

# 「計算と論理」

## Software Foundations

### その0

五十嵐 淳

`igarashi@kuis.kyoto-u.ac.jp`

京都大学

October 1, 2013

# 今日のメニュー

- 講義・演習の概要説明
- (工学部3号館1F演習室1へ移動)
- 教科書のダウンロード・動作確認

# 担当教員について

- 名前: 五十嵐 淳 (いがらし あつし)
- 所属: 情報学研究科 通信情報システム専攻 計算機ソフトウェア分野
- オフィス: 総合研究7号館 224号室 (火曜日の4限は在室予定)
- 講義についての連絡: `cal13@fos.kuis...`
- 講義 WWW ページ: `http://www.fos.kuis.kyoto-u.ac.jp/~igarashi/class/cal/`

# TA

- 名前: 奥村 健太郎 (おくむら けんたろう)
- 所属: 情報学研究科 通信情報システム専攻 計算機ソフトウェア分野
- オフィス: 総合研究7号館 227号室

# 講義内容

## シラバスより

数理論理学の基礎と，数理論理学を用いた計算機プログラムの検証について講述する．また，講義を補完するため，証明支援系（計算機上で数学的証明を行うシステム）である Coq を用いた演習を行う．

# 数理論理学

判断 (judgment) について (数理的手法で) 考える学問

判断 真偽を考えることが可能な文

- 命題論理: 単純な判断を組み合わせて複合的な判断を構成する「接続詞」の理論
  - ▶ 「かつ」「または」「ならば」「～ではない」
- 述語論理: 量化を伴う判断の理論
  - ▶ 「任意の について～である」「ある が存在して～である」
- (様相論理: 真偽を修飾する副詞の理論)
  - ▶ 「必然的に～である」「～である可能性がある」「未来永劫～である」

# 数理論理学: 意味論と証明論

- 意味論...与えられた判断が「真である」とはどういうことかを考える
  - ▶ 真理値表 (論理関数) は命題論理の意味論のひとつ
- 証明論...与えられた命題の「証明」とは何か,「証明が同じ・違う」とはどういうことかを考える
  - ▶ 様々な証明 (記述) 体系: 自然演繹, シーケント計算, ヒルベルト流公理
  - ▶ 証明の構造について考える
    - ★ 証明の等しさ, 簡単さ

# 計算機プログラムの検証

「計算機プログラム」の正しさの証明を与える

- 「正しさ」の基準  $\Rightarrow$  判断として書かれた仕様 (specification)
- 例:

リストを反転させる Scheme 関数 `rev` の仕様

任意のリスト `xs` について  $(\text{rev} (\text{rev} \text{xs})) = \text{xs}$

Q. これだけで仕様として十分といえるだろうか？  
(他にも `rev` が満たすべき仕様はないだろうか？)

- c.f. 単体 (unit) テスト



# 証明支援系 Coq を用いた演習

証明支援系: 計算機で数学をするためのソフトウェア

- 数学的対象 (数, リスト, 木など) 定義とその対象を操作するプログラムの記述言語
    - ▶ Scheme のような関数型プログラミング
    - ▶ ただし静的に型がついている
    - ▶ そして文法がちょっと変わっている
  - それらの対象に対する性質を述べる判断の記述言語
  - 判断の証明の記述言語
  - 証明の検査機能
  - (自動証明機能)
- を使って, 色々なプログラムや証明を書く

# 講義内容

## シラバスより

数理論理学の基礎と，数理論理学を用いた**計算機プログラムの検証**について講述する．また，講義を補完するため，証明支援系（計算機上で数学的証明を行うシステム）である**Coq**を用いた**演習**を行う．

## 講義の(裏)テーマ

証明 = プログラム

(Curry-Howard 同型対応としても知られる論理と計算の関係)

# 教科書

Benjamin C. Pierce, et al. Software Foundations.  
<http://www.cis.upenn.edu/~bcpierce/sf/>

- 注意: オンライン・テキストで予告なく内容が変わる可能性あり
- この講義では, 2013年9月時点でのスナップショットを使う
  - ▶ 講義 WWW ページから「ダウンロード用」のリンクあり
  - ▶ ユーザ名: cal2013
  - ▶ パスワード: cookadoodledoo

# 成績評価

- 宿題 30%
  - ▶ 宿題提出システムへの登録が必要 (次のスライド)
- 期末試験 70%
- 随意課題によりさらに加点

# 重要: 宿題提出システムへの登録

以下の要領でメールを cal13@fos.kuis... へ送信

- Subject: 提出システム登録願
- 本文先頭4行に, 氏名, 氏名のふりがな, 学籍番号, メールアドレス, をこの順で明記

To: cal13@fos.kuis.kyoto-u.ac.jp

Subject: 提出システム登録願

---

五十嵐 淳

いがらし あつし

12345678

igarashi@kuis.kyoto-u.ac.jp

# 宿題：10/8 午前10:30 締切

- 端末室での設定と動作確認を済ませる
- 宿題提出システムへの登録を済ませる
- Preface, Basics を予習し，今日配る質問用紙に，予習時に生じた質問と自分なりの予想回答を記入
  - ▶ (提出は授業開始時)

# これから端末室で行うこと

- Linux 環境にログイン
- 環境設定
- 教科書のダウンロード
- Coq の動作確認

# Linux 環境にログイン

- Windows が立ち上がったら VirtualBox を起動
- ログイン



# 環境設定

環境変数 PGHOME の値を

`/usr/share/emacs/site-lisp/ProofGeneral` に設定する .

- (デフォルトの) bash の場合: `~/.bashrc` に  
`export PGHOME=/usr/share/emacs/site-lisp/ProofGeneral`  
の一行を追加
- (`source ~/.bashrc` を実行して上の設定を反映させる .)
- 他のシェルの場合: 自分でできますね? :-)

# 教科書のダウンロード・解凍・展開

- 前のページにある URL から教科書の .zip ファイルをダウンロード
- 適当なディレクトリに解凍・展開
- sf というディレクトリができる
  - ▶ index.html をブラウザで読み込む
  - ▶ \*.v が 各章の Coq ファイル

# Coq の動作確認 (方法1)

Proof General という Emacs から Coq を呼び出すための emacs lisp ソフトウェアを使う

## Proof General 起動方法

```
% cd <教科書のディレクトリ>  
% proofgeneral Basics.v
```

- 軍人さんが現れた後，ファイルの内容が表示される
- C-c C-n で，ファイルの内容が少しずつ (決まった単位で) Coq に送られ，処理済部分の背景が青くなる．
- C-c C-u は逆 (undo)
  - ▶ ツールバーの左右矢印でも操作可能

# Coq の動作確認 (方法2)

CoqIDE という Coq 用統合開発環境を使う

## CoqIDE 起動方法

```
% cd <教科書のディレクトリ>  
% coqide Basics.v
```

- ファイルの内容が表示される
- ツールバーの下矢印で、ファイルの内容が少しずつ (決まった単位で) coq に送られ、処理済部分の背景が緑になる。
- 上矢印は逆で undo する。
  - ▶ ショートカットキーもあります

# 自宅などで Coq を使う

- Coq 8.4 をインストール
  - ▶ <http://coq.inria.fr/download> から
    - ★ Linux: ソースファイルをダウンロードしてコンパイル (かディストリビューションのパッケージをインストール)
    - ★ Windows, Mac: バイナリをダウンロードしてインストール
  - ▶ CoqIDE は付属している
- Proof General をインストール (オプション)
  - ▶ Linux (Ubuntu) の場合: proofgeneral-coq というパッケージがある
  - ▶ Windows, Mac の場合: Emacs をインストールして, <http://proofgeneral.inf.ed.ac.uk/> あたりから頑張ってください

# 受講上の注意

- ノート PC 持込受講を歓迎します
- 実際に証明を書いてみないと身につきません
- 書かれている記号の意味をよくよく考えましょう
  - ▶ みようみまねでいつの間にか証明ができるのは危険な徴候