Friedrich-Schiller Universität, Fakultät für Mathematik und Informatik

Lehrstuhl Programmiersprachen und Compiler

Automatische Programmverifikation

Seminar im SS 2003 Mi 16.30 – 18.00 CZ3 SR 129

Vorträge

Einführung:	Seminarteilnahme Einführung in das Thema		Jürgen Winkler	10. 4.
Zusicherungen, Programmzustände		Fut 89: 3, 4	Mathias Goldau	24. 4
Spezifizieren von Programmen		Fut 89: 5	Juliana Kreissig	8. 5.
Verifikationsregeln		Fut 89: 6	Lutz Kohl	14. 5.
Weakest precondition 1 Zuw, Sequ, IF		Fut 89: 8.1, 8.2.1-5	Thomas Puhl	21. 5.
Weakest precondition 2 WHILE, FOR		Fut 89: 8.2.6, Kau 2000: 3.4.2	Georg Laube	28. 5.
Weakest precondition 3 Proz-Aufruf		Fut 89: 10, Gri 83	Carsten Freining	4. 6.
Beispiele 1: Sortieren und Suchen		Fut 89: 9.1, 9.2	Heiko, Stark	11.6.
FPP, NPPV, SPARK		FKW 2002, Folien Win 2002 (s. psc.informatik ...)	Jürgen Winkler	18. 6.

Die automatische Verifikation von Programmen wird in Zukunft immer wichtiger werden, da

- die Anzahl der sicherheitskritischen Programme stetig zunimmt:
  Einsatz in Auto, in medizinischen Geräten, im Flugzeug
- generell die Qualitätsansprüche an die SW steigen werden, wie das bei anderen Produkten wie technischen Geräten oder Medikamenten ebenso der Fall ist.

Das MIT zählt die SW-Verifikation zu den *10 wichtigsten Zukunftstechnologien*
(http://www.technologyreview.com/articles/emerging0203.asp?p=0)

Hoare sieht einen automatischen Programmbeweiser als eine „Grand Challenge“ J. ACM 50,1 (2003) 63 .. 69 und LNCS 2622, S.262 .. 272

Viele Werkzeuge zu automatischen Verifikation beruhen auf dem wp-Kalkül, in welchem die Semantik der einzelnen Sprachkonstrukte mittels der weakest precondition (schwächste Vorbedingung) definiert wird.

Bei der Anwendung dieses Verfahrens spezifiziert der Programmkonstrukteur das Programm in anschaulicher Weise durch die Angabe von Vor- und Nachbedingungen. Der Programmbeweiser berechnet dann daraus die Verifikationsbedingungen und versucht diese zu beweisen.

Das Seminar gibt eine Einführung in die Grundlagen der wp-Semantik und behandelt Beispiele von Programmbeweisern.

Hinweise zur Vortragsgestaltung finden sich unter
psc.informatik.uni-jena.de/lehre/sem/good-presentation.htm

Literatur

FKW 2002	Freining / Kauer / Winkler: Ein Vergleich der Programmbeweiser FPP, NPPV und SPARK. Ada Deutschland Tagung 2002, Jena, 127..146
Fut 89	Futschek, Gerald: Programmentwicklung und Verifikation. Springer, Wien usw., 1989
Gri 83	Gries, David: The Science of Programming. Springer, New York etc. 1983
Kau 2000	Kauer, Stefan: Automatische Erzeugung von Verifikations- und Falsifikationsbedingungen sequentieller Programme. Diss., FSU Jena, 2000
Luc 90	Luckham, David: Programming with Specifications. Springer, New York etc., 1990.
Mey 92	Meyer, Bertrand: Eiffel – The Language -. Prentice Hall, New York etc., 1992
Win 2002	Winkler, J F H: Folien zu Vergleich FPP, NPPV und SPARK (psc.informatik.uni-jena.de/FPP/Vo-FPP-0210.pdf)

J F H Winkler, 2003.Mai.12

Seminarteilnahme

1) Vortrag halten

Ziel: Üben eines verständlichen und die Hörer ansprechenden Vortrages. (z.B. auch Begrüssung)

Vortrag ausarbeiten, Vortragsdesign: Gliederung, Zeitbedarf, Üben

Hilfsmittel: Sprache, Schrift, Bild usw.

Vortragsinhalt: muß vom Vortragenden bestens verstanden sein.

2) Vorträge besprechen (positiv und negativ)

Nach Form, Stil und Verständlichkeit

Nach Inhalt, d.h. fachliche Diskussion

3) Schriftliche Ausarbeitung

Titel, Seitennr., usw.

Soll für kompetenten Nichtteilnehmer Teilnahme “ersetzen“ können

4) Ein Thema vertieft kennenlernen

=> „How to make a good presentation“

psc.informatik.uni-jena.de/lehre/sem/good-presentation.htm