Zilly

Lakebolt Research

https://www.lakebolt.com/

1 Zilly

1.1 Abstract syntax

Types

$$t ::= Z \tag{1}$$
$$t ::= Fun(t_p,\ t_r) \tag{2}$$
$$t ::= Lazy(t_b) \tag{3}$$

Expressions

$$e ::=\ i \quad (\ where\ i\ =\ 0 \mid 1 \mid 2 \mid ...\ ) \tag{4}$$
$$e ::=\ x \quad (where\ x\ is\ a\ symbol\ ) \tag{5}$$
$$e ::=\ Lambda(x : t, e_x) \tag{6}$$
$$e ::=\ Apply(e_f,\ e_a) \tag{7}$$
$$e ::=\ If(e_c,\ e_t,\ e_f) \tag{8}$$
$$e ::=\ Defer(e_x) \tag{9}$$
$$e ::=\ Less(e_x, e_y) \tag{10}$$
$$e ::=\ Minus(e_x, e_y) \tag{11}$$
$$e ::=\ Random(e_x) \tag{12}$$
$$e ::=\ Formula(e_x) \tag{13}$$

Statements

$$a ::= Define(x : t, e); \tag{14}$$
$$a ::= Assign(x, e); \tag{15}$$

Zilly subsets: Though small, Zilly has three subsets worth mentioning:

* Eliminating (15) precludes an imperative programming style
* Eliminating (3) and (9) precludes lazy evaluation and reactive behavior
* Eliminating (3), (9), and (15) still leaves a Turing-complete language