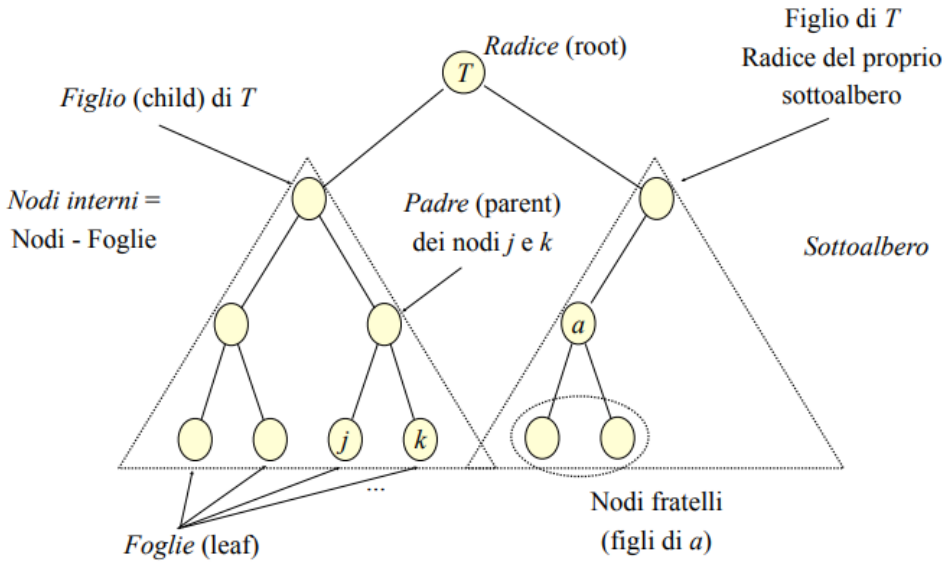
ALBERI

NOMENCLATURA

Albero: Struttura dati dinamica non lineare in quanto ogni elemento (nodo) può avere un successore (discendente)
Radice: primo elemento di un albero
Figli: discendenti diretti di un nodo (**padre**)
Fratelli: nodi con lo stesso padre
Foglia: nodo da cui non discende da nessun altro nodo
Sottoalbero: ogni nodo che non è la radice è la radice di un albero contenuto nell'albero dato
Alberi n-ari: possono avere un numero qualsivoglia di figli per ciascun nodo.

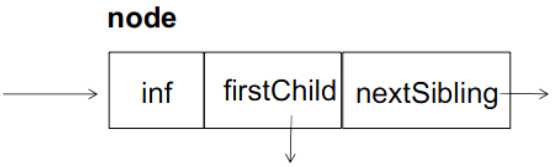


Alberi binari: possono avere 0, 1, o al più 2 figli per ciascun nodo

IMPLEMENTAZIONE PUNTATORI PRIMO-FIGLIO/FRATELLO

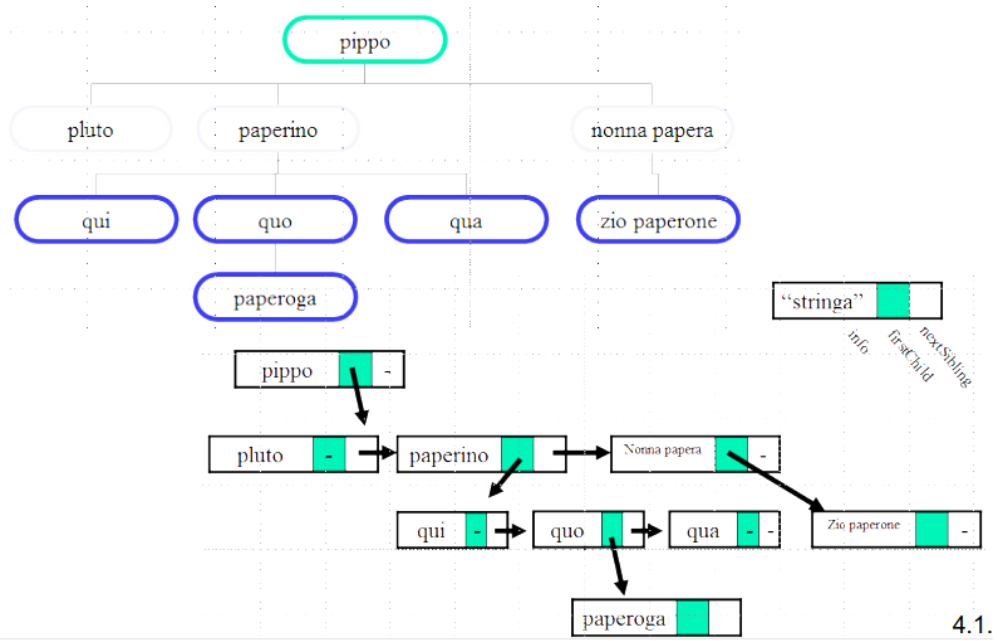
Ciascun elemento (node) contiene

- un campo informativo
- un puntatore al primo figlio (firstChild)
- un puntatore al fratello successivo (nextSibling)



Le *foglie* hanno valore NULL nel campo firstChild

ESEMPIO



Alcune osservazioni:

- Non c'è da nessuna parte un rimando al padre pertanto supporta l'esplorazione dell'albero dalla radice alle foglie
- Se si vuole navigare l'albero in ordine inverso bisogna aggiungere un puntatore al padre (parent).

STRUCT "NODE" E IL TIPO DI DATO "TREE"

```
struct node {
   tipo_inf inf;
   node* parent; //opzionale
   node* firstChild;
   node* nextSibling;
};

typedef node* tree; //punta alla radice
                    dell'albero
tree root; //variabile di tipo tree
```

PRIMITIVE

- `node* new_node(tipo_inf i)`: crea un nuovo nodo con valore informativo i.
- `void insert_child(tree p, tree c)`: aggiorna p inserendo il sottoalbero radicato in c come primo figlio di p
- `void insert_sibling(node* n, tree t)`: aggiorna n inserendo il sottoalbero radicato in t come primo figlio di n

Nelle primitive `insert_child` e `insert_sibling` il primo parametro viene passato per valore e non per riferimento (anche se dentro la funzione è necessario l'aggiornamento del parametro) perché viene modificato l'oggetto puntato dal parametro formale e non il valore del puntatore.

- `tipo_inf get_info(node* n)`: restituisce il contenuto informativo del nodo n
- `node* get_parent(node* n)`: Restituisce il padre del nodo n
- `node* get_firstChild(node* n)`: restituisce il primo figlio del nodo n, se esiste
- `node* get_nextSibling(node* n)`: restituisce il fratello successive del nodo n, se esiste