WS 2017/2018 15.11.2017

Übungen zur Vorlesung
Programmanalyse
Blatt 4

Prof. Dr. Roland Meyer,
M. Sc. Sebastian Wolff
M. Sc. Elisabeth Neumann Abgabe bis 22.11.2017 um 12 Uhr

Aufgabe 4.1 (Funktionenverbände)
Sei $(D, \leq)$ ein vollständiger Verband und sei $F_D := \{f : D \to D \mid f \text{ ist monoton}\}$ die Menge aller monotonen Funktionen über D. Es gelte $f_1 \preceq f_2$ genau dann, wenn $f_1(d) \leq f_2(d)$ für alle $d \in D$.

Zeigen Sie, dass $(F_D, \preceq)$ ein vollständiger Verband ist.

Aufgabe 4.2 (Procedure Summaries)
Betrachten Sie folgendes Programm.

```
proc [ main() ]^1
    [ g := 2 ]^2
    [ l := g ]^3
    [ work() ]^4_5
    [ ret := 1 − ret ]^6
end
```

```
proc [ work() ]^7
    [ l := g ]^8
    if [ l > 0 ]^9 then
        [ work() ]^10_11
    [ ret := l ]^12
end
```

Führen Sie eine *Copy-Propagation-Analyse* durch. Gesucht ist für jeden Block die Menge der Variablen, die am Ausgang definitiv den gleichen Wert wie g haben (vorwärts-must-Analyse). Zum Beispiel hat *ret* nach Block 12 den gleichen Wert wie g, nicht aber nach Block 6.

Benutzen Sie den Verband $(\mathbb{P}(\mathit{Var}), \supseteq)$, mit $\mathit{Var} = \{g, \mathit{ret}, l\}$. Benutzen Sie ferner die Menge der globalen Variablen $\mathit{Glob} = \{g, \mathit{ret}\}$.

Gehen Sie wie folgt vor.

a) Stellen Sie die Transferfunktionen auf, welche Sie für die Berechnung einer Procedure Summary für die *work()*-Prozedur benötigen. Geben Sie insbesondere f_{call} und f_{return} an.

Hinweis: Sie benötigen f_{10} und f_{11} nicht, da diese durch f_{call} und f_{return} abgedeckt werden.

b) Stellen Sie das Summary-Gleichungssystem auf.

Hinweis: Sie benötigen kein Y_{11}, wenn Sie strikt der Definition von $\mathit{callret}(\cdot)$ aus der Vorlesung folgen.

c) Vereinfachen Sie ihr Gleichungssystem so weit wie möglich, um sich in der folgenden Teilaufgabe etwas Arbeit zu sparen.

d) Lösen Sie das Summary-Gleichungssystem durch Fixpunkt-Iteration auf dem Verband der monotonen Funktionen.

Hinweis: Stellen Sie eine Funktion durch ihre Funktionsvorschrift $Y_i(X)$ dar.

e) Schließen Sie die Datenflussanalyse, unter Verwendung der soeben erstellten Procedure-Summary, ab.

Aufgabe 4.3 (Procedure Summaries und *call-return*-Verhalten)
Im Programm aus Aufgabe 5.2 wird die Variable g nach ihrer ersten Belegung nicht mehr verändert. Dadurch kann man sicher sein, dass alle Kopien von g auch nach der Ausführung von *work()* noch Kopien von g sind. Betrachten Sie nun das folgende Programm mit einer modifizierten *work()*-Prozedur:

```
proc [ main() ]^1
    [ g := 2 ]^2
    [ l := g ]^3
    [ work() ]^4_5
    [ ret := 1 - ret ]^6
end
```

```
proc [ work() ]^7
    if [ g > 0 ]^9 then
        [ work() ]^10_11
    else
        [ g := g - 1 ]^12
    [ ret := g ]^13
end
```

Hier wird in der *work()*-Prozedur die Variable g geändert. Damit ist am Ende der Ausführung auch l (in *main()*) keine Kopie von g. Adaptieren Sie die Analyse so, dass diese Situation erkannt wird und das richtige Ergebnis geliefert wird.

Hinweis: Um zu erkennen, ob g in der Unterfunktion geändert wurde, kann f_{call} eine Hilfsvariable einführen, die beim Rücksprung durch f_{return} ausgewertet wird.

Abgabe bis 22.11.2017 um 12 Uhr im Kasten neben Raum IZ 343