1

Capitolo D30
arborescenze

Contenuti delle sezioni

26 pagine

D300.01 *In questo capitolo introduciamo le caratteristiche fondamentali dei particolari digrafi chiamati arborescenze. Queste caratteristiche si possono ritrovare in una grande varietà di configurazioni e in molti processi studiati in ambiti danche molto diversi.*

Questo permette di ricondurre ad arborescenze e a loro arricchimenti molti argomenti della matematica discreta e delle sue applicazioni computazionali che altrimenti verrebbero sviluppati in modo inutilmente dispersivo.

In effetti ricondursi alle arborescenze, oltre a permettere di servirsi di schemi visivi chiarificanti, consente di impostare efficaci procedimenti computazionali.

D30 a. arborescenze [1]

D30a.01 Ricordiamo che per **arborescenza** si intende un digrafo dotato di radice, cioè di un nodo dal quale sono raggiungibili tutti gli altri, digrafo tale che ogni altro suo nodo è raggiungibile dalla radice mediante un unico cammino.

Per tali digrafi si usa anche il termine di albero con radice (*rooted tree*).

//input pD30a01

Nel seguito denotiamo con **Arb** la classe delle arborescenze e con Arb_S l'insieme delle arborescenze con i nodi appartenenti all'insieme S.
Le proprietà generali delle arborescenze si riconoscono facilmente.

Si nota innanzi tutto come la scelta della radice implichi l'orientazione di ciascuno dei collegamenti tra due nodi.

D30a.02 Prop. Un'arborescenza è un digrafo connesso.

Dim.: Infatti si può passare da un qualsiasi nodo p a un altro qualsivoglia q percorrendo a ritroso il cammino dalla radice a p e successivamente seguendo il cammino dalla radice a q ∎

Osserviamo che la precedente procedura, come accade a molte altre riguardanti strutture grafiche, è facile da descrivere, ma poco efficiente all'esecuzione.

D30a.03 Prop. Un'arborescenza non presenta circuiti.

Dim.: In caso contrario un nodo p che si trova su un circuito, essendo raggiungibile dalla radice con un dato cammino, lo sarebbe anche attraverso il cammino ottenuto prolungando il precedente con gli archi che formano il circuito a iniziare da p ∎

//input pD30a03

D30a.04 Prop. La radice di un'arborescenza non ha alcun predecessore ed è unica.

Dim.: Se un'arborescenza avesse due radici diverse, queste verrebbero a trovarsi su un circuito.
Se la radice avesse un predecessore p, si avrebbe il circuito costituito dal cammino che va dalla radice a p e dall'arco da p alla radice. ∎

D30a.05 Prop. In un nodo q di un'arborescenza diverso dalla radice entra uno e un solo arco.

Dim.: Il nodo q deve presentare un arco entrante, quello conclude il cammino dalla radice allo stesso q; se vi fossero due archi entranti $\langle p_1, q\rangle$ e $\langle p_2, q\rangle$, si potrebbe raggiungere q dalla radice sia attraverso il cammino ottenuto aggiungendo $\langle p_1, q\rangle$ al cammino dalla radice a p_1, sia attraverso il cammino ottenuto aggiungendo $\langle p_2, q\rangle$ al cammino dalla radice a p_2 ∎

D30a.06 Prop. In un'arborescenza non vi possono essere semicammini chiusi che non siano cammini.

Dim.: Una tale sequenza di archi, infatti, dovrebbe contenere almeno una coppia di archi della forma $\langle p_1, q\rangle$ e $\langle p_2, q\rangle$ e quindi il nodo q sarebbe raggiungibile dalla radice da più cammini ∎

//input pD30a06

Dunque in un'arborescenza si ha un unico nodo, la radice, privo di predecessori, mentre i nodi rimanenti sono dotati di uno e un solo predecessore: questi si dicono **nodi figli** e //a06 l'unico predecessore di un nodo q figlio si dice **padre del nodo** q. //a06

D30a.07 Prop. Un'arborescenza possiede n nodi sse presenta $n-1$ archi, ovvero sse presenta $n-1$ nodi figli.

Dim.: Infatti la corrispondenza che a ciascuno degli $n-1$ nodi figli dell'arborescenza associa l'arco che entra in esso è chiaramente biunivoca e coinvolge tutti gli archi del digrafo e tutti i suoi nodi ad eccezione dalla radice ▮

D30a.08 Prop. Un'arborescenza è la riduzione riflessivo-transitiva di un digrafo ordinato e può essere raffigurata senza segnare la direzione degli archi, ma solo rendendo riconoscibile la sua radice.

Dim.: Un'arborescenza, essendo priva di circuiti, è un digrafo privo di cappi e antisimmetrico; inoltre essa è un digrafo antitransitivo, privo di archi ottenibili da altri per transitività, poiché in caso contrario ogni nodo dotato di padre a sua volta dotato di padre avrebbe almeno due archi entranti ▮

Si osservi che la definizione di arborescenza è soddisfatta anche dal digrafo nodo, il digrafo costituito da un solo nodo e privo di archi, dal digrafo di due nodi e un arco che li collega e da ogni catena, digrafo al quale si può dare la forma

$$lggg\{q_0, q_1, ..., q_n\}\ ,\ \{i = 1, 2, ..., n :| \langle q_{i-1}, q_i\rangle\}\Big\rangle\ .$$

D30a.09 Prop. Un'arborescenza possiede almeno un nodo privo di successori.

Dim.: Se così non fosse si avrebbero cammini di lunghezza grande a piacere, e quindi, essendo finito l'insieme dei nodi, si dovrebbero avere cammini che toccano più volte un nodo, cioè circuiti, evenienza esclusa in a03 ▮

Conviene segnalare che hanno grande interesse anche quelle che chiamiamo **arborescenze.N** e che introduciamo in :f relazioni il cui terreno è costituito da un insieme numerabile nel quale si ha una radice dalla quale tutti gli altri nodi sono raggiungibili con un solo cammino.

D30a.10 Un nodo privo di discendenti si dice **foglia** o **nodo terminale**. Si hanno quindi due bipartizioni dell'insieme dei nodi di un'arborescenza non ridotta al digrafo nodo: la bipartizione riguardante la radice ed ai nodi dotati di predecessore e quella che distingue tra nodi padre e nodi foglia.

Per un'arborescenza $\Psi = \langle Q, U\rangle$ si può denotare con $Root(\Psi)$ la sua radice, con $Fath(\Psi) := Q \,_{\mathsf{R}}\, U^{\top}$ l'insieme dei nodi padre e con $Leav(\Psi) := Q \setminus Fath(T)$ l'insieme delle sue foglie.
Se $q \in Q \setminus Root(\Psi)$ denoteremo suo padre con $fath(q)$, cioè scriviamo $fath(q) := q \,_{\mathsf{R}}\, U^{\top}$.

//input pD30a10

D30a.11 Due nodi che discendono da uno stesso padre si dicono essere **nodi fratelli** l'uno dell'altro. Con atteggiamento esclusivamente patrilineare si parla anche di **nodi abiatici** (figli dei figli), di **nodi nonni** (padri dei padri), di **nodi cugini** (discendenti di secondo livello dallo stesso nonno), di **nodi nipoti** (per evitare ambiguità conviene usare questo termine solo per i figli dei fratelli e non per i figli dei figli). L'insieme di tutti i nodi figli di uno stesso padre q lo chiamiamo **prole del nodo** q.

Evidentemente l'essere nodi fratelli e l'essere nodi cugini sono relazioni simmetriche e antiriflessive; l'essere fratelli è anche transitiva, come l'essere fratelli o cugini, ma in generale non è transitiva la relazione essere cugini.

//input pD30a11

D30a.12 Dunque i termini adottati per le arborescenze provengono in parte dalla genealogia e in parte dalla descrizione della topologia delle piante, due settori nei quali questi digrafi da tempo hanno fornito schematizzazioni semplici ed efficaci.

Potrebbero infastidire le raffigurazioni che stiamo dando con radici in alto e foglie in basso; queste raffigurazioni sono invece in sintonia con il termine discendente usato per le genealogie.

Le raffigurazioni qui prevalenti le chiamiamo del tipo alto-basso: tutti i collegamenti con nodi discendenti vanno verso posizioni inferiori.
Per talune applicazioni invece può essere preferibile una delle tre raffigurazioni varianti del tipo prevalente, cioè uno dei tipi basso-alto, sinistra-destra e destra-sinistra.

D30a.13 In un'arborescenza Ψ si ha una corrispondenza biunivoca tra nodi e cammini che hanno la radice come estremità iniziale.
Quindi i nodi di un'arborescenza si ripartiscono in vari livelli nel seguente modo:

– Al livello 0 poniamo la radice;
– al livello 1 poniamo i suoi discendenti (figli della radice);
– al livello 2 poniamo i discendenti dei nodi di livello 1 (abiatici della radice);
– . . . . . .

Un'arborescenza è quindi la riduzione irriflessiva e intransitiva di un digrafo graduato.

Si definisce **altezza di un'arborescenza** il massimo dei livelli attribuiti ai suoi nodi.

Può essere utile per un'arborescenza Ψ denotare con $Ord(\Psi)$ la relazione d'ordine ottenuta dalla Ψ mediante chiusura riflessivo-transitiva, ossia la relazione d'ordine della quale la Ψ è la riduzione irriflessiva e intransitiva.

D30a.14 Un'arborescenza, come ogni digrafo acircuitale, possiede **cammini massimali**, cioè cammini che non possono essere estesi ad altri cammini: si tratta evidentemente dei cammini che vanno dalla radice a un nodo senza figli, ossia a una sua foglia.
I cammini massimali di un'arborescenza sono dunque in corrispondenza biunivoca con le sue foglie.

L'altezza di un'arborescenza è la massima lunghezza dei suoi cammini, cioè la massima lunghezza dei suoi cammini massimali. Un'arborescenza di altezza h si dice anche **arborescenza-h**.

Le arborescenze-1 sono dette anche **arborescenze elementari** e sono costituite solo da una radice e dai suoi figli.
A meno di isomorfismi, le arborescenze-1 si possono caratterizzare completamente mediante il numero dei figli: 1, 2, 3, ... o, equivalentemente, con il numero dei nodi 2, 3, 4, ...

//input pD30a14

Il digrafo nodo si può considerare come una arborescenza-0.

Prop. Ogni sottodigrafo S di una arborescenza Ψ che sia debolmente connesso è una sua sottoarborescenza.

Dim.: Preso un nodo qualsiasi ν di S, grazie alla debole connessione, i nodi di S in parte sono suoi discendenti, in parte suoi ascendenti fino a giungere a un Q privo di ascendenti in S. I nodi rimanenti, per la debole connessione, devono essere discendenti di Q e quindi questo nodo è radice di tutti i rimanenti di S ▮

D30a.15 Si dice **sottoarborescenza di un digrafo** un suo sottodigrafo che sia una arborescenza. Un digrafo in genere possiede molte sottoarborescenze; alcune di esse sono facili da trovare, ma individuare la loro totalità può essere oneroso e possono rendersi necessari algoritmi specifici.

Evidentemente le sottoarborescenze di un digrafo sono da ricercare all'interno di ciascuna delle sue componenti connesse

Due tipi di sottoarborescenze facili da individuare sono i sottocammini (privi di nodi ripetuti) e le sottoarborescenze di altezza 1 ottenute da un nodo e da tutti gli archi che vanno da esso a un nodo diverso.

Evidentemente le sottoarborescenze delle sottoarborescenze di un digrafo D sono sottoarborescenze di D.

Si dice **sottoarborescenza massimale di un digrafo** una sottoarborescenza che non può estendersi con l'aggiunta di qualche altro arco.

Dall'osservazione precedente segue che le sottoarborescenze massimali (delle componenti connesse) di un digrafo sono le più interessanti, in quanto tutte le altre possono ottenersi da queste mediante eliminazioni di singoli archi.

Le sottoarborescenze sono sottodigrafi utili in molte applicazioni; in particolare può essere utile individuare le sottoarborescenze che permettono di raggiungere da un nodo radice del digrafo connesso tutti gli altri suoi nodi.

Sulla ricerca delle sottoarborescenze di un digrafo ritorneremo in seguito. Per ora ci limitiamo agli esempi della figura che segue e a osservare che le sottoarborescenze non possono contenere cappi e non possono toccare diverse componenti connesse del digrafo: quindi la ricerca delle sottoarborescenze conviene studiarla su digrafi connessi e privi di cappi.

//input pD30a15

D30a.16 Sulle sottoarborescenze di una arborescenza si possono fare considerazioni un po' più stringenti.

Tra le sottoarborescenze di un'arborescenza Ψ sono ben evidenti le elementari: esse sono in corrispondenza biunivoca con i nodi padre di Ψ e si ripartiscono naturalmente secondo i livelli ai quali si trovano i rispettivi padri.

In ogni arborescenza si trova almeno un padre i cui figli sono tutti foglie, cioè una sottoarborescenza-1 costituita da un padre e dalla sua prole. Se da un'arborescenza non elementare si elimina una prole costituita da sole foglie (il cui numero chiamiamo s), si ottiene una nuova arborescenza avente il numero dei padri diminuito di 1 e il numero delle foglie diminuito di $s-1$.

Un'arborescenza si può decomporre progressivamente eliminando a ogni passo una prole costituita solo da foglie fino a rimanere con la arborescenza-1 formata dalla radice e dai suoi figli.

//input pD30a16

D30a.17 È utile considerare anche il processo inverso consistente nella progressiva costruzione di un'arborescenza che inizia con la arborescenza-1 formata dalla sua radice e dai figli di questa e prosegue con fasi in ciascuna delle quali viene aggiunta una nuova arborescenza-1 U, mediante la fusione di una foglia dell'arborescenza in crescita con la radice di U.

Con un processo di questo genere si può costruire una qualsiasi arborescenza.

Quindi le arborescenze-1 astratte caratterizzate dai diversi numeri di figli si possono vedere come le componenti elementari di ogni arborescenza, come i "mattoni" con i quali si può "edificare" ogni arborescenza.

D30a.18 Sopra le arborescenze si possono operare anche decomposizioni e composizioni che non riguardano solo arborescenze elementari.

Consideriamo una qualsiasi arborescenza Ψ di altezza maggiore o uguale a 2 ed un qualsiasi suo nodo ν che conviene pensare diverso dalla radice e dalle foglie.
Il sottodigrafo ottenuto eliminando da Ψi nodi discendenti di ν è una arborescenza che chiamiamo **sottoarborescenza dei nondiscendenti** di ν.
Un sottidigrafo leggermente diverso (con un nodo in meno) si ottiene eliminando da Ψ l'arco entrante in ν, ν e i suoi discendnti.4
Per queste operazioni di riduzione di una arborescenza si parla anche di **potatura dell'arborescenza**.

Anche il sottodigrafo ottenuto eliminando da Ψ tutti i nodi diversi da ν e dai suoi nodi discendenti è una arborescenze che viene detta **sottoarborescenza dei discendenti** di ν. Evidentemente questa arborescenza ha ν come radice.

Si dice **sottoarborescenza discendente** di una data arborescenza ogni sua sottoarborescenza costituita dai discendenti di un suo nodo.

Le sottoarborescenze ottenute riducendole a un suo nodo e ai suoi discndenti sono in biiezione con i suoi nodi padre (diversi dalla radice se ci liitiamo alle sottoarborescenze proprie).

D30a.19 Poniamoci dal punto di vista opposto. Due arborescenze qualsiasi Ψ_1 e Ψ_2 si possono comporre confondendo una foglia f della prima con la radice della seconda. Questa operazione verrà detta **innesto dell'arborescenza** Ψ_2 sulla foglia f di Ψ_1.

Si osservi che questa composizione è un'operazione ternaria, in quanto è definita assegnando due arborescenze arbitrarie e una qualsiasi foglia della prima.

A questo punto si può tirare una conclusione.

(1) Prop.: Ogni arborescenza si può ottenere innestando opportunamente un certo numero di arborescenze-1. Se si utilizzano m arborescenze-1 che complessivamente presentano F foglie si ottiene un'arborescenza con F archi, $F+1$ nodi, m padri ed $F-m+1$ foglie.

Questo enunciato si dimostra facilmente per induzione ∎

Da una arborescenza si possono ottenere sottoarborescenze anche agendo su loro archi singoli o su insiemi di archi mutuamente noncomparabili che si possono pensare costituire insiemi di taglio.
All'opposto possono essere interessanti composizioni di arborescenze che vengono collegate con nuovi archi che vanno da una arborscenza "leader" a radici di arborescenze che vengono "appese" sotto di essa.

D30a.20 Si dice **arborescenza uniforme-**f o anche **arborescenza satura-**f un'arborescenza ottenibile per innesto di arborescenze-1 tutte con lo stesso numero f di figli e quindi tutte isomorfe.

Evidentemente ogni arborescenza-1 può considerarsi un'arborescenza uniforme; inoltre un'arborescenza è uniforme-f sse ogni suo nodo padre possiede f figli, ovvero se ha tutti i nodi padre con grado uscente pari ad f.

Le arborescenze uniformi-2 sono dette anche **arborescenze binarie**.

Diciamo **arborescenza-CMLU** un'arborescenza i cui cammini massimali hanno la stessa lunghezza. Un'arborescenza è una arborescenza CMLU sse tutte le sue foglie sono allo stesso livello, pari alla lunghezza comune di tutti i suoi cammini massimali.

//input pD30a20

D30a.21 Sono molto particolari le arborescenze CMLU uniformi.

//input pD30

(1) Prop.: Un'arborescenza CMLU uniforme-f di altezza h presenta $1 + f + f^2 + ... + f^h = \dfrac{f^{h+1}-1}{f-1}$ nodi; di questi $1 + f + f^2 + ... + f^{h-1} = \dfrac{f^h-1}{f-1}$ sono padri ed f^h foglie ∎

In particolare le arborescenze binarie CMLU di altezza h presentano $2^{h+1} - 1$ nodi, $2^h - 1$ padri e 2^h foglie.

D30a.22 Si dice **arborescenza uniforme sui livelli**, un'arborescenza tale che i nodi di uno stesso livello posseggono tutti lo stesso numero di figli. In particolare si considerano le arborescenze CMLU uniformi sui livelli.
Queste, a meno di isomorfismi, sono individuate dall'altezza h e dalla h-upla $\langle d_0, d_1, ..., d_{h-1}\rangle$ dei gradi uscenti dei padri ai successivi livelli 0, 1, ..., $h-1$. Si parla quindi di arborescenze uniformi-$\langle d_0, d_1, ..., d_{h-1}\rangle$ sui livelli. Per esempio l'arborescenza CMLU $\langle 4, 2, 3\rangle$-uniforme è rappresentata da:

//input pD30a22

D30a.23 Consideriamo ora le **anticatene** di un'arborescenza, cioè gli insieme di suoi nodi tali che due di essi non appartengono a uno stesso cammino: si tratta quindi di insiemi di nodi mutuamente noncomparabili rispetto alla relazione d'ordine dell'arborescenza.

Anticatene immediatamente percepibili sono quelle costituite da nodi dello stesso livello.

Vi sono tuttavia anticatene costituite da nodi di livelli diversi.

//input pD30a23

Evidentemente un sottoinsieme nonvuoto di una anticatena è ancora una anticatena.
Quindi tra le anticatene interessano principalmente le massimali, cioè quelle che non possono essere ampliate con l'aggiunta di un ulteriore nodo. Infatti ogni anticatena non massimale si può ottenere da qualche catena massimale attraverso eliminazioni di nodi che possono essere scelti ad arbitrio, anche uno per volta.

Due anticatene "estreme" di un'arborescenza sono date dalla sola radice e dall'insieme delle sue foglie; entrambe sono anticatene massimali.

Altre anticatene evidenti sono quelle fornite da tutti i nodi di un dato livello.
Queste possono essere nonmassimali.
Sono invece massimali le anticatene costituite dai nodi di un livello inferiore o uguale alla minima delle lunghezze dei cammini massimali.

Per un'arborescenza con foglie di livello minore di un certo h non superiore alla sua altezza l'anticatena dei nodi di livello h non è massimale, ma lo diventa con l'aggiunta di foglie di basso livello.

Una anticatena K di un'arborescenza Ψ individua tre tipi di sottoarborescenze di Ψ:
(1) la sottoarborescenza costituita dai nodi dei cammini tra radice e nodi di K, estremità incluse;
(2) h sottoarborescenze dei discendenti dei nodi di K (per ogni foglia si ha solo un grafo nodo).
(3) sottoarborescenze costituite dalle classi di connessione dei nodi che non appartengono alle sottoarborescenze precedenti.

Per esempio nella figura che segue si ha una anticatena di 4 nodi (rappresentati dai pallini) che individua, oltre alla sottoarborescenza di tipo (1) avente 7 nodi, 4 sottoarborescenze di tipo (2) e 2 di tipo (3).

//input pD30a23B

D30a.24

Data un'arborescenza Ψ e una sua anticatena K la sottoarborescenza di tipo (1), ossia l'arborescenza ottenuta eliminando tutti i nodi successori di ciascuno dei nodi della K viene detta anche **sottoarborescenza sovrastante** della Ψ definita dalla K
Ogni sottoarborescenza di tipo (2), ossia ciascuna delle arborescenze formate da un nodo della K e dai suoi discendenti, si chiama anche **sottoarborescenza sottostante** della Ψ.

Per l'anticatena costituita dalla sola radice, la sottoarborescenza sovrastante si riduce al digrafo nodo e si ha una sola sottoarborescenza sottostante coincidente con l'intera arborescenza Ψ.

Per l'anticatena costituita dell'insieme delle foglie di una arborescenze si ha come sottoarborescenza sovrastante l'intera arborscenza di partenza, mentre le sottoarborescenze sottostanti sono i grafi nodo costituiti da una sola foglia.

Si hanno sottoarborescenze (3) sse l'anticatena non è massimale.

Una anticatena nonmassimale K, si può ampliare fino a diventare massimale individuando le sue sottoarborescenze di tipo (3) ed aggiungendo a K le unioni di anticatene massimali di queste sottoarborescenze.
In particolare si può ampliare K con la totalità delle loro foglie, cioè con tutte le foglie che non sono discendenti di nodi appartenenti a K.

D30 b. arborescenze distese

D30b.01 Per praticare il controllo effettivo delle attività che si svolgono in modi osservabili e di quelle svolte da automatismi e governate da procedure computazionali è sostanzialmente indispensabile che gli oggetti materiali e le informazioni che sono coinvolte si presentino ordinati.

Come vedremo anche nei paragrafi seguenti, un gran numero delle accennate attività fanno riferimento ad arborescenze che possono avere migliaia e anche molti milioni di nodi.

Spesso quindi si rende necessario scorrere i nodi di un'arborescenza in una ben definita sequenza, ossia, come si dice, si rende necessario **visitare i nodi di un'arborescenza**.

Sovente risulta necessario fare riferimento a un ordine totale dei nodi delle arborescenze [b04].

In effetti l'esigenza di una visita ordinata si può manifestare per ogni genere di struttura discreta in relazione a necessità concrete.

Per le arborescenze si sanno ottenere vantaggiose sequenzializzazioni dei nodi ampliando l'ordinamento parziale fornito dalla struttura mediante relazioni d'ordine totale delle proli dei nodi padre, cioè imponendo per ogni nodo padre un ordine totale dei figli.

D30b.02 Chiamiamo **arborescenza distesa**, in breve **arborescenza.d** una struttura della forma $\boldsymbol{D} := \langle Q, U, \boldsymbol{F}\rangle$, dove $\langle Q, U\rangle$ è un'arborescenza ed $\boldsymbol{F}$ una relazione che concerne le coppie di nodi che sono fratelli e che ridotta a ciascuna delle proli costituisce un ordine totale. Denotiamo con **ArbD** l'insieme delle arborescenze distese.

Se p e q sono fratelli, $p\boldsymbol{F}q$ si esprime dicendo che p è **nodo che precede un fratello** q.

Alternativamente si dice anche che p è figlio più a sinistra di q; questa terminologia deriva dalla raffigurazione più usuale delle arborescenze distese, quella che abbiamo adottato implicitamente fin dall'inizio e che chiamiamo **raffigurazione alto-basso**.

Secondo questa raffigurazione per una arborescenza distesa elementare la radice è posta in alto, i suoi figli si trovano immediatamente sotto e sono presentati a partire da sinistra nell'ordine dato da $\boldsymbol{F}$. Per una arborescenza composita si procede con il suddetto criterio per gli insiemi ordinati degli elementi di tutte le proli, cioè per tutti i figli dei vari nodi padre.

Per esempio nella seguente figura $b\boldsymbol{F}e$, $c\boldsymbol{F}d$, $f\boldsymbol{F}j$, $g\boldsymbol{F}h$, $h\boldsymbol{F}i$.

//input pD30b02

In un'altra raffigurazione usuale di un'arborescenza.d si chiama ragionevolmente **raffigurazione sinistra-destra**; essa presenta la radice nella posizione più a sinistra, i figli alla destra di ogni padre e ciascuno dei fratelli posto sotto quello che lo precede immediatamente e sopra quello che lo segue immediatamente. Una raffigurazione sinistra-destra si ottiene da una alto-basso con una rotazione antioraria di 90° e con una riflessione rispetto ad una linea orizzontale.

La raffigurazione sinistra-destra si può presentare facilmente con stampe effettuate con le cosiddette "indentazioni". Per esempio la stampa che segue equivale alla figura precedente.

//input pD30b02b

Nella pratica le arborescenze.d e i loro arricchimenti hanno molte più applicazioni delle corrispondenti strutture prive di ordinamenti delle proli. Spesso quindi si adotta una terminologia sbrigativa e le arborescenze distese vengono chiamate arborescenze tout court; in effetti vengono ancor più spesso chiamate alberi, mentre qui questo termine riguarda grafi nonorientati (e aciclici).

D30b.03 Due cammini massimali diversi C e C' sopra un'arborescenza distesa presentano una parte iniziale comune che può ridursi alla sola radice e due parti distinte: si individua quindi una coppia di nodi fratelli, il primo su C il secondo su C', che risultano essere i primi diversi sui due cammini.
Si dice che tra i cammini C precede C' sse il suo primo nodo diverso precede il primo nodo diverso su C'.

Si ha quindi per ogni arborescenza distesa un naturale ordinamento totale dei cammini massimali.
Anche per i cammini massimali si può parlare di **cammino più a sinistra** o di **cammino più a destra** riferendosi alla raffigurazione alto-basso, oppure di **cammino più in alto** o di **cammino più in basso** riferendosi alla raffigurazione sinistra-destra.

Questo ordinamento totale dei cammini massimali di un'arborescenza distesa, ha le caratteristiche dell'ordinamento lessicografico delle sequenze dei nodi che codificano i cammini stessi.

Inoltre in virtù della biiezione tra cammini massimali e foglie, iquesto ordinamento totale nduce un ordinamento totale delle foglie dell'arborescenza distesa.

L'ordinamento dei cammini massimali induce un ordinamento totale naturale anche sui nodi di ciascuno dei livelli: infatti i nodi del livello s si possono considerare le foglie dell'arborescenza distesa ottenuta eliminando tutti i nodi ai livelli superiori ad s.

Evidentemente si ha un ordinamento totale anche per tutti i cammini che iniziano nella radice.

Inoltre si ha un ordinamento totale su una qualsiasi anticatena dell'arborescenza: i nodi di una anticatena, infatti, costituiscono le foglie dell'arborescenza distesa ottenuta come sottoarborescenza superiore definita dalla anticatena, ossia conservando della originale solo i nodi che si trovano sui cammini tra radice e nodi dell'anticatena.

D30b.04 Per molti scopi si rendono necessari meccanismi che consentono di scorrere o "visitare" uno dopo l'altro tutti i nodi di un'arborescenza distesa $\boldsymbol{D} = \langle Q, \Psi, \boldsymbol{F} \rangle$ al fine di effettuare analisi, scelte o conteggi.

Risulta allora necessario presentare i loro nodi secondo un ordine totale definito chiaramente e tale da poter essere utilizzato agevolmente da vari algoritmi.

In genere risulta opportuno che questo ordine totale rispetti l'ordine U^* dall'arborescenza sottostante $\langle Q, U \rangle$ e rispetti l'ordine in ciascuna prole, cioè $\boldsymbol{F}$: si chiede quindi un ordine totale che costituisca un sovrainsieme dell'insieme di coppie di nodi $U^+ \cup \boldsymbol{F}$.
Talora occorre invece rispettare l'ordine opposto di U^+ e/o l'ordine opposto di $\boldsymbol{F}$.

Le arborescenze con migliaia o milioni di nodi che si incontrano nelle applicazioni, possono essere presentate esplicitamente mediante strutture informative oppure implicitamente attraverso regole che permettono di passare da un nodo ai nodi adiacenti; In alcuni di questi casi la visita dei nodi può costituire un problema non facile.

Per effettuare la visita dei nodi di queste arborescenze "impegnative", oltre che a scegliere opportunamente il primo nodo, occorre individuare regole e procedimenti che permettano di passare da un nodo qualsiasi al successivo secondo l'ordine previsto e di segnalare quando questo "avanzamento" non è possibile, in quanto si è raggiunto proprio sull'ultimo nodo da visitare.

D30b.05 Il primo ordine totale che estende $U^+ \cup \boldsymbol{F}$ cui si può pensare è detto **ordine per livelli** e si ottiene considerando primariamente i successivi livelli (radice, suoi figli, suoi abiatici, ...) e secondariamente l'ordine di precedenza dei nodi di ciascun livello.

Accade però che per molte arborescenze impegnative risulta poco agevole operare con un meccanismo di visita secondo questo ordine. Infatti nella relativa sequenza si incontrano molte coppie di nodi consecutivi che sull'arborescenza sono collegati da semicammini piuttosto lunghi, cioè nodi con un ascendente comune a un livello sensibilmente inferiore.

Nella maggior parte delle elaborazioni che richiedono visite risulta molto più conveniente fare riferimento a un ordinamento totale secondo il quale nella maggior parte dei passi in avanti ilnodo successivo sia poco distante e facile da raggiungere dal nodo attuale.

D30b.06 Il procedimento di visita più conveniente è dato dal seguente **algoritmo di visita Alto-Basso-sinistra-destra** che, servendoci dell'abbreviazione di *Top-Down-left-right*, chiamiamo **algoritmo TDlr**.

(1) Si considera come primo nodo da visitare la radice dell'arborescenza distesa e le si aggiunge un segno semplice (spesso chiamato "tag")per segnalare che essa è stata visitata.
(2) Si stabilisce se qualcuno dei figli del nodo corrente non sia ancora visitato; nel caso affermativo si passa al nodo più a sinistra di questi figli, lo si segna come già visitato e si resta nella istruzione (2).
(3) Se nessuno dei figli del nodo corrente rimane da visitare si cerca di risalire al suo predecessore; se questo è possibile si passa all'istruzione (2).
(4) La suddetta risalita non è possibile sse il nodo corrente è la radice e sono stati già visitati tutti i suoi figli e i loro discendenti: quando si giunge a una tale situazione la visita risulta completata.

L'algoritmo viene illustrato dalla seguente figura:

//input pD30b06

D30b.07 Oltre che segnare i nodi successivamente visitati, si possono emettere sequenzialmente loro contrassegni in modo da ottenere una sequenza che definisce il loro ordine totale. Per esempio per l'arborescenza che segue

//input pD30b07

l'algoritmo TDlr porta alla seguente codifica della sequenza ordinata dei nodi:

$$abefcgdhiklj$$

Conviene osservare esplicitamente che

visitare i nodi di un grafo equivale a individuare un loro ordine totale .

D30b.08 Osserviamo che la precedente stringa si può ottenere da molte altre arborescenze aventi come insieme dei nodi $\{a, b, ..., k\}$. Si ha invece una rappresentazione fedele delle arborescenze distese mediante una stringa, cioè una loro codifica, aggiungendo semplici prestazioni all'algoritmo TDlr che riguardino due nuovi segni diversi dai contrassegni dei nodi.

Per questi ruoli utilizziamo le parentesi aperta e chiusa per segnalare, risp., che si scende a un figlio e che si risale verso un padre e chiediamo all'algoritmo di emettere anche uno di questi segni in corrispondenza di ogni cambiamento di livello sull'arborescenza.

Nella visita dell'arborescenza precedente si avrebbe l'emissione della stringa:

$$a(b(e)(f))(c(g))(d(h)(i(k)(l))(j))$$

Questa si può chiamare **codifica completamente parentesizzata**; essa è fedele, in quanto non è difficile ricostruire da essa l'arborescenza che esprime. Essa però è piuttosto pesante: in generale per un'arborescenza con n nodi (e con $n-1$ archi) la codifica completamente parentesizzata è costituita da $3n-2$ simboli: n sono i contrassegni dei nodi e gli archi, dato che ciascuno di essi viene percorso una volta in discesa e una volta in risalita, contribuiscono con $n-1$ parentesi aperte e altrettante chiuse.

D30b.09 La semplice ispezione della stringa precedente mostra che i fattori ")(" possono essere evitati: in effetti, dopo averli tolti, possono essere ripristinati andandoli a inserire prima di ogni contrassegno di nodo non preceduto da "(" e diverso dalla radice.
Evitando i fattori ")(" si ottiene la cosiddetta **codifica essenzialmente parentesizzata** della arborescenza. Nel caso precedente si ha la stringa:

$$a(b(ef)c(g)d(hi(kl)j))$$

Osserviamo che in questa codifica ogni parentesi aperta corrisponde a un arco che scende da un nodo padre (quello alla sinistra della parentesi stessa) al suo figlio più a sinistra e ogni perentesi chiusa corrisponde al risalire un arco che scende a un figlio più a destra.

Si ha quindi una biiezione tra parentesi aperte e chiuse e due parentesi così collegate si dicono **parentesi coniugate**.
Ad ogni sottoarborescenza elementare dell'arborescenza, ovvero ad ogni sua sottoarborescenza di altezza positiva, è associata nella codifica parentesizzata una coppia di parentesi coniugate.

Per passare da una parentesi aperta alla coniugata, si scorre la codifica alla destra della aperta aggiornando un contatore posto inizialmente a 0: a ogni nuova aperta si aumenta il contatore di 1, a ogni chiusa gli si sottrae 1 e si è raggiunta la chiusa coniugata quanto il contatore torna a 0.

Esaminando l'andamento di questo contatore per tutta una visita si constata che il suo valore a ogni passo fornisce il livello de nodo che si sta toccando.

D30b.10 Nella codifica essenzialmente parentesizzata di una arborescenza distesa T si riconoscono facilmente molte delle sue caratteristiche: i nodi padre sono quelli seguiti da una parentesi aperta, i rimanenti sono foglie.
Ogni sottoarborescenza di altezza positiva corrisponde alla sottostringa che inizia con il suo nodo padre e una parentesi aperta e si conclude con la parentesi chiusa coniugata di questa aperta.
L'arborescenza è costituita da m padri, cioè si ottiene innestando m arborescenze elementari, sse la lunghezza della sua codifica essenzialmente parentesizzata è $n+2\,m$.

D30b.11 L'algoritmo TDlr riveste grande importanza, in quanto sono molto numerose le situazioni riconducibili ad arborescenze distese che sono sottoposte ad analisi ed elaborazioni basate sulla sequenzializzazione dei loro nodi.

Spesso occorre considerare il procedimento TDlr come uno **schema di algoritmo**.
Infatti nella maggior parte delle situazioni applicative i passaggi da un padre a un figlio o viceversa non sono azioni elementari, ma richiedono manovre complesse dipendenti dalle regole o dalle strutture informative attraverso le quali si possono controllare effettivamente le arborescenze distese.

Si pensi per esempio al gioco degli scacchi, gioco ormai ampiamente praticato con successo dai computers. Esso fa riferimento a una arborescenza i cui nodi sono le configurazioni della scacchiera alle quali si può giungere con mosse che rispettano le regole, configurazioni alle quali occorre aggiungere un ricordo della configurazione dalla quale ciascuna è stataraggiute.

I nodi di livello pari sono riservati alle configurazioni della scacchiera nelle quali deve muovere il giocatore bianco, i nodi di livello dispari alle configurazioni nelle quali sta per muovere il nero.

Ogni scelta di mossa corrisponde a un arco che porta a un nodo/configurazione figlio, il numero dei possibili figli è mediamente molto alto e lo stesso ordinamento dei figli di un padre richiede parecchie precisazioni.

Ciascuno dei due giocatori deve scegliere ogni mossa in modo di portarsi verso una sottoarborescenza nella quale prevalgano le situazioni a lui favorevoli. L'analisi delle porzioni delle sottoarborescenze "conseguenti" alle possibili scelte riguarda quantità enormi di nodi e richiede accorgimenti che si collocano nella secolare cultura del gioco degli scacchi e che solo in misura limitata conviene far discendere dalle considerazioni generali sulle arborescenze.

Lo schema di algoritmo presentato in precedenza serve solo al livello della impostazione generale delle manovre sulla arborescenza del gioco.

D30b.12 In molte applicazioni delle arborescenze distese, insieme o in alternativa all'ordinamento TDlr, si deve considerare l'analogo ordinamento ottenuto dando la precedenza ai figli più a destra invece che ai più a sinistra. Questo ordinamento, richiamando l'espressione "Top-Down-right-left" (Alto-Basso-destra-sinistra), lo chiamiamo **ordinamento TDrl**; esso si ottiene dall'algoritmo variante del TDlr ottenuto scambiando la direzione sinistra con la destra.

Applicando tale procedimento all'arborescenza.d degli esempi precedenti si ottengono le sequenzializzazioni e le codifiche parentesizzate che seguono:

$$adjilkhcgbfe \qquad a(d(j)(i(l)(k))(h))(c(g))(b(f)(e)) \qquad a(d(ji(lk)h)c(g)b(fe))$$

Vi sono anche importanti manovre (in particolare quelle relative alla problematica del `parsing` `(we)`) nelle quali si individuano arborescenze distese cominciando dalle foglie e dalle sottoarborescenze più in basso: in questi casi servono gli ordinamenti inversi dei precedenti e le codifiche ottenute riflettendo le precedenti e scambiando le parentesi.

Per tali oggetti useremo le sigle rlBU, che abbrevia *right-left, Bottom-Up*, e lrBU, che sta per *left-right, bottom-up*.

Per l'arborescenza precedente si hanno stringhe come le seguenti:

$$((j(lk)ih)d(g)c(fe)b)a \qquad efbgchklijda$$

D30b.13 Accanto alla visita di una arborescenza distesa, conviene prendere in considerazione la seguente nozione.

Si dice **rotta per un nodo** su un'arborescenza distesa $\boldsymbol{D}$ la sequenza di interi costruita con le seguenti regole ricorsive:

- alla radice si assegna la sequenza di 0 interi, cioè la stringa muta ␣ ;
- al g-esimo dei successivi f figli di un nodo già dotato della rotta w si attribuisce la rotta $w(g-1)$, sequenza ottenuta giustapponendo alla rotta del padre l'intero $g-1$.

Chiaramente la rotta per un nodo costituisce una codifica del cammino dalla radice al nodo stesso; inoltre tutti e soli i nodi di livello s sono caratterizzati da una rotta di lunghezza s.

L'insieme delle rotte dei nodi di una arborescenza distesa $\boldsymbol{D}$ è un linguaggio avente come insieme di simboli $\{0, 1, ..., F-1\}$, dove F denota il massimo numero di discendenti di uno dei padri nella $\boldsymbol{D}$, cioè il massimo grado uscente di $\boldsymbol{D}$. Le proprietà che seguono sono piuttosto evidenti.

(1) Prop.: Il linguaggio delle rotte di un'arborescenza distesa è chiuso rispetto al passaggio ai prefissi e l'ordinamento lessicografico delle sue stringhe corrisponde all'ordinamento TDlr dei suoi nodi ▮

(2) Prop.: Ogni linguaggio finito sull'alfabeto A chiuso rispetto al passaggio ai prefissi si può ottenere come insieme delle rotte dei nodi di una sottoarborescenza superiore della arborescenza distesa uniforme delle stringhe di opportuna lunghezza sull'alfabeto A.

D30 c. arborescenze degli assemblaggi

D30c.01 Moltissime costruzioni di configurazioni discrete e di oggetti materiali compositi (si pensi in particolare al Lego o a qualsiasi altro gioco di costruzioni di strutture materiali) si ottengono attraverso successive operazioni con le quali si aggiungono sempre più componenti.

La tendenza di operare in questo modo, dopo essersi sviluppata nella produzione industriale (catena di montaggio), si è imposta anche nella definizione di progetti operativi e nella produzione di strumenti software (organizzazione del software mediante moduli e mediante strati funzionali).

Lo stesso procedere dell'esecuzione di molti programmi si può utilmente vedere come costruzione graduale della soluzione di un problema.

In molte costruzioni si procede aggiungendo a ogni passo un componente elementare: in questi casi parleremo di **processi di assemblaggio**.

Per costruzioni elaborate, in genere, si procede costruendo parti diverse separatamente (talora in reparti diversi e diversamente attrezzati) che successivamente vengono collegate in costrutti via via più corposi fino a ottenere il prodotto completo.
Per questi processi useremo il nome **processi di montaggio**, termine al quale vogliamo attribuire maggiore comprensività di assemblaggio.

Vedremo che i montaggi e gli assemblaggi si rappresentano naturalmente due tipi di arborescenze e che a esse si può ricondurre una grande varietà di processi e di configurazioni discrete.

Cominciamo ora prendendo in esame assemblaggi di alcune delle configurazioni discrete più semplici.

D30c.02 Particolarmente semplici da ottenere mediante assemblaggi sono le collezioni di sequenze combinatorie come le disposizioni con e senza ripetizioni, le combinazioni con e senza ripetizioni e le permutazioni di un insieme finito.

Si può costruire una di queste collezioni di sequenze **s** individuando le sue successive componenti, che supponiamo costituite dagli interi di $(n] = \{1, 2, ..., n\}$, a partire dalla più a sinistra.
Si tratta quindi di un assemblaggio: per ogni componente sono possibili diverse scelte che si sanno determinare con criteri semplici; inoltre il numero di passi costruttivi è prevedibile con semplici considerazioni.

Vediamo come la costruzione di una di queste collezioni si può descrivere mediante un'arborescenza che viene detta **arborescenza degli assemblaggi** relativa al tipo di sequenze in causa.

La radice corrisponde all'inizio della costruzione. I nodi di livello 1 sono etichettati dai vari interi che possono dare inizio a una sequenza accettabile. Ogni nodo di livello ℓ è etichettato da una delle sequenze di ℓ interi costituenti il prefisso di almeno una delle sequenze combinatorie ammissibili.

Con una tale arborescenza degli assemblaggi si fornisce quindi una descrizione costruttiva di una intera collezione S di configurazioni combinatorie soddisfacenti le regole che le caratterizzano.

Spesso una tale arborescenza fornisce utili informazioni sull'insieme S.

D30c.03 Le arborescenze degli assemblaggi delle disposizioni con ripetizione degli interi di $(n]$ di lunghezza s sono le arborescenze uniformi-s-n. Infatti a ogni passo della costruzione di una di queste sequenze si può scegliere come ulteriore componente uno qualsiasi degli interi di $(n]$.

Per esempio per $n = 2,\ s = 3$:

//input pD30c03

D30c.04 Le arborescenze degli assemblaggi delle disposizioni senza ripetizioni degli interi di (n] di lunghezza r sono le arborescenze uniformi a strati aventi come sequenza dei gradi $\langle n, n-1, ..., n-r+1\rangle$. Infatti a ogni passo della costruzione di una di queste sequenze diminuisce di 1 il numero degli interi che rimangono utilizzabili.
Per esempio per $n = 4,\ s = 3$:

//input pD30c04

D30c.05 Per l'assemblaggio delle permutazioni di (n] la precedente costruzione procede fino all'n-esimo passo, ovvero fino a che si ottiene un'arborescenza nella quale le foglie non hanno fratelli.

Per esempio per il caso precedente in cui $n = 4,\ s = 3$:

//input pD30c05

D30c.06 Le arborescenze degli assemblaggi delle combinazioni senza ripetizioni di (n] di lunghezza s possono costruirsi con le seguenti manovre.
Innanzitutto si costruisce un cammino più a sinistra che presenta s nodi ciascuno dei quali destinato ad avere un determinato numero di figli.
Si procede poi a individuare i figli e gli ulteriori discendenti dei nodi che si trovano ai livelli 1,2, ..., $s-1$ abbassando il numero dei figli quando si passa a un fratello immediatamente a destra e mantenendo il numero dei figli quando si passa da un nodo al suo figlio più a sinistra.
Per assegnare le etichette ai nodi basta far crescere gradualmente le etichette di una prole a partire dall'intero successivo a quello che etichetta il padre.

Per esempio per $n = 6,\ s = 3$:

//input pD30c06

D30c.07 Le arborescenze degli assemblaggi delle combinazioni con ripetizioni coincidono con le precedenti come arborescenze distese; per l'assegnazione delle etichette occorre invece far crescere le etichette di una prole gradualmente a partire dall'intero che etichetta il padre.

Per esempio per $n = 5,\ s = 2$:

//input pD30c07

La differenza nelle etichette che si riscontra per le arborescenze precedenti corrisponde alla biiezione tra combinazioni senza ripetizioni degli interi di (n] di lunghezza r e combinazioni con ripetizioni degli interi di $(n-s+1]$ della stessa lunghezza data da

$$\ulcorner \big\langle s_1, s_2, ..., s_r\big\rangle \mapsto \big\langle s_1, s_2-1, ..., s_r-s+1\big\rangle \urcorner \ .$$

D30c.08 Vediamo ora l'arborescenza degli assemblaggi delle partizioni di un intero positivo n presentate in ordine lessicografico decrescente.

I cammini hanno diverse lunghezze: spostandosi da sinistra a destra queste lunghezze vanno crescendo da 1 ad n.

Questa arborescenza distesa viene costruita con un procedimento ricorsivo che inizia assegnando alla radice n figli con le etichette da n ad 1 e quindi, procedendo con $j = n, ..., 1$ a costruire sul nodo etichettato da j l'arborescenza degli assemblaggi delle partizioni di $n - j$ con addendi non superiori a j.

Quindi al nodo con etichetta h per il quale si dispone di k nodi dove k è la differenza tra $n - h$ e la somma delle etichette dei nodi suoi ascendenti (alla radice si attribuisce etichetta 0) si assegnano $\min(h, k)$ figli con etichette decrescenti da $\min(h, k)$ ad 1.

//input pD30c08

D30c.09 Un assemblaggio intravvisto in precedenza consiste nella costruzione di un'arborescenza mediante successivi innesti di arborescenze elementari.
Per un generico grafo dotato di circuiti non si trova un procedimento costruttivo altrettanto evidente: questo corrisponde a una intrinseca minore complessità delle arborescenze e giustifica che si cerchi di utilizzarle per varie costruzioni efficienti.

Vari assemblaggi si incontrano studiando alcuni tipi di polinomi associabili ai digrafi che incontreremo.

Altri assemblaggi riguardano l'individuazione di particolari tipi di cammini. Una distinzione sostanziale tra cammini euleriani e cammini hamiltoniani si può attribuire alla possibilità di individuare facilmente mediante una sorta di assemblaggio i cammini del primo tipo e nella impossibilità di fare cosa analoga per i cammini hamiltoniani.
Questo inoltre mostra che si possono individuare senza difficoltà solo assemblaggi riguardanti configurazioni caratterizzabili esclusivamente mediante regole locali.

Vedremoanche che interessanti assemblaggi riguardano la individuazione di sottoalberi dei grafi e più in generale costruzioni realizzabili mediante i cosiddetti algoritmi greedy [D48h].

D30c.10 Visitare con il procedimento TDlr un'arborescenza degli assemblaggi corrisponde alla generazione delle relative configurazioni secondo un ordine di tipo lessicografico.

La presa in visione di nodi di queste arborescenze può essere utile per la definizione dei meccanismi di generazione di moltissime configurazioni combinatorie. Questo risulta particolarmente evidente per le sequenze combinatorie viste in precedenza.
Le caratteristiche del cammino più a sinistra dicono come generare la prima sequenza.
Delle caratteristiche del cammino più a destra si derivano le regole per concludere la generazione.

Come effettuare il passaggio da una sequenza alla successiva si ricava dall'analisi dei cosiddetti golfi.
Diciamo **golfo di un'arborescenza distesa** ogni sua sottoarborescenza che ha come radice un nodo che possiede almeno due figli ed è costituita da due cammini che passano per i due figli consecutivi f_1 e f_2 e terminano in due foglie consecutive l_1 e l_2, la prima diversa da f_1 e la seconda da f_2.
Nelle arborescenze degli assemblaggi il cammino più a sinistra di un golfo corrisponde alla porzione finale della sequenza che deve essere modificata per passare alla sequenza successiva.

D30c.11 **(1) Eserc.** Descrivere l'arborescenza degli assemblaggi delle matrici binarie 2×2.

(2) Eserc. Descrivere l'arborescenza degli assemblaggi per le partizioni di un intero con addendi eventualmente nulli.

(3) Eserc. Descrivere l'arborescenza degli assemblaggi delle stringhe palindrome di lunghezza 4 sulle lettere a e b.

(4) Eserc. Si consideri l'arborescenza Ψ_1 degli assemblaggi delle permutazioni degli interi 1, 2 e 3; descrivere l'arborescenza Ψ_2 delle possibili costruzioni della Ψ_1 ottenute mediante innesti di arborescenze-1 iniziando dalla radice.
Farsi anche un'idea di cosa si possa individuare un'arborescenza Ψ_3 che stia alla Ψ_2 come questa sta alla Ψ_1.

D30 d. arborescenze di montaggio ed espressioni

D30d.01 Accanto alle arborescenze può essere utile considerare anche le **antiarborescenze**, digrafi ottenibili dalle arborescenze invertendo l'orientamento dei loro archi.
Questa inversione di orientamento porta a una biiezione tra arborescenze e antiarborescenze ed evidentemente rende queste due specie di strutture sostanzialmente equivalenti ed intercambiabili.

Le antiarborescenze consentono di descrivere i processi di montaggio. Gli archi che risalgono da un gruppo di fratelli al loro nodo padre rappresentano efficacemente la costruzione di un pezzo a partire da componenti elementari (foglie) o da altre componenti più ridotte. Alla radice corrisponde l'intero oggetto da produrre con il montaggio.

Una antiarborescenza consente di visualizzare i procedimenti per l'organizzazione della produzione dell'oggetto attraverso montaggi di componenti che possono essere ottenute operando in parallelo persone diverse e se conveniente in impianti diversi.

Alla nozione di montaggio si può ricondurre l'utilizzo di antiarborescenze in geografia per schematizzare bacini fluviali: in questo caso alla radice corrisponde la foce ed ai nodi i punti di confluenza dei vari affluenti e le sorgenti come nodi foglia.

Va osservato che questo genere di modelli si può adottare solo per fiumi con foce a estuario, e non riesce a trattare le isole fluviali e le foci a delta.

D30d.02 Il più importante esempio di arborescenza di montaggio nell'ambito matematico riguarda la descrizione e il trattamento delle espressioni.

Consideriamo un'espressione algebrica come la seguente:

$$(1) \qquad (a+b)*(c-d)+e/f \quad .$$

Essa individua un valore o un possibile insieme di valori la cui determinazione richiede che si esegua un complesso di calcoli presentato dall'espressione stessa.

Questi calcoli consistono nella esecuzione delle operazioni che compaiono nell'espressione sopra valori attuali forniti dagli operandi e dai risultati di operazioni che devono essere state eseguite in precedenza.

La valutazione di un'espressione richiede di rispettare le priorità di esecuzione dei vari operatori.
Queste priorità possono essere determinate dai tipi degli operatori e da convenzioni che li riguardano, ma sono modificabile con l'inserimento nella stessa espressione di coppie di parentesi coniugate; ciascuna di queste coppie ha l'effetto di delimitare una sottoespressione che si vuole venga valutata prima di far agire gli operatori ad essa esterni e in grado di utilizzarla.

L'esecuzione di ogni operatore si può assimilare a una fase di un montaggio, quella che riguarda la elaborazione degli operandi stabilita dall'operatore con il fine della costruzione di un risultato che risulta associato all'operatore stesso.

Una chiara definizione del significato operativo di un'espressione è data da un'arborescenza distesa con nodi etichettati, avente le successive foglie etichettate dagli operandi che si incontrano scorrendo l'espressione da sinistra a destra e ogni operatore come etichetta del nodo padre degli operandi sui quali l'operatore deve agire; questi possono essere operandi espliciti, cioè presenti nell'espressione oppure operandi impliciti forniti da sottoespressioni che vanno eseguiti in precedenza.

D30d.03 Da una figura come la seguente

//input pD30d02

risulta chiaro il fluire delle informazioni elaborate, dai valori forniti dalle foglie ai risultati delle operazioni associati ai nodi padre e ai corrispondenti operatori fino al risultato complessivo associato alla radice.

Questa costruzione viene detta **albero sintattico** dell'espressione.

Nelle implementazioni delle espressioni mediante dispositivi elettronici ad ogni nodo padre viene associata una memoria destinata ad accogliere, in genere temporaneamente e in un caso definitivamente, i risultati della elaborazione, che consistono nei valori forniti dai dispositivi che implementano gli operatori.

Nel cap. C14, servendoci della nozione di grammatica, ci poniamo in grado di trattare un'ampia gamma di espressioni. Vedremo in particolare che la visita lrBU dell'albero sintattico di un'espressione $\mathcal{E}$ porta alla cosiddetta **codifica di Lukasiewicz** di tale $\mathcal{E}$.
In particolare per l'espressione precedente si ha la codifica:

$$ab + cd - *ef/+$$

La notazione di Lukasiewicz permette di automatizzare il calcolo dell'espressione con un procedimento semplice ed efficace che viene ampiamente utilizzato dagli strumenti per i calcoli automatici.

In breve: si scorre la codifica di Lukasiewicz da sinistra a destra utilizzando un deposito a pila (*stack*) per depositarvi operandi espliciti e intermedi e in corrispondenza di ogni operatore si richiede la sua esecuzione sopra gli operandi in cima alla pila e la loro sostituzione con il risultato ottenuto.

D30d.04 Eserc. Dimostrare che una endofunzione entro un insieme finito corrisponde alla somma di digrafi costituiti da cicli nei nodi dei quali si innestano controarborescenze.

D30 e. applicazioni delle arborescenze

D30e.01 Le arborescenze e le svariate strutture ottenute arricchendo adeguatamente delle arborescenze vengono spesso chiamate piuttosto genericamente **strutture ad albero**.
Esse hanno una grandissima varietà di applicazioni: abbiamo già accennato che si incontrano strutture ad albero nello studio delle genealogie, nella botanica, in geografia e nella elaborazione delle espressioni numerico-algebriche.

L'elevata importanza applicativa delle arborescenze è dovuta alla possibilità di associarle a processi che si incontrano con grande frequenza vuoi nello studio di varie questioni matematiche, vuoi nelle procedure di elaborazione di dati, vuoi in attività pratiche.
Questi processi si presentano in forme anche molto diverse, ma risulta conveniente ricondurli a pochi tipi fondamentali.

D30e.02 Dopo gli insiemi di assemblaggi e i processi di montaggio, prendiamo in considerazione i **processi di partizione degli insiemi**.

Un'arborescenza elementare si può far corrispondere a una partizione di un insieme, ovvero a una operazione materiale di suddivisione in parti di un oggetto tangibile che possa essere suddiviso in parti senza difficoltà (asta di legno, torta, figura piana, ...).

Una qualsiasi arborescenza, potendosi decomporre in arborescenze elementari, si può associare a un processo di suddivisione di un insieme o di un oggetto composito,processo che si sviluppa attraverso successive fasi di suddivisione che si potrebbero motivare con opportunità di distinguere o con necessità di selezionare e raffinare.
In questi casi si parla di **arborescenze di partizioni**.

Ogni cammino su tali arborescenze che inizi nella radice corrisponde all'esecuzione di una sequenza di selezioni.
Osserviamo che la presenza di un figlio unico in una arborescenza di partizione costituisce un elemento superfluo, possibilmente da evitare, in quanto non porta contributi all'attività di selezione. D'altra parte queste situazioni possono rendere conto di scelte effettuate con insufficiente consapevolezza strategica.

D30e.03 L'esempio forse più evidente di questi processi è fornito dalla seguente azione.
Si prende in mano una sottile asta di legno; in una prima fase la si spezza in alcune parti; in eventuali fasi successive si spezzano in due o più parti quelle ottenute in precedenza.
Ogni operazione di spezzamento è un modello di partizione di un insieme ordinato e corrisponde a un'arborescenza distesa elementare.
La sequenza dei pezzetti ottenuti presi nell'ordine secondo il quale formavano l'asta originaria corrisponde a un ordine totale entro le foglie dell'arborescenza.

L'insieme da suddividere potrebbe avere gli elementi in corrispondenza biunivoca con le foglie dell'arborescenza oppure essere l'unione di insiemi disgiunti associati alle foglie.

Si possono considerare quindi arborescenze di partizione di un primo e di un secondo tipo; talune arborescenze del secondo tipo si possono considerare sottoarborescenze superiori di ideali arborescenze del primo tipo.

D30e.04 Gli organigrammi delle organizzazioni a struttura rigidamente gerarchica costituiscono i tipici esempi di arborescenze di partizione.

In un tale organigramma viene presentato un insieme di esecutori subalterni che vengono associati alle foglie.

Con arborescenze del primo tipo si possono anche organizzare partizioni per righe (o colonne) delle matrici e partizioni analoghe per prodotti cartesiani di dimensioni superiori.

Con arborescenze di partizione si possono descrivere suddivisioni amministrative o operative del territorio (regioni, province, comuni, vie, numeri civici, appartamenti, vani) e si possono esprimere le suddivisioni gerarchiche dei testi (capitoli, paragrafi, sottoparagrafi, periodi, frasi, parole, caratteri).

Gli schemi tassonomici riguardanti le modalità con cui si manifestano molti fenomeni studiati empiricamente sono invece i tipici casi di arborescenze di partizione del secondo tipo.

Ad essi si possono ricondurre anche gli schemi di classificazione utilizzati per gli archivi bibliografici.
Un tale schema è dato da un'arborescenza distesa con i nodi etichettati dalle denominazioni delle categorie di diversa portata nelle quali si ripartisce una disciplina.
I collegamenti di queste arborescenze riguardano il passaggio da categorie comprensive a settori più specialistici; si possono però avere anche collegamenti non gerarchici tra nodi rappresentanti categorie con aspetti comuni.
Per questi collegamenti, in effetti, uno schema ad arborescenza pura risulta insufficiente, in quanto può adeguarsi solo a una visione basata su valutazioni unilaterali degli argomenti trattati.

D30e.05 Ad ogni nodo di un'arborescenza è associato l'insieme delle foglie sue discendenti. Ad un livello più ricco a ogni nodo di un'arborescenza distesa è associato l'insieme ordinato delle foglie discendenti.
Passare da un nodo a uno dei suoi figli comporta la restrizione dell'insieme o della sequenza delle foglie raggiungibili.

Quindi ogni arborescenza si può far corrispondere a un processo di partizione entro insiemi finiti ed ogni arborescenza distesa si può associare a un processo di partizione di una sequenza finita.

D30e.06 In connessione con le arborescenze di partizione può essere utile esaminare le **arborescenze di indagine**.
Tale genere di struttura viene adottata per descrivere processi nei quali si tratta di esaminare insiemi di possibilità; queste in genere non sono date in forma esplicita, ma sono individuate come potenziali risultati di indagini che vengono effettuate attraverso analisi o esperimenti successivi.

Le arborescenze di indagine sono usate in modo più o meno palese nello studio dei giochi, nei procedimenti di ricerca di configurazioni ottime (ossia nei procedimenti tesi a massimizzare qualche significativo parametro di merito), nei procedimenti di costruzione di soluzioni di vari problemi (mediante procedimenti chiamati in latino "*divide et impera*" ed in inglese "*divide and conquer*").
Queste strutture costituiscono dunque il supporto di algoritmi di larghissima applicazione.

È opportuno osservare che alle arborescenze di indagine ed alle arborescenze di partizione si possono collegare le arborescenze degli assemblaggi.
In effetti una indagine si può considerare la costruzione passo passo della codifica di un risultato.

Inoltre un processo di selezione, cioè un cammino che inizia dalla radice di un'arborescenza di partizione, corrisponde la costruzione passo passo di una codifica del sottoinsieme o dell'elemento che si va selezionando e questa costruzione può considerarsi un assemblaggio.

Ad un processo di partizione progressiva di un insieme si può ricondurre la stesura delle successive fasi di realizzazione di un progetto di ampia portata.

D30e.07 Una anticatena K si dice **anticatena più bassa** o anticatena più dettagliata di una seconda K' sse ogni nodo di K è raggiungibile da (o coincide con) un nodo di K' .

Ricordiamo che una partizione di un insieme S $\{P_0, ..., P_{k-1}\}$ si dice **partizione più fine** di un'altra $\{Q_0, .., Q_{h-1}\}$ sse ciascuno dei Q_j è eprimibile come unione di alcuni dei sottoinsiemi disgiunti P_i.

Passare da una partizione $\mathcal{Q}$ di un insieme a una più fine $\mathcal{P}$ corrisponde a suddividere qualcuna delle parti della $\mathcal{Q}$ in più sottoinsiemi.
In altre parole, per un'arborescenza di partizione di un insieme S il passare da un'anticatena a una più dettagliata corrisponde al passare da una partizione di S a una più fine.

Questo genere di manovra trova molti esempi nella pratica: il passaggio dalla sudddivisione del territorio italiano in regioni a quella in provincie e poi a quella in comuni; il suddividere un righello dapprima in centimetri, poi in mezzi centimetri e successivamente in millimetri; il tagliare un trancio rettangolare di pizza prima in strisce orizzontali e poi con tagli verticali.

D30e.08 Ricordiamo che si dice **infimo di due partizioni-s** la meno fine di quelle che sono fini almeno quanto ciascuna delle due.

Come esempio di questa operazione binaria tra partizioni si può considerare l'insieme degli abitanti di un comune e le sue partizioni rispetto al sesso e all'anno di nascita; la partizione di questa popolazione che distingue sia per sesso, che per anno di nascita, costituisce l'infimo delle due precedenti.

In effetti, spesso, passare all'infimo di due partizioni di un insieme corrisponde a considerare la possibilità di distinzione fornita dalla precisazione di due parametri aventi capacità discriminanti indipendenti.

Di questa operazione tra partizioni si può considerare l'esempio, visivamente efficace, di due suddivisioni in intervalli di un segmento:

//input pD30d12

D30e.09 Ogni arborescenza distesa priva di figli unici si può ricondurre ad un'arborescenza distesa binaria, cioè a un'arborescenza nella quale ogni nodo padre ha non più di due figli.
Per fare ciò basta, per ogni arborescenza-1 con un numero di figli $r > 2$, aggiungere $r-2$ nodi intermedi tra il padre e i figli.

//input pD30d13

Questa sorta di "allungamento" di un'arborescenza distesa può portare a qualche vantaggio, in quanto le manovre sulle arborescenze binarie possono essere implementate in modo particolarmente efficiente e corrispondono a **scelte dicotomiche**, cioè a scelte basate su prove che possono avere solo risposta positiva o negativa e quindi con caratteristiche di elementarità e relativi vantaggi.

D30e.10 Nelle figure che seguono sono mostrati un esempio di albero di indagine e una raffigurazione piana delle ripartizioni dell'insieme delle possibilità alle quali si riferisce.

//input pD30d14

//input pD30d14B

Ai nodi e agli archi si possono associare numeri (interi o reali), che esprimono valori di distanza, di costi o di tempi richiesti da spostamenti, portata di tratti di vie di comunicazione, probabilità e tantoaltro.

Per "modellizzare" realtà più complesse potrebbe rendersi necessario anche associare ai nodi e agli archi sequenze di attributi numerici o strutturali.

In particolare si potrebbero avere espressioni di calcolo associate agli archi di un digrafo che descriva un procedimento complesso in presenza di alternative e di consequenzialità: digrafi di questo tipo sono i diagrammi di flusso, cioè strutture grafiche arricchite che sono in grado di rappresentare gli algoritmi secondo ben studiate modalità canoniche.

D30e.11 Concludiamo questa panoramica con un cenno alle applicazioni delle arborescenze alle strutture informative.

Tra i digrafi arricchiti che vengono utilizzati per rappresentare le strutture di dati per il computer, le arborescenze arricchite rivestono ruoli particolari.

Esse per esempio individuano procedimenti di accesso controllati alle informazioni che possono associarsi ai nodi di un digrafo complesso.
A questa situazione si possono riferire le organizzazioni dei complessi di files variamente accessibili nell'ambito di un sistema operativo.

D30e.12 Molte strutture informative sono organizzate in modo da presentare particolari pregi operativi.

Tra queste vanno citati i **quadtrees**, strutture in grado di trattare efficientemente regioni piane attraverso la loro suddivisione in aree quadrate ciascuna ripartibili in quattro quadrati di estensioni dimezzate; Con un quadtree su più livelli si possono rappresentare con buona approssimazione regioni con confini anche molto irregolari. che possono essere progressivamente.

Simili ad essi sono gli **octrees**, loro corrispondenti tridimensionali utilizzati per descrivere regioni dello spazio attraverso gerarchie di cubi ciascuno dei quali ripartibile in otto cubi con lati dimezzati.

Arborescenze molti utili vengono prese in considerazione per algoritmi di ordinamento e ricerca. Tra i tipi di strutture informative che si possono considerare vi sono le liste con rinvii unidirezionali o bidirezionali, gli alberi binari o n-ari, le code e reti più complesse.

D30 f. arborescenze.N

D30f.01 Si dice **arborescenza.N** ogni relazione il cui terreno è un insieme numerabile N che presenta un unico nodo (radice) dal quale tutti i rimanenti sono raggiungibili attraverso un unico cammino.

Chiaramente ogni arborescenza.N è un digrafo.N.
L'insieme delle arborescenze.N lo denotiamo con **Arb.N** e con $\mathsf{Arb.N}_{\mathsf{N}}$ denotiamo l'insieme delle arborescenze.N il cui terreno è l'insieme numerabile N.

Interessano in particolare le arborescenze.N avente come insieme dei nodi $\mathbb{N}$ o un suo sottoinsieme. Segnaiamo anche che in molte applicazioni si incontrano arborescenze.N il cui terreno è un insieme di stringhe su due distinti alfabeti [C14a] o più in generale un insieme di configurazioni discrete costruibile progressivamente facendo riferimento alla stessa arborescenza.N. In particolare questo accade per l'insieme delle partizioni degli insiemi della forma $\{1, 2, ..., m\}$ [D20i].

D30f.02 Le proprietà basilari delle arborescenze.N si possono considerare varianti di molte delle proprietà trovate in a: per le arborescenze finite, in quanto sono ottenibili con piccoli adattamenti delle dimostrazioni riguardanti queste ultime. Qui ci limitiamoad enunciarle.

(1) Prop.: Un'arborescenza.N è un digrafo.N connesso ▮

(2) Prop.: Un'arborescenza.N non presenta circuiti ▮

(3) Prop.: La radice di un'arborescenza.N non ha alcun predecessore ed è unica ▮

(4) Prop.: In un nodo q di un'arborescenza.N diverso dalla radice entra uno e un solo arco ▮

(5) Prop.: In un'arborescenza.N non vi possono essere semicammini chiusi ▮

(6) Prop.: Un'arborescenza.N è la riduzione riflessivo-transitiva di un digrafo.N ordinato graduato ▮

D30f.03 Alle arborescenze.N si attribuiscono molte nozioni introdotte per le arborescenze finite, ancora attraverso semplici adattamenti delle argomentazioni usate per le arborescenze finite.

Anche tra i nodi delle arborescenze.N si hanno le relazioni essere padre, essere figlio, essere fratelli, essere cugini, essere peredecessore, essere successore, ssere foglia.

Anche i nodi delle arborescenze.N si ripartiscono sui successivi livelli; mentre per le arborescenze finite l'insieme dei livelli è un intervallo della forma $[h]$, per le arborescenze.N non finite l'insieme dei livelli è $\mathbb{N}$.

Anche per le arborescenze.N rivestono importanza i cammini che iniziano dalla radice; ora si possono avere o meno cammini che si concludono con una foglia, cioè con un nodo che non ha successori.

Anche nelle arborescenze.N si individuano sottoarborescenze finite e in particolare sottoarborescenze elementari; esse inoltre possono avere sottoarborescenze numerabili.
Ora si possono avere anche sottoarborescenze.N la cui individuazione richiede regole che devono avere carattere algoritmico, ben diverse dagli elenchi finiti che consentono di individuare le sottoarborescenze finite.

D30f.04 Vediamo alcuni semplici esempi di arborescenze.N; altri esempi più elaborati si possono dare in contesti specializzati, come quelli riguardanti dimostrazioni induttive, grammatiche acontestuali, costruzioni di famiglie di configurazioni di carattere progressivo; tra queste ultime segnaliamo le parole palindrome, leparole di Fibonacci, le tavole di Young standard, gli insiemi di cammini specifici in spazi $\mathbb{N}^{\times d}$.

Chiamiamo **arborescenza.N binaria** la struttura

$$\left\langle \{0,1\}^* \ , \ \{w \in \{0,1\}^* \ :| \ \langle w, w0 \langle, \langle w, w1 \langle \} \right\rangle \ .$$

Questa arborescenza.N può servire per inquadrare procedimenti di analisi ricerca dicotomica potenzialmente illimitata. Tutte le sue sottoarborescenze.N elementari sono isomorfe ala arborescenza-1 con 2 figli.

Più in generale, per ogni $k = 2, 3, 4, ...$ si possono definire le arborescenze.N nelle quali ogni nodo possiede k figli e viene identificato da una sequenza di interi dell'intervallo $[k)$.

L'arborescenza.N relativa a $k = 3$ potrebbe servire per esaminare ricerche o indagini tricotomiche.

D30f.05 Anche le arborescenze.N si possono arricchire con ordinamenti totali dei sottoinsiemi dei figli di uno stesso padre.
In effetti questo arricchimento risulta naturale nel caso delle arborescenze.N descritte nel paragrafo precedente.

Questa situazione è piuttosto generale: tutte le arborescenze.N che si intendono utilizzare per inquadrare famiglie illimitate e progressive di configurazioni discrete si devono servire di un algoritmo che per ogni nodo forniscono i suoi figli e questi vengono individuati in qualche ordine che si può facilmente individuare.

`L'esposizione in https://www.mi.imati.cnr.it/alberto/ e https://arm.mi.imati.cnr.it/Matexp/matexp_main.php`