

ML 演習 最終回

おおいわ

June 18, 2000

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 1～5 のうち最低1問を解け。
- め切: 2000年9月30日(土) 24:00
- 提出先:
 - 課題1を含む場合:
ml-report-prolog@yl.is.s.u-tokyo.ac.jp
 - その他: ml-report@yl.is.s.u-tokyo.ac.jp
- 題名: Report 8-課題番号 学生証番号

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 ocamllyacc, 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 図6 (p. 9)
- 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 (Tk)
 - 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

- 6・7回の Mini-ML interpreter を更に拡張せよ。
 - 例えば...
 - exception, reference の導入...。
 - datatype など...。
 - 遅延評価ベースのインタプリタの作成...。
 - call/cc ??
 - などなど...。

課題4

- これまでの課題のプログラムを他のプログラミング言語で書き直し、記述性、効率などを具体的な事実に基づいて比較せよ。
 - C, C++, Java, Pascal, Scheme, Prolog, ...

課題5

- 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を実装せよ。
 - syntax (構文) と semantics (意味) を明確にすること。