

Seminar *Diskrete und hybride dynamische Systeme*

TEMPORALE LOGIK: EINE EINFÜHRUNG

Armin M. Warda

Gliederung:

- Einleitung & Grundlagen
- Temporale Logik
- Computation Tree Logic — CTL
- Temporal Logic of Actions — TLA
- Echtzeit-Logik
- Zusammenfassung

Ziele:

1. T.L. zur
Spezifikation
Beschreibung
2. Grundlagen
für Verifikation
→ U. Aldenhoff

Temporale Logik:

- deskriptive Modellierungstechnik
- für dynamische ereignisdiskrete Systemen zur
 - Spezifikation
 - Verifikation
- es gibt verschiedene Semantiken:
 - Branching–Time
 - Linear–Time
- \exists auch quantitative temporale Logiken: Zeitbewertung, „Real–Time“
- verschiedene Ansätze zur (automatischen) Verifikation:
 - Model–Checking (→ Ulrich Aldenhoff)
 - Theorem–Beweiser (→ Ulrich Aldenhoff)

- **Aussagenlogik:** Aussagen

= Satz, für den es Sinn macht zu fragen, ob er wahr oder falsch ist.

„*Borussia Dortmund ist Deutscher Fußballmeister 1995/96.*“

- Aussagenvariablen
- logischen Operatoren $\wedge, \vee, \neg, \Rightarrow, \iff, \dots$
- Erfüllbarkeit, Unerfüllbarkeit, Allgemeingültigkeit

- **Prädikatenlogik:** Prädikate

= Abbildung $P : M \rightarrow \{\text{falsch, wahr}\}$

$P(n, m) := (n \text{ teilt } m), \quad \exists n : \forall m : P(n, m)$

- auch Quantoren: $\exists x : P(x), \forall x : P(x)$
- gebundene und freie Variablen
- Erfüllbarkeit, Unerfüllbarkeit, Gültigkeit

Syntax:

1. **Alphabet:** $\Sigma = \{\neg, \Rightarrow, (,), p_1, p_2, p_3, \dots\}$
2. **Menge wohlgeformter Formeln \mathcal{WFF} :**
 - (a) $p_i \in \mathcal{WFF}$
 - (b) $(\neg A) \in \mathcal{WFF}$, falls $A \in \mathcal{WFF}$
 - (c) $(A \Rightarrow B) \in \mathcal{WFF}$, falls $A, B \in \mathcal{WFF}$
3. **Axiome:** Für beliebige $A, B, C \in \mathcal{WFF}$:
 - (L1) $(A \Rightarrow (B \Rightarrow A))$
 - (L2) $((A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)))$
 - (L3) $((\neg A) \Rightarrow (\neg B)) \Rightarrow (B \Rightarrow A)$
4. **Ableitungsregel:** Für beliebige $A, B \in \mathcal{WFF}$:
 - (MP) $A, A \Rightarrow B \vdash B$ „Modus Ponens“

Sei $\Gamma \subset \mathcal{WFF}$.

Ableitung / Beweis = Sequenz von \mathcal{WFF} s A_1, A_2, \dots, A_n ,

wenn für alle $i = 1, \dots, n$ gilt:

1. $A_i \in \Gamma$, oder
2. A_i Axiom, oder
3. $A_j, A_k \vdash A_i$ mit $j, k < i$.

$\Gamma \vdash A_n$ ist ein *Satz*, „ A_n ist von der *Annahme* Γ ableitbar.“

$\Gamma = \emptyset$: $\vdash A_n$ ist ein *Theorem*.

Satz: $\{A, (B \Rightarrow (A \Rightarrow C))\} \vdash (B \Rightarrow C)$

Beweis:

- | | | |
|-----|---|--------------------------|
| (1) | $(B \Rightarrow (A \Rightarrow C)) \Rightarrow ((B \Rightarrow A) \Rightarrow (B \Rightarrow C))$ | (L2) |
| (2) | $(B \Rightarrow (A \Rightarrow C))$ | Annahme |
| (3) | $((B \Rightarrow A) \Rightarrow (B \Rightarrow C))$ | aus (1) und (2) mit (MP) |
| (4) | $(A \Rightarrow (B \Rightarrow A))$ | (L1) |
| (5) | A | Annahme |
| (6) | $(B \Rightarrow A)$ | aus (4) und (5) mit (MP) |
| (7) | $(B \Rightarrow C)$ | aus (3) und (6) mit (MP) |

Satz: $\{\neg(\neg A)\} \vdash A$

Beweis: *Übung.*

Interpretation, Belegung = Abbildung $\hat{\mathcal{M}} : \{p_1, p_2, p_3, \dots\} \rightarrow \{\text{falsch, wahr}\}$

- $\hat{\mathcal{M}}(p_i) = \text{wahr}$: Schreibweise $\hat{\mathcal{M}} \models p_i$ („gültig“, $\hat{\mathcal{M}}$ ist *Modell*)
- $\hat{\mathcal{M}}(p_i) = \text{falsch}$: Schreibweise $\hat{\mathcal{M}} \not\models p_i$

$\hat{\mathcal{M}}$ induktiv auf ganz \mathcal{WFF} fortsetzen: $\mathcal{M} : \mathcal{WFF} \rightarrow \{\text{falsch, wahr}\}$:

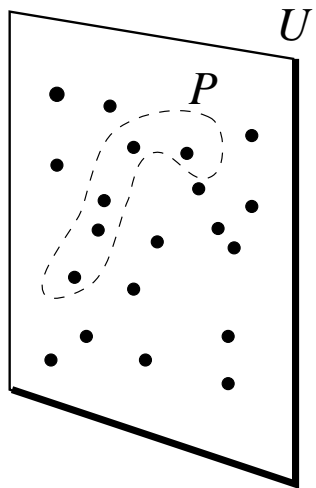
$$\mathcal{M} \models p_i \quad \text{gdw.} \quad \hat{\mathcal{M}} \models p_i$$

$$\mathcal{M} \models \neg B \quad \text{gdw.} \quad \mathcal{M} \not\models B$$

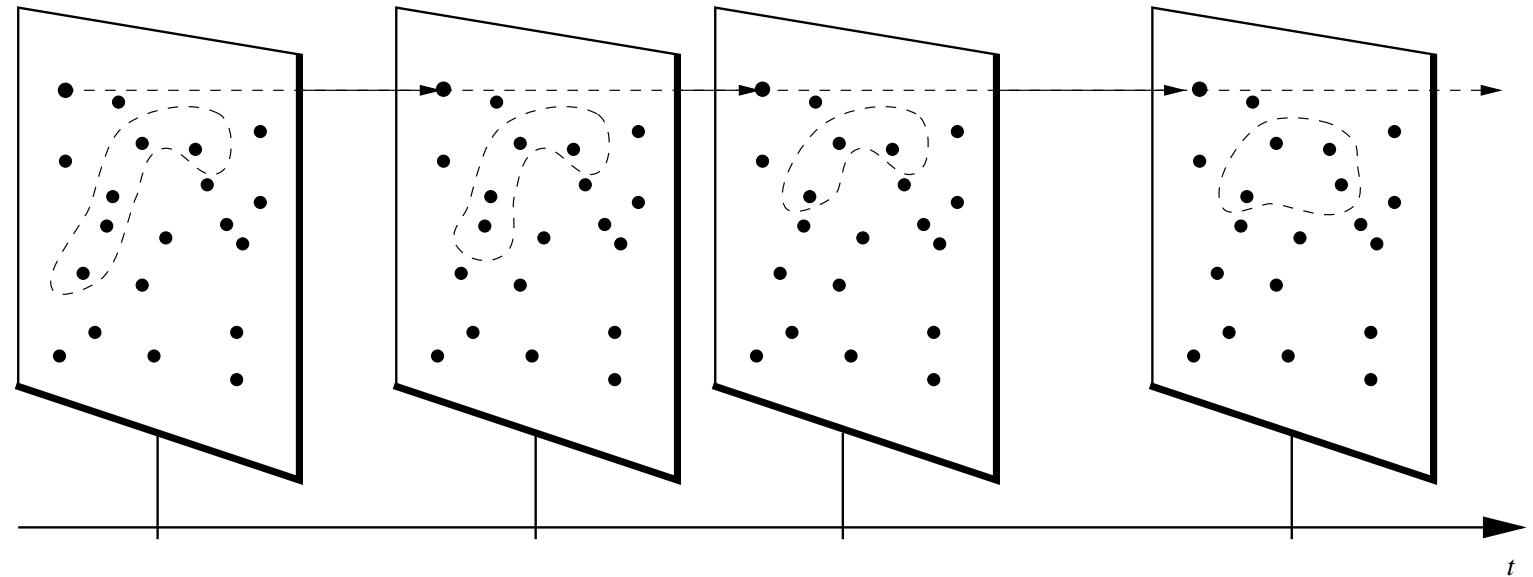
$$\mathcal{M} \models (B \Rightarrow C) \quad \text{gdw.} \quad \mathcal{M} \not\models B \text{ oder } \mathcal{M} \models C$$

Verbinden von Syntax und Semantik:

- *vollständig*, falls $\forall B \forall \mathcal{M}$: aus $\mathcal{M} \models B$ folgt $\vdash B$
- *korrekt*, falls $\forall B \forall \mathcal{M}$: aus $\vdash B$ folgt $\mathcal{M} \models B$



(a) Aussagen- und
Praedikatenlogik



(b) temporale Logik

Temporale Logik ist eine spezielle *modale Logik*, bei der Formeln nicht generell — sondern *unter Umständen* (= Modus!) — wahr oder falsch sind.

- „*Borussia Dortmund ist Deutscher Fußballmeister 1995/96.*“
- „*Borussia Dortmund ist Deutscher Fußballmeister.*“

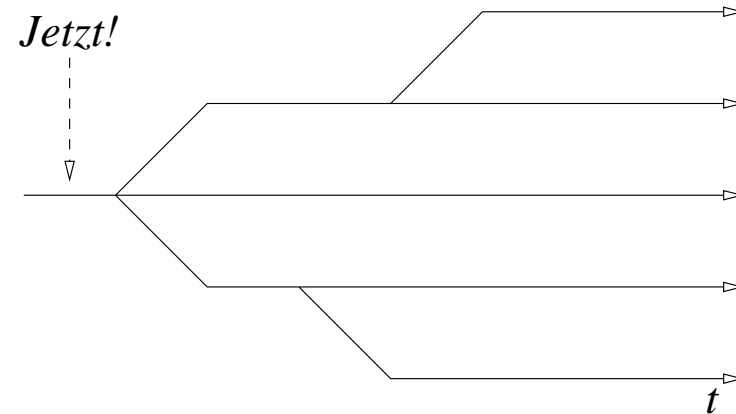
Ist dies wahr?

Spezielle temporale Operatoren:

immer, schließlich, möglicherweise, irgendwann, ...

Was bedeuten sie $\Gamma \rightarrow$ Frage der Semantik.

Was für eine Semantik $\Gamma \rightarrow$ Branching-Time oder Linear-Time



Syntax einfacher temporallogischer Formeln:

1. Jede atomare Proposition ist eine temporale Formel.
2. Wenn p und q temporale Formeln sind,
dann sind $(\neg p)$ und $(p \wedge q)$ temporale Formeln.
3. Wenn p eine temporale Formel ist,
dann sind $(\Box p)$ und $(\star p)$ temporale Formeln.

Semantik einfacher temporallogischer Formeln bezüglich *Pfadstruktur* $\mathcal{M} = (S, \Pi, L)$ erklären:

1. S nicht-leere Menge von *Zuständen*.
2. $\Pi \subseteq S^\infty$ nicht-leere Menge von unendlichen *Pfaden* $\pi = \langle s_0, s_1, s_2, \dots \rangle$.
3. L Abbildung, die jedem Zustand s die Menge aller atomarer Proposition, die im Zustand s gelten, zuordnet.

Branching–Time Semantik: $\mathcal{M}, s \models_B p,$

wenn p in der Pfadstruktur \mathcal{M} in dem Zustand s als wahr interpretiert wird.

$$\mathcal{M}, s \models_B P \quad \text{gdw.} \quad P \in L(s)$$

$$\mathcal{M}, s \models_B p \wedge q \quad \text{gdw.} \quad \mathcal{M}, s \models_B p \text{ und } \mathcal{M}, s \models_B q$$

$$\mathcal{M}, s \models_B \neg p \quad \text{gdw.} \quad \text{nicht } \mathcal{M}, s \models_B p$$

$$\mathcal{M}, s \models_B \Box p \quad \text{gdw.} \quad \forall \pi = \langle s_0, s_1, \dots \rangle \in \Pi, s_0 = s : \forall n \geq 0 : \mathcal{M}, s_n \models_B p$$

$$\mathcal{M}, s \models_B \star p \quad \text{gdw.} \quad \forall \pi = \langle s_0, s_1, \dots \rangle \in \Pi, s_0 = s : \exists n \geq 0 : \mathcal{M}, s_n \models_B p$$

Linear–Time Semantik: $\mathcal{M}, \pi \models_L p,$

wenn p in der Pfadstruktur \mathcal{M} für die Sequenz $\pi = \langle s_0, s_1, \dots \rangle$ als wahr interpretiert wird.

$$\mathcal{M}, \pi \models_L P \quad \text{gdw.} \quad P \in L(s_0)$$

$$\mathcal{M}, \pi \models_L p \wedge q \quad \text{gdw.} \quad \mathcal{M}, \pi \models_L p \text{ und } \mathcal{M}, \pi \models_L q$$

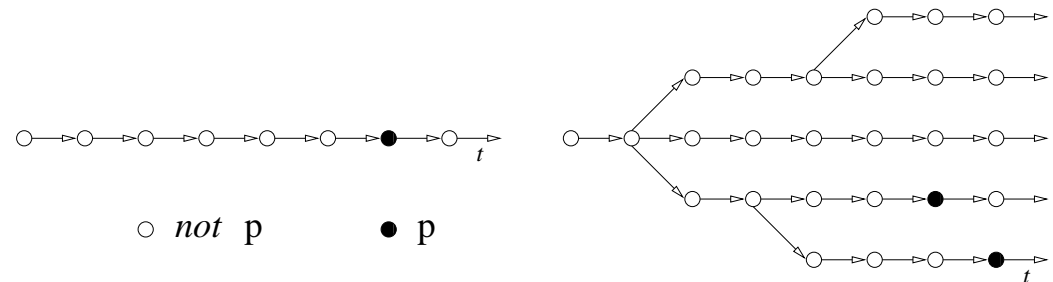
$$\mathcal{M}, \pi \models_L \neg p \quad \text{gdw.} \quad \text{nicht } \mathcal{M}, \pi \models_L p$$

$$\mathcal{M}, \pi \models_L \Box p \quad \text{gdw.} \quad \forall n \geq 0 : \mathcal{M}, \pi^n \models_L p$$

$$\mathcal{M}, \pi \models_L \star p \quad \text{gdw.} \quad \exists n \geq 0 : \mathcal{M}, \pi^n \models_L p$$

$$f := ((\Box p) \Rightarrow (q \wedge (\star r)))$$

$$g := \neg \Box \neg p$$



f bedeutet unter beiden Interpretationen:

Wenn p von dem Zustand s_0 an immer erfüllt ist, dann muß q in s_0 gelten und es muß irgendwann r erfüllt sein.

g bedeutet in der Linear-Time Semantik:

p gilt in s_0 oder irgendwann in der Zukunft.

Die Formel ist in der Linear-Time Semantik also äquivalent mit $\star p$.

In der Branching-Time Semantik bedeutet sie aber „nicht niemals p “:

Es ist möglich, daß p in s_0 oder irgendwann in der Zukunft gilt.

„Sometimes“ is sometimes „not never“ — but not always

Klassifizierung von Eigenschaften:

1. **Sicherheitseigenschaften (Safety Properties):** bestimmte Ereignisse oder Zustände treten nie ein. *„Nichts schlechtes passiert.“*
2. **Lebendigkeitseigenschaften (Liveness Properties):** bestimmte Ereignisse oder Zustände treten schließlich ein. *„Etwas gutes passiert.“*

„Immer wenn der Reaktor betrieben wird, dann ist die Kühlung eingeschaltet“

„Wenn der Not–Aus–Knopf betätigt wird, dann wird die Anlage heruntergefahren“

Sicherheitseigenschaften werden bereits durch endliche Präfixe eines Verhaltens verletzt, Lebendigkeitseigenschaften nur durch das gesamte (unendlich lange) Verhalten.

Lebendigkeit durch Fairness ausdrücken:

Ein System ist **fair** bzgl. einer Aktion \mathcal{A} ,
wenn \mathcal{A} unendlich häufig ausgeführt wird,
vorausgesetzt \mathcal{A} ist „genügend häufig“ ausführbar.

„genügend häufig“ =

- **starke Fairness** oder **Compassion**:
= unendlich häufig
- **schwache Fairness** oder **Justice**:
= nicht unendlich häufig nicht

CTL*-Syntax (Grammatik):

$\langle \text{log-Op-1} \rangle$	$::=$	\neg
$\langle \text{log-Op-2} \rangle$	$::=$	$\wedge \mid \vee \mid \Rightarrow \mid \Leftrightarrow$
$\langle \text{temp-Op-1} \rangle$	$::=$	$\mathbf{X} \mid \mathbf{G} \mid \mathbf{F}$
$\langle \text{temp-Op-2} \rangle$	$::=$	$\mathbf{U} \mid \mathbf{V}$
$\langle \text{P-Quantor} \rangle$	$::=$	$\mathbf{E} \mid \mathbf{A}$
$\langle \text{inf-Z-Quantor} \rangle$	$::=$	$\mathbf{F}^\infty \mid \mathbf{G}^\infty$
$\langle \text{Z-Formel} \rangle$	$::=$	$\langle \text{atomare-Proposition} \rangle \mid (\langle \text{Z-Formel} \rangle)$ $\mid \langle \text{log-Op-1} \rangle \langle \text{Z-Formel} \rangle \mid \langle \text{Z-Formel} \rangle \langle \text{log-Op-2} \rangle \langle \text{Z-Formel} \rangle$ $\mid \langle \text{P-Quantor} \rangle \langle \text{P-Formel} \rangle$
$\langle \text{P-Formel} \rangle$	$::=$	$\langle \text{Z-Formel} \rangle \mid (\langle \text{P-Formel} \rangle)$ $\mid \langle \text{log-Op-1} \rangle \langle \text{P-Formel} \rangle \mid \langle \text{temp-Op-1} \rangle \langle \text{P-Formel} \rangle$ $\mid \langle \text{P-Formel} \rangle \langle \text{log-Op-2} \rangle \langle \text{P-Formel} \rangle$ $\mid \langle \text{P-Formel} \rangle \langle \text{temp-Op-2} \rangle \langle \text{P-Formel} \rangle$ $\mid \langle \text{inf-Z-Quantor} \rangle \langle \text{P-Formel} \rangle$
$\langle \text{CTL}^*\text{-Formel} \rangle$	$::=$	$\langle \text{Z-Formel} \rangle$

CTL*-Semantik (Sprechweisen):

temporale Operatoren:

$X p$	<i>next time p</i>	im nächsten Zustand gilt p
$G p$	<i>always p</i>	p gilt immer
$F p$	<i>sometimes p</i>	p gilt irgendwann
$p U q$	<i>p until q</i>	p gilt solange bis q gilt und irgendwann gilt q
$p V q$	<i>p releases q</i>	q gilt solange bis p gilt

Pfad-Quantoren:

$E p$	<i>exists path: p</i>	es gibt einen Pfad auf dem p gilt
$A p$	<i>for all paths: p</i>	für alle Pfade gilt p

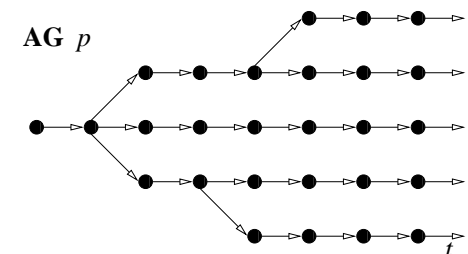
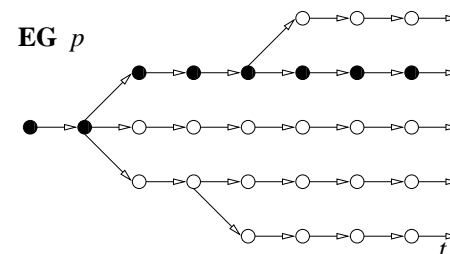
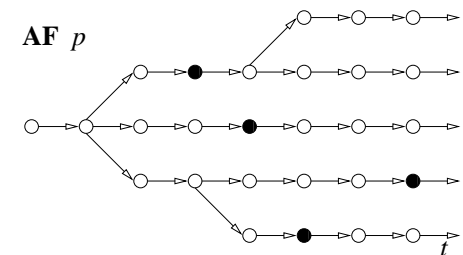
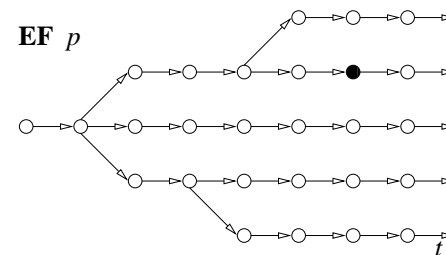
infinite Zustands-Quantoren:

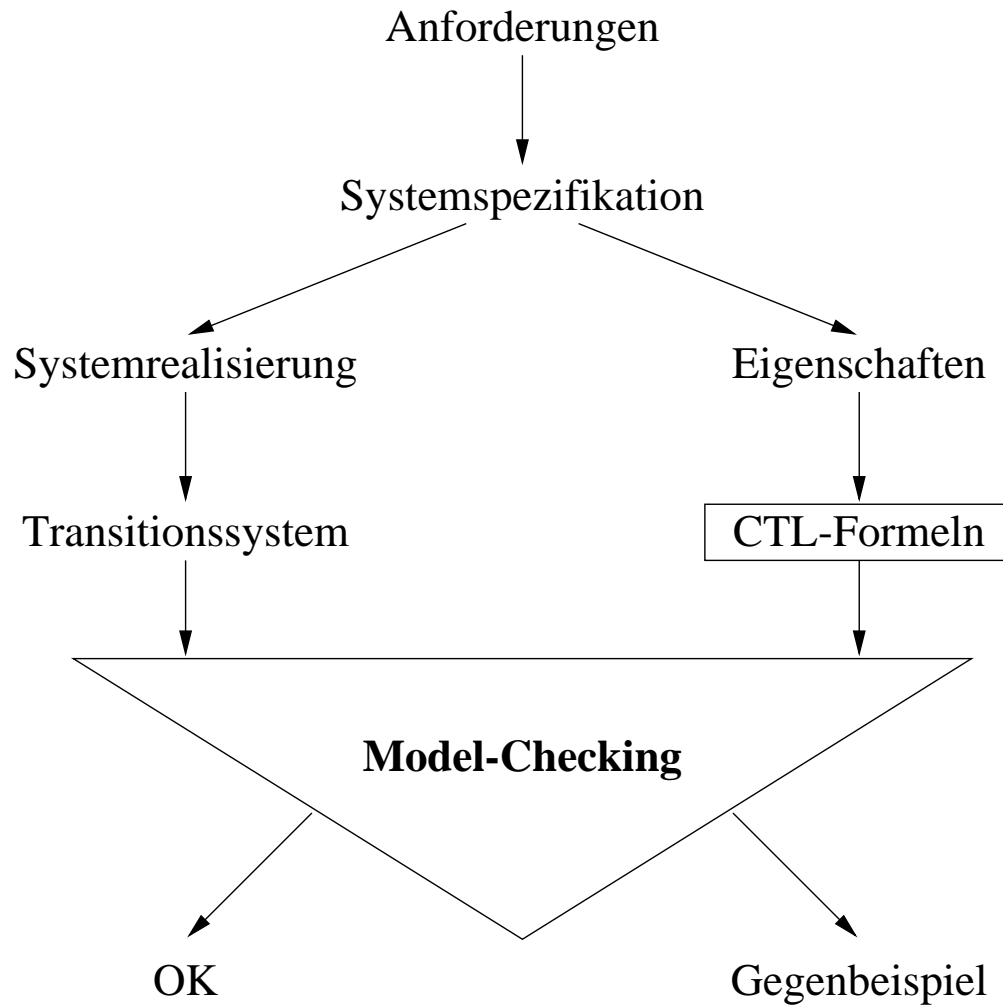
$F^\infty p$	<i>infinitely often p</i>	p gilt unendlich oft
$G^\infty p$	<i>almost everywhere p</i>	p gilt immer bis auf endlich viele Ausnahmen

CTL-Syntax (Grammatik):

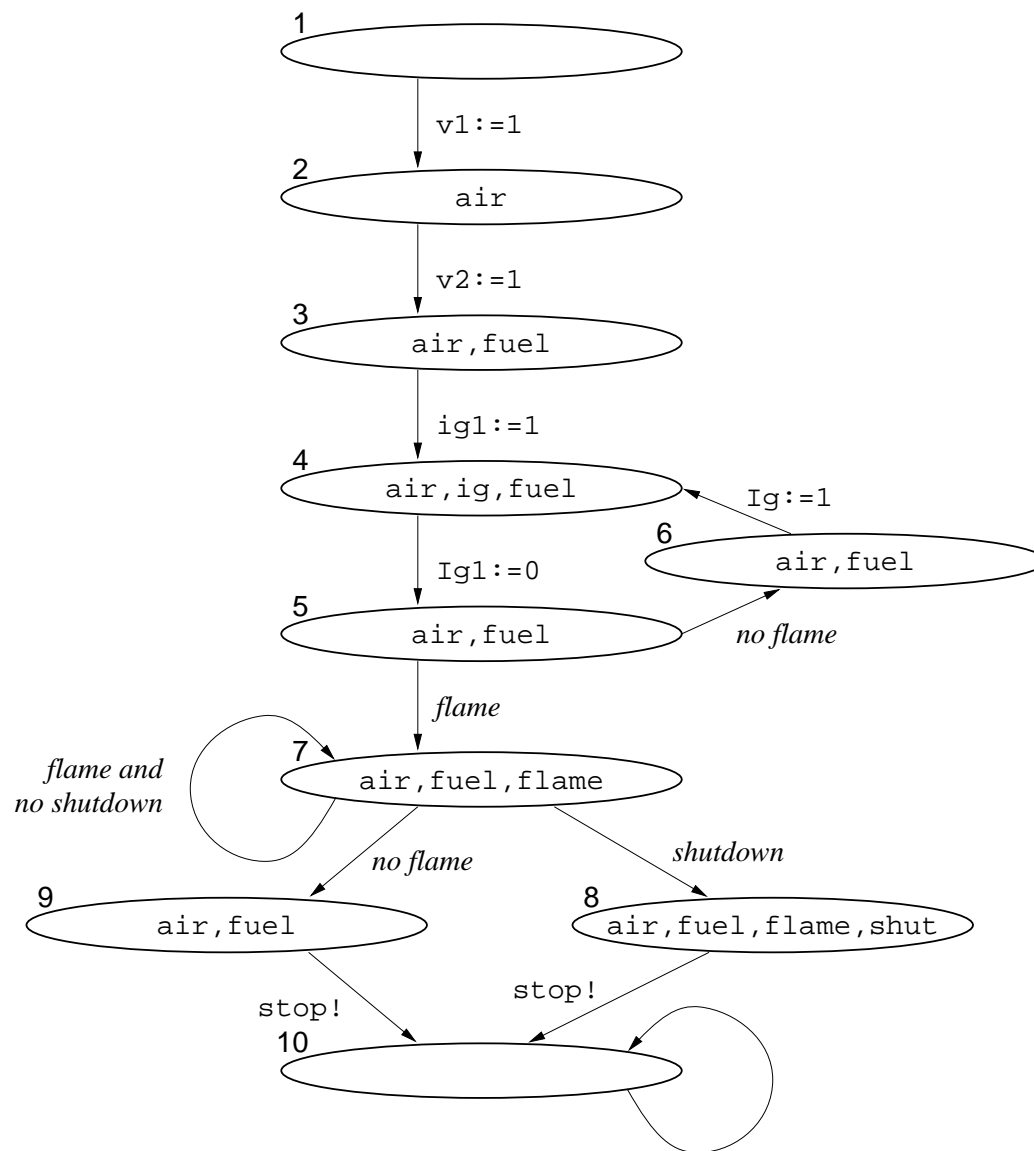
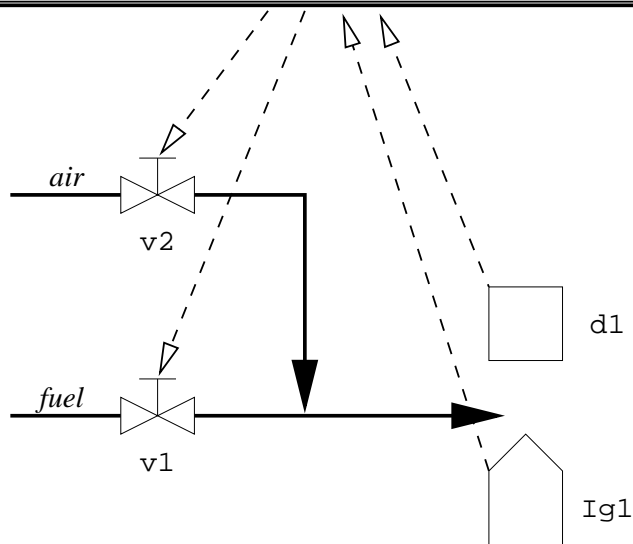
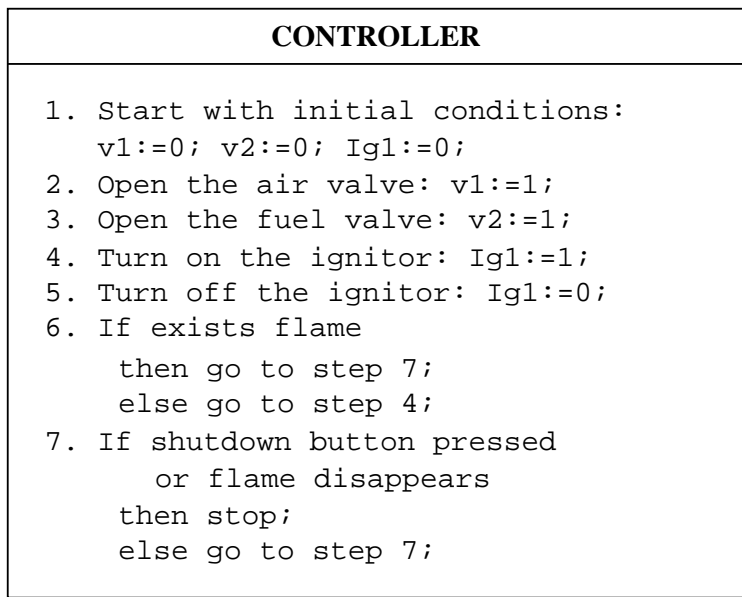
$$\begin{aligned} \langle \text{CTL-Formel} \rangle & ::= \langle \text{atomare-Proposition} \rangle \mid (\langle \text{CTL-Formel} \rangle) \\ & \mid \langle \text{log-Op-1} \rangle \langle \text{CTL-Formel} \rangle \mid \langle \text{CTL-Formel} \rangle \langle \text{log-Op-2} \rangle \langle \text{CTL-Formel} \rangle \\ & \mid \langle \text{P-Quantor} \rangle \langle \text{temp-Op-1} \rangle \langle \text{CTL-Formel} \rangle \\ & \mid \langle \text{P-Quantor} \rangle (\langle \text{CTL-Formel} \rangle \langle \text{temp-Op-2} \rangle \langle \text{CTL-Formel} \rangle) \end{aligned}$$

1. **EF** p : potentiell p , p ist möglich,
nicht niemals p .
2. **AF** p : p ist unausweichlich,
 p ist unvermeidbar.
3. **EG** p : potentiell immer p ,
immer p ist möglich.
4. **AG** p : immer p , p ist invariant.





irgendwo hier erfolgt der Wechsel von nicht-formalen zu formalen Methoden



Spezifikation & Verifikation mit CTL:

- System: als Transitionssystem modelliert
- geforderte Eigenschaften: CTL-Formeln

Eine sicherlich wünschenswerte Eigenschaft ist, daß es möglich sein muß, daß die Flamme irgendwann mal brennt:

$$\mathbf{EF} (air \wedge fuel \wedge flame)$$

Es soll aber ausgeschlossen sein, daß Brennstoff kontinuierlich fließen kann ohne daß eine Flamme brennt:

$$\neg \mathbf{EF} \mathbf{EG} (fuel \wedge \neg flame)$$

Kanonische TLA-Formel:
$$\Psi = \underbrace{\underbrace{Init}_{\text{initial}} \wedge \underbrace{\square [Next]_v}_{\text{next-step}}}_{\text{safety}} \wedge \underbrace{Fair}_{\text{fairness}}$$

1. *Init*: Prädikat über Variablen, Initialzustände

Bsp.: $Init := x = 0$

2. *Next*: Next-State-Relation, Disjunktion von Aktionen

Bsp.: $Next := A \vee B$

Aktion A: Prädikat über Zustandspaar

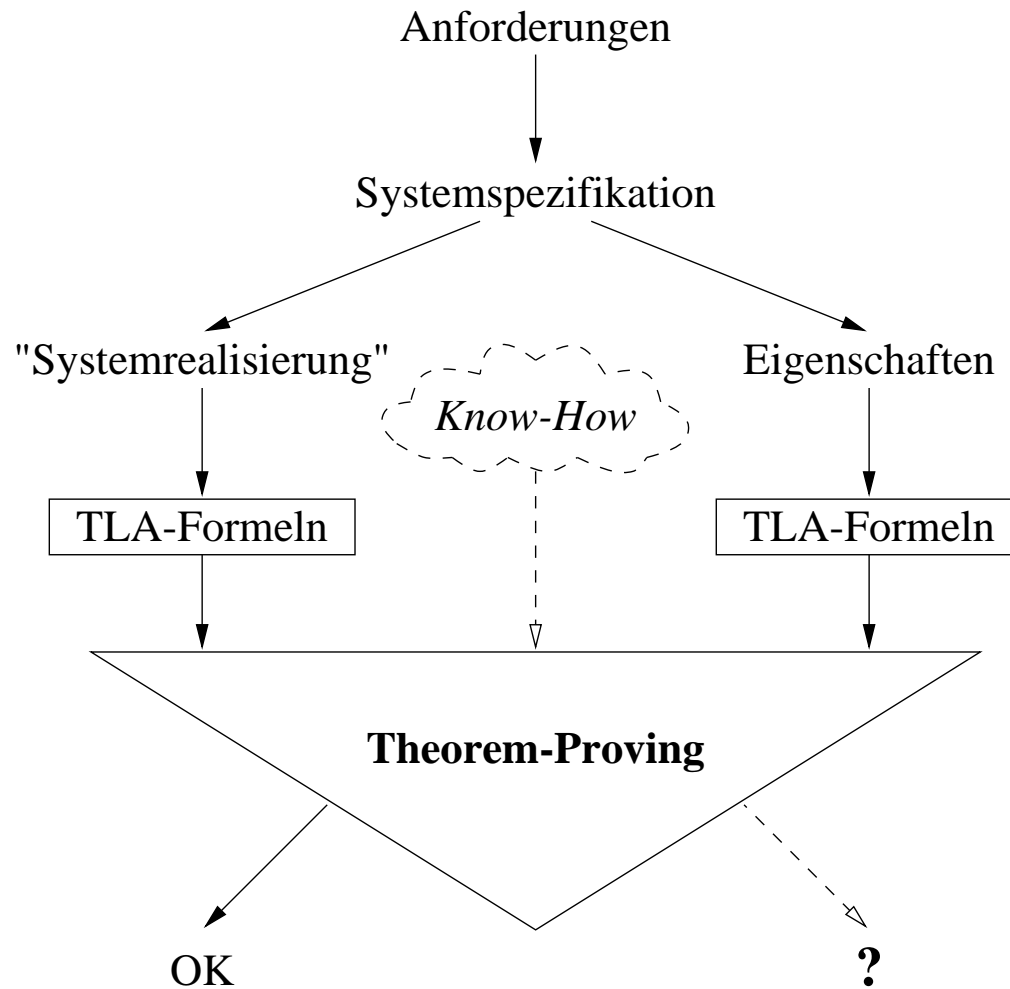
Bsp.: $A := x > 0 \wedge x' = x - 1$

$[Next]_v$: Abkürzung für $Next \vee (v' = v)$

$\square [Next]_v$ immer *Next*-Schritt oder *Stottersschritt*

3. *Fair*: Konjunktion von Fairness-Bedingungen über Aktionen

Bsp.: $Fair := WF_v(A) \wedge SF_v(B)$



irgendwo hier erfolgt der Wechsel von nicht-formalen zu formalen Methoden

Beispiel: Spezifikation von Uhren

$$\begin{aligned}
 Init_1 &:= \wedge h = 0 \wedge m = 0 \\
 \mathcal{A}_1 &:= \wedge m < 59 \\
 &\quad \wedge h' = h \wedge m' = m + 1 \\
 \mathcal{B}_1 &:= \wedge m = 59 \wedge h < 23 \\
 &\quad \wedge h' = h + 1 \wedge m' = 0 \\
 \mathcal{C}_1 &:= \wedge m = 59 \wedge h = 23 \\
 &\quad \wedge h' = 0 \wedge m' = 0 \\
 Next_1 &:= \mathcal{A}_1 \vee \mathcal{B}_1 \vee \mathcal{C}_1 \\
 \Phi_1 &:= Init_1 \wedge \square [Next_1]_{(h,m)}
 \end{aligned}$$

$$\begin{aligned}
 Init_2 &:= \wedge h = 0 \wedge m = 0 \wedge s = 0 \\
 \mathcal{A}_2 &:= \wedge s < 59 \\
 &\quad \wedge h' = h \wedge m' = m \wedge s' = s + 1 \\
 \mathcal{B}_2 &:= \wedge s = 59 \wedge m < 59 \\
 &\quad \wedge h' = h \wedge m' = m + 1 \wedge s' = 0 \\
 \mathcal{C}_2 &:= \wedge s = 59 \wedge m = 59 \wedge h < 23 \\
 &\quad \wedge h' = h + 1 \wedge m' = 0 \wedge s' = 0 \\
 \mathcal{D}_2 &:= \wedge s = 59 \wedge m = 59 \wedge h = 23 \\
 &\quad \wedge h' = 0 \wedge m' = 0 \wedge s' = 0 \\
 Next_2 &:= \mathcal{A}_2 \vee \mathcal{B}_2 \vee \mathcal{C}_2 \vee \mathcal{D}_2 \\
 \Phi_2 &:= Init_2 \wedge \square [Next_2]_{(h,m,s)}
 \end{aligned}$$

$$(\exists s : \Phi_2) \Rightarrow \Phi_1$$

(\exists = Verstecken interner Variablen)

- Φ_2 implementiert Φ_1 ,
- Φ_2 ist eine Verfeinerung von Φ_1 , oder
- Φ_1 ist eine Abstraktion von Φ_2 .

$$\begin{aligned}
 \text{step2} & := \wedge PC = 2 \wedge PC' = 3 \\
 & \quad \wedge v1' = 1 \\
 \text{step3} & := \wedge PC = 3 \wedge PC' = 4 \\
 & \quad \wedge v2' = 1 \\
 \text{step4} & := \wedge PC = 4 \wedge PC' = 5 \\
 & \quad \wedge Ig1' = 1 \\
 \text{step5} & := \wedge PC = 5 \wedge PC' = 6 \\
 & \quad \wedge Ig1' = 0 \\
 \text{step6flame} & := \wedge PC = 6 \wedge PC' = 7 \\
 & \quad \wedge d1 = 1 \\
 \text{step6fail} & := \wedge PC = 6 \wedge PC' = 4 \\
 & \quad \wedge d1 = 0 \\
 \text{step7shut} & := \wedge PC = 7 \wedge PC' = 8 \\
 & \quad \wedge shut = 1 \\
 \text{step7fail} & := \wedge PC = 7 \wedge PC' = 8 \\
 & \quad \wedge d1 = 0 \\
 \text{flameOn} & := \wedge v1 = 1 \wedge v2 = 1 \\
 & \quad \wedge Ig1 = 1 \wedge d1' = 1 \\
 \text{flameFail} & := \wedge d1 = 1 \wedge d1' = 0 \\
 \text{stop} & := \wedge PC = 8 \wedge PC' = \text{stopped} \\
 & \quad \wedge v1' = 0 \wedge v2' = 0 \\
 & \quad \wedge Ig1' = 0 \wedge d1' = 0
 \end{aligned}$$

$$\begin{aligned}
 \text{Init} & := \wedge v1 = 0 \wedge v2 = 0 \\
 & \quad \wedge Ig1 = 0 \wedge d1 = 0 \\
 & \quad \wedge PC = 2 \\
 \mathcal{N} & := \vee \text{step2} \vee \text{step3} \\
 & \quad \vee \text{step4} \vee \text{step5} \\
 & \quad \vee \text{step6flame} \vee \text{step6fail} \\
 & \quad \vee \text{step7shut} \vee \text{step7fail} \\
 & \quad \vee \text{flameOn} \vee \text{flameFail} \vee \text{stop} \\
 v & := (PC, v1, v2, Ig1, d1) \\
 F & := \wedge \text{WF}_v(\text{step2}) \\
 & \quad \wedge \text{WF}_v(\text{step3}) \\
 & \quad \wedge \text{WF}_v(\text{step4}) \\
 & \quad \wedge \text{WF}_v(\text{step5}) \\
 & \quad \wedge \text{WF}_v(\text{step6flame}) \\
 & \quad \wedge \text{WF}_v(\text{step6fail}) \\
 & \quad \wedge \text{WF}_v(\text{step7shut}) \\
 & \quad \wedge \text{WF}_v(\text{step7fail}) \\
 & \quad \wedge \text{SF}_v(\text{flameOn}) \\
 & \quad \wedge \text{WF}_v(\text{stop}) \\
 B & := \exists PC : \text{Init} \wedge \square [\mathcal{N}]_v \wedge F
 \end{aligned}$$

Spezifikation & Verifikation mit TLA:

- System: TLA–Formeln
- geforderte Eigenschaften: TLA–Formeln

Eine sicherlich wünschenswerte Eigenschaft ist, daß es möglich sein muß, daß die Flamme irgendwann mal brennt:

$$B \Rightarrow \diamond (v1 = 1 \wedge v2 = 1 \wedge d1 = 1)$$

Es soll aber ausgeschlossen sein, daß Brennstoff kontinuierlich fließen kann ohne daß eine Flamme brennt:

$$B \Rightarrow \neg \diamond \square (v1 = 1 \wedge v2 = 1 \wedge d1 = 0)$$

Wichtige zeitbeschränkte Eigenschaften:

- **Beschränkte Antwort**

(*bounded-response*):

„Jedem Aufruf p folgt innerhalb 3 Zeiteinheiten eine Antwort q .“

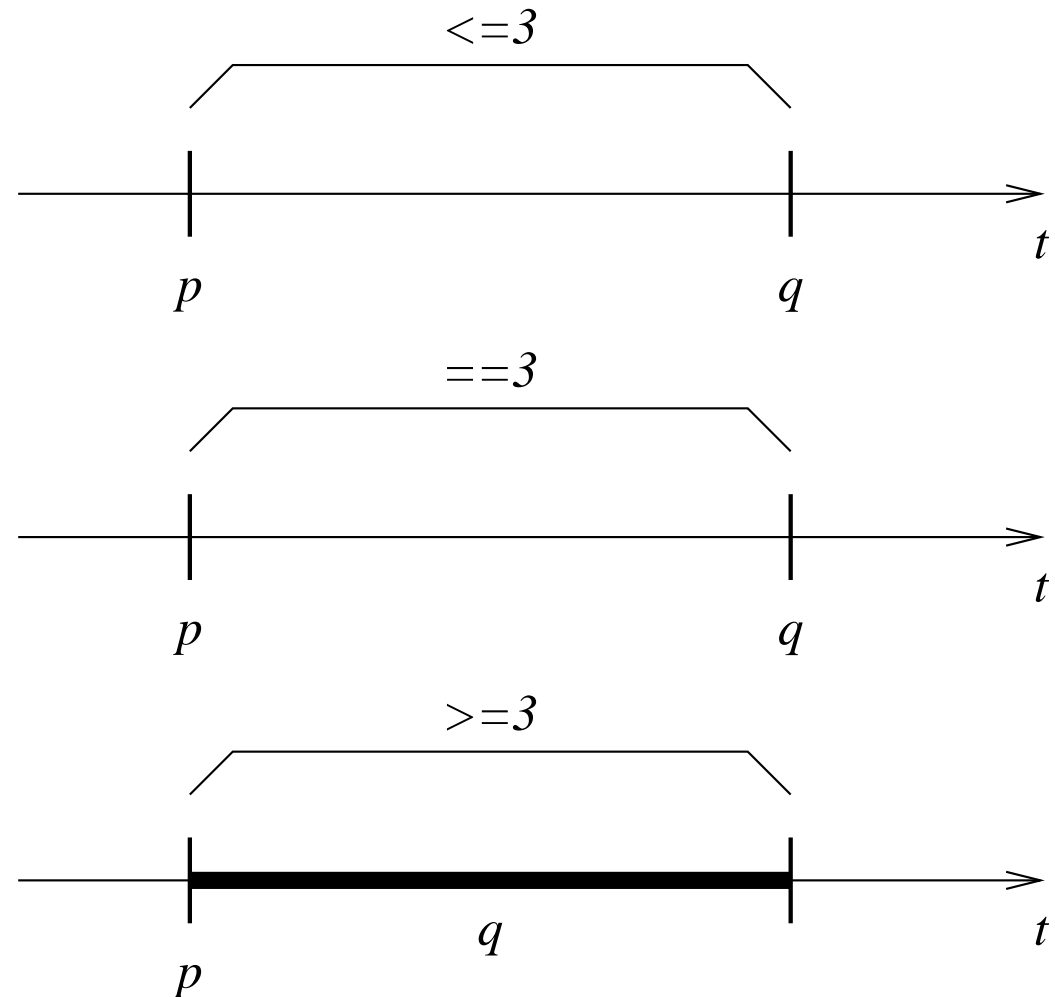
- **Pünktlichkeit** (*punctuality*):

„Jedem Aufruf p folgt nach genau 3 Zeiteinheiten eine Antwort q .“

- **Beschränkte Invarianz**

(*bounded-invariance*):

„Nach jedem Aufruf p ist q mindestens 3 Zeiteinheiten gültig.“



Einführung von Zeitschranken:

1. Beschränkte temporale Operatoren, z.B.:

- BR: $\Box (p \Rightarrow (\Diamond_{[0,3]} q))$, $p \rightsquigarrow_{\leq 3} q$
- P : $\Box (p \Rightarrow (\Diamond_{[3,3]} q))$, $p \rightsquigarrow_{=3} q$
- BI: $\Box (p \Rightarrow (\Box_{\leq 3} q))$

2. Freeze-Quantification, z.B.:

- BR: $\Box (x : (p \Rightarrow (\Diamond y : (q \wedge y \leq x + 3))))$
- P : $\Box (x : (p \Rightarrow (\Diamond y : (q \wedge y = x + 3))))$
- BI: $\Box (x : (p \Rightarrow (\Box y : (q \vee y > x + 3))))$

3. Explizite Uhrenvariable, z.B.:

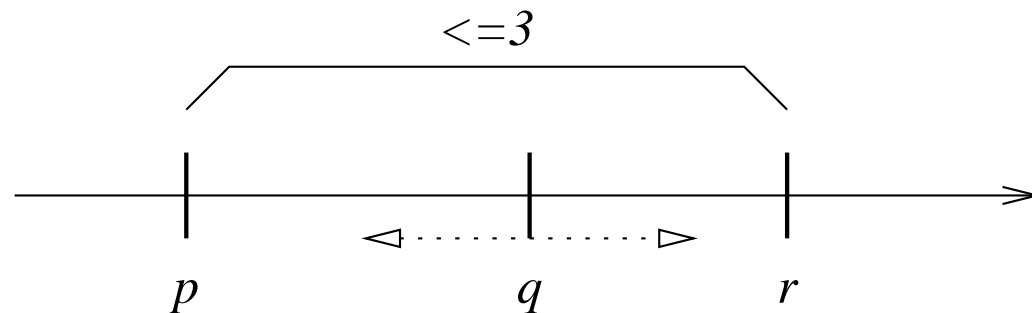
- BR: $\forall x : \Box ((p \wedge T = x) \Rightarrow (\Diamond (q \wedge T \leq x + 3)))$
- P : $\forall x : \Box ((p \wedge T = x) \Rightarrow (\Diamond (q \wedge T = x + 3)))$
- BI: $\forall x : \Box ((p \wedge T = x) \Rightarrow (\Box (q \vee T > x + 3)))$

Beschr. temp. Op. „<“ Freeze Quant. „<“ Explizite Uhrenvar.

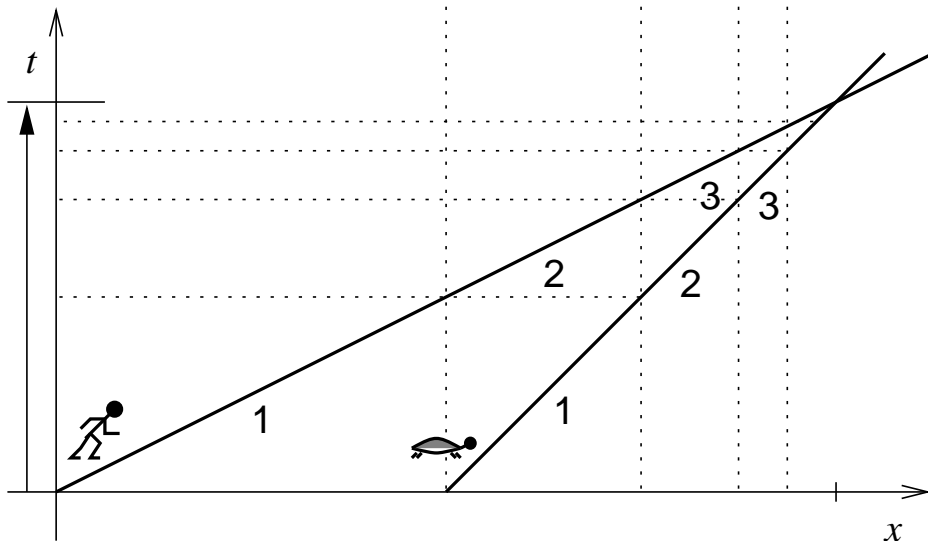
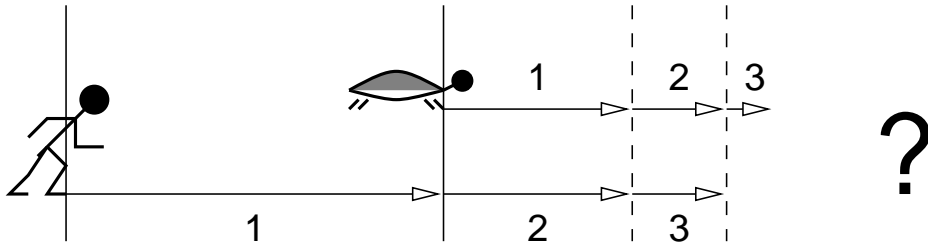
$$p \mathbf{U}_I q \iff x : (p \mathbf{U} y : (q \wedge y \in x + I))$$

$$x : \phi \iff \exists x : (T = x \wedge \phi) \iff \forall x : (T = x \wedge \phi)$$

Nicht-lokale Zeitschranken: $\Box x : (p \Rightarrow \Diamond (q \wedge \Diamond z : (r \wedge z \leq x + 3)))$



Bezug auf absolute Zeit: $\Box ((T \equiv 0 \pmod{2}) \Rightarrow q)$



Forderung „Non-Zenoness“:

1. *Monotonie*:

$$(T \in \mathcal{R}_{\geq 0}) \wedge \square (T' \in [T, \infty[)$$

2. *Fortschritt*:

$$\forall t \in \mathcal{R}_{\geq 0} : \diamond (T > t)$$

Genügen abzählbar viele Bedingungen:

$$\forall t \in \mathcal{N} : \diamond (T > t)$$

Häufig Zustands- und Zeitschritte
exklusiv:

$$\square (V' = V \vee T' = T)$$

- wesentliche semantische Unterschiede zwischen Linear- und Branching-Time
- Branching-Time mächtiger
- Verifikation: Model-Checking oder Theorem-Proving
- Methoden zur strukturierten Komposition von Spezifikationen notwendig
- Toolunterstützung notwendig