STRUMENTI FORMALI PER LA BIOINFORMATICA

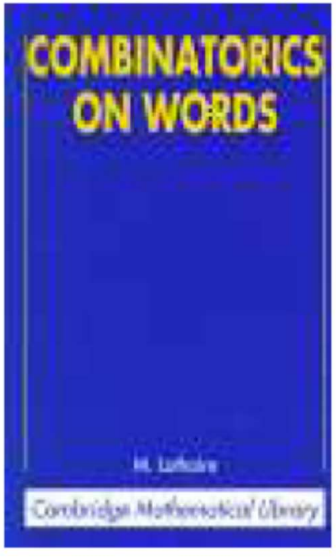
Combinatoria delle parole (Parte 1)

Questi lucidi sono basati su una traduzione in italiano di un corso in inglese tenuto dal Prof. Dominique Perrin dell'Universitè de Paris Est.

E sui libri:
M. Lothaire, Algebraic Combinatorics on Words, Encyclopedia Math. Appl., vol. 90, Cambridge Univ. Press, Cambridge, 2002.

Jeffrey Shallit, Second Course in Formal Languages and Automata Theory, Cambridge Univ. Press, Cambridge, 2009.

Referenza


COMBINATORICS
ON WORDS
M. Lothaire
Cambridge Mathematical Library

Il **numero di occorrenze di un carattere** a in una stringa x viene indicato da $|x|_a$.

Il **numero di occorrenze di un carattere** a in una stringa x viene indicato da $|x|_a$.

Esempio:

$$|aab|_a = 2, \quad |baa|_a = 2, \quad |baa|_b = 1, \quad |baa|_c = 0$$

Il **numero di occorrenze di un carattere** a in una stringa x viene indicato da $|x|_a$.

Esempio:

$$|aab|_a = 2, \quad |baa|_a = 2, \quad |baa|_b = 1, \quad |baa|_c = 0$$

La **lunghezza** di una stringa x è il numero di simboli in x.

Il **numero di occorrenze di un carattere** a in una stringa x viene indicato da $|x|_a$.

Esempio:

$$|aab|_a = 2, \quad |baa|_a = 2, \quad |baa|_b = 1, \quad |baa|_c = 0$$

La **lunghezza** di una stringa x è il numero di simboli in x.

La lunghezza di x è denotata con $|x|$.

Il **numero di occorrenze di un carattere** a in una stringa x viene indicato da $|x|_a$.

Esempio:

$$|aab|_a = 2, \quad |baa|_a = 2, \quad |baa|_b = 1, \quad |baa|_c = 0$$

La **lunghezza** di una stringa x è il numero di simboli in x.

La lunghezza di x è denotata con $|x|$.

Esempio: $|ab| = 2$, $|abaa| = 4$.

Due stringhe

$$x = a_1 a_2 \cdots a_h, \quad y = b_1 b_2 \cdots b_k,$$

con $a_i, b_j \in \Sigma$, $1 \leq i \leq h$, $1 \leq j \leq k$, si dicono **eguali** se

Due stringhe

$$x = a_1 a_2 \cdots a_h, \quad y = b_1 b_2 \cdots b_k,$$

con $a_i, b_j \in \Sigma$, $1 \leq i \leq h$, $1 \leq j \leq k$, si dicono **eguali** se $h = k$ e $a_i = b_i$, per $i = 1, \ldots, h$.

Due stringhe

$$x = a_1 a_2 \cdots a_h, \quad y = b_1 b_2 \cdots b_k,$$

con $a_i, b_j \in \Sigma$, $1 \leq i \leq h$, $1 \leq j \leq k$, si dicono **eguali** se $h = k$ e $a_i = b_i$, per $i = 1, \ldots, h$.

In due stringhe uguali i caratteri letti ordinatamente da sinistra a destra coincidono.

Due stringhe

$$x = a_1 a_2 \cdots a_h, \quad y = b_1 b_2 \cdots b_k,$$

con $a_i, b_j \in \Sigma$, $1 \leq i \leq h$, $1 \leq j \leq k$, si dicono **eguali** se $h = k$ e $a_i = b_i$, per $i = 1, \ldots, h$.

In due stringhe uguali i caratteri letti ordinatamente da sinistra a destra coincidono.

Esempio: $aba \neq baa$, $baa \neq ba$.

Date le stringhe

$$x = a_1 a_2 \cdots a_h, \quad y = b_1 b_2 \cdots b_k,$$

con $a_i, b_j \in \Sigma$, $1 \leq i \leq h$, $1 \leq j \leq k$, la **concatenazione** (di x e y) è definita da

$$x \cdot y = a_1 a_2 \cdots a_h b_1 b_2 \cdots b_k$$

Date le stringhe

$$x = a_1 a_2 \cdots a_h, \quad y = b_1 b_2 \cdots b_k,$$

con $a_i, b_j \in \Sigma$, $1 \leq i \leq h$, $1 \leq j \leq k$, la **concatenazione** (di x e y) è definita da

$$x \cdot y = a_1 a_2 \cdots a_h b_1 b_2 \cdots b_k$$

La concatenazione di due stringhe x e y è spesso denotata xy (invece che $x \cdot y$).

Date le stringhe

$$x = a_1 a_2 \cdots a_h, \quad y = b_1 b_2 \cdots b_k,$$

con $a_i, b_j \in \Sigma$, $1 \leq i \leq h$, $1 \leq j \leq k$, la **concatenazione** (di x e y) è definita da

$$x \cdot y = a_1 a_2 \cdots a_h b_1 b_2 \cdots b_k$$

La concatenazione di due stringhe x e y è spesso denotata xy (invece che $x \cdot y$).

- **Esempio:** $x = vice$, $y = capo$, $z = stazione$ $xy = vicecapo$, $yx = capovice \neq xy$

$$(xy)z = vicecapostazione = x(yz)$$

Operazioni sulle stringhe

La concatenazione **non è commutativa**, in generale $xy \neq yx$.

La concatenazione **non è commutativa**, in generale $xy \neq yx$.

La concatenazione **è associativa**:

$$(xy)z = x(yz)$$

(possiamo scrivere senza parentesi la concatenazione di tre o più stringhe).

La concatenazione **non è commutativa**, in generale $xy \neq yx$.

La concatenazione **è associativa**:

$$(xy)z = x(yz)$$

(possiamo scrivere senza parentesi la concatenazione di tre o più stringhe).

$$|xy| = |x| + |y|$$

Stringa vuota

La **stringa vuota** ϵ è la stringa che non contiene nessun simbolo.

Stringa vuota

La **stringa vuota** ϵ è la stringa che non contiene nessun simbolo.

Proprietà della stringa vuota:

$$x\epsilon = \epsilon x = x$$

$$|\epsilon| = 0$$

Stringa vuota

La **stringa vuota** ϵ è la stringa che non contiene nessun simbolo.

Proprietà della stringa vuota:

$$x\epsilon = \epsilon x = x$$

$$|\epsilon| = 0$$

Nota :

$$\emptyset \neq \epsilon, \quad \emptyset \neq \{\epsilon\}$$

$\emptyset$ è un sottoinsieme di Σ^*, $\epsilon \in \Sigma^*$;
$|\emptyset| = 0 \neq 1 = |\{\epsilon\}|$.

Def. Data una stringa x, una *sottostringa* di x è una qualsiasi sequenza di simboli consecutivi della stringa x. Un *prefisso* di x è una qualsiasi sequenza di simboli consecutivi iniziali della stringa x. Un *suffisso* di x è una qualsiasi sequenza di simboli consecutivi terminali della stringa x.

Def. Data una stringa x, una *sottostringa* di x è una qualsiasi sequenza di simboli consecutivi della stringa x. Un *prefisso* di x è una qualsiasi sequenza di simboli consecutivi iniziali della stringa x. Un *suffisso* di x è una qualsiasi sequenza di simboli consecutivi terminali della stringa x.

Se $x = uyv$ è la concatenazione di stringhe u, y, v (eventualmente vuote) allora:

Sottostringa

Def. Data una stringa x, una *sottostringa* di x è una qualsiasi sequenza di simboli consecutivi della stringa x. Un *prefisso* di x è una qualsiasi sequenza di simboli consecutivi iniziali della stringa x. Un *suffisso* di x è una qualsiasi sequenza di simboli consecutivi terminali della stringa x.

Se $x = uyv$ è la concatenazione di stringhe u, y, v (eventualmente vuote) allora:

- y è una **sottostringa** di x,

Sottostringa

Def. Data una stringa x, una *sottostringa* di x è una qualsiasi sequenza di simboli consecutivi della stringa x. Un *prefisso* di x è una qualsiasi sequenza di simboli consecutivi iniziali della stringa x. Un *suffisso* di x è una qualsiasi sequenza di simboli consecutivi terminali della stringa x.

Se $x = uyv$ è la concatenazione di stringhe u, y, v (eventualmente vuote) allora:

- y è una **sottostringa** di x,
- u è un **prefisso** di x,

Sottostringa

Def. Data una stringa x, una *sottostringa* di x è una qualsiasi sequenza di simboli consecutivi della stringa x. Un *prefisso* di x è una qualsiasi sequenza di simboli consecutivi iniziali della stringa x. Un *suffisso* di x è una qualsiasi sequenza di simboli consecutivi terminali della stringa x.

Se $x = uyv$ è la concatenazione di stringhe u, y, v (eventualmente vuote) allora:

- y è una **sottostringa** di x,
- u è un **prefisso** di x,
- v è un **suffisso** di x

Sottostringa

Def. Data una stringa x, una *sottostringa* di x è una qualsiasi sequenza di simboli consecutivi della stringa x. Un *prefisso* di x è una qualsiasi sequenza di simboli consecutivi iniziali della stringa x. Un *suffisso* di x è una qualsiasi sequenza di simboli consecutivi terminali della stringa x.

Se $x = uyv$ è la concatenazione di stringhe u, y, v (eventualmente vuote) allora:

- y è una **sottostringa** di x,
- u è un **prefisso** di x,
- v è un **suffisso** di x

Una sottostringa (prefisso, suffisso) di x è **propria** se non coincide con x.

Esempio: La stringa 472 ha

Esempio: La stringa 472 ha

- prefissi: $\epsilon, 4, 47, 472,$

Esempio: La stringa 472 ha

- prefissi: $\epsilon, 4, 47, 472,$
- suffissi: $\epsilon, 2, 72, 472,$

Esempio: La stringa 472 ha

- prefissi: $\epsilon, 4, 47, 472,$
- suffissi: $\epsilon, 2, 72, 472,$
- sottostringhe: $\epsilon, 4, 7, 2, 47, 72, 472$

Esempio: La stringa 472 ha

- prefissi: $\epsilon, 4, 47, 472,$
- suffissi: $\epsilon, 2, 72, 472,$
- sottostringhe: $\epsilon, 4, 7, 2, 47, 72, 472$
- La stringa 42 non è sottostringa di 472.

Inversa o riflessione di una stringa

Definizione

L'inversa (o reverse o riflessione) $\mathbf{w}^R$ *di una stringa* w *è la stringa ottenuta scrivendo i caratteri di* w *da destra verso sinistra.*

Inversa o riflessione di una stringa

Definizione

L'inversa (o reverse o riflessione) $\mathbf{w}^R$ *di una stringa* w *è la stringa ottenuta scrivendo i caratteri di* w *da destra verso sinistra.*

$\epsilon^R = \epsilon$ e se $w = a_1 \cdots a_n$, con a_j lettere, allora

$$\mathbf{w}^R = a_n a_{n-1} \cdots a_1.$$

Inversa o riflessione di una stringa

Definizione

L'inversa (o reverse o riflessione) $\mathbf{w}^R$ *di una stringa* w *è la stringa ottenuta scrivendo i caratteri di* w *da destra verso sinistra.*

$\epsilon^R = \epsilon$ e se $w = a_1 \cdots a_n$, con a_j lettere, allora

$$\mathbf{w}^R = a_n a_{n-1} \cdots a_1.$$

Esempio: $x = roma$, $x^R = amor$.

Inversa o riflessione di una stringa

Definizione

L'inversa (o reverse o riflessione) $\mathbf{w}^R$ *di una stringa* w *è la stringa ottenuta scrivendo i caratteri di* w *da destra verso sinistra.*

$\epsilon^R = \epsilon$ e se $w = a_1 \cdots a_n$, con a_j lettere, allora

$$\mathbf{w}^R = a_n a_{n-1} \cdots a_1.$$

Esempio: $x = roma$, $x^R = amor$.

Proprietà:

$$(x^R)^R = x, \quad (xy)^R = y^R x^R$$

Sia $m \geq 1$ un intero non negativo. La *potenza m-esima* di una stringa x è la concatenazione di x con sé stessa $m - 1$ volte.
Per convenzione la potenza 0 di una stringa è la stringa vuota.

Sia $m \geq 1$ un intero non negativo. La *potenza m-esima* di una stringa x è la concatenazione di x con sé stessa $m-1$ volte. Per convenzione la potenza 0 di una stringa è la stringa vuota.

Definizione

Sia x una stringa. Poniamo
PASSO BASE: $x^0 = \epsilon$
PASSO RICORSIVO: $x^m = x^{m-1}x$, *per* $m > 0$.

Esempi:

Esempi:

$$
\begin{aligned}
x &= ab \\
x^0 &= \epsilon \\
x^1 &= x = ab \\
x^2 &= (ab)^2 = abab \\
y &= a^2 = aa \\
y^3 &= a^2a^2a^2 = a^6 \\
\epsilon^0 &= \epsilon \\
\epsilon^2 &= \epsilon
\end{aligned}
$$

Nota. È necessario racchiudere tra parentesi la stringa da elevare alla potenza se ha lunghezza maggiore di uno.

Nota. È necessario racchiudere tra parentesi la stringa da elevare alla potenza se ha lunghezza maggiore di uno.

$$(ab)^2 = abab \neq abb = ab^2$$

Nota. È necessario racchiudere tra parentesi la stringa da elevare alla potenza se ha lunghezza maggiore di uno.

$$(ab)^2 = abab \neq abb = ab^2$$

L'elevamento a potenza ha *precedenza* rispetto alla concatenazione.

Nota. È necessario racchiudere tra parentesi la stringa da elevare alla potenza se ha lunghezza maggiore di uno.

$$(ab)^2 = abab \neq abb = ab^2$$

L'elevamento a potenza ha *precedenza* rispetto alla concatenazione.

Anche la riflessione ha *precedenza* rispetto alla concatenazione.

Nota. È necessario racchiudere tra parentesi la stringa da elevare alla potenza se ha lunghezza maggiore di uno.

$$(ab)^2 = abab \neq abb = ab^2$$

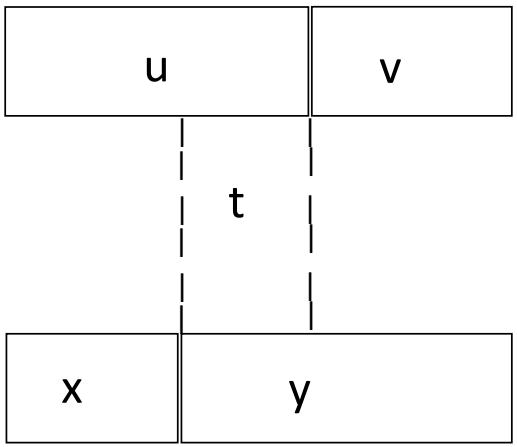
L'elevamento a potenza ha *precedenza* rispetto alla concatenazione.

Anche la riflessione ha *precedenza* rispetto alla concatenazione.

$b^R = b$, quindi $ab^R = ab$.

$$(ab)^R = ba \neq ab^R = ab$$

Lemma di Levi



Lemma (Lemma di Levi)

Siano $u, v, x, y \in A^*$ *e supponiamo che* $uv = xy$. *Se* $|u| \geq |x|$, *esiste* $t \in A^*$ *tale che* $u = xt$ *e* $y = tv$. *Se* $|u| < |x|$, *esiste* $t \in A^+$ *tale che* $x = ut$ *e* $v = ty$.

Lemma di Levi

$$uv = xy,\ |u| \geq |x| \quad \Rightarrow \quad \exists t \in A^*\ u = xt,\ y = tv$$

Esempio:

$$(abba)(aab) = (abb)(aaab)$$

$$u = abba,\ v = aab,\ x = abb,\ y = aaab$$

Chi è t?

Lemma di Levi

$$uv = xy,\ |u| \geq |x| \quad \Rightarrow \quad \exists t \in A^*\ u = xt,\ y = tv$$

Esempio:

$$(abba)(aab) = (abb)(aaab)$$

Lemma di Levi

$$uv = xy,\ |u| \geq |x| \quad \Rightarrow \quad \exists t \in A^*\ u = xt,\ y = tv$$

Esempio:

$$(abba)(aab) = (abb)(aaab)$$

$$u = abba,\ v = aab,\ x = abb,\ y = aaab$$

Lemma di Levi

$$uv = xy,\ |u| \geq |x| \quad \Rightarrow \quad \exists t \in A^*\ u = xt,\ y = tv$$

Esempio:

$$(abba)(aab) = (abb)(aaab)$$

$$u = abba,\ v = aab,\ x = abb,\ y = aaab$$

$$u = xa, \quad y = av$$

Lemma di Levi

$$uv = xy,\ |u| \geq |x| \quad \Rightarrow \quad \exists t \in A^*\ u = xt,\ y = tv$$

Esempio:

$$(abba)(aab) = (abb)(aaab)$$

$$u = abba,\ v = aab,\ x = abb,\ y = aaab$$

$$u = xa, \quad y = av$$

$$t = a$$

Primo Teorema di Lyndon-Schützenberger

Per motivare il primo teorema di Lyndon-Schützenberger, consideriamo il problema seguente: sotto quali condizioni una stringa può avere un prefisso proprio e un suffisso che sono uguali?

Esempi:
amaca inizia e termina con *a*, *barba* inizia e termina con *ba*.
ababab inizia e termina con *abab*.

Primo Teorema di Lyndon-Schützenberger

Teorema (Lyndon-Schützenberger)

Siano $x, y, z \in A^+$. *Allora* $xy = yz$ *se e solo se esistono* $u \in A^+$, $v \in A^*$ *e un intero* $e \geq 0$ *tali che* $x = uv$ *e* $z = vu$ *e* $y = (uv)^e u = u(vu)^e$.

Primo Teorema di Lyndon-Schützenberger

$$(\forall x \in A^+ \ \forall y \in A^+ \ \forall z \in A^+ \ \ xy = yz) \quad \Leftrightarrow$$

$$(\exists u \in A^+ \ \exists v \in A^* \ \exists e \in \mathbb{N} \ e \geq 0$$

$$x = uv, \ z = vu, \ y = (uv)^e u = u(vu)^e)$$

Esempi.

amaca: $x = amac$, $y = a$, $z = maca$

barba: $x = bar$, $y = ba$, $z = rba$

ababab: $x = ab$, $y = abab$, $z = ab$.

Secondo Teorema di Lyndon-Schützenberger

Il secondo teorema di Lyndon-Schützenberger risponde al problema seguente: sotto quali condizioni due stringhe possono commutare?

Cioè quando abbiamo $xy = yx$?

Secondo Teorema di Lyndon-Schützenberger

Quando abbiamo $xy = yx$?

Secondo Teorema di Lyndon-Schützenberger

Quando abbiamo $xy = yx$?

$$x = abab = (ab)^2,$$
$$y = ababab = (ab)^3$$

Secondo Teorema di Lyndon-Schützenberger

Quando abbiamo $xy = yx$?

$$x = abab = (ab)^2,$$
$$y = ababab = (ab)^3$$

$$xy = (ab)^2(ab)^3 = (ab)^5 = (ab)^3(ab)^2 = yx$$

Secondo Teorema di Lyndon-Schützenberger

Quando abbiamo $xy = yx$?

$$x = abab = (ab)^2,$$
$$y = ababab = (ab)^3$$

$$xy = (ab)^2(ab)^3 = (ab)^5 = (ab)^3(ab)^2 = yx$$

$$x^3 = (ab)^6 = y^2$$

Secondo Teorema di Lyndon-Schützenberger

Teorema (Lyndon-Schützenberger)

Siano $x, y \in A^+$. Le seguenti tre condizioni sono equivalenti.

1. $xy = yx$.
2. *Esistono* $z \in A^+$ *e interi* $h, k > 0$ *tali che* $x = z^h$ *e* $y = z^k$.
3. *Esistono interi* $i, j > 0$ *tali che* $x^i = y^j$.

Una parola w è primitiva se $w = v^n$ implica $n = 1$.

Una parola w è primitiva se $w = v^n$ implica $n = 1$.

Nota che la parola vuota non è primitiva.

Una parola w è primitiva se $w = v^n$ implica $n = 1$.

Nota che la parola vuota non è primitiva.

abab è primitiva?

Una parola w è primitiva se $w = v^n$ implica $n = 1$.

Nota che la parola vuota non è primitiva.

$abab$ è primitiva?

No: $abab = (ab)^2$.

Una parola w è primitiva se $w = v^n$ implica $n = 1$.

Nota che la parola vuota non è primitiva.

$abab$ è primitiva?

No: $abab = (ab)^2$.

aba, abb sono parole primitive.

Proposizione

Ogni parola non vuota è potenza di un'unica parola primitiva.

Proposizione

Ogni parola non vuota è potenza di un'unica parola primitiva.

$$w = v^n$$

Proposizione

Ogni parola non vuota è potenza di un'unica parola primitiva.

$w = v^n$

$n > 1$? Sì: w non è primitiva, $0 < |v| < |w|$

Proposizione

Ogni parola non vuota è potenza di un'unica parola primitiva.

$w = v^n$

$n > 1$? Sì: w non è primitiva, $0 < |v| < |w|$

$v = z^m$

Proposizione

Ogni parola non vuota è potenza di un'unica parola primitiva.

$w = v^n$

$n > 1$? Sì: w non è primitiva, $0 < |v| < |w|$

$v = z^m$

$m > 1$? Sì: v non è primitiva, $0 < |z| < |v| < |w|$, $w = z^{mn}$

Proposizione

Ogni parola non vuota è potenza di un'unica parola primitiva.

$w = v^n$

$n > 1$? Sì: w non è primitiva, $0 < |v| < |w|$

$v = z^m$

$m > 1$? Sì: v non è primitiva, $0 < |z| < |v| < |w|$, $w = z^{mn}$

Si intuisce una dimostrazione formale della proposizione basata sul principio di induzione.

Due parole x, y sono coniugate se esistono parole u, v tali che $x = uv, y = vu$.

La relazione di coniugazione è una relazione di equivalenza.

Una *classe di coniugazione* è una classe di questa relazione di equivalenza.

Una classe di coniugazione è spesso chiamata necklace.

Esempio 1

Sia $w = banana$.

$$
\begin{array}{cccccc}
b & a & n & a & n & a \\
a & n & a & n & a & b \\
n & a & n & a & b & a \\
a & n & a & b & a & n \\
n & a & b & a & n & a \\
a & b & a & n & a & n
\end{array}
$$

Esempio 2

Sia $w = abraca$

$$
\begin{array}{cccccc}
a & b & r & a & c & a \\
b & r & a & c & a & a \\
r & a & c & a & a & b \\
a & c & a & a & b & r \\
c & a & a & b & r & a \\
a & a & b & r & a & c
\end{array}
$$

Quante coniugate ha la stringa *abab*?

Quante coniugate ha la stringa *abab*?

La stringa *abab* ha due coniugate: *abab* e *baba*.

Quante coniugate ha la stringa *abab*?

La stringa *abab* ha due coniugate: *abab* e *baba*.

Quante coniugate ha la stringa *aaa*?

Quante coniugate ha la stringa *abab*?

La stringa *abab* ha due coniugate: *abab* e *baba*.

Quante coniugate ha la stringa *aaa*?

La stringa *aaa* ha una sola coniugata, è coniugata solo di sé stessa.

Se w è una parola primitiva, tutte le sue coniugate sono primitive.

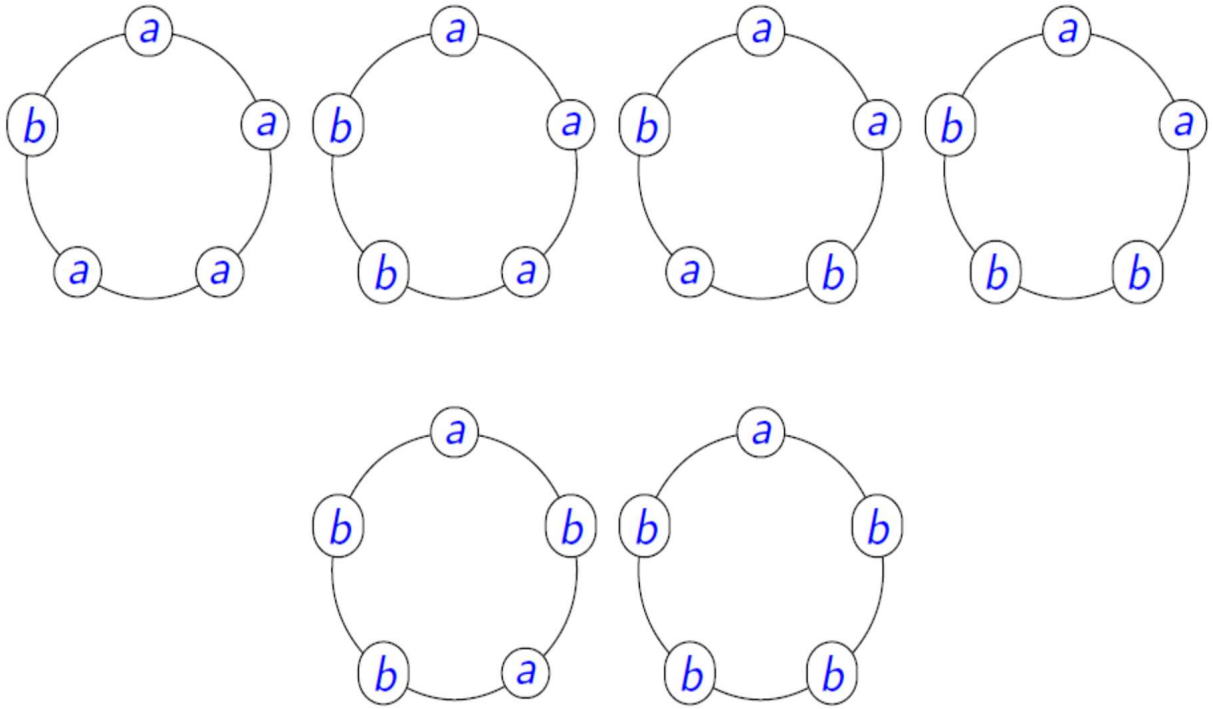
Un necklace è *primitivo* se è la classe di coniugazione di una parola primitiva.

Proposizione

Una parola primitiva di lunghezza n ha n distinte coniugate.

Infatti assumiamo $rs = sr$ con r, s non vuote. Per il secondo teorema di Lyndon-Schützenberger esiste una parola x tale che $r = x^i$, $s = x^j$ e $rs = x^{i+j}$ non è primitiva.

I sei necklace primitivi di lunghezza 5 sull'alfabeto $\{a, b\}$



Definizione

Sia $A = \{a_0, \ldots, a_k\}$ *un alfabeto e sia* $a_0 < a_1 < \ldots < a_k$ *un ordinamento degli elementi di* A. *Siano* $x, y \in A^*$.
Diremo che $x < y$ *rispetto all'***ordine lessicografico** *se* x *e* y *verificano una delle condizioni seguenti:*

1. $y = xz$ *con* $z \in A^+$, *cioè* x *è un prefisso di* y *e* $x \neq y$.

Definizione

Sia $A = \{a_0, \ldots, a_k\}$ *un alfabeto e sia* $a_0 < a_1 < \ldots < a_k$ *un ordinamento degli elementi di* A. *Siano* $x, y \in A^*$.
Diremo che $x < y$ *rispetto all'***ordine lessicografico** *se* x *e* y *verificano una delle condizioni seguenti:*

1. $y = xz$ *con* $z \in A^+$, *cioè* x *è un prefisso di* y *e* $x \neq y$.
2. $x = zax'$, $y = zby'$, *con* $z, x', y' \in A^*$, $a, b \in A$ *e* $a < b$.

Definizione

Sia $A = \{a_0, \ldots, a_k\}$ un alfabeto e sia $a_0 < a_1 < \ldots < a_k$ un ordinamento degli elementi di A. Siano $x, y \in A^*$.
*Diremo che $x < y$ rispetto all'***ordine lessicografico** *se x e y verificano una delle condizioni seguenti:*

1. *$y = xz$ con $z \in A^+$, cioè x è un prefisso di y e $x \neq y$.*
2. *$x = zax'$, $y = zby'$, con $z, x', y' \in A^*$, $a, b \in A$ e $a < b$.*

Le parole in un dizionario sono ordinate in base all'ordine lessicografico.

Definizione

Sia $A = \{a_0, \ldots, a_k\}$ un alfabeto e sia $a_0 < a_1 < \ldots < a_k$ un ordinamento degli elementi di A. Siano $x, y \in A^*$.
*Diremo che $x < y$ rispetto all'***ordine lessicografico** *se x e y verificano una delle condizioni seguenti:*

1. *$y = xz$ con $z \in A^+$, cioè x è un prefisso di y e $x \neq y$.*
2. *$x = zax'$, $y = zby'$, con $z, x', y' \in A^*$, $a, b \in A$ e $a < b$.*

Le parole in un dizionario sono ordinate in base all'ordine lessicografico.

- Esempio. Supponiamo $a < b < \ldots < z$.

Definizione

Sia $A = \{a_0, \ldots, a_k\}$ un alfabeto e sia $a_0 < a_1 < \ldots < a_k$ un ordinamento degli elementi di A. Siano $x, y \in A^*$.
*Diremo che $x < y$ rispetto all'***ordine lessicografico** *se x e y verificano una delle condizioni seguenti:*

1. *$y = xz$ con $z \in A^+$, cioè x è un prefisso di y e $x \neq y$.*
2. *$x = zax'$, $y = zby'$, con $z, x', y' \in A^*$, $a, b \in A$ e $a < b$.*

Le parole in un dizionario sono ordinate in base all'ordine lessicografico.

- Esempio. Supponiamo $a < b < \ldots < z$.

  latte $<$ latteria, castagna $<$ castello.

Nota. Date due qualsiasi parole $x, y \in A^*$, con $x \neq y$, risulta $x < y$ oppure $y < x$.

Proposizione

Siano $x, y \in A^*$. Valgono le seguenti proprietà.

(1) *$x < y$ se e solo se $zx < zy$, per ogni $z \in A^*$.*

(2) *Se $x < y$ e x non è prefisso di y, allora $xu < yv$ per ogni $u, v \in A^*$.*

Ordine lessicografico

(1) Siano $x, y \in A^*$. Allora $x < y$ se e solo se $zx < zy$, per ogni $z \in A^*$.

Ordine lessicografico

(1) Siano $x, y \in A^*$. Allora $x < y$ se e solo se $zx < zy$, per ogni $z \in A^*$.

- Esempio. Supponiamo $a < b$.
  $ab < aba$ e per ogni $z \in A^*$, $zab < zaba$.

Ordine lessicografico

(1) Siano $x, y \in A^*$. Allora $x < y$ se e solo se $zx < zy$, per ogni $z \in A^*$.

- Esempio. Supponiamo $a < b$.
  $ab < aba$ e per ogni $z \in A^*$, $zab < zaba$.

- Esempio. Supponiamo $a < b$.
  $aa < ab$ e per ogni $z \in A^*$, $zaa < zab$.

(2) Se $x < y$ e x non è prefisso di y, allora $xu < yv$ per ogni $u, v \in A^*$.

(2) Se $x < y$ e x non è prefisso di y, allora $xu < yv$ per ogni $u, v \in A^*$.

- Esempio. Supponiamo $a < b$.
  $aa < ab$ e per ogni $u, v \in A^*$, $aau < abv$

Parole di Lyndon

Una parola di Lyndon è una parola primitiva che è la più piccola nella sua classe di coniugazione rispetto all'ordine lessicografico.
Denotiamo con L l'insieme delle parole di Lyndon.
Le "prime" parole di Lyndon su $\{a, b\}$, con $a < b$.

$$a, b$$

$$ab$$

$$aab, abb$$

$$aaab, aabb, abbb$$

$$aaaab, aaabb, aabab, aabbb, ababb, abbbb$$

Parole di Lyndon

Sia $A = \{a, b\}$ con $a < b$.

Le parole a, b, $aaab$, $abbb$, $aabab$ e $aababaabb$ sono parole di Lyndon.

Invece $abab$, aba e $abaab$ non sono parole di Lyndon.

La parola $abab$ non è primitiva, $aab < aba$ e $aabab < abaab$.

Sia $w = banana$.

tutte le coniugate		tutte le coniugate ordinate
banana		abanan
ananab		anaban
nanaba		ananab
anaban	$\rightarrow$	banana
nabana	ordine	nabana
abanan	lessicografico	nanaba

- Sia $w = abraca$
- Ordiniamo lessicograficamente tutte le coniugate di w.

$$
\begin{array}{cccccc}
a & a & b & r & a & c \\
a & b & r & a & c & a \\
a & c & a & a & b & r \\
b & r & a & c & a & a \\
c & a & a & b & r & a \\
r & a & c & a & a & b
\end{array}
$$

Proposizione

Una parola è una parola di Lyndon se e solo se è minore di ogni suo suffisso proprio diverso dalla parola vuota.

Esempio.
$A = \{a, b\}$, con $a < b$.

- $aaaab < b$

Proposizione

Una parola è una parola di Lyndon se e solo se è minore di ogni suo suffisso proprio diverso dalla parola vuota.

Esempio.
$A = \{a, b\}$, con $a < b$.

- $aaaab < b$
- $aaaab < ab$

Proposizione

Una parola è una parola di Lyndon se e solo se è minore di ogni suo suffisso proprio diverso dalla parola vuota.

Esempio.
$A = \{a, b\}$, con $a < b$.

- $aaaab < b$
- $aaaab < ab$
- $aaaab < aab$

Proposizione

Una parola è una parola di Lyndon se e solo se è minore di ogni suo suffisso proprio diverso dalla parola vuota.

Esempio.
$A = \{a, b\}$, con $a < b$.

- $aaaab < b$
- $aaaab < ab$
- $aaaab < aab$
- $aaaab < aaab$

Proposizione

Una parola è una parola di Lyndon se e solo se è minore di ogni suo suffisso proprio diverso dalla parola vuota.

Esempio.
$A = \{a, b\}$, con $a < b$.

- $aaaab < b$
- $aaaab < ab$
- $aaaab < aab$
- $aaaab < aaab$

  $aaaab$ è una parola di Lyndon.

Proposizione

Una parola è una parola di Lyndon se e solo se è minore di ogni suo suffisso proprio diverso dalla parola vuota.

Sufficienza: proviamo che se w è una parola minore di ogni suo suffisso proprio diverso dalla parola vuota, allora w è una parola di Lyndon.

Sia $w = uv$ con u, v non vuote. Poiché $w < v$ e w non è prefisso di v, risulta $w < vu$. Inoltre w è primitiva altrimenti $w = z^n$ con $n > 1$ e quindi $z < w < z$, una contraddizione. Quindi $w \in L$.

Necessità: sia $w \in L$ e sia $w = uv$ con u, v non vuote. Proviamo che $w < v$.
Assumiamo prima che $w = vt$. Poiché w è una parola di Lyndon, $w < tv$.
Quindi $w = uv < tv$ con $|uv| = |tv|$ e quindi $|u| = |t|$. Questo implica $u < t$ e quindi $vu < vt = w$, una contraddizione.

Quindi v non è un prefisso di $w = uv$ e $v \neq w$. Allora $v < uv$ oppure $uv < v$. Ma v non è un prefisso di uv e $v < uv$ implicherebbe $vu < uv = w$, una contraddizione. Concludiamo che $w = uv < v$.

Proposizione

Se $\ell, m \in L$ con $\ell < m$, allora ℓm è una parola di Lyndon.

Esempio. Le "prime" parole di Lyndon su $\{a, b\}$, con $a < b$.

$$a, b \in L, a < b \Rightarrow ab \in L,$$

Proposizione

Se $\ell, m \in L$ con $\ell < m$, allora ℓm è una parola di Lyndon.

Esempio. Le "prime" parole di Lyndon su $\{a, b\}$, con $a < b$.

$a, b \in L, a < b \Rightarrow ab \in L$,

$a, ab \in L, a < ab \Rightarrow aab \in L$,

Proposizione

Se $\ell, m \in L$ con $\ell < m$, allora ℓm è una parola di Lyndon.

Esempio. Le "prime" parole di Lyndon su $\{a, b\}$, con $a < b$.

$a, b \in L, a < b \Rightarrow ab \in L$,

$a, ab \in L, a < ab \Rightarrow aab \in L$,

$ab, b \in L, ab < b \Rightarrow abb \in L$,

Proposizione

Se $\ell, m \in L$ con $\ell < m$, allora ℓm è una parola di Lyndon.

Esempio. Le "prime" parole di Lyndon su $\{a, b\}$, con $a < b$.

$a, b \in L, a < b \Rightarrow ab \in L,$

$a, ab \in L, a < ab \Rightarrow aab \in L,$

$ab, b \in L, ab < b \Rightarrow abb \in L,$

$a < aab \Rightarrow aaab \in L,$
$a < abb \Rightarrow aabb \in L,$
$abb < b \Rightarrow abbb \in L,$

Proposizione

Se $\ell, m \in L$ con $\ell < m$, allora ℓm è una parola di Lyndon.

Esempio. Le "prime" parole di Lyndon su $\{a, b\}$, con $a < b$.

$a, b \in L, a < b \Rightarrow ab \in L,$

$a, ab \in L, a < ab \Rightarrow aab \in L,$

$ab, b \in L, ab < b \Rightarrow abb \in L,$

$a < aab \Rightarrow aaab \in L,$
$a < abb \Rightarrow aabb \in L,$
$abb < b \Rightarrow abbb \in L,$

$a < aaab \Rightarrow aaaab \in L,$
$a < aabb \Rightarrow aaabb \in L,$
$aab < ab \Rightarrow aabab \in L,$
$a < abbb \Rightarrow aabbb \in L,$
$ab < abb \Rightarrow ababb \in L,$
$abbb < b \Rightarrow aabb \in L.$

Proposizione

Se $\ell, m \in L$ con $\ell < m$, allora ℓm è una parola di Lyndon.

Prova (cenni).

Mostriamo prima che $\ell m < m$. Se ℓ è un prefisso di m, allora $m = \ell m'$. La stringa m' è un suffisso proprio e non vuoto di $m \in L$. Quindi $m < m'$ il che implica $\ell m < \ell m' = m$. Altrimenti ℓ non è un prefisso di m e allora $\ell < m$ implica $\ell m < m$.

Sia v un suffisso proprio non vuoto di ℓm. Se v è un suffisso di m, allora $m < v$ e allora $\ell m < m < v$. Altrimenti, abbiamo $v = v' m$. Quindi $\ell < v'$ e allora $\ell m < v' m = v$.

Teorema (Lyndon)

Ogni parola si fattorizza in modo unico come un prodotto di parole di Lyndon in cui ogni fattore è maggiore o uguale al successivo (nonincreasing product).

Quindi ogni parola w può essere scritta in modo unico

$$w = \ell_1 \cdots \ell_m$$

con $\ell_1, \ldots, \ell_m \in L$ e $\ell_1 \geq \ldots \geq \ell_m$.

Esistenza: Poiché le lettere sono in L, ogni parola ha una fattorizzazione in parole di Lyndon.

Consideriamo una fattorizzazione $w = \ell_1 \cdots \ell_m$ in parole di Lyndon e con m minimale. Se $\ell_i < \ell_{i+1}$ per qualche i, allora $w = \ell_1 \cdots \ell_{i-1}(\ell_i \ell_{i+1}) \cdots \ell_m$ è una fattorizzazione in parole di Lyndon poiché $\ell_i \ell_{i+1} \in L$.

Unicità: Assumiamo che $\ell_1 \cdots \ell_m = \ell'_1 \cdots \ell'_{m'}$ con $\ell_i, \ell'_i \in L$, $\ell_1 \geq \ldots \geq \ell_m$ e $\ell'_1 \geq \ldots \geq \ell'_{m'}$. Assumiamo che ℓ_1 è più lunga di ℓ'_1. Quindi $\ell_1 = \ell'_1 \cdots \ell'_i u$ con u prefisso non vuoto di ℓ'_{i+1}. Allora $\ell_1 < u \leq \ell'_{i+1} \leq \ell'_1 < \ell_1$, una contraddizione.

Esempio.
Sia $A = \{a, b, c, d\}$ con $a < b < c < d$.

Sia $w = bbcbacad$. Le stringhe $bbc, b, acad$ sono parole di Lyndon e $w = (bbc)(b)(acad)$. Inoltre $bbc > b > acad$. Quindi la fattorizzazione di Lyndon di w è $(bbc, b, acad)$.

Sia $x = aababb$. Le stringhe aab, abb sono parole di Lyndon e $x = (aab)(abb)$. Siccome $aab < abb$, la stringa x è una parola di Lyndon. Quindi la fattorizzazione di Lyndon di x è (x).

Sia $y = abbaab$. Le stringhe abb, aab sono parole di Lyndon e $y = (abb)(aab)$. Inoltre $abb > aab$. Quindi la fattorizzazione di Lyndon di y è (abb, aab).

Sesquipotenze di parole di Lyndon

Una sesquipotenza di una parola x è una parola della forma $x^n p$ con $n \geq 1$ e con p prefisso proprio di x.

Sia S l'insieme delle sesquipotenze delle parole di Lyndon.

$$S = \{(pq)^n p \mid p \in A^*, q \in A^+, n \geq 1, pq \in L\}$$

Esempio. Sia $A = \{a, b\}$, con $a < b$.
La stringa aab è una parola di Lyndon e $(aab)^2, aab, (aab)^5 aa$ sono sesquipotenze di aab, quindi sono elementi di S.

Sia P l'insieme di tutte le parole che sono diverse dalla parola vuota e che sono prefissi di qualche parola di Lyndon.

$$P = \{w \mid w \in A^+ \text{ e } wA^* \cap L \neq \emptyset\}$$

Se A è finito, come nel nostro caso, esiste una lettera massimale in A. Sia c la lettera massimale in A e sia

$$P' = P \cup \{c^k \mid k \geq 2\}$$

Esempio. Sia $A = \{a, b\}$, con $a < b$.
La stringa $aababb$ è una parola di Lyndon. Quindi $a, aa, aab, aaba, aabab, aababb \in P$. Invece $b^3 \in P'$ e $b^3 \notin P$.

$$P = \{w \mid w \in A^+ \text{ e } wA^* \cap L \neq \emptyset\}$$

$$P' = P \cup \{c^k \mid k \geq 2\}$$

$$S = \{(pq)^n p \mid p \in A^*, q \in A^+, n \geq 1, pq \in L\}$$

Proposizione (J. P. Duval)

$$S = P'$$

La prova usa il lemma seguente.

Lemma

Per ogni parola p e $a \in A$ tali che pa è un prefisso di una potenza di una parola di Lyndon, e per ogni lettera $b > a$, pb è una parola di Lyndon.

Esempio. Sia $A = \{a, b\}$, con $a < b$.
La stringa $aababb$ è una parola di Lyndon e $aababbaa$ è un prefisso di $(aababb)^2$. Cambiando l'ultima lettera in $aababbaa$ con b, otteniamo $aababbab$. Il lemma afferma che $aababbab$ è una parola di Lyndon. Si noti che $aababbab$ si ottiene concatenando due parole di Lyndon $aababb, ab$ tali che $aababb < ab$.

Proposizione (J. P. Duval)

$$S = P'$$

Prova.
Nota che $L \subseteq S$ perché se $x \in L$ allora $x = (pq)^n p$ con $n = 1$, $q = x$ e con p uguale alla parola vuota.
Inoltre $\{c^k \mid k \geq 2\} \subseteq S$ perché $c \in L$.

Sia $S = \{(pq)^n p \mid p \in A^*, q \in A^+, n \geq 1, pq \in L\}$ e
$P = \{w \mid w \in A^+ \text{ e } wA^* \cap L \neq \emptyset\}$.
Dimostriamo che $S \setminus \{c^k \mid k \geq 2\} \subseteq P$.

Sia $w = x^n p$ con $n \geq 1$, p prefisso proprio di x, x parola di Lyndon e $w \neq c^k$, $k \geq 2$. Quindi tra le lettere di cui x è concatenazione esiste una lettera non massimale. Poniamo $x = p'aq$ con a lettera non massimale. La stringa w è un prefisso di $x^{n+1}p'a$. Sia b una lettera tale che $b > a$. Per il lemma, $x^{n+1}p'b$ è in L e ha come prefisso w. Quindi $w \in P$.

Viceversa sia w un prefisso di $x \in L$, con w parola non vuota. Dimostriamo che $w \in S$.

Usiamo l'induzione su $|w|$. Se $|w| = 1$, allora $w \in L$ e sappiamo che $w \in S$.
Assumiamo $|w| > 1$. Poniamo $x = ws$ e $w = va$ con $a \in A$. La parola v è un prefisso di x. Quindi, per ipotesi induttiva, $v = y^n p$ con $y \in L$, $n \geq 1$ e con p prefisso proprio di y. Poniamo $y = pbu$ con $b \in A$. Ora $x = ws = vas = y^n pas$ e y inizia con pb.
Quindi $pb \leq x < pas$ da cui abbiamo $pb \leq pa$ e allora $b \leq a$.
Se $a = b$, allora $w = y^n pb$ è una sesquipotenza di y. Se $b < a$, allora $w = va = y^n pa$ e $y^n pb = (pbu)^n pb$ è un prefisso della potenza y^{n+1} della parola di Lyndon y. Per il lemma precedente $w = va = y^n pa \in L$.