

2006年1月9日

2006年度後期・数理解析・計算機数学 III レポート課題

評価方法について

ともにプログラムの正しさと明快さを評価します。全ての問に答えなくてもいいが、コンパイルできないプログラムの提出は避けて下さい。

レポートの提出方法

メールで提出して下さい。宛先は `computer-lecture-2006-aw-3@math.nagoya-u.ac.jp` です。届いたら確認のメールを送ります。

提出期限

提出期限は2006年2月6日(火)。
修論を提出するM2の場合、期限は2006年1月30日(火)。

課題

問A・Bの両方に答えて下さい。

問A

第V章を元に以下のプログラムを作して下さい。

1. 関数 `extract_closest` および `add_edge` を定義せよ。(練習問題1.1)
2. 経路を記録するコードを書きなさい。(練習問題1.2)
3. ライブラリーのデータ構造を利用したバージョンを元に、以下の引数を使ってコードをファンクター化せよ。

```
module type Graph = sig
  type graph
  type vertex
  type edge
  val fold_graph : (string -> vertex -> 'a -> 'a) -> graph -> 'a -> 'a
  val get_vertex : graph -> string -> vertex
  val fold_vertex : (string -> edge -> 'a -> 'a) -> vertex -> 'a -> 'a
  val length : edge -> float
end
```

即ち、以下のインターフェースを満すファンクターを定義し、

```
module Path :  
  functor (G : Graph) ->  
    sig  
      val shortest :  
        graph:G.graph -> seen:VSet.t -> nexts:path VMap.t -> VMap.key -> path  
      val shortest_path : G.graph -> VMap.key -> VMap.key -> path  
    end
```

さらに

```
type edge = float  
type vertex = (string * edge) list  
type graph = (string * vertex) list
```

のときに Graph の実装も書きなさい。

4. 前問の Graph の実装において、各頂点に 2 次元座標を与える。

```
type point = {x:float; y:float}  
type edge = float  
type vertex = point * (string * edge) list  
type graph = (string * vertex) list
```

亀グラフィックスを参考にして、LablTk の Canvas の上でグラフを表示せよ。

5. 二つの頂点をもらい、その間の最短経路を赤で表示するようにせよ。

問 B

以下の関数の型を見て、その関数の定義を推論せよ。

1. $f1 : ('a \rightarrow 'b) \rightarrow ('b \rightarrow 'c) \rightarrow ('a \rightarrow 'c)$
2. $f2 : ('a * 'b) \text{ list} \rightarrow ('b * 'a) \text{ list}$
3. $f3 : (\text{int} \rightarrow 'a) \rightarrow 'a$
4. $f4 : 'a \text{ list} \rightarrow (\text{int} * 'a) \text{ list}$
5. $f5 : (('a \rightarrow 'a) \rightarrow 'b) \rightarrow 'b$