Fachbereich 6 · Mathematik
Theoretische Informatik

Kurt Sieber
Benedikt Meurer

Theorie der Programmierung
Wintersemester 2006/07

Übungsblatt 2

Aufgabe 1
Denken Sie sich small step Regeln für die Konjunktion $e_1 \&\& e_2$ und die Disjunktion $e_1 \| e_2$ aus, und zeigen Sie, dass es sich bei Ihren Regeln um abgeleitete Regeln handelt.

Aufgabe 2
Die in der Vorlesung definierte small 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. Im Gegensatz dazu bedeutet *call by name* (auch *lazy evaluation* genannt), dass der aktuelle Parameter eingesetzt wird, *ohne* dass man ihn vorher auswertet.

a. Ändern Sie unsere small step Regeln so ab, dass Sie eine *call by name* Semantik erhalten. Achten Sie darauf, dass Ihre Semantik deterministisch ist, d.h. dass Satz 1 der Vorlesung auch für Ihre Semantik gilt.

b. Geben Sie einen Ausdruck an, der in den beiden Semantiken unterschiedliche 'Endergebnisse' liefert.

c. Welcher der beiden Parameterübergabe-Mechanismen wird in den Ihnen bekannten Programmiersprachen verwendet?

Aufgabe 3
Definieren Sie $\mathit{free}\,(e)$ und $e\,[e'/id]$ für den Fall, dass e ein **let**-Ausdruck ist. (Hinweis: Sie erhalten die 'richtige' Definition, indem Sie den **let**-Ausdruck als syntaktischen Zucker auffassen und in die Kernsyntax übersetzen.)

Aufgabe 4
Bestimmen Sie die small step Semantik für jeden der folgenden Ausdrücke. Machen Sie sich (vorher) intuitiv klar, wie die Bindungen in den Ausdrücken verlaufen, und überzeugen Sie sich davon, dass diese intuitive Auffassung jeweils mit der formalen Semantik übereinstimmt.

a. $\textbf{let}\ x = 0$
$\quad\textbf{in}\ (\lambda x.\,\textbf{if}\ x\ \textbf{then}\ 1\ \textbf{else}\ 2)\,(x \geq 0)$

b. $\textbf{let}\ x = 0$
$\quad\textbf{in let}\ x = (x = x)$
$\quad\quad\textbf{in}\ x$

c. $\textbf{let}\ f = \lambda x.\,\textbf{if}\ x = 0\ \textbf{then}\ 1\ \textbf{else}\ f\,(x-1)$
$\quad\textbf{in}\ f\,3$

d. $\textbf{let}\ f = \lambda x.\,0$
$\quad\textbf{in let}\ f = \lambda x.\,\textbf{if}\ x = 0\ \textbf{then}\ 1\ \textbf{else}\ f\,(x-1)$
$\quad\quad\textbf{in}\ f\,3$

e. $\textbf{let}\ x = 1$
$\quad\textbf{in let}\ f = \lambda y.\,y + x$
$\quad\quad\textbf{in let}\ x = 2$
$\quad\quad\quad\textbf{in}\ f\,x$

f. $\textbf{let}\ g = \lambda x.\,x + 1$
$\quad\textbf{in let}\ f = \lambda x.\,g\,x$
$\quad\quad\textbf{in let}\ g = \lambda x.\,x * x$
$\quad\quad\quad\textbf{in}\ f\,(g\,2)$