

Programmazione Funzionale – Febbraio 2022

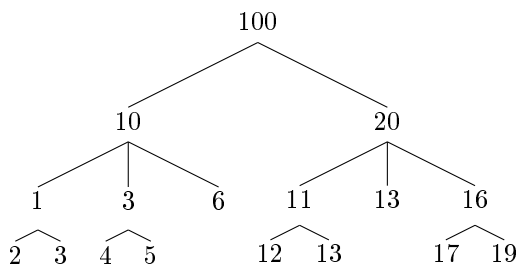
Nota: è indispensabile specificare il tipo e dare una descrizione dichiarativa di ogni funzione ausiliaria utilizzata (anche locale), altrimenti non verrà presa in considerazione (ad eccezione delle funzioni il cui tipo e specifica sono già dati nel testo).

Definizione: dato un albero n-ario t , con nodi di tipo α , e una lista associativa $\text{assoclist}: (\text{int} \times \alpha \text{ list}) \text{ list}$, diciamo che un ramo di t **rispetta i livelli** di assoclist se ogni nodo del ramo che si trova al livello n dell'albero (cioè ogni nodo in posizione n nella lista che rappresenta il ramo) appartiene alla lista associata a n in assoclist , se assoclist contiene una coppia avente n come primo elemento (altrimenti va bene qualsiasi nodo).

Si consideri ad esempio l'albero sotto rappresentato.

- L'unico ramo che rispetta i livelli della lista $[(2, [2; 16; 4]); (1, [10; 20; 40])]$ è $[100; 20; 16; 17]$: non ci sono vincoli per la radice (livello 0, cioè posizione 0 nella lista), né per quelli di livello maggiore di 2; il nodo di livello 1, cioè 20, appartiene alla lista associata a 1 ($[10; 20; 40]$) e il nodo di livello 2 (16) appartiene a $[2; 16; 4]$.
- Se listanodi è una lista che contiene 100, allora qualsiasi ramo rispetta i livelli di $[(0, \text{listanodi})]$, altrimenti nessuno.

- Non esiste alcun ramo che rispetti i livelli di $[(1, [100; 20; 30]); (2, [1; 3; 4])]$: l'unico ramo che ha a livello 1 un nodo di $[100; 20; 30]$ è quello di destra, ma nel sottoalbero di destra nessun ramo ha un nodo in $[1; 3; 4]$ a livello 2.



Si noti che la definizione non pone alcun vincolo sulla lunghezza del ramo: non è necessario che il ramo contenga nodi di livello n per ogni n al quale assoclist associa un valore. Per cui, ad esempio, il ramo $[100; 10; 6]$ rispetta i livelli di $[(1, [10; 20; 40]); (3, [200; 60; 50])]$, anche se non ha nodi di livello 3. In altri termini, i vincoli si applicano ai nodi del ramo, non ai livelli rappresentati nella lista.

1. Scrivere una funzione

good_node: $\alpha \rightarrow \beta \rightarrow (\alpha \times \beta \text{ list}) \text{ list} \rightarrow \text{bool}$,
 tale che `good_node level x assoclist = true` in 2 casi: o x appartiene alla lista associata a `level` in assoclist , oppure assoclist non associa a `level` alcun valore.

2. Definire un tipo di dati $\alpha \text{ tree}$ per la rappresentazione di alberi n-ari e scrivere una funzione

path: $\alpha \text{ tree} \rightarrow (\text{int} \times \alpha \text{ list}) \text{ list} \rightarrow \alpha \text{ list}$,
 tale che `path t assoclist` riporti, se esiste, un cammino dalla radice a una foglia dell'albero che rispetti i livelli di assoclist ; se un tale cammino non esiste, la funzione solleverà un'eccezione.