Analyse und Verifikation

LVA 185.276, VU 2.0, ECTS 3.0
SS 2018

– Vorbesprechung –

(Stand: 08.03.2018)

Jens Knoop

Technische Universität Wien
Information Systems Engineering
Compilers and Languages

Software-Abhängigkeit: Allumfassend (1)

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

- Handels- und Buchungsplattformen (Bücher, Veranstaltungstickets, Reise- und Hotelbuchungen, Autos, Immobilien, etc.)
- Social Media
- Online banking
- Navigationssyteme
- Elektronische Gesundheitsakte
- ...

Software-Abhängigkeit: Allumfassend (2)

Software ist heutzutage unverzichtbar auch 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, ...)

Software-Abhängigkeit: Allumfassend (3)

...weit mehr als nur

- ein Lebensstil

ist es buchstäblich

- unser (tägliches) Überleben!

das zunehmend von der korrekten Funktionsweise von Software abhängt!

Zugleich wird Software (auch sicherheitskritische) nicht nur immer allgegenwärtiger, sondern auch immer komplexer!

Qualitäts-, insbes. Korrektheitssicherung (1)

Testen alleine nicht ausreichend zur stringenten

- Qualitäts-, insbesondere Korrektheitssicherung.

Einige weit bekanntgewordene Fehlerbeispiele aus Hard- und Software:

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

$\rightsquigarrow$ siehe dazu auch Aufgabenblatt 1!

Qualitäts-, insbes. Korrektheitssicherung (2)

Formale Analyse- und Verifikationsmethoden zur

- Qualitäts-, insbesondere Korrektheitssicherung unverzichtbar!

Tatsächlich wird der Einsatz formaler Methoden zur

- Analyse, Verifikation und Transformation/Optimierung

von Software in vielen Bereichen (Software-Industrie, Luftfahrt-, Automobilindustrie,...) zunehmend

- üblich und selbstverständlich.
- ist 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 Einstieg (1)

"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 Einstieg (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 Software, d.h. von Programmen und Programmsystemen.

Ziele der Lehrveranstaltung

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

Voraussetzungen, Anrechenbarkeit

Voraussetzungen

- Abgeschlossenes Bachelor-Studium.
- Grundlagen in Theoretischer Informatik, grundlegende Programmierkenntnisse.
- Kenntnisse im Übersetzerbau, etwa aus LVA 185.A48 Übersetzerbau VU 4.0 und LVA 185.A04 Optimierende Compiler VU 2.0 oder einer vergleichbaren Veranstaltung sind hilfreich, wenn auch nicht zwingend erfor- derlich.

Anrechenbarkeit für die Master-Studiengänge:

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

Anmeldung, Ablauf und Beurteilung

Anmeldung

- Via TISS bis 16. März 2018, in 2er-Gruppen (in Ausnahmefällen in 1er- oder 3er-Gruppen)
- *Abmeldung:* Via TISS bis 30. März 2018.

Ablauf und Beurteilung

- Vorlesungsteil (i.d.R. wöchentlich)
- Übungsteil in 2er-Gruppen (i.d.R. wöchentlich)
- Mündliche Abschlussprüfung (über Vorlesungs- und Übungsstoff; Übungs- und Prüfungsteil müssen beide positiv sein)

Zeit und Ort für Vorlesung und Übung

- Ab Mittwoch, den 07.03.2018: 16:15 - 17:45 Uhr, Hörsaal EI3a, Elektrot. Institutsg., Gußhausstr. 25-29, 2. Stock, 1040 Wien
  - Besprechung der Übungsaufgaben der Vorwoche sowie der neuen Aufgaben im Regelfall zu Anfang bzw. Ende der Vorlesungseinheiten.
- Im Regelfall mittwochs, beginnend mit dem 07.03.2018 (Abgabefrist für Aufgabenblatt 1 zwei Wochen), ein neues Aufgabenblatt (im Web erhältlich); insgesamt ca. 8 Aufgabenblätter.

Vorlesungsmaterialien, Aufgaben, Termine

Vorlesungsmaterialien, Aufgaben, Termine

- Webseite der Lehrveranstaltung:

`www.complang.tuwien.ac.at/knoop/auv185276_ss2018.html`

Fragen, Probleme

- Vorlesung, Übung
- Sprechstunde (Mittwochs, 15 Uhr - 16 Uhr, bitte vorher kurz anmelden)
- Elektronisch (`knoop@complang.tuwien.ac.at`)

Leseempfehlungen

- 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*. Springer-V., 3. Auflage, 2009.
- Flemming Nielson, Hanne Riis Nielson, Chris Hankin. *Principles of Program Analysis*. Springer-V., 2. Auflage, 2005.
- Stephen S. Muchnick. *Advanced Compiler Design Implementation*. Morgan Kaufman Publishers, 1997.

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

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

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!