OCaml (あるいは F#) のプログラミングに関する課題
2016.4.21 (演習後に修正したもの；変更点は太字)

この授業は、OCaml/F# などの関数型プログラミング言語を習う授業ではなく、これらを道具として使って、一般のプログラミング言語の様々な機能を記述します。 (メタ言語として使います。)

したがって、OCaml/F# の全ての機能 (素晴しい機能) を網羅するつもりはなく、ここで使うのは、「OCaml/F# のなるべく小さなサブセット (他のプログラム言語の処理を記述するのに必要なだけの最小限のサブセット)」です。

この授業で、これらの言語に興味を持った人は、インターネットあるいは市販参考書を見て自分でプログラムを書いてみるとよいでしょう。

- 課題 1. ウェブページのリンク先の OCaml プログラムをすべて実行し、コメントや計算結果を参考に意味を理解せよ。必要ならば、プログラムの一部を改変して実行してみなさい。(この部分は、レポートとしての提出は不要です。ただし、必ず自分の手でやっておくことが必要です。)

- 課題 2 (OCaml を初めてさわったという人向け)

  以下の関数を書きなさい。入力は正の整数 1 つ。出力はその (正の) 約数の個数。

  たとえば、28 を入力すると (約数は 1,2,4,7,14,28 なので) 6 が返り、13 を入力すると (約数は 1,13 なので)2 が返る。なお、厳密にいえば -2 や -7 も 28 の約数であるが、ここでは、正の約数の個数のみを数えることにする。

  なお、このような関数は、`int -> int` という型をもつはずである。

- 課題 2' (OCaml を初めてさわったという人向け)

  以下の関数を書きなさい。入力は正の整数 1 つ。関数は、「それが, 1 なら終了、偶数なら 2 でわり、3 以上の奇数なら 3 倍して 1 を足す」という操作を行い再帰的に動作する。1 になるまで延々と計算しつづけるので (無限ループかもしれない)、100 回まわったら、その時点の答えを返しておわり。

- 課題 2" (OCaml を初めてさわったという人向け； もっともお勧め)

  課題 2' において「100 回まわったらおわり」をはずしたものを書きなさい。(つまり、無限ループするかもしれない関数でよい。)

- 課題 3 (OCaml 経験者向け)

  2 週目のスライドの binarytree データ型を**適当に変形して**、「ソートされたデータの集合」あらわそう。つまり、「木の左にあるデータは、常に、木の右にあるデータより小さいか等しい」という性質をみたす二分木のみを相手にする。

  そのような二分木と、データ v があたえられたとき、v を適切な位置に挿入する (ソートされた状態をたもつ) 操作を OCaml 関数として実現せよ。

  また、そのような二分木と、データ v があたえられたとき、その二分木にふくまれる v を削除する操作を OCaml 関数として実現せよ。ただし、v が含まれていなかったときは何もしないで入力の二分木をそのまま返す。

  余力があれば： 上記のような操作後に、木が「バランスしない」場合に、バランスさせた木に変形して返してください。(木がバランスする、とは、木のすべてのノードにおいて、左部分木の高さと、右部分木の高さが同じか、差が 1 であると定義します。)

- 課題 4 (上のものでは物足りない人だけに向けた optional 課題)

  2 週目のスライドの最後にかいてある CK 抽象機械を実行するインタプリタを OCaml で書いてください。

提出先: Manaba システム, 提出締切:　来週月曜深夜

上記の課題 2 と課題 2' と 課題 3 のどれか 1 つ (以上) を 1 つのレポートに書いて提出してください。課題 4 もできた人は、それに追加して (1 つのレポートとして) 出してください。