid \alpha \in \mathrm{Act}\}$) be an image-finite LTS, $p,q \in$ Proc two states. Then

$$p \sim q \quad \text{iff} \quad \underbrace{\text{for all } \phi \in \mathcal{M}.\ (\, p \models \phi \Leftrightarrow q \models \phi \,)}_{\text{logical equivalence!}}$$

Hence, if $p \nsim q$, there exists a **distinguishing** HML formula!

Image-finite LTS

Definition (Image-finite LTS)

An LTS (Proc, Act, $\{\xrightarrow{\alpha} \mid \alpha \in \text{Act}\}$) is image-finite if for every $p \in \text{Proc}$ and every $\alpha \in \text{Act}$, the set

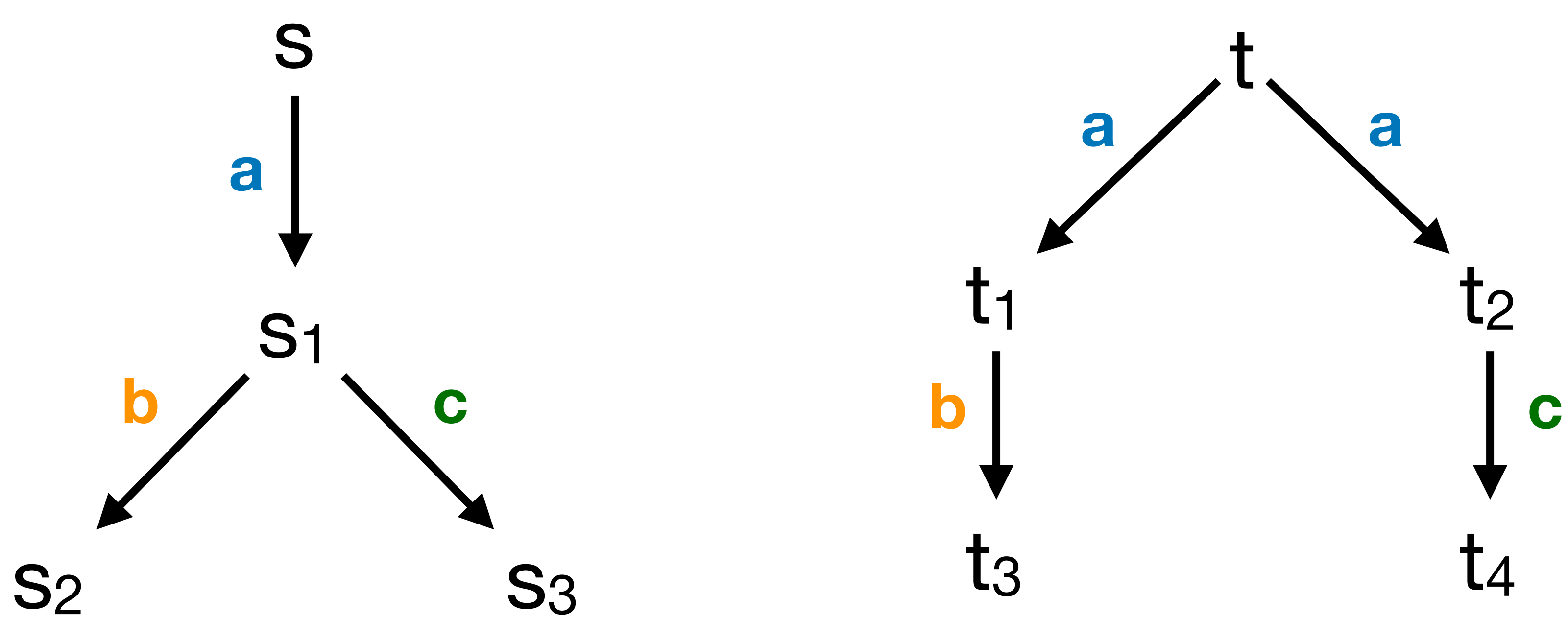
$$\{ p' \in \text{Proc} \mid p \xrightarrow{\alpha} p' \} \text{ is finite}$$

The following CCS processes are not image finite

$$\text{Rep} \overset{\text{def}}{=} \mathbf{a}.0 \mid \text{Rep} \qquad\qquad P \overset{\text{def}}{=} \sum_{i \geq 0} \mathbf{a}.0$$

Distinguishing formulas

We have already showed that s $\nsim$ t by using the game characterisation of strong bisimilarity.



Can we do the same by using Hennessy-Milner theorem?

Example Session in CAAL

http://caal.cs.aau.dk