

Grundlagen der Programmkonstruktion

Übungsblatt 2

Organisatorisches

Ablauf

Lösen Sie die Aufgaben auf diesem Übungsblatt. Kommen Sie mit diesen Lösungen in die Übungseinheit. In der Übungseinheit sollen Sie in der Lage sein, die Lösung der gestellten Aufgaben zu präsentieren. Dabei steht im Vordergrund, wie Sie zu ihrer Lösung gelangen und weniger das korrekte Ergebnis. Die Aufgaben dieses Übungsblatts sollen Sie in Gruppen (wie in der ersten Übungseinheit eingeteilt) erarbeiten und auch in Gruppen präsentieren.

Fragen

Bei Fragen wenden Sie sich bitte in den Sprechstunden an die Tutoren (Termine siehe <http://www.complang.tuwien.ac.at/franz/pk#tutoren>). Sollte in der Angabe etwas unklar sein, sehen Sie bitte zunächst auf der Homepage der Lehrveranstaltung nach, ob dort bereits ergänzende Anmerkungen zum Übungsteil stehen. Wichtige Ankündigungen zum Übungsteil werden auch in der Vorlesung und online bekannt gegeben.

1 Aufgabe 1: Lambda-Kalkül - ggT (1,5)

Reduzieren Sie den Lambda-Ausdruck $((F F) 4) 2$ zur Normalform, wobei

$$F = (\lambda u. \lambda v. \lambda w. v == w ? v : v > w ? ((u u) (v - w)) w : ((u u) (w - v)) v)$$

Hinweise

Beachten Sie, dass der ternäre Operator `?:` ineinander geschachtelt verwendet wird. Ersetzen Sie F durch den Lambda-Ausdruck erst so spät wie möglich, um unnötig große Ausdrücke zu vermeiden.

Beispiel

Die Funktion entspricht einer rekursiven Variante des euklidischen Algorithmus. Die Reduktion des Ausdrucks $((F F) 1) 1$ sieht folgendermaßen aus (anstatt von λ wird `\` verwendet um das Tippen zu erleichtern):

```
((F F) 1) 1  ->
(((\u.\v.\w.v == w ? v : v > w ? ((u u) (v - w)) w : ((u u) (w - v)) v) F) 1) 1  ->
((\v.\w.v == w ? v : v > w ? ((F F) (v - w)) w : ((F F) (w - v)) v) 1) 1  ->
(\w.1 == w ? 1 : 1 > w ? ((F F) (1 - w)) w : ((F F) (w - 1)) 1) 1  ->
1 == 1 ? 1 : 1 > 1 ? ((F F) (1 - 1)) 1 : ((F F) (1 - 1)) 1  ->
1
```

2 Aufgabe 2: Grammatik (1)

Gegeben ist folgende Grammatik:

$$\begin{aligned} \text{Ausdruck} &= [-] \text{Wert} \{ + \text{Wert} \} \\ &| [-] \text{Wert} \{ - \text{Wert} \} \end{aligned}$$

$$\begin{aligned} \text{Wert} &= \text{Zahl} \\ &| (\text{Ausdruck}) \end{aligned}$$

$$\begin{aligned} \text{Zahl} &= \mathbf{0} \\ &| \mathbf{1} \{ \mathbf{0} \mid \mathbf{1} \} \end{aligned}$$

Ermitteln Sie, ob die folgenden Wörter syntaktisch korrekte Ausdrücke sind. Begründen Sie ihre Antwort.

- 0
- -0
- 0100
- 1011
- 1000
- (1)
- ((1))
- 1+1
- -1-1
- --1
- -1+-1
- 1+1+1
- 1+1-1
- 1+1+(-1)
- -(-1)
- 1+(-1-1)
- -(1-1)+(1-1)

3 Aufgabe 3: Rekursion nachvollziehen (1,5)

Versuchen Sie folgende rekursive Methode nachzuvollziehen. Ermitteln Sie alle Aufrufe von `hanoi(3, 1, 2, 3)`; und die Ausgabe der Methode in der richtigen Reihenfolge (siehe Beispiel unten).

```
public static void hanoi(int n, int a, int b, int c) {
    if(n == 0) {
        return;
    } else {
        hanoi(n - 1, a, c, b);
        System.out.println("Von " + a + " nach " + c);
        hanoi(n - 1, b, a, c);
    }
}
```

Beispiel

Zur besseren Übersichtlichkeit ist es hilfreich, entsprechend der Rekursionstiefe die Aufrufe und Ausgaben einzurücken.

Die Lösung für `hanoi(1, 1, 2, 3)`; sieht folgendermaßen aus:

```
hanoi(1, 1, 2, 3);
    hanoi(0, 1, 3, 2);
    "Von 1 nach 3"
    hanoi(0, 2, 1, 3);
```

Die ersten drei Zeilen Ihrer Lösung sollten folgendermaßen aussehen:

```
hanoi(3, 1, 2, 3);
    hanoi(2, 1, 3, 2);
        hanoi(1, 1, 2, 3);
```

4 Aufgabe 4: Theoriefrage (1)

Beantworten Sie die folgenden Theoriefragen tiefgehend:

- Wann ist etwas statisch und wann dynamisch?
- Wann spricht man von statischer und wann von dynamischer Semantik einer Programmiersprache?
- Was sind Zusicherungen? Wie können wir sie ausdrücken?