Capitolo D50:
Stringhe, linguaggi formali e automi

Contenuti delle sezioni

D50:0.01 Questo è il primo dei capitoli dedicati ai linguaggi formali, alle macchine sequenziali (o automi). Queste entità verranno in seguito utilizzate per affrontare la computabilità, intesa come complesso delle questioni generali riguardanti le attività computazionali, e la complessità delle computazioni.

Il capitolo inizia, per motivi di autonomia e completezza, esponendo le nozioni basilari su stringhe e linguaggi, già esposte come elementi iniziali della esposizione in B10: e riprese successivamente quando servivano ad introdurre altre entità tendenzialmente semplici.

Successivamente vengono trattate con una certa sistematicità le relazioni che riguardano stringhe e linguaggi e le prime proprietà circa la fattorizzazione di queste entità, anche in vista delle successive trattazioni della teoria dei codici e della combinatorica delle parole.

D50:a. Nozioni iniziali su stringhe e linguaggi

D50:a.01 Accade spesso di studiare sequenze finite di oggetti, eventualmente ripetuti, occupandosi solo delle proprietà che derivano da come tali oggetti si dispongono nelle sequenze stesse. Esempi comuni riguardano le manipolazioni di elenchi di locuzioni e testi, naturali o artificiali; in questi casi gli oggetti delle sequenze sono abbastanza elementari. Si studiano anche sequenze di oggetti molto complessi, ad es. della sequenza dei processori costituenti un sistema multiprocessore e della sequenza dei residui di aminoacidi costituenti una proteina di medie dimensioni (10000 residui); in certe considerazioni ciascuno di questi oggetti complessi viene considerato solo mediante il simbolo che ne identifica la struttura interna e la posizione che occupa nella sequenza.

D50:a.02 Nell'ambito di questi studi gli oggetti che compongono le sequenze si possono quindi considerare "elementari" o "atomici": essi quindi vengono chiamati **caratteri**, **lettere** o **segni** ed un insieme finito di tali oggetti viene detto **alfabeto**; le loro sequenze finite sono dette **stringhe** o **parole [di lunghezza finita]**.

Esempi di alfabeti sono il singoletto contenente solo un segno come $\{|\}$; l'alfabeto dei bits $\{0, 1\}$; l'insieme delle cifre decimali; l'insieme delle lettere utilizzate in una lingua naturale (comprendente o

meno lettere accentate e distinguendo o meno fra minuscole e maiuscole); l'insieme dei caratteri di un sistema di codifica come l'ASCII o Unicode. Osserviamo che i caratteri ASCII si codificano con 7 o 8 bits e gli Unicode con 16 bits. Vi sono quindi considerazioni nelle quali i caratteri di uno di questi codici sono considerati elementari, altre in cui sono stringhe sull'alfabeto dei bits.

Introduciamo il termine **insieme delle lettere per alfabeti** ed il corrispondente simbolo $\boxminus$ al fine di avere la possibilità di servirsi di scritture come $\mathsf{A} \subset \boxminus$ e $\{\mathtt{a}_1, ..., \mathtt{a}_r\} \subset \boxminus$ per introdurre la notazione A per un alfabeto e le notazioni $\mathtt{a}_i$ per dei caratteri.

D50:a.03 Consideriamo un alfabeto $\mathsf{A} = \{\mathtt{a}_1, ..., \mathtt{a}_n\} \subset_F \boxminus$. Una stringa su tale alfabeto è quindi una sequenza
$w = \langle \mathtt{a}_{j_1}, \mathtt{a}_{j_2}, ..., \mathtt{a}_{j_r} \rangle \in \mathsf{A}^r$ per un certo intero positivo r; questo viene detto **lunghezza** di w e si denota $\ell(w)$, $w^{\vdash}$ o $|w|$.
Chiamiamo **giustapposizione** delle stringhe $w = \langle \mathtt{a}_{j_1}, \mathtt{a}_{j_2}, ..., \mathtt{a}_{j_r} \rangle$ e $x = \langle \mathtt{a}_{h_1}, \mathtt{a}_{h_2}, ..., \mathtt{a}_{h_s} \rangle$ la stringa
$w \cdot x := \langle \mathtt{a}_{j_1}, \mathtt{a}_{j_2}, ..., \mathtt{a}_{j_r}, \mathtt{a}_{h_1}, \mathtt{a}_{h_2}, ..., \mathtt{a}_{h_s} \rangle$. Evidentemente questa operazione binaria è associativa e questo consente di evitare la scrittura di parentesi: invece di $w \cdot (x \cdot y)$ o $(w \cdot x) \cdot y$ si può scrivere $w \cdot x \cdot y$. Una stringa può considerarsi ottenuta come giustapposizione dei caratteri che sono le sue successive componenti; si può quindi scrivere $w = \mathtt{a}_{j_1} \cdot \mathtt{a}_{j_2} \cdot ... \cdot \mathtt{a}_{j_r}$. Spesso poi il segno della giustapposizione, come altri segni del genere "prodotto", si trascura per brevità e si adotta una scrittura come la $w = \mathtt{a}_{j_1}\mathtt{a}_{j_2}...\mathtt{a}_{j_r}$. Quindi si può immaginare che una stringa sia ottenuta per accostamento materiale dei caratteri componenti e che la giustapposizione di w e x si ottenga "scrivendo" la w seguita immediatamente dalla x sopra un dispositivo sequenziale di registrazione (nastro).

D50:a.04 Per l'insieme delle stringhe su un alfabeto A si scrive: $\mathsf{A}^+ := \dot{\bigcup}_{r=1}^{\infty} \mathsf{A}^r$; A^+ munito della giustapposizione è un semigruppo chiamato **semigruppo libero** sull'alfabeto A.
Si osserva che la giustapposizione è commutativa se l'alfabeto ha un solo carattere, mentre con due o più caratteri in genere non lo è; ad esempio $\mathtt{ab} \cdot \mathtt{a} \neq \mathtt{a} \cdot \mathtt{ab}$.

I semigruppi liberi $\langle \mathsf{A}^+, \cdot \rangle$ non hanno unità; a ciascuno di essi, come ad ogni semigruppo, si può però aggiungere una nuova entità che indichiamo con μ, che considereremo stringa su ogni alfabeto e che chiamiamo **stringa muta** o **stringa vuota**, chiedendo che giustapposta a sinistra o a destra a qualsiasi stringa la lasci invariata: $w \cdot \mu = \mu \cdot w = w$. A questa stringa si attribuisce la lunghezza 0, $\mu^{\vdash} := 0$, e si scrive $\mathsf{A}^0 := \{\mu\}$ per ogni alfabeto A.

Definito $\mathsf{A}^* := \dot{\bigcup}_{r \in \mathbb{N}} \mathsf{A}^r$, si osserva che $\langle \mathsf{A}^*, \cdot, \mu \rangle$ costituisce un monoide del quale μ è l'unità; esso è detto **monoide libero** sull'alfabeto A.

D50:a.05 Una stringa $w \in \mathsf{A}^r$ si può considerare una funzione del tipo $\{(r] \longmapsto \mathsf{A}\}$ oppure del tipo $\{[r) \longmapsto \mathsf{A}\}$ a seconda che si attribuiscano alle sue componenti rispettivamente gli indici 1,...,r o gli indici 0,...,$r-1$. Secondo le due convenzioni, evidenziando le successive componenti, si scrive rispettivamente $w = w(1)...w(r)$ e $w = w(0)...w(r-1)$; nel seguito seguiremo prevalentemente la prima convenzione.
Scriviamo $w =_{\mathsf{A}} \mathtt{a}_{j_1}\mathtt{a}_{j_2}\ldots\mathtt{a}_{j_r}$ per indicare sia l'uguaglianza delle stringhe w ed $\mathtt{a}_{j_1}\mathtt{a}_{j_2}\ldots\mathtt{a}_{j_r}$, sia il fatto che ciascuno degli $\mathtt{a}_{j_1}, \mathtt{a}_{j_2}, \ldots, \mathtt{a}_{j_r}$ è un carattere dell'alfabeto A.

La lunghezza è definita per ogni elemento di A^* e $\forall w, x \in \mathsf{A}^* \ : \ (w \cdot x)^{\vdash} = w^{\vdash} + x^{\vdash}$. In termini algebrici questa relazione dice che la funzione lunghezza costituisce un morfismo di ogni monoide libero $\langle \mathsf{A}^*, \cdot, \mu \rangle$ sul monoide $\langle \mathbb{N}, +, 0 \rangle$. Le classi di equivalenza di questo morfismo sono gli insiemi A^r delle stringhe della stessa lunghezza.

D50:a.06 Si dice **linguaggio [formale]** sull'alfabeto A ogni sottoinsieme di A^*. Indichiamo con $\mathbf{Lng}_{\mathsf{A}}$ la loro collezione, cioè $\mathfrak{P}(\mathsf{A}^*)$.

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

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

Un tale insieme è evidentemente isomorfo ad $\mathbb{N}$ e fornisce una cosiddetta **rappresentazione unadica** degli interi naturali; si tratta della rappresentazione adottata da molti popoli primitivi (ad eccezione dello zero la cui introduzione storicamente ha richiesto un lungo travaglio): secondo essa un intero $n > 0$ è rappresentato da n repliche affiancate di un segno (o da n sassolini, da n conchiglie, ...) e la somma di due interi positivi è ottenuta accostando le sequenze di segni esprimenti gli addendi.
Per alfabeti A con due o più elementi, A^* ha una struttura molto più complessa e ricca di applicazioni di $\mathbb{N}$.

D50:a.07 Stringhe e linguaggi, nonostante la semplicità della loro definizione, costituiscono strumenti utili in moltissimi settori.
Essi vengono studiati da vari punti di vista. Alcune loro proprietà si ricavano con metodi algebrici che, come vedremo nel prossimo paragrafo, fanno riferimento a semigruppi, semianelli e serie di potenze. Questi metodi però hanno limiti piuttosto marcati: in effetti le strutture algebriche citate si basano su assiomi piuttosto deboli.
In effetti si rendono necessari vari approcci di tipo costruttivo: nei capitoli D52: e D54: vedremo due classi di strumenti che consentono di individuare e manipolare linguaggi, i riconoscitori e le grammatiche.
Inoltre molte proprietà dei linguaggi richiedono studi di tipo combinatorio nei quali si devono analizzare contemporaneamente biiezioni con configurazioni discrete di vario genere, meccanismi enumerativi, proprietà algebriche e costruzioni algoritmiche.

D50:a.08 Per quanto riguarda le applicazioni, occorre innanzitutto segnalare che mediante stringhe si possono esprimere, ovvero **codificare**, tutti gli oggetti studiati dalla matematica e tutte le informazioni sottoposte ad elaborazioni. Questo rende possibile fare sistematico riferimento ai procedimenti che operano su stringhe, a cominciare dallo studio dei fondamenti della matematica e delle procedure computazionali.
Molti oggetti della matematica discreta si studiano e si elaborano attraverso loro codifiche ed esse hanno un ruolo importante in vari punti della complessa rete di collegamenti che sussiste fra le configurazioni discrete.

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 cerca di tenere sotto controllo i linguaggi naturali e vari fenomeni e processi 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].

D50:a.09 I linguaggi forniscono numerosi spunti enumerativi. Il più semplice proviene dalla cosiddetta **successione di enumerazione -I** del linguaggio L, successione dei numeri delle sue stringhe caratterizzate delle successive lunghezze:

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

Altri spunti riguardano i diversi modi di individuare le singole stringhe di un linguaggio quando si opera con strumenti come le grammatiche (v. D54: per i fenomeni di ambiguità).

D50:b. Operazioni su stringhe e linguaggi

D50:b.01 Nel seguito del capitolo useremo caratteri come A, B, ... per denotare alfabeti, L, M, N, X, Y, ... per denotare linguaggi, a, b, c, ... per caratteri, u, v, w, x, y, z denoteranno stringhe, mentre m, n, r, s riguarderanno interi naturali.

Si dice **giustapposizione** dei linguaggi L ed M l'insieme delle stringhe ottenibili giustapponendo una stringa di L con una di M: $\mathsf{L}\cdot\mathsf{M} := \{w \in \mathsf{L},\ x \in \mathsf{M}\ :\mid w\cdot x\}$.

Di un linguaggio si possono quindi considerare le potenze positive:

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

Si pone inoltre $\mathsf{L}^0 := \{\mu\}$.

D50:b.02 Si dice poi **chiusura per giustapposizione** o **cross-chiusura** del linguaggio L l'unione delle sue potenze positive; la indicheremo con $\mathsf{L}^+ := \bigcup_{r=1}^{\infty}\mathsf{L}^r$.
Si dice invece **star-chiusura** di L l'unione delle sue potenze non negative; la denoteremo $\mathsf{L}^* := \bigcup_{r=0}^{\infty}\mathsf{L}^r$.

Si provano facilmente relazioni come le seguenti:

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

$$\mathsf{L}^+ = \mathsf{L}\cdot\mathsf{L}^* = \mathsf{L}^*\cdot\mathsf{L}\ ;\qquad \mathsf{L}^* = \mathsf{L}^+ \Longleftrightarrow \mu\in\mathsf{L}^+ \Longleftrightarrow \mu\in\mathsf{L}\ .$$

D50:b.03 L'operazione binaria di unione di linguaggi talora viene indicata con il segno "+": questo rende più evidenti alcune proprietà algebriche dei linguaggi che li avvicinano agli usuali campi numerici.
Si provano facilmente le seguenti uguaglianze:

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

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

Si ha però l'idempotenza della somma:

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

quindi non esiste operazione inversa della somma. Peraltro talora il segno "−" viene usato per indicare la differenza insiemistica di linguaggi.
Sopra un alfabeto di due o più caratteri in genere $\mathsf{L}\cdot\mathsf{M}\neq\mathsf{M}\cdot\mathsf{L}$, mentre sopra un alfabeto di un carattere $\mathsf{L}\cdot\mathsf{M} = \mathsf{M}\cdot\mathsf{L}$.
Per i linguaggi $\emptyset$ e $\{\mu\}$ si ha:

$$\mathsf{L}+\emptyset = \emptyset+\mathsf{L} = \mathsf{L}\ ;\qquad \mathsf{L}\cdot\emptyset = \emptyset\cdot\mathsf{L} = \emptyset\ ;\qquad \mathsf{L}\cdot\mu = \mu\cdot\mathsf{L} = \mathsf{L}\ ;$$

$$\emptyset^+ = \emptyset\ ;\qquad \emptyset^* = \mu\ ;\qquad \mu^+ = \mu^* = \mu\ .$$

L'insieme dei linguaggi sopra un alfabeto costituisce quindi un semianello, non commutativo sse l'alfabeto ha più di un carattere (v. T15h02); lo zero e l'unità di questo semianello sono $\emptyset$ e $\{\mu\}$.
Nelle precedenti relazioni abbiamo adottato un'altra usuale semplificazione evitando le parentesi graffe intorno a μ, cioè confondendo una stringa con il linguaggio costituito da questo solo elemento.

D50:b.04 Utilizzando i segni "+" e "·" le espressioni per i linguaggi finiti assumono aspetto di polinomi: ad esempio si scrive a^3+b^3 invece di $\{a^3, b^3\}$. Questo agevola sviluppi come quello che porta a scrivere

$$(a+b^2+b^2a)\cdot(\mu+a) = a+b^2+b^2a+a^2+b^2a^2 \ .$$

Ora i caratteri hanno il ruolo di "variabili non commutative"; l'idempotenza della somma non consente coefficienti superiori ad 1 nè coefficienti negativi; i soli coefficienti sono lo zero e l'unità, ovvero $\emptyset$ e μ; questi costituiscono il semianello, isomorfo a quello dei bits $\mathbb{B}$, su cui sono definiti i polinomi.
Analogamente un linguaggio infinito $\mathsf{L} \subseteq \mathsf{A}^*$ si può considerare come serie nelle variabili formali non commutative costituenti A sul semianello $\{\emptyset, \mu\}$ e si può scrivere:

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

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

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

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

Quindi per indicare che una stringa z è una potenza della w si scrive $z \in w^*$ e per indicare che tutte le stringhe del linguaggio M si possono esprimere come prodotti di stringhe di L si scrive $\mathsf{M} \subseteq \mathsf{L}^*$.

D50:b.05 Per le operazioni definite sui linguaggi si possono dimostrare senza difficoltà varie altre relazioni.

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

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

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

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

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

D50:b.06 Le uguaglianze trovate inducono a riferire i linguaggi ad una struttura algebrica più ricca di quella di semianello.
Chiamiamo, con [J .H. Conway 1971] **algebra di Kleene (classica)** una struttura della forma $\langle \boldsymbol{K}, +, \cdot, ^*, 0, 1\rangle$, dove $\langle \boldsymbol{K}, +, \cdot, 0\rangle$ è un semianello, l'operazione binaria "+" è idempotente $(\forall L \in \boldsymbol{K} \ : \ L+L=L)$, $1 \in \boldsymbol{K}\setminus 0$ è l'unità per l'operazione "·" e dove " * " è un'operazione unaria con le seguenti proprietà:

$$(L+M)^* = (L^*\cdot M)^*\cdot L^* \ ; \qquad (L\cdot M)^* = 1 + L\cdot(M\cdot L)^*\cdot M \ ; \qquad L^*\cdot L^* = L^* \ .$$

Inoltre, posto $L^+ := L\cdot L^*$, vale la successione di uguaglianze

$$\forall r\in\mathbb{N} \ : \ L^* = (L^r)^*\cdot(1 + L + \ldots + L^{r-1}) \ .$$

In particolare si ha l'algebra di Kleene dei linguaggi sull'alfabeto A :

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

Osserviamo anche che $\langle \{\emptyset, \mu\}, +, \cdot, ^*, \emptyset, \mu \rangle$ si può considerare la più semplice algebra di Kleene; essa può chiamarsi **algebra di Kleene booleana**.

D50:b.07 Una operazione unaria su stringhe e linguaggi che individua una utile simmetria è la **riflessione**. Si dice **riflessa** della stringa $w =_{\mathsf{A}} \mathtt{a}_{j_1}\mathtt{a}_{j_2}...\mathtt{a}_{j_r}$ la stringa ottenuta "leggendo i suoi caratteri da destra a sinistra", cioè $w^{\leftarrow} := \mathtt{a}_{j_r}...\mathtt{a}_{j_2}\mathtt{a}_{j_1}$. Ad esempio $(\mathtt{ROMA})^{\leftarrow} = \mathtt{AMOR}$.
Per la giustapposizione di due stringhe si ha $(w \cdot x)^{\leftarrow} = x^{\leftarrow} \cdot w^{\leftarrow}$; questo si esprime dicendo che la riflessione è un **antisomorfismo dei monoidi liberi**.
Si dice **riflesso** del linguaggio L l'insieme delle stringhe riflesse delle stringhe costituenti L:

$$\mathsf{L}^{\leftarrow} := \{w \in \mathsf{L} \nmid w^{\leftarrow}\} \ .$$

Evidentemente la riflessione è un'involuzione dei monoidi liberi e delle collezioni di linguaggi:

$$(\mathsf{L}^{\leftarrow})^{\leftarrow} = \mathsf{L} \ .$$

Si distinguono quindi le stringhe invarianti per riflessione, stringhe dette **palindrome** : una palindroma di lunghezza pari è `ANNA` ed una di lunghezza dispari è `ANILINA`.

D50:b.08 Eserc. Verificare che l'insieme delle palindrome su A è:

$$\sum_{w \in \mathsf{A}^*} ww^{\leftarrow} + \sum_{w \in \mathsf{A}^*} w\mathsf{A}w^{\leftarrow}.$$

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

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

D50:b.09 La riflessione e la giustapposizione sono state definite sui linguaggi estendendo operazioni definite sulle stringhe. Queste **estensioni booleane** possono effettuarsi per tutte le operazioni unarie, binarie, ... definite sugli elementi di una qualsiasi struttura algebrica per renderle applicabili ai sottoinsiemi della struttura stessa. In genere usando notazioni opportune non si generano confusioni se non si distingue fra operazioni e loro estensioni.
Per estensione booleana si ricavano facilmente le seguenti uguaglianze atte a chiarire le relazioni fra la riflessione e le altre operazioni sui linguaggi:

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

D50:b.10 Ricordiamo che in algebra un sottoinsieme T di un generico semigruppo S si dice: **ideale a sinistra** sse $ST = T$; **ideale a destra** sse $TS = T$; **ideale bilatero** sse $STS = T$; **sottosemigruppo** sse $TT \subseteq T$. Un sottoinsieme N di un monoide M avente $\mathtt{1}$ come unità si dice **sottomonoide** di M sse $\mathtt{1} \in N$ e $NN \subseteq N$, ovvero sse $NN = N$.

Vediamo come sono questi insiemi nel caso del monoide libero A^*.

D50:b.11 Prop. Relativamente ad A^*, gli ideali a sinistra sono linguaggi della forma $\mathsf{A}^*\mathsf{M}$, gli ideali a destra hanno forma $\mathsf{M}\mathsf{A}^*$, gli ideali bilateri forma $\mathsf{A}^*\mathsf{M}\mathsf{A}^*$, i sottosemigruppi forma M^+ ed i sottomonoidi T hanno forma M^*, cioè sono t.c. $T^* = T$; in tutte le espressioni precedenti M è un linguaggio su A qualsiasi ∎

Relativamente all'alfabeto A, i linguaggi della forma $\mathsf{A}^*\mathsf{N}$ sono detti **linguaggi cometa**, quelli della forma $\mathsf{M}\mathsf{A}^*$, **linguaggi anticometa**, quelli della forma $\mathsf{A}^*\mathsf{N}\mathsf{A}^*$ **linguaggi bicometa**. Si noti che i riflessi delle comete

sono le anticomete, mentre le collezioni delle bicomete, dei sottosemigruppi e dei sottomonoidi sono invarianti per riflessione. Possiamo quindi affermare che per riflessione le nozioni di ideale a sinistra e ideale a destra sono mutuamente duali, mentre le nozioni di ideale bilatero, sottosemigruppo e sottomonoide sono nozioni autoduali.

I linguaggi L^+ e L^* sono detti rispettivamente **sottosemigruppo** e **sottomonoide generato** da L. In accordo con questa dizione, semigruppi e monoidi liberi sono risp. semigruppi e monoidi generati dai relativi alfabeti.

Dato un sottomonoide M di A^* si può porre il problema di individuare un suo insieme di generatori, cioè un linguaggio contenuto in A^* che generi M. Naturalmente sono più interessanti i sistemi di generatori più ridotti, cioè minimali per l'inclusione.

D50:b.12 Prop. Ogni sottomonoide M di A^* ha un unico insieme di generatori minimale per inclusione esprimibile con la scrittura:

$$(\mathsf{M} \setminus \mu) \setminus (\mathsf{M} \setminus \mu)^2. \blacksquare$$

Osserviamo che l'espressione precedente (che spesso si trova scritta $(\mathsf{M} - \mu) - (\mathsf{M} - \mu)^2$) dà solo una indicazione genericamente formale per la ricerca dell'insieme di generatori minimale di M: un procedimento effettivo va individuato per singoli M o per collezioni particolari di linguaggi sulla base di proprietà specifiche.

D50:b.13 Si dice **morfismo** dal monoide $\langle M, \cdot, 1\rangle$ nel monoide $\langle N, \odot, \mathbf{1}\rangle$ ogni applicazione $\psi : M \longmapsto N$ t.c. $\forall a, b \in M \ : \ \psi(u \cdot v) = \psi(u) \odot \psi(v)$ e $\psi(1) = \mathbf{1}$.
Un morfismo da un monoide libero A^* in un monoide N risulta definito quando si forniscono i trasformati $\psi(\mathsf{a}_i)$ per tutti gli $\mathsf{a}_i \in \mathsf{A}$, cioè per tutti i suoi elementi generatori: infatti per ogni $\mathsf{a}_{j_1}...\mathsf{a}_{j_r} \in \mathsf{A}^*$ si ottiene $\psi(\mathsf{a}_{j_1}...\mathsf{a}_{j_r}) = \psi(\mathsf{a}_{j_1}) \odot ... \odot \psi(\mathsf{a}_{j_r})$.

I morfismi tra due monoidi liberi A^* e B^* sono detti anche **sostituzioni di caratteri con stringhe** e ciascuno di essi è definito dalla funzione $\lceil\, \mathsf{a}_i \in \mathsf{A} \mapsto \psi(\mathsf{a}_i) \,\rceil$.
In particolare un morfismo ψ è una **cancellazione di caratteri** sse $\psi(\mathsf{a}_i) = \mu$ per qualche $\mathsf{a}_i \in \mathsf{A}$; **rispetta la lunghezza** sse per ogni $\mathsf{a}_i \in \mathsf{A}$ si ha $\psi(\mathsf{a}_i) \in \mathsf{B}$; è un **isomorfismo** sse per ogni $\mathsf{a}_i \in \mathsf{A}$ si ha $\psi(\mathsf{a}_i) \in \mathsf{B}$ e $\psi(\mathsf{a}_i) = \psi(\mathsf{a}_j) \Longrightarrow \mathsf{a}_i = \mathsf{a}_j$, cioè sse la sua riduzione ad A è una biiezione di $\{\mathsf{A} \lhd\!\longrightarrow\!\rhd \mathsf{B}\}$.
Morfismi importanti sono gli isomorfismi del tipo $\mathsf{A}^* \lhd\!\longrightarrow \mathsf{B}^*$, trasformazioni invertibili e quindi tali da consentire, per ogni $w \in \mathsf{A}^*$, la ricostruzione a partire da $\psi(w)$ della w stessa. Queste trasformazioni sono l'oggetto della teoria dei codici, una teoria complessa e ricca di importanti applicazioni (v. D60:, D61:).

D50:b.14 Per estensione booleana le sostituzioni si applicano anche ai linguaggi :

$$\psi(\mathsf{L}) := \sum_{\mathsf{a}_{j_1}...\mathsf{a}_{j_r} \in \mathsf{L}} \psi(\mathsf{a}_{j_1}) \cdot ... \cdot \psi(\mathsf{a}_{j_r}).$$

Questa espressione si può utilizzare per definire più in generale l'applicazione ad un linguaggio di una funzione del tipo $\psi \in \{\mathsf{A} \longmapsto \mathbf{Lng}_{\mathsf{B}}\}$ la quale ad 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:

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

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

$$\psi(\mu + \mathtt{a}^2 + \mathtt{ba}^2\mathtt{b}) = \mu + \mathtt{cd}^+\mathtt{c}^2\mathtt{d}^+\mathtt{c} + \mathtt{cdc}^2\mathtt{d}^+\mathtt{c}^2\mathtt{d}^+\mathtt{c}^2\mathtt{dc}.$$

D50:c. Relazioni tra stringhe e derivazioni di linguaggi

D50:c.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** o **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** di w, e si scrive $u \preceq_{\mathtt{u}} w$, sse, essendo $u =_{\mathsf{A}} \mathtt{a}_{j_1} \cdots \mathtt{a}_{j_r}$, si ha $w \in \mathsf{A}^*\mathtt{a}_{j_1}\mathsf{A}^* \cdots \mathsf{A}^*\mathtt{a}_{j_r}\mathsf{A}^*$.
L'insieme dei prefissi di $w = \mathtt{abaa}$ è $\{\mu, \mathtt{a}, \mathtt{ab}, \mathtt{aba}, \mathtt{abaa}\}$; l'insieme dei suoi suffissi $\{\mu, \mathtt{a}, \mathtt{aa}, \mathtt{baa}, \mathtt{abaa}\}$; per avere tutti gli infissi di w occorre aggiungere $\mathtt{b}$ e $\mathtt{ba}$ all'unione dei suoi prefissi e dei suoi suffissi; per avere tutte le sottostringhe di w occorre aggiungere all'insieme dei suoi infissi $\mathtt{a}^3$.

Osserviamo che l'insieme degli infissi di w è l'insieme dei prefissi dei vari suffissi di w e coincide con l'insieme dei suffissi dei prefissi di w.

D50:c.02 Munendo A^* con ciascuna di queste relazioni si ottiene un digrafo infinito graduato, i cui livelli corrispondono alle successive potenze cartesiane A^r.
Quando $\mathsf{A} = \{\mid\}$ (ed in genere quando A è un singoletto) questi digrafi graduati si riducono al digrafo infinito

$$\mu \longrightarrow \mid \longrightarrow \mid\mid \longrightarrow \mid\mid\mid \longrightarrow \mid\mid\mid\mid \longrightarrow ...$$

isomorfo a quello di $\mathbb{N}$ munito della relazione successore *scsr* che con un solo carattere è una funzione.

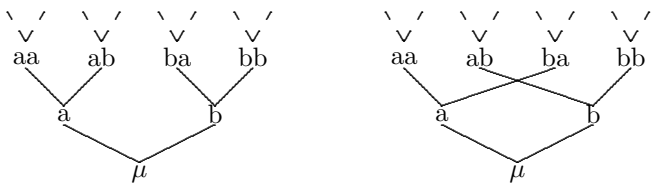
D50:c.03 Vediamo ora le caratteristiche delle relazioni precedenti per $|\mathsf{A}| \geq 2$. Osserviamo che tra le relazioni precedenti considerate come insiemi di coppie di stringhe su un dato alfabeto A, cioè come sottoinsiemi di $\mathsf{A}^* \times \mathsf{A}^*$ per un dato alfabeto A, sussistono le seguenti inclusioni:

$$\emptyset \subset \preceq_{\mathtt{p}} \cap \preceq_{\mathtt{s}} \subset \preceq_{\mathtt{p}} \quad , \quad \preceq_{\mathtt{s}} \subset \preceq_{\mathtt{p}} \cup \preceq_{\mathtt{s}} \subset \preceq_{\mathtt{i}} \subset \preceq_{\mathtt{u}} .$$

Queste relazioni si riducono convenientemente alle loro riduzioni riflessivo-transitive (uniche) che indicheremo rispettivamente con $\prec_{\mathtt{p}\,\mathtt{I}}$, $\prec_{\mathtt{s}\,\mathtt{I}}$, $\prec_{\mathtt{i}\,\mathtt{I}}$ e $\prec_{\mathtt{u}\,\mathtt{I}}$. Per esse vale una relazione di inclusione analoga alla precedente:

$$\emptyset \subset \prec_{\mathtt{p}\,\mathtt{I}} \cap \prec_{\mathtt{s}\,\mathtt{I}} \subset \prec_{\mathtt{p}\,\mathtt{I}} \quad , \quad \prec_{\mathtt{s}\,\mathtt{I}} \subset \prec_{\mathtt{p}\,\mathtt{I}} \cup \prec_{\mathtt{s}\,\mathtt{I}} = \prec_{\mathtt{i}\,\mathtt{I}} \subset \prec_{\mathtt{u}\,\mathtt{I}}.$$

Se $w \neq \mu$, $w\mathrm{succ}_{\mathtt{p}_\mathtt{I}}$ e $w\mathrm{succ}_{\mathtt{s}_\mathtt{I}}$ sono costituiti da una stringa: quindi A^* munito rispettivamente di $\prec_{\mathtt{p}_\mathtt{I}}$ e di $\prec_{\mathtt{s}\,\mathtt{I}}$ costituisce due arborescenze che si estendono illimitatamente verso l'alto (v. D28e).



Munendo A^* con $\prec_{\mathtt{i\ I}}$ si ha un digrafo infinito graduato, ma non un'arborescenza, in quanto $w\mathrm{succ}_{\mathtt{iI}}$ contiene una sola stringa sse w ha il primo e l'ultimo carattere coincidenti; in caso contrario ne contiene 2.
Con $\prec_{\mathtt{u\ I}}$ si ha un digrafo infinito graduato ottenuto dal precedente per aggiunta di numerosi archi (v. Eserc. .18:3.16).

D50:c.04 Della riflessione si può considerare l'estensione cartesiana applicata alle coppie o alle sequenze di stringhe: $\langle w, x\rangle^{\leftarrow} := \langle w^{\leftarrow}, x^{\leftarrow}\rangle$ e quindi agli insiemi di coppie, cioè alle relazioni tra stringhe. In particolare si verifica facilmente che le relazioni $\preceq_{\mathtt{p}}$ e $\preceq_{\mathtt{s}}$ sono l'una riflessa dell'altra e che $\preceq_{\mathtt{i}}$ e $\preceq_{\mathtt{u}}$ sono invarianti per riflessione. Questo corrisponde alle implicazioni:

$$w \preceq_{\mathtt{p}} x \iff (w^{\leftarrow}) \preceq_{\mathtt{s}} (x^{\leftarrow}) \ ;$$

$$w \preceq_{\mathtt{i}} x \iff (w^{\leftarrow}) \preceq_{\mathtt{i}} (x^{\leftarrow}) \ ; \qquad w \preceq_{\mathtt{u}} x \iff (w^{\leftarrow}) \preceq_{\mathtt{u}} (x^{\leftarrow}) \ .$$

D50:c.05 Abbiamo visto vari ordini parziali di A^*; spesso è utile completare un ordine parziale in un ordine totale (la cosa è sempre possibile in virtù di un teorema di G. Birkhoff).
Considereremo due ordini totali basati sopra un ordine totale $\leq$ dell'alfabeto A, cioè su una sequenzializzazione $\langle \mathtt{a}_1, \mathtt{a}_2, ..., \mathtt{a}_n\rangle$ dell'alfabeto.
Di 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 x **secondo l'ordine lessicografico**, e si scrive $w <_{lx} x$, sse si trova $k \in \{1, ..., r\}$ t.c. $i < k \Longrightarrow \mathtt{a}_{j_i} = \mathtt{a}_{h_i}$ e $\mathtt{a}_{j_k} < \mathtt{a}_{h_k}$. La relazione $\leq_{lx}$ è una relazione d'ordine totale in ogni A^r e non è difficile passare da una stringa di questo insieme a quella immediatamente precedente o successiva.

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 $x = \mathtt{a}_{h_1}...\mathtt{a}_{h_s}$:
- scorrendo le due stringhe contemporaneamente si cerca il primo $k \leq \min(r, s)$ t.c. $i < k \Longrightarrow \mathtt{a}_{j_i} = \mathtt{a}_{h_i}$ e $\mathtt{a}_{j_k} \neq \mathtt{a}_{h_k}$;
- se non si trova tale k: se $w = x$ si decide $w =_{lx} x$; se w è prefisso proprio di x si decide $w <_{lx} x$; se x è prefisso di w si decide $w >_{lx} x$;
- quando si incontra il suddetto k: se $\mathtt{a}_{j_k} < \mathtt{a}_{h_k}$ si decide $w <_{lx} x$, se viceversa si stabilisce $w >_{lx} x$.

D50:c.06 L'ordinamento lessicografico è un ordine totale che estende $\preceq_{\mathtt{p}}$; il suo primo elemento è μ e la stringa immediatamente successiva secondo $\leq_{lx}$ di una generica w è $w\mathtt{a}_1$. Non esiste invece una stringa immediatamente precedente l'ordine lessicografico per alcuna stringa che non abbia la forma $w\mathtt{a}_1$; 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 di $u\mathtt{a}_i x\mathtt{a}_1{}^+$ ed altre, mentre fra w e $w\mathtt{a}_1{}^k\mathtt{a}_h y$ con $k \geq 0$ e $\mathtt{a}_h > \mathtt{a}_1$ si trovano le infinite stringhe di $w\mathtt{a}_1{}^k\mathtt{a}_1{}^+$ ed altre. Quindi l'ordinamento lessicografico non è un ordine sequenziale.

D50:c.07 Un ordinamento sequenziale di A^* che estende $\preceq_{\mathtt{p}}$ e risulta spesso utile è l'ordinamento **secondo lunghezza-lessicografico** indicato $\leq_{ll}$. Esso si basa sul seguente algoritmo di confronto di due generiche stringhe diverse w e x. Primariamente si confrontano le loro lunghezze; se $w^{\vdash} < x^{\vdash}$ si decide $w <_{ll} x$, se $w^{\vdash} > x^{\vdash}$ si pone $w >_{ll} x$; se le due lunghezze coincidono si effettua il confronto lessicografico e si decide di conseguenza, cioè si pone $w >_{ll} x$ sse $w >_{lx} x$.
L'ordine $\leq_{ll}$ è un ordine sequenziale: infatti non è difficile passare da una stringa a quella immediatamente precedente o successiva. È semplice anche organizzare una procedura che scorra secondo $\leq_{ll}$ quante stringhe di A^* si vogliono ponendole in corrispondenza biunivoca con gli interi naturali. Si inizia con μ, corrispondente a 0; si procede primariamente sulle successive lunghezze 1,2, ...; nell'ambito di un A^r si comincia con $\mathtt{a}_1{}^r$; si passa da una generica $u\mathtt{a}_i\mathtt{a}_n{}^s$ con $i = 1, 2, ..., r$ ed $s = 0, ..., r - i$

alla immediatamente successiva $u\mathtt{a}_{i+1}\mathtt{a}_1{}^s$; infine di fronte ad $\mathtt{a}_n{}^r$, ultima stringa di A^r, si aumenta la lunghezza passando ad $\mathtt{a}_1{}^{r+1}$, prima stringa di A^{r+1}.
La corrispondenza biunivoca che si stabilisce in tal modo alla stringa $\mathtt{a}_{j_1}...\mathtt{a}_{j_r}$ associa

$$j_1\, n^{r-1} + j_2\, n^{r-2} + j_3\, n^{r-3}2 + \cdots + j_{r-1}\, n + j_r\ .$$

Inoltre alla stringa muta si associa lo 0. La stringa di L che in tal modo si associa ad un numero naturale si dice che lo rappresenta nel **sistema di numerazione** n**-adico**. Si osservi che se $\mathsf{A} = \{\mid\}$ questa formula a " $\mid^k$ " associa l'intero k, in accordo con la rappresentazione unadica degli interi naturali.

D50:c.08 Definiamo **derivata da sinistra rispetto ad una stringa** w di un generico linguaggio L il linguaggio

$$w \setminus\!\setminus \mathsf{L} := \{z \in \mathsf{A}^* \text{ t.c. } w \cdot z \in \mathsf{L}\},$$

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

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

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

D50:c.09 **(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.: $\forall \mathtt{a}_{j_1}, ..., \mathtt{a}_{j_k} \in \mathsf{A} \ :\ (\mathtt{a}_{j_1}\mathtt{a}_{j_2}...\mathtt{a}_{j_k}) \setminus\!\setminus \mathsf{L} = \mathtt{a}_{j_k} \setminus\!\setminus (...(\mathtt{a}_{j_2} \setminus\!\setminus (\mathtt{a}_{j_1} \setminus\!\setminus \mathsf{L}))...)$ ∎

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

D50:c.10 Si definisce **derivata da destra rispetto ad una stringa** z del linguaggio L

$$\mathsf{L} /\!\!/ z := \{w \in \mathsf{A}^* \text{ t.c. } w \cdot z \in \mathsf{L}\}.$$

Questa operazione è la duale per riflessione 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 :c.09.

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

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

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

D50:c.11 È naturale introdurre le estensioni booleane delle derivazioni rispetto a stringhe.
Si dice **derivata da sinistra rispetto ad un linguaggio** X di L :

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

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

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

D50:c.12 **(1) Prop.**: $\mathsf{X} \setminus\!\setminus \mathsf{L} \ = \ \{w \in \mathsf{A}^* \text{ t.c. } \mathsf{X}w \cap \mathsf{L} \neq \emptyset\}$; $\mathsf{L} /\!\!/ \mathsf{Y} \ = \ \{w \in \mathsf{A}^* \text{ t.c. } w\mathsf{Y} \cap \mathsf{L} \neq \emptyset\}$ ∎

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

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

(4) Prop.: $\mathsf{X} \setminus\!\setminus \mathsf{L} \neq \emptyset \iff \mathsf{X} \cap \mathsf{L} \neq \emptyset$; $\mathsf{L} /\!\!/ \mathsf{Y} \neq \emptyset \iff \mathsf{L} \cap \mathsf{Y} \neq \emptyset$ ∎

$\mathsf{L} /\!\!/ Y \neq \emptyset \iff \mathsf{L} \cap \mathsf{A}^* Y \neq \emptyset$ ∎

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

D50:c.13 Con le operazioni di derivazione si possono esprimere prefissi, suffissi ed infissi. Per questo è comodo servirsi di scritture come $w /\!\!/ s := w /\!\!/ \mathsf{A}^s$ e $s \setminus\!\setminus w := \mathsf{A}^s \setminus\!\setminus w$. Evidentemente se $w =_{\mathsf{A}} \mathtt{a}_{j_1}...\mathtt{a}_{j_r}$:

$$s \in [r] \implies w /\!\!/ s = \mathtt{a}_{j_1}...\mathtt{a}_{j_{r-s}} \;\;,\;\; s \setminus\!\setminus w = \mathtt{a}_{j_{s+1}...\mathtt{a}_{j_r}} \;\;,\;\; w = (w /\!\!/ s) \cdot ((r-s) \setminus\!\setminus w);$$

$$s,t \in [r], s+t \le r \implies (s \setminus\!\setminus w) /\!\!/ t = s \setminus\!\setminus (w /\!\!/ t) = \mathtt{a}_{j_{s+1}}...\mathtt{a}_{j_{r-t}} \;\;;$$

$$s \setminus\!\setminus w = \mu \iff w /\!\!/ s = \mu \iff \ell(w) = s \;\;;\;\; s \setminus\!\setminus w = \emptyset \iff w /\!\!/ s = \emptyset \iff \ell(w) < s \;\;.$$

Per estensione booleana si definiscono $s \setminus\!\setminus \mathsf{L} := \mathsf{A}^s \setminus\!\setminus \mathsf{L}$ e $\mathsf{L} /\!\!/ s := \mathsf{L} /\!\!/ \mathsf{A}^s$.

D50:c.14 Prop. Consideriamo una stringa w ed il relativo alfabeto minimo A.
L'insieme dei prefissi di w è $w /\!\!/ (\mathsf{A}^*)$;
l'insieme dei suffissi di w è $(\mathsf{A}^*) \setminus\!\setminus w$;
l'insieme degli infissi di w è $(\mathsf{A}^*) \setminus\!\setminus (w /\!\!/ (\mathsf{A}^*)) = ((\mathsf{A}^*) \setminus\!\setminus w) /\!\!/ (\mathsf{A}^*)$ ∎

D50:c.15 Prop. Consideriamo un linguaggio L ed 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 $(\mathsf{A}^*) \setminus\!\setminus (\mathsf{L} /\!\!/ (\mathsf{A}^*)) = ((\mathsf{A}^*) \setminus\!\setminus \mathsf{L}) /\!\!/ (\mathsf{A}^*)$ ∎
Le ultime due espressioni si possono semplificare nella $(\mathsf{A}^*) \setminus\!\setminus \mathsf{L} /\!\!/ (\mathsf{A}^*)$.

D50:c.16 Eserc. Mostrare che $w\mathrm{succ}_{\mathtt{u_I}}$ contiene un numero di elementi pari alla lunghezza delle stringa ottenuta eliminando da w ogni carattere coincidente con quello che lo precede e quindi che $1 \le |w\mathrm{succ}_{\mathtt{u_I}}| \le w^{\vdash\dashv}$.
Mostrare anche che $|w \prec_{\mathtt{u_I}}| = |\mathsf{A}| \cdot (w^{\vdash\dashv} + 1) - w^{\vdash\dashv}$.

D50:c.17 Eserc. Mostrare che, per L e M linguaggi qualsiasi:

$$o(\mathsf{L}^+) = o(\mathsf{L}) \;; \qquad o(\mathsf{L}^*) = \mu \;; \qquad o(\mathsf{L} + \mathsf{M}) = o(\mathsf{L}) + o(\mathsf{M}) \;; \qquad o(\mathsf{L} \cdot \mathsf{M}) = o(\mathsf{L}) \cdot o(\mathsf{M}).$$

D50:c.18 Eserc. Per l'operazione di derivazione da destra dimostrare:

$$\forall v, w \in \mathsf{A}^* \;:\; \mathsf{L} /\!\!/ (vw) = (\mathsf{L} /\!\!/ w) /\!\!/ v \quad ; \quad \forall w \in \mathsf{A}^* \;:\; (w \setminus\!\setminus \mathsf{L})^{\leftarrow} = \mathsf{L}^{\leftarrow} /\!\!/ w^{\leftarrow},$$

D50:c.19 Eserc. Dimostrare le seguenti proprietà della derivata da sinistra rispetto a linguaggi:

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

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

$$\mu \setminus\!\setminus \mathsf{L} = \mathsf{L} \;; \qquad \mathsf{X} \setminus\!\setminus \mu = o(\mathsf{X}) \;;$$

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

D50:c.20 Eserc. (a) Trovare le relazioni corrispondenti alle precedenti per la derivazione da destra.
(b) Dimostrare che: $\mathsf{X} \setminus\!\setminus (\mathsf{L} /\!\!/ \mathsf{Y}) = (\mathsf{X} \setminus\!\setminus \mathsf{L}) /\!\!/ \mathsf{Y}$.
(c) Assunti $L := a^2b^2 + (ab)^3 + (ab^2)^2$. $X := a\,b^*\,a$ e $Y := a\,a^*\,b$, si calcoli il linguaggio $\mathsf{X} \setminus\!\setminus (\mathsf{L} /\!\!/ \mathsf{Y})$.

D50:d. Fattorizzazioni di stringhe

D50:d.01 Una stringa $w \in \mathsf{A}^+$ si dice **parola primitiva** sse non si può esprimere come potenza di un'altra stringa (più corta) cioè sse $w \in z^+$ per qualche $z \in \mathsf{A}^+$ implica $w = z$. Indichiamo con $\mathsf{Primw}_\mathsf{A}$ il linguaggio delle parole primitive su A. Una stringa primitiva ed una non primitiva sono:

$$abbcabccabbacacbac \qquad \text{e} \qquad (acb)^5 = acbacbacbacbacb.$$

Come mostra questo esempio, a parità di lunghezza, le stringhe non primitive si possono considerare più regolari, ovvero meno complesse, delle primitive della stessa lunghezza, nel senso che possono individuarsi con meno informazioni. Viceversa le primitive sono da considerare poco abbreviabili.

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

D50:d.02 Vediamo quindi alcuni fatti collegati alla non primitività.

Prop. Per ogni $w \in \mathsf{A}^+$ si individua una sola parola primitiva z t.c. $w \in z^+$.

Dim.: Se $w^{\vdash\dashv}$ è 1 o un numero primo, la stessa w è primitiva. In caso contrario si esaminano i successivi prefissi di w aventi come lunghezza 1 o un divisore proprio di $w^{\vdash\dashv}$; per ciascuno di essi, che denotiamo con z, si controlla se $z^{w^{\vdash\dashv}/z^{\vdash\dashv}} = w$; la prima z che soddisfa questa richiesta è la stringa trovata. Un ulteriore prefisso proprio z_1 può soltanto una potenza di z. ∎

Evidentemente due stringhe che sono potenze di una stessa stringa commutano.

D50:d.03 Prop. Consideriamo due stringhe $w, x \in \mathsf{A}^+$; $wz = zw \iff \mathsf{A}^+ \ni z \;[\![\; w, x \in z^+$.

Dim.: "$\Longrightarrow$" Sia $wx = xw$; se $w^{\vdash\dashv} = x^{\vdash\dashv}$ deve essere $w = x$ e $z := w = x$. In caso contrario supponiamo che $w^{\vdash\dashv} < x^{\vdash\dashv}$; si può scrivere $x = w^h v$ con $0 \leq v^{\vdash\dashv} < w^{\vdash\dashv}$. Se $v^{\vdash\dashv} = 0$ si sceglie $z := w$ e $x = z^h$. Se $v^{\vdash\dashv} > 0$ si ha $w^h \setminus\!\setminus (wx) = w^h \setminus\!\setminus (xw)$ e quindi $wv = vw$ con $v^{\vdash\dashv} < w^{\vdash\dashv} < x^{\vdash\dashv}$. Siamo quindi ricondotti ad una ricerca analoga alla iniziale, ma per stringhe aventi somma delle lunghezze inferiore. Per induzione si può supporre di trovare z t.c. $w, v \in z^+$ e quindi anche $x = w^h v \in z^+$

"$\Longleftarrow$" la implicazione è evidente. ∎

D50:d.04 Prop. $w^m = v^n$ per qualche coppia di interi positivi $\langle m, n\rangle \iff$ in A^+ si trova una stringa z t.c. $w, v \in z^+$.

Dim.: Evidentemente se $m = n$ basta assumere $z := w = v$. Se $m = 1$ si assume $z := v$; se $n = 1$ si assume $z := w$. Se m ed n hanno un divisore comune e più precisamente $m = kp$ ed $n = kq$ con $k > 1$, allora $w^m = v^n$ implica $w^p = v^q$. Resta quindi da dimostrare l'asserto per m ed n mutuamente primi e per simmetria possiamo limitarci al caso $m < n$ (e $w^{\vdash\dashv} > v^{\vdash\dashv}$). Si può scrivere $w = v^h y$ con $0 < y^{\vdash\dashv} < v^{\vdash\dashv}$; esaminato l'inizio di w si trova $vy = yv$ e per la proposizione precedente si trova z t.c. $y, v \in z^+$ ed anche $w = v^h y \in z^+$ ∎

D50:d.05 Introduciamo ora una nozione, quella di occorrenza, la quale consente di approfondire lo studio delle regolarità delle parole e permette di definire due utili relazioni di equivalenza nei monoidi liberi.

Consideriamo la stringa $w = w(1)...w(s) \in \mathsf{A}^s$ con $\mathsf{A} = \{\mathtt{a}_1, ..., \mathtt{a}_n\}$. Ogni coppia $\langle \mathtt{a}, i\rangle \in \mathsf{A} \times (\!|s|\!]$ si dice **occorrenza** del carattere $\mathtt{a}$ nella stringa w sse $w(i) = \mathtt{a}$. Più in generale ogni coppia $\langle u, i\rangle \in \mathsf{A}^{\leq s} \times (\!|s|\!]$ si dice **occorrenza** di u nella w sse $w = xuy$ con $x^{\vdash\dashv} = i - 1$.

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

D50:d.06 Assegnamo ad A un ordine totale $\leq$, cioè poniamo in sequenza i caratteri che costituiscono questo alfabeto; supponiamo ad esempio che sia $\mathtt{a}_1 < \mathtt{a}_2 < \cdots < \mathtt{a}_n$.
Diciamo **vettore di Parikh** di w relativo ad A e $\leq$:

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

Ad esempio relativamente all'alfabeto $\{a, b, c, d\}$ ordinato secondo tradizione risulta:

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

La **funzione di Parikh** " $^{\mathcal{P}rk}$ " associa ad ogni stringa di A^* un elemento di $\mathbb{N}^n$, è suriettiva e nonbiiettiva sse $|\mathsf{A}| > 1$; la relativa equivalenza, che indicheremo con $\sim_{\mathtt{p}}$, è costituita dalle classi di stringhe di A^* ottenibili l'una dall'altra permutando i caratteri che le compongono.
Ad esempio $(accdadaca)^{\mathcal{P}rk} = (a^4c^2dcd)^{\mathcal{P}rk} = (c^2adca^2dca)^{\mathcal{P}rk} = \langle 4, 0, 3, 2 \rangle$.
Queste classi corrispondono ai multinsiemi basati sull'insieme ordinato A; esse sono quindi costituite da permutazioni con ripetizioni (v. B13e17) e le loro cardinalità sono date da coefficienti multinomiali: ad esempio il numero di stringhe equivalenti ad ESSERE è $6!/(3!\,2!\,1!) = 720/(6\,2) = 60$.

D50:d.07 Le classi di $\sim_{\mathtt{p}}$ sono interamente contenute negli insiemi A^s di stringhe aventi la stessa lunghezza; in altri termini la partizione di A^* relativa a $\sim_{\mathtt{p}}$ è un raffinamento di quella relativa alla lunghezza.
In genere è utile individuare le classi di una equivalenza attraverso loro elementi rappresentativi facilmente individuabili. Un atteggiamento spesso conveniente consiste nell'assumere come rappresentativo di ciascuna classe l'elemento minimo secondo un opportuno ordine totale dell'insieme ambiente.

D50:d.08 Come elementi rappresentativi delle classi di $\sim_{\mathtt{p}}$, come per altri raffinamenti della partizione della lunghezza, conviene assumere le stringhe minime secondo l'ordine lessicografico $\leq_{lx}$: la generica classe $w^{\natural} \sim_{\mathtt{p}}$ viene dunque rappresentata da

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

Sul piano algebrico si osserva che

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

dove con $\boldsymbol{+}$ indichiamo la somma termine a termine degli elementi di $\mathbb{N}^n$; la funzione di Parikh quindi è un morfismo di monoidi da $\langle \mathsf{A}^*, \cdot, \mu \rangle$ su $\langle \mathbb{N}^n, \boldsymbol{+}, 0^n \rangle$.

D50:d.09 Una proprietà che contribuisce alla regolarità delle parole riguarda la possibilità di trovare in esse occorrenze ripetute.
Si dice **coppia sovrapposta** sull'alfabeto A una stringa della forma $u = vy = xv$ con $v, x, y \in \mathsf{A}^+$; si dice invece **overlap** su A una stringa della forma $auaua$ con $a \in \mathsf{A}$ ed $u \in \mathsf{A}^+$. Un overlap fornisce un caso particolare di coppia sovrapposta: infatti può scriversi $auaua = vy = xv$ con $v = aua$, $x = au$ e $y = ua$.

D50:d.10 Prop. Ogni coppia sovrapposta $u = vy = xv$ sull'alfabeto A ha un overlap come prefisso ed un overlap come suffisso.

Dim.: Supponiamo sia $x^{\vdash} \leq v^{\vdash}$ e scriviamo $z := x \setminus\!\setminus v$, in modo che sia $u = xzy = xv = vy$. Evidentemente se $z = \mu$ si ha $x = v$ e $u = xx$; dunque lo stesso u è un overlap. In caso contrario indichiamo con $\mathtt{a}$ il primo carattere di v, osserviamo che $\mathtt{a}$ è l'iniziale anche di z, poniamo $z_1 := a \setminus\!\setminus z$ e osserviamo che z_1 è suffisso di z, v ed u. Posto $w := v /\!\!/ z_1$ si ha $u = awawaz_1$ e quindi si ha un overlap prefisso di u. La dimostrazione della presenza di un overlap suffisso si dimostra procedendo con le stringhe trasposte di quelle usate sopra ∎

D50:d.11 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 $w = xvy = x'vy'$ con $x^{\vdash\dashv} = i - 1$ e $x'^{\vdash\dashv} = i' - 1$. Si danno tre possibilità:

– si hanno **occorrenze disgiunte** sse $i + r < i'$, cioè sse si può scrivere $w = xvzvy'$ con $z \in \mathsf{A}^+$;

– si hanno **occorrenze adiacenti** sse $i + r = i'$, cioè sse si può scrivere $w = xvvy'$;

– si hanno **occorrenze sovrapposte** sse $i + r > i'$, cioè sse si può scrivere $w = xv'y'$ con v' coppia sovrapposta.

Si può ritenere che la presenza di occorrenze ripetute in una parola contribuisca alla sua regolarità e che questa sia accentuata se sono presenti occorrenze adiacenti, cioè fattori al quadrato, al cubo etc.

D50:d.12 **Prop.** Una stringa ha come infisso una coppia sovrapposta sse ha come infisso un overlap.

Dim.: Evidentemente la presenza di un overlap in una parola costituisce un caso particolare di presenza di una coppia sovrapposta. Per il viceversa dalla :d.04 segue che la presenza di una coppia sovrapposta implica la presenza di un overlap ∎

D50:d.13 Due parole w ed x si dicono **[ciclicamente] coniugate** sse una di esse si può ottenere permutando ciclicamente l'altra; in tal caso si scrive $w \sim_{\mathsf{c}} x$.

Chiaramente $w \sim_{\mathsf{c}} x$ sse w si può fattorizzare come $w = uv$ e $x = vu$, ovvero sse in A^+ si trova una stringa u t.c. $wu = ux$.

L'equivalenza $\sim_{\mathsf{c}}$ è un raffinamento della equivalenza $\sim_{\mathsf{p}}$ tra stringhe ottenibili attraverso permutazioni qualsiasi delle componenti: ad esempio $abc \sim_{\mathsf{p}} acb$, mentre $abc \not\sim_{\mathsf{c}} acb$.

Ad esempio le classi di $\sim_{\mathsf{c}}$ contenute in $\{a, b\}^3$ sono $\{a^3\}$, $\{b^3\}$, $\{aab, aba, baa\}$ e $\{abb, bab, bba\}$; esse coincidono con le classi di $\sim_{\mathsf{p}}$. Le classi contenute in $\{a, b\}^4$ sono invece $\{a^4\}$, $\{b^4\}$, $\{aaab, aaba, abaa, baaa\}$, $\{aabb, abba, baab, bbaa\}$, $\{abbb, babb, bbab, bbba\}$ e $\{aabb, abba, bbaa, baab\}$ e $\{abab, baba\}$; per avere le classi relative a $\sim$ p basta fondere le ultime due scritte.

Come per $\sim_{\mathsf{p}}$, anche le classi di $\sim_{\mathsf{c}}$ sono opportunamente rappresentate dalle stringhe minime secondo un ordine lessicografico.

D50:d.14 Si osserva che se una stringa w è potenza di una stringa più corta, $w = z^q$ 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 particolari le stringhe primitive risultano più interessanti delle non primitive in quanto consentono maggiore concisione. Vedremo che tra le primitive risultano particolarmente utili le minime lessicografiche delle classi di $\sim_{\mathsf{c}}$.

D50:d.15 Si dice **parola di Lyndon** sull'alfabeto A una stringa di A^+ che è primitiva ed è la minima secondo l'ordine lessicografico nella propria classe di stringhe coniugate. Indichiamo con $\mathsf{Lynd}_{\mathsf{A}}$ il linguaggio delle parole di Lyndon su A.

In particolare possiamo scrivere:

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

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

È allora evidente che $a^+b^+ \subset \mathsf{Lynd}_{\{a,b\}}$ e che $\mathsf{A} \subset \mathsf{B} \implies \mathsf{Lynd}_{\mathsf{A}} \subset \mathsf{Lynd}_{\mathsf{B}}$. Quindi tutti i linguaggi di Lyndon sono infiniti.

D50:d.16 Prop. Una stringa $w \in \mathsf{A}^+$ appartiene a $\mathsf{Lynd}_{\mathsf{A}}$ sse precede strettamente secondo l'ordine lessicografico ogni suo suffisso proprio.
..... COW65 ∎

D50:d.17 Prop. Una stringa $w \in \mathsf{A}^+$ appartiene a $\mathsf{Lynd}_{\mathsf{A}}$ sse appartiene ad A oppure sse è esprimibile come $w = x \cdot y$ con $x, y \in \mathsf{Lynd}_{\mathsf{A}}$ e $w \leq_{lx} y$.
..... COW66 ∎

D50:d.18 Teorema Ogni stringa $w \in \mathsf{A}^+$ possiede una unica fattorizzazione mediante una sequenza di parole di Lyndon lessicograficamente non crescenti.
..... COW67 ∎

Le varie componenti di questo testo sono accessibili in http://www.mi.imati.cnr.it/∼alberto