

Verifikation verteilter Systeme

Seminar Systementwurf
Hartmut Lackner

1

Übersicht

- Einleitung
- Rekapitulation
 - Temporale Logik
 - Eigenschaften verteilter Systeme
- Verifikation verteilter System
 - Beweistheoretischer-Ansatz
 - Maschinelle Synthese
 - Automatisierte Verifikation

Seminar Systementwurf
Verifikation verteilter Systeme
Hartmut Lackner

2

Einleitung

→ Klassifizierung von Systemen

- Zwei Klassen von Systemen
 - Sequentielle
 - Transformation: Eingabe, Berechnung, Ausgabe
 - Korrektheit: Vor-/Nachbedingung
 - Reaktive verteilte
 - Kontinuierlich berechnend, nicht terminierend
 - Korrektheit: Zeitbezug

Seminar Systementwurf
Verifikation verteilter Systeme
Hartmut Lackner

3

Einleitung

→ Reaktive verteilte Systeme

- Sequentielle Systeme können parallele Architekturen ausnutzen
- Reaktive Systeme können auch in sequentiellen Architekturen implementiert werden
- Verteilte Systeme sind reaktiv

Seminar Systementwurf
Verifikation verteilter Systeme
Hartmut Lackner

4

Einleitung

→ Anwendung der TL

- Beweistheoretisch
 - Korrektheitsbeweis mittels deduktiver Systeme (Axiome und Schlussregeln)
- Modelltheoretisch
 - Automatisierung durch Entscheidungsverfahren

Seminar Systementwurf
Verifikation verteilter Systeme
Hartmut Lackner

5

Rekapitulation

→ PLTL

- Linear Time Structure: $M = (S, x, L)$
- In Struktur M erfüllt Formel p Zeitlinie x : $M, x \models p$

■ „p until q“:	$(p \text{ U } q)$	■ „lasttime p“:	Xp
■ „nexttime p“:	Xp	■ „past p“:	Fp
■ „sometimes q“:	Fq	■ „past p“:	Gp
■ „always p“:	Gp	■ „p before q“:	$(p \text{ B } q)$
■ „p before q“:	$(p \text{ B } q)$	■ „p unless q“:	$(p \text{ U}_w q)$
■ „p unless q“:	$(p \text{ U}_w q)$		

Seminar Systementwurf
Verifikation verteilter Systeme
Hartmut Lackner

6

Rekapitulation

→ Eigenschaften für Korrektheit

- Eigenschaften für Korrektheit
 - Sicherheit: „Nichts schlimmes passiert“
 - Lebendigkeit: „Etwas Gutes wird passieren“
- Beispiele
 - Sicherheit:
 1. Partielle Korrektheit: $at_l_0 \wedge \phi \Rightarrow G(at_l_k \Rightarrow \psi)$
 2. Gegenseitiger Ausschluss: $G(\neg(atCS_1 \wedge atCS_2))$
 3. Verklemmungsfreiheit: $G(enabled_1 \vee \dots \vee enabled_m)$
 - Lebendigkeit:

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laickner 7

Rekapitulation

→ Eigenschaften für Korrektheit

- Beispiele
 - Lebendigkeit:
 1. Grundlegende Lebendigkeit: $Fp, F p$ oder $\overset{\infty}{G} p$
 2. Temporale Implikation: $G(p \Rightarrow Fq)$
 3. Intermittierende Aussage: $G((at_l \wedge \phi) \Rightarrow F(at_l' \wedge \phi'))$
 4. Totale Korrektheit: $at_l_0 \wedge \phi \Rightarrow F(at_l_k \wedge \psi)$
 5. Garantierter Zugang: $G(atTry_i \Rightarrow F atCS_i)$
 6. Allg. gar. Zugang: $G(at_l \Rightarrow F at_l')$
 7. Beantwortung: $G(req_i \Rightarrow F grant_i)$
 8. Fairness, Abwesenheit von unverlangten Antworten,...

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laickner 8

Verifikation verteilter Systeme

→ Beweistheoretischer-Ansatz

- Nachteile
 - Komplex
 - Fehleranfällig
- Vorteile
 - Menschliche Intuition
- Beweissystem (Manna & Pnueli)
 - Beweis temporaler Formeln
 - Domänen
 - Programmbeweise

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laickner 9

Verifikation verteilter Systeme

→ Beweistheoretischer-Ansatz: Regeln

- $A_1 \dots A_n$ sind Voraussetzungen und B ist die Schlussfolgerung

$$\frac{A_1 \dots A_n}{B}$$
- Die Invarianzregel (INVAR) ist ausreichend um die meisten Sicherheitseigenschaften zu beweisen

$$\frac{\phi \quad G\{\phi \Rightarrow X\phi\}}{G\phi}$$
 - Diese Regel ist eine Induktionsregel

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laickner 10

Verifikation verteilter Systeme

→ Beweistheoretischer-Ansatz: Beispiel am Peterson Algorithmus

```

flag[0] = 0
flag[1] = 0
turn = 0

L:
flag[0] = 1
turn = 1
while( flag[1] && turn == 1 );
// do nothing
// critical section
...
// end of critical section
flag[0] = 0

M:
flag[1] = 1
turn = 0
while( flag[0] && turn == 0 );
// do nothing
// critical section
...
// end of critical section
flag[1] = 0
  
```

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laickner 11

Verifikation verteilter Systeme

→ Beweistheoretischer-Ansatz: Beispiel am Peterson Algorithmus

- Zustände eines Prozesses:
 - 0: unkritisch (NCS)
 - 1: versuch (TRY)
 - 2: kritisch (CS)
- Invarianten:
 - $y_i = \text{wahr}$: Prozess i möchte kritisch werden oder ist es bereits
 - $y_i = \text{falsch}$: Prozess ist nicht kritisch und möchte es nicht werden
 - $t = \text{wahr}$: Prozess 1 hat die Initiative
 - $t = \text{falsch}$: Prozess 2 hat die Initiative

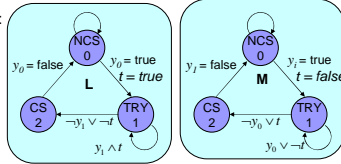
Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laickner 12

Verifikation verteilter Systeme

→ Beweistheoretischer-Ansatz: INVAR

Ausgangszustand:

- L: 0 (NCS)
- M: 0 (NCS)
- y1: false
- y2: false
- t: null



$G\phi_1, \phi_1: y_1 \equiv atm_1 \vee atm_2$

$G\psi_1, \psi_1: y_2 \equiv atm_1 \vee atm_2$

$G\phi_2, \phi_2: atm_2 \wedge atm_1 \Rightarrow t$

$G\psi_2, \psi_2: atm_2 \wedge atm_1 \Rightarrow \neg t$

$G\phi, \phi: \neg(atm_2 \wedge atm_2)$

$\frac{G(\phi \Rightarrow X\phi)}{G\phi}$ $\frac{G(\phi_1 \Rightarrow X\phi_1)}{G\phi_1}$ $\frac{G(\phi_2 \Rightarrow X\phi_2)}{G\phi_2}$
 Analog $G\psi_1$ Analog $G\psi_2$

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laackner

13

Verifikation verteilter Systeme

→ Beweistheoretischer-Ansatz: LIVE

- Möglichkeit für einen hilfreichen Prozess
- Der nächste ausgeführte Schritt in Prozess
 P_k ist p: $X_k p : enabled_k \Rightarrow (executed_k \Rightarrow Xp)$

$G(\phi \Rightarrow X(\phi \vee \psi))$

$G(\phi \Rightarrow X_k \psi)$

$\frac{G(\phi \Rightarrow \psi \vee enabled_k)}{G(\phi \Rightarrow F\psi)}$

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laackner

14

Verifikation verteilter Systeme

→ Beweistheoretischer-Ansatz: CHAIN

- In einigen Fällen ist es erforderlich LIVE-Formeln zu verknüpfen: $G(\phi_i \Rightarrow F(\bigvee_{j < i} \phi_j \vee \psi))$
- Notwendig um „Garantierten Zugang“ zugewährleisten:

$G(\bigvee_{i < j} \phi_i \Rightarrow F\psi)$

$\phi_2: atm_1$

$\phi_4: atm_1 \wedge atm_1 \wedge t$

$\phi_3: atm_1 \wedge atm_2 \wedge t$

$\phi_2: atm_1 \wedge atm_0 \wedge t$

$\phi_1: atm_1 \wedge atm_1 \wedge \neg t$

$\psi: atm_2$

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laackner

15

Verifikation verteilter Systeme

→ Maschinelle Synthese

- Ansatz: Automatische Überführung von High-Level Spezifikation zu Temporaler Logik
- Ignoriert Programmteile, die nicht zur Synchronisation beitragen
- Nutzung von Entscheidungsverfahren
- Konstruktion eines endlichen Modells
- Exponentielle Komplexität

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laackner

16

Verifikation verteilter Systeme

- Vielen Dank für Aufmerksamkeit

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laackner

17

Quellennachweis

- E. Allen Emerson. *Temporal and Modal Logic*
- Manna Z and Pnueli A. *Verification of Concurrent Programs A Temporal Proof System*
- Peterson G L. *Myths about the Mutual Exclusion Problem*

Seminar Systementwurf
Verifikation verteilter Systeme
Harmut Laackner

18