1

Capitolo B06
liste, prodotti cartesiani, prodotti numerici

Contenuti delle sezioni

a. prefissi, suffissi, erosioni, infissi e sottosequenze p. 3
b. liste p. 11
c. prodotto cartesiano di liste p. 20
d. griglia-ZZ p. 25
e. prodotto di numeri naturali p. 30
f. potenze cartesiane di liste e potenze di numeri naturali p. 34
g. lista delle stringhe di fissata lunghezza p. 36

39 pagine

B06 0.01 Anche in questo capitolo si procede con la gradualità e con i molti dettagli che consentono di definire in modo convincente procedimenti costruttivi che possono essere proposti come affidabili, ampiamente condivisibili e utili a risolvere in una prospettiva di sistematicità problemi di rilievo.

Qui a partire da semplici algoritmi si introducono le nozioni basilari di lista, di prodotto cartesiano, di prodotto di numeri naturali e interi e di interi elevati a potenze intere.
Si accostano dunque questioni generali limitandosi alle loro versioni che riguardano oggetti finiti e costruibili.

Si inizia riesaminando le nozioni di prefisso, suffisso e infisso di stringhe generiche per disporre di elementi che favoriscano la successiva definizione e discussione delle liste di stringhe.
Vengono poi introdotte le liste a due livelli: il primo più preciso e dettagliato riguarda le liste in forma canonica, definite in modo semplice ma con il rischio di risultare rigide e pletorico; il secondo poco preciso, basato sulla prospettiva di maggiore versatilità con la promessa di successivi chiarimenti e volto a rendere presentabili molte liste di interesse pratico.
Le liste sono anche l'occasione di insistere sulla necessità di introdurre contrassegni per costruzioni formali che possono essere richiamate più volte al fine di poterle renderle più maneggevoli e versatili.

Si prosegue con il prodotto cartesiano di due liste di caratteri e più in generale di liste di stringhe per poi focalizzare l'attenzione sui più particolari prodotti di stringhe e di liste di stringhe prive di ripetizioni.

Si utilizzano le liste di caratteri prive di ripetizioni per introdurre gli alfabeti con il ruolo di insiemi finiti basilari; subito dopo si passa alle potenze cartesiane di alfabeti e di liste prive di ripetizioni, giungendo a definire gli insiemi costituiti da tutte le stringhe di una data lunghezza costituite dai caratteri di un dato alfabeto. ambienti adatti ad ospitare molte nozioni combinatorie

Gli insiemi delle stringhe di data lunghezza s su un alfabeto di n caratteri sono proposti come ambienti di lavoro di primaria importanza, dopo aver constatato che i loro elementi possono essere facilmente generati e sottoposti a importanti esami e rielaborazioni.

Inoltre si possono individuare concretamente insiemi di sequenze di caratteri o di liste di stringhe di elevato interesse combinatorio in quanto queste sequenze si riveleranno adatte a supportare molte costruzioni utili per definire varie strutture formali w per molteplici applicazioni.

In particolare si prendono in esame i sottoinsiemi di tutte le stringhe su un alfabeto di un dato numero n di caratteri che sono nonripetitive e di lunghezza n; ciascuna di tali stringhe viene chiamata permutazione dell'alfabeto e può assumersi il compito di definire un ordinamento degli n caratteri, relazione alla quale risulta spesso necessario fare riferimento.

B06 a. prefissi, suffissi, erosioni, infissi e sottosequenze

B06 a.01 Serve un algoritmo per un esecutore MSM che, date due stringhe w e u, decide se w si può ottenere giustapponendo ad u una opportuna stringa x, cioè se si può trovare una stringa x tale che $w = u_{\shortmid} x$.

Il procedimento richiesto si ottiene organizzando un cosiddetto **scorrimento parallelo** da sinistra a destra dei due nastri T_w e T_u su cui sono presentate all'esecutore MSM, risp., la stringhe w e u.

Inizialmente la testine di lettura di ciacuno dei due nastri si trova sulla casella occupata dalla stringa da esaminare.
In uno stato iniziale q_1 si legge da T_w un carattere, lo si carica in un registro R_w, si avanza di una casella e si passa allo stato q_2.

In q_2 si legge da T_u un carattere, lo si carica in un registro R_u, si avanza di una casella e si passa allo stato q_3.

In q_3 si confrontano i contenuti di R_w ed R_u; se coincidono si va in q_4, se no in q_5.

In q_4 si confronta R_w con $\dashv$: in caso affermativo si va in f_1 dove si decide che, essendo $u = w$, la stringa u è effettivamente prefisso della w, mentre in caso contrario si torna in q_1.

In q_5 si confronta R_u con $\dashv$: se coincidono resta stabilito che la u è prefisso della w diverso da tale stringa e si va in f_1; in caso contrario si va in f_0 e qui si segnala che la u non è prefisso della v.

Se si trova x tale che $w = u_{\shortmid} x$ si dice che u è un **prefisso** di w; in particolare se la stringa x è la stringa muta, cioè se $u = w$, si dice che u è prefisso improprio di w, mentre se x presenta almeno un carattere si dice che u è prefisso proprio di w.

L'aggiunta al precedente algoritmo di poche istruzioni, nel caso u sia prefisso di w consente di emettere la stringa x; questa stringa si dice ottenuta per **cancellazione da sinistra** della u dalla w.

Per esempio `aba` è prefisso di `abaacb` e `tergi` è prefisso di `tergicristallo`; le due rispettive cancellazioni sono `acb` e `cristallo`.

La definizione consente di affermare che ogni stringa è prefisso (improprio) di se stessa.

L'algoritmo precedente consente di riscontrare un prefisso anche quando $u = \sqcup$ e in tal caso fornisce $x = w$ come risultato della cancellazione dalla w di tale u. Quindi si può affermare che la stringa muta è prefisso di ogni stringa.

Abbiamo dunque trovate due situazioni ciascuna delle quali può essere considerata caso particolare di una situazione più comprensiva tra le stringhe su qualsiasi alfabeto.

La possibilità di esprimere con precisione una situazione generale consentita dalla disponibilità della stringa muta risulta quindi vantaggiosa per l'attività di formulazione di enunciati con prospettive di riutilizzo e quindi con qualche miglioramento delle capacità costruttive.

B06 a.02 Date due stringhe w e v, con una procedura simile alla precedente basata su uno scorrimento parallelo delle due stringhe, ma procedendo da destra a sinistra, si può decidere se la w si può ottenere giustapponendo ad una opportuna stringa y la v, cioè se si può trovare una y tale che $w = y_{\shortmid} v$.

Se questo accade si dice che v è un **suffisso** o anche un postfisso della stringa w, mentre la y, stringa anch'essa facilmente individuabile, si dice ottenuta per **cancellazione da destra** della v dalla w.

Inoltre, similmente al caso del prefisso, si ha che ogni stringa w è suffisso di se stessa e che $\sqcup$ si può considerare suffisso di qualsiasi stringa.

Per esempio `cba` è suffisso di `edcba`, essendo `ed` la corrispondente cancellazione da destra, mentre `cani` è suffisso di `Dominicani` avendo `Domini` come cancellazione; ed `arum` è suffisso di `rosarum`. In grammatica un tale suffisso viene detto "desinenza" e la corrispondente cancellazione da destra viene chiamata radice.
È evidente che in molte lingue le tipologie dei cambiamenti di genere (maschile/femminile), dei cambiamenti di numero (singolare/plurale) e le declinazioni (in particolare nel latino) sono esprimibili in termini di prefissi, suffissi, giustapposizione e cancellazione.

Una stringa può essere sia prefisso che suffisso di un'altra: `abra` è prefisso e suffisso di `abracadabra`.

Si possono inoltre definire in modo prevedibile i termini suffisso proprio e suffisso improprio.

Due stringhe u e v, da considerare secondo il loro ordine di presentazione, si dicono costituire, risp., il primo e il secondo fattore di una **bifattorizzazione** della stringa w sse $w = u_{\cdot}v$.
Evidentemente primo fattore di una stringa è da considerare sinonimo di prefisso di tale stringa e secondo fattore va considerato sinonimo di suffisso della detta stringa.

Nella lingua italiana sono numerose le parole, chiamate "composte", che possono essere bifattorizzate in due fattori ciascuno dei quali ha un chiaro significato: calzascarpe, fendinebbia, mangiafumo, pianoforte, tirapiedi, ...

B06 a.03 Una affermazione come "la stringa *abac* è prefisso proprio della *abaccba*", decidibile mediante il semplice algoritmo sopra descritto, stabilisce un collegamento tra le due stringhe il cui riconoscimento potrebbe risultare utile.

Questo fatto induce a rendere il collegamento tra due stringhe u e w comunicabile mediante l'affermazione "u è prefisso di w"; questo enunciato va considerato una entità riconoscibile con sicurezza e utilizzabilein qualche argomentazione.

Si è quindi indotti a cercare questa proprietà e altre simili (ad esempio "essere suffisso proprio", "essere stringhe diverse") attribuibili a coppie di stringhe in seguito a una verifica algoritmica o attraverso qualche altra argomentazione affidabile; trovata una di queste proprietà risulta opportuno segnalarla a un buon livello di accessibilità e di leggibilità, in quanto questo può contribuire a farla utilizzare per qualche argomentazione riguardante stringhe.

La stessa considerazione si può fare per altre affermazioni riguardanti proprietà di coppie di altri oggetti formali, anche grazie al fatto che tutti gli oggetti formali sono in qualche modo sono esprimibili mediante stringhe.
Ricordiamo in particolareconnettivi come "essere minore o uguale a" ed "essere maggiore di" esaminate in B04d riguardanti coppie di numeri interi individuabili con stringhe unadiche.
Osserviamo inoltre che sono esprimibili mediante stringhe anche definizioni, notazioni, espressioni, schemi metodologici e altri strumenti espositivi.

Anche le affermazioni concernenti stringhe quali "essere prefisso" o "non essere suffisso proprio" sono da considerare entità (esprimenti connettivi relazionali algoritmici, ovvero decidibili) per le quali si possono trovare proprietà che possono risultare interessanti e che quindi conviene esaminare ed enunciare.

Anticipiamo inoltre che molte proprietà dei molti generi di entità che si incontrano possono essere segnalate utilmente come

In particolare vedremo che coppie di grande importanza sono quelle con il primo membro qualificato come "elemento" ed il secondo come "insieme" del quale l'elemento fa parte .

B06 a.04 Per formulare gran parte delle argomentazioni matematiche conviene adottare notazioni specifiche concise; vediamo quali notazioni concise possono essere vantaggiose per le nozioni precedentemente introdotte.

Per affermare che la stringa u è prefisso della stringa w introduciamo la notazione $u \ \mathsf{pfx} \ w$.
Per affermare invece che la stringa v è suffisso della stringa w adottiamo la notazione $v \ \mathsf{sfx} \ w$.

Introduciamo anche simboli che denotano concisamente le cancellazioni.

Se si riscontrano le bifattorizzazioni $w = u \cdot v = x \cdot y$ scriviamo

$$v = u \blacksquare w \qquad \text{e} \qquad x = w \blacksquare y \ .$$

Dunque con i simboli $\blacksquare$ ed $\blacksquare$ denotiamo due operazioni binarie tra stringhe che diciamo, risp., **erosione da sinistra** ed **erosione da destra**.

Anche i simboli pfx, sfx, $\blacksquare$ ed $\blacksquare$ sono da collocare in $\mathbb{A}M$.

A questo punto conviene sottolineare che anche nello studio di entità elementari come le stringhe risultano necessarie (e utili) molte accurate precisazioni e che questa esigenza induce a rendere disponibili notazioni differenziate atte a facilitare presentazioni per le quali in genere si devono raggiungere buoni compromessi tra accuratezza, generalità, concisione e agevole comunicazione di significati.

Preannunciamo anche che l'adozione di notazioni accurate favorisce la possibilità di mettere a punto strumenti per la manipolazione automatica di definizioni di enunciati e in genere di formule che consentiranno di sostenere, sia sul piano della coerenza che sul piano della accessibilità, la documentazione, l'utilizzo e gli sviluppi dell'*apparato*.

B06 a.05 Le operazioni binarie di erosione sono state definite solo per coppie di stringhe particolari: una delle due deve essere prefisso o suffisso dell'altra. A questo punto queste operazioni dovrebbero essere qualificate come operazioni binarie costruibili parziali.

In linea di massima risulta più conveniente trattare con operazioni binarie su stringhe a pieno titolo, cioè eseguibili su tutte le coppie di stringhe su un determinato alfabeto.
Questo scopo si può raggiungere introducendo un nuovo oggetto da affiancare alle stringhe sull'alfabeto di lavoro corrente $\mathbb{A}W_{.,\mathcal{P}}$ che per brevità qui scriviamo $\mathbb{A}W$. Vedremo che questo ha anche l'effetto di rendere più flessibile il trattamento delle stringhe.

Introduciamo quindi il segno $\emptyset$ diverso da tutti gli altri segni in $\mathbb{A}W_{.,\mathcal{P}}$ e denotiamo con $\overline{\mathcal{A}}$ l'alfabeto così allargato.
Il segno $\emptyset$ denota una entità, l'**insieme vuoto**, che risulta importante per lo studio degli insiemi, ma al quale ora attribuiamo solo il significato di "nessuna stringa".
Vedremo anche che questa aggiunta risulta opportuna in quanto rende più agevole l'estensione dello studio delle stringhe allo studio dei linguaggi formali, cioè degli insiemi di stringhe [C32a04].

Per completare questa prima introduzione di $\emptyset$ chiediamo che questa entità si possa giustapporre a sinistra e a destra a qualsiasi stringa w e a se stessa per ottenere se stessa; chiediamo che produca se stessa anche se combinata con qualsiasi stringa in una operazione di erosione.

Facciamo quindi le richieste formali che seguono.

$$\text{Per ogni stringa } w \text{ su } \mathsf{A} \ : \quad \emptyset \cdot w := \emptyset \qquad\qquad w \cdot \emptyset := \emptyset \qquad\qquad \emptyset \cdot \emptyset := \emptyset \ .$$

Si segnala che un oggetto con questa proprietà (in algebra) viene detta elemento assorbente rispetto alla operazione binaria giustapposizione.

A causa della $w \cdot \emptyset = \emptyset \cdot w = \emptyset$ possiamo qualificare come stringhe su $\overline{\mathcal{A}}$ tutte le stringhe su $\mathcal{A}$ e la $\emptyset$.

Conveniamo poi che l'espressione $u \blacksquare w$ assuma come valore $\emptyset$ sse u non è prefisso della w e conveniamo che $w \blacksquare v$ abbia come valore $\emptyset$ sse v non è suffisso della w .

Chiediamo infine che $\emptyset$ sia elemento assorbente anche rispetto alle erosioni:

$$\emptyset \blacksquare w \ := \ w \blacksquare \emptyset \ := \ w \blacksquare \emptyset \ := \ \emptyset \blacksquare w \ := \ \emptyset \ .$$

B06 a.06 A questo punto possiamo affermare che

– per w, u e v stringhe qualsiasi su $\overline{\mathcal{A}}$ l'uguaglianza $w = u \shortmid v$ equivale alla $w = u \shortmid (u \blacksquare w)$;

– per w, x e y stringhe qualsiasi l'uguaglianza $w = x \shortmid y$ equivale alla $w = (w \blacksquare x) \shortmid y$.

Si constata poi che i prefissi dei prefissi di una data stringa sono essi stessi prefissi della detta stringa. Similmente si osserva che i suffissi dei suffissi di una data stringa sono essi stessi suffissi della detta stringa.

Queste proprietà sono espresse anche dalle formule che seguono.

Siano w , u e v stringhe qualsiasi.

Si trova una stringa v tale che $w = u \shortmid v \iff w = u \shortmid (u \blacksquare w)$;

si trova una stringa u tale che $w = u \shortmid v \iff w = (w \blacksquare v) \shortmid v$;

$u \ \mathsf{pfx} \ v, v \ \mathsf{pfx} \ w \implies u \ \mathsf{pfx} \ w$; $\qquad u \ \mathsf{sfx} \ v, v \ \mathsf{sfx} \ w \implies u \ \mathsf{sfx} \ w$;

$w = u \shortmid z \shortmid v \iff w = u \shortmid ((u \blacksquare w) \blacksquare v) \shortmid v$.

Servendoci di $\emptyset$ estendiamo la portata delle operazioni di cancellazione tra stringhe chiedendo che si abbia

$$u \blacksquare w := \begin{cases} v & \text{sse } w = u \shortmid v \text{ per qualche stringa } v \\ \emptyset & \text{sse altrimenti} \end{cases} .$$

e simmetricamente

$$w \blacksquare v := \begin{cases} u & \text{sse } w = u \shortmid v \text{ per qualche stringa } u \\ & \text{sse altrimenti} \end{cases} .$$

Inoltre si constata facilmente che per u, v e w stringhe arbitrarie si ha

$$(u \blacksquare w) \blacksquare v = u \blacksquare (w \blacksquare v) .$$

Si constata inoltre che

$$(u \blacksquare w) \ \mathsf{sfx} \ w \quad , \quad (w \blacksquare v) \ \mathsf{pfx} \ w \quad , \quad (u \blacksquare w) \blacksquare v = u \blacksquare (w \blacksquare v) \ \mathsf{ifx} \ w \ .$$

Qui abbiamo anticipata la definizione di ifx che sarà data in a08.

B06 a.07 Consideriamo una stringa w e due interi naturali h e k.

Si dice **annichilazione da sinistra di lunghezza** h della stringa w la stringa ottenuta eliminando da w le sue prime h occorrenze di carattere se $h \leq |w|$, oppure se $h > |w|$ l'entità $\emptyset$.
Questa stringa si esprime con la scrittura $h \ \mathsf{nihil} \ w$.

Similmente si dice **annichilazione da destra di lunghezza** k della stringa w la stringa ottenuta eliminando da w le sue ultime k occorrenze di carattere se $k \leq |w|$, l'entità $\emptyset$ in caso contrario.
Questo risultato si esprime scrivendo $w \ \mathsf{nihil} \ k$.

I precedenti risultati si possono ottenere con algoritmi facilmente definibili, soprattutto se gli interi sono forniti mediante le loro rappresentazioni unadiche.

Il simbolo nihil lo qualifichiamo come connettivo compositivo costruibile.

Si dimostrano le seguenti proprietà per h e k interi naturali e w e y stringhe qualsiasi.

$$0 \ \mathsf{nihil} \ w = w \ \mathsf{nihil} \ 0 = w \qquad \text{e} \qquad |w| \ \mathsf{nihil} \ w = w \ \mathsf{nihil} \ |w| = \sqcup \ .$$

$$h \text{ nihil } (k \text{ nihil } w) \;=\; (h+k) \text{ nihil } w \qquad \text{e} \qquad (w \text{ nihil } h) \text{ nihil } k \;=\; w \text{ nihil } (h+k)\ .$$
$$h \le k \le |w| \iff (k \text{ nihil } w) \text{ pfx } (h \text{ nihil } w) \qquad \text{e} \qquad h \le k \le |w| \iff (w \text{ nihil } k) \text{ pfx } (w \text{ nihil } h)\ .$$

$$h \text{ nihil } (w \shortmid y) \;=\; \begin{cases} (h \text{ nihil } w) \shortmid y & \text{sse} \quad h \le |w| \\ (h - |w|) \text{ nihil } y & \text{sse} \quad h > |w| \end{cases}\ .$$
$$u \text{ pfx } w \implies (u \blacksquare w) \;=\; |u| \text{ nihil } w \qquad , \qquad v \text{ sfx} w \implies (w \blacksquare v \;=\; w \text{ nihil } |v|\ .$$

B06 a.08 Una stringa z si dice **infisso** della stringa w sse si trovano due opportune stringhe x ed y che giustapposte, risp., a sinistra e a destra della z forniscono la w, ovvero sse accade che $w = x \shortmid z \shortmid y$.

Per esprimere concisamente il fatto che la stringa z è un infisso della stringa w ci serviamo del simbolo ifx, da porre in $\mathbb{A}M$, scrivendo z ifx w .
Anche questo segno costituisce un connettivo relazionale decidibile.
Con esso la constatazione precedente si enuncia anche scrivendo

$$\begin{aligned} z \text{ ifx } w \quad &\iff \quad \text{si trovano due stringhe } x \text{ e } y \;\; ST \quad = \; x \blacksquare w \blacksquare y \\ &\iff \quad (x \shortmid z) \text{ pfx } w \;\wedge\; (z \shortmid y) \text{ sfx } w \end{aligned}\ .$$

Osserviamo che si può affermare che la stringa z è infisso della w in particolare:
quando $z = w$, caso in cui $x = y = \sqcup$ e in cui si parla di infisso improprio;
quando z è prefisso della w, caso in cui $x = \sqcup$;
quando z è suffisso della w, caso in cui $y = \sqcup$.

Per esempio gli infissi di `abcde` sono: $\sqcup$, `a`, `b`, `c`, `d`, `e`, `ab`, `bc`, `cd`, `de`, `abc`, `bcd`, `cde`, `abcd`, `bcde`, `abcde`; gli infissi di `aababa` sono invece: $\sqcup$, `a`, `b`, `aa`, `ab`, `ba`, `aab`, `aba`, `bab`, `aaba`, `abab`, `baba`, `aabab`, `ababa` e `aababa`.

Si può anche affermare che $\sqcup$ è infisso di ogni stringa.

Evidentemente $(x \text{ ifx } y) \wedge (y \text{ ifx } z) \implies (x \text{ ifx } z$.

B06 a.09 Invece dei termini prefisso, suffisso e infisso di una stringa si usano anche, risp., i termini **fattore sinistro**, **fattore destro** e **fattore centrale** di tale stringa.

Secondo questa terminologia una espressione come $w = x \shortmid y$ si dice **bifattorizzazione** della w e una espressione come $w = x \shortmid z \shortmid y$ si dice **trifattorizzazione** della w.

Si constata facilmente che valgono le due seguenti affermazioni:

gli infissi di una stringa sono tutti e soli i prefissi dei suoi suffissi .

gli infissi di una stringa sono tutti e soli i suffissi dei suoi prefissi .

Si osserva che ciascuna di queste due affermazioni si ottiene dall'altra applicando alla arbitraria stringa in esame l'operazione di riflessione per ottenere una stringa che è lecito dichiarare arbitraria.

Per coppie di affermazioni come le precedenti si dice tra di esse intercorre la relazione **dualità per riflessione** o **dualità-LR**, dove LR sta a significare sinistra-destra.
Ialtre parole diciamo che ciascuna delle due precedenti affermazioni è duale per riflessione, ovvero duale-LR, dell'altra.

Si osserva che vi sono stringhe z e w con z ifx w tali che la prima si può trovare in più posizioni diverse della w: ad esempio nella $abcddefddefddghi$ il suo infisso $ddefdd$ si trova a partire dalla posizione 4 e a partire dalla posizione 8.

Le eventuali più occorrenze di un infisso in una stringa si posson distinguere con le loro diverse posizioni iniziali. In particolare con le loro posizioni si possono distinguere le diverse occorrenze di un carattere in una stringa.
Chiaramente un infisso in una stringa si può individuare con la sua posizione iniziale e con la successiva della finale, oppure con la sua posizione iniziale e con la sua lunghezza.
L'infisso di una stringa w che inizia nella posizione i e ha lunghezza s si può individuare con l'espressione $(i-1 \text{ nihil } w \text{ nihil } (|w|-s-i+1)$.

B06 a.10 Può accadere di effettuare due costruzioni che forniscono stringhe sulle quali è dubbio se coincidano oppure una sia prefisso dell'altra.

Nell'ambito di una indagine lo stabilire quale delle due possibilità si verifichi costituisce una restrizione delle situazioni possibili e questo equivale a una precisazione della conoscenza e quindi un vataggioso aumento delle informazioni disponibili.

In linea generale è opportuno disporre di termini che consentano di esprimere i diversi gradi di conoscenza che si sono raggiunti.

Diciamo quindi che u è **prefisso in senso stretto** di v se u è prefisso di v e non coincide con essa.

Parlando di due costruzioni che hanno condotto a due stringhe w e x risulta più stringente enunciare "w è prefisso in senso stretto di x" piuttosto che limitarsi ad affermare "w è prefisso di x".

La precisazione "*in senso stretto*" costituisce un rafforzamento per molte relazioni, oltre che per la "essere prefisso di".
In particolare si dice che u è **suffisso in senso stretto** di v se u è suffisso di v e non coincide con essa; analoga distinzione si ha per la relazione "essere infisso di".

Per talune applicazioni possono rendersi opportune altre espressioni verbali per distinguere gli infissi di una stringa. In relazione a una data stringa w :

una stringa z si dice **infisso in senso stretto a sinistra** della w sse z è infisso e non prefisso della v;
una stringa z si dice **infisso in senso stretto a destra** della w sse z è infisso e non suffisso della w;
una stringa z si dice **infisso in senso stretto a sinistra e a destra** della w sse z è infisso, non prefisso e non suffisso della w.
Si dice invece più genericamente **infisso in senso stretto** della v sse z è infisso della w e non coincide con essa.

B06 a.11 Introduciamo un'altro importante connettivo relazionale tra stringhe: si dice che una stringa v nonmuta è una **sottosequenza** della stringa x (necessariamente nonmuta) sse con un qualche scorrimento con possibili salti da sinistra verso destra della x si possono incontrare tutti i successivi caratteri costituenti la v. Quando questo si verifica si dice anche che x è una **sovrasequenza** della v.

Per affermare che v è sottosequenza di x scriviamo v sbseq x ; equivalentemente si può utilizzare la notazione **spseq** per esprimere concisamente la relazione "essere sovrasequenza" in scritture come x **spseq** v .

Evidentemente una affermazione della forma v sbseq x equivale alla affermazione x **spseq** v. Una tale situazione si può esprere dicendo che sbseq e **spseq** sono (denotano) due relazioni trasposta l'una dell'altra, oppure affermando che sono mutamente trasposte.

Chiaramente tutti gli infissi di una stringa x sono anche sue sottosequenze; precisamente sono tutte e sole le sottosequenze i cui successivi caratteri si possono trovare in posizioni consecutive.
Esempi di sottosequenze che non sono infissi sono:

$$\textit{trave}\ \mathsf{sbseq}\ \textit{alternative}\quad,\quad \textit{ratte}\ \mathsf{sbseq}\ \textit{eratostene}\quad,$$
$$\textit{osti}\ \mathsf{sbseq}\ \textit{miosotis}\quad,\quad \textit{badar}\ \mathsf{sbseq}\ \textit{abracadabra}\ .$$

B06 a.12 Abbiamo visto poco sopra che il connettivo sbseq è evidentemente meno stringente, ossia applicabile a più coppie di stringhe, della ifx (la quale a sua volta è evidentemente più applicabile sia della pfx che della sfx).

In altre parole $v\ \mathsf{ifx}\ w \Longrightarrow v\ \mathsf{sbseq}\ w$, ma non viceversa, così come si constata che $v\ \mathsf{pfx}\ x \Longrightarrow v\ \mathsf{ifx}\ x$, ma non viceversa e che $v\ \mathsf{sfx}\ x \Longrightarrow v\ \mathsf{ifx}\ x$, ma non viceversa.

Si tratta comunque di un connettivo relazionale decidibile, come mostra il seguente algoritmo che consente, date due stringhe v e w, di decidere se v sbseq w o meno.

Si organizza una iterazione primaria sulle successive caselle del nastro T_v contenente i successivi caratteri a_i della v e si avvia uno scorrimento del nastro di ingresso T_w contenente la w a partire dalla casella che precede il suo primo carattere; questo scorrimento sarà portato avanti in linea subordinata con la possibilità di effettuare arretramenti.

Per ogni a_i si ricerca procedendo su T_w la prima occorrenza di tale carattere. Se la si trova si procede sul successivo carattere della v; se non la si trova si arretra sulla precedente a_{i-1} riscontrata e si riprende ad avanzare cercando una sua occorrenza nella w più a destra di quella precedentemente trovata.

Se nessuno di questi tentativi ha successo si decide che v non è sottosequenza di w. Se tutte le occorrenze di carattere della v si trovano in posizioni successive nella w si decide che v sbseq w.

Si osserva che questo algoritmo richiede di ricordare le posizioni dei caratteri riscontrati in modo da poter modificare precedenti riscontri. Questo richiede di organizzare una sequenza di queste posizioni modificabili, cosa non semplicissima che potrà essere meglio chiarita disponendo della nozione di arborescenza distesa [D30].

Si osserva che la stessa decisione si raggiunge mediante scorrimenti sia della v che della w da destra verso sinistra e che con i due diversi modi di procedere si possono trovare diverse sequenze di posizioni per le occorrenze di carattere della sottosequenza v nella w.

A questo proposito si parla di occorrenze di una sottosequenza v nella x ciascuna individuabile con la sequenza delle $|v|$ posizioni in x delle successive occorrenze dei caratteri della v.

Per esempio in *novecentonovantanove* si trovano varie occorrenze di *nov*, quelle individuate dalle sequenze di posizioni:

$$\langle 1,2,3\rangle\ ,\ \langle 1,2,12\rangle\ ,\ \langle 1,2,19\rangle\ ,\ \langle 1,9,12\rangle\ ,\ \langle 1,9,19\rangle\ ,\ \langle 1,11,12\rangle\ ,$$
$$\langle 1,11,19\rangle\ ,\ \langle 1,18,19\rangle\ ,\ \langle 10,11,12\rangle\ ,\ \langle 10,11,19\rangle\ ,\ \langle 10,18,19\rangle\ ,\ \langle 17,18,19\rangle\ .$$

Si possono porre i due seguenti problemi:

data una v e una w, individuare la lista di tutte le sequenze delle posizioni delle sottosequenze equivalenti alla v nella stringa w.

data una w, individuare la lista di tutte le stringhe nonmute v che compaiono come sottosequenze entro la w.

B06 a.13 La nozione di sottosequenza si può precisare ricorrendo alla nozione di funzione finita che definiremo in [B12].

Consideriamo le stringhe v e w e per le loro lunghezze scriviamo $m := |v|$ e $n := |w|$; più in dettaglio scriviamo $v = a_1\, a_2 \ldots a_m$ e $w = b_1\, b_2 \ldots b_n$.

Affermare v sbseq w equivale a dichiarare che nell'insieme di funzioni crescenti del genere $\lceil [m) \longmapsto_{<} [n) \rceil$ si trova una ϕ tale che per ogni $i = 1, 2, ..., m$ si ha $b_{\phi(a_i)} = a_i$.

L'esistenza di più occorrenze di una sottosequenza in una stringa corrisponde alla possibilità di trovare più funzioni tra interi naturali con le caratteristiche della precedente ϕ.

Per esempio per la stringa $rstuwxyz$ in quanto sottosequenza della $arbcrsdesfyghtuijuk$ si trovano 16 funzioni ϕ.

Questo accade anche per $v = abra$ e $w = abracadabra$, caso in cui il ruolo della funzione ϕ può essere svolto dalle funzioni individuate dalle seguenti quaterne di interi da 1 a 11:

1 2 3 4 1 2 3 6 1 2 3 8 1 2 3 11 1 2 10 11 1 9 10 11 4 9 10 11 6 8 10 11 8 9 10 11

Nel caso siano $v = cad$ e $w = abracadabra$ si ha invece una sola possibile funzione ϕ, quella individuata dai tre interi naturali successivi 5, 6 e 7 .

Si osserva che in *abracadabra* vi sono solo due occorrenze di *abra* come infisso, molte meno delle nove quaterne di interi che individuano occorrenze di sottosequenze.
Questa constatazione e molte altre simili dicono che la relazione "essere sottosequenza" è ben più complessa da controllare della ifx.

B06 b. liste

B06 b.01 Procediamo ora a esaminare le liste, i costrutti formali che generalizzano le coppie e che rivestono grande importanza per due ordini di motivi.

Sul piano della pratica della elaborazione dei dati esse consentono di esprimere una grande quantità di informazioni che hanno importanza applicativa, possibilità che induce ad aggiungere molti esempi di liste potenzialmente utili a quelle già presentate.

Sul piano della presentazione di argomentazioni generali di interesse matematico-informatico, le liste sono proposte come costrutti che intervengono nelle definizioni di tutte le strutture che vengono attribuite ai dati compositi, strutture che si incontrano nella programmazione e in molti capitoli della matematica: le strutture combinatorie (difficilmente separabili dalle strutture di dati), le strutture algebriche, le strutture topologiche e parecchie altre.

Cominciamo a definire le liste in forme canoniche avanzando richieste formalmente semplici ma accurate e con lo svantaggio di richiedere l'utilizzo di molti tipi di caratteri e il rischio di produrre espressioni prolisse, costrutti che contribuirebbero a rendere alquanto pesanti alla lettura molte parti dell'*esposizione* delle argomentazioni generali.

Presenteremo anche liste formalmente più libere che seguono forme che le rendono più compatte e di lettura più agevole e che in molti importanti campi applicativi consentono espressioni più concise.
Queste liste nei prossimi primi sviluppi e nella pratica diffusa finiscono con l'essere gestite in modo poco accurato, piuttosto elastico, con l'adozione di varie abbreviazioni, semplificazioni e metafore e quindi che possono essere correttamente interpretate solo facendo appello alla intuizione.
Per queste liste in forme più libere ci limitiamo ad avanzare vaghe promesse di successive precisazioni.

B06 b.02

Nelle prossime pagine, dopo aver definite le forme canoniche delle liste, introdurremo loro varianti, tendenzialmente raccomandabili in casi circoscritti, che permettono di usare scritture specifiche ed espressioni verbali e simboliche semplificate ai fini della maggiore leggibilità.

Prima di definire le forme canoniche delle liste introduciamo gli elementi di un formalismo del quale ci serviremo solo per definire i suddetti costrutti formali, ma dal quale in seguito ci svincoleremo proponendo modi di dire e modi di strutturare le formule più sbrigativi.

Introduciamo quattro segni basilari che chiamiamo **ursegni**:

- ≔ urdefinitore.
- [urdelimitatore iniziale,
- , urseparatore,
-] urdelimitatore terminale.

Questi segni ci servono solo per le definizioni di quelle che chiamiamo **quaterne-dsd** e **quaterne-dsda**, definizioni aventi la forma

$$\mathsf{T}_{\mathcal{K}o\mathcal{L}} \;\coloneqq\; [\langle _ , _ , _ \rangle] \quad \text{e}$$
$$\mathsf{Q}_{\mathcal{K}o\mathcal{L}} \;\coloneqq\; [\langle _ , _ , _ \rangle _ \mathsf{A}] \;.$$

Ogni quaterna ha lo scopo di individuare un cosiddetto "genere di liste", tipo di entità qui contrassegnamo con $\mathcal{K}o\mathcal{L}$.

Nelle precedenti definizioni compaiono il simbolo $\mathsf{T}_{\mathcal{K}o\mathcal{L}}$ che ricopre il ruolo del contrassegno delle terna-dsd individuata alla destra dell'urdefinitore e il simbolo $\mathsf{Q}_{\mathcal{K}o\mathcal{L}}$ per fare da contrassegno per la quaterna-dsda che compare a destra del suo urdefinitore.
I segni chiamati a comporre la terna e la quaterna sua estensione sono:

- il carattere $\langle$ detto delimitatore iniziale.
- il carattere "," detto separatore.
- il carattere $\rangle$ detto delimitatore terminale.
- il carattere A che deve contrassegnare una stringa nonripetitiva precisata preliminarmente che fornisce l'alfabeto dei cosiddetti "items" delle liste del genere che viene definito.

I due delimitatori e il separatore si dicono **demarcatori** per il genere di liste, entità che è lecito caratterizzare con il segno $\mathcal{K}o\mathcal{L}$ o con la equivalente terna-dsd.

L'alfabeto A si vuole in grado di individuare tutti gli oggetti che si vogliono rappresentare con gli items delle liste del genere $\mathcal{K}o\mathcal{L}$. Di esso non possono far parte i quattro ursegni, i quattro segni costituenti la quaterna $\mathsf{Q}_{\mathcal{K}o\mathcal{L}}$ e i contrassegni $\mathsf{T}_{\mathcal{K}o\mathcal{L}}$ e $\mathsf{Q}_{\mathcal{K}o\mathcal{L}}$. Spesso A può far parte di un insieme di caratteri di lavoro $\mathbb{A}W$.

Ogni quaterna-dsda consente di stabilire quali liste possono essere attribuite al genere $\mathcal{K}o\mathcal{L}$ che si sta introducendo, liste che servono a esprimere sequenze di dati, ciascuno fornito da una stringa su A.
I primi tra questi costrutti sono le coppie di stringhe di caratteri in A, fornite da stringhe della forma

$$\langle \ \textit{stringa su } \mathsf{A} \ , \ \textit{stringa su } \mathsf{A} \ \rangle \ .$$

Di questa espressione di coppia la stringa che costituisce il secondo infisso evidenziato viene detta "primo membro", "primo componente" o "primo item"; la stringa evidenziata come quarto infisso viene invece chiamata "secondo membro", "secondo componente" o "secondo item" della coppia.

B06 b.03 Le coppie costituiscono un esempio primario di stringhe articolate. La quaterna-dsda consente di introdurre anche stringhe articolate che costituiscono particolarizzazioni lessicali delle coppie delle quali si possono intuire le svariate possibilità di contribuire ad espressioni composite.

Il genere $\mathcal{K}o\mathcal{L}$ viene individuato dalla quaterna-dsda $\mathsf{Q}_{\mathcal{K}o\mathcal{L}}$, ossia dalla terna-dsd $\mathsf{T}_{\mathcal{K}o\mathcal{L}}$ insieme all'alfabeto A e sostanzialmente si può identificare con la quaterna e con la coppia della terna e dell'alfabeto.

Diciamo **terna in forma canonica** del genere $\mathcal{K}o\mathcal{L}$, o equivalentemente secondo la quaterna $\mathsf{Q}_{\mathcal{K}o\mathcal{L}}$, una stringa della forma

$$\langle \ \textit{primo item} \ , \ \textit{secondo item} \ , \ \textit{terzo item} \ \rangle \ ,$$

dove i tre items sono stringhe sull'alfabeto A.

Come nelle coppie in forma canonica i segni $\langle$ ed $\rangle$ hanno il ruolo dei delimitatori e la virgola il ruolo di separatore.

Occorre aggiungere che nella pratica in genere non si distinguono le coppie coniugate di delimitatori da quelle simili ma di dimensioni diverse.
Segnaliamo anche che talora invece del termine "terna" si usa il termine "tripla" e invece di "quaterna" si usa "quadrupla".

Similmente si definiscono quaterne nella forma canonica $\mathcal{K}o\mathcal{L}$, le stringhe articolate della forma

$$\langle \ \textit{primo item} \ , \ \textit{secondo item} \ , \ \textit{terzo item} \ , \ \textit{quarto item} \ \rangle \ .$$

Procedendo in modo analogo si definiscono le quintuple che presentano 5 items, le sestuple con 6 items e così via.

Più in generale per ogni intero positivo s maggiore o uguale a 2, si definiscono come s-uple in forma canonica $\mathcal{K}o\mathcal{L}$ le stringhe articolate che tra i due delimitatori $\langle$ ed $\rangle$ presentano s items inframmezzati da $s-1$ occorrenze del separatore ",".

Si definisce infine più genericamente **lista in forma canonica** del genere $\mathbf{Q}_{\mathcal{K}o\mathcal{L}}$ una stringa avente la forma di una s-upla del genere $\mathbf{Q}_{\mathcal{K}o\mathcal{L}}$, per qualche imprecisato il numero s dei suoi items.

Si conviene inoltre di includere tra le liste in forma canonica anche le stringhe della forma $\langle$ *stringa su* **A** $\rangle$, da considerare liste con 1 item e la stringa $\langle\ \rangle$ da considerare lista con 0 items; questa viene detta anche **lista vuota** e spesso viene considerata indistinguibile dalle liste vuote dei vari altri generi.

B06 b.04 Spesso serve individuare una lista in forma canonica costituita da un certo numero s di items che si lascia indeterminato; per un tale oggetto useremo scritture della forma

$$L\ :=\langle\ I_1\ ,\ I_2\ ,\ \ldots\ ,\ I_s\ \rangle\ . \tag{1}$$

La parte iniziale "$L :=$" serve per attribuire al simbolo L il ruolo di **identificatore** o di contrassegno della lista presentata alla destra del segno medesimo, segno con il ruolo di definitore e che attribuiamo all'alfabeto degli scansori in uso e sul quale ritorneremo.

Gli I_j servono a rappresentare i successivi items e il segno "..." intende dare l'idea della indefinitezza del numero degli items e contribuisce a manifestare la genericità della lista, cioè la possibilità di precisarla quando questo fosse necessario.

Conviene dire esplicitamente che tra gli items di una lista si possono avere items ripetuti; le liste nelle quali questo non accade sono dette **liste nonripetitive**; anticipiamo inoltre che queste liste sono usate per un compito importante, quello di definire gli insiemi espliciti.

I vari I_i possono essere: stringhe su **A** o più in precisamente espressioni in grado di individuare entità che devono risultare chiaramente definite ogni volta che la lista L viene utilizzata .

Sottolineiamo che la definizione delle liste canoniche consente che in esse compaiano altre liste canoniche di un altro genere che vengono dette sue **sottoliste**.
Se si vuole rispettare la richesta di non appartenenza all'alfabeto **A** dei tre demarcatori della lista complessiva le sottoliste devono essere di un genere completamente diverso, ossia presentare tre demarcatori diversi da quelli della complessiva.
Ad esempio una lista relativa ai marcatori "$\langle$", "," e "$\rangle$" può contenere come items delle liste relative a demarcatori che potrebbero essere "[", ";" e "]".
Occorre tuttavia segnalare che la richiesta su alfabeti e demarcatori può essere derogata richiedendo soltanto che ogni sottolista deve essere precisamente riconoscibile come tale. Questo consente di avere liste articolate esprimibili con pochi demarcatori e anche di evitare dei separatori tra items successive. Sui dettagli di queste forme semplificate delle liste per ora non intendiamo entrare per poter procedere più comprensibilmente.

La possibilità di disporre di liste contenenti altre liste a loro volta contenenti sottoliste fa parte di un meccanismo espressivo che viene adottato in vari altri sviluppi e viene chiamato **ricorsione**; va notato che esso non porta a circoli viziosi in virtù della precisa riconoscibilità delle sottoliste di ogni lista che, ovviamente, sono meno elaborate della lista, unica, di cui sono items e che si può chiamare la loro **sovralista**.

Questa possibile articolazione delle liste ci induce a ribadire la segnalazione che le liste aprono la possibilità di definire una vasta gamma di tipi di entità composite che a loro volta, prevedibilmente, hanno la possibilità di servire in molteplici applicazioni.

Conviene ribadire l'importanza dell'adozione dei contrassegni nel linguaggio delle espressioni matematiche che stiamo delineando: ciascuno di tali simboli ha la possibilità di rimpiazzare in più brani espositivi e concisamente una stringa articolata; inoltre ogni lista richiamata da un contrassegno ha la possibilità di essere particolarizzata in forme diverse nel corso delle argomentazioni nelle quali viene menzionata in quanto portatrice di caratteristiche tendenzialmente generiche.
I contrassegni si rivelano sostanzialmente indispensabili per rendere meno prolissi e più leggibili i passi espositivi elaborati, mantenendo se necessario, una loro versatilità.

Il segno ":=" ha il ruolo del definitore di contrassegni e si può anche qualificare come **connettivo tra contrassegno e stringa contrassegnata**.

In ogni definizione della forma (1) si deve pensare che il contrassegno L faccia parte dell'alfabeto di lavoro in uso, mentre ":=" appartiene a $\mathbb{A}M$.

Segnaliamo esplicitamente che nel seguito molte delle precisazioni lessicali come quelle sopra esposte in genere saranno trascurate, ritenendo che possano essere ricostruite dal lettore sulla base dello scopo e degli elementi del contesto.
Nello stesso spirito in genere trascureremo anche di segnalare la dipendenza delle scritture da una esplicitata quaterna-dsda.

B06 b.05 La prospettiva di servirsi di una grande varietà di liste apre la notevole possibilità di trattare liste aventi come items stringhe da considerare non come mere giustapposizioni di caratteri con il solo pregio di essere distinguibili, ma da considerare dotate di strutture formali che rispecchiano loro funzioni e quindi loro significati.

Fondamentalmente si consente che ciascuno degli items di una lista del genere $\mathcal{K}o\mathcal{L}$ possa essere una lista di un genere definito in precedenza facendo riferimento a una quaterna-dsda; questa, se si volesse portare avanti una formalizzazione facilmente garantibile come priva di incertezze , dovrebbe essere diversa dalla $\mathsf{Q}_{\mathcal{K}o\mathcal{L}}$.
Per esemplificare a questa nuova quaterna diamo la forma

$$\mathsf{Q}_{\mathcal{K}o\mathcal{L}1} \ := \ \mathsf{[}\,[\,_\,;\,_\,]\,_\,\mathsf{B}\mathsf{]} \ .$$

Questa quaterna si serve di demarcatori, di caratteri costituenti l'alfabeto B e di simboli ausiliari definiti prima di B che possono essere stati inseriti nello stesso A e per questo va considerata **quaterna-dsda subordinata** della $\mathsf{Q}_{\mathcal{K}o\mathcal{L}}$.

Si possono quindi prendere in esame stringhe articolate aventi una forma suggerita da una scrittura come

$$M \ := \ [\ J_1\ ;\ J_2\ ;\ \ldots\ ,\ J_s\ ]\ ,$$

nella quale sono in evidenza i segni delimitatori "[" e "]", il segno separatore ";" e i simboli J_j rappresentanti stringhe su B.

Prevedibilmente si consente di avere liste contenenti come items diverse sottoliste che possono dipendere da diverse quaterne-dsda subordinate ad una stessa quaterna-dsda precedentemente introdotta; non si pongono limiti neppure ai livelli di subordinazione delle quaterne-dsda.

In tal modo si possono trattare liste i cui items, tutti o in parte, sono liste; e proseguendo nell'ampliamento dei tipi di stringhe composite, si possono trattare liste con (alcuni) items costituiti da liste che a loro volta possono contenere items costituiti da liste e così via.

B06 b.06 In molti contesti le forme canoniche delle liste possono essere sostituite da costrutti formali anche molto diversi, per ragioni che comprendono la semplificate delle scritture, la maggiore evidenza tipografica di suoi items articolati e/o differenziati, e esigenze estetiche.

Per questo definiamo in generale **lista** una stringa strutturata in modo da presentare una sequenza (finita) di occorrenze di stringhe, gli items della lista, per i quali si chiede soltanto che possano essere univocamente delimitati e quindi resi disponibili, mediante algoritmi.

Una lista quindi è in grado di esprimere una entità complessiva nella quale si distinguono successive entità componenti, e a ciascuna lista complessiva può essere assegnato un ruolo diverso da quelli affidati ai suoi items; evidentemente il ruolo di una lista si basa sui ruoli delle sue sottoliste e quindi può essere considerata un loro arricchimento.

Come si è detto, la sequenza degli items di una lista potrebbe essere vuota, contenere uno o pochi item o all'opposto potrebbe presentare un gran numero di componenti variamente articolati.

Si possono proporre molte liste di componenti omogenei nel significato e nella forma la cui utilità può essere facilmente prospettata; evidenti esempi sono gli elenchi di nomi o sigle, le bibliografie e i vettori numerici.

Altre liste utili nelle applicazioni, invece, presentano componenti disomogenee: sono tali le liste delle informazioni di natura diversa che caratterizzano una entità complessa come una persona, un libro, una città, un prodotto, una organizzazione.

Si possono adottare elaborazioni facilmente immaginabili che modificano le singole liste in vari modi:
si possono cambiare i segni che delimitano e separano gli items;
si possono cambiare singoli items; si può accrescere o ridurre la sequenza degli items.
Queste possibilità rendono le liste degli strumenti versatili.

Si possono gestire anche liste annidate in altre liste con nidi di più livelli servendosi di un unica scelta di delimitatori e separatore; infatti con semplici algoritmi che gestiscono le cosiddettw pile si possono distinguere i delimitatori e i separatori dei vari livelli tenendo conto di quante occorrenze di delimitatori iniziali e di quanti finali si incontrano scorrendo la lista complessiva.

Una lista annidata in un'altra lista può essere modificata anche sostituendo qualche suo item con un'altra lista dello stesso genere o viceversa è possibile sostituire una sua sottolista composita con una semplice stringa che fa da suo contrassegno.

In programmazione queste possibilità si concretizzano in algoritmi che possono essere facilmente adattabili. In effetti alcuni importanti linguaggi di programmazione assumono le liste come primari oggetti da elaborare e seguono come paradigma principale la manipolazione di liste.

B06 b.07 Le liste più semplici sono le liste i cui items sono segni che seguono convenzioni concordate entro gruppi di agenti che cooperano; un raggruppamento di esemplari di segni diversi che possono destare interesse può essere registrato su un nastro come stringa eventualmente delimitata tra due segni come $\vdash$ e $\dashv$, oppure equivalentemente sotto forma di lista canonica attraverso una terna-dsd precedentemente scelta in modo da essere agevolmente maneggiabile [B01c].

Una lista di segni nonripetitiva può costituire una cosiddetta **presentazione ordinata di un alfabeto**; se questo alfabeto viene utilizzato più volte deve essere ben riconoscibile e conviene sia identificato da un contrassegno o da un nome peculiare.

Consideriamo una tale lista nella sua forma canonica

$$\mathsf{A} \;=\; \left\langle a_{j_1}, a_{j_2}, ..., a_{j_s} \right\rangle \,,$$

ove ciascuno degli a_{j_r} rappresenta un carattere dell'alfabeto che si sta introducendo.

Questa lista si può considerare equivalente alla stringa che scriviamo

$$\mathsf{a} \;=\; a_{j_1}\, a_{j_2} \,...\, a_{j_s}$$

e argomentando elasticamente si può identificare con essa.

Se si ha una lista di stringhe tendenzialmente associate a significati omogenei, ad esempio una lista di nomi di un qualche genere di oggetti tangibili o mentali, e se di queste stringhe in un brano delle considerazioni generali compaiono numerose occorrenze, può risultare conveniente rimpiazzarle con loro contrassegni semplici, in particolare con caratteri di un alfabeto introdotto con una apposita onvenzione.
Un tale rimpiazzamento spesso viene chiamato "codifica della lista".

Per le possibili elaborazioni sulle suddette stringhe è necessario mantenere disponibile un collegamento tra caratteri della codifica e stringhe codificate.
Questo può essere convenientemente gestito utilizzando una lista i cui items siano coppie aventi come primo membro un carattere codificante e come secondo la corrispondente stringa e servendosi di due terne-dsd. Ovviamente la sequenza dei caratteri codificanti e la sequenza delle stringhe codificate devono essere nonripetitive.
Questa lista di coppie viene spesso chiamata **tavola di codifica**; si osserva che questa tavola ha la forma di una lista i cui items sono sottoliste (coppie).

B06 b.08 Le precedenti considerazioni suggeriscono che molte manipolazioni sulle liste possono essere viste come varianti delle manipolazioni sulle stringhe rispetto alle quali sono più complesse per vari dettagli, ma hanno le stesse organizzazioni complessive.

Un semplice esempio di questa situazione riguarda da una parte liste come la

$$\langle alpha, delta, omega, Lambda, delta, lambda, beta, alpha, Lambda, epsilon\rangle$$

e dall'altra la stringa di caratteri introdotti appositamente

$$\alpha\delta\omega\Lambda\delta\lambda\beta\alpha\Lambda\epsilon\ .$$

La corrispondent tavola di codifica potrebbe essere la seguente lista di coppie:

$$\Big\langle [alpha, \alpha]; [beta, \beta]; [gamma, \gamma]; [delta, \delta], [lambda, \lambda], [Lambda, \Lambda], [epsilon, \epsilon]\Big\rangle\ .$$

Si osserva che una tale lista, che si dice avere la struttura della tabella a due colonne, consente di caratterizzare una sostituzione alfabetica acontestuale che permette di automatizzare le trasformazioni delle stringhe (espressioni) contenenti lettere greche in stringhe contenenti i corrispondenti nomi in italiano. Questa operazione di decodifica si può automatizzare in tanti altri casi.
Per l'operazione inversa di codifica si possono avere problemi che vengono discussi più oltre.

B06 b.09 Vedremo che sono spesso utili liste con items elaborati ma omogenei, in particolare liste con tutti gli items costituiti da coppie, terne o multiple, e liste con lo stesso numero di items.
A questo genere si possono assegnare le liste di vettori appartenenti a uno spazio vettoriale [B32a, B45d, G40a] e anche le liste di s-grammi su un dato alfabeto [:g].

Quando si trattano liste di items eterogenei che presentano nidi di sottoliste di diversi livelli, ovvero liste di liste di liste e così via, può essere opportuno organizzare le liste con di demarcatori diversi, ma questo non è tassativo.

In certi contesti una lista può essere presentata elasticamente senza essere racchiusa tra i due delimitatori servendosi solo di un separatore; questo è il caso delle liste che da sole sono registrate su un nastro loro riservato.

Una lista di items chiaramente distinguibili potrebbe essere presentata trascurando i separatori. In particolare questo si può fare per le liste di items omogenei, a patto che ciascuno di essi possa essere individuato e estratto o eventualmente qualificato da qualche ben definito algoritmo.

Tra questi casi segnaliamo le liste di nomi che presentano la lettera iniziale maiuscola e le rimanenti minuscole e le liste di stringhe che iniziano con un carattere con il ruolo di escape (\ nel linguaggio TEX, $ nel linguaggio PHP) e liste di stringhe racchiuse tra due delimitatori; questi nel linguaggio HTML sono & e ";" per delimitare le codifiche ASCII dei caratteri speciali e coppie di stringhe della forma <tag> e </tag> .

Per liste più elaborate è conveniente parlare di liste a vari livelli, le liste meno comprensive essendo considerate di livello immediatamente superiore al livello delle liste delle quali sono items.

Si parla anche di liste primarie non facenti parte di altre liste, di liste secondarie ciascuna contenuta interamente in una primaria e quindi meno comprensiva di questa sua sovralista, di liste di terzo livello e così via.

Con atteggiamento opposto le liste costituite solo da coppie o in generale solo da sottoliste di forma simile e riconoscibile (ad esempio racchiuse da specifici delimitatori), possono mancare dei separatori primari.

La possibilità di servirsi di liste di forme diverse viene garantita dalla disponibilità di algoritmi in grado di trasformarle in una forma canonica. Vedremo che alcuni di questi algoritmi sono facilmente e sistematicamente individuabili e per i loro dettagli rinviamo allo studio degli automi a stati finiti [C13].

B06 b.10 I singoli items di una lista di s items possono essere individuate dai numeri interi positivi da 1 a s.

Spesso conviene fare riferimento a operatori che chiamiamo **proiettori** parametrizzati da questi interi: il proiettore relativo all'intero positivo i applicato a una lista di almeno i items porta all'item nella posizione i.
Data la lista $L := \langle a_1, a_2, ..., a_s \rangle$ per ogni $i = 1, 2, ..., s$, poniamo

$$\boldsymbol{Prj}_i(L) := a_i \ .$$

Valgono quindi uguaglianze della seguente forma:

$$L = \left\langle \boldsymbol{Prj}_1(L), \boldsymbol{Prj}_2(L), ..., \boldsymbol{Prj}_s(L) \right\rangle \ .$$

In relazione con la possibilità di collegare biunivocamente liste e stringhe, l'azione di questi proiettori è in stretto collegamento con la individuazione di una occorrenza di carattere in una data posizione di una stringa [B01d03].

Può essere significativo ricordare che in molte attività organizzative risultano importanti elenchi di oggetti attribuibili a un unico tipo (persone, prodotti, paragrafi del presente testo, articoli di legge, versi di una poesia, fasi di una attività, ...), ciascun oggetto individuato da una lista di stringhe articolate con il ruolo di suoi attributi (nome, indirizzo, caratteristiche fisiche, qualifiche, ...).

Spesso in queste circostanze conviene identificare ciascun oggetto con il numero intero, in genere positivo o naturale, che fornisce la sua posizione in un elenco con buone caratteristiche di controllabilità formale.
In particolare accade di rappresentare vantaggiosamente le persone mediante un numero di matricola, un progressivo di lista elettorale, un numero di attesa a uno sportello, ...

Spesso queste numeri naturali con il ruolo degli identificatori vengono sostituite da stringhe composite come codice fiscale, codice di carta di identità in corso di validità, SPID, ... che per certe operazioni sono meglio utilizzabili, ad esempio per migliore memorizzazione e per ragioni di sicurezza.
A queste informazioni spwsso si assegna il ruolo di **coordinate** riguardanti gli oggetti che individuano.

Molte di queste informazioni devono essere gestite per riuscire ad informatizzare servizi per il pubblico. Una gestione con buone garanzie di sicurezza di queste informazioni richiede la disponibilità di accurate liste di coordinate e di tavole di codifica per collegamenti tra liste di coordinate alternative.

Ricordiamo anche che molti libri etichettano le prime pagine con numeri romani e le successive con interi positivi, coordinate che nelle versioni digitali vanno collegate con i numeri progressivi delle pagine.

B06 b.11 Per le liste si possono facilmente definire operazioni e relazioni che costituiscono generalizzazioni di quelle definite per le stringhe.

Per questo prendiamo in considerazione le due liste generiche che trattiamo, a partire da una loro forma canonica

$$\mathsf{A} = \langle a_1, a_2, ..., a_h \rangle \qquad , \qquad \mathsf{B} = \langle b_1, b_2, ..., b_k \rangle$$

Si dice **giustapposizione delle due liste** la lista

$$\mathsf{A} \shortmid \mathsf{B} := \langle a_1, a_2, ..., a_h, b_1, b_2, ..., b_k \rangle$$

Si dimostra facilmente che questa operazione binaria è associativa, mentre raramente risulta commutativa e che ha come elemento neutro la lista vuota che possiamo scrivere $\vec{0} := \langle\rangle$.

Si dice **lista riflessa-I della lista A** la lista

$$\mathsf{A}^{\leftarrow} := \langle a_h, a_{h-1}, ..., a_2, a_1 \rangle \ .$$

La riflessione-I delle stringhe è una operazione unaria involutoria e le liste invarianti per questa involuzione si dicono **liste palindrome**; esse devono avere una delle forme $\mathsf{C} \shortmid (\mathsf{C}^{\leftarrow})$ e $\mathsf{C} \shortmid \langle c \rangle \shortmid (\mathsf{C}^{\leftarrow})$, con C lista arbitraria e c stringa arbitraria.

Può essere utile associare a una generica lista **A** altre due liste ottenute con riflessioni: la lista riflessa-C della **A** ottenuta riflettendo separatamente i singoli items.

$$\mathsf{A}^{\leftarrow C} := \langle a_1^{\leftarrow}, a_2^{\leftarrow}, ..., a_{h-1}^{\leftarrow}, a_h^{\leftarrow} \rangle$$

e la cosiddetta riflessa-IC della **A** ottenuta riflettendo sia la sequenza degli items che i singoli items

$$\mathsf{A}^{\leftarrow IC} := \langle a_h \leftarrow, a_{h-1}^{\leftarrow}, ..., a_2^{\leftarrow}, a_1^{\leftarrow} \rangle \ .$$

Di una lista, similmente a quanto si fa per le stringhe, si possono considerare le liste prefisse, suffisse, infisse e le sottoparole; per le corrispondenti relazioni adottiamo gli stessi connettivi relazionali usati per le stringhe: pfx, sfx, ifx e **sbwrd**.

Per questi connettivi valgono proprietà formali strettamente simili a quelle dei corrispondenti connettivi delle stringhe.

B06 b.12 Consideriamo una generica lista $\mathsf{a} = \langle a_1, a_2, ..., a_h \rangle$.

Ad essa si attribuiscono due lunghezze: la lunghezza-I, detta anche **cardinale della lista** e che corrisponde al numero dei suoi items e la lunghezza-C che fornisce il numero dei caratteri che formano la sua scrittura, e quindi il numero dei caratteri che servono per delimitatori e separatori aumentato del numero dei caratteri che servono a presentare tutti i successivi items.
Per denotare la lunghezza-I della lista **A** scriviamo lenI(**a**) o $|\mathsf{a}|$, mentre denotiamo la sua lunghezza-C con lenC(**A**).
Quindi lenI $(\langle est, nord, ovest, sud \rangle) = 4$, mentre lenC$(\langle est, nord, ovest, sud \rangle) = 20$ (considerando superfluo segnalare che "20" equivale a "||||||||||||||||||||") .

Conviene segnalare esplicitamente che per cardinale di una lista che presenta sottoliste ed eventualmente sottosottoliste e simili, ogni sottolista viene considerata come un unico componente.

Per esempio alla lista $\langle tetraedro; [cubo, ottaedro]; [dodecaedro, icosaedro]\rangle$ si attribuisce il cardinale 3.

Ogni coppia della forma $\langle a_i, i\rangle$ con a_i item della lista **a** si dice **occorrenza** di a_i nella **a**.
Per ogni stringa w di caratteri dell'alfabeto degli items della lista **a** si dice **numero di occorrenze nella lista** di w il numero delle occorrenze $\langle w, i\rangle$ tra gli items della lista; questo intero naturale lo denotiamo con $\mathsf{nocc}_w(\mathsf{a})$

B06 b.13 L'osservazione delle caratteristiche comuni tra il ruolo dei caratteri nelle stringhe e il ruolo delle stringhe nelle liste, consente di definire facilmente molti algoritmi utili per operare sulle liste e conseguentemente di trasformare molte argomentazioni sulle stringhe in argomentazioni sulle liste.

Per stabilire se una certa stringa w compare tra i componenti di una lista occorre organizzare una iterazione su questi componenti e per ciascuno di essi uno scorrimento parallelo sui suoi caratteri e su quelli della w con il compito di decidere della loro uguaglianza.
L'aggiunta di semplici manovre consente di conoscere il numero delle eventuali occorrenze di una stringa entro una lista.

È semplice anche definire un algoritmo che date due liste **A** e **B** stabilisce se tutti i componenti della prima compaiono come componenti nella seconda.

Le due lista **A** e **B** si dicono **equivalenti per occorrenza** sse i componenti di ciascuna di esse sono componenti anche dell'altra.
Per annunciare questa equivalenza scriviamo **A** ***AE***occ **B** ; in questo enunciato compare ***AE***occ , simbolo qualificabile come connettivo relazionale decidibile da collocare in $\mathbb{A}M$.

B06 b.14 Tra le liste si distinguono quelle che non presentano componenti ripetuti, che abbiamo chiamate **liste nonripetitive** o liste iniettive da quelle che presentano ripetizioni e sono dette **liste ripetitive**.
È semplice definire un algoritmo che effettui questa distinzione.

Un algoritmo leggermente più elaborato da una qualsiasi lista **a** elimina tutti i componenti ripetuti servendosi, oltre che di un nastro dal quale leggere la lista data, di un nastro T_u per sola scrittura sul quale fornire la lista nonripetitiva richiesta.

Inizialmente si riproduce su T_u il primo item di **a**; quindi si organizza la iterazione sui rimanenti componenti della **a** che confronta ciascuno di essi con i componenti presenti su T_u e solo nel caso di differenza lo accoda su tale nastro.

La lista ottenuta in tal modo si dice **lista ridotta nonripetitiva** della **A**; tale lista la denotiamo con RdcNr(**A**) , scrittura nella quale il simbolo RdcNr, collocabile in $\mathbb{A}M$, rappresenta una trasformazione effettivamente applicabile a tutte le liste.

Evidentemente gli enunciati **A** ***AE***occ **B** e RdcNr(**A**) ***AE***occ RdcNr(**B**) sono equivalenti.

Da queata osservazione segue che per stabilire se due liste con cardinali elevati sono equivalenti per occorrenza conviene in genere passare alle rispettive ridotte ed effettuare su di esse la decisione della equivalenza per occorrenza.

B06 c. prodotto cartesiano di liste

B06 c.01 Consideriamo due liste **a** e **b** i cui rispettivi cardinali denotiamo con h e k e per le quali usiamo le scritture

$$\mathsf{a} = \langle a_1, ..., a_h \rangle \text{ e } \mathsf{b} = \langle b_1, ..., b_k \rangle .$$

In particolare teniamo presente il caso in cui queste sono liste di caratteri, cioè strutture con contenuti informativi equivalenti, risp., a quelli delle due stringhe per le quali usiamo le scritture

$$\mathsf{a} = a_1\, a_2 \,...\, a_h \text{ e } \mathsf{b} = b_1\, b_2 \,...\, b_k$$

Consideriamo poi le coppie $\langle a_i, b_j \rangle$ aventi come prima componente una componente di **a** e come seconda componente una componente di **b**.

Non è difficile delineare un algoritmo per la costruzione di una lista di tutte queste coppie.

Dopo aver predisposto un nastro di uscita che inizialmente contiene solo quello che proponiamo come delimitatore primario $\Big\langle$ della lista risultato, si organizza una iterazione primaria sui successivi componenti di **a**.

Per ciascuno di questi, a_i, si organizza una iterazione secondaria con la quale si scorre **b** e si procede a emettere su un nastro di uscita le coppie $\langle a_i, b_1 \rangle$, $\langle a_i, b_2 \rangle$, ..., $\langle a_i, b_k \rangle$ separati dal segno ";" al quale assegnamo il ruolo di separatore primario.

Alla fine delle h successive fasi primarie riguardanti, risp., a_1, a_2, ... , a_h, sul nastro di uscita si ottiene la lista delle coppie

$$\begin{array}{c} \Big\langle \langle a_1, b_1 \rangle; \langle a_1, b_2 \rangle; ...; \langle a_1, b_k \rangle; \\ \langle a_2, b_1 \rangle; \langle a_2, b_2 \rangle; ...; \langle a_2, b_k \rangle; \\ \cdots\cdots\cdots\cdots\cdots \\ \langle a_h, b_1 \rangle; \langle a_h, b_2 \rangle; ...; \langle a_h, b_k \rangle \Big\rangle . \end{array} \tag{1}$$

La lista così ottenuta la chiamiamo **prodotto cartesiano** delle liste **a** e **b** e la denotiamo con $\mathsf{a} \times \mathsf{b}$.

Dunque con il simbolo $\times$ esprimiamo una operazione binaria costruibile riguardante liste.
Le liste di partenza di questa costruzione si dicono, risp., **primo fattore** e **secondo fattore** del prodotto cartesiano.

B06 c.02 Le coppie costituenti $\mathsf{a} \times \mathsf{b}$ sono state presentate in linee successive aventi lo stesso numero di items e ben incolonnate.

A questa scrittura si possono far corrispondere presentazioni materiali di coppie di oggetti collocati sopra un supporto piano che potrebbe essere un foglio di carta, uno schermo di una apparecchiatura elettronica o una variante della scacchiera.
Si ha una scacchiera vera e propria se $\mathsf{a} = \langle a, b, c, d, e, f, g, h \rangle$ e $\mathsf{b} = \langle 1, 2, 3, 4, 5, 6, 7, 8 \rangle$, e quindi si hanno $* \cdot 8 = 64$ caselle.

Queste presentazioni dei prodotti cartesiani costituiscono i primi esempi, molto particolari, delle presentazioni delle importanti entità chiamate matrici [B16], che ora consideriamo prodotti cartesiani con ogni casella dotata di un valore (in particolare numerico o costituito da un pezzo degli scacchi o dal vuoto).

Per queste presentazioni bidimensionali useremo il termine **schieramenti**.

Conviene osservare che con gli attuali schermi elettronici si possono esaminare matrici di estensioni molto grandi effettuando scorrimenti orizzontali e verticali di un complessivo schieramento virtuale.

Queste manovre mediante dispositivi digitali possono rivestire rilevante importanza pratica: si pensi ai sistemi di gestione dei cosiddetti spread sheets, alle griglie di icone visibili sugli schermi degli smart phones, alla consultazione di mappe geografiche e alla osservazione di navigatori satellitari su autoveicoli (sottoposti a modifiche continue gestite da un sistema globale automatizzato); si consideri inoltre che su tutti gli schermi digitali vengono presentati schieramenti di pixels, di picture elements, ossia di puntini colorati.

Possiamo quindi dire che i prodotti cartesiani di liste, le matrici e più in generale gli schieramenti, hanno assunto primaria importanza per le odierne attività di gestione delle informazioni.

Le coppie $\langle a_i, b_j \rangle$ corrispondenti a un particolare b_j e ai successivi items a_i di **A** presi nell'ordine di comparsa in tale lista si dicono costituire una **riga** del prodotto cartesiano e di ogni matrice su di esso; le coppie $\langle a_i, b_j \rangle$ corrispondenti a un particolare a_i e ai diversi items b_j di **B** presi nell'ordine di comparsa in tale lista si dicono costituire una **colonna** del prodotto cartesiano e di ogni matrice su di esso.

Ciascuna di queste sequenze si può considerare equivalente a una lista ottenuta modificando la lista c01(1) che si serve del solo separatore primario ";" sostituendo con un separatore diverso (ed esempio una "/") ogni ";" che separa una coppia ultima di una riga da una coppia prima della riga successiva (coppia avente il primo membro modificato nel suo successivo nella lista **A**).

Si osserva che la lista prodotto $\mathbf{a} \times \mathbf{b}$ presenta intere righe ripetute sse la lista primo fattore **a** presenta ripetizioni sse, mentre la lista prodotto presenta intere colonne ripetute sse la lista secondo fattore **b** presenta ripetizioni.

Viceversa se la lista prodotto $\mathbf{a} \times \mathbf{b}$ presenta due items ripetuti:

se questi si trovano su una stessa riga si devono avere almeno due colonne ripetute,

se si trovano sulla stessa colonna si devono avere almeno due righe ripetute,

se si trovano solo ripetizioni su righe e colonne mutuamente diverse non si hanno nè righe nè colonne ripetute.

Per quanto riguarda le riduzioni senza ripetizioni delle liste in esame se si hanno ripetizioni sulle righe si ha

$$\mathsf{RdcNr}(\mathbf{a} \times \mathbf{b}) \;=\; \mathsf{RdcNr}(\mathbf{a})) \times \mathbf{b} \;;$$

se si hanno ripetizioni sulle colonne si ha

$$\mathsf{RdcNr}(\mathbf{a} \times \mathbf{b}) \;=\; \mathbf{a} \times \mathsf{RdcNr}(\mathbf{b}) \;.$$

B06 c.03 Per presentare un prodotto cartesiano di liste (o una matrice) a un esecutore automatico può essere vantaggioso dotarlo di un dispositivo di **memoria a griglia rettangolare**, dispositivo che chiamiamo anche nastro bidimensionale.

Un tale dispositivo ricorda i banchi di memoria dei computers utilizzati fino agli anni 1970 formati da nuclei di ferrite disposti a griglia sopra un telaio rettangolare e attraversati da fili elettrici per la scrittura e la lettura delle informazioni binarie portate da ciascun anellino.
Una griglia rettangolare ricorda anche, con una forzatura non eccessiva, un disco magnetico: gli spostamenti in orizzontale su una griglia corrispondono su un disco alla esecuzione di spostamenti su circonferenze con centro nel perno combinati con spostamenti radiali.

Si osserva che un dispositivo di memoria a griglia rettangolare può essere letto e scritto efficientemente da una testina di lettura/scrittura in grado di spostarsi sia in orizzontale che in verticale.

È opportuno segnalare che tutte le manovre eseguibili su griglie rettangolari si possono eseguire con risultati equivalenti anche su nastri monodimensionali, ma attraverso sequenze operative alquanto macchinose e che richiedono molti più passi.
Infatti per emulare gli spostamenti in orizzontale si hanno differenze (facilmente immaginabili) solo negli accorgimenti da adottare alle estremità a sinistra e a destra delle righe di caselle.
Invece uno spostamento in verticale di h righe a partire dalla posizione m di una qualsiasi riga sopra una griglia rettangolare con righe di n caselle utili deve essere simulato su un nastro monodimensionale effettuando uno spostamento in orizzontale di $n - m$ posizioni, da salti di $h - 1$ righe, cioè di $h - 1$ occorrenze del separatore delle righe (se $h > 1$ e d uno spostamento in orizzontale di $m - 1$ posizioni.

Tale manovra si attua su una MSM senza difficoltà ricorrendo a nastri ausiliari con funzioni dette di **contatore** sui quali vengono registrate e ritoccate rappresentazioni dei numeri che determinano gli spostamenti da effettuare; queste registrazioni numeriche sono più facilmente comprensibili se i numeri sono espressi con notazioni unadiche, cioè da repliche di un carattere apposito, tipicamente "|" .

La possibilità di effettuare sopra una griglia rettangolare tutte le manovre attuabili sopra un nastro bidimensionale costituisce un primo esempio di una cosiddetta **simulazione** di un meccanismo mediante un altro con caratteristiche diverse, in questo caso con un meccanismo simulatore dotato di possibilità operative più ridotte e, necessariamente, meno efficienti.

Come chiariremo [C20, C21 e C47], le simulazioni di un meccanismo con un altro contribuiscono a chiarire la intercambiabilità tra molti tipi di strumenti utilizzabili per la soluzione di problemi, e quindi contribuiscono a comprendere e a gestire la adattabilità e la versatilità dell'*apparato*matematico-informatico.

B06 c.04 L'elencazione degli elementi di un prodotto cartesiano $\mathbf{a}\times\mathbf{b}$ può effettuarsi anche organizzando una iterazione primaria su tutti i componenti della seconda lista fattore **b** e per ciascuno di questi, b_j, scorrendo i componenti del primo fattore **a** per emettere su un nastro di uscita le coppie $\langle a_1, b_j\rangle$, $\langle a_2, b_j\rangle$, ..., $\langle a_h, b_j\rangle$.

Alla fine delle successive fasi relative a b_1, b_2, ... , b_k si ottiene sul nastro di uscita l'elenco delle coppie

$$
\begin{aligned}
\Big\langle &\langle a_1, b_1\rangle;\langle a_2, b_1\rangle;...,\langle a_h, b_1\rangle;\\
&\langle a_1, b_2\rangle;\langle a_2, b_2\rangle;...,\langle a_h, b_2\rangle;\\
&\quad\cdots\cdots\cdots\cdots\cdots\\
&\langle a_1, b_k\rangle;\langle a_2, b_k\rangle;...;\langle a_h, b_k\rangle\Big\rangle.
\end{aligned}
\tag{1}
$$

Questo modo di procedere differisce dal precedente solo per lo scambio dei ruoli tra le due liste fattori.
Diciamo **caselle diagonali** le caselle contenenti le cosiddette **coppie diagonali** del prodotto cartesiano, ossia le coppie con le due componenti uguali.
La seconda presentazione del prodotto cartesiano si può pensare come visualizzazione sopra una griglia rettangolare ottenuta da quella presentata in b02 ruotandola nello spazio di 180° intorno alla sua diagonale, ossia la linea delle caselle diagonali.

$$\langle a_1, b_1\rangle\ ,\ \langle a_2, b_2\rangle\ ,\ \langle a_3, b_3\rangle\ ,\ ...\ ,$$

La seconda presentazione si dice ottenuta dalla prima mediante la trasformazione che può chiamarsi **trasposizione del supporto bidimensionale**.

Le due presentazioni delle coppie $\langle a_i, b_j \rangle$ dipendono dai due procedimenti di elencazione adottabili, ciascuno dei quali dipendente dalla priorità dello scorrimento di uno o dell'altro dei due fattori **a** e **b**, nonché dal fatto che ciascuno di essi è stato scorso per componenti crescenti.
Per **a** abbiamo imposto l'ordinamento degli elementi che, servendoci del segno "$\prec$" per esprimere precedenza, denotiamo con la scrittura $a_1 \prec a_2 \prec ... \prec a_h$; per gli elementi di **b** l'ordinamento che deriva dall'elenco della sua definizione lo denotiamo con la scrittura $b_1 \prec b_2 \prec ... \prec b_k$.

Le due diverse elencazioni di $\mathbf{a} \times \mathbf{b}$ discendono la prima dall'aver scelto di scorrere primariamente **a** e secondariamente **b** e la seconda dalla scelta opposta.
Nel primo caso si dice che gli elementi di $\mathbf{a} \times \mathbf{b}$ sono presentati secondo l'**ordinamento lessicografico** delle coppie $\langle a_i, b_j \rangle$, nel secondo si dice che gli elementi di $\mathbf{a} \times \mathbf{b}$ sono presentati secondo l'**ordinamento antilessicografico**.

Entrambi gli ordinamenti, lo ripetiamo, sono determinati dagli ordinamenti implicati dalle letture da sinistra a destra della liste **a** e della lista **b**.

B06 c.05 Se gli ordini secondo i quali si susseguono gli items di **a** e di **b** presentano motivi di interesse, ci si aspetta che presentino interesse anche i derivati ordinamenti lessicografico e antilessicografico del prodotto cartesiano $\mathbf{a} \times \mathbf{b}$.

In questi casi in genere risultano interessanti anche le sequenzializzazioni opposte che denotiamo con $a_h \prec ... \prec a_2 \prec a_1$ e con $b_k \prec ... \prec b_2 \prec b_1$ e di conseguenza anche le tre elencazioni di $\mathbf{a} \times \mathbf{b}$ ottenute sostituendo la prima sequenzializzazione di **a** con la sua opposta, oppure sostituendo la precedente sequenzializzazione di **b** con la opposta, oppure sostituendo entrambe le sequenzializzazioni.

I collegamenti tra le elencazioni che si ottengono, si possono descrivere anche come risultati dell'effettuazione di rotazioni di 180° (o di suoi multipli) nello spazio del piano del supporto materiale intorno a uno o entrambi gli assi mediani delle due griglie rettangolari individuate in precedenza.

Si osserva che si hanno trasformazioni da griglia a griglia anche effettuando rotazioni di 90° (o di suoi multipli) intorno all'asse ortogonale al piano passante per l'incrocio dei due assi mediani nel piano, entità che chiamiamo **asse ortogonale alla griglia rettangolare**; queste trasformazioni si possono raffigurare anche come conseguenze di rotazioni di 180° intorno alla verticale passante per la casella di una qualsiasi coppia $\langle a_i, b_j \rangle$.

Le due presentazioni più interessanti degli elementi di $\mathbf{a} \times \mathbf{b}$ sono la prima presentata in a01(1) e quella ottenuta ruotando questa di 90° intorno all'asse ortogonale alla griglia passante per $\langle a_1, b_1 \rangle$:

$$
\begin{array}{c}
\Big\langle \langle a_1, b_k \rangle; \langle a_2, b_k \rangle; ..., \langle a_h, b_k \rangle; \\
\cdots\cdots\cdots\cdots\cdots \\
\langle a_1, b_2 \rangle; \langle a_2, b_2 \rangle; ...; \langle a_h, b_2 \rangle; \\
\langle a_1, b_1 \rangle; \langle a_2, b_1 \rangle; ...; \langle a_h, b_1 \rangle \Big\rangle \ .
\end{array}
\tag{1}
$$

Esse verranno dette, risp., **raffigurazione matriciale** e **raffigurazione geografica** del prodotto cartesiano, subordinate, nell'ordine, alla sequenzializzazione $a_1 \prec a_2 \prec ... \prec a_h$ della lista **a** e alla $b_1 \prec b_2 \prec ... \prec b_k$ della **b**.

B06 c.06 Le considerazioni sopra i prodotti cartesiani di liste e le scritture a griglie che le raffigurano si possono riprendere in forma semplificata e concisa per i prodotti cartesiani di coppie di sequenze di caratteri, cioè per liste di digrammi così ottenute.

Consideriamo più in particolare due stringhe prive di caratteri ripetuti, ad esempio `a` := `abc` e `p` := `1234567`.
Visualizziamo il corrispondente elenco dei digrammi, cioè la lista delle stringhe di due caratteri, il primo fornito da `a` e il secondo `p`:

$$\begin{array}{llllllll} & \texttt{a1} & \texttt{a2} & \texttt{a3} & \texttt{a4} & \texttt{a5} & \texttt{a6} & \texttt{a7} \\ (1) \qquad & \texttt{b1} & \texttt{b2} & \texttt{b3} & \texttt{b4} & \texttt{b5} & \texttt{b6} & \texttt{b7} \\ & \texttt{c1} & \texttt{c2} & \texttt{c3} & \texttt{c4} & \texttt{c5} & \texttt{c6} & \texttt{c7}\ . \end{array}$$

Questo schieramento si riconduce a un prodotto cartesiano come precedentemente definito sostituendo ogni digramma `xi` con la corrispondente coppia $\langle \texttt{x}, \texttt{i} \rangle$. Dunque lo schieramento precedente si può considerare la presentazione più concisa del prodotto cartesiano

$$\begin{array}{llllllll} & \Big\langle \langle \texttt{a},1\rangle & \langle \texttt{a},2\rangle & \langle \texttt{a},3\rangle & \langle \texttt{a},4\rangle & \langle \texttt{a},5\rangle & \langle \texttt{a},6\rangle & \langle \texttt{a},7\rangle \\ (2) \qquad & \langle \texttt{b},1\rangle & \langle \texttt{b},2\rangle & \langle \texttt{b},3\rangle & \langle \texttt{b},4\rangle & \langle \texttt{b},5\rangle & \langle \texttt{b},6\rangle & \langle \texttt{b},7\rangle \\ & \langle \texttt{c},1\rangle & \langle \texttt{c},2\rangle & \langle \texttt{c},3\rangle & \langle \texttt{c},4\rangle & \langle \texttt{c},5\rangle & \langle \texttt{c},6\rangle & \langle \texttt{c},7\rangle \Big\rangle\ . \end{array}$$

Si osserva che analoghe raffigurazioni si possono dare per i digrammi ottenuti da due stringhe non necessariamente nonripetitive; la restrizione della nonripetitività delle stringhe è servita solo per dare un esempio più leggibile.

Va anche detto che i digrammi da stringhe senza caratteri ripetuti consentono di individuare univocamente i nodi delle griglie che li raffigurano, cosa che non possono fare i digrammi ottenuti da due stringhe che presentano almeno una ripetizione di caratteri.

Stesse considerazioni valgono per le griglie rettangolari che raffigurano prodotti di liste di stringhe.

B06 d. griglia-ZZ

B06 d.01 Procediamo ora a estendere il modello-$\mathbb{Z}$ introdotto in B04d; si tratta di una estensione che invece dei numeri interi riguarda le coppie di numeri interi appartenenti a determinati intervalli.

Le considerazioni che seguono, come quelle del paragrafo precedente, possono essere viste come elementi per lo sviluppo di una sorta di "protogeometria delle griglie rettangolari" che sarà ripresa in B20 e nei capitoli successivi per essere assunta come ambiente di avvio per la geometria analitica.

Consideriamo dunque un intero positivo N che genericamente richiediamo essere "sufficientemente elevato".
Ponendoci dal punto di vista delle possibili applicazioni, scegliamo un N maggiore dei numeri interi positivi che si presumono essere richiesti dalle istanze dei problemi che si intendono affrontare con l'aiuto di un esemplare del nuovo modello, numeri che potranno essere citati nelle argomentazioni che si intendono ricavare dai risultati delle elaborazioni effettuate o pensate sugli esemplari.

Assegnamo poi all'intervallo di numeri interi $I := \mathbf{[}-N : N\mathbf{]}$ il ruolo dell'insieme dei numeri interi che si intendono trattare.
Adottando un tono colloquiale, diciamo di associare le coppie $\langle i, j\rangle$ di $I \times I$ a punti di un supporto piano disposti con regolarità in posizioni equidistanziate su un sistema costituito da linee parallele anch'esse equidistanziate, sia orizzontali che verticali, le quali si incontrano nei punti preannunciati in modo da costituire i nodi di una griglia regolare a maglie quadrate.

Ci proponiamo di definire costruttivamente un esemplare di modello osservabile che consideriamo ottenuto da quella che chiamiamo **modellizzazione-$\mathbb{ZZ}$**, con azioni che ampliano quelle esposte in B04c e in B04d per definire la modellizzazione-$\mathbb{N}$ e il suo ampliamento, la modellizzazione-$\mathbb{Z}$.

L'esemplare preannunciato si colloca su una piattaforma materiale sufficientemente estesa che sia stata dichiarata piana in quanto si sia riscontrato che raggi luminosi sufficientemente collimati si propagano mantenendosi a una distanza piccolissima e costante dalla sua superficie.

Questa piattaforma si pensa posata su un'area piana della superficie terrestre che viene giudicata disposta orizzontalmente, che siasufficientemente estes e nella quale si possano individuare una orientazione verso il nord (e quindi le orientazioni verso est, sud, ovest, ecc.).

Si inizia adagiando sulla piattaforma un opportuno esemplare di modello-$\mathbb{Z}$ riguardante l'intervallo che denotiamo con I con orientazione WE, da ovest verso est; si prosegue per ogni i di I ad assegnare la coppia $\langle i, 0\rangle$ al punto su cui risulta posato il simulacro dell'intero i.

A questo insieme di punti diamo il nome di **intervallo-ZZ orizzontale** del piano-ZZ, mentre il punto $\mathbf{O} := \langle 0, 0]ra$ lo chiamiamo **origine del modello-$\mathbb{ZZ}$**.

B06 d.02 A questo punto si ricorre agli oggetti rigidi che abbiamo chiamati regoli e compassi [B04c03]. Tra i regoli uno era stato assunto come campione delle unità di misura delle lunghezze per il modello-$\mathbb{Z}$, mentre se ne possono predisporre altri le cui lunghezze sono multipli della lunghezze unitaria, quando si ritenesse importante rendere più agevoli le misurazioni delle lunghezze elevate.

Una punta del compasso può essere tenuta fissa su un punto della piattaforma piana per fare da perno, mentre la seconda viene fatta ruotare per far comparire sulla piattaforma una traccia che supponiamo sufficientemente sottile i cui punti vengono ad avere la stessa distanza dal punto perno; questa traccia la chiamiamo **arco di circonferenza fisica**.

Collocando la punta perno prima in $\langle -1,0\rangle$ poi in $\langle 1,0\rangle$ e tenendo la punta tracciante a distanza di due unità di lunghezza si tracciano due archi di circonferenza fino a individuare due punti Q_+ e Q_- del piano, il primo a nord dell'origine $\mathsf{O} = \langle 0,0\rangle$(al di sopra per chi scorre la retta-ZZ da ovest verso est per punti con primi componenti crescenti) e il secondo Q_- a sud dell'origine.

Dopo aver constatato che sia Q_+ che Q_- distano 2 unità sia da $\langle -1,0\rangle$ che da $\langle 1,0\rangle$, si osserva che questidue punti e l'origine risultano allineati; ogni raggio luminoso passante per questi tre punti viene detto **ortogonale** al segmento considerato orizzontale o diretto da ovest a est.

Per chi guarda la piattaforma dall'alto si dice che il segmento iniziale è stato "disposto orizzontalmente" e che è orientato da sinistra verso destra, mentre un raggio a esso ortogonale viene detto "disposto verticalmente"; questo viene orientato in modo che scorrendolo si incontri il punto Q_- prima del punto Q_+, ossia orientato da sud a nord.

B06 d.03 Con operazioni che si servono del regolo unitario o del compasso sostanzialmente simili a quelle adottate per il segmento orizzontale, su un raggio luminoso passante per Q_-, O e Q_+, si individua la sequenza di $n+1$ punti che inizia in O e prosegue verso l'alto con punti che presentano la distanza 1 dal precedente; successivamente si individua la sequenza degli $n+1$ punti che inizia in O e prosegue verso il basso con punti che distano 1 dal precedente.

Ai $2n+1$ punti disposti verticalmente così individuati si associano in successione le coppie $\langle 0,-n\rangle$, ...,$\langle 0,-2\rangle$, $\langle 0,-1\rangle$, $\langle 0,0\rangle$, $\langle 0,1\rangle$, $\langle 0,2\rangle$, ..., $\langle 0,n\rangle$.

Servendosi del citato esemplare osservabile del modello-$\mathbb{Z}$ si constata empiricamente che il segmento verticale della piattaforma-ZZ è sovrapponibile al segmento orizzontale.

Inoltre si constata che muovendo l'esemplare di modello-$\mathbb{Z}$ in modo da collocarlo traslato sui punti del segmento orizzontale del modello-$\mathbb{ZZ}$ i punti di questo risultano sovrapposti; una sovrapposizione analoga si ha per i punti del segmento verticale.

Queste osservazioni inducono a individuare vari oggetti tangibili le cui forme si possono giudicare invariabili nel tempo e che quindi qualifichiamo come **corpi rigidi** (come si è fatto introducendo i regoli). Si assume che sottoponendo questi corpi ai movimenti di traslazione e di rotazione restano invariate le distanze tra coppie di loro punti distinguibili.
A queste invarianze aggiungiamo le invarianze relative al passaggio da un corpo rigido a un suo corrisponente corpo che mediante opportuni strumenti ottici assumiamo che sia il suo riflesso in uno specchio accuratamente piano.
Tutti questi movimenti rispetto ai quali non cambiano le distanze tra coppie di punti distinguibili li diciamo **movimenti-ZZ rigidi**.

Il segmento orientato da ovest verso est individuato inizialmente sulla piattaforma lo consideriamo rappresentato formalmente dalla lista

$$\overline{\langle -n,-1\rangle \text{ , } \langle n,0\rangle} \;:=\; \Big\langle \langle -n,0\rangle, ..., \langle -1,0\rangle, \langle 0,0\rangle, \langle 1,0\rangle, ..., \langle n,0\rangle \Big\rangle \;.$$

Inoltre lo chiamiamo **asse** Ox dell'esemplare del modello-ZZ.

Il segmento orientato da sud verso nord costruito ortogonalmente al primo viene rappresentato formalmente con la scrittura

$$\overline{\langle -n,-1\rangle \text{ , } \langle 0,n\rangle} \;:=\; \Big\langle \langle 0,-n\rangle, ..., \langle 0,-1\rangle, \langle 0,0\rangle, \langle 0,1\rangle, ..., \langle 0,n\rangle \Big\rangle$$

e lo chiamiamo **asse** Oy dell'esemplare.

B06 d.04 Con ulteriori manovre effettuate con raggi luminosi, regoli e compassi si procede ad individuare altri $2\,n$ segmenti orientati orizzontali relativi a coppie di interi con il secondo membro j fisso per ogni segmento; questo si attua per i valori $j = -n, ..., -1, 1, 2, ..., n$.
Similmente si individuano altri $2\,n$ segmenti orientati verticali relativi a coppie di interi con il primo componente i fisso per ogni segmento; questo si attua per i valori $i = -n, ..., -1, 1, 2, ..., n$.

Al segmento orizzontale con secondo membro j risultano associate successivamente le coppie $\langle -n, j\rangle$, ..., $\langle -1, j\rangle$, $\langle 0, j\rangle$, $\langle 1, j\rangle$, ...,$\langle n, j\rangle$; ed al segmento verticale relativo al primo membro i risultano associate le coppie $\langle i, -n\rangle$, ..., $\langle i, -1\rangle$, $\langle i, 0\rangle$, $\langle i, 1\rangle$, ...,$\langle i, n\rangle$.

I segmenti verticali non presentano coppie in comune e si dice che tra di essi sussiste la relazione di **parallelismo**, o equivalentemente si dice che sono **segmenti mutuamente paralleli**.
Anche i segmenti verticali non presentano coppie in comune e si dicono mutuamente paralleli, ossia si dice che tra di essi sussiste la relazione di parallelismo.

La disgiunzione tra segmenti orizzontali o verticali distinti si accorda con l'osservazione empirica che per distanze limitate (non astronomiche) due raggi molto collimati che localmente si verifichino essere paralleli si mantengono alla stessa distanza quando ci si allontana dell'origene e quindi consentono di dichiarare che non si possono incontrare in alcun punto comune degli esemplari fisici del modello-$\mathbb{ZZ}$.

Limiti di questo esemplare di modello sono le estensioni astronomiche e le dimensioni estremamente piccole dei punti e dei segmemti, situazioni nelle quali non si può escludere che accurate osservazioni inducano a dichiarare il modello inadeguato rispetto alle esigenze che hanno portato alle suddette osservazioni.
Per una dichiarazione limitativa di questi tipi usiamo il termine **recusazione del modello**.

Possiamo pensare di disporre molti esemplari di modello-$\mathbb{ZZ}$ che differiscono nelle scelte di N e dell'unità di misura delle lunghezze. Essi nei limiti sopra segnalati e consentendo opportuni adattamenti si possono assumere coerenti e si possono dichiarare equivalenti nei limiti della possibilità di recusazione.

Possiamo anche pensare a esemplari nei quali le coordinate orizzontali e $\mathbf{[}r, s\mathbf{]}$ le coordinate verticali variano in intervalli della forma $\mathbf{[}p, q\mathbf{]}$ e ai quali si chiede solo che $k - h$ sia positivo e sufficientemente grande rispetto alle esigenze applicative.
Anche questi esemplari, non solo quadrati e simmetrici rispetto agli assi ma anche rettangolari e dissimmetrici, risultano coerenti ai precedenti con griglie quadrate e simmetriche rispetto ai due assi e sono in grado di agevolare una maggiore gamma di applicazioni.

Possiamo quindi definire come modello-$\mathbb{ZZ}$ il complesso comprendente gli esemplari costruibili come si è detto, i corrispondenti schieramenti di coppie di interi e le corrispondenze biunivoche tra i suddetti esemplari fisici e i cprrispondenti esemplari formali.

B06 d.05 Definiamo **casella** di un esemplare di modello-$\mathbb{ZZ}$ la zona delimitata da una quaterna di nodi della forma

$$\Big\langle \langle i, j\rangle, \langle i+1, j\rangle, \langle i+1, j+1\rangle, \langle i, j+1\rangle \Big\rangle \ ;$$

si tratta di quadrati di lato unitario che richiamano le tessere di un mosaico semplice e regolare.

Le caselle di un esemplare di modello-$\mathbb{ZZ}$ possono essere utili quando si possono considerare dei contenitori nei quali si dispongono degli oggetti da studiare in quanto collocati con regolarità sull'esemplare. Per esaminare una tale collocazione di oggetti sulla griglia risulta utile la raffigurazione a caselle della griglia.
Una evidente applicazione di questo genere di collocazione è dato dalle usuali raffigurazioni dei pezzi che si trovano su una scacchiera in un dato momento di una partita di scacchi o di dama.

Nelle raffigurazioni dei prodotti cartesiani di due intervalli di interi le scritture delle coppie costituenti i componenti sono riducibili: può essere sufficiente segnalare la casella origine e i componenti degli assi orizzontale e verticale.

Nelle raffigurazioni dei prodotti cartesiani di interi di uso pratico è possibile limitarsi a segnalare solo pochi numeri in grado di individuare righe e colonne di caselle.

Per esempio le raffigurazioni matriciale e geografica a nodi del prodotto $\mathbf{(}7\mathbf{]} \times \mathbf{(}5\mathbf{]}$ sono:

Per le raffigurazioni matriciale e geografica mediante caselle quadrate di $\mathbf{[}5\mathbf{]} \times \mathbf{[}4\mathbf{]}$ abbiamo:

Nella pratica queste raffigurazioni possono presentare l'intero contenuto informativo di un prodotto cartesiano di due intervalli finiti di oggetti presentati con un proprio ordinamento sequenziale (lettere, numeri romani, codici dei comuni italiani, codici fiscali, prefissi telefonici, ...).
Le raffigurazioni dei prodotti cartesiano sono significative in quanto, come vedremo, i prodotti cartesiani si possono giudicare come degli "ambienti neutri ma accoglienti", cioè insiemi nei quali possono essere collocati vantaggiosamente molti oggetti che si intendono esaminare, manipolare o utilizzare in argomentazioni o dimostrazioni.

Sui prodotti cartesiani di intervalli di interi avremo modo di sviluppare sistematiche considerazioni di natura geometrica discreta [B20, B21, B22, B23, B24 e D21].

B06 d.06 Come mostra l'esempio di $\{p, q\} \times \{1, 2, 3\}$, per un prodotto cartesiano di stringhe (ma anche di liste) sono possibili 8 raffigurazioni:

(1)

$$\begin{array}{ccc} p1 & p2 & p3 \\ q1 & q2 & q3 \end{array} \qquad \begin{array}{cc} p3 & q3 \\ p2 & q2 \\ p1 & q1 \end{array} \qquad \begin{array}{ccc} q3 & q2 & q1 \\ p3 & p2 & p1 \end{array} \qquad \begin{array}{cc} q1 & p1 \\ q2 & p2 \\ q3 & p3 \end{array}$$

$$\begin{array}{ccc} q1 & q2 & q3 \\ p1 & p2 & p3 \end{array} \qquad \begin{array}{cc} q3 & p3 \\ q2 & p2 \\ q1 & p1 \end{array} \qquad \begin{array}{ccc} p3 & p2 & p1 \\ q3 & q2 & q1 \end{array} \qquad \begin{array}{cc} p1 & q1 \\ p2 & q2 \\ p3 & q3 \end{array}$$

Si può pensare che seconda, terza e quarta siano ottenute dalla prima con rotazioni intorno ad assi ortogonali alla piattaforma portante per angoli multipli di 90°, mentre le ultime 4 sono ottenibili dalle precedenti con riflessioni che scambiano $p1$ con $q1$, $p2$ con $q2$ e $p3$ con $q3$.

B06 d.07 Un prodotto cartesiano nel quale uno dei fattori è una lista con un solo item non differisce sostanzialmente dall'altro fattore:

(1) $$\langle a_1, ..., a_h \rangle \times \langle e \rangle \;=\; \Big\langle \langle a_1, e \rangle, ..., \langle a_h, e \rangle \Big\rangle \;.$$

Infatti gli elementi dei due insiemi si mettono in corrispondenza biunivoca "spensieratamente", cioè senza dover ricorrere ad alcuno collegamento minimamente elaborato, e le costruzioni elaborazioni e le argomentazioni riguardanti una lista possono ripetersi, previe minime modifiche formali, per l'altra lista fornendo risultati strettamente corrispondenti, cioè riconducibili gli uni agli altri mediante prevedibili riadattamenti.

Questa situazione costituisce un esempio particolarmente semplice di quella che possiamo chiamare **equivalenza informativa** delle due espressioni paragonate.

Per quanto riguarda i prodotti cartesiani nei quali uno o entrambi i fattori coincidono con l'insieme vuoto si impongono le definizioni

$$\mathbf{A} \times \emptyset \;:=\; \emptyset \qquad \text{e} \qquad \emptyset \times \mathbf{B} \;=\; \emptyset \qquad \text{e} \qquad \emptyset \times \emptyset \;:=\; \emptyset \,. \tag{2}$$

Infatti entrambi i membri delle precedenti uguaglianze si possono rappresentare con la scrittura $\langle\rangle \;:=\; \sqcup$, con la quale denotiamo la **stringa muta**.

B06 d.08 Accanto al prodotto cartesiano di due liste $\mathbf{A}$ e $\mathbf{B}$ si può considerare il prodotto cartesiano delle stesse liste prese nell'ordine opposto $\mathbf{B} \times \mathbf{A}$.

Questo viene detto **prodotto cartesiano trasposto** del preceedente ed ha come items le coppie $\langle b_j, a_i \rangle$; ogni coppia $\langle b_j, a_i \rangle$ viene detta trasposta della $\langle a_i, b_j \rangle$

Il passaggio da un prodotto cartesiano di due liste al suo trasposto è evidentemente una involuzione.

Un prodotto cartesiano di liste coincide con il suo trasposto sse i suoi due fattori coincidono.

Un prodotto cartesiano di liste della forma $\mathbf{A} \times \mathbf{A}$ si dice **quadrato cartesiano** o seconda potenza cartesiana della lista $\mathbf{A}$; esso si denota anche con $\mathbf{A}^{\times 2}$ o con l'espressione sua semplificata $\mathbf{A}^2$.

B06 d.09 Date tre liste $\mathbf{A}$, $\mathbf{B}$ e $\mathbf{C}$ si possono prendere in considerazione anche le composizioni come $\mathbf{A} \times (\mathbf{B} \times \mathbf{C})$ e $(\mathbf{A} \times \mathbf{B}) \times \mathbf{C}$.

A rigore le due composizioni precedenti conducono a due liste diverse, in quanto la prima contiene items della forma $\Big\langle a_i, \langle b_j, c_k \rangle \Big\rangle$ e la seconda componenti della forma $\Big\langle \langle a_i, b_k \rangle, c_h \Big\rangle$.
Tuttavia i contenuti informativi delle due liste si equivalgono e inoltre equivalgono ai contenuti della lista i cui items hanno la forma delle terne $\langle a_i, b_j, c_k \rangle$, più semplice delle due precedenti.

In genere quindi alla terza lista si dà la forma $\mathbf{A} \times \mathbf{B} \times \mathbf{C}$ e si conviene di identificare le tre liste e le rispettive componenti rendendo lecito scrivere

$$\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) \;=\; \mathbf{A} \times (\mathbf{B} \times \mathbf{C}) \;=\; \mathbf{A} \times \mathbf{B} \times \mathbf{C} \quad \text{e}$$
$$\Big\langle a_i, \langle b_j, c_k \rangle \Big\rangle \;=\; \Big\langle \langle a_i, b_k \rangle, c_h \Big\rangle \;=\; \Big\langle a_i, b_j, c_k \Big\rangle \,.$$

Considerazioni analoghe si possono svolgere per i prodotti cartesiani di quattro e più liste.

Questo stato di cose porta ad assumere che il prodotto cartesiano di liste goda lalla proprietà associativa.

Di una lista $\mathbf{A}$ si possono considerare anche composizioni come $\mathbf{A} \times (\mathbf{A} \times \mathbf{A})$ e $(\mathbf{A} \times \mathbf{A}) \times \mathbf{A}$ e, considerando opportuno privilegiare l'equivalenza dei contenuti informativi al rispetto di regole semplici e rigide, si identificano le liste fornite dalle precedenti espressioni e la lista i cui items sono terne di items della $\mathbf{A}$.

A questa lista si dà il nome di **terza potenza** o **potenza cubica** della $\mathbf{A}$ e la si denota con $\mathbf{A}^{\times 3}$ o più semplicemente con $\mathbf{A}^3$.

Procedendo in questa direzione si definisce per n arbitrario intero positivo la potenza n-esima di una lista $\mathbf{A}$, $\mathbf{A}^{\times n}$ o più semplicemente $\mathbf{A}^n$; si tratta della lista i cui items sono tutte le n-uple di items della $\mathbf{A}$.

B06 e. prodotto di numeri naturali

B06 e.01 Ricordiamo che il cardinale di una lista L è il numero dei suoi items e conveniamo di denotare questo numero naturale non solo con $|\mathsf{L}|$ e con $\mathsf{lenl}(\mathsf{L})$, ma anche con $\mathsf{L}^{\#}$ o con $\mathrm{card}(\mathsf{L})$.

Due liste con lo stesso cardinale si dicono **liste equicardinali**.

Consideriamo due liste A avente cardinale A e B contenente B items ed esaminiamo il loro prodotto cartesiano $\mathsf{A} \times \mathsf{B}$.
Consideriamo inoltre la lista C avente lo stesso cardinale di A ($|\mathsf{C}| = A$) e la lista D equicardinale di B ($|\mathsf{D}| = B$)

Evidentemente il prodotto cartesiano $\mathsf{C} \times \mathsf{D}$ presenta lo stesso numero di componenti di $\mathsf{A} \times \mathsf{B}$, in quanto si individua facilmente una corrispondenza biunivoca tra gli items delle liste dei due prodotti.

Possiamo quindi affermare per quattro liste arbitrarie A, B, C e D:

$$|\mathsf{A}| \;=\; |\mathsf{C}| \;,\; |\mathsf{B}| \;=\; |\mathsf{D}| \;\;\Longrightarrow\;\; |\mathsf{C} \times \mathsf{D}| \;=\; |\mathsf{A} \times \mathsf{B}| \;.$$

In altre parole il cardinale di un prodotto cartesiano di liste dipende solo dali cardinali delle sue liste fattori e non da altre loro caratteristiche. Questo giustifica la definizione che segue.

Si dice **prodotto di due interi positivi** h e k il cardinale di ogni prodotto cartesiano di liste $\mathsf{A} \times \mathsf{B}$, nel quale A è una qualsiasi lista di h items e B è una qualsiasi lista di k items. Tale numero positivo lo denotiamo con $h \cdot k$.

Si completa la definizione di prodotto di due interi naturali, in accordo con le uguaglianze d07(2) chiedendo che per ogni coppia di interi positivi h e k si abbia

$$0 \cdot k \;:=\; h \cdot 0 \;:=\; 0 \cdot 0 \;:=\; 0 = |\emptyset| \;. \tag{1}$$

La notazione $h \cdot k$ per il prodotto degli interi naturali h e k spesso si semplifica con la notazione $h\,k$; in taluni contesti, in particolare nei documenti scritti a mano, risultano convenienti anche notazioni della forma $h{\times}k$.

Si può quindi enunciare

$$|\mathsf{A}| \cdot |\mathsf{B}| \;:=\; |\mathsf{A}|\,|\mathsf{B}| \;:=\; |\mathsf{A}|{\times}|\mathsf{B}| \;:=\; |\mathsf{A} \times \mathsf{B}| \;. \tag{1}$$

Va anche segnalato che nell'ambito di molti linguaggi di programmazione procedurali il prodotto dei numeri forniti, risp., da due identificatori `h` e `k` si denota scrivendo `h*k` .

In effetti all'interno di un programma è necessario che ogni richiesta di una operazione prodotto (come ogni richiesta di ogni altra operazione numerica e non) sia formulata in modo da poter essere interpretata correttamente da programmi traduttori efficienti e veloci.
Si consente sempre, per non ridurre l'espressività del linguaggio di programmazione, che gli identificatori delle variabili su valori numerici (e di ogni altro genere) possano essere costituiti da più caratteri, mentre al contrario nelle formule matematiche, grazie alla disponibilità di diverse fonti tipografiche, è quasi sempre sufficiente (e opportuno dal punto di vista della leggibilità) identificare le variabili con una sola lettera.

Per esempio in molti linguaggi di programmazione due espressioni matematiche come $a\,x + b\,y + c$ e $(a{+}x^2){-}3(c{+}y^2)\,z\,w$ si traducono, risp., con le scritture `a*x+b*y+c` e `(a+x*x)-3*(c+y*y)*z*w`.

B06 e.02 Vedremo che l'operazione prodotto si estende dai numeri interi naturali ad una vasta gamma di entità (numeri razionali, reali, complessi, trasformazioni, ... su molte delle quali, come per gli interi naturali, si può effettuare anche una operazione binaria chiamata somma che in genere è commutativa e la cui riduzione unaria è dotata di inverso o opposto.

Quando vedremo le regole formali del prodotto di interi naturali sarà opportuno tenere conto che in buona parte valgono per insiemi numerici molto estesi (numeri razionali, numeri reali, numeri complessi e non solo).

La portata e l'utilità delle operazioni chiamate prodotto risulteranno ben chiari dal suo impiego per la definizione delle notazioni posizionali [B11d], nelle svariate **formule enumerative** [D20] e in buona parte di tutte le formule matematiche e fisiche [tomo F, tomo W].

Se $|\mathsf{a}| = \text{ıııı}$ e $|\mathsf{b}| = \text{ııııııııı}$, per il cardinale di $\mathsf{a} \times \mathsf{b}$ si può scrivere:

$$\text{ııııııııı ııııııııı ııııııııı ııııııııı} \;=\; \text{ıııı} \cdot \text{ııııııııı} \;.$$

Già questo semplice esempio mostra che un numero positivo un po' grande quando viene espresso come prodotto di due interi sensibilmente inferiori può essere individuato da una scrittura unadica molto più corta e più facilmente interpretabile a vista della pura scrittura unadica.

Questo corrisponde al fatto che un insieme costituito da un numero esprimibile come prodotto di oggetti identici (di tacche, di quadratini, di pallini, ...) può essere presentato in modo visivamente più chiaro con i suoi elementi schierati a griglia o a matrice piuttosto che allineati (o, peggio, collocati a caso).

Queste considerazioni fanno intravvedere che servendosi della nozione di prodotto si arrivi a notazioni per i numeri naturali in grado di trattare anche interi positivi molto grandi che risultano sensibilmente più maneggevoli delle rappresentazioni unadiche.
Tuttavia prima di precisare queste notazioni conviene ampliare le conoscenze che riguardano il prodotto cartesiano e il prodotto di interi naturali.

B06 e.03 I prodotti cartesiani $\mathsf{A} \times \mathsf{B}$ e $\mathsf{B} \times \mathsf{A}$, dato che sono messi in corrispondenza biunivoca dalla trasformazione che a ciascuna coppia $\langle a_i, b_j \rangle$ associa la riflessa $\langle b_j, a_i \rangle$, hanno cardinali uguali:

$$|\mathsf{A} \times \mathsf{B}| \;=\; |\mathsf{B} \times \mathsf{A}| \;.$$

Da qui si ricava la **proprietà commutativa del prodotto** di naturali

(1) $$\text{per } \; h \; \text{ e } \; k \text{ naturali qualsiasi } \; : \; \; h \cdot k = k \cdot h \;.$$

In particolare si ha

(2) $$h \cdot 1 = 1 \cdot h = h \;.$$

Ricordiamo inoltre [a07] che

(3) $$h \cdot 0 = 0 \cdot h = 0 \;.$$

La proprietà commutativa del prodotto spesso viene presentata con la frase "scambiando l'ordine dei fattori il prodotto non cambia".

B06 e.04 Si considerino due liste A e B; si suddivida B come giustapposizione di due sottoliste $\mathsf{B} = \mathsf{B}_1 \,, \mathsf{B}_2$ e per i cardinali in gioco si scriva $h := |\mathsf{A}|$, $k := |\mathsf{B}|$, $k_1 := |\mathsf{B}_1|$ e $k_2 := |\mathsf{B}_2|$.

Si trova allora $\mathsf{A} \times \mathsf{B} \;=\; (\mathsf{A} \times \mathsf{B}_1) \,, (\mathsf{A} \times \mathsf{B}_2)$.

Passando ai cardinali si ottiene

$$h \cdot (k_1 + k_2) \;=\; h \cdot k_1 + h \cdot k_2 \;. \tag{1}$$

Grazie alla arbitrarietà delle liste in gioco questa uguaglianza vale per qualsiasi scelta degli interi positivi h, k_1 e k_2.

Si verifica facilmente che essa vale anche quando uno degli interi in gioco è uguale a 0; quindi essa vale per qualsiasi scelta di h, k_1 e k_2 tra gli interi naturali.

Consideriamo poi una suddivisione della prima lista come $\mathsf{A} \;=\; \mathsf{A}_1 \,, \mathsf{A}_2$ e introduciamo $h_1 := |\mathsf{A}_1|$ e $h_2 := |\mathsf{A}_2|$.

Con considerazioni analoghe alle precedenti si trova

$$(h_1 + h_2) \cdot k \;=\; h_1 \cdot k + h_2 \cdot k \;, \tag{2}$$

uguaglianza valida per h_1, h_2 e k interi naturali qualsiasi.

Le uguaglianze (1) e (2) sono ottenibili l'una dall'altra grazie alla proprietà commutativa del prodotto dei numeri naturali [e03] e in virtù della arbitrarietà 'a delle liste coinvolte.

Si dice che queste uguaglianze esprimono la proprietà di **distributività della somma rispetto al prodotto** per gli interi naturali.

Casi particolari come $6 \cdot (4 + 3) = 6 \cdot 4 + 6 \cdot 3$ e come $(5 + 3) \cdot 6 = 5 \cdot 6 + 5 \cdot 3$ si possono leggere agevolmente in figure facilmente interpretabili.

Occorre osservare che per il significato delle espressioni nei primi membri delle uguaglianze precedenti si è assunto, senza dichiararlo esplicitamente, che all'operatore prodotto viene attribuita la precedenza di esecuzione rispetto all'operatore somma.

Questa nozione di precedenza di esecuzione verrà approfondita in C14.

Aggiungiamo l'uguaglianza riguardante la separazione in due sottoliste sia della A che della B derivabile da (1) e (2):

$$(h_1 + h_2) \cdot (k_1 + k_2) \;=\; h_1 \cdot k_1 + h_1 \cdot k_2 + h_2 \cdot k_1 + h_2 \cdot k_2 \;. \tag{3}$$

B06 e.05 A partire da tre liste A, B e C si possono individuare due costruzioni cartesiane in corrispondenza con le seguenti scritture nelle quali i delimitatori coniugati per le sottoespressioni "(" e ")", come spesso accade, servono per imporre che le operazioni entro ciascuna coppia di parentesi siano eseguite prima di quelle che compaiono al di fuori di essa.

$(\mathsf{A} \times \mathsf{B}) \times \mathsf{C}$, insieme delle terne rappresentabili con le scritture $\Big\langle \langle \mathtt{a}, \mathtt{b} \rangle, \mathtt{c} \Big\rangle$ e

$\mathsf{A} \times (\mathsf{B} \times \mathsf{C})$, insieme delle terne rappresentabili con le scritture $\Big\langle \mathtt{a}, \langle \mathtt{b}, \mathtt{c} \rangle \Big\rangle$,

dove $\mathtt{a}$, $\mathtt{b}$ e $\mathtt{c}$ denotano elementi qualsiasi, risp., di A, di B e di C.

È immediato constatare che le coppie di espressioni aventi, risp., le forme $\Big\langle \langle \mathtt{a}, \mathtt{b} \rangle, \mathtt{c} \Big\rangle$ e $\Big\langle \mathtt{a}, \langle \mathtt{b}, \mathtt{c} \rangle \Big\rangle$ individuano una corrispondenza biunivoca tra $(\mathsf{A} \times \mathsf{B}) \times \mathsf{C}$ e $\mathsf{A} \times (\mathsf{B} \times \mathsf{C})$.

Essa è dovuta alla presenza nella prima posizione di un qualsiasi componente $\mathtt{a} \in \mathsf{A}$, nella seconda di un qualsiasi $\mathtt{b} \in \mathsf{B}$ e nella terza di un arbitrario $\mathtt{c} \in \mathsf{C}$.

Queste presenze portano ad altre due corrispondenze biunivoche tra ciascuna delle liste dei suddetti costrutti con la lista delle terne $\langle \mathtt{a}, \mathtt{b}, \mathtt{c} \rangle$ che può essere denotata con $\mathsf{A} \times \mathsf{B} \times \mathsf{C}$ e può essere considerata come una composizione (cartesiana) di tre liste.

I componenti delle tre composizioni cartesiane delle tre liste si possono raffigurare con una certa efficacia come cubetti identici disposti in modo da far combaciare le facce di due di tali solidi adiacenti; la figura ottenuta viene detta **griglia cuboidale** [v.a. G36d07]. Nel caso di tre insiemi fattori aventi cardinali, risp., 2, 3 e 4 si ha la figura

Passando ai cardinali, assumendo che $h := |\mathsf{A}|$, $k := |\mathsf{B}|$ ed $l := |\mathsf{C}|$, essendo A, B e C liste qualsiasi, si ha che per h, k ed l interi naturali qualsiasi

$$(h \cdot k) \cdot l \;=\; h \cdot (k \cdot l)\ . \tag{1}$$

Questa uguaglianza esprime la cosiddetta proprietà di **associatività del prodotto** di interi naturali. Questa relazione rende lecito esprimere l'intero naturale precedente scrivendo semplicemente $h \cdot k \cdot l$, oppure $h{\times}k{\times}l$ o ancor più concisamente $h\,k\,l$; in un linguaggio di programmazione procedurale si avrà h*jk*l .

Tre composizioni come le precedenti spesso si possono identificare senza generare confusione: effettuando un cosiddetto **abuso di notazione** veniale, si può scrivere

$$\mathsf{A} \times \mathsf{B} \times \mathsf{C} \;=\; (\mathsf{A} \times \mathsf{B}) \times \mathsf{C} \;=\; \mathsf{A} \times (\mathsf{B} \times \mathsf{C})\ . \tag{2}$$

Accettando queste identificazioni si attribuisce l'associatività anche al prodotto cartesiano.

Si osservi che la associatività del prodotti cartesiano e del prodotto numerico dipendono dalla fondamentale e ben comprensibile associatività della giustapposizione tra stringhe.

B06 f. potenze cartesiane di liste e potenze di numeri naturali

B06 f.01 La definizione in c01 consente di considerare anche il prodotto cartesiano di una lista nonripetitiva **A** per se stessa $\mathbf{A}^{\times 2}$; tale costruzione viene detta **quadrato cartesiano** o **seconda potenza cartesiana** della lista **A**.

È utile considerare anche le successive potenze cartesiane delle liste nonripetitive:

$\mathbf{A}^{\times 3} := (\mathbf{A} \times \mathbf{A}) \times \mathbf{A}$, **terza potenza cartesiana**;
$\mathbf{A}^{\times 4} := ((\mathbf{A} \times \mathbf{A}) \times \mathbf{A}) \times \mathbf{A}$, **quarta potenza cartesiana**;
. . . . . .
$\mathbf{A}^{\times d} := \mathbf{A}^{\times (d-1)} \times \mathbf{A}$, d-**esima potenza cartesiana** per $d = 1, 2, 3, \ldots$.

Da questa ultima richiesta discende l'opportunità di stsbilire che la scrittura $\mathbf{A}^{\times 1}$ equivalga alla **A**.

Conviene inoltre definire come potenza cartesiana nulla $\mathbf{A}^{\times 0} := \{ \sqcup \}$.

Applicando ripetutamente l'associatività si ottiene la seguente sua generalizzazione.
Per d ed e interi positivi arbitrari si ha:

(1) $$\mathbf{A}^{\times d} \times \mathbf{A}^{\times e} = \mathbf{A}^{\times (d+e)} .$$

Nelle formule nelle quali si possono distinguere con chiarezza gli insiemi dai numeri in gioco, le precedenti notazioni si possono semplificare scrivendo $\mathbf{A}^{d}$ invece di $\mathbf{A}^{\times d}$.

B06 f.02 Gli elementi della generica potenza cartesiana d-esima di un alfabeto A si possono identificare con le stringhe su A aventi lunghezza d. Per esempio se $\mathsf{A} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$, i 27 items di A^3, $\langle \mathsf{a}, \mathsf{a}, \mathsf{a} \rangle$, $\langle \mathsf{a}, \mathsf{a}, \mathsf{b} \rangle$, $\langle \mathsf{a}, \mathsf{a}, \mathsf{c} \rangle$, $\langle \mathsf{a}, \mathsf{b}, \mathsf{a} \rangle$, $\langle \mathsf{a}, \mathsf{b}, \mathsf{b} \rangle$,... , $\langle \mathsf{c}, \mathsf{c}, \mathsf{b} \rangle$ e $\langle \mathsf{c}, \mathsf{c}, \mathsf{c} \rangle$ si possono identificare, risp., con i trigrammi `aaa, aab, aac, aba, abb, ... , ccb` e `ccc` .

Quindi l'insieme delle stringhe di lunghezza d sull'alfabeto A si può individuare con la scrittura $\mathsf{A}^{\times d}$ o con la sua semplificazione A^{d}, cioè con la potenza cartesiana d-esima dell'alfabeto.

La precedente uguaglianza f01(1) equivale ad affermare, in tutta evidenza, che ogni stringa di lunghezza $d + e$ su un qualsiasi alfabeto A si può ottenere come giustapposizione di una stringa di lunghezza d e di una stringa di lunghezza e.

A questo punto risulta chiaramente giustificata la definizione di potenza cartesiana nulla di un qualsiasi alfabeto A come la lista costituita dalla sola stringa muta $\sqcup$: questa infatti è l'unica stringa di lunghezza 0 sopra un qualsiasi alfabeto.
L'oggetto formale $\sqcup$ si può definire anche come unica lista in grado di rappresentare la potenza cartesiana nulla di una qualsiasi lista.

B06 f.03 Fissati due interi naturali d maggiore di 2 ed h, esaminiamo i cardinali delle potenze cartesiane seconda, terza, ..., d-esima di una lista **A** che presenta h items.
Questi cardinali sono forniti, risp., dai prodotti di due, tre, ..., d repliche dell'intero naturale h e vengono chiamati,risp., seconda, terza, ..., d-esima **potenza** dell'intero h.
Per questi numeri si scrive:

(1) $$h^2 := h \cdot h \qquad h^3 := h \cdot h \cdot h \qquad \ldots \qquad h^d := h^{d-1} \cdot h .$$

Passando dalle liste ai cardinali, dalla

$$\mathbf{A}^{\times (d+e)} = \mathbf{A}^{\times d} \times \mathbf{A}^{\times e}$$

si ricava la seguente uguaglianza tra espressioni numeriche:

$$h^{d+e} \;=\; h^d \cdot h^e \;. \tag{2}$$

Per aumentare la portata della notazione esponenziale si pone $h^1 := h$ e, in accordo con la $\mathsf{A}^0 = \{\mu\}$, si definisce $h^0 := 1$.

Queste definizioni si giustificano pienamente sul piano della efficacia espositiva con la possibilità di estendere la validità della $h^{d+e} \;=\; h^d \cdot h^e$ a ogni scelta degli esponenti d ed e tra gli interi naturali.

In particolare abbiamo $0 \cdot 0 = 0$, $0^3 = 0$ e $0^d := 0$ per ogni d intero positivo.

B06 f.04 Si potrebbe anche cercare di attribuire un intero naturale come valore della scrittura 0^0; una tale attribuzione risulterebbe utile se si accordasse con le altre uguaglianze generali per il prodotto e la somma di naturali e con la precedente $0^d = 0$.

Si verifica però che ogni scelta in proposito porta a conclusioni in conflitto con qualche formula.

Se si pone $0^0 \;:=\; 0$ cercando di estendere la $0^d = 0$ a ogni d naturale, si ha una situazione discorde dalla $h^0 = 1$ per ogni h intero positivo; se si pone $0^0 \;:=\; 1$, cercando di estendere la $h^0 = 1$ a ogni h naturale, si ha una situazione discorde dalla $0^d = 0$ per ogni d intero positivo.

Quindi non è possibile attribuire un valore numerico alla scrittura $0^0 = 0$ che abbia utilità strumentale e che non presenti incoerenze.

Va osservato tuttavia che Donald Knuth (v. *Concrete Mathematics* 5.1) suggerisce che nell'ambito della combinatorica si adotti la definizione $0^0 := 1$, in modo da estendere la funzione polinomiale x^0 per x numero reale, numero complesso o variabile formale, entità giudicata molto più utile della eventuale alternativa offerta dalla funzione 0^x.

In effetti la funzione x^0 si può trattare vantaggiosamente insieme agli altri polinomi [B33], in particolare con quelli della forma x^n, e va considerato che i polinomi sono strumenti molto efficaci per affrontare numerosi problemi di matematica sia discreta che continua [W60c].

Ricordiamo anche che nell'ambito dell'analisi matematica la scrittura 0^0 associata a un valore reale limite individua una cosiddetta **forma indeterminata** [I17], entità che viene definita più opportunamente come insieme di specifiche funzioni di variabile reale.

B06 g. lista delle stringhe di fissata lunghezza

B06 g.01 In questa sezione ci serviamo di una generica stringa $\mathsf{a} = a_1\, a_2 \,...\, a_n$ nonripetitiva, che potremmo anche chiamare alfabeto ordinato, e della corrispondente lista nonripetitiva di caratteri $\mathsf{A} = \langle a_1, a_2, ..., a_n \rangle$, lista portatrice di informazioni equivalenti a quelle fornite dalla stringa e che per semplicità identifichiamo con essa.

A questi due oggetti formali è utile associare tre connettivi relazionali decidibili strettamente collegati riguardanti gli a_i che qui scriviamo, eisp., $\prec_{\mathtt{I}}$, $\prec$ e $\preceq$, sottintendendo il fatto che dipendono da a (o da A).

Cominciamo con il presentare $\prec_{\mathtt{I}}$; si tratta di un connettivo che può essere definito con una lista completa delle coppie di caratteri in a $\langle a_i, a_{i+1} \rangle$ che la costituiscono, a sua volta equivalente alla lista di tutte le terne della forma $a_i \;\prec_{\mathtt{I}}\; a_{i+1}$.
Esplicitamente questa lista è

$$\Big\langle \langle a_1, a_2 \rangle; \langle a_2, a_3 \rangle; ...; \langle a_{n-1}, a_n \rangle \Big\rangle \ .$$

Per essa useremo anche la scrittura

$$\langle a_1 \prec_{\mathtt{I}} a_2 \prec_{\mathtt{I}} ... \prec_{\mathtt{I}} a_n \rangle \ ,$$

che si può considerare una sua abbreviazione.

Il segno $\prec_{\mathtt{I}}$ si dice denotare la **relazione di precedenza immediata** tra i caratteri a_1, a_2, ..., a_n.

La relazione $\preceq$ viene definita con la seguente lista di coppie derivabile dalla precedente (estesa in modo da ottenere la riflessività e la transitività) e che conviene visualizzare con una griglia triangolare

$$\begin{array}{ccccccc}
\langle a_1, a_1 \rangle & \langle a_1, a_2 \rangle & ... & \langle a_1, a_{h-1} \rangle & \langle a_1, a_h \rangle \\
 & \langle a_2, a_2 \rangle & ... & \langle a_2, a_{h-1} \rangle & \langle a_2, a_h \rangle \\
 & & & \cdots & \cdots \\
 & & & \langle a_{h-1}, a_{h-1} \rangle & \langle a_{h-1}, a_h \rangle \\
 & & & & \langle a_h, a_h \rangle
\end{array}$$

La relazione denotata con $\preceq$ viene detta **relazione di ordine totale** tra i caratteri di una stringa a.
Si osserva che si può definire anche con l'enunciato

$$\text{per} \;\; i, j = 1, 2, ..., h \;\; : \;\; a_i \preceq a_j \;\; \text{sse} \;\; i \leq j \ .$$

Il termine **ordine totale** sta a indicare che dati due caratteri diversi a_i e a_j di a, si può sempre decidere se $a_i \preceq a_j$ oppure $a_j \preceq a_i$.

Infine definiamo $\prec$ chiedendo che per ogni $\langle a_i, a_j \rangle$ componente di a^2 sia $a_i \;\prec\; a_j$ sse $a_i \;\preceq\; a_j$ e $a_i \neq a_j$.

Facendo ricorso alla nozione di chiusura che sarà esaminato in B54e, il connettivo $\preceq$ si dice **chiusura transitiva riflessiva** di $\prec_{\mathtt{I}}$ e il connettivo $\prec$ si dice **chiusura transitiva** di $\prec_{\mathtt{I}}$.

Si osserva che da ciascuno dei tre connettivi si possono costruire gli altri due e per questo diciamo che si tratta di tre **entità informativamente equivalente**.

La definizione di $\prec$ equivale a chiedere che valgano tutte le affermazioni $a_i \prec a_{i+h}$ per ogni $i = 1, ..., n-2$ e ogni $h = 1, ..., n - i$.

Da queste richieste discende che, individuati con due diversi processi due caratteri a_i e a_j, consultando la a si riesce a decidere se $a_i \prec a_j$, oppure se $a_j \prec a_i$, oppure se $a_i = a_j$.

Una tale decisione si dice anche "risoluzione di un trilemma" o "precisazione di una tricotomia".

B06 g.02 Consideriamo il prodotto cartesiano $\mathsf{A} \times \mathsf{A}$ e identifichiamolo con la lista di digrammi

$$\mathsf{a} \times \mathsf{a} \;= \big\langle a_1\,a_1, a_1\,a_2, ..., a_1\,a_n, a_2\,a_1, ..., a_2\,a_n, ... a_n\,a_1, ..., a_n\,a_n \big\rangle$$

Si osserva che la sequenza degli items di questa lista è tale che si può passare da uno di essi (diverso dall'ultimo) al successivo con una manovra piuttosto semplice che si avvale della sola sequenzializzazione dei caratteri determinata dalla stringa $\mathsf{a} = a_1\,a_2\,...\,a_n$ e che si può effettuare la generazione di una lista equivalente a quelle considerate in precedenza per definire il prodotto cartesiano di liste.

Algoritmo: 1 Dato il digramma $a_{j_1}\,a_{j_2}$ si vuole ottenere il successivo.

Se $a_{j_2} \prec a_h$ il digramma successivo è $a_{j_1}\,a_{j_2+1}$.

Se $a_{j_2} = a_n$ e $a_{j_1} \prec a_n$, allora il successivo è $a_{j_1+1}\,a_1$.

Se $a_{j_2} = a_n$ e $a_{j_1} = a_n$ si enuncia che non esiste un digramma successivo di quello dato in quanto questo è l'ultimo della lista.

L'algoritmo che inizia con la emissione di $a_1\,a_1$ in quanto digramma iniziale e procede con la iterazione del predenente algoritmo di passaggio al digramma successivo seguito dalla sua emissione oppure dall'arresto per raggiunto ultimo digramma costituisce uno strumento che genera l'intera lista $\mathsf{a} \times \mathsf{a}$ a partire da a.
Un tale algoritmo viene detto **algoritmo di generazione sequenziale di digrammi** e determina un ordinamento totale tra i digrammi di a^2 che viene detto ordinamento lessicografico derivato da $\preceq$.

Si possono individuare facilmente alcune interessanti sottoliste di a^2:
la sottolista dei digrammi $a_{j_1}\,a_{j_2}$ tali che $a_{j_1} \prec a_{j_2}$;
la sottolista dei digrammi $a_{j_1}\,a_{j_2}$ tali che $a_{j_1} = a_{j_2}$;
la sottolista dei digrammi $a_{j_1}\,a_{j_2}$ tali che $a_{j_2} \prec a_{j_1}$.

La prima si può identificare con la relazione $\prec$, la seconda con la relazione identità tra caratteri di a e la terza con la relazione trasposta della $\prec$ per la quale si adotta la notazione $\succ := \prec^{\mathsf{T}}$.

Si può inoltre prendere in considerazione la sottolista dei digrammi $a_{j_1}\,a_{j_2}$ tali che $a_{j_1} \prec a_{j_2}$ o $a_{j_1} = a_{j_2}$, che identifichiamo con il connettivo relazionale $\preceq$.
Infine può servire la relazione trasposta della $\preceq$ che in genere si denota con $\succeq$, cioè con la relazione ottenuta per riflessione di $\preceq$ rispetto alla diagonale della griglia di presentazione primaria di $\mathsf{a} \times \mathsf{a}$.

B06 g.03 Dopo la seconda potenza cartesiana di una stringa nonripetitiva di caratteri a è prevedibilmente utile, ossia naturale, proporsi la costruzione delle successive potenze cartesiane a^d con d arbitrario intero maggiore di 1.

Per il generico item di questa lista, cioè la generica stringa di d caratteri che occorrono nella a usiamo la scrittura $a_{j_1}\,a_{j_2}\,...\,a_{j_d}$ nella quale gli j_i denotano interi ai quali si chiede di assumere tutti i valori di $\{1, 2, ..., n\}$; si dice anche che gli j_i corrono sull'intero $\{1, 2, ..., n\}$.
Queste stringhe (o le equivalenti liste) nella analisi combinatoria sono dette **disposizioni con ripetizione** dei caratteri nell'alfabeto a [B13a]. Si osserva chequesta espressione sostituisce la denominazione più corretta "disposizioni con possibili ripetizioni" giudicata troppo prolissa.

La costruzione della lista si può effettuare con un algoritmo che estende quello proposto per la generazione della lista lessicografica del quadrato cartesiano a^2.

Per un particolare d si potrebbero organizzare d iterazioni annidate, la primaria si occupa della emissione dei caratteri nella prima posizione, la iterazione sua subordinata della emissione dei caratteri

nella seconda posizione e così via fino alla d-esima iterazione che effettua la emissione dei caratteri nell'ultima posizione.

Conviene tuttavia adottare un algoritmo per la generazione sequenziale che rispetto al precedente ha maggiore portata in quanto non è vincolato a un d prefissato e risulta più compatto.

Esso assume come informazioni caratterizzanti la istanza del problema che risolve, la lista a e il parametro dimensionale d.

Con una prima fase genera la prima stringa costituita da d repliche del primo carattere di a, cioè emette ${a_1}^d$.

Successivamente organizza la trasformazione di un d-gramma precedentemente generato nel successivo che poi si preoccupa di accodare come nuovo item della lista che sta costruendo.

Questo passaggio a una nuova stringa risulta possibile fino a che si cerca di modificare l'ultima stringa, che deve essere la ${a_n}^d$, e quando si riscontra tale impossibilità viene decisa la conclusione della generazione della lista richiesta.

Ci resta dunque di presentare la manovra centrale consistente nella trasformazione di una stringa $w = a_{j_1}\, a_{j_2} \ldots a_{j_d}$ nella successiva.

Partendo dalla ultima posizione, la d, si cerca di trasformare a_{j_d} nel successivo carattere a_{j_d+1}.
Questo è possibile solo se $a_{j_d} \prec a_h$ e in tal caso si emette $a_{j_1}\, a_{j_2} \ldots a_{j_d}{+}1$.

Se invece $a_{j_d} = a_h$ si arretra nella w fino a trovare il primo carattere dalla destra a_{j_i} che precede a_h.

Questo lo si trova sse la w non è l'ultima stringa $w = {a_h}^d$ e in tal caso si modifica il carattere nella posizione i nel successivo $a_{j_i} + 1$ e si pongono occorrenze di a_1 nelle posizioni successive emettendo come nuova stringa $aj_1 \ldots a_{j_i} + 1\, {a_1}^{d-i}$.

Se invece si è cercato di modificare la $w = {a_h}^d$, la ricerca verso sinistra di un a_h non ha successo e si conclude la generazione della lista a^d.

Anche le stringhe che sono items della a^d sono in un ordine chiamato ordinamento lessicografico derivato da $\preceq$, a sua volta determinato da a. Questa relazione d'ordine sequenziale la denotiamo con $\preceq_{lxg}$ e va considerata una estensione della $\preceq$.

B06 g.04 Presentiamo anche l'algoritmo tricotomico che, date due stringhe u e v di lunghezza d di caratteri di a, stabilisce se $u \prec_{lxg} v$, se $u = v$ oppure se $v \prec_{lxg} u$.

Esso si serve di una iterazione che effettua lo scorrimento parallelo da sinistra a destra delle posizioni delle due stringhe fino alla prima posizione nella quale i due caratteri trovati sono diversi.

Se questa posizione non si trova le due stringhe coincidono.
Se la si trova si decide che viene prima la stringa nella quale il primo carattere non coincidente viene prima nella a.

Evidentemente il cardinale della a^d è $h^d = |\mathsf{a}|^d$.

Conviene ribadire esplicitamente che i connettivi $\prec_{\mathtt{I}}$, $\prec$, $\preceq$ e le loro trasposte dipendono da a, ovvero da A e che questa dipendenza può essere esplicitata usando notazioni più complete, per esempio ${\prec^{\mathsf{a}}}_I$, $\prec^{\mathsf{a}}$ e $\preceq^{\mathsf{a}}$.
In genere sono da preferire le notazioni più semplici per avere espressioni più leggibili.

B06 g.05 Interessano i casi speciali delle sottoliste di stringhe ottenibili selezionando le stringhe di a^d con algoritmi relativamente semplici che richiedono il soddisfacimento di proprietà collegabili a caratteristiche interessanti per le applicazioni.

Si distinguono facilmente le stringhe che non presentano ripetizioni.
Evidentemente queste devono avere lunghezze d minori o uguali al numero $|\mathsf{a}|$ dei caratteri disponibili.
Queste stringhe in combinatorica sono dette **disposizioni senza ripetizioni**.

Altri casi particolari di stringhe interessanti sono individuati tenendo conto dell'ordinamento dei caratteri di un alfabeto fornito da una stringa nonripetitiva.

Per questa distinzione risultano molto chiare le stringhe di numeri positivi.
Si dicono stringhe crescenti su un alfabeto ordinato le stringhe $a_{j_1}\, a_{j_2} \ldots a_{j_s}$ tali che

$$a_{j_1} \prec a_{j_2} \prec \ldots \prec a_{j_{s-1}} \prec a_{j_s} \ .$$

Si dicono invece stringhe nondecrescenti su un alfabeto ordinato le stringhe $a_{j_1}\, a_{j_2} \ldots a_{j_s}$ tali che

$$a_{j_1} \preceq a_{j_2} \preceq \ldots \preceq a_{j_{s-1}} \preceq a_{j_s} \ .$$

Ovviamente l'insieme delle prime è un sottoinsieme proprio dell'insieme delle stringhe nondecrescenti (almeno se $s \geq 2$).

Evidentemente la lista delle stringhe riflesse-I di una sequenza di stringhe crescenti costituisce una sequenza di stringhe decrescenti e la sequenza delle stringhe riflesse-I di una sequenza di stringhe noncrescenti costituisce una sequenza di stringhe nondecrescenti.

Testo fruibile in https://www.mi.imati.cnr.it/alberto/ e https://arm.mi.imati.cnr.it/Matexp/matexp_main.php