

LVA 185.A04 Optimierende Compiler (WS 20)

Leit- und Kontrollfragen IX

Di, 01.12.2020

Stoff: Vorlesungsteil III – Kapitel 12.1

Interprozedurale DFA: Der funktionale Ansatz: Wertparameter, lokale Variablen
(Ohne Abgabe, ohne Beurteilung; zur Selbsteinschätzung)

Teil III, Kapitel 12.1 ‘Der fkt. Ansatz: Wertparameter, lokale Variablen’

1. Welche Analyseprobleme treten bei Hinzunahme von
 - (a) Wertparametern
 - (b) lokalen Variablenfür Prozeduren gegenüber dem parameter- und variablenlosen Fall neu hinzu?
2. Wie werden
 - (a) Prozeduraufrufe mit Wertparameterübergabe
 - (b) Prozeduren mit lokalen Variablenin Flussgraphsystemen dargestellt?
3. Wie in interprozeduralen Flussgraphen?
4. Wozu dienen Datenflussanalysestapel? Was modellieren sie?
5. Stapel werden üblicherweise ausgehend vom leeren Stapel konstruiert. Datenflussanalysestapel nicht. Warum?
6. Welche Operationen werden für die Manipulation von Datenflussanalysestapeln benötigt?
7. Welche Rolle kommt den Rückkehrfunktionen in der Spezifikation eines interprozeduralen Datenflussanalyseproblems zu?
8. Welche Funktionalität haben diese Rückkehrfunktionen?
9. Wie beurteilen Sie folgende Aussage:
 - Ein DFA-Stapeleintrag modelliert genau einen Aktivierungseintrag des Laufzeitsystemstapels.Richtig? Halbrichtig? Falsch? Begründen Sie Ihre Antwort.
10. Haben Prozeduren Wertparameter oder/und lokale Variablen arbeiten die lokalen abstrakten Semantikfunktionen auf Datenflussanalysestapeln. Diese Funktionen müssen in der Spezifikation eines entsprechenden interprozeduralen DFA-Problems angegeben werden. Richtig oder falsch? Begründen Sie Ihre Antwort.
11. Wie transformieren/verändern
 - (a) Aufruffunktionen (an Aufrufkanten)
 - (b) Rückkehrfunktionen (an Rückkehrkanten)
 - (c) ‘gewöhnliche’ Funktionen (an Nichtaufruf-/rückkehrkanten)einen Datenflussanalysestapel? Geben Sie eine informelle Beschreibung.
12. Welche Komponenten gehören zur Spezifikation eines interprozeduralen DFA-Problems für Programme mit Wertparametern oder/und lokale Variablen?

13. Welche Gruppen von DFA-Funktionen auf DFA-Stapeln können gemäß ihrer wesentlichen strukturellen Eigenschaften unterschieden werden?
14. Was sind die signifikanten Funktionen der DFA-Funktionen auf DFA-Stapeln? Welche Rolle spielen sie?
15. Welche Eigenschaften bezeichnen
 - (a) S-Monotonie
 - (b) S-Distributivität?
16. Ist auch S-Additivität als Eigenschaft sinnvoll? Wenn ja, wofür? Wenn nein, warum nicht?
17. Wann lässt sich auf
 - (a) S-Monotonie
 - (b) S-Distributivität
 von DFA-Funktionen auf DFA-Stapeln schließen? Was ist hinreichend dafür?
18. Die Festlegungen:

$$\sqcap STK =_{df} \text{newstack}(\sqcap \{top(stk) \mid stk \in STK\})$$

$$\sqcup STK =_{df} \text{newstack}(\sqcup \{top(stk) \mid stk \in STK\})$$
 für $STK \subseteq STACK$ verwerfen nahezu alle Einträge von DFA-Stapeln. Warum sind diese Festlegungen dennoch sinnvoll? Warum führt das Verwerfen zu keinen Problemen?
19. Wie ist die *IMOP*-Lösung eines interprozeduralen DFA-Problems für Programme mit Werteparametern oder/und lokale Variablen definiert?
20. Aus wie vielen Stufen besteht der *IMaxFP*-Ansatz für die Lösung eines interprozeduralen DFA-Problems für Programme mit Werteparametern oder/und lokale Variablen?
21. Welche Stufen sind das?
22. Wodurch sind diese Stufen gegeben/festgelegt?
23. Was besagen das interprozedurale
 - (a) Sicherheits-
 - (b) Koinzidenz-
 - (c) Terminations-
 Theorem für interprozedurale DFA-Probleme und Programme mit Werteparametern oder/und lokale Variablen?
24. Warum ist es möglich, für die Implementierung des generischen Fixpunktalgorithmusses für interprozedurale DFA-Probleme und Programme mit Werteparametern oder/und lokale Variablen auf DFA-Stapel zu verzichten?
25. Was ist stattdessen ausreichend?
26. Warum ist das für die interprozedurale Schnitt-über-alle-Pfade-Lösung nicht möglich?
27. Wie hängen die stapelgestützte Schnitt-über-alle-Pfade-Lösung mit der stapellosen maximalen Fixpunktlösung zusammen? Warum liefert die stapellose maximale Fixpunktlösung eine Beschreibung der der stapelgestützten Schnitt-über-alle-Pfade-Lösung? Welcher Zusammenhang besteht?
28. Skizzieren Sie den interprozeduralen DFA-Rahmen für Programme mit Werteparametern oder/und lokalen Variablen graphisch.

29. Erklären Sie informell die Bedeutung des Gleichungssystems:

$$\llbracket n \rrbracket_{Stk} = \begin{cases} Id_{STACK} & \text{if } n \in \{s_0, \dots, s_k\} \\ \bigsqcap \{ \llbracket (m, n) \rrbracket_{Stk} \circ \llbracket m \rrbracket_{Stk} \mid m \in pred_{flowGraph}(n) \} & \text{otherwise} \end{cases}$$

und

$$\llbracket e \rrbracket_{Stk} = \begin{cases} \llbracket e \rrbracket_{Stk}^* & \text{if } e \in E \setminus E_{call} \\ \llbracket e_r \rrbracket_{Stk}^* \circ \llbracket end(callee(e)) \rrbracket_{Stk} \circ \llbracket e_c \rrbracket_{Stk}^* & \text{otherwise} \end{cases}$$

30. Erklären Sie informell die Bedeutung des Gleichungssystems:

$$inf(n) = \begin{cases} newstack(c_s) & \text{if } n = s_0 \\ \bigsqcap \{ \llbracket e_c \rrbracket_{Stk}^*(inf(src(e))) \mid e \in caller(flowGraph(n)) \} & \text{if } n \in start(S) \setminus \{s_0\} \\ \bigsqcap \{ \nu \llbracket (m, n) \rrbracket_{Stk}(inf(m)) \mid m \in pred_{flowGraph}(n) \} & \text{otherwise} \end{cases}$$