1

Capitolo C10
stringhe e linguaggi formali

Contenuti delle sezioni

34 pagine

C100.01 Questo è il primo dei capitoli del tomo C dedicato ai linguaggi formali, alle macchine sequenziali (o automi), alle considerazioni generali sulle attività computazionali e sulla loro complessità e alla introduzione della teoria dei codici e della crittografia.

Questi capitoli si sforzano di essere autonomi e il presente primo capitolo espone con una certa completezza le nozioni basilari su stringhe e linguaggi, già presentate come elementi iniziali dell'*esposizione* in B01 e riprese successivamente quando servivano a introdurre altre entità che si possono basare solo sulle stringhe e sulle liste [B06, B17].

Successivamente vengono presentati i linguaggi della gerarchia di Chomsky insieme alle macchine che li caratterizzano [C12-C16], fino a giungere alla macchine di Turing e alla computabilità [C20, C21].

Dopo aver introdotte alcune collezioni di linguaggi controllati con grammatiche e meccanismi meno classici [C23, C26], si esamina la fattorizzazione dei linguaggi [C30] e si espongono teorie più astratte sui linguaggi [C32-C35].

In C47 viene introdotto lo studio delle complessità computazionale e in C60 sono presentate alcune parole infinite.

Gli ultimi capitoli [C65, C66] sono dedicati alla introduzione delle probabilità discrete e dei codici.

C10 a. stringhe, linguaggi e monoidi liberi

C10a.01 In molte attività conoscitive e in molte manipolazioni materiali si devono esaminare sequenze finite di oggetti (più o meno compositi) curandosi solo delle posizioni che essi occupano nelle sequenze stesse.
Gli oggetti componenti di queste sequenze vengono individuati da alcune caratteristiche che sono giudicate sostanziali in dipendenza dagli scopi per i quali gli oggetti stessi saranno utilizzati.
Sulla base di queste caratteristiche due oggetti di una sequenza possono essere giudicati diversi o indistinguibili; nel secondo caso si parla di repliche di un unico tipo di oggetto.

Tra le attività concernenti sequenze di oggetti vi sono le manipolazioni di elenchi di parole, di locuzioni, di frasi o di espressioni che fanno parte di lingue naturali o artificiali.
In questi casi gli oggetti componenti di sequenze, chiamati anche items di liste, vengono considerati elementari e si possono identificare con le loro scritture.

Per queste scritture in genere si auspica costituiscano dei validi significanti in grado di evitare dubbi identificativi e di facilitare la comprensione dei significati e dei ruoli.

Altre attività riguardano sequenze di oggetti sensibilmente complessi per i quali si chiede comunque che si possano rappresentare mediante scritture convenzionali che facciano riferimento a definizioni modellistiche anche articolate (che sul piano pratico possono essere soggette a discussioni).

Alcuni esempi: un elenco anagrafico, l'elenco dei processori costituenti un sistema multiprocessore, la sequenza dei residui di aminoacidi costituenti una proteina di medie dimensioni (10000 residui), un catalogo astronomico, un listino di prodotti tecnologici portatori di interrelazioni importanti per le applicazioni prevedibili.

In relazione agli oggetti ai quali si possono associare per le loro possibili applicazioni, per le sequenze che stiamo prospettando, per le loro componenti che sono considerate elementari e per gli oggetti tangibili dei quali queste componenti sono i modelli proponiamo, risp., i termini **sequenze rappresentative**, **items rappresentativi** e **oggetti rappresentati**.

C10a.02 Accade che nelle varie aree disciplinari dei molti generi di sequenze rappresentative vengono effettute e comunicate analisi nelle quali ciascun item, per quanto complesso possa essere l'oggetto tangibile che esso rappresenta, quando lo si incontra viene trattato solo in quanto segno (unità semiologica) che serve solo a distinguerlo dagli items da considerare diversi e ad assimilarlo agli items che si è convenuto di considerare indistinguibili.

Quando si dovesse andare più a fondo sugli oggetti tangibili rapresentati si dovrebbe ricorrere alle precisazioni delle relazioni fra oggetti rappresentati e items rappresentativi secondo il modello applicativo adottato.

Per completezza va detto che nelle scritture della tradizione pittografica e ideografica (in particolare nelle scritture della tradizione cinese) le unità simboliche hanno anche il compito di richiamare, almeno per sommi capi, gli oggetti rappresentati.

Conviene segnalare esplicitamente che l'ordine secondo il quale gli items compaiono nelle sequenze può essere collegato a motivi molto diversi: disposizione temporale, collocazione spaziale di qualche genere, ordinamento secondo vari tipi di valutazione e altro.
Questa varietà di situazioni corrisponde alla amplissima utilizzazione delle sequenze nelle attività e negli studi.

C10a.03 Nell'ambito delle considerazioni sulle sequenze rappresentative gli items loro componenti si possono qualificare come segni che sono stati dotati o che potrebbero essere dotati di significati univoci, ossia potrebbero qualificarsi come contrassegni.

In un complesso di attività che riguarda un corpo di sequenze rappresentative risulta utile raggruppare i segni degli items in insiemi che per realismo dobbiamo dichiarare finiti ma che possiamo pensare estendibili quando interessano eventi conoscitivi o costruttivi che si sviluppano nel tempo.

Questi raggruppamenti di segni sono realizzati o realizzabili mediante elenchi nonripetitivi che quando sono opportune precisazioni semantiche si possono riconoscere come entrate della prima colonna di una opportuna "tabella esplicativa". Questi elenchi nonripetitivi li chiamiamo **repertori di contrassegni**; in alcuni contesti è conveniente chiamarli **alfabeti**.

Per taluni complessi di attività questi repertori riguardano elenchi di poche decine di segni elementari. Tipici esempi sono l'elenco delle 26 lettere minuscole dell'alfabeto inglese, l'elenco dei 62 caratteri alfanumerici e l'elenco dei caratteri ASCII visualizzabili; questi elenchi presentano il vantaggio di essere pienamente gestibili dai più comuni strumenti informatico-telematici.
In questi casi i segni possono essere chiamati "caratteri" o "lettere". e i loro insiemi "alfabeti elementari".
Oltre agli alfabeti elementari sono in uso alfabeti di segni più elaborati, il più importante di questi è l'alfabeto *Unicode*.

Le sequenze finite di caratteri saranno dette **stringhe**; in alcuni ambiti disciplinari (algebra, logica) viene usato il termine **parole**. Ricordiamo che accanto alle parole, che sono sequenze di caratteri di lunghezza finita, sono studiate anche le parole infinite, che sono successioni di caratteri [C60].

C10a.04 Segnaliamo alcuni alfabeti elementari particolarmente utilizzati:

il singoletto contenente solo un particolare segno che potrebbe essere $\{\mid\}$ e chiamato "tacca";

l'alfabeto dei bits $\{0, 1\}$;

l'insieme delle cifre decimali;

l'insieme delle lettere utilizzate in una lingua naturale (che potrebbe comprendente o meno lettere dotate di segni diacritici e potrebbe distinguere o meno tra lettere minuscole e maiuscole);

l'insieme dei caratteri di un sistema standardizzato di codifica come `ASCII (wi)` o `Unicode (wi)`.

Ricordiamo per questi ultimi che tutti i caratteri ASCII si codificano con 7 o con 8 bits, mentre in Unicode i più basilari richiedono 16 bits e altri 32 e altri multipli di 16.
Va segnalato che in certe argomentazioni i caratteri ASCII o Unicode sono trattati come elementari, mentre in altre sono da considerare come stringhe sull'alfabeto dei bits.
Quale punto di vista scegliere dipende dai fini dello sviluppo argomemtativo che si sta effettuando e a queste scelte conviene lasciare libertà ed elasticità ai singoli esecutori di procedure o ai singoli redattori di documenti.

C10a.05 Per ragioni di praticità espositiva introduciamo il termine **insieme dei caratteri per alfabeti** ed il corrispondente simbolo $\boxminus$.

Questo consente di servirsi di scritture come $\mathsf{A} \subset_{\varphi} \boxminus$ e $\{\mathtt{a}_1, ..., \mathtt{a}_r\} \subset \boxminus$ per introdurre semplici notazioni per alfabeti (come A) e per caratteri (come gli $\mathtt{a}_i$).

$\boxminus$ va considerato insieme finito che potrebbe essere individuato da una ampia indagine sopra le attività SciTech, cioè sulle ricerche e sulle realizzazioni di carattere scientifico e tecnologico; esso inoltre va considerato un insieme in continua evoluzione.

Consideriamo dunque un alfabeto $\mathsf{A} = \{\mathsf{a}_1, ..., \mathsf{a}_n\} \subset_\varphi \boxplus$. Per la stringa esemplare su tale alfabeto usiamo la scrittura

$$w \;=\; \langle \mathsf{a}_{j_1}, \mathsf{a}_{j_2}, ..., \mathsf{a}_{j_r} \rangle \in \mathsf{A}^r \;,$$

ove r è un intero positivo e a_{j_1}, ...,a_{j_r} sono caratteri di A; r viene detto **lunghezza della stringa** w e questa quantità si può denotare con $|w|$, $w^{\vdash\dashv}(w)$ o $\mathsf{len}(w)$.

C10a.06 Chiamiamo **giustapposizione** o concatenazione delle stringhe $w = \langle \mathsf{a}_{j_1}, \mathsf{a}_{j_2}, ..., \mathsf{a}_{j_r} \rangle$ e $x = \langle \mathsf{a}_{h_1}, \mathsf{a}_{h_2}, ..., \mathsf{a}_{h_s} \rangle$ la stringa

$$w \shortmid x \;:= \big\langle \mathsf{a}_{j_1}, \mathsf{a}_{j_2}, ..., \mathsf{a}_{j_r}, \mathsf{a}_{h_1}, \mathsf{a}_{h_2}, ..., \mathsf{a}_{h_s} \big\rangle \;.$$

Evidentemente questa operazione binaria è associativa: se w, x e y denotano tre stringhe, la stringa ottenuta giustapponendo prima w ed x, poi $w \shortmid x$ con y si assume coincidere con la stringa ottenuta in un primo momento con la giustapposizione di x e y ed in un secondo momento con la giustapposizione di w con $x \shortmid y$.

Queste due costruzioni quando si vogliono distinguere è ragionevole denotarle, risp., con $(w \shortmid x) \shortmid y$ e con $w \shortmid (x \shortmid y)$.
Riconosciuta l'associatività della giustapposizione si può evitare la scrittura di alcune coppie di parentesi: invece di ciascuna delle precedenti scritture si può scrivere più semplicemente $w \shortmid x \shortmid y$.

Spesso si adotta anche la semplificazione consistente nel trascurare il segno che ricorda la giustapposizione e si scrive concisamente $w\,x\,y$.

In seguito svolgeremo varie considerazioni su stringhe e linguaggi di tono algebrico e per alcuni aspetti vicine a quelle può tradizionali riguardanti i campi numerici; per analogia spesso l'operazione di giustapposizione verrà chiamata anche **prodotto** e conseguentemente invece del segno "$\shortmid$" verrà usato il segno "$\cdot$".

Conviene rilevare esplicitamente che la assunzione della associatività per la giustapposizione discende dalla opportunità di utilizzare sul piano formale le stringhe per rappresentare i messaggi che si scambiano sistematicamente gli agenti matematico-informatici, messaggi la cui associatività risulta chiaramente giustificata.

Questo consente di descrivere, con efficacia, le stringhe come oggetti ottenuti mediante accostamento materiale di entità tangibili semplici (per esempio come affiancamento di caratteri tipografici) che assumono i ruoli delle sue successive componenti.

C10a.07 Ogni stringa, potendosi considerare la giustapposizione dei suoi caratteri, può essere presentat con una scrittura come

$$w \;=\; \mathsf{a}_{j_1} \shortmid \mathsf{a}_{j_2} \shortmid \mathsf{a}_{j_3} \shortmid ... \shortmid \mathsf{a}_{j_{r-1}} \shortmid \mathsf{a}_{j_r} \;.$$

Spesso il segno della giustapposizione, come altri segni attribuibili a qualche genere di "operazione prodotto", per brevità si riduce un piccolo spazio tra i due successivi operandi (caratteri componenti) o anche si fa scomparire.
Vengono quindi adottate scritture come

$$w \;=\; \mathsf{a}_{j_1}\mathsf{a}_{j_2}\mathsf{a}_{j_3}...\mathsf{a}_{j_{r-1}}\mathsf{a}_{j_r} \;=\; \mathsf{a}_{j_1}\;\mathsf{a}_{j_2}\;\mathsf{a}_{j_3}\;...\;\mathsf{a}_{j_{r-1}}\;\mathsf{a}_{j_r} \;.$$

Si può quindi descrivere la giustapposizione di due stringhe w ed e x come il processo di "scrivere" la w seguita immediatamente dalla x sopra un dispositivo in grado di registrare in successione i caratteri della w e dopo di questi i caratteri della x.

Tale dispositivo di registrazione sequenziale dei caratteri viene convenzionalmente chiamato **nastro**. Esso si può descrivere come un supporto che presenta una sequenza di sezioni elementari che chiamiamo **caselle**, ciascuna delle quali in grado di registrare un carattere dell'alfabeto con cui si lavora.

I nastri vengono presentati disposti orizzontalmente e si conviene di denotare come loro prima casella la più a sinistra, in sintonia con le modalità di scrittura e lettura che procedono da sinistra a destra (modalità prevalenti nei paesi "occidentali").

Come si è visto in B17 e come vedremo in C21, risulta utile servirsi di nastri che, oltre a poter essere registrati da sinistra a destra, possono essere percorsi in avanti e all'indietro e avere le caselle modificate.

Inoltre in molti sviluppi argomentativi un nastro, cioè un oggetto tangibile e quindi finito, quando le circostanze lo richiedono, si consente possa essere dotato di ulteriori caselle da aggiungere sulla destra ed anche sulla sinistra.

A questo punto è opportuno segnalare che finora abbiamo denotato i caratteri dell'alfabeto di lavoro servendoci della fonte teletype e quindi di notazioni come $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$, $\mathtt{a}_j$ o $\mathtt{b}_{h_i}$.
Tuttavia per il seguito ci riserviamo invece di usare notazioni come a, b, c, a_j o b_{h_i} in quanto più agevolmente inseribili in espressioni matematiche elaborate.

C10a.08 Per l'insieme delle stringhe su un alfabeto A si utilizza la notazione

$$\mathsf{A}^+ \;:=\; \dot{\bigcup}_{r=1}^{+\infty} \mathsf{A}^r \;.$$

A^+ può leggersi "A cross" e munito della giustapposizione dal punto di vista dell'algebra costituisce un semigruppo chiamato **semigruppo libero sull'alfabeto** A [B41a].

Si osserva che la giustapposizione è commutativa sse l'alfabeto ha un solo carattere, mentre per un alfabeto di due o più caratteri non lo è; ad esempio $\mathtt{ab}\,\mathtt{a} \neq \mathtt{a}\,\mathtt{ab}$.

Conviene osservare esplicitamente che la noncommutatività della giustapposizione è una caratteristica che rende le stringhe strumenti in grado di rappresentare gli eventi nel mondo reale, in quanto entità osservabili nel loro svolgersi nel tempo e strettamente vincolate dalle relazioni "prima di" e "dopo di".

I semigruppi liberi della forma $\langle \mathsf{A}^+, \, \rangle$ non hanno unità; tuttavia a ciascuno di essi, come a ogni semigruppo [B41a], si può aggiungere una nuova entità che denotiamo con $\sqcup$, che considereremo stringa su ciascuno degli alfabeto utilizzabili e che chiamiamo **stringa muta** o stringa vuota, chiedendo che giustapposta a sinistra o a destra a qualsiasi stringa la lasci invariata:

$$w \, \sqcup \;=\; \sqcup \, w \;=\; w \;.$$

Essa quindi ha il ruolo di unità bilatera per la giustapposizione.

Alla stringa muta si attribuisce la lunghezza 0, $\sqcup^{\vdash} := 0$; inoltre per ogni alfabeto A si scrive $\mathsf{A}^0 := \{ \sqcup \}$.

Si introduce allora la notazione

$$\mathsf{A}^* \;:=\; \dot{\bigcup}_{r\in\mathbb{N}} \mathsf{A}^r \;,$$

da leggersi "A star" e si osserva che $\langle \mathsf{A}^*, \, , \sqcup \rangle$ costituisce un monoide del quale $\sqcup$ è l'unità; esso è detto **monoide libero su un alfabeto** A.

C10a.09 Una stringa $w \in \mathsf{A}^r$ si può considerare una funzione del genere $\lceil (\mathrm{r}\,] \longmapsto \mathsf{A} \rceil$ oppure del genere $\lceil [r) \longmapsto \mathsf{A} \rceil$ a seconda che si attribuiscano alle sue componenti, risp., indici come $1,...,r$ o gli indici $0,...,r-1$.

Secondo le due convenzioni, si evidenziano i successivi caratteri componenti con le scritture, risp., $w = w(1)...w(r)$ oppure $w = w(0)...w(r-1)$; nel seguito seguiremo prevalentemente la prima convenzione.

Quando risulterà opportuno scriveremo $w =_{\mathsf{A}} \mathtt{a}_{j_1}\mathtt{a}_{j_2}\dots\mathtt{a}_{j_r}$ per segnalare sia l'uguaglianza delle stringhe a sinistra e a destra del segno "=", sia il fatto che ciascuno dei simboli $\mathtt{a}_{j_1}$, $\mathtt{a}_{j_2}$,...,$\mathtt{a}_{j_r}$ rappresenta un puro carattere dell'alfabeto A con cui si lavora.

Per lunghezza di una stringa $w \in \mathsf{A}^*$ si è definito l'intero naturale che esprime il numero delle sue componenti, cioè l'entità fornita dalla stringa ottenuta trasformando ogni carattere della w in in unico segno (per esempio $\mid$). Si constata che

$$\forall w, x \in \mathsf{A}^* \ : \ (w \mid x)^{\vdash} = w^{\vdash} + x^{\vdash} \ .$$

In termini algebrici questa relazione dice che la funzione lunghezza costituisce un morfismo di ogni monoide libero $\langle \mathsf{A}^*, \mid, \sqcup \rangle$ sul monoide $\langle \mathbb{N}, +, 0 \rangle$.

Evidentemente si tratta di un epimorfismo che è biiettivo sse $|\mathsf{A}| = 1$.

Le classi di equivalenza di questo morfismo sono per i diversi $r \in \mathbb{N}$ gli insiemi A^r delle stringhe aventi la stessa lunghezza r.

Per ogni $w \in \mathsf{A}^*$ e per ogni $a \in \mathsf{A}$ denotiamo con $|w|_a$ il numero di caratteri a che si incontrano scorrendo w. Evidentemente $|w| = \sum_{a \in \mathsf{A}} |w_a|$.

Più in generale per ogni $\mathsf{B} \subseteq \mathsf{A}$ definiamo $|w|_{\mathsf{B}} := \sum_{a \in \mathsf{B}} |w|_a$.

Definiamo **alfabeto minimo della parola** w, e denotiamo con w^{minalf}, il più ridotto sottoinsieme di A tale che $w \in \left(w^{\text{minalf}}\right)^*$.

Evidentemente $a \in w^{\text{minalf}} \iff |w|_a > 0$ e $\sqcup^{\text{minalf}} = \emptyset$.

C10a.10 Si dice **linguaggio [formale] sull'alfabeto** A ogni sottoinsieme di A^*.

Denotiamo con $\mathbf{Lng}_{\mathsf{A}}$ la loro collezione, cioè definiamo

$$\mathbf{Lng}_{\mathsf{A}} := \mathfrak{P}(\mathsf{A}^*) \ .$$

Nello studio dei linguaggi i casi relativi ad alfabeti di una sola lettera sono molto particolari. In effetti in questo caso si hanno monoidi liberi come

$$\{\mid\}^* = \{\sqcup, \mid, \mid\mid, \mid\mid\mid, \mid\mid\mid\mid, \mid\mid\mid\mid\mid, \mid\mid\mid\mid\mid\mid, \mid\mid\mid\mid\mid\mid\mid, ...\}.$$

Un tale insieme è evidentemente isomorfo ad $\mathbb{N}$ e fornisce una cosiddetta **rappresentazione unadica degli interi naturali** si tratta della rappresentazione che, sostanzialmente, è stata adottata da tutti i popoli primitivi.
Va tuttavia segnalata la eccezione dello zero, entità non associabile ad alcun oggetto tangibile e quindi largamente adottato dopo un rilevante travaglio.

Secondo questa rappresentazione un intero $n > 0$ è rappresentato da n repliche affiancate di un determinato segno (che possono rappresentare n tacche, n sassolini (calculi), n conchiglie, n nodi su una corda, ...).

La somma di due interi naturali è ottenuta, ben comprensibilmente, accostando le sequenze di segni esprimenti i due addendi e corrisponde nei modelli concreti all'accostamento di due sequenze di tacche,

all'aggregazione di due raggruppamenti di sassolini o di conchiglie, al congiungimnto di due corde che portno nodi, ...

Per alfabeti A con due o più elementi, A^* ha una struttura molto più complessa e ricca di applicazioni di $\mathbb{N}$.
È quindi ragionevole pensare che i numeri naturali servano per esprimere efficacemente caratteristiche essenziali dei raggruppamenti di oggetti più o meno tangibili, ma non siano in grado di rappresentare le caratteristiche dei raggruppamenti di oggetti tra i quali sussistono rilevanti correlazioni, cioè raggruppamenti dotati di qualche complessità.

C10a.11 Stringhe e linguaggi, nonostante la semplicità della loro definizione, costituiscono strumenti utili in moltissimi settori e vengono studiati da svariati punti di vista.

In particolare i linguaggi formali sono in grado di fornire tutte le espressioni per le entità che possono essere oggetto di studi razionali e attendibili.

Alcune proprietà dei linguaggi si ricavano con metodi algebrici che, come vedremo tra breve, fanno riferimento a semigruppi [B41a], semianelli [B41c] e algebre di Kleene [C32].

Questi metodi però hanno limiti piuttosto marcati: in effetti le strutture algebriche citate si basano su assiomi piuttosto deboli che portano ad algoritmi raramente stringenti quanto quelli sui numeri interi e sulle funzioni numeriche.

In effetti per disporre di linguaggi efficacemente utilizzabili si rendono necessari altri approcci di tipo costruttivo: nei capitoli C12 e C14 vedremo due classi di strumenti che consentono di individuare e manipolare linguaggi relativamente semplici, i riconoscitori e le grammatiche.

Più avanti vedremo alcuni strumenti più prestanti come le macchine di Turing, e i sistemi i riscrittura; di questi strumenti sono stati proposti numerosi arricchimenti in grado di affrontare classi mediamente estese di problemi.

Molte proprietà dei linguaggi richiedono studi di proprietà combinatorie nei quali si devono analizzare congiuntamente biiezioni con configurazioni discrete di vario genere (in particolare geometrico), meccanismi enumerativi, proprietà algebriche e costruzioni algoritmiche.

Un genere di strumenti che fornisce metodi costruttivi che possono essere computazionalmente stringenti è costituito dalle serie di potenze di variabili noncommutative.

C10a.12 Per quanto riguarda le applicazioni, occorre innanzitutto segnalare che mediante stringhe si possono esprimere, ovvero **codificare**, tutti gli oggetti studiati dalla matematica e dalle discipline che di essa si avvalgono e tutte le informazioni che si intendono sottoporre a elaborazioni automatiche.

Questo spinge a fare sistematico riferimento ai procedimenti che operano su stringhe, a cominciare dai procedimenti che servono allo studio dei fondamenti della matematica e dalla logica, fino ai procedimenti computazionali, alle tecniche per la manipolazione simbolica e all'analisi dei testi.

Molti oggetti della matematica discreta si studiano e si elaborano attraverso loro codifiche e queste rivestono un ruolo importante in vari punti della molto articolata rete di collegamenti che sussiste tra le numerose configurazioni discrete che vengono studiate anche per applicazioni di notevole rilevanza pratica.

I linguaggi formali trovano applicazioni importanti e dirette con i linguaggi di programmazione [*Aho, Sethi, Ullman 1988*], con i linguaggi per la comunicazione [*Goldfarb 1990*] e con la crittografia [*Berstel, Perrin 1985*].

Con essi inoltre si possono tenere sotto controllo i linguaggi naturali e vari fenomeni e processi della organizzazione delle conoscenze e della gestione della comunicazione.

Segnaliamo in particolare l'attuale importanza pratica degli algoritmi per l'analisi dei testi [*Crochemore, Rytter 1994*] e per la compressione dei dati [*Bell, Cleary, Witten 1990*].

C10a.13 I linguaggi formali forniscono vari spunti per lo sviluppo di strumenti enumerativi.

Il più semplice proviene dalla cosiddetta **successione di enumerazione-L di un linguaggio** L, successione dei numeri delle sue stringhe raggruppate secondo le successive lunghezze:

$$\left\langle\ r \in \mathbb{N}\ \vdots\ |\mathsf{L} \cap \mathsf{A}^r|\ \right\rangle\ .$$

Altri temi enumerativi riguardano i diversi modi di individuare le stringhe di un linguaggio quando si opera con strumenti come le grammatiche; in particolare questo conduce alla precisazione di alcuni generi di espressioni con significato ambiguo [C14].

C10 b. operazioni su stringhe e linguaggi

C10b.01 Nel seguito del capitolo useremo i caratteri A, B, ... per denotare alfabeti, L, M, N, X, Y, ... per denotare linguaggi, a, b, c, ... per i caratteri, u, v, w, x, y, z per denotare stringhe; le lettere m, n, r, s individueranno invece interi naturali.

Si dice **giustapposizione dei linguaggi** L ed M l'insieme delle stringhe ottenibili giustapponendo una stringa di L con una di M:

$$\mathsf{L}\cdot\mathsf{M} \;:=\; \{w \in \mathsf{L},\; x \in \mathsf{M} \;:|\; w\cdot x\}\;.$$

Dunque la giustapposizione di linguaggi è l'estensione booleana della giustapposizione di stringhe.

Di un linguaggio si possono quindi considerare tutte le potenze positive:

$$\mathsf{L}^1 \;:=\; \mathsf{L} \qquad , \qquad \mathsf{L}^2 := \mathsf{L}\cdot\mathsf{L} \qquad , \qquad \ldots \mathsf{L}^r := \mathsf{L}^{r-1}\cdot\mathsf{L} \qquad .$$

Inoltre risulta utile definire la sua potenza nulla

$$\mathsf{L}^0 \;:=\; \{\sqcup\}\;.$$

C10b.02 Si dice **chiusura per giustapposizione del linguaggio** o **chiusura-cross del linguaggio** L l'unione delle sue potenze positive; la denoteremo con

$$\mathsf{L}^+ \;:=\; \bigcup_{r=1}^{\infty} \mathsf{L}^r\;.$$

Chiaramente $\mathsf{L}^+ \;=\; \mathsf{L}\big[\,\cdot\,\big]$.

Si dice invece **chiusura-star del linguaggio** L l'unione delle sue potenze nonnegative; la denoteremo con

$$\mathsf{L}^* \;:=\; \bigcup_{r=0}^{\infty} \mathsf{L}^r\;.$$

Si provano facilmente relazioni come le seguenti:

$$\forall r,s \in \mathbb{N} \;:\; \mathsf{L}^r\cdot\mathsf{L}^s \;=\; \mathsf{L}^s\cdot\mathsf{L}^r \;=\; \mathsf{L}^{r+s} \qquad , \qquad \mathsf{L}^* \;=\; \{\sqcup\}\dot\cup\mathsf{L}^+ \qquad ,$$

$$\mathsf{L}^+ \;=\; \mathsf{L}\cdot\mathsf{L}^* \;=\; \mathsf{L}^*\cdot\mathsf{L}\;; \qquad , \qquad \mathsf{L}^* \;=\; \mathsf{L}^+ \iff \sqcup \in \mathsf{L}^+ \iff \sqcup \in \mathsf{L}\;.$$

C10b.03 In molti contesti l'operazione binaria di unione di linguaggi viene denotata con il segno "+" e conseguentemente l'unione in generale viene espressa con la notazione della sommatoria: questo atteggiamento non del tutto chiaro rende più familiari alcune proprietà algebriche dei monoidi liberi di stringhe, in quanto le avvicina a proprietà degli usuali campi numerici.
Questa convenzione si accorda con quella di esprimere la giustapposizione con il segno "·" e di chiamarla "prodotto".

Si ottengono quindi le seguenti uguaglianze:

$$\mathsf{L}+(\mathsf{M}+\mathsf{N}) \;=\; (\mathsf{L}+\mathsf{M})+\mathsf{N} \qquad , \qquad \mathsf{L}\cdot(\mathsf{M}\cdot\mathsf{N}) \;=\; (\mathsf{L}\cdot\mathsf{M})\cdot\mathsf{N} \;\text{ (associatività)} \qquad ,$$

$$(\mathsf{L}+\mathsf{M})\cdot\mathsf{N} \;=\; \mathsf{L}\cdot\mathsf{N}+\mathsf{M}\cdot\mathsf{N} \qquad , \qquad \mathsf{L}\cdot(\mathsf{M}+\mathsf{N}) \;=\; \mathsf{L}\cdot\mathsf{M}+\mathsf{L}\cdot\mathsf{N} \;\text{ (distributività)} \qquad .$$

Conviene segnalare esplicitamente che, contrariamente a quanto accade per i campi, si ha l'idempotenza della somma:

$$\mathsf{L}+\mathsf{L} \;=\; \mathsf{L}\;;$$

quindi non si può disporre di una operazione inversa della somma.

Va anche segnalato che, in accordo con le suddette convenzioni "algebrizzanti", talora viene usato il segno "−" per denotare la differenza insiemistica di linguaggi, cioè che si considera $\mathsf{L}-\mathsf{M}$ equivalente a $\mathsf{L}\setminus\mathsf{M}$ e che talore invece di $\sqcup$ si usa $\mathbf{1}$.

Mentre ovviamente sopra un alfabeto di un carattere la giustapposizione di linguaggi è commutativa, sopra un alfabeto di due o più caratteri per gran parte delle coppie di linguaggi $\langle\mathsf{L},\mathsf{M}\rangle$ si riscontra la noncommutatività, cioèaccade che $\mathsf{L}\cdot\mathsf{M}\neq\mathsf{M}\cdot\mathsf{L}$.

Per i linguaggi $\emptyset$ e $\{\sqcup\}$ si trovano le seguenti relazioni:

$$\mathsf{L}+\emptyset \;=\; \emptyset+\mathsf{L} \;=\; \mathsf{L} \quad , \quad \mathsf{L}\cdot\emptyset \;=\; \emptyset\cdot\mathsf{L} \;=\; \emptyset \quad , \quad \mathsf{L}\cdot\sqcup \;=\; \sqcup\cdot\mathsf{L} \;=\; \mathsf{L} \quad ,$$
$$\emptyset^{+} \;=\; \emptyset \quad , \quad \emptyset^{*} \;=\; \sqcup \quad , \quad \sqcup^{+} \;=\; \sqcup^{*} \;=\; \sqcup \quad .$$

L'insieme dei linguaggi sopra un alfabeto A costituisce quindi un semianello avente $\emptyset$ come zero e $\{\sqcup\}$ come unità; questo semianello è noncommutativo sse A ha più di un carattere [T15h02].

Nell'ultima delle precedenti relazioni abbiamo adottato un'altra semplificazione piuttosto usuale evitando le parentesi graffe intorno a $\sqcup$, cioè identificando una stringa con il linguaggio costituito da questo solo elemento.

C10b.04 Utilizzando i segni "+" e "·" le espressioni per i linguaggi finiti assumono aspetto di espressioni polinomiali: ad esempio invece di $\{a^3,b^3\}$ si scrive a^3+b^3, cioè una espressione utilizzata anche per oggetti molto familiari come i polinomi .
Questo agevola la scrittura di formule come la seguente:

$$(a+b^2+b^2\,a)\cdot(\sqcup+a) \;=\; a+b^2+b^2\,a+a^2+b^2\,a^2 \ .$$

In queste espressioni l'idempotenza della somma non consente coefficienti superiori a 1 e ovviamente non hanno senso i coefficienti negativi; i soli coefficienti delle espressioni monomiali possono essere lo zero e l'unità, ovvero $\emptyset$ e $\sqcup$; questi due elementi costituiscono il terreno del semianello isomorfo a quello dei bits $\mathbb{B}$.

In queste espressioni i caratteri svolgono un ruolo che li fa considerare delle 'variabili noncommutative" sopra un semianello.
Dunque le stringhe e linguaggi sono retti da formule contenenti espressioni polinomiali in variabili noncommutative nelle quali non compare alcun coefficiente.

Coerentemente un linguaggio infinito $\mathsf{L}\subseteq\mathsf{A}^*$ si può considerare come serie nelle variabili formali noncommutative costituenti A sul semianello avente come terreno $\{\emptyset,\sqcup\}$.
Questo conduce a scritture come la seguente:

$$\mathsf{L} \;=\; \sum_{w\in\mathsf{L}} w \;=\; \sum_{\mathsf{a}_{j_1}\ldots\mathsf{a}_{j_r}\in\mathsf{A}^*} \mathsf{a}_{j_1}\ldots\mathsf{a}_{j_r}\,\big\langle \mathsf{a}_{j_1}\ldots\mathsf{a}_{j_r}\big|\mathsf{L}\big\rangle \quad ,$$

nella quale si sono introdotti i coefficienti (alla Dirac) con valori binari

$$\big\langle \mathsf{a}_{j_1}\ldots\mathsf{a}_{j_r}\big|\mathsf{L}\big\rangle \;:=\; \begin{cases} \sqcup & \text{sse } \mathsf{a}_{j_1}\ldots\mathsf{a}_{j_r}\in\mathsf{L}, \\ \emptyset & \text{altrimenti.} \end{cases}$$

In particolare si hanno la serie delle potenze della stringa w e la serie delle potenze del linguaggio L:

$$w^* \;=\; \sqcup+w+w^2+\ldots \;=\; \sum_{r=0}^{\infty} w^r \qquad , \qquad \mathsf{L}^* \;=\; \sqcup+\mathsf{L}+\mathsf{L}^2+\ldots \;=\; \sum_{r=0}^{\infty}\mathsf{L}^r \qquad .$$

Quindi per affermare che una stringa z è una qualche potenza della w si scrive $z\in w^*$, mentre per enunciare che tutte le stringhe del linguaggio M si possono esprimere come prodotti di stringhe appartenenti ad L si scrive $\mathsf{M}\subseteq\mathsf{L}^*$.

C10b.05 Per le operazioni definite sui linguaggi si possono dimostrare senza difficoltà varie altre relazioni.

Prop. $\forall\, \mathsf{L}, \mathsf{M} \subseteq \mathsf{A}^*$:

$$(\mathsf{L}+\mathsf{M})^* \;=\; (\mathsf{L}^* \cdot \mathsf{M})^* \cdot \mathsf{L}^* \;=\; (\mathsf{M}^* \cdot \mathsf{L})^* \cdot \mathsf{M}^* \quad , \quad (\mathsf{L} \cdot \mathsf{M})^* \;=\; \mu + \mathsf{L} \cdot (\mathsf{M} \cdot \mathsf{L})^* \cdot \mathsf{M} \quad ,$$

$$\mathsf{L}^* \cdot \mathsf{L}^* \;=\; \mathsf{L}^* \quad , \quad \mathsf{L}^+ \cdot \mathsf{L}^+ \;=\; \mathsf{L} \cdot \mathsf{L}^+ \;=\; \mathsf{L}^2 \cdot \mathsf{L}^* \quad , \quad \mathsf{L}^* \cdot \mathsf{L}^+ \;=\; \mathsf{L}^+ \cdot \mathsf{L}^* \;=\; \mathsf{L}^+ \quad ,$$

$$(\mathsf{L}^*)^* \;=\; \mathsf{L}^* \quad , \quad (\mathsf{L}^+)^+ \;=\; \mathsf{L}^+ \quad , \quad (\mathsf{L}^+)^* \;=\; (\mathsf{L}^*)^+ \;=\; \mathsf{L}^* \,; \qquad (\mu + \mathsf{L})^+ \;=\; (\mu + \mathsf{L})^* \;=\; \mathsf{L}^* \quad ,$$

$$\forall r \in \mathbb{N} \;:\; (\mathsf{L}^+)^r \;=\; \mathsf{L}^r \cdot \mathsf{L}^* \qquad , \qquad \forall r \in \mathbb{N} \;:\; \mathsf{L}^* \;=\; (\mathsf{L}^r)^* \cdot \mathsf{L}^{[r)} \qquad \blacksquare$$

C10b.06 Le uguaglianze trovate che coinvolgono gli operatori unari $^+$ e * inducono a riferire le costruzioni insiemistiche e basate sulla giustapposizione dei linguaggi a una struttura algebrica più ricca della struttura di semianello.

Seguendo *John Horton Conway (1971)* chiamiamo **algebra di Kleene classica** una struttura della forma $\langle \boldsymbol{K}, \oplus, \odot, {}^{\circledast}, \mathbf{0}, \mathbf{1} \rangle$ nella quale

$\langle \boldsymbol{K}, \oplus, \odot, \mathbf{0} \rangle$ è un semianello ;
l'operazione binaria "$\oplus$" è idempotente: $(\forall L \in \boldsymbol{K} \;:\; L \oplus L \;=\; L)$;
"$\mathbf{1}$" è l'unità per l'operazione "$\odot$" ,
"$^{\circledast}$" è un'operazione unaria,

per la quale valgono le seguenti proprietà per arbitrari $L, M \in \boldsymbol{K}$:

$$(L \oplus M)^{\circledast} \;=\; (L^{\circledast} \odot M)^{\circledast} \odot L^{\circledast} \quad , \quad (L \odot M)^{\circledast} \;=\; \mathbf{1} \oplus L \odot (M \odot L)^{\circledast} \odot M \quad ,$$

$$L^{\circledast} \odot L^{\circledast} \;=\; L^{\circledast}$$

e dove, posto $L^{\oplus} := L \odot L^{\circledast}$, vale la successione di uguaglianze

$$\forall r \in \mathbb{N} \;:\; L^{\circledast} \;=\; (L^r)^{\circledast} \odot (\mathbf{1} \oplus L \oplus \cdots \oplus L^{r-1}) \;\; .$$

Oltre la ridottissima struttura con il terreno ridotto a un solo elemento che fa da zero e da unità, la più semplice algebra di Kleene classica che chiamiamo **algebra di Kleene booleana** è

$$\langle\, \{\emptyset, \mu\}, +, \cdot, {}^*, \emptyset, \mu \,\rangle$$

Molto più articolata è l'**algebra di Kleene dei linguaggi** sull'alfabeto A:

$$\langle\, \mathbf{Lng}_{\mathsf{A}}, +, \cdot, {}^*, \emptyset, \mu \,\rangle \;\; .$$

La struttura algebrica precedente e altre strutture a lei vicine verranno trattate con una certa ampiezza in C32.

C10b.07 Una operazione unaria su stringhe e linguaggi che conduce a una utile simmetria negli insiemi di stringhe e linguaggi è la **riflessione**.

Si dice **stringa riflessa** della stringa $w =_{\mathsf{A}} \mathtt{a}_{j_1}\mathtt{a}_{j_2}...\mathtt{a}_{j_r}$ e si denota con $w^{\leftarrow}$ o w^{T} la stringa ottenuta "leggendo i caratteri della w da destra a sinistra", cioè la stringa $w^{\leftarrow} := \mathtt{a}_{j_r}...\mathtt{a}_{j_2}\mathtt{a}_{j_1}$.
Ad esempio: $(\mathtt{ROMA})^{\leftarrow} \;=\; \mathtt{AMOR}$.

Per la giustapposizione di due stringhe si constata che $(w \cdot x)^{\leftarrow} \;=\; x^{\leftarrow} \cdot w^{\leftarrow}$; questo si esprime in termini algebrici dicendo che l'operazione unaria riflessione è un **antiisomorfismo dei monoidi liberi**.

Si dice **riflesso del linguaggio** L, e si scrive $\mathsf{L}^{\leftarrow}$, l'insieme delle stringhe riflesse delle stringhe di L:

$$\forall \mathsf{L} \in \mathbf{Lng} \;:\; \mathsf{L}^{\leftarrow} := \{w \in \mathsf{L} :|\; w^{\leftarrow}\} \;\; .$$

Evidentemente la riflessione dei linguaggi è l'estensione booleana della riflessione tra stringhe; inoltre essa è un'involuzione per la collezione dei linguaggi sopra un qualsiasi alfabeto:

$$\forall \mathsf{L} \in \mathbf{Lng} \ : \ (\mathsf{L}^{\leftarrow})^{\leftarrow} \ = \ \mathsf{L} \ .$$

Tra le stringhe si distinguono quelle invarianti per riflessione, stringhe dette **stringhe palindrome** : una palindroma di lunghezza pari è `ANNA` ed una di lunghezza dispari è `ANILINA`. un'altra palindroma famosa è `SATOR AREPO TENET OPERA ROTAS` .
Conviene considerare che anche la stringa muta sia una palindroma.

Un linguaggio P si dice **linguaggio palindromo** sse $w \in \mathsf{P} \Longrightarrow w^{\leftarrow} \in \mathsf{P}$.
Denotiamo con $\mathbf{LngPlndrm}_{\mathsf{A}}$ la collezione dei linguaggi palindromi su A e con $\boldsymbol{Palnd}_{\mathsf{A}}$ l'insieme delle stringhe palindrome su A.

(1) Prop.: La collezione dei linguaggi costituiti da sole stringhe palindrome su A è contenuta propriamente nella collezione dei linguaggi palindromi su A, ossia $\mathfrak{P}(\boldsymbol{Palnd}_{\mathsf{A}}) \subset \mathbf{LngPlndrm}_{\mathsf{A}}$ ∎

Dim.: Chiaramente $\forall \mathsf{P} \subseteq \boldsymbol{Palnd}_{\mathsf{A}} \ : \ \mathsf{P} \in \mathbf{LngPlndrm}_{\mathsf{A}}$; mentre per l'inclusione stretta basta osservare che $\mathtt{aab} + \mathtt{baa} \in \mathbf{LngPlndrm}_{\{a,b\}}$ mentre e $(\mathtt{aab} + \mathtt{baa}) \cap \boldsymbol{Palnd}_{\{a,b\}} \ = \ \emptyset$.

C10b.08 **(1) Eserc.** Verificare che l'insieme delle palindrome su A si può esprimere come:

$$\boldsymbol{Palnd}_{\mathsf{A}} \ := \ \sum_{w \in \mathsf{A}^*} w \, w^{\leftarrow} + \sum_{w \in \mathsf{A}^*} w \, \mathsf{A} \, w^{\leftarrow}$$

e che, se $n := |\mathsf{A}|$, la successione di enumerazione-L di tale linguaggio è:

$$\left| \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & \dots & r & \dots \\ 1 & n & n & n^2 & n^2 & n^3 & n^3 & \dots & n^{\lceil r/2 \rceil} & \dots \end{matrix} \right| \ .$$

(2) Eserc. Constatare che $\forall y \in \boldsymbol{Palnd}_{\mathsf{A}} \ : \ y^2, y^3, ..., y^+, y^* \in \mathbf{LngPlndrm}_{\mathsf{A}}$.

(3) Eserc. Constatare che $\mathbf{LngPlndrm}_{\mathsf{A}} \ \in \ [\, +, {}^{\leftarrow} \,]$

(4) Eserc. Mostrare controesempi che portano alla relazione $\mathbf{LngPlndrm}_{\mathsf{A}} \notin [\, \cdot \,]$ e constatare che si può solo affermare $\lceil \ Y, Z \in \mathbf{LngPlndrm}_{\mathsf{A}} \implies Y \cdot Z + Z \cdot Y \in \mathbf{LngPlndrm}_{\mathsf{A}} \ \rceil$.

(5) Eserc. Constatare che $Y \subseteq \boldsymbol{Palnd}_{\mathsf{A}} \not\Rightarrow \ Y^+, Y^* \in \mathbf{LngPlndrm}_{\mathsf{A}}$ e quindi
$Z \in \mathbf{LngPlndrm}_{\mathsf{A}} \not\Rightarrow \ Z^+, Z^* \in \mathbf{LngPlndrm}_{\mathsf{A}}$.

(6) Eserc. Constatare che $\forall \mathsf{L} \in \mathbf{Lng}_{\mathsf{A}} \ : \ \mathsf{L} + \mathsf{L}^{\leftarrow} \in \mathbf{LngPlndrm}_{\mathsf{A}} \ \wedge \ (\mathsf{L} + \mathsf{L}^{\leftarrow})^+ \in \mathbf{LngPlndrm}_{\mathsf{A}}$

C10b.09 La riflessione e la giustapposizione sono state definite sui linguaggi attraverso la semplice estensione booleana delle corrispondenti operazioni definite sulle stringhe. Analoghe **estensioni booleane** si possono effettuare per tutte le operazioni unarie, binarie e infinitarie definite sugli elementi dei linguaggi e di una qualsiasi struttura algebrica per renderle applicabili ai sottoinsiemi (del terreno) della struttura stessa.

Se si adotta la stessa notazione per una operazione e per la sua estensione booleana in genere non si generano confusioni difficili da chiarire osservando il contesto.
Questo si riscontra in particolare se si semplifica il simbolo $\odot^{be}$ con $\odot$ e il simbolo $+^{be}$ con $+$.

Esaminando le estensioni booleane si giustificano facilmente le uguaglianze che seguono, uguaglianze che adottano la semplificazione suddetta e che aiutano a chiarire le relazioni tra la riflessione e le altre operazioni sui linguaggi:

$$(\mathsf{L} + \mathsf{M})^{\leftarrow} \ = \ \mathsf{L}^{\leftarrow} + \mathsf{M}^{\leftarrow} \quad , \quad (\mathsf{L} \cdot \mathsf{M})^{\leftarrow} \ = \ \mathsf{M}^{\leftarrow} \cdot \mathsf{L}^{\leftarrow} \quad ,$$
$$(\mathsf{L}^{+})^{\leftarrow} \ = \ (\mathsf{L}^{\leftarrow})^{+} \quad , \quad (\mathsf{L}^{*})^{\leftarrow} \ = \ (\mathsf{L}^{\leftarrow})^{*} \quad .$$

Ricordiamo inoltre che la riflessione tra stringhe e linguaggi è alla base della dualità-LR.

C10b.10 Per talune attività, dato un linguaggio L serve individuare il suo cosiddetto sottolinguaggio palindromo

$$\mathsf{L} \cap \boldsymbol{Palnd} \,. \tag{1}$$

In altre circostanze si chiede il suo sottolinguaggio antipalindromo $\mathsf{M} \subseteq \mathsf{L}$ tale che $\mathsf{M} \cap \mathsf{M}^{\leftarrow} = \emptyset$, cioè l'insieme ottenuto eliminando da L le sue stringhe palindrome. Questo linguaggio è fornito dalle espressioni equivalenti

$$\mathsf{L} \setminus \boldsymbol{Palnd} \;=\; \mathsf{L} \cap \boldsymbol{Palnd} \;=\; \mathsf{L} \setminus (\mathsf{L} \cap \mathsf{L}^{\leftarrow}) \,. \tag{2}$$

Occorre osservare che le espressioni proposte (1) e (2) quando si devono trovare caratteristiche utilizzabili dei due sottolinguaggi, servono solo ad impostare i due problemi effettivi, ma la loro soluzione deve essere trovata tenendo conto delle specifiche caratteristiche del linguaggio da ridurre e delle esigenze che si vogliono soddisfare con il sottolinguaggio risultato.

C10b.11 Nella lingua italiana, ma non solo, si trovano molte parole ottenute dalla giustapposizione di due sillabe con il frequente raddoppio della iniziale della seconda:

babà, bebè, cacca, cocco, lalla, mamma, nanna, nonno, papa, pappa, popo, pupu, tata, tette, ...

Molte di queste parole sono tra le prime che i piccoli sono in grado di controllare, grazie a una loro sostanziale semplicità costruttiva; la presenza di una doppia ha l'effetto di una pausa che facilita ulteriormente in controllo della parola.

Un altro schema che raccoglie parole di facile controllo consiste in un tetragramma palindromo preceduto da una consonante: cassa, nassa, sonno, zappa, ...

Segnaliamo anche le giustapposizioni di due palindrome: esse godono di due proprietà di simmetria abbastanza semplici, anche se meno nette della proprietà che caratterizza le palindrome.
Se u e v sono due palindrome, allora $(u \cdot v)^{\leftarrow} = v \cdot u$.

C10b.12 Buona parte dei giochi enigmistici consistono nel presentare un testo, tipicamente un testo poetico, nel quale sono individuabili due o più parole incognite che si chiede di individuare. Queste parole vengono rappresentate da schemi convenzionali (xxxxxx, xxxyyyxxx, xxyxxxzxx, ...), oppure che sono alluse da immagini espresse in versi collegati da rime e somiglianze ritmiche, spesso facendo intervenire termini ambigui per omonimia, sinonimia e polisemia.
Le parole incognite risultano raggiungibili attraverso collegamenti esprimibili in termini di operazioni su stringhe come giustapposizione, riflessione o permutazione.
Il solutore deve quindi ricostruire percorsi basati su collegamenti criptici che possono essere sia formali che dovuti a significati non immediatamente riconoscibili.

Si parla di "anagramma" se si devono indovinare una parola o un frase e una sua permutata (*trilussa* e *salustri*), (*arrosti* con *artrosi*).

Più in particolare si parla di "scambio" quando le due parole differiscono solo per il cambiamento di posizione di due occorrenze di lettere (*creole* con *ercole*).

Si chiama "scambio sillabico" uno scambiano di due sillabe.

Si chiama "bifronte" il gioco nel quale si devono indovinare una parola e la sua riflessa (*acetone* insieme a *enoteca*).

Si parla di "cambio" o di "metatesi" quando si modifica una sola lettera (*malta*, *malva*).

Si ha una "zeppa" se da una prima parola si ottiene una seconda mediante inserimento di una lettera (da *casa* a *casta*) o di più lettere (da *lino* a *liocorno*).

Viceversa viene detto "scarto" il gioco che vede una prima parola e una seconda ottenuta eliminando dalla prima una lettera (da *geloso* a *gelso*).

Loro prevedibili varianti sono la "zeppa sillabica" (*ditale* con *digitale*, *scorno*, *scorfano* e *matto* con *mattone*, che più in particolare è un esempio di falso accrescitivo) e lo "scarto sillabico" (*anonimo*, *animo*).

Una "sciarada" richiede di trovare due parole che giustapposte forniscono una terza parola o frase incognita (*ciocco* e *latino* danno *cioccolatino*)

C10 c. morfismi tra linguaggi

C10c.01 Ricordiamo che in algebra un sottoinsieme T di un generico semigruppo S si dice:

ideale da sinistra sse $S\,T = T$, cioè sse è un sottoinsieme invariante per traslazioni da sinistra;

ideale da destra sse $T\,S = T$, cioè sse è un sottoinsieme invariante per traslazioni da destra;

ideale bilatero sse $S\,T\,S = T$, cioè sse è un sottoinsieme invariante per traslazioni da destra e da sinistra;

sottosemigruppo sse $T\,T \subseteq T$.

Evidentemente gli ideali bilateri sono anche ideali da sinistra e ideali da destra.
Inoltre i sottosemigruppi sono anche ideali da sinistra e ideali da destra (nonché ideali bilateri).

Un sottoinsieme N di un monoide M avente $\mathbf{1}$ come unità si dice **sottomonoide** di M sse $\mathbf{1} \in N$ e $N\,N \subseteq N$, ovvero sse $N\,N = N$.

Esaminiamo questi tipi di sottostrutture nel caso del monoide libero A^*.

C10c.02 **Prop.** Consideriamo i linguaggi sopra un alfabeto A:

I linguaggi ideali da sinistra sono esattamente i linguaggi della forma $\mathsf{A}^*\,\mathsf{M}$ per qualche $\mathsf{M} \in \mathbf{Lng}_{\mathsf{A}}$; in altre parole: la collezione dei linguaggi ideali a sinistra è $\{\mathsf{M} \in \mathbf{Lng}_{\mathsf{A}} :| \mathsf{A}^*\mathsf{M}\}$.

In accordo con la dualità-LR, l'insieme dei linguaggi ideali da destra è $\{\mathsf{M} \in \mathbf{Lng}_{\mathsf{A}} : \mathsf{M}\,\mathsf{A}^*\}$.

L'insieme dei linguaggi ideali bilateri è $\{\mathsf{M} \in \mathbf{Lng}_{\mathsf{A}} : \mathsf{A}^*\mathsf{M}\mathsf{A}^*\}$.
Si noti che $\mathsf{A}^*\mathsf{M} \cap \mathsf{M}\,\mathsf{A}^* = \mathsf{A}^*\mathsf{M}\mathsf{A}^*$.

I linguaggi sottosemigruppi hanno forma $\mathsf{T} = \mathsf{M}^+$ per qualsiasi $\mathsf{M} \in \mathbf{Lng}_{\mathsf{A}}$, cioè sono i linguaggi tali che $\mathsf{T}^+ = \mathsf{T}$.

I linguaggi sottomonoidi hanno forma $\mathsf{T} = \mathsf{M}^*$, cioè sono i linguaggi tali che $\mathsf{T}^* = \mathsf{T}$; equivalentemente possiamo affermare che l'insieme dei sottomonoidi di A^* è $\{\mathsf{L} \subseteq \mathsf{A}^* :| \mathsf{L}^*\}$ ∎

Relativamente all'alfabeto A, si dicono **linguaggi cometa** i linguaggi della forma $\mathsf{A}^*\,\mathsf{N}$; sono detti invece **linguaggi anticometa** quelli della forma $\mathsf{M}\,\mathsf{A}^*$; si chiamano invece **linguaggi bicometa** i linguaggi della forma $\mathsf{A}^*\,\mathsf{N}\,\mathsf{A}^* = \mathsf{A}^*\,\mathsf{N} \cap \mathsf{N}\,\mathsf{A}^*$.

Si constata che i riflessi delle comete sono le anticomete, mentre le collezioni delle bicomete, dei sottosemigruppi e dei sottomonoidi sono invarianti per riflessione.

Inoltre possiamo affermare che un linguaggio bicometa della forma $\mathsf{A}^*\mathsf{N}\mathsf{A}^*$ è un linguaggio palindromo sse N è palindromo.

Possiamo inoltre affermare che rispetto alla riflessione le collezioni degli ideali da sinistra e degli ideali da destra sono mutuamente duali-LR, mentre le collezioni dei linguaggi ideali bilateri, dei linguaggi sottosemigruppi e dei linguaggi sottomonoidi sono autoduali-LR.

(1) Eserc. Constatare che le collezioni dei linguaggi cometa, quella dei linguaggi anticometa e quella dei linguaggi bicometa sono collezioni (numerabili) invarianti per somma e giustapposizione e di conseguenza per chiusura-cross e per sommatoria numerabile, ovvero che appartengono alla collezione $\left[\ +, \cdot, {}^{\oplus}, \sum^{+\infty}\ \right]$.

C10c.03 Abbiamo visto che $\langle \mathsf{L}^+, \shortmid \rangle \in \mathbf{Sgrp}$ e che $\langle \mathsf{L}^*, \shortmid, \sqcup \rangle \in \mathbf{Mnd}$; possiamo quindi affermare che L^+ è il sottosemigruppo di A^+ generato da L, mentre L^* è il sottomonoide di A^+ generato da L.

In accordo con questa terminologia algebrica, possiamo affermare che semigruppi liberi e monoidi liberi sono , risp., i semigruppi e i monoidi generati dagli alfabeti.

Dato un sottomonoide M di A^*, è spesso utile individuare un suo insieme di generatori, cioè un linguaggio contenuto in A^* che generi M.
Evidentemente sono più interessanti i sistemi di generatori più ridotti, cioè minimali per l'inclusione.

(1) Prop.: Ogni sottomonoide M di A^* ha un unico insieme di generatori minimale per inclusione individuabile con l'espressione di impostazione:

$$(\mathsf{M} \setminus \text{µ}) \setminus (\mathsf{M} \setminus \text{µ})^2.$$

Dim.: Ogni insieme di generatori di M deve essere contenuto in $\mathsf{M} \setminus \text{µ}$ e per essere minimale non può contenere alcuna stringa in $(\mathsf{M} \setminus \text{µ})^2$, cioè è contenuto nel linguaggio definito dall'espressione.
Viceversa ogni stringa di M è ottenibile come prodotto di elementi del linguaggio sopra espresso ∎

Conviene ribadire che l'espressione precedente (che spesso si trova scritta $(\mathsf{M} - \text{µ}) - (\mathsf{M} - \text{µ})^2$) fornisce solo una indicazione genericamente formale per la ricerca dell'insieme di generatori minimale di M.
Se si rende necessario controllare efficacemente l'insieme generatore minimale di uno specifico $\mathsf{M} <_{Mnd} \mathsf{A}^*$ (in particolare se occorre decidere se tale insieme è finito o infinito) occorre individuare un procedimento effettivo che si basa sulle proprietà specifiche che determinano M.
Le stesse considerazioni valgono per le espressioni b10(1) e (2).

C10c.04 In generale si pone il problema della utilizzabilità delle espressioni matematiche, ossia delle formule che esprimono risultati matematici.

Alcune delle espressioni come la (1) e le due sopraccitate servono solo ad inquadrare un problema che dovrà essere approfondito tenendo conto di proprietà specifiche.

All'opposto talune espressioni, in particolare le uguaglianze che si incontrano nell'algebra elementare e nell'analisi numerica, contengono tutte le indicazioni che consentono calcoli effettivi e lasciano aperti solo questioni riguardanti le tolleranze consentite alle approssimazioni.

Vi sono poi situazioni intermedie rappresentate da espressioni contenenti operazioni ben definite, cioè calcolabili con algoritmi consolidati, e operazioni definite in modo generico, vaghe, che richiedono ulteriori approfondimenti che coinvolgono operandi che non si sanno completamente dominare.

Per gli approfondimenti che molte formule lasciano in sospeso riveste notevole importanza il livello di generalità consentito per gli argomenti che occorrono nelle espressioni. Tipicamente contano gli operandi e i domini delle funzioni e delle relazioni.

La portata di una formula cresce al crescere dei domini di variabilità dei suoi argomenti. Questa tendenza si può trovare in conflitto con la utilizzabilità effettiva dell'espressione.
Una espressione riguardante domini più estesi tende ad essere meno utilizzabile di una versione della stessa formula riguardante domini meno estesi.
I compromessi tra queste esigenze conflittuali sono oggetto di studi di notevole imoportanza. Queste tematiche sono definite in modo particolarmente chiaro quando espresse in termini di digrafi e di linguaggi formali.

C10c.05 Ricordiamo che si dice **morfismo** dal monoide $\langle M, \cdot, 1\rangle$ nel monoide $\langle N, \odot, \mathbf{1}\rangle$ ogni applicazione $\psi \in \lceil M \longmapsto N \rceil$ tale che $\forall u, v \in M \ : \ \psi(u \cdot v) = \psi(u) \odot \psi(v)$ e $\psi(1) = \mathbf{1}$.

(1) Prop.: Consideriamo un insieme G, un monoide M, una funzione $\mathsf{f} \in \lceil G \longmapsto M \rceil$ e l'iniezione naturale $\boldsymbol{i} := \text{natinj}(G, G^*) = \lceil g \in G \mapsto G^* \rceil$.
Esiste un unico morfismo di monoidi $\phi \in \lceil G^* \longmapsto M \rceil$ tale che $\mathsf{f} = \text{natinj}(G, G^*) \circ_{lr} \phi$, cioè tale da rendere commutativo il diagramma funzionale

//input pC10c05 ∎

Questa proprietà di G^*, chiamata **proprietà universale** equivale ad affermare che G^* sia un **monoide libero**.

La situazione della proposizione (1) nel caso G sia un alfabeto finito A conduce al morfismo ψ dal monoide libero A^* in un monoide $\langle M, \odot, \mathbf{1}\rangle$ che risulta definito quando si forniscono i trasformati $\psi(\mathtt{a}_i)$ per tutti i caratteri $\mathtt{a}_i \in \mathsf{A}$, cioè per tutti gli elementi generatori del dominio: infatti per ogni $\mathtt{a}_{j_1}, \mathtt{a}_{j_2}, ..., \mathtt{a}_{j_r} \in \mathsf{A}^*$ si ottiene

$$\psi(\mathtt{a}_{j_1}\,\mathtt{a}_{j_2}\,...\,\mathtt{a}_{j_r}) \;=\; \psi(\mathtt{a}_{j_1}) \odot \mathtt{a}_{j_2}) \odot \,...\, \odot \psi(\mathtt{a}_{j_r}) \;.$$

C10c.06 I morfismi tra due monoidi liberi A^* e B^* sono anche detti, motivatamente, **sostituzioni di caratteri con stringhe**.

Ciascuno di tali morfismi ψ è definito dalla funzione $\gamma = \lceil\, \mathtt{a}_i \in \mathsf{A} \mapsto \psi(\mathtt{a}_i) \,\rceil$ e gli si può assegnare la forma $\psi = \gamma^{\otimes} = \lceil\, \mu \mapsto \mu \,\rceil \cup \gamma \cup \gamma \circ \gamma \cup \gamma^{\circ 3} \cup \ldots$.

Sono interessanti alcuni sottoinsiemi specifici di un insieme di morfismi $\mathbf{M} := \lceil\, A^* \longmapsto B^* \,\rceil$.

Un morfismo $\psi \in \mathbf{M}$ si dice **cancellazione di caratteri** sse $\psi(\mathtt{a}_i) = \mu$ per qualche $\mathtt{a}_i \in \mathsf{A}$.

SLa funzione $\psi \in \mathbf{M}$ si dice **funzione che rispetta la lunghezza** sse per ogni $\mathtt{a}_i \in \mathsf{A}$ si ha $\psi(\mathtt{a}_i) \in \mathsf{B}$.

Le cancellazioni di caratteri non rispettano la lunghezza ∎

Un morfismo $\gamma^{\otimes} =: \psi$ va chiamato **isomorfismo** sse $\gamma \in \lceil\, A \lhd\!\longrightarrow\!\rhd B \,\rceil$, ovvero sse

$$\forall \mathtt{a}_i, \mathtt{a}_j \in \mathsf{A} \quad : \quad \psi(\mathtt{a}_i) = \psi(\mathtt{a}_j) \Longrightarrow \mathtt{a}_i = \mathtt{a}_j \;.$$

Tale trasformazione rispetta la lunghezza.

Si chiamano **morfismi invertibili** i morfismi del genere $\mathsf{A}^* \lhd\!\longrightarrow \mathsf{B}^*$; si tratta di trasformazioni invertibili e quindi tali da consentire, per ogni $w \in \mathsf{A}^*$, la ricostruzione di w a partire da $\psi(w)$ come $w = (\psi^{-\mathbf{1}})(\psi(w))$.

Queste trasformazioni sono l'oggetto della teoria dei codici, una disciplina, oggi assai sviluppata e sostegno di importanti applicazioni [C65, C66].

C10c.07 Per estensione booleana le sostituzioni si applicano anche ai linguaggi definendo:

$$(1) \qquad \forall \mathsf{L} \in \mathbf{Lng}_{\mathsf{A}} \;:\; \psi(\mathsf{L}) := \sum_{\mathtt{a}_{j_1}\,\mathtt{a}_{j_2}\,...\,\mathtt{a}_{j_r} \in \mathsf{L}} \psi(\mathtt{a}_{j_1}) \cdot \psi(\mathtt{a}_{j_2}) \cdot \,...\, \cdot \psi(\mathtt{a}_{j_r}) \;.$$

Questa espressione si può utilizzare per definire più in generale l'applicazione a un linguaggio su un alfabeto A di una funzione del genere $\psi \in \lceil\, \mathsf{A} \longmapsto \mathbf{Lng}_{\mathsf{B}} \,\rceil$ la quale a ogni carattere di A fa corrispondere un linguaggio su B.

Questa funzione si dice **sostituzione di caratteri con linguaggi**; essa non costituisce un morfismo tra monoidi liberi ma un morfismo tra algebre di Kleene di linguaggi. Infatti si dimostra facilmente che:

$$(2) \qquad \psi(\mathsf{L} + \mathsf{M}) \;=\; \psi(\mathsf{L}) + \psi(\mathsf{M}) \quad , \quad \psi(\mathsf{L} \cdot \mathsf{M}) \;=\; \psi(\mathsf{L}) \cdot \psi(\mathsf{M}) \quad , \quad \psi(\mathsf{L}^*) \;=\; (\psi(\mathsf{L}))^* \;.$$

Ad esempio., se $\psi(\mathtt{a}) := \mathtt{c}\,\mathtt{d}^+\,\mathtt{c}$ e $\psi(\mathtt{b}) := \mathtt{c}\,\mathtt{d}\,\mathtt{c}$ si ha:

$$\psi(\,\mu + \mathtt{a}^2 + \mathtt{b}\,\mathtt{a}^2\,\mathtt{b}) \;=\; \mu + \mathtt{c}\,\mathtt{d}^+\,\mathtt{c}^2\,\mathtt{d}^+\mathtt{c} + \mathtt{c}\,\mathtt{d}\,\mathtt{c}^2\,\mathtt{d}^+\,\mathtt{c}^2\,\mathtt{d}^+\,\mathtt{c}^2\,\mathtt{d}\,\mathtt{c} \;.$$

C10c.08 Gli isomorfismi del genere $\lceil\, \mathsf{A}^* \lhd\!\longrightarrow \{0,1\}^* \,\rceil$ sono chiamati anche **codifiche binarie**. Essi consentono di rappresentare mediante sequenze di bits i più svariati tipi di stringhe e quindi i più diveersi tipi di informazioni mantenendo la possibilità di risalire alla loro forma originaria.

Le rappresentazioni binarie delle informazioni consentono di registrarle, analizzarle e modificarle servendosi degli odierni strumenti dell'informatica e delle telecomunicazioni e quindi con grandi vantaggi in termini di velocità, di versatilità e di componibilità delle elaborazioni e con la conseguente possibilità di raccogliere, strutturare, riutilizzare e far circolare grandi collezioni di dati [v.a `big data (we)`].

In particolare si hanno le codifiche binarie delle decine di caratteri utilizzati correntemente con strumenti elettronici seguendo il codice `ASCII (wi)`, e le codifiche binarie delle decine di migliaia di segni degli alfabeti delle lingue di qualche importanza seguendo il codice `Unicode (wi)`.

I procedimenti e i metodi che consentono di codificare e decodificare le informazioni afferiscono alla sopraccitata teoria dei codici.

C10 d. relazioni tra stringhe e derivazioni di linguaggi

C10d.01 Consideriamo $u, v, w \in \mathsf{A}^*$.

Si dice che u è **prefisso** o **fattore sinistro** di w, e si scrive $u \preceq_{\mathtt{p}} w$, sse $w \in u\,\mathsf{A}^*$.

Si dice che u è **suffisso**, **postfisso** o **fattore destro** di w, e si scrive $u \preceq_{\mathtt{s}} w$ sse $w \in \mathsf{A}^*\,u$.

Si dice che u è **infisso** o **fattore** di w, e si scrive $u \preceq_{\mathtt{i}} w$, sse $w \in \mathsf{A}^*\,u\,\mathsf{A}^*$.

Si dice che u è sottostringa o **sottosequenza** di w, e si scrive $u \preceq_{\mathtt{u}} w$, sse, essendo $u =_{\mathsf{A}} \mathtt{a}_{j_1}\,\mathtt{a}_{j_2} \cdots \mathtt{a}_{j_r}$, si ha

$w \in \mathsf{A}^*\,\mathtt{a}_{j_1}\,\mathsf{A}^*\,\mathtt{a}_{j_2}\,\mathsf{A}^* \cdots \mathsf{A}^*\,\mathtt{a}_{j_r}\mathsf{A}^*$.

Le relazioni tra stringhe $\preceq_{\mathtt{p}}$, $\preceq_{\mathtt{s}}$, $\preceq_{\mathtt{i}}$ e $\preceq_{\mathtt{u}}$ sono evidentemente delle relazioni d'ordine parziale; questo ordine non è totale sse $|\mathsf{A}| \geq 2$.

Accanto a queste quatto relazioni si introducono quattro funzioni del genere $\lceil\, \mathsf{A}^* \longmapsto \mathbf{Lng}_{\mathsf{A}} \,\rceil$ che ad ogni stringa $w \in \mathsf{A}^*$ associano:

l'insieme dei suoi prefissi: $\mathbf{Pfx}(w) = w^{\mathbf{Pfx}}$;

l'insieme dei suoi suffissi: $\mathbf{Sfx}(w) = w^{\mathbf{Sfx}}$;

l'insieme dei suoi infissi: $\mathbf{Ifx}(w) = w^{\mathbf{Ifx}}$;

l'insieme delle sue sottosequenze (o sottotringhe): $\mathbf{Useq}(w) = w^{\mathbf{Useq}} = \mathbf{Sbseq}(w)$.

L'insieme dei prefissi di $w = \mathtt{abaa}$ è $\mathbf{Pfx}(w) = \{\sqcup, \mathtt{a}, \mathtt{ab}, \mathtt{aba}, \mathtt{abaa}\}$;
per l'insieme dei suffissi di w abbiamo $w^{\mathbf{Sfx}} = \mathbf{Sfx}(w) = \{\sqcup, \mathtt{a}, \mathtt{aa}, \mathtt{baa}, \mathtt{abaa}\}$;
l'insieme dei suoi infissi è $\mathbf{Ifx}(w) = \{\sqcup, \mathtt{a}, \mathtt{b}, \mathtt{aa}, \mathtt{ab}, \mathtt{ba}, \mathtt{aba}, \mathtt{baa}, \mathtt{abaa}\}$;
Per avere l'insieme delle sottostringhe di w $\mathbf{Useq}(w)$ occorre aggiungere all'insieme dei suoi infissi $\mathtt{a}^3$, cioè $w^{\mathbf{Useq}} = \{\sqcup, \mathtt{a}, \mathtt{b}, \mathtt{aa}, \mathtt{ab}, \mathtt{ba}, \mathtt{aaa}, \mathtt{aba}, \mathtt{baa}, \mathtt{abaa}\}$.

C10d.02 Sono utili anche le estensioni booleane delle 4 funzioni precedenti che noi denoteremo con gli stessi simboli; esse appartengono al genere $\lceil\, \mathbf{Lng}_{\mathsf{A}} \longmapsto \mathbf{Lng}_{\mathsf{A}} \,\rceil$. Si segnala anche che queste funzioni applicate a stringhe portano a linguaggi finiti, ossia a linguaggi polinomiali.

Esse possono essere considerate endofunzioni entro $\mathbf{Lng}_{\mathsf{A}}$ e questo consente di comporre queste ed altre funzioni che trasformano linguaggi in linguaggi. formali.

Ad esempio si osserva che l'insieme degli infissi di w è un linguaggio ottenibile come insieme dei prefissi dei vari suffissi di w e coincide con l'insieme dei suffissi dei vari prefissi di w, ossia

$$\forall w \in ASs^* \;:\; \mathbf{Pfx}(\mathbf{Sfx}(w)) = \mathbf{Sfx}(\mathbf{Pfx}(w)) = \mathbf{Ifx}(w) \;.$$

Questa doppia uguaglianza vale anche quando l'argomento è un linguaggio, ossia

$$\forall \mathsf{L} \subseteq \mathsf{A}^* \;:\; \mathbf{Pfx}(\mathbf{Sfx}(\mathsf{L})) = \mathbf{Sfx}(\mathbf{Pfx}(\mathsf{L})) = \mathbf{Ifx}(\mathsf{L}) \;.$$

Si constata facilmente che le endofunzioni entro A^* **Pfx**, **Sfx**, **Ifx** e **Useq** sono idempotenti.

Queste quattro endofunzioni consentono di individuare quattro collezioni di linguaggi particolari caratterizzati come loro punti fissi.

La collezione $[\,\mathbf{Pfx}\,]$ dei linguaggi invarianti per **Pfx**, linguaggi L tali che $(\mathbf{Pfx}(\mathsf{L}) = \mathsf{L}$.
La collezione $[\,\mathbf{Pfx}\,]$ dei linguaggi invarianti per **Sfx**, linguaggi L tali che $(\mathbf{Sfx}(\mathsf{L}) = \mathsf{L}$.
La collezione $[\,\mathbf{Ifx}\,]$ dei linguaggi invarianti per **Ifx**, linguaggi L tali che $(\mathbf{Ifx}(\mathsf{L}) = \mathsf{L}$.
La collezione $[\,\mathbf{Useq}\,]$ dei linguaggi invarianti per **Useq**, linguaggi L tali che $(\mathbf{Useq}(\mathsf{L}) = \mathsf{L}$.

Si osserva chei linguaggi invarianti per **Useq** sono invarianti anche per **Ifx**, ma vi sono linguaggi invarianti per **Ifx** che non sono invarianti per **Useq**; concisamente abbiamo

$$[\ \textsf{Useq}\ ] \subset [\ \textsf{Ifx}\ ]\ .$$

Si trova inoltre che $[\ \textsf{Ifx}\ ] \subset [\ \textsf{Pfx}\ ]$ e dualmente-LR abbiamo $[\ \textsf{Ifx}\ ] \subset [\ \textsf{Sfx}\ ]$.

C10d.03 Munendo A^* con ciascuna delle relazioni $\preceq_{\mathtt{x}}$ per $\mathtt{x} \in \{\mathtt{p}, \mathtt{s}, \mathtt{i}, \mathtt{u}\}$ si ottiene un digrafo infinito graduato, i cui livelli contengono elementi delle successive potenze cartesiane A^r.

Quando $\mathsf{A} = \{\mid\}$ (ed in genere quando A è costituito da un solo carattere) si ha il semplice digrafo numerabile e funzionale

$$\sqcup \longrightarrow \mid \longrightarrow \mid\mid \longrightarrow \mid\mid\mid \longrightarrow \mid\mid\mid\mid \longrightarrow \ \ldots$$

isomorfo a quello di $\mathbb{N}$ munito della relazione "immdiatamente minore" $<_{\mid}$.

Si osserva che per ogni linguaggio M su un solo carattere, in particolare su $\mid$, abbiamo

$$\textsf{Pfx}(M) = \textsf{Sfx}(M) = \textsf{Ifx}(M) = \textsf{Sbseq}(M)\ .$$

Inoltre se M è finito $\textsf{Pfx}(M) = \textsf{Pfx}(W)$ dove W denota la stringa più lunga in M, mentre se M è infinito $\textsf{Pfx}(M) = \{\mid\}^*$.

C10d.04 Sono molto più articolate delle precedenti le relazioni $u \preceq_{\mathtt{p}} w$, $u \preceq_{\mathtt{s}} w$, $u \preceq_{\mathtt{i}} w$ $u \preceq_{\mathtt{u}} w$ per un dato alfabeto A di due o più caratteri.

Innanzi tutto si osserva che esse sono relazioni di ordine parziale. Si osserva poi che tra le relazioni precedenti considerate come insiemi di coppie di stringhe su A, cioè come sottoinsiemi di $\mathsf{A}^* \times \mathsf{A}^*$, sussistono le seguenti inclusioni:

$$(1) \qquad \emptyset \ \subset\ \preceq_{\mathtt{p}} \cap \preceq_{\mathtt{s}} \ \subset\ \preceq_{\mathtt{p}}\ , \preceq_{\mathtt{s}} \ \subset\ \preceq_{\mathtt{p}} \cup \preceq_{\mathtt{s}} \ \subset\ \preceq_{\mathtt{i}} \ \subset\ \preceq_{\mathtt{u}}\ .$$

Insieme a queste relazioni interessano le loro associate riduzioni nonriflessive (univocamente determinate) che sono relazioni d'ordine:stretto; le denoteremo, risp., con

$$\prec_{\mathtt{p}} \quad , \quad \prec_{\mathtt{s}} \quad , \quad \prec_{\mathtt{i}} \quad , \quad \prec_{\mathtt{u}}\ ;$$

Per esse useremo i termini, risp., prefisso proprio (o fattore sinistro proprio), suffisso proprio (o fattore destro proprio), infisso proprio o fattore proprio e sottostringa propria o sottosequenza propria.

Per esse vale una relazione di inclusione analoga alla (1):

$$(2) \qquad \emptyset \ \subset\ \prec_{\mathtt{p}} \cap \prec_{\mathtt{s}} \ \subset\ \prec_{\mathtt{p}}\ , \prec_{\mathtt{s}} \ \subset\ \prec_{\mathtt{p}} \cup \prec_{\mathtt{s}} \ \subset\ \prec_{\mathtt{i}} \ \subset\ \prec_{\mathtt{u}}\ .$$

Inoltre sono spesso utili le corrispondenti relazioni di predecessore immediato, ovvero le loro riduzioni radice-star, e le rispettive relazioni trasposte di successore immediato.
Limitiamoci a considerare solo le relazioni di predecessore immediato; le denoteremo, risp., con $\prec_{\mathtt{pI}}$, $\prec_{\mathtt{sI}}$, $\prec_{\mathtt{iI}}$ e $\prec_{\mathtt{uI}}$ e per esse useremo i termini, risp., **prefisso immediato**, **suffisso immediato**, **infisso immediato** e **sottostringa immediata**.

Anche per esse vale una relazione di inclusione analoga alla (1):

$$\emptyset \ \subset\ \prec_{\mathtt{pI}} \cap \prec_{\mathtt{sI}} \ \subset\ \prec_{\mathtt{pI}}\ , \prec_{\mathtt{sI}} \ \subset\ \prec_{\mathtt{pI}} \cup \prec_{\mathtt{sI}} = \prec_{\mathtt{iI}} \ \subset\ \prec_{\mathtt{uI}}\ .$$

Queste relazioni si constata che sono fornite dalle seguenti espressioni:

$$\prec_{\mathtt{pI}} = \bigcup_{w \in \mathsf{A}^*} w \times (w \mid \mathsf{A}) \quad , \quad \prec_{\mathtt{sI}} = \bigcup_{w \in \mathsf{A}^*} w \times (\mathsf{A} \mid w) \quad ,$$

$$\prec_{\mathtt{pI}} \cup \prec_{\mathtt{sI}} = \ \sqcup \times \mathsf{A} \ \dot\cup \bigcup_{w\in\mathsf{A}^*} \Big(w \times (w \shortmid \mathsf{A})) \cup (w \times (\mathsf{A} \shortmid w)\Big) \quad ,$$

$$\prec_{\mathtt{pI}} \cap \prec_{\mathtt{sI}} = \Big\{\langle a,n\rangle \in \mathsf{A}\times\mathbb{N} \ :| \ \langle a^n, a^{n+1}\rangle\Big\} \quad , \quad \prec_{\mathtt{iI}} = \bigcup_{w\in\mathsf{A}^*} w \times \{w \shortmid \mathsf{A} \cup \mathsf{A} \shortmid w\} \quad .$$

C10d.05 Per ogni $w \in A^+$ gli insiemi $w \prec_{\mathtt{pI}}$ e $w \prec_{\mathtt{sI}}$ sono costituiti da una sola stringa: quindi A^* munito, risp., dell'insieme di archi $\dot\bigcup_{w\in A^*} \Big(\bigcup_{x\in w^{\succ}{}_{\mathtt{pI}}} \langle w,x\rangle\Big)$ e dell'insieme di archi $\dot\bigcup_{w\in A^*} \Big(\bigcup_{x\in w^{\succ}{}_{\mathtt{sI}}} \langle w,x\rangle\Big)$ costituisce due arborescenze illimitate; esse di solito vengono raffigurate svilupparsi verso l'alto [D30a].

//input pC10d04

//input pC10d04B

Munendo il terreno A^* con l'insieme $\dot\bigcup_{w\in A^*} \Big(\bigcup_{x\in w\,\wr \succ_{\mathtt{iI}}} \langle w,x\rangle\Big)$ si ottiene un digrafo.N graduato che non è un'arborescenza, in quanto per ogni $w \in \mathsf{A}\mathsf{A}^+$ l'insieme $\prec_{\mathtt{iI}}(w)$ contiene una sola stringa sse $\left|w^{\mathrm{minalf}}\right| = 1$, mentre in caso contrario ne contiene almeno $\left|w^{\mathrm{minalf}}\right|$.

//input pC10d05C

//input pC10d05D

Il terreno A^+ munito degli archi corrispondenti alle coppie costituenti l'ordine $\succ_{\mathtt{uI}}$ conduce a un digrafo.N graduato che si può ottenere dal digrafo.N che corrisponde a $\succ_{\mathtt{iI}}$ per aggiunta di numerosi archi.

//input pC10d05E

(1) Eserc. Relativamente a $\mathsf{A} = \{a,b,c\}$, tracciare i primi quattro livelli, cioè fino a raggiungere le stringhe di lunghezza 3, dei digrafi.N rappresentanti $\succ_{\mathtt{iI}}$ e $\succ_{\mathtt{uI}}$.

C10d.06 Della operazione di riflessione si può considerare l'estensione cartesiana applicata alle coppie o alle sequenze di stringhe e successivamente l'estensione booleana di questa estensione cartesiana. Utilizzando lo stesso simbolo per queste estensioni per le trasformazioni delle sequenze di stringhe e linguaggi abbiamo

$$\langle w,x\rangle^{\leftarrow} := \langle w^{\leftarrow}, x^{\leftarrow}\rangle \quad , \quad \langle \mathsf{L},\mathsf{M}\rangle^{\leftarrow} := \langle \mathsf{L}^{\leftarrow}, \mathsf{M}^{\leftarrow}\rangle \ .$$

$$\big\langle w_1, w_2, ..., w_n\big\rangle^{\leftarrow} := \big\langle {w_1}^{\leftarrow}, {w_2}^{\leftarrow}, ..., {w_n}^{\leftarrow}\big\rangle \quad , \quad \big\langle \mathsf{L}_1, \mathsf{L}_2, ..., \mathsf{L}_n\big\rangle^{\leftarrow} := \big\langle {\mathsf{L}_1}^{\leftarrow}, {\mathsf{L}_2}^{\leftarrow}, ..., {\mathsf{L}_n}^{\leftarrow}\big\rangle \ .$$

Per l'estensione della riflessione alle coppie di stringhe e di linguaggi sarà utilizzato preferibilmente il termine specifico **biriflessione**.

Si verifica facilmente che le relazioni d'ordine $\preceq_{\mathtt{p}}$ e $\preceq_{\mathtt{s}}$ sono l'una la biriflessa dell'altra e che $\preceq_{\mathtt{i}}$ e $\preceq_{\mathtt{u}}$ sono invarianti per biriflessione.

Questo discende dalle implicazioni sulle coppie di stringhe:

$$w \preceq_{\mathtt{p}} x \iff (w^{\leftarrow}) \preceq_{\mathtt{s}} (x^{\leftarrow}) \quad , \quad w \preceq_{\mathtt{i}} x \iff (w^{\leftarrow}) \preceq_{\mathtt{i}} (x^{\leftarrow}) \quad , \quad w \preceq_{\mathtt{u}} x \iff (w^{\leftarrow}) \preceq_{\mathtt{u}} (x^{\leftarrow}) \; .$$

Può servire anche l'operazione unaria sulle coppie di stringhe o di linguaggi consistente nello scambio dei due membri della coppia stessa. Per questa operazione usiamo il termine trasposizione e scriviamo

$$\langle w, x\rangle^{\mathsf{T}} := \langle x, w\rangle \qquad , \qquad \langle \mathsf{L}, \mathsf{M}\rangle^{\mathsf{T}} = \langle \mathsf{M}, \mathsf{L}\rangle \; .$$

Queste due operazioni unarie sono evidentemente delle involuzioni e possono essere applicate successivamente ottenendo

$$(\langle w, x\rangle^{\leftarrow})^{\mathsf{T}} = \left(\langle w, x\rangle^{\mathsf{T}}\right)^{\leftarrow} = \langle x^{\leftarrow}, w^{\leftarrow}\rangle \qquad , \qquad (\langle \mathsf{L}, \mathsf{M}\rangle^{\leftarrow})^{\mathsf{T}} = \left(\langle \mathsf{L}, \mathsf{M}\rangle^{\mathsf{T}}\right)^{\leftarrow} = \langle \mathsf{M}^{\leftarrow}, \mathsf{L}^{\leftarrow}\rangle \; .$$

C10d.07 In varie applicazioni risulta utile affiancare a ciascuno dei 4 ordini parziali di A^* visti in precedenza l'ordine opposto e gli ordini che differiscono da questi due solo in quanto ottenuti procedendo sulle stringhe da destra a sinistra invece che da sinistra a destra.

Molti algoritmi riguardanti stringhe e linguaggi per essere ben definiti richiedono che si faccia riferimento a un ordine totale delle stringhe che estenda uno degli ordini parziali precedenti $\preceq_{\mathtt{p}}$, $\preceq_{\mathtt{s}}$, $\preceq_{\mathtt{i}}$ e $\preceq_{\mathtt{u}}$.
Se denotiamo uno degli ordini precedenti con $\preceq_{\mathtt{y}}$ e il richiesto ordine totale con $\sqsubseteq$, chiediamo che sia

$$w \preceq_{\mathtt{y}} x \implies w \sqsubseteq x \; ,$$

mentre non chiediamo il viceversa: $w \sqsubseteq x \not\implies w \preceq x$.

Segnaliamo che un teorema di Garret Birkhoff garantisce che risulta possibile l'estensione di ogni ordine parziale sopra un terreno finito o numerabile a ordine totale compatibile.

Introduciamo due ordini totali basati sopra un ordine totale $\leq$ dell'alfabeto A, cioè basati su una sequenzializzazione delle lettere di A che enunciamo scrivendo $\langle \mathtt{a}_1 < \mathtt{a}_2 < ... < \mathtt{a}_n \rangle$.

Cominciamo con il confronto tra due stringhe della stessa lunghezza $w =_{\mathsf{A}} \mathtt{a}_{j_1}...\mathtt{a}_{j_r}$ e $x =_{\mathsf{A}} \mathtt{a}_{h_1}...\mathtt{a}_{h_r}$; si dice che w **precede secondo l'ordine lessicografico** x, e si scrive $w <_{lxg} x$, sse si trova $k \in \{1, ..., r\}$ tale che $i < k \Longrightarrow \mathtt{a}_{j_i} = \mathtt{a}_{h_i}$ e $\mathtt{a}_{j_k} < \mathtt{a}_{h_k}$.
Per esempio, riferendoci all'usuale ordinamento alfabetico $\langle \mathtt{a} < \mathtt{b} < \mathtt{c} < .. < \mathtt{z}\rangle$, abbiamo

$$\mathtt{acbaz} <_{lxg} \mathtt{acbba} <_{lxg} \mathtt{accaa} \; .$$

C10d.08 Facciamo riferimento a un alfabeto A ordinato dalla richiesta $\langle a_1 < a_2 < \cdots < a_n\rangle$ e ricordiamo che per ogni $r \in \mathbb{P}$ le stringhe di A^r sono ordinate totalmente dalla relazione $\leq_{lxg}$.

Risulta semplice passare da una stringa di questo insieme a quella immediatamente precedente o a quella immediatamente successiva e conseguentemente risulta possibile procedere alla visita di A^r secondo l'ordinamento $\leq_{lxg}$ o secondo l'ordine totale opposto $\geq_{lxg}$.

Descriviamo l'algoritmo della suddetta visita di A^r denotando con $w = a_{i_1} a_{i_2} \cdots a_{i_r}$ la stringa di A^r attualmente visitata.

[1] si genera come prima stringa ${a_1}^r$;
[2] se la lettera nella posizione più a destra della w $a_{i_r} =: a_s$ non è a_n, l'ultima in A, la si sostituisce con la successiva, a_{s+1}; quindi si emette come nuova stringa quella ottenuta e si torna alla [2];
[3] se $a_{i_r} = a_n$ si arretra cercando la posizione più a destra della w, j, nella quale si abbia $a_{i_j} =: a_t < a_n$;
[4] se si trova questo $j \geq 1$, si pone a_{t+1} nella posizione j e si pone a_1 nelle posizioni successive $h = j, j+1, ..., r$; quindi si emette come nuova stringa quella ottenuta e si torna alla [2];

[5] se non si trova j la w precedentemente ottenuta è l'ultima di A^r e si arresta l'esecuzione.

L'ordine totale lessicografico si estende all'intero A^* mediante il seguente algoritmo di confronto di due stringhe generiche $w = \mathtt{a}_{j_1}...\mathtt{a}_{j_r}$ e $x = \mathtt{a}_{h_1}...\mathtt{a}_{h_s}$:

[1] scorrendo le due stringhe in parallelo si cerca il primo $k \leq \min(r,s)$ tale che $i < k \Longrightarrow \mathtt{a}_{j_i} = \mathtt{a}_{h_i}$ e $\mathtt{a}_{j_k} \neq \mathtt{a}_{h_k}$;

[2] se non si trova tale k sono possibili tre situazioni:
- se $w = x$ si decide $w =_{lxg} x$;
- se w è prefisso proprio di x si decide $w <_{lxg} x$;
- se x è prefisso di w si decide $w >_{lxg} x$;

[3] quando si incontra il suddetto k:
- se $\mathtt{a}_{j_k} < \mathtt{a}_{h_k}$ si decide $w <_{lxg} x$,
- se $\mathtt{a}_{j_k} > \mathtt{a}_{h_k}$ si stabilisce $w >_{lxg} x$.

C10d.09 Eserc. Consideriamo $u, v \in \mathsf{A}^+$; si constati che

(a) $u <_{lxg} v \iff v \in u\,\mathsf{A}^+ \ \vee \ u =: w\,\mathtt{a}\,x\ ,\ v =: w\,\mathtt{b}\,y$ con $w, x, y \in \mathsf{A}^*, \mathtt{a}, \mathtt{b} \in \mathsf{A}, \mathtt{a} < \mathtt{b}$.

(b) $\forall w \in \mathsf{A}^* \ : \ u <_{lxg} v \iff u\,w <_{lxg} v\,w$.

(c) $u <_{lxg} v \implies \lceil \forall x, y \in \mathsf{A}^* \ : \ u\,x <_{lxg} v\,y \rceil$.

L'ordinamento lessicografico è un ordine totale che estende l'ordinamento parziale $\preceq_{\mathtt{p}}$; il suo primo elemento è $\sqcup$ e la stringa immediatamente successiva secondo $\leq_{lxg}$ di una generica w è $w\mathtt{a}_1$.
Non è invece possibile individuare una stringa che precede immediatamente secondo $<_{lxg}$ una qualsiasi parola di $\mathsf{A}^* \setminus \{\sqcup, \mathtt{a}_1\}$: infatti tra due stringhe $u\,\mathtt{a}_i\,x$ e $u\,\mathtt{a}_h\,y$ con $\mathtt{a}_i < \mathtt{a}_h$ si trovano le infinite stringhe della forma $u\mathtt{a}_i\,\mathtt{a}_j\,\mathsf{A}^*$ per tutti gli $\mathtt{a}_j < \mathtt{a}_h$ mentre tra w e $w\,{\mathtt{a}_1}^m\,\mathtt{a}_k\,y$ con $k \geq 1$, $\mathtt{a}_k > \mathtt{a}_1$ e $y \in \mathsf{A}^*$ si trovano le infinite stringhe dell'insieme $w\,{\mathtt{a}_1}^m\,\mathtt{a}_1\,\mathsf{A}^*$.

Quindi l'ordinamento lessicografico non è un ordine sequenziale.

C10d.10 Un ordinamento sequenziale di A^* che estende $\preceq_{\mathtt{p}}$ e che risulta spesso utile è l'ordinamento l'**ordinamento secondo lunghezza-lessicografico** che denotiamo con $\leq_{llx}$.
Esso si basa sul seguente algoritmo di confronto di due generiche stringhe diverse w e x.

[1] Primariamente si confrontano le lunghezze di w e x; se $w^{\vdash} < x^{\vdash}$ si decide $w <_{llx} x$, mentre se $w^{\vdash} > x^{\vdash}$ si stabilisce $w >_{llx} x$;

[2] se le due lunghezze coincidono si effettua il loro confronto lessicografico e si decide di conseguenza, cioè si pone $w <_{llx} x$ sse $w <_{lxg} x$.

L'ordine $\leq_{llx}$ è un ordine sequenziale ed è agevole passare da una stringa w a quella immediatamente precedente o successiva. ¶Per passare dalla w alla successiva, se w è l'ultima secondo $\leq_{lxg}$ tra quelle aventi la sua stessa lunghezza $r := w^{\vdash}$ si passa alla prima della lunghezza successiva, cioè alla ${\mathtt{a}_1}^{r+1}$; se invece non è l'ultima si passa alla immediatamente successiva secondo $\leq_{lxg}$ tra quelle della stessa lunghezza.

Per passare dalla w diversa dalla stringa muta alla precedente, sse w è la prima secondo $\leq_{lxg}$ tra quelle aventi la sua stessa lunghezza $r := w^{\vdash}$ si passa all'ultima della lunghezza precedente, cioè alla ${\mathtt{a}_n}^{r-1}$; se invece non è la prima si passa alla immediatamente precedente secondo $\leq_{lxg}$ tra quelle della stessa lunghezza r.

Evidentemente risulta semplice anche organizzare una procedura che a partire da una qualsiasi $w \in \mathsf{A}^*$ avanzi (o retroceda) secondo $\leq_{llx}$ di un qualsiasi numero t di passi.

C10d.11 Vediamo in dettaglio una procedura di visita illimitata che pone in corrispondenza biunivoca A^* con l'insieme degli interi naturali $\mathbb{N}$.

Per questo introduciamo la notazione $\mathsf{sopi}_{n,r}$ con $r \in \mathbb{N}$ per esprimere la somma delle potenze di n n^i per $i = 0, 1, 2, ..., r$ e osserviamo che tale somma esprime il cardinale delle stringhe su A con $|\mathsf{A}| = n$ aventi lunghezza minore o uguale a r, ossia

$$\mathsf{sopi}_{n,r} \ := \ 1 + n + n^2 + \cdots + n^r \ = \ \left|\mathsf{A}^{\leq r}\right| .$$

Inoltre utilizziamo la variabile intera `m` da utilizzare come fosse una variabile del linguaggio c++ con il ruolo di indice corrente delle stringhe successivamente visitate.

[1] Si inizia con $\sqcup$, cui si fa corrispondere l'intero `m = 0` ;
[2] si procede primariamente sulle successive lunghezze delle stringhe $r = 1, 2, 3, ...$;
  [2.1] nell'ambito di un A^r si comincia con $\mathsf{a}_1{}^r$ e le si associa $\mathsf{sopi}_{n,r-1}$, cioè l'intero corrente `m` precedentemente aumentato di 1 (in c++ si scriverebbe `++m;`);
  [2.2] si passa da una generica $u\,\mathsf{a}_i\,\mathsf{a}_n{}^q$ con $i = 1, 2, ..., n-1$ e $q = 0, ..., |u| - 1$ alla immediatamente successiva $u\,\mathsf{a}_{i+1}\,\mathsf{a}_1{}^q$; a questa si associa `++m`;
  [2.3] quando si è di fronte ad $\mathsf{a}_n{}^r$, ultima stringa di A^r, si aumenta la lunghezza passando ad $\mathsf{a}_1{}^{r+1}$, prima stringa di A^{r+1}, le si associa `++m` e si passa a [2.2].

La corrispondenza biunivoca che si stabilisce in tal modo, oltre ad associare 0 alla $\sqcup$, alla stringa $\mathsf{a}_{j_1}...\mathsf{a}_{j_r}$ associa

$$m \ = \ \mathsf{sopi}_{n,r-1} + j_1\, n^{r-1} + j_2\, n^{r-2} + j_3\, n^{r-3}\, 2 + \cdots + j_{r-1}\, n + j_r \ . \tag{1}$$

La stringa di A^* che in tal modo si associa biunivocamente a un numero naturale `m` si dice che lo rappresenta, fedelmente, nel **sistema di numerazione n-adico**.

Si osserva che se $\mathsf{A} = \{\mathsf{I}\}$ questa formula alla stringa "I^k" associa l'intero naturale k, in accordo con la rappresentazione unadica degli interi naturali.

C10d.12 Definiamo **derivata da sinistra rispetto a una stringa** w o **erosione da sinistra** di w dal linguaggio L il linguaggio

$$w \setminus\!\setminus \mathsf{L} \ := \ \{z \in \mathsf{A}^* \,\mathbb{[}\, w \cdot z \in \mathsf{L}\} \ .$$

Definiamo anche la **funzione o piccolo del linguaggio** L come

$$\mathbf{o}(\mathsf{L}) \ := \ \begin{cases} \sqcup & \text{sse } \sqcup \in \mathsf{L} \ , \\ \emptyset & \text{altrimenti} \ . \end{cases}$$

Per la precedente operazione e per la precedente funzione si trovano le proprietà che seguono.

(1) Prop.: $\forall v, w \in \mathsf{A}^* \ : \ (v \cdot w) \setminus\!\setminus \mathsf{L} \ = \ w \setminus\!\setminus (v \setminus\!\setminus \mathsf{L})$ ∎

(2) Prop.: $\forall \mathsf{a}_{j_1} \cdot ... \cdot \mathsf{a}_{j_k} \in \mathsf{A}^+ \ : \ (\mathsf{a}_{j_1}\, \mathsf{a}_{j_2} \,...\, \mathsf{a}_{j_k}) \setminus\!\setminus \mathsf{L} \ = \ \mathsf{a}_{j_k} \setminus\!\setminus (...(\mathsf{a}_{j_2} \setminus\!\setminus (\mathsf{a}_{j_1} \setminus\!\setminus \mathsf{L}))...)$ ∎

(3) Prop.: $w \in \mathsf{L} \iff \mathbf{o}(w \setminus\!\setminus \mathsf{L}) = \sqcup$ ∎

C10d.13 Si definisce **derivata da destra rispetto a una stringa** z o **erosione da destra** di z dal linguaggio L

$$\mathsf{L} /\!\!/ z \ := \ \{w \in \mathsf{A}^* \,\mathbb{[}\, w \cdot z \in \mathsf{L}\} \ .$$

Questa operazione è la duale-LR della derivazione da sinistra; infatti si ha

$$\forall z \in \mathsf{A}^*, \mathsf{L} \subseteq \mathsf{A}^* \ : \ \mathsf{L} /\!\!/ z \ = \ (z^{\leftarrow} \setminus\!\setminus \mathsf{L}^{\leftarrow})^{\leftarrow}.$$

Valgono quindi le uguaglianza duali per riflessione di quelle mostrate in `d12` .

(1) Prop.: $\forall v, w \in \mathsf{A}^* \ : \ \mathsf{L} /\!\!/ (v \cdot w) \ = \ (\mathsf{L} /\!\!/ w) /\!\!/ v$ ∎

(2) Prop.: $\forall \mathsf{a}_{j_1}, ..., \mathsf{a}_{j_k} \in \mathsf{A} \ : \ \mathsf{L} /\!\!/ (\mathsf{a}_{j_1}\, \mathsf{a}_{j_2} \dots \mathsf{a}_{j_k}) \ = \ ((...(\mathsf{L} /\!\!/ \mathsf{a}_{j_k}) /\!\!/ ... /\!\!/ \mathsf{a}_{j_2}) /\!\!/ \mathsf{a}_{j_1})$ ∎

(3) Prop.: $w \in \mathsf{L} \iff \mathbf{o}(\mathsf{L} /\!\!/ w) = \sqcup$ ∎

C10d.14 Comprensibilmente risultano utili anche le estensioni booleane delle derivazioni rispetto a stringhe.

Si dice **derivata da sinistra rispetto a un linguaggio** X di L :

$$\mathsf{X} \setminus\!\!\setminus \mathsf{L} \ := \bigcup_{x \in \mathsf{X}} x \setminus\!\!\setminus \mathsf{L} \ . \tag{1}$$

Si dice **derivata da destra rispetto a un linguaggio** Y di L :

$$\mathsf{L} /\!\!/ \mathsf{Y} \ := \bigcup_{y \in \mathsf{Y}} \mathsf{L} /\!\!/ y \ . \tag{2}$$

(3) Prop.: $\mathsf{X} \setminus\!\!\setminus \mathsf{L} \ = \ \{w \in \mathsf{A}^* \ \left[\!\right] \ \mathsf{X}\, w \cap \mathsf{L} \neq \emptyset\} \ ; \ \mathsf{L} /\!\!/ \mathsf{Y} \ = \ \{w \in \mathsf{A}^* \ \left[\!\right] \ w\, \mathsf{Y} \cap \mathsf{L} \neq \emptyset\}$ ∎

(4) Prop.: $\mathsf{L} /\!\!/ \mathsf{Y} \ = \ ((\mathsf{Y}^{\leftarrow}) \setminus\!\!\setminus (\mathsf{L}^{\leftarrow}))^{\leftarrow} \qquad \mathsf{X} \setminus\!\!\setminus \mathsf{L} = ((\mathsf{L}^{\leftarrow}) /\!\!/ (\mathsf{X}^{\leftarrow}))^{\leftarrow}$ ∎

(5) Prop.: $\mathbf{o}(\mathsf{X} \setminus\!\!\setminus \mathsf{L}) = \sqcup \iff \mathbf{o}(\mathsf{L} \setminus\!\!\setminus \mathsf{X}) = \sqcup \iff \mathsf{X} \cap \mathsf{L} \neq \emptyset \iff \mathbf{o}(\mathsf{L} /\!\!/ \mathsf{X}) = \sqcup \iff \mathbf{o}(\mathsf{X} /\!\!/ \mathsf{L}) = \sqcup$ ∎

(6) Prop.: $\mathsf{X} \setminus\!\!\setminus \mathsf{L} \neq \emptyset \iff \mathsf{X}\mathsf{A}^* \cap \mathsf{L} \neq \emptyset \qquad , \qquad \mathsf{L} /\!\!/ \mathsf{Y} \neq \emptyset \iff \mathsf{L} \cap \mathsf{A}^* \mathsf{Y} \neq \emptyset$ ∎

(7) Prop.: $\mathsf{X} \setminus\!\!\setminus \mathsf{L}\,\mathsf{M} \ = \ (\mathsf{X} \setminus\!\!\setminus \mathsf{L})\,\mathsf{M} + (\mathsf{L} \setminus\!\!\setminus \mathsf{X}) \setminus\!\!\setminus \mathsf{M} \qquad , \qquad (\mathsf{L}\,\mathsf{M}) /\!\!/ \mathsf{Y} \ = \ \mathsf{L}\,(\mathsf{M} /\!\!/ \mathsf{Y}) + \mathsf{L} /\!\!/ (\mathsf{Y} /\!\!/ \mathsf{M})$ ∎

C10d.15 Con le operazioni di derivazione si possono esprimere prefissi, suffissi e infissi.

Per chiarire questi collegamenti è comodo servirsi delle scritture abbreviate

$$w /\!\!/ s \ := \ w /\!\!/ \mathsf{A}^s \qquad \text{e} \qquad s \setminus\!\!\setminus w \ := \ \mathsf{A}^s \setminus\!\!\setminus w \quad , \quad \text{per} \ \ s \in \mathbb{N} \ .$$

Inoltre consideriamo $r \in \mathbb{P}$ e $w \ =_{\mathsf{A}} \ \mathsf{a}_{j_1} \dots \mathsf{a}_{j_r}$.

(1) Prop.:

$$\forall s \in [\mathbf{r}] \ : \ w /\!\!/ s = \mathsf{a}_{j_1} \dots \mathsf{a}_{j_{r-s}} \ , \ s \setminus\!\!\setminus w = \mathsf{a}_{j_{s+1}} \dots \mathsf{a}_{j_r} \ , \ w = (w /\!\!/ s) \cdot ((r-s) \setminus\!\!\setminus w)$$ ∎

(2) Prop.:

$$\forall \ s \in [\mathbf{r}] \ , \ t \in [\mathbf{r} - \mathbf{s}] \ : \ (s \setminus\!\!\setminus w) /\!\!/ t \ = \ s \setminus\!\!\setminus (w /\!\!/ t) \ = \ \mathsf{a}_{j_{s+1}} \dots \mathsf{a}_{j_{r-t}}$$ ∎

(3) Prop.:

$$\forall \ s \in [\mathbf{r}] \ : \ s \setminus\!\!\setminus w = \sqcup \iff w /\!\!/ s = \sqcup \iff \mathsf{len}(w) = s$$ ∎

(4) Prop.:

$$\forall \ s \in [\mathbf{r}] \ : \ s \setminus\!\!\setminus w = \emptyset \iff w /\!\!/ s = \emptyset \iff \mathsf{len}(w) < s$$ ∎

Possono essere utili anche le estensioni booleane delle abbreviazioni precedenti

$$s \setminus\!\!\setminus \mathsf{L} := \mathsf{A}^s \setminus\!\!\setminus \mathsf{L} \qquad \text{e} \qquad \mathsf{L} /\!\!/ s := \mathsf{L} /\!\!/ \mathsf{A}^s \ .$$

(5) Prop.: Consideriamo una stringa w e il relativo alfabeto minimo A.

L'insieme dei prefissi di w è $\mathbf{Pfx}(w) \ = \ w /\!\!/ \mathsf{A}^*$;
l'insieme dei suffissi di w è $\mathbf{Sfx}(w) \ = \ \mathsf{A}^* \setminus\!\!\setminus w$;
l'insieme degli infissi di w è $\mathbf{Ifx}(w) \ = \ \mathsf{A}^* \setminus\!\!\setminus (w /\!\!/ \mathsf{A}^*) \ = \ (\mathsf{A}^* \setminus\!\!\setminus w) /\!\!/ \mathsf{A}^*$ ∎

(6) Prop.: Consideriamo un linguaggio L e il relativo alfabeto minimo A.

L'insieme dei prefissi di L è $\mathsf{L} /\!\!/ \mathsf{A}^*$;
l'insieme dei suffissi di L è $\mathsf{A}^* \setminus\!\!\setminus \mathsf{L}$;
l'insieme degli infissi di L è $\mathsf{A}^* \setminus\!\!\setminus (\mathsf{L} /\!\!/ \mathsf{A}^*) \ = \ (\mathsf{A}^* \setminus\!\!\setminus \mathsf{L}) /\!\!/ \mathsf{A}^*$ ∎

L'ultima uguaglianza consente di introdurre la semplificazione

$$\mathsf{A}^* \setminus\!\setminus \mathsf{L} /\!/ \mathsf{A}^* \;:=\; \mathsf{A}^* \setminus\!\setminus \mathsf{L} /\!/ \mathsf{A}^*) \;=\; (\mathsf{A}^* \setminus\!\setminus \mathsf{L} /\!/ \mathsf{A}^* \,.$$

C10d.16 **(1) Eserc.** Mostrare che il numero degli elementi di $\succ_{\mathbf{uI}\ R} w$ soddisfa le relazioni

$$|\mathsf{A}| + 1 \;\leq\; |\succ_{\mathbf{uI}\ R} w| \;\leq\; |\mathsf{A}| \cdot (w^{\vdash\dashv} + 1) \,.$$

(2) Eserc. Mostrare che, per L e M linguaggi qualsiasi:

$$\mathbf{o}(\mathsf{L}^+) = \mathbf{o}(\mathsf{L}) \quad , \quad \mathbf{o}(\mathsf{L}^*) = \sqcup \quad , \quad o(\mathsf{L} + \mathsf{M}) = \mathbf{o}(\mathsf{L}) + \mathbf{o}(\mathsf{M}) \quad , \quad \mathbf{o}(\mathsf{L} \cdot \mathsf{M}) = \mathbf{o}(\mathsf{L}) \cdot \mathbf{o}(\mathsf{M}).$$

C10d.17 **(1) Eserc.** Dimostrare le seguenti proprietà della derivata da sinistra rispetto a linguaggi:

$$\mathsf{X} \setminus\!\setminus (\mathsf{L}\,\mathsf{M}) \;=\; (\mathsf{X} \setminus\!\setminus \mathsf{L}) \cdot \mathsf{M} + \mathsf{L} \cdot ((\mathsf{L} \setminus\!\setminus \mathsf{X}) \setminus\!\setminus \mathsf{M}) \qquad , \qquad (\mathsf{X}\,\mathsf{Y}) \setminus\!\setminus \mathsf{L} \;=\; \mathsf{Y} \setminus\!\setminus (\mathsf{X} \setminus\!\setminus \mathsf{L}) \qquad ,$$

$$\mathsf{X} \setminus\!\setminus (\mathsf{L} + \mathsf{M}) \;=\; \mathsf{X} \setminus\!\setminus \mathsf{L} + \mathsf{X} \setminus\!\setminus \mathsf{M} \qquad , \qquad (\mathsf{X} + \mathsf{Y}) \setminus\!\setminus \mathsf{L} = \mathsf{X} \setminus\!\setminus \mathsf{L} + \mathsf{Y} \setminus\!\setminus \mathsf{L} \qquad ,$$

$$\sqcup \setminus\!\setminus \mathsf{L} \;=\; \mathsf{L} \qquad , \qquad \mathsf{X} \setminus\!\setminus \sqcup \;=\; \mathbf{o}(\mathsf{X}) \qquad ,$$

$$\mathsf{X} \setminus\!\setminus \mathsf{L}^+ \;=\; ((\mathsf{L}^* \setminus\!\setminus \mathsf{X}) \setminus\!\setminus \mathsf{L}) \cdot \mathsf{L}^* \qquad , \qquad \mathsf{X} \setminus\!\setminus \mathsf{L}^* \;=\; \mathbf{o}(\mathsf{X}) + ((\mathsf{L}^* \setminus\!\setminus \mathsf{X}) \setminus\!\setminus \mathsf{L}) \cdot \mathsf{L}^* \qquad .$$

C10d.18 **(2) Eserc.** (a) Trovare le relazioni corrispondenti alle precedenti per la derivazione da destra.

(b) Constatare che $\forall\, \mathsf{X}, \mathsf{L}, \mathsf{Y} \in \mathbf{Lng} \;:\; \mathsf{X} \setminus\!\setminus (\mathsf{L} /\!/ \mathsf{Y}) = (\mathsf{X} \setminus\!\setminus \mathsf{L}) /\!/ \mathsf{Y}$ e conseguentemente giustificare l'abbreviazione $\mathsf{X} \setminus\!\setminus \mathsf{L} /\!/ \mathsf{Y} \;:=\; \mathsf{X} \setminus\!\setminus (\mathsf{L} /\!/ \mathsf{Y})$.

(c) Assunti $\mathsf{L} := a^2\, b^2 + (a\, b)^3 + (a\, b^2)^2$, $\mathsf{X} \;:=\; a\, b^*\, a$ e $\mathsf{Y} := a\, a^*\, b$, trovare una significativa espressione per il linguaggio $\mathsf{X} \setminus\!\setminus \mathsf{L} /\!/ \mathsf{Y}$.

C10 e. fattorizzazioni di stringhe

C10e.01 Una stringa $w \in \mathsf{A}^+$ si dice **stringa primitiva** o anche parola primitiva, sse non si può esprimere come potenza di un'altra stringa (ovviamente più corta), cioè sse non si trova alcuna $z \in \mathsf{A}^+$ per la quale sia $w = z^k$ per $k \in \{2, 3, 4, ...\}$.

Una stringa primitiva e una nonprimitiva sono, risp.:

$$a\,b\,b\,c\,a\,b\,c\,c\,a\,b\,b\,a\,c\,a\,c\,b\,a\,c \qquad \text{e} \qquad a\,c\,b\,a\,c\,b\,a\,c\,b\,a\,c\,b\,a\,c\,b = (a\,c\,b)^5 \ .$$

Come mostra questo esempio, a parità di lunghezza, le stringhe nonprimitive si possono giudicare più regolari delle primitive della stessa lunghezza.

Questa maggiore regolarità si può chiamare anche minore complessità.

Più operativamente si stabilisce che una stringa w va considerata poco complessa quando si trova una espressione E che consente di individuarla per la quale sia $|E| < |w|$.

Questo significa che le stringhe nonprimitive possono essere individuate fornendo meno informazioni di quante richiedono le primitive della stessa lunghezza. In altre parole le stringhe primitive sono meno abbreviabili delle nonprimitive.

Denotiamo con $\mathsf{Primw}_\mathsf{A}$ il linguaggio delle stringhe primitive su A. Per esempio

$$\begin{aligned}\mathsf{Primw}_{\{a,b\}} \ = \ \{&a, b, ab, ba, aab, aba, abb, baa, bab, bba,\\ &aaab, aaba, aabb, abaa, abba, abbb, baaa, baab, babb, bbab, bbaa, bbab, bbba, ...\}\end{aligned} \ .$$

Ovviamente per ogni alfabeto A l'insieme di stringhe $\mathsf{Primw}_\mathsf{A}$ è invariante per permutazione dei caratteri di A.

C10e.02 Vediamo alcuni fatti collegati alla non primitività delle stringhe.

(1) Prop.: Le stringhe di lunghezza 1 sono tutte primitive, ovviamente.
Le stringhe la cui lunghezza è un numero primo aut sono primitive aut sono potenze di una lettera ∎

(2) Prop.: Per ogni $w \in \mathsf{A}^+$ si può effettivamente individuare una sola parola primitiva z tale che $w \in z^+$.

Dim.: Se $w^{\vdash\dashv} = 1$ $w \in \mathsf{A}$. Se $w^{\vdash\dashv}$ è un numero primo si vede se è una potenza della sua prima lettera e in caso contrario è primitiva.

Consideriamo quindi una w la cui lunghezza $s := \mathsf{len}(w)$ abbia sottomultipli propri maggiori di 1.

Organizziamo l'esame dei successivi prefissi di w aventi come lunghezza uno dei suddetti sottomultipli procedendo per sottomultipli crescenti.

Per ciascuno di questi prefissi, che denotiamo con z, si controlla se soddisfa il requisito $z^{w^{\vdash\dashv}/z^{\vdash\dashv}} = w$.

Il primo prefisso z che soddisfa questo requisito è la stringa primitiva cercata: infatti ogni ulteriore prefisso y tale che $y^{|w|/|y|} = w$ deve essere una potenza di z; infatti

$$y^{|w|/|y|} \ = \ z^{|w|/|z|} \iff y^{|w|\,|z|} \ = \ z^{|w|\,|y|} \iff y = z^{|y|/|z|} \ .$$

Se non si trova alcun prefisso che soddisfa il requisito la w è primitiva ∎

C10e.03 Prop. Due stringhe diverse commutano sse sono potenze di una stessa stringa (primitiva).

Dim.: Consideriamo due stringhe $w, x \in \mathsf{A}^+$.
Se $w = z^n$ e $x = z^m$ per qualche stringa z è ovvio che sia $w\,x \ = \ z^{m+n} \ = \ x\,w$.

Quindi basta dimostrare $w\,x = x\,w \implies \mathsf{A}^+ \ni z \;⟦\; w, x \in z^+$, cosa che faremo costruttivamente con l'algoritmo che segue che permette di precisare come ottenere la z a partire da w e x.

Algoritmo: Non può essere $w^{\vdash\!\dashv} = x^{\vdash\!\dashv}$, in quanto la loro commutazione implica la loro uguaglianza, esclusa per ipotesi.
In caso contrario basta esaminare il caso $w^{\vdash\!\dashv} < x^{\vdash\!\dashv}$. È allora possibile scrivere $x = w^h\,v$ con $h \in \mathbb{P}$ e $0 \leq v^{\vdash\!\dashv}$. Se $v^{\vdash\!\dashv} = 0$ come stringa richiesta si può scegliere $z := w$ e $x = z^h$. Se $0 < v^{\vdash\!\dashv}$ si ha $w^{h+1}\,v = w^h\,v\,w$ e quindi $w\,v = v\,w$ con $v^{\vdash\!\dashv} < x^{\vdash\!\dashv}$.
Siamo quindi ricondotti a una ricerca del tipo di quella iniziale, ma limitata a coppie di stringhe aventi somma delle lunghezze inferiore.
Si può quindi procedere con successive fasi che ripetono le manovre precedenti con la w rimpiazzata da una stringa v via via più corta e dopo un numero finito di fasi si individua la z tale che $w, v \in z^+$ e quindi anche tale che $x = w^h\,v \in z^+$ ∎

C10e.04 **Prop.** Consideriamo due stringhe $w, x \in \mathsf{A}^+$.
Si trova una coppia di interi positivi $\langle m, n\rangle$ tali che sia $w^m = x^n \iff \mathsf{A}^+ \ni z$ tale che $w, x \in z^+$.

Dim.: "$\Longleftarrow$": è implicazione ovvia .

"$\Longrightarrow$": Se $m = n$ si assume $z := w = x$. Se $m = 1$ si assume $z := x$ e, dualmente-LR, se $n = 1$ si assume $z := w$.
Se m ed n hanno un divisore comune $k > 1$, si può scrivere $m = k\,p$ ed $n = k\,q$; in tal caso $w^m = x^n$ equivale a $w^p = x^q$.

Resta quindi da dimostrare $⟦\, w^m = x^n \implies \mathsf{A}^+ \ni z$ tale che $w, x \in z^+ \,⟧$ per m ed n mutuamente primi e, data l'equivalenza dei ruoli di w e x, possiamo limitarci a supporre $m < n$, ovvero $w^{\vdash\!\dashv} > x^{\vdash\!\dashv}$.

Si può scrivere $w = x^h\,y$ con $0 < y^{\vdash\!\dashv} < x^{\vdash\!\dashv}$ e $h \in \mathbb{P}$; dall'esame dell'inizio di w si ottiene $y\,x^n = x^n\,y$ e quindi, dato che $y^{\vdash\!\dashv} < x^{\vdash\!\dashv}$, dopo eventuali successive riduzioni dell'uguaglianza precedente, si ottiene $x\,y = y\,x$; per questa e utilizzando l'algoritmo in e03 per questa uguaglianza si può individuare la stringa z tale che $x, y \in z^+$ e $w = x^h\,y \in z^+$ ∎

C10e.05 Riprendiamo la nozione di occorrenza di carattere in una stringa, nozione che aiuta a precisare i rapporti tra una stringa e i suoi infissi, le sue eventuali regolarità e permette di definire due utili relazioni di equivalenza nei monoidi liberi.

Assumiamo ancora $\mathsf{A} = \{\mathtt{a}_1, ..., \mathtt{a}_n\}$ e Consideriamo la stringa $w =_{\mathsf{A}} w(1)...w(s) \in \mathsf{A}^s$.
Ogni coppia $\langle \mathtt{a}_j, i\rangle \in \mathsf{A} \times$ (s] si dice **occorrenza di un carattere** $\mathtt{a}_j$ nella stringa w sse $w(i) = \mathtt{a}_j$.

Più in generale ogni coppia $\langle u, i\rangle \in \mathsf{A}^{\leq s} \times$ (s] si dice **occorrenza dell'infisso** u nella w sse $w = x\,u\,y$ con $x^{\vdash\!\dashv} = i - 1$.

Denotiamo con $\mathrm{Occ}(u, w)$ l'insieme delle occorrenze di u nella w e denotiamo con $|w|_u$ o con $\mathrm{occ}(u, w)$ il numero delle occorrenze di u nella stringa w.

(1) Prop.: $\displaystyle\sum_{a_i \in \mathsf{A}} |w|_{a_i} = s = |w|$ ∎

(2) Prop.: $|w|_u > 0 \iff \mathrm{Occ}(u, w) \neq \emptyset \iff u \in \mathbf{Ifx}(w)$ ∎

Studiando le parole e i linguaggi su A può essere significativo anche l'alfabeto minimo di una parola w, i.e. l'insieme delle lettere che occorrono in w, ossia l'insieme delle $\mathtt{a}_i \in \mathsf{A} \;⟦\; |w|_{\mathtt{a}_i} > 0$; tale alfabeto lo denotiamo con w^{minalf} .
Palesemente $w^{\mathrm{minalf}} = \mathbf{Ifx}(w) \cap \mathsf{A}$.

C10e.06 Assegnamo ad A un ordine totale $<$, cioè poniamo in sequenza i caratteri che costituiscono questo alfabeto; più precisamente supponiamo che sia $\mathtt{a}_1 < \mathtt{a}_2 < \cdots < \mathtt{a}_n$.

Diciamo **vettore di Parikh della stringa** w relativo ad A e $\leq$ la sequenza di lunghezza $n = |\mathsf{A}|$ che segue

$$\mathcal{P}rk(w) \;=\; w^{\mathcal{P}rk} \;:=\; \big\langle\, , |w|_{\mathtt{a}_1}, |w|_{\mathtt{a}_2}, ..., |w|_{\mathtt{a}_n} \big\rangle \quad .$$

Per esempio relativamente all'alfabeto $\{a, b, c, d\}$ ordinato nel modo usuale risulta:

$$(accdadaca)^{\mathcal{P}rk} = \langle 4, 0, 3, 2\rangle.$$

La **funzione di Parikh** "$\mathcal{P}rk$" associa a ogni stringa di un A^* un elemento di $\mathbb{N}^{|\mathsf{A}|}$, è suriettiva e noniniettiva sse $|\mathsf{A}| > 1$.
In particolare è evidente che per ogni stringa (anche nonpalindroma) w si ha $\mathcal{P}rk(w^{\leftarrow}) = \mathcal{P}rk(w)$.

La relativa equivalenza, che definiamo $\sim_{Prk} := \mathcal{P}rk \circ_{lr} (\mathcal{P}rk^{-\mathbf{1}})$, è costituita dalle coppie di stringhe di A^* ottenibili l'una dall'altra permutando le occorrenze di carattere che le compongono.

Per esempio $(accdadaca)^{\mathcal{P}rk} = (a^4c^2dcd)^{\mathcal{P}rk} = (c^2adca^2da)^{\mathcal{P}rk} = \langle 4, 0, 3, 2\rangle$ e quindi $accdadaca \;\sim_{Prk}\; a^4c^2dcd \;\sim_{Prk}\; c^2adca^2da$.

Queste classi corrispondono ai multiinsiemi basati sull'insieme ordinato A; esse quindi sono in biiezione con permutazioni con ripetizioni di A [B13e17] e i loro cardinali sono dati da coefficienti multinomiali. Per esempio il numero di stringhe equivalenti ad "*essere*" è $6!/(3!\,2!\,1!) = 720/(6 \cdot 2) = 60$, mentre la sequenza in ordine lessicografico delle stringhe dell'insieme $essere_{\,\natural\,} \sim_{Prk}$ comprende:

$$\begin{gathered} eeerss\ eeesrs\ eeerss\ eeress\ eerses\ eersse\ ereess\ ereses\ eresse\ ersees\ ersese\ erssee \\ \cdots\cdots \\ \cdots\cdots \\ \cdots\cdots \\ seseer\ sesere\ sesree\ sreees\ sreese\ sresee\ srseee\ ssreee\ sseree\ sseere\ sseree\ ssreee\ . \end{gathered}$$

C10e.07 Le classi dell'equivalenza $\sim_{Prk}$ sono interamente contenute negli insiemi A^s di stringhe aventi la stessa lunghezza; in altri termini la partizione di A^* relativa a $\sim_{Prk}$ è un raffinamento di quella relativa alla uguagkianza delle lunghezza.

In genere è utile individuare le classi di una equivalenza attraverso loro elementi rappresentativi facilmente individuabili. Un atteggiamento di questo tipo spesso conveniente consiste nell'assumere come rappresentativo di ciascuna classe l'elemento minimo secondo un opportuno ordine totale dell'insieme ambiente.

Come elementi rappresentativi delle classi di $\sim_{Prk}$, come per i rappresentativi di altre equivalenze entro insiemi $\mathsf{A}^* = \{\mathtt{a}_1, \mathtt{a}_2, ..., \mathtt{a}_n\}^*$, conviene assumere le stringhe minime secondo l'ordine lessicografico $\leq_{lxg}$: la generica classe $w_{\,\natural\,} \sim_{Prk}$ viene dunque rappresentata da

$$\mathtt{a}_1{}^{|w|_{\mathtt{a}_1}} \,\shortmid\, \mathtt{a}_2{}^{|w|_{\mathtt{a}_2}} \,\shortmid\, \ldots \,\shortmid\, \mathtt{a}_n{}^{|w|_{\mathtt{a}_n}} \quad .$$

Dal punto di vista dell'insieme dei vettori di Parikh $\mathbb{N}^{\times n}$ si osserva che

$$\forall w, x \in \mathsf{A}^* \;:\; (w \shortmid x)^{\mathcal{P}rk} = w^{\mathcal{P}rk} \boldsymbol{+} x^{\mathcal{P}rk} \;,$$

dove con $\boldsymbol{+}$ denotiamo la somma termine a termine delle n-uple di $\mathbb{N}^{\times n}$; la funzione di Parikh quindi è un epimorfismo di monoidi da $\big\langle \mathsf{A}^*, \cdot, \sqcup \big\rangle$ su $\big\langle \mathbb{N}^n, \boldsymbol{+}, \mathbf{0}_n \big\rangle$.

Anche della funzione di Parikh risulta utile la estensione booleana; essa è un morfismo tra algebre di Kleene e c'è da aspettarsi che gli insiemi di sequenze di interi naturali ottenibili dalla applicazione di

questa funzione del genere $\lceil\, \mathbf{Lng}_{\mathsf{A}} \longmapsto \mathbb{N}^{\times|\mathsf{A}|} \,\rceil$ ad alcuni linguaggi forniscano caratteristiche interessanti dei linguaggi stessi.

C10e.08 Una proprietà che contribuisce alla regolarità delle stringhe, più debole dell'essere esprimibile come potenza di una stringa più corta [e01, ..., e04] riguarda la possibilità di trovare in esse occorrenze ripetute.

Si dice **coppia sovrapposta** sull'alfabeto A una stringa su A che può scriversi $w = v\,y = x\,v$ per qualche $v, x, y \in \mathsf{A}^+$.
Un esempio di coppia sovrapposta è ABRACADABRA ; per essa $v =$ ABRA.

Si dice invece **stringa con overlap** su A una stringa della forma $v\,u\,v\,u\,v$ con $v \in \mathsf{A}^*$ ed $u \in \mathsf{A}^+$; l'infisso ripetuto $v\,u\,v$ si dice **overlap della stringa** $vuvuv$.

Una stringa con overlap fornisce un caso particolare di coppia sovrapposta: infatti si può scrivere $v\,u\,v\,u\,v = w\,y = z\,w$ con $w = v\,u\,v$, $y = u\,v$ e $z = v\,u$.

Si osserva che le stringhe che sono quadrati di stringa, cioè che hanno la forma $u\,u$, sono particolari stringhe con overlap ($v = \sqcup$); tali sono anche tutte le stringhe potenze k-esime di stringhe con $k = 3, 4, ...$, cioè stringhe della forma u^k; per una tale stringa vi sono $k-1$ scelte dell'overlap: u, u^2,..., u^{k-1} .

C10e.09 **(1) Prop.**: Ogni coppia sovrapposta $w = v\,y = x\,v$ sull'alfabeto A presenta un overlap come prefisso e un overlap come suffisso.

Dim.: Possiamo limitarci al caso in cui sia $x^{\vdash} < v^{\vdash}$, dopo aver osservato che il caso $x^{\vdash} = v^{\vdash}$ si riduce a $x = v$ e dopo aver segnalato la possibilità di trattare il caso $x^{\vdash} > v^{\vdash}$ per dualità-LR.

Poniamo $z := x \setminus\!\setminus v$, in modo che sia $w = x\,z\,y = x\,v = v\,y$ e osserviamo che $x^{\vdash} < v^{\vdash}$ implica $|z| > 0$.
Denotiamo con a il primo carattere di v, osserviamo che a è anche il carattere iniziale di z; quindi introduciamo $z_1 := a \setminus\!\setminus z$ e osserviamo che z_1 è suffisso di z, v e w.

Posto $t := v /\!\!/ z_1$ si ha $w = a\,t\,a\,t\,a\,z_1$ e quindi si trova una stringa con overlap che è prefisso di t.
La dimostrazione della presenza di un overlap suffisso si ottiene per dualità-LR, ossia procedendo come sopra, ma servendosi delle stringhe riflesse di quelle usate in precedenza ∎

C10e.10 Consideriamo una parola $w \in \mathsf{A}^s$ che contiene due occorrenze di $v \in \mathsf{A}^r$ con $0 < 2\,r \leq s$ che scriviamo $\langle v, i\rangle$ e $\langle v, i'\rangle$; scriviamo inoltre $w = x\,v\,y = x'\,v\,y'$ con $x^{\vdash} = i - 1$ e $x'^{\vdash} = i' - 1$.

Si danno tre possibilità per le occorrenze di v:
- si hanno **occorrenze sconnesse** sse $i + r < i'$, cioè sse si può scrivere $w = x\,v\,z\,v\,y'$ con $z \in \mathsf{A}^+$;
- si hanno **occorrenze adiacenti** sse $i + r = i'$, cioè sse si può scrivere $w = x\,v\,v\,y'$;
- si hanno **occorrenze sovrapposte** sse $i + r > i'$, cioè sse si può scrivere $w = x\,v'\,y'$ con v' coppia sovrapposta.

Si può stabilire che la presenza di occorrenze ripetute in una parola contribuisca alla sua regolarità e che questa sia accentuata se nella parola sono presenti occorrenze adiacenti, cioè fattori esprimibili come quadrati o come potenze superiori a due.

(1) Prop.: Una stringa w ha come infisso una coppia sovrapposta $\iff$ w ha come infisso un overlap.

Dim.: "$\Longrightarrow$": Dalla e04 segue che la presenza di una coppia sovrapposta implica la presenza di un overlap .
"$\Longleftarrow$": Come si è detto in e08(1), la presenza di un overlap in una parola costituisce un caso particolare di presenza di una coppia sovrapposta ∎

C10e.11 Due stringhe w ed x si dicono **stringhe ciclicamente coniugate** sse una di esse si può ottenere permutando ciclicamente l'altra; per segnalare questa situazione scriviamo $w \sim_{\mathsf{C}} x$.

Chiaramente $w \sim_{\mathsf{C}} x$ sse w ed x si possono fattorizzare, risp., come $w = u\,v$ e $x = v\,u$ sse in A^+ si trova una stringa u tale che $w\,u = u\,x$.

La partizione di A^* criptomorfa all'equivalenza $\sim_{\mathsf{C}}$ è un raffinamento della partizione criptomorfa all'equivalenza $\sim_{Prk}$: infatti $\sim_{Prk}$ aggrega stringhe ottenibili attraverso permutazioni qualsiasi dei caratteri componenti, mentre $\sim_{\mathsf{C}}$ aggrega a una stringa solo quelle ottenibili attraverso permutazioni cicliche. Per esempio $a\,b\,c \sim_{Prk} a\,c\,b$, mentre $a\,b\,c \not\sim_{\mathsf{C}} a\,c\,b$.

Constatiamo che le classi di $\sim_{\mathsf{C}}$ contenute in $\{a,b\}^3$ sono $\{a^3\}$, $\{b^3\}$, $\{aab, aba, baa\}$ e $\{abb, bab, bba\}$ e che esse coincidono con le classi di $\sim_{Prk}$.

Le classi dell'equivalenza $\sim_{\mathsf{C}}$ contenute in $\{a,b\}^4$ sono $\{a^4\}$, $\{b^4\}$, $\{aaab, aaba, abaa, baaa\}$, $\{aabb, abba, bbaa, baab\}$, $\{abbb, babb, bbab, bbba\}$ e $\{abab, baba\}$; ora il loro insieme non coincide con l'insieme delle classi relative a $\sim \mathcal{P}rk$ dal quale si ottengono sostituendo la quarta e la sesta classe con la loro unione $\{aabb, abba, bbaa, baab, abab, baba\}$.

Sia le classi di $\sim_{\mathsf{C}}$ che le classi di $\sim_{Prk}$ sono opportunamente rappresentate dalle rispettive stringhe minime secondo un ordine lessicografico.
Nell'ambito di $\{a,b\}^4$ questo sistema di rappresentativi per $\sim_{\mathsf{C}}$ è $\{a^4,\ aaab,\ aabb,\ abab,\ aaab, b^4\}$, mentre per $\sim_{Prk}$ è $\{a^4,\ aaab,\ aabb,\ abbb, b^4\}$.

Può essere chiarificante scrivere

$$w \sim_{\mathsf{C}} x \implies \mathcal{P}rk(w) = \mathcal{P}rk(x)\ , \quad \text{mentre} \quad \mathcal{P}rk(w) = \mathcal{P}rk(x) \not\Longrightarrow w \sim_{\mathsf{C}} x\ .$$

C10e.12 Si osserva che se una stringa w è potenza di una stringa più corta, cioè se $w = z^q$ con $q > 1$, ogni stringa ottenuta permutando ciclicamente w è potenza q-esima di una permutazione ciclica di z.

Quindi tra le classi di $\sim_{\mathsf{C}}$ si distinguono quelle costituite solo da stringhe primitive dalle classi formate da stringhe che sono tutte potenze proprie di stringhe più corte.

Volendo esprimere le varie stringhe di A^* come prodotti di stringhe più corte, le stringhe primitive risultano più interessanti delle nonprimitive in quanto consentono maggiore concisione.
Vedremo nella sezione che segue che tra le primitive risultano particolarmente utili le minime lessicografiche delle classi di $\sim_{\mathsf{C}}$.

C10 f. parole di Lyndon

C10f.01 Si dice **parola di Lyndon** sull'alfabeto A munito di un ordine totale $<$ una stringa di A^+ che è primitiva ed è la minima secondo l'ordine lessicografico nella propria classe di $\sim_{\mathsf{c}}$, classe di stringhe ciclicamente coniugate, ossia inferiore-lxg alle sue rotazioni.

Denotiamo con $\mathsf{Lynd}_{\mathsf{A},<}$ il linguaggio delle parole di Lyndon per $\langle \mathsf{A}, < \rangle$. Quando $<$ si può considerare implicito, per esempio se A è costituito dalle prime lettere di un alfabeto ben riconosciuto, usiamo la scrittura semplificata $\mathsf{Lynd}_{\mathsf{A}}$.

In particolare possiamo scrivere:

$$\mathsf{Lynd}_{\{a,b\}} = \{a,\ b,\ ab,\ aab,\ abb,\ aaab,\ aabb,\ abbb,\ aaaaab,\ aaabb,\ aabab,\ aabbb,\ ababb,\ abbbb,\ aaaaab, ...\}$$

$$\begin{aligned}\mathsf{Lynd}_{\{a,b,c\}} = \{&a,b,c,\ ab,ac,bc,\ aab,aac,abb,abc,acc,bbc,bcc,\\ &aaab,aaac,aabb,aabc,aacb,aacc,abac,abbb,abbc,abcb,abcc,acbb,acbc,accc,\\ &aaaab,aaabb,aaabc,aaacb,aaacc,aabab,aabac,aabbb,aabbc,aabcb,aabcc,\\ &aacab,aacac,aacbb,aacbc,aaccb,aaccc,ababb,ababc,abacc,abbab,abbac,\\ &abbbb,abbbc,abbcb,abbcc,abcac,abcbb,abcbc,abccb,abccc,acacb,acacc,\\ &acbbb,acbbc,acbcc,accbb,accbc,acccb,acccc,\\ &bbbbc,bbbcc,bbcbc,bbccc,bcbcc,bcccc,\quad aaaaab,...\}\ .\end{aligned}$$

C10f.02 Sono evidenti gli enunciati che seguono riguardanti alfabeti A e B.

(1) Prop.: (a) $\mathsf{Lynd}_{\mathsf{I}} = \{\mathsf{I}\}$.

(b) $\mathsf{A} \subset \mathsf{B} \implies \mathsf{Lynd}_{\mathsf{A}} \subset \mathsf{Lynd}_{\mathsf{B}}$.

(c) $|\mathsf{Lynd}_{\mathsf{A}}| = \aleph_0$.

(d) $\mathsf{Lynd}_{\{a,b\}} \supset \{b\} \dot\cup a^+ b^+$.

(e) $\mathsf{A} = \{a_1 < a_2 < \cdots < a_r\} \implies a_1{}^+ a_2{}^+ \cdots a_r{}^+ \subset \mathsf{Lynd}_{\mathsf{A}}$.

(f) $\mathsf{Lynd}_{\mathsf{A}} = \mathsf{A} \ \dot\cup\ \mathsf{Lynd}_{\mathsf{A}} \cap \mathsf{A}\,\mathsf{A}^+$ ∎

C10f.03 **(1) Prop.**: Una stringa $w \in \mathsf{A}\,\mathsf{A}^+ = \mathsf{A}^{\geq 2}$ appartiene a $\mathsf{Lynd}_{\mathsf{A}}$ sse precede strettamente secondo l'ordine lessicografico ogni suo suffisso proprio, ovvero

$$\mathsf{Lynd}_{\mathsf{A}} = \left\{ w \in \mathsf{A}^+ \ \middle|\ \forall x \in w^{\mathsf{Sfx}} \cap \mathsf{A}^+ \ :\ w \leq_{lxg} x \right\} .$$

Dim.: Introduciamo l'insieme delle stringhe su A che precedono secondo $<_{lxg}$ ogni loro suffisso proprio, ovvero poniamo

$$\mathsf{L} := \left\{ w \in \mathsf{A}^+ \ \middle|\ \forall x \in w^{\mathsf{Sfx}} \cap \mathsf{A}^+ \ :\ w \leq_{lxg} x \right\} .$$

Osserviamo che la proprietà vale per le lettere di $\mathsf{A} \subset \mathsf{Lynd}_{\mathsf{A}}$ in quanto non posseggono suffissi propri; per le stringhe di lunghezza maggiore o uguale a 2 procediamo a dimostrare due relazioni di inclusioni opposte.

$\lceil \mathsf{Lynd}_{\mathsf{A}} \subseteq \mathsf{L} \rceil$: Consideriamo $w \in \mathsf{L}$, la stringa $x \in w^{\mathsf{Sfx}} \cap \mathsf{A}^+$ tale che sia $w <_{lxg} x$ ed introduciamo $u := w \mathbin{/\!\!/} x$, ossia $u \in \mathsf{A}^+$ tale che $w = u\,x$.

Procediamo a dimostrare, per assurdo, che $x \notin w^{\mathsf{Pfx}}$. Se fosse $w = x\,t$ con $t \in \mathsf{A}^+$ sarebbe $x\,t = u\,x$ e quindi, grazie a e08, si trovano $y, z \in \mathsf{A}^*$ ed $i \in \mathbb{N}$ tali che si avrebbe

$$u = y\,z \ \ ,\ \ t = z\,y \ \ ,\ \ (y\,z)^i\,y \quad \text{e quindi} \qquad w = (y\,z)^{i+1}\,y\ .$$

La parola $w \in \mathsf{Primw}$ sarebbe inferiore per $<_{lxg}$ di ogni sua coniugata ciclica, cioè sarebbe

$$(y\,z)^{i+1}\,y \;<_{lxg}\; y\,(y\,z)^{i+1} \quad \text{e quindi} \quad ,(z\,y)^{i+1} \;<_{lxg}\; (y\,z)^{i+1}\ .$$

Giustapponendo la y a destra ai due membri si avrebbe $(z\,y)^{i+1}\,y <_{lxg} (y\,z)^{i+1}\,y = w$, contraddicendo la precedenza lessicografica di ciascuna delle parole di Lyndon rispetto alle sue coniugate cicliche. Dunque ogni iparola di Lyndon w appartiene ad L.

$\lceil\mathsf{L} \subseteq \mathsf{Lynd_A}\rceil$: Consideriamo $w \in \mathsf{L} \cap \mathsf{A}\,\mathsf{A}^+$ e le stringhe $u, v \in \mathsf{A}^+$ tali che $u\,v = w$ e si abbia la disuguaglianza $w <_{lxg} v$; ovviamente $w <_{lxg} v\,u$ e quindi $w \in \mathsf{Lynd_A}$ ∎

C10f.04 Dato che $\mathsf{A} \subset \mathsf{Lynd_A}$, evidentemente ogni stringa su A si può esprimere come prodotto di parole di Lyndon. Le fattorizzazioni che si servono di fattori di Lyndon sono dette **fattorizzazioni di Lyndon**; esse godono di proprietà notevoli.

Cominciamo con l'osservare esplicitamente che una parola di Lyndon può presentare diverse fattorizzazioni di Lyndon. Due esempi:

$$aabb \;=\; aab\,{}_{\shortmid}\,b \;=\; a\,{}_{\shortmid}\,abb \qquad , \qquad ababb \;=\; ab\,{}_{\shortmid}\,abb \;=\; a\,{}_{\shortmid}\,b\,{}_{\shortmid}\,abb$$

(1) Prop.: $l, m \in \mathsf{Lynd_A} \;\wedge\; l <_{lxg} m \;\Longrightarrow\; l\,m \in \mathsf{Lynd}$.

Dim.: Chiaramente $l\,m <_{lxg} m$; se $l <_{\mathsf{pfx}} m$ scrivendo $m =: l\,m'$ abbiamo $m <_{lxg} m'$ e di conseguenza $l\,m \leq_{lxg} l\,m' = m$; se invece $l \notin m\,\mathsf{A}^*$, essendo $l <_{\mathsf{pfx}} m$, si ha ancora $l\,m \leq_{lxg} l\,m' = m$.

Prendiamo ora una $v \in m^{\mathsf{Sfx}}$: dato che $m \in \mathsf{Lynd_A}$, grazie a f03(1), abbiamo $l\,m \leq_{lxg} m \leq_{lxg} v$. Introduciamo anche $v' \in l^{\mathsf{Sfx}}$: si ha $l \leq_{lxg} v'$ e quindi $l\,m \leq_{lxg} v'\,m$.
Dunque $l\,m$ precede lessicograficamente ogni suo suffisso proprio e, grazie a f03(1) $l\,m \in \mathsf{Lynd_A}$ ∎

(2) Prop.: Sia $w \in \mathsf{Lynd_A} \setminus \mathsf{A}$ e sia m la più lunga delle parole in $w^{\mathsf{Sfx}} \cap \mathsf{Lynd_A} \setminus \{w\}$, insieme che contiene almeno l'ultima lettera di w. Introdotto l tale che $w = l\,m$, allora $l \in \mathsf{Lynd_A}$ e $l <_{lxg} l\,m <_{lxg} m$.

Dim.: Se $l \in \mathsf{A}$ l'enunciato vale. In caso contrario scegliamo $v \in l^{\mathsf{Sfx}}$. Dato che $v\,m \notin \mathsf{Lynd_A}$, scegliamo $t \in (v\,m)^{\mathsf{Sfx}} \setminus \{v\,m\}$ tale che $t <_{lxg} v\,m$.
Allora se $v <_{lxg} t$, essendo $v <_{lxg} t <_{lxg} v\,m$ si deduce che esiste $s <_{lxg} m$ tale che $t = v\,s$ e questa s sarebbe un suffisso proprio di m lessicograficamente inferiore ad m, contro la $m \in \mathsf{Lynd_A}$; dunque $t \leq_{lxg} v$; di conseguenza $l <_{lxg} l\,m <_{lxg} t \leq_{lxg} v$, consentendo di affermare $l <_{lxg} v$ e, grazie a f03(1), $l \in \mathsf{Lynd_A}$.
Se invece $l <_{lxg} m$, dato che $w = l\,m \in \mathsf{Lynd_A}$, si ottiene $l\,m <_{lxg} m$ ∎

Gli enunciati (1) e (2) conducono ad un'altra caratterizzazione delle parole di Lyndon.

(3) Prop.: $w \in \mathsf{Lynd_A} \setminus \mathsf{A} \;\Longleftrightarrow\; w = l\,m$ con $l, m \in \mathsf{Lynd_A} \;\wedge\; l <_{lxg} m$ ∎

C10f.05 L'enunciato f04(1) suggerisce una procedura ricorsiva in grado di procedere illimitatamente nella generazione delle parole di un $\mathsf{Lynd_A}$ a partire dalle lettere dell'alfabeto A.

Questa procedura presenta una iterazione primaria (illimitata) che procede a generare le parole di Lyndon delle successive lunghezze $\ell = 2, 3, 4, ...$
In ogni stadio relativo ad una ℓ si ha una iterazione secondaria che riguarda coppie di parole già acquisite che presentano coppie di lunghezze $\langle h, k\rangle$ con $h = 1, ..., \ell - 1$ e $k = \ell - h$.
Una iterazione terziaria riguarda le coppie di parole $\langle l, m\rangle$ relative ad ogni attuale coppia $\langle h, k\rangle$ e per ciascuna di queste $\langle l, m\rangle$ verifica se $l <_{lxg} m$ ed in caso affermativo accoda $l\,m$ alla successione delle parole di Lyndon.

Descriviamo le prime azioni della procedura nel caso $\mathsf{A} = \{a, b\}$. Per $\ell = 2$ aggiunge alla lista delle lettere la sola $a\,{}_{\shortmid}\,b$. Per $\ell = 3$ emette $a\,{}_{\shortmid}\,ab$ ed $ab\,{}_{\shortmid}\,b$. Per $\ell = 4$ emette $a\,{}_{\shortmid}\,aab$, $a\,{}_{\shortmid}\,abb$, $aab\,{}_{\shortmid}\,b$ e $abb\,{}_{\shortmid}\,b$.

. . . . . .

Si vede come si presentano anche parole ripetute e risulta chiara la convenienza di eliminare al più presto le ripetizioni per evitare successive manovre inutili.

La coppia $\langle l, m\rangle \in \mathsf{Lynd_A}^{\times 2}$ tale che $w := l\,m \in \mathsf{Lynd_A}$ ed m di lunghezza massimale tra i suffissi di w che sono parole di Lyndon si dice **fattorizzazione-mls** della w (o fattorizzazione standard, o fattorizzazione con suffisso di lunghezza massimale). Denotiamo con $\mathsf{fmls}(w)$ questa coppia (evidentemente unica).

(1) Prop.: Consideriamo $w \in \mathsf{Lynd_A} \cap \mathsf{A}\,\mathsf{A}^+$ e scriviamo $\langle l, m\rangle := \mathsf{fmls}(w)$.
Allora $\forall x \in \mathsf{Lynd_A} \cap (w_{\not\sqsupset} >_{lxg}) \;:\; \langle w, x\rangle = \mathsf{fmls}(w\,x) \iff x <_{lxg} m$ ∎

C10f.06 **Teorema** (**teorema di Lyndon**)

Ogni stringa $w \in \mathsf{A}^+$ possiede una unica fattorizzazione mediante una sequenza di parole di Lyndon lessicograficamente noncrescenti.

Dim.: Come già osservato [f04], ogni $w \in \mathsf{A}^+$ possiede almeno una fattorizzazione di Lyndon; scegliamone una la cui lunghezza n sia minimale e scriviamola $w =: l_1\,l_2 \ldots l_n$.

Se vi fosse $i \in [n-1]$ tale che $l_i <_{lxg} l_{i+1}$, grazie a f04(1) si avrebbe $l_i\,l_{i+1} \in \mathsf{Lynd_A}$ e quindi la fattorizzazione non sarebbe minimale. Quindi la generica w possiede almeno una fattorizzazione di Lyndon nondecrescente-lxg.

Dimostriamo per assurdo l'unicità di questa fattorizzazione supponendo che esista un'altra fattorizzazione di w con gli stessi requisiti e scriviamo

$$l_1\,l_2 \ldots l_n \;=\; l'_1\,l'_2 \ldots l'_n$$
$$\text{ove } \forall i \in [1,n] \;:\; l_i, l'_i \in \mathsf{Lynd} \text{ e } \forall i \in [1,n-1] \;:\; l_i \geq_{lxg} l_{i+1} \;\wedge\; l'_i \geq_{lxg} l'_{i+1}\ .$$

Se vi fosse un $j \in [1,n-1] \;\nexists\; |l_j| > |l'_j|$, si avrebbe $l_j = l'_j \cdots l'_{j+h}\,u$ con $h \in \mathbb{N}$ e $u \in {l'_j}^{\mathsf{Sfx}}$.
Grazie a f03(1) si avrebbe allora $l_j <_{lxg} u \leq_{lxg} l'_{j+h+1} \leq_{lxg} l'_j \leq_{lxg} l_j$, un evidente assurdo.
Quindi $l_l = l'_j$ ∎

C10f.07 Anche la precedente dimostrazione suggerisce un algoritmo per fattorizzare una qualsiasi stringa mediante parole di Lyndon.

Un algoritmo più veloce è suggerito dal seguente fatto.

(1) Prop.: Sia $l_1\,l_2\,\cdots\,l_n$ una fattorizzazione della $w \in \mathsf{A}^+$ mediante una sequenza di parole di Lyndon noncrescente secondo $<_{lxg}$. Allora l_n è il suffisso di w minimo secondo $<_{lxg}$.

Dim.: Sia v il suffisso di w minimo-lxg e scriviamo $w = u\,v$; in forza di f04(1) si ha $v \in \mathsf{Lynd}$.
Se $u = \sqcup\!\sqcup$ l'asserto è dimostrato. Se no scriviamo s il suffisso di u minimo-lxg e scriviamo $w =: rsv$.
Non può essere $s <_{lxg} v$, in quanto, grazie a f04(1), sarebbe $s\,v \in \mathsf{Lynd}$ da cui $s\,v <_{lxg} v$, contro la definizione di v; dunque $s \geq_{lxg} v$ e proseguendo questo processo si giunge alla enunciata fattorizzazione di w ∎

C10f.08 Si osservi che da f07(1) segue direttamente f06.
Va anche segnalato che l'algoritmo suggerito da f07(1) non è il più veloce possibile.
Art M. Duval nel 1980 ha dimostrato che conviene procedere sulla w da fattorizzare non da destra verso sinistra, ma da sinistra a destra, in quanto in tal modo si ottiene la fattorizzazione con un numero di confronti tra lettere lineare nella lunghezza della stringa sottoposta.