九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Factorizing Strings into Combinatorial Objects

杉本, 志穂

https://doi.org/10.15017/1928634

出版情報：九州大学, 2017, 博士（理学）, 課程博士
バージョン：
権利関係：

KYUSHU UNIVERSITY

（別紙様式２）

氏　　名　：　杉本　志穂

論 文 名　：　Factorizing Strings into Combinatorial Objects
（組合せ的文字列分解）

区　　分　：　甲

論　文　内　容　の　要　旨

文字列分解(string factorization)とは，与えられた制約の下で文字列を非空文字列（項）の列に分解する問題である．たとえば，Lyndon分解は，「各項がLyndon文字列」「項の列が辞書式順序に関して単調非増加」という二つの制約を満たす分解であり一意に定まる．一方，LZ77圧縮法の中核をなすLZ分解は，「各項が既出」という制約を満たす項数最小の分解の一つである．ここで，項が既出であるとは，項と同一の文字列が項の左方に出現するときをいう．LZ分解は，「各項が既出」という制約を「各項が右極大な既出文字列」と強めることで一意性を獲得しつつ，強めない場合の最小項数と同じ項数をもつ．Lyndon分解とLZ分解の項数は，文法圧縮における文法サイズの下界を与えることが知られている．また，様々な文字列処理の問題において，前処理として入力文字列にこれらの分解を施すことでアルゴリズムの高速化に成功した事例も数多く報告されている．このように，文字列分解の研究は，文字列組合せ論分野と文字列アルゴリズム分野の双方へ貢献することが期待できる．

本論文では，既存あるいは新たに定義した文字列分解問題に対し，その組合せ的性質を究明し，効率的なアルゴリズムを開発した．

第一に，逆向きLZ分解問題およびその変種に対し省領域なオンラインアルゴリズムを開発した．逆向きLZ分解とは，「各項は右極大な逆向き既出文字列」という制約をもつ分解である．ここで，項が逆向き既出であるとは，項を反転した文字列が項の左方に出現するときをいう．$O(n \log\sigma)$ 時間・$O(n \log n)$ ビット領域を要するKolpakov-Kucherovアルゴリズムに比べ，提案アルゴリズムは$O(n \log^2 n)$ 時間・$O(n \log \sigma)$ビット領域で動作し省領域である．ここで，n は入力長，σ はアルファベットサイズを表す．また，自己参照付き逆向きLZ分解という変種を定義し，上記と同じ計算量をもつ二つのオンラインアルゴリズムを示している．

第二に，項数最小回文分解およびその変種を求めるオンラインアルゴリズムを開発した．ここで，項数最小回文分解とは，「各項が回文」という制約の下で項数を最小にする分解をいう．本論文では項数最小回文分解の一つを$O(n \log n)$ 時間・$O(n)$ 領域で計算するオンラインアルゴリズムを開発した．また，「各項が極大回文」という制約の下で項数最小の分解を求める項数最小極大回文分解問題に対して最初のオンラインアルゴリズムを与えた．計算量は前述の項数最小回文分解アルゴリズムと同じである．　一方，「各項が回文」「全ての項が異なる」という制約をもつ素回文分解を定義しこの問題がNP完全であることを示した．

第三に，閉文字列分解を $O(n)$ 時間で計算するアルゴリズムを開発した．閉文字列分解とは，「各項が右極大な閉文字列」という制約を満たす分解であり一意に定まる．ここで，文字列

wが閉文字列であるとは，文字列u（$|u|<|w|$）が存在して，u が w の接頭辞かつ接尾辞であり，かつ，u がこの2回を除いて w 中に出現しないときをいう．

第四に，アーベル同値性にかかわる問題を取り上げ，連長分解を前処理として用いる効率的アルゴリズムを開発した．文字列 x, y がアーベル同値であるとは，すべての文字について x と y における出現回数が等しいときをいう．本論文では，(1)アーベル平方を $O(mn)$ 時間，(2)アーベル周期を $O(mn)$ 時間，(3)最長共通アーベル部分文字列を $O(m^2n)$ 時間で計算するアルゴリズムを開発した．ここで，m は入力文字列の連長圧縮のサイズであり常に $m \leq n$ である．これらのアルゴリズムは，連長圧縮での圧縮率が高いときに既存手法よりも高速に動作する．すなわち，(1)は，$O(n^2)$ 時間アルゴリズム (Cummings & Smyth 1997; Crochemore et al. 2013)と同等以上であり，$(\sigma m^2)/(m-\sigma)=\omega(n)$のときには $O(\sigma(m^2+n))$ 時間アルゴリズム (Amir et al. 2014)より高速である．(2)は，$\log\log n = \omega(m^3)$ のとき $O(n(\log\log n+\log\sigma))$ 時間アルゴリズム (Kociumaka et al. 2013) より高速である．(3)は，$\sigma n = \omega(m^2)$ のときに既存の $O(\sigma n^2)$時間アルゴリズム (Grabowski 2015) より高速であり，$\log n \log^* n = \omega(m)$ のときに既存の $O(n \log^2 n \log^* n)$ 時間アルゴリズム (Badkobeh et al. 2016)より高速である．