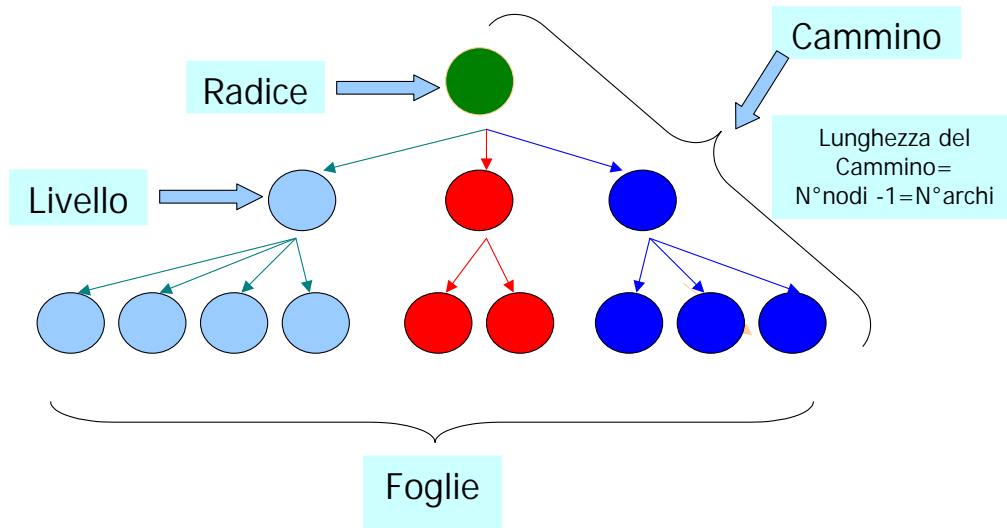


## Linearizzazione di alberi

### Il problema

Si supponga di avere un albero n-ario:



Conservare tale struttura non lineare, nella memoria centrale o di massa di un calcolatore, richiede la sua **trasformazione in una struttura lineare**, una lista contenente i nodi dell'albero in un ordine pre-determinato.

Il problema enunciato ricorre in tutti i casi di memorizzazione di alberi. Un caso interessante di alberi è quello degli alberi binari ed, in particolare, degli alberi binari che descrivono le espressioni aritmetiche. La linearizzazione di tali alberi porta a quella che è nota come la rappresentazione in **notazione polacca** di espressioni aritmetiche.

### La trasformazione di un albero in struttura lineare

I metodi esaminati sono:

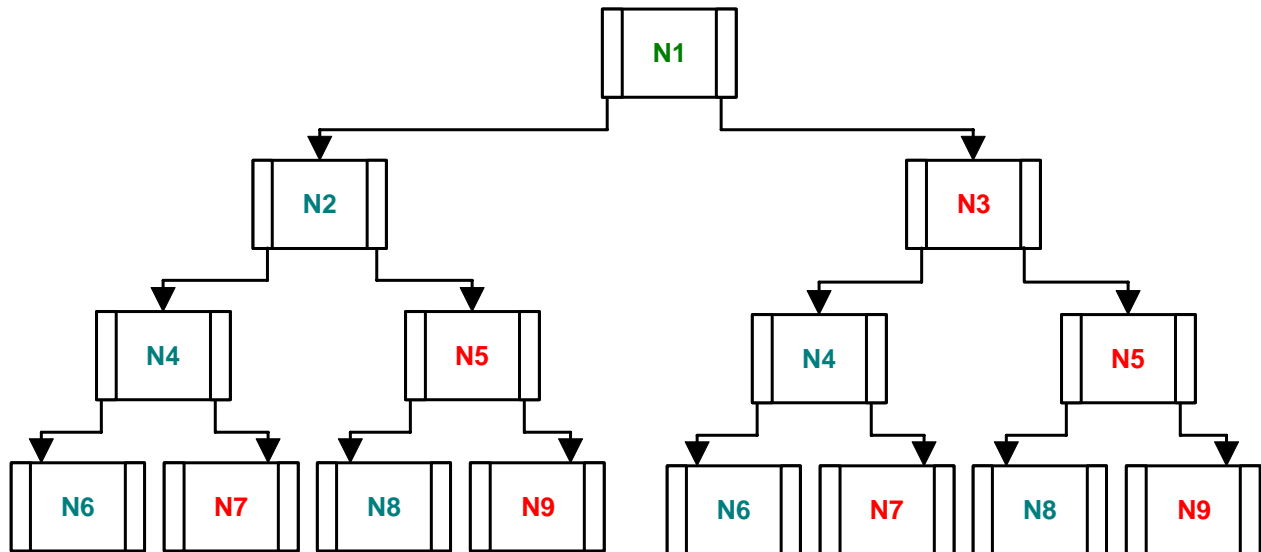
- *linearizzazione di alberi binari;*
- *linearizzazione di alberi n-ari.*

### La linearizzazione di alberi binari

La rappresentazione, attraverso una struttura di dati lineare (lista), di una struttura non lineare quale un albero binario si basa sull'aggiunta di due puntatori al contenuto di ogni nodo: il primo indica l'indirizzo del figlio di sinistra ed il secondo indica l'indirizzo del figlio di destra. Nel caso di nodi finali (foglie) o con un solo figlio, l'indirizzo relativo è posto "a massa", attraverso l'assegnazione di un valore *null*.

Si noti come l'aggiunta dei 2 puntatori implica l'accrescimento dell'occupazione di memoria di ciascun nodo.

Nella figura seguente viene riportata la rappresentazione grafica a puntatori di un albero binario.



Va da sé che la visita di alberi binari che adottino la tecnica di rappresentazione anzidetta consista in un procedimento ricorsivo che prevede alternativemente:

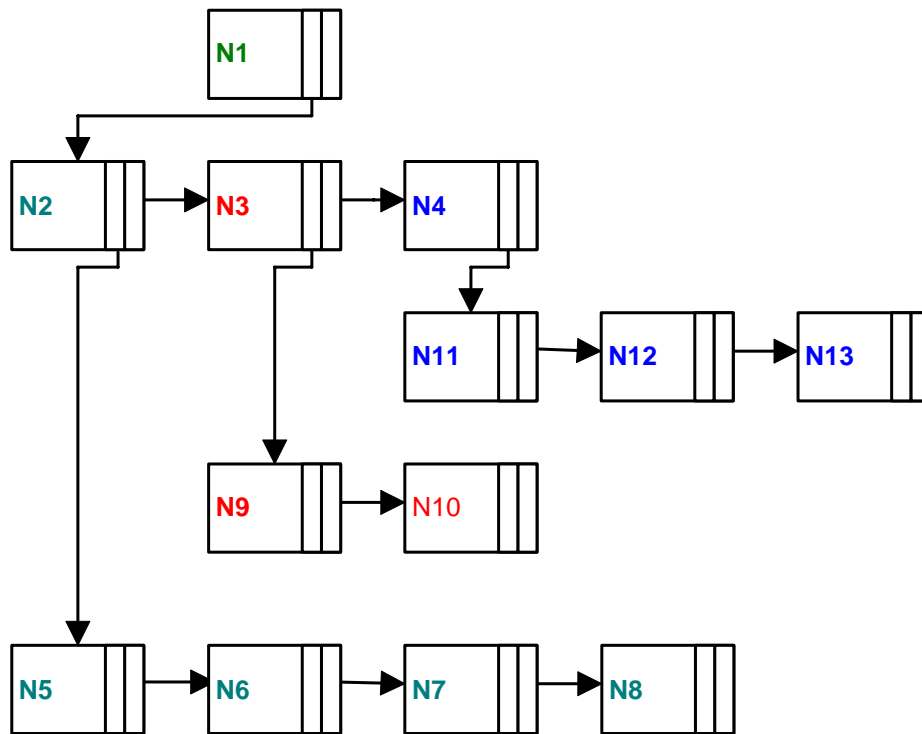
- la visita della radice dell'albero, poi quella ordinata di uno e l'altro dei figli (*pre-ordine*); se il figlio visitato per primo è il sinistro si dirà *pre-ordine sinistro*, mentre se il figlio visitato per primo è il destro si dirà *pre-ordine destro*;
- la visita ordinata di uno e l'altro dei figli, poi quella della radice dell'albero (*post-ordine*); se il figlio visitato per primo è il sinistro si dirà *post-ordine sinistro*, mentre se il figlio visitato per primo è il destro si dirà *post-ordine destro*;
- la visita ordinata di uno dei figli, poi quella della radice dell'albero e quindi quella dell'altro figlio (*in-ordine*); se il figlio visitato per primo è il sinistro si dirà *in-ordine sinistro*, mentre se il figlio visitato per primo è il destro si dirà *in-ordine destro*;

### La linearizzazione di alberi n-ari

Quando il grado di uscita massimo di un albero, cioè il numero massimo di figli, è limitato (ad esempio ogni nodo non ha più di 4 figli) e non varia molto da nodo a nodo, è immediato estendere le tecniche utilizzate per gli alberi binari ad alberi ternari, quaternari, ecc.

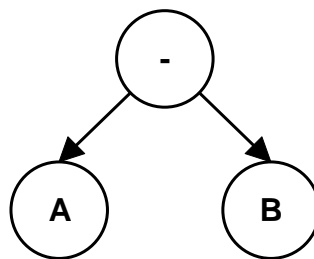
Se però il numero di figli di un nodo è altamente variabile (p.e. da 0 a 100) o non limitato, le tecniche utilizzate per gli alberi binari determinerebbero spreco. Vengono pertanto adoperate tecniche che consentono di rendere dinamicamente variabile il numero di figli di ogni nodo. Ciò può avvenire in vari modi. Uno dei più semplici consiste nell'associare ad ogni nodo un puntatore al primo figlio della lista dei propri figli ed un altro puntatore al proprio fratello di destra.

Anche in questo caso la visita di alberi, che adottino la tecnica di rappresentazione descritta, consiste in un procedimento ricorsivo che tiene conto della funzione dei due puntatori.



### La notazione polacca

Una significativa applicazione degli alberi binari è costituita dalla rappresentazione di una generica espressione aritmetica. Infatti quest'ultima può essere descritta graficamente mediante un albero che ricorsivamente riporti nella radice un operatore aritmetico e nei due figli il primo ed il secondo operando rispettivamente. Nella figura sottostante l'albero rappresenta l'espressione  $(A - B)$  oppure  $(B - A)$ , a seconda che l'ordine di visita dell'albero preveda prima la visita del figlio sinistro o quella del figlio destro.



Applicando la tecnica di attraversamento in *pre-ordine*, tale rappresentazione grafica corrisponde alla stringa di simboli  $-AB$  oppure  $-BA$  a seconda che si adoperi il *pre-ordine sinistro* o il *pre-ordine destro* rispettivamente. Le due stringhe vengono dette **notazione polacca pre-fissa sinistra** e **notazione polacca pre-fissa destra** rispettivamente.

Applicando la tecnica di attraversamento in *post-ordine*, tale rappresentazione grafica corrisponde alla stringa di simboli  $AB-$  oppure  $BA-$  a seconda che si adoperi il *post-ordine sinistro* o il *post-ordine destro* rispettivamente. Le due stringhe vengono dette **notazione polacca post-fissa sinistra** e **notazione polacca post-fissa destra** rispettivamente.

Infine, usando la tecnica di attraversamento *in-ordine*, tale rappresentazione grafica corrisponde alla stringa di simboli  $A-B$  oppure  $B-A$  a seconda che si adoperi l'*in-ordine sinistro* o l'*in-ordine*

*destro* rispettivamente. Le due stringhe vengono dette **notazione polacca in-fissa sinistra** e **notazione polacca in-fissa destra** rispettivamente.

Le notazioni polacche pre-fisse e post-fisse si prestano, come vedremo immediatamente, alla memorizzazione, in forma non ambigua, delle espressioni aritmetiche in un calcolatore. La notazione in-fissa, invece, soffre di ambiguità interpretativa e, pertanto, richiedendo ulteriori specifiche, viene meno frequentemente adoperata.

Le notazioni polacche pre-fisse e post-fisse possono consentire la esecuzione dell'espressione aritmetica a cui equivalgono se, percorrendo la stringa da sinistra verso destra, ogni operatore incontrato viene ricorsivamente applicato, rispettivamente, ai due operandi che immediatamente lo precedono o seguono. Tali due operandi sono da intendersi il sinistro e poi il destro se la notazione è sinistra, il destro e poi il sinistro se la notazione è destra.