

Definizione degli alberi n-ari per induzione strutturale

- 1 L'albero costituito da un unico nodo r senza genitori è un albero, con radice r ;
- 2 Se t_1, t_2, \dots, t_n sono alberi, con radici, rispettivamente, r_1, r_2, \dots, r_n , e r è un nuovo nodo, allora la struttura che si ottiene aggiungendo il nodo r come genitore di r_1, r_2, \dots, r_n è un albero;
- 3 Nient'altro è un albero

Rappresentazione

```
type 'a ntree = Tr of 'a * 'a ntree list
```

- Una foglia etichettata da **n** è rappresentata da **Tr(n,[])** (non ha sottoalberi)

```
(* leaf: 'a -> 'a ntree *)  
let leaf x = Tr(x, [])
```

- L'albero con radice **n** e sottoalberi **t1,...,tn** è rappresentato da **Tr(n,[t1;...;tn])**

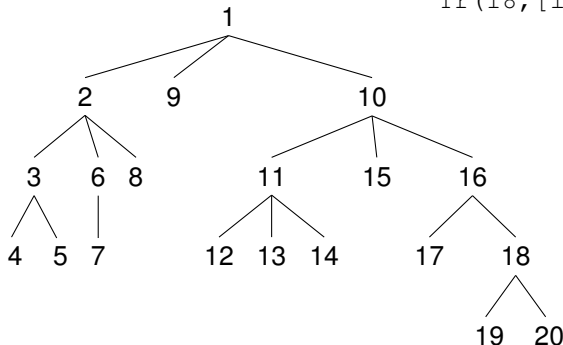
La definizione è **mutuamente ricorsiva**:

- **'a ntree** definito in termini di **'a ntree list**
- **'a ntree list** definito in termini di **'a ntree**

Esiste l'albero vuoto?

Esempio

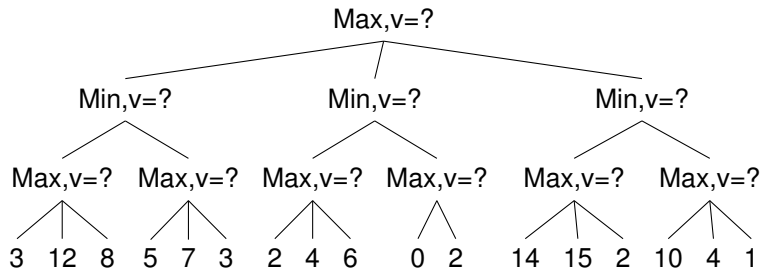
```
let t = Tr(1,[Tr(2,[Tr(3,[leaf 4; leaf 5]);  
                Tr(6,[leaf 7]);  
                leaf 8]));  
        leaf 9;  
        Tr(10,[Tr(11,[leaf 12; leaf 13; leaf 14]);  
              leaf 15;  
              Tr(16,[leaf 17;  
                    Tr(18,[leaf 19; leaf 20])])])])])
```



Alberi di minimax: esempio

A ogni foglia è assegnato un valore (utilità per Max della situazione corrispondente).

Il giocatore Min cerca di minimizzare l'utilità, Max di massimizzarla.



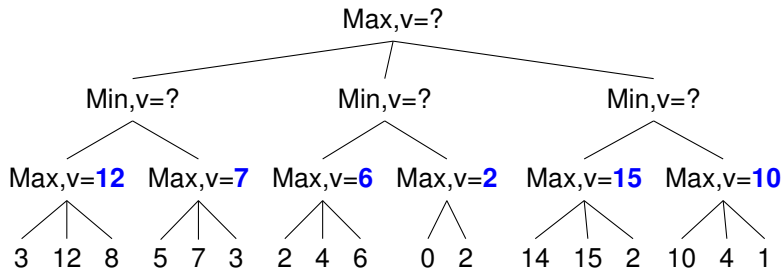
I valori si propagano verso l'alto

- Max massimizza il valore dei figli

Alberi di minimax: esempio

A ogni foglia è assegnato un valore (utilità per Max della situazione corrispondente).

Il giocatore Min cerca di minimizzare l'utilità, Max di massimizzarla.



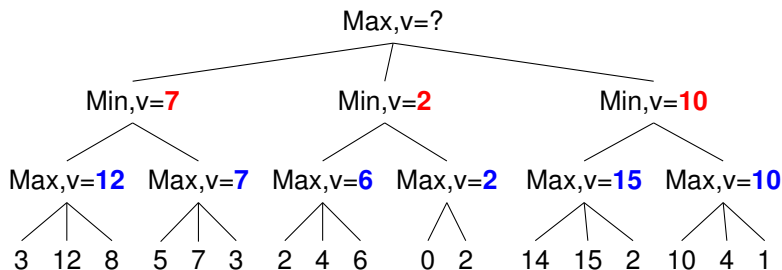
I valori si propagano verso l'alto

- Max massimizza il valore dei figli
- Min minimizza il valore dei figli

Alberi di minimax: esempio

A ogni foglia è assegnato un valore (utilità per Max della situazione corrispondente).

Il giocatore Min cerca di minimizzare l'utilità, Max di massimizzarla.



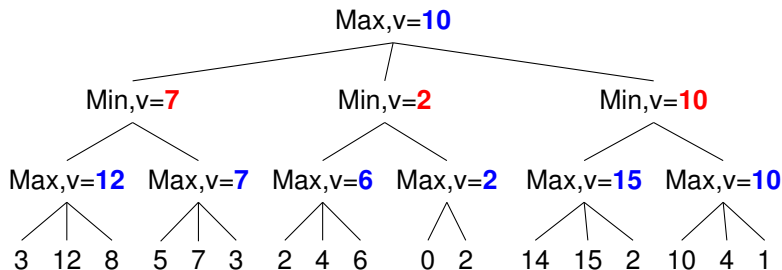
I valori si propagano verso l'alto

- Max massimizza il valore dei figli
- Min minimizza il valore dei figli
- Max massimizza il valore dei figli

Alberi di minimax: esempio

A ogni foglia è assegnato un valore (utilità per Max della situazione corrispondente).

Il giocatore Min cerca di minimizzare l'utilità, Max di massimizzarla.



I valori si propagano verso l'alto

- Max massimizza il valore dei figli
- Min minimizza il valore dei figli
- Max massimizza il valore dei figli: sceglierà la mossa di destra

Dal compito d'esame di Luglio 2013

Un albero di minimax è un albero n-ario le cui foglie sono etichettate da interi, e i nodi intermedi sono etichettati da una coppia di elementi, costituita da uno dei due valori *Min* o *Max* e da un intero.

L'intero è detto **valore numerico del nodo** (intermedio o foglia).

Rappresentiamo gli alberi di minimax mediante le seguenti dichiarazioni di tipo:

```
type player = Min | Max
type minmaxtree = Leaf of int
                | Node of (player * int) * minmaxtree list
```


Alberi di minimax (II)

I valori numerici dei nodi di un albero di minimax sono **ben assegnati** se:

- per ogni nodo intermedio etichettato da (Max, n) , n è il massimo valore numerico dei figli del nodo;
- per ogni nodo intermedio etichettato da (Min, n) , n è il minimo valore numerico dei figli del nodo.

(le foglie possono avere qualsiasi valore numerico).

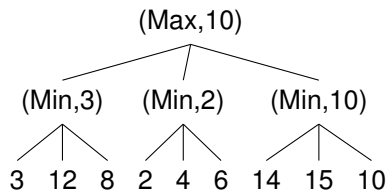
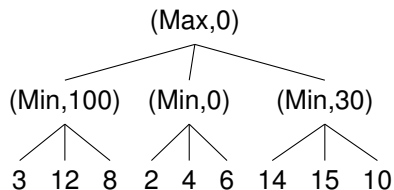
Scrivere un programma con una funzione

propagate: minmaxtree -> minmaxtree,

che, applicata a un qualsiasi albero T di minimax, restituisca l'albero che si ottiene da T sostituendo (eventualmente) i valori dei nodi intermedi in modo che siano ben assegnati.

Esempio

Dall'albero rappresentato a sinistra (i cui valori non sono ben assegnati) si otterrà quello rappresentato a destra:



Vai al codice