MODEL CHECKING
3 – TEMPORALE LOGIKEN

Sommersemester 2009

Dr. Carsten Sinz, Universität Karlsruhe

Kripke-Struktur

- Definition: Sei A eine Menge von Aussagevariablen. Eine Kripke-Struktur M über A ist ein Tupel

$$M = (S, I, T, L)$$

- S: Zustandsmenge (endlich)
- I⊆S: Menge der Initialzustände
- T: S×S: *totale* Übergangsrelation (*total*: für jeden Zustand s gibt es ein s' mit T(s,s'))
- $L : S \to \mathbb{P}(A)$ Label-Funktion, die anzeigt, welche Variablen in einem Zustand wahr sind

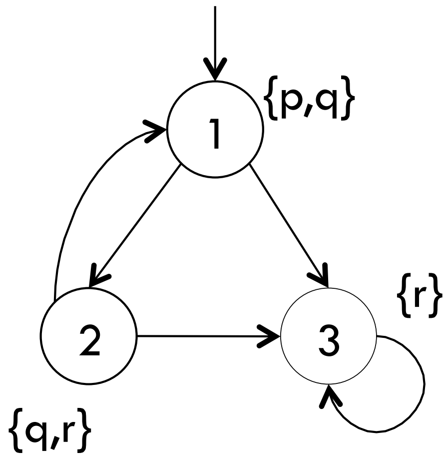
Kripke-Strukturen: Pfade

- **Definition:** Ein Pfad π in einer Kripke-Struktur $M = (S, I, T, L)$ ist eine unendliche Folge von Zuständen $\pi = s_0 s_1 s_2 \ldots$ mit $s_0 \in I$ und $s_i \to s_{i+1}$ für alle i≥0.
- **Anmerkung:** Häufig keine Startzustände explizit benannt, dann soll I=S gelten
- Für jeden Zustand s∈S kann die Menge der Pfade, die in diesem Zustand beginnen, als unendlicher Baum dargestellt werden.

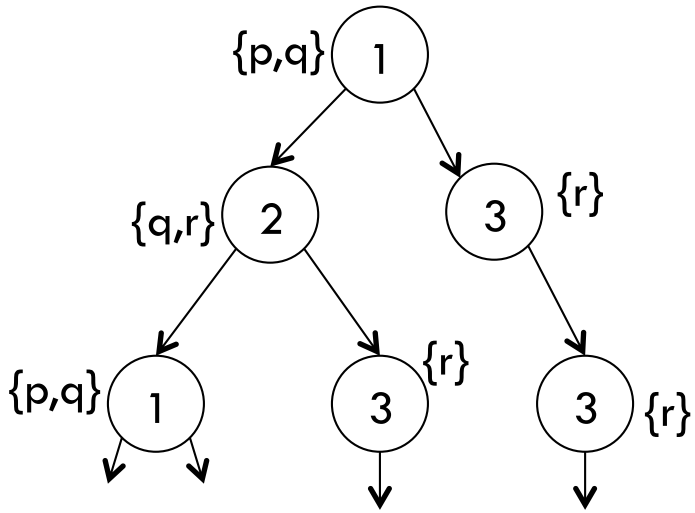
Beispiel Kripke-Struktur

S = { 1, 2, 3 }
A = { p, q, r }

M:

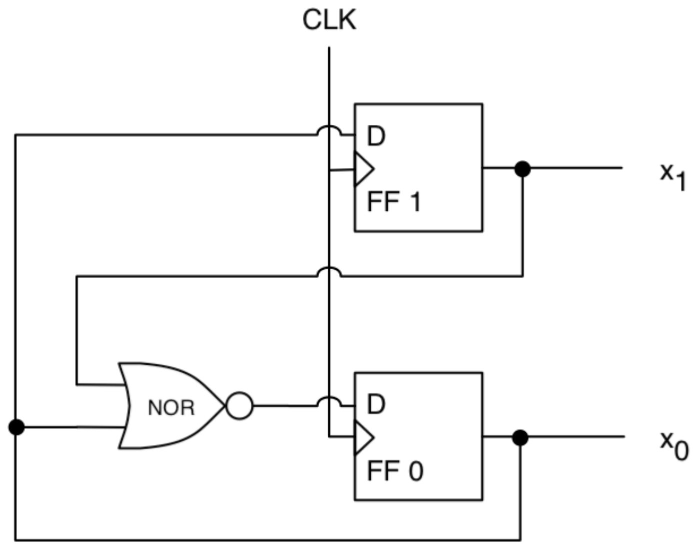


Pfade als unendlicher Baum
(ausgehend von Zustand 1)



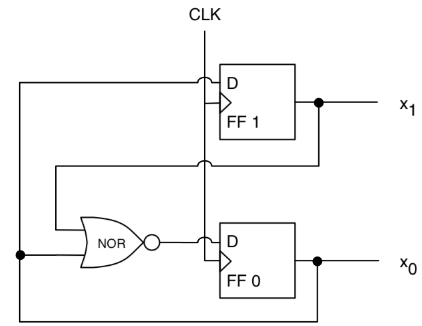
Hardware-Schaltung als Kripke-Struktur

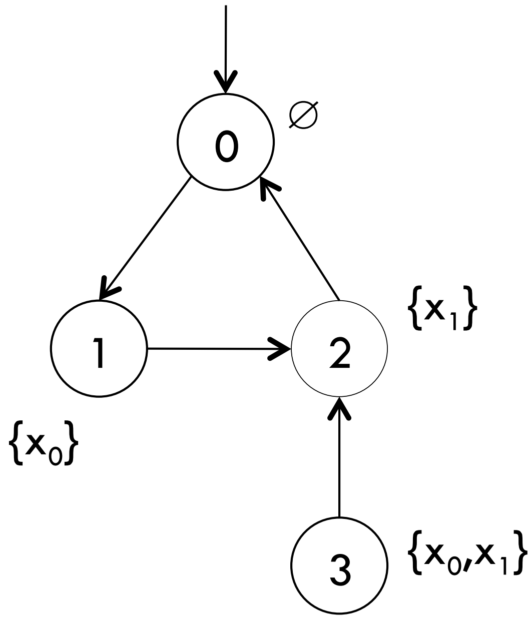
- Kripke-Strukturen dienen zur Modellierung z.B. von Hardware-Schaltungen
- **Beispiel:** 2-Bit-Zähler, der kontinuierlich von 0 bis 2 zählt



Beispiel Kripke-Struktur

- Kripke-Struktur M über $A=\{x_1,x_2\}$:





Pfad(e) ausgehend von Zustand 0:
$\pi_0 = 0,1,2,0,1,2,0,1,2,\ldots$

Eigenschaften von Kripke-Strukturen

- Wie können Eigenschaften von Kripke-Strukturen dargestellt werden?
- Zum Beispiel:
  - Wenn der Zähler nicht in Zustand 3 startet, so wird dieser Zustand nie erreicht.
  - Jeder der Zustände 0, 1, 2 wird unendlich oft erreicht.
  - Auf den Zustand 2 folgt immer der Zustand 0.

Temporallogiken

- Beschreiben Eigenschaften von möglichen Pfaden in einem System (bzw. einer Kripke-Struktur)
- Zeitliches Verhalten im Fokus:
  - Eine Eigenschaft soll *irgendwann einmal* gelten.
  - Eine Eigenschaft (z.B. Fehler) soll *nie* eintreten.
  - Eine Eigenschaft soll *spätestens nach x Schritten* eintreten.

Die Temporallogik CTL*

- CTL*: *computation tree logic*
- CTL* beschreibt mögliche Läufe einer nicht-deterministischen Schaltung (*computation trees*) bzw. Pfade in einer Kripke-Struktur
- Aussagenlogik erweitert um Pfad-Quantoren und Temporal-Operatoren

CTL*: Syntax (I)

- Pfad-Quantoren:
  - **A** (*for all*): für alle Berechnungspfade gilt…
  - **E** (*exists*) : es gibt einen Pfad, auf dem gilt...
- Temporal-Operatoren:
  - **X** (*next*): im nächsten Zustand des Pfades gilt...
  - **F** (eventually / in the future): in irgendeinem zukünftigen Zustand des Pfades gilt...
  - **G** (*always* / *globally*): auf allen Zuständen des Pfades gilt...
  - **U** (*until*): eine Eigenschaft gilt solange, bis eine andere Eigenschaft eintritt
  - **R** (*release*): eine Eigenschaft gilt, solange eine andere Eigenschaft auch gilt

CTL*: Syntax (II)

- A: Menge der aussagenlogischen Variablen
- Zustandsformeln (gelten für *Zustände*):
  - Jedes $a \in A$ ist eine Zustandsformel.
  - Wenn f und g Zustandsformeln, so auch $\neg f$, $f \wedge g$ und $f \vee g$.
  - Ist f eine Pfadformel, so sind **E**f und **A**f Zustandsformeln
- Pfadformeln (gelten für *Pfade*):
  - Jede Zustandsformel ist auch eine Pfadformel (dies ist so zu verstehen, dass sie für den *ersten* Zustand des Pfades gilt)
  - Sind f und g Pfadformeln, so auch $\neg f$, $f \wedge g$, $f \vee g$, **X**f, **F**f, **G**f, f**U**g, f**R**g.

CTL*: Beispiele

- **AF**x: auf allen Pfaden gilt irgendwanneinmal x
- **AG** (Req → **AF** Ack)
- **AG**(**AF** DeviceEnabled)
- **AG**($\neg(x_0 \wedge x_1)$)

Was ist mit:

- **AE**p, **AX**q, **AA**p, **A**p∧**E**q ?
- **XX**p, **FXE**p, **EFX**p ?

Semantik von CTL*-Formeln

- Gegeben
  - eine CTL*-Formel f,
  - eine Kripke-Struktur M und
  - ein Zustand s (oder ein Pfad π) in M
- **Frage:** Gilt f in der Kripke-Struktur M in Zustand s (bzw. auf Pfad π)?
- **Notation:** $M, s \models f \quad \text{bzw.} \quad M, \pi \models f$
- **Definition:** Für einen Pfad $\pi = s_0 s_1 s_2 \ldots$ bezeichne π^k den Suffix von π, der in s_k beginnt, d.h. $\pi^k = s_k s_{k+1} \ldots$

CTL*: Semantik

Zustandsformeln:

$$
\begin{array}{lcl}
M, s \models p & \Leftrightarrow & p \in L(s) \\
M, s \models \neg f_1 & \Leftrightarrow & M, s \not\models f_1 \\
M, s \models f_1 \vee f_2 & \Leftrightarrow & M, s \models f_1 \ \text{ oder } \ M, s \models f_2 \\
M, s \models f_1 \wedge f_2 & \Leftrightarrow & M, s \models f_1 \ \text{ und } \ M, s \models f_2 \\
M, s \models \mathbf{E} g_1 & \Leftrightarrow & \text{es gibt einen Pfad } \pi = s \ldots, \text{ so dass } M, \pi \models g_1 \\
M, s \models \mathbf{A} g_1 & \Leftrightarrow & \text{für alle Pfade } \pi = s \ldots \ \text{ gilt } M, \pi \models g_1
\end{array}
$$

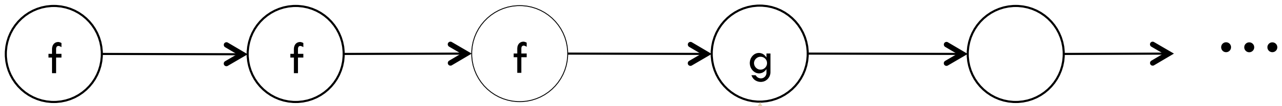
$[f_1, f_2 : \ \text{Zustandsformeln}; \quad g_1, g_2 : \ \text{Pfadformeln}]$

CTL*: Semantik

$$
\begin{array}{lcl}
M,\pi \models f_1 & \Leftrightarrow & \pi = s\ldots \text{ und } M,s \models f_1 \\
M,\pi \models \neg g_1 & \Leftrightarrow & M,\pi \not\models g_1 \\
M,\pi \models g_1 \vee g_2 & \Leftrightarrow & M,\pi \models g_1 \;\text{ oder }\; M,\pi \models g_2 \\
M,\pi \models g_1 \wedge g_2 & \Leftrightarrow & M,\pi \models g_1 \;\text{ und }\; M,\pi \models g_2 \\
M,\pi \models \mathbf{X} g_1 & \Leftrightarrow & M,\pi^1 \models g_1 \\
M,\pi \models \mathbf{F} g_1 & \Leftrightarrow & \text{es gibt ein } k \geq 0 \text{, so dass } M,\pi^k \models g_1 \\
M,\pi \models \mathbf{G} g_1 & \Leftrightarrow & \text{für alle } k \geq 0 \text{ gilt } M,\pi^k \models g_1 \\
M,\pi \models g_1 \mathbf{U} g_2 & \Leftrightarrow & \text{es gibt ein } k \geq 0 \text{, so dass } M,\pi^k \models g_2 \text{, und} \\
 & & \text{für alle } 0 \leq j < k \text{ gilt } M,\pi^j \models g_1 \\
M,\pi \models g_1 \mathbf{R} g_2 & \Leftrightarrow & \text{für alle } k \geq 0 \text{: wenn für jedes } j < k \; M,\pi^j \not\models g_1, \\
 & & \text{so gilt } M,\pi^k \models g_2
\end{array}
$$

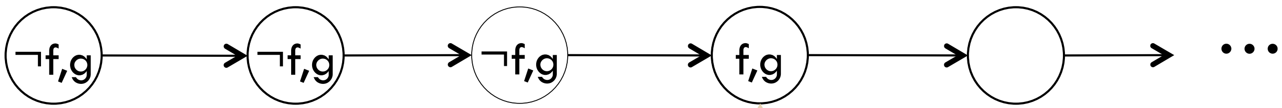
Darstellung der Operatoren **U** und **R**

f**U**g:



Es gibt einen Zustand, ab dem (mindestens einmal) g gilt.
Bis zu diesem Zustand gilt immer f.

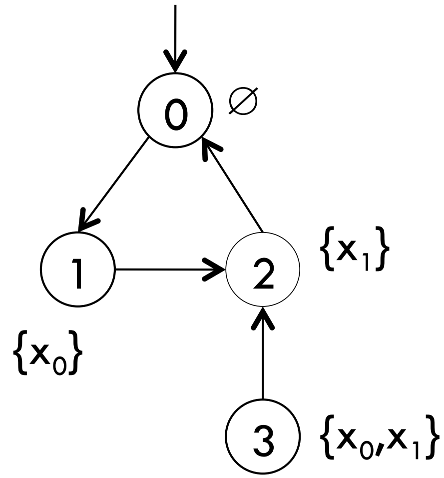
f**R**g:



g gilt so lange (einschließlich), bis irgendwann einmal f wahr wird.
Es muss allerdings keinen Zustand geben, in dem f wahr wird.

Semantik: Beispiel

2-bit-Zähler 0-2:



- Zustandsformeln:
  - $M,0 \models x_0$? $M,1 \models x_0 \vee x_1$?
  - In welchen Zuständen gilt $\mathbf{EF}(x_0 \wedge x_1)$, in welchen $\neg(x_0 \wedge x_1) \Rightarrow \mathbf{AG}\neg(x_0 \wedge x_1)$?
- Pfadformeln:
  - betrachte π=012012...
  - $M,\pi \models \neg x_0$? $M,\pi \models \mathbf{F}(x_1)$?
  - $M,\pi \models x_1 \mathbf{U} \neg x_0$? $M,\pi \models \mathbf{X}(\neg x_0 \wedge x_1)$?

Einige Äquivalenzen in CTL*

$$f \wedge g \equiv \neg(\neg f \vee \neg g) \qquad f\,\mathbf{R}\,g \equiv \neg(\neg f\,\mathbf{U}\,\neg g)$$

$$\mathbf{F}\,f \equiv \mathit{true}\,\mathbf{U}\,f \qquad \mathbf{G}\,f \equiv \neg\,\mathbf{F}\,\neg f$$

$$\mathbf{A}\,f \equiv \neg\,\mathbf{E}\,\neg f$$

- **Konsequenz:**
  1. Die CTL*-Operatoren **R, F, G** und **A** können eliminiert werden (ebenso die Konjunktion ∧).
  2. Mit den Operatoren ∨,¬, **X, U** und **E** können sämtliche CTL*-Formeln ausgedrückt werden.
  3. Beweise und Algorithmen müssen nur für diese Operatoren (∨,¬, **X, U** und **E**) ausgeführt werden.

Fragestellungen der Temporallogik

- Gilt eine Eigenschaft f in einem Zustand s (einem Pfad π) einer Kripke-Struktur M? $M, s \models f$ $M, \pi \models f$
- Gilt eine Eigenschaft f in jedem Zustand (auf jedem Pfad) einer Kripke-Struktur M? $M \models f$
- Ist eine Eigenschaft f (in jeder Kripke-Stuktur) gültig? $\models f$
- Gilt eine Eigenschaft f unendlich oft auf allen Pfaden einer Kripke-Struktur M? fairness

Model-Checking-Problem der Temporallogik

- **Gegeben:** Eine Kripke-Struktur M=(S,T,L) und eine Eigenschaft f
  - Wir betrachten hier den Fall, dass alle Zustände Initialzustände sind.
- **Frage:** Bestimme alle Zustände s∈S, in denen f gilt, d.h. bestimme die Menge

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

Sub-Logiken von CTL*: (1) CTL

- Wichtige Sub-Logiken von CTL*
  - CTL: branching-time logic
  - LTL: linear-time logic
- CTL: Pfadformeln eingeschränkt
  - Sind f und g *Zustandsformeln*, so sind **X**f, **F**f, **G**f, f**U**g und f**R**g Pfadformeln.
  - **Konsequenz**: In einer Zustandsformel muss jedem Temporaloperator (**X**, **F**, **G**, **U**, **R**) direkt ein Pfadquantor vorangehen.

CTL: Semantik

$\pi = s_0 s_1 s_2 \ldots$

$$
\begin{array}{lcl}
M, s \models p & \Leftrightarrow & p \in L(s) \\
M, s \models \neg f_1 & \Leftrightarrow & M, s \not\models f_1 \\
M, s \models f_1 \vee f_2 & \Leftrightarrow & M, s \models f_1 \ \text{ oder } \ M, s \models f_2 \\
M, s \models f_1 \wedge f_2 & \Leftrightarrow & M, s \models f_1 \ \text{ und } \ M, s \models f_2 \\
M, s \models \mathbf{E} g_1 & \Leftrightarrow & \text{es gibt einen Pfad } \pi = s \ldots, \text{ so dass } M, \pi \models g_1 \\
M, s \models \mathbf{A} g_1 & \Leftrightarrow & \text{für alle Pfade } \pi = s \ldots \text{ gilt } M, \pi \models g_1 \\
M, \pi \models \mathbf{X} f_1 & \Leftrightarrow & M, s_1 \models f_1 \\
M, \pi \models \mathbf{F} f_1 & \Leftrightarrow & \text{es gibt ein } k \geq 0, \text{ so dass } M, s_k \models f_1 \\
M, \pi \models \mathbf{G} f_1 & \Leftrightarrow & \text{für alle } k \geq 0 \text{ gilt } M, s_k \models f_1 \\
M, \pi \models f_1 \mathbf{U} f_2 & \Leftrightarrow & \text{es gibt ein } k \geq 0, \text{ so dass } M, s_k \models f_2, \text{ und} \\
 & & \text{für alle } 0 \leq j < k \text{ gilt } M, s_j \models f_1 \\
M, \pi \models f_1 \mathbf{R} f_2 & \Leftrightarrow & \text{für alle } k \geq 0\text{: wenn für jedes } j < k \ M, s_j \not\models f_1, \\
 & & \text{so gilt } M, s_k \models f_2
\end{array}
$$

CTL-Operatoren

- Pfadquantoren und Temporaloperatoren treten immer gemeinsam auf
- 10 mögliche Kombinationen: **EX, EF, EG, EU, ER, AX, AF, AG, AU, AR**
- Diese 10 Kombinationen können mit Hilfe der 3 Basiskombinationen **EX, EG** und **EU** geschrieben werden, denn:

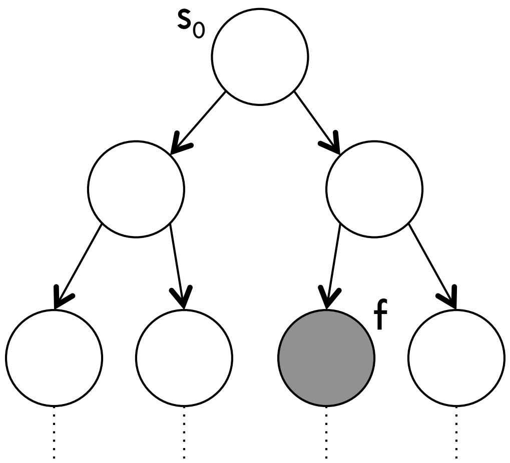
$$\mathbf{A}\,\mathbf{F}\,f \equiv \neg\,\mathbf{E}\,\mathbf{G}\,\neg f \qquad \mathbf{E}\,\mathbf{F}\,f \equiv \mathbf{E}[\mathit{true}\,\mathbf{U}\,f]$$

$$\mathbf{A}[f\,\mathbf{R}\,g] \equiv \neg\,\mathbf{E}[\neg f\,\mathbf{U}\,\neg g] \qquad \mathbf{E}[f\,\mathbf{R}\,g] \equiv \neg\,\mathbf{A}[\neg f\,\mathbf{U}\,\neg g]$$

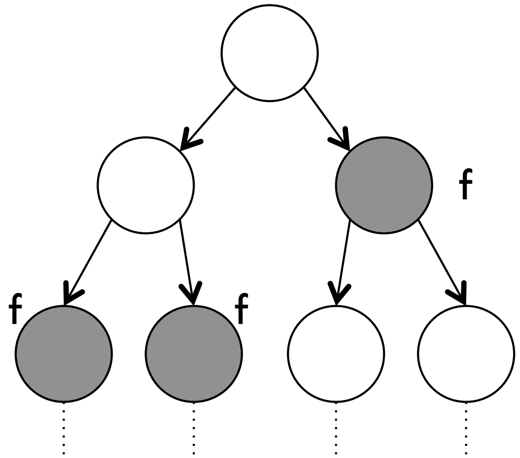
$$\mathbf{A}\,\mathbf{X}\,f \equiv \neg\,\mathbf{E}\,\mathbf{X}\,\neg f \qquad \mathbf{A}\,\mathbf{G}\,f \equiv \neg\,\mathbf{E}\,\mathbf{F}\,\neg f$$

$$\mathbf{A}[f\,\mathbf{U}\,g] \equiv \neg\,\mathbf{E}[\neg g\,\mathbf{U}(\neg f \wedge \neg g)] \wedge \neg\,\mathbf{E}\,\mathbf{G}\,\neg g$$

Darstellung typischer CTL-Operatoren (I)

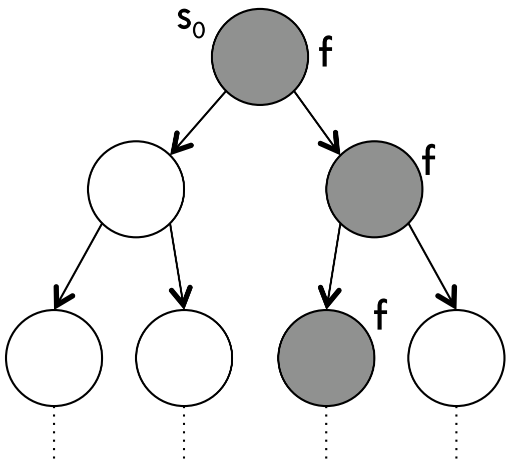


$$M, s_0 \models \mathbf{EF}\, f$$

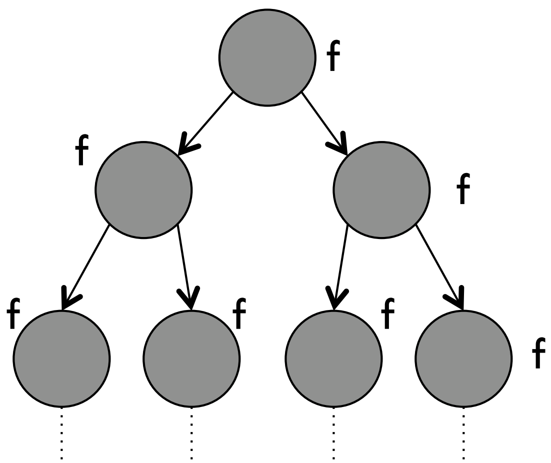


$$M, s_0 \models \mathbf{AF}\, f$$

Darstellung typischer CTL-Operatoren (II)



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



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

Sub-Logiken von CTL*: (2) LTL

- Einziger Pfadquantor ist **A**, alle Formeln haben die Form **A**f (für eine Pfadformel f)
- Pfadformeln eingeschränkt:
  - *Jedes p $\in$A ist eine Pfadformel*
  - Sind f und g Pfadformeln, so auch $\neg$f, f$\wedge$g, f$\vee$g, **X**f, **F**f, **G**f, f**U**g, f**R**g.

- **Konsequenzen:**
  - Nur ein Pfadquantor (**A**), dieser wird höufig nicht geschrieben
  - Formel f in **A**f bezieht sich immer nur auf *einen* Pfad

Semantik von LTL

$\pi = s_0 s_1 s_2 \ldots$

$$
\begin{array}{lcl}
M, \pi \models p & \Leftrightarrow & p \in L(s_0) \\
M, \pi \models \neg f_1 & \Leftrightarrow & M, \pi \not\models f_1 \\
M, \pi \models f_1 \vee f_2 & \Leftrightarrow & M, \pi \models f_1 \;\; \text{oder} \;\; M, \pi \models f_2 \\
M, \pi \models f_1 \wedge f_2 & \Leftrightarrow & M, \pi \models f_1 \;\; \text{und} \;\; M, \pi \models f_2 \\
M, \pi \models \mathbf{X} f_1 & \Leftrightarrow & M, \pi^1 \models f_1 \\
M, \pi \models \mathbf{F} f_1 & \Leftrightarrow & \text{es gibt ein } k \geq 0, \text{ so dass } M, \pi^k \models f_1 \\
M, \pi \models \mathbf{G} f_1 & \Leftrightarrow & \text{für alle } k \geq 0 \text{ gilt } M, \pi^k \models f_1 \\
M, \pi \models f_1 \mathbf{U} f_2 & \Leftrightarrow & \text{es gibt ein } k \geq 0, \text{ so dass } M, \pi^k \models f_2, \text{ und} \\
 & & \text{für alle } 0 \leq j < k \text{ gilt } M, \pi^j \models f_1 \\
M, \pi \models f_1 \mathbf{R} f_2 & \Leftrightarrow & \text{für alle } k \geq 0\text{: wenn für jedes } j < k \; M, \pi^j \not\models f_1, \\
 & & \text{so gilt } M, \pi^k \models f_2
\end{array}
$$

Weitere Sub-Logiken von CTL*

- ACTL*: CTL* ohne Pfadquantor **E**
- ACTL: CTL ohne Pfadquantor **E**
- Um implizite **E**-Quantoren durch Negation zu vermeiden, wird verlangt, dass Formeln in *positiver Normalform* (Negation nur vor atomaren Aussagen) angeschrieben werden.
- ACTL und ACTL* verbreitet in kompositioneller Verifikation und Abstraktion

Beispiel: Bahnschranke

- A = { SchrankeOffen, Rot, Gelb, Gruen, Zug }
- Gewünschte Eigenschaften:
  - Wenn ein Zug durchfährt, so ist die Schranke geschlossen und die Ampel rot
  - Wenn die Ampel grün ist, ist die Schranke offen
  - Wenn die Ampel rot ist und kein Zug durchfährt, so schaltet die Ampel im nächsten Schritt auf Rot-Gelb, und danach auf Grün
  - Ampel wird immer wieder (unendlich oft) grün

$$\mathbf{AG}((\mathrm{Rot} \Rightarrow \neg\mathrm{Gelb} \wedge \neg\mathrm{Gruen}) \wedge (\mathrm{Gelb} \Rightarrow \neg\mathrm{Rot} \wedge \neg\mathrm{Gruen}) \wedge (\mathrm{Gruen} \Rightarrow \neg\mathrm{Gelb} \wedge \neg\mathrm{Rot}))$$

$$\mathbf{AG}(\mathrm{Zug} \Rightarrow \neg\mathrm{SchrankeOffen} \wedge \mathrm{Rot})$$

$$\mathbf{AG}(\mathrm{Gruen} \Rightarrow \mathrm{SchrankeOffen})$$

$$\mathbf{AG}(\mathrm{Rot} \wedge \neg\mathrm{Zug} \Rightarrow \mathbf{AX}(\mathrm{Gelb} \wedge \mathbf{AX}\,\mathrm{Gruen}))$$

$$\mathbf{AG}(\mathbf{AF}\,\mathrm{Gruen})$$

Typische CTL-Spezifikationen

- Auf eine Anforderung erfolgt irgendwann eine Bestätigung:

  AG(Anforderung ⇒ **AF** Bestätigung)

- Eine Eigenschaft (*liveliness property*) tritt auf jedem Pfad unendlich oft auf:

  AG(**AF** live)

- Von jedem Zustand aus ist es möglich, in einen "sicheren" Zustand zu kommen:

  AG(**EF** safe)

Ausdrucksstärke

- **Definition:** Seien L und L' Temporallogiken, $f \in L$ und $g \in L'$.
  - f und g heißen *äquivalent* ($f \equiv g$), wenn $K \vDash f \Leftrightarrow K \vDash g$ für alle Kripkestrukturen K gilt.
  - L' *subsumiert* L ($L \leq L'$), wenn es für alle $f \in L$ ein $g \in L'$ gibt, so dass $f \equiv g$.
  - Ist $L \lneq L'$, so heißt L' *ausdrucksstärker* als L.
- **Satz:**
  - $CTL \lneq CTL^*$, $LTL \lneq CTL^*$
  - $CTL \nleq LTL$, $LTL \nleq CTL$

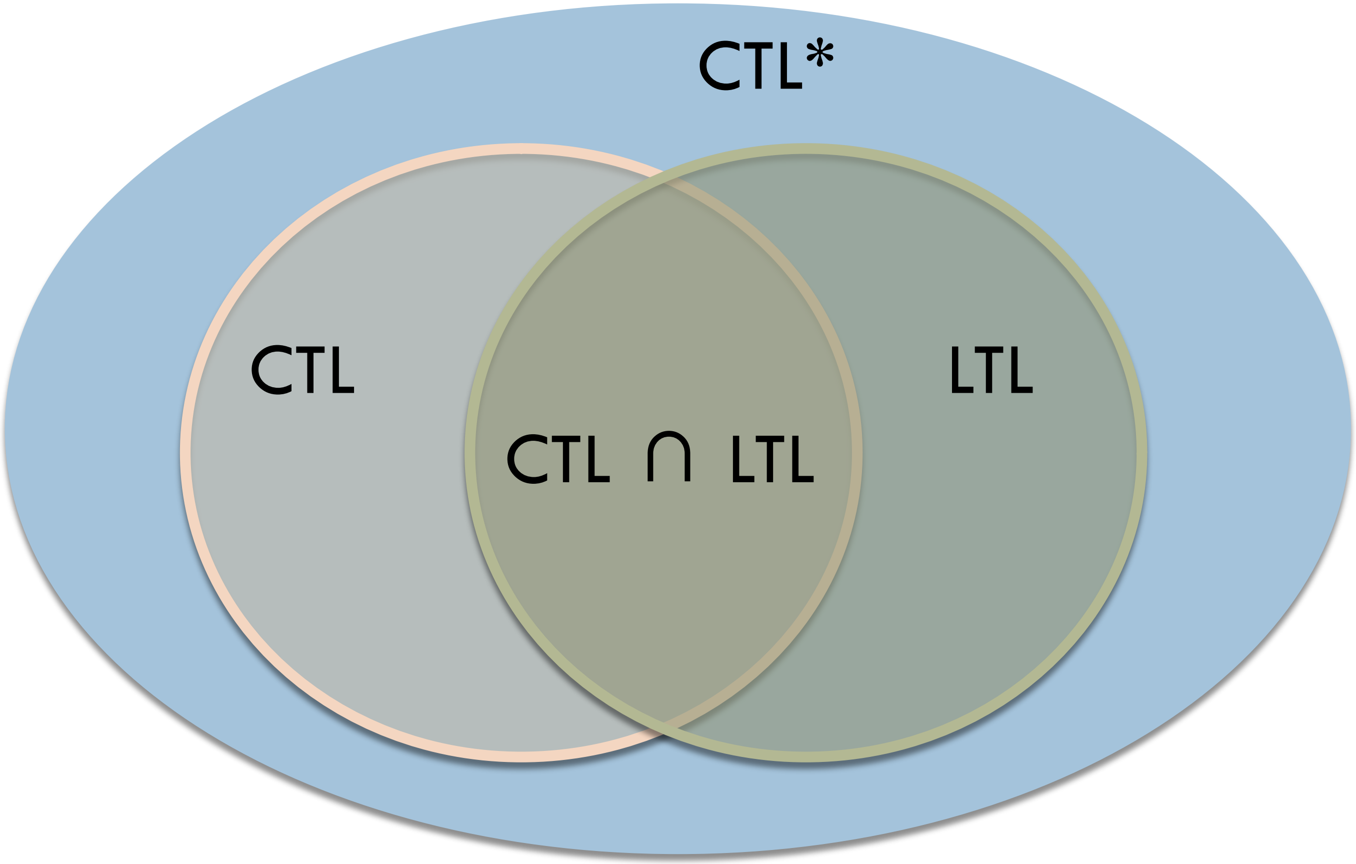
$CTL \lneq CTL^*$

- **Feststellung**: Es gibt keine CTL-Formel, die das folgende ausdrückt: Es gibt einen Pfad, auf dem unendlich oft p gilt.
  - **EG EF** p funktioniert nicht. Warum?
  - **EGF** p ist eine CTL*-Formel mit dieser Bedeutung
- **Satz**: Es gibt eine CTL*-Formel f, zu der es keine äquivalente CTL-Formel gibt.

LTL $\not\leq$ CTL, CTL $\not\leq$ LTL

- **Feststellung:** Es gibt CTL-Eigenschaften, die nicht in LTL ausgedrückt werden können und umgekehrt.
  - LTL-Formel "**AFG** q" lässt sich in CTL nicht ausdrücken
  - CTL-Formel "**AG EF** q" lässt sich in LTL nicht ausdrücken
- Häufig behauptet:
  Die meisten Spezifikationen liegen in CTL ∩ LTL.

Einordnung der Temporallogiken bzgl. Ausdrucksstärke



Weitere wichtige Eigenschaften von CTL und LTL

- CTL und LTL "**können nicht zählen**", d.h. die Eigenschaft "f gilt in jedem k-ten Schritt" (für $k \geq 2$) lässt sich in CTL und LTL nicht ausdrücken
  - Beachte: *Genau* in jedem k-ten Schritt geht, z.B. für $k=2$:
    $\mathbf{A}(p \land \mathbf{G}(p \Leftrightarrow \mathbf{X}\neg p))$
    lässt sich spezifizieren