Concurrency: Theory, Languages and Programming

– Equivalences for Concurrency –

Session 10 – January 8, 2003

Uwe Nestmann

EPFL-LAMP

Repetition of Algebraic Notions

relations/functions

composition

comparison, containment

preorder/equivalence

reflexivity

symmetry

transitivity

kernel of a (reflexive) preorder

comparison, containment vs fine/coarse

congruence

by definition?

Automata

An **automaton**
over an **action alphabet** *Act*:

- a set : the **states**
- a state : the **start state**
- a subset : the **accepting states**
- a subset *Act* : the **transitions**

A transition is also written .

Example Automaton

Let *Act* be .

Let be defined as

Automata (II)

An automaton is

- **finite-state**, if is finite, and
- **deterministic** if for each pair *Act* there is **exactly one transition** . (Note the similarity to a function *Act* .)

Question: Would the formulation "**at most one transition**" yield less deterministic automata?

Note:"Complete" an automaton?

Behavior: *Language* of an Automaton

Let be an automaton over *Act*.
Let be a string over *Act*. Then:

is said to **accept** , if there is a path in
— from to some accepting state —
whose arcs are labeled successively .

The **language** of , denoted by ,
is the set of strings accepted by .

denotes the empty string.

<u>**Fact:**</u> The language of any finite-state automaton is *regular*.

Regular Sets

(a mathematical model *)*

<u>Definition:</u> A set of strings over *Act* is **regular** if it can be built from

the **empty set** and the **singleton sets** (*Act*),

using the operations of

union ( ),

concatenation ( ), and

iteration ( ).

In regular sets, we sometimes write for and for .

Regular Expressions

(* *syntax to indicate the elements of the mathematical model* *)

Definition: The set of **regular expressions** over *Act* is generated by the following grammar:

where *Act*.

In regular expressions, we often write for . . .

regular expressions	*regular sets*
,	

“Denotational Semantics”

RegExps	RegSets

in the image of the semantics function ,
all of , , and , are operators on sets so they entail the calculation of the actual set that they represent

compare to Arithmetic Expressions and Natural Numbers

note that is not surjective . . . ***why?***

Some Laws on Regular Expressions

Be Careful . . .

<u>Note:</u>
The regular set means "no path". But:
The regular expression means "empty path".

As an example, compare with .

Arden's rule

Theorem:
For any sets of strings and , the equation

has

as a **solution**.
Moreover, this solution is unique if .

Example Automaton

Determine the language of the previous automaton
as the regular expression describing
the strings accepted in the initial state.

Write down a set of equations,
one equation for each state.

Solve the set of equations . . .

Determinism / Nondeterminism

Analyze the two automata of § 2.4 of [Mil99].

Message1:
Language equivalence is blind for nondeterminism!

In fact, every nondeterministic automaton can be converted into a determinstic one that accepts the same language.

Message2:
Language equivalence is blind for deadlocks!

Example?

Message3 (less important):
Language equivalence requires accepting states.

Labeled Transition Systems

<u>Definition:</u>

An **LTS** over an **action alphabet** *Act*:

- a set of **states**
- a ternary **transition relation** *Act*

A transition is also written .

If we call a **derivative** of .

Equivalence on LTS ?

Example: Compare   and   in

Induce simulation of paths
through step-by-step simulation of actions . . .

(Strong) Simulation on LTS

<u>Definition:</u> **(learn it by heart!)**
Let be an LTS.

1. Let be a binary relation over .
   is a **(strong) simulation** over if, whenever ,

   if then there is such that and .

2. **(strongly) simulates** , written ,
   if there is a (strong) simulation such that .

The relation is sometimes called *similarity*.

Properties of Simulations

Lemma:
If and are simulations, then

is also a simulation.

is also a simulation ?

is also a simulation ?

Definition: Let be a LTS.

$\bigcup$ is simulation over

Lemma:

is the largest simulation over .

is a reflexive preorder over .

Is any simulation a preorder?

Working with Simulation

What do we do with simulations?

- exhibiting a simulation:
  "*guessing*" a relation that contains
- checking a simulation:
  check that a given relation is in fact a simulation.

Fortunately, clever people developed algorithms and respective tools (CWB, ABC) that are good at "guessing" simulations.

In fact, they ***generate*** relations algorithmically that—by construction—fulfil the property of being a simulation.

Results on (semi-)decidability are very important for such tools.

Home-Working with Simulation

Example: Find all non-trivial simulations in

How many are there ?

Trivial pairs are any pairs with elements from
(because there are no transitions),
as well as any identity on .

Trivial simulations are those that either
(0) are empty, or
(1) contain only trivial pairs, or
(2) contain at least one trivial pair that is not reachable from a contained non-trivial one.

Towards Equivalence

Simulation is only a preorder,
thus it allows us to *distinguish* states.

We want instead an equivalence,
which would allow us to *equate* states.

The mathematical way: just take the “kernel”

if and

However, there are two different natural candidates !

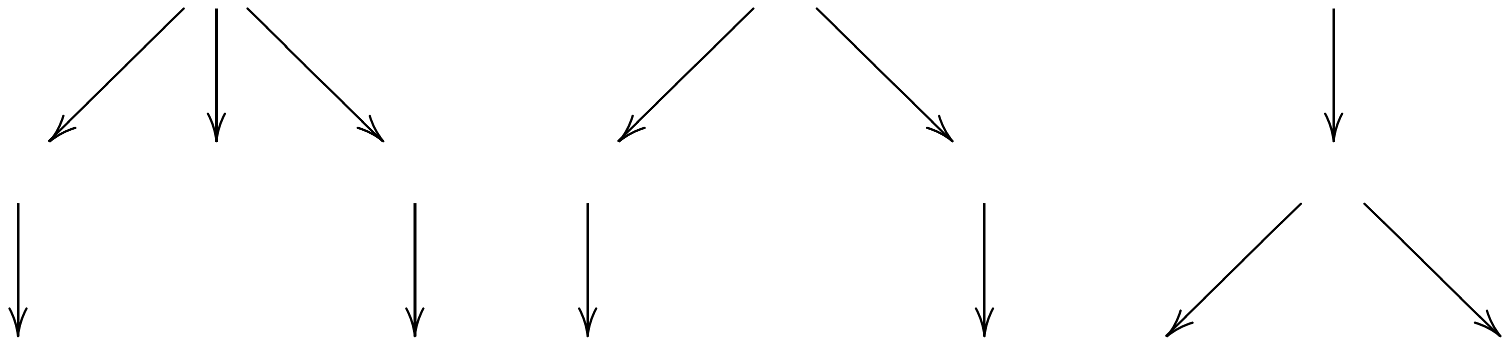
- mutual simulation
- bisimulation

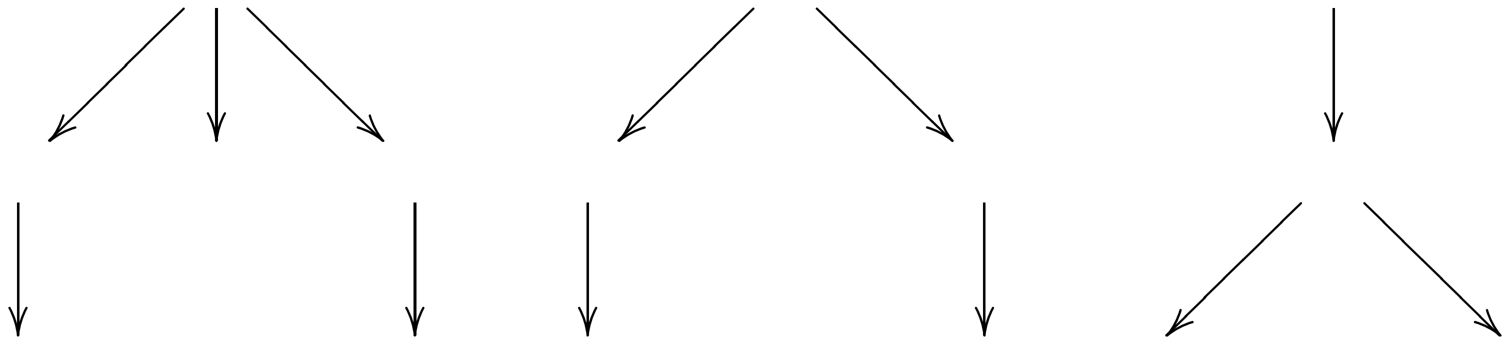
Mutual Simulation: Back and Forth

Definition:
Let be a LTS. Let .

and are **mutually similar**, written ,
if there is a pair of simulations and
with (i.e., with and).

Example: Mut. Sim. vs Lang. Equiv.





Mutual Simulation (II)

Proposition:

is an equivalence relation.

Proof?

Typical research-craftsmen work . . .

(Strong) Bisimulation

Definition: **(learn it by heart!)**
A binary relation over is
a **(strong) bisimulation** over the LTS
if both and its converse are (strong) simulations.

and are **(strongly) bisimilar**, written ,
if there is a (strong) bisimulation such that .

Alternatively:

$\bigcup$ is (strong) bisimulation over $\mathcal{T}$

(Strong) Bisimulation (II)

Proposition:

- is (itself) a (strong) bisimulation.
- is an equivalence relation.

Proof?

Again, typical research-craftsmen work . . .

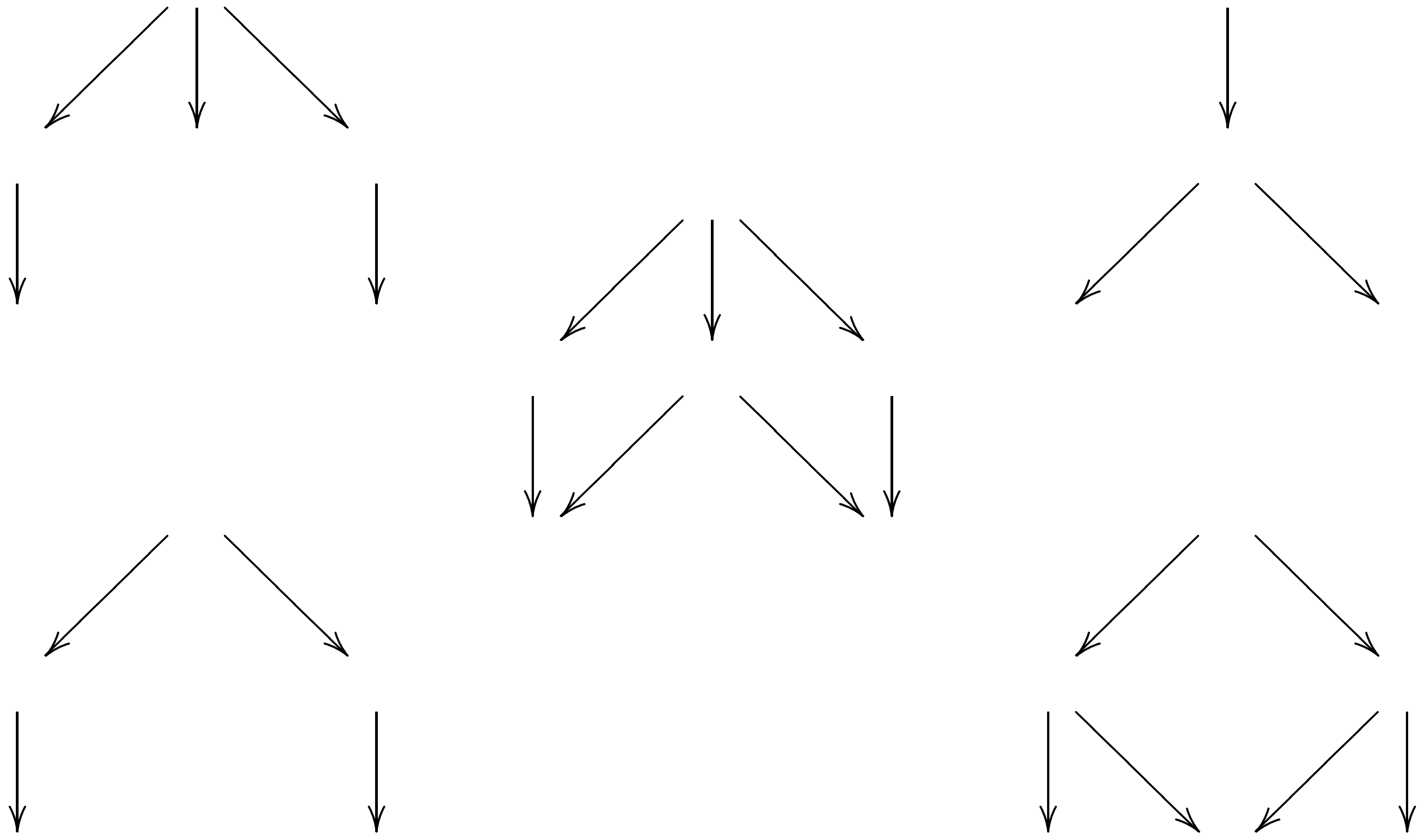
Example

Prove .

Write out ...

Minimization ?!

Example: Mutual vs Bi



Isomorphism on LTS

Definition:
Let be two LTS over *Act* for .

and are **isomorph(ic)**,
written ,
if there is a **bijection** on between and
that preserves , i.e., with

iff .

Isomorphism on LTS (II)

Proposition:

is an equivalence relation
(on the domain of LTSs).

Proof?

Be careful with the interpretation of reflexivity, symmetry, and transitivity . . .

Isomorphism vs Bisimulation

“Problem”:
Isomorphism compares two transition systems;
Bisimulation (at least as we have defined it) compares two states.

Redefine to be a bisimulation
if and are simulations on their respective domains, i.e.,
.

Redefine to the whole domain of LTSs.
Be careful with the interpretation of reflexivity, symmetry, and transitivity . . .

Isomorphism vs Bisimulation

1. **reachability**

Isomorphism vs Bisimulation

2. **copying**

Isomorphism vs Bisimulation

3. **recursion/unfolding**

Which is the Best Equivalence ?

language equivalence
mutual simulatity
bisimilarity
isomorphism
identity

To be remembered:
What are the intuitive distinguishing aspects
between all of these notions of equivalence? (Exam . . .)