

LVA 185.276 Analyse und Verifikation (SS 21)

Leit- und Kontrollfragen VII

Mi, 12.05.2021

Stoff: Vorlesungsteil IV – Kapitel 11, 12 und 13

Analyse – Abstrakte parallele Semantiken, parallele Analysesemantiken; Parallele Datenflussanalyse; Datenflussanalyse vs. axiomatische Verifikation

(Ohne Abgabe, ohne Beurteilung)

Teil IV, Kapitel 11 ‘Abstrakte parallele Semantiken, parallele Analysesemantiken’

1. Welche zentralen Phänomene treten beim Übergang von sequentiellen zu parallelen Programmen neu auf?
2. Was sind *verschränkte* Vorgänger?
3. Was sind (*syntaktisch*) *sequentialisierte* Flussgraphen?
4. Was sind parallele Pfade?
5. Wie sind die
 - (a) parallele Aufsammlungsemantik
 - (b) Schnitt-über-alle-Pfade-Semantik
 - (c) Vereinigung-über-alle-Pfade-Semantikparalleler Programme bzw. Flussgraphen definiert?
6. Treten bei der Definition dieser Semantiken im Vergleich zu ihren Gegenstücken neue Probleme auf? Begründe die Antwort.

Teil IV, Kapitel 12 ‘Parallele Datenflussanalyse’

1. Was wird als *Pfad-* oder *Zustandsexplosionsproblem* bezeichnet?
2. In welcher Weise treten das *Pfad-* bzw. *Zustandsexplosionsproblem* bei parallelen Programmen auf?
3. Welches sind die zentralen Probleme, die beim Übergang von der Analyse sequentieller zur Analyse paralleler Programme gelöst werden müssen?
4. Warum können das *Pfad-* und *Zustandsexplosionsproblem* für Bitvektoranalysen paralleler Programme vermieden werden?
5. Wie lässt sich die Unentscheidbarkeit von *SUPP-* und *VUPP-*Semantik beweisen?
6. Was sind sog. *kontrollflussfreie* Datenflussanalysen?
7. Welche Bedeutung hat kontrollflussfreie Analyse für Bitvektoranalysen paralleler Programme?
8. Worin liegt der Schlüssel für die Effizienz und Skalierbarkeit von Bitvektoranalysen für parallele Programme?
9. Funktioniert der Schlüssel auch für andere Problemklassen?
10. Was ist der funktionale denotationelle Semantikansatz? Welche Rolle spielt er für die Analyse paralleler Programme?

11. Im welchem Zusammenhang steht die funktionale denotationelle Semantik zur denotationellen Semantik aus Kapitel 7? Welche Resultate gelten und verbinden die beiden?
12. Was ist der *Rang* eines parallelen Flussgraphen? Wofür dient er im Rahmen paralleler Datenflussanalyse?
13. Welche Rolle spielen (syntaktisch) sequentialisierte Flussgraphen im Rahmen paralleler Bitvektoranalysen?
14. Warum reicht es, sich für die hierarchische Analyse paralleler Flussgraphen für Bitvektorprobleme auf 'syntaktisch sequentialisierte' Graphen abzustützen?
15. Bitvektoranalysen sind einfach vom sequentiellen auf parallele Programme zu übertragen. Für auf Bitvektoreigenschaften aufbauende Programmtransformationen und -optimierungen gilt das auch. Richtig oder falsch? Begründe die Antwort.

Teil IV, Kapitel 13 'Datenflussanalyse vs. axiomatische Verifikation'

1. Skizziere den Zusammenhang und Analogien zwischen axiomatischer Programmverifikation und reverser Datenflussanalyse.
2. Veranschauliche den Zusammenhang von Datenflussanalyse und reverser Datenflussanalyse graphisch.
3. Illustriere Idee und Zusammenhang axiomatischer Programmverifikation aus Sicht stärkster Nachbedingungen und schwächster Vorbedingungen.
4. In welcher Weise unterscheiden sich die
 - (a) Aufgaben
 - (b) Eigenschaften
 auf die sich axiomatische Verifikation und Datenflussanalyse im allgemeinen konzentrieren?
5. Was ergibt sich aus diesen Unterschieden für die Skalierbarkeit von axiomatischer Verifikation und Datenflussanalyse im Austausch gegen Genauigkeit, Vollständigkeit der Resultate?

Teil I – IV, Verschiedene Kapitel

1. Erweitere die Programmiersprache WHILE um das Schleifenkonstrukt

`repeat π until b end`

Definiere die Semantik der repeat-Schleife im Stil der natürlichen Semantik durch Angabe entsprechender NS-Regeln, ohne sich dabei auf die Existenz der while-Schleife und deren natürliche Semantik abzustützen; die repeat-Schleife soll dabei ihre "gewohnte" Bedeutung erhalten.

2. Zeige mithilfe der natürlichen Semantik von WHILE, dass die Programme π und π' von SET 6 angesetzt auf den Zustand $\sigma \in \Sigma$ mit $\sigma(x) = \mathbf{13}$ und $\sigma(y) = \mathbf{5}$ regulär terminieren und die Werte der Variablen x , y und z in den jeweiligen Endzuständen übereinstimmen.
3. In welcher konzeptuellen Weise unterscheiden sich denotationelle und axiomatische Semantik einer voneinander?
4. Was besagt das Prinzip der strukturellen Induktion?
5. Was wäre ein gutes Beispiel für eine Aussage, die mithilfe struktureller Induktion bewiesen würde?
6. Führe diesen Beweis.
7. Erläutere die denotationelle Semantik der while-Schleife.
8. Wie lässt sich das Sicherheitstheorem aus Kapitel 7 beweisen?

9. Spezifiziere als reverse Datenflussanalyse eine anforderungsgetriebene Datenflussanalyse, mit deren Hilfe bestimmt werden kann, ob eine bestimmte Variable an einem bestimmten Programmpunkt möglicherweise nicht initialisiert ist.
10. Was ist die Aufsammelsemantik einer Datenflussanalyse? Wie ist sie definiert?