Universität Bremen

Korrekte Software: Grundlagen und Methoden SoSe 2021

Serge Autexier
Christoph Lüth

9. Übungsblatt

Ausgabe: 24.06.21
Abgabe: 29.06.21 10:00

Dieses Übungsblatt ist ein PDF-Formular. Bitte in einem PDF-Viewer Ihrer Wahl ausfüllen, abspeichern und in Ihrem Gitlab-Abgabe-Repository committen. Alternativ können Sie die Lösungen in der Markdown-Datei `uebung-XX.md` eintragen und diese committen.

Gruppe:

Name: Matrikelnummer:
Name: Matrikelnummer:
Name: Matrikelnummer:

9.1 *Spezifikation von Zeichenketten*

Zeichenketten sind in C einfach Felder von **char**. (Für diese Übung sind **char** und **int** erst einmal das gleiche.) Wir können sie zu Sequenzen von Zeichen, oder Zeichenketten, abstrahieren, so wie wir Feldern von ganzen Zahlen zu Sequenzen (Listen) von ganzen Zahlen abstrahiert haben.

In dieser Übung wollen wir ine nützliche Funktion auf Zeichenketten spezifizieren. Dazu spezifizieren wir erst einmal die Abstraktionsfunktion string(a), die gegeben ein Feld von **char** eine Sequenz von Zeichen zurückgibt, wobein ein Zeichen 0 (`'\0'`, ASCII-Zeichenl NULL) das Ende der Zeichenkette darstellt, d.h. eine Zeichenkette ist eine Liste von Zeichen, die nicht null sind. Spezifizieren Sie so die Abstraktionsfunktion string(a) unter Nutzung der elementaren Abstraktionsfunktion seq:

Hinweis: Starten Sie mit $string(a) = xs \iff \ldots$ und spezifizieren Sie dann, unter welchen Umständen xs das Ergebnis von $seq(a, n)$ sein kann.

9.2 *Wörter*

Als Anwendungsbeispiel wollen wir jetzt eine Funktion spezifizieren, die eine Zeichenkette in eine Liste von Wörtern zerlegt. Dazu definieren wir ein Wort als eine Zeichenkette, die kein Leerzeichen[1] enthält.

Als Beispiel für rekursive Funktionsdefinitionen können die Funktionen aus der Vorlesung dienen, oder folgende Funktion, welche aus einer Liste von Listen eine lange Liste macht:

$$concat([]) == []$$
$$concat(xs : xss) == xs{+}{+}(concat(xss))$$

(1) Spezifizieren Sie ein Prädikat nospace(w), welches wahr ist gdw. das Argument w kein Leerzeichen enthält (also ein Wort ist). Nutzen Sie dazu das Prädikat is_space(c), welches wahr ist, wenn das Argument c ein Leerzeichen ist:

(2) Spezifizieren Sie eine Funktion remspace(s), welche alle Leerzeichen aus einer Zeichenkette entfernt:

[1] "white space", also Leerzeichen, Tabulator oder Zeilenvorschub

(3) Spezifizieren Sie, dass das Ergebnis von remspace(s) keine Leerzeichen enthält und damit ein Wort ist:

(4) Spezifizieren Sie damit eine Funktion words(s), welche aus einer Zeichenkette s eine Liste von Zeichenketten ws erzeugt, so dass:

- alle Elemente von ws Wörter sind, und
- sie verkettet daselbe ergeben wie s ohne die Leerzeichen.

Hinweis:

Nutzen Sie die in der Vorlesung vorgestellten Funktionen $++$, : etc. auf Sequenzen; ggf. können Sie auch weitere Hilfsfunktionen definieren. Beachten Sie, dass Ihre Definitionen nicht rekursiv bzw. ausführbar zu sein brauchen.

Die Funktionen concat und words gibt es mit der hier intendierten Funktionalität auch in Haskell.