

# LVA 185.276 Analyse und Verifikation (SS 2020)

## Selbsteinschätzungstest 5

Mi, 25.03.2020

*Stoff: Vorlesungsteil III, Kapitel 6 und 7*

*Programmanalyse, Datenflussanalyse*

(Ohne Abgabe, ohne Beurteilung)

### Teil III, Kapitel 6 ‘Programmanalyse’

1. Welche grundsätzlichen Probleme stellen sich für Programmanalyse? Für Entwicklung und Anwendung von Programmanalyseverfahren?
2. Welche Lichtblicke gibt es, mit diesen Problemen erfolgreich umzugehen?
3. Welche konkreten Beispiele (ggf. auch aus anderen Bereichen als der Programmanalyse und -verifikation) können Ihre Antworten zu den vorigen beiden Fragen veranschaulichen und untermauern?

### Teil III, Kapitel 7 ‘Datenflussanalyse’

1. Datenflussanalyse stützt sich auf *vollständige Verbände* zur mathematischen Modellierung interessierender Programmeigenschaften. Warum reichen nicht *Verbände*?
2. Gibt es Programmklassen, für die vollständige Verbände nicht nötig wären und *Verbände* doch reichen? Warum?
3. Ist die Forderung nach vollständigen Verbänden für operationelle und denotationelle Datenflussanalyse in gleicher Weise nötig? Begründen Sie Ihre Antwort.
4. Monotonie, Distributivität, Additivität sind *Erhaltungseigenschaften*. Was erhalten sie?
5. In welchen (Implikations-) Beziehungen stehen Monotonie, Distributivität, Additivität zueinander?
6. Distributivität, Additivität sind sog. *Homomorphieeigenschaften*. Was bedeutet das? Was ist eine Homomorphieeigenschaft?
7. In welchem Sinn ist die DFA-Aufsammelsemantik als globale Flussgraphsemantik *nichtdeterministisch*?
8. Auf welche Weise(n) *determinieren* wir die nichtdeterministische DFA-Aufsammelsemantik, machen sie *deterministisch*?
9. Welche Bestandteile legen ein Datenflussanalyseproblem fest? Welche Rolle spielen sie für das Datenflussanalyseproblem?
10. Manche Datenflussanalyseprobleme werden als *Bitvektorprobleme* oder *Gen/Kill-Probleme* bezeichnet. Was steckt hinter diesen Bezeichnungen?
11. Was verstehen wir als *spezifizierende* Lösung(en) eines Datenflussanalyseproblems, was als seine *berechenbare(n)* Lösung(en)? Warum wird diese Unterscheidung vorgenommen? Begründen Sie Ihre Antwort.
12. Welche Eigenschaften muss die Spezifikation eines Datenflussanalyseproblems  $P$  erfüllen, damit
  - (a) die spezifizierende Lösung von  $P$  existiert?
  - (b) die berechenbare Lösung von  $P$  existiert?
  - (c) die berechenbare Lösung von  $P$  effektiv, in endlicher Zeit berechnet werden kann?
  - (d) die berechenbare Lösung von  $P$  korrekt bzgl. der spezifizierenden Lösung von  $P$  ist?
  - (e) berechenbare und spezifizierende Lösung existieren und übereinstimmen?

13. Terminierung, Sicherheit und Koinzidenz sind drei zentrale Begriffe aus der Theorie der Datenflussanalyse. Was bedeuten sie?
14. Im Rahmen der axiomatischen Semantik haben wir bei den Hoare-Kalkülen für partielle und totale Korrektheit von Kalkül-Korrektheit und Kalkül-Vollständigkeit gesprochen. In welcher Weise finden sich Korrektheit und Vollständigkeit inhaltlich in der Theorie der Datenflussanalyse wieder? Was bedeuten Korrektheit und Vollständigkeit in der Theorie der Datenflussanalyse?
15. Auf wen gehen die Resultate zur Unentscheidbarkeit der spezifizierenden Lösung von Datenflussanalyseproblemen zurück? Auf wen die Resultate zur Unentscheidbarkeit des Konstantenpropagations- und -faltungsproblems zurück? Worauf beruhen die Ideen zum Beweis der jeweiligen Unentscheidbarkeiten? Welche Hilfsergebnisse gehen ein? Auf wen gehen sie zurück?
16. Als *anytime*-Algorithmen bezeichnet man Algorithmen, die jederzeit vor ihrer regulären Terminierung abgebrochen werden können und deren bis dahin bzw. zum Abbruchzeitpunkt gelieferten oder vorliegenden Ergebnisse möglicherweise nicht vollständig oder präzise, aber korrekt und deshalb nützlich sind. Ist der generische Fixpunktalgorithmus zur Berechnung der *MaxFP*-Lösung ein *anytime*-Algorithmus? Begründen Sie Ihre Antwort.
17. Was hat die besonders aus dem amerikanischen Rechtswesen bekannte (Eides-) Formel *die Wahrheit, die ganze Wahrheit und nichts als die Wahrheit* zu sagen, mit Datenflussanalyse zu tun?
18. Sei  $A$  eine Programmanalyse, die mögliche Programmfehler aufdecken soll. Manche Fehler entdeckt  $A$ ; manche Fehler entdeckt  $A$  nicht; manche von  $A$  gemeldete Fehler sind bei genauerer Nachschau tatsächlich keine. Erfüllt  $A$  die Forderung, *die Wahrheit, die ganze Wahrheit und nichts als die Wahrheit* zu sagen? Kann  $A$  trotzdem 'nützlich' sein? Begründen Sie Ihre Antwort.
19. Was muss für die Anzahl tatsächlicher Fehlerstellen in einem Programm und die von von  $A$  als Fehlerstellen gemeldeten gelten, die bei genauerer Nachschau in *true positives* und *false positives* geschieden werden können, damit  $A$  guten Gewissens die vor einem amerikanischen Gericht geforderte Eidesformel leisten könnte?
20. Die Theorie der Datenflussanalyse lässt sich als Rahmenwerk (engl. *framework*) und auch als Werkzeugkiste (engl. *tool kit*, *tool box*) auffassen. Warum? In welchem Sinn sind beide Sichtweisen gerechtfertigt?