8. Simulation und Äquivalenz

8.1 Simulation

Simulation: a relation

Def. R simulates L with iff

a) \( l_0 \xrightarrow{r_0} \) \( r_0 \) simulates \( l_0 \)

b) If \( \xrightarrow{l \rightarrow l'} \) \( \xrightarrow{r \rightarrow r'} \) and \( r' \) simulates \( l' \)

Vice versa

Def. R simulates L with iff

a) \( l_0 \xrightarrow{r_0} \) \( r_0 \) simulates \( l_0 \)

b) If \( \xrightarrow{l \rightarrow l'} \) \( \xrightarrow{r \rightarrow r'} \) and \( r' \) simulates \( l' \)

Let’s construct

L: R:

Process L has two traces: a.b, a.c
Process R has same traces.
“Systems with same traces are equivalent!”
L and R are not equivalent, ... by no means!

R is “more liberal” than L.

intuitively:
R simulates L
L does not simulate R

L does not simulate R
Simulation equivalence

**Def.:** P and Q are *simulation equivalent* iff P simulates Q and Q simulates P.

**Observation.** Simulation equivalence is an equivalence relation on processes.

**Def.:** Let ~ be an equivalence relation on processes. Then ~ is a *congruence* (w.r.t. \(\times\)) iff for all processes P, Q, R holds:

- If \(P \sim Q\), then \(P \times R \sim Q \times R\).

**Observation.** Simulation equivalence is no congruence!

---

**Example**

Remember: L and R simulate one another

\[ L \times S \text{ and } R \times S \text{ do not simulate one another} \]
How gain a simulation congruence?

Observation:
A slightly more tight relation
makes simulation equivalence a congruence:
R simulates L with \( r \) and
L simulates R with \((r)^{-1}\).

L simulates R with

R simulates L with

is not the reverse of

Bisimulation harmonizes with the Temporal Logic CTL *

Theorem.
Two states are bisimilar
iff they share the same CTL * properties.

Consequence:
Specify a system in terms of CTL*.
This may yield various different implementations.
They all are bisimilar.

Variant: L is weakly simulated by R

a) If
b) If

Caution!
Weak bisimulation is no congruence

Def. \( \rho \) is a bisimulation from L to R iff
R simulates L with \( \rho \) and
L simulates R with \((\rho)^{-1}\).

Def. L and R are bisimilar iff
there exists a bisimulation from L to R.

\( \text{sim} = \{(l_0, r_0), (l_0, r_2), (l_1, r_1), (l_2, r_1)\} \) is a bisimulation from L to R.

Theorem. Bisimulation is a congruence.

Consequence:
Specify a system in terms of CTL*.
This may yield various different implementations.
They all are bisimilar.
Examples for weak Bisimilarity

P and Q are weakly bisimilar:

R, S and T are pairwise not weakly bisimilar:

Caution!
Weak bisimulation is no congruence
Failure Trace Equivalence

... like Failure equivalence.
But now you continue along a trace

$$a \{f\} c \{e\} d$$
is a failure trace of L but not of R

$$a \{f\} c \{e\} d$$

Ready Trace Equivalence

In a trace, between each two actions,
present the alternative actions.

$$[a,\{c\},b]$$
is a ready trace of L but not of R

Tree Equivalence

Unfold the transition systems as trees

$$L \equiv_U R$$
iff both trees are isomorphic

Structural Equivalence

Equivalence:

$$L \equiv_K R$$
iff the transition systems are isomorphic

Further equivalences

- Ready equivalence
- Ready Simulation equivalence
- Ready Trace Simulation equivalence
- Completed Simulation equivalence
- Failure Simulation equivalence
- Failure Trace Simulation equivalence
- Simulation equivalence

... 152 ones

8. Simulation und Äquivalenz

8.3 Weitere Kongruenzen

Ende
8. Simulation und Äquivalenz

8.4 Temporal Logic

How to express properties of systems that perform discrete steps?

CTL*, intuitively

From a transition system to its tree

Once more: a process and its tree

Computation Tree Logic $CTL^*$

$p = \downarrow$

eventually $p$  
globally $p$  
next $p$  
p until $q$

AGEF
Typical applications

“Never something bad happens” \[\text{AG safely}\]
“No deadlock reachable” \[\text{AG enabled}\]
„With a series of clicks you reach p“ \[\text{EF } p\]
“Whatever happens – you will succeed” \[\text{AF Goal}\]
“Each requirement is followed by an acknowledgement” \[\text{AG(req U AF ack)}\]
“It makes sense to wait” \[\text{AG AF avail}\]
“You always can properly terminate” \[\text{AG EF exit}\]

Expressiveness

Why just THIS logic?

Theorem.
Two states are bisimilar
iff they share the same CTL* properties.

Consequence:
Specify a system in terms of CTL*.
This may yield various different implementations.
They all are bisimilar.

Why not just First order logic (predicate logic)?

Example:
Whenever process A sends a message to process B, then B eventually sends an acknowledgement to A.

First order:
\[\forall t \, (\text{send}(A,B,t) \implies \exists t' \, (\text{greater}(t',t) \land \text{send}(B,A,t')))]\]

CTL*:
\[\text{AG} \, (\text{Send} \, (A,B) \implies \text{AF Send} \, (B,A))\]

8. Simulation und Äquivalenz

8.4 Temporal Logic