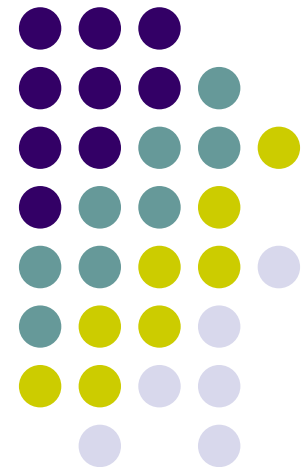


CTL / LTL

SE Systemanalyse

Lars Biermann



Gliederung



- 1. Motivation
- 2. Was ist CTL?
 - Syntax
 - Semantik
 - Beispiel
- 3. Was ist LTL?
 - Syntax
 - Semantik
 - Beispiel
- 4. Zusammenfassung
- 5. Literatur

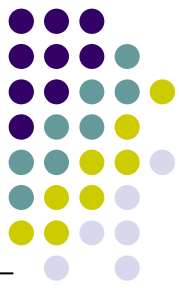


Modelchecking

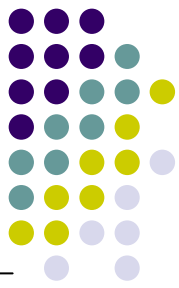
Modellierung -> Spezifikation -> **Verifikation**

- Abstraktion des physischen Systems
 - Modell M
- nachzuweisende formale Eigenschaften des Systems
 - Formel ϕ
- *Gilt* $M \models \phi$?
- ϕ = Formel in CTL bzw. LTL

Beispiel



- Internet-Protokoll:
 - Immer wenn eine Nachricht abgeschickt wird, wird sie irgendwann später den Empfänger erreichen
 - Es ist nie der Fall, dass eine Nachricht unterwegs ist, die nicht vorher vom Sender abgeschickt wurde
- Eigenschaften von Systemen beschreiben



Was ist CTL? (1)

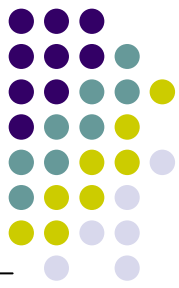
= Computational Tree Logic

- ist eine temporale Logik
 - Fundament: Aussagenlogik
 - formales System, das beschreibt wie sich der Wahrheitswert von Aussagen über die Zeit verändert
 - für nichtterminierende, nebenläufige Systeme
z.B. Betriebssysteme,
Kommunikationsprotokolle



Idee von CTL

- Ausdruck nicht statisch wahr oder falsch wie in Aussagen- oder Prädikatenlogik
 - Bsp.: $x \wedge y$
- dynamischer Wahrheitsbegriff:
 - mehrere Zustände mit unterschiedlichen Wahrheitswerten

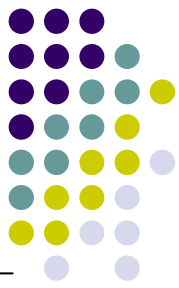


Modell

- $\mathcal{M} = (S, \rightarrow, L)$
 - S – Menge von Zuständen
 - \rightarrow – Übergangsrelation,
 $\forall s \in S, \exists s' \in S$ mit $s \rightarrow s'$
 - L – Markierungsfunktion
 $L : S \rightarrow \mathcal{P}(\text{Atome})$

Atome = nicht aus anderen Aussagen zusammengesetzt
- jedes Modell als gerichteter Graph darstellbar

Beispiel



- $\mathcal{M} = (S, \rightarrow, L)$
 - $S = \{s_0, s_1, s_2\}$
 - $\{s_0 \rightarrow s_1, s_0 \rightarrow s_2, s_1 \rightarrow s_0, s_1 \rightarrow s_2, s_2 \rightarrow s_2\}$
 - $L(s_0) = \{p, q\}$
 $L(s_1) = \{q, r\}$
 $L(s_2) = \{r\}$



Konstrukte

- Pfadquantoren: E, A
 - E – entlang mindestens eines Pfades („there exists one path“)
 - A – entlang aller Pfade („along all paths“)
- Temporale Operatoren: X, F, G, U
 - X – unmittelbar folgender Zustand („next xt state“)
 - F – ein irgendwann folgender Zustand („some future state“)
 - G – alle folgenden Zustände („globally“)
 - U – ϕ gilt ununterbrochen bis ψ eintritt („until“)



Syntax – CTL

- Sei \mathcal{P} = Menge atomarer Aussagen
- induktive Definition von CTL-Formeln:
$$\begin{aligned} \phi ::= & \top \mid \perp \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \\ & \mid (\phi \rightarrow \phi) \mid AX\phi \mid EX\phi \mid AF\phi \mid EF\phi \\ & \mid AG\phi \mid EG\phi \mid A[\phi U \phi] \mid E[\phi U \phi] \end{aligned}$$

wobei $p \in \mathcal{P}$

Semantik



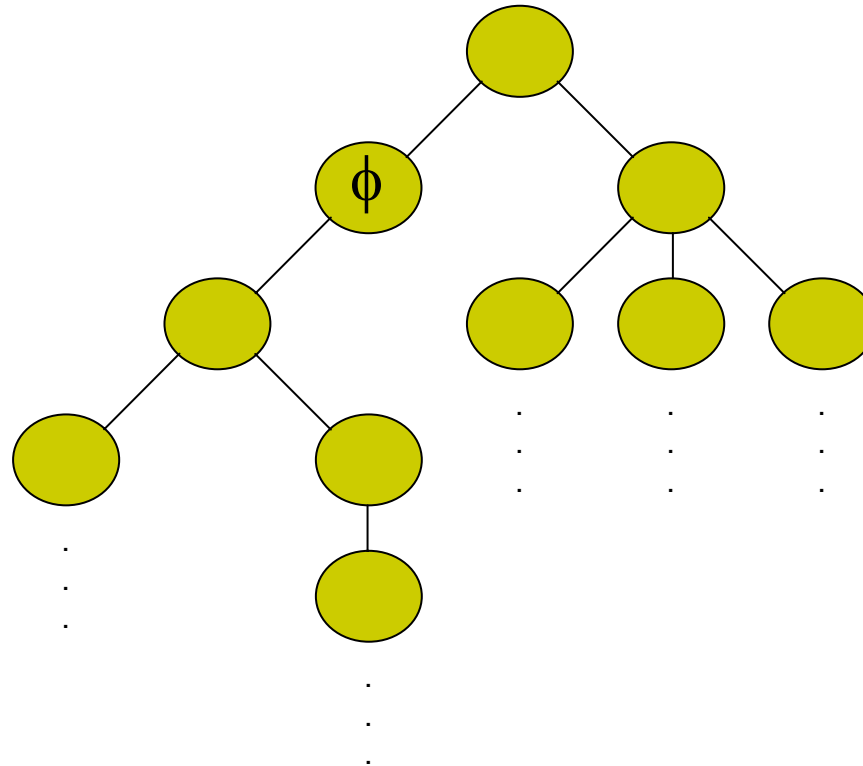
Erfüllungsrelation \models induktiv wie folgt definiert:

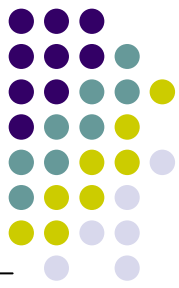
1. $\mathcal{M}, s \models \top$ und $\mathcal{M}, s \not\models \perp, \forall s \in S$
2. $\mathcal{M}, s \models p$ gdw. $p \in L(s)$
3. $\mathcal{M}, s \models \neg\phi$ gdw. $\mathcal{M}, s \not\models \phi$
4. $\mathcal{M}, s \models \phi_1 \wedge \phi_2$ gdw. $\mathcal{M}, s \models \phi_1$
und $\mathcal{M}, s \models \phi_2$
5. $\mathcal{M}, s \models \phi_1 \vee \phi_2$ gdw. $\mathcal{M}, s \models \phi_1$
oder $\mathcal{M}, s \models \phi_2$
6. $\mathcal{M}, s \models \phi_1 \rightarrow \phi_2$ gdw. $\mathcal{M}, s \not\models \phi_1$
oder $\mathcal{M}, s \models \phi_2$

Semantik - $EX\phi$



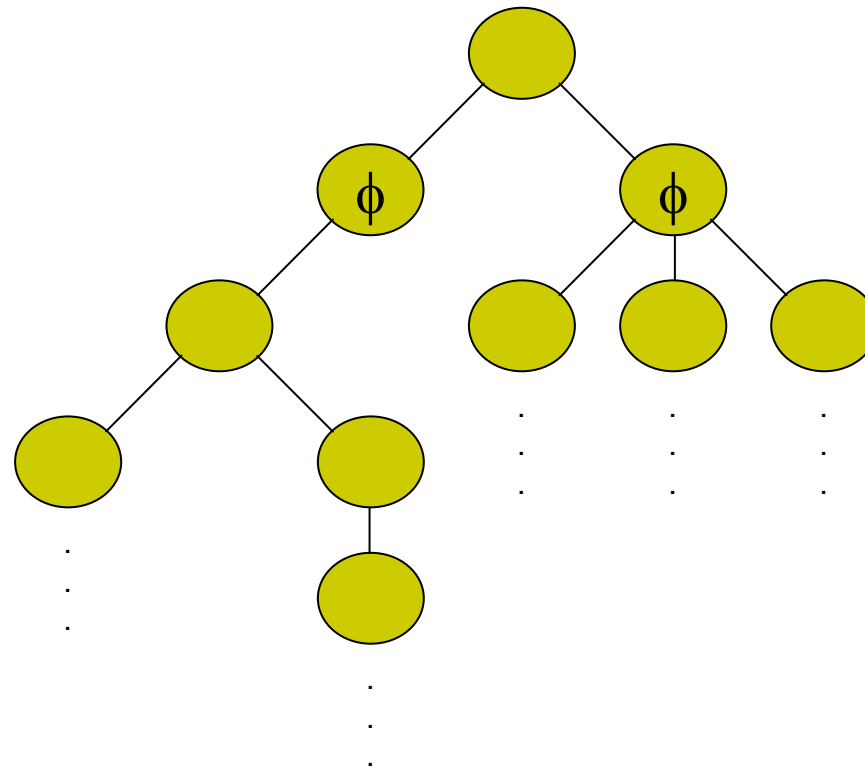
- System, dessen Startzustand $EX\phi$ erfüllt

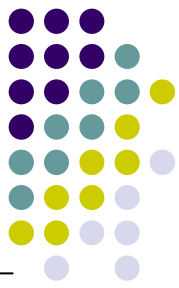




Semantik - $AX\phi$

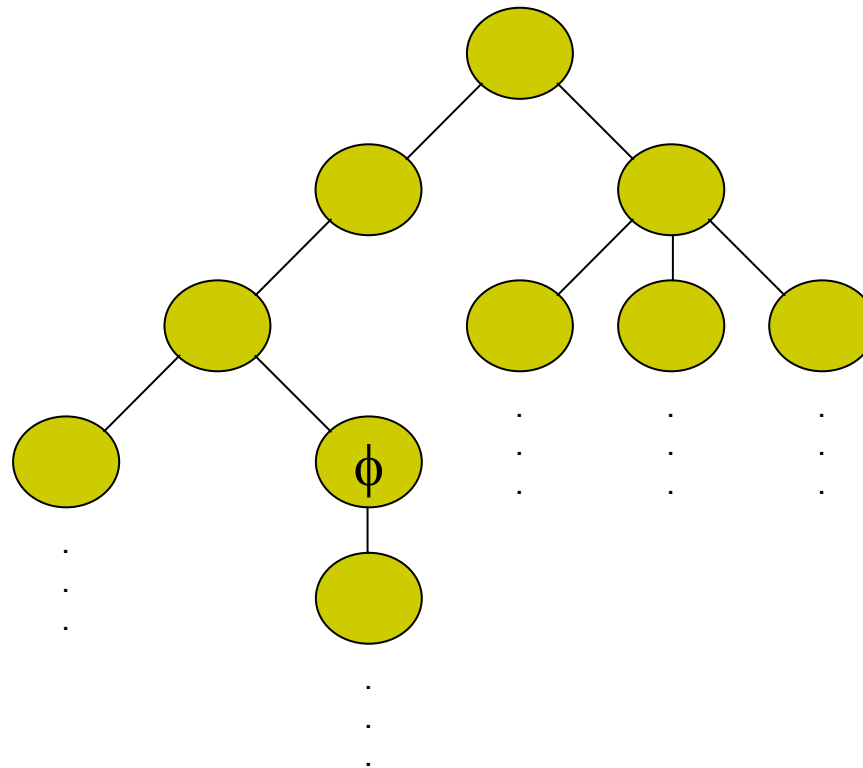
- System, dessen Startzustand $AX\phi$ erfüllt

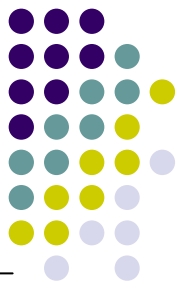




Semantik - $EF\phi$

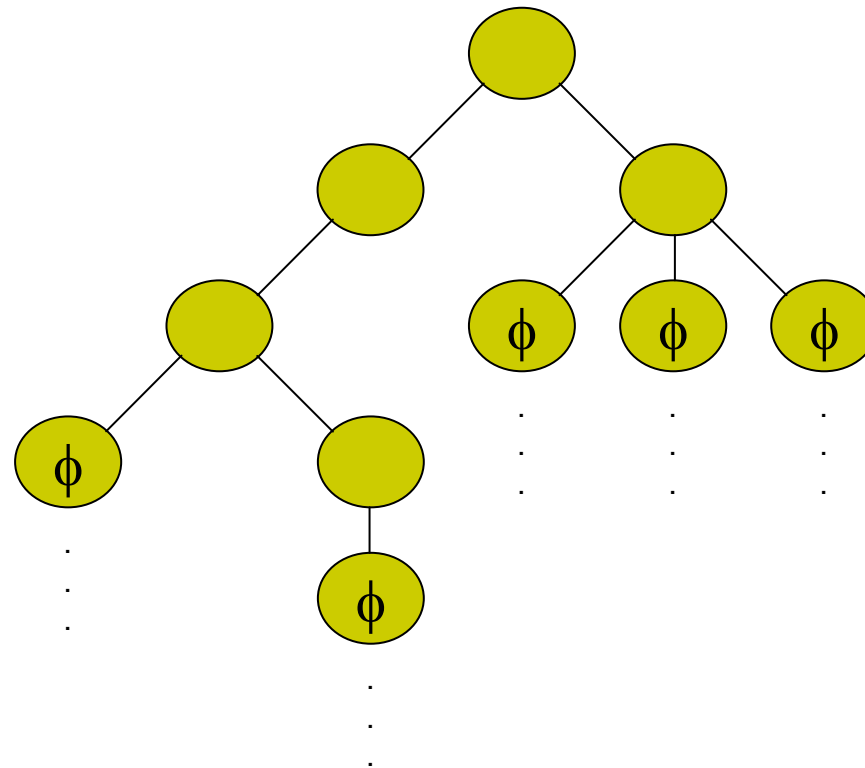
- System, dessen Startzustand $EF\phi$ erfüllt

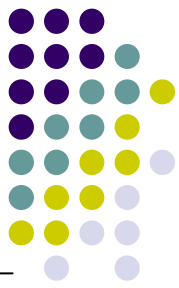




Semantik - $AF\phi$

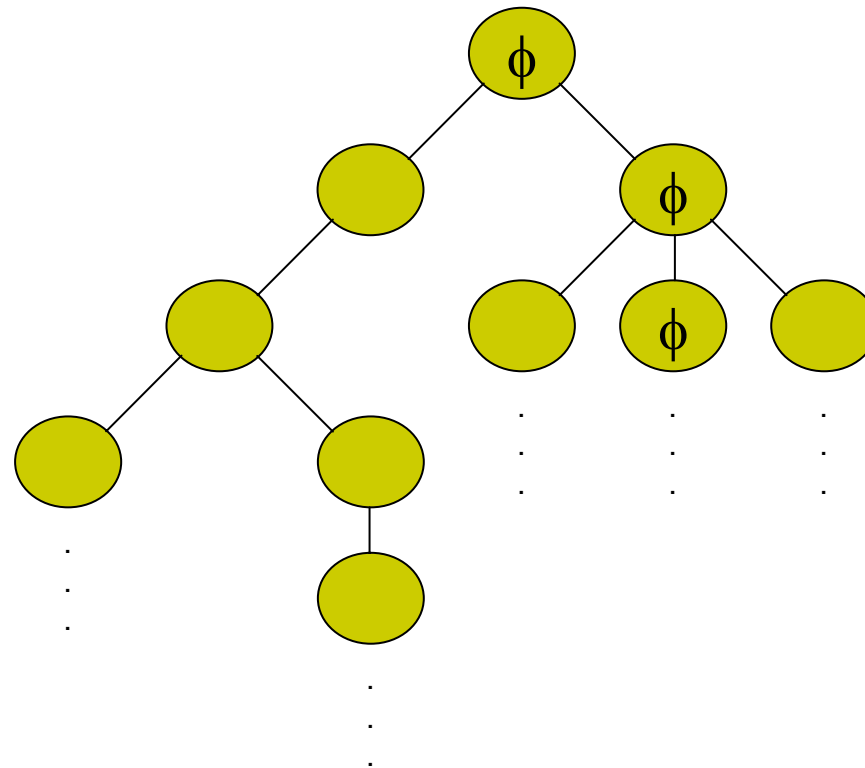
- System, dessen Startzustand $AF\phi$ erfüllt

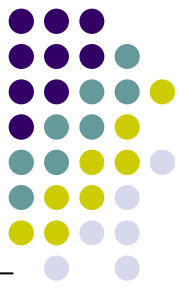




Semantik - $EG\phi$

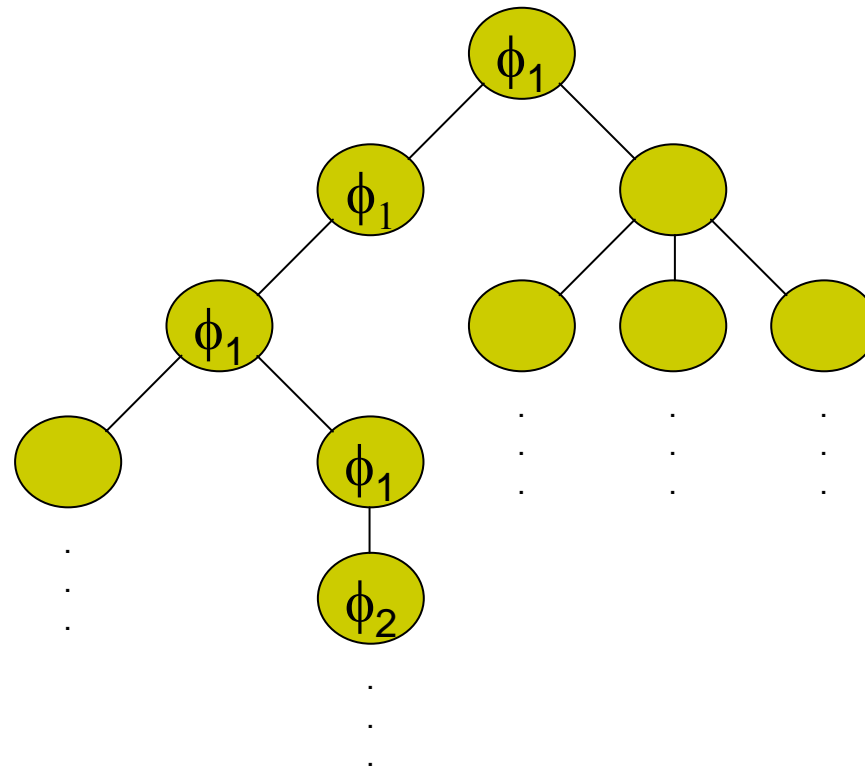
- System, dessen Startzustand $EG\phi$ erfüllt





Semantik – $E[\phi_1 U \phi_2]$

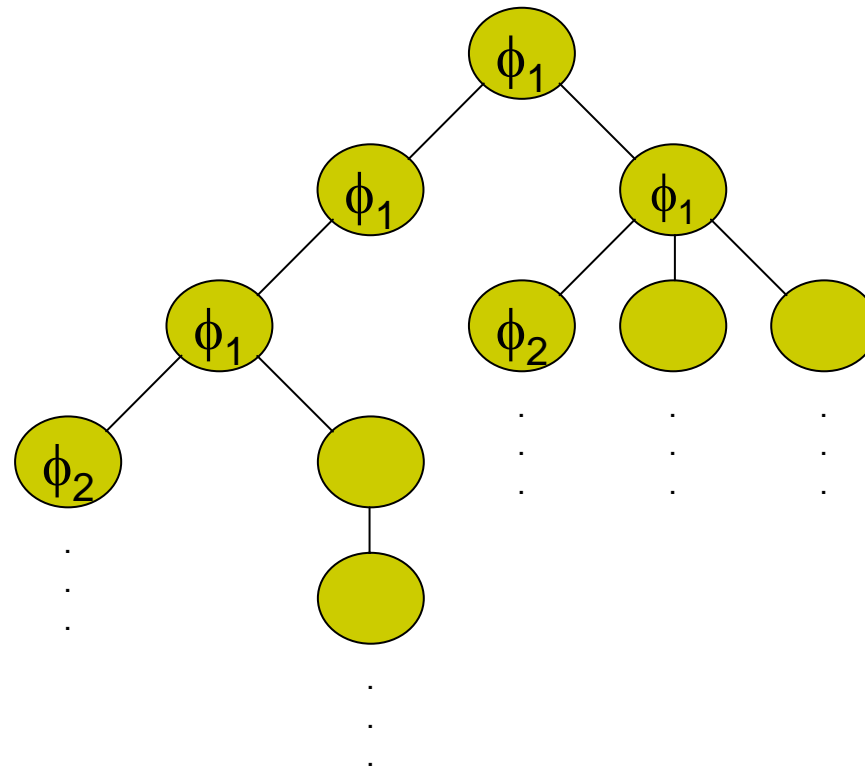
- System, dessen Startzustand $E[\phi_1 U \phi_2]$ erfüllt





Semantik – $A[\phi_1 U \phi_2]$

- System, dessen Startzustand $A[\phi_1 U \phi_2]$ erfüllt

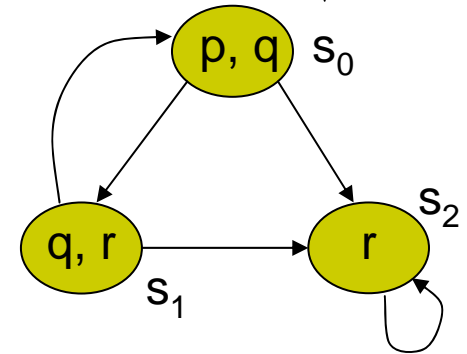


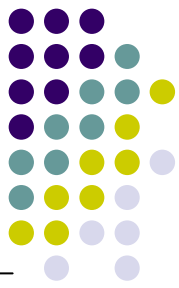
Beispiel (Forts.)



- Gilt?

- $\mathcal{M}, s_0 \models p \wedge q$
- $\mathcal{M}, s_0 \models \top$
- $\mathcal{M}, s_0 \models EX(q \wedge r)$
- $\mathcal{M}, s_0 \models AF r$
- $\mathcal{M}, s_2 \models AG r$
- $\mathcal{M}, s_0 \models A[p U r]$



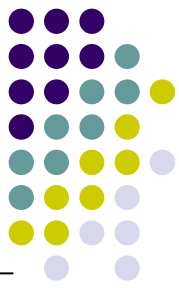


LTL

= Linear-Time Temporal Logic

- ähnlich CTL, aber:
 - Formeln haben nur Bedeutung auf einzelnen Pfaden
 - > keine Pfadquantoren E und A
 - man nimmt Bezug auf Pfade des Systems nicht auf Zustandsbaum

Syntax - LTL



- Sei \mathcal{P} = Menge atomarer Aussagen
- induktive Definition von LTL-Formeln:

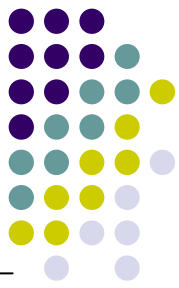
$$\begin{aligned} \phi ::= & p \mid (\neg\phi) \mid (\phi \wedge \phi) \\ & \mid X\phi \mid F\phi \mid G\phi \mid (\phi U \phi) \end{aligned}$$

wobei $p \in \mathcal{P}$



Semantik - LTL

- Formel wird über Pfad oder Menge von Pfaden ausgewertet
- Menge von Pfaden erfüllt ϕ , wenn jeder Pfad in der Menge ϕ erfüllt
 - System erfüllt ϕ in Zustand s , wenn jeder Pfad der in s beginnt ϕ erfüllt
- Pfad $\pi := s_1 \rightarrow s_2 \rightarrow \dots$
 $\pi^i := s_i \rightarrow s_{i+1} \rightarrow \dots$
- mit Erfüllungsrelation \models definieren wir wann Pfad π LTL-Formel ϕ erfüllt



Semantik - LTL (Forts.)

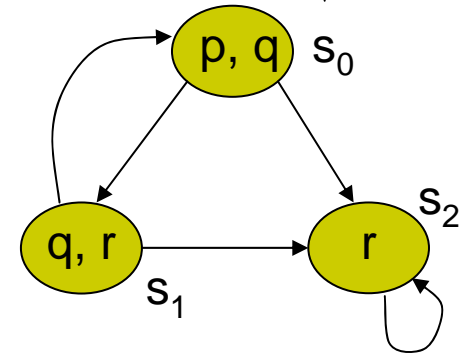
1. $\pi \models \top$
2. $\pi \models \neg\phi$ gdw. $\pi \not\models \phi$
3. $\pi \models p$ gdw. $p \in L(s_1)$
4. $\pi \models \phi_1 \wedge \phi_2$ gdw. $\pi \models \phi_1$ und $\pi \models \phi_2$
5. $\pi \models X\phi$ gdw. $\pi^2 \models \phi$
6. $\pi \models G\phi$ gdw. $\forall i \geq 1, \pi^i \models \phi$
7. $\pi \models F\phi$ gdw. $\exists i \geq 1, \pi^i \models \phi$
8. $\pi \models \phi U \psi$ gdw. $\exists i \geq 1$ mit $\pi^i \models \psi$
und $\forall j = 1, \dots, i - 1$ gilt $\pi^j \models \phi$

Beispiel (Forts.)



- Gilt?

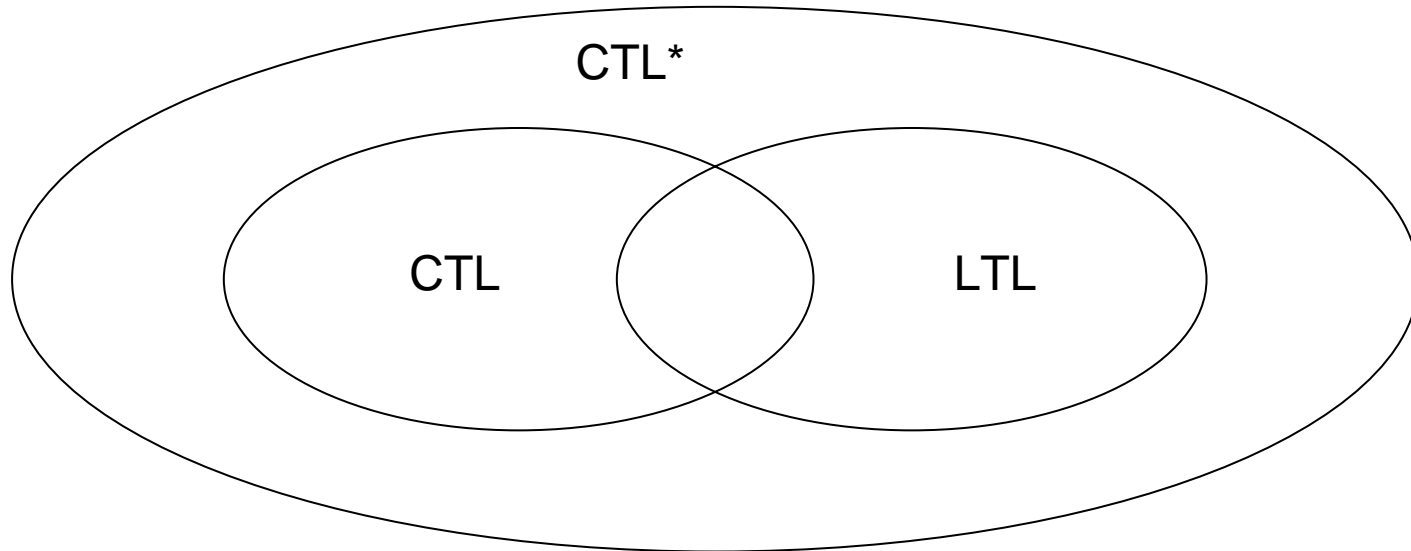
- $\pi \models p \wedge q$
- $\pi \models \top$
- $\pi \models X(q \wedge r)$
- $\pi \models F r$
- $\pi \models G r$
- $\pi \models p U r$



CTL: $\mathcal{M}, s_0 \models p \wedge q$
 $\mathcal{M}, s_0 \models \top$
 $\mathcal{M}, s_0 \models EX(q \wedge r)$
 $\mathcal{M}, s_0 \models AF r$
 $\mathcal{M}, s_2 \models AG r$
 $\mathcal{M}, s_0 \models A(p U r)$



- Obermenge von CTL und LTL





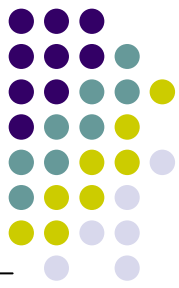
Abgrenzung CTL – LTL

- In CTL und nicht in LTL:
 - $AG\ EF\ p$
- In LTL und nicht in CTL:
 - $A[GF\ p \rightarrow F\ q]$ (bzw. $GF\ p \rightarrow F\ q$)
- Nicht in CTL und LTL
 - $E[GF\ p]$

Zusammenfassung



- CTL + LTL sind
 - wichtige Spezifikationsprachen für reaktive Systeme
 - wesentlich für Modelchecking
 - nur zwei unter vielen
 - Reguläre Ausdrücke, Zustandsdiagramme, Kripke-Strukturen, μ -Kalkül



Literatur

- Huth, M., Ryan, M.: Logic in Computer Science. Cambridge University Press, 2003.
- Emerson, E.A.: Temporal and Modal Logic. 1995.
- <http://www2.informatik.hu-berlin.de/~kschmidt/modelchecking/modelchecking.html>