OCaml (あるいは F#) のプログラミングに関する課題

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

  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

  以下の関数を書きなさい。入力は正の整数 2 つ：1 引数が 1 あるいは第 2 引数が 1 なら 第 1 引数を返して終了、そうでなければ「第 1 引数が偶数なら 2 で割り、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。ただし、第 2 引数は 1 を引く。つまり、f(4,5) を呼ぶと、次に f(2,4) を呼ぶ。また、f(7,5) を呼ぶと、次に、f(22,4) を呼ぶ。

  第 2 引数が 1 つずつ減っていくので、いつかは終了するので、そのときの第 1 引数の値が返るはずである。

- 課題 3 (発展課題)

  講義スライドにかいてあるインタプリタを改良して、整数だけでなく、浮動小数点数 (float 型の値) の四則演算ができるようにしなさい。(ただし、整数と浮動小数点を「まぜて」使うプログラムを処理するのは大変なので、そのようなものは考えなくてよい。)

提出先: Manaba システム, 提出締切:　来週水曜深夜