Übungen 8: Ein- und Ausgabe

Programmiertechniken in der Computerlinguistik I · Wintersemester 2001/2002

1. Leerzeichen

Das vordefinierte Prädikat tab/1 schreibt soviele Leerschläge heraus, wie sein Argument enthält. Zeige, dass dieses Prädikat leicht selbst definiert werden kann, indem du ein äquivalentes my_tab/1 implementierst.

2. Gross- und Kleinschreibung

Trotz amtlicher Rechtschreibereform fürs Deutsche trifft man in weniger amtlichen Bereichen – wie etwa in Emails – auf ganz unterschiedliche Schreibungen. Einige schreiben alles klein, einige nur den Satzanfang gross, einige wie im Englischen Eigennamen gross. Daneben gibt's TIppfehler, oder Grossschreibung wird zum Ausdruck von Emotionen (SCHREIEN) verwendet.

a) Definiere die Prädikate ucfirst/2 (upper case) und lcfirst/2 (lower case), die genau dann wahr sein sollen, wenn das 2. Argument die Variante mit gross- bzw. kleingeschriebenen Anfangsbuchstaben des 1. Arguments ist. Die Argumente sollen Atome sein und die Prädikate müssen nur im Modus ucfirst(+Atom,?Atom) bzw. lcfirst(+Atom, ?Atom) funktionieren. Die deutschen Umlaute ä,ö,ü... sollen ebenfalls berücksichtigt werden.

```
?- ucfirst(gern,X).
X = Gern
yes

?- lcfirst('Äpfel', X).
X = äpfel
yes
```

Tipp: Mit einem geeigneten Hilfsprädikat, das die Beziehung zwischen gross- und kleingeschriebenen Varianten der Buchstaben ausdrückt, wird der Schreibaufwand in Übung 2 markant verringert. Denke auch daran, dass in ASCII und ISO-Latin 8859-1 die Codes von Gross- und Kleinbuchstaben systematisch miteinander verbunden sind!

b) Definiere die Prädikate uc/2 und lc/2, die genau dann wahr sein sollen, wenn das 2. Argument die gross- bzw. kleingeschriebene Variante des 1.Arguments ist.

```
?- uc(sbb, X).
X = SBB
yes

?- lc('Wenn Fliegen hinter Fliegen fliegen.', X).
X = 'wenn fliegen hinter fliegen fliegen.'
yes
```

c) Definiere das Prädikat lc_file(+InputDatei,+OutputDatei), das alle Prologterme aus InputDatei einliest und die kleingeschriebene Variante dieser Terme in OutputDatei hineinschreibt. Die OutputDatei wird also erzeugt!

Beispiel: In der Datei "bsp.txt" auf Laufwerk C befindet sich der Text der linken Spalte von untenstehender Tabelle. Nach dem Aufruf von

```
?- lc_file('C:/bsp.txt', 'C:/lcbsp.txt').
yes
```

enthält die Datei "lcbsp.txt" auf Laufwerk C den Text der rechten Spalte von untenstehender Tabelle.

Inhalt von Datei 'C:/bsp.txt'	Inhalt von Datei 'C:/lcbsp.txt'
'Wenn'. hinter. 'Fliegen'. 'Fliegen'. fliegen. ','. fliegen. 'Fliegen'. 'Fliegen'. nach. '.'.	wenn. hinter. fliegen. fliegen. fliegen. ','. fliegen. fliegen. fliegen. nach. '.'.

Tipps: Verwende write_canonical/1 zum Herausschreiben der Prologterme, damit das Komma und der wörtliche Punkt korrekt zitiert werden.

```
?- write('.').
.
yes
?- write_canonical(',').
','
yes
```

Wer mit SICStus Prolog unter MacOS arbeitet, muss die Dateinamen, die sich nicht auf das momentan aktuelle Arbeitsverzeichnis beziehen (dieses wird im Fenster "Console" oben angezeigt), im UNIX-Stil angeben. D.H. aus der Datei "bsp.txt" auf dem Volumen "MacHD" die Datei "lcbsp.txt" zu erzeugen, muss der Aufruf

```
?- lc_file('/MacHD/bsp.txt', '/MacHD/lcbsp.txt').
yes
```

gemacht werden.

Unter http://www.ifi.unizh.ch/cl/siclemat/lehre/ws0102/pcl1/bsp.txt findest du die Beispieldatei.