2019 年度夏の LA シンポジウム [4]

In-Place Re-Pair

クップル ドミニク * 井 智弘 † 古谷 勇 ‡ 高畠 嘉将 † 酒井 健輔 †
後藤 啓介 §

文法圧縮は、入力文字列 T のみを導出する文法を計算し T の圧縮表現として利用する圧縮手法である。Re-Pair [2] は文法圧縮手法の一つである。様々な実際的な実験により、文法圧縮手法の中で Re-Pair は高い圧縮性能を持つことが知られている [2, 3]。

定義 1. bigram は文字のペアである。bigram b の頻度は T の中で b が重複せずに出現する回数の最大値である。

Re-Pair は、 一番頻度の大きい bigram b を確定し、b の頻度が 2 以上だったら、b の全部の出現を非終端記号で置換する。この操作を、最大頻度の bigram の頻度が 1 になるまで再帰的に繰り返す。

定義 2. T の長さを n、アルファベットサイズを σ、文法の規則のサイズを π とする。

Re-Pair を提案した論文 [2] では、線形時間・領域で動作するアルゴリズムが提案されているが、計算領域を詳しく解析すると $5n + 4\sigma^2 + 4\pi + \sqrt{n}$ words であり、定数が大きい。最近、より省領域なアルゴリズムが提案されている [1]。このアルゴリズムは T の領域に加え $\epsilon n + \sqrt{n}$ words 使用する（ϵ は任意の正の実数）。ただし、T は n words 使用する整数配列で表現され、アルゴリズムはテキスト領域を書き換え可能であることを仮定している。本研究では、さらに省領域なアルゴリズムを提案する：

定理 1. Re-Pair はテキスト領域を含めて $n \lg \max(n, \pi + \sigma)$ bits 領域、$O(n^2)$ 時間で計算できる。ただし、テキスト領域は書き換え可能であるとする。

大雑把に言うと、提案アルゴリズムは置き換えの圧縮効果によって得られる空き領域を利用し、バッチ処理で bigram の頻度を計算する。その際、追加領域を使用しない heapsort [4] を利用している。

*九州大学
†九州工業大学
‡北海道大学
§富士通研究所

参考文献

[1] P. Bille, I. L. Gørtz, and N. Prezza. Space-efficient Re-Pair compression. In *Proc. DCC*, pages 171–180, 2017.

[2] N. J. Larsson and A. Moffat. Offline dictionary-based compression. In *Proc. DCC*, pages 296–305, 1999.

[3] D. S. N. Nunes, F. A. da Louza, S. Gog, M. Ayala-Rincón, and G. Navarro. A grammar compression algorithm based on induced suffix sorting. In *Proc. DCC*, pages 42–51, 2018.

[4] J. W. J. Williams. Algorithm 232 - heapsort. *Communications of the ACM*, 7(6):347–348, 1964.