Temporale Logiken

CTL und LTL

Anne Rubruck

Fachbereich Informatik
Universität Hamburg

Seminar FGI 3, WS13/14

Gliederung

Motivation

Verifikation durch Model Checking Algorithmen

- Modell M des Systems
- von M zu erfüllende Spezifikation ϕ
- M erfüllt ϕ, gdw. $M \models \phi$

Motivation

Verifikation durch Model Checking Algorithmen

- Modell M des Systems
- von M zu erfüllende Spezifikation ϕ
- M erfüllt ϕ, gdw. $M \models \phi$

Motivation

Verifikation durch Model Checking Algorithmen

- Modell M des Systems
- von M zu erfüllende Spezifikation ϕ
- M erfüllt ϕ, gdw. $M \models \phi$

Motivation

Verifikation durch Model Checking Algorithmen

- Modell M des Systems
- von M zu erfüllende Spezifikation ϕ
- M erfüllt ϕ, gdw. $M \models \phi$

Das Modell

Reaktive Systeme

- parallele Ausführung von Aufgaben
- Reagieren auf Benutzereingaben
- nicht Terminieren
- Beispiel?
- repräsentiert durch Modell M mit Zuständen $s \in S$ und Pfaden $s_1 \to s_2 \to \ldots$

Das Modell

Reaktive Systeme

- parallele Ausführung von Aufgaben
- Reagieren auf Benutzereingaben
- nicht Terminieren
- Beispiel?
- repräsentiert durch Modell M mit Zuständen $s \in S$ und Pfaden $s_1 \rightarrow s_2 \rightarrow \ldots$

Das Modell

Reaktive Systeme

- parallele Ausführung von Aufgaben
- Reagieren auf Benutzereingaben
- nicht Terminieren
- Beispiel?
- repräsentiert durch Modell M mit Zuständen $s \in S$ und Pfaden $s_1 \rightarrow s_2 \rightarrow \ldots$

Das Modell

Reaktive Systeme

- parallele Ausführung von Aufgaben
- Reagieren auf Benutzereingaben
- nicht Terminieren
- Beispiel?
- repräsentiert durch Modell M mit Zuständen $s \in S$ und Pfaden $s_1 \rightarrow s_2 \rightarrow \ldots$

Das Modell

Reaktive Systeme

- parallele Ausführung von Aufgaben
- Reagieren auf Benutzereingaben
- nicht Terminieren
- Beispiel?
- repräsentiert durch Modell M mit Zuständen $s \in S$ und Pfaden $s_1 \rightarrow s_2 \rightarrow \ldots$

Das Modell

Reaktive Systeme

- parallele Ausführung von Aufgaben
- Reagieren auf Benutzereingaben
- nicht Terminieren
- Beispiel?
- repräsentiert durch Modell M mit Zuständen $s \in S$ und Pfaden $s_1 \rightarrow s_2 \rightarrow \dots$

Temporale Logik

temporale Logiken

- erweitern klassische Aussagenlogik um temporale Operatoren
- Wahrheitsgehalt von Ausdrücken ϕ ändert sich dynamisch

Temporale Logik

temporale Logiken

- erweitern klassische Aussagenlogik um temporale Operatoren
- Wahrheitsgehalt von Ausdrücken ϕ ändert sich dynamisch

Temporale Logik

temporale Logiken

- erweitern klassische Aussagenlogik um temporale Operatoren
- Wahrheitsgehalt von Ausdrücken ϕ ändert sich dynamisch

Computation Tree Logic
Einleitung

- kurz: CTL
- E. Clarke und E. A. Emerson (1981)
- Verifikation von Hardware, Kommunikationsprotokollen und Software
- nicht deterministisch, verzweigend

Computation Tree Logic
Einleitung

- kurz: CTL
- E. Clarke und E. A. Emerson (1981)
- Verifikation von Hardware, Kommunikationsprotokollen und Software
- nicht deterministisch, verzweigend

Computation Tree Logic

Einleitung

- kurz: CTL
- E. Clarke und E. A. Emerson (1981)
- Verifikation von Hardware, Kommunikationsprotokollen und Software
- nicht deterministisch, verzweigend

Computation Tree Logic
Einleitung

- kurz: CTL
- E. Clarke und E. A. Emerson (1981)
- Verifikation von Hardware, Kommunikationsprotokollen und Software
- nicht deterministisch, verzweigend

Computation Tree Logic
Syntax

Definition der CTL-Syntax:

$$\phi ::= \bot \mid \top \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid AX\phi \mid EX\phi \mid$$

$$A[\phi U \phi] \mid E[\phi U \phi] \mid AG\phi \mid EG\phi \mid AF\phi \mid EF\phi$$

Computation Tree Logic
Temporale Konnektive

Pfadquantoren

- A: »along **a**ll paths«
- E: »there **e**xists one path«

Bedingungen

- X: »ne**x**t state«
- F: »some **f**uture state«
- G: »**g**lobally«
- U: »**u**ntil«

X, F, G und U müssen immer A oder E folgen!

Computation Tree Logic

Temporale Konnektive

Pfadquantoren

- A: »along **a**ll paths«
- E: »there **e**xists one path«

Bedingungen

- X: »ne**x**t state«
- F: »some **f**uture state«
- G: »**g**lobally«
- U: »**u**ntil«

X, F, G und U müssen immer A oder E folgen!

Computation Tree Logic
Temporale Konnektive

Pfadquantoren

- A: »along **a**ll paths«
- E: »there **e**xists one path«

Bedingungen

- X: »ne**x**t state«
- F: »some **f**uture state«
- G: »**g**lobally«
- U: »**u**ntil«

X, F, G und U müssen immer A oder E folgen!

Computation Tree Logic

Bindungsstärke

Bindungsstärke absteigend:

$$[\neg, AG, EG, AF, EF, AX, EX], [\wedge, \vee], [\rightarrow, AU, EU]$$

Computation Tree Logic
Quiz: Sind dies korrekte Ausdrücke in CTL?

CTL-Syntax:

$$\phi ::= \bot \mid \top \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid AX\phi \mid EX\phi \mid$$

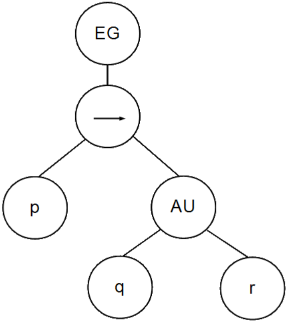
$$A[\phi U \phi] \mid E[\phi U \phi] \mid AG\phi \mid EG\phi \mid AF\phi \mid EF\phi$$

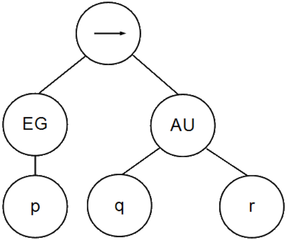
Bindungsstärke:

$$[\neg, AG, EG, AF, EF, AX, EX], [\wedge, \vee], [\rightarrow, AU, EU]$$

Computation Tree Logic
Herleitungsbäume

Welcher Herleitungsbaum gehört zu $EGp \rightarrow A[qUr]$ und welcher zu $EG(p \rightarrow A[qUr])$?





Computation Tree Logic

Kripke-Modell

Modell $M = (S, \rightarrow, L)$

- Zustände $s \in S$
- Relation $\rightarrow$, welche jedem Zustand $s \in S$ mindestens einen Folgezustand $s' \in S$ zuordnet
- $\forall s \in S$ eine Menge $L(s)$ von atomaren Termen p, welche in Zustand s wahr sind
- Das System, bzw. Modell M, erfüllt eine Spezifikation ϕ, falls ϕ für alle Zustände $s \in S$ wahr ist. Schreibweise: $M, s \models \phi$

Computation Tree Logic

Kripke-Modell

Modell $M = (S, \rightarrow, L)$

- Zustände $s \in S$
- Relation $\rightarrow$, welche jedem Zustand $s \in S$ mindestens einen Folgezustand $s' \in S$ zuordnet
- $\forall s \in S$ eine Menge $L(s)$ von atomaren Termen p, welche in Zustand s wahr sind
- Das System, bzw. Modell M, erfüllt eine Spezifikation ϕ, falls ϕ für alle Zustände $s \in S$ wahr ist. Schreibweise: $M, s \models \phi$

Computation Tree Logic

Kripke-Modell

Modell $M = (S, \rightarrow, L)$

- Zustände $s \in S$
- Relation $\rightarrow$, welche jedem Zustand $s \in S$ mindestens einen Folgezustand $s' \in S$ zuordnet
- $\forall s \in S$ eine Menge $L(s)$ von atomaren Termen p, welche in Zustand s wahr sind
- Das System, bzw. Modell M, erfüllt eine Spezifikation ϕ, falls ϕ für alle Zustände $s \in S$ wahr ist. Schreibweise: $M, s \models \phi$

Computation Tree Logic

Kripke-Modell

Modell $M = (S, \rightarrow, L)$

- Zustände $s \in S$
- Relation $\rightarrow$, welche jedem Zustand $s \in S$ mindestens einen Folgezustand $s' \in S$ zuordnet
- $\forall s \in S$ eine Menge $L(s)$ von atomaren Termen p, welche in Zustand s wahr sind
- Das System, bzw. Modell M, erfüllt eine Spezifikation ϕ, falls ϕ für alle Zustände $s \in S$ wahr ist. Schreibweise: $M, s \models \phi$

Computation Tree Logic

Kripke-Modell

Modell $M = (S, \rightarrow, L)$

- Zustände $s \in S$
- Relation $\rightarrow$, welche jedem Zustand $s \in S$ mindestens einen Folgezustand $s' \in S$ zuordnet
- $\forall s \in S$ eine Menge $L(s)$ von atomaren Termen p, welche in Zustand s wahr sind
- Das System, bzw. Modell M, erfüllt eine Spezifikation ϕ, falls ϕ für alle Zustände $s \in S$ wahr ist. Schreibweise: $M, s \models \phi$

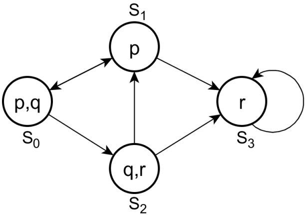
Computation Tree Logic

Beispiel eines Kripke-Modells

$M = (S, \rightarrow, L)$, mit:

$s_0 \rightarrow s_1$, $s_0 \rightarrow s_2$, $s_1 \rightarrow s_0$, $s_1 \rightarrow s_3$, $s_2 \rightarrow s_1$, $s_2 \rightarrow s_3$, $s_3 \rightarrow s_3$

$L(s_0) = \{p, q\}$, $L(s_1) = \{p\}$, $L(s_2) = \{q, r\}$, $L(s_3) = \{r\}$



Computation Tree Logic
Semantik

- $M, s \models \top$ und $M, s \not\models \bot$ $\forall s \in S$
- $M, s \models p$ falls $p \in L(s)$
- $M, s \models \neg\phi$ falls $M, s \not\models \phi$
- $M, s \models \phi_1 \land \phi_2$ falls $M, s \models \phi_1$ und $M, s \models \phi_2$
- $M, s \models \phi_1 \lor \phi_2$ falls $M, s \models \phi_1$ oder $M, s \models \phi_2$

Computation Tree Logic
Semantik

- $M, s \models \top$ und $M, s \not\models \bot \ \forall s \in S$
- $M, s \models p$ falls $p \in L(s)$
- $M, s \models \neg\phi$ falls $M, s \not\models \phi$
- $M, s \models \phi_1 \wedge \phi_2$ falls $M, s \models \phi_1$ und $M, s \models \phi_2$
- $M, s \models \phi_1 \vee \phi_2$ falls $M, s \models \phi_1$ oder $M, s \models \phi_2$

Computation Tree Logic
Semantik

- $M, s \models \top$ und $M, s \not\models \bot \ \forall s \in S$
- $M, s \models p$ falls $p \in L(s)$
- $M, s \models \neg\phi$ falls $M, s \not\models \phi$
- $M, s \models \phi_1 \wedge \phi_2$ falls $M, s \models \phi_1$ und $M, s \models \phi_2$
- $M, s \models \phi_1 \vee \phi_2$ falls $M, s \models \phi_1$ oder $M, s \models \phi_2$

Computation Tree Logic
Semantik

- $M, s \models \top$ und $M, s \not\models \bot \ \forall s \in S$
- $M, s \models p$ falls $p \in L(s)$
- $M, s \models \neg\phi$ falls $M, s \not\models \phi$
- $M, s \models \phi_1 \wedge \phi_2$ falls $M, s \models \phi_1$ und $M, s \models \phi_2$
- $M, s \models \phi_1 \vee \phi_2$ falls $M, s \models \phi_1$ oder $M, s \models \phi_2$

Computation Tree Logic

Semantik

- $M, s \models \top$ und $M, s \not\models \bot \; \forall s \in S$
- $M, s \models p$ falls $p \in L(s)$
- $M, s \models \neg\phi$ falls $M, s \not\models \phi$
- $M, s \models \phi_1 \wedge \phi_2$ falls $M, s \models \phi_1$ und $M, s \models \phi_2$
- $M, s \models \phi_1 \vee \phi_2$ falls $M, s \models \phi_1$ oder $M, s \models \phi_2$

Computation Tree Logic
Semantik

- $M, s \models \phi_1 \rightarrow \phi_2$ falls $M, s \not\models \phi_1$ oder $M, s \models \phi_2$
- $M, s \models AX\phi$ falls $\forall s', s \rightarrow s'$ gilt $M, s' \models \phi$
  (in jedem Folgezustand)
- $M, s \models EX\phi$ falls $\exists s', s \rightarrow s'$, mit $M, s' \models \phi$
  (in einem Folgezustand)

Computation Tree Logic

Semantik

- $M, s \models \phi_1 \to \phi_2$ falls $M, s \not\models \phi_1$ oder $M, s \models \phi_2$
- $M, s \models AX\phi$ falls $\forall s', s \to s'$ gilt $M, s' \models \phi$
  (in jedem Folgezustand)
- $M, s \models EX\phi$ falls $\exists s', s \to s'$, mit $M, s' \models \phi$
  (in einem Folgezustand)

Computation Tree Logic
Semantik

- $M, s \models \phi_1 \rightarrow \phi_2$ falls $M, s \not\models \phi_1$ oder $M, s \models \phi_2$
- $M, s \models AX\phi$ falls $\forall s', s \rightarrow s'$ gilt $M, s' \models \phi$
(in jedem Folgezustand)
- $M, s \models EX\phi$ falls $\exists s', s \rightarrow s'$, mit $M, s' \models \phi$
(in einem Folgezustand)

Computation Tree Logic
Semantik

- $M, s \models AG\phi$ falls $\forall$ Pfade $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ und $\forall s_i$ entlang des Pfades gilt $M, s_i \models \phi$
- $M, s \models EG\phi$ falls $\exists$ Pfad $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ und $\forall s_i$ entlang des Pfades gilt $M, s_i \models \phi$

Computation Tree Logic
Semantik

- $M, s \models AG\phi$ falls $\forall$ Pfade $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ und $\forall s_i$ entlang des Pfades gilt $M, s_i \models \phi$
- $M, s \models EG\phi$ falls $\exists$ Pfad $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ und $\forall s_i$ entlang des Pfades gilt $M, s_i \models \phi$

Computation Tree Logic
Semantik

- $M, s \models AF\phi$ falls $\forall$ Pfade $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ $\exists s_i$ entlang des Pfades, mit $M, s_i \models \phi$
- $M, s \models EF\phi$ falls $\exists$ Pfad $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ und $\exists s_i$ entlang des Pfades, mit $M, s_i \models \phi$

Computation Tree Logic
Semantik

- $M, s \models AF\phi$ falls $\forall$ Pfade $s \to s' \to s'' \to \ldots$ $\exists s_i$ entlang des Pfades, mit $M, s_i \models \phi$
- $M, s \models EF\phi$ falls $\exists$ Pfad $s \to s' \to s'' \to \ldots$ und $\exists s_i$ entlang des Pfades, mit $M, s_i \models \phi$

Computation Tree Logic

Semantik

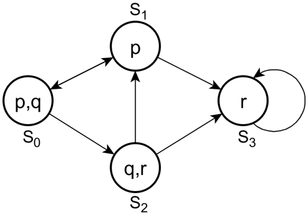
- $M, s \models A[\phi_1 U \phi_2]$ falls $\forall$ Pfade $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ gilt $\phi_1 U \phi_2$, d.h. $\exists s_i$, $M, s_i \models \phi_2$ und $\forall j < i$ gilt $M, s_j \models \phi_1$
  (ϕ_1 ist wahr entlang eines Pfades für dessen letzten Zustand ϕ_2 wahr ist)
- $M, s \models E[\phi_1 U \phi_2]$ falls $\exists$ Pfad $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ gilt $\phi_1 U \phi_2$, d.h. $\exists s_i$, $M, s_i \models \phi_2$ und $\forall j < i$ gilt $M, s_j \models \phi_1$

Computation Tree Logic
Semantik

- $M, s \models A[\phi_1 U \phi_2]$ falls $\forall$ Pfade $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ gilt $\phi_1 U \phi_2$, d.h. $\exists s_i$, $M, s_i \models \phi_2$ und $\forall j < i$ gilt $M, s_j \models \phi_1$
  (ϕ_1 ist wahr entlang eines Pfades für dessen letzten Zustand ϕ_2 wahr ist)
- $M, s \models E[\phi_1 U \phi_2]$ falls $\exists$ Pfad $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ gilt $\phi_1 U \phi_2$, d.h. $\exists s_i$, $M, s_i \models \phi_2$ und $\forall j < i$ gilt $M, s_j \models \phi_1$

Computation Tree Logic

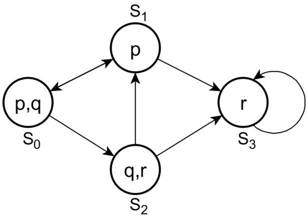
Quiz: Erfüllt Modell M die Verifikation ϕ?



- $M, s \models \top$ und $M, s \not\models \bot \ \forall s \in S$
- $M, s \models p$ falls $p \in L(s)$
- $M, s \models \neg\phi$ falls $M, s \not\models \phi$
- $M, s \models \phi_1 \wedge \phi_2$ falls $M, s \models \phi_1$ und $M, s \models \phi_2$
- $M, s \models \phi_1 \vee \phi_2$ falls $M, s \models \phi_1$ oder $M, s \models \phi_2$

Computation Tree Logic

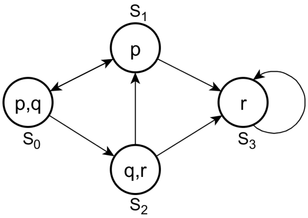
Quiz: Erfüllt Modell M die Verifikation ϕ?



- $M, s \models \phi_1 \rightarrow \phi_2$ falls $M, s \not\models \phi_1$ oder $M, s \models \phi_2$
- $M, s \models AX\phi$ falls $\forall s', s \rightarrow s'$ gilt $M, s' \models \phi$ (in jedem Folgezustand)
- $M, s \models EX\phi$ falls $\exists s', s \rightarrow s'$, mit $M, s' \models \phi$ (in einem Folgezustand)

Computation Tree Logic

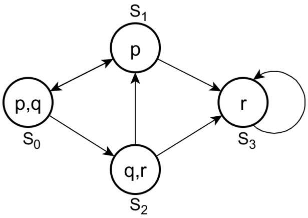
Quiz: Erfüllt Modell M die Verifikation ϕ?



- $M, s \models AG\phi$ falls $\forall$ Pfade $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ und $\forall s_i$ entlang des Pfades gilt $M, s_i \models \phi$
- $M, s \models EG\phi$ falls $\exists$ Pfad $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ und $\forall s_i$ entlang des Pfades gilt $M, s_i \models \phi$

Computation Tree Logic

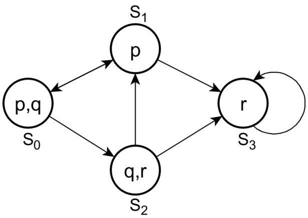
Quiz: Erfüllt Modell M die Verifikation ϕ?



- $M, s \models AF\phi$ falls $\forall$ Pfade $s \rightarrow s' \rightarrow s'' \rightarrow \ldots \exists s_i$ entlang des Pfades, mit $M, s_i \models \phi$
- $M, s \models EF\phi$ falls $\exists$ Pfad $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ und $\exists s_i$ entlang des Pfades, mit $M, s_i \models \phi$

Computation Tree Logic

Quiz: Erfüllt Modell M die Verifikation ϕ?



- $M, s \models A[\phi_1 U \phi_2]$ falls $\forall$ Pfade $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ gilt $\phi_1 U \phi_2$, d.h. $\exists s_i$, $M, s_i \models \phi_2$ und $\forall j < i$ gilt $M, s_j \models \phi_1$ (ϕ_1 ist wahr entlang eines Pfades für dessen letzten Zustand ϕ_2 wahr ist)
- $M, s \models E[\phi_1 U \phi_2]$ falls $\exists$ Pfad $s \rightarrow s' \rightarrow s'' \rightarrow \ldots$ gilt $\phi_1 U \phi_2$, d.h. $\exists s_i$, $M, s_i \models \phi_2$ und $\forall j < i$ gilt $M, s_j \models \phi_1$

Linear-Time Temporal Logic
Einleitung

- kurz: LTL
- A. Pnueli (1977)
- Validierung anhand einzelner Pfade oder einer Menge von Pfaden

Linear-Time Temporal Logic
Syntax

Definition der LTL-Syntax:

$$\phi ::= \bot \mid \top \mid p \mid (\neg \phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi U \phi) \mid (G \phi) \mid (F \phi) \mid (X \phi)$$

Linear-Time Temporal Logic
Semantik

- $\pi \models \top$
- $\pi \models p$ falls $p \in L(s_1)$
- $\pi \models \neg\phi$ falls $\pi \not\models \phi$
- $\pi \models \phi_1 \wedge \phi_2$ falls $\pi \models \phi_1$ und $\pi \models \phi_2$
- $\pi \models \phi_1 \vee \phi_2$ falls $\pi \models \phi_1$ oder $\pi \models \phi_2$

Linear-Time Temporal Logic

Semantik

- $\pi \models \top$
- $\pi \models p$ falls $p \in L(s_1)$
- $\pi \models \neg\phi$ falls $\pi \not\models \phi$
- $\pi \models \phi_1 \wedge \phi_2$ falls $\pi \models \phi_1$ und $\pi \models \phi_2$
- $\pi \models \phi_1 \vee \phi_2$ falls $\pi \models \phi_1$ oder $\pi \models \phi_2$

Linear-Time Temporal Logic

Semantik

- $\pi \models \top$
- $\pi \models p$ falls $p \in L(s_1)$
- $\pi \models \neg\phi$ falls $\pi \not\models \phi$
- $\pi \models \phi_1 \wedge \phi_2$ falls $\pi \models \phi_1$ und $\pi \models \phi_2$
- $\pi \models \phi_1 \vee \phi_2$ falls $\pi \models \phi_1$ oder $\pi \models \phi_2$

Linear-Time Temporal Logic
Semantik

- $\pi \models \top$
- $\pi \models p$ falls $p \in L(s_1)$
- $\pi \models \neg\phi$ falls $\pi \not\models \phi$
- $\pi \models \phi_1 \wedge \phi_2$ falls $\pi \models \phi_1$ und $\pi \models \phi_2$
- $\pi \models \phi_1 \vee \phi_2$ falls $\pi \models \phi_1$ oder $\pi \models \phi_2$

Linear-Time Temporal Logic
Semantik

- $\pi \models \top$
- $\pi \models p$ falls $p \in L(s_1)$
- $\pi \models \neg\phi$ falls $\pi \not\models \phi$
- $\pi \models \phi_1 \wedge \phi_2$ falls $\pi \models \phi_1$ und $\pi \models \phi_2$
- $\pi \models \phi_1 \vee \phi_2$ falls $\pi \models \phi_1$ oder $\pi \models \phi_2$

Linear-Time Temporal Logic
Semantik

- $\pi \models X\phi$ falls $\pi^2 \models \phi$
- $\pi \models G\phi$ falls $\forall i \geq 1 : \pi^i \models \phi$
- $\pi \models F\phi$ falls $\exists i \geq 1 : \pi^i \models \phi$
- $\pi \models \phi_1 U \phi_2$ falls $\exists i \geq 1 : \pi^i \models \phi_2$ und $\forall j = 1, \ldots, i-1 : \pi^j \models \phi_1$

Linear-Time Temporal Logic
Semantik

- $\pi \models X\phi$ falls $\pi^2 \models \phi$
- $\pi \models G\phi$ falls $\forall i \geq 1 : \pi^i \models \phi$
- $\pi \models F\phi$ falls $\exists i \geq 1 : \pi^i \models \phi$
- $\pi \models \phi_1 U \phi_2$ falls $\exists i \geq 1 : \pi^i \models \phi_2$ und $\forall j = 1, \ldots, i-1 : \pi^j \models \phi_1$

Linear-Time Temporal Logic

Semantik

- $\pi \models X\phi$ falls $\pi^2 \models \phi$
- $\pi \models G\phi$ falls $\forall i \geq 1 : \pi^i \models \phi$
- $\pi \models F\phi$ falls $\exists i \geq 1 : \pi^i \models \phi$
- $\pi \models \phi_1 U \phi_2$ falls $\exists i \geq 1 : \pi^i \models \phi_2$ und $\forall j = 1, \ldots, i-1 : \pi^j \models \phi_1$

Linear-Time Temporal Logic

Semantik

- $\pi \models X\phi$ falls $\pi^2 \models \phi$
- $\pi \models G\phi$ falls $\forall i \geq 1 : \pi^i \models \phi$
- $\pi \models F\phi$ falls $\exists i \geq 1 : \pi^i \models \phi$
- $\pi \models \phi_1 U \phi_2$ falls $\exists i \geq 1 : \pi^i \models \phi_2$ und $\forall j = 1, \ldots, i-1 : \pi^j \models \phi_1$

Linear-Time Temporal Logic

Semantik

Verifikation

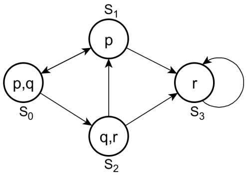
$M, s_i \models \phi$ falls $\forall \pi^i$ gilt: $\pi^i \models \phi$

Linear-Time Temporal Logic

Quiz: Gilt $M, s_i \models \phi$?

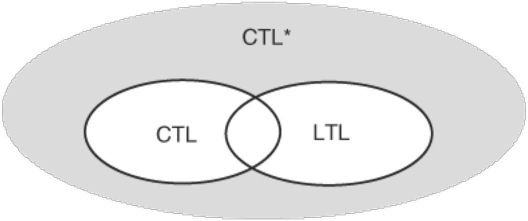
Verifikation

$M, s_i \models \phi$ falls $\forall \pi^i$ gilt: $\pi^i \models \phi$



- $\pi \models X\phi$ falls $\pi^2 \models \phi$
- $\pi \models G\phi$ falls $\forall i \geq 1 : \pi^i \models \phi$
- $\pi \models F\phi$ falls $\exists i \geq 1 : \pi^i \models \phi$
- $\pi \models \phi_1 U \phi_2$ falls $\exists i \geq 1 : \pi^i \models \phi_2$ und $\forall j = 1, \ldots, i-1 : \pi^j \models \phi_1$

Zusammenfassung



Quellen und weiterführende Literatur

M. Huth, M. Ryan.
Logic in Computer Science - Modelling and Reasoning about Systems.
Cambridge, 2004.

E. A. Emerson.
Temporal and Modal Logic.
Handbook of Theoretical Computer Science. 16, the MIT Press, 1990

S. Kripke.
Semantical Considerations On Modal Logic.
Acta Philosophica Fennica. 16:83–94, 1963.