Model Checking

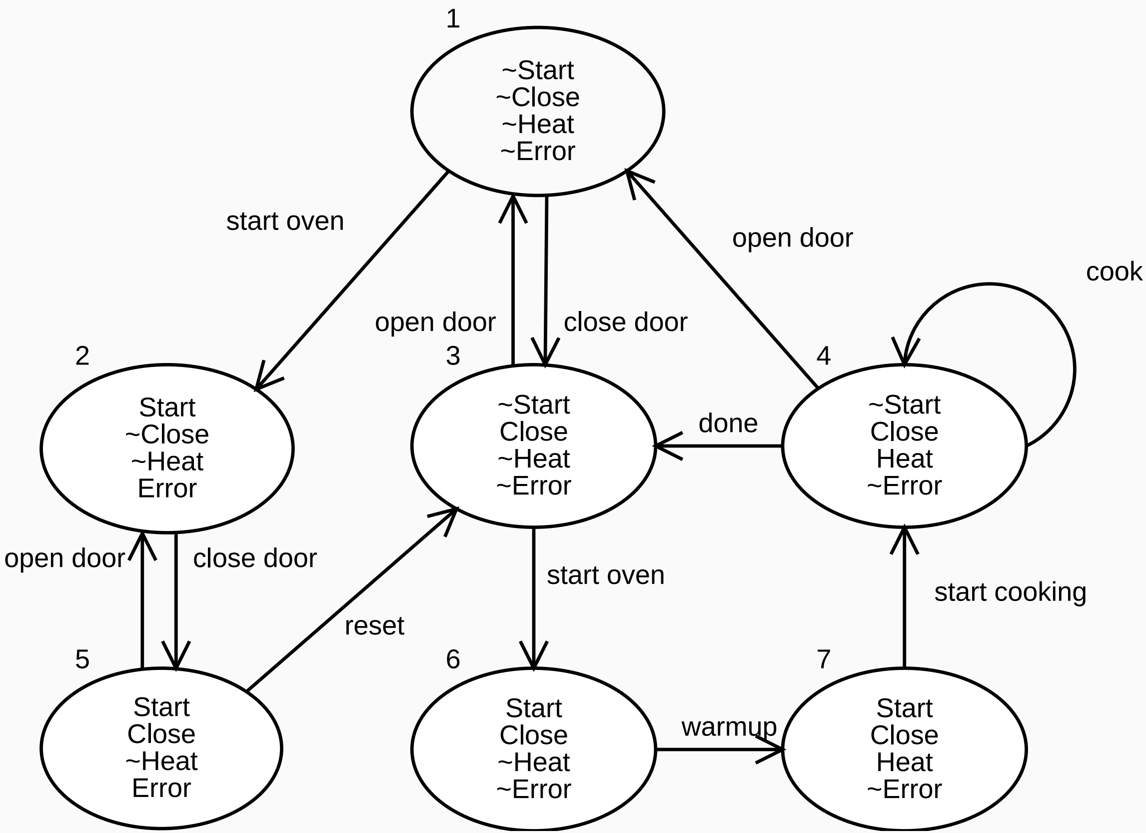
Andreas Zeller

Lehrstuhl Softwaretechnik
Universität des Saarlandes, Saarbrücken

Übersicht

- Motivation
- Temporale Logik
- Model Checking
- Binary Decision Diagrams
- Boolesche Programme

Ein Mikrowellen-Herd



Ein Mikrowellen-Herd (2)

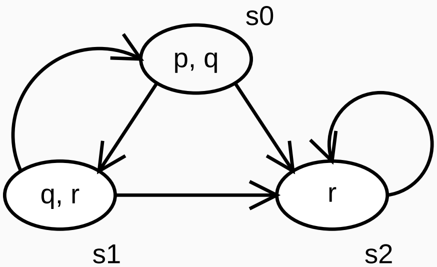
Offene Fragen:

- Folgt auf den Zustand `Start` stets der Zustand `Heat`?
- Kann `Heat` vor `Close` auftreten?
- Kann `Start` nach `Error` auftreten?

Temporale Logik

Zur Beschreibung von *Abläufen in der Zeit* setzt man *temporale Logik* ein.

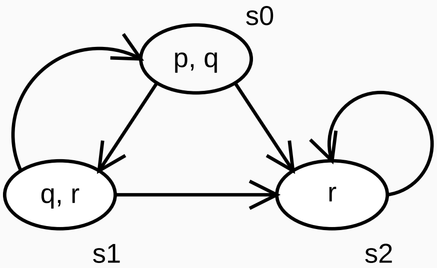
Semantisches Modell: Zustandsübergänge werden in einen *unendlichen Baum* („Berechnungsbaum“) entflochten

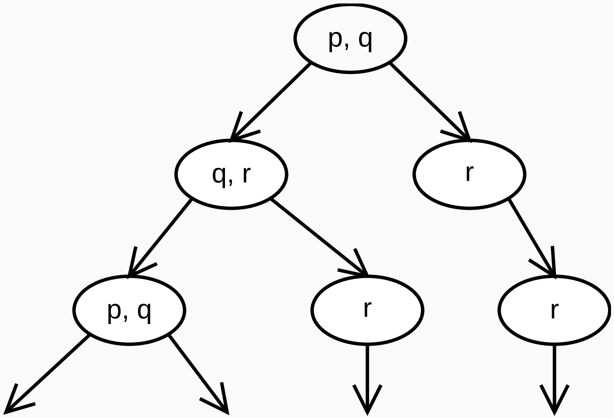


Temporale Logik

Zur Beschreibung von *Abläufen in der Zeit* setzt man *temporale Logik* ein.

Semantisches Modell: Zustandsübergänge werden in einen *unendlichen Baum* („Berechnungsbaum“) entflochten

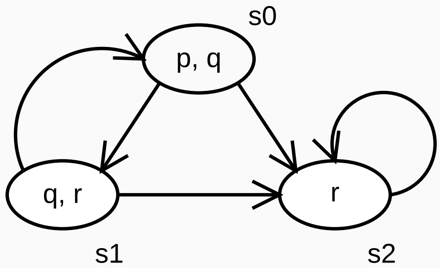


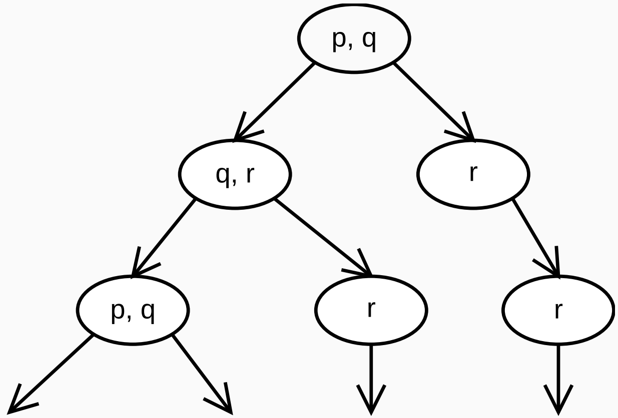


Temporale Logik

Zur Beschreibung von *Abläufen in der Zeit* setzt man *temporale Logik* ein.

Semantisches Modell: Zustandsübergänge werden in einen *unendlichen Baum* („Berechnungsbaum“) entflochten





Über die *Pfade* des Berechnungsbaums lassen sich Aussagen treffen – mit der *Computation Tree Logic* (CTL)

CTL – Syntax

Eine Formel g in CTL ist aufgebaut wie folgt:

$$g ::= T \,|\, g \,|\, \neg g \,|\, g_1 \vee g_2 \,|\, g_1 \wedge g_2$$

CTL – Syntax

Eine Formel g in CTL ist aufgebaut wie folgt:

$$g ::= T \,|\, g \,|\, \neg g \,|\, g_1 \vee g_2 \,|\, g_1 \wedge g_2$$

EX g g gilt in einem Folgezustand („next“)

AX g g gilt in allen Folgezuständen

CTL – Syntax

Eine Formel g in CTL ist aufgebaut wie folgt:

$$g ::= T \,|\, g \,|\, \neg g \,|\, g_1 \vee g_2 \,|\, g_1 \wedge g_2$$

EX g — g gilt in einem Folgezustand („next“)

AX g — g gilt in allen Folgezuständen

EF g — g gilt irgendwann in einem Pfad („finally“)

AF g — g gilt irgendwann in allen Pfaden

CTL – Syntax

Eine Formel g in CTL ist aufgebaut wie folgt:

$$g ::= T \,|\, g \,|\, \neg g \,|\, g_1 \vee g_2 \,|\, g_1 \wedge g_2$$

EX g	g gilt in einem Folgezustand („next“)
AX g	g gilt in allen Folgezuständen
EF g	g gilt irgendwann in einem Pfad („finally“)
AF g	g gilt irgendwann in allen Pfaden
EG g	g gilt stets in einem Pfad („globally“)
AG g	g gilt stets in allen Pfaden

CTL – Syntax

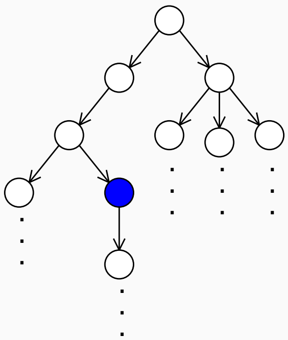
Eine Formel g in CTL ist aufgebaut wie folgt:

$$g ::= T \,|\, g \,|\, \neg g \,|\, g_1 \vee g_2 \,|\, g_1 \wedge g_2$$

EX g	g gilt in einem Folgezustand („next“)
AX g	g gilt in allen Folgezuständen
EF g	g gilt irgendwann in einem Pfad („finally“)
AF g	g gilt irgendwann in allen Pfaden
EG g	g gilt stets in einem Pfad („globally“)
AG g	g gilt stets in allen Pfaden
E$(f$ **U** $g)$	f gilt in einem Pfad, bis g gilt („until“)
A$(f$ **U** $g)$	f gilt in allen Pfaden, bis g gilt

Die wichtigsten CTL-Idiome

(in Klammern: alternative Schreibweise aus *Modallogik*)

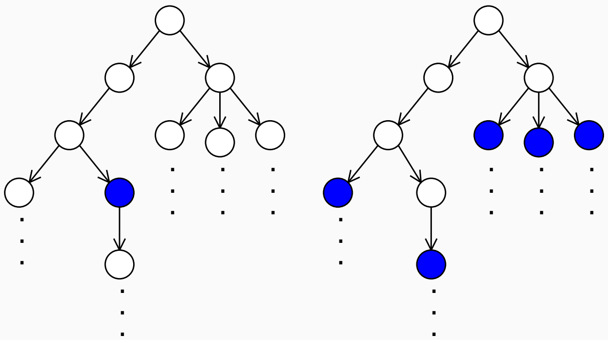


$M, s_0 \models \mathbf{EF}\ f$

$(M, s_0 \models \exists\Diamond\ f)$

Die wichtigsten CTL-Idiome

(in Klammern: alternative Schreibweise aus *Modallogik*)

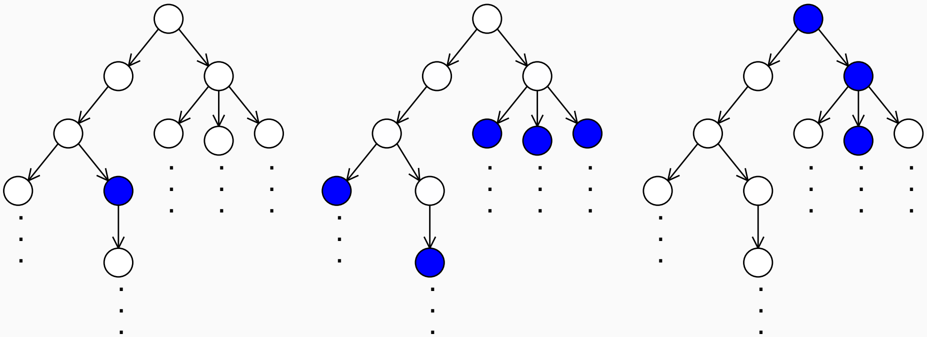


$M, s_0 \models \mathbf{EF}\, f$ $\quad$ $M, s_0 \models \mathbf{AF}\, f$

$(M, s_0 \models \exists \Diamond\, f)$ $\quad$ $(M, s_0 \models \forall \Diamond\, f)$

Die wichtigsten CTL-Idiome

(in Klammern: alternative Schreibweise aus *Modallogik*)

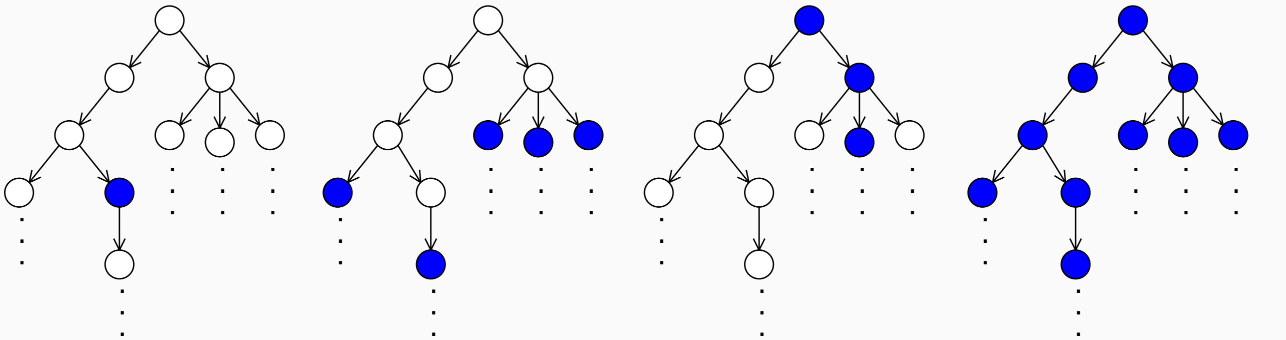


$M, s_0 \vDash \mathbf{EF}\, f$ $\quad$ $M, s_0 \vDash \mathbf{AF}\, f$ $\quad$ $M, s_0 \vDash \mathbf{EG}\, f$

$(M, s_0 \vDash \exists\Diamond\, f)$ $\quad$ $(M, s_0 \vDash \forall\Diamond\, f)$ $\quad$ $(M, s_0 \vDash \exists\Box\, f)$

Die wichtigsten CTL-Idiome

(in Klammern: alternative Schreibweise aus *Modallogik*)



$M, s_0 \models \mathbf{EF}\, f$	$M, s_0 \models \mathbf{AF}\, f$	$M, s_0 \models \mathbf{EG}\, f$	$M, s_0 \models \mathbf{AG}\, f$
$(M, s_0 \models \exists\Diamond\, f)$	$(M, s_0 \models \forall\Diamond\, f)$	$(M, s_0 \models \exists\Box\, f)$	$(M, s_0 \models \forall\Box\, f)$

$M, s \models f$: f gilt im Modell M im Zustand s

CTL – Beispiele

$\mathbf{AG}\,(\neg p)$ „p gilt nie“

CTL – Beispiele

$\mathbf{AG}\,(\neg p)$	„p gilt nie“
$\mathbf{AF}\,\mathbf{A}(p\ \mathbf{U}\ q)$	„p gilt immer, bis q gilt“

CTL – Beispiele

$\mathbf{AG}\,(\neg p)$	„p gilt nie“
$\mathbf{AF}\,\mathbf{A}(p\ \mathbf{U}\ q)$	„p gilt immer, bis q gilt“
$\mathbf{EF}\,(p)$	„Irgendwo wird irgendwann p gelten“

CTL – Beispiele

$\mathbf{AG}\,(\neg p)$	„p gilt nie“
$\mathbf{AF}\,\mathbf{A}(p\;\mathbf{U}\;q)$	„p gilt immer, bis q gilt“
$\mathbf{EF}\,(p)$	„Irgendwo wird irgendwann p gelten“
$\mathbf{AG}\,(\mathit{Start} \rightarrow \mathbf{AF}\,\mathit{Heat})$	„Aus *Start* folgt stets *Heat*“

CTL – Beispiele

$\mathbf{AG}\ (\neg p)$	„p gilt nie“
$\mathbf{AF}\ \mathbf{A}(p\ \mathbf{U}\ q)$	„p gilt immer, bis q gilt“
$\mathbf{EF}\ (p)$	„Irgendwo wird irgendwann p gelten“
$\mathbf{AG}\ (\mathit{Start} \rightarrow \mathbf{AF}\ \mathit{Heat})$	„Aus *Start* folgt stets *Heat*“
$\mathbf{AF}\ (\mathbf{AG}\ \mathit{Deadlock})$	„Stets gibt es irgendwann einen Deadlock, der für immer anhält“

CTL – Beispiele

$\mathbf{AG}\,(\neg p)$	„p gilt nie“
$\mathbf{AF}\,\mathbf{A}(p\,\mathbf{U}\,q)$	„p gilt immer, bis q gilt“
$\mathbf{EF}\,(p)$	„Irgendwo wird irgendwann p gelten“
$\mathbf{AG}\,(\mathit{Start} \rightarrow \mathbf{AF}\,\mathit{Heat})$	„Aus *Start* folgt stets *Heat*“
$\mathbf{AF}\,(\mathbf{AG}\,\mathit{Deadlock})$	„Stets gibt es irgendwann einen Deadlock, der für immer anhält“

$$\mathbf{AG}\,(\mathit{floor} = 2 \wedge \mathit{direction} = \mathit{up} \wedge \mathit{ButtonPressed5} \rightarrow (\mathit{direction} = \mathit{up}\,\mathbf{U}\,\mathit{floor} = 5)$$

„Der Aufzug ändert seine Richtung nicht, bis er den 5. Stock erreicht hat.“

CTL – Semantik

(M: Modell, s: Zustand; $L(S)$: Gültige Aussagen in s)

$$
\begin{array}{lll}
M,s \models \top, M,s \not\models \bot & \text{für alle Zustände } s & \\
M,s \models p & \Leftrightarrow p \in L(s) & \\
M,s \models \neg f & \Leftrightarrow M,s \not\models f & \\
M,s \models f_1 \wedge f_2 & \Leftrightarrow M,s \models f_1 \quad \text{und} \quad M,s \models f_2 & \\
M,s \models f_1 \vee f_2 & \Leftrightarrow M,s \models f_1 \quad \text{oder} \quad M,s \models f_2 & \\
M,s \models f_1 \rightarrow f_2 & \Leftrightarrow M,s \not\models f_1 \quad \text{oder} \quad M,s \models f_2 &
\end{array}
$$

CTL – Semantik (2)

(M: Modell, s: Zustand)

$$
\begin{array}{lll}
M, s \models \mathbf{AX}\, f & \Leftrightarrow & \forall s_1 | s \to s_1 \cdot M, s_1 \models f \\
M, s \models \mathbf{EX}\, f & \Leftrightarrow & \exists s_1 | s \to s_1 \cdot M, s_1 \models f \\
M, s_1 \models \mathbf{AG}\, f & \Leftrightarrow & \forall \pi = (s_1 \to s_2 \to \ldots) \cdot \forall s_i \cdot M, s_i \models f \\
M, s_1 \models \mathbf{EG}\, f & \Leftrightarrow & \exists \pi = (s_1 \to s_2 \to \ldots) \cdot \forall s_i \cdot M, s_i \models f \\
M, s_1 \models \mathbf{AF}\, f & \Leftrightarrow & \forall \pi = (s_1 \to s_2 \to \ldots) \cdot \exists s_i \cdot M, s_i \models f \\
M, s_1 \models \mathbf{EF}\, f & \Leftrightarrow & \exists \pi = (s_1 \to s_2 \to \ldots) \cdot \exists s_i \cdot M, s_i \models f \\
M, s_1 \models \mathbf{A}(f_1\, \mathbf{U}\, f_2) & \Leftrightarrow & \forall \pi = (s_1 \to s_2 \to \ldots) \cdot \\
 & & \quad \exists s_i \cdot (M, s_i \models f_2) \wedge \big(\forall j < i \cdot (M, s_j \models f_1)\big) \\
M, s_1 \models \mathbf{E}(f_1\, \mathbf{U}\, f_2) & \Leftrightarrow & \exists \pi = (s_1 \to s_2 \to \ldots) \cdot \\
 & & \quad \exists s_i \cdot (M, s_i \models f_2) \wedge \big(\forall j < i \cdot (M, s_j \models f_1)\big)
\end{array}
$$

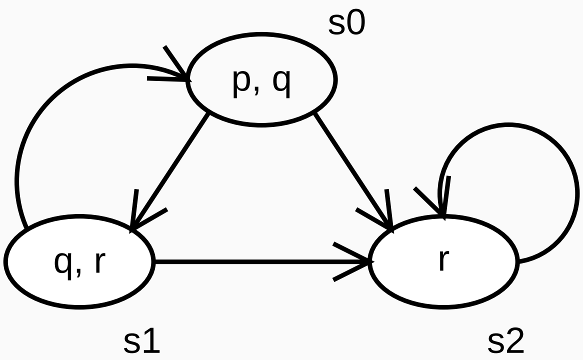
CTL – Einige Äquivalenzen

- $f \vee g \equiv \neg(\neg f \wedge \neg g)$
- $f \rightarrow g \equiv \neg(f \wedge \neg g)$
- $\mathbf{EG}\, f \equiv \neg \mathbf{AF}\, \neg f$
- $\mathbf{AG}\, f \equiv \neg \mathbf{EF}\, \neg f$
- $\neg \mathbf{AX}\, f \equiv \neg \mathbf{AX}\, \neg f$
- $\mathbf{AF}\, f \equiv \mathbf{A}(\top\ \mathbf{U}\ f)$
- $\mathbf{EF}\, f \equiv \mathbf{E}(\top\ \mathbf{U}\ f)$
- $\mathbf{A}(f\ \mathbf{U}\ g) \equiv \neg\big(\mathbf{E}(\neg g\ \mathbf{U}\ (\neg f \wedge \neg g)) \vee \mathbf{EG}\, \neg g\big)$

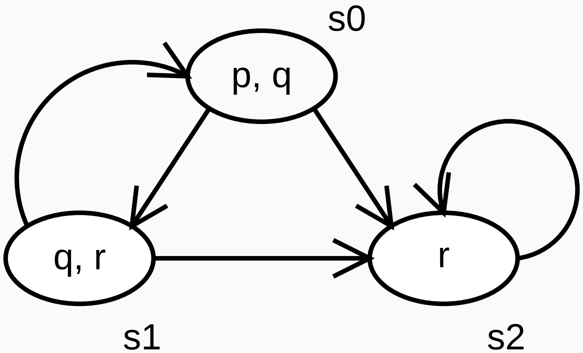
Tatsächlich reichen die Operatoren **AF**, **E U** und **EX** zusammen mit $\wedge$ und $\neg$ aus, um alle CTL-Formeln zu schreiben (Übung).

Was gilt?

$$M, s_0 \models p \wedge q$$



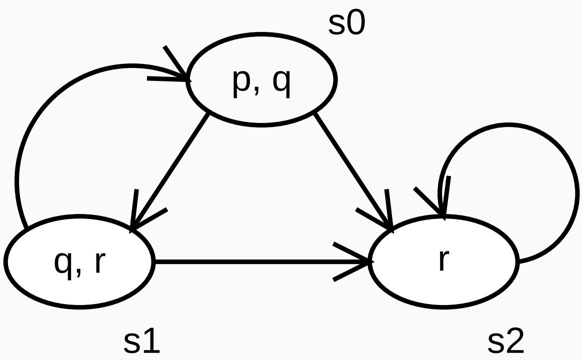
Was gilt?



$M, s_0 \models p \wedge q$ ja

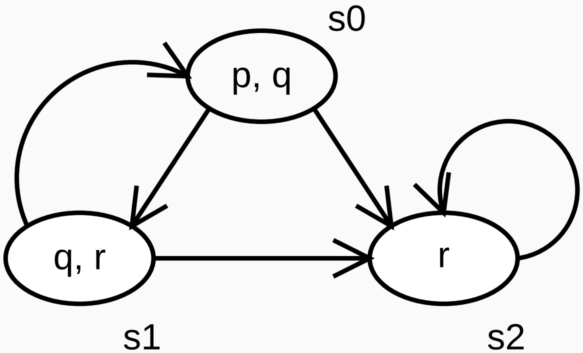
$M, s_0 \models \neg r$

Was gilt?



$M, s_0 \models p \wedge q$	ja
$M, s_0 \models \neg r$	ja
$M, s_0 \models \top$	

Was gilt?



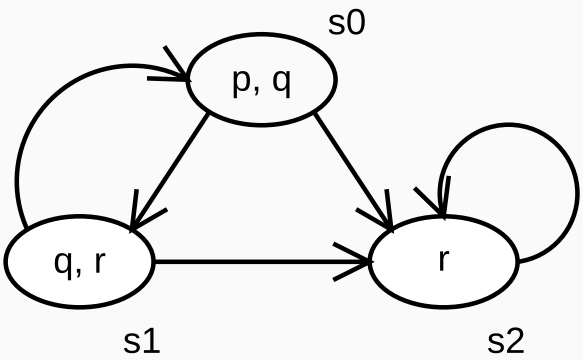
$M, s_0 \models p \wedge q$ ja

$M, s_0 \models \neg r$ ja

$M, s_0 \models \top$ ja

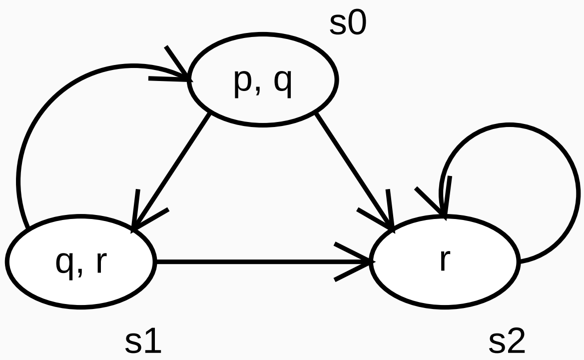
$M, s_0 \models \mathbf{EX}\,(q \wedge r)$

Was gilt?



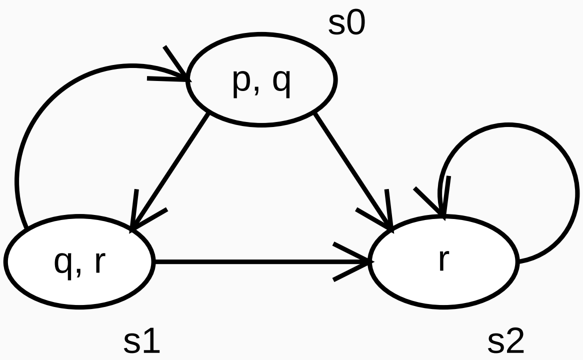
$M, s_0 \models p \land q$	ja
$M, s_0 \models \neg r$	ja
$M, s_0 \models \top$	ja
$M, s_0 \models \textbf{EX}\,(q \land r)$	ja
$M, s_0 \models \neg\textbf{AX}\,(q \land r)$	

Was gilt?



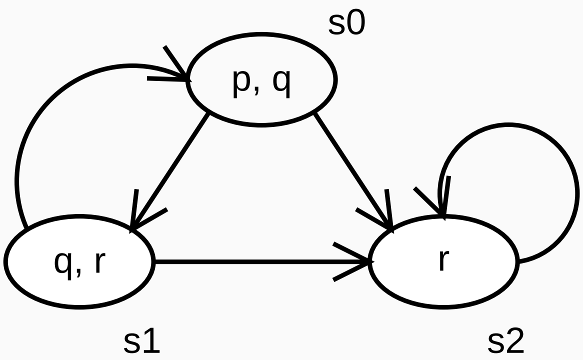
$M, s_0 \models p \wedge q$	ja
$M, s_0 \models \neg r$	ja
$M, s_0 \models \top$	ja
$M, s_0 \models \textbf{EX}\ (q \wedge r)$	ja
$M, s_0 \models \neg\textbf{AX}\ (q \wedge r)$	ja
$M, s_0 \models \neg\textbf{EF}\ (p \wedge r)$	

Was gilt?



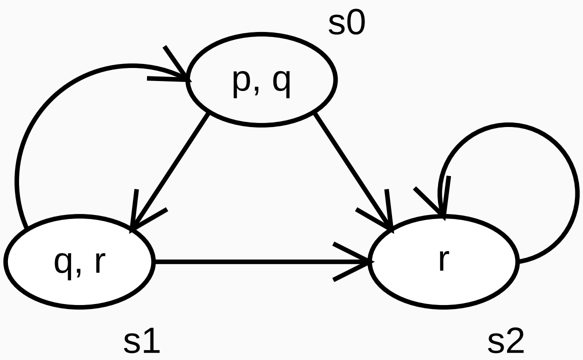
$M, s_0 \models p \wedge q$	ja
$M, s_0 \models \neg r$	ja
$M, s_0 \models \top$	ja
$M, s_0 \models \mathbf{EX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{AX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{EF}\,(p \wedge r)$	ja
$M, s_2 \models \mathbf{EG}\,r$	

Was gilt?



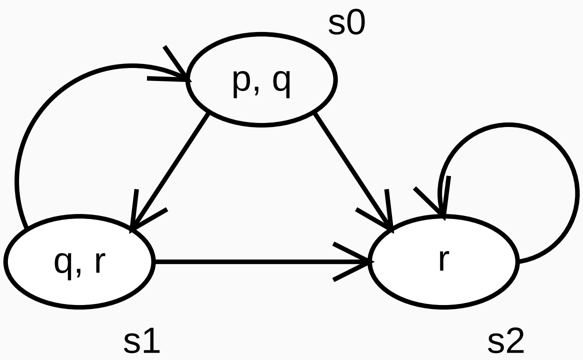
$M, s_0 \models p \wedge q$	ja
$M, s_0 \models \neg r$	ja
$M, s_0 \models \top$	ja
$M, s_0 \models \mathbf{EX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{AX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{EF}\,(p \wedge r)$	ja
$M, s_2 \models \mathbf{EG}\,r$	ja
$M, s_2 \models \mathbf{AG}\,r$	

Was gilt?



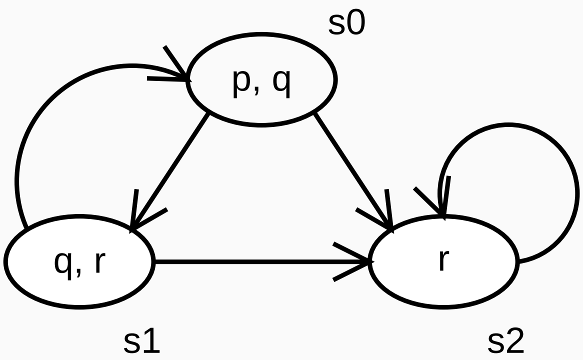
$M, s_0 \models p \wedge q$	ja
$M, s_0 \models \neg r$	ja
$M, s_0 \models \top$	ja
$M, s_0 \models \mathbf{EX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{AX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{EF}\,(p \wedge r)$	ja
$M, s_2 \models \mathbf{EG}\,r$	ja
$M, s_2 \models \mathbf{AG}\,r$	ja
$M, s_0 \models \mathbf{AF}\,r$	

Was gilt?



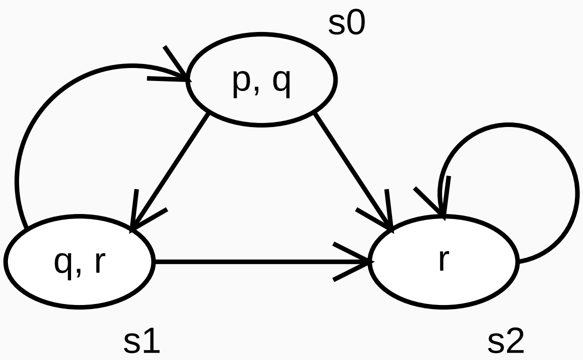
$M, s_0 \models p \wedge q$	ja
$M, s_0 \models \neg r$	ja
$M, s_0 \models \top$	ja
$M, s_0 \models \mathbf{EX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{AX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{EF}\,(p \wedge r)$	ja
$M, s_2 \models \mathbf{EG}\,r$	ja
$M, s_2 \models \mathbf{AG}\,r$	ja
$M, s_0 \models \mathbf{AF}\,r$	ja
$M, s_0 \models \mathbf{E}\big((p \wedge q)\,\mathbf{U}\,r\big)$	

Was gilt?



$M, s_0 \models p \wedge q$	ja
$M, s_0 \models \neg r$	ja
$M, s_0 \models \top$	ja
$M, s_0 \models \mathbf{EX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{AX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{EF}\,(p \wedge r)$	ja
$M, s_2 \models \mathbf{EG}\,r$	ja
$M, s_2 \models \mathbf{AG}\,r$	ja
$M, s_0 \models \mathbf{AF}\,r$	ja
$M, s_0 \models \mathbf{E}((p \wedge q)\,\mathbf{U}\,r)$	ja
$M, s_0 \models \mathbf{A}(p\,\mathbf{U}\,r)$	

Was gilt?



$M, s_0 \models p \wedge q$	ja
$M, s_0 \models \neg r$	ja
$M, s_0 \models \top$	ja
$M, s_0 \models \mathbf{EX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{AX}\,(q \wedge r)$	ja
$M, s_0 \models \neg\mathbf{EF}\,(p \wedge r)$	ja
$M, s_2 \models \mathbf{EG}\,r$	ja
$M, s_2 \models \mathbf{AG}\,r$	ja
$M, s_0 \models \mathbf{AF}\,r$	ja
$M, s_0 \models \mathbf{E}((p \wedge q)\,\mathbf{U}\,r)$	ja
$M, s_0 \models \mathbf{A}(p\,\mathbf{U}\,r)$	ja

Anwendung: CORBA-Protokoll

(Kamel, Leue, Holzmann et al 1998)

After sending an `SRequest`*, the GIOP agent should eventually receive a corresponding* `SReply`

$$\mathbf{AG}\ (SRequest \rightarrow \mathbf{AF}\ SReply)$$

The agent should never receive an `SReply` *for a request that is not outstanding*

$$\mathbf{AF}\ SReply \rightarrow \mathbf{A}\big(\neg SReply\ \mathbf{U}\ (SRequest \wedge \neg SReply)\big)$$

Anwendung: CORBA-Protokoll (2)

Servers may only issue `CloseConnection` *messages when* `Reply` *messages have been sent in response to all received* `Request` *messages that require replies*

$$\mathbf{AF}\, close \rightarrow \big(\mathbf{AG}\, (request \rightarrow \mathbf{A}(\mathbf{A}(\neg close\, \mathbf{U}\, reply)\, \mathbf{U}\, close))\big)$$

Anwendung: CORBA-Protokoll (2)

Servers may only issue `CloseConnection` *messages when* `Reply` *messages have been sent in response to all received* `Request` *messages that require replies*

$$\mathbf{AF}\, close \rightarrow \big(\mathbf{AG}\, (request \rightarrow \mathbf{A}(\mathbf{A}(\neg close\, \mathbf{U}\, reply)\, \mathbf{U}\, close))\big)$$

Was passiert bei $\pi = request, request, reply, close$?

Anwendung: CORBA-Protokoll (2)

Servers may only issue `CloseConnection` *messages when* `Reply` *messages have been sent in response to all received* `Request` *messages that require replies*

$$\mathbf{AF}\, close \rightarrow \big(\mathbf{AG}\,(request \rightarrow \mathbf{A}(\mathbf{A}(\neg close\,\mathbf{U}\, reply)\,\mathbf{U}\, close))\big)$$

Was passiert bei $\pi = request, request, reply, close$?

Verbesserte Fassung:

$$\mathbf{AF}\, close \rightarrow \big(\mathbf{AG}\,(request \rightarrow \mathbf{A}(\mathbf{A}(\neg close\,\mathbf{U}\, reply)\,\mathbf{U}\, close))\big) \wedge N$$

mit $N \equiv \#(request) = \#(reply)$

Model Checking

Gegeben:

- Modell M als endlicher Automat mit Ursprungszustand s_0
- CTL-Formel f (Spezifikation)

Model Checking prüft, ob alle Pfade durch den Automaten (= des Berechnungsbaums) die Spezifikation erfüllen ($M, s_0 \models f$)

Falls $M, s_0 \not\models f$, wird *Gegenbeispiel* konstruiert

Vorgehensweise

Wir bestimmen alle Zustände s_i aus M, die f erfüllen.
(Ist s_0 in diesen Zuständen, so gilt $M, s_0 \models f$)

Übersicht:

1. Wir schreiben die CTL-Formel so um, daß nur **AF** , **E U** und **EX** zusammen mit $\wedge$ und $\neg$ auftreten.
2. Wir *attribuieren* die Zustände des Modells mit erfüllten Teilformeln von f.

Attribuierung

Wir arbeiten die Formel f rekursiv durch – *bottom up* von den Teilformeln $f_1, f_2, \ldots$ zur Gesamtformel:

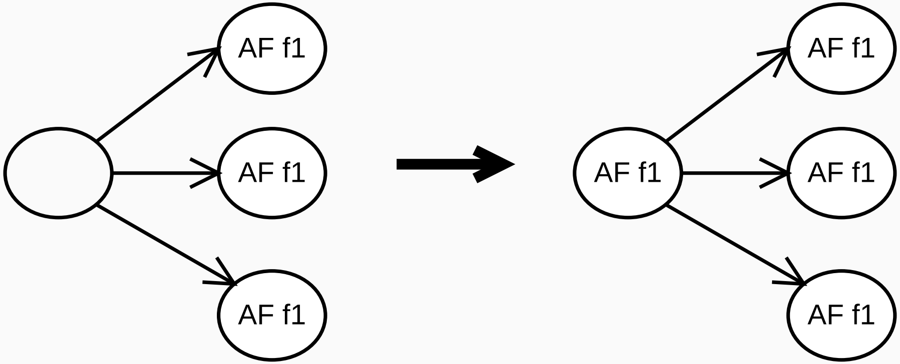
Hat f die Form

- $\perp$ – Keine Zustände werden attribuiert
- p – Attribuiere s mit p wenn $p \in L(s)$
- $f_1 \wedge f_2$ – Attribuiere s mit $f_1 \wedge f_2$, wenn s mit f_1 und f_2 attribuiert ist
- $\neg f_1$ – Attribuiere s mit $\neg f_1$, wenn s nicht mit f_1 attribuiert ist
- **EX** f_1 – Attribuiere jeden Zustand mit **EX** f_1, dessen Nachfolger mit f_1 attribuiert ist

Attribuierung (2)

Hat f die Form **AF** f_1:

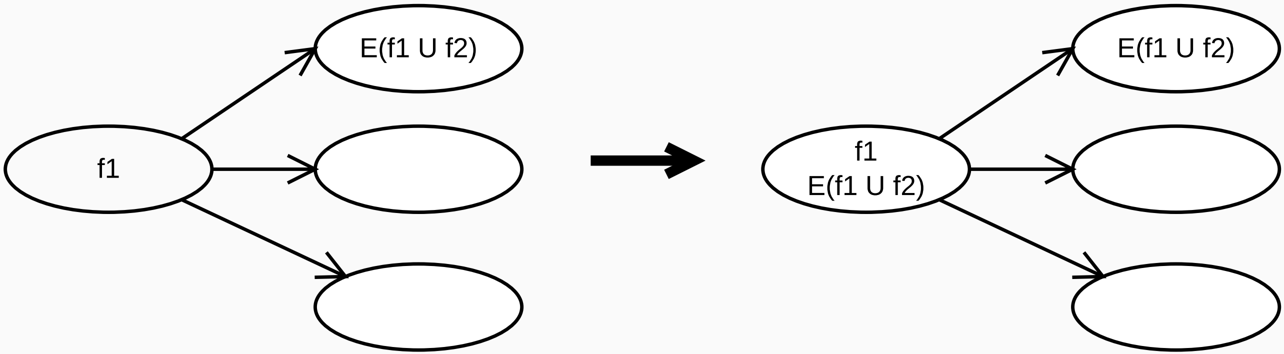
- Ist irgendein Zustand mit f_1 attribuiert, attribuiere ihn mit **AF** f_1
- Attribuiere jeden Zustand, dessen Nachfolger mit **AF** f_1 attribuiert sind, ebenfalls mit **AF** f_1 – bis keine Änderung mehr auftritt



Attribuierung (3)

Hat f die Form $\mathbf{E}(f_1 \mathbf{U} f_2)$:

- Ist irgendein Zustand mit f_2 attribuiert, attribuiere ihn mit $\mathbf{E}(f_1 \mathbf{U} f_2)$
- Attribuiere jeden Zustand, der mit f_1 attribuiert ist, mit $\mathbf{E}(f_1 \mathbf{U} f_2)$, wenn einer seiner Nachfolger mit $\mathbf{E}(f_1 \mathbf{U} f_2)$ attribuiert ist – bis keine Änderung mehr auftritt



Attribuierung (4)

Nachdem f und alle Unterformeln bearbeitet wurden, erfüllen genau die Zustände f, die mit f attribuiert sind.

Komplexität des Verfahrens: $O(f \cdot V \cdot (V + E))$ mit
f = Anzahl der Operatoren,
V = Anzahl der Zustände (Knoten),
E = Anzahl der Übergänge (Kanten)

Attribuierung (4)

Nachdem f und alle Unterformeln bearbeitet wurden, erfüllen genau die Zustände f, die mit f attribuiert sind.

Komplexität des Verfahrens: $O(f \cdot V \cdot (V + E))$ mit
f = Anzahl der Operatoren,
V = Anzahl der Zustände (Knoten),
E = Anzahl der Übergänge (Kanten)

Effizientere Variante in $O(f \cdot (V + E))$

- Basiert auf **EG** statt **AF**
- Benutzt Breitensuche auf Modell, um **E** zu berechnen
- Benutzt *stark zusammenhängende Komponenten* (= Subgraph, in dem es von jedem Knoten einen Pfad zu jedem anderen Knoten gibt), um **EG** effizient zu berechnen.

Beispiel: Mikrowellen-Herd

Kommt nach dem Start stets Hitze ($f = \mathbf{AG}\,(Start \rightarrow \mathbf{AF}\,Heat)$)?

Beispiel: Mikrowellen-Herd

Kommt nach dem Start stets Hitze ($f = \mathbf{AG}\,(Start \rightarrow \mathbf{AF}\,Heat)$)?

Umformung nach

$$
\begin{aligned}
f &= \mathbf{AG}\,(\neg Start \vee \mathbf{AF}\,Heat) \\
&= \neg\mathbf{EF}\,(Start \wedge \neg\mathbf{AF}\,Heat) \\
&= \neg\mathbf{E}(\top\ \mathbf{U}\ (Start \wedge \neg\mathbf{AF}\,Heat)
\end{aligned}
$$

Beispiel: Mikrowellen-Herd

Kommt nach dem Start stets Hitze ($f = \mathbf{AG}\,(\mathit{Start} \rightarrow \mathbf{AF}\,\mathit{Heat})$)?

Umformung nach

$$\begin{aligned} f &= \mathbf{AG}\,(\neg \mathit{Start} \vee \mathbf{AF}\,\mathit{Heat}) \\ &= \neg \mathbf{EF}\,(\mathit{Start} \wedge \neg \mathbf{AF}\,\mathit{Heat}) \\ &= \neg \mathbf{E}(\top\ \mathbf{U}\ (\mathit{Start} \wedge \neg \mathbf{AF}\,\mathit{Heat}) \end{aligned}$$

Wir definieren:

$$f_1 = \mathbf{AF}\,\mathit{Heat}$$

Beispiel: Mikrowellen-Herd

Kommt nach dem Start stets Hitze ($f = \mathbf{AG}\,(Start \rightarrow \mathbf{AF}\,Heat)$)?

Umformung nach

$$\begin{aligned} f &= \mathbf{AG}\,(\neg Start \vee \mathbf{AF}\,Heat) \\ &= \neg\mathbf{EF}\,(Start \wedge \neg\mathbf{AF}\,Heat) \\ &= \neg\mathbf{E}(\top\ \mathbf{U}\ (Start \wedge \neg\mathbf{AF}\,Heat) \end{aligned}$$

Wir definieren:

$f_1 = \mathbf{AF}\,Heat$

$f_2 = \neg f_1$

Beispiel: Mikrowellen-Herd

Kommt nach dem Start stets Hitze ($f = \mathbf{AG}\,(Start \rightarrow \mathbf{AF}\,Heat)$)?

Umformung nach

$$\begin{aligned} f &= \mathbf{AG}\,(\neg Start \vee \mathbf{AF}\,Heat) \\ &= \neg \mathbf{EF}\,(Start \wedge \neg \mathbf{AF}\,Heat) \\ &= \neg \mathbf{E}(\top\ \mathbf{U}\ (Start \wedge \neg \mathbf{AF}\,Heat) \end{aligned}$$

Wir definieren:

$$f_1 = \mathbf{AF}\,Heat$$

$$f_2 = \neg f_1$$

$$f_3 = Start \wedge f_2$$

Beispiel: Mikrowellen-Herd

Kommt nach dem Start stets Hitze ($f = \textbf{AG}\ (Start \rightarrow \textbf{AF}\ Heat)$)?

Umformung nach

$$\begin{aligned} f &= \textbf{AG}\ (\neg Start \vee \textbf{AF}\ Heat) \\ &= \neg \textbf{EF}\ (Start \wedge \neg \textbf{AF}\ Heat) \\ &= \neg \textbf{E}(\top\ \textbf{U}\ (Start \wedge \neg \textbf{AF}\ Heat) \end{aligned}$$

Wir definieren:

$$\begin{aligned} f_1 &= \textbf{AF}\ Heat \\ f_2 &= \neg f_1 \\ f_3 &= Start \wedge f_2 \\ f_4 &= \textbf{E}(\top\ \textbf{U}\ f_3) \end{aligned}$$

Beispiel: Mikrowellen-Herd

Kommt nach dem Start stets Hitze ($f = \mathbf{AG}\,(\mathit{Start} \rightarrow \mathbf{AF}\,\mathit{Heat})$)?

Umformung nach

$$\begin{aligned} f &= \mathbf{AG}\,(\neg \mathit{Start} \vee \mathbf{AF}\,\mathit{Heat}) \\ &= \neg \mathbf{EF}\,(\mathit{Start} \wedge \neg \mathbf{AF}\,\mathit{Heat}) \\ &= \neg \mathbf{E}(\top\ \mathbf{U}\ (\mathit{Start} \wedge \neg \mathbf{AF}\,\mathit{Heat}) \end{aligned}$$

Wir definieren:

$$\begin{aligned} f_1 &= \mathbf{AF}\,\mathit{Heat} \\ f_2 &= \neg f_1 \\ f_3 &= \mathit{Start} \wedge f_2 \\ f_4 &= \mathbf{E}(\top\ \mathbf{U}\ f_3) \\ f &= \neg f_4 \end{aligned}$$

Beispiel: Mikrowellen-Herd

Kommt nach dem Start stets Hitze ($f = \mathbf{AG}\,(Start \rightarrow \mathbf{AF}\,Heat)$)?

Umformung nach

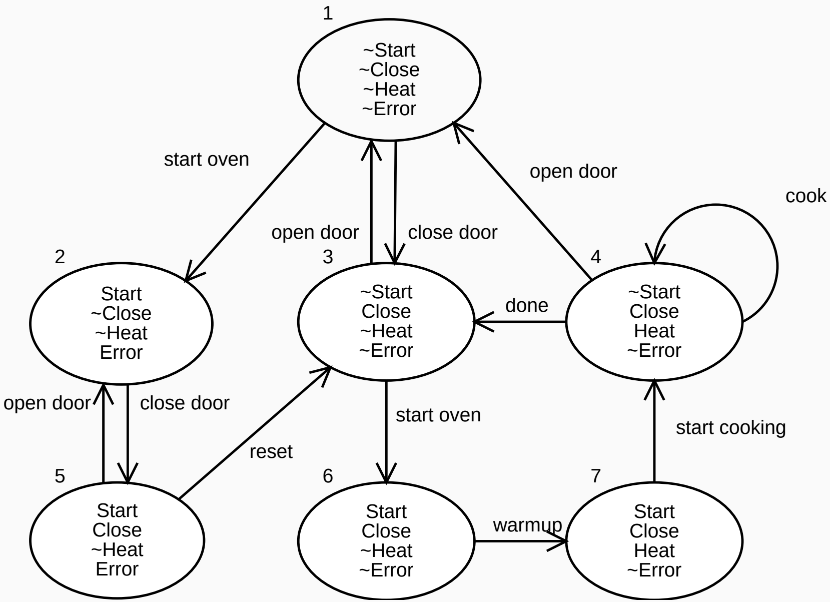
$$\begin{aligned} f &= \mathbf{AG}\,(\neg Start \vee \mathbf{AF}\,Heat) \\ &= \neg\mathbf{EF}\,(Start \wedge \neg\mathbf{AF}\,Heat) \\ &= \neg\mathbf{E}(\top\ \mathbf{U}\ (Start \wedge \neg\mathbf{AF}\,Heat) \end{aligned}$$

Wir definieren:

$$\begin{aligned} f_1 &= \mathbf{AF}\,Heat \\ f_2 &= \neg f_1 \\ f_3 &= Start \wedge f_2 \\ f_4 &= \mathbf{E}(\top\ \mathbf{U}\ f_3) \\ f &= \neg f_4 \end{aligned}$$

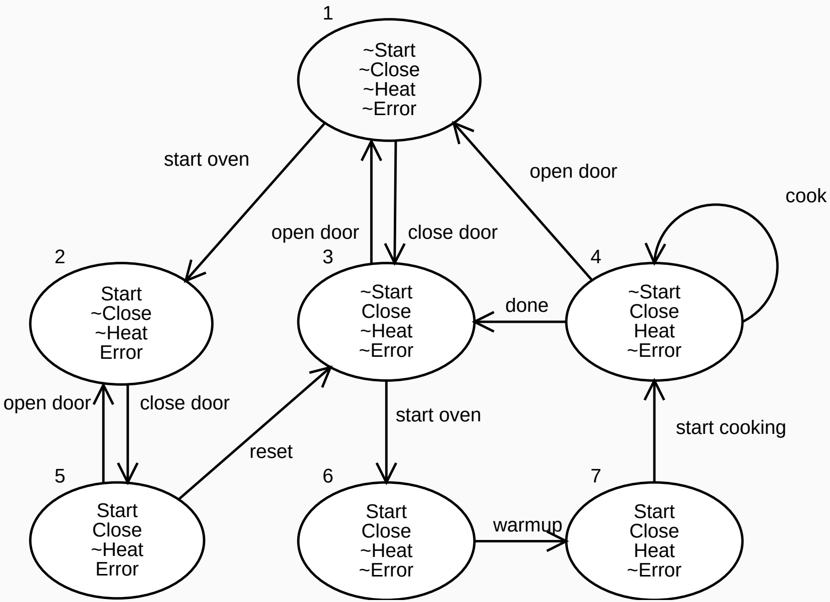
Annahme: SAT(f) berechnet Menge der Zustände, die mit f attribuiert sind (Übung): $M, 1 \models f \Leftrightarrow 1 \in \mathrm{SAT}(f)$

Beispiel: Mikrowellen-Herd (2)



$$f_1 = \mathbf{AF}\ \mathit{Heat}$$
$$f_2 = \neg f_1$$
$$f_3 = \mathit{Start} \wedge f_2$$
$$f_4 = \mathbf{E}(\top\ \mathbf{U}\ f_3)$$
$$f = \neg f_4$$
$$\mathrm{SAT}(f_1) =$$

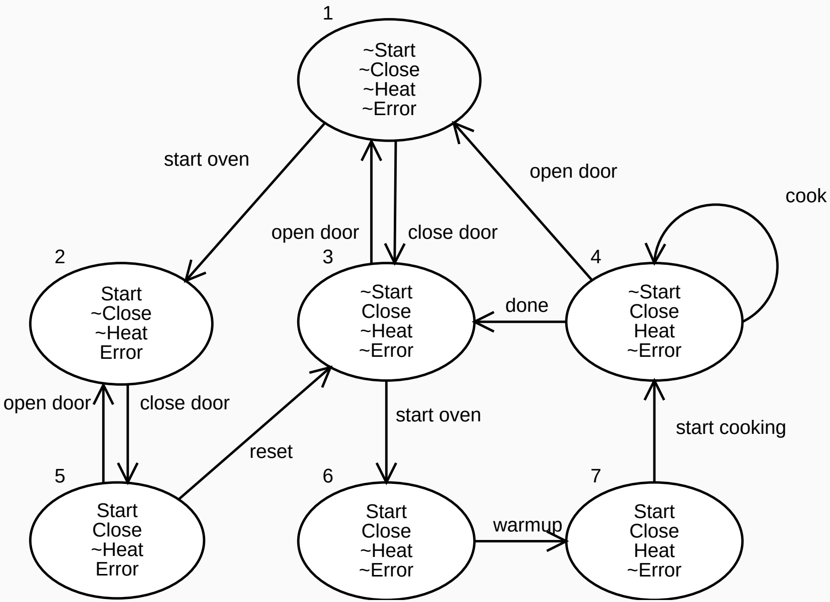
Beispiel: Mikrowellen-Herd (2)



$$f_1 = \textbf{AF}\ Heat$$
$$f_2 = \neg f_1$$
$$f_3 = Start \wedge f_2$$
$$f_4 = \textbf{E}(\top\ \textbf{U}\ f_3)$$
$$f = \neg f_4$$

$$\mathsf{SAT}(f_1) = \{4, 6, 7\}$$
$$\mathsf{SAT}(f_2) =$$

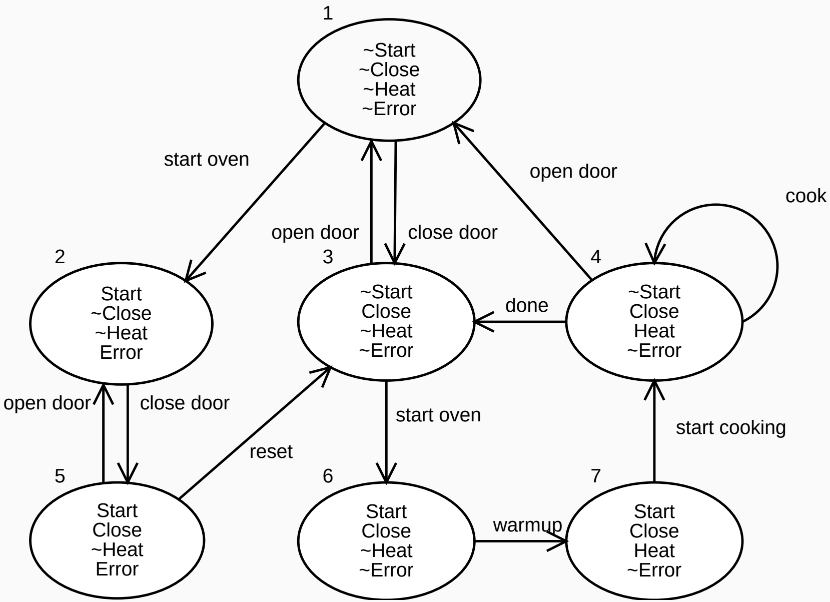
Beispiel: Mikrowellen-Herd (2)



$$f_1 = \mathbf{AF}\ \mathit{Heat}$$
$$f_2 = \neg f_1$$
$$f_3 = \mathit{Start} \wedge f_2$$
$$f_4 = \mathbf{E}(\top\ \mathbf{U}\ f_3)$$
$$f = \neg f_4$$

$$\mathrm{SAT}(f_1) = \{4, 6, 7\}$$
$$\mathrm{SAT}(f_2) = \{1, 2, 3, 5\}$$
$$\mathrm{SAT}(f_3) =$$

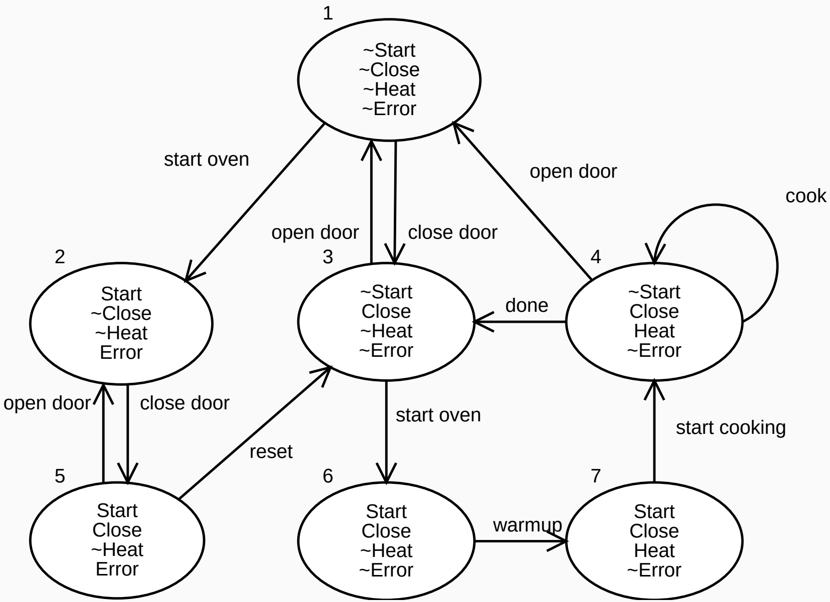
Beispiel: Mikrowellen-Herd (2)



$$f_1 = \mathbf{AF}\ \mathit{Heat}$$
$$f_2 = \neg f_1$$
$$f_3 = \mathit{Start} \wedge f_2$$
$$f_4 = \mathbf{E}(\top\ \mathbf{U}\ f_3)$$
$$f = \neg f_4$$

$$\mathrm{SAT}(f_1) = \{4, 6, 7\}$$
$$\mathrm{SAT}(f_2) = \{1, 2, 3, 5\}$$
$$\mathrm{SAT}(f_3) = \{2, 5\}$$
$$\mathrm{SAT}(f_4) =$$

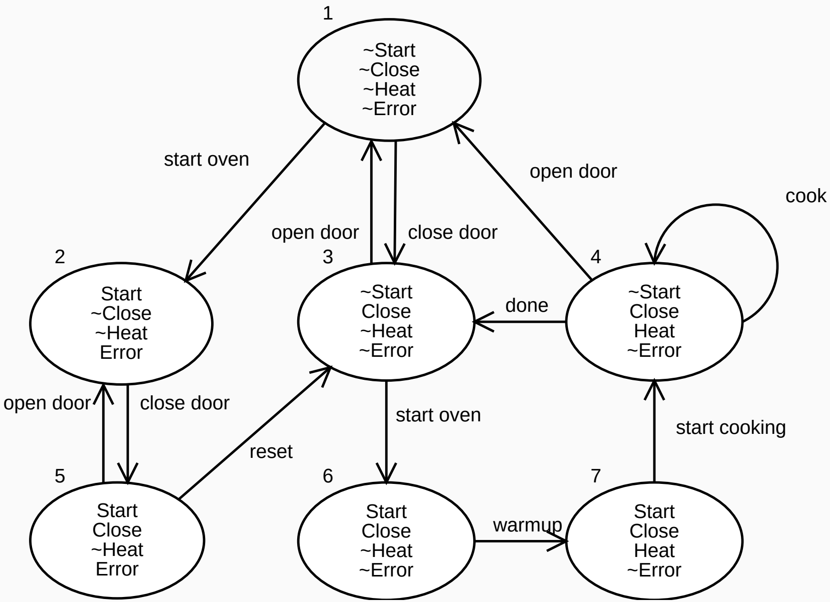
Beispiel: Mikrowellen-Herd (2)



$$f_1 = \mathbf{AF}\ \mathit{Heat}$$
$$f_2 = \neg f_1$$
$$f_3 = \mathit{Start} \wedge f_2$$
$$f_4 = \mathbf{E}(\top\ \mathbf{U}\ f_3)$$
$$f = \neg f_4$$

$$\mathrm{SAT}(f_1) = \{4, 6, 7\}$$
$$\mathrm{SAT}(f_2) = \{1, 2, 3, 5\}$$
$$\mathrm{SAT}(f_3) = \{2, 5\}$$
$$\mathrm{SAT}(f_4) = \{1, 2, 5\}$$
$$\mathrm{SAT}(f) =$$

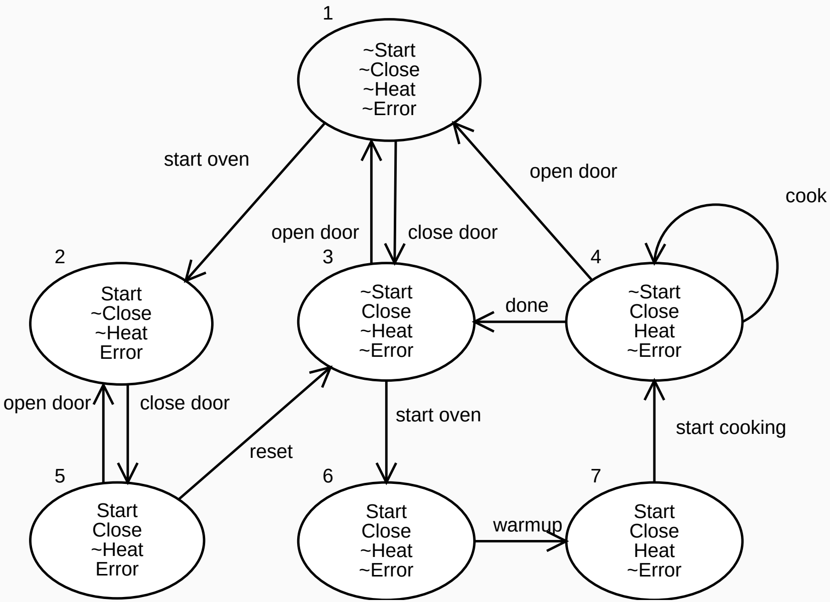
Beispiel: Mikrowellen-Herd (2)



$$f_1 = \textbf{AF}\ Heat$$
$$f_2 = \neg f_1$$
$$f_3 = Start \wedge f_2$$
$$f_4 = \textbf{E}(\top\ \textbf{U}\ f_3)$$
$$f = \neg f_4$$

$$\mathrm{SAT}(f_1) = \{4, 6, 7\}$$
$$\mathrm{SAT}(f_2) = \{1, 2, 3, 5\}$$
$$\mathrm{SAT}(f_3) = \{2, 5\}$$
$$\mathrm{SAT}(f_4) = \{1, 2, 5\}$$
$$\mathrm{SAT}(f) = \{3, 4, 6, 7\}$$

Beispiel: Mikrowellen-Herd (2)



$$f_1 = \mathbf{AF}\ \mathit{Heat}$$
$$f_2 = \neg f_1$$
$$f_3 = \mathit{Start} \wedge f_2$$
$$f_4 = \mathbf{E}(\top\ \mathbf{U}\ f_3)$$
$$f = \neg f_4$$

$$\mathrm{SAT}(f_1) = \{4, 6, 7\}$$
$$\mathrm{SAT}(f_2) = \{1, 2, 3, 5\}$$
$$\mathrm{SAT}(f_3) = \{2, 5\}$$
$$\mathrm{SAT}(f_4) = \{1, 2, 5\}$$
$$\mathrm{SAT}(f) = \{3, 4, 6, 7\}$$

$$1 \notin \mathrm{SAT}(f)\ !$$

Beispiel: Mikrowellen-Herd (3)

Kommt die Hitze stets, nachdem die Tür geschlossen ist ($f = \mathbf{A}(\neg \mathit{Heat}\ \mathbf{U}\ \mathit{Close})$)?

Beispiel: Mikrowellen-Herd (3)

Kommt die Hitze stets, nachdem die Tür geschlossen ist
($f = \mathbf{A}(\neg \mathit{Heat}\ \mathbf{U}\ \mathit{Close})$)?

Umformung nach

$$\begin{aligned} f &= \neg\big(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \mathbf{EG}\ \neg \mathit{Close}\big) \\ &= \neg\big(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \neg \mathbf{AF}\ \mathit{Close}\big) \end{aligned}$$

Beispiel: Mikrowellen-Herd (3)

Kommt die Hitze stets, nachdem die Tür geschlossen ist
($f = \mathbf{A}(\neg \mathit{Heat}\ \mathbf{U}\ \mathit{Close})$)?

Umformung nach

$$\begin{aligned} f &= \neg\big(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \mathbf{EG}\ \neg \mathit{Close}\big) \\ &= \neg\big(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \neg \mathbf{AF}\ \mathit{Close}\big) \end{aligned}$$

Wir definieren:

$f_1 = \mathbf{AF}\ \mathit{Close}$

Beispiel: Mikrowellen-Herd (3)

Kommt die Hitze stets, nachdem die Tür geschlossen ist
($f = \mathbf{A}(\neg \mathit{Heat}\ \mathbf{U}\ \mathit{Close})$)?

Umformung nach

$$\begin{aligned} f &= \neg\big(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \mathbf{EG}\ \neg \mathit{Close}\big) \\ &= \neg\big(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \neg \mathbf{AF}\ \mathit{Close}\big) \end{aligned}$$

Wir definieren:

$f_1 = \mathbf{AF}\ \mathit{Close}$

$f_2 = \neg f_1$

Beispiel: Mikrowellen-Herd (3)

Kommt die Hitze stets, nachdem die Tür geschlossen ist ($f = \mathbf{A}(\neg \mathit{Heat}\ \mathbf{U}\ \mathit{Close})$)?

Umformung nach

$$
\begin{aligned}
f &= \neg\big(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \mathbf{EG}\ \neg \mathit{Close}\big)\\
&= \neg\big(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \neg \mathbf{AF}\ \mathit{Close}\big)
\end{aligned}
$$

Wir definieren:

$f_1 = \mathbf{AF}\ \mathit{Close}$

$f_2 = \neg f_1$

$f_3 = \mathit{Heat} \wedge \neg \mathit{Close}$

Beispiel: Mikrowellen-Herd (3)

Kommt die Hitze stets, nachdem die Tür geschlossen ist
($f = \mathbf{A}(\neg \mathit{Heat}\ \mathbf{U}\ \mathit{Close})$)?

Umformung nach

$$
\begin{aligned}
f &= \neg(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \mathbf{EG}\ \neg \mathit{Close}) \\
&= \neg(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \neg \mathbf{AF}\ \mathit{Close})
\end{aligned}
$$

Wir definieren:

$$
\begin{aligned}
f_1 &= \mathbf{AF}\ \mathit{Close} \\
f_2 &= \neg f_1 \\
f_3 &= \mathit{Heat} \wedge \neg \mathit{Close} \\
f_4 &= \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3))
\end{aligned}
$$

Beispiel: Mikrowellen-Herd (3)

Kommt die Hitze stets, nachdem die Tür geschlossen ist
($f = \mathbf{A}(\neg \mathit{Heat}\ \mathbf{U}\ \mathit{Close})$)?

Umformung nach

$$\begin{aligned} f &= \neg(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \mathbf{EG}\ \neg \mathit{Close}) \\ &= \neg(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \neg \mathbf{AF}\ \mathit{Close}) \end{aligned}$$

Wir definieren:

$$\begin{aligned} f_1 &= \mathbf{AF}\ \mathit{Close} \\ f_2 &= \neg f_1 \\ f_3 &= \mathit{Heat} \wedge \neg \mathit{Close} \\ f_4 &= \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3)) \\ f_5 &= \neg f_4 \end{aligned}$$

Beispiel: Mikrowellen-Herd (3)

Kommt die Hitze stets, nachdem die Tür geschlossen ist
($f = \mathbf{A}(\neg \mathit{Heat}\ \mathbf{U}\ \mathit{Close})$)?

Umformung nach

$$\begin{aligned} f &= \neg(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \mathbf{EG}\ \neg \mathit{Close}) \\ &= \neg(\mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ (\mathit{Heat} \wedge \neg \mathit{Close})) \vee \neg \mathbf{AF}\ \mathit{Close}) \end{aligned}$$

Wir definieren:

$f_1 = \mathbf{AF}\ \mathit{Close}$

$f_2 = \neg f_1$

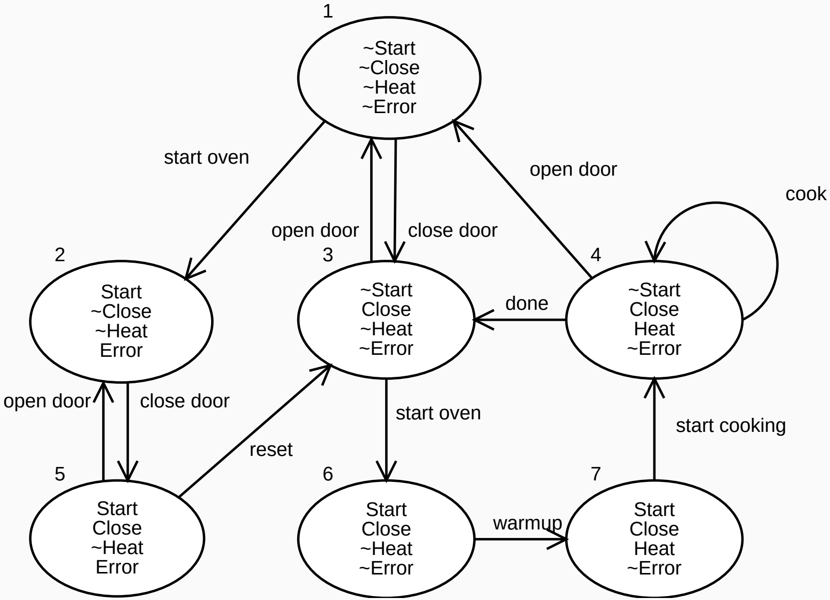
$f_3 = \mathit{Heat} \wedge \neg \mathit{Close}$

$f_4 = \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3))$

$f_5 = \neg f_4$

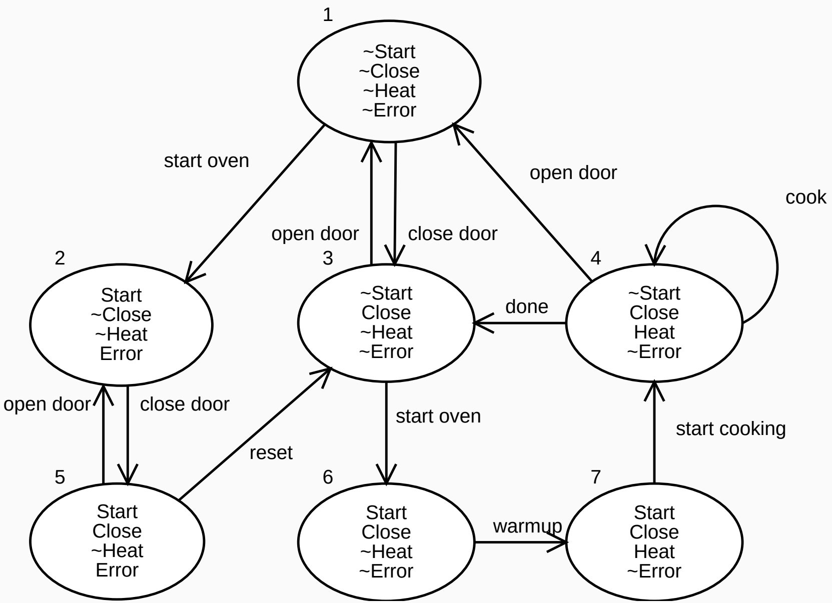
$f = f_5 \vee f_2$

Beispiel: Mikrowellen-Herd (4)



$$
\begin{aligned}
f_1 &= \mathbf{AF}\, \mathit{Close} \\
f_2 &= \neg f_1 \\
f_3 &= \mathit{Heat} \wedge \neg \mathit{Close} \\
f_4 &= \mathbf{E}(\neg \mathit{Close}\, \mathbf{U}\, f_3)) \\
f_5 &= \neg f_4 \\
f &= f_5 \vee f_2
\end{aligned}
$$

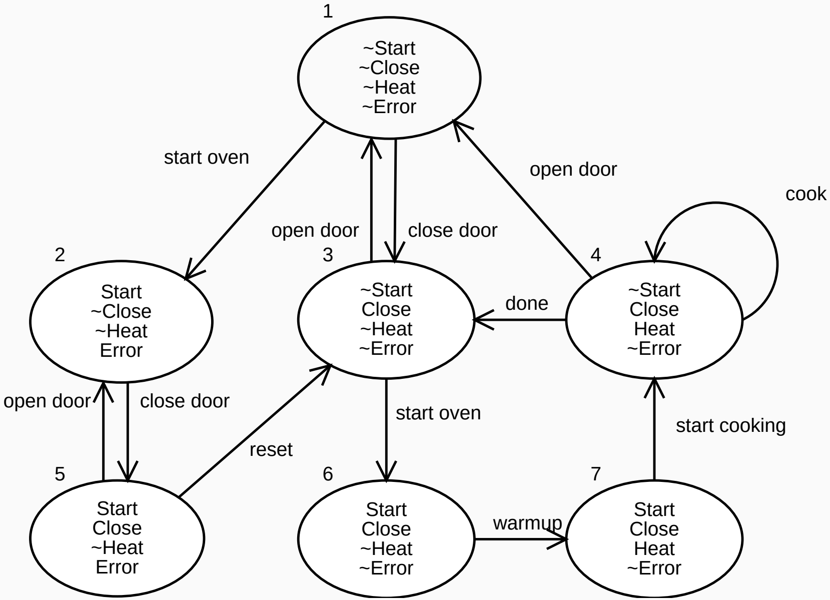
Beispiel: Mikrowellen-Herd (4)



$$
\begin{aligned}
f_1 &= \mathbf{AF}\ Close \\
f_2 &= \neg f_1 \\
f_3 &= Heat \land \neg Close \\
f_4 &= \mathbf{E}(\neg Close\ \mathbf{U}\ f_3)) \\
f_5 &= \neg f_4 \\
f &= f_5 \lor f_2
\end{aligned}
$$

$$\mathrm{SAT}(f_1) =$$

Beispiel: Mikrowellen-Herd (4)

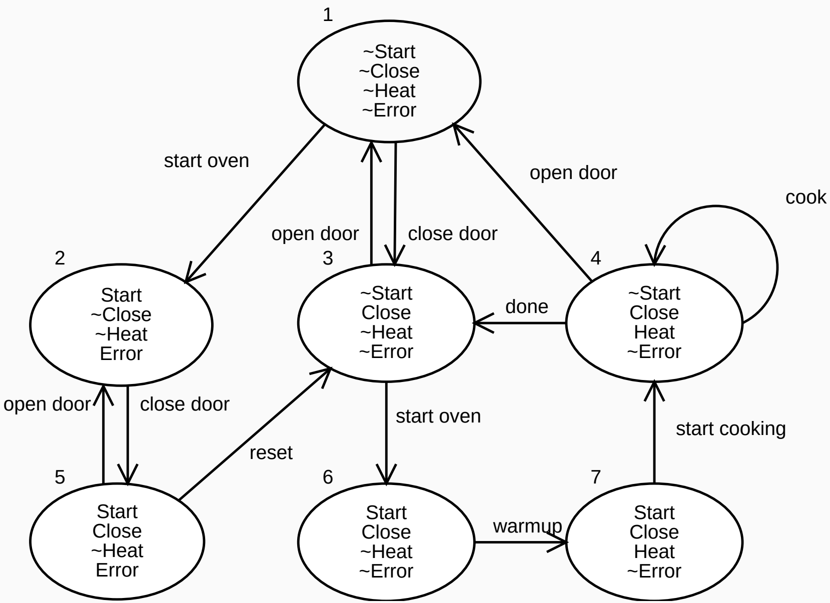


$$
\begin{aligned}
f_1 &= \mathbf{AF}\ \mathit{Close} \\
f_2 &= \neg f_1 \\
f_3 &= \mathit{Heat} \wedge \neg \mathit{Close} \\
f_4 &= \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3)) \\
f_5 &= \neg f_4 \\
f &= f_5 \vee f_2
\end{aligned}
$$

$\mathsf{SAT}(f_1) = \{1, 2, \ldots, 7\}$

$\mathsf{SAT}(f_2) =$

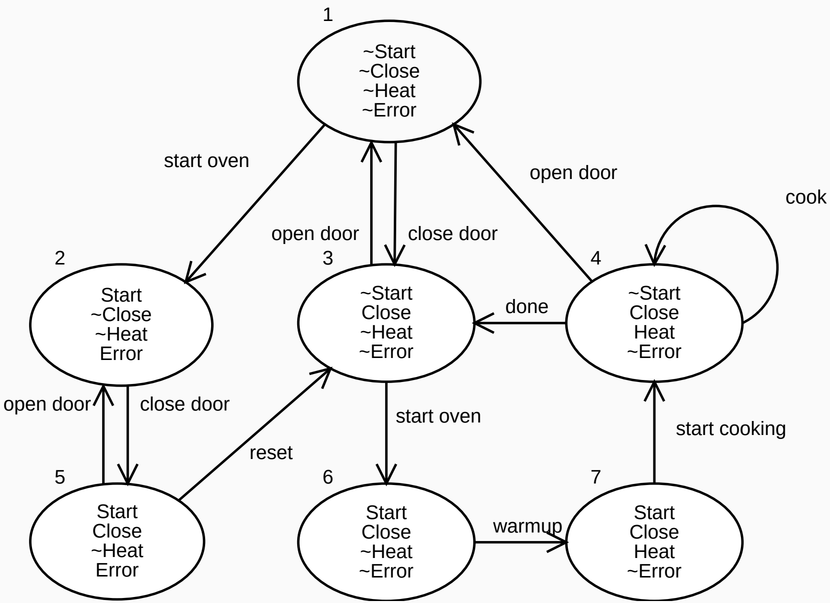
Beispiel: Mikrowellen-Herd (4)



$$f_1 = \mathbf{AF}\ \mathit{Close}$$
$$f_2 = \neg f_1$$
$$f_3 = \mathit{Heat} \wedge \neg \mathit{Close}$$
$$f_4 = \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3))$$
$$f_5 = \neg f_4$$
$$f = f_5 \vee f_2$$

$$\mathsf{SAT}(f_1) = \{1, 2, \ldots, 7\}$$
$$\mathsf{SAT}(f_2) = \emptyset$$
$$\mathsf{SAT}(f_3) =$$

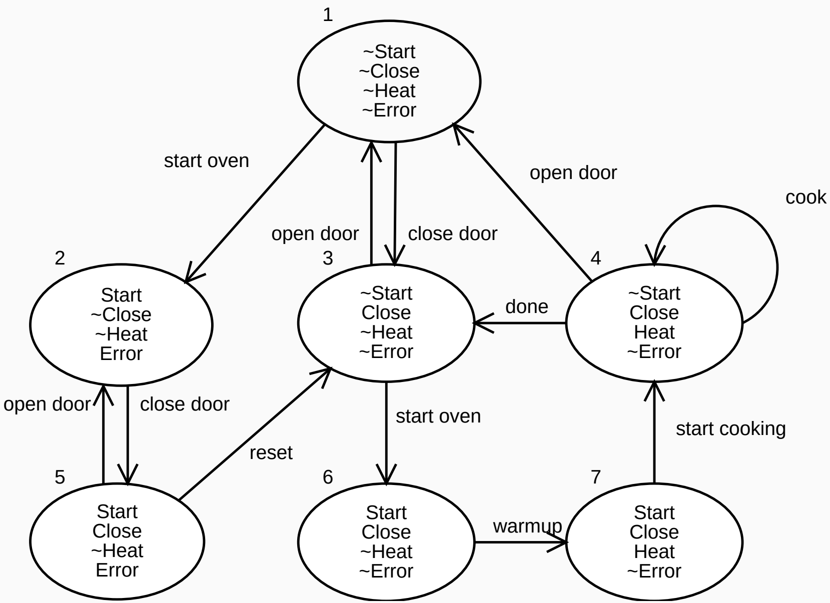
Beispiel: Mikrowellen-Herd (4)



$$f_1 = \mathbf{AF}\ \mathit{Close}$$
$$f_2 = \neg f_1$$
$$f_3 = \mathit{Heat} \wedge \neg \mathit{Close}$$
$$f_4 = \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3))$$
$$f_5 = \neg f_4$$
$$f = f_5 \vee f_2$$

$$\mathsf{SAT}(f_1) = \{1, 2, \ldots, 7\}$$
$$\mathsf{SAT}(f_2) = \emptyset$$
$$\mathsf{SAT}(f_3) = \emptyset$$
$$\mathsf{SAT}(f_4) =$$

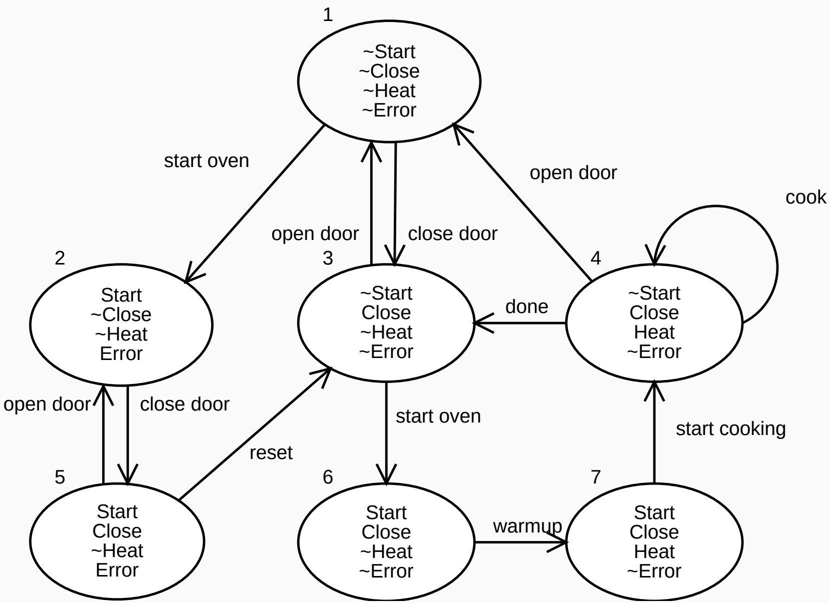
Beispiel: Mikrowellen-Herd (4)



$$
\begin{aligned}
f_1 &= \mathbf{AF}\ \mathit{Close} \\
f_2 &= \neg f_1 \\
f_3 &= \mathit{Heat} \land \neg \mathit{Close} \\
f_4 &= \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3)) \\
f_5 &= \neg f_4 \\
f &= f_5 \lor f_2
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{SAT}(f_1) &= \{1, 2, \ldots, 7\} \\
\mathsf{SAT}(f_2) &= \emptyset \\
\mathsf{SAT}(f_3) &= \emptyset \\
\mathsf{SAT}(f_4) &= \emptyset \\
\mathsf{SAT}(f_5) &=
\end{aligned}
$$

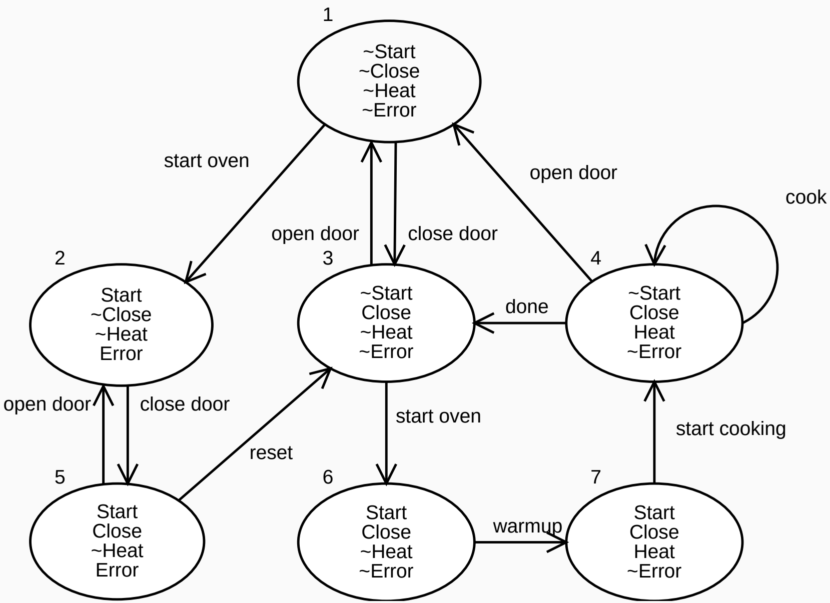
Beispiel: Mikrowellen-Herd (4)



$$f_1 = \mathbf{AF}\ \mathit{Close}$$
$$f_2 = \neg f_1$$
$$f_3 = \mathit{Heat} \wedge \neg \mathit{Close}$$
$$f_4 = \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3))$$
$$f_5 = \neg f_4$$
$$f = f_5 \vee f_2$$

$$\mathrm{SAT}(f_1) = \{1, 2, \ldots, 7\}$$
$$\mathrm{SAT}(f_2) = \emptyset$$
$$\mathrm{SAT}(f_3) = \emptyset$$
$$\mathrm{SAT}(f_4) = \emptyset$$
$$\mathrm{SAT}(f_5) = \{1, 2, \ldots, 7\}$$
$$\mathrm{SAT}(f) =$$

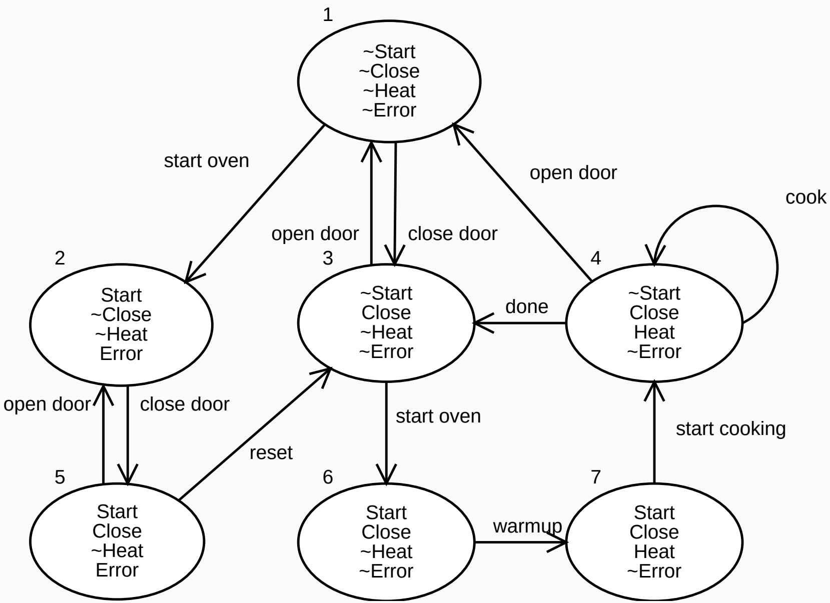
Beispiel: Mikrowellen-Herd (4)



$$f_1 = \mathbf{AF}\ \mathit{Close}$$
$$f_2 = \neg f_1$$
$$f_3 = \mathit{Heat} \wedge \neg \mathit{Close}$$
$$f_4 = \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3))$$
$$f_5 = \neg f_4$$
$$f = f_5 \vee f_2$$

$$\mathrm{SAT}(f_1) = \{1, 2, \ldots, 7\}$$
$$\mathrm{SAT}(f_2) = \emptyset$$
$$\mathrm{SAT}(f_3) = \emptyset$$
$$\mathrm{SAT}(f_4) = \emptyset$$
$$\mathrm{SAT}(f_5) = \{1, 2, \ldots, 7\}$$
$$\mathrm{SAT}(f) = \{1, 2, \ldots, 7\}$$

Beispiel: Mikrowellen-Herd (4)



$$
\begin{aligned}
f_1 &= \mathbf{AF}\ \mathit{Close} \\
f_2 &= \neg f_1 \\
f_3 &= \mathit{Heat} \wedge \neg \mathit{Close} \\
f_4 &= \mathbf{E}(\neg \mathit{Close}\ \mathbf{U}\ f_3)) \\
f_5 &= \neg f_4 \\
f &= f_5 \vee f_2
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{SAT}(f_1) &= \{1,2,\ldots,7\} \\
\mathsf{SAT}(f_2) &= \emptyset \\
\mathsf{SAT}(f_3) &= \emptyset \\
\mathsf{SAT}(f_4) &= \emptyset \\
\mathsf{SAT}(f_5) &= \{1,2,\ldots,7\} \\
\mathsf{SAT}(f) &= \{1,2,\ldots,7\}
\end{aligned}
$$

$1 \in \mathsf{SAT}(f)$!

Alternative Logiken

LTL (Linear Time Logic) Formeln beziehen sich nicht auf Zustandsbaum, sondern auf (alle) *Pfade* des Systems.

Pfadquantoren **E**, **A** aus CTL entfallen; dafür können Boolesche Verknüpfungen und Temporaloperatoren beliebig verschachtelt werden:

$\mathbf{G}\,\mathbf{F}\,p \rightarrow \mathbf{F}\,q$ „gilt p immer, dann auch irgendwann q“.

Alternative Logiken

LTL (Linear Time Logic) Formeln beziehen sich nicht auf Zustandsbaum, sondern auf (alle) *Pfade* des Systems.

Pfadquantoren **E**, **A** aus CTL entfallen; dafür können Boolesche Verknüpfungen und Temporaloperatoren beliebig verschachtelt werden:

$\mathbf{G}\,\mathbf{F}\,p \rightarrow \mathbf{F}\,q$ „gilt p immer, dann auch irgendwann q“.

CTL* Kombination aus LTL und CTL; CTL$^* \supseteq$ LTL $\cup$ CTL

Pfadquantoren und Temporaloperatoren können wie in LTL beliebig verschachtelt werden:

$\mathbf{E}(\mathbf{G}\,\mathbf{F}\,p)$ „Es gibt einen Pfad, in dem p unendlich oft wahr wird“

Erstaunlicherweise keine erhöhte Komplexität!

Problem: Zustandsexplosion

Auch wenn wir ein Modell effizient prüfen können, kann das Modell selbst immer noch *sehr groß* sein.

Beispiel: 100 (boolesche) Variablen $\Rightarrow 2^{100}$ Zustände

Lösung: *Ordered Binary Decision Diagram* (OBDD) – ermöglicht *geteilte Darstellung* von Formeln

Zwei-Bit-Vergleicher

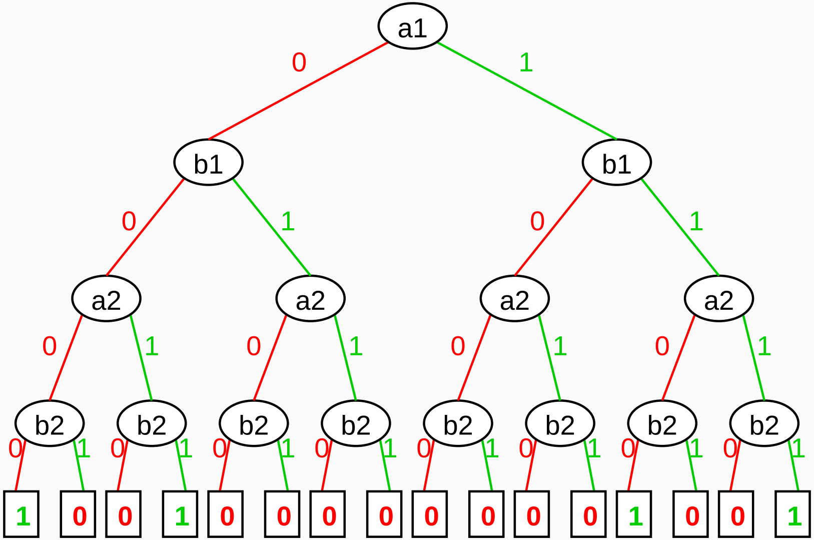
Problem: Zwei Bitfolgen a_1a_2 und b_1b_2 vergleichen

$$f(a_1, b_1, a_2, b_2) = (a_1 \leftrightarrow b_1) \wedge (a_2 \leftrightarrow b_2)$$

Zwei-Bit-Vergleicher

Problem: Zwei Bitfolgen a_1a_2 und b_1b_2 vergleichen

$$f(a_1, b_1, a_2, b_2) = (a_1 \leftrightarrow b_1) \wedge (a_2 \leftrightarrow b_2)$$

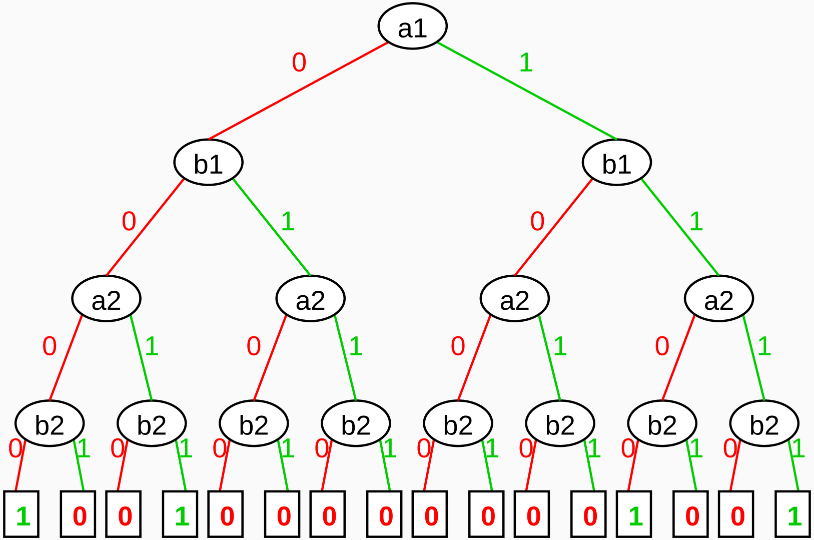


Als Entscheidungsbaum

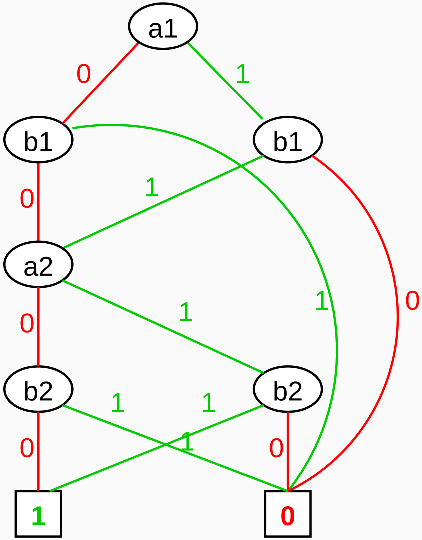
Zwei-Bit-Vergleicher

Problem: Zwei Bitfolgen $a_1 a_2$ und $b_1 b_2$ vergleichen

$$f(a_1, b_1, a_2, b_2) = (a_1 \leftrightarrow b_1) \wedge (a_2 \leftrightarrow b_2)$$



Als Entscheidungsbaum



Als OBDD

OBDDs und Model Checking

Mit OBDDs können auch große Zustandsräume effizient abgelegt werden; gleichzeitig gibt es effiziente Operationen auf OBDDs

Symbolisches Model Checking stellt Modell (= Entscheidungsbaum) und Spezifikation (= logische Formel) als OBDDs dar und prüft, ob Formel im Modell enthalten ist.

⇒ Auch große (Software-)Systeme werden so modellierbar

Anwendungen von Model Checking

- Hardware (routinemäßig)
- Protokollverifikation (AT&T, CORBA)
- Software (wenn Abstraktion zu endl. Automat existiert)
- Raumsonden (NASA)
- Testdatengenerierung (via Gegenbeispiel)
- Space Shuttle Resttungsprogramm (NASA)

Boolesche Programme

Läßt sich Model Checking auch auf komplexe Programme anwenden?

Grundidee: Abbildung des Programms in *Boolesches Programm*:

- Alle Datentypen werden zu Booleans
- Alle Bedingungen werden zu Booleschen Prädikaten

Auf der so gewonnenen *Abstraktion* kann dann Model Checking ausgeführt werden

Boolesche Programme

Läßt sich Model Checking auch auf komplexe Programme anwenden?

Grundidee: Abbildung des Programms in *Boolesches Programm*:

- Alle Datentypen werden zu Booleans
- Alle Bedingungen werden zu Booleschen Prädikaten

Auf der so gewonnenen *Abstraktion* kann dann Model Checking ausgeführt werden

Ansatz von Microsoft Research zum Verifizieren von Gerätetreibern (mit Andreas Podelski vom MPI)

Boolesche Programme

Läßt sich Model Checking auch auf komplexe Programme anwenden?

Grundidee: Abbildung des Programms in *Boolesches Programm*:

- Alle Datentypen werden zu Booleans
- Alle Bedingungen werden zu Booleschen Prädikaten

Auf der so gewonnenen *Abstraktion* kann dann Model Checking ausgeführt werden

Ansatz von Microsoft Research zum Verifizieren von Gerätetreibern (mit Andreas Podelski vom MPI)

Mehr in dessen Vorlesungen!

Zusammenfassung

Model Checking

- ist ein effizientes Verfahren zur Verifikation endlicher Modelle
- basiert auf spezieller Logik (CTL, LTL, CTL*), in der Anfragen gestellt werden
- prüft Anfragen durch Attribuierung des Modells
- benutzt OBDDs, um Zustandsexplosion zu vermeiden
- findet breite industrielle Anwendung

Literatur

The SPIN Model Checker http://netlib.bell-labs.com/netlib/spin/whatispin.html

The SLAM Project http://research.microsoft.com/projects/slam/

Model Checking (Clarke, Grumberg, Peled)