Space-Efficient Algorithms for Computing Unique Substrings

三重野, 琢也

（別紙様式２）

氏　　名　：　三重野　琢也

論 文 名　：　Space-Efficient Algorithms for Computing Unique Substrings
(ユニーク部分文字列計算のための省領域アルゴリズム)

区　　分　：　甲

論　文　内　容　の　要　旨

近年のネットワーク技術やセンサ技術などの急速な発展により，大量のディジタルデータが日々生み出され続けている．そして，そのような大規模なデータを解析して新たな価値を創出する重要性が，多くの分野で指摘されている．しかし，扱うデータのサイズが巨大である場合，メモリや通信帯域を圧迫するため，効率的な処理が困難となる．そこで，大規模データのための省領域なデータ処理手法の開発が重要となる．計算機可読なあらゆるディジタルデータは，記号の列，即ち文字列として表現することができる．そこで，本研究では文字列データ処理のための省領域なデータ構造とアルゴリズムの開発，およびそれに資する文字列の組合せ的性質の解明に取り組んだ．

文字列中にちょうど一度だけ出現する部分文字列を，ユニーク部分文字列という．そして，文字列中の位置 p に対して，位置 p を包含する最短のユニーク部分文字列を p に対する最短ユニーク部分文字列 (Shortest Unique Substring; SUS) と呼ぶ．本研究では，このユニーク部分文字列および SUS に着目し，(A) ユニーク部分文字列の組合せ的性質，(B) SUS のための省領域データ構造，そして (C) 準動的文字列上でのユニーク部分文字列計算アルゴリズムについて研究を行った．

(A) では，(A-1) 文字列中に存在する SUS の個数の解析と (A-2) 連長圧縮と極小ユニーク部分文字列 (Minimal Unique Substring; MUS) の間の関係の解明に取り組んだ．(A-1) については，文字列中のすべての位置に対して SUS となり得る部分文字列の総数を解析し，その最大値の厳密な上下界を示すことに成功した．それまで，SUS に関するアルゴリズム的な先行研究は多く存在していたが，その一方で，非自明な組合せ的性質は知られていなかった．本成果は，SUS に関して非自明な組合せ的性質を示した初めての結果である．

(A-2) については，連長圧縮と呼ばれる文字列の圧縮手法に着目し，連長圧縮された文字列のサイズと極小ユニーク部分文字列 (MUS) の数の関係について解析した．連長圧縮文字列のサイズを r, 文字列中の MUS の個数を m としたとき, $m \leq 2r - 1$ であることを証明した．さらに，この MUS の個数の上界が厳密であること, 即ち, $m = 2r - 1$ を満たす文字列の系列が存在することを示した．

(B) では，(B-1) 連長圧縮に基づく省領域な SUS データ構造の開発と (B-2) 簡潔データ構造に基づく省領域な SUS データ構造の開発に取り組んだ．(B-1) については，クエリとして位置 p が与えられたときに p に対する SUS を計算する問題 (SUS 問題) について取り組み，連長圧縮文字列のサイズ r に線形な $O(r)$ 領域のデータ構造を新たに提案した．この結果は，(A-2) で得られた組合せ的性質に基づいた結果である．SUS 問題に対する既存のデータ構造のサイズはいずれも $O(n)$ 領域であった．ここで n は圧縮されていない通常の文字列の長さである．連長圧縮文字列のサイズ r は常に n 以下であることが言えるため，提案データ構造の領域計算量は $O(n)$ よりも悪くなることはない．特に，入力文字列が連長圧縮によって十分に小さく表現できる場合は，$O(n)$ よ

りも省領域になり得る．

(B-2) については，(B-1) と同様に SUS 問題に取り組み，簡潔データ構造を活用した $2.6n + o(n)$ ビット領域のデータ構造を新たに提案した．また，一般化された SUS 問題に対して $4n + o(n)$ ビット領域のデータ構造を提案した．計算機のワードサイズが $\log n$ ビットであるとすると，既存の $O(n)$ ワード領域のデータ構造と比較しておよそ $\log n$ 倍の省領域化であると言える．また，これらのデータ構造はいずれも線形時間で構築可能であり，構築のための作業領域もまた省領域である．

(C) では，(C-1) 準動的文字列上での MUS 計算アルゴリズムの開発と (C-2) 準動的文字列上での極小ユニーク回文部分文字列 (Minimal Unique Palindromic Substring; MUPS) 計算アルゴリズムの開発に取り組んだ．本論文で扱う準動的文字列の問題設定では，文字列 T の末尾に新たな文字を追加する操作と，T の先頭一文字を削除する操作が許されている．

(C-1) については，極小ユニーク部分文字列 (MUS) を準動的文字列上で計算するアルゴリズムを新たに提案した．文字列中のすべての MUS はオフライン，即ち全体の文字列が既知である場合に線形時間で計算できることが知られていたが，一方でオンライン（末尾への文字追加に対応した問題設定）や準動的文字列の場合の効率的な計算手法は知られていなかった．本成果は，準動的文字列に対して MUS を効率的に計算する初めての結果である．

(C-2) については，回文かつユニークな部分文字列に着目し，極小ユニーク回文部分文字列 (MUPS) を準動的文字列上で計算するアルゴリズムを新たに提案した．MUPS もまた，オフラインの場合には線形時間で計算可能である一方，オンラインや準動的文字列の場合の効率的な計算手法は知られていなかった．また，準動的文字列上の MUPS を効率的に計算するために，回文木と呼ばれるデータ構造を利用した．本成果では，回文木を準動的文字列上で高速に更新するアルゴリズムも同時に提案した．