「計算と論理」
Software Foundations
その9

五十嵐 淳
igarashi@kuis.kyoto-u.ac.jp

京都大学

December 25, 2012

本日のメニュー

Logic.v (Coq の論理)

- 量化と含意
- 連言 (「かつ」)
- 選言 (「または」)
- 偽・矛盾
- 否定 (「〜でない」)
- 特称量化 (「ある x が存在して〜」)
- 等号
- 命題としての関係
- 非形式的証明

Coq の論理

Coq の論理に関するプリミティブ

- `Inductive` 定義
- 全称量化 (`forall`)
- 含意 (`->`)

その他の論理結合子 (「かつ」「または」など) はこれらを使って全て定義可能

カリー・ハワードの同型対応

カリー・ハワードの対応 (その 2)

論理の世界	計算 (プログラム) の世界
命題	型
証拠・証明	データ・式
論理結合子	データ型 (データ構造)
ならば・全称量化	関数型
かつ	??
または	??
矛盾	??

量化と含意: 含意は量化の特殊ケース!

- **量化 = 含意の前提 (の証明) に名前をつけたもの**

```
Definition funny_prop1 := forall n,
  forall (E : beautiful n), beautiful (n+3).
```

- **funny_prop1 の証明…自然数 n，beautiful n の証拠 E を引数として，beautiful (n + 3) の証拠を返す関数**
- **結論 (返り値型) に E への言及がない ⇒ 下でも同じ**

```
Definition funny_prop1' := forall n,
  forall (_ : beautiful n), beautiful (n+3).
Definition funny_prop1'' :=
  forall n, beautiful n -> beautiful (n+3).
```

本日のメニュー

Logic.v (Coq の論理)

- 量化と含意
- 連言 (「かつ」)
  - 同値 (if and only if)
- 選言 (「または」)
- 偽・矛盾
- 否定 (「〜でない」)
- 特称量化 (「ある x が存在して〜」)
- 等号
- 命題としての関係
- 非形式的証明

連言 (conjunction)

「P かつ Q」の (帰納的な) 定義

```
Inductive and (P Q : Prop) : Prop :=
  conj : P -> Q -> (and P Q).
Notation "P /\ Q" := (and P Q) : type_scope.
```

- **命題をパラメータとする命題定義**
- **直観**: `and P Q` ($P \wedge Q$) **の証拠は** `P` **の証拠と** `Q` **の証拠から構成される**
- **逆に** $P \wedge Q$ **の証拠があれば，そこから** `P` **の証拠と** `Q` **の証拠が取り出せる**

自然演繹における導出規則

$$\frac{\Gamma \vdash P : \mathtt{Prop} \qquad \Gamma \vdash Q : \mathtt{Prop}}{\Gamma \vdash P \wedge Q : \mathtt{Prop}} \quad (\text{P-}\wedge)$$

$$\frac{\Gamma \vdash P \qquad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \quad (\wedge\text{-I})$$

$$\frac{\Gamma \vdash P \wedge Q}{\Gamma \vdash P} \quad (\wedge\text{-E1})$$

$$\frac{\Gamma \vdash P \wedge Q}{\Gamma \vdash Q} \quad (\wedge\text{-E2})$$

$$\frac{\Gamma \vdash P \wedge Q \qquad \Gamma, H1 : P, H2 : Q \vdash R}{\Gamma \vdash R} \quad (\wedge\text{-E3})$$

「かつ」の証明(1)

導出規則も単なるコンストラクタなので，今までのタクティクで証明できる

```
Check conj.

Theorem and_example :
  (beautiful 0) /\ (beautiful 3).
Proof.
  apply conj.    (* もしくは split. *)
  Case "left".  apply b_0.
  Case "right".  apply b_3.   Qed.

Print and_example.
```

「かつ」の証明(2)

仮定に「かつ」が現れる場合:

```
Theorem proj1 : forall P Q : Prop,
  P /\ Q -> P.
Proof.
  intros P Q H.
  inversion H as [HP HQ].
  apply HP.  Qed.
```

- **H は conj P Q HP HQ の形のはず** $\Longrightarrow$ `inversion` **で分解**

カリー・ハワードの対応 (その 2，続き)

論理の世界	計算 (プログラム) の世界
かつ	ペア型

```
Inductive and (P Q : Prop) : Prop :=
  conj : P -> Q -> (and P Q).

Inductive prod (X Y : Type) : Type :=
  pair : X -> Y -> prod X Y.
```

- 「P かつ Q」の証明は「P の証明」と「Q の証明」のペア

(論理的)同値

同値(if and only if)**は，両方向の含意の連言**:

```
Definition iff (P Q : Prop) :=
  (P -> Q) /\ (Q -> P).
Notation "P <-> Q" := (iff P Q)
  (at level 95, no associativity) : type_scope.
```

同値性に関する性質

```
Theorem iff_implies : forall P Q : Prop,
  (P <-> Q) -> P -> Q.
Proof.   (* 実は proj1 の特殊ケース *)


Qed.

Theorem iff_sym : forall P Q : Prop,
  (P <-> Q) -> (Q <-> P).
Proof.   (* 実は and_commut の特殊ケース *)


Qed.
```

本日のメニュー

Logic.v (Coq の論理)

- 量化と含意
- 連言 (「かつ」)
- 選言 (「または」)
  - 「かつ」「または」と andb, orb
- 偽・矛盾
- 否定 (「〜でない」)
- 特称量化 (「ある x が存在して〜」)
- 等号
- 命題としての関係
- 非形式的証明

選言 (disjunction)

「P または Q」の (帰納的な) 定義

```
Inductive or (P Q : Prop) : Prop :=
  | or_introl : P -> or P Q
  | or_intror : Q -> or P Q.

Notation "P \/ Q" := (or P Q) : type_scope.
```

- **直観—**`or P Q` (`P \/ Q`) **の証拠を構成する方法は二通り**:
  - `P` **の証拠から構成**
  - `Q` **の証拠から構成**

自然演繹における導出規則

$$\frac{\Gamma \vdash P : \mathtt{Prop} \qquad \Gamma \vdash Q : \mathtt{Prop}}{\Gamma \vdash P \vee Q : \mathtt{Prop}} \quad (\text{P-}\vee)$$

$$\frac{\Gamma \vdash P}{\Gamma \vdash P \vee Q} \quad (\vee\text{-I1})$$

$$\frac{\Gamma \vdash Q}{\Gamma \vdash P \vee Q} \quad (\vee\text{-I2})$$

$$\frac{\Gamma \vdash P \vee Q \qquad \Gamma, H : P \vdash R \qquad \Gamma, H : Q \vdash R}{\Gamma \vdash R} \quad (\vee\text{-E})$$

「または」についての証明(1)

```
Check or_introl.
Check or_intror.
```

「または」についての証明(2)

```
Theorem or_commut : forall P Q : Prop,
  P \/ Q -> Q \/ P.
Proof.
  intros P Q H.
  inversion H as [HP | HQ].
    Case "left".  apply or_intror. apply HP.
    Case "right". apply or_introl. apply HQ.
Qed.
```

H は

- `or_introl` **P Q HP (ただし HP : P) の形か**
- `or_intror` **P Q HQ (ただし HQ : Q) の形**

のいずれか $\Longrightarrow$ `inversion` **は** 2 **つのサブゴールを生成**

「または」についての証明(3)

```
Theorem or_commut' : forall P Q : Prop,
  P \/ Q -> Q \/ P.
Proof.
  intros P Q H.
  inversion H as [HP | HQ].
    Case "left".  right.  apply HP.
    Case "right". left. apply HQ.
Qed.
```

- left. は apply or_introl. の略
- right. は apply or_intror. の略

カリー・ハワードの対応 (その 2，続き)

論理の世界	計算 (プログラム) の世界
または	直和型

```
Inductive or (P Q : Prop) : Prop :=
  | or_introl : P -> or P Q
  | or_intror : Q -> or P Q.

Inductive sum (X Y : Type) : Type :=
  | inl : X -> sum X Y
  | inr : Y -> sum X Y.
```

- `sum nat bool` は「自然数か真偽値」の型
- `inl nat bool 2 :  sum nat bool`
- `inr nat bool true :  sum nat bool`

「かつ」「または」と andb, orb

論理結合子 $\wedge$, $\vee$ と真偽値上の関数 andb, orb の関係

```
Theorem andb_true__and : forall b c,
  andb b c = true -> b = true /\ c = true.

Theorem and__andb_true : forall b c,
  b = true /\ c = true -> andb b c = true.

Theorem andb_false : forall b c,
  andb b c = false -> b = false \/ c = false.
```

など

宿題： 1/9 午前10:00 締切

- Exercise: `proj2` (1), `and_assoc` (2), `even__ev` (2), `iff_properties` (1), `or_distributes_over_and_2` (2), `bool_prop` (2),
- **解答を書き込んだ `Logic.v` をまるごとオンライン提出システムを通じて提出**
- **以下をコメント欄に明記**:
  - **講義・演習に関する質問，わかりにくいと感じたこと，その他気になること．(「特になし」はダメです．)**
  - **友達に教えてもらったら、その人の名前，他の資料 (web など) を参考にした場合，その情報源 (URL など)．**