SECD Machine Implementation of $\lambda_{\rightarrow}$

TAZMILUR SAAD, Independent, USA

This work documents an autodidactic implementation of an interpreter for a simply typed lambda calculus extended unit, sums and product types. The interpreter is based on the SECD machine.

1 CALCULUS

A typing context is a set of variables equipped with a type.

$$\Gamma ::= \varnothing \mid \Gamma, x : \tau$$

The types in our system includes the unit type, sum types, product types, function types and the base types.

$$\tau ::= \mathbb{1} \mid \tau + \tau \mid \tau \times \tau \mid \tau \rightarrow \tau \mid \texttt{Nat} \mid \texttt{Bool}$$

The values in our language include lambda abstractions, unit, sums, products, natural numbers and booleans.

$$v ::= \lambda x : \tau . e \mid () \mid (v, v) \mid \text{inl } v \mid \text{inr } v \mid n \mid b$$

$$n ::= 0 \mid 1 \mid 2 \mid \ldots$$

$$b ::= true \mid false$$

Expressions includes variables, values, sums, products, applications, let bindings, cases and projections

$$e ::= x \mid v \mid \text{inl } e \mid \text{inr } e \mid (e, e) \mid (e, e).1 \mid (e, e).2 \mid ee$$

$$\mid \text{let } x = e \text{ in } e \mid \text{case } e \text{ of } \{\text{inl } x \rightarrow e,\ \text{inr } x \rightarrow e\}$$

Expressions are evaluated using the following evaluation contexts.

$$E ::= [\cdot] \mid Ee \mid vE \mid \text{inl } E \mid \text{inr } E \mid (E, e) \mid (v, E) \mid E.1 \mid E.2$$

$$\mid \text{case } E \text{ of } \{\text{inl } x \rightarrow e,\ \text{inr } x \rightarrow e\} \mid \text{let } x = E \text{ in } e$$

The typing judgements and the operational semantics are shown in Figure 1 and 3 respectively.

2 SOUNDNESS

2.1 Preservation

Lemma 2.1 (Permutation). *If $\Gamma \vdash e : \tau$ and Δ is a permutation of Γ, then $\Delta \vdash e : \tau$. Moreover, the latter derivation has the same depth as the former.*

Proof. A permutation is a bijection $\Delta : \Gamma \rightarrow \Gamma$. We will use Δ as a bijection and Δ as a typing context interchangeably. We proceed by structural induction on the typing derivations.

[Unit, Nat, Bool]: Since Δ is a bijection, there exists some $e' : \tau'$ in Δ such that $\Delta(e' : \tau') = e : \tau$. Therefore, $\Delta \vdash e : \tau$ and the depth does not change.

[Var]: If $e : \tau \in \Gamma$, then $e : \tau \in \Delta$ since Δ is a bijection. We can apply the judgement and the depth does not change.

[Inl, Inr, Pair, Proj 1, Proj 2, App]: We assume the inductive hypothesis, substitute $\Delta \vdash e : \tau$ for $\Gamma \vdash e : \tau$ and apply the judgement. The depth does not change.

Author's address: Tazmilur Saad, Independent, Jersey City, NJ, USA, ssaad@colgate.edu.

$$\frac{}{\Gamma \vdash () : \mathbb{1}}\ \textsc{Unit} \qquad \frac{}{\Gamma \vdash n : \texttt{Nat}}\ \textsc{Nat} \qquad \frac{}{\Gamma \vdash b : \texttt{Bool}}\ \textsc{Bool}$$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}\ \textsc{Var} \qquad \frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda x : \tau_1 . e : \tau_1 \to \tau_2}\ \textsc{Lam} \qquad \frac{\Gamma \vdash e_1 : \tau_1 \to \tau_2 \qquad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1 e_2 : \tau_2}\ \textsc{App}$$

$$\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \text{ inl } e : \tau_1 + \tau_2}\ \textsc{Inl} \qquad \frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \text{ inr } e : \tau_1 + \tau_2}\ \textsc{Inr}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma, x : \tau_1 \vdash e_2 : \tau_2}{\Gamma \vdash \text{ let } x = e_1 \text{ in } e_2 : \tau_2}\ \textsc{Let} \qquad \frac{\Gamma \vdash e : \tau_1 + \tau_2 \quad \Gamma, x_1 : \tau_1 \vdash e_1 : \tau \quad \Gamma, x_2 : \tau_2 \vdash e_2 : \tau}{\Gamma \vdash \text{case } e \text{ of } \{\text{inl } x_1 \to e_1,\ \text{inr } x_2 \to e_2\} : \tau}\ \textsc{Case}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (e_1, e_2) : \tau_1 \times \tau_2}\ \textsc{Pair} \qquad \frac{\Gamma \vdash (e_1, e_2) : \tau_1 \times \tau_2}{\Gamma \vdash e_1 : \tau_1}\ \textsc{Proj 1} \qquad \frac{\Gamma \vdash (e_1, e_2) : \tau_1 \times \tau_2}{\Gamma \vdash e_2 : \tau_2}\ \textsc{Proj 2}$$

Fig. 1. Simply typed lambda calculus

$$\frac{e \to e'}{E[e] \to E[e']}\ \textsc{context}$$

$$\begin{aligned}
(\lambda x : \tau . e) v &\to e[v/x] && \beta\textsc{-reduction}\\
\text{let } x = v \text{ in } e &\to e[v/x] && \textsc{Let}\\
(v_1, v_2).1 &\to v_1 && \textsc{Proj 1}\\
(v_1, v_2).2 &\to v_2 && \textsc{Proj 2}\\
\text{case inl } v \text{ of } \{e_1,\ e_2\} &\to e_1 && \textsc{Inl}\\
\text{case inr } v \text{ of } \{e_1,\ e_2\} &\to e_2 && \textsc{Inr}
\end{aligned}$$

Fig. 2. Operational Semantics

$$b[v/x] = b \qquad n[v/x] = n \qquad ()[v/x] = () \qquad (ee)[v/x] = e[v/x]e[v/x]$$

$$x_1[v/x_1] = v \qquad x_2[v/x_1] = x_2 \qquad (\text{inl } e)[v/x] = \text{inl } e[v/x] \qquad (\text{inl } r)[v/x] = \text{inr } e[v/x]$$

$$(e, e)[v/x] = (e[v/x], e[v/x])$$

Fig. 3. Substitution Rules

[Let, Case, Lam]: We assume the inductive hypothesis and substitute $\Delta \vdash e : \tau$ for $\Gamma \vdash e : \tau$. Each of these derivations extend Γ with some term $x : \tau_x$, but we can also extend Δ by adding a mapping from this term to itself and apply the judgement. The depth does not change. □

Lemma 2.2 (Weakening). *If* $\Gamma \vdash e : \tau_1$ *and* $x \notin dom(\Gamma)$*, then* $\Gamma, x : \tau_2 \vdash e : \tau_1$*. Moreover, the latter derivation has the same depth as the former.*

Proof. We proceed by structural induction on the typing derivation.

[Unit, Nat, Bool]: These judgements do not assume anything about the context. If we extend the context with an element from outside the domain, we can reach the same conclusions. Moreover, there are no subterms so the derivation tree's depth does not change.

[Var]: If we extend Γ with an element from outside the domain, $x : \tau \in \Gamma$ still holds and we can apply the judgement.

[Inl, Inr, Pair, Proj 1, Proj 2, App]: We assume the inductive hypothesis. These rules do not extend Γ, so addition of another element from outside the domain allows us to reach the same conclusion.

[Let, Case, Lam]: We assume the inductive hypothesis. If we extend the Γ with element from outside the domain, the subsequent extensions as part of the antecedent will still be valid modulo renaming. We can still apply the judgement. □

Lemma 2.3 (Substitution). *If* $\Gamma, x : \tau_1 \vdash e : \tau_2$ *and* $\Gamma \vdash v : \tau_1$ *then* $e[v/x] : \tau_2$.

Proof. We proceed by structural induction on the typing derivation of $\Gamma, x : \tau_1 \vdash e : \tau_2$.

[Unit, Nat, Bool]: By definition of substitution.

[Var]:

$$\frac{z : \tau_1 \in \Gamma, x : \tau}{\Gamma, x : \tau \vdash z : \tau_1} \quad \Gamma \vdash v : \tau \qquad \text{(Assumptions)}$$

$$x = z \implies z : \tau \in \Gamma \implies \Gamma \vdash z[v/x] : \tau = \tau_1 \qquad \text{(Substitution)}$$

$$x \neq z \implies z : \tau_1 \in \Gamma \implies \Gamma \vdash z[v/x] : \tau_1 \qquad \text{(Substitution)}$$

[Lam]:

$$\begin{aligned}
&\frac{\Gamma, x : \tau, z : \tau_1 \vdash e : \tau_2}{\Gamma, x : \tau \vdash \lambda z : \tau_1.e : \tau_1 \rightarrow \tau_2} \quad \Gamma \vdash v : \tau && \text{(Assumption)}\\
&\Gamma, x : \tau, z : \tau_1 \vdash e : \tau_2 && \text{(Subterm)}\\
\implies &(\Gamma, z : \tau_1), x : \tau \vdash e : \tau_2 && \text{(Permutation)}\\
\implies &\Gamma \vdash e[v/z] : \tau_2 && \text{(Inductive Hypothesis)}\\
&\Gamma \vdash v : \tau && \text{(Assumption)}\\
\implies &\Gamma, z : \tau_1 \vdash v : \tau && \text{(Weakening)}\\
\implies &\Gamma \vdash \lambda z : \tau_1.v : \tau_1 \rightarrow \tau && \text{(Lam)}
\end{aligned}$$

[App]:

$$\begin{aligned}
&\frac{\Gamma, x : \tau \vdash e_1 : \tau_1 \rightarrow \tau_2 \qquad \Gamma, x : \tau \vdash e_2 : \tau_1}{\Gamma, x : \tau \vdash e_1 e_2 : \tau_2}\ \textsc{App} \quad \Gamma \vdash v : \tau && \text{(Assumptions)}\\
&\Gamma, x : \tau \vdash e_1 : \tau_1 \rightarrow \tau_2 \qquad \Gamma, x : \tau \vdash e_2 : \tau_1 && \text{(Subterms)}\\
\implies &\Gamma \vdash e_1[v/x] : \tau_1 \rightarrow \tau_2 \qquad \Gamma \vdash e_2[v/x] : \tau_1 && \text{(Induction Hypothesis)}\\
\implies &\Gamma \vdash e_1[v/x] e_2[v/x] : \tau_2 && \text{(App)}
\end{aligned}$$

[Inl, Inr]: Similar derivation applies for Inr.

$$\frac{\Gamma, x : \tau \vdash e : \tau_1}{\Gamma, x : \tau \vdash \text{ inl } e : \tau_1 + \tau_2} \quad \Gamma \vdash v : \tau \qquad \text{(Assumptions)}$$

$$\Gamma, x : \tau \vdash e : \tau_1 \qquad \text{(Subterm)}$$

$$\implies \Gamma \vdash e[v/x] : \tau_1 \qquad \text{(Inductive Hypothesis)}$$

$$\implies \Gamma \vdash \text{inl } e[v/x] : \tau_1 + \tau_2 \qquad \text{(Inl)}$$

[Let]:

$$\frac{\Gamma, x : \tau \vdash e_1 : \tau_1 \qquad \Gamma, x : \tau, z : \tau_1 \vdash e_2 : \tau_2}{\Gamma, x : \tau \vdash \text{ let } z = e_1 \text{ in } e_2 : \tau_2} \quad \Gamma \vdash v : \tau \qquad \text{(Assumptions)}$$

$$\Gamma, x : \tau \vdash e_1 : \tau_1 \implies \Gamma \vdash e_1[v/x] : \tau_1 \qquad \text{(Subterm 1, Inductive Hypothesis)}$$

$$\Gamma, x : \tau, z : \tau_1 \vdash e_2 : \tau_2 \qquad \text{(Subterm 2)}$$

$$\implies (\Gamma, z : \tau_1), x : \tau \vdash e_2 : \tau_2 \qquad \text{(Permutation)}$$

$$\implies \Gamma, z : \tau_1 \vdash e_2[v/x] : \tau_2 \qquad \text{(Inductive Hypothesis)}$$

$$\implies \Gamma \vdash \text{ let } z = e_1[v/x] \text{ in } e_2[v/x] : \tau_2 \qquad \text{(Let)}$$

[Pair]:

$$\frac{\Gamma, x : \tau \vdash e_1 : \tau_1 \qquad \Gamma, x : \tau \vdash e_2 : \tau_2}{\Gamma, x : \tau \vdash (e_1, e_2) : \tau_1 \times \tau_2} \quad \Gamma \vdash v : \tau \qquad \text{(Assumptions)}$$

$$\Gamma, x : \tau \vdash e_1 : \tau_1 \qquad \Gamma, x : \tau \vdash e_2 : \tau_2 \qquad \text{(Subterms)}$$

$$\implies \Gamma \vdash e_1[v/x] : \tau_1 \qquad \Gamma \vdash e_2[v/x] : \tau_2 \qquad \text{(Inductive Hypothesis)}$$

$$\implies \Gamma \vdash (e_1[v/x], e_2[v/x]) : \tau_1 \times \tau_2 \qquad \text{(Pair)}$$

[Proj 1, Proj 2]: A similar derivation applies for Proj 2.

$$\frac{\Gamma, x : \tau \vdash (e_1, e_2) : \tau_1 \times \tau_2}{\Gamma, x : \tau \vdash e_1 : \tau_1} \quad \Gamma \vdash v : \tau \qquad \text{(Assumptions)}$$

$$\Gamma, x : \tau \vdash (e_1, e_2) : \tau_1 \times \tau_2 \qquad \text{(Subterm)}$$

$$\implies \Gamma \vdash (e_1, e_2)[v/x] : \tau_1 \times \tau_2 \qquad \text{(Inductive Hypothesis)}$$

$$\implies \Gamma \vdash (e_1[v/x], e_2[v/x]) : \tau_1 \times \tau_2 \qquad \text{(Substitution)}$$

$$\implies \Gamma \vdash e_1[v/x] : \tau_1 \qquad \text{(Proj 1)}$$

□

Theorem 2.4 (Preservation). *If*

Proof. Admitted. □

2.2 Progress

Lemma 2.5 (Canonical Forms). *If*

Proof. Admitted. □

Theorem 2.6 (Progress). *If* $\cdot \vdash x : \tau$ *is a well typed term then* x *is a value or there exists some* y *such that* $x \mapsto y$.

Proof. Admitted. □

3 ACKNOWLEDGEMENTS

Thanks to Anton Lorenzen for his guidance on the project.