プログラミング演習B
ML編　第7回

2011/6/21（コミ）
2011/6/22（情報・知能）

住井

http://www.kb.ecei.tohoku.ac.jp/~sumii/class/proenb2011/ml7/

今日のポイント

1. 単純な言語処理系（インタプリタ）の作り方（やや発展的内容）

レポートについて

電気・情報系内のマシンから

http://130.34.188.208/ （情報・知能）

http://130.34.188.209/ （コミ）

にアクセスし、画面にしたがって提出せよ。

締め切りは一週間後厳守（ただし課題7.3のみ8月20日）。

- 「プログラム」のテキストボックスがある課題では、プログラムとしてsmlに入力した文字列のみを過不足なく正確にコピー＆ペーストして提出せよ。（smlの出力は「プログラム」ではなく考察に含めて書くこと。）
- プログラムの課題でも必ず考察を書くこと。
- 提出したレポートやプログラムの実行結果は「提出状況」から確認できる。
  - 質問はml-enshu@kb.ecei.tohoku.ac.jpにメールせよ。
  - レポートの不正は試験の不正と同様に処置する。

注意

8月の締切当日まで計算機室が利用できるかどうか未定のため、あらかじめ

http://localweb.ecei.tohoku.ac.jp/sysinfo/

などでよく確認すること。

プログラムと
プログラミング言語

- プログラム：
  計算機に対する命令書

- プログラミング言語：
  プログラム（計算機に対する命令書）を記述するための言語

コンパイラとインタプリタ

- コンパイラ：
  ある言語のプログラムを、**別の言語に変換する方法**を記述したプログラム
  - コンパイラ自身もまた何らかの言語で書かれている
- インタプリタ：
  ある言語のプログラムを**実行する方法**を記述したプログラム
  - インタプリタ自身もまた何らかの言語で書かれている

今日の内容

- 単純な命令型言語「MyC」のインタプリタの作り方を見る

言語処理系の基礎

- **字句解析** (lexical analysis)：
  abc+xy*z **のような文字列を**
  「abc」「+」「xy」「*」「z」
  のような字句(token)の列にわける
- **構文解析** (parsing)：
  字句の列をさらに +(abc,*(xy,z))
  のような構文木(parse tree)にする

コンパイラでもインタプリタでも必要

LexとYacc

- Lex：**字句解析器**(lexer)を簡単に生成するためのツール
  - Standard ML用はML-Lex
- Yacc：**構文解析器**(parser)を簡単に生成するためのツール
  - Standard ML用はML-Yacc

MyC言語の構文

S ::=	$x = i$	定数代入
|	$x = y + z$	加算
|	$\{ S_1 ; \ldots ; S_n \}$	逐次実行
|	while (x>y) S	条件付反復
|	print x	出力

ただし$S, S_1, \ldots, S_n$は文、
x, y, zは変数、 iは整数定数

ML-LexとML-Yaccによる記述

http://www.kb.ecei.tohoku.ac.jp/~sumii/class/proenb2011/myc/

- syntax.sml：構文木を表すバリアント型の定義
- myc.lex：字句解析器の記述
- myc.grm：構文解析器の記述
- myc.sml：インタプリタの定義

(sources.cm：smlで実行するための記述)

実行方法

1. UNIXのコマンドラインで
   ml-yacc myc.grm を実行
2. smlを起動し
   CM.make "sources.cm"
   を評価
3. MyC.run ()を評価し、
   「MyCの文の後に;をつけてEnter」
   の繰り返し（Ctrl-cで中止）

例

- MyC.run () ;

（中略）

```
n = 10 ;
sum = 0 ;
minus = -1 ;
zero = 0 ;
while (n > zero) { sum = sum + n ; n = n + minus } ;
print sum ;
```

55

課題7．1

「12×34」を計算して出力するMyC言語のプログラムを考え、先のインタプリタ(MyC.run)の上で実行せよ。

課題7．2

先のMyCインタプリタを改造し、
減算文「x = y - z」と
乗算文「x = y * z」を追加せよ。

追加した文を用いる、できるだけ
自明でないプログラムを作り、
実行せよ。

課題7．3 (optional)

MyCインタプリタをさらに拡張し、
できるだけ自明でない機能を実装せよ
（例えばif文や関数定義など）。

または、MyC言語以外の任意の言語の
インタプリタもしくはコンパイラを実
装せよ。既存の言語（の一部）でも、
独自に考えた言語でも良い。