Diplomprüfung Theoretische Informatik 12.10.98

Kurs Programmverifikation 1682 (Buch Francez)

Dauer 25 Min

Prüfer: Dr. Heinemann

Wie ist die Semantik von Programmen definiert? Wie sieht sie konktet bei der "Goto"- Programmiersprache aus ?

Nennen Sie Beispiele für externe/interne Variablenbelegung und syntaktischer Fortsetzung.

Definition einer Spezifikation ?

Was versteht man unter partieller Korrektheit (Definition aufschreiben) ?

Beschreiben sie die Methode F. Wie sieht es hinsichtlich der Korrektheit und Vollständigkeit aus ?

Was bedeuten diese Begriffe ?

Geben sie die Syntax der PLW an. Wie ist diese Definition aufgebaut ?

Angabe der Semantik der PLW. Was ist eine Struktur ?

Sei S= (T,Q,G), was genau sind Q und G ?

Was ist eine Belegung ? Wie sind W und WW definiert (genau)?

Hoare-Kalkül : Was wird damit gezeigt ? Regel (SEQ) : Kann sie immer benutzt werden ? Warum ?

Wie kommt man an die Zwischenprädikate ?

Relative Vollständigkeit von H* : Was versteht man unter relativer Vollständigkeit ? ("post" aufschreiben).

Kurzer Anriß von RVL

DIPLOM-PRÜFUNG Theoretische Informatik, 27.3.95

Kurs: **Programmverifikation** (1682 Basis Buch **Francez)**
Dauer 20 min
Prüfer Dr B Heinemann

Die im folgenden aufgeführten Angaben pg. 24 beziehen sich stets auf das Buch von N Francez

Wie ist die operationale Semantik von Programmen definiert ? (pg. 11) Wie sieht das im Fall der PLF-Programme aus ? Wie lauten Syntax und Semantik der PLF-Programme ? (pg.31-33)

Was ist eine Spezifikation ? Hier ist es nicht nur wichtig die Definition von pg. 14 zu kennen, sondern auch zu erläutern, daß die Vor- und Nachbedingung Prädikate darstellen deren Implementation später durch das jeweilige Programm geieistet werden muß.

Was sind Spezifikationsvariable und wofür werden sie benötigt ? (pg. 16) Hauptsächlicher Gebrauch ist dabei die Bewahrung von Startwer-ten.

Was ist {p}S{q} ? Trivial zwar, doch im Eifer des Gefechts verfiel ich in eine groß angelegte Erklärung von Spezifikation und Implementation usw Gewünscht war :
Das ist die symbolische Darstellung für partielle Korrektheit.

Wie beweist man in PLF partielle Korrektheit ? Durch die Mett-rode F. Sie mußte in allen Einzelheiten beschrieben werden. (pg 39). Wichtig Die verification condition muß für jeden basic path nachgewiesen werden,

Was beweist man mit der Methode F' ? Die Terminierung von PLF-Programmen, noch nicht die totale Korrektheit Angabe der Methode F* im Detail (pg 45). Unbedingte Voraussetzung : Wohlfundierte Mengen (pg.45).

Die parametrisierte induktive Zusicherung benötigt das Element der wohlfundierten Menge zur Kontrolle der Schleifen. Wozu ist das Element dann auch zwischen cut points in der verification condition nötig, obwohl dort nicht unbedingt eine Schleife vorliegen muß ? Weil es für den Beweis der semantic completeness als Restanzahl von basic paths dient. (Wußte ich im Zusammenhang mit der Erklärung der Methode F'.)

Angabe der Syntax und Semantik von PLW-Programmen (pg. 69/70)

Angabe des Hoare-Kalküls (pg. 72-75)

Besonderheit beim Beweis der relative completeness des Kalküls H die expressiveness pg. 79 (Beispiel pc. 80)

Besonderheit beim Beweis der arithmetical completeness des Kalküls H* :
die arithmetical extension (pg. 87).

Darstellung der Rekursionsregel REC und der Auswirkung auf die Beweisführung (pg. 155). Sozusagen ein Unterbeweis (die Korrektheit des Rekursionsrumpfes) in der Beweisführung der Korrektheit des Aufrufs.

Syntax und Semantik von SVL. (pg. 1891191). Besonderes Augenmerk dabei auf die await-Regel und Vergleich der Semantik mit der while-Regel.

Erklärung der Begriffe termination/deadlock/diverges In SVL. (pg 193)

Was ist interference free und wo wird dieser Begriff benötigt ? Definition angegeben (pg. 195) und Verwendung in der Regel SCONC. die kurz erörtert wurde

Abschließend wurde noch auf die auxiliary variables und deren Verwendung eingegangen

Die Prüfung verlief in einer guten Atmosphäre. Dr. Heinemann ist aus meiner Sicht uneingeschränkt als Prüfer weiter zu empfehlen. Der Kurs verlangt eine andere Arbeitsweise hinsichtlich der Erarbeitung der Inhalte, nicht nur der Sprache wegen Die sonst gewohnten starken Formalismen bei anderen Kurse, wie Logik oder Berechenbarkelt, treten in den Hintergrund, wohingegen man Eigeninitiative für die “persönliche Strukturierung” des Kurses aufbringen muß Aus meiner Sicht dennoch ein guter Schritt in die richtige Richtung für die Theoretische Informatik