Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik (20)
Johann Wolfgang Goethe-Universität Frankfurt am Main

Funktionale Programmierung I

Sommersemester 2000

Übungsblatt Nr. 6

Abgabe: Dienstag 16. Mai 2000 vor der Vorlesung

Aufgabe 1 (15 Punkte)

In der Vorlesung wurde die parallele $\rightarrow_1$-Reduktion vorgestellt. Geben Sie Terme s_k, t_k an, so daß sowohl $s_k \rightarrow_1 t_k$ als auch

$$\exists u_k : \; s_k \rightarrow^{no,k} u_k \rightarrow_1 t_k$$

gilt, wobei $\rightarrow^{no,k}$ für k Reduktionsschritte in Normalordnung steht.

Aufgabe 2 (15 Punkte)

In Haskell können Infixoperatoren mittels `infix`, `infixl` und `infixr` erklärt und mit einer Präzedenz sowie Assoziativität versehen werden.

Zum Beispiel ist in der Prelude für die append-Funktion `(++)`

```
infixr 5 ++
```

vorgegeben. Warum ist die rechtsassoziative Auswertung für `(++)` sinnvoll? Begründen Sie die Komplexitätsunterschiede.

Aufgabe 3 (20 Punkte)

In der Vorlesung wurden kontextuelle Approximation und Äquivalenz eingeführt. Beweisen oder widerlegen Sie folgende Aussagen:

1. $(\lambda xs.\texttt{case}_{List}\ xs\ (\texttt{Nil -> Nil})\ (\texttt{Cons}\ y\ ys\ \texttt{->}\ \texttt{Cons}\ y\ ys)) \ \leq_c\ (\lambda xs.xs)$
2. $(\lambda xs.\texttt{case}_{List}\ xs\ (\texttt{Nil -> Nil})\ (\texttt{Cons}\ y\ ys\ \texttt{->}\ \texttt{Cons}\ y\ ys))\ zs \ \sim_c\ (\lambda xs.xs)\ zs$
3. $(\lambda xs.\texttt{case}_{List}\ xs\ (\texttt{Nil -> Nil})\ (\texttt{Cons}\ y\ ys\ \texttt{->}\ \texttt{Cons}\ y\ ys)) \ \sim_c\ (\lambda xs.xs)$

Hinweis: Zusammenhang zwischen 2 und 3 über die Kongruenzeigenschaft von $\sim_c$