CS738: Advanced Compiler Optimizations

Simply Typed Lambda Calculus

Amey Karkare

karkare@cse.iitk.ac.in

http://www.cse.iitk.ac.in/~karkare/cs738
Department of CSE, IIT Kanpur

Reference Book

Types and Programming Languages by Benjamin C. Pierce

Simple Types over Bool

$$
\begin{array}{lll}
T & := & \text{– Types} \\
& \text{Bool} & \text{– Boolean Type} \\
& T \to T & \text{– Function Type}
\end{array}
$$

type constructor $\to$ is right-associative, i.e., $T_1 \to T_2 \to T_3$ stands for $T_1 \to (T_2 \to T_3)$

Examples

For each of the type below, write a function (in your favourite programming language) that has the required type:

- ▶ Bool $\to$ Bool
- ▶ Bool $\to$ Bool $\to$ Bool
- ▶ (Bool $\to$ Bool) $\to$ Bool
- ▶ (Bool $\to$ Bool) $\to$ Bool $\to$ Bool
- ▶ (Bool $\to$ Bool $\to$ Bool) $\to$ Bool
- ▶ (Bool $\to$ Bool $\to$ Bool) $\to$ Bool
- ▶ ((Bool $\to$ Bool) $\to$ Bool) $\to$ Bool

The Abstract Syntax

Simply Typed λ-terms with conditions and Booleans

t	:=	x		– *Variable*
	|	$\lambda x : T. t$		– *Abstraction*
	|	t t		– *Application*
	|	true		– *constant true*
	|	false		– *constant false*
	|	if t then t else t		– *conditional*

Recap: The Set of Values

v	:=		– *values*
		$\lambda x : T. t$	– *Abstraction Value*
	|	true	– *value true*
	|	false	– *value false*

Evaluation

$$\frac{t_1 \rightarrow t_1'}{t_1\ t_2 \rightarrow t_1'\ t_2} \qquad (\text{E-APP1})$$

$$\frac{t_2 \rightarrow t_2'}{v\ t_2 \rightarrow v\ t_2'} \qquad (\text{E-APP2})$$

$$(\lambda x : T_1. t_1)v_2 \rightarrow [x \mapsto v_2]t_1 \qquad (\text{E-APPABS})$$

The Typing Relation

► A *Typing Context* or *Type Environment*, Γ, is a sequence of variables with their types

► $\Gamma, x : T$ denotes extending Γ with a new variable x having type T
  ► The name x is assumed to be distinct from any existing names in Γ

The Typing Relation

$$\frac{\Gamma, x : T_1 \vdash t_2 : T_2}{\Gamma \vdash \lambda x : T_1.\, t_2 : T_1 \to T_2} \qquad \text{(T-ABS)}$$

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T} \qquad \text{(T-VAR)}$$

$$\frac{\Gamma \vdash t_1 : T_1 \to T_2 \qquad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1\, t_2 : T_2} \qquad \text{(T-APP)}$$

Inversion of the Typing Relation

- If $\Gamma \vdash x : R$, then $x : R \in \Gamma$.
- If $\Gamma \vdash \lambda x : T_1.\, t_2 : R$, then $R = T_1 \to R_2$ for some R_2 with $\Gamma, x : T_1 \vdash t_2 : R_2$.
- If $\Gamma \vdash t_1\, t_2 : R$, then $\exists T_1$ *s.t.* $\Gamma \vdash t_1 : T_1 \to R$ and $\Gamma \vdash t_2 : T_1$.
- If $\Gamma \vdash \texttt{true} : R$, then $R = $ Bool.
- If $\Gamma \vdash \texttt{false} : R$, then $R = $ Bool.
- If $\Gamma \vdash \texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 : R$, then
  - $\Gamma \vdash t_1 : $ Bool
  - $\Gamma \vdash t_2 : R$
  - $\Gamma \vdash t_3 : R$

Exercises

- For each of the term t below, find context Γ and type T such that
$$\Gamma \vdash t : T$$

  - t is $\lambda x.\, x$
  - t is $(\,(x\, z)\,(y\, z)\,)$
  - t is $\lambda y.\, x$
  - t is $x\, x$

Uniqueness of Types

- In a given type context Γ, A term t, such that the free variables of t are in Γ, has at most one type.
- If t is typeable, then its type is unique.
- Moreover, there is just one derivation of this typing built from the inference rules.

Some Properties

- ▶ **Permutation:** If $\Gamma \vdash t : T$ and Δ is a permutation of Γ, then $\Delta \vdash t : T$.
  - ▶ The derivation with Δ has the same depth as the derivation with Γ.
- ▶ **Weakening:** If $\Gamma \vdash t : T$ and $x \notin$ domain(Γ), then $\Gamma, x : S \vdash t : T$.
  - ▶ The derivation with $\Gamma, x : S$ has the same depth as the derivation with Γ.

Progress

- ▶ **Progress:** A well-typed term is not stuck.
  - ▶ If $\vdash t : T$, then t is either a value or there exists some t' such that $t \to t'$.

Preservation

- ▶ **Preservation of Types under Substitution:** If $\Gamma, x : S \vdash t : T$ and $\Gamma \vdash s : S$, then $\Gamma \vdash [x \mapsto s]t : T$.
- ▶ **Preservation:** If a well-typed term takes a step of evaluation, then the resulting term is also well-typed.
  - ▶ If $\Gamma \vdash t : T$ and $t \to t'$, then $\Gamma \vdash t' : T$.