8. Simulation und Äquivalenz

8.4 Temporal Logic
CTL*, intuitively

How to express properties of systems that perform discrete steps?
From a transition system to its tree
Once more: a process and its tree
Computation Tree Logic $CTL^*$

$p = \bullet$

eventually $p$  globally $p$  next $p$  $p$ until $q$
AGEF
Valid formulas

AG ( V EX )

EX

AX

AGEF

EFG
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” \(\text{AG EF exit}\)
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) \rightarrow \exists t' \ (\text{greater}(t',t) \land \text{send}(B,A,t')))\)

CTL*:
$\text{AG} \ (\text{Send}(A,B) \rightarrow \text{AF} \ \text{Send}(B,A))$
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 bisimular.
8. Simulation und Äquivalenz

8.4 Temporal Logic