Universal Data Compression Based on the Burrows and Wheeler-Transformation: Theory and Practice

Bernhard Balkenhol*
Fakultät für Mathematik
Universität Bielefeld
Postfach 100 131
33501 Bielefeld
Germany

Stefan Kurtz†
Technische Fakultät
Universität Bielefeld
Postfach 100 131
33501 Bielefeld
Germany

September 8, 1998

Abstract

A very interesting recent development in data compression is the Burrows and Wheeler-Transformation [3]. The idea is to permute the input sequence in such a way that characters with a similar context are grouped together. We provide a thorough analysis of the Burrows and Wheeler-Transformation from an information theoretic point of view. Based on this analysis, we systematically consider techniques to efficiently implement a data compression program based on the transformation. We show that our program achieves a better compression rate than other programs that have similar requirements in space and time.

*E-mail: bernhard@mathematik.uni-bielefeld.de
†E-mail: kurtz@techfak.uni-bielefeld.de

1 Introduction and Overview

A very interesting recent development in data compression is the Burrows and Wheeler-Transformation [3]. The idea is to permute the input sequence in such a way that characters with a similar context are grouped together. This property allows a locally adaptive statistical compression scheme to achieve compression rates that are close to the best known rates. However, the important point is that these rates can be achieved with much less computational effort than previous programs based on statistical modeling techniques. Thus data compression based on the Burrows and Wheeler-Transformation is fast and it leads to good compression results.

Up until now the Burrows and Wheeler-Transformation has not been thoroughly analyzed from an information theoretic point of view. One of the main contributions of this paper is to provide such an analysis. Assuming that our information source is modeled by a context tree [27], we will show that the Burrows and Wheeler-Transformation permutes the source in such a way, that the permutation can be partitioned into intervals, one for each leaf of the context tree. Due to the property of the context tree, the subsequence of the source in each such interval is i.i.d. As a consequence, a data compression scheme based on the Burrows and Wheeler-Transformation can (in principle) achieve the same compression rates as any other context tree based method, but with much less space requirement.

Based on these theoretical insights, we systematically consider practical and engineering aspects. That is, we describe techniques to efficiently implement a data compression program based on the Burrows and Wheeler-Transformation. We consider some new techniques as well as some well-known techniques. We always carefully motivate their applicability, possibly modify them, and show how to make them work well in practice. This always includes the analysis of space and time requirements. Our contributions are as follows:

- We discuss in which cases run length encoding should be applied, and in which cases better not.
- We explain why the alphabet should always be encoded, and provide a new efficient alphabet encoding technique.
- We describe how to efficiently construct the Burrows and Wheeler-Transformation in $O(kn)$ worst case time and $12n$ bytes of space, where k is the size of the input alphabet and n is the length of the input sequence. These complexity results are achieved using suffix trees [26] and the implementation techniques of [14].
- We provide a technique to efficiently encode runs of zeros after the move-to-front transformation.