ELIOM

A CORE ML LANGUAGE FOR TIERLESS WEB PROGRAMMING

Prathyusha Butti
Jordan L. Siaha

Radanne G., Vouillon J., Balat V. (2016) Eliom: A Core ML Language for Tierless Web Programming.
In: Igarashi A. (eds) Programming Languages and Systems. APLAS 2016.
Lecture Notes in Computer Science, vol 10017. Springer, Cham

THE PROBLEM

- Programmers must learn many languages and tools, keep all the numerous software artifacts involved synchronized in a Web site and make them communicate properly.

MOTIVATION

- The goal of a modern client-server Web application framework should be to make it possible to build dynamic Web pages in a *composable* way.

- This is where so-called *tierless* languages come into play. Such languages unify the client and server part of the application in one language with seamless communication.

SUMMARY OF THE APPROACH

- An Eliom application is composed of a single program which is decomposed by the compiler into two parts.
- The first part runs on a Web server and can manage several connections and sessions at the same time, with the possibility of sharing data between sessions, and to keep state for each browser or tab currently running the application.
- The client program, compiled statically to JavaScript, is sent to each client by the server program together with the HTML page, in response to the initial HTTP request. It persists until the browser tab is closed or until the user follows an external link.

ELIOM FEATURES

- Composition
- Leveraging the type system
- Explicit communication
- A simple and efficient execution model

ELIOM CODING EXAMPLES

```
let%server s = ...
let%client c = ...
```

```
let%server x : int fragment = [%client 1 + 3 ]
```

```
let%server s : int = 1 + 2
let%client c : int = ~%s + 1
```

```
let%server x : int fragment = [%client 1 + 3 ]
let%client c : int = 3 + ~%x
```

ELIOM GRAMMAR

$$
\begin{array}{rll}
p ::= \texttt{let}_s\ x = e_s\ \texttt{in}\ p \mid \texttt{let}_c\ x = e_c\ \texttt{in}\ p \mid e_c & & \text{(Programs)} \\
e_s ::= c_s \mid x \mid \texttt{Y} \mid (e_s\ e_s) \mid \lambda x.e_s \mid \{\!\{\ e_c\ \}\!\} & & \text{(Expressions)} \\
e_c ::= c_c \mid x \mid \texttt{Y} \mid (e_c\ e_c) \mid \lambda x.e_c \mid f\%e_s & & \\
f ::= x \mid c_s & & \text{(Converter)} \\
c_s \in \mathit{Const}_s & c_c \in \mathit{Const}_c & \text{(Constants)}
\end{array}
$$

Fig. 1. ELIOM$_\varepsilon$'s grammar

TYPE SYSTEM

$$\sigma_\varsigma ::= \forall \alpha^*.\tau_\varsigma \qquad \text{(TypeSchemes)}$$

$$\tau_s ::= \alpha \mid \tau_s \to \tau_s \mid \{\tau_c\} \mid \tau_s \rightsquigarrow \tau_c \mid \kappa \text{ for } \kappa \in ConstType_s$$

$$\tau_c ::= \alpha \mid \tau_c \to \tau_c \mid \kappa \text{ for } \kappa \in ConstType_c \qquad \text{(Types)}$$

TYPING RULES

Common rules

$$\textsc{Var}\ \frac{(x:\sigma)_\varsigma \in \Gamma \qquad \sigma \succ \tau}{\Gamma \rhd_\varsigma x : \tau} \qquad \textsc{Lam}\ \frac{\Gamma, (x:\tau_1)_\varsigma \rhd_\varsigma e : \tau_2}{\Gamma \rhd_\varsigma \lambda x.e : \tau_1 \to \tau_2} \qquad \textsc{Const}\ \frac{\mathit{TypeOf}_\varsigma(c) \succ \tau}{\Gamma \rhd_\varsigma c : \tau}$$

$$\textsc{App}\ \frac{\Gamma \rhd_\varsigma e_1 : \tau_1 \to \tau_2 \qquad \Gamma \rhd_\varsigma e_2 : \tau_1}{\Gamma \rhd_\varsigma (e_1\ e_2) : \tau_2} \qquad \textsc{Let}\ \frac{\Gamma \rhd_\varsigma e_1 : \tau_1 \qquad \Gamma, (x : \mathit{Close}(\tau_1, \Gamma))_\varsigma \rhd_\varsigma e_2 : \tau_2}{\Gamma \rhd_\varsigma \texttt{let}\ x = e_1\ \texttt{in}\ e_2 : \tau_2}$$

$$\textsc{Y}\ \frac{}{\Gamma \rhd_\varsigma \texttt{Y} : ((\tau_1 \to \tau_2) \to \tau_1 \to \tau_2) \to \tau_1 \to \tau_2}$$

Server rules

$$\textsc{Fragment}\ \frac{\Gamma \rhd_c e_c : \tau_c}{\Gamma \rhd_s \{\!\{\ e_c\ \}\!\} : \{\tau_c\}}$$

Client rules

$$\textsc{Injection}\ \frac{\Gamma \rhd_s f : \tau_s \rightsquigarrow \tau_c \qquad \Gamma \rhd_s e_s : \tau_s}{\Gamma \rhd_c f\%e_s : \tau_c}$$

Eliom$_\varepsilon$'s rules

$$\textsc{Prog}\ \frac{\Gamma \rhd_\varsigma e : \tau_1 \qquad \Gamma, (x : \mathit{Close}(\tau_1, \Gamma))_\varsigma \blacktriangleright p : \tau_2}{\Gamma \blacktriangleright \texttt{let}_\varsigma\ x = e\ \texttt{in}\ p : \tau_2} \qquad \textsc{Return}\ \frac{\Gamma \rhd_c e_c : \tau_c}{\Gamma \blacktriangleright e_c : \tau_c}$$

$$\mathit{Close}(\tau, \Gamma) = \forall \alpha_0 \ldots \alpha_n.\tau \text{ where } \{\alpha_0, \ldots, \alpha_n\} = \mathit{FreeTypeVar}(\tau) \backslash \mathit{FreeTypeVar}(\Gamma)$$

Fig. 2. Typing rules for Eliom$_\varepsilon$

THANK YOU