

Progetto - Parte 3 - Gruppo 2

Callegari Filippo, Dreossi Talissa, Rosso Luca
Linguaggi e compilatori

Assunzioni e indicazioni per funzionamento codice

Le assunzioni rispetto alla specifica sono:

- il codice proposto è stato testato con ghc 8.6.5;
- i pesi sono sempre interi positivi, sia per un albero a valori interi che per uno a valori in virgola mobile;
- il nodo radice ha altezza pari a zero;
- il nodo radice ha sempre peso zero perché consideriamo gli archi che i nodi hanno con il padre e non con il figlio.

Per quanto riguarda i test abbiamo casi sia per alberi con valori interi che in virgola mobile in quanto abbiamo realizzato un unico parser.

Esercizio 1.a

La grammatica è la seguente (dove ϵ sta per il terminale "blanks"):

$$\begin{aligned} S &= a (C) \\ C &= S \epsilon C \mid S \end{aligned}$$

Per la definizione del parser utilizziamo due grammatiche distinte in modo tale da poter gestire gli errori in maniera coerente.

Esercizio 1.b

Il linguaggio che si genera è il seguente:

$$\{w \in T^* \mid w = \{\alpha a^* \alpha^R \cup \alpha (* \alpha^R \cup \alpha)^* \alpha^R \cup \epsilon\} \text{ con } \alpha \neq \epsilon\}$$

Infatti, considerato che $\{w \in T^* \mid w = w^R\}$ rappresenta l'insieme delle stringhe palindroma, gli unici alberi che vengono rappresentati da stringhe palindroma sono gli alberi con solo il nodo radice. Questo perché appena abbiamo un sottoalbero dobbiamo introdurre una parentesi aperta e una chiusa che quindi impediscono che la stringa sia palindroma. Pertanto dobbiamo considerare tutte le stringhe palindroma riproducibili a partire dall'alfabeto T rimuovendo solo l'albero con unico nodo radice ("a").

Esercizio 1.c

L'idea base della funzione `maximalWeight` è quella di visitare l'intero albero sommando di volta in volta il peso dell'arco successivo e applicando poi la funzione `maximum` a tutti i valori ottenuti. Si scende infatti ricorsivamente nella lista di sottoalberi.