

Grundlagen der Programmkonstruktion

Freiwillige Übung 3

1 Lambda

1.1 Lambda-Ausdruck 1

Reduzieren Sie den folgenden Ausdruck mit der β -Regel, bis Sie eine Normalform erhalten, oder begründen Sie, warum es keine Normalform gibt.

$$(\lambda u.0) ((\lambda v.(v v)) (\lambda w.(w w)))$$

1.2 Lambda-Ausdruck 2

Reduzieren Sie den folgenden Ausdruck mit der β -Regel, bis Sie eine Normalform erhalten. Hinweis: Sie müssen zumindest einmal mit Hilfe der α -Regel eine gebundene Variable umbenennen.

$$(\lambda v.(\lambda v.(v * v))(v + v))$$

Versuchen Sie dasselbe mit folgendem Ausdruck:

$$(\lambda v.(\lambda v.(4 * v))(v * v))$$

Beschreiben Sie die Funktion, die diese Lambda-Ausdrücke berechnen.

1.3 Lambda-Ausdruck 3

Reduzieren Sie den folgenden Ausdruck mit der β -Regel, bis Sie eine Normalform erhalten:

$$(\lambda p.\lambda a.\lambda b.((p b) a))(\lambda x.\lambda y.y)$$

Können Sie Ihr Ergebnis mit der α oder η -Regel in den Ausdruck $\lambda x.\lambda y.x$ umwandeln?

1.4 Lambda-Ausdruck 4

Gegeben ist der folgende Ausdruck

$$F = (\lambda u.(\lambda v.v == 0 ? 1 : 2 * ((u u)(v - 1))))$$

Reduzieren Sie den Ausdruck “ $(F F) 2$ ” bis Sie eine Normalform erhalten. Werten Sie mathematische Ausdrücke wie $2 - 1$ bzw. den ternären Operator sofort aus.

2 Grammatik

Gegeben ist die folgende Grammatik

```
expr: atom
     | '(' { expr } ')'
     ;

atom: id
     | num
     ;

num: '0'..'9'{'0'..'9'}
id: ('a'..'z' | 'A'..'Z') {'a'..'z' | 'A'..'Z' | '0'..'9'}
```

Diese Grammatik beschreibt (fast) alle gültigen Ausdrücke der Programmiersprache LISP.

2.1 Aufgabe 1

Sind die folgenden Ausdrücke gültige `expr`?

- `()`
- `(1 (2 (3 4)))`
- `() ()`
- `(a1 2)`
- `3`
- `(a (1) (2))`
- `(3)`
- `(list 1 2 (list 3 4))`
- `(-3)`
- `(lambda (x) (f x))`

2.2 Aufgabe 2

Man kann in LISP Lambda-Ausdrücke verwenden. Der Ausdruck `(lambda (v) e)` entspricht dabei dem Ausdruck $\lambda v.e$.

Wandeln Sie die folgenden Lambda-Ausdrücke in ihre entsprechenden LISP-Ausdrücke um:

- $\lambda x.x$
- $\lambda f.(\lambda y.(f y))$
- $(\lambda x.x) 1$
- $\lambda f.(\lambda y.(f y)) \sin$
- $(\lambda x.0) 3$
- $\lambda p.(\lambda a.(\lambda b.((p b) a)))$

Wandeln Sie die β -Reduktion wenn möglich auf die LISP-Ausdrücke an.