

Progetto di Linguaggi e Compilatori 1 – Parte 1
A.A. 2015/16

Gruppo 14

Marco Bucchiarone
Emanuele Tonetti
Francesco Zilli

1 Esercizio 1

La funzione Haskell `boundedMaximum(n [BST t])` data una lista di BST `t` trova se esiste

$$\max_n t = \max(\forall x \in t \mid x < n).$$

Il primo esercizio è stato risolto tramite la funzione `boundedMaximum`, la quale accetta un intero `n` e una lista di BST `x` e ritorna una lista di interi, che sono l'elenco di tutti i `maxn` degli elementi di ciascun BST. `Maxnt` la funzione che passato un BST ritorna l'elemento dello stesso che è il maggiore dei minoranti, che è ricavato applicando la funzione `max`, attraverso un `foldl1` aumentato di funzionalità, alla lista degli elementi strettamente minori di `n` ricavata con una funzione di visita del BST in modo da sfruttare la natura della struttura dati considerata. La funzione è testabile attraverso `ghci` e si forniscono opportuni test case dentro il file `es1queryghci.txt`, che comprende una rappresentazione più leggibile dei BST proposti in esso.

il metodo è stato impla

2 es2