

LVA 185.A04 Optimierende Compiler (WS 20)

Leit- und Kontrollfragen X

Do, 10.12.2020

Stoff: Vorlesungsteil III – Kapitel 12.2 - 12.7, Kapitel 13

Interprozedurale DFA: Der funktionale und kontextbasierte Ansatz

(Ohne Abgabe, ohne Beurteilung; zur Selbsteinschätzung)

Teil III, Kapitel 12.2 - 12.7 ‘Der fkt. Ansatz: Erweiterungen, Anwendungen’

1. Was unterscheidet
 - (a) gewöhnliche
 - (b) formaleProzeduraufrufe?
2. In objektorientierten Sprachen spricht man üblicherweise nicht von formalen Prozeduraufrufen, sondern wovon?
3. Wie lässt sich die Schlüsselidee zur Behandlung formaler Prozeduraufrufe im interprozeduralen DFA-Rahmen zusammenfassen und beschreiben?
4. Auf welche Weise wird die DFA-Semantik formaler Prozeduraufrufe mathematisch modelliert/beschrieben?
5. Ändert sich durch die Behandlung formaler Prozeduraufrufe die Stufigkeit interprozeduraler DFA? Wenn ja, wie? Wenn nein, warum nicht?
6. Ändern sich durch die Behandlung formaler Prozeduraufrufe einzelne Stufen des interprozeduralen DFA-Rahmens? Wenn ja, wie? Wenn nein, warum nicht?
7. Welche Rolle spielt die sog. Klassenhierarchieanalyse für die Behandlung formaler Aufrufe in der Analyse objektorientierter Programme?
8. Was für eine Analyse wäre die natürliche Entsprechung der Klassenhierarchieanalyse für prozedurale Programmiersprachen?
9. Referenzparameter werfen Alias-Probleme auf. Was ist damit gemeint?
10. Für die Analyse von Referenzparametern werden ‘mag’- und ‘muss’-Varianten unterschieden.
 - (a) Wie unterscheiden sich diese Varianten?
 - (b) Warum sind beide Varianten nötig?Belegen Sie Ihre Antwort anhand von Beispielen.
11. Sind die ‘mag’- und ‘muss’-Fragen für Referenzparameter entscheidbar?
12. Was muss bei Hinzutreten von statisch geschachtelten Prozeduren für die interprozedurale Analyse bedacht werden? Welche Probleme treten neu auf?
13. Welche Möglichkeiten bieten sich an, um mit statisch geschachtelten Prozeduren in interprozeduraler DFA umzugehen?
14. Wie können Aliasphänomene durch Referenzparameter in Bitvektoranalysen für z.B. ihre vier kanonischen Vertreter berücksichtigt werden?
15. Wir betrachten die Probleme
 - (a) einfacher
 - (b) Kopier-Konstanten. Sind beide Probleme in gleicher Weise für interprozedurale DFA geeignet? Wenn ja, warum? Wenn nein, warum nicht?

Teil III, Kapitel 13 ‘Der kontextbasierte Ansatz’

1. Welche Vorteile versuchen kontextbasierte Ansätze für interprozedurale DFA im Vergleich zum funktionalen Ansatz zu erreichen?
2. Welche Nachteile nimmt man dafür in Kauf?
3. Was unterscheidet kontextsensitive von kontextinsensitiven Ansätzen?
4. Was ist die Idee von
 - (a) Aufrufsequenzen (engl. call strings)
 - (b) Annahmemengen (engl. assumption sets)?
5. Was sind (einfache) Beispiele möglicher
 - (a) Aufrufsequenzen
 - (b) Annahmemengen?
6. Wie lässt sich die Idee von Aufrufsequenzen verfeinern?
7. Wie ist das maximale Fixpunktsystem interprozeduraler DFA dafür anzupassen?
8. Ist die Länge von Aufrufsequenzen grundsätzlich beschränkt? Wenn ja, warum? Wenn nein, warum nicht?
9. Ist die Länge von Aufrufsequenzen nicht beschränkt, wie geht man damit um?
10. Was sind naheliegende Möglichkeiten zur Behandlung von
 - (a) Funktionszeigern
 - (b) Virtuellen Funktions-/Methodenaufrufen
 - (c) Überladenen Funktionen?
11. Auf welcher Idee bauen klonbasierte Verfahren für interprozedurale DFA auf?
12. Für welche Probleme werden sie häufig eingesetzt?