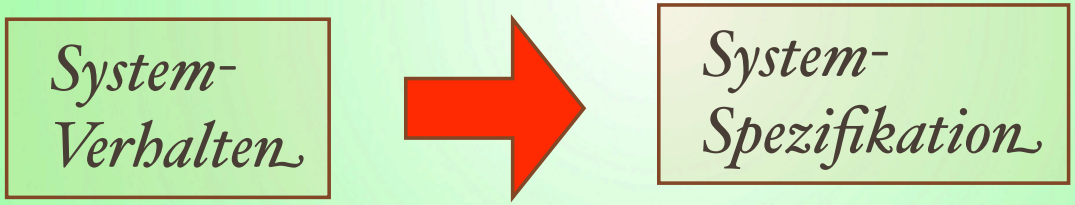
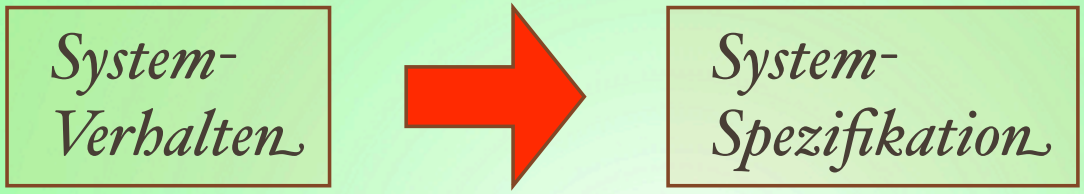
Einführung in LTL unter MAUDE

Verifikation eines Systems



Verifikation eines Systems



„system enjoys property"

Theorem Proving:

„Systems ***formula*** *implies property* ***formula.***"

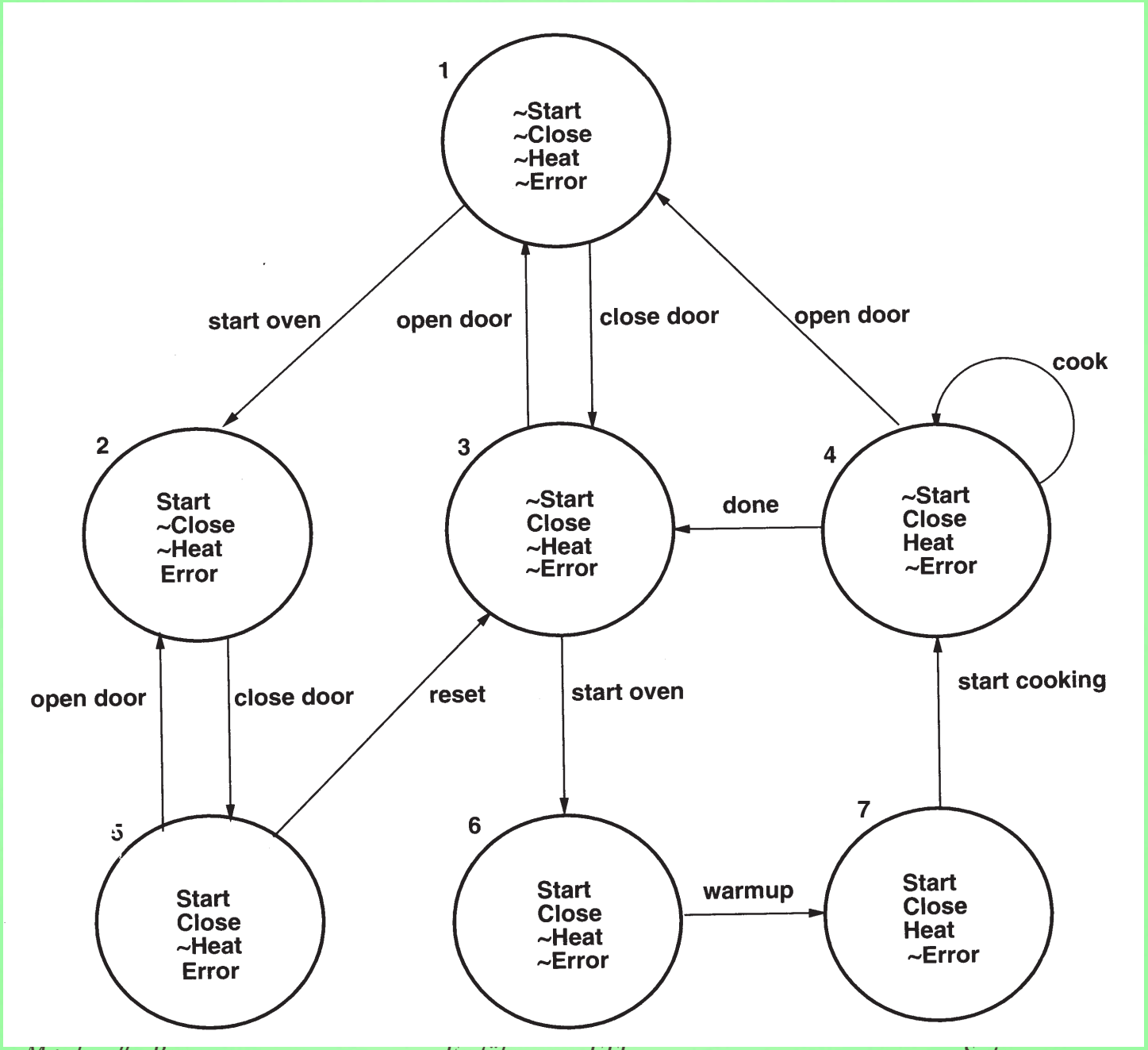
Model Checking: $\phi \Longrightarrow f$

„Systems ***semantics*** *is model of property* ***formula.***"

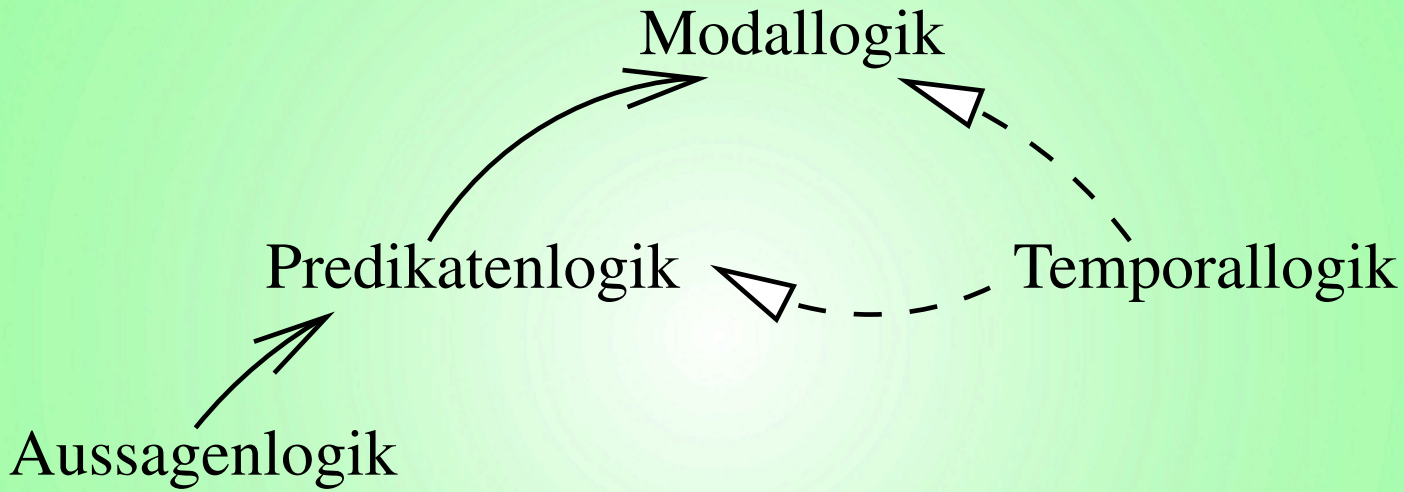
$$M, s \models f$$

Definition 5.29 *Eine* Kripke-Struktur *ist ein Tupel* $M := (S, S_0, R, L)$, *für das folgendes gilt:*

1. S *endliche Zustandsmenge,*
2. $S_0 \subseteq S$ *Menge von Anfangszuständen,*
3. $R \subseteq S \times S$ *links totale*[6] *(Transitions-) Relation,*
4. $L : S \to 2^{AP}$ *Abbildung, die jedem Zustand* s *eine Menge* $L(s) \subseteq AP$ *von aussagenlogischen atomaren Formeln zuordnet (die in diesem Zustand gelten).*
5. *Ein* Pfad *oder eine* Rechnung *aus* $s \in S$ *ist eine Folge* $\pi = s_0, s_1, s_2, \ldots$ *mit* $s_0 = s$ *und* $\forall i \geq: R(s_i, s_{i+1})$


1
~Start
~Close
~Heat
~Error
start oven
open door
close door
open door
cook
2
Start
~Close
~Heat
Error
3
~Start
Close
~Heat
~Error
done
4
~Start
Close
Heat
~Error
open door
close door
reset
start oven
start cooking
5
Start
Close
~Heat
Error
6
Start
Close
~Heat
~Error
warmup
7
Start
Close
Heat
~Error

5.6 Temporale Logik



Beispiel 5.32 Spezifikation eines Aufzuges (Fragment)

I. Jede Anforderung des Aufzugs wird auch erfüllt.

II. Der Aufzug passiert keinen Stockwerk (SW) mit einer nicht erfüllten Anforderung.

Beispiel für physikalisches Bewegungsgesetz: $z(t) = -\frac{1}{2}gt^2$

I. Jede Anforderung des Aufzugs wird auch erfüllt.

I. $\forall t, \forall n(app(n,t) \Rightarrow \exists t' > t \,.\, serv(n,t'))$

$H(t)$ Position des Fahrstuhls zur Zeit t,

$app(n,t)$ offene Anforderung von Stockwerk n zur Zeit t,

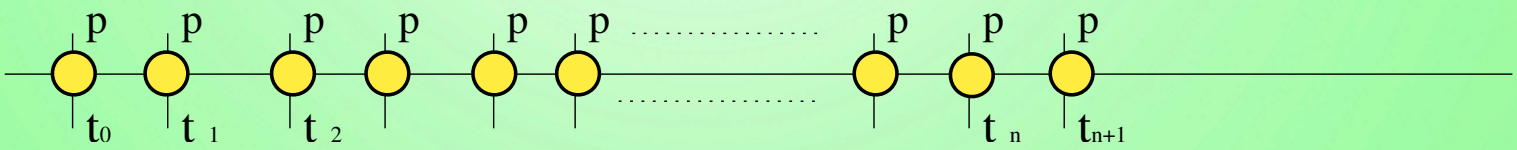
$serv(n,t)$ Fahrstuhl bedient Stockwerk n

Temporale Logik für die Informatik: Pnueli 1977:

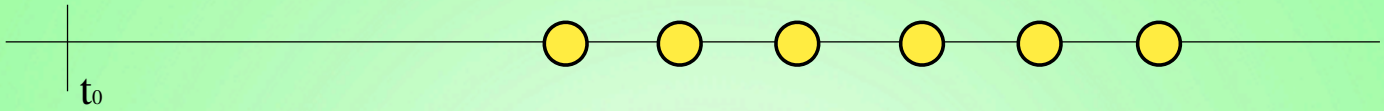
Linear Temporal Logic: LTL

$\Diamond p$ irgendwann einmal gilt p

$\Box p$ von jetzt an gilt immer p



$\Diamond\Box p$ bedeutet?



$\Box\Diamond p$ bedeutet?

“Abwickeln” der Kripke-Struktur

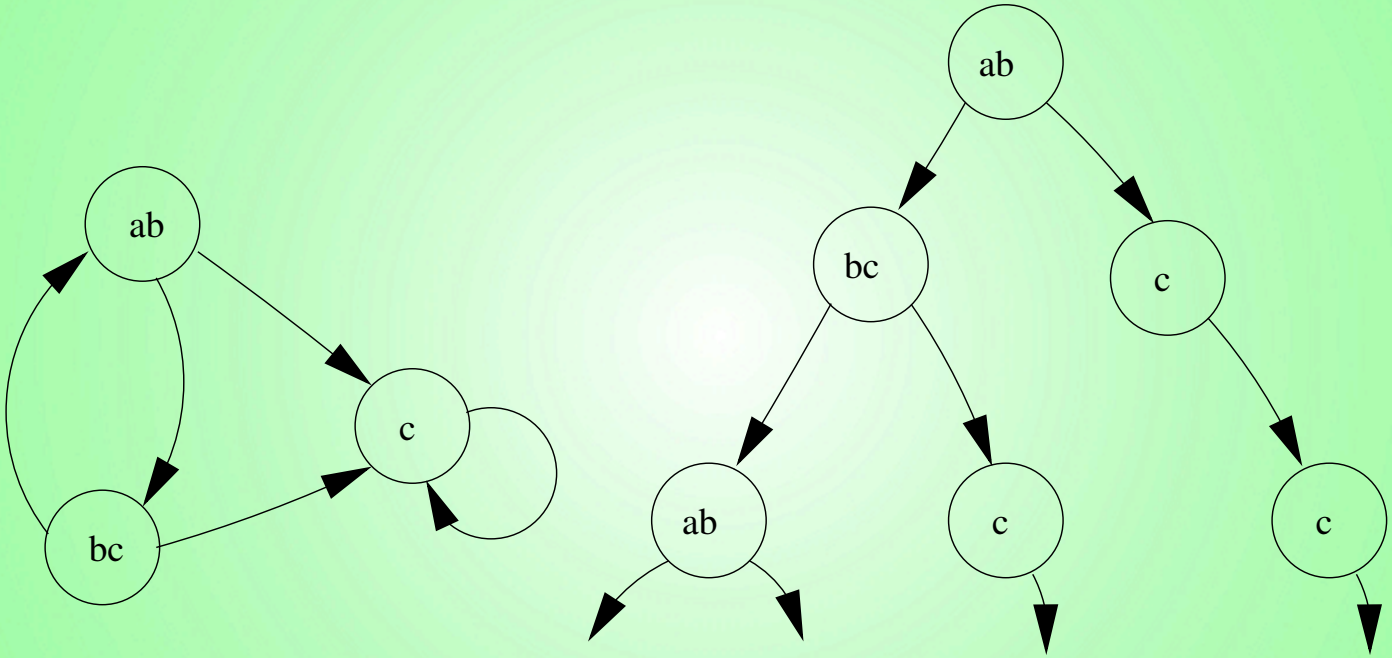
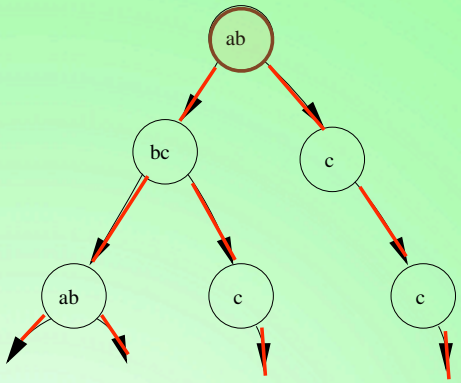


Abbildung 5.15: Abwicklung einer Kripke-Struktur

Zustandsquantoren

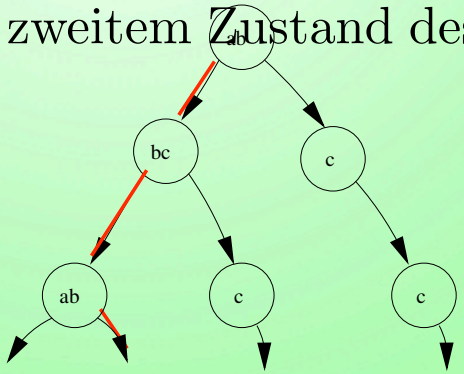


A “für alle Pfade”,

E “es gibt ein Pfad”,

Pfadquantoren

Xp *next time* : p gilt im zweitem Zustand des Pfades (vorher $\bigcirc$),



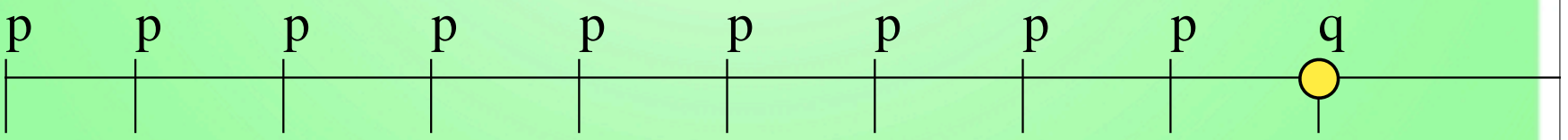
Pfadquantoren

Xp *next time* : p gilt im zweitem Zustand des Pfades (vorher $\bigcirc$),

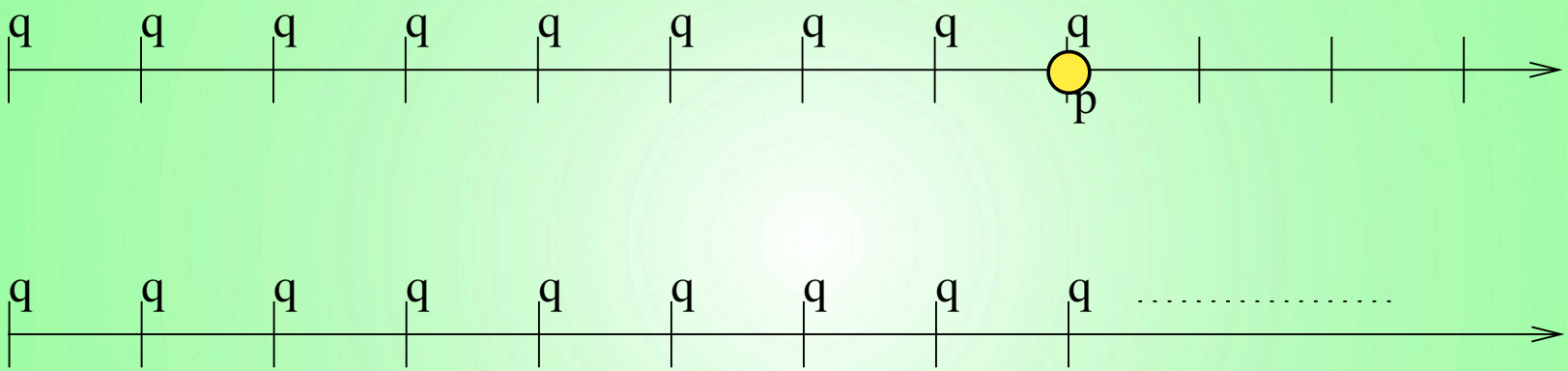
Fp *eventually, in the future* : p gilt in einem Zustand des Pfades (vorher $\Diamond$),

Gp *always, globally* : p gilt in allen Zuständen des Pfades (vorher $\Box$),

pUq *until* : es gibt einen Zustand auf dem Pfad, in dem q gilt und vor diesem Zustand gilt immer p.



pRq *release* : Dual zu pUq. q gilt bis einschließlich des ersten Zustands, in dem p gilt oder q gilt immer.



Angenommen, dass f_1, f_2 Zustands- und g_1, g_2 Pfad-Formeln sind, so ist $\models$ definiert durch:

8. $M, \pi \models \neg g_1 \quad \Leftrightarrow \quad M, \pi \not\models g_1.$
9. $M, \pi \models g_1 \vee g_2 \quad \Leftrightarrow \quad M, \pi \models g_1$ oder $M, \pi \models g_2.$
10. $M, \pi \models g_1 \wedge g_2 \quad \Leftrightarrow \quad M, \pi \models g_1$ und $M, \pi \models g_2.$
11. $M, \pi \models X g_1 \quad \Leftrightarrow \quad M, \pi^1 \models g_1.$
12. $M, \pi \models F g_1 \quad \Leftrightarrow \quad \exists k \geq 0 \,.\, M, \pi^k \models g_1.$
13. $M, \pi \models G g_1 \quad \Leftrightarrow \quad \forall k \geq 0 \,.\, M, \pi^k \models g_1.$

14. $M, \pi \models g_1 U g_2 \quad \Leftrightarrow \quad \exists k \geq 0 \,.\, M, \pi^k \models g_2$ und für alle $0 \leq j < k$ gilt $M, \pi^j \models g_1.$
15. $M, \pi \models g_1 R g_2 \quad \Leftrightarrow \quad \forall j \geq 0$, wenn für jeden $i < j$ $M, \pi^i \not\models g_1$ gilt, dann $M, \pi^j \models g_2.$

Model Checking

Für eine gegebene Kripke-Struktur $M = (S, R, L)$ und eine gegebene temporal-logische Formel f ist zu berechnen:

$$\{s \in S \mid M, s \models f\}$$

M ist hier als Graph explizit gegeben.