「プログラミング言語の基礎概念」補助資料
演習システムの使い方

五十嵐 淳
京都大学 大学院情報学研究科通信情報システム専攻
e-mail: igarashi@kuis.kyoto-u.ac.jp

平成 31 年 4 月 16 日

この文書は，五十嵐淳著「プログラミング言語の基礎概念」(サイエンス社) で使われるオンライン演習システムの使い方を簡単に述べるものである．

「プログラム意味論」受講者のみなさんへ: アカウントを取得したらアカウント名と氏名・ふりがなを五十嵐まで知らせてください．

1 演習システムへの登録

http://www.fos.kuis.kyoto-u.ac.jp/~igarashi/CoPL/ にアクセスし，新規ユーザ登録のリンクをクリックし，登録画面を呼び出す．希望するユーザ ID，氏名，メイルアドレスを入力し,「登録」ボタンを押す．パスワードが入力したメイルアドレスに届くので，それを確認した上でトップページ http://www.fos.kuis.kyoto-u.ac.jp/~igarashi/CoPL/ からログインする．入力アドレスを間違えた場合には登録のキャンセルをする．

2 演習システムの使い方

演習システムの URL は http://www.fos.kuis.kyoto-u.ac.jp/~igarashi/CoPL/ である．ここにアクセスし，ユーザ ID とパスワード (忘れた場合は同ページから取り寄せられる) を入力しログインする．

ログイン直後は，左にサイドバー,「おしらせ」と「おすなば」(下を参照のとこ) が表示されているはずである．

左のサイドバーから，問題セクション・問題番号をクリックすると，問題文 (通常，与えられた判断の導出を答えるもの) と解答を入力するフォームのページになるので，解答となる導出を入力し「フォームの解答を送信」ボタンを押す (もしくは解答を記入したファイル

をアップロードし,「ファイルを送信」ボタンを押す). 導出システムの推論規則は問題文からリンクされておりオンラインで見ることができる.

サイドバーでは, その他に「おすなば」「おしらせ」「統計」「ログアウト」をクリックすることでそれぞれのコマンドが実行できる.

2.1 おすなば

「おすなば」では, (問題とは関係なく) フォームに導出を入力することで正しいかどうかをチェックすることができる. どの導出システムの規則でチェックするかの指定を間違えないように!

2.2 おしらせ

演習システムに関するおしらせが表示される.

2.3 統計

ここで自分の進度がどれくらいかチェックすることができる.

2.4 ログアウト

ログアウトできる.

3 導出の構文

演習システムに入力する際の導出の表記 (ASCII 表記と呼ぶ) を説明する. あるシステムの判断 $\mathcal{J}$ の ASCII 表記を $\mathcal{J}^\flat$ とする. (通常はテキストにあるタイプライタ体の文字そのままだが, 記号を使う場合には適宜その ASCII 表記が与えられる. 表 1 参照.) この時, 導出

$$\mathcal{D} \equiv \frac{\mathcal{D}_1 \quad \cdots \quad \mathcal{D}_n}{\mathcal{J}_0}\ \textsc{RuleName}$$

に対して, その (一次元)ASCII 表記 $\mathcal{D}^\flat$ は, 再帰的に

$\mathcal{J}_0^\flat$ by RuleName {
  $\mathcal{D}_1^\flat$;
  $\vdots$
  $\mathcal{D}_{n-1}^\flat$;
  $\mathcal{D}_n^\flat$
}

表 1: 記号の ASCII 表記変換表

教科書での記号	ASCII 表記
$\Downarrow$	`evalto`
$\longrightarrow$	`--->`
$\longrightarrow^*$	`-*->`
$\longrightarrow_d$	`-d->`
$\vdash$	`|-`
$\bullet$	(省略)
(環境，型環境先頭の) $\bullet$,	(省略)
(`fun` 式，`match` 式，関数型の) $\rightarrow$	`->`
(NamelessML3 の) $\Longrightarrow$	`==>`
(EvalContML1,4 の) $\Rightarrow$	`=>`
$\gg$	`>>`

で与えられる．例えばシステム Nat の導出

$$\dfrac{\dfrac{}{\texttt{Z times S(Z) is Z}}\,\textsc{T-Zero} \quad \dfrac{\dfrac{}{\texttt{Z plus Z is Z}}\,\textsc{P-Zero}}{\texttt{S(Z) plus Z is S(Z)}}\,\textsc{P-Succ}}{\texttt{S(Z) times S(Z) is S(Z)}}\,\textsc{T-Succ}$$

は，

```
S(Z) times S(Z) is S(Z) by T-Succ {
  Z times S(Z) is Z by T-Zero {};
  S(Z) plus Z is S(Z) by P-Succ {
    Z plus Z is Z by P-Zero {}
  }
}
```

となる．(大文字・小文字の区別に注意すること．P-Zero は `P-Zero` になる．)

また，Java のように `//` から行末まで，OCaml のように `(*` と `*)` で囲まれた部分はコメントとして扱われる．

4 その他の (非) 機能

- 以前に入力した解答を問い合わせる機能は (今のところ) ないので，必要があれば手元で保存すること．(前の解答を使いたい，というケースが多くある．)
- 判断の後の「`by RuleName { ... }`」の代わりに 「`?`」を書くと「未完成の (部分) 導出」としてシステムに受け付けてもらうことができる．(導出を並べる際の末尾のセミコロンは必要である．)

- 規則の前提にある括弧で囲まれた部分は，付帯条件であり，導出検査器が自動的に検査する条件なので，解答となる導出には何も書かなくてよい．
- EvalML1–5 などでは以下の判断の省略形が使用可能である．

省略形	展開形
i_1 `is less than` i_2	i_1 `less than` i_2 `is true`
i_1 `is not less than` i_2	i_1 `less than` i_2 `is false`

- PolyTypingML4 において，型変数は ’ (アポストロフィ) で始まる英数名 (英小文字もしくはアンダスコアで始まり，0 個以上の英数字・アンダスコア・アポストロフィが続く) を使う．
- EvalRefML3 において *Loc* の要素は `@` で始まる英数名を使う．
- EvalContML1,4 において継続表現末尾に現れる ≫ _ は省略してよい．

5 講義受講者への注意とアドバイス

- アカウントを取得したらアカウント名と氏名を五十嵐まで知らせること．
- 解答をプログラムで自動化してもらって構わない．その場合，期末にプログラムとその説明をレポートに加えること．
- (関数などでない限り) 値のわからない式がでてきたら OCaml 処理系を使うべし．
- 上にも書いたように解答を保存してあとで取り出す機能はない．
- 問題は順に解く必要はない．むしろ，面倒になってきたら次のセクションに飛ぶ方が吉．