Andreas Podelski
Fachbereich 14 – Informatik
Universität des Saarlandes,
Max-Planck-Institut für Informatik

Programmiersprachen (WS 1996/97)

31. Oktober 1996

Übungsblatt 2

Abdelwaheb Ayari
Hubert Baumeister

Aufgabe 1: (3 Punkte)

1. Definiere einen Datentyp für natürliche Zahlen in unärer Darstellung, also mittels Konstruktoren `zero` und `succ` (Nachfolger).
   Zum Beispiel ist 2 in unärer Darstellung `succ(succ(zero))`.
2. Schreibe ML-Funktionen `even` und `odd`, die für eine natürliche Zahl in unärer Darstellung festellen ob sie gerade (even) oder ungerade (odd) ist. Verwende bei der Definition von `even` die Funktion `odd` und umgekehrt.
3. Reduziere die Terme `even (succ(succ(succ(zero))))` und `even (succ(succ(zero)))` mit Hilfe der in der Vorlesung angegebenen Reduktionssemantik.

Aufgabe 2: (2 Punkte)
Zeige durch strukturelle Induktion, daß `append`(l_1, l_2) für alle endlichen Listen l_1 und l_2 terminiert. (Hinweis: Zeige, daß der Wert von `append`(l_1, l_2) ein Konstruktorterm ist.)

Aufgabe 3: (5 Punkte)
Implementiere ein Stringersetzungssystem in SML. Stelle dazu eine Funktion
`tes : (string * string) list * string -> string` bereit. (Hinweis: Implementiere ein Listenersetzungssystem für Listen von Zeichen. Benutze zum Umwandeln von strings in Zeichenlisten die SML-Funktionen `explode : string -> string list` und `implode : string list -> string`.)

- Das erste Argument `(string * string) list` ist eine Liste L von Stringersetzungsregeln $l \to r$, wobei l und r strings sind.
- Das zweite Argument ist ein String, der mit dem Stringersetzungssystem reduziert werden soll. Der reduzierte String ist das Ergebnis der Funktion.

Beispiel: `tes ([("c","d"),("ab","c")], "abab")` liefert `"dd"`.

1. Definiere ein Stringersetzungssystem welches in einem String aus a's und b's die b's nach links und die a's nach rechts verschiebt.

2. Definiere ein Stringersetzungssystem welches in einem nichtleeren String aus a's und b's den String `"a"` generiert wenn eine ungerade Anzahl von a's in dem String enthalten ist, ansonsten den String `"b"` generiert.

Wende `tes` und die jeweiligen Stringersetzungssystem auf die Strings

1. `"aaabbbbbababababbaababbab"`
2. `"bababaabbababbabaaabbababababab"`
3. `"bababbbaaaababbbabaababababbbbbbbbaaaaa"`
4. `"bbbbbbaabbbaaaaaaaaabbbbbbaaabaaaaaaaaaabbbaaaaa"`

an.