

# AXECHOP: A Grammar-based Compressor for XML

Gregory Leighton, Jim Diamond, Tomasz Müldner

Jodrey School of Computer Science, Acadia University, Wolfville, NS, Canada B4P 2R6

Email: {005985L, jim.diamond, tomasz.muldner}@acadiau.ca

XML is gaining widespread acceptance as a standard for storing and transmitting structured data. One of the drawbacks of XML is that it is quite verbose: an XML representation of a set of data can easily be ten times as large as a more economical representation of the data. To overcome this limitation, we present a compression scheme tailored specifically to XML named AXECHOP.

The compression strategy used in AXECHOP begins by dividing the source XML document into structural and data segments. The former is represented using a byte tokenization scheme that preserves the original structure of the document (i.e. it maintains the proper nesting and ordering of elements, attributes, and data values). The MPM compression algorithm is used to generate a context-free grammar capable of deriving this original structure, and the grammar is passed through an adaptive arithmetic coder before being written to the compressed file. The document's data is organized into a series of containers (where container membership is determined by the identity of the XML element or attribute that encloses the data) and then the Burrows-Wheeler Transform (BWT) is applied to the contents of each dictionary, with the results being appended to the compressed file.

Table 1 presents testing results comparing AXECHOP with a general-purpose text compression program (bzip2) and two XML-specific compression programs (XBMill and XMLPPM) on a corpus of seven XML files. The compression achieved by each program is expressed using a bits-per-character (bpc) metric. XBMill employs the BWT to compress both the structure and data of an XML file, while XMLPPM uses a quite different strategy centered on the PPM compression algorithm and is included as a benchmark against which our results can be measured. The results indicate that AXECHOP's grammar-based approach is especially effective on small- to medium-sized XML documents. On larger documents with lengthy structure strings, the ability of the BWT to efficiently compress large blocks of data comes into play, leading to a better measure for XBMill on 50000emp.

**Table 1: Experimental Results**

File	Structure Size(bytes)	Original Size (bytes)	bzip2 (bpc)	XBMill (bpc)	XMLPPM (bpc)	AXECHOP (bpc)
file1	16	235	7.387	9.055	4.596	6.332
macbeth	11238	179168	1.559	1.681	1.418	1.602
weblog	129	2214	2.399	2.678	1.771	2.786
v1-warpeace10	32929	3213992	1.965	2.040	1.818	1.950
50000emp	900002	8619966	0.458	0.395	0.383	0.409
soap	22	760	5.147	6.168	4.232	4.895
peergroup_adv	15	265	7.185	8.845	5.615	6.521
<b>average(bpc)</b>			3.729	4.409	2.833	3.499