Analyse und Verifikation

LVA 185.276, VU 2.0, ECTS 3.0
SS 2016

– Vorbesprechung –

(Stand: 02.03.2016)

Jens Knoop

Technische Universität Wien
Institut für Computersprachen

Software-Abhängigkeit unseres Lebensstils

Unser tägliches Leben, unser Lebensstil, hängt zunehmend vom korrekten Funktionieren von Software ab!

- Personal Computers, Mobile Phones, Smart Watches,...
  - Online banking, Elektronische Gesundheitsakte, Handels- und Buchungsplattformen (Bücher, Veranstaltungstickets, Reise- und Hotelbuchungen, Gebrauchtwagen, Immobilien, etc.), Social Media, Navigationsssyteme, etc.

Software-Abhängigkeit: Mittel- & unmittelbar

Software ist heutzutage unverzichtbar zur Steuerung

- sicherheitskritischer Anwendungen und Systeme mit unmittelbaren Auswirkungen und Verantwortung für
  - Leib und Leben (Medizintechnik (Operationsroboter, Bestrahlungsgeräte), Luftfahrzeug- und Automobilbau (fly/drive-by-wire, führerlose autonome Fahrzeuge wie Drohnen und Automobile, ABS, ESP, Airbag), Schienenfahrzeugbau (Lokomotiven, Stellwerke), (Industrie-) Anlagensteuerung (Chemieanlagen), Infrastruktur zur Daseinsvorsorge (Kraftwerke, Strom- und Telefonnetze),...)
  - Hohe Sach- und Vermögenswerte (Unternehmens-IT, Finanzindustrie, "unser" Bankkonto, ...)

Mit Fug und Recht

Es ist nicht nur

- unser tägliches Leben, unser Lebensstil

der zunehmend von der korrekten Funktionsweise von Software abhängt!

Mit Fug und Recht

Es ist nicht nur

- unser tägliches Leben, unser Lebensstil

der zunehmend von der korrekten Funktionsweise von Software abhängt!

Es ist

- buchstäblich unser (tägliches) Überleben!

Mit Fug und Recht

Es ist nicht nur

- unser tägliches Leben, unser Lebensstil

der zunehmend von der korrekten Funktionsweise von Software abhängt!

Es ist

- buchstäblich unser (tägliches) Überleben!

Gleichzeitig werden sicherheitskritische Anwendungen nicht nur immer

- allgegenwärtiger

sondern auch immer

- komplexer!

Konsequenz

Daraus folgt:

- Testen alleine zur Qualitätssicherung nicht ausreichend.
- Formale Methoden zur Programmanalyse und -verifikation unverzichtbar!

Einige weit bekanntgewordene Beispiele

...in Stichworten:

- Pentium-Bug
- Ariane-Absturz
- Toyota-Prius
- Mars-Sonde Pathfinder
- ...

$\rightsquigarrow$ mehr dazu siehe 1. Aufgabenblatt!

Beobachtung

Der Einsatz formaler Methoden zur

- Analyse, Verifikation und Transformation (Optimierung)

von Programmen und Programmsystemen wird

- in vielen Bereichen der Industrie (Software-Industrie, Luftfahrt- und Automobilindustrie,...) zunehmend üblich und selbstverständlich.
- Grundlage auch für neue Geschäftsmodelle und Firmengründungen; (gerade auch) aus Universitäten heraus (Coverity Inc. (USA), AbsInt Angewandte Informatik GmbH (DE), Symtavision GmbH (DE), Rapita Systems Ltd. (UK),...).

Zwei Referenzen zum Beleg und Einstieg 1(2)

"Why formal methods and verification are ready to become mainstream applications..."

- Steve P. Miller, Michael W. Whalen, Darren D. Cofer. Software Model Checking Takes Off. Communications of the ACM 53(2):58-64, 2010.

  "Although formal methods have been used in the development of safety- and security-critical systems for years, they have not yet achieved widespread industrial use in software or systems engineering. However, two important trends are making the industrial use of formal methods practical [...]
  The second is the growing power of formal verification tools, particularly model checkers."

Zwei Referenzen zum Beleg und Einstieg 2(2)

"How Coverity built a bug-finding tool, and a business, around the unlimited supply of bugs in software systems..."

- Al Bessey, Ken Block, Ben Chelf, Andy Chou, Bryan Fulton, Seth Hallem, Charles Henri-Gros, Asya Kamsky, Scott McPeak, Dawson Engler. A Few Billion Lines of Code Later: Using Static Analysis to Find Bugs in the Real World. Communications of the ACM 53(2):66-75, 2010.

  "In 2002, Coverity commercialized a research static bug finding tool. [...] We built our tool to find generic errors (such as memory corruption and data races) and system-specific or interface-specific violations (such as violations of function- ordering constraints. The tool, like all static bug finders, leveraged the fact that programming rules often map clearly to source code; thus static inspection can find many of their violations."

Inhalte und Fokus der Lehrveranstaltung

Bedeutung von Korrektheit, Vollständigkeit, Optimalität in

- Analyse
  Datenflussanalyse, abstrakte Interpretation, Modellprüfung, symbolische Methoden,...
- Verifikation
  Beweiskalküle, Methode von Hoare, Korrektheit, Vollständigkeit, stärkste Nachbedingungen, schwächste Vorbedingungen,...
- Transformation
  Programm"verbesserung" (Optimierung),...

von Programmen und Programmsystemen.

Ziele der Lehrveranstaltung

- Vertiefte Einsicht in fundamentale Prinzipien und Konzepte in Programmanalyse, -verifikation und -transformation (-optimierung).
- Herausarbeiten und Verstehen von Gemeinsamkeiten, Analogien und Unterschieden zwischen Programm-analyse und -verifikation.
- Erkennen, Einschätzen und Bewerten der Möglichkeiten und Grenzen insbesondere automatischer Programm-analyse, -verifikation und -optimierung.

Was Sie mitbringen sollten...

Voraussetzungen

- Grundlagen in Theoretischer Informatik, grundlegende Programmierkenntnisse
- Kenntnisse im Übersetzerbau, etwa aus der LVA Übersetzerbau 185.A48 VU 4.0 oder einer vergleichbaren Veranstaltung) sind hilfreich, wenn auch nicht zwingend erforderlich
- Abgeschlossenes Bachelor-Studium

Eine perfekte (Grundlagen-) Ergänzung

...und Vertiefung in diesem Semester durch Mitbesuch von:

- LVA 185.A48 Übersetzerbau, VU 4.0, ECTS 6.0, Prof. Dr. Anton Ertl, Prof. Dr. Andreas Krall: www.complang.tuwien.ac.at/ubvl/index.html
- LVA 185.A49 Abstrakte Maschinen, UE 2.0, ECTS 3.0, Prof. Dr. Andreas Krall: www.complang.tuwien.ac.at/ubvl/index.html
- LVA 185.A50 Dynamische Übersetzer, VU 2.0, ECTS 3.0, Prof. Dr. Andreas Krall: www.complang.tuwien.ac.at/ubvl/index.html
- LVA 185.A64 Übersetzer für parallele Systeme, VU 2.0, ECTS 3.0, Dr. Hans Moritsch: www.complang.tuwien.ac.at/ubvl/index.html

Anrechenbarkeit

...non scholae, sed vitae discimus.

Anrechenbar für die Master-Studiengänge:

- 066 931 Computational Intelligence
- 066 937 Software Engineering & Internet Computing

Ablauf und Beurteilung der Lehrveranstaltung

- Vorlesungsteil (i.d.R. wöchentlich)
- Übungsteil in 2er-Gruppen (i.d.R. wöchentlich)
- Mündliche Abschlussprüfung (über Vorlesungs- und Übungsstoff)

Literaturhinweise

- Janusz Laski, William Stanley. *Software Verification and Analysis*. Springer-V., 2009.
- Hanne Riis Nielson, Flemming Nielson. *Semantics with Applications: An Appetizer*. Springer-V., 2007.
- Krzysztof R. Apt, Frank S. de Boer, Ernst-Rüdiger Olderog. *Verification of Sequential and Concurrent Programs*. 3. Auflage, Springer-V., 2009.
- Flemming Nielson, Hanne Riis Nielson, Chris Hankin. *Principles of Program Analysis*. 2. Auflage, Springer-V., 2005.
- Stephen S. Muchnick. *Advanced Compiler Design Implementation*. Morgan Kaufman Publishers, 1997.

...weitere Literaturhinweise, insbesondere auf Originalarbeiten, werden im Verlauf der Veranstaltung angegeben.

Anmeldung

Zweistufig:

- 1. Stufe: Individuelle Anmeldung
  ...jeder für sich über ein arbeitsbereichsinternes elektronisches Anmeldesystem bis Freitag, den 11.03.2016:
  www.complang.tuwien.ac.at/anmeldung
- 2. Stufe: Gruppenbildung
  ...mithilfe desselben Anmeldesystems: eines der Gruppenmitglieder kreiert die Gruppe

Siehe auch Angaben zur Anmeldung auf der Webseite der LVA:
www.complang.tuwien.ac.at/knoop/auv185276_ss2016.html

Zeit und Ort für Vorlesung und Übung

- Dienstag, 16:30 - 18:00 Uhr s.t., Seminarraum 125, Elektrot. Institutsg., Gußhausstr. 25-29, Stiege 1, 2. Stock, 1040 Wien
  - 16:30 Uhr - bis ca. 17:30 Uhr: Vorlesung
  - ab ca. 17:30 Uhr - 18:00 Uhr: Übung (Abgabe und Besprechung der Übungsaufgaben der Vorwoche, Ausgabe der neuen Übungsaufgaben (im Web erhältlich)).
- Im Regelfall jeden Dienstag, beginnend mit dem 08.03.2016, ein neues Aufgabenblatt; insgesamt ca. 8 Abgaben.

Bei Fragen und Problemen

- Webseite der Lehrveranstaltung:
  `www.complang.tuwien.ac.at/knoop/auv185276_ss2016.html`
- Fragen und Antworten in Vorlesung und Übung
- Sprechstunde: Mittwochs, 15 Uhr - 16 Uhr
  (bitte vorher kurz anmelden)
- E-Mail: `knoop@complang.tuwien.ac.at`

Interesse an gefördertem Auslandsstudium?

Die Erasmus/LLP-Programmlinie der EU bietet eine Vielzahl lohnender Möglichkeiten, z.B.

- Linköping University, Schweden
- Aalto University, Finnland
- The University of Copenhagen, Dänemark
- Universität Halle-Wittenberg, Deutschland
- Universität Paderborn, Deutschland
- Universidad Politècnica de València, Spanien
- ...

Mehr dazu: `www.complang.tuwien.ac.at/knoop/erasmus`

Ich wünsche Ihnen

...viel Erfolg bei dieser Lehrveranstaltung und dass Sie auch über die unmittelbare Veranstaltung hinaus davon profitieren!

Nicht zuletzt:
Vorlesung und Übung leben mit Ihnen! Ihre Rückmeldungen, Anregungen, Verbesserungsvorschläge sind willkommen!
Natürlich auch Hinweise, wenn Ihnen etwas gut gefallen hat!