Lyndon 分解における逆問題

九州大学
中島　祐人, 岡部　駿志, 井　智弘,
稲永　俊介, 坂内　英夫, 竹田　正幸

Lyndon 文字列 [Lyndon, 1954]

定義

文字列 w が Lyndon 文字列であるとは，w を巡回シフトしたすべての文字列より w の辞書式順序が小さいことである．

w の巡回シフト

w = a a b a b b ＜ b a b b a a

Lyndon 文字列

Lyndon 分解 [Chen, Fox, Lyndon, 1958]

定義

文字列の列 $u_1^{p_1},\ldots,u_m^{p_m}$ が w の Lyndon 分解 LF_w であるとは，次の条件を満たすことである：

$u_1 > \ldots > u_m$ が Lyndon 文字列，$p_i \geq 1\ (1 \leq i \leq m)$，かつ $w = u_1^{p_1}\ldots u_m^{p_m}$.

$$w = \underset{u_1}{\mathrm{a\,b\,c}}\,|\,\underset{u_2}{\mathrm{a\,b\,b}}\,|\,\underset{u_2}{\mathrm{a\,b\,b}}\,|\,\underset{u_3}{\mathrm{a\,a\,b\,c}}\,|\,\underset{u_4}{\mathrm{a}}\,|\,\underset{u_4}{\mathrm{a}}\,|\,\underset{u_4}{\mathrm{a}}$$

$$LF_w = \underset{u_1^1}{(\mathrm{abc})^1}\,|\,\underset{u_2^2}{(\mathrm{abb})^2}\,|\,\underset{u_3^1}{(\mathrm{aabc})^1}\,|\,\underset{u_4^3}{(\mathrm{a})^3}$$

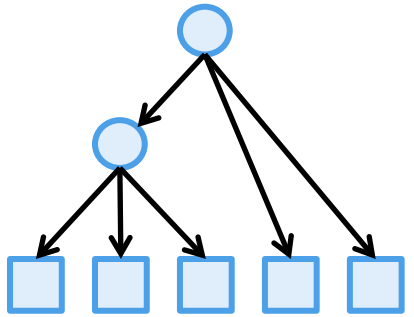
※ 分解の各要素を項（factor）とよぶ.

任意の文字列 w に対して，LF_w は一意に決まる.

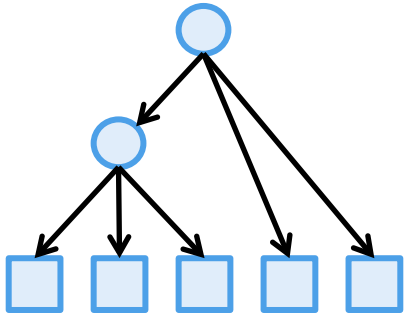
文字列構造の逆問題

例　接尾辞に対する問題

- 順問題：文字列から接尾辞木を求める.

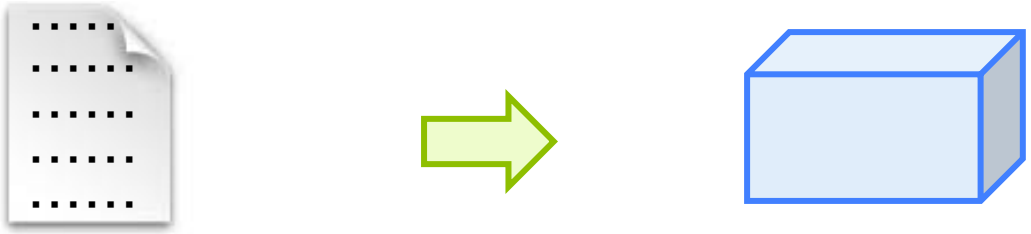
abaac ⇒ 

- 逆問題：接尾辞木から文字列を求める.

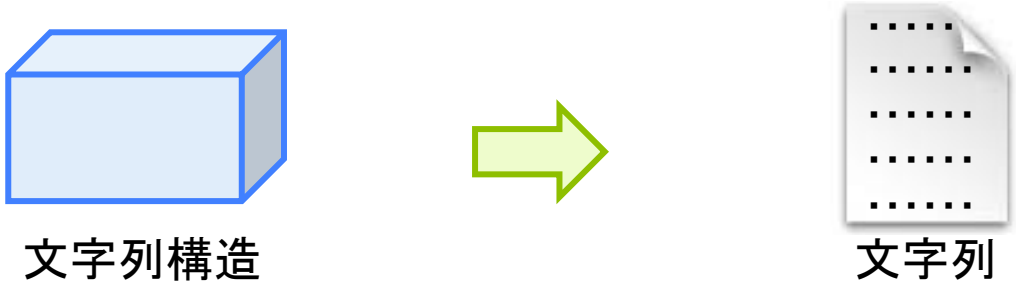
 ⇒ abaac

文字列構造の逆問題

- 順問題 : 文字列から文字列構造を求める.



- 逆問題 : 文字列構造から文字列を求める.



文字列構造の組み合わせ的性質を明らかにする.

問題の入力について

- Lyndon 分解の逆問題における入力を以下のように定義.

$$S = (s_1, p_1), \ldots, (s_m, p_m) \qquad s_i, p_i : \text{自然数}$$

  - i 番目の組は, Lyndon 分解の i 番目の factor に対応.
  - s_i : i 番目の factor の Lyndon 文字列の長さ.
  - p_i : i 番目の factor の Lyndon 文字列の繰り返しの数.

例

$$S = (3, 1), (3, 2), (4, 1), (1, 3)$$

$$(\mathrm{abc})^1(\mathrm{abb})^2(\mathrm{aabc})^1\ (\mathrm{a})^3$$

主結果1

問題 1

自然数の組の列 $S = ((s_1, p_1), \ldots, (s_m, p_m))$ があたえられたとき，$LF_w = u_1^{p_1}, \ldots, u_m^{p_m} (|u_i| = s_i)$ となるアルファベットサイズ最小の w を求めよ.

定理 1

問題 1 は $O(n)$ 時間で解ける.

n : 文字列の長さ

本結果の意義

素朴な手法では，指数時間を要するため効率的である.

定理1のアイデア

- Lyndon word u_{i-1} から u_i を計算する.
  ← u_i は u_{i-1} より辞書式順序が小さい Lyndon word のうち
  辞書式順序が最大の Lyndon word.

$u_1 \succ \cdots \succ u_{i-1} \succ u_i \succ \cdots \succ u_m$

左から順に計算する.（u_1 は辞書式順序最大の Lyndon word とする.）

例　$S = (3, 1), (3, 2), (4, 1), (1, 3)$

$\Sigma = \{z, y, \ldots\}$ と仮定.

yzz ≻ yyz ≻ yyyz ≻ y

u_i を $O(s_i)$ 時間で計算できる.

定理1のアイデア

補題

任意の文字列 A について，A の i 文字目を辞書式順序が1つ小さい文字に置換，i+1 文字目以降の文字を辞書式順序が最大の文字に置換した文字列 B が Lyndon word となる最大の i とすると，B は，A より辞書式順序が小さい Lyndon word のうち辞書式順序が最大の文字列である.

例

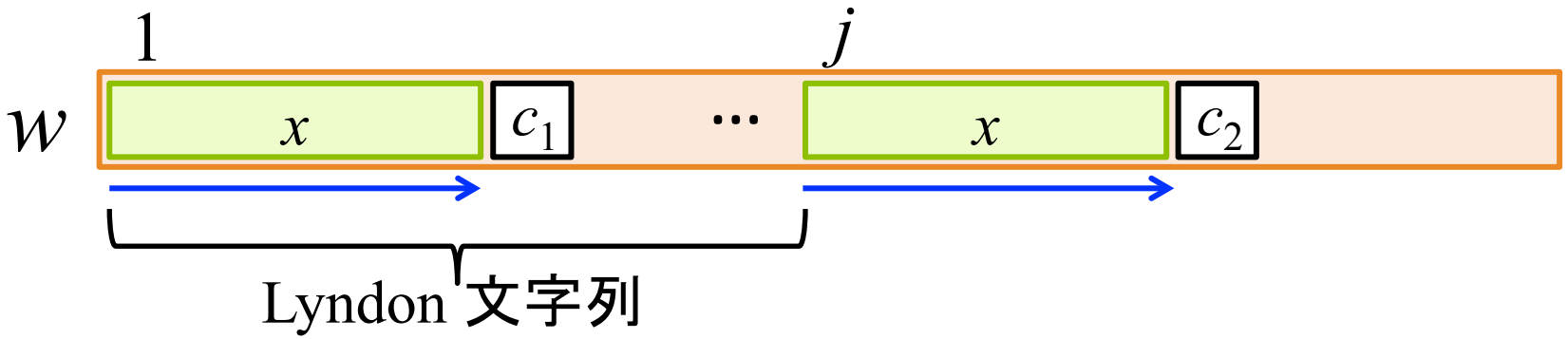
$$A = \mathrm{x\,y\,y\,x\,y\,z}$$

$$B = \mathrm{x\,x\,z\,z\,z\,z}$$

- Duval [1983] のオンライン線形時間 Lyndon 分解アルゴリズムを利用して，A から B を計算する.

Duval のアルゴリズム [Duval, 1983]

- $w[1..|x|] = w[j..j+|x|-1] = x$ かつ
  $w[1..j-1]$ は Lyndon 文字列 かつ $c_1 \neq c_2$ $(c_1, c_2 \in \sum)$ と仮定.

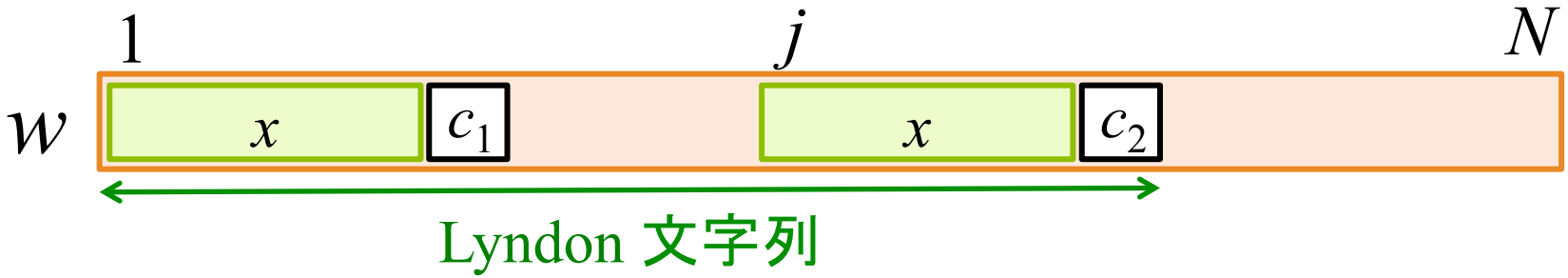


- この状態で，アルゴリズムは文字 c_1 と c_2 の辞書式順序を比較.
- この比較の結果によって，2つの場合が存在.

Duval のアルゴリズム [Duval, 1983]

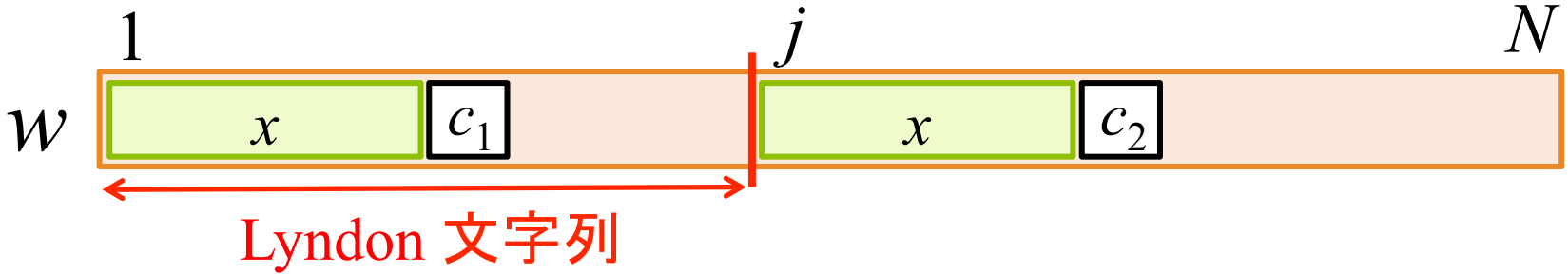
状態１: $c_1 < c_2$ のとき

$w[j..j+|x|]$ は Lyndon 文字列.



状態２: $c_1 > c_2$ のとき

$w[1..j\text{-}1]$ は Lyndon 分解の項.



- Duval のアルゴリズムでは，文字を一文字ずつ比較する.

主結果2

問題 2

自然数の組の列 $S = ((s_1, p_1), \ldots, (s_m, p_m))$ があたえられたとき，$LF_w = u_1^{p_1}, \ldots, u_m^{p_m} (|u_i| = s_i)$ となる w の最小アルファベットサイズを求めよ.

定理 2

問題 2 は $O(m)$ 時間で解ける.

本結果の意義

問題 1 に対する結果を用いることで，$O(n)$ 時間で解けるが，

$m \leq n$ であるため，効率的である.

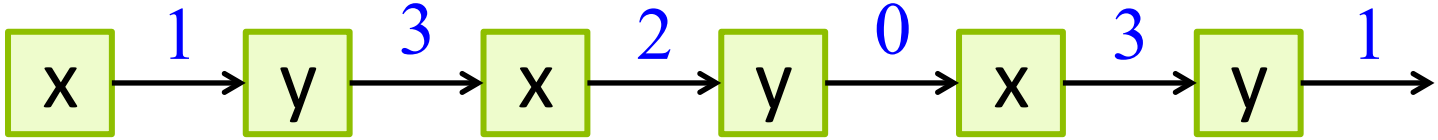
定理2のアイデア

問題 1 のアルゴリズムを以下の圧縮表現上でシミュレート.

- 辞書式順序最大の文字の連続をまとめて表現する.

例　$\Sigma = \{z, y, \ldots\}$ (z は辞書式順序最大の文字).

$$w = x\,z\,y\,z\,z\,z\,x\,z\,z\,y\,x\,z\,z\,z\,y\,z$$



主結果3

問題 3

自然数の組の列 $S = ((s_1, p_1), \ldots, (s_m, p_m))$ と 自然数 k が
あたえられたとき，$LF_w = u_1^{p_1}, \ldots, u_m^{p_m} (|u_i| = s_i)$ となる
アルファベットサイズ k 以内のすべての w を求めよ.

定理 3

問題 3 は $O(K \cdot s_{max} + n')$ 時間，$O(n')$ 追加領域で解ける.
K : 解の数，$s_{max} = \max\{s_i \mid 1 \le i \le n\}$, $n' = \sum s_i$

本結果の意義

素朴な手法では，$O(nk^n)$ 時間で解ける.

本手法は素朴な手法に比べ効率的である.

定理3のアイデア

以下のような木構造を出力とする.

例 $S = (2,2),(3,1),(2,1)$

各ノードが,
解に対応する.