Scribe: Patrick N and Muthuramakrishnan V Lecturer: Andrew Myers

1 Strong normalization and logical relations

We want to prove all terms terminate. In other words we want to show that every expression has a normal form. It agrees with the denotational semantics at base types and this implies strong normalization. But both these facts will require a new proof technique, logical relations.

We prove by induction on typing derivation. We want the following as our induction hypothesis

$$\vdash e : \tau \implies \exists v.e \longrightarrow^* v \tag{1}$$

We can show that this easily holds for the base types.

$$\vdash e : B \wedge e \longrightarrow^* v \iff \mathcal{C}[\![\vdash e : B]\!]\emptyset = v$$

Consider at this typing inference rule

$$\frac{\vdash e_0 : \tau_1 \to \tau_2 \vdash e : \tau_1}{\vdash e_0\ e_1 \tau_2}$$

Just because e_0 terminates, it does not imply that e_0, it does not imply that e_0 when e_1 is substituted will terminate. Hence our induction hypothesis is not strong enough.

Idea: Define a family of relations $\mathcal{R}_\tau$ indexed on type. The *logical relation* is defined by induction on type structure. $\mathcal{R}_\tau(e)$ is a unary relation with $e \in \mathcal{R}_\tau$. So our induction hypothesis would be now.

$$\mathcal{R}_\tau(e) \implies \ \vdash e.\tau\ \wedge\ \exists\ v.e\ \longrightarrow^*\ v$$

Notice that we define the logical relation in such a way that it implies the fact that we are trying to prove. We formally define the logical relation as:

$$\begin{array}{lcl} \mathcal{R}_B(e) & \equiv & \vdash e : B \wedge \exists v.e \longrightarrow^* v \\ \mathcal{R}_{\tau_1 \to \tau_2}(e) & \equiv & e : \tau_1 \to \tau_2 \wedge \exists v.e \to^* v \wedge \forall e'.\mathcal{R}_{\tau_1}(e') \implies \mathcal{R}_{\tau_2}(e\ e') \end{array}$$

Lemma 1 $\mathcal{R}_\tau(e) \implies \ \vdash e.\tau\ \wedge\ \exists\ v.e\ \longrightarrow^*\ v$

Proof: We need an additional lemma for this.

Lemma 2 $\vdash e : \tau\ \wedge\ e \to e'\ \wedge\ \mathcal{R}_\tau(e') \iff \mathcal{R}_\tau(e)$

Proof: We prove by induction on τ.

- $\tau = B$. $\mathcal{R}_\tau(e') \implies e' \longrightarrow^* v$. Hence $e \longrightarrow e' \longrightarrow^* v$
- $\tau = \tau_1 \to \tau_2$. Assume an arbitrary e'' where $\mathcal{R}_{\tau_1}(e'')$.

$$\begin{array}{lcl} e\ e'' \to e'\ e'' & \implies & e'\ e'' \longrightarrow^* v \\ & \implies & \forall e''.\mathcal{R}_{\tau_1}(e'') \\ & \implies & \mathcal{R}_{\tau_2}(e'\ e'') \end{array}$$

Now we proceed on to the strong normalization hypothesis that every typed-lambda term has normal form. This we prove by induction on typing derivations.

$$\frac{}{\Gamma\ \vdash\ \lambda x\!:\!\tau_1.\, e' : \tau_1 \to \tau_2}$$

Consider $\Gamma \vdash e : \tau \implies \mathcal{R}_\tau(e)$, if free terms are in e then it will not reduce to a value. For this we introduce a substitution operator γ.

$$\gamma = \{x_1 \mapsto v_1, x_2 \mapsto v_2, \ldots, x_n \mapsto v_n\}$$

We lift this definition to expression in the following manner: $\gamma(e)$ means e with $x_1, x_2, \ldots, x_n$ substituted by γ, i.e. $\gamma(e) = e\{v_1/x_1, \ldots, v_n/x_n\}$.

We say a substitution satisfies Γ as:

$$\begin{aligned}\gamma \models \Gamma \iff & dom(\gamma) = dom(\Gamma) \\ & \wedge\ \forall x \in dom(\gamma).\gamma(x) \in \textbf{Value} \wedge \mathcal{R}_{\Gamma(x)}(\gamma(x))\end{aligned}$$

We can say $\gamma(x) \in$ **Value** because we are having call by value semantics. If it were Call by Name semantics we have to show for Subst $\gamma(e) = \{x_1 \mapsto e_1, x_2 \mapsto e_2, \ldots, x_n \mapsto e_n\}$.

Let us recall the substitution lemma

$$\Gamma \vdash e : \tau \wedge \gamma \models \Gamma \implies \gamma(e) : \tau$$

Our induction hypothesis now turns out to be

$$\Gamma \vdash e : \tau \wedge \gamma \models \Gamma \implies \mathcal{R}_\tau(\gamma(e))$$

Strong normalization: We specialize to $\Gamma = \emptyset$, $\gamma = \emptyset$. So if we prove our induction hypothesis we are done by setting $\Gamma = \emptyset$ and $\gamma = \emptyset$.

We now show that $\Gamma \vdash e : \tau \wedge \gamma \models \Gamma \implies R_\tau(\gamma(e))$ using the substitution lemma. Recall the syntax of $\lambda^{\rightarrow}$.

$$e \quad ::= \quad b \ \mid \ x \ \mid \ e_0 e_1 \ \mid \ \lambda x\!:\!\tau.\, e$$

So we have the following cases:

- Case $e = b$: Since b is a base value, $\vdash e : B \wedge b \longrightarrow^* v$. Thus, by the definition of logical relations, $R_B(\gamma(b))$.

- Case $e = x$: We need to show that $\Gamma \vdash x : \tau \wedge \gamma \models \Gamma \implies R_\tau(\gamma(x))$. Since x is a variable and $\Gamma \vdash x : \tau$, so $\tau = \Gamma(x)$ and $\vdash e : \Gamma(x)$. Moreover, since the evaluation rules for $\lambda^{\rightarrow}$ is CBV, $\gamma(x)$ is a value. Therefore, $R_\tau(\gamma(x))$.

- Case $e = e_0\ e_1$: We need to show that $\Gamma \vdash e_0\ e_1 : \tau \wedge \gamma \models \Gamma \implies R_\tau(\gamma(e_0\ e_1))$. By typing derivation, we have:

  $$\frac{\Gamma \vdash e_0 : \tau_1 \rightarrow \tau \quad \Gamma \vdash e_1 : \tau_1}{\Gamma \vdash e_0\ e_1 : \tau}$$

  Thus, by the induction hypothesis on the two typing judgments, $R_{\tau_1 \longrightarrow \tau}(\gamma(e_0))$ and $R_{\tau_1}(\gamma(e_1))$. It then follows from the definition of $R_{\tau_1 \rightarrow \tau}$ that $R_\tau(\gamma(e_0)\ \gamma(e_1))$. And finally, $R_\tau(\gamma(e_0)\ \gamma(e_1)) = R_\tau(\gamma(e_0\ e_1))$.

- Case $e = \lambda x\!:\!\tau_1.\, e_2$: Assume $\Gamma \vdash \lambda x : \tau_1.e_2 : \tau_1 \rightarrow \tau_2 \wedge \gamma \models \Gamma$. In order to show that $R_{\tau_1 \rightarrow \tau_2}(e)$, we need to show that

  $$(\vdash \gamma(e) : \tau_1 \rightarrow \tau_2) \wedge (\exists v.\gamma(e) \longrightarrow^* v) \wedge (\forall e''.R_{\tau_1}(e'') \implies R_{\tau_2}(\gamma(e)\ e''))$$

  For the first clause, it can be shown by using the substitution lemma on our assumptions, i.e.

  $$\Gamma \vdash \lambda x\!:\!\tau_1.\, e_2 : \tau_1 \rightarrow \tau_2 \wedge \gamma \models \Gamma \implies \ \vdash \gamma(\lambda x\!:\!\tau_1.\, e_2) : \tau_1 \rightarrow \tau_2$$

  The second clause follows from the definition of γ, $\gamma(\lambda x\!:\!\tau_1.\, e_2) = \lambda x\!:\!\tau_1.\, \gamma(e_2)$, which is a value. We now need to prove the third clause.

  Consider an arbitrary e'' and assume $R_{\tau_1}(e'')$. It needs to be shown that $R_{\tau_2}(\gamma(e)\ e'')$. We first note that $\gamma(e) = \lambda x\!:\!\tau_1.\, (\gamma\backslash x)\ e_2$, where $\gamma\backslash x$ is simply γ on all values except x. And since the evaluation rules of $\lambda^{\rightarrow}$ is CBV, we have the following

  $$\begin{aligned} & \gamma(e)\ (e'') \longrightarrow^* \gamma(e)\ v'' \\ \longrightarrow \quad & ((\gamma\backslash x)(e_2))\{v''/x\} = \gamma'(e_2)\end{aligned}$$

where $\gamma' = \gamma[x \mapsto v'']$. Recall the lemma

$$\vdash e : \tau \wedge e \longrightarrow e' \wedge R_\tau(e') \iff R_\tau(e)$$

Thus, if $R_{\tau_2}(\gamma'(e_2))$, then $R_{\tau_2}(\gamma(e)\ e'')$. Therefore, we now only need to show $R_{\tau_2}(\gamma'(e_2))$.

We now prove this by the typing derivations of one of our assumptions. Recall the assumption, $\Gamma \vdash \lambda x\!:\!\tau_1.\, e : \tau_1 \rightarrow \tau_2$. Its typing derivation has the form

$$\frac{\Gamma, x : \tau_1 \vdash e_2 : \tau_2}{\Gamma \vdash \lambda x\!:\!\tau_1.\, e_2 : \tau_1 \rightarrow \tau_2}$$

It is now important to notice that $\gamma' \models \Gamma, x : \tau_1$. This is because $\gamma \models \Gamma$ and $\gamma'(x) = v'' \in R_{\Gamma(x)}$, where $\Gamma(x) = \tau_1$. Hence, by our induction hypothesis, $\gamma'(e_2) \in R_{\tau_2}$, and this completes our proof.

Logical relations is a powerful technique that can be used to prove properties of more complex languages.