# Type Safety via Logical Relations

Andrew Tolmach

May 16, 2024

These notes are based heavily on Amal Ahmed's lectures at the Oregon Programming Languages Summer School in 2015 (`www.cs.uoregon.edu/research/summerschool/summer15/curriculum.html`) and Lau Skorstengaard's notes based on those lectures (*An Introduction to Logical Relations: Proving Program Properties Using Logical Relations*, `https://www.cs.uoregon.edu/research/summerschool/summer16/notes/AhmedLR.pdf`).

We can use the machinery of unary logical relations (or logical "predicates") to give an alternative proof of type safety that doesn't rely on the notions of progress and preservation. The basic idea is to define a predicate that captures the property of safe evaluation (i.e. evaluation without getting "stuck") and show that every (syntactically) well-typed term has this property.

In these notes, we will show type safety for the simply-typed lambda calculus extended with booleans as in Pierce Ch. 9. The proof can be expanded to cover further extensions such as pairs; this is left as an exercise.

Recall that a term $\mathtt{t}$ is in *normal form*, written $NF(\mathtt{t})$, iff there is no $\mathtt{t}'$ such that $\mathtt{t} \to \mathtt{t}'$. We write $VAL(\mathtt{t})$ iff $\mathtt{t}$ is a (syntactic) value. Finally, we define

$$SAFE(\mathtt{t}) \triangleq \forall \mathtt{t}'.\mathtt{t} \to^* \mathtt{t}' \wedge NF(\mathtt{t}') \implies VAL(\mathtt{t}')$$

to capture terms whose evaluation will never lead to a stuck state.

Then we wish to prove

Theorem (Type Safety): If $\vdash \mathtt{t} : \mathtt{T}$ then $SAFE(\mathtt{t})$.

We'll prove this by defining a type-indexed predicate on terms that holds when the term is *semantically well-typed* (in a sense defined below), and then showing that (A) syntactically well-typed terms are semantically well-typed, and (B) semantically well-typed terms are safe. This is completely analogous to the structure of the normalization proof in Ch. 12.

Naturally, the challenge is to come up with a workable definition of semantic well-typedness for which both (A) and (B) can be shown. Again, as with normalization, we will arrange things so that the proof of (B) is very easy; the effort will be in proving (A).

To characterize semantically well-typed terms, we give a *value interpretation* $\mathcal{V}[\![\mathtt{T}]\!]$ of each type $\mathtt{T}$ as a set of (closed) values that belong to that type, taking care only to include values that are "well-behaved" with respect to safety. In particular, we must make sure that functions in the value interpretation are not only themselves (trivially) values, but also don't get stuck when the function is applied. (Similarly, if we extend our language to include values of other compound types, such as pairs, we must ensure that the sub-components of the value are well-behaved.) To do this, we

also need a more general notion of a *term interpretation* $\mathcal{T}[\![\mathtt{T}]\!]$ of type $\mathtt{T}$ as a set of terms that will safely evaluate to values in the value interpretation.

We give mutually recursive definitions of these predicates as follows:

$$\begin{array}{rcl} \mathcal{V}[\![\mathtt{Bool}]\!] & \triangleq & \{\mathtt{true}, \mathtt{false}\} \\ \mathcal{V}[\![\mathtt{T}_1 \to \mathtt{T}_2]\!] & \triangleq & \{\lambda \mathtt{x}{:}\mathtt{T}_1 . \mathtt{t}_2 \mid (FV(\mathtt{t}_2) - \{\mathtt{x}\}) = \emptyset \wedge \forall \mathtt{v} \in \mathcal{V}[\![\mathtt{T}_1]\!] . [\mathtt{x} \mapsto \mathtt{v}]\mathtt{t}_2 \in \mathcal{T}[\![\mathtt{T}_2]\!]\} \end{array}$$

$$\mathcal{T}[\![\mathtt{T}]\!] \triangleq \{\mathtt{t} \mid FV(\mathtt{t}) = \emptyset \wedge \forall \mathtt{t}'. \mathtt{t} \to^* \mathtt{t}' \wedge NF(\mathtt{t}') \Longrightarrow \mathtt{t}' \in \mathcal{V}[\![\mathtt{T}]\!]\}$$

Here $FV(\mathtt{t})$ is the set of free variables of $\mathtt{t}$. Observe that these predicates are structurally well-founded, because the recursive mentions are at smaller types.

Here are some simple but useful consequences of these definitions:

- $\mathtt{t} \in \mathcal{V}[\![\mathtt{T}]\!] \Longrightarrow VAL(\mathtt{t})$.
- $\mathtt{t} \in \mathcal{V}[\![\mathtt{T}]\!] \Longrightarrow \mathtt{t} \in \mathcal{T}[\![\mathtt{T}]\!]$, i.e, $\mathcal{V}[\![\mathtt{T}]\!] \subseteq \mathcal{T}[\![\mathtt{T}]\!]$.
- $\mathtt{t} \in \mathcal{T}[\![\mathtt{T}]\!] \wedge NF(\mathtt{t}) \Longrightarrow \mathtt{t} \in \mathcal{V}[\![\mathtt{T}]\!]$.
- If $\mathtt{t} \in \mathcal{T}[\![\mathtt{T}]\!]$ then $\mathtt{t}$ is closed.

Notice that we do *not* require values or expressions in the interpretation of type $\mathtt{T}$ to be (syntactically) well-typed, only that they be closed. This is unusual in logical relations proofs, but it makes sense here since we are trying to develop a non-syntactic notion of type safety.

Now we would like to proceed by proving something like:

$\emptyset \vdash \mathtt{t} : \mathtt{T} \Longrightarrow \mathtt{t} \in \mathcal{T}[\![\mathtt{T}]\!]$

By expanding out the definition of $\mathtt{t} \in \mathcal{T}[\![\mathtt{T}]\!]$ and comparing it to that of $SAFE(\mathtt{t})$, we can see that this is essentially a strengthening of the original Type Safety theorem statement, with $VAL(\mathtt{t}')$ replaced by the (stronger) $\mathtt{t}' \in \mathcal{V}[\![\mathtt{T}]\!]$.

Naturally, the proof should be by structural induction on the typing derivation in the hypothesis. But it is clear that we will get stuck in the abstraction case because the typing context will no longer be empty. (Again, this happened in the normalization proof too.) So we need to generalize our theorem by considering how to "close off" terms. To this end, we can define the following *context interpretation* characterizing safe closing substitutions for a context.

$$\mathcal{G}[\![\Gamma]\!] \triangleq \{\sigma : Var \to Value \mid Dom(\sigma) = Dom(\Gamma) \wedge \forall \mathtt{x} \in Dom(\Gamma), \sigma(\mathtt{x}) \in \mathcal{V}[\![\Gamma(x)]\!]\}$$

Using this, we can finally define our actual logical predicate for semantic type safety:

$$\Gamma \vDash \mathtt{t} : \mathtt{T} \triangleq \forall \sigma \in \mathcal{G}[\![\Gamma]\!] . \sigma(\mathtt{t}) \in \mathcal{T}[\![\mathtt{T}]\!]$$

and phrase the two lemmas we will actually prove

Lemma A: If $\Gamma \vdash \mathtt{t} : \mathtt{T}$ then $\Gamma \vDash \mathtt{t} : \mathtt{T}$.

In proof developments based on logical relations, this is usually called the "Fundamental Property" or "Basic Lemma."

Lemma B: If $\emptyset \vDash \mathtt{t} : \mathtt{T}$ then $SAFE(\mathtt{t})$.

Combining these lemmas (with $\Gamma = \emptyset$ in Lemma A) directly gives us type safety.

We start by proving Lemma B, which, as promised, is very easy.

*Proof.* Suppose $\emptyset \vDash \mathtt{t} : \mathtt{T}$. Then we can pick $\sigma$ in the definition of $\vDash$ to be the empty substitution, so we have $\mathtt{t} \in \mathcal{T}[\![\mathtt{T}]\!]$. Thus, $\mathtt{t} \to^* \mathtt{t}' \wedge \mathit{NF}(\mathtt{t}') \Longrightarrow \mathtt{t}' \in \mathcal{V}[\![\mathtt{T}]\!]$. Since $\mathtt{t}' \in \mathcal{V}[\![\mathtt{T}]\!]$ implies $\mathit{VAL}(\mathtt{t}')$, we have $\mathtt{t} \to^* \mathtt{t}' \wedge \mathit{NF}(\mathtt{t}') \Longrightarrow \mathit{VAL}(\mathtt{t}')$, but this is exactly the definition of $\mathit{SAFE}(\mathtt{t})$. □

Now we proceed to prove Lemma A, by induction on the structure of the deriviation of $\Gamma \vdash \mathtt{t} : \mathtt{T}$.

*Case* T-Var: $\mathtt{t} = \mathtt{x} \quad \mathtt{x} : \mathtt{T} \in \Gamma$

Choose any $\sigma \in \mathcal{G}[\![\Gamma]\!]$. We must show that $\sigma(\mathtt{x}) \in \mathcal{T}[\![\mathtt{T}]\!]$. By definition of $\mathcal{G}[\![]\!]$, we have $\sigma(\mathtt{x}) \in \mathcal{V}[\![\mathtt{T}]\!] \subseteq \mathcal{T}[\![\mathtt{T}]\!]$.

*Case* T-True: $\mathtt{t} = \mathtt{true} \quad \Gamma \vdash \mathtt{true} : \mathtt{Bool}$

For any $\sigma$, we have $\sigma(\mathtt{true}) = \mathtt{true} \in \mathcal{V}[\![\mathtt{Bool}]\!] \subseteq \mathcal{T}[\![\mathtt{Bool}]\!]$.

*Case* T-False: $\mathtt{t} = \mathtt{false} \quad \Gamma \vdash \mathtt{false} : \mathtt{Bool}$

Similar.

*Case* T-If: $\mathtt{t} = \mathtt{if}\ \mathtt{t}_1\ \mathtt{then}\ \mathtt{t}_2\ \mathtt{else}\ \mathtt{t}_3 \quad \Gamma \vdash \mathtt{t}_1 : \mathtt{Bool} \quad \Gamma \vdash \mathtt{t}_2 : \mathtt{T} \quad \Gamma \vdash \mathtt{t}_3 : \mathtt{T}$

Choose any $\sigma$ in $\mathcal{G}[\![\Gamma]\!]$. We must show $\sigma(\mathtt{t}) = \mathtt{if}\ \sigma(\mathtt{t}_1)\ \mathtt{then}\ \sigma(\mathtt{t}_2)\ \mathtt{else}\ \sigma(\mathtt{t}_3) \in \mathcal{T}[\![\mathtt{T}]\!]$.

By applying the induction hypothesis to the three sub-derivations and instantiating at $\sigma$, we get $\sigma(\mathtt{t}_1) \in \mathcal{T}[\![\mathtt{Bool}]\!]$, $\sigma(\mathtt{t}_2) \in \mathcal{T}[\![\mathtt{T}]\!]$, and $\sigma(\mathtt{t}_3) \in \mathcal{T}[\![\mathtt{T}]\!]$. In particular, this means that $\sigma(\mathtt{t}_1)$, $\sigma(\mathtt{t}_2)$, and $\sigma(\mathtt{t}_3)$ are closed.

Unfolding the definition of $\mathcal{T}[\![\mathtt{T}]\!]$, we must show

- $\mathtt{if}\ \sigma(\mathtt{t}_1)\ \mathtt{then}\ \sigma(\mathtt{t}_2)\ \mathtt{else}\ \sigma(\mathtt{t}_3)$ is closed. This follows directly from the fact that $\sigma(\mathtt{t}_1)$, $\sigma(\mathtt{t}_2)$, and $\sigma(\mathtt{t}_3)$ are closed.
- $\mathtt{if}\ \sigma(\mathtt{t}_1)\ \mathtt{then}\ \sigma(\mathtt{t}_2)\ \mathtt{else}\ \sigma(\mathtt{t}_3) \to^* \mathtt{t}' \wedge \mathit{NF}(\mathtt{t}') \Longrightarrow \mathtt{t}' \in \mathcal{V}[\![\mathtt{T}]\!]$.

  So suppose $\mathtt{if}\ \sigma(\mathtt{t}_1)\ \mathtt{then}\ \sigma(\mathtt{t}_2)\ \mathtt{else}\ \sigma(\mathtt{t}_3) \to^* \mathtt{t}'$ and $\mathit{NF}(\mathtt{t}')$. Analyzing the stepping rules, we can see that this can only happen if, by repeated uses of E-If, there is a normal form $\mathtt{t}'_1$ such that $\sigma(\mathtt{t}_1) \to^* \mathtt{t}'_1$ and

  $\mathtt{if}\ \sigma(\mathtt{t}_1)\ \mathtt{then}\ \sigma(\mathtt{t}_2)\ \mathtt{else}\ \sigma(\mathtt{t}_3) \to^* \mathtt{if}\ \mathtt{t}'_1\ \mathtt{then}\ \sigma(\mathtt{t}_2)\ \mathtt{else}\ \sigma(\mathtt{t}_3) \to^* \mathtt{t}'$

  Now, since $\sigma(\mathtt{t}_1) \in \mathcal{T}[\![\mathtt{Bool}]\!]$ and $\mathit{NF}(\mathtt{t}'_1)$, we know $\mathtt{t}'_1 \in \mathcal{V}[\![\mathtt{Bool}]\!] = \{\mathtt{true}, \mathtt{false}\}$. So suppose $\mathtt{t}'_1 = \mathtt{true}$ (the $\mathtt{false}$ case is similar). Then, by E-True,

  $\mathtt{if}\ \mathtt{t}'_1\ \mathtt{then}\ \sigma(\mathtt{t}_2)\ \mathtt{else}\ \sigma(\mathtt{t}_3) \to \sigma(\mathtt{t}_2) \to^* \mathtt{t}'$

  Since $\sigma(\mathtt{t}_2) \in \mathcal{T}[\![\mathtt{T}]\!]$ and $\mathit{NF}(\mathtt{t}')$, we have $\mathtt{t}' \in \mathcal{V}[\![\mathtt{T}]\!]$, as required.

*Case* T-Abs: $\mathtt{t} = \lambda \mathtt{x}{:}\mathtt{T}_1 . \mathtt{t}_2 \quad \mathtt{T} = \mathtt{T}_1 \to \mathtt{T}_2 \quad \Gamma, \mathtt{x} : \mathtt{T}_1 \vdash \mathtt{t}_2 : \mathtt{T}_2$

Choose any $\sigma$ in $\mathcal{G}[\![\Gamma]\!]$. We must show $\sigma(\mathtt{t}) = \lambda \mathtt{x}{:}\mathtt{T}_1 . \sigma(\mathtt{t}_2) \in \mathcal{T}[\![\mathtt{T}_1 \to \mathtt{T}_2]\!] = \mathcal{T}[\![\mathtt{T}]\!]$. Equivalently, unfolding the definition, we must show that

- $\lambda \mathtt{x}{:}\mathtt{T}_1 . \sigma(\mathtt{t}_2)$ is closed. From the sub-derivation, we know that the free variables of $\mathtt{t}_2$ are in $\mathit{Dom}(\Gamma) \cup \{\mathtt{x}\}$. Then, by the definition of $\mathcal{G}[\![]\!]$, the only possible free variable in $\sigma(\mathtt{t}_2)$ is $\mathtt{x}$, so $\lambda \mathtt{x}{:}\mathtt{T}_1 . \sigma(\mathtt{t}_2)$ is indeed closed.

- $\lambda \mathtt{x}{:}\mathtt{T}_1.\sigma(\mathtt{t}_2) \to^* \mathtt{t}' \wedge \mathit{NF}(\mathtt{t}') \Longrightarrow \mathtt{t}' \in \mathcal{V}[\![\mathtt{T}_1 \to \mathtt{T}_2]\!]$.

  Suppose $\lambda \mathtt{x}{:}\mathtt{T}_1.\sigma(\mathtt{t}_2) \to^* \mathtt{t}'$. Then, since $\lambda \mathtt{x}{:}\mathtt{T}_1.\sigma(\mathtt{t}_2)$ is already a value, we must have $\mathtt{t}' = \lambda \mathtt{x}{:}\mathtt{T}_1.\sigma(\mathtt{t}_2)$. To show that $\lambda \mathtt{x}{:}\mathtt{T}_1.\sigma(\mathtt{t}_2) \in \mathcal{V}[\![\mathtt{T}_1 \to \mathtt{T}_2]\!]$, we pick an arbititrary $\mathtt{v} \in \mathcal{V}[\![\mathtt{T}_1]\!]$ and show that $[\mathtt{x} \mapsto \mathtt{v}](\sigma(\mathtt{t}_2)) \in \mathcal{T}[\![\mathtt{T}_2]\!]$. Since $\sigma$ maps variables to closed terms, $[\mathtt{x} \mapsto \mathtt{v}](\sigma(\mathtt{t}_2)) = \sigma'(\mathtt{t}_2)$, where $\sigma' = (\sigma, \mathtt{x} \mapsto \mathtt{v})$, i.e., the substitution that extends $\sigma$ with a mapping from $\mathtt{x}$ to $\mathtt{v}$. But it is easy to see that $\sigma' \in \mathcal{G}[\![\Gamma, \mathtt{x} : \mathtt{T}_1]\!]$. So applying the induction hypothesis to the sub-derivation gives $\sigma'(\mathtt{t}_2) \in \mathcal{T}[\![\mathtt{T}_2]\!]$, as required.

*Case* T-App $\quad \mathtt{t} = \mathtt{t}_1\ \mathtt{t}_2 \quad \Gamma \vdash \mathtt{t}_1 : \mathtt{T}_2 \to \mathtt{T} \quad \Gamma \vdash \mathtt{t}_2 : \mathtt{T}_2$

Choose any $\sigma$ in $\mathcal{G}[\![\Gamma]\!]$. We must show $\sigma(\mathtt{t}_1\ \mathtt{t}_2) = \sigma(\mathtt{t}_1)\ \sigma(\mathtt{t}_2) \in \mathcal{T}[\![\mathtt{T}]\!]$. Applying the induction hypothesis on the sub-derivations and instantiating at $\sigma$ gives us $\sigma(\mathtt{t}_1) \in \mathcal{T}[\![\mathtt{T}_2 \to \mathtt{T}]\!]$ and $\sigma(\mathtt{t}_2) \in \mathcal{T}[\![\mathtt{T}_2]\!]$. In particular, this means that $\sigma(\mathtt{t}_1)$ and $\sigma(\mathtt{t}_2)$ are closed.

Unfolding the definition of $\mathcal{T}[\![\mathtt{T}]\!]$, we must show that

- $\sigma(\mathtt{t}_1)\ \sigma(\mathtt{t}_2)$ is closed. This follow immediately from the fact that $\sigma(\mathtt{t}_1)$ and $\sigma(\mathtt{t}_2)$ are closed.
- $\sigma(\mathtt{t}_1)\ \sigma(\mathtt{t}_2) \to^* \mathtt{t}' \wedge \mathit{NF}(\mathtt{t}') \Longrightarrow \mathtt{t}' \in \mathcal{V}[\![\mathtt{T}]\!]$.

  So suppose $\sigma(\mathtt{t}_1)\ \sigma(\mathtt{t}_2) \to^* \mathtt{t}'$ and $\mathit{NF}(\mathtt{t}')$. Analyzing the stepping rules, we can see that this can only happen if, by repeated use of E-App1, there is a normal form $\mathtt{t}'_1$ such that $\sigma(\mathtt{t}_1) \to^* \mathtt{t}'_1$ and

  $\sigma(\mathtt{t}_1)\ \sigma(\mathtt{t}_2) \to^* \mathtt{t}'_1\ \sigma(\mathtt{t}_2) \to^* \mathtt{t}'$.

  Now, since $\sigma(\mathtt{t}_1) \in \mathcal{T}[\![\mathtt{T}_2 \to \mathtt{T}]\!]$ and $\mathit{NF}(\mathtt{t}'_1)$, we know that $\mathtt{t}'_1 \in \mathcal{V}[\![\mathtt{T}_2 \to \mathtt{T}]\!]$, so $\mathtt{t}'_1$ has the form $\lambda \mathtt{x}{:}\mathtt{T}_2.\mathtt{t}_{11}$, where $\forall \mathtt{v} \in \mathcal{V}[\![\mathtt{T}_2]\!], [\mathtt{x} \mapsto \mathtt{v}]\mathtt{t}_{11} \in \mathcal{T}[\![\mathtt{T}]\!]$.

  So, by repeated use of E-App2, there is a normal form $\mathtt{t}'_2$ such that $\sigma(\mathtt{t}_2) \to^* \mathtt{t}'_2$ and

  $\mathtt{t}'_1\ \sigma(\mathtt{t}_2) = (\lambda \mathtt{x}{:}\mathtt{T}_2.\mathtt{t}_{11})\ \sigma(\mathtt{t}_2) \to^* (\lambda \mathtt{x}{:}\mathtt{T}_2.\mathtt{t}_{11})\ \mathtt{t}'_2 \to^* \mathtt{t}'$.

  Since $\sigma(\mathtt{t}_2) \in \mathcal{T}[\![\mathtt{T}_2]\!]$ and $\mathit{NF}(\mathtt{t}'_2)$, we know that $\mathtt{t}'_2 \in \mathcal{V}[\![\mathtt{T}_2]\!]$. Hence $\mathtt{t}'_2$ is a value, and so, by E-AppAbs,

  $(\lambda \mathtt{x}{:}\mathtt{T}_2.\mathtt{t}_{11})\ \mathtt{t}'_2 \to [\mathtt{x} \mapsto \mathtt{t}'_2]\mathtt{t}_{11} \to^* \mathtt{t}'$

  Finally, since $[\mathtt{x} \mapsto \mathtt{t}'_2]\mathtt{t}_{11} \in \mathcal{T}[\![\mathtt{T}]\!]$ and $\mathit{NF}(\mathtt{t}')$, we have $\mathtt{t}' \in \mathcal{V}[\![\mathtt{T}]\!]$, as required. □