UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Hardcoding Finite Automata

By

ERNEST KETCHA NGASSAM

Supervisors:

Bruce W. Watson and Derrick G. Kourie

Submitted in partial fulfilment of the requirements for the degree of
MAGISTER SCIENTIA (Computer Science)
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria
NOVEMBER 2003

To Maurice & Madeleine Ngassam, late Pierre Tatchou Tchoukwam, late Pauline Tchetgnia, & late Jean Petji.

Abstract

The so-called conventional approach to implement finite automata is by mean of a matrix to represent the transition function. Of course, if the transition table is very sparse, linked lists might be used as an alternative. Such approaches therefore depend on the computer's main memory capabilities to optimally hold the table for better processing.

For various computational problems using finite automata as a basic solution-model, the processing may be an important factor to be considered. This work aims to investigate a relatively new implementation approach that relies on hardcoding. A hardcoded algorithm uses simple instructions to represent the transition table. The algorithm is written in such a way that the transition matrix is part of its instructions as opposed to the traditional table-driven approach in which the table is external data that is to be accessed by the algorithm. This work includes a general performance analysis of both approaches through an empirical study. We firstly investigate the processing speed required to accept or reject a symbol by some randomly generated single states of some automata. Then, a more advanced experiment is performed based on the previous, for the test of acceptance of randomly generated strings by randomly generated finite automata .

The main result of this work is that the hardcoded implementations of finite automata outperform the table-driven implementation up to some threshold. This therefore emphasizes that many applications using finite automata as basic model may be optimized by replacing the table-driven implementation with a hardcoded implementation, resulting to better performances.

Keywords: Hardcoding, Automata, Pattern matching, Lexical analyzer, Algorithms, Experimentation, Performance, Grammars, Language, Parsing

Acknowledgements

I would like to thank Derrick G. Kourie and Bruce W. Watson, my supervisors, for their constant supports and unterminated suggestions during this research.

All my gratitude to Professor Bruce Watson for providing me the idea leading to the achievement of this work. Many thanks to Professor Derrick Kourie for its constant proof reading and critics throughout this research.

Of course, I am grateful to my parents, *Maurice and Madeleine Ngassam*, for their patience and *love*. Without them this work would never have come into existence. My thanks go also to my daughter Orline Ketcha, my son Ryan Ketcha, and my wife Liliane Ketcha who provided me with all the moral support needed to achieve such a work.

Finally, I wish to thank the following: Lisette Ngassam, Guy Ngassam, Laurent Ngassam, Mirabelle Ngassam, Orline Ngassam and Floriant Ngassam for their constant support and *love* through the path of achieving this goal.

Pretoria
October 31, 2003

Ernest Ketcha Ngassam

Table of Contents

List of Tables

List of Figures