CLE Type System

Benjamin Flin

August 2021

1 Introduction

2 CLE Types Grammar

The following is a small grammar for CLE types τ. l and r represent levels and can be thought of as arbitrary identifiers. $\alpha, \beta, \theta, \sigma, \phi, \pi, \gamma$ are sets of remote levels.

$$\tau ::= l\ \sigma \mid l\ (\alpha_1, \ldots, \alpha_n) \to \theta$$

$l\ \sigma$ describes a global variable in level l which can be shared with levels inside σ. $l\ (\alpha_1, \ldots, \alpha_n) \to \theta$ describes a function in level l with arguments shareable with $\alpha_1, \ldots \alpha_n$ where the return value can be shared with θ. The return value θ determines from which level where the function can be called from.

For any given program, there is a set, $\mathcal{U}$, which describes the set of levels under consideration. That set is called the universe set.

The terms of the inhabitants of these types are an idealized subset of llvm, where the only control flow changes are from breaks and function calls.

3 Type rules

Here we assume all functions and global variables have cle types associated with them. We will focus on how to infer such types in the next section. There are several types of judgements, each of which is enumerated below:

3.1 Judgements

1. $\Gamma \vdash e : \tau$. Function, instruction or global variable e has type τ.
2. $\Gamma \vdash e \Leftarrow \phi : l\ \sigma$. Basic block or terminator e has type $l\ \sigma$ with constraining set ϕ.
3. $\Gamma \vdash i \Leftarrow l\ \phi$. Instruction list is constrained by level set ϕ with level l.

3.2 Rules for top-level entities

$$\frac{\tau = l\ \sigma}{\Gamma \vdash @x : \tau}\ \text{global-decl} \qquad \frac{c : l\ \sigma' \qquad \sigma' \supseteq \sigma}{\Gamma \vdash @x = c : l\ \sigma}\ \text{global-def}$$

$$\frac{\Gamma[\%1 \mapsto l\ \alpha_1, \ldots, \%n \mapsto l\ \alpha_n] \vdash bb_0 \Leftarrow \mathcal{U} : l\ \theta}{\Gamma \vdash @f(\%1, \ldots, \%n)\ \{bb_0\} : l\ (\alpha_1, \ldots, \alpha_n) \to \theta}\ \text{fn-def}$$

3.3 Rules for basic blocks and instruction lists

$$\frac{\Gamma \vdash instrs \Leftarrow l\ \phi \qquad \Gamma \vdash term \Leftarrow \phi : l\ \sigma}{\Gamma \vdash \%b = instrs\ term \Leftarrow \phi : l\ \sigma}\ \text{bb}$$

$$\frac{\Gamma \vdash instr : l\ \sigma \qquad \sigma \subseteq \phi \qquad \Gamma[\%a \mapsto l\ \sigma] \vdash instrs \Leftarrow l\ \phi}{\Gamma \vdash \%a = instr; instrs \Leftarrow l\ \phi}\ \text{instrs}$$

3.4 Rules for special instructions and terminators

$$\frac{\begin{array}{c}\Gamma(\%1) = l\ \alpha_1, \ldots, \Gamma(\%n) = l\ \alpha_n \\ \Gamma \vdash f : r\ (\beta_1, \ldots, \beta_n) \to \theta \\ \alpha_1 \supseteq \beta_1, \ldots, \alpha_n \supseteq \beta_n \\ r \in \alpha_1, \ldots, r \in \alpha_n \\ l \in \theta \\ \theta \supseteq \sigma\end{array}}{\Gamma \vdash \text{call } @f(\%1, \ldots, \%n) : l\ \sigma}\ \text{call}$$

$$\frac{\Gamma(\%v) = l\ \gamma' \qquad \Gamma(\%a) = l\ \gamma \qquad \gamma' \supseteq \gamma}{\Gamma \vdash \text{store } \%v, \%a : l\ \sigma}\ \text{store}$$

$$\frac{\Gamma(\%a) = l\ \pi \qquad \Gamma \vdash \%b_1 \Leftarrow \pi \cap \phi : l\ \gamma \qquad \Gamma \vdash \%b_2 \Leftarrow \pi \cap \phi : l\ \gamma' \qquad \gamma \cap \gamma' \supseteq \sigma}{\Gamma \vdash \text{br } \%a, \%b_1, \%b_2 \Leftarrow \phi : l\ \sigma}\ \text{break}$$

$$\frac{\Gamma(\%a) = l\ \pi \qquad \pi \supseteq \sigma}{\Gamma \vdash \text{ret } \%a \Leftarrow \phi : l\ \sigma}\ \text{ret}$$

3.5 General rule for instructions

The rules for all other instructions can be derived from general instruction form which takes in a number of arguments:

$$\text{instr } \%a_1, \ldots, \%a_n$$

Thus,

$$\frac{\Gamma(\%a_1) = l\ \pi_1 \cdots = \Gamma(\%a_n) = l\ \pi_n \qquad \pi_1 \supseteq \sigma, \ldots, \pi_n \supseteq \sigma}{\Gamma \vdash \text{instr } \%a_1, \ldots \%a_n : l\ \sigma}\ \text{instr}$$