

# Formale Grundlagen der Informatik 3

Prof. Dr. Christoph Walther / Visar Januzaj, Nathan Wasser  
Technische Universität Darmstadt — Wintersemester 2011/12

## Hausübung 1

---

Geben Sie die handschriftliche Lösung für diese Übung **am 6. Dezember 2011 vor oder nach der Vorlesung** von 16:15 bis 17:55 in S202/C205 ab. Heften Sie Ihre Blätter **oben links** zusammen und versehen Sie alle Blätter **gut lesbar** mit Ihren Namen und Ihren Matrikel-Nummern sowie dem Namen Ihres Tutors. Es können **maximal vier** Übungsteilnehmer eine gemeinsam erarbeitete Lösung einreichen.

Lesen Sie vor dem Anfertigen einer Lösung die Aufgaben gründlich durch. Verwenden Sie Definitionen und Notationen wie im Rahmen dieser Veranstaltung vorgestellt. Für Fragen bezüglich dieser Übung können Sie das Forum zur Veranstaltung sowie die Sprechstunden der Tutoren nutzen.

Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Mit der Abgabe einer Lösung bestätigen Sie, dass Sie die alleinigen Autoren des gesamten Materials sind. Bei Unklarheiten zu diesem Thema finden Sie weiterführende Informationen unter

<http://www.informatik.tu-darmstadt.de/Plagiarism>  
oder sprechen Sie Ihren Betreuer an.

### Aufgabe 1.1 (Implementieren und Spezifizieren) (12 Punkte)

Betrachten Sie die folgende Datenstruktur für S-Expressions.

```
structure sexpr[@I] <=  
  nil,  
  atom(data : @I),  
  cons(car : sexpr[@I], cdr : sexpr[@I])
```

- Schreiben Sie eine Prozedur `maximum(s : sexpr[N]) : N`, die das größte Element eines S-Expressions natürlicher Zahlen berechnet. Falls der S-Expression keine Elemente besitzt, soll 0 geliefert werden.
- Schreiben Sie eine Prozedur `isIn(i : @I, s : sexpr[@I]) : bool`, die überprüft, ob `i` im S-Expression `s` vorkommt.
- Formulieren Sie die folgenden Eigenschaften von `maximum` durch  $\mathcal{L}$ -Lemmata:
  - Das Maximum eines S-Expressions ist 0, oder Element des S-Expressions.
  - Das Maximum eines S-Expressions ist größer oder gleich jedem Element des S-Expressions.

### Aufgabe 1.2 (Kalküle) (17 Punkte)

Betrachten Sie die Sprache  $L = \{w \mid |w|_a = |w|_b = |w|_c\} \subset \{a, b, c\}^*$ . Sie sollen Kalküle entwerfen, um das Wortproblem von  $L$  zu lösen. In Ihren Kalkülregeln dürfen  $|w|_a$ ,  $|w|_b$  und  $|w|_c$  **nicht** vorkommen.

- Geben Sie einen nicht-deterministischen Kalkül an, um für Wörter aus  $\{a, b, c\}^*$  festzustellen, ob das Wort in  $L$  ist. Als Sprache ihres Kalküls soll  $\{a, b, c\}^*$  genommen werden.
- Geben Sie einen deterministischen Kalkül an, um für Wörter aus  $\{a, b, c\}^*$  festzustellen, ob das Wort in  $L$  ist. Als Sprache ihres Kalküls soll  $\{a, b, c\}^* \times \mathbb{N}^3$  genommen werden. Ihr Kalkül soll zeichenweise vorgehen.

- (c) Lösen Sie die folgenden Instanzen des Wortproblems für  $L$  mit Hilfe Ihrer Kalküle, d. h. geben Sie jeweils an, ob das Wort in  $L$  ist und belegen Sie dies durch Angabe geeigneter Herleitungen beider Kalküle.
- abac
  - bcaacb
- (d) (**Bonus**) Geben Sie einen deterministischen Kalkül an, um für Wörter aus  $\{a, b, c\}^*$  festzustellen, ob das Wort in  $L$  ist. Als Sprache ihres Kalküls soll  $\{a, b, c\}^* \times \{A, B, C\}$  genommen werden. Ihr Kalkül soll zeichenweise vorgehen.

### Aufgabe 1.3 (Testen von Lemmata) (11 Punkte)

Betrachten Sie die Prozedur `mix`:

```
function mix(k : list[@I], l : list[@I]) : list[@I] <=
if ? $\emptyset$ (k)
  then l
  else if ? $\emptyset$ (l)
    then k
    else hd(k) :: mix(tl(l), tl(k))
  end_if
end_if
```

Die folgenden Lemmata lassen sich in  $\checkmark$ eriFun mit „Generate Lemma“ aus dem Menü „Program“ für `mix` erzeugen. Zur Lösung dieser Aufgabe sollten Sie jedoch  $\checkmark$ eriFun nicht benötigen.

- (a) lemma mix is associative <=  $\forall x, y, z : \text{list}[@I]$   
 $\text{mix}(\text{mix}(x, y), z) = \text{mix}(x, \text{mix}(y, z))$
- (b) lemma mix is commutative <=  $\forall x, y : \text{list}[@I]$   
 $\text{mix}(x, y) = \text{mix}(y, x)$
- (c) lemma mix is idempotent <=  $\forall x, y : \text{list}[@I]$   
 $\text{if}\{x = y, \text{mix}(x, y) = x, \text{true}\}$
- (d) lemma mix is left-cancelable <=  $\forall x1, x2, y, z : \text{list}[@I]$   
 $\text{if}\{x1 = x2, \text{if}\{\text{mix}(x1, y) = \text{mix}(x2, z), y = z, \text{true}\}, \text{true}\}$
- (e) lemma mix is right-cancelable <=  $\forall x1, x2, y, z : \text{list}[@I]$   
 $\text{if}\{x1 = x2, \text{if}\{\text{mix}(y, x1) = \text{mix}(z, x2), y = z, \text{true}\}, \text{true}\}$
- (f) lemma mix is injective <=  $\forall x1, x2, y1, y2 : \text{list}[@I]$   
 $\text{if}\{\text{mix}(x1, x2) = \text{mix}(y1, y2), \text{if}\{x1 = y1, x2 = y2, \text{false}\}, \text{true}\}$

Geben Sie für jedes dieser Lemmata an, ob es gilt. Geben Sie für jedes ungültige Lemma ein Gegenbeispiel in Form einer widerlegenden Substitution an.