Logical Relations and Strong Normalization

Solomon Aduol Maina

December 19, 2016

1 Introduction

Logical relations are a proof technique that yield simple proofs of theorems that are not readily solved by standard induction on typing relations or stepping relations. In this project we give proofs of strong normalization of the Simply Typed Lambda Calculus (STLC), System **T** and System **F** using logical relations.

2 Strong Normalization of STLC

2.1 The Syntax, Statics and Dynamics of STLC

The syntax of the Simply Typed Lambda Calculus is given by the following grammar:

$$\begin{aligned} &\textit{Typ}\ \tau &&::= nat \mid \tau_1 \longrightarrow \tau_2 \\ &\textit{Exp}\ e &&::= x \mid 0 \mid \mathbf{S}(e) \mid \lambda(x:\tau).e \mid e_1\ e_2 \end{aligned}$$

The typing rules of STLC are as follows:

$$\frac{}{\Gamma, x:\tau \vdash x:\tau}\ \text{Typ-Var}$$

$$\frac{}{\Gamma \vdash 0 : nat}\ \text{Typ-Zero}$$

$$\frac{\Gamma \vdash e : nat}{\Gamma \vdash \mathbf{S}(e) : nat}\ \text{Typ-Succ}$$

$$\frac{\Gamma, x:\tau_1 \vdash e:\tau_2}{\Gamma \vdash \lambda(x:\tau_1).e : \tau_1 \longrightarrow \tau_2}\ \text{Typ-Lam}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \longrightarrow \tau_2 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash e_1\ e_2 : \tau_2}\ \text{Typ-App}$$

The closed values of STLC are defined by the following rules:

$$\frac{}{0\ \mathbf{Val}}\quad \text{Val-Zero}$$

$$\frac{e\ \mathbf{Val}}{\mathbf{S}(e)\ \mathbf{Val}}\quad \text{Val-Succ}$$

$$\frac{}{\lambda(x:\tau).e\ \mathbf{Val}}\quad \text{Val-Lam}$$

The transition rules for STLC are as follows:

$$\frac{e \longmapsto e'}{\mathbf{S}(e) \longmapsto \mathbf{S}(e')}\quad \text{Step-Succ}$$

$$\frac{e_1 \longmapsto e_1'}{e_1\ e_2 \longmapsto e_1'\ e_2}\quad \text{Step-App}$$

$$\frac{}{(\lambda(x:\tau).e_1)\ e_2 \longmapsto e_1[x ::= e_2]}\quad \text{Step-Lam}$$

2.2 Why Logical Relations?

One might try to prove that STLC is normalizing by proving the following:

Proposition 1. If $\vdash e : \tau$ then there exists a value v such that $e \longmapsto^* v$.

Since the hypothesis includes just the assumption $\vdash e : \tau$, the obvious strategy would be to prove by induction on the derivation $\vdash e : \tau$, but the Typ-App case fails:

$$\frac{\vdash e_1 : \tau_1 \longrightarrow \tau_2 \qquad \vdash e_2 : \tau_1}{\vdash e_1\ e_2 : \tau_2}\quad \text{Typ-App}$$

By the induction hypothesis there exists a value f such that $e_1 \longmapsto^* f$. By canonical forms, $f = \lambda(x : \tau_1).e'$, hence by the Step-App rule,

$$e_1\ e_2 \longmapsto^* (\lambda(x:\tau_1).e')\ e_2 \longmapsto e'[x := e_2]$$

We are stuck in this case firstly because we do not know that terminating terms are closed under converse multistep evaluation, secondly because we do not know that substituting terminating terms into arbitrary expressions creates terminating expressions, and thirdly (and most importantly) we have no information about the body e' of the lambda expression since the induction hypothesis does not apply to e'.

The solution to this problem is to define a unary logical relation $\mathcal{R}_\tau$ over expressions of a closed type τ such that if $e \in \mathcal{R}_\tau$ then there exists a value v such that $e \longmapsto^* v$. This relation should satisfy the property that if $e \in \mathcal{R}_{\tau_1 \longrightarrow \tau_2}$ and $e' \in \mathcal{R}_{\tau_1}$, then $(e\ e') \in \mathcal{R}_{\tau_2}$. We then show that if $\vdash e : \tau$, then $e \in \mathcal{R}_\tau$, so there exists a value v such that $e \longmapsto^* v$. This thus leads us to the following definition:

Definition 1. We define a unary relation $\mathcal{R}_\tau$ on closed expressions of the type τ by the following rules: if e is a closed expression, then $e \in \mathcal{R}_\tau$ if and only if

1. if $\tau = nat$, then there exists a value v such that $e \longmapsto^* v$, and
2. if $\tau = \tau_1 \longrightarrow \tau_2$, then
   - there exists a unique v such that $e \longmapsto^* v$, and
   - if $e' \in \mathcal{R}_{\tau_1}$, then $(e\ e') \in \mathcal{R}_{\tau_2}$

We then attempt to prove the following:

Proposition 2. If $\vdash e : \tau$, then $e \in \mathcal{R}_\tau$.

Again the proof proceeds by induction over the derivation $\vdash e : \tau$. We now find that the Typ-App case is provable:

$$\frac{\vdash e : \tau_1 \longrightarrow \tau_2 \qquad \vdash e' : \tau_1}{\vdash e\ e' : \tau_2}\ \text{Typ-App}$$

By the induction hypothesis $e \in \mathcal{R}_{\tau_1 \longrightarrow \tau_2}$ and $e' \in \mathcal{R}_{\tau_1}$, so $(e\ e') \in \mathcal{R}_{\tau_2}$. However, there is a further complication in the Typ-Lam Case:

$$\frac{x : \tau_1 \vdash e : \tau_2}{\vdash \lambda(x : \tau_1).e : \tau_1 \longrightarrow \tau_2}\ \text{Typ-Lam}$$

$\lambda(x : \tau_1).e$ is already a value. Assume that $e' \in \mathcal{R}_{\tau_1}$. Then $(\lambda(x : \tau_1).e)\ e' \longmapsto e[x := e']$ We are stuck again, firstly because we do not know that terminating terms are closed under converse evaluation, and because we do not know anything about e' since e' is not a closed term and hence the induction hypothesis does not apply to it. We therefore need to generalize $\mathcal{R}_\sigma$ so that it applies to open terms as well as closed terms. This leads to the following definition:

Notation 1. Let γ be a function from a finite set $\{x_1, \ldots, x_n\}$ of expression variables to closed expressions. Let e be an arbitary expression. Then

$$\hat{\gamma}(e) = e[(x_1, \ldots, x_n) := (\gamma(x_1), \ldots, \gamma(x_n))]$$

Definition 2. Let e be an (arbitary) expression, τ a type and $\Gamma = \{x_1 : \tau_1, \ldots, x_n : \tau_n\}$ a context. Then $e \in \mathcal{S}(\Gamma, \tau)$ if and only if when γ is a function from $\{x_1, \ldots, x_n\}$ to closed expressions such that $\gamma(x_i) : \tau_i$ and $\gamma(x_i) \in \mathcal{R}_{\tau_i}$, then $\hat{\gamma}(e) \in \mathcal{R}_\tau$

We also need to prove the following proposition:

Proposition 3. Assume that $\vdash e : \tau$ and $\vdash e' : \tau$. Assume that $e \longmapsto e'$. If $e' \in \mathcal{R}_\tau$, then $e \in \mathcal{R}_\tau$.

Proof. Proof. The proof proceeds by induction over the type τ.

1. (Case $\tau = nat$) Since $e' \in \mathcal{R}_{nat}$, then there exists a value v such that $e' \longmapsto^* v$, thus $e \longmapsto^* v$ as $e \longmapsto e'$ by assumption, so $e \in \mathcal{R}_{nat}$.

2. (Case $\tau = \tau_1 \longrightarrow \tau_2$) Since $e' \in \mathcal{R}_{\tau_1 \longrightarrow \tau_2}$, then there exists a value v such that $e' \longmapsto^* v$, thus $e \longmapsto^* v$ as $e \longmapsto e'$ by assumption. Now assume that $e_1 \in \mathcal{R}_{\tau_1}$. Then $(e\ e_1) \longmapsto (e'\ e_1)$. Since $e' \in \mathcal{R}_{\tau_1 \longrightarrow \tau_2}$, then $(e'\ e_1) \in \mathcal{R}_{\tau_2}$. By the induction hypothesis, $(e\ e_1) \in \mathcal{R}_{\tau_2}$, thus $e \in \mathcal{R}_{\tau_1 \longrightarrow \tau_2}$. This completes the proof.

□

2.3 Completing the Proof

We are now in a position to prove our desired proposition:

Proposition 4. If $\Gamma \vdash e : \tau$ then $e \in \mathcal{S}(\Gamma, \tau)$.

Proof. Assume that $\Gamma \vdash e : \tau$ with $\Gamma = \{x_1 : \tau_1, \ldots, x_n : \tau_n\}$ a context. Let γ be a function from $\{x_1, \ldots, x_n\}$ to closed expressions such that $\gamma(x_i) : \tau_i$ and $\gamma(x_i) \in \mathcal{R}_{\tau_i}$. The proof proceeds by induction on the derivation $\Gamma \vdash e : \tau$.

- (Case Typ-Var)

$$\frac{}{\Gamma', x : \tau \vdash x : \tau}\ \text{Typ-Var}$$

Then $x = x_i$ for some $1 \leq i \leq n$, so $\hat{\gamma}(x) = \gamma(x_i)$ which is in $\mathcal{R}_{\tau_i}$ by the definition of γ, so $x \in \mathcal{S}(\Gamma, \tau)$.

- (Case Typ-Zero)

$$\frac{}{\Gamma \vdash 0 : nat}\ \text{Typ-Zero}$$

Then $\hat{\gamma}(0) = 0$ which is in $\mathcal{R}_{nat}$ so $0 \in \mathcal{S}(\Gamma, \tau)$.

- (Case Typ-Succ)

$$\frac{\Gamma \vdash e : nat}{\Gamma \vdash \mathbf{S}(e) : nat}\ \text{Typ-Succ}$$

Then $\hat{\gamma}(\mathbf{S}(e)) = \mathbf{S}(\hat{\gamma}(e))$. By the induction hypothesis, $e \in \mathcal{S}(\Gamma, nat)$, so $\hat{\gamma}(e) \in \mathcal{R}_{nat}$, so $\hat{\gamma}(e) \longmapsto^* v$ with v a value, so $\mathbf{S}(\hat{\gamma}(e)) \longmapsto^* \mathbf{S}(v)$ which is a value so $\mathbf{S}(\hat{\gamma}(e)) \in \mathcal{R}_{nat}$, so $\mathbf{S}(e) \in \mathcal{S}(\Gamma, nat)$.

- (Case Typ-App)

$$\frac{\Gamma \vdash e_1 : \tau_1 \longrightarrow \tau_2 \qquad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1\ e_2 : \tau_2}\ \text{Typ-App}$$

Then $\hat{\gamma}(e_1\ \ e_2) = (\hat{\gamma}(e_1))\ (\hat{\gamma}(e_2))$. By the induction hypothesis, $e_1 \in \mathcal{S}(\Gamma, \tau_1 \longrightarrow \tau_2)$ and $e_2 \in \mathcal{S}(\Gamma, \tau_1)$, so $\hat{\gamma}(e_1) \in \mathcal{R}_{\tau_1 \longrightarrow \tau_2}$ and $\hat{\gamma}(e_2) \in \mathcal{R}_{\tau_1}$, so $(\hat{\gamma}(e_1))\ (\hat{\gamma}(e_2)) \in \mathcal{R}_{\tau_2}$, so $(e_1\ e_2) \in \mathcal{S}(\Gamma, \tau)$.

- (Case Typ-Lam)

  $$\frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda(x : \tau_1).e : \tau_1 \longrightarrow \tau_2} \quad \text{Typ-Lam}$$

  Assume without loss of generality that x is not in Γ since otherwise we can rename x to satisfy this condition. Then $\hat{\gamma}(\lambda(x : \tau_1).e) = \lambda(x : \tau_1).\hat{\gamma}(e)$ which is a value. Assume that $e' \in \mathcal{R}_{\tau_1}$. Let $\theta = \gamma \otimes [x \mapsto e']$. By the induction hypothesis, $e \in \mathcal{S}((\Gamma, x : \tau_1), \tau_2)$, so $\hat{\theta}(e) \in \mathcal{R}_{\tau_2}$. Now

  $$(\lambda(x : \tau_1).\hat{\gamma}(e))\ e' \longmapsto \hat{\gamma}(e)[x := e'] = \hat{\theta}(e)$$

  which is in $\in \mathcal{R}_{\tau_2} \ldots$ so $(\lambda(x : \tau_1).\hat{\gamma}(e))\ e' \in \mathcal{R}_{\tau_2}$ as $\mathcal{R}_{\tau_2}$ is closed under converse evaluation. Therefore $\hat{\gamma}(\lambda(x : \tau_1).e) \in \mathcal{R}_{\tau_1 \longrightarrow \tau_2}$. This completes the proof.

Corollary 1. If $\vdash e : \tau$ then there exists a value v such $e \longmapsto^* v$

Since the transition rules of STLC are deterministic we get the following:

Corollary 2. If $\vdash e : \tau$ then there exists a unique value v such $e \longmapsto^* v$

3 Strong Normalization of System T

3.1 Syntax, Statics and Dynamics of T

System **T** is an extension of the Simply Typed Lambda Calculus, with the added term $\mathbf{Prec}(e_1, e_2)(e)$

$$\begin{aligned} &\mathit{Typ}\ \tau\ \ ::= \mathit{nat} \mid \tau_1 \longrightarrow \tau_2 \\ &\mathit{Exp}\ e\ \ ::= x \mid 0 \mid \mathbf{S}(e) \mid \lambda(x : \tau).e \mid e_1\ e_2 \mid \mathbf{Prec}(e_1, e_2)(e) \end{aligned}$$

We add the typing rule

$$\frac{\Gamma \vdash e : \mathit{nat} \qquad \Gamma \vdash e_0 : \tau \qquad \Gamma \vdash e_1 : \mathit{nat} \longrightarrow (\tau \longrightarrow \tau)}{\Gamma \vdash \mathbf{Prec}(e_0, e_1)(e) : \tau} \ \text{Typ-Prec}$$

We also add the following transition rules:

$$\frac{e \longmapsto e'}{\mathbf{Prec}(e_0, e_1)(e) \longmapsto \mathbf{Prec}(e_0, e_1)(e')} \quad \text{Step-Prec-1}$$

$$\frac{}{\mathbf{Prec}(e_0, e_1)(0) \longmapsto e_0} \quad \text{Step-Prec-2}$$

$$\frac{e\ \mathbf{Val}}{\mathbf{Prec}(e_0, e_1)(\mathbf{S}(e)) \longmapsto (e_1\ e)\ (\mathbf{Prec}(e_0, e_1)(e))} \quad \text{Step-Prec-3}$$

3.2 The Proof of Strong Normalization

As with STCL, we prove the following

Proposition 5. If $\Gamma \vdash e : \tau$ then $e \in \mathcal{S}(\Gamma, \tau)$.

Proof. The proof procees by induction over the derivation $\Gamma \vdash e : \tau$. Since **T** is an extension of STLC, it suffices to prove the Typ-Prec case.

$$\frac{\Gamma \vdash e : nat \qquad \Gamma \vdash e_0 : \tau \qquad \Gamma \vdash e_1 : nat \longrightarrow (\tau \longrightarrow \tau)}{\Gamma \vdash \mathbf{Prec}(e_0, e_1)(e) : \tau}\text{Typ-Prec}$$

Then $\hat{\gamma}(\mathbf{Prec}(e_0, e_1)(e)) = \mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(\hat{\gamma}(e))$. By the induction hypothesis, $e \in \mathcal{S}(\Gamma, nat)$, $e_0 \in \mathcal{S}(\Gamma, \tau)$ and $e_1 \in \mathcal{S}(nat \longrightarrow (\tau \longrightarrow \tau))$, so $\hat{\gamma}(e) \in \mathcal{R}_{nat}$, $\hat{\gamma}(e_0) \in \mathcal{R}_\tau$ and $\hat{\gamma}(e_1) \in \mathcal{R}_{nat \longrightarrow (\tau \longrightarrow \tau)}$. Therefore, there exists a value v such that $\hat{\gamma}(e) \in \mathcal{R}_{nat} \longmapsto^* v$, so

$$\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(\hat{\gamma}(e)) \longmapsto^* \mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(v)$$

Here we see that we need to strengthen the converse evaluation lemma to the following:

Proposition 6. Assume that $\vdash e : \tau$ and $\vdash e' : \tau$. Assume that $e \longmapsto^* e'$. If $e' \in \mathcal{R}_\tau$, then $e \in \mathcal{R}_\tau$.

The proof of this lemma is a carbon copy of the original lemma, but with $\longrightarrow$ substituted for $\longrightarrow^*$.
Now that we have reduced the case $\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(\hat{\gamma}(e))$ to proving the proposition for $\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(v)$ with v a value, then we can apply natural number induction to finish the proof as follows:

- (Case $v = 0$) Then $\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(v) \longmapsto \hat{\gamma}(e_0)$ which is in $\mathcal{R}_\tau$, so $\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(v) \in \mathcal{R}_\tau$ as $\mathcal{R}_\tau$ is closed under converse evaluatio

- (Case $v = \mathbf{S}(e')$ with e' a value) Then

  $$\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(\mathbf{S}(e')) \longmapsto ((\hat{\gamma}(e_1))\ e')\ (\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(e'))$$

  Since e' is a value, then by the (natural number) induction hypothesis, $\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(e') \in \mathcal{R}_\tau$, and since $\hat{\gamma}(e_1) \in \mathcal{R}_{nat \longrightarrow (\tau \longrightarrow \tau)}$ and e' is a value (and hence is in $\mathcal{R}_{nat}$), it follows that

  $$((\hat{\gamma}(e_1))\ e')\ (\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(e')) \in \mathcal{R}_\tau,$$

  so $\mathbf{Prec}(\hat{\gamma}(e_0), \hat{\gamma}(e_1))(\mathbf{S}(e')) \in \mathcal{R}_\tau$ as $\mathcal{R}_\tau$ is closed under converse evaluation. This completes the proof.

□

Remark 1. System **T** retains its strong normalization property if we add any "finite" type to it e.g. boolean values, binary trees or any type which is the solution of a polynomial type equation.

Remark 2. A call-by-value semantics would require that we prove the additional property that the relation is closed under forward multistep evaluation. This is because in the Typ-Lam case where we have $\lambda x : \tau.e$ being applied to an expression e' which is in our relation, then we first have to evaluate e' to a value until we substitute for x in the body of e, so we have to prove that our relation is closed under forward multistep evaluation.

4 Strong Normalization of System F

4.1 Syntax, Statics and Dynamics of F

The syntax of **F** is given by the following grammar:

$$\begin{aligned} &\mathit{Typ}\ \tau ::= t \mid \tau \longrightarrow \tau' \mid \forall t.\tau \\ &\mathit{Exp}\ e ::= x \mid \lambda(x : \tau).e \mid e\ e' \mid \Lambda t.e \mid e\ [\tau] \end{aligned}$$

Let Δ be a set of type variables. The well-formedness of types is given by the following rules:

$$\frac{}{\Delta, t\ \mathbf{WF} \vdash t\ \mathbf{WF}}\ \text{WF-Var}$$

$$\frac{\Delta \vdash \tau_1\ \mathbf{WF} \qquad \Delta \vdash \tau_2\ \mathbf{WF}}{\Delta \vdash \tau_1 \longrightarrow \tau_2\ \mathbf{WF}}\ \text{WF-lam}$$

$$\frac{\Delta,\ t\ \mathbf{WF} \vdash \tau\ \mathbf{WF}}{\Delta \vdash \Lambda t.\tau\ \mathbf{WF}}\ \text{WF-Lam}$$

The typing rules of **F** are given by the following rules:

$$\frac{}{(\Delta, (\Gamma, x : \tau)) \vdash x : \tau}\ \text{Typ-Var}$$

$$\frac{\Delta \vdash \tau_1\ \mathbf{WF} \qquad (\Delta, (\Gamma, x : \tau_1)) \vdash e : \tau_2}{(\Delta, \Gamma) \vdash \lambda(x : \tau_1).e : \tau_1 \longrightarrow \tau_2}\ \text{Typ-lam}$$

$$\frac{(\Delta, \Gamma) \vdash e_1 : \tau_2 \longrightarrow \tau \qquad (\Delta, \Gamma) \vdash e_2 : \tau_2}{(\Delta, \Gamma) \vdash (e_1\ e_2) : \tau}\ \text{Typ-app}$$

$$\frac{(\Delta \cup \{t\}, \Gamma) \vdash e : \tau}{(\Delta, \Gamma) \vdash \Lambda t.e : \forall t.\tau}\ \text{Typ-Lam}$$

$$\frac{(\Delta, \Gamma) \vdash e : \forall t.\tau' \qquad \Delta \vdash \tau\ \mathbf{WF}}{(\Delta, \Gamma) \vdash e\ [\tau] : \tau'[t := \tau]}\ \text{Typ-App}$$

The closed values of **F** are defined by the following rules:

$$\frac{}{\lambda(x : \tau).e\ \mathbf{Val}}\ \text{Val-lam}$$

$$\frac{}{\Lambda t.e\ \textbf{Val}}\ \text{Val-Lam}$$

The dynamics of **F** are defined by the following rules:

$$\frac{e \longmapsto e'}{(e\ e_1) \longmapsto (e'\ e_1)}\ \text{Step-app-1}$$

$$\frac{}{(\lambda(x:\tau).e)\ e' \longmapsto e[x := e']}\ \text{Step-lam}$$

$$\frac{e \longmapsto e'}{(e\ [\tau]) \longmapsto (e'\ [\tau])}\ \text{Step-App}$$

$$\frac{}{(\Lambda t.e)\ [\tau] \longmapsto e[t := \tau]}\ \text{Step-Lam}$$

4.2 Defining the Relation

If we proceed as in STLC and define the desired relation on types, then for the case $\forall t.\tau$, the obvious strategy would be to say if $(\Delta, \Gamma) \vdash \forall t.\tau$, then e is in the relation at type $\forall t.\tau$ if and only if for all closed types τ', we have $e\ [\tau']$ is in the relation at type $\tau[t := \tau']$. However, $\tau[t := \tau']$ is not structurally smaller than $\forall t.\tau$. This therefore calls for some indirection. In particular, we want to find some condition P (which we can show holds), so that if P holds, then e is in the relation at type $\tau[t := \tau']$. As it turns out, this definition suffices:

Definition 3. Let e be a closed expression and τ a type. Let δ be a function from type variables to closed types. Let η be a function from type variables to the set of unary relations $\mathcal{R}_\sigma$ over expressions of the closed type σ that are closed under converse evaluation. Then $e \in \mathcal{CHT}(\delta, \eta, \tau)$ if and only if the following hold:

1. if $\tau = t$, then $e \in \mathcal{CHT}(\delta, \eta, \tau)$ if and only if $e \in \eta(t)$,
2. if $\tau = \tau_1 \longrightarrow \tau_2$, then $e \in \mathcal{CHT}(\delta, \eta, \tau)$ if and only if
   - there exists a value e' such that $e \longmapsto^* e'$, and
   - if $e_1 \in \mathcal{CHT}(\delta, \eta, \tau_1)$, then $(e\ e_1) \in \mathcal{CHT}(\delta, \eta, \tau_2)$
3. if $\tau = \forall t.\tau'$, then $e \in \mathcal{CHT}(\delta, \eta, \tau)$ if and only if
   - there exists a value e' such that $e \longmapsto^* e'$, and
   - for any closed type σ and any relation $\mathcal{R}_\sigma$ over expressions of type σ that is closed under converse evaluation, we have

$$e\ [\sigma] \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau')$$

4.3 Two Key Lemmata

This definition gives us the desired property because we now have the following fact which gives us a way of proving that $e \in \mathcal{CHT}(\delta, \eta, \tau[t := \tau'])$, assuming that we can prove the appropriate hypothesis:

Lemma 1. *Let e be a closed expression. Let τ be a type. Let σ be a closed type and $\mathcal{R}_\sigma$ a unary relation over expressions of type σ that is closed under converse evaluation. Let δ be a function from a set of type variables to closed types and η a function from a set of type variables to the set of unary relations $\mathcal{R}_\sigma$ over expressions of the closed type σ that are closed under converse evaluation. Assume that $e \in R_\sigma$ if and only if $e \in \mathcal{CHT}(\delta, \eta, \sigma)$. Then $e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto R_\sigma], \tau)$ if and only if $e \in \mathcal{CHT}(\delta, \eta, \tau[t := \sigma])$.*

Proof. The proof proceeds by induction over τ.

Case $(\tau = t')$ $(\Rightarrow)$ Assume that

$$e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau)$$

If $t = t'$, then $\tau[t := \sigma] = \sigma$. Since $e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto R_\sigma], \tau)$, it follows that $e \in R_\sigma$, thus $e \in \mathcal{CHT}(\delta, \eta, \sigma)$. If $t \neq t'$, then $\tau[t := \sigma] = t'$. Since $e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto R_\sigma], \tau)$, it follows that $e \in \eta(t')$, thus $e \in \mathcal{CHT}(\delta, \eta, t')$.
$(\Leftarrow)$ Assume that

$$e \in \mathcal{CHT}(\delta, \eta, \tau[t := \sigma])$$

If $t = t'$, then $\tau[t := \sigma] = \sigma$, thus $e \in \mathcal{R}_\sigma$, so $e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto R_\sigma], \tau)$. If $t \neq t'$, then $\tau[t := \sigma] = t'$, so $e \in \eta(t')$, hence $e \in (\eta \otimes [t \mapsto R_\sigma])(t')$ and $e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto R_\sigma], \tau)$

Case $(\tau = \tau_1 \longrightarrow \tau_2)$ Then $(\tau_1 \longrightarrow \tau_2)[t := \sigma] = (\tau_1[t := \sigma]) \longrightarrow (\tau_2[t := \sigma])$.
$(\Rightarrow)$ Assume that

$$e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau)$$

Assume that

$$e' \in \mathcal{CHT}(\delta, \eta, \tau_1)$$

By the induction hypothesis, $e' \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto R_\sigma], \tau_1)$, thus $(e\ e') \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto R_\sigma], \tau_2)$, thus $(e\ e') \in \mathcal{CHT}(\delta, \eta, \tau_2[t := \sigma])$ by another application of the induction hypothesis. Thus $e \in \mathcal{CHT}(\delta, \eta, \tau[t := \sigma])$
$(\Leftarrow)$ Assume that

$$e \in \mathcal{CHT}(\delta, \eta, \tau[t := \sigma])$$

Assume that

$$e' \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto R_\sigma], \tau_1[t := \sigma])$$

By the induction hypothesis, $e' \in \mathcal{CHT}(\delta, \eta, \tau_1[t := \sigma])$, thus $(e\ e') \in \mathcal{CHT}(\delta, \eta, \tau_2[t := \sigma])$. By another application of the induction hypothesis, $(e\ e') \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau_2)$, thus
$e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau)$.

Case $(\tau = \forall t'.\tau')$ Assume without loss of generality that $t' \neq t$ since otherwise we can rename t' so that $t' \neq t$. Then $\tau[t := \sigma] = \forall t'.\tau'[t := \sigma]$.
$(\Rightarrow)$ Assume that

$$e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta[t \mapsto R_\sigma], \forall t'.\tau')$$

Let ρ be a closed type and R_ρ a unary relation over expressions of type ρ. Then

$$e\ [\rho] \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma] \otimes [t' \mapsto \rho], \eta \otimes [t \mapsto \mathcal{R}_\sigma] \otimes [t' \mapsto \mathcal{R}_\rho], \tau')$$

By the induction hypothesis

$$e\ [\rho] \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau'[t := \sigma]),$$

so $e \in \mathcal{CHT}(\delta, \eta, \forall t'.\tau'[t := \sigma])$.
$(\Leftarrow)$ Assume that

$$e \in \mathcal{CHT}(\delta, \eta, \forall t'.\tau'[t := \sigma])$$

Let ρ be a closed type and R_ρ a unary relation over expressions of type ρ. Then

$$e\ [\rho] \in \mathcal{CHT}(\delta \otimes [t' \mapsto \rho], \eta \otimes [t' \mapsto \mathcal{R}_\rho], \tau'[t := \sigma])$$

By the induction hypothesis,

$$e\ [\rho] \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma] \otimes [t' \mapsto \rho], \eta \otimes [t \mapsto \mathcal{R}_\sigma] \otimes [t' \mapsto \mathcal{R}_\rho], \tau'),$$

so $e \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta[t \mapsto R_\sigma], \forall t'.\tau')$. This completes the proof.

□

As in the STLC, our proof will require that the relation is closed under converse evaluation:

Lemma 2. Assume that e and e' are closed expressions. Assume that τ is a type. Assume that $e \longmapsto e'$. Let δ be a function from some set of type variables to closed types. Let η be a function from some set of type variables to the set of unary relations $\mathcal{R}_\sigma$ over expressions of the closed type σ that are closed under converse evaluation. Assume that $e' \in \mathcal{CHT}(\delta, \eta, \tau)$. Then $e \in \mathcal{CHT}(\delta, \eta, \tau)$.

Proof. Assume the hypothesis in the statement of the proposition. The proof proceeds by induction on τ.

Case $(\tau = t)$ Since $\eta(t)$ is closed under converse evaluation, then $e' \in \eta(t)$ implies that $e \in \eta(t)$, so $e \in \mathcal{CHT}(\delta, \eta, \tau)$

Case $(\tau = \tau_1 \longrightarrow \tau_2)$ Since $e' \in \mathcal{CHT}(\delta, \eta, \tau)$, then there exists a value e'' such that $e' \longmapsto^* e''$. Also, $e \longmapsto e'$ by assumption, hence $e \longmapsto^* e''$ with e'' a value. Now assume that $e_1 \in \mathcal{CHT}(\delta, \eta, \tau_1)$. Since $e \longmapsto e'$, it follows that $(e\ e_1) \longmapsto (e'\ e_1)$. Since $e' \in \mathcal{CHT}(\delta, \eta, \tau_1 \longrightarrow \tau_2)$, then $(e'\ e_1) \in \mathcal{CHT}(\delta, \eta, \tau_2)$, hence by the induction hypothesis, $(e\ e_1) \in \mathcal{HT}(\delta, \eta, \tau_2)$, hence $e \in \mathcal{CHT}(\delta, \eta, \tau)$.

Case $(\tau = \forall t.\tau')$ $(\Leftarrow)$ Since $e' \in \mathcal{CHT}(\delta, \eta, \tau)$, then there exists a value e'' such that $e' \longmapsto^* e''$. Also $e \longmapsto e'$ by assumption, hence $e \longmapsto^* e''$ with e'' a value. Let σ be a closed type and $\mathcal{R}_\sigma$ a unary relation over expressions of type σ that is closed under converse evaluation. We want to show that

$$e\ [\sigma] \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau'), \tag{1}$$

Since $e' \in \mathcal{CHT}(\delta, \eta, \tau)$, then

$$e'\ [\sigma] \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau')$$

By the Step-App rule, $(e\ [\sigma]) \longmapsto (e'\ [\sigma])$, hence by the induction hypothesis, Eq. (1) holds. Thus $e \in \mathcal{CHT}(\delta, \eta, \tau)$. This completes the proof.

□

4.4 Completing the Proof

We are now in a position to extend our relation to open types and complete the proof.

Notation 2. Let δ be a function from a finite set $\{t_1, \ldots, t_n\}$ of type variables to closed types. Let τ be an (arbitary) type. Let e be an (arbitary) expression. Then

$$\hat{\delta}(t) = t[(t_1, \ldots, t_n) := (\delta(t_1), \ldots, \delta(t_n))]$$

and

$$\hat{\hat{\delta}}(e) = e[(t_1, \ldots, t_n) := (\delta(t_1), \ldots, \delta(t_n))]$$

Definition 4. Let Δ be a set of type variables, Γ a context and τ a type. We say that $e \in \mathcal{HT}(\Delta, \Gamma, \tau)$ if and only if the following condition holds:

- if δ is a function from type variables to closed types satisfying $\Delta \subseteq \mathit{dom}(\delta)$, and
- η is a function from type variables to the set of unary relations $\mathcal{R}_\sigma$ over expressions of the closed type σ that are closed under converse evaluation, with η satisfying $\Delta \subseteq \mathit{dom}(\eta)$, and
- γ is a function from expression variables to closed expressions such that if $(x : \tau') \in \Gamma$, then $x \in \mathit{dom}(\gamma)$, $\gamma(x) : \hat{\delta}(\tau')$ and $\gamma(x) \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau'))$,

then $\hat{\gamma}(\hat{\hat{\delta}}(e)) \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau))$

Proposition 7. Assume that $(\Delta, \Gamma) \vdash e : \tau$. Then $e \in \mathcal{HT}(\Delta, \Gamma, \tau)$

Proof. Assume that $(\Delta, \Gamma) \vdash e$. Assume that

- δ is a function from type variables to closed types satisfying $\Delta \subseteq dom(\delta)$,
- η is a function from type variables to the set of unary relations $\mathcal{R}_\sigma$ over expressions of the closed type σ that are closed under converse evaluation, with η satisfying $\Delta \subseteq dom(\eta)$, and
- γ is a function from expression variables to closed types such that if $(x : \tau') \in \Gamma$, then $x \in dom(\gamma)$, $\gamma(x) : \hat{\delta}(\tau')$ and $\gamma(x) \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau'))$

We want to show that $\hat{\gamma}(\hat{\delta}(e)) \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau))$. The proof proceeds by induction over the derivation $(\Delta, \Gamma) \vdash e$.

Case Typ-Var Then $e = x$, so $\hat{\gamma}(\hat{\delta}(e)) = \gamma(x)$ with $\gamma(x) \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau))$ by assumption, hence $e \in \mathcal{HT}(\Delta, \Gamma, \tau)$

Case Typ-lam Then $e = \lambda(x : \tau_1).e'$, $\tau = \tau_1 \longrightarrow \tau_2$, with $\Delta \vdash \tau_1$ **WF** and $(\Delta, (\Gamma, x : \tau_1)) \vdash e' : \tau_2$. We may assume without loss of generality that $x \notin dom(\gamma)$ since we may rename x to fulfil this condition. Then $\hat{\gamma}(\hat{\delta}(e)) = \lambda(x : \tau_1.\hat{\gamma}(\hat{\delta}(e')))$ which is already a value. Now assume that
$e_1 \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau_1))$. Let $\theta = \gamma \otimes [x \mapsto e_1]$. Observe that

$$\begin{aligned}(\lambda(x : \tau_1).\hat{\gamma}(\hat{\delta}(e')))\ e_1 &\longmapsto (\lambda(x : \tau_1).\hat{\gamma}(\hat{\delta}(e')))\ e_1 \\ &= \hat{\gamma}(\hat{\delta}(e'))[x := e_1] = \hat{\theta}(\hat{\delta}(e'))\end{aligned}$$

which is in $\mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau_2))$ since $e' \in \mathcal{HT}(\Delta, (\Gamma, x : \tau_1), \tau_2)$ by the induction hypothesis. By Lemma 2,
$(\lambda(x : \tau_1).\hat{\gamma}(\hat{\delta}(e')))\ e_1 \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau_2))$, thus $e \in \mathcal{HT}(\Delta, \Gamma, \tau)$.

Case Typ-app Then $e = (e_1\ e_2)$ with $(\Delta, \Gamma) \vdash e_1 : \tau_1 \longrightarrow \tau_2$ and $(\Delta, \Gamma) \vdash e_2 : \tau_2$. By the induction hypothesis, $e_1 \in \mathcal{HT}(\Delta, \Gamma, \tau_1 \longrightarrow \tau_2)$ and $e_2 \in \mathcal{HT}(\Delta, \Gamma, \tau_2)$, thus $\hat{\gamma}(\hat{\delta}(e_1)) \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau_1))$ and $\hat{\gamma}(\hat{\delta}(e_2)) \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau_2))$. Consequently, $\hat{\gamma}(\hat{\delta}((e_1\ e_2))) = (\hat{\gamma}(\hat{\delta}(e_1)))\ (\hat{\gamma}(\hat{\delta}(e_2))) \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau_2))$, thus $e \in \mathcal{HT}(\Delta, \Gamma, \tau)$.

Case Type-Lam Then $e = \Lambda t.e'$ and $\tau = \forall t.\tau'$ with $((\Delta \cup \{t\}), \Gamma) \vdash e' : \tau'$. Assume without loss of generality that $t \notin dom(\delta)$ since we can rename t to fulfil this condition. Observe that $\hat{\gamma}(\hat{\delta}(\Lambda t.e')) = \Lambda t.\hat{\gamma}(\hat{\delta}(e'))$. Now let σ be a closed type. Let $\mathcal{R}_\sigma$ be a unary relation over expressions of type σ that is closed under converse evaluation. We want to show that

$$\Lambda t.\hat{\gamma}(\hat{\delta}(e'))\ [\sigma] \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tag{2}$$

Observe that $\Lambda t.\hat{\gamma}(\hat{\delta}(e'))\ [\sigma] \longmapsto \hat{\gamma}(\hat{\delta}(e'))[t := \sigma]$. Now let $\theta = \delta \otimes [t \mapsto \sigma]$. By the induction hypothesis, $e' \in \mathcal{HT} \in (\Delta \cup \{t\}, \Gamma, \tau')$, thus

$$\hat{\gamma}(\hat{\theta}(e')) \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau')$$

but $\hat{\gamma}(\hat{\theta}(e')) = \hat{\gamma}(\hat{\delta}(e'))[t := \sigma]$, thus

$$\hat{\gamma}(\hat{\delta}(e'))[t := \sigma] \in \mathcal{CHT}(\delta \otimes [t \mapsto \sigma], \eta \otimes [t \mapsto \mathcal{R}_\sigma], \tau')$$

It follows by Lemma 2 that Eq. (2) holds. Therefore, $e \in \mathcal{HT}(\Delta, \Gamma, \tau)$.

Case Type-App Then $e = e'\ [\tau']$, $\tau = \tau''[t := \tau']$ with $(\Delta, \Gamma) \vdash e' : \forall t.\tau''$ and $\Delta \vdash \tau'$ **WF**. Observe that

$$\hat{\gamma}(\hat{\delta}(e'\ [\tau'])) = (\hat{\gamma}(\hat{\delta}(e')))\ [\hat{\delta}(\tau')] \tag{3}$$

and

$$\hat{\delta}(\tau''[t := \tau']) = (\hat{\delta}(\tau''))[t := \hat{\delta}(\tau')]$$

We want to show that

$$\hat{\gamma}(\hat{\delta}(e'\ [\tau'])) \in \mathcal{CHT}(\delta, \eta, (\hat{\delta}(\tau''))[t := \hat{\delta}(\tau')]) \tag{4}$$

Now define a unary relation $\mathcal{Q}$ on expressions k of the closed type $\hat{\delta}(\tau')$ by $k \in \mathcal{Q}$ if and only if $k \in \mathcal{CHT}(\delta, \eta, \hat{\delta}(\tau'))$. By Lemma 2, $\mathcal{Q}$ is closed under converse evaluation. By the induction hypothesis, $e' \in \mathcal{HT}(\Delta, \Gamma, \forall t.\tau'')$, so $\hat{\gamma}(\hat{\delta}(e')) \in \mathcal{CHT}(\delta, \eta, \forall t.\tau'')$. Consequently,

$$\hat{\gamma}(\hat{\delta}(e'))[\hat{\delta}(\tau')] \in \mathcal{CHT}(\delta \otimes [t \mapsto \hat{\delta}(\tau')], \eta \otimes [t \mapsto \mathcal{Q}], \hat{\delta}(\tau'')),$$

so

$$\hat{\gamma}(\hat{\delta}(e'\ [\tau']))[\hat{\delta}(\tau')] \in \mathcal{CHT}(\delta \otimes [t \mapsto \hat{\delta}(\tau')], \eta \otimes [t \mapsto \mathcal{Q}], \hat{\delta}(\tau'')),$$

since $\hat{\gamma}(\hat{\delta}(e'\ [\tau'])) = (\hat{\gamma}(\hat{\delta}(e')))\ [\hat{\delta}(\tau')]$ by Eq. (3). By Lemma 1,

$$\hat{\gamma}(\hat{\delta}(e'\ [\tau']))[\hat{\delta}(\tau')] \in \mathcal{CHT}(\delta \otimes [t \mapsto \hat{\delta}(\tau')], \eta \otimes [t \mapsto \mathcal{Q}], \hat{\delta}(\tau''))$$

if and only if

$$\hat{\gamma}(\hat{\delta}(e'\ [\tau'])) \in \mathcal{CHT}(\delta, \eta, (\hat{\delta}(\tau''))[t := \hat{\delta}(\tau')])$$

Therefore, Eq. (4) holds, hence $e \in \mathcal{HT}(\Delta, \Gamma, \tau)$. This completes the proof.

□

Corollary 3. If $\vdash e : \tau$, then there exists a unique value v such that $e \longmapsto^* v$.

5 Conclusion

Logical relations are a powerful yet simple technique for proving tricky theorems. In this project we used only unary logical relations, but a binary logical relation features prominently in the proof of parametricity in System **F**. Logical relations are also used in proving such properties as type safety, non-interference in security-typed languages, compiler correctness and soundness of logics.

References

[1] Harper, Robert. Practical Foundations for Programming Languages. Cambridge: Cambridge University Press, 2013.

[2] Ahmed, Amal. Oregon Programming Languages Summer School. Northeastern University, 2013.