Concurrent System Design

Proofs and Process Equivalences

Valentín Cassano
cassanv@mcmaster.ca

Department of Computing and Software
McMaster University

Winter 2011

On what a lecture/tutorial should not be about

The art of transmitting information
from the notes of the lecturer
to the notes of the students
without passing through the minds
of either

What is a proof?

Proof

A proof is sufficient evidence or argument for the truth of a proposition.

Mathematical Proof

A (mathematical) proof is a convincing demonstration that some mathematical statement is *necessarily* true. (Mathematical) Proofs are obtained from deductive reasoning, rather than from inductive or empirical arguments.

Formal Proof

A (formal) proof is a finite sequence of sentences (called well-formed formulas in the case of a formal language) each of which is an axiom or follows from the preceding sentences in the sequence by a rule of inference.

Example

Question:

Show that processes $S1$ and $S2$ describe the same behavior:

```
P = ( a -> b -> P).
Q = ( c -> b -> Q ).
||S1 = ( P || Q ).

S2 = ( a -> c -> b -> S2 | c -> a -> b -> S2 ).
```

Process equivalence

Process equivalence

Two processes P_1 and P_2 are equivalent if they (somehow) exhibit the same behavior.

Some desirable properties of behavioral equivalence are:

- Each process is equivalent to itself (reflexivity).
- It should support stepwise derivation of processes (transitivity).
- Two behaviourally equivalent processes can be used interchangeably as part of large process descriptions without affecting the overall behaviour (congruence).
- Behavioural equivalence based on the observable behaviour of processes (not on their structure).

First Attempt

Definition (Trace)

A (finite) trace is a sequence of action names $\langle a_1, \ldots, a_n \rangle \in \alpha(P)^*$ s.t there exists a sequence of transitions $s_0 \rightarrow^{a_1} s_1 \rightarrow \cdots \rightarrow^{a_n} s_n$ in the LTS described by process P.

Trace Equivalence

Two processes P_1 and P_2 are equivalent iff $\tau(P_1) = \tau(P_2)$.

Questions:

- Is trace equivalence reasonable for reactive machines that interact with their environment?
- If you want coffee and you hate tea, which machine would you like to interact with?

Example

Second Attempt

Isomorphism Equivalence

Two processes P_1 and P_2 are equivalent iff there is an isomorphism between LTS_{P_1} and LTS_{P_2}.

This notion may be proven too restrictive, i.e., it may rule out some processes ought to be considered to be equivalent.

Third attempt

(Strong) Bisimulation

A binary relation R over the states of a LTS is a (strong) bisimulation iff whenever $(s_1, s_2) \in R$ then the following conditions hold:

- if $s_1 \rightarrow^a s_1'$ then there is a transition $s_2 \rightarrow^a s_2'$ s.t. $(s_1', s_2') \in R$.
- if $s_2 \rightarrow^a s_2'$ then there is a transition $s_1 \rightarrow^a s_1'$ s.t. $(s_1', s_2') \in R$.

Two states s_1 and s_2 are bisimilar iff there is a bisimulation that relates them.

Example

Questions/Comments?