Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Simon Wimmer

Lambda Calculus
Winter Term 2017/18
Solutions to Exercise Sheet 6

Exercise 1 (Reduction Relation with Closures)

For the evaluation of lambda terms that is closer to evaluation of programs in functional programming languages, one usually replaces textual substitution $t[v/x]$ with a more lazy approach that records the binding $x \mapsto v$ in an environment. These bindings are used whenever need the value of a variable v.

In this approach abstractions $\lambda x.t$ do not evaluate to themselves, but to a pair $(\lambda x.t)[e]$, where e is the current environment. We call such pairs function *closures*.

a) Define a big-step reduction relation for the lambda calculus with function closures and environments.

b) Add explicit error handling for the case where the biding of a free variable v cannot be found in the enviroment. Introduce an explicit value **abort** to indicate such an exception in the relation.

Solution

a)

$$\frac{e(x) = v}{e \vdash x \Rightarrow_{cbv} v} \qquad e \vdash \lambda x.t \Rightarrow_{cbv} (\lambda x.v)[e]$$

$$\frac{e \vdash t_1 \Rightarrow_{cbv} (\lambda x.t)[e'] \qquad e \vdash t_2 \Rightarrow_{cbv} v' \qquad e' + (x \mapsto v') \vdash t \Rightarrow_{cbv} v}{e \vdash t_1\, t_2 \Rightarrow_{cbv} v}$$

b) We just need to add rules to propagate errors, and modify the existing rules to ensure that no subexpression evaluates to **abort**.

$$\frac{x \notin e}{e \vdash x \Rightarrow_{cbv} \mathbf{abort}} \qquad \frac{e(x) = v}{e \vdash x \Rightarrow_{cbv} v} \qquad e \vdash \lambda x.t \Rightarrow_{cbv} (\lambda x.v)[e]$$

$$\frac{e \vdash t_1 \Rightarrow_{cbv} (\lambda x.t)[e'] \qquad e \vdash t_2 \Rightarrow_{cbv} v' \qquad e' + (x \mapsto v') \vdash t \Rightarrow_{cbv} v \qquad v' \neq \mathbf{abort}}{e \vdash t_1\, t_2 \Rightarrow_{cbv} v}$$

$$\frac{e \vdash t_1 \Rightarrow_{cbv} \mathbf{abort} \qquad e \vdash t_2 \Rightarrow_{cbv} v}{e \vdash t_1\, t_2 \Rightarrow_{cbv} \mathbf{abort}} \qquad \frac{e \vdash t_2 \Rightarrow_{cbv} \mathbf{abort}}{e \vdash t_1\, t_2 \Rightarrow_{cbv} \mathbf{abort}}$$

Exercise 2 (Reduction Relation with Pattern Matching)

In this exercise, we consider a λ-calculus extended with a special set of constructor values and pattern matching. Constructor values are constructed according to the following grammar:

$$c \;::=\; C\ (c_1, \ldots, c_n) \text{ for } n \geq 0$$

where C is one from a distinguished set of *constructor* symbols.

We illustrate pattern matching by example. The expression

$$\textbf{match}\ C_1\ (\mathsf{false})\ \textbf{with}\ C_2\ () \to \mathsf{true} \mid C_1\ (x) \to x$$

should evaluate to false, while

$$\textbf{match}\ C_2\ (\mathsf{false})\ \textbf{with}\ C_2\ () \to \mathsf{true} \mid C_1\ (x) \to x$$

should evaluate to **abort**.

a) Define a big-step reduction relation for this language.

b) Prove that the two derivations stated informally above are indeed possible in the relation.

Solution

a) We need two things here: a relation that determines wether a pattern matches a constructor term and produces the corresponding variable bindings; and additional cases in the big step semantics that work trough the patterns of a **match**-construct case by case.

The relation for matching a single pattern can look like this:

$$e \vdash v \Downarrow x \Rightarrow e + (x \mapsto v)$$

$$\frac{e \vdash c_1 \Downarrow p_1 \Rightarrow e_1 \qquad e_1 \vdash c_2 \Downarrow p_2 \Rightarrow e_2 \qquad \ldots}{e \vdash C\ c_1\ c_2\ \ldots\ c_n \Downarrow C\ p_1\ p_2\ \ldots\ p_n \Rightarrow e_n}$$

We did not explicitly give the rules that produce and propagate **abort** in the case of a mismatch.

Now we extend the big-step relation from the last exercise:

$$\frac{e \vdash c_1 \Downarrow p_1 \Rightarrow e_1 \qquad e_1 \vdash t_1 \Rightarrow_{cbv} v}{e \vdash \textbf{match}\ c\ \textbf{with}\ p_1 \to t_1 \mid \ldots \Rightarrow_{cbv} v}$$

$$\frac{e \vdash c_1 \Downarrow p_1 \Rightarrow \textbf{abort} \qquad e \vdash \textbf{match}\ c\ \textbf{with}\ p_2 \to t_2 \mid \ldots \Rightarrow_{cbv} v}{e \vdash \textbf{match}\ c\ \textbf{with}\ p_1 \to t_1 \mid p_2 \to t_2 \mid \ldots \Rightarrow_{cbv} v}$$

$$e \vdash \textbf{match}\ c\ \textbf{with}\ [] \Rightarrow_{cbv} \textbf{abort}$$

- We illustrate the first case only. We first show:

$$\frac{\{\} \vdash \mathsf{false} \Downarrow x \Rightarrow \{x \mapsto \mathsf{false}\}}{\{\} \vdash C_1(\mathsf{false}) \Downarrow C_1(x) \Rightarrow \{x \mapsto \mathsf{false}\}}$$

  and

$$\{\} \vdash C_1(\mathsf{false}) \Downarrow C_2() \Rightarrow \mathbf{abort}\,.$$

  Then:

$$\frac{\dfrac{\{\} \vdash C_1(\mathsf{false}) \Downarrow C_1(x) \Rightarrow \{x \mapsto \mathsf{false}\} \qquad \{x \mapsto \mathsf{false}\} \vdash x \Rightarrow_{cbv} \mathsf{false}}{e \vdash \mathbf{match}\ C_1\ (\mathsf{false})\ \mathbf{with}\ C_1\ (x) \to x \Rightarrow_{cbv} \mathsf{false}} \quad e \vdash C_1\ (\mathsf{false}) \Downarrow C_2() \Rightarrow \mathbf{abort}}{e \vdash \mathbf{match}\ C_1\ (\mathsf{false})\ \mathbf{with}\ C_2\ () \to \mathsf{true} \mid C_1\ (x) \to x \Rightarrow_{cbv} \mathsf{false}}$$

Homework 3 (Normal Forms)

Recall the inductive definition of the set NF of *normal forms*:

$$\frac{t \in \mathrm{NF}}{\lambda x.t \in \mathrm{NF}}$$

$$\frac{n \geq 0 \qquad t_1 \in \mathrm{NF} \qquad t_2 \in \mathrm{NF} \qquad \ldots \qquad t_n \in \mathrm{NF}}{x\ t_1\ t_2\ \ldots\ t_n \in \mathrm{NF}}$$

Show that this set precisely captures all normal forms, i.e.:

$$t \in \mathrm{NF} \Leftrightarrow \nexists t'.t \to_\beta t'$$

Homework 4 (Normal Forms & Big Step)

Show:

$$t \in \mathrm{NF} \wedge t \Rightarrow_n u \Longrightarrow u = t$$

Homework 5 (Proofs with Small-steps and Big-steps)

Let $\omega := \lambda x.xx$ and

$$t \ := \ (\lambda x.(\lambda xy.x)zy)(\omega\omega((\lambda xy.x)y))\,.$$

Prove the following:

a) $t \Rightarrow_n z$

b) $t \to_{cbv}^3 t$

c) $t \not\to_{cbn}^+ t$