

Assignment 3

Canonicity for System T

15-814: Types and Programming Languages
Jan Hoffmann & C.B. Aberlé

Due Tuesday, September 24, 2024
75 pts

In this assignment, you are given a proof of a key property of System T – Canonicity – which makes use of the fundamental technique of *logical relations*. Some of the sub-proofs of key lemmas & inductive cases contained in this proof have been omitted, and you are tasked with filling them in. The proof has been written as a gentle introduction to the main ideas of logical relations; read the proof carefully to familiarize yourself with these ideas, and then write out the missing steps. For your convenience, these missing pieces are listed as tasks at the end of this handout.

1 A Primer on Logical Relations

A desirable property of a programming language such as System T is that every closed program in that language should compute to a well-defined value of the corresponding type. Given a type τ , the property that every closed expression $e : \tau$ evaluates to a value (i.e. canonical form) of type τ is known as *canonicity* for the type τ . If a given term $e : \tau$ has this property, we say that e *satisfies canonicity*. Here, we will consider the problem of proving the following:

Theorem 1 (Canonicity for nat) *For all closed programs $e : \text{nat}$ in System T, there exists $n \in \mathbb{N}$ such that $e \Downarrow \bar{n}$.*

As we have already seen in lecture, a naïve attempt at proving the related property of normalization by induction on typing derivations for System T fails for the case of function application, and this same difficulty arises when proving Canonicity. What is needed is some systematic way of strengthening the induction hypothesis so as to force this case to go through as well. For this purpose, we now introduce a central concept of PL Theory: *logical relations*.

N.B. In this assignment, we work with the call-by-name dynamics of T as given in the lecture notes. This simplifies the proof slightly but the technique also applies to the call-by-value dynamics.

Recall the following definitions.

Numerals For $n \in \mathbb{N}$, \bar{n} is defined as the n -fold composition of s applied to z .

Evaluation Relation We write $e \Downarrow v$ if $e \mapsto^* v$ and $v \text{ val}$.

2 Logical Relations – Gluing Syntax to Semantics

Consider the application rule T_{ap} for System T

$$\frac{\Gamma \vdash e_1 : \sigma \rightarrow \tau \quad \Gamma \vdash e_2 : \sigma}{\Gamma \vdash e_1(e_2) : \tau}$$

Suppose we are attempting to argue by induction on derivations that a closed expression $e_1(e_2) : \text{nat}$ satisfies canonicity, for some closed programs $e_1 : \text{nat} \rightarrow \text{nat}$ and $e_2 : \text{nat}$. As we have seen, it is not enough for both e_1 and e_2 to satisfy canonicity on their own; however, if e_1 additionally had the property that $e_1(e')$ satisfied canonicity for any expression $e' : \text{nat}$ satisfying canonicity, the result would follow immediately. What is needed, then, is to somehow *relativize* the induction hypothesis to the types occurring in a derivation so as to allow for this stronger hypothesis to occur in such a proof.

Note that whether or not a given term $e : \text{nat}$ satisfies canonicity can instead be expressed as a *relation* between e and some natural number n , namely $e \Downarrow \bar{n}$. Upon reflection, it is clear also that the strengthened induction hypothesis given above for a term $e : \text{nat} \rightarrow \text{nat}$ can also be expressed as a relation between e and some mathematical object – this time a function $f : \mathbb{N} \rightarrow \mathbb{N}$, with the relation being as follows:

- for any closed expression $e' : \text{nat}$ and $n \in \mathbb{N}$ such that $e \Downarrow \bar{n}$, we have $e(e') \Downarrow \overline{f(n)}$.

These relations effectively allow us to simulate the dynamics of (syntactic) expressions by linking (or *gluing*) them to corresponding (semantic) mathematical objects, and derive key properties of the former from the latter.

To make this construction precise, we first define, for each System T type τ , a corresponding set $|\tau|$ of mathematical objects, by induction on the structure of τ :

- $|\text{nat}| = \mathbb{N}$
- $|\sigma \rightarrow \tau|$ is the set of functions from $|\sigma|$ to $|\tau|$.

We then define, for each type τ , a corresponding relation $\Vdash_\tau \subseteq \{(e, v) \mid e : \tau \text{ and } v \in |\tau|\}$, again by induction on τ :

- $e \Vdash_{\text{nat}} n \iff e \Downarrow \bar{n}$
- $e \Vdash_{\sigma \rightarrow \tau} f \iff \forall e' : \sigma, v \in |\sigma| (e' \Vdash_\sigma v \implies e(e') \Vdash_\tau f(v))$.

When $e \Vdash_\tau v$ for $e : \tau$ and $v \in |\tau|$, we say that e *realizes* – or *is a realizer for* – v (c.f. the definition of realizers for the λ -calculus).

A key lemma regarding \Vdash_τ as defined above, which one generally expects to hold for logical relations of this sort, is the following:

Lemma 2 (Head Expansion) *If $e : \tau$ and $e' : \tau$ with $v \in |\tau|$ such that $e \mapsto e'$ and $e' \Vdash_\tau v$, then $e \Vdash_\tau v$.*

Proof: See Task 1 at end of handout. □

N.B. You may use the Head Expansion lemma freely in all of your subsequent proofs.

3 The Fundamental Theorem

The canonicity of nat would straightforwardly follow if it were the case that, for every closed expression $e : \tau$ there exists some $|e| \in |\tau|$ such that $e \Vdash_\tau |e|$. This property is almost, but not quite, what is commonly called the *Fundamental Theorem of the Logical Relation* (FTLR). As stated, this version of the fundamental theorem is still not amenable to a proof by induction on derivations, since it applies only to closed terms, and the presence in System T of rules such as T_{abs}

$$\frac{\Gamma, x : \sigma \vdash e : \tau}{\Gamma \vdash \text{lam}\{x.e\} : \sigma \rightarrow \tau}$$

requires us to consider *open terms* as well. However, there is an easy fix for this, which is to simply treat open terms as functions on closed terms. For this purpose, we extend the definitions of $|\tau|$ and \Vdash_τ above to apply to contexts Γ as follows:

- For $\Gamma = x_1 : \tau_1, \dots, x_n : \tau_n$, define $|\Gamma| = \{(v_1, \dots, v_n) \mid v_1 \in |\tau_1|, \dots, v_n \in |\tau_n|\}$
- For $\Gamma = x_1 : \tau_1, \dots, x_n : \tau_n$ with $e_1 : \tau_1, \dots, e_n : \tau_n$ and $v_1 \in |\tau_1|, \dots, v_n \in |\tau_n|$, define

$$(e_1, \dots, e_n) \Vdash_\Gamma (v_1, \dots, v_n) \iff e_1 \Vdash_{\tau_1} v_1 \text{ and } \dots \text{ and } e_n \Vdash_{\tau_n} v_n$$

For notational simplicity, given $\Gamma = x_1 : \tau_1, \dots, x_n : \tau_n$, we write $\gamma : \Gamma$ to mean $\gamma = (e_1, \dots, e_n)$ such that $e_1 : \tau_1, \dots, e_n : \tau_n$. Then given $\gamma = (e_1, \dots, e_n) : \Gamma$ and $\Gamma \vdash e : \tau$, write $\widehat{\gamma}(e)$ for the result of substituting each e_i for x_i in e , i.e. $[e_1/x_1, \dots, e_n/x_n](e)$.

We can now state (and prove) the proper Fundamental Theorem of Logical Relations.

Theorem 3 (FTLR) 1. for all $\Gamma \vdash e : \tau$, there exists a function $|e| : |\Gamma| \rightarrow |\tau|$
 2. such that for all $\gamma : \Gamma$ and $v_\gamma \in |\Gamma|$ with $\gamma \Vdash_\Gamma v_\gamma$, we have $\widehat{\gamma}(e) \Vdash_\tau |e|(v_\gamma)$

Proof: We proceed by induction on the derivation of $\Gamma \vdash e : \tau$.

Case: T_{var} – we have

$$\overline{\Gamma', x : \tau \vdash x : \tau}$$

1. Define $|x| : |\Gamma| \rightarrow |\tau|$ by

$$|x|(u_1, \dots, u_m, v) = v$$

for all $(u_1, \dots, u_m) \in |\Gamma'|$.

2. Then given $\gamma', e : \Gamma', x : \tau$ with $v_{\gamma'} \in |\Gamma'|$, $v_e \in |\tau|$ such that

$$\gamma', e \Vdash_{\Gamma', x : \tau} v_{\gamma'}, v_e,$$

this in particular means that

$$\widehat{\gamma', e}(x) = e \Vdash_\tau v_e = |x|(v_{\gamma'}, v_e,)$$

Case: T_z – we have

$$\frac{}{\Gamma \vdash z : \mathbf{nat}}$$

1. Define $|z| : |\Gamma| \rightarrow \mathbb{N}$ by $|z|(v) = 0$ for all $v \in |\Gamma|$.
2. By construction, for all $\gamma : \Gamma$ with $v_\gamma \in |\Gamma|$ such that $\gamma \vdash_\Gamma v_\gamma$, we have

$$\widehat{\gamma}(z) = z \Vdash_{\mathbf{nat}} 0 = |z|(v_\gamma)$$

Case: T_s – we have

$$\frac{\Gamma \vdash e : \mathbf{nat}}{\Gamma \vdash s(e) : \mathbf{nat}}$$

By induction hypothesis, we have $|e| : |\Gamma| \rightarrow \mathbb{N}$ such that $\widehat{\gamma}(e) \Vdash_{\mathbf{nat}} |e|(v_\gamma)$ for all $\gamma : \Gamma$ and $v_\gamma \in |\Gamma|$ such that $\gamma \vdash_\Gamma v_\gamma$.

1. Define $|s(e)| : |\Gamma| \rightarrow \mathbb{N}$ by $|s(e)|(v) = 1 + |e|(v)$
2. Then for all $\gamma : \Gamma$ and $v_\gamma \in |\Gamma|$ such that $\gamma \vdash_\Gamma v_\gamma$, we have

$$\widehat{\gamma}(s(e)) = s(\widehat{\gamma}(e)) \Vdash_{\mathbf{nat}} 1 + |e|(v_\gamma) = |s(e)|(v_\gamma)$$

Case: T_{rec} – we have

$$\frac{\Gamma \vdash e : \mathbf{nat} \quad \Gamma \vdash e_0 : \tau \quad \Gamma, x : \mathbf{nat}, y : \tau \vdash e_1 : \tau}{\Gamma \vdash \mathbf{rec}\{e_0; x.y.e_1\}(e) : \tau}$$

By induction hypothesis we have functions $|e| : |\Gamma| \rightarrow \mathbb{N}$, $|e_0| : |\Gamma| \rightarrow |\tau|$, $|e_1| : |\Gamma| \times \mathbb{N} \times |\tau| \rightarrow |\tau|$ such that

$$\widehat{\gamma}(e) \Vdash_{\mathbf{nat}} |e|(v_\gamma) \quad \text{and} \quad \widehat{\gamma}(e_0) \Vdash_\tau |e_0|(v_\gamma) \quad \text{and} \quad \widehat{\gamma, e', e''} \Vdash_\tau |e_1|(v_\gamma, v_{e'}, v_{e''})$$

for all $\gamma : \Gamma$, $e' : \mathbf{nat}$, $e'' : \tau$ with $v_\gamma \in |\Gamma|$, $v_{e'} \in \mathbb{N}$, $v_{e''} \in |\tau|$ such that $\gamma \vdash_\Gamma v_\gamma$ and $e' \Vdash_{\mathbf{nat}} v_{e'}$ and $e'' \Vdash_\tau v_{e''}$.

Define an auxiliary function $g' : \mathbb{N} \times |\Gamma| \rightarrow |\tau|$ as follows:

$$\begin{aligned} g'(0, v) &= |e_0|(v) \\ g'(n+1, v) &= |e_1|(v, n+1, g'(n, v)) \end{aligned}$$

Then

1. Define $|\mathbf{rec}\{e_0; x.y.e_1\}(e)| : |\Gamma| \rightarrow |\tau|$ by

$$|\mathbf{rec}\{e_0; x.y.e_1\}(e)|(v) = g'(|e|(v), v)$$

2. Then for all $\gamma : \Gamma$ with $v_\gamma \in |\Gamma|$ such that $\gamma \vdash_\Gamma v_\gamma$, we have

$$\widehat{\gamma}(\mathbf{rec}\{e_0; x.y.e_1\}(e)) = \mathbf{rec}\{\widehat{\gamma}(e_0); x.y.\widehat{\gamma}(e_1)\}(\widehat{\gamma}(e))$$

By induction on $|e|(v_\gamma)$, we will show that $\mathbf{rec}\{\widehat{\gamma}(e_0); x.y.\widehat{\gamma}(e_1)\}(\widehat{\gamma}(e)) \Vdash_\tau g'(|e|(v_\gamma), v_\gamma)$

(a) When $|e|(v_\gamma) = 0$ we have that $\hat{\gamma}(e) \mapsto^* \bar{0}$, and therefore

$$\text{rec}\{\hat{\gamma}(e_0); x.y.\hat{\gamma}(e_1)\}(\hat{\gamma}(e)) \mapsto^* \hat{\gamma}(e_0) \Vdash_\tau |e_0|(v_\gamma) = g'(0, |\gamma|)$$

and so the result follows by head expansion.

(b) If $|e|(v_\gamma) = 1 + n$ for some n such that $\text{rec}\{\hat{\gamma}(e_0); x.y.\hat{\gamma}(e_1)\}(\bar{n}) \Vdash_\tau g'(n, v_\gamma)$ it follows that

$$\begin{aligned} & \text{rec}\{\hat{\gamma}(e_0); x.y.\hat{\gamma}(e_1)\}(\hat{\gamma}(e)) \\ \mapsto^* & \text{rec}\{\hat{\gamma}(e_0); x.y.\hat{\gamma}(e_1)\}(\bar{1+n}) \\ \mapsto & ([\bar{n+1}/x] \text{rec}\{\hat{\gamma}(e_0); x.y.\hat{\gamma}(e_1)\}(\bar{n})[y](\hat{\gamma}(e_1))) \\ \Vdash_\tau & |e_1|(v_\gamma, n+1, g'(n, v_\gamma)) \\ = & g'(n+1, v_\gamma) \end{aligned}$$

so the result follows by Head Expansion.

Case: T_{abs} – See **Task 2** at end of handout.

Case: T_{ap} – See **Task 3** at end of handout.

This completes the proof of the fundamental theorem. □

Corollary 4 (Canonicity for nat) For $e : \text{nat}$ there exists $n \in \mathbb{N}$ such that $e \Downarrow \bar{n}$.

Proof: See **Task 4** at end of handout. □

Task 1 (20pts) Prove Lemma 2 (Head Expansion).

Task 2 (25 pts) Complete the proof for **Case:** T_{lam} in the proof of Theorem 3 (FTLR).

Task 3 (25 pts) Complete the proof for **Case:** T_{ap} in the proof of Theorem 3 (FTLR).

Task 4 (5 pts) Prove Corollary 4 (Canonicity for nat), using FTLR.

Hint: You may use the following proposition whenever you need it on this homework.

Proposition 1 (Compatibility) Multistep transition is compatible with evaluation contexts (i.e. rules E_{ap1} , E_s , E_{rec1}).

This means that the following rules are admissible for multistep transition.

$$\frac{e \mapsto^* e'}{e(e_1) \mapsto^* e'(e_1)} E_{ap1}^* \quad \frac{e \mapsto^* e'}{s(e) \mapsto^* s(e')} E_s^* \quad \frac{e \mapsto^* e'}{\text{rec}(e, e_0, x.y.e_1) \mapsto^* \text{rec}(e', e_0, x.y.e_1)} E_{rec1}^*$$