Programmazione Funzionale

Esercitazione 6 – Principi per l'Analisi Lessicale

Esercizio 1. Definire la funzione implode che prende una lista di caratteri e costruisce la stringa corrispondente.
Ad esempio, `implode [ 'c' ; 'o' ; 's' ; 'a' ; ' ' ; 'e']` restituisce la stringa `"cosa e"`.

Esercizio 2. Definire la funzione explode che prende una stringa e ritorna la lista di caratteri corrispondente.
Ad esempio, `explode "una frase"` restituisce `['u' ; 'n' ; 'a' ; ' ' ; 'f' ; 'r' ; 'a' ; 's' ; 'e' ]`.

Esercizio 3. Una volta definite le funzioni precedenti definire la funzione `blankOnce` che trasforma tanti spazi vuoti in un singolo spazio vuoto `' '`. Ad esempio `"una frase"` diventa `"una frase"`.

Esercizio 4. Consideriamo di aver dichiarato il seguente tipo:

```
type espr = Int of int | Plus of espr * espr ;;
```

Definire una funzione `interpret : string -> espr` che prende in entrata una stringa e gli associa l'espressione corrispondente. Nei casi in cui non si puo associare un espressione la funzione sollevera un eccezione.

- Ad esempio, `interpret "3"` restituisce Int 3.
  Invece `interpret "Plus(3,Plus(1,2))"` restituisce l'epressione `Plus(3,Plus(1,2))`.

In un secondo tempo ottimizzare la funzione per cancellare i spazzi vuoti e.g. (' ') e i caratteri `\n`.