Hennessy-Milner Logic [1].

Colloquium in honor of Robin Milner.

Carlos Olarte.

Pontificia Universidad Javeriana

28 April 2010.

[1] Based on the talks: [1,2,3]

Prof. Robin Milner (R.I.P).

LIX, Ecole Polytechnique.

Motivation

How to Verify the Correctness of a Concurrent System?

By Using Equivalences

$$impl \equiv spec$$

- $\equiv$ is an equivalence (e.g., bisimulation)
- The specification and the implementation are written in the same language, e.g., CCS.
- Spec provides the full specification of the intended behavior.

Motivation

How to Verify the Correctness of a Concurrent System?

Model Checking Approach

$$impl \models Property$$

- $\models$ is the satisfaction relation.
- Property is a partial specification of the intended behavior.
- Property is a formula in logic.

Motivation

Specification of Properties

- Henessy-Milner Logic : a Modal logic to express properties of reactive systems.
- Modalities: Necessity and Possibility .
  - The action a cannot happen.
  - After an action a, the systems can perform an action b.
  - After an a action, the system never exhibits a b action.
- More Examples:
  - A coffee is given after a coin is inserted.
  - After a coin is inserted, either a coffee or a tea are dispensed.

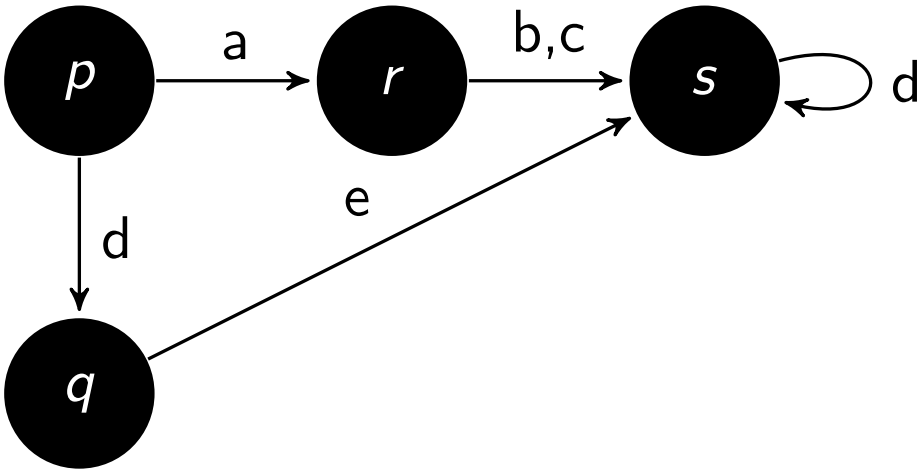
Backgroud

Labelled Transition Systems (LTS)

A LTS is a triple $\langle S, A, T \rangle$ where

- S is a set of states .
- A is a set of actions (e.g., $a, \overline{a}, b, c, \tau, ...$).
- $T \subseteq S \times \mathcal{L} \times S$ is the transition relation:

$$s_1 \xrightarrow{a} s_2 \text{ means } (s_1, a, s_2) \in T$$

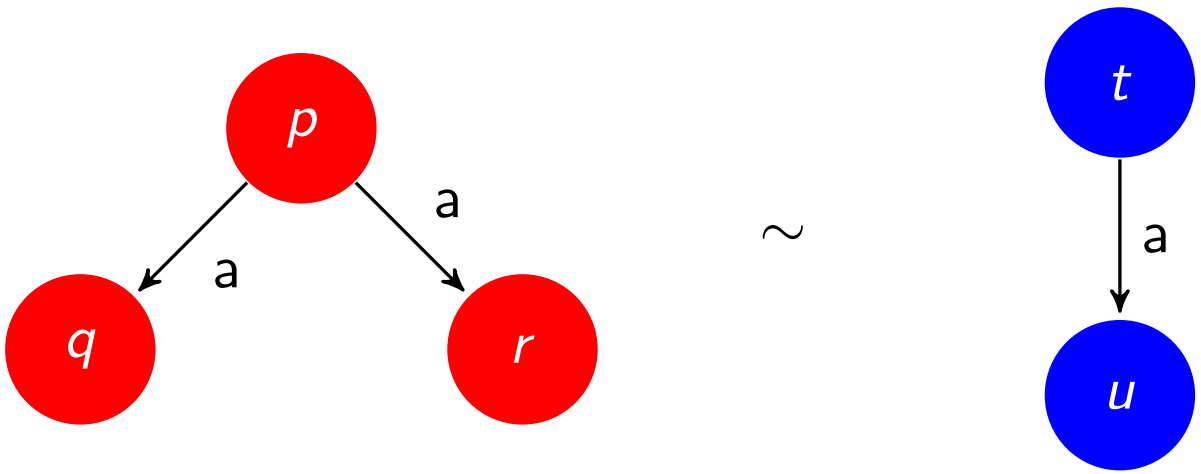


Backgroud

Bisimulation

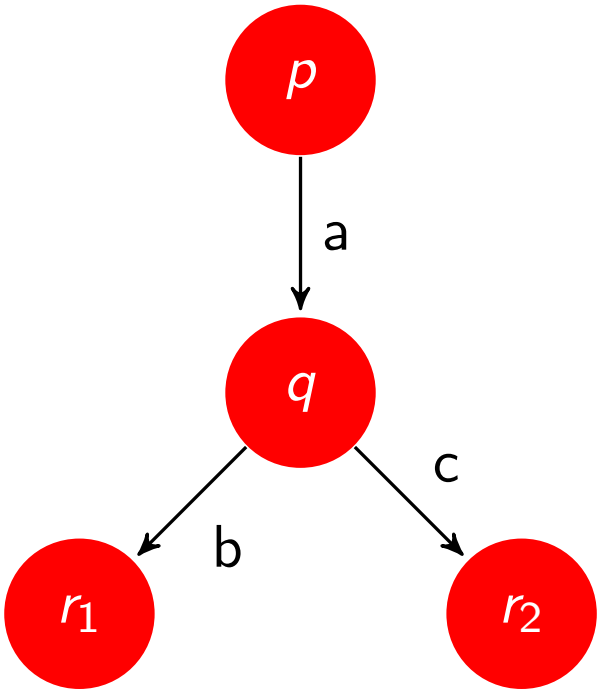
Given two LTSs, a relation $R \subseteq S \times S'$ is called a bisimulation whenever:

- If $(p, q) \in R$ and $p \xrightarrow{a} p'$ then there exists q' s.t. $q \xrightarrow{a} q'$ and $(p', q') \in R$.
- If $(q, p) \in R$ and $q \xrightarrow{a} q'$ then there exists p' s.t. $p \xrightarrow{a} p'$ and $(q', p') \in R$.

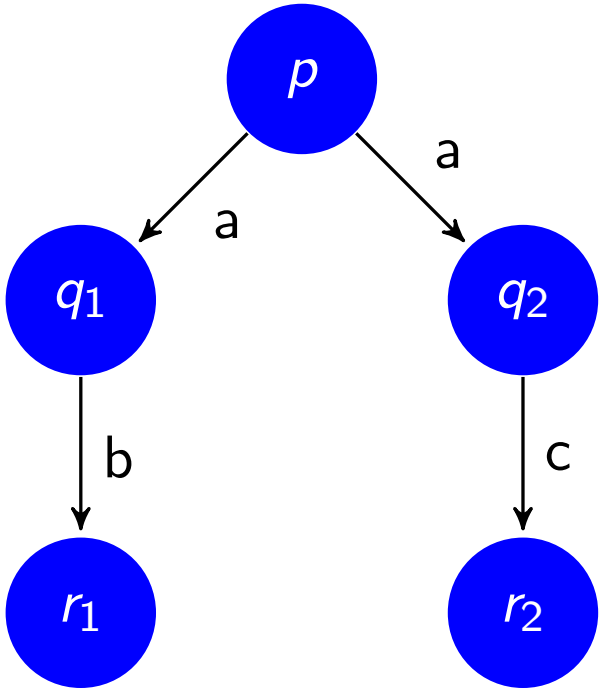


Bisimilarity is the finest reasonable equivalence, where reasonable means that we can observe only the behavior and not the state-space.

Two systems that are not bisimilar



$S1 = a.(b + c)$



$S2 = a.b + a.c$

$$S1 \not\sim S2$$

Notice that in S_1: $p \xrightarrow{a} q \xrightarrow{b} r_1$ while in S_2: $p \xrightarrow{a} q_2 \not\xrightarrow{b}$

Henessy-Milner Logic

Syntax

Let A be a set of actions. Formulae in HM Logic are build from:

HM Logic Syntax

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

- `tt`: The constant *true* formula.
- `ff`: The constant *false* formula.
- $\Phi_1 \wedge \Phi_2$: Conjunction.
- $\Phi_1 \vee \Phi_2$: Disjunction.
- $[A]\Phi$: Read as box A Φ. For all A-derivation, Φ holds.
- $\langle A \rangle \Phi$: Read as diamond A Φ. There exists an A-derivation s.t. Φ holds.

Henessy-Milner Logic

Semantics (Intuition)

- The formula $\texttt{tt}$ is satisfied for all process.
- No process satisfies $\texttt{ff}$.
- The operands $\wedge$ and $\vee$ are interpreted as usual in logic.
- $[A]\Phi$ means, all a-sucessor ($a \in A$) satisfies Φ.
- $\langle A \rangle \Phi$ means, there exists an a-sucessor ($a \in A$) that satisfies Φ.

Henessy-Milner Logic

Semantics

Let $(Proc, \mathcal{L}, \{\xrightarrow{a} \mid a \in \mathcal{L}\})$ be an LTS.

Validity of $P \models \Phi$

$P \models \mathtt{tt}$ for each $P \in Proc$

$P \not\models \mathtt{ff}$

$P \models \Phi \wedge \Theta$ iff $P \models \Phi$ and $P \models \Theta$

$P \models \Phi \vee \Theta$ iff $P \models \Phi$ or $P \models \Theta$

$P \models [A]\Phi$ iff $P' \models \Phi$ for all $P' \in Proc, a \in A$ s.t. $P \xrightarrow{a} P'$

$P \models \langle A \rangle \Phi$ iff $P \xrightarrow{a} P'$ for some $P' \in Proc, a \in A$ s.t. $P' \models \Phi$

We say that a formula Φ is:

- Satisfiable : if there exists P s.t., $P \models \Phi$.
- Unsatisfiable : if no process satisfies it.
- Valid if all processes satisfy it.

Examples

- $P \models \langle tick \rangle \texttt{tt}$: P can do an *tick*.
- $P \models \langle tick \rangle \langle tock \rangle \texttt{tt}$: P can do a *tick* and then a *tock*.
- $P \models \langle \{tick, tock\} \rangle \texttt{tt}$: P can do a *tick* or a *tock*.
- $P \models [tick] \texttt{ff}$: P cannot do a *tick*.
- $P \models \langle tick \rangle [tock] \texttt{ff}$: P performs a *tick* and goes to a state from which there are no *tock* transitions.
- $P \models \langle tick \rangle \texttt{ff}$: This is always false.
- $P \models [tick] \texttt{tt}$: This is always true.

Examples

Continuation

Let $\mathcal{L}$ be the set of actions, $A \subseteq \mathcal{L}$ and $\overline{A}$ the complement of A.

- $P \models [\mathcal{L}]\mathtt{ff}$: P is a deadlock (it cannot perform any action).
- $P \models \langle\mathcal{L}\rangle\mathtt{tt}$: P can perform some action.
- $P \models \langle\mathcal{L}\rangle\mathtt{tt} \wedge [\overline{\{a\}}]\mathtt{ff}$: a must happen next.
- $P \models \langle\mathcal{L}\rangle\mathtt{tt} \wedge [\mathcal{L}]\Phi$: Φ holds after one step.

Negation and De Morgan laws

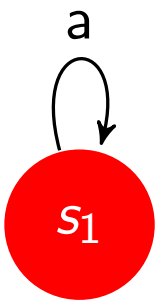
$$P \models \neg\Phi \text{ iff } P \not\models \Phi$$

- $\neg\mathtt{tt} = \mathtt{ff}$.
- $\neg\mathtt{ff} = \mathtt{tt}$.
- $\neg(\Phi \wedge \Theta) = \neg\Phi \vee \neg\Theta$.
- $\neg(\Phi \vee \Theta) = \neg\Phi \wedge \neg\Theta$.
- $\neg[A]\Phi = \langle A\rangle\neg\Phi$.
- $\neg\langle A\rangle\Phi = [A]\neg\Phi$.

With the subsets $\{\langle\cdot\rangle, \vee, \mathtt{ff}, \neg\}$, $\{[\cdot], \wedge, \mathtt{tt}, \neg\}$ or $\{\langle\rangle, [\cdot], \vee, \wedge, \mathtt{tt}, \mathtt{ff}\}$ one gets the full logic.

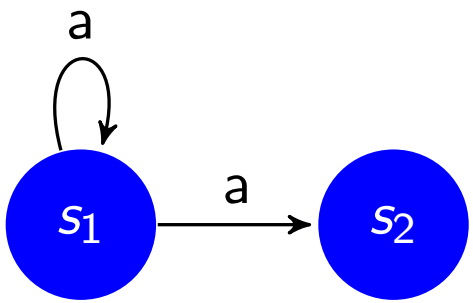
More Examples

Formulae Distinguishing Systems



$C = a.C$

$C \not\models \langle a \rangle [a]\mathtt{ff}$

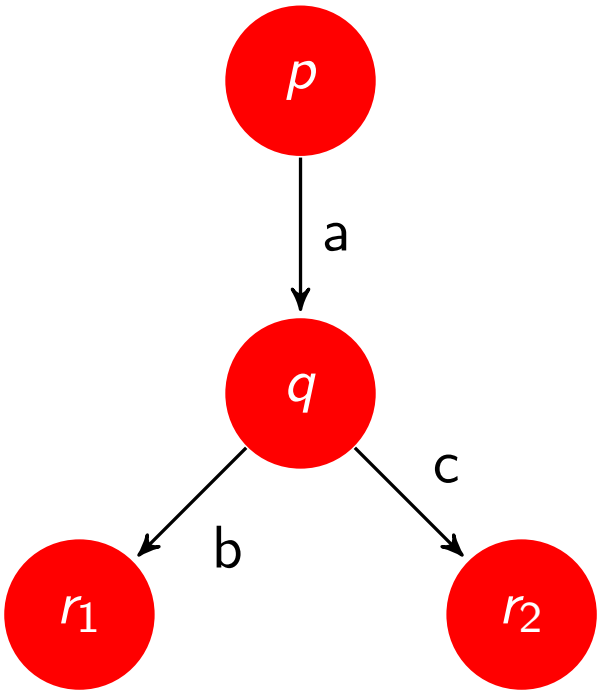


$D = a.D + a.nil$

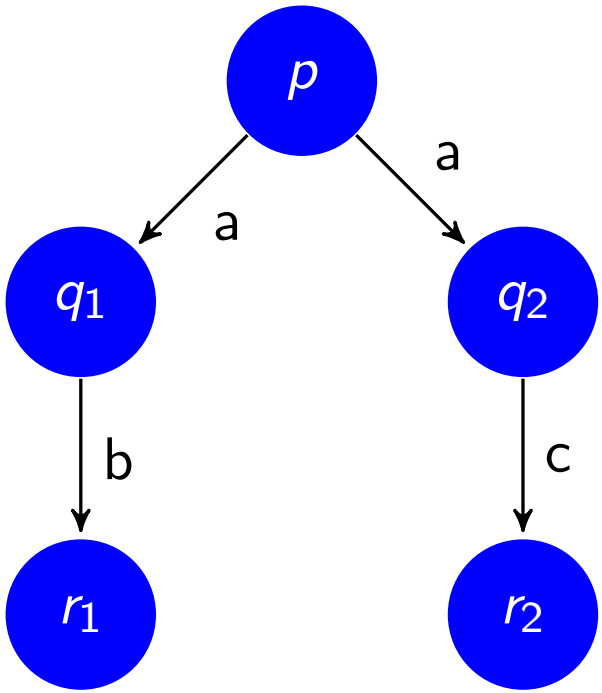
$D \models \langle a \rangle [a]\mathtt{ff}$

More Examples

Formulae Distinguishing Systems



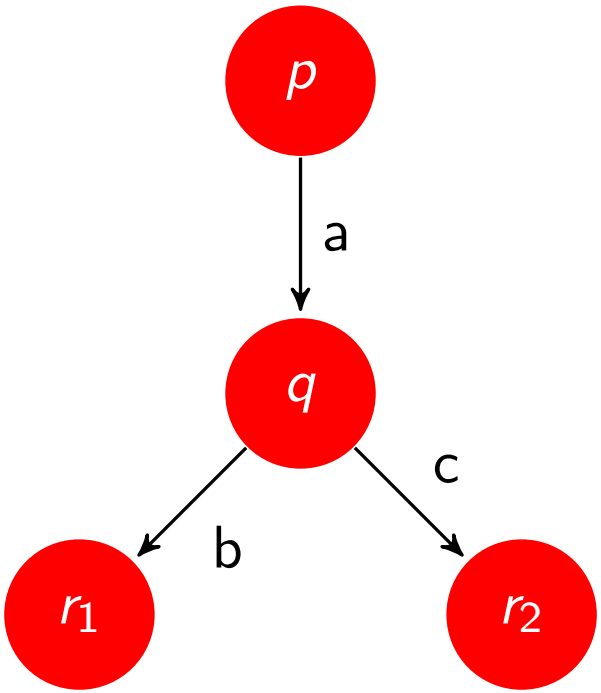
$\models \langle a \rangle (\langle b \rangle \mathtt{tt} \wedge \langle c \rangle \mathtt{tt})$



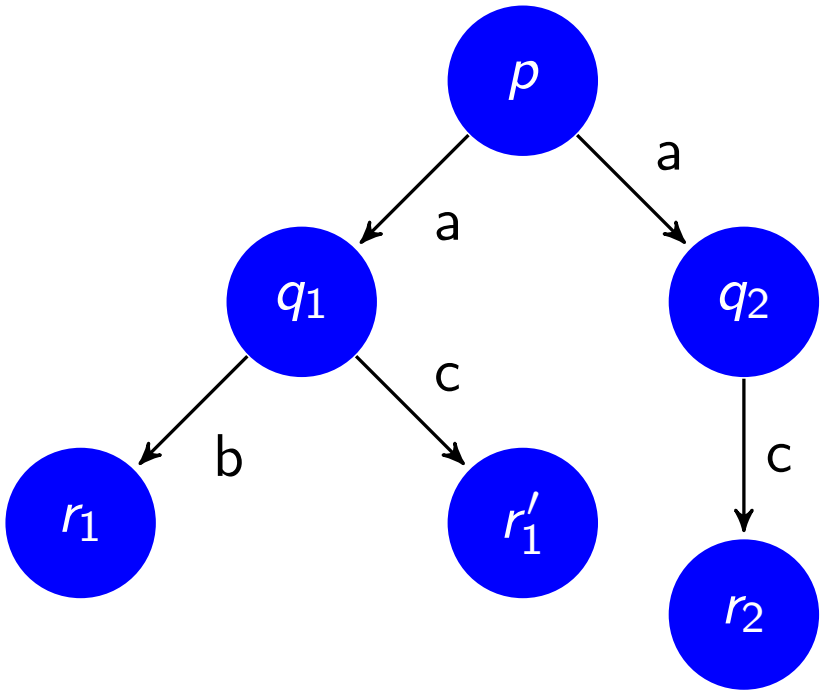
$\not\models \langle a \rangle (\langle b \rangle \mathtt{tt} \wedge \langle c \rangle \mathtt{tt})$

More Examples

Formulae Distinguishing Systems



$\not\models \langle a \rangle(\neg\langle b \rangle \mathtt{tt})$



$\models \langle a \rangle(\neg\langle b \rangle \mathtt{tt})$

HM Login and Strong Bilimilarity

Image-Finite System

The LTS $(Proc, \mathcal{L}, \{\xrightarrow{a} \mid a \in \mathcal{L}\})$ is image-finite if for every $P \in Proc$ and every $a \in A$ the set

$$\{P' \in Proc \mid P \xrightarrow{a} P'\}$$

is finite.

Theorem (Henessy-Milner)

Let $(Proc, \mathcal{L}, \{\xrightarrow{a} \mid a \in \mathcal{L}\})$ be a image-finite LTS and $P, Q \in Proc$. The following sentences are equivalent:

1. $P \sim Q$ (P and Q are strongly bisimilar).
2. For every HM formula Φ, $P \models \Phi \Leftrightarrow Q \models \Phi$.

I make no claim that everything can be done by algebra ... It is perhaps equally true that not everything can be done by logic; thus one of the outstanding challenges in concurrency is to find the right marriage between logic and behavioral approaches
– Robin Milner.

Sources

1. Pawel Sobocinski. Bisimulation, Games and Hennessy Milner logic. Lecture 1 of Modelli Matematici dei Processi Concorrenti.
2. Martin Wirsing and Axel Rauschmayer. Prozessalgebra: Hennessy-Milner Logic. Basierend auf Lecture Notes von Rocco De Nicola.
3. Modal Logic. `http://www.doc.ic.ac.uk/~pg/Concurrency/course.html`.