Logical Relations

Natalie Ravenhill

July 28, 2016

1 Adequacy

Now that we have proved Correctness, we have proved half of Adequacy, as it is the following theorem:

Theorem 1. *If* $\vdash e : Nat$ *(ie. e is a closed term of type* Nat*) and* $[\![e]\!] = n$ *then* $[\![e]\!] = n \Leftrightarrow e \mapsto^* \underline{n}$

where $[\![e]\!] = [\![\vdash e : Nat]\!]$ and $\underline{n}$ represents the numeral n, instead of the natural number n.

1.1 $\Leftarrow$

The right to left direction is a corollary of the correctness proof:

Corollary 1. *If* $\vdash e : Nat$ *and* $[\![e]\!] = n$ *then* $e \mapsto^* \underline{n} \Rightarrow [\![e]\!] = n$

Proof. Rewrite $e \mapsto^* \underline{n}$ as $e_0 \mapsto \cdots \mapsto e_n \mapsto \underline{n}$, for any $n \geq 0$. Applying correctness to these evaluations gives us $[\![e_0]\!] = \cdots = [\![e_n]\!] = [\![\underline{n}]\!]$. Therefore we have $[\![e_n]\!] = [\![\underline{n}]\!]$.

Now we need to prove $\forall n \in \mathbb{N}.[\![n]\!] = n$, which we prove by induction on n:

If $\underline{n} = zero$, then by the denotational semantics for $zero$ we have $[\![e_n]\!] = 0$.

If $\underline{n} = s(v)$, then in the rule, we have $[\![v]\!]$, which equals v by the inductive hypothesis. Therefore, we use the case in the successor rule for a number and get $v + 1$.

Therefore $\forall n \in \mathbb{N}.[\![n]\!] = n$, so $[\![e]\!] = [\![n]\!] = n$.

□

1.2 $\Rightarrow$

For the other direction, we want to show that if $[\![\cdot \vdash e : Nat]\!] = n$ then $e \mapsto^* n$. We cannot prove this by induction, so we need to define a logical predicate to use in the proof, inductively on types, which is the following:

$$Good_{Nat} = \{e \mid \ \vdash e : Nat \ \wedge \ ([\![e]\!] = n \Rightarrow e \mapsto^* \underline{n})\}$$

$$Good_{A \to B} = \{e \mid \ \vdash e : A \to B \ \wedge \ \forall e' \in Good_A (e \ e') \in Good_B\}$$

Now we the proof is that every well typed term is *Good*, so we also defined *Good* on typing contexts:

$$Good_{Ctx}(\cdot) = \{<>\}$$

where $<>$ is the empty substitution and $\cdot$ is the empty context.

$$Good_{Ctx}(\Gamma, A) = \{(\gamma, e/x) \mid \gamma \in Good_{Ctx}(\Gamma) \ \wedge \ e \in Good_A\}$$

For example, if we have the context $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ then:

$$Good_{Ctx}(x_1 : A_1, \ldots, x_n : A_n) = \{(e_1/x_1, \ldots, e_n/x_n) \mid e_i \in Good_{A_i}\}$$

From this we know that:

- $FV([\gamma](e)) = \emptyset$, because for any expression e, $FV(e) \subseteq \Gamma$ and $[\gamma]$ substitutes all the free variables in e with expressions.
- If $\gamma \in Good_{Ctx}(\Gamma)$ then $\gamma(x_i)$ has no free variables, because every expression in γ that substitutes some x_i is in $Good_{A_i}$, so is a closed term.

Now we have this, we can prove the Fundamental Lemma, which is the following:

Lemma 1. *If $\Gamma \vdash e : A$ and $\gamma \in Good_{Ctx}(\Gamma)$, then $[\gamma](e) \in Good_A$*

$[\gamma](e)$ is the expression obtained by applying a substitution γ to an expression e. We can define it inductively in the following way:

$$[\gamma](zero) = zero$$

$$[\gamma](x) = \begin{cases} [\gamma](x) & \text{if } x \in dom(\gamma) \\ x & otherwise \end{cases}$$

This says that if x is present in γ, (there is a substitution for it), replace x with the result of the substitution in γ. Otherwise there is nothing to replace x with, so we return the variable unaltered.

$$[\gamma](s(e)) = s([\gamma](e))$$

$$[\gamma](case(e, z \mapsto e_0, s(v) \mapsto e_S)) = case([\gamma](e), z \mapsto [\gamma](e_0), s(v) \mapsto [\gamma](e_S))$$

$$[\gamma](e\ e') = ([\gamma]e)([\gamma]e')$$

$$[\gamma](\lambda x : A.e) = \lambda x : A.[\gamma]e$$

$$[\gamma](fix\ x : A.e) = fix\ x : A.[\gamma]e$$

For a non empty $\gamma = e_1/x_1, \ldots e_n/x_n$, we have:

$$[e_1/x_1, \ldots, e_n/x_n]e = [e_1/x_1]([e_2/x_2](\ldots [e_n/x_n]e)$$

2 Expansion Lemma

To prove this lemma, we need another lemma for the λ-abstraction case. This lemma is called the **Expansion Lemma**:

Lemma 2. *If* $\vdash e : A$ *and* $e \mapsto e'$ *and* $e' \in Good_A$ *then* $e \in Good_A$

Proof. By induction on types.

The base case will be when $\vdash e : Nat$. We have $e' \in Good_{Nat}$, so we have $\vdash e' : Nat$ and $[\![e']\!] = n \Rightarrow e' \mapsto^* \underline{n}$. We need to show that $e \in Good_{Nat}$. We already have $\vdash e : Nat$ as it is one of our assumptions, so we just prove $[\![e]\!] = n \Rightarrow e \mapsto^* \underline{n}$.

Assume $[\![e]\!] = n$. Correctness in the empty context is $e \mapsto e' \Rightarrow [\![e]\!] = [\![e']\!]$, so we use this to get $[\![e]\!] = [\![e']\!] = n$.

We can use $[\![e']\!] = n$ to get $e' \mapsto^* \underline{n}$ from $e' \in Good_{Nat}$. As $e \mapsto e'$ was an assumption, we now have $e \mapsto e' \ \wedge \ e' \mapsto^* \underline{n}$, so we have $e \mapsto^* \underline{n}$. Therefore $e \in Good_{Nat}$.

The inductive case will be when $\vdash e : A \to B$. We have $e \mapsto e'$ and $e' \in Good_{A \to B}$ and we need to show that $e \in Good_{A\to B}$. We already have $\vdash e : A \to B$, as it is one of our assumptions, so we just prove $\forall a \in Good_A.\ e\ a \in Good_B$.

Let $a : A$ be an expression such that $a \in Good_A$. Then we have $e'\ a \in Good_B$ from $e' \in Good_A$. We use $\vdash e : A \to B$ and $\vdash a : A$ (obtained from $a \in Good_A$) with the typing rule for function application to get $\vdash e\ a : B$. We use the congruence rule on $e \mapsto e'$ to get $e\ a \mapsto e'\ a$.

Now we can apply the inductive hypothesis on $\vdash e\ a : B$, $e\ a \mapsto e'\ a$ and $e'\ a \in Good_B$ to get $e\ a \in Good_B$

Therefore $\forall a \in Good_A . e\ a \in Good_B$, so $e \in Good_{A \to B}$

Now we have proved all of the cases, so we know the lemma holds for expressions of any type A.

□

3 Main Lemma

Now we can prove the Main Lemma:

Proof. By induction on $\Gamma \vdash e : A$

Variables $[\gamma]x = [\gamma]x$, as $x \in dom[\gamma]$, so we will always have $[\gamma]x \in Good_A$.

Zero $[\gamma](zero) = zero$, so we need to prove that $zero \in Good_{Nat}$. Therefore we first prove $\vdash zero : Nat$. As $zero$ is a constant, then we have it in any typing context, including the empty context. We must also prove $[\![zero]\!] = n \Rightarrow zero \mapsto^* \underline{n}$. 0 is the only possible value of n, as defined by the denotational semantics, so we need $zero \mapsto^* \underline{0}$. $zero$ is our representation of the numeral $\underline{0}$, so it maps to this in 0 steps. Therefore $zero \in Good_{Nat}$.

Successor $[\gamma]s(e) = s([\gamma]e)$, so we must prove $s([\gamma]e) \in Good_{Nat}$. As we have a derivation of $\Gamma \vdash s(e) : Nat$, from the typing rule we also have $\Gamma \vdash e : Nat$. Therefore we can apply the inductive hypothesis to this and $\gamma \in Good_{Ctx}(\Gamma)$ to get $[\gamma]e \in Good_{Nat}$. Now there are two things we must show:

1. $\vdash s([\gamma]e) : Nat$
   We have $\vdash [\gamma]e : Nat$ from $[\gamma]e \in Good_{Nat}$, so we use the typing rule on this to get $\vdash s([\gamma]e) : Nat$
2. $[\![s([\gamma]e)]\!] = n \Rightarrow s([\gamma]e) \mapsto^* \underline{n}$
   Assume $[\![s([\gamma]e)]\!] = n$. By the definition of $[\![s([\gamma]e)]\!]$, we must have $[\![[\gamma]e]\!] = n - 1$, because this is the only case that does not give $\perp$ as the final output. From $[\gamma]e \in Good_{Nat}$ and this we have $[\gamma]e \mapsto^* \underline{n-1}$. Using the congruence rule for successor, with this as our assumption, we

have $s([\gamma]e) \mapsto^* s(\underline{n-1})$, which is the same as the numeral $\underline{n}$. Therefore $s([\gamma]e) \mapsto^* \underline{n}$

Therefore $s([\gamma]e) \in Good_{Nat}$, so by the definition of $[\gamma]$ we have $[\gamma]s(e) \in Good_{Nat}$.

Case $[\gamma](case(e, z \mapsto e_0, s(v) \mapsto e_S)) = case([\gamma](e), z \mapsto [\gamma](e_0), s(v) \mapsto [\gamma](e_S)$, so we need to prove that $case([\gamma](e), z \mapsto [\gamma](e_0), s(v) \mapsto [\gamma](e_S)) \in Good_A$, for some type A. As we have a derivation of $\Gamma \vdash case(e, z \mapsto e_0, s(v) \mapsto e_S : A)$, from the typing rule we also have derivations for $\Gamma \vdash e : Nat$, $\Gamma \vdash e_0 : A$ and $\Gamma, v : Nat \vdash e_S : A$.

Therefore we can apply the inductive hypothesis to this and $\gamma \in Good_{Ctx}(\Gamma)$ to get $[\gamma]e \in Good_{Nat}$ and $[\gamma]e_0 \in Good_A$.

For e_S, we have $[\gamma']e_S \in Good_A$, where $\gamma' = (\gamma, e/v)$.

Now we have three cases for the proof, depending on the value of $[\gamma]e$:

1. When $[\gamma]e = zero$, the evaluation rule gives us $[\gamma]e_0$, so we must prove $[\gamma]e_0 \in Good_A$. We have the derivation tree for $\Gamma \vdash e_0 : A$, so we apply the inductive hypothesis with this and $\gamma \in Good_{Ctx}(\Gamma)$ to get $[\gamma]e_0 \in Good_A$
2. When $[\gamma]e = s(v)$, the evaluation rule gives us $[\gamma]e_S$, so we must prove $[\gamma]e_S \in Good_A$. We have the derivation tree for $\Gamma, v : Nat \vdash e_S : A$, so we apply the inductive hypothesis with this and $\gamma, e/v \in Good_{Ctx}(\Gamma, Nat)$ to get $[\gamma']e_S \in Good_A$
3. When e does not evaluate to a value, we need to show that for any expression e, its case statement is still in $Good_A$, so we need to show:

   Lemma 3. *If* $[\![e]\!] = \perp$ *and* $\vdash e : A$ *and* $e \mapsto^\infty$ *then* $e \in Good_A$

   Proof. In the *Lemmas* file. □

   By the definition of the denotational semantics, when we have $[\![[\gamma]e]\!] = \perp$, then $[\![case([\gamma]e, z \mapsto [\gamma]e_0, s(v) \mapsto [\gamma]e_S]\!] = \perp$.

   We know that our case expression is a well typed closed term by the following lemma:

   Lemma 4. *If* $\gamma \in Good_{Ctx}(\Gamma)$ *and* $\Gamma \vdash e : A$ *then* $\vdash [\gamma](e) : A$

   Proof. In the *Lemmas* file. □

   Using our original assumption for $\Gamma \vdash case(e, z \mapsto e_0, s(v) \mapsto e_S)$ to get $\vdash [\gamma]case(e, z \mapsto e_0, s(v) \mapsto e_S)$.

   We use the lemma for non termination, to get $case([\gamma]e, z \mapsto [\gamma]e_0, s(v) \mapsto [\gamma]e_S) \in Good_A$

Application $[\gamma](e_0\, e_1) = ([\gamma]e_0)([\gamma]e_1)$, so we need to prove that $([\gamma]e_0)([\gamma]e_1) \in Good_B$. As we have a derivation of $\Gamma \vdash e_0\ e_1 : B$, from the typing rule, we have derivations for $\Gamma \vdash e_0 : A \to B$ and $\Gamma \vdash e_1 : A$. Therefore we can apply the inductive hypothesis to these derivations and $\gamma \in Good_{Ctx}(\Gamma)$ to get $[\gamma]e_0 \in Good_{A\to B}$ and $[\gamma]e_1 \in Good_A$.

From $[\gamma]e_0 \in Good_{A\to B}$, we know that $\forall e' \in Good_A.([\gamma]e_0\ e') \in Good_B$. As $[\gamma]e_1 \in Good_A$, then $([\gamma]e_0)([\gamma]e_1) \in Good_B$.

λ-Abstraction $[\gamma](\lambda x : A.\ e) = \lambda x : A.\ [\gamma]e$ so we must prove $\lambda x : A.\ [\gamma]e \in Good_{A\to B}$. As we have a derivation of $\Gamma \vdash \lambda x : A.\ e : A \to B$, from the typing rule, we have $\Gamma, x : A \vdash e : B$. Let $\gamma' = (\gamma, e'/x)$ for some expression $e' \in Good_A$ (so we have $\gamma' \in Good_{Ctx}(\Gamma, A)$). Then using the induction hypothesis we have $[\gamma']e \in Good_B$. Now there are two things we need to show:

1. $\vdash \lambda x : A.\ [\gamma]e : A \to B$
   We can prove this using the following lemma:

   Lemma 5. *If* $\gamma \in Good_{Ctx}(\Gamma)$ *and* $\Gamma \vdash e : A$ *then* $\vdash [\gamma](e) : A$

   Proof. In the *Lemmas* file. □

   As we have $\Gamma \vdash \lambda x : A.\ e : A \to B$, then by using the lemma, we have $\vdash [\gamma]\lambda x : A.e : A = \vdash \lambda x : A.\ [\gamma]e : A \to B$

2. $\forall e' \in Good_A.\ (\lambda x : A.\ [\gamma]e)\ e' \in Good_B$
   Let $e' : A$ be an expression such that $e' \in Good_A$. Using the evaluation rule for λ abstraction we have $(\lambda x : A.\ [\gamma]e)\ e' \mapsto [e'/x][\gamma]e$, which can be simplified to $[\gamma, e'/x]e = [\gamma']e$. We have $\vdash e' : A$ from $e' \in Good_A$ and $\vdash \lambda x : A.\ [\gamma]e : A \to B$ from the previous case, so we use the typing rule for function application to get $\vdash (\lambda x : A.\ [\gamma]e)\ e' : B$.

   As $[\gamma']e \in Good_B$, we can use the Expansion Lemma to get $(\lambda x : A.\ [\gamma]e)\ e' \in Good_B$. Therefore we know $\forall e' \in Good_A.\ (\lambda x : A.\ [\gamma]e)\ e' \in Good_B$.

Fixpoint $[\gamma](fix\ x : A.e) = fix\ x : A.[\gamma]e$ so we must prove $fix\ x : A.[\gamma]e \in Good_A$. As we have a derivation of $\Gamma \vdash fix\ x : A.\ e : A$, from the typing rule, we have $\Gamma, x : A \vdash e : A$. Let $\gamma' = (\gamma, e'/x)$ for some expression $e' \in Good_A$ (so we have $\gamma' \in Good_{Ctx}(\Gamma, A)$). Then using the induction hypothesis we have $[\gamma']e \in Good_A$. Let $e' = fix\ x : A\ e$. Then $[\gamma']e = [\gamma, fix\ x : A\ e/x]e = [\gamma][fix\ x : A\ e/x]e$. Then as $[\gamma](fix\ x : A.e) \mapsto [\gamma]([fix\ x : A\ e/x]e)$ and $[\gamma]([fix\ x : A\ e/x]e) \in Good_A$, we use the Expansion Lemma with $\vdash fix\ x : A\ e : A$ to get $[\gamma](fix\ x : A.e) \in Good_A$.

□

We can now prove the second direction of Adequacy as a corollary of the Main Lemma:

Corollary 2. *If* $\vdash e : Nat$ *and* $[\![e]\!] = n$ *then* $[\![e]\!] = n \Rightarrow e \mapsto^* \underline{n}$

Proof. Using the Main Lemma, as we have $\vdash e : Nat$ and the empty substitution will be in $Good_{Ctx}(\cdot)$, we will have $[<>]e \in Good_{Nat}$. $[<>]e = e$, so we have $e \in Good_{Nat}$.

Expanding this definition gives us $[\![e]\!] = n \Rightarrow e \mapsto^* \underline{n}$, which is what we wanted to prove. $\square$