Laboratorio di Programmazione a.a. 2013-14

Problema 13

15 Aprile 2014

Descrizione

Rielabora il programma che applica la tecnica di *memoization* per determinare una sottosequenza comune più lunga (LCS) e definisci un metodo statico `allLcs` per calcolare la lista di tutte le soluzioni diverse fra loro.

Per quanto riguarda le procedure che operano sulle liste, considera esclusivamente *liste di stringhe*. A tale proposito utilizza la classe (atipica) `StringList`, che emula in Java le liste di Scheme, purché gli elementi siano tutti di tipo `String`. Il protocollo per operare con questo tipo di liste è riformulato come segue:

Tipo `IntList` :

`StringList.NULL`	costante lista vuota [`null`]
`StringList.listNull( lst )`	pedicato per verificare se una lista è vuota [`null?`]
`StringList.listCar( lst )`	primo elemento di una lista [`car`]
`StringList.listCdr( lst )`	lista che si ottiene eliminando il primo elemento [`cdr`]
`StringList.listCons( s, lst )`	lista che si ottiene aggiungendo un nuovo elemento all'inizio [`cons`]

Nota che in Java occorre premettere ai nomi di "costanti" e "procedure" il nome della classe che li definisce. Per esempio, la lista `("a" "" "bc")` può essere definita come segue:

```
StringList.listCons("a",StringList.listCons("",StringList.listCons("bc",StringList.NULL)))
```

Esempi — l'ordine delle stringhe in una lista non è vincolante, ma non si devono avere ripetizioni:

```
MyLCS.allLcs( "cane", "topo" )                  →  ("")

MyLCS.allLcs( "arto", "atrio" )                 →  ("ato" "aro")

MyLCS.allLcs( "salva", "saliva" )               →  ("salva")

MyLCS.allLcs( "artigianato", "trevigiano" )     →  ("tigiano" "rigiano")


MyLCS.allLcs( "articolazione", "alcaloide" )
                      →  ("alaoe" "alaie" "acaoe" "acaie" "acloe" "aclie" "acoie")
```

(La classe `StringList` deve essere collocata nella stessa cartella in cui si trova la tua soluzione, per esempio codificata nel file `MyLCS.java`)