Re-Pair Achieves High-Order Entropy

Gonzalo Navarro* 
Deptartment of Computer Science
University of Chile, Chile
gnavarro@dcc.uchile.cl

Luís Russo†
Deptartment of Computer Science
University of Lisbon, Portugal
lsr@di.fc.ul.pt

Re-Pair is a dictionary-based compression method invented in 1999 by J. Larsson and A. Moffat [Off-line dictionary-based compression. Proc. IEEE, 88(11):1722–1732, 2000], lacking up to now an efficiency analysis. We show that Re-Pair compresses a sequence $T[1,n]$ over an alphabet of size $\sigma$ to at most $2^nH_k+o(n\log \sigma)$ bits, for any $k=o(\log \sigma n)$, where $H_k$ is either the classical information-theory or the empirical $k$-th order entropy (in the latter, the model is inferred from the sequence statistics).

Re-Pair repeatedly finds the most frequent pair $ab$ of symbols in the sequence and replaces its occurrences by a new symbol $A$, adding a rule $A \rightarrow ab$ to a dictionary, until every pair appears only once. Re-Pair can be implemented in linear time and space, and it decompresses very fast. At an arbitrary step $d$, the current sequence $C=c_1c_2\ldots c_p$ mixes original and newly created symbols, while the dictionary contains exactly $d$ rules. In the beginning, $C=T$, $p=n$ and $d=0$. At each step, $d$ grows by 1 and $p$ decreases at least by 2. We point out that Re-Pair compression has the following properties at any step: $\sigma+d \leq n$; the size of the compressed data $p+2d$ integers does not increase; the frequency of the most common pair does not increase; the same text cannot be represented with the same number of rules in distinct ways, i.e., if $\text{expand}(XY) = \text{expand}(ZW)$ then $X=Z$ and $Y=W$.

**Theorem 1** The Re-Pair compression algorithm outputs at most $2^nH_k(T)+o(n\log \sigma)$ bits for any $k=o(\log \sigma n)$ (so $\log \sigma = o(\log n)$ must hold to achieve $k>0$).

*Proof. We study $p+2d$ when the most frequent pair occurs at most $b = \log^2 n$ times. This is achieved in at most $n/(b+1)$ steps, hence $2d[\log n] < 2(n/b)(\log(n)+1) = O(n/\log n) = o(n)$. Consider the parsing $\text{expand}(c_1c_2)$, $\text{expand}(c_3c_4)$, $\ldots$ of $t = \lceil p/2 \rceil$ strings that do not appear more than $b$ times. R. Kosaraju and G. Manzini [Compression of low entropy strings with Lempel-Ziv algorithms. SIAM Journal on Computing, 29(3):893–911, 1999] showed that in such a case $t \log t \leq nH_k(T) + t \log(n/t) + t \log b + \Theta(t(1+k \log \sigma))$. Further algebra, especially when $t[\log n] > n/\log n$, gives the bound for $t[\log n]$.

*Supported by FCT through the Multiannual Funding Programme for LaSIGE and grant SFRH/BPD/34373/2006.