

Homework Assignment #5

CSE505

Due November 2

In this assignment, you are going to add *records* to the call-by-value (untyped) lambda calculus. A record is a mapping from a finite set of *labels* to values. For example,

$$\{l_1 = (\lambda x.x), l_2 = \text{true}, l_3 = (\lambda x.(x \text{ true}))\}$$

is a record mapping l_1 , l_2 and l_3 to values. Here l is a metavariable ranging over an infinite set of labels (distinct from the set of variable names represented by metavariable x). The record's components are accessed (or *projected*) by label, using “dot notation.” For example, the value of

$$\{l_1 = (\lambda x.x), l_2 = \text{true}, l_3 = (\lambda x.(x \text{ true}))\}.l_2$$

is true.

The new cases of the language's syntax are as follows:

$$\begin{aligned} e &::= \dots \\ &\quad \{l_1 = e_1, \dots, l_n = e_n\} \quad n \geq 1 \\ &\quad e.l \end{aligned}$$

Note the use of a new metavariable n for integers. The metavariable is used to express the fact that records may have any positive length (for simplicity, we disallow zero-length records). This notational device will be useful in the assignment. We also use the metavariable m for integers. In general, you can invent any new notation necessary for the assignment, as long as you define it before use.

1 Values

Show the new cases of the grammar for values.

2 Operational Semantics

- a. Show the new rules of the operational semantics. Record components should be evaluated in left-to-right order.

Tip: Think about how the operational semantics of *pairs* would look. A pair is denoted (e_1, e_2) , with component accessors named “fst” and “snd.” For example, $(\text{true}, \lambda x.x).\text{fst}$ evaluates to true. Records generalize pairs in two ways. First, components are accessed by label name rather than by distinguished accessors, fst and snd. Second, records may contain an unbounded (but finite) number of components, rather than only two.

- b. Show the new parts of the grammar for stuck expressions.