Logical Relations Part 1

Lecture 1 Wednesday, August 30, 2017

1 Logical Relations

We often want to show that a property holds of all (well-typed) terms in a language. A standard way of proving such a property is to perform induction on the structure of the term. However, for some properties, it may be difficult to find the right induction hypothesis, since evaluation of the program may not preserve the structure of terms.

For example, consider trying to prove strong normalization (i.e., termination) of terms in the simply-typed lambda calculus, and specifically consider the case for application $e_1\ e_2$. The induction hypothesis might say that terms e_1 and e_2 are strongly normalizing, and so e_1 will evaluate to an abstraction $\lambda x.\, e$ and e_2 will evaluate to a value v. But $(\lambda x.\, e)\ v$ steps to $e\{v/x\}$, and our inductive hypothesis has nothing to say about this new term!

Logical relations are a powerful technique to help specify the a sufficiently strong inductive hypothesis. In this lecture, we will prove the strong normalization of simply-typed lambda calculus using logical relations. This proof technique was invented by Tait in 1967, to prove strong normalization (i.e., normalization in a language with full-beat reduction, i.e., nondeterministic evaluation order). Plotkin coined the phrase "logical relations", but don't try to read too much into the name. Logical relations are so-called because they preserve the property of interest across logical implication (i.e., functions) and types with logical counterparts under the Curry-Howard isomorphism.

2 Proof of Normalization Using Logical Relations

(This proof is based on the presentation in Chapter 12 of "Types and Programming Languages" by Benjamin C. Pierce, MIT Press, 2002.)

Suppose we have a simply-typed lambda calculus with unit. (That is, the only base type is **Unit**.) The syntax is defined as follows.

$$
\begin{aligned}
e &::= x \mid \lambda x{:}\tau.\, e \mid e_1\ e_2 \mid () \\
v &::= \lambda x{:}\tau.\, e \mid () \\
\tau &::= \textbf{Unit} \mid \tau_1 \to \tau_2
\end{aligned}
$$

We start by defining a logical relation for each type: for each type τ, the set R_τ will be a set of closed terms of type τ, and we write $R_\tau(e)$ if $e \in R_\tau$. (For this example, we are using unary logical relations, and so could maybe call them *logical predicates*.)

$$
\begin{aligned}
R_{\textbf{Unit}}(e) &\text{ iff } e \text{ halts} \\
R_{\tau_1 \to \tau_2}(e) &\text{ iff } e \text{ halts and for all } e' \text{ such that } R_{\tau_1}(e'), \text{ we have } R_{\tau_2}(e\ e')
\end{aligned}
$$

Note that if we consider the Curry-Howard isomorphism and regard function types as logical implication, the definition of $R_{\tau_1 \to \tau_2}$ preserves the logical relation over implication: if $R_{\tau_1}(e')$ and $R_{\tau_1 \to \tau_2}(e)$ then $R_{\tau_2}(e\ e')$.

The proof of strong normalization proceeds in two steps. First we show that if $e \in R_\tau$, then e terminates. Second, we show that if e is a well-typed expression of type τ, then $e \in R_\tau$.

Lemma 1. *If $R_\tau(e)$ then e terminates.*

Proof. Immediate from the definition of R_τ. □

To prove the second step, we use two lemmas. The first lemma shows that membership in R_τ is invariant under evaluation. The second lemma shows quite directly that if e is a well-typed expression of type τ, then $e \in R_\tau$. However, in this second lemma we need to strengthen the inductive hypothesis to handle expressions with free variables. We do so by considering expressions with free variables have the variables substituted with suitable values.

Lemma 2. *If $e \longrightarrow e'$ then $R_\tau(e)$ iff $R_\tau(e')$.*

Proof. Assume $\vdash e : \tau$ and $e \longrightarrow e'$. First, note that it is clear in this language that e terminates if and only if e' terminates.

Now, consider the $\Rightarrow$ direction, which we will prove by induction on the structure of type τ. Suppose that $R_\tau(e)$. For the base case $\tau = \textsf{Unit}$, the result is immediate. Consider the case $\tau = \tau_1 \to \tau_2$. We need to show that for all $e_1 \in R_{\tau_1}$, we have $R_{\tau_2}(e'\ e_1)$. But we have $e\ e_1 \longrightarrow e'\ e_1$, which by the induction hypothesis for type τ_2 means that $R_{\tau_2}(e'\ e_1)$.

The other direction ($\Leftarrow$) is similar. □

Lemma 3. *If $x_1 : \tau_1, \ldots, x_n : \tau_n \vdash e : \tau$ and $v_1, \ldots, v_n$ are closed values of type $\tau_1, \ldots, \tau_n$, and for all $i \in 1..n$ we have $R_{\tau_i}(v_i)$, then $R_\tau(e\{v_1/x_1\} \ldots \{v_n/x_n\})$.*

Proof. We proceed by induction on the typing derivation of $x_1 : \tau_1, \ldots, x_n : \tau_n \vdash e : \tau$.

- **Case T-Var.** Here, $e = x_i$ and $\tau = \tau_i$. Since $R_{\tau_i}(v_i)$ and $v_i\{v_1/x_1\} \ldots \{v_n/x_n\} = v_i$, the result is immediate.
- **Case T-Unit.** Here, $e = ()$ and $\tau = \textsf{Unit}$. Since $R_{\textsf{Unit}}(\,())$ and $()\{v_1/x_1\} \ldots \{v_n/x_n\} = ()$, the result is immediate.
- **Case T-App.** Here, $e = e_1\ e_2$, and we have the following:

  $$x_1 : \tau_1, \ldots, x_n : \tau_n \vdash e_1 : \tau_2 \to \tau$$
  $$x_1 : \tau_1, \ldots, x_n : \tau_n \vdash e_2 : \tau_2$$

  By the inductive hypothesis, we have $R_{\tau_2 \to \tau}(e_1\{v_1/x_1\} \ldots \{v_n/x_n\})$ and $R_{\tau_2}(e_2\{v_1/x_1\} \ldots \{v_n/x_n\})$. By the definition of $R_{\tau_2 \to \tau}$ we have

  $$\begin{aligned} &R_\tau(e_1\{v_1/x_1\} \ldots \{v_n/x_n\}\ e_2\{v_1/x_1\} \ldots \{v_n/x_n\}) \\ =&R_\tau((e_1\ e_2)\{v_1/x_1\} \ldots \{v_n/x_n\} \end{aligned}$$

- **Case T-Abs.** Here $e = \lambda x : \tau_x.\, e'$ and $\tau = \tau_x \to \tau'$ and we have:

  $$x_1 : \tau_1, \ldots, x_n : \tau_n, x : \tau_x \vdash e' : \tau'$$

  Clearly, e is a value, and so e terminates. So we need to show that for any e'' such that $R_{\tau_x}(e'')$, we have $R_{\tau'}((e\{v_1/x_1\} \ldots \{v_n/x_n\})\ e'')$.

  Since $R_{\tau_x}(e'')$, by Lemma 1 e'' will terminate, i.e., will evaluate to a value v''. Moreover, by Lemma 2 we have $R_{\tau_x}(v'')$. So we have

  $$\begin{aligned} (e\{v_1/x_1\} \ldots \{v_n/x_n\})\ e'' \longrightarrow^* &(e\{v_1/x_1\} \ldots \{v_n/x_n\})\ v'' \\ = &((\lambda x : \tau_x.\, e')\{v_1/x_1\} \ldots \{v_n/x_n\})\ v'' \\ \longrightarrow &e'\{v_1/x_1\} \ldots \{v_n/x_n\}\{v''/x\} \end{aligned}$$

  But by the inductive hypothesis, we have $R_{\tau'}(e'\{v_1/x_1\} \ldots \{v_n/x_n\}\{v''/x\})$.

□

Strong normalization follows easily from Lemma 3.

Theorem 1. *If $\vdash e : \tau$ then e terminates.*