

Übungsblatt 4

Abgabe bis: **Donnerstag, 23. Mai 2002, 12.00 Uhr**

Abgabemöglichkeiten:

- Büro D1.09 (Andreas Abel)
- Büro Z1.05 (Sandra Nentwich-Mertel)
- email `abel@informatik.uni-muenchen.de`

Aufgabe 1: [Reduktionsstrategien]

In der Vorlesung wurden folgende Abkürzungen definiert:

$$\begin{aligned}\mathbf{true} &= \lambda t \lambda e. t \\ \mathbf{false} &= \lambda t \lambda e. e \\ \mathbf{if} &= \lambda b \lambda t \lambda e. b t e\end{aligned}$$

Reduzieren Sie die folgenden Terme zu Normalform (überspringen Sie dabei keine wesentlichen Schritte), und zwar einmal mit der Strategie “call-by-name” und einmal mit “call-by-value”. (Insgesamt sind also vier Reduktionsfolgen gefragt.)

1. $t_1 = \mathbf{if\ false\ (if\ true\ (if\ false\ u\ v)\ w)\ u.}$
2. $t_2 = (\lambda x. \mathbf{if\ x\ x\ x}) (\mathbf{if\ true\ false\ true})$

Dabei sind u , v und w beliebige λ -Terme in Normalform.
(8 Punkte)

Aufgabe 2: [Boolsche Funktionen im λ -Kalkül]

Geben Sie λ -terme an für die boolschen Funktionen **and**, **or** und **imp** (logische Implikation).

(3 Punkte)

(Bitte wenden!)

Aufgabe 3: [Currying]

Für diese Aufgabe erweitern wir den λ -Kalkül um Ganzzahl-Konstanten 0, 1, -1, ..., Addition + und Multiplikation *. Gecurryte Fassungen der arithmetischen Operationen sind:

$$\begin{aligned}\mathbf{add} &= \lambda x \lambda y. x + y \\ \mathbf{mult} &= \lambda x \lambda y. x * y\end{aligned}$$

Finden Sie die kürzesten Implementierungen der folgenden Funktionen durch λ -Terme. Verwenden Sie dazu `add` und `mult`.

- a) Nachfolger (erhöht das Argument um eins).
- b) Verdopplung.

(2 Punkte)

Aufgabe 4: [Optionstyp]

In SML gibt es den folgenden Datentyp:

```
datatype 'a option =  
  NONE  
  | SOME of 'a
```

Ein Bewohner dieses Typs ist entweder `NONE` oder `SOME a` mit einem Datum `a`. Dieser Typ eignet sich für die Implementation partieller Funktionen, die manchmal ein ordentliches Ergebnis liefern (`SOME a`) und manchmal keines (`NONE`). (In Haskell heisst dieser Datentyp `Maybe`.)

Kodieren Sie diesen Datentyp im λ -Kalkül. Geben Sie λ -Terme `some`, `none` (Konstruktoren) und `case` (Destruktor) an mit folgendem Verhalten.

$$\begin{aligned}\mathbf{case\ none\ } t_1\ t_2 &=_{\beta} t_1 \\ \mathbf{case\ (some\ } a) t_1\ t_2 &=_{\beta} t_2\ a\end{aligned}$$

(3 Punkte)

Viel Erfolg!