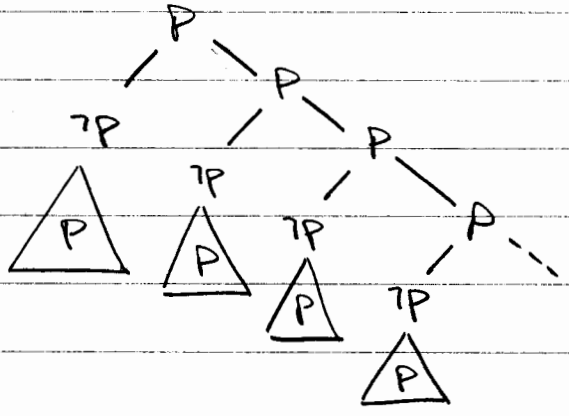
Nachtrag zu letzter Vorlesung:

Clarke-Buch: LTL-Formel $FG\,p$ nicht in CTL darstellbar

Frage: $FG\,p \equiv AF(AG\,p)$

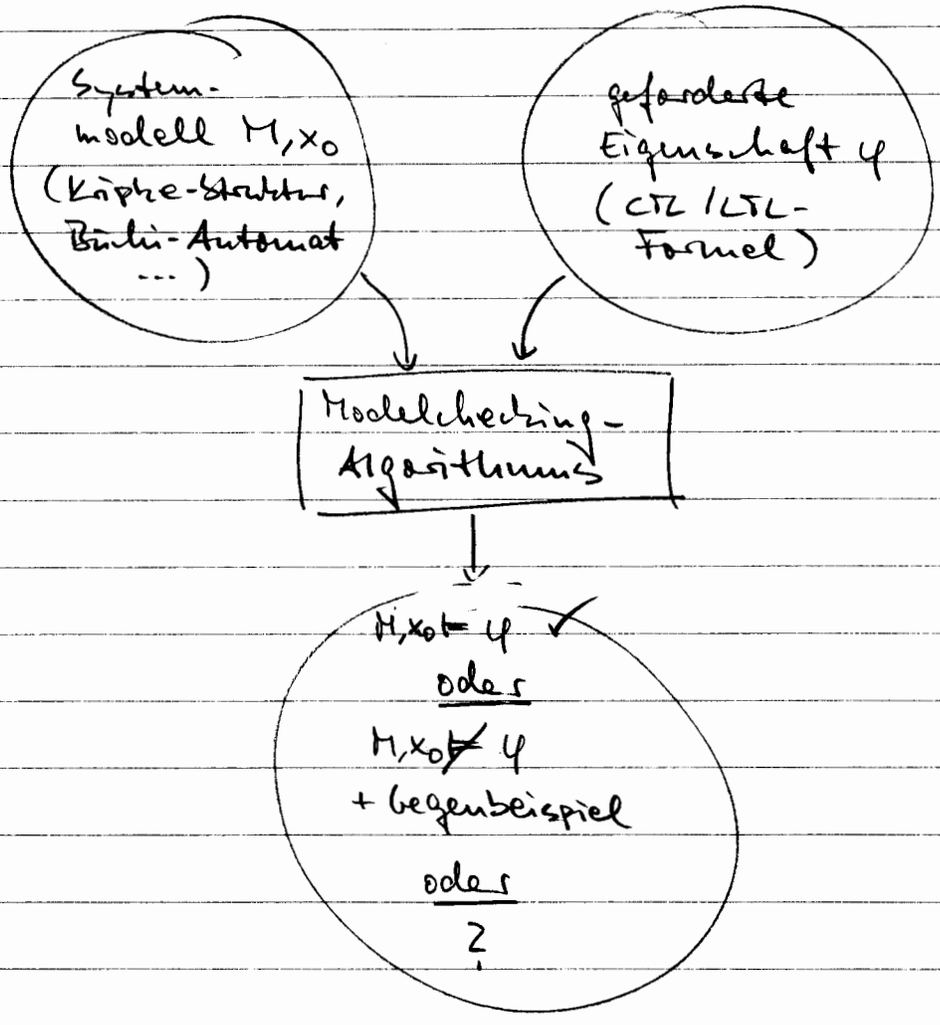
Gegenbeispiel (Dank an Philipp Rhode):




$FG\,p$? 1. Bestimme Menge aller Pfade
2. Für jeden Pfad gilt:
entweder: $Gp$ (der eine rechts außen)
oder: Es gibt nur einen Knoten mit $\neg p$.
$\Rightarrow FG\,p$ gilt.

$AF(AG\,p)$? Es muss möglich sein, auf allen Pfaden einen Knoten zu finden, ab dem $AG\,p$ gilt.
Das ist auf dem Pfad rechts außen nicht möglich, da immer die Möglichkeit besteht, noch nach links in einen Knoten mit $\neg p$ abzubiegen.
$\Rightarrow AF(AG\,p)$ gilt nicht.

## Modelchecking:




## CTL - Modelchecking

1. Umformung der CTL-Formel für φ, so dass nur noch verwendet werden: ¬, ∨, EX, EU, EG

$$AG\,p = \neg EF \neg p$$
$$AF\,p = \neg EG \neg p$$
$$AX\,p = \neg EX \neg p$$

Frage u. Hinweis auf Schreibweise

$$A(p\,U\,q) = \underbrace{AF\,q}_{} \wedge \neg E(\neg p\,U\,\neg(p \vee q))$$

$$= \neg EG \neg q \wedge \neg E(\neg p\,U\,\neg(p \vee q))$$
$$= \neg(EG \neg q \vee E(\neg p\,U\,\neg(p \vee q)))$$

$EF\ p = E(true\ U\ p)$

2. Für jede Unterformel von innen (atomare Propositionen) nach außen in der Verschachtelung: Absuchen des Transitionssystems und markieren der Zustände mit den gültigen Unterformeln.

Bsp.: $\varphi = E(f\ U\ (\neg EX\ g)) \vee EG\ \neg h$

von innen nach außen:

1. $f, g, h$
2. $EX\ g, \neg h$
3. $\neg EX\ g, EG\ \neg h$
4. $E(f\ U\ \neg EX\ g)$
5. $\varphi$

⇒ Folgende Teilalgorithmen werden gebraucht:

(a) $M, x_0 \models p$

(b) $M, x_0 \models \neg p$

(c) $M, x_0 \models p_1 \vee p_2$

(d) $M, x_0 \models EX\ p$

(e) $M, x_0 \models E(p_1\ U\ p_2)$

(f) $M, x_0 \models EG\ p$

(a) - (d): Alle Zustände einmal besuchen und Gültigkeit der Unterformel analysieren anhand der bereits untersuchten Unterformeln für diesen Zustand bzw. seine Folgezustände (EX).

(e): $M, x_0 \models E(f_1 \, U \, f_2)$:

~~Für alle Zustände, in denen $f_2$ gilt:~~

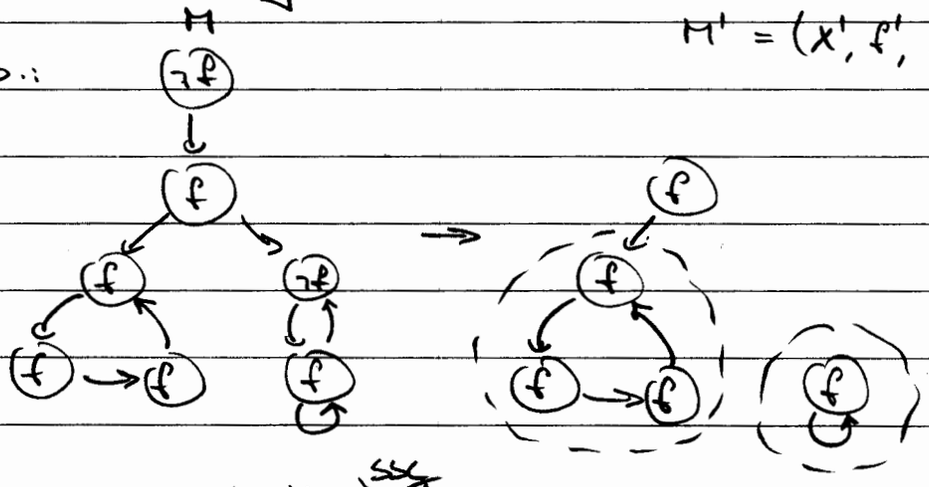
Rückwärtssuche: alle Zustände, die über einen Pfad erreichbar sind, auf dem jeder Zustand mit $f_1$ markiert ist, werden mit $E(f_1 U f_2)$ markiert.

(f): $M, x_0 \models EG\, f$

1. Eliminierung aller Zustände mit $\neg f$.

$M' = (X', f', \rightarrow')$

Bsp.:



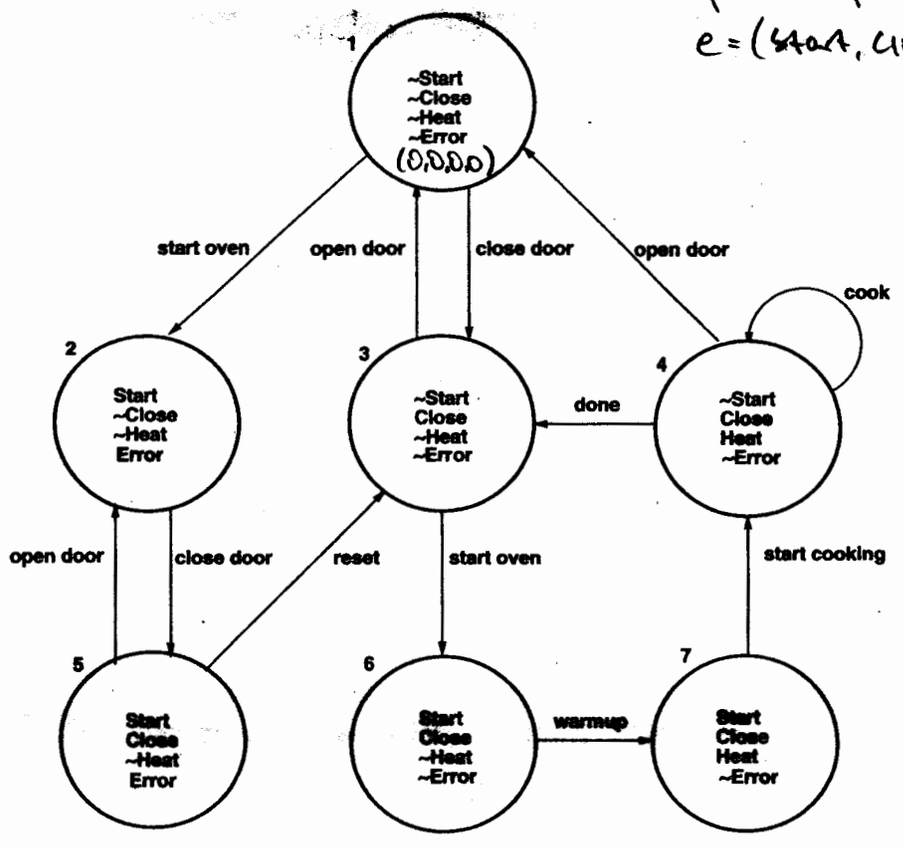

2. Bestimmung nicht-trivialer stark zus.hängender Teilgraphen in $M'$ (des Mengen~~graphen~~ alter SSK)

3. $M, x \models EG\, f \iff$

1. $x \in X'$
2. Es existiert ein Pfad von $x$ in einen ~~nicht-triv. stark zus.hängenden~~ Teilgraphen von $M'$

Bsp. aus Clarke et al, Model Checking, MIT Press, 1999:
Mikrowelle

Eigenschaftsvektor (Tupel)
e = (start, close, Heat, ~~Error~~)




(stehen lassen)

Zu überprüfende CTL-Formel: AG(start → AF Heat)

$= \neg EF \neg(\text{start} \rightarrow AF\,\text{Heat})$

$= \neg EF(\text{start} \wedge \neg AF\,\text{Heat})$

$= \neg EF(\text{start} \wedge \neg EG \neg \text{Heat})$

(EF als Abkürzung für E(true U f))

$= \neg E(\text{true}\ U\ (\text{start} \wedge EG \neg \text{Heat})$

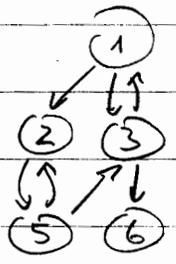
Notation: $S(p)$ = Menge aller Zustände, die mit der Unterformel $p$ markiert sind.

$$\neg E\ (\text{true}\ U\ (\text{start} \wedge EG \neg \text{Heat}))$$

1. $S(\text{start}) = \{2,5,6,7\}$, $S(\neg \text{Heat}) = \{1,2,3,5,6\}$
   (in Graph markieren)

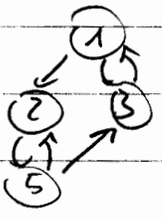
2. $S(EG \neg \text{Heat}) = \{1,2,3,5\}$

$M'$:




(Menge nicht-triv. st. zus. hängender Teilgraphen

SSC:




3. $S(\text{start} \wedge EG \neg \text{Heat}) = \{2,5\}$

4. $S(E(\text{true}\ U\ (\text{start} \wedge EG \neg \text{Heat})) = \{1,2,3,4,5,6,7\}$

   Beginnen mit $\{2,5\}$,
   dann rückwärts

5. $S(\neg E(\text{true}\ U\ (\text{start} \wedge EG \neg \text{Heat})) = \emptyset$

⇒ Anforderung ist nicht erfüllt.

## LTL-Modelchecking

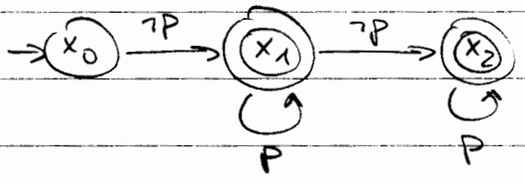
$M, x_0$ : Büchi-Automat $A_M$ ($F \subseteq X$ : Menge von akzeptierenden Zuständen

$\varphi$ : LTL-Formel

Lauf ist akzeptierend, wenn min. ein Zustand aus $F$ unendlich oft besucht wird.)

Bsp.:

$A_M$:




$\varphi$:

$FG\,p$

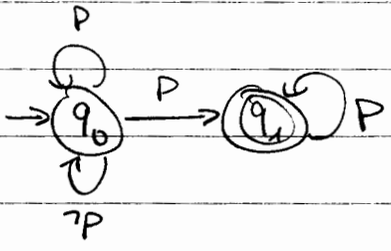
$\mathcal{L}\{M\} = (\neg p\, p^\omega) + (\neg p\, p^* \neg p\, p^\omega)$

↓

(min 1x ?)

1. $FG\,p \;\hat{=}\; (\neg p + p)^* p^\omega$

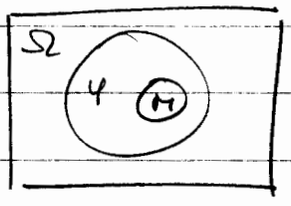
2. dazugehöriger Büchi-Automat $A_\varphi$:




$M, x_0 \models \varphi$

$\Leftrightarrow \quad \mathcal{L}\{A_M\} \subseteq \mathcal{L}\{A_\varphi\}$

$\Leftrightarrow \quad \mathcal{L}\{A_M\} \cap \overline{\mathcal{L}\{A_\varphi\}} = \emptyset$




Benötigt wird:
1. Schnitt
2. Komplement
3. Test auf Leere

## Symbolic Model Checking (z.B. SMV):

- bisher: Suche über einzelne Zustände

  Problem: Davon gibt es viele (exponentiell in Anzahl der Teilsysteme)

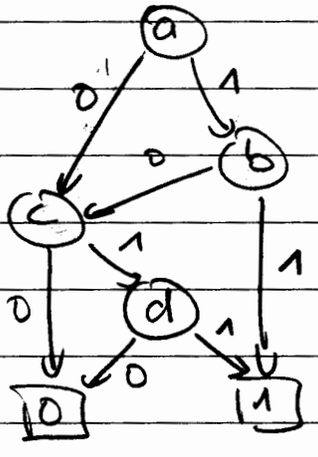
- Effizienzgewinn (praktisch) durch „symbolische" Repräsentation von Zustandsmengen.

= Logische Formel repräsentiert die Menge von Zuständen, in denen sie gilt.

- Effiziente Datenstrukturen für diese Formeln:

  OBDDs (Ordered Binary Decision Diagrams, Akers, 1960; Bryant 1986)

Bsp.: $(a \wedge b) \vee (c \wedge d)$




→ Details siehe Clarke et al.