

Formale Grundlagen der Informatik 3

Prof. Dr. Christoph Walther / Simon Siegler, Nathan Wasser
Technische Universität Darmstadt — Wintersemester 2010/11

Praktische Übung 2

Version 1 vom 19. November 2010

Diese Übung wird in der **KW 49 (06.–10.12.2010)** testiert. Reichen Sie dazu Ihre mit `veriFun 3.2.2` erstellte `.vf`-Lösungsdatei bis spätestens **Freitag, 03.12.2010 23:59 Uhr** online auf der Abgabeplattform ein. Später eintreffende Lösungen können nicht akzeptiert werden.

Bitte reservieren Sie bis zur oben genannten Frist auf der Abgabeplattform einen Testattermin für die Testwoche, zu dem alle Mitglieder Ihrer Praktikumsgruppe erscheinen können. Das Erscheinen zum vereinbarten Testattermin ist eine notwendige Voraussetzung, um das Testat zu bekommen.

Bei Fragen und Problemen können Sie neben den Poolbetreuungszeiten der Tutoren auch das Forum zur Veranstaltung 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 der alleinige Autor 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.

Vorbemerkung

In dieser praktischen Übung betrachten wir den deterministischen Kalkül aus der 2. Präsenzübung, welcher für zwei Zeichenketten $\alpha, \beta \in \Sigma^*$ überprüft ob α eine Teilkette von β ist. Sie sollen in dieser praktischen Übung beweisen, dass der Kalkül korrekt und vollständig ist.

Ein deterministischer Kalkül lässt sich leicht durch eine rekursive Prozedur implementieren, wie die Prozedur `isSubstring` in `Praxis2.vf` für den beschriebenen Kalkül. Überzeugen Sie sich zunächst selbst, dass die Prozedur korrekt ist, indem Sie sie im Interpreter testen. Schauen Sie sich auch die benötigten Hilfsprozeduren an, um sich damit vertraut zu machen. (Den Ordner `isSubstring termination` dürfen Sie ignorieren.)

Sie werden zum Lösen der Aufgaben Hilfslemmata schreiben und beweisen müssen.

Bauen Sie bei der Lösung der nachfolgenden Aufgaben auf der Datei `Praxis2.vf` auf, die bereits einige grundlegende Definitionen enthält.

Praktische Übung 2.1 (Eigenschaften der Relation \in)

Die Prozedur \in ist eine Implementierung der Relation „ist Teilkette von“, also $k \in l$ genau dann, wenn k eine Teilkette¹ von l ist. Überzeugen Sie sich zunächst selbst, dass dies der Fall ist. Betrachten Sie dazu auch die Hilfsprozedur \rightarrow und überlegen Sie sich, welche Relation diese Prozedur implementiert.

Die Reflexivität und Transitivität von \in werden Sie für den Beweis der Korrektheit und Vollständigkeit des Kalküls benötigen.

- Schreiben Sie im Kommentarfenster der Prozedur \rightarrow (Menü **Program \ Comment...**) welche Relation sie implementiert.
- Beweisen Sie das Lemma „ \in is reflexive“.
- Beweisen Sie das Lemma „ \in is transitive“. Hierbei hilft eventuell das Öffnen bestimmter Prozeduraufrufe mittels der Beweisregel **Unfold Procedure**. Überlegen Sie sich gut, welche Prozeduraufrufe Sie öffnen wollen, da dies sehr leicht den Goalterm unübersichtlich machen kann.

Praktische Übung 2.2 (Korrektheit des Kalküls)

In dieser Aufgabe sollen Sie die Korrektheit des Kalküls beweisen. Behalten Sie beim Beweis immer die Mengen der Hypothesen und Induktionshypothesen, die Sie für den momentanen Goalterm annehmen dürfen, im Auge. Oft sieht ein Goalterm komplizierter aus, als er tatsächlich ist. Mit **Use Lemma** können neben bewiesenen und unbewiesenen Lemmata auch die Induktionshypothesen benutzt werden.

- Beweisen Sie das Hilfslemma „ $isSubstring(k, l, acc) \rightarrow acc \langle \rangle k \in acc \langle \rangle l$ “ (\dagger).

Hinweis: Erstellen und beweisen Sie ein Hilfslemma $snoc(k, x) \langle \rangle l = ?$, wobei Sie $?$ so wählen, dass (1.) das Lemma stimmt und (2.) die Prozedur $snoc$ in $?$ nicht vorkommt.

- Beweisen Sie mit Hilfe des Hilfslemmas (\dagger) das Hauptlemma „ $isSubstring$ is correct“.

Praktische Übung 2.3 (Vollständigkeit des Kalküls)

In dieser Aufgabe sollen Sie die Vollständigkeit des Kalküls beweisen. Achten Sie auch hier immer auf die Hypothesen und Induktionshypothesen, die Sie beim Beweis zur Verfügung haben.

- Beweisen Sie das Hilfslemma „ $acc \langle \rangle k \in acc \langle \rangle l \rightarrow isSubstring(k, l, acc)$ “ (\ddagger).
- Beweisen Sie mit Hilfe des Hilfslemmas (\ddagger) das Hauptlemma „ $isSubstring$ is complete“.

¹Für $\alpha, \beta \in \Sigma^*$ ist α eine Teilkette von β genau dann, wenn $\beta = \gamma_1 \alpha \gamma_2$, für $\gamma_1, \gamma_2 \in \Sigma^*$.