Temporallogik

Ablaufverhalten von Programmen

- Programmabläufe: Lineare Sequenzen von
 - Programmzuständen
 - Ereignissen
 - Beobachtungen
- Potentiell endlos
 - o Interaktive Programme (z.B. GUI-Anwendungen, Server)
 - o Allgemein: Systeme ohne Terminierungsgarantie

ISP Software-Engineering

Ziele & Gliederung

Spez-78

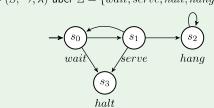
Temporallogik (Forts.)

Endliche Abläufe

- endliche Pfade $u \in S^*$, z.B. $s_0s_1s_0s_1s_3$
- Beschriftung: endliche Wörter, z.B. $\lambda(u) = wait \ serve \ wait \ serve \ halt \in \Sigma^*$

Beispiel (Transitionssystem T)

 $T = (S, \rightarrow, \lambda)$ über $\Sigma = \{wait, serve, halt, hang\}$:



isp Software-Engineering

Zustände und Abläuf

Spez-79

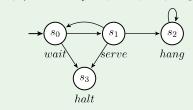
Temporallogik (Forts.)

Unendliche Abläufe

- unendliche Pfade, z. B. $s_0s_1s_0s_1s_0s_1...$
- Beschriftung: unendliche Wörter, z.B. $w = wait \ serve \ wait \ serve \ wait \ serve \dots$

Beispiel (Transitionssystem T)

 $T = (S, \rightarrow, \lambda)$ über $\Sigma = \{wait, serve, halt, hang\}$:



isp Software-Engineerin

Spez-80

Temporallogik (Forts.)

Unendliche Sequenzen

- ullet A^ω beschreibt die Menge unendlicher Sequenzen aus Elementen der Menge A
- u^{ω} bezeichnet die unendliche Wiederholung einer endlichen Sequenz $u \in A^*$
- Beispiel: $(wait\ serve)^{\omega}$ ist die unendliche Sequenz wait serve wait serve $\dots \in \Sigma^{\omega}$
- Begriffe: " ω -Wörter" (Elemente $w \in \Sigma^{\omega}$) und " ω -Sprachen" (Teilmengen $L \subseteq \Sigma^{\omega}$)

(Hintergrund: ω bezeichnet üblicherweise die kleinste nicht endliche Größe, z.B. bei Ordinalzahlen)

ISP Software-Engineering

Spez-81

Temporallogik (Forts.)

Temporale Eigenschaften

Spezifizieren "zeitlicher" Zusammenhänge, z.B:

- davor, danach, als nächstes, irgendwann, nie, immer ...
- Iteratoren: "Direkt vor Aufruf der Methode next (), wird stets die Methode hasNext () benutzt."
- Programmzustände: "Ein mit fail beschrifteter Zustand wird nie durchlaufen."
- "Solange die Verbindung nicht geschlossen wird, erfolgt immer wieder eine Synchronisation."

ISP Software-

Martin Leucker Spez-82

Temporallogik (Forts.)

Mathematische Logiken:

- unmissverständliche Aussagen
- "Bedeutung" klar definiert
- gut untersuchte Eigenschaften
- automatische Analysen

isp

Martin Leucker Spez-83

Temporallogik (Forts.)

Prädikatenlogik

"Ein mit fail beschrifteter Zustand wird nie durchlaufen."

- $\neg \exists i : Fail(i)$
- ullet Universum U: Positionen in einem Wort w
- Prädikat $Fail \subseteq U$: Menge der Positionen in einem Wort w, die mit fail beschriftet sind
- Modelle: Wörter
- Modell für diese Formel: $w = wait \ serve \ halt$
- ullet kein Modell: $w=wait\ serve\ wait\ serve\ fail\ serve\ wait$

ISP Software-Engineering

Spezifikation Ziele & Gliederung Überblick Nebenläufige Aktivität Petri-Netze

Sequenzdiagramme Temporallogik Algebraische Spezifikati

> Martin Leuck Spez-84

Temporallogik (Forts.)

LTL: Linear-time temporal logic

- Formalismus zum (möglichst intuitiven) spezifizieren zeitlicher Zusammenhänge
- Aussagen werden über Wörtern (z.B. den Läufen eines Systems) ausgewertet

ISP Software-Engineering

Spezifikation
Ziele & Gliederung
Überblick
Nebenläufige Aktivitäter
Petri-Netze
Aktivitätsdiagramme
Zustände und Abläufe

Zustandsdiagramme Sequenzdiagramme Temporallogik Algebraische Spezifikation

Martin Leucke Spez-85

Temporallogik (Forts.)

Formel: φ

Die Formel φ hält für eine Ausführung, wenn φ in dem ersten Zustand s_0 der Ausführung hält.



isp Software-Engineering

Spezifikation
Ziele & Gliederung
Überblick
Nebenläufige Aktivität
Petri-Netze
Aktivitätsdiagramme
Zustände und Abläufe
Zustandsdiagramme

Sequenzdiagramme Temporallogik Algebraische Spezifika

> Martin Leucker Spez-88

Temporallogik (Forts.)

Next: $\mathcal{X} \varphi$

Die Formel $\mathcal{X} \varphi$ hält in dem Zustand s_i , wenn φ in dem Zustand s_{i+1} hält.



ISP Software-Engineering

Spezifikation
Ziele & Gliederung
Überblick
Nebenlänlige Aktivitäten
Petri-Netze
Aktivitätedagramme
Zustände und Abläufe
Zuständelingsamme
Sequenzilagramme
Sequenzilagramme
Temporalogig
Algebraische Spezifikation

lartin Leucker Spez-89

Temporallogik (Forts.)

Globally: $\mathcal{G}\varphi$

Die Formel $\mathcal{G}\, \varphi$ hält in dem Zustand s_i , wenn φ in allen Zuständen s_j für $j\geq i$ hält.



ISP Software-Engineering

Ziele & Gliederung
Überblick
Nebenläufige Aktivitär
Petri-Netze
Aktivitätsdiagramme
Zustände und Abläufe
Zustandsdiagramme
Sequenzdiagramme
Temporallogik
Algebraische Spezifika

Temporallogik (Forts.)

Finally: $\mathcal{F}\varphi$

Die Formel $\mathcal{F} \varphi$ hält in dem Zustand s_i , wenn es einen Zustand s_j für $j \geq i$ gibt, in dem φ hält.



ISP Software-Engineering

Spezifikation
Ziele & Gliederung
Überblick
Nebenläufige Aktivitäten
Petri-Netze
Aktivitätsdiagramme
Zustände und Abläufe
Zustandediagramme
Sequenzidagramme
Tempozallogik
Algebraische Spezifikation

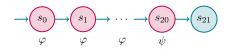
Martin Leucker Spez-91

Martin Leucker Spez-90

Temporallogik (Forts.)

Until: $\varphi \mathcal{U} \psi$

Die Formel $\varphi \mathcal{U} \psi$ hält in dem Zustand s_i , wenn es einen Zustand s_j für $j \geq i$ gibt, in dem ψ hält und φ in allen Zuständen s_k für $i \le k < j$ hält.



Achtung: Es muss nicht unbedingt einen Zustand geben, in dem φ hält.

isp Software-Engineering

Spezifikation Ziele & Gliederung

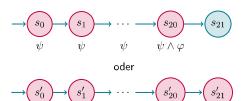
Zustände und Abläufe

Temporallogik (Forts.)

Release: $\varphi \mathcal{R} \psi$

Die Formel $\varphi \mathcal{R} \psi$ hält in dem Zustand s_i , wenn es einen Zustand s_j für $j \geq i$ gibt, in dem φ hält und ψ in allen Zuständen s_k für $i \le k \le j$ hält.

Wenn es keinen solchen Zustand s_j gibt, dann hält $\varphi \mathcal{R} \psi$, wenn ψ in allen Zuständen s_k für $k \geq i$ hält.



ψ

isp Software-Engineering

Zustände und Abläufe

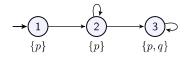
Spez-93

Temporallogik (Forts.)

Propositionen in Transitionssystemen

Transitionssystem $T=(S, \rightarrow, s_1, \lambda)$ über $\Sigma=2^{AP}$ mit:

- $AP = \{p, q\}$,
- $S = \{s_1, s_2, s_3\}$,
- $\rightarrow = \{(s_1, s_2), (s_2, s_2), (s_2, s_3), (s_3, s_3)\},\$
- $\lambda : \{s_1 \mapsto \{p\}, s_2 \mapsto \{p\}, s_3 \mapsto \{p, q\}\}$



Gibt es einen Pfad in T, auf dem niemals q gilt?

ISP

Spez-92

Software-Engineering

Spez-98

Temporallogik (Forts.)

Für ein ω -Wort $w=w_0w_1...\in \Sigma^\omega$ ($w_i\in \Sigma$) ist die Modell-Relation ⊨ induktiv definiert:

> $w \models \mathsf{True}$ $w \models p$ falls $p \in w_0$ falls $w \models \varphi_1$ oder $w \models \varphi_2$ $w \models \varphi_1 \vee \varphi_2$ $w \models \neg \varphi$ falls $nicht w \models \varphi$ falls $w^{(1)} \models \varphi$ $w \models \mathcal{X} \varphi$ falls $\exists_i : w^{(i)} \models \varphi_2$ und $w \models \varphi_1 \ \mathcal{U} \varphi_2$ $\forall_{j,0 \le j < i} : w^{(j)} \models \varphi_1$

Dabei bezeichnet $w^{(n)}$ das Suffix $w_n w_{n+1} w_{n+2} \dots$ von wab der Position n.

ISP Software-Engineering

Ziele & Gliederung stände und Abläufe ustandsdiagrams

Spez-100

ISP

Temporallogik (Forts.)

Alle weiteren Operatoren können wie folgt als syntaktische Abkürzungen betrachtet werden:

$$\begin{array}{lll} \varphi_1 \wedge \varphi_2 & :\equiv & \neg(\neg\varphi_1 \vee \neg\varphi_2) \\ \varphi_1 \to \varphi_2 & :\equiv & \neg\varphi_1 \vee \varphi_2 \\ \varphi_1 \leftrightarrow \varphi_2 & :\equiv & (\varphi_1 \to \varphi_2) \wedge (\varphi_2 \to \varphi_1) \\ \mathcal{F} \varphi & :\equiv & \mathrm{True} \ \mathcal{U} \varphi & \mathrm{,finally} \ \varphi'' \\ \mathcal{G} \varphi & :\equiv & \neg \mathcal{F} \neg \varphi & \mathrm{,globally} \ \varphi'' \\ \varphi_1 \ \mathcal{R} \varphi_2 & :\equiv & \neg(\neg\varphi_1 \ \mathcal{U} \neg \varphi_2) & \mathrm{,} \varphi_1 \ \mathrm{releases} \ \varphi_2'' \end{array}$$

ISP Software-

Martin Leucker Spez-101

Temporallogik (Forts.)

Praktische Beispiele

In den folgenden Beispielen betrachten wir diese Bereiche:

immer: alle Zustände

vor ψ : alle Zustände vor dem ersten in dem ψ hält (wenn es einen solchen Zustand gibt)

nach ψ : alle Zustände nach dem ersten (inkl.) in dem ψ hält (wenn es einen solchen Zustand gibt)

> Martin Leucker Spez-102

Temporallogik (Forts.)

Beispiel (Abwesenheit)

Die Formel φ hält

immer: $\mathcal{G} \neg \varphi$

vor ψ : $(\mathcal{F}\psi) \rightarrow (\neg \varphi \mathcal{U} \psi)$

nach ψ : $\mathcal{G}(\psi \rightarrow (\mathcal{G} \neg \varphi))$

nicht

isp Software-Engineering

Spezifikation
Ziele & Gilederung
Überblick
Nebenläufige Aktivitäts
Petri-Netze
Aktivitätsdisgramme
Zustände und Abläufe
Zustandsdisgramme
Temporallogik

Spez-103

Temporallogik (Forts.)

Beispiel (Existenz)

Die Formel φ hält in der Zukunft

immer: $\mathcal{F}\varphi$

vor ψ : $\mathcal{G} \neg \psi \lor \neg \psi \mathcal{U}(\varphi \land \neg \psi)$

nach ψ : $\mathcal{G} \neg \psi \lor \mathcal{F}(\psi \land \mathcal{F} \varphi)$

isp Software-Engineering

Spezifikation
Ziole & Gliederung
Überblick
Nebenläufige Aktivitäten
Petri-Netze
Aktivitätsdiagramme
Zustände und Abläufe
Zustandsdiagramme
Sequenzdiagramme
Temporallogiik

Spez-104

Temporallogik (Forts.)

Beispiel (Universalität)

Die Formel φ hält

immer: $\mathcal{G} \varphi$

vor ψ : $(\mathcal{F}\psi) \rightarrow (\varphi \mathcal{U} \psi)$

 $\mathbf{nach}\ \psi\mathbf{:}\ \mathcal{G}(\psi\!\to\!\mathcal{G}\,\varphi)$

isp Software-Engineering

Spez-105