Coq による証明済み型検査器を用いた算術式インタプリタの実装

Arch B2 nem

概要　「型システム入門」[1] 8 章型付き算術式に定理証明支援家 Coq を用いて証明し,OCaml を用いてパーサーをつけることで簡易的なインタプリタを実装した.

1　経緯

以前から興味を持っていた型理論や圏論が定理証明支援系によって形式的な証明がなされていることを知り, 定理証明支援系である Coq をやってみた.

Coq は Software Foundations[2](以下 SF) の有志による日本語翻訳版 [3] を用いて学んだ. SF は 4 巻ありそれぞれ,1 巻は Coq,2 巻はプログラミング言語論の形式化,3 巻はアルゴリズムの形式化,4 巻は形式手法の解説という構成になっており,2 巻まで終わらせた. その後, 型理論の解説書である「型システム入門」を Coq を用いて形式化を始めた. その際,SF2 巻にはなかった型推論の形式化とそのインタプリタの実装を最終目標として設定し, 今学期は型付きラムダ計算の型検査器の形式化とそのインタプリタの実装を目標として設定した.

2　ツール

2.1　定理証明支援系 Coq

定理証明支援系とは, 型理論と証明論の対応であるカリー=ハワード同型対応を基にしてコンピュータ上での形式的な証明を可能とするためのシステムである.　定理証明支援系には,Agda,Lean,Idris,Isabelle,F*などといったものもある. その中で Coq は上記の SF を含め日本語による資料が他の支援系に比べて多いことから今回は Coq を選んだ.

2.2　OCaml

Coq で記述・証明した関数を他の言語に変換する Extranction という拡張機能を用いて, 式の評価を行う関数及び型付けを行う関数を OCaml に変換,OCaml で記述したレキサー及びパーサーを用いてこれらの関数の実行環境を実装した.

なお,OCaml での実装には字句解析器生成器 ocamllex, 構文解析器生成器 Menhir, ビルドシステム Dune を使用した.

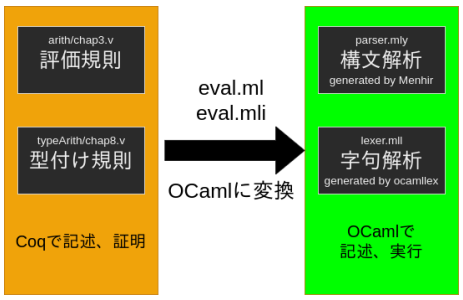

図 1　構成

3　構文

型付き算術式の構文を以下に記す.

Term ::=
- true
- false
- 0
- succ Term
- pred Term
- iszero Term
- if Term then Term else Term

Value ::=
| true
| false
| NatValue

NatValue ::=
| 0
| succ NatValue

Type ::=
| Bool
| Nat

4 評価規則

t を評価すると t' となること (1 ステップ評価) を $t \longrightarrow t'$ と書く. また, 評価の反射的推移的閉包 (多ステップ評価) を $\longrightarrow *$ で表す.

$$\frac{t1 \longrightarrow t1'}{if\ t1\ t2\ t3 \longrightarrow if\ t1'\ t2\ t3}\ (\text{E\_If})$$

$$\frac{}{if\ true\ t2\ t3 \longrightarrow t2}\ (\text{E\_IfTrue})$$

$$\frac{}{if\ false\ t2\ t3 \to t3}\ (\text{E\_IfFalse})$$

$$\frac{t1 \longrightarrow t1'}{succ\ t1 \longrightarrow succ\ t1'}\ (\text{E\_Succ})$$

$$\frac{}{pred\ 0 \longrightarrow 0}\ (\text{E\_PredZero})$$

$$\frac{\text{NatValue nv1}}{pred\ succ\ nv1 \longrightarrow nv1}\ (\text{E\_PredSucc})$$

$$\frac{t1 \longrightarrow t1'}{pred\ t1 \longrightarrow pred\ t1'}\ (\text{E\_Pred})$$

$$\frac{}{iszero\ 0 \longrightarrow true}\ (\text{E\_IsZeroZero})$$

$$\frac{\text{NatValue nv1}}{iszero\ succ\ nv1 \longrightarrow false}\ (\text{E\_IsZeroSucc})$$

$$\frac{t1 \longrightarrow t1'}{iszero\ t1 \longrightarrow iszero\ t1'}\ (\text{E\_IsZero})$$

証明済みの性質

評価規則の健全性を評価の決定性と停止性によって証明した.

定理. 評価の決定性
すべての項に対して評価は一意に定まる
$\forall$ t $\forall$ t1 $\forall$ t2 $(t \longrightarrow t1) \land (t \longrightarrow t2) \to t1 = t2$

定理. 停止性
すべての項に対して評価を進めると, ある正規形に達する.
$\forall$ t $\exists$ t' $(t \longrightarrow *t') \land (\neg \exists t''\ t' \longrightarrow t'')$

その他, 大ステップ評価を定義し, 意味論の一致なども証明した.

5 型付け規則

項 t が型 T に型付けされることを $t\colon T$ と書く.

$$\frac{}{true\colon Bool}\ (\text{T\_True})$$

$$\frac{}{false\colon Bool}\ (\text{T\_False})$$

$$\frac{t1\colon Bool\ t2\colon T\ t3\colon T}{if\ t1\ t2\ t3\colon T}\ (\text{T\_If})$$

$$\frac{}{0\colon Nat}\ (\text{T\_Zero})$$

$$\frac{t1\colon Nat}{succ\ t1\colon Nat}\ (\text{T\_Succ})$$

$$\frac{t1\colon Nat}{pred\ t1\colon Nat}\ (\text{T\_Pred})$$

$$\frac{t1\colon Nat}{iszero\ t1\colon Bool}\ (\text{T\_IsZero})$$

参考文献

証明済みの性質

型付け規則の健全性は進行定理と保存定理によって証明を行った.

定理. 進行

正しく型付けされた任意に項 t は値であるか, $t \longrightarrow t'$ となる t' が存在する.

$\forall$ t $(Value\ t) \vee (\exists t', t \longrightarrow t')$

定理. 保存

正しく型付けされた任意の項 t と $t \longrightarrow t'$ となる t' は型が一致する.

$\forall$ t $\forall$ t' $(t \longrightarrow t') \wedge (t\colon T) \rightarrow t'\colon T$

6 まとめ

今学期の目標であった型付きラムダ計算を証明, 実装することはできなかった主な要因は本旨あまり関わらない性質についての証明に時間をかけたことや Coq において証明しやすい定義が存在しその通りに記述することに不慣れであったこと, 数学の知識が不足していたことなどが挙げられる. そのため, 今後の形式化においては一定以上の時間をかけて証明できないあまり重要でない性質については飛ばすようにする. また, 今学期の経験である程度 Coq に慣れてきたので今後は進捗が生まれやすくなるだろう.

7 今後

「型システム入門」の形式化を進めて, 来学期には型推論器の形式化を終えたい.

Coq の正しさを保証している CIC と ZFC の互換性について理解をしたいので数学基礎論の知識をつける予定である.

Coq は可読性が低いので他の証明支援系を使うことも検討する. そのためにも Agda を少し触れたい.

参考文献

[1] オーム社／ Benjamin C. Pierce 著、住井 英二郎 監訳、遠藤 侑介 訳、酒井 政裕 訳、今井 敬吾 訳、黒木 裕介 訳、今井 宣洋 訳、才川隆文 訳、今井 健男 訳 https://www.ohmsha.co.jp/book/9784274069116/

[2]「型システム入門」の作者 Benjamin C. Pierce 氏などによって継続的にメンテナンスされている Coq の教科書. https://softwarefoundations.cis.upenn.edu/

[3] https://www.chiguri.info/sfja/index.html