Histola: Rethinking programs as series of interactions

TOMAS PETRICEK, University of Kent, United Kingdom

xx

1 INTRODUCTION

given an expression e is the greatest lie in programming language research!

2 IDEA

represent program as list of interactions

3 CONSEQUENCES: TYPE SYSTEM

3.1 Well constructed means well typed

3.2 Dependent types

4 CONSEQUENCES: USABILITY

4.1 Refactoring

4.2 Dual view

5 FORMALISM

$$\begin{array}{rcl} \iota & = & \mathsf{def}\ v\ \mathsf{as}\ r' \\ & | & \mathsf{dot}\ r, n\ \mathsf{as}\ r' \\ & | & \mathsf{abstract}\ (r_1, \ldots, r_k), r\ \mathsf{as}\ r' \\ & | & \mathsf{apply}\ r, (r_1, \ldots, r_k)\ \mathsf{as}\ r' \\ & | & \mathsf{eval}\ r \end{array}$$

$$p = \iota_1, \ldots, \iota_k$$

state

$$\begin{array}{rcl} v & = & o \mid (r_1, \ldots, r_n) \to r \\ e & = & v \mid r.n \mid r(r_1, \ldots, r_k) \end{array}$$

$$s = (E, V) \text{ where } E = \{r_1 \mapsto e_1, \ldots, r_k \mapsto e_k\}, V = \{r_1 \mapsto v_1, \ldots, r_l \mapsto v_l\}$$

evaluation assume we have $o.m \rightsquigarrow v$

$$\mathsf{apply}\ (\mathsf{def}\ v\ \mathsf{as}\ r')\ (E, V) = (E \cup \{r' \mapsto v\}, V) \tag{1}$$

$$\mathsf{apply}\ (\mathsf{dot}\ r, n\ \mathsf{as}\ r')\ (E, V) = (E \cup \{r' \mapsto r.n\}, V) \tag{2}$$

$$\mathsf{apply}\ (\mathsf{abstract}\ (r_1, \ldots, r_k), r\ \mathsf{as}\ r')\ (E, V) = (E \cup \{r' \mapsto (r_1, \ldots, r_k) \to r\}, V) \tag{3}$$

$$\mathsf{apply}\ (\mathsf{apply}\ r, (r_1, \ldots, r_k)\ \mathsf{as}\ r')\ (E, V) = (E \cup \{r' \mapsto r(r_1, \ldots, r_k)\}, V) \tag{4}$$

$$\mathsf{apply}\ (\mathsf{eval}\ r)\ (E, V) = (E, V \cup \{r \mapsto v\}) \text{ where } (r \mapsto v) \in E \tag{5}$$

$$\begin{array}{l} \mathsf{apply}\ (\mathsf{eval}\ r)\ (E, V) = (E, V \cup \{r \mapsto v\}) \\ \qquad \text{where } (r \mapsto r_0.n) \in E \\ \qquad\qquad (E', V') = \mathsf{apply}\ (\mathsf{eval}\ r_0)\ (E, V) \\ \qquad\qquad (r \mapsto o) \in V' \\ \qquad\qquad o.m \rightsquigarrow v \end{array} \tag{6}$$

$$\begin{array}{l} \mathsf{apply}\ (\mathsf{eval}\ r)\ (E, V) = (E, V \cup \{r \mapsto v\}) \\ \qquad \text{where } (r \mapsto r_0(r_1, \ldots, r_k)) \in E \\ \qquad\qquad (E', V') = \mathsf{apply}^*\ (\mathsf{eval}\ r_0, \ldots, \mathsf{eval}\ r_k)\ (E, V) \\ \qquad\qquad (r_0 \mapsto (r'_1, \ldots, r'_k) \to r') \in V' \\ \qquad\qquad (r_1 \mapsto v_1) \in V', \ldots, (r_k \mapsto v_k) \in V' \\ \qquad\qquad (E'', V'') = \mathsf{apply}^*\ (\mathsf{def}\ v_1\ \mathsf{as}\ r'_1, \ldots, \mathsf{def}\ v_k\ \mathsf{as}\ r'_k, \mathsf{eval}\ r')\ (E, V) \\ \qquad\qquad (r' \mapsto v) \in V'' \end{array} \tag{7}$$

getting completions judgements