

Was ist für die Klausur relevant ?

Kapitel 1: Einführung

Grundlegende Begriffe und Konzepte: Funktion, Spezifikation, Algorithmus, Programm, Verifikation, Konstruktion und alles, was damit zusammenhängt. Fähigkeit, die Konzepte an Minibeispielen anzuwenden.

Kapitel 2.1: Aussagenlogik

Grundlegende Begriffe wie Aussage, Wahrheitswert, Formel, Belegung, Erfüllbarkeit. Sicherer Umgang mit aussagenlogischen Umformungen, eindeutige und exakte Notation. Konjunktive Normalform: Begriffe und Fähigkeit der Umformung in eine solche.

Kapitel 2.2: Prädikatenlogik

Unterschied zur Aussagenlogik: Variable, Prädikate, Funktionen, Quantoren. Erklären dieser Begriffe an Beispielen, aktive und passive Beherrschung der Notation. Rechenregeln für Quantoren, Anwendung der Prädikatenlogik auf Minibeispiele aus dem täglichen Leben und der Arithmetik.

Was ist für die Klausur relevant ?

Kapitel 3: Grundlagen logischer Programmierung

Ziele und theoretische Grenzen logischer Programmierung.
Anwendung der Techniken Resolution und Unifikation auf Formelmengen.

Kapitel 4.1: Verifikation mit Hoare-Tripeln

Hoare-Tripel: Aktive und passive Beherrschung der Notation.
Zusammenhang zwischen unterschiedlichen Stärken von Vorbedingungen und Nachbedingungen.
Exakte Verifikation kleinster Programmteile: Finden von stärksten Nachbedingungen und schwächsten Vorbedingungen.

Kapitel 4.2: Verzweigungen

Finden von stärksten Nachbedingungen und schwächsten Vorbedingungen in Anwendungsbeispielen.

Kapitel 4.3: Schleifen

Bestimmung und Verifikation von Invariantenbedingungen sowie von Variantenzahlen für die Terminierung, Verifikation von kleinen Schleifen mit vollständiger Induktion (**als einziger Verifikationstechnik**)

Was ist für die Klausur relevant ?

Kapitel 4.4: Modulare und funktionale Programmierung

Call-by-reference und call-by-value: Erklärung der Unterschiede, Vorteile und Nachteile der beiden Parameterübergabetechniken.

Funktionale Programmierung: Umwandlung von Prozeduren mit Ausgabeparametern in Prozeduren mit Rückgabewerten.

Formulierung von Berechnungen in funktionaler Präfixnotation
(Lisp-Notation: nur Funktionsaufrufe, nicht Funktionsdeklarationen)

Kapitel 4.5: Rekursion

Nachvollziehen des Programmablaufs vorgegebener rekursiver Programme, Bestimmen von notwendigen Vorbedingungen.

Erklärung der Konzepte primitiv rekursiv, endrekursiv und linear rekursiv.

Erkennen dieser Konzepte an Beispielen.

Umformung von endrekursiven Programmen in Schleifen.

Kapitel 5: Entwurf und Analyse von Algorithmen

Grundideen der Algorithmen Lineare Suche und Binärsuche (keine weiteren Algorithmen).

Kenntnis der Komplexitätseigenschaften der Algorithmen Lineare Suche, Binärsuche, Selectionsort, Mergesort und Quicksort.

Begründung, warum Komplexitätsklassen mit Landau-Symbolen ein gutes Maß zur Bewertung von Algorithmen sind (zum Beispiel im Vergleich zu iterativ / rekursiv).

NP-Vollständigkeit: Grundidee und praktische Relevanz.