STRUMENTI FORMALI PER LA BIOINFORMATICA

Combinatoria delle parole (Parte 2)

Definizione

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

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

Definizione

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

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

Definizione

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

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

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

Definizione

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

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

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

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

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 non è primitiva perché $abab = (ab)^2$.

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

Nota che la parola vuota non è primitiva.

$abab$ non è primitiva perché $abab = (ab)^2$.

aba, abb sono parole primitive.

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 $w = banana$.

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

Definizione

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.

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

Le parole *abab*, *aba* and *abaab* **non** sono parole di Lyndon.

La parola *abab* non è primitiva. Per quanto riguarda *aba* e *abaab*, non sono parole di Lyndon perché $aab < aba$ e $aabab < abaab$.

Proposizione

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

Sia $A = \{a, b\}$ con $a < b$. Le parole a, b, $aaab$ e $abbb$ sono parole di Lyndon.

Sia $A = \{a, b\}$ con $a < b$. Le parole a, b, $aaab$ e $abbb$ sono parole di Lyndon.

Le parole a e b non hanno suffissi propri diversi dalla parola vuota.

Sia $A = \{a, b\}$ con $a < b$. Le parole a, b, $aaab$ e $abbb$ sono parole di Lyndon.

Le parole a e b non hanno suffissi propri diversi dalla parola vuota.

I suffissi propri diversi dalla parola vuota di $aaab$ sono b, ab, aab e $aaab < b$, $aaab < ab$, $aaab < aab$.

Sia $A = \{a, b\}$ con $a < b$. Le parole a, b, $aaab$ e $abbb$ sono parole di Lyndon.

Le parole a e b non hanno suffissi propri diversi dalla parola vuota.

I suffissi propri diversi dalla parola vuota di $aaab$ sono b, ab, aab e $aaab < b$, $aaab < ab$, $aaab < aab$.

I suffissi propri diversi dalla parola vuota di $abbb$ sono b, bb, bbb e $abbb < b$, $abbb < bb$, $abbb < bbb$.

Sia $A = \{a, b\}$ con $a < b$. Le parole *aabab* e *aababaabb* sono parole di Lyndon.

Sia $A = \{a, b\}$ con $a < b$. Le parole *aabab* e *aababaabb* sono parole di Lyndon.

I suffissi propri diversi dalla parola vuota di *aabab* sono *b*, *ab*, *bab*, *abab* e $aabab < b$, $aabab < ab$, $aabab < bab$, $aabab < abab$.

Sia $A = \{a, b\}$ con $a < b$. Le parole $aabab$ e $aababaabb$ sono parole di Lyndon.

I suffissi propri diversi dalla parola vuota di $aabab$ sono b, ab, bab, $abab$ e $aabab < b$, $aabab < ab$, $aabab < bab$, $aabab < abab$.

I suffissi propri diversi dalla parola vuota di $aababaabb$ sono b, bb, abb, $aabb$, $baabb$, $abaabb$, $babaabb$, $ababaabb$ e $aababaabb < b$, $aababaabb < bb$, $aababaabb < abb$, $aababaabb < aabb$, $aababaabb < baabb$, $aababaabb < abaabb$, $aababaabb < babaabb$, $aababaabb < ababaabb$.

L'overlap di due stringhe x e y è il più lungo suffisso proprio di x che è anche un prefisso proprio di y.

Esempio.
Siano $x = abacaba$ e $y = acabaca$.

La stringa a è un suffisso proprio di x che è anche un prefisso proprio di y. Ma anche $acaba$ è un suffisso proprio di x che è un prefisso proprio di y e $|acaba| = 5 > 1 = |a|$. Inoltre x non ha suffissi propri più lunghi che siano anche prefissi propri di y.

Quindi l'overlap di $abacaba$ e $acabaca$ è la stringa $acaba$ di lunghezza 5.

L'overlap di due stringhe x e y è il più lungo suffisso proprio di x che è anche un prefisso proprio di y.

Qual è l'overlap di *bbb* e *aaa*?

L'overlap di x e y è uguale all'overlap di y e x?

L'overlap di x e y ha la stessa lunghezza dell'overlap di y e x?

L'overlap di due stringhe x e y è il più lungo suffisso proprio di x che è anche un prefisso proprio di y.

Qual è l'overlap di *bbb* e *aaa*?

L'overlap di *bbb* e *aaa* è la parola vuota.

L'overlap di x e y è uguale all'overlap di y e x?

No, l'overlap di *abacaba* e *acabaca* è *acaba* mentre l'overlap di *acabaca* e *abacaba* è *abaca*.

L'overlap di x e y ha la stessa lunghezza dell'overlap di y e x?

No, l'overlap di *ca* e *ab* è *a* che ha lunghezza 1 mentre l'overlap di *ab* e *ca* è la parola vuota che ha lunghezza 0.

Il bordo di una parola non vuota w è l'overlap di w e sé stessa.

Quindi è la più lunga parola u che è sia un prefisso proprio che un suffisso proprio di w.

Esempi:

Il bordo di *amaca* è *a*, il bordo di *barba* è *ba*.

Il bordo di *ababab* è *abab*.

Il bordo di *aaa* è *aa*, il bordo di *ca* è la parola vuota.

Un *bordo* di w è una parola che è sia un prefisso proprio che un suffisso proprio di w.

La parola aaa ha come bordi la parola vuota, a e aa.

La parola ca ha come bordo solo la parola vuota.

Una parola $w \in \Sigma^+$ è *bordered* se ha un bordo non vuoto. Altrimenti, w è *unbordered*.

Le stringhe $amaca$, $barba$, $ababab$, aaa sono bordered. La stringa ca è unbordered.

Proposizione

Ogni parola di Lyndon è unbordered.

La prova di questo risultato è un semplice corollario della proposizione che caratterizza le parole di Lyndon come quelle parole minori di ogni loro suffisso proprio diverso dalla parola vuota.

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 $\ell_1 \geq \ldots \geq \ell_m$.
La sequenza $\mathrm{CFL}(w) = (\ell_1, \ldots, \ell_m)$ è chiamata la *decomposizione di Lyndon* (o *fattorizzazione di Lyndon*) di w.
La notazione $\mathrm{CFL}(w)$ è dovuta al fatto che il teorema della fattorizzazione è in genere attribuito a Chen, Fox e Lyndon.
Le parole di Lyndon $\ell_1, \ldots, \ell_m$ sono anche chiamate *fattori di Lyndon* di w.

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 $\mathrm{CFL}(w) = (bbc, b, acad)$.

Sia $x = aababb$. Le stringhe aab, abb sono parole di Lyndon e $x = (aab)(abb)$. Ma $aab < abb$.
La stringa x è una parola di Lyndon. Quindi $\mathrm{CFL}(x) = (x) = (aababb)$.

Sia $y = abbaab$. Le stringhe abb, aab sono parole di Lyndon e $y = (abb)(aab)$.
Inoltre $abb > aab$. Quindi $\mathrm{CFL}(y) = (abb, aab)$.

Proposizione

Sia $w \in \Sigma^+$, *sia* ℓ_1 *il suo più lungo prefisso che è una parola di Lyndon e sia* w' *tale che* $w = \ell_1 w'$. *Se* $w' = 1$, *allora* $\mathsf{CFL}(w) = (\ell_1)$. *Se* $w' \neq 1$, *allora* $\mathsf{CFL}(w) = (\ell_1, \mathsf{CFL}(w'))$.

$\mathsf{CFL}(w)$ è calcolabile in tempo lineare.

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

La stringa $x = aababb$ è una parola di Lyndon, quindi $\mathrm{CFL}(x) = (x) = (aababb)$.

Sia $y = abbaab$. Le stringhe a, ab, abb sono parole di Lyndon mentre $abba$, $abbaa$, $abbaab$ non sono parole di Lyndon. Quindi $\mathrm{CFL}(y) = (abb, \mathrm{CFL}(aab))$. Siccome aab è una parola di Lyndon, abbiamo $\mathrm{CFL}(y) = (abb, aab)$.

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

Sia $w = bbcbacad$. La stringa b è una parola di Lyndon, bb non è una parola di Lyndon mentre bbc lo è perché $bbc < c$, $bbc < bc$. Poiché $bbcb$, $bbcba$, $bbcbac$, $bbcbaca$, $bbcbacad$ non sono parole di Lyndon, abbiamo $\mathrm{CFL}(w) = (bbc, \mathrm{CFL}(bacad))$.

La stringa b è una parola di Lyndon, mentre ba, bac, $baca$, $bacad$ non sono parole di Lyndon. Quindi
$\mathrm{CFL}(w) = (bbc, b, \mathrm{CFL}(acad))$. Siccome $acad$ è una parola di Lyndon, abbiamo $\mathrm{CFL}(w) = (bbc, b, acad)$.

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

Sia $z = bbcbcacad$. La stringa b è una parola di Lyndon, bb non è una parola di Lyndon mentre bbc lo è perché $bbc < c$, $bbc < bc$. La stringa $bbcb$ non è una parola di Lyndon ma $bbcbc$ lo è. Poiché $bbcbca$, $bbcbcac$, $bbcbcaca$, $bbcbcacad$ non sono parole di Lyndon, abbiamo $\mathrm{CFL}(z) = (bbcbc, \mathrm{CFL}(acad))$. Siccome $acad$ è una parola di Lyndon, abbiamo $\mathrm{CFL}(z) = (bbcbc, acad)$.

Varianti e applicazioni

- Nyldon words [Charlier,Philibert,Stipulanti, 2018; S. Gard, 2021]
- Generalized Lyndon words [Dolce, Restivo, Reutenauer, 2018]
- Inverse Lyndon Words [Bonizzoni, De Felice, Zaccagnino, Zizza, 2018 e 2021]

Definizione

Una parola w è una parola inversa di Lyndon (inverse Lyndon word) se $w > s$, per ogni s, con s suffisso proprio e diverso dalla parola vuota di w.

Esempio.

Sia $\Sigma = \{a, b\}$ con $a < b$. La parola *aaba* non è una parola inversa di Lyndon poiché $aaba < ba$. Analogamente, $aabba < ba$ e quindi *aabba* non è una parola inversa di Lyndon.

Definizione

Una parola w è una parola inversa di Lyndon (inverse Lyndon word) se $w > s$, per ogni s, con s suffisso proprio e diverso dalla parola vuota di w.

Esempio.

Le parole a, b e $aaaaa$ sono parole inverse di Lyndon su $\{a, b\}$, con $a < b$. Tutti i suffissi propri di $aaaaa$ sono anche prefissi propri di $aaaaa$. Quindi, per ogni s, con s suffisso proprio e diverso dalla parola vuota di $aaaaa$, risulta $aaaaa > s$.

Definizione

Una parola w è una parola inversa di Lyndon (inverse Lyndon word) se $w > s$, per ogni s, con s suffisso proprio e diverso dalla parola vuota di w.

Esempio Le parole *bbba* e *baaab* sono parole inverse di Lyndon su $\{a, b\}$, con $a < b$. I suffissi propri e diversi dalla parola vuota di *bbba* sono a, ba, bba. Quindi, per ogni s, con s suffisso proprio e diverso dalla parola vuota di *bbba*, risulta $bbba > s$.

Analogamente, i suffissi propri e diversi dalla parola vuota di *baaab* sono b, ab, aab, $aaab$. Quindi, per ogni s, con s suffisso proprio e diverso dalla parola vuota di *baaab*, risulta $baaab > s$.

Definizione

Una parola w è una parola inversa di Lyndon (inverse Lyndon word) se $w > s$, per ogni s, con s suffisso proprio e diverso dalla parola vuota di w.

Esempio Le parole *bbaba* e *bbababbaa* sono parole inverse di Lyndon su $\{a, b\}$, con $a < b$.

Lemma

Ogni prefisso non vuoto di una parola inversa di Lyndon è una parola inversa di Lyndon.

Esempio Le parole *bbaba* e *bbababbaa* sono parole inverse di Lyndon su $\{a, b\}$, con $a < b$. La prima è un prefisso della seconda. Tutti i prefissi non vuoti di *bbababbaa* sono parole inverse di Lyndon.

Definizione

Sia $(\Sigma, <)$ *un alfabeto totalmente ordinato. Sia* $<_{in}$ *l'inversa of* $<$*, definita da*

$$\forall a, b \in \Sigma \quad b <_{in} a \Leftrightarrow a < b$$

L'ordine lessicografico inverso, denotato $<_{in}$*, su* $(\Sigma^*, <)$ *è l'ordine lessicografico su* $(\Sigma^*, <_{in})$*.*

Esempio Sia $\Sigma = \{a, b, c, d\}$ con $a < b < c < d$. Allora $dab < dabd$ e $dabda < dac$. Abbiamo $d <_{in} c <_{in} b <_{in} a$. Quindi $dab <_{in} dabd$ e $dac <_{in} dabda$.

Anti-Lyndon words e inverse Lyndon words

Sia L_{in} l'insieme delle parole di Lyndon in Σ^* rispetto all'ordine lessicografico inverso.
Una stringa $w \in L_{in}$ è chiamata una *parola anti-Lyndon*.

Proposizione

Una parola $w \in \Sigma^+$ è una parola inversa di Lyndon se e solo se è un prefisso non vuoto di una potenza di una parola anti-Lyndon.

Quindi le parole inverse di Lyndon sono le sesquipotenze delle parole anti-Lyndon.

Sia L_{in} l'insieme delle parole di Lyndon in Σ^* rispetto all'ordine lessicografico inverso.
Una stringa $w \in L_{in}$ è chiamata una *parola anti-Lyndon*.

Proposizione

Una parola $w \in \Sigma^+$ è una parola anti-Lyndon se e solo se è una parola inversa di Lyndon ed è unbordered.

Quindi una parola anti-Lyndon è anche inversa di Lyndon. Ma non è vero il contrario. Ad esempio *aaa* è una parola inversa di Lyndon sull'alfabeto $\{a\}$ ma non è una parola anti-Lyndon.

Proposizione 1
Una parola $w \in \Sigma^+$ è una parola anti-Lyndon se e solo se è una parola inversa di Lyndon ed è unbordered.

Proposizione
Una parola $w \in \Sigma^+$ è una parola inversa di Lyndon se e solo se è un prefisso non vuoto di una potenza di una parola anti-Lyndon.

Esempio.
Le stringhe a, b, $bbba$, $bbaba$ e $bbababbaa$ sono parole inverse di Lyndon su $\{a, b\}$, con $a < b$.
Inoltre sono unbordered. Per la Proposizione 1 sono parole anti-Lyndon.

Proposizione

Una parola $w \in \Sigma^+$ è una parola inversa di Lyndon se e solo se è un prefisso non vuoto di una potenza di una parola anti-Lyndon.

Esempio.
Le stringhe $aaaaa$ e $baaab$ sono parole inverse di Lyndon su $\{a, b\}$, con $a < b$.

La stringa $aaaaa$ è una potenza della parola anti-Lyndon a.

Inoltre la stringa $baaa$ è una parola anti-Lyndon. Quindi $baaab$ è un prefisso non vuoto della potenza $(baaa)^2$ della parola anti-Lyndon $baaa$.

Siano $x, y \in \Sigma^+$ con Σ alfabeto ordinato. Scriveremo $x \ll y$ se $x < y$ ma x non è prefisso di y. In questo caso $x = zax'$, $y = zby'$, con $z, x', y' \in \Sigma^*$, $a, b \in \Sigma$ e $a < b$.

Definizione

Una fattorizzazione inversa di Lyndon (inverse Lyndon factorization) di $w \in \Sigma^+$ è una sequenza $(f_1, f_2, \ldots, f_n)$ di parole tali che

(1) $w = f_1 \cdots f_n$,

(2) *per ogni $j \in \{1, \ldots, n\}$, la parola f_j è una parola inversa di Lyndon,*

(3) $f_1 \ll f_2 \ll \cdots \ll f_n$

Purtroppo data una stringa w non sempre esiste una sola fattorizzazione inversa di Lyndon di w.

Purtroppo data una stringa w non sempre esiste una sola fattorizzazione inversa di Lyndon di w.

Ad esempio, con $a < b < c < d$, per $w = dabdadacddbdc$ abbiamo $dab \ll dadacd \ll db \ll dc$ e $dabda \ll dac \ll ddbdc$. Inoltre ogni stringa in $\{dab, dadacd, db, dc, dabda, dac, ddbdc\}$ è una parola inversa di Lyndon.

Purtroppo data una stringa w non sempre esiste una sola fattorizzazione inversa di Lyndon di w.

Ad esempio, con $a < b < c < d$, per $w = dabdadacddbdc$ abbiamo $dab \ll dadacd \ll db \ll dc$ e $dabda \ll dac \ll ddbdc$. Inoltre ogni stringa in $\{dab, dadacd, db, dc, dabda, dac, ddbdc\}$ è una parola inversa di Lyndon.

Quindi $(dab, dadacd, db, dc)$ e $(dabda, dac, ddbdc)$ sono due fattorizzazioni inverse di Lyndon di $w = dabdadacddbdc$. La sequenza $(dab, dadac, ddbdc)$ è ancora un'altra fattorizzazione inversa di Lyndon di $w = dabdadacddbdc$.

Fattorizzazione inversa di Lyndon canonica (ICFL)

- La fattorizzazione inversa di Lyndon canonica (ICFL) di una stringa w è **unica**

  $$\mathrm{ICFL}(dabdadacddbdc) = (dab, dadac, ddbdc)$$

  ⋄ come CFL

- può essere calcolata in **tempo lineare**

  ⋄ come CFL

Qual è la definizione di $\mathrm{ICFL}(w)$?

Definizione (bounded right extension)

Sia $w \in \Sigma^+$*, sia* p *una parola inversa di Lyndon che è un prefisso proprio non vuoto di* $w = pv$*.*
La **bounded right extension** $\overline{p}$ *di* p *(relativamente a* w*), se esiste, è un prefisso non vuoto di* v *tale che:*

(1) $\overline{p}$ *è una parola inversa di Lyndon,*

(2) pz' *è una parola inversa di Lyndon, per ogni prefisso proprio non vuoto* z' *di* $\overline{p}$*,*

(3) $p\overline{p}$ *non è una parola inversa di Lyndon,*

(4) $p \ll \overline{p}$*.*

Bounded right extension

Data una parola w, si può dimostrare che o nessun prefisso di w ha una bounded right extension oppure questo prefisso è unico (e la coppia $(p, \overline{p})$ è unica).

Il primo caso si verifica se e solo se w è una parola inversa di Lyndon.

La coppia $(p, \overline{p})$ è chiamata la coppia canonica associata a w.

Proposizione

Sia $w \in \Sigma^+$ una parola che non è una parola inversa di Lyndon. Una coppia di parole $(p, \overline{p})$ è la coppia canonica associata a w se e solo se sono soddisfatte le seguenti condizioni.

(1) *$z = p\overline{p}$ è il più corto prefisso non vuoto di w che non è una parola inversa di Lyndon.*

(2) *$p = ras$ e $\overline{p} = rb$, con $r, s \in \Sigma^*$, $a, b \in \Sigma$ dove r è il più corto prefisso di $p\overline{p}$ tale che $p\overline{p} = rasrb$, con $a < b$.*

(3) *$\overline{p}$ è una parola inversa di Lyndon.*

Esempio
Sia $\Sigma = \{a, b\}$ con $a < b$. La stringa $w = babaaabb = yb$ non è una parola inversa di Lyndon. Invece $babaaab = y$ è una parola inversa di Lyndon, quindi il più corto prefisso non vuoto di w che non è una parola inversa di Lyndon è w. Conseguentemente $w = p\overline{p}$.
Cerchiamo l'unica coppia $(p, \overline{p})$. Siccome la stringa p inizia con b, la stringa $\overline{p}$ non può iniziare con a, altrimenti non avremmo $p \ll \overline{p}$. Quindi i possibili prefissi candidati per p sono ba, $babaaa$ e $babaaab$. Il primo caso non può verificarsi perché implicherebbe $\overline{p} = baaabb$ ma $baaabb$ non è una parola inversa di Lyndon. Il terzo caso non può verificarsi perché implicherebbe $\overline{p} = b$ cioè $(p, \overline{p}) = (babaaab, b)$ e non avremmo $p \ll \overline{p}$.
Infine la coppia $(p, \overline{p}) = (babaaa, bb)$ è la coppia richiesta.

Definizione

Sia $w \in \Sigma^+$.

(Passo Base) *Se* w *è una parola inversa di Lyndon allora* $\mathrm{ICFL}(w) = (w)$.

(Passo Ricorsivo) *Se* w *non è una parola inversa di Lyndon,*

1. *trova il più corto prefisso non vuoto* x *di* w *che non è una parola inversa di Lyndon, quindi* $x = rasrb = p\overline{p}$.
2. *Trova* $(p, \overline{p})$.
3. *Sia* $w = pv$ *e sia* $\mathrm{ICFL}(v) = (m'_1, \ldots, m'_k)$

$$\mathrm{ICFL}(w) = \begin{cases} (p, \mathrm{ICFL}(v)) & \text{se } \overline{p} = rb \text{ è prefisso di } m'_1 \\ (pm'_1, m'_2, \ldots, m'_k) & \text{se } m'_1 \text{ è prefisso di } r \end{cases}$$

1. Qual è il più corto prefisso non vuoto *rasrb* di $w = dabdabdadac$ che non è una parola inversa di Lyndon?

dabdabdadac

2. $p = dabdab, \overline{p} = dad$

2. $p = dabdab, \overline{p} = dad$
3. $\mathrm{ICFL}(dadac) = dadac$
4. dad è prefisso di $dadac$, allora $\mathrm{ICFL}(w) = (dabdab, dadac)$.

Fattorizzazione inversa di Lyndon canonica (ICFL)

La definizione di ICFL(w) è molto complicata...

Fattorizzazione inversa di Lyndon canonica (ICFL)

La definizione di $\mathrm{ICFL}(w)$ è molto complicata...

Come semplificarla?

Fattorizzazione inversa di Lyndon canonica (ICFL)

La definizione di ICFL(w) è molto complicata...

Come semplificarla?

Studiandone le proprietà.

Fattorizzazione inversa di Lyndon canonica (ICFL)

Per ogni $w \in \Sigma^+$, denotiamo con $\mathrm{CFL}_{in}(w)$ la fattorizzazione di Lyndon di w rispetto all'ordine lessicografico inverso.
Costruzione alternativa diretta di $\mathrm{ICFL}(w)$ da $\mathrm{CFL}_{in}(w)$?

Fattorizzazione inversa di Lyndon canonica (ICFL)

Per ogni $w \in \Sigma^+$, denotiamo con $\mathrm{CFL}_{in}(w)$ la fattorizzazione di Lyndon di w rispetto all'ordine lessicografico inverso.
Costruzione alternativa diretta di $\mathrm{ICFL}(w)$ da $\mathrm{CFL}_{in}(w)$?

Recentemente ottenuta:

Fattorizzazione inversa di Lyndon canonica (ICFL)

Per ogni $w \in \Sigma^+$, denotiamo con $\mathrm{CFL}_{in}(w)$ la fattorizzazione di Lyndon di w rispetto all'ordine lessicografico inverso.
Costruzione alternativa diretta di $\mathrm{ICFL}(w)$ da $\mathrm{CFL}_{in}(w)$?

Recentemente ottenuta:

Paola Bonizzoni, Clelia De Felice, Brian Riccardi, Rocco Zaccagnino, Rosalba Zizza, *Unveiling the Connection Between the Lyndon Factorization and the Canonical Inverse Lyndon Factorization via a Border Property*, **49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)**, Leibniz International Proceedings in Informatics (LIPIcs), 2024

Proposizione

Per ogni $w \in \Sigma^+$, $\mathsf{ICFL}(w)$ *è un grouping di* $\mathsf{CFL}_{in}(w)$.

Proposizione

Per ogni $w \in \Sigma^+$, $\mathsf{ICFL}(w)$ *è un grouping di* $\mathsf{CFL}_{in}(w)$.

Sfortunatamente, $\mathsf{ICFL}(w)$ non è sempre l'unico grouping di $\mathsf{CFL}_{in}(w)$.

Fattorizzazione inversa di Lyndon canonica (ICFL)

Sia $\Sigma = \{a, b, c, d\}$, $a < b < c < d$, and $w = dabadabdabdadac$.

Fattorizzazione inversa di Lyndon canonica (ICFL)

Sia $\Sigma = \{a, b, c, d\}$, $a < b < c < d$, and
$w = dabadabdabdadac$.

Risulta $\mathrm{CFL}_{in}(w) = (daba, dab, dab, dadac)$ e
$\mathrm{ICFL}(w) = (daba, dabdab, dadac)$.
$\mathrm{ICFL}(w)$ è un grouping di $\mathrm{CFL}_{in}(w)$.

Fattorizzazione inversa di Lyndon canonica (ICFL)

Sia $\Sigma = \{a, b, c, d\}$, $a < b < c < d$, and
$w = dabadabdabdadac$.

Risulta $\mathrm{CFL}_{in}(w) = (daba, dab, dab, dadac)$ e
$\mathrm{ICFL}(w) = (daba, dabdab, dadac)$.
$\mathrm{ICFL}(w)$ è un grouping di $\mathrm{CFL}_{in}(w)$.

Invece $(dabadab, dabda, dac)$ non è un grouping di $\mathrm{CFL}_{in}(w)$.

Fattorizzazione inversa di Lyndon canonica (ICFL)

Sia $\Sigma = \{a, b, c, d\}$, $a < b < c < d$, and
$w = dabadabdabdadac$.

Risulta $\mathrm{CFL}_{in}(w) = (daba, dab, dab, dadac)$ e
$\mathrm{ICFL}(w) = (daba, dabdab, dadac)$.
$\mathrm{ICFL}(w)$ è un grouping di $\mathrm{CFL}_{in}(w)$.

Invece $(dabadab, dabda, dac)$ non è un grouping di $\mathrm{CFL}_{in}(w)$.

Sia $y = dabadabdabdabdadac$.
Risulta $\mathrm{CFL}_{in}(y) = (daba, dab, dab, dab, dadac)$ e
$\mathrm{ICFL}(y) = (daba, (dab)^3, dadac)$.

Fattorizzazione inversa di Lyndon canonica (ICFL)

Sia $\Sigma = \{a, b, c, d\}$, $a < b < c < d$, and
$w = dabadabdabdadac$.

Risulta $\mathrm{CFL}_{in}(w) = (daba, dab, dab, dadac)$ e
$\mathrm{ICFL}(w) = (daba, dabdab, dadac)$.
$\mathrm{ICFL}(w)$ è un grouping di $\mathrm{CFL}_{in}(w)$.

Invece $(dabadab, dabda, dac)$ non è un grouping di $\mathrm{CFL}_{in}(w)$.

Sia $y = dabadabdabdabdadac$.
Risulta $\mathrm{CFL}_{in}(y) = (daba, dab, dab, dab, dadac)$ e
$\mathrm{ICFL}(y) = (daba, (dab)^3, dadac)$.

$\mathrm{ICFL}(y)$ è un grouping di $\mathrm{CFL}_{in}(y)$.

Fattorizzazione inversa di Lyndon canonica (ICFL)

Sia $\Sigma = \{a, b, c, d\}$, $a < b < c < d$, and
$w = dabadabdabdadac$.

Risulta $\mathrm{CFL}_{in}(w) = (daba, dab, dab, dadac)$ e
$\mathrm{ICFL}(w) = (daba, dabdab, dadac)$.
$\mathrm{ICFL}(w)$ è un grouping di $\mathrm{CFL}_{in}(w)$.

Invece $(dabadab, dabda, dac)$ non è un grouping di $\mathrm{CFL}_{in}(w)$.

Sia $y = dabadabdabdabdadac$.
Risulta $\mathrm{CFL}_{in}(y) = (daba, dab, dab, dab, dadac)$ e
$\mathrm{ICFL}(y) = (daba, (dab)^3, dadac)$.

$\mathrm{ICFL}(y)$ è un grouping di $\mathrm{CFL}_{in}(y)$.

La fattorizzazione inversa di Lyndon $(dabadab, (dab)^2, dadac)$
è un altro grouping di $\mathrm{CFL}_{in}(y)$.

Fattorizzazione inversa di Lyndon canonica (ICFL)

Ma ci sono altre proprietà rispetto alle quali ICFL è unica.

Ma ci sono altre proprietà rispetto alle quali ICFL è unica.

Per ogni $w \in \Sigma^+$, ICFL(w) è l'unica fattorizzazione inversa di Lyndon di w che ha la proprietà del bordo.

Ma ci sono altre proprietà rispetto alle quali ICFL è unica.

Per ogni $w \in \Sigma^+$, ICFL(w) è l'unica fattorizzazione inversa di Lyndon di w che ha la proprietà del bordo.

Definizione (Border property)

Sia $w \in \Sigma^+$. Una fattorizzazione $(m_1, \ldots, m_k)$ di w ha la border property *se ogni bordo non vuoto z di m_i non è un prefisso di m_{i+1}, $1 \leq i \leq k-1$.*

Ma ci sono altre proprietà rispetto alle quali ICFL è unica.

Per ogni $w \in \Sigma^+$, ICFL(w) è l'unica fattorizzazione inversa di Lyndon di w che ha la proprietà del bordo.

Definizione (Border property)

Sia $w \in \Sigma^+$. *Una fattorizzazione* $(m_1, \ldots, m_k)$ *di* w ha la border property *se ogni bordo non vuoto* z *di* m_i *non è un prefisso di* m_{i+1}, $1 \leq i \leq k-1$.

Permette una costruzione alternativa diretta di ICFL(w) da CFL$_{in}$(w).

Fattorizzazione inversa di Lyndon canonica (ICFL)

Risultato recentemente ottenuto: costruzione alternativa diretta di ICFL(w) senza passare da $\mathrm{CFL}_{in}(w)$.