

CuCh machine  
Linguaggi di Programmazione  
a.a. 2019-2020

Edoardo De Matteis  
1746561

Mirko Giacchini  
matricola

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Sintassi</b>	<b>1</b>
<b>3</b>	<b>Semantica operativa</b>	<b>2</b>
3.1	Dynamic eager . . . . .	2
3.2	Dynamic lazy . . . . .	2
3.3	Static eager . . . . .	3
3.4	Static lazy . . . . .	3
<b>4</b>	<b>Osservazioni</b>	<b>4</b>

## 1 Introduzione

Una **Curry-Church** machine implementa un interprete minimale di un linguaggio funzionale non tipato.

## 2 Sintassi

*syntax.sml*

$$\begin{aligned} FUN ::= & Const\ k \mid Var\ x \\ & \mid Sum(M, N) \mid Fn(x, M) \\ & \mid Let(x, M, N) \mid App(M, N) \end{aligned} \tag{1}$$

$$K ::= 0 \mid 1 \mid \dots \tag{2}$$

$$X ::= A \mid \dots \mid Z \mid a \mid \dots \mid z \mid \dots \tag{3}$$

$$ENV : VAR \rightarrow VAR \times FUN \times ENV \tag{4}$$

$$find : ENV \times FUN \rightarrow (FUN \times ENV) \cup EXC \quad (5)$$

In  $ENV$  nel codominio il prodotto cartesiano presenta  $ENV$  poichè necessario nelle valutazioni con scoping statico, nel mondo dinamico non è necessario e semplicemente lo si ignora.  $EXC$  è l'insieme delle eccezioni.

### 3 Semantica operativa

$$\mapsto \subseteq ENV \times FUN \times VAL \equiv ENV \vdash FUN \mapsto VAL \quad (6)$$

#### 3.1 Dynamic eager

$$\overline{E \vdash Const\ k \mapsto Const\ k}$$

$$\overline{E \vdash Var\ x \mapsto v} \quad E(x) = v$$

$$\frac{E \vdash M \mapsto v1 \quad E \vdash N \mapsto v2}{E \vdash Sum(M, N) \mapsto v} \quad (v=v1+v2)$$

$$\overline{E \vdash Fn(x, M) \mapsto (x, M)}$$

$$\frac{E \vdash M \mapsto (x, M') \quad E \vdash N \mapsto v \quad E(x, v) \vdash M' \mapsto v'}{E \vdash App(M, N) \mapsto v'}$$

$$\frac{E \vdash M \mapsto v \quad E(x, v) \vdash N \mapsto v'}{E \vdash Let(x, M, N) \mapsto v'}$$

#### 3.2 Dynamic lazy

$$\overline{E \vdash Const\ k \mapsto Const\ k}$$

$$\frac{E \vdash M \mapsto v}{E \vdash Var\ x \mapsto v} \quad E(x) = M$$

$$\frac{E \vdash M \mapsto v1 \quad E \vdash N \mapsto v2}{E \vdash Sum(M, N) \mapsto v} \quad (v=v1+v2)$$

$$\overline{E \vdash Fn(x, M) \mapsto (x, M)}$$

$$\frac{E \vdash M \mapsto (x, M') \quad E(x, N) \vdash M' \mapsto v}{E \vdash App(M, N) \mapsto v}$$

$$\frac{E(x, M) \vdash N \mapsto v}{E \vdash Let(x, M, N) \mapsto v}$$

### 3.3 Static eager

$$\overline{E \vdash Const\ k \mapsto Const\ k}$$

$$\frac{E \vdash M \mapsto v}{E \vdash Var\ x \mapsto v} \quad E(x) = M$$

$$\frac{E \vdash M \mapsto v1 \quad E \vdash N \mapsto v2}{E \vdash Sum(M, N) \mapsto v} \quad (v=v1+v2)$$

$$\overline{E \vdash Fn(x, M) \mapsto (x, M, E)}$$

$$\frac{E \vdash M \mapsto (x, M', E') \quad E \vdash N \mapsto v \quad E'(x, v) \vdash M' \mapsto v'}{E \vdash App(M, N) \mapsto v'}$$

$$\frac{E \vdash M \mapsto v \quad E(x, v) \vdash N \mapsto v'}{E \vdash Let(x, M, N) \mapsto v'}$$

### 3.4 Static lazy

$$\overline{E \vdash Const\ k \mapsto Const\ k}$$

$$\frac{E' \vdash M \mapsto v}{E \vdash Var\ x \mapsto v} \quad E(x) = (M, E')$$

$$\frac{E \vdash M \mapsto v1 \quad E \vdash N \mapsto v2}{E \vdash Sum(M, N) \mapsto v} \quad (v=v1+v2)$$

$$\overline{E \vdash Fn(x, M) \mapsto (x, M, E)}$$

$$\frac{E \vdash M \mapsto (x, M', E') \quad E'(x, N) \vdash M' \mapsto v}{E \vdash App(M, N) \mapsto v}$$

$$\frac{E(x, M, E) \vdash N \mapsto v}{E \vdash \text{Let}(x, M, N) \mapsto v}$$

## 4 Osservazioni

Un programma che mette in mostra le differenze tra una valutazione eager ed una lazy è il programma

$$\text{let } x = x \text{ in } x$$

che in una semantica static eager (come in SML) dà errore perchè  $x$  non è definita. In una semantica dynamic lazy va invece in loop

$$\frac{\begin{array}{c} \dots \\ \hline \langle (x, x) \rangle \vdash x \mapsto \\ \hline \langle (x, x) \rangle \vdash x \mapsto \end{array}}{\emptyset \vdash \text{Let}(x, x, x) \mapsto}$$

Inoltre in SML non è possibile eseguire il più piccolo transinfinito

$$\omega = (fn\ x \Rightarrow x\ x)(fn\ x \Rightarrow x\ x)$$

per via del sistema dei tipi,  $FUN$  non essendo tipato non presenta questo problema e si entra in loop come si può dimostrare:

$$\frac{\begin{array}{c} \dots \\ \hline \langle (x, Fn(x, x\ x))(x, x) \rangle \vdash x\ x \mapsto (x, x\ x) \\ \hline \langle (x, Fn(x, x\ x)) \rangle \vdash x \mapsto (x, x\ x) \end{array} \quad \frac{\dots}{\langle (x, Fn(x, x\ x))(x, x) \rangle \vdash x\ x \mapsto}}{\frac{\emptyset \vdash Fn(x, x\ x) \mapsto (x, x\ x) \quad \langle (x, Fn(x, x\ x)) \rangle \vdash x\ x \mapsto}{\emptyset \vdash App(Fn(x, x\ x), Fn(x, x\ x)) \mapsto}}$$

Figura 1: dynamic lazy

$$\frac{\emptyset \vdash Fn(x, x\ x) \mapsto (x, x\ x, \emptyset) \quad \emptyset \vdash Fn(x, x\ x) \mapsto (x, x\ x, \emptyset) \quad \begin{array}{c} \dots \\ \hline \langle (x, x\ x) \rangle \vdash x\ x \mapsto \\ \hline \langle (x, x\ x) \rangle \vdash x \mapsto \end{array}}{\emptyset \vdash Fn(x, x\ x)\ Fn(x, x\ x) \mapsto}$$

Figura 2: static eager