Prüfungsprotokoll Vordiplom Praktische Informatik

Kurs 1612 - Konzepte imperativer Programmierung

Prüfer: Prof. Dr. Helbig
Datum: 10.04.2007
Note: 2,0

Tatsächlich beschränkten sich die Fragen genau auf den Katalog, der aus den anderen Prüfungsberichten bekannt sind:
Pointerkonzept an Beispielen erläutert, Zeiger an einem Beispiel umgehängt, Deklaration für einen Bin. Baum, Übergabearten bei Funktionen und Prozeduren (hier bestand ich darauf, das "var" nach Skript nicht für Funktionen vorgesehen ist, was er zu honorieren schien.).
Wann braucht man globale Variablen, obwohl man diese doch nicht nutzen sollte? Hier wußt ich die Antwort nicht - ich glaube er wollte dann Felder hören und ich erläuterte, warum diese ausnahmsweise per Referenz übergeben werden: Speicher und Zeitersparnis.
Welche Schleife kann man außer der for-Schleife noch für Iterationen verwenden. Ich erläuterte dann die Konstrukte von while und repeat . Sollte diese ineinander umwandeln.
Hoare-Kalkül: Was denn oben und unten stehen würde: Prämisse und Konklusion! Und? Programmformeln! Ja was steht denn da genau? Wusste nicht was er wollte. Er wollte H1,..., Hn als Prämisse sehen und H als Konklusion. Außerdem wollte genau wissen was Prädikate seien und was diese in der Programmformel bedeuten, wobei er sich nicht mit Vor- und Nachbedingung zufrieden gab.
Syntaxdiagramm einer PASCAL-Fkt.: Komplett hingeschrieben von *function declaration* bis *formal parameter list*. Iterative und rekursive Funktion für die Fibonacci-Zahlen hinschreiben. -ging alles ruckzuck. Dabei wollte er genau erläutert haben wann iterative und wann rekursive Lösungen besser sind: Rekursive Datenstrukturen, Lineares Problem, Hilfsstapel nötig oder nicht...
Kontextfreie Grammatik: 4-Tupel...
Schrieb Ersetzungsregel so hin „<Nichtterminal> ::= $w_1|w_2...w_n$“. Er war schein`s relativ zufrieden damit. Dann malte ich noch unsicher die 2 Striche dazu: „<Nichtterminal> ::= $w_1|w_2|...|w_n$“ (so wie es im Skript steht) und bei Ihm ging „eine Rakete los“. Es sei nun falsch – der Beisitzer nickte das ab - und ich hätte wohl nicht so viel davon verstanden.
Also das ist leider schon sehr gravierend, wenn ein Prüfer den richtigen Ausdruck für falsch hält. Dazu meinte er noch "<Nichtterminal>“ hätte doch nichts mit der Menge N zu tun. Ich sagte es sei doch ein Element der Menge. Das stimmte ihn auch nicht zufrieden und ich solle mir doch darüber klar werden, dass da auch „xyz“ stehen könnte. An diesem Punkt hielt er sich immer wieder mit denselben Erklärungen auf. Ich glaube auch die Antwort, dass die w_i aus der Vereinigungsmenge T und N Stern sei stimmte ihn nicht zufrieden bzw. ruhig (:-). Vielleicht wollte er auf die paarweise Disjunktheit hinaus.
Tipp: Man sollte sich nur auf seinen Fragenkatalog einschießen, aber alles 100% draufhaben, denn sonst wird man möglicherweise verunsichert. Das heißt, dass man mit relativ wenig Aufwand eine 2 machen kann. Für bessere Noten braucht man entweder Glück, dass er nicht irgendwo hängen bleibt und/oder die Fähigkeit die Dinge wirklich in einem sehr allg. Zusammenhang zu verstehen so, dass man mit jeder seiner Fragen etwas anfangen kann.
Sicherlich habe ich den Fehler gemacht mich auf Diskussionen einzulassen, die nur wertvolle Zeit gekostet haben und der Prüfungsatmosphäre auch nicht zuträglich waren. Im Umgang war er allerdings stets freundlich.

Vordimplomsprüfung in Kurs 01612: “Konzepte imperativer Programmierung”
Datum: 09.08.2007
Prüfer: Prof. Dr. Hermann Helbig
Beisitzer: Dr. Rainer Osswald
Themen: Syntax, Syntaxdiagramme, kontextfreie Grammatik,
Parameterübergabe bei Prozeduren, Funktionen, Zeigerdefinition, Schleifen
Zeigeroperationen, Rekursion und Iteration, Verifikation nach Hoare

Note: 1,3

Fragen
-Antworten

Was verstehen Sie unter einer syntaktischen Struktur bei einer Programmiersprache?
-siehe Kurstext unter: “Syntax”

Wie lässt sich eine solche Syntax definieren?
-Durch eine Grammatik, genauergesagt : kontexfreie Grammatik, oder Syntaxdiagramme

Was für Grammatiken kennen sie noch?
-verschiedene Typen aus der Chomskyhierarchie, Grammatiken für endliche/reguläre Sprachen, kontextsensitive ... (ist allerdings nicht prüfungsrelevant, Prof. Helbig wollte hier wahrscheinlich einfach erfahren, wie weit man aus dem Tellerrand schauen konnte)

Was ist dann das besondere bei den kontextfreien Grammatiken?
-Formale Definition einer Grammatik angeben, dann die Produktionseigenschaft der kontextfreien
Grammatik erwähnen. N -> (N vereinigt T)*
-N sind Nichterminale, T sind Terminale

Was gilt für N und T ?
-Elemente aus T werden ausschließlich aus Elementen von N abgeleitet

Was gilt aber überhaupt vorher, viel einfacher erstmal?
- (nach langem Überlegen)
Für jedes Element a aus N gilt, a istNichtElement von T
->Prof. Helbig wollte dazu jedoch die mathematisch kompakte Mengenschreibweise:
N geschnitten T = leere Menge,
Es war nicht so, dass ich diese elementare Schreibweise nicht kannte,
nur wusste ich in der Prüfunssituation nicht, was er genau von mir wollte,
ansonsten steht dies aber auch im Kurstext unter
“Kontextfreien Grammatik”, dort am besten auch die Mengenrelationen zwischen
N,P und T,P ansehen.

Schreiben Sie das Syntaxdiagramm einer Funktion auf
- siehe Kurstext: “Funktionen”,
Darauf achten, dass bei Funktionen “in”-Paramer per call by value übergeben werden

Was ist call by value/reference?
-siehe Kurstext
An was für eine Variablennutzung erinnert die Wirkung der Übergabe per call by reference?
-globale Variablen
Wozu kann das bei all den Seiteneffekten trotzdem gut sein?
-gemeinsame Datennutzung ohne ständige Übergabe
Nee, eher um z.b die Duplizierung von großen übergebenen Arrays zu verhindern
-(hätte er die Frage etwas direkter im Bezug auf call by reference und nicht globale Variablen gestellt, hätte ich ihm diese Antwort auch geben können...)

Definieren sie einen Zeigertyp
-Siehe Kurstext, Beispiele zur Referenzzuweisung und Wertezuweisung von Zeigervariablen wurden ebenfalls verlangt

Schreiben Sie die rekursive Variante der Finonacci-Funktion
-Zunächst musste ich die mathematische Funktionsdefinition aufschreiben,dann die Funktion. Steht alles im Exkurskapitel drin, jedoch hatte Prof. Helbig mich darauf hingewiesen bei Bedarf eine andere Funktion zu stellen, da die Fibonacci-Zahlen zum Exkurs des Kurses gehörten.

Was sind die Vorteile der Rekursion?:
-gute Lesbarkeit, geringe Programmkomplexität
Nachteile?
-unnötig wiederholende Funktionsaufrufe,
für F(n) = 1 Aufruf
F(n-1)=1
F(n-2)=2
F(n-3)=3
F(n-4)=5
F(1) =Fib(n) Aufrufe
bei einer iterativen Schleifenvariante führt man die Rechnungen für die Aufrufe nur je einmal durch.
OK!
Was für Schleifenkonstrukte kennen Sie?
-for, while, repeat until, zu den letzten beiden die Unabhängigkeit voneinander erläutern, und aufschreiben, wie eine while durch eine repeat until Schleife ersetzt wird

Wie sieht eine Regel im Hoare-Kalkül aus?
-Folgerungspfeil, drüber: Prämissen aus Programmformeln
drunter: Konklusionsprogrammformel, Beispiel für if-then-else-Regel angeben.

Schreiben sie mal eine Programmformel auf
-Beispiel: {P} S {Q}
Was genau steht in P?
-Prädikatenlogischer Ausdruck
Konkreter?
-Bedingungen für Eingabewerte, dementsprechend Q die Bedingungen der Eingabewerte nach Ausführung von S
Wie geht man letzlich bei der Verifikation mit einer Inferenzregel vor?
-von unten nach oben, dabei zu jedem Codefragment die entsprechende Ableitungsregel finden
Was macht mal also?
-Wieder eine Frage, wo ich nicht genau wusste, was Prof. Helbig hören wollte, im Prinzip habe ich meine Antwort wiederholt. Er wollte aber hören, dass man dabei die äußere Programmcodestruktur dekompositionieren muss und dann eine rückwärts gerichtete Deduktion vollzieht:
Hat man eine whileschleife: was steht in der Bedingung? In der Bedingung könnte wieder eine Funktion stehen, indieser Funktion wieder eine Schleife usw... bis Code nicht mehr zu zerlegen ist von der äußeren Struktur.
-Auch hier muss ich ehrlich sagen, dass mir dies schon klar war, jedoch wusste ich nicht, dass er es eben genauso hören wollte und nicht allgemeiner..., hier sagte Prof. Helbig übrigens, dass dies noch kein Student richtig in einer Prüfung bei ihm gesagt hat, jetzt weiß ich allerdings auch warum: auf seine Fragen weiß man meist einfach nicht, was er genau hören will, obwohl die Antwort selbst im Kopf ist.

Fazit: Ich kann Prof. Helbig als Prüfer schon empfehlen, seine Fragen sind eigentlich alle zu beantworten, wenn man alle nötigen Fakten aus dem Kurstext gelernt hat. Das Problem bei manchen Fragen von ihm ist, dass er manchmal sehr allgemein fragt, und dann etwas sehr genaues hören möchte, wobei man selbst als Prüfling nicht genau weiß, was er genau hören will. Das, was er wiederum hören will, ist nicht sonderlich schwer zu wissen, jedoch sind seine Fragen bezogen darauf manchmal ein Rätsel für sich. Ansonsten ist er ein netter Prüfer, der die Prüfung gerne sehr schnell, kurz und schmerzlos macht. Es wurden auch bei mir die meisten Prüfungsfragen wie aus den Protokollen gestellt, hier und da mal etwas abgeändert.

Prüfungsprotokoll zur Vordiplomsprüfung
Kurs 1612. Konzepte imperative Programmierung

Prüfer: Prof. Dr. Helbig
Datum: 16.12.2005
Note: 2,3

Hier die Fragen, an die ich mich noch erinnern kann.

Kontextfreie Grammatik:
- 4-Tupel (N,T,P,S) mit
    - N – Nichtterminale
    - T – Terminale
    - P - Regeln
    - S - Startsymbol

Wie sieht eine allgemeine Regel aus:
Für A *aus* N und b *aus* (N u T)* gilt: A=> b

Iterationen nennen und Syntaxdiagramme erstellen:
- for-Schleife
- while-Schleife
- repeat-Schleife

Sind while und repeat unabhängig?
- Ja, man kann die eine Schleife durch die andere ersetzen, was dann auch noch durch Aufschreiben und Erläutern zu zeigen war.

Ausprogrammierung einer rekursiven Funktion zur Berechnung der Fibonacci-Zahlen. Dabei sollten die Eigenschaften sowie Vor- und Nachteile von iterativ und rekursiv erläutert werden.
- Siehe Skript

Zeigerkonzept in Pascal

Parameterübergabe by value und by reference

Wie wird Verifizierung durchgeführt?
- Mit Hilfe der Hoare-Regeln

If-then-else – Anweisung aufschreiben und erläutern.
- Sinnvoll ist es hier die Regel von unten nach oben zu entwickeln, so wie man sie auch zur Verifizierung einsetzt.

Besonders empfehlen kann ich Prof. Helbig nicht. Ich habe fast immer Probleme gehabt zu verstehen, worauf er mit seiner Fragestellung hinaus wollte. Das hat die sowieso schon vorhandene Nervosität immer wieder angetrieben und zu weiteren Unsicherheiten geführt. Da ich bei einer Fragestellung einen kompletten Blackout und auch sonst teilweise Unsicherheiten in der Formulierung hatte, bin ich mit der Benotung aber bestens zufrieden.

Gedächtnisprotokoll zur Diplomvorprüfüng Praktische Informatik

Kurs 1612 - Konzepte imperativer Programmierung (am 13.12.2000)

bei H. Prof. Dr. Helbig

Fragen: Unterschied zwischen Parameterübergabe by value und by reference. Nachfrage, ob die Möglichkeit Übergabe per pointer, d.h. mit In- und Ausgabeparameter, wodurch man Variablen im aufrufenden Block vom aufgerufenen Block aus verändern kann, gut sei. Die Benutzung muß auf jeden Fall diszipliniert, bewußt erfolgen.

Ausprogrammierung einer procedure zur Bestimmung der Fibonacci-Zahlen:

Erstens rekursiv - zweitens iterativ.

Dabei wurden Details aus dem Kurstext verändert,

welche bei der Ausprogrammierung natürlich berücksichtigt werden

mußten.

Was ist das Zeiger-Konzept in PASCAL.

Das Beispiel einer Verzeigerung in einer Linearen Liste ging etwas an der

Frage vorbei.

Welche Iterationen gibt es in PASCAL:

for - Schleife, bei fester Unter- und Obergrenze. while - Schleife, wenn Schleife nicht unbedingt einmal durchlaufen werden muß.

repeat - until, wenn Schleife mindestens einmal zu durchlaufen ist. Sind while-Schleife und repeat-until unabhängig voneinander. Ja, weil eine Schleife durch die andere ersetzt werden kann. Anschreiben, wie while durch repeat-until bzw. umgekehrt ersetzt werden kann.

Aussagenlogik:

Anschreiben der Regeln von de Morgan.

Prädikatenlogik:

Durch Anwendung welcher Regeln wird Verifizierung durchgeführt. Mit Hilfe der Hoare-Regeln.

Anschreiben des Inferenzschemas für die Programmformel „IF THEN... ELSE"
L Erläutern, daß von unten nach oben vorgegangen wird, bei der Verifizierung und im Inferenzschema Programmformeln stehen. Anschreiben einer prädikatenlogischen Formel aus einer natürlichsprachlichen Formulierung heraus. Wie „für alle syntaktisch korrekten Programme gilt, daß sie mit END. enden"..

Prof. Dr. Helbig ist als Prüfer zu empfehlen, besonders fürStudenten, die in der Theorie firm sind, Schwerpunkte hierbei sind Aussagen- und Prädikatenlogik, aber auch die Programmierung muß im Detail verstanden und parat sein, allerdings wurde nicht die Beherrschung dynamischer Datenstrukturen geprüft, großer Wert wurde auf das Verstehen des Vorgehens bei der Verifizierung gelegt. Ich hatte bisweilen Schwierigkeiten gleich zu erkennen, auf was der Prüfer mit seiner Fragestellung hinauswollte.

Prüfung: Vordiplom Praktische Informatik - Konzepte Imperativer Programmierung (Kurs: 1612)
Prüfer: Prof. Helbig
Datum: 24.1.2000
Note: 1.0

Wie wird die Syntax (z.B. einer Programmiersprache) definiert?

- Kontextfreie Grammatik oder Syntaxdiagramme

Definition einer kontextfreien Grammatik?

- Tupel (N, T, P, S) usw. wie im Kurstext.

Geben Sie das Syntaxdiagramm für die Definition einer Funktion in Pascal an.

- Wie im Kurstext. *Hr. Helbig meinte das reicht noch nicht. Was ist denn mit den unterschiedlichen Möglichkeiten der Bindung für die Variablen?*
  Ich fragte nach: „Call by Value / by Ref.?)
  Hr. Helbig: ja, genau
  Ich habe dann erklärt, daß im Kurstext (zumindest soweit ich weiß) für Funktionen nur by Value möglich ist, nur für Prozeduren hat man beide Möglichkeiten. Bei meinem TurboPascal-Kompiler ist allerdings auch für Funktionen By Ref. möglich. Wir einigten uns dann darauf, daß das wahrscheinlich ein Unterschied zwischen Standard- und TurboPascal ist...

Wie lautet die Syntax für Call by Ref.?

- „var“ vor die Variable

Wie wirkt sich das aus?

- Wert der übergeben Variablen kann auch für den aufrufenden Block (Hauptprogramm) verändert werden usw. ...

Wie wird in Pascal ein Zeiger-Typ definert?

- Beispiel gegeben

Seien x und w Zeiger. Was ist der Unterschied zwischen w := x und w^ := x^ ?

- Gleichsetzen der Adressen / gleichsetzen der Inhalte, Bildchen malen...

Wie lautet die Typen-Definition für einen binären Baum?

- Siehe Kurstext

Wie kann man einen binären Baum als Graphen definieren?

- Baum mit max. zwei Nachfolgern pro Knoten. Ein Baum ist ein zykelfreier, zusammenhängender, gerichteter Graph. *Hr. Helbig: das reicht aber noch nicht! Eingangsgrad maximal = 1 !*

Geben Sie die Funktion an, die die Fibbonacci-Zahlen iterativ berechnet.

- Siehe Kurstext, fib2. (Dauerte ganz schön lang, im Vergleich zu den anderen Fragen)

Wie kann man die Semantik eines Programms beschreiben ?

- Programmformel, Def. für Programmformel z.B. {P} S {Q}. mit P, Q Prädikaten über den Variablen des Programms S ... (siehe Kurstext)

Wie sieht (allgemein) eine Regel im Hoare-Kalkül aus ?

- Über dem Strich: Prämisse aus mehreren Programmformel. Unten die Konklusion, die aus der Prämisse folgt.

In welcher „Richtung“ macht man meistens den Beweis eines Programms in der „Praxis“ ?

- In der Praxis von unten nach oben in der Hoare-Regel (man sucht für ein Programm in der Konklusion die passenden Prämissen und teilt dafür das Programm in Stücke auf)

Wo endet das Spiel (= die logische Deduktion) nach „oben“ hin ?

- Bei den Axiomen: Nullaxiom, Zuweisungsaxiom.

Geben Sie die Hoare-Regel für if-then-else an .

- Siehe Kurstext.

Hr. Prof. Helbig ist ein sehr netter Prüfer. Über die Note hab ich mich natürlich riesig gefreut. Schade daß es bisher so wenige Prüfungsprotokolle für ihn gibt – also bitte selber auch eines schreiben, gell !

Vordiplomprüfung
Konzepte Imperativer Programmierung

Prüfer: Prof. Helbig
Termin: 17.05.1999
Dauer: 30 min
Note: 1,7

Das waren die wichtigsten Fragen, an die ich mich noch erinnern kann:

F: Was für Programmiersprachen kennen Sie'? Beispiele?
A: imperativ, objektorientiert usw. siehe Kursunterlagen (Hier habe ich gedacht, er meint niedere/höhere PS, aber er wollte die Programmierparadigmen hören = steigender Streßpegel)

F: Wo tritt bei PASCAL das funktionale Konzept auf?
A: bei den Funktionen (richtig geraten...)

F: Unterschied Funktionen und Prozeduren?
A: Parameterlibergabe etc. siehe Kursunterlagen

F: Unterschied Wertparameter, Referenzparameter?
A: kurze Erkl&ung

F: Wie werden Referenzparameter kenntlich gemacht?
A: Schlüsselwort var in der Liste der formalen Parameter

F: Logik: was sind die algebraischen Eigenschaften der Operatoren'?
A: bei and und or Kommutativitat, Assoziativität, Distributivität. neutrale Elemenle, bei not doppelte Verneinung (Was meint der jetzt schon wieder?! Bis bei mir der Groschen gefallen war, hatte ich schon wieder etliche Gramm Adrenalin im Blut...)

F: Wie kann man 'fur alle x gilt P(x)' durch 'es gibt' ausdrücken?
A: not(not(es gibt ein x mit not P(x))) (einfache Übung zur Beruhigung)

F: Welche Iterationsverfahren gibt es? Zeichnen Sie Syntaxdiagramm.
A: for, while, repeat

F: Sind while und repeat unabhängig?
A: while und repeat können einander ersetzen (vorführen)

F: Wie sieht eine Typdefinition für array und fir record aus?
A: Beispielcode angeben

F: Was ist ein binärer Baum'?
A: Knoten, Kanten, Wurzel, Blätter erklären

F: Wie kann man einen binären Baum als Array darstellen?
A: Knoten durchnumerieren, der linke Nachfolger von Node[i] ist Node[2i], der rechte ist Node [2i+l] (Das kam unerwartet. Nach einigen Winks mit dem Zaunpfahl bin ich dann drauf gekommen, aber doch eher durch Raten.)

F: Wie als record?
A: Typdefinition wie im Kurstext (Hier war ich von eben noch etwas konfus, und machte für den linken und rechten Teilbaum Verweise auf Knoten, es müssen aber Verweise auf Bäume sein.)

F: Wie wird die Syntax einer Programmiersprache definiert?
A: Syntaxdiagramm oder kontextfreie Grammatik

F: Wie kann man die Semantik definieren?
A: Hoare-Regeln

F: Erklären Sie, was eine Programmformel ist.
A: Vorbedingung, Anweisung, Nachbedingung erklären

F: Leiten Sie die Hoare-Regeln für if-then-else und für while her.
A: darstellen, wie es im Kurstext erläutert ist

Fazit:
Ich hatte fast immer Probleme herauszufinden, was Prof.Helbig eigentlich von mir wollte, und ich mußte zum Teil nachfragen. Was ich ihm hoch anrechne, ist die Tatsache, daß er bei kleinen Unsicherheiten nicht weiter in den Wunden bohrte, sondern korrigierte und zur nächsten Frage überging. Dadurch hielt sich der Prüfungsstreß in Grenzen. Weil ich in einigen Punkten Unsicherheiten zeigte und er mir auf die Sprünge helfen mußte, gab es ‘nur’ eine 1,7. Vom Gefühl her hatte ich mich schlechter eingeschätzt, kann also nicht meckern. Wer in der Theorie recht fit ist, hat mit Prof.Helbig einen angenehmen Prüfer, der es kurz und schmerzlos erledigt, Programmierfreaks dagegen kommen nicht so auf ihre Kosten.

Smja Hilberl

Rüfungsprotokoll

Konzepte Imperativer Programmierung

Prüfer: Prof. Helbig

Termin: QI 1999

Dauer: 35 min

Note: 2

Prüfungsthemen

Welche Iterationsverfahren kennen Sie?

for - while - repeat. Syntaxbeispiele hinschreiben.

Sind while und repeat unabhängig?

Ja, weil man auf eines von beiden verzichten kann. Beide Varianten hinschreiben.

procedure / function

Wie werden die Ergebnisse zurtickgegeben?

Syntaxdiagramm hinschreiben, dabei kommt es natürlich auf die Darstellung der Unterschiede an. Der Rest kann kompakt dargestellt werden.

call by value / call by reference

Fibunacci Zahlen

Rekursiv und iterativ implementieren. Die Definition der FZ wird bei Bedarf gestellt.

Grammatiken

Definition

P={n E N ::= v E V* mit V = N U T} Dabei ist V* die Menge aller Zeichenketten gebildet aus den Elementen aus N vereinigt T, die sogenannte Hülle.

Resümee

Mir hat es etwas an der Metasprache für dieses Gebiet gefehlt. Ich bin mit einigen, mir in diesem Zusammenhang unbekannten Variablen konfrontiert worden.

Das hat Nervosität erzeugt, die mir dann bei der iterativen Implementierung der Fibunacci Zahlen zum Nachteil gereichte.

Prof Helbig ist ein empfehlenswerter Prüfer. Nervosität ist unangebracht und unnötig. Er ist ruhig und gedulig, führt einen aber auch beharrlich zur Beantwortung seiner Fragen. Weiterer Pluspunkt: Wenn man etwas richtig macht, dann sagt er es auch! Die Bewertung empfinde ich als gerechtfertigt.

Viel Erfolg!