

Semantik von Programmiersprachen I

Sommersemester 2014

Übungsblatt 1

Aufgabe 1

λ -Abstraktionen dienen zur Beschreibung von Funktionen. Welche Funktionen werden durch die folgenden λ -Abstraktionen beschrieben? Machen Sie sich jeweils klar, wie Definitions- und Zielbereich der Funktion aussehen, d.h. auf welche Argumente die Funktion angewandt werden kann, ohne dass ein Laufzeitfehler entsteht, und wie das Resultat in diesem Fall aussieht.

- a. $\lambda x. 0$
- b. $\lambda x. x$
- c. $\lambda x. \text{if } x > 0 \text{ then } x \text{ else } 0 - x$
- d. $\lambda x. x \bmod 2 = 0$
- e. $\lambda x. \text{if } x \text{ then } \textit{false} \text{ else } \textit{true}$
- f. $\lambda x. \lambda y. x$
- g. $\lambda x. \lambda y. \text{if } x \geq y \text{ then } x - y \text{ else } y - x$
- h. $\lambda x. \lambda y. \text{if } x \text{ then } y \text{ else } \textit{false}$
- i. $\lambda f. \lambda x. f x$
- j. $\lambda f. \lambda x. f (f x)$
- k. $\lambda f. \lambda g. \lambda x. f (g x)$
- l. $\lambda p. \lambda q. \lambda x. \text{if } p x \text{ then } q x \text{ else } \textit{false}$
- m. $\lambda f. \lambda g. \lambda p. \lambda x. \text{if } p x \text{ then } f x \text{ else } g x$

Aufgabe 2

Bestimmen Sie (wie bei dem einführenden Beispiel in der Vorlesung) die big step Semantik der folgenden Ausdrücke. Machen Sie sich vorher intuitiv klar, ob der Ausdruck ein Ergebnis liefert und—wenn ja—welches.

- a. **let** $distance = \lambda x. \lambda y. \mathbf{if} \ x \geq y \ \mathbf{then} \ x - y \ \mathbf{else} \ y - x$
in $distance \ 0$
- b. **let** $twice = \lambda f. \lambda x. f \ (f \ x)$
in $twice \ (\lambda x. x * x) \ 5$
- c. $(\lambda x. x \ x) \ (\lambda x. x \ x)$
- d. $(\lambda x. 0) \ (1/0)$
- e. **let** $x = 1/0$ **in** 0
- f. **if** $1 > 0$ **then** 1 **else** $1/0$
- g. **let** $f = \lambda x. 1/0$ **in** f
- h. **let** $f = \lambda x. 1/0$ **in** $f \ 0$
- i. **let** $f = \lambda x. 1/0$ **in** $\lambda x. f \ (f \ x)$

Aufgabe 3

Die in der Vorlesung definierte big step Semantik benutzt das *call by value* Prinzip (auch *eager evaluation* genannt) bei der Parameterübergabe, d.h. der aktuelle Parameter wird ausgewertet, bevor er für den formalen Parameter eingesetzt wird. Die Alternative ist das sogenannte *call by name* Prinzip (auch *lazy evaluation*), wo der aktuelle Parameter eingesetzt wird, *ohne* dass man ihn vorher auswertet

- a. Ändern Sie die big step Semantik aus der Vorlesung so ab, dass Sie eine *call by name* Semantik erhalten.
- b. Geben Sie einen Ausdruck an, der in einer der beiden Semantiken ein Ergebnis liefert und in der anderen nicht.