ML 演習 最終回

おおいわ

July 17, 2001

今回の内容

- 最終課題
  - Prolog インタプリタの作成
    - most-general unifier, substitution
  - その他

最終課題

- 次の課題 1～7 のうち1問以上を解け。
  - 課題1: 基本課題
  - 課題 2～7: Optional 課題
    - 難度はまちまちです。
  - 2問以上の回答も大歓迎

課題一覧

1. Prolog インタプリタの作成 *
2. 実用アプリケーション製作 **
3. OCaml のオブジェクト指向について **
4. Mini-ML インタプリタの拡張 *~***
5. 言語の設計に関する考察 ***
6. 新しいプログラミング言語の設計 ****
7. (ICFP Programming Contest ***?)

課題1 (1)

- 簡単な Prolog 処理系を作れ。
  - rule のリストと query を受け取り、MGU の1つを返す関数を作れ。
  - literal としては、atom, variable, compound term をサポートせよ。
  - cut はサポートしなくても良い。

課題1 (2)

- parser として今回も ocamlyacc, ocamllex を用いた物を用意してあるので必要に応じて用いてよい。
- 余力があれば cut の実装や、対話型処理系の実装などにも挑戦すると良い。

課題1 ヒント (1)

- substitution の表現
  - type substitution = (identifier * term list)
  - 例: [“X”, Atom “abraham”;
    “Y”, Atom “haran”;
    “Z”, Atom “nachor”]
- mgu を求めるアルゴリズム
  - Prolog 演習 第2回レジュメ (pp. 6～10)
  - 前回の型推論も参考になるはず

課題1 ヒント (2)

- query “?- q, q'.” を解くアルゴリズム:
  - rule (fact) を1つ1つ見ていく。
  - rule を a :- b, b', b''. とする。
    - rename することを忘れないこと。
  - もし、q と a の間に mgu θ があるなら、query “?- $b\theta$, $b'\theta$, $b''\theta$, $q'\theta''$ を解く。query が空になったら、その時点までの mgu を全て合成したものが答え。
  - θが無かったり sub-query に失敗したら次の rule を調べる。

課題 2 (1)

- OCaml の次の機能の 1 つ以上を用い、実用アプリケーションを作成せよ。
  - portable graphics
  - GUI (lablTk)
  - ソケット入出力 (unix)
  - ocamllex, ocamlyacc
  - Thread
  - その他... (GTK, png など外部モジュールも可)

課題 2 (2)

- 各モジュールの使い方はマニュアルを。
- ocamlc は通常の gcc などとライブラリを指定する方法が異なるので注意。
  - gcc:
    % gcc -o foobar foo.o **-ltermcap**
  - ocamlc:
    % ocamlc -o foobar **unix.cma** foo.cmo
- モジュール (.cmo) のリンク順も注意。

課題 3

- Objective Caml の “Objective” 部分の機能について調べよ。
  - 型システム
  - C++, Java, Modula-2 など他言語の OOP サポート機構との比較
  - 実装手法
  - 具体的な記述性　　など。

課題 4

- 第 5〜7 回の Mini-ML interpreter を更に拡張せよ。
  - 例えば…
    - Exception, reference
    - Datatype (Variant, Record)
    - First-class continuation (call/cc)
    - Module と 抽象データ型
    - オブジェクト指向?
    - 部分評価?　　など…

課題 5

- これまでの課題を OCaml 以外の 2 つ以上の言語で再実装し、その記述性や効率などについて深く考察せよ。
  - きちんと同じ機能を持つ実装を作ること。
  - 説明していない OCaml の機能 (配列など) も必要に応じて考慮して適切な比較をすること。

課題 6 (1)

- OCaml を含む既存の言語の不満な点を示し、その問題点を解決する言語を設計し処理系 (と応用例) を実装せよ。
  - syntax (構文) と semanatics (意味) を明確にすること。
  - 特定の目的に特化して考えても良いが、きちんと適切な対照言語を見つけてどんな問題を解決したか示すこと。

課題 6 (2)

- 例
  - Domain-specific languages
    - Web スクリプティングのための言語
    - ハードウェアのシミュレーションのための言語
    - ゲームの思考ルーチンのための言語　など…
  - 汎用 scripting 言語
    - 打倒 Perl, Ruby, Python
  - 汎用言語
    - 打倒 C++, C#, Java, OCaml…

課題(?) 7

- ICFP Programming Contest に参加した人はレポートで最終課題と認定します。
  - http://cristal.inria.fr/ICFP2001/prog-contest/
  - 72時間 (7/27～29)、人数制限なし
  - 結構難しいけどやりがいはあります
    - Caml じゃなくてもいいけど...

提出方法

- 〆切: 2001年9月30日 (日) 24:00
- メール提出: ml-report@yl.is.s.u-tokyo.ac.jp
  - 題名: Report 8-*課題番号 学生証番号*
- 紙提出: 1F レポートボックスへ
  - プログラムはメールか Web で
  - 提出したことをメールで通知してください

1回提出した後の追加も受け付けます。

単位について (1)

- 判定条件 (原則)
  - 予告通り「期限追従」「全提出」の OR
- 解けそうもない人は…
  - 早め (7~9月) に相談 (ml-query 宛)
    - これまでの提出状況を勘案して目標設定します
    - 代替条件を検討します
      - 例: 第1回～第4回必須課題 + 第8回課題 × 2～3
    - 特に4年生はきちんと相談してください

単位について (2)

- 昨年もあった事態ですが…
  - 成績票の事務への提出後・直前に泣き付く
    - 具体的な日程は未定
  - (4年生) 卒業判定会議直前に泣き付く

…など手遅れになる前に課題提出を。

- 今年は hard limit 2週間以内の negotiation は無条件で *reject* させてもらいます
  - 追加課題出しても間に合わないでしょ…