Institut für Informatik
der Universität München
Prof. Dr. M. Hofmann
Dr. J. Johannsen, Dr. M. Lange

WS 03/04

13.11.03

Übungen zur Vorlesung
Informatik I

Blatt 4

Abgabe der Hausaufgaben spätestens am 24.11.03, 11:00 Uhr. Programmieraufgaben über
`http://miles.tcs.informatik.uni-muenchen.de/info1/abgabe.php`,
schriftliche Aufgaben auf Papier im Briefkasten in der Theresienstraße 39, 1. Stock. Notieren Sie Namen, Matrikelnummern und Ihre Übungsgruppe auf den Blättern. Bearbeitung in Gruppen zu max. 3 Personen ist zulässig. Besprechung der Aufgaben in den Übungen ab 1.12.03.

Auf diesem Blatt gibt es erstmals Programmieraufgaben. Diese werden in elektronischer Form abgegeben. Beachten Sie, dass jede Programmieraufgabe in einer separaten Datei abzugeben ist. Der Name der entsprechenden Datei wird vorgegeben. Halten Sie sich unbedingt daran, und beachten Sie auch Groß- und Kleinschreibung. Beachten Sie auch, dass das Abgabesystem nur syntaktisch korrekte Ocaml-Programme akzeptiert. Sie können Ihre Dateien auf syntaktische Korrektheit überprüfen, indem Sie sie mit dem Kommando `ocamlc` *dateiname* kompilieren. Erhalten Sie keine Fehlermeldung, so ist Ihr Programm syntaktisch korrekt.

Programmieraufgabe P-17 (`istprim.ml`): **5 Punkte**
Implementieren Sie in Ocaml die Funktion **istPrim**: **nat** → **bool**, wie sie auf den Vorlesungsfolien Seite 62 angegeben ist.

Programmieraufgabe P-18 (`fixpunkt.ml`): **5 Punkte**
Ein Wert y heißt ein Fixpunkt der Funktion f, wenn $f(y) = y$ gilt. Implementieren Sie in Ocaml eine Funktion **fixpunkt**: $(\alpha \to \alpha) \times \alpha \to \alpha$, die als Argumente eine Funktion f und einen Startwert x erhält und nach einem Fixpunkt von f sucht. Genauer ist **fixpunkt**(f, x) wie folgt definiert: falls es ein $n \in \mathbb{N}$ gibt, so dass $f^n(x)$ ein Fixpunkt von f ist, so soll dieser zurückgegeben werden, andernfalls ist **fixpunkt**(f, x) undefiniert.

Programmieraufgabe P-19 (`zahlenwert.ml`): **5 Punkte**
Implementieren Sie in Ocaml die folgende Funktion **zahlenwert**(s: **String**, b: **nat**) **nat**. Das erste Argument ist eine Zeichenkette, bei der Sie davon ausgehen können, dass sie nur aus Ziffern besteht. Das zweite ist eine natürliche Zahl, bei der Sie davon ausgehen können, dass sie zwischen 2 und 10 einschliesslich liegt. Die Zeichenkette s wird als Darstellung einer Zahl zur Basis b gesehen. Der Rückgabewert der Funktion ist der Wert dieser Zahl. Genauer, sei $s = s_n \ldots s_0$ wobei die s_i Ziffern sind. Dann soll $\sum_{i=0}^{n} s_i \cdot b^i$ berechnet werden.

Hinweise: Sie können die Funktion **zahlenwert** durchaus durch mehrere Ocaml-Funktionen implementieren. So werden Sie z.B. sicherlich eine Funktion brauchen, die `char`-Werte in `int`-Werte umwandelt. Zum Verarbeiten der Strings brauchen Sie evtl. die folgenden Funktionen:

`String.length: string -> int`	liefert die Länge einer Zeichenkette
`String.get: string -> int -> char`	liefert das Zeichen an der angegebenen Position
`String.sub: string -> int -> int -> string`	liefert einen Teilstring; die Argumente sind Start- und Endposition innerhalb des gegebenen Strings

Schriftliche Aufgabe S-20: **5 Punkte**

Betrachten Sie die folgende Grammatik in BNF für `float`-Literale.

$$
\begin{aligned}
\langle\texttt{float}-\texttt{literal}\rangle &::= [\langle\texttt{Vorzeichen}\rangle]\langle\texttt{Mantisse}\rangle[\langle\texttt{Exponent}\rangle] \\
\langle\texttt{Vorzeichen}\rangle &::= - \mid + \\
\langle\texttt{Mantisse}\rangle &::= \langle\texttt{Ziffer}\rangle^{+}[.\langle\texttt{Ziffer}\rangle^{*}] \\
\langle\texttt{Exponent}\rangle &::= \{e \mid E\}[\langle\texttt{Vorzeichen}\rangle]\langle\texttt{Ziffer}\rangle^{+} \\
\langle\texttt{Ziffer}\rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{aligned}
$$

Verbessern Sie diese Grammatik so, dass `float`-Literale immer entweder einen Dezimalpunkt oder ein `e` oder ein `E` enthalten.