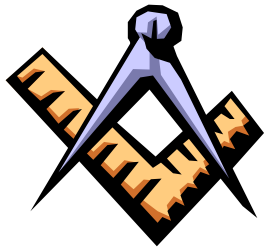
Lineare Temporale Logik

H. Peter Gumm

Philipps-Universität Marburg

Sommersemester 2007

Logiken für Kripke Strukturen



Logiken, die nicht nur Zustandseigenschaften, sondern auch das dynamische Verhalten beschreiben können.

Mögliche logische Sprachen für Kripke-Strukturen unterscheiden sich bezüglich

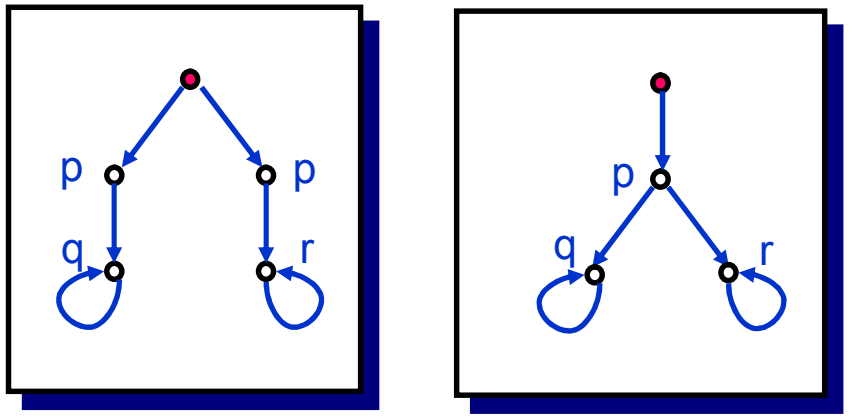
- **Ausdrucksstärke:** welche interessierenden Eigenschaften sind ausdrückbar
- **Trennschärfe:** wann lassen sich zwei Zustände durch eine Formel unterscheiden ?

Wir werden drei Sprachen untersuchen: LTL, CTL und CTL*.

LTL und CTL haben zwar vieles gemeinsam, dennoch gibt es in jeder Sprache (LTL und CTL) einen Ausdruck, der in der jeweils anderen nicht ausdrückbar ist. CTL* umfaßt LTL und CTL.

Logiken für Kripke Strukturen

- Einbeziehen des dynamischen Verhaltens
  - gegeben durch Relation R
- LTL - lineare temporale Logik
  - Aussagen über Spuren
  - welche Eigenschaften gelten für alle Spuren
- CTL - Computation Tree Logic
  - Aussagen über Berechnungen und Zustände
  - welche Optionen bestehen in einem Zustand
- CTL* - beinhaltet LTL und CTL
  - allerdings schwieriger zu handhaben

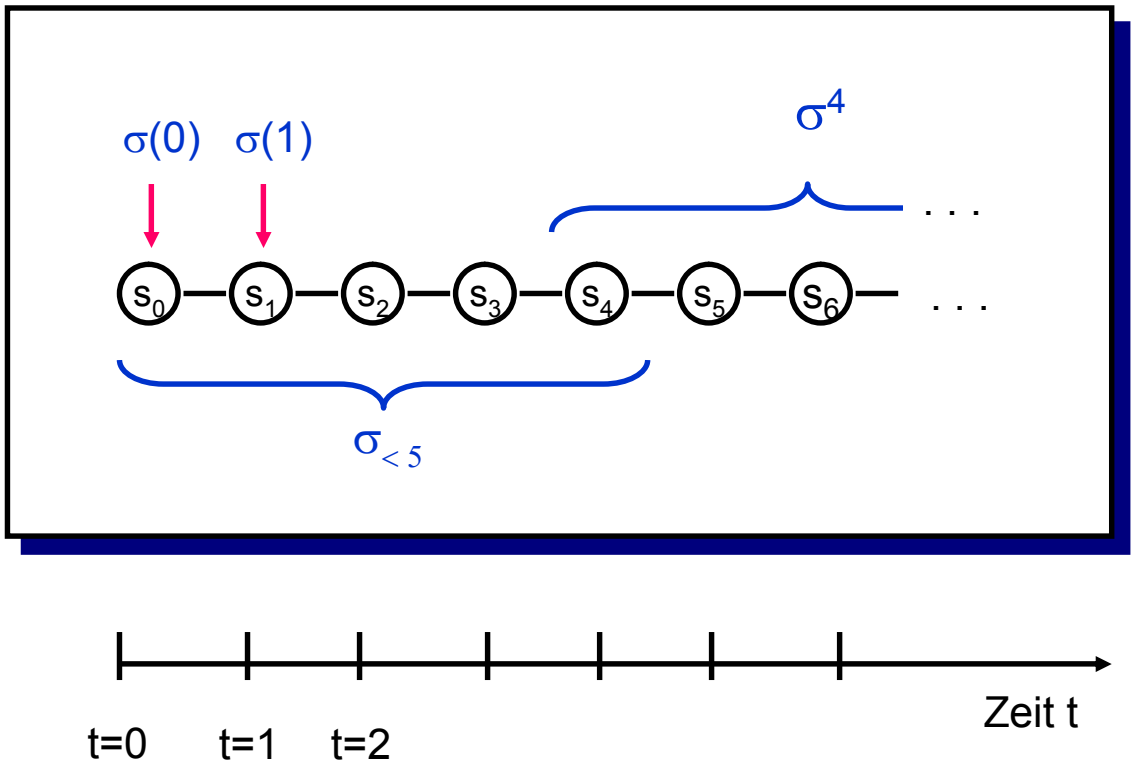


Die Berechnungen, die in den rot markierten Punkte beginnen, sind anhand ihrer Spuren nicht unterscheidbar.

Die Option, einen Zustand mit q oder r zu erreichen bleibt in dem rechten System länger offen.

Notation für Folgen

- Sei $\sigma = (s_0, s_1, s_2, \ldots)$ eine unendliche Folge,
  - $\sigma: \mathbb{N} \rightarrow S$.
- i-tes Glied:
  - $\sigma(i) = s_i$.
- Restfolge ab i-tem Glied :
  - $\sigma^i := (s_i, s_{i+1}, s_{i+2}, \ldots)$
  - Formal: $\sigma^i(n) = \sigma(i+n)$
- i-ter Präfix :
  - $\sigma_{<i} := (s_0, s_1, s_2, \ldots, s_{i-1})$
- $S^\omega := [\mathbb{N} \rightarrow S]$
  - Menge aller unendlichen Folgen in S
- S*
  - Menge aller endlichen Folgen



Berechnung, Spur

Ein Prozess startet in einem Zustand s_0 und durchläuft Reihe von Folgezuständen

$$s_0, s_1, s_2, \ldots .$$

Wir betrachten nur nicht-terminierende Prozesse.

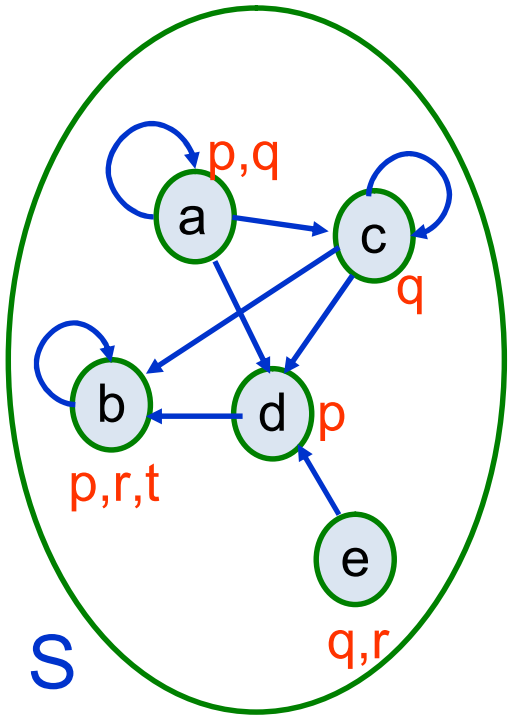
Berechnung (Pfad):
Die unendliche Folge $s = (s_0, s_1, s_2, \ldots)$ der durchlaufenen Zustände.

Zustand eines Prozesses ist nie direkt beobachtbar (siehe Digitaluhr), sondern nur gewisse Eigenschaften (z.B.: Display)

Spur (einer Berechnung)
Die Folge **der beobachteten Eigenschaften**

Eine Spur ist also ein beobachtbares Verhalten des Systems.

Berechnung und Spur in Kripke-Struktur



Berechnung: Unendliche Folge von Zuständen

$\sigma = (s_0, s_1, s_2, \ldots) \in S^\omega$

mit

$\forall\, i \in \mathbb{N}.\ (s_i, s_{i+1}) \in R.$

Menge aller in s startenden unendlichen Pfade:

$\text{Paths}(s) := \{ \sigma \in S^\omega \mid \sigma(0) = s \ \wedge\ \forall\, k \in \mathbb{N}.\ \sigma(k)\ R\ \sigma(k+1) \}$

Spur (engl. trace) der Berechnung $\sigma = (s_0, s_1, s_2, \ldots)$:

$\tau = \text{trace}(\sigma) = (L(s_0), L(s_1), L(s_2), \ldots)$

Menge aller in s startenden Spuren:

$\text{Traces}(s) := \{ \text{trace}(\sigma) \mid \sigma \in \text{Paths}(s) \}$

Beispiel einer Berechnung

$\sigma = (a, a, c, b, b, b, \ldots)$

zugehörige Spur:

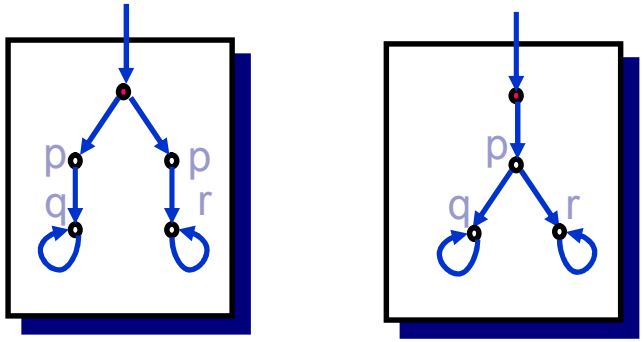
$\tau = (\{p,q\}, \{p,q\}, \{q\}, \{p,r,t\}, \{p,r,t\}, \{p,r,t\}, \ldots)$

Spur-Äquivalenz

Zustand eines Prozesses ist nie direkt beobachtbar ist, sondern nur dessen Eigenschaften.

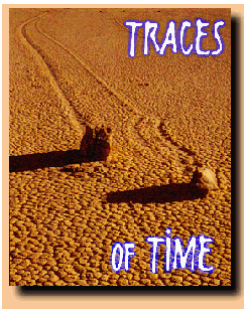
Spur-Äquivalenz: Pfade heißen spur-äquivalent, wenn ihre Spuren gleich sind, d.h ihr beobachtbares Verhalten ist gleich.

Zwei Pfade σ und σ' heißen spur-äquivalent, falls
trace(σ) = trace(σ').



Beispiel: Zu jeder Berechnung in einem System gibt es eine spur-äquivalente Berechnung im anderen.

Temporale Deutung



Berechnung ist Pfad $\mathbf{s} = (\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \ldots)$, mit $s_0 \in I$, $\mathbf{s}_i \in S$ für alle $i \in \mathbb{N}$

$\mathbb{N}$ kann als Zeitverlauf gedeutet werden:

$\mathbf{s}_t$: Zustand zur Zeit t

$L(s_t)$: Eigenschaften des Systems zur Zeit t.

Zeitmodell ist

diskret — im Gegensatz zu kontinuierlich

linear — Zu jedem Zeitpunkt gibt es genau einen nächsten Zeitpunkt

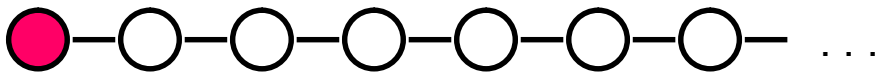
initial — Es gibt einen Anfangszeitpunkt

Lineare Temporale Logik :

Sprache und Kalkül um Eigenschaften von Zustandsfolgen auszudrücken, ohne die Zeit explizit zu erwähnen.

Eigenschaften von Folgen

init p - Am Anfang gilt p

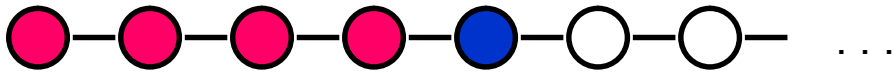


◊ p - sometimes p

□ p - always p

O p - nexttime p

p **U** q - p until q



Zustand, in dem p gilt

Zustand, in dem q gilt

LTL – die Logik der Berechnungen

Wir definieren nun eine Sprache, in der wir Aussagen über Folgen von Zuständen formulieren können. Diese Sprache heißt (P)LTL (propositional) linear temporal logic.

BNF Definition von temporalen Aussagen:

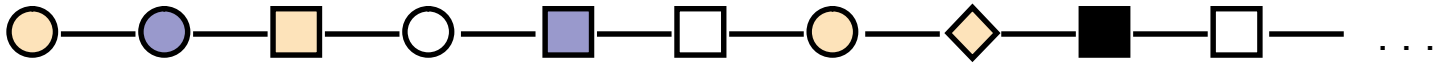
```
TProp  ::= init Prop
         | X TProp
         | F TProp
         | G TProp
         | TProp U TProp
         | TProp W TProp
         | TProp ∧ TProp
         | TProp ∨ TProp
         | TProp ⇒ TProp
         | TProp ⇔ TProp
         | ¬ TProp
         | true
         | false
         | ( TProp )
```

Sprechweisen:

anfangs
next, danach, O
irgendwann, sometimes,eventually, ◊
immer, always, forever, □
bis, until
weak until, unless, waits for

Das Schlüsselwort **init** wird meist unterschlagen, es ergibt sich stets aus dem Kontext.

Beispiele



Hier gehen wir von einer Menge **AP** = { braun, blau, schwarz, weiß, rund,quadratisch, karo } aus.

Einige Eigenschaften, die oben gelten:

init (braun $\wedge$ rund)
F(schwarz)
X (blau)
XX quadratisch
X (rund $\Rightarrow$ **X** braun)
braun **U** blau
rund **W** quadratisch
F (karo **W** schwarz)

Einige Eigenschaften, die oben **nicht** gelten:

G (braun $\wedge$ rund)
G X (blau)
G (braun **U** blau)
braun **U G** blau

Eigenschaften, die oben noch gelten können:

GF braun
FG braun
FG schwarz **U** weiß
G $\neg$ rot

Was bedeuten wohl folgende Eigenschaften :

(F braun **) U** rot
(braun **U** blau**) U** braun
G rot **U** weiß
G $\neg$ **G** rot

Formale Semantik von LTL

(S, AP, L) Kontext über AP , also $\models_L \subseteq S \times AP$ mit $s \models_L p :\Leftrightarrow p \in L(s)$

P Menge aller aus AP gebildeten zusammengesetzten Aussagen, P = B(AP).
S^ω die Menge aller Zustandsfolgen (über S).

Für eine beliebige Folge $\sigma \in S^\omega$ und ein beliebiges φ aus **TProp** müssen wir

$\sigma \models \varphi$ definieren, also $\models \subseteq S^\omega \times TProp$

TProp ::= **init** P
| **X** TProp
| **F** TProp
| **G** TProp
| TProp **U** TProp
| TProp **W** TProp
| TProp $\wedge$ TProp
| .. etc. ..

$\sigma \models$ **init p** $:\Leftrightarrow$ $p \in L(s_0)$

$\sigma \models$ **X** φ $:\Leftrightarrow$ $\sigma^1 \models \varphi$

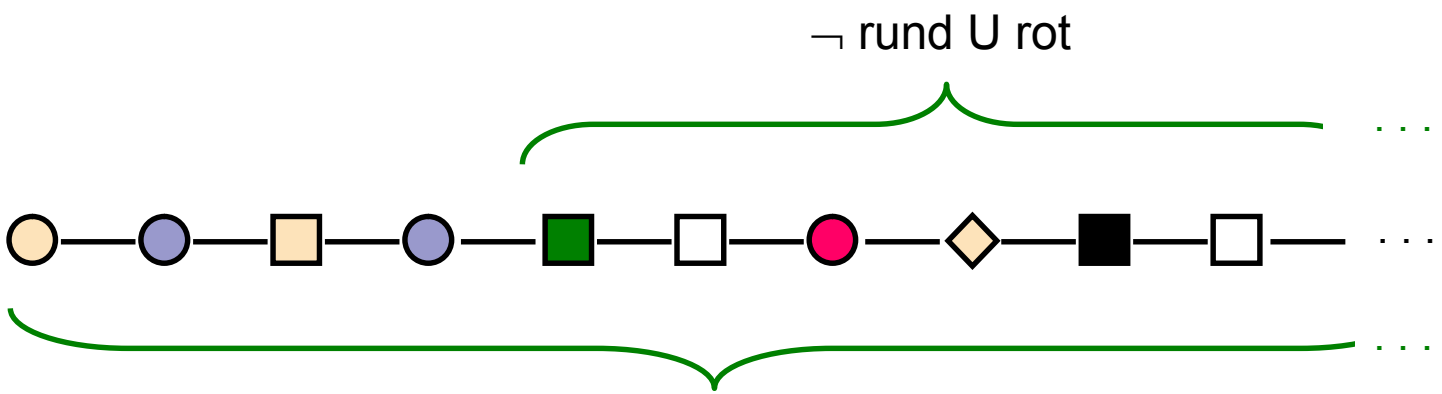
$\sigma \models \varphi$ **U** ψ $:\Leftrightarrow$ $\exists\, k \in Nat.\ \sigma^k \models \psi \wedge \forall\, i < k.\ \sigma^i \models \varphi.$

Die übrigen temporalen Operatoren verstehen wir als Abkürzungen :

F φ : Abkürzung für **true U** φ
G φ : Abkürzung für $\neg$**F**$\neg$ φ
φ **W** ψ : Abkürzung für **G** $\varphi \vee (\varphi$ **U** $\psi)$

Die Semantik der Booleschen Junktoren definieren wir wie vorher.

Beispiele – ganz sicher



F (¬rund U rot)

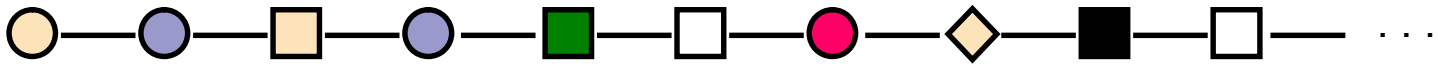
(blau ∨ braun) U (¬rund U rot)

F rot U (grün U weiß)

F (schwarz ∧ X weiß)

(rund U braun) U (¬braun U rot)

Beispiele - möglicherweise



F (schwarz $\wedge$ rund)	- möglicherweise
G F rot	- möglicherweise
F G rot	- möglicherweise
$\neg$pink U dreieckig	- möglicherweise
$\neg$grün U pink	- nein
rund U rot	- nein

G $\neg$pink	- Widerlegung durch endliche Folge, Verifikation durch unendliche
G F rot	- Widerlegung und Verifikation durch unendliche Folge
$\neg$pink U dreieckig	- Widerlegung und Verifikation durch endliche oder unendliche Folge

Strukturen und Modelle der LTL

Linear temporaleStruktur : Paar **(C, σ)** mit **C=(S,AP,L)** Kontext und $\sigma \in S^{\omega}$.

(C,σ) heißt Modell *von* $\varphi \in$ PLTL , falls $\sigma \models_L \varphi$.

φ heißt
- erfüllbar, falls es ein Modell von φ gibt.
- widersprüchlich, falls φ nicht erfüllbar ist
- allgemeingültig , falls jede linear temporale Struktur ein Modell von φ ist.

Temporallogische Aussagen φ und ψ heißen semantisch äquivalent, ($\varphi \equiv \psi$), wenn für jede linear temporale Struktur (C,σ) gilt :

$\sigma \models_L \varphi$ gdw. $\sigma \models_L \psi$.

Beispiele : Mit rot, gelb,grün ∈ **P** gilt :

rot ⇒ G (rot ⇒ (¬ grün ∧ ¬ gelb))	ist erfüllbar, aber nicht allgemeingültig
G rot ∧ F (grün U ¬ rot)	ist widersprüchlich
(grün W ff) ⇒ G grün	ist allgemeingültig und
(tt U grün) $\equiv_{LTL}$ F grün	(semantische Äquivalenz)

Wichtige Äquivalenzen

- Dualität
  - $\neg G\phi \equiv F\neg\phi$
  - $\neg F\phi \equiv G\neg\phi$
  - $\neg X\phi \equiv X\neg\phi$
- Idempotenz
  - $\neg GG\phi \equiv G\neg\phi$
  - $\neg FF\phi \equiv F\neg\phi$
  - $\phi \mathrel{U} \phi \mathrel{U} \psi \equiv \phi \mathrel{U} \psi$
- Absorption
  - $FGF\phi \equiv GF\phi$
  - $GFG\phi \equiv FG\phi$
- Distributivität X
  - $XF\phi \equiv FX\phi$
  - $XG\phi \equiv GX\phi$
  - $X(\phi \mathrel{U} \psi) \equiv X\phi \mathrel{U} X\psi$
- Distributivität U
  - $\phi \mathrel{U} (\psi \vee \chi) \equiv (\phi \mathrel{U} \psi) \vee (\phi \mathrel{U} \chi)$
  - $(\phi \wedge \psi) \mathrel{U} \chi \equiv (\phi \mathrel{U} \chi) \wedge (\psi \mathrel{U} \chi)$
- Spezialfälle
  - $F(\phi \vee \psi) \equiv F\phi \vee F\psi$
  - $G(\phi \wedge \psi) \equiv F\phi \wedge F\psi$

Fixpunktgleichungen

Fixpunkte

- $\Box F\phi \equiv \phi \vee X F\phi$
- $\Box G\phi \equiv \phi \wedge X G\phi$
- $\Box \phi \, U \, \psi \equiv \psi \vee (\phi \wedge X (\phi \, U \, \psi))$

Fixpunktgleichungen legen rekursive Definition nahe:

$F\phi := \phi \vee XF\phi$

- wo ist der Rekursionsanfang ?
- Lösung nicht eindeutig:
  - $true = \phi \vee X\, true$
  - $F\phi = \phi \vee X F\phi$

Approximation von unten

- Fixpunktgleichung: $f = \psi \vee (\phi \wedge X f))$
- Eine Lösung:

$$\phi \, U \, \psi \;=\; \psi \vee (\phi \wedge X \, (\phi \, U \, \psi \,) \,)$$

- Approximation von unten :

$P_0 = \text{false},$

$P_1 = \psi \vee (\phi \wedge X \text{ false}) = \psi$

$P_2 = \psi \vee (\phi \wedge X \, \psi)$

$P_3 = \psi \vee (\phi \wedge X \, (\psi \vee (\phi \wedge X \, \psi) \,) \,)$

$= \psi \vee (\phi \wedge X\psi) \vee (\phi \wedge X \, \phi \wedge X \, X \, \psi \,)$

allgemein:

$P_{i+1} = \psi \vee (\phi \wedge X \, (P_i) \,)$

$\phi \, U \, \psi = P_0 \vee P_1 \vee \ldots$

Approximation von oben

- Fixpunktgleichung: $f = \psi \vee (\phi \wedge X f))$
- Andere Lösung:

$$\phi \, W \, \psi \;=\; \psi \vee (\phi \wedge X \, (\phi \, W \, \psi \,) \,)$$

- Approximation von oben :

$P_0 = \text{true},$
$P_1 = \psi \vee (\phi \wedge X \, \text{true}) = \psi \vee \phi$
$P_2 = \psi \vee (\phi \wedge X \, (\psi \vee \phi))$
$P_3 = \psi \vee (\phi \wedge X \, (\psi \vee (\phi \wedge X \, (\psi \vee \phi) \,) \,) \,)$
$= \psi \vee (\phi \wedge X \, \psi \,) \vee (\phi \wedge X \, \phi \wedge X \, X \, \psi \vee \phi \,)$
$P_i = \psi \vee (\phi \wedge X \psi) \vee \ldots \vee (\phi \wedge X \phi \wedge XX \phi \wedge \ldots \wedge X^i \, (\phi \vee \psi))$

allgemein:

$\phi \, W \, \psi = P_0 \wedge P_1 \wedge \ldots$

Fixpunktberechnung

- $F\phi = \phi \vee X\, F\phi$
  $= \phi \vee X\,(\phi \vee X\, F\,\phi)$
  $= \phi \vee X\,\phi \vee X\,X\,F\,\phi =$
  $= \phi \vee X\,\phi \vee X\,X\,(\phi \vee X\, F\,\phi)$
  $= \phi \vee X\,\phi \vee X\,X\,\phi \vee X\,X\,X\,F\,\phi =$

  $= \ldots$

- Approximation von unten :

  $P_0 = \text{false}$,
  $P_1 = \phi \vee X\ \text{false} = \phi$
  $P_2 = \phi \vee X\,\phi \vee X\,X\ \text{false} = \phi \vee X\,\phi$
  $P_3 = \phi \vee X\,\phi \vee X\,X\,\phi$

allgemein:

$P_{i+1} = \phi \vee XP_i$

$F\phi = P_0 \vee P_1 \vee \ldots$

Beweis: $\sigma \vDash F\phi \Leftrightarrow \exists\, n \in \mathbb{N}.\ \sigma^n \vDash \phi$
$\Leftrightarrow \exists\, n \in \mathbb{N}.\ \sigma \vDash X^n \phi$
$\Leftrightarrow \exists\, n \in \mathbb{N}.\ \sigma \vDash P_n.$

- $G\phi = \phi \wedge X\, G\phi$
  $= \phi \wedge X\,(\phi \wedge X\, G\,\phi)$
  $= \phi \wedge X\,\phi \wedge X\,X\,G\,\phi =$
  $= \phi \wedge X\,\phi \wedge X\,X\,(\phi \wedge X\, G\,\phi)$
  $= \phi \wedge X\,\phi \wedge X\,X\,\phi \wedge X\,X\,X\,G\,\phi =$

  $= \ldots$

- Approximation von oben:

  $Q_0 = \text{true}$,
  $Q_1 = \phi \wedge X\ \text{true}$
  $Q_2 = \phi \wedge X\,\phi \wedge X\ \text{true}$
  $Q_3 = \phi \wedge X\,\phi \wedge X\,X\,\phi \wedge X\ \text{true}$

allgemein:

$Q_{i+1} = \phi \wedge XQ_i$

$G\phi = Q_0 \wedge Q_1 \wedge \ldots$

Beweis: $\sigma \vDash G\,\phi \Leftrightarrow \forall\ n \in \mathbb{N}.\ \sigma^n \vDash \phi$
$\Leftrightarrow \forall\ n \in \mathbb{N}.\ \sigma \vDash X^n \phi$
$\Leftrightarrow \forall\ n \in \mathbb{N}.\ \sigma \vDash Q_n\,.$

Releases

- $\neg G\phi \equiv F\neg\phi$
- $\neg F\phi \equiv G\neg\phi$
- $\neg X\phi \equiv X\neg\phi$
- $\neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$
- $\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$

aber:

- $\neg(\phi \, U \, \psi) \equiv \neg\phi \; ? \; \neg\psi$

- Idee: Neuer Operator R mit
- $\phi \, R \, \psi \; := \; \neg(\neg\phi \, U \, \neg\psi)$

Dann gelten

- $\neg(\phi \, U \, \psi) \equiv \neg \, \phi \, R \, \neg\psi$
- $\neg(\phi \, R \, \psi) \equiv \neg\phi \, U \neg\psi$

Semantik:

$\sigma \models \phi \, R \, \psi \quad \Leftrightarrow \; \neg \, (\exists k \in \mathbb{N}.\sigma^k \models \neg\psi \wedge \forall \, j<k.\sigma^k \models \neg\phi)$

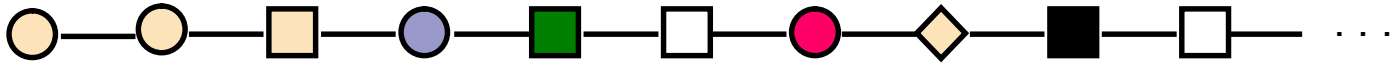
$\Leftrightarrow \forall \, k \in \mathbb{N}.(\, \sigma^k \models \psi \vee \exists \, j<k. \, \sigma^j \models \phi \,)$

$\Leftrightarrow \forall \, k \in \mathbb{N}.(\, \sigma^k \models \psi \,) \vee \exists \, r \in \mathbb{N}. \, (\, \sigma^r \models \phi \wedge \forall \, j \leq r.\sigma^j \models \psi)$

wähle r minimal mit $\sigma^r \models \phi$

Releases

$$\sigma \vDash \phi \, R \, \psi \quad \Leftrightarrow \quad \forall\, k \in \mathbb{N}.(\sigma^k \vDash \psi) \vee \exists\, r \in \mathbb{N}.\ (\sigma^r \vDash \phi \wedge \forall\, j \leq r. \sigma^j \vDash \psi)$$



- quadratisch R braun
- ¬(blau R braun)
- braun R braun

Rekursionsgleichungen

Die folgenden Äquivalenzen können verwendet werden, um ausgehend von **X** alle anderen temporalen Operatoren (rekursiv) zu definieren :

$$\mathbf{F}\varphi \equiv \varphi \vee \mathbf{X}\mathbf{F}\varphi \qquad \mathbf{G}\varphi \equiv \varphi \wedge \mathbf{X}\mathbf{G}\varphi$$

Es gilt nämlich :

1. **F**φ ist die **stärkste** Eigenschaft **E**, die die Gleichung $\mathbf{E} \equiv \varphi \vee \mathbf{X}\mathbf{E}$ erfüllt
2. **G**φ ist die **schwächste** Eigenschaft **E**, die die Gleichung $\mathbf{E} \equiv \varphi \wedge \mathbf{X}\mathbf{E}$ erfüllt

Es gilt weiterhin :

$$\varphi \mathbf{U} \psi \equiv \psi \vee (\varphi \wedge \mathbf{X}(\varphi \mathbf{U} \psi))$$

$$\varphi \mathbf{W} \psi \equiv \psi \vee (\varphi \wedge \mathbf{X}(\varphi \mathbf{W} \psi))$$

Daher wird die Gleichung $\mathbf{E} \equiv \psi \vee (\varphi \wedge \mathbf{X}\,\mathbf{E})$ sowohl durch $\mathbf{E} = \varphi \mathbf{U} \psi$, als auch durch $\mathbf{E} = \varphi \mathbf{W} \psi$ gelöst. Inwieweit wird einer der Operatoren (**U** bzw. **W**) durch die Gleichung definiert ?

Ausdrucksstärke

Mit Hilfe der temporalen Logik lassen sich viele wichtige Eigenschaften von Systemverläufen ausdrücken:

unendlich oft φ	:	**G F** φ	(man schreibt auch $\mathbf{F}^{\infty}$ für **G F**)
ab irgendwann φ	:	**F G** φ	(man schreibt auch $\mathbf{G}^{\infty}$ für **F G**)

Wichtig sind auch ***Fairness-Aussagen***, die ausdrücken, dass jede Transition, die oft genug bereit ist, auch oft genug ausgeführt wird. Seien n Transitionen gegeben, dann drücken wir für k = 1,..,n aus

$enabled_k$: Transition k ist bereit

$executed_k$: Transition k wird ausgeführt.

Unbedingte Fairness :	$\bigwedge_{k=1,..,n} (\ \mathbf{G\ F}\ executed_k\ )$
Schwache Fairness :	$\bigwedge_{k=1,..,n} (\ \mathbf{F\ G}\ enabled_k \Rightarrow \mathbf{G\ F}\ executed_k\ )$
Starke Fairness :	$\bigwedge_{k=1,..,n} (\ \mathbf{G\ F}\ enabled_k \Rightarrow \mathbf{G\ F}\ executed_k\ )$

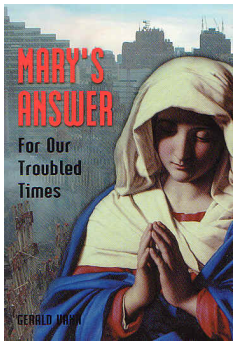
Spezifikationsmuster

- Gewisse Muster tauchen in Spezifikationen immer wieder auf.
  Sie lassen sich in LTL formulieren
  - http://patterns.projects.cis.ksu.edu/documentation/patterns/ltl.shtml
- Wir benutzen hier W (weak until, unless) mit der Definition

  $p\ W\ q \quad :\Leftrightarrow \quad (G\ p) \vee (p\ U\ q)$.

Universalität	p gilt immer…
…	$G\ p$
… vor jedem r	$F\ r \Rightarrow (p\ U\ r)$
… ab erstem q	$G\ (q \Rightarrow G\ p)$
.. zwischen q und r	$G((q \wedge \neg r \wedge Fr) \Rightarrow (p\ U\ r))$

Existenz	p gilt irgendwann …
…	$F\ p$
.. vor dem ersten r	$\neg r\ W\ (p \wedge \neg r)$
.. nach einem q	$G\neg q \ \vee \ F\ (q \wedge F\ p)$
.. zwischen q und r	$G\ (q \wedge \neg r \Rightarrow (\neg r\ W\ (p \wedge \neg r))$



Spezifikationsmuster

- Wichtig sind Präzedenz und Response-Eigenschaften

Präzedenz	s vor p …
… immer	$\neg p \; W \; s$
… vor r	$F\, r \Rightarrow (\neg p \; U \; (s \vee r))$
… nach q	$G\neg q \vee F(q \wedge (\neg p \; W \; s))$
..zwischen q und r	$G\,((q \wedge \neg r \wedge Fr) \Rightarrow (\neg p \; U \; (s \vee r)))$

Response	s reagiert auf p …
… immer	$G\,(p \Rightarrow F\, s)$
… vor r	$F\, r \Rightarrow (p \Rightarrow (\neg r U\, (s \wedge \neg r))) \; U \; r$
… nach q	$G\,(q \Rightarrow G(p \Rightarrow Fs))$
..zwischen q und r	$G((q \wedge \neg r \wedge F\, r) \Rightarrow (p \Rightarrow (\neg r \; U \; (s \wedge \neg r))) \; U \; r)$

LTL für Spezifikationen

- Viele Eigenschaften lassen sich leicht durch LTL-Formeln beschreiben, man kann aber auch undurchsichtige Formeln bauen.
- Unhandlich wird es, wenn die Zeitdauer mitkodiert werden soll:
  - Nach höchstens k Schritten gilt …
- Es existieren Spracherweiterungen in denen man häufige Muster bequem formulieren kann. Diese werden dann nach LTL compiliert.
- Alternative Kalküle - z.B. UNITY kommen mit einer Teilsprache von LTL aus. Hier kann man temporale Operatoren nicht schachteln. Mit Zustandseigenschaften **p** und **q** verwendet man nur die Konnektoren **init**, **unless** und **leadsTo** :

  p leadsTo q : $\mathbf{G}\,(\,p \Rightarrow \mathbf{F}\,q\,)$

  p unless q : $\mathbf{G}\,(\,p \Rightarrow p\ \mathbf{W}\ q\,)$

LTL für Zustände und Systeme

- LTL-Formeln gelten für Pfade σ:

  $\sigma \models \phi$

- Für Kripke-Struktur K mit Zustand $s \in S$ definiere

  $K,s \models A\phi$ $:\Leftrightarrow \forall\, \sigma \in \text{Paths}(s).\ \sigma \models \phi$

- und falls $I \subseteq S$ die Menge der Anfangszustände ist:

  $K \models \phi$ $:\Leftrightarrow \forall\, s \in I.\ K,s \models A\phi$

In NuSMV: **`LTLSPEC`** ϕ

Vorsicht: Es kann sein, dass weder $K,s \models A\phi$ noch $K,s \models A\neg\phi$ gilt, bzw weder $K \models \phi$ noch $K \models \neg\phi$!

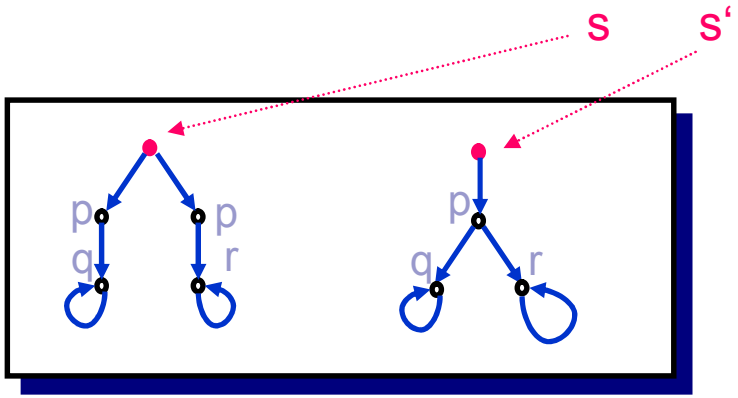
LTL für Zustände von Kripke Strukturen

Sei **K = (S,R,L)** eine Kripke-Struktur über P. Wie genau können wir Zustände **s** durch LTL-Formeln spezifizieren ?

$$s \models A\,\varphi$$

Zustände **s,** und **s'** heißen **LTL-äquivalent**, falls für jede LTL-Formel φ gilt:

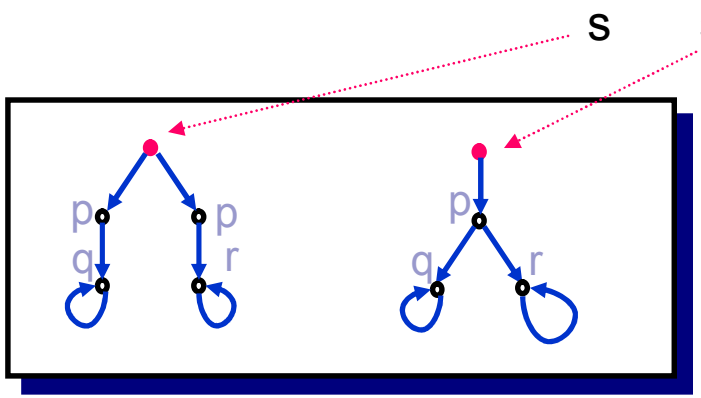
$$s \models A\varphi \quad \Leftrightarrow \quad s' \models A\varphi$$



Sind s und s' LTL-äquivalent ?

Spur-Äquivalenz

Zwei Zustände **s,** und **s'** heißen **spur-äquivalent**, falls Traces(s) = Traces(s').



s und **s'** sind offensichtlich spur-äquivalent

Da die Gültigkeit von LTL-Formeln über die Spuren definiert ist, folgt sofort:

Sind s und s' spur-äquivalent, dann sind sie LTL-äquivalent

Anfangsstücke von Berechnungen

- Sei L(s) endlich für jedes $s \in S$. Die Spur einer Berechnung lässt sich bis zu jedem beliebigen endlichen Zeitpunkt eindeutig durch eine LTL-Formel festlegen:
- Für einen Zustand s setze

$$\Delta(s) = \bigwedge\{\phi \mid \phi \in L(s)\} \wedge \bigwedge\{\neg\phi \mid \phi \notin L(s)\}$$

- für eine Folge $\sigma = (s_0, s_1, \ldots, s_n, s_{n+1}, \ldots)$ und $k \in \mathbb{N}$ setze

$$\Delta_k(\sigma) = \Delta(s_0) \wedge X\Delta(s_1) \wedge \ldots \wedge X^k\Delta(s_k).$$

- Für einen beliebigen Pfad τ gilt dann:

$$\tau \models \Delta_k(\sigma) \Leftrightarrow \text{trace}(\tau)_{\leq k} = \text{trace}(\sigma)_{\leq k}$$

- Folgerung: Sind s und t LTL-äquivalent, so gibt es zu jedem $k \in \mathbb{N}$ und zu jedem $\sigma \in$ Paths(s) ein $\tau \in$ Paths(t) mit $\text{trace}(\sigma)_{\leq k} = \text{trace}(\tau)_{\leq k}$.
- Beweis: Sei $\sigma \in$ Paths(s). Dann gilt $s \not\models \neg\Delta_k(\sigma)$, also $t \not\models \neg\Delta_k(\sigma)$. Also muss ein Pfad $\tau \in$ Paths(t) existieren mit $\tau \models \neg\Delta_k(\sigma)$.

LTL und Spur-Äquivalenz

Satz: Sei K eine bild-endliche Kripke-Struktur. Genau dann sind s und t LTL-äquivalent, wenn sie spur-äquivalent sind.

Seien s und t LTL-äquivalent und $\sigma \in Paths(s)$. Zu jedem $k \in \mathbb{N}$ es eine Berechnung t_k mit $trace(\sigma)_{\leq k} = trace(\tau_k)_{\leq k}$. Wir bauen nun aus den τ_k einen gemeinsamen Pfad δ, der Spur-äquivalent zu σ ist.

- Wir beginnen mit $\delta(0)=t$.
- Angenommen, $(\delta(0),\ldots,\delta(i))$ ist bereits konstruiert. Dann gilt $trace(\delta(0),\ldots,\delta(i)) = trace(\sigma)_{\leq i} = trace(\tau_k)_{\leq i}$ für alle $k \geq i$. Für unendlich viele der τ_k gilt also $trace(\tau_k)_{\leq i} = trace(\delta(0),\ldots,\delta(i))$.

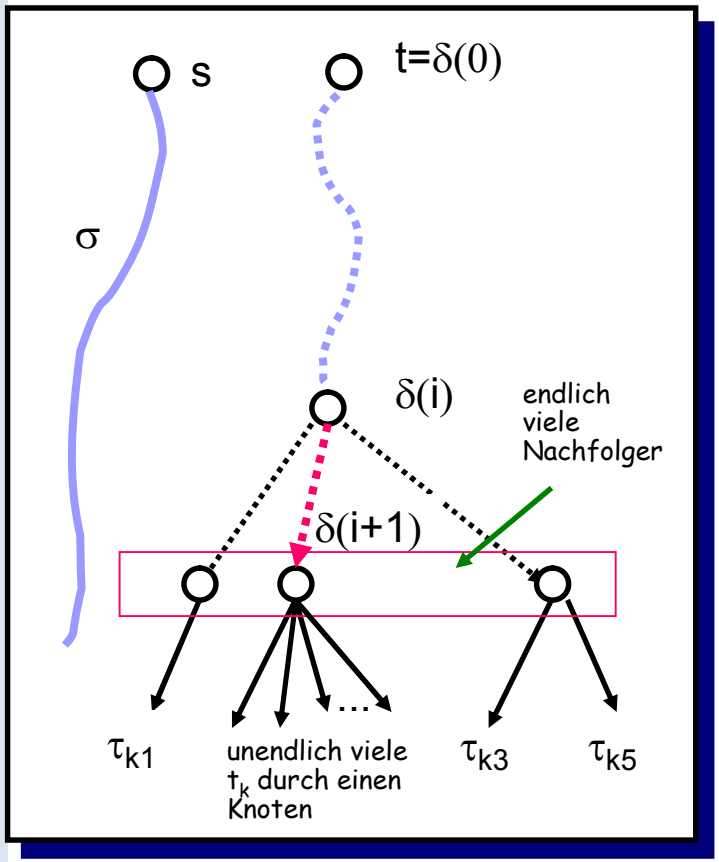
- Da $\delta(i)$ nur endlich viele Nachfolger hat, können wir einen davon auswählen, durch den immer noch unendlich viele der τ_k gehen. Dieser wird unser $\delta(i+1)$.

Auf diese Weise konstruieren wir einen unendlichen Pfad $\delta = (\delta(0),\delta(1),\ldots,\delta(n),\ldots)$ so dass für jedes $i \in \mathbb{N}$ gilt:

$trace(\delta)_{\leq i} = trace(\tau_k)_{\leq i}$ für unendlich viele k.

Es folgt, dass $trace(\delta)_{\leq i} = trace(\sigma)_{\leq i}$ für alle i gilt, also

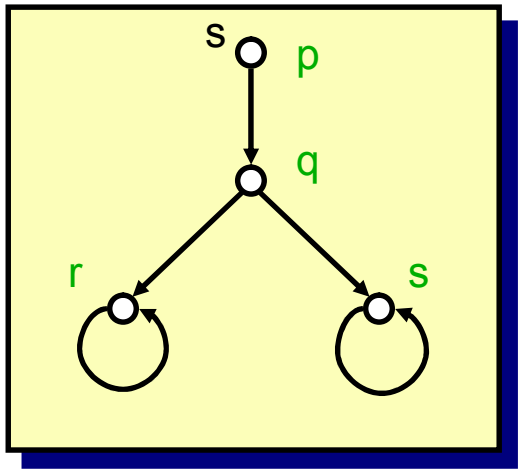
$trace(\delta) = trace(\sigma)$.



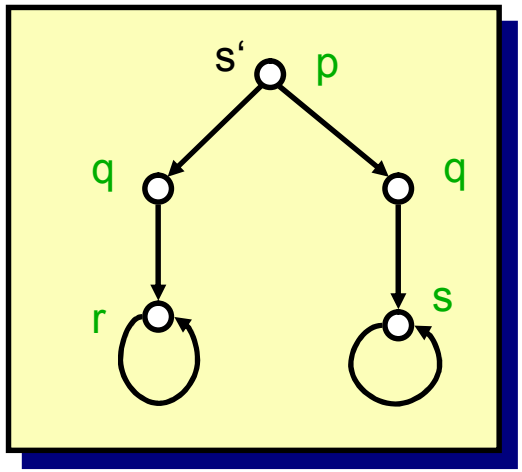
Ist Lineare Äquivalenz adäquat ?

- In den folgenden Systemen sind **s** und **s'** linear-äquivalent.
- Andererseits:
  - kein Nachfolger von **s** ist linear-äquivalent zu einem Nachfolger von **s'**
  - also doch ein merklicher Unterschied ...
- Das erste System erscheint irgendwie „nützlicher"
  - die Entscheidung, in welchen Ast man absteigt, kann aufgeschoben werden.
  - aber ... diesen Unterschied kann man in LTL offensichtlich nicht ausdrücken,

P :



Q :



$\mathbf{w_1} \equiv \mathbf{A}\,\mathbf{X}\,(\mathbf{A}\,(\,\mathbf{X}\,r \vee \mathbf{X}\,s)\,)$

$\mathbf{w_2} \equiv \mathbf{A}\,\mathbf{X}\,(\mathbf{A}\,\mathbf{X}\,r \vee \mathbf{A}\,\mathbf{X}\,s)\,)$

Weder w_1 noch w_2 sind LTL-Formeln. Wenn man aber beliebige Schachtelung der Pfad-Quantoren **A** und **E** zuläßt, dann trennt w_2 den Zustand s vom Zustand s'.

Aufgaben

Sei $\underline{\sigma}$ eine Zustandsfolge und k$\in$ Nat. Die Folge $\underline{\sigma}$' mit $\sigma'(n) = (\ \textbf{if}\ n \leq k\ \textbf{then}\ \underline{\sigma}(n)\ \textbf{else}\ \underline{\sigma}(n-1))$ entsteht aus $\underline{\sigma}$ durch Einfügung eines *Stotterschrittes*. Zwei Folgen $\underline{\sigma}$ und $\underline{\tau}$ sind *stotter-äquivalent* (in Zeichen $\underline{\sigma} \approx \underline{\tau}$), wenn sie durch Einfügung endlich vieler Stotter-schritte in die gleiche Folge überführt werden können.

1. a) Zeigen Sie: Sind $\underline{\sigma} \approx \underline{\tau}$, dann gilt für alle temporal-logischen Ausdrücke φ , in denen **X** nicht vorkommt : $\underline{\sigma} \models \varphi$ gdw. $\underline{\tau} \models \varphi$.
   b) Gilt auch die Umkehrung ?

2. Offensichtlich lassen sich **F** und **G** mithilfe von **X** und **U** ausdrücken.
   a) Zeigen Sie, daß sich **G**(φ **U** ψ) ohne Zuhilfenahme von **U** (oder **W**) ausdrücken läßt.
   b)* Läßt sich φ **U** ψ ohne Zuhilfenahme von **U** (oder **W**) ausdrücken ?

3. a) Ist es möglich, mit Hilfe von **X, F, G, U** einen temporal-logischen Ausdruck anzugeben, der besagt, daß p zu allen geraden Zeitpunkten wahr ist?
   b) Definieren Sie temporale Operatoren AllEven und SomeEven als Lösungen rekursiver gleichungen, so daß gilt :
   AllEven p gdw. p ist zu allen geraden Zeitpunkten wahr
   SomeEven p gdw. p ist an mindestens einem geraden Zeitpunkten wahr
   Begründen Sie, daß Ihre Definition korrekt ist !

Aufgabe: Single Pulser

- Spezifizieren Sie durch Formeln temporaler Logik das Verhalten des Single Pulser:
  - In jedem Intervall, in dem e gilt, gibt es genau einen Zeitpunkt zu dem a gilt
- Modellieren Sie den Single Pulser in NuSMV (mit einer booleschen Inputvariablen e) und zeigen Sie, dass das NuSMV-Modell die gewünschten Eigenschaften erfüllt

- Beispiel für korrektes Verhalten: