Adequacy of PCF

Natalie Ravenhill

? August 2016

Aim of the Project

- Study the operational and denotational semantics of the programming language PCF
- Prove that these semantics are equivalent at base type, by proving a theorem called Adequacy

Adequacy

Theorem

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

Relates operational semantics to denotational semantics.

Prerequisites

- **Domain Theory**
- Logical Relations

Domains including Natural Numbers, Single Element Domain, Product of Domains, **Continuous Functions**

Fixpoint Theorem

Continuous Functions are used to model fixpoint recursion.

Theorem

Every continuous function $f : X \to X$ has a least fixpoint, which is the limit of the chain $\bot \sqsubseteq f(\bot) \sqsubseteq f^2(\bot) \sqsubseteq \ldots$

Proof.

in 2 steps:

1. Prove limit of chain is a fixpoint
2. Limit $\sqsubseteq$ any other fixpoint

Prerequisites

- Domain Theory
- **Logical Relations**

Logical Relations

Definition

An n-ary **logical relation** *is a family* $\mathcal{R} = \{R_A\}_{A \in Type}$ *of n-ary relations such that* $R_A \subseteq [\![A]\!] \times \ldots [\![A]\!]$ *(i.e. an n-tuple) for any A and*

$$R_{A \to B}(f_1, \ldots f_n) \Leftrightarrow$$

for all $(d_1, \ldots d_n) \in [\![A]\!]^n$*, if* $R_B(d_1, \ldots d_n)$ *then* $R_B(f_1(d_1), \ldots f_n(d_n))$

This is uniquely determined by $R_{\mathsf{Nat}} \subseteq [\![\mathsf{Nat}]\!] \times \cdots \times [\![\mathsf{Nat}]\!]$

PCF

Syntax of PCF:

$$
\begin{aligned}
&A ::= \mathsf{Nat} \mid A \to B \\
&e ::= \lambda x : A.e \mid e\ e \mid x \\
&\mid\ z\ \mid s(e)\ \mid\ case\ (e,\ z\ \to\ e_0\ , \\
&s(n) \to e_s) \\
&\mid \mathsf{fix}\ x : A\ .\ e
\end{aligned}
$$

Correctness

Now we can relate the Operational Semantics to the Denotational semantics with the following theorem:

Theorem

If $\Gamma \vdash e : A$ *and* $e \mapsto e'$ *and* $\gamma \in [\![\Gamma]\!]$, *then*
$[\![\Gamma \vdash e : A]\!]\gamma = [\![\Gamma' \vdash e' : A]\!]\gamma$

which says that for a well typed expression e, if it maps to another expression e', then its denotation will be equal to that of the new expression in its new context.

Proof.

by induction on $e \mapsto e'$, so the cases are on the evaluation rules. We can use the fact that $f(fix(f)) = fix(f)$ and a substitution lemma □

Substitution Lemma

Lemma

If $\Gamma \vdash e : A$ *and* $\Gamma, x : A \vdash e' : C$ *and* $\gamma \in [\![\Gamma]\!]$, *then*
$[\![\Gamma \vdash [e/x]e' : C]\!]\gamma = [\![\Gamma, x : A \vdash e' : C]\!](\gamma, [\![\Gamma \vdash e : A]\!]\gamma/x)$

Proof.

By induction on e' □

Adequacy

Adequacy is the following theorem:

Theorem

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

Correctness is one part of this. If we wanted to prove the opposite direction, we cannot do this just using induction. So we need Logical Relations for the other direction.

Logical Predicate

We defined the following logical predicate (i.e./ a unary logical relation:

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

$$\mathsf{Adeq}_{A \to B} = \{e \mid \ \vdash e : A \to B \ \wedge \ \forall e' \in \mathsf{Adeq}_A (e \ e') \in \mathsf{Adeq}_B\}$$

Logical Predicate

Now to prove adequacy, we just prove that every well typed term is in Adeq, so we also defined Adeq on typing contexts:

$$\text{Adeq}_{Ctx}(\cdot) = \{<>\}$$

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

$$\text{Adeq}_{Ctx}(\Gamma, A) = \{(\gamma, e/x) \mid \gamma \in \text{Adeq}_{Ctx}(\Gamma) \ \wedge \ e \in \text{Adeq}_A\}$$

Main Lemma

Lemma

If $\Gamma \vdash e : A$ *and* $\gamma \in \mathrm{Adeq}_{Ctx}(\Gamma)$, *then* $[\gamma](e) \in \mathrm{Adeq}_A$

Proof.

By induction on derivations of $\Gamma \vdash e : A$ $\square$

Lemmas

Expansion Lemma

If $\vdash e : A$ *and* $e \mapsto e'$ *and* $e' \in \mathrm{Adeq}_A$ *then* $e \in \mathrm{Adeq}_A$

Proof.

By induction on types. □

Non Termination Lemma

If $[\![e]\!] = \bot$ *and* $\vdash e : A$ *and* $e \mapsto^{\infty}$, *then* $e \in \mathrm{Adeq}_A$

Proof.

By induction on types. □

Lemmas

Substitution Lemma

If $\gamma \in \mathsf{Adeq}_{Ctx}(\Gamma)$ *and* $\Gamma \vdash e : A$ *then* $\vdash [\gamma](e) : A$

Proof.

By induction on Γ. $\square$

Adequacy of Case

Adequacy of Fixpoint

Other Proofs

- Plotkin - Computable elements *logical predicates*
- Streicher - *logical relation* between closed terms and denotations (on a variant of PCF where fixpoint is a constant)

Logical Relation *(Streicher)*

We define a binary logical relation R, which contains a family of relations $R_A \subseteq [\![A]\!] \times \{e \mid \vdash e : A\}$, defined by induction on types for each type as follows:

$$dR_{\mathsf{Nat}}e \Leftrightarrow \forall n \in \mathbb{N}.\ d = n \Rightarrow e \mapsto^* n$$

$$fR_{A\to B}e \Leftrightarrow \forall d \in [\![A]\!].\ \forall e' \in \{e \mid \vdash e : A\}.\ dR_A e' \Rightarrow f(d)R_B e\ e'$$

Evaluation

What Went Well:

- Learned about Domain Theory and Logical Relations
- Proved standard theorems in Semantics (Type Safety, Soundness, Adequacy, ...)

Challenges:

- Choosing a logical relation that would make Adequacy easy to prove using our semantics
- Proving the fixpoint case for our semantics using the domain theoretic fixpoint theorem

Future Work

- Adequacy of more complicated languages (e.g./ FPC)
- Different Models of PCF
- Other ways of constructing the Logical Relation
- Formalisation?