

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Weaknesses of Compressed Text Indexes . . . . .	2
1.2	Contributions of the Thesis . . . . .	3
1.3	Thesis Organization . . . . .	5
<b>2</b>	<b>Basic Concepts</b>	<b>7</b>
2.1	Entropy, Modeling and Coding . . . . .	7
2.2	Statistical Encoding . . . . .	9
2.3	Variable-Length Integer Encoding . . . . .	10
2.4	Rank and Select Queries . . . . .	11
2.5	Searchable Partial Sums with Indels . . . . .	14
2.6	Classical Full-Text Indexes . . . . .	15
2.7	Backward Search . . . . .	16
2.8	The Burrows-Wheeler Transform . . . . .	16
2.9	Compressed Text Indexes . . . . .	17
2.9.1	The FM-index Family . . . . .	18
2.9.2	The Compressed Suffix Array (CSA) . . . . .	19
2.9.3	The Lempel-Ziv Index . . . . .	21
2.10	Re-Pair . . . . .	22
<b>3</b>	<b>Compressed Text Indexes: From Theory to Practice</b>	<b>24</b>
3.1	Practical Binary Rank and Select Queries . . . . .	25
3.1.1	Rank Queries . . . . .	25
3.1.2	Select Queries . . . . .	30
3.1.3	SelectNext Queries . . . . .	35
3.1.4	Discussion . . . . .	36
3.2	Implementing Indexes . . . . .	39
3.2.1	Implementing the FM-index . . . . .	39
3.2.2	The Alphabet-Friendly FM-index . . . . .	40
3.3	The Pizza&Chili Site . . . . .	42
3.3.1	Indexes . . . . .	43
3.3.2	Texts . . . . .	44

3.4	Experimental Results . . . . .	45
3.4.1	Construction . . . . .	47
3.4.2	Counting . . . . .	48
3.4.3	Locating . . . . .	49
3.4.4	Extracting . . . . .	51
3.5	Discussion and Open Challenges . . . . .	53
<b>4</b>	<b>Locally Compressed Suffix Arrays</b>	<b>55</b>
4.1	Locally Compressed Suffix Array (LCSA) . . . . .	56
4.1.1	Basic LCSA Idea . . . . .	57
4.1.2	Compression using $\Psi$ . . . . .	58
4.1.3	Stronger Compression based on $\Psi$ . . . . .	59
4.1.4	Compressing the Dictionary . . . . .	61
4.2	Analysis of Compression Ratio . . . . .	64
4.3	Towards a Text Index . . . . .	66
4.3.1	A Smaller Classical Index . . . . .	66
4.3.2	A Compressed Self-Index . . . . .	67
4.4	Experimental Results . . . . .	67
<b>5</b>	<b>Statistical Encoding of Sequences</b>	<b>76</b>
5.1	A New Entropy-Bounded Data Structure . . . . .	77
5.1.1	Data Structures for Substring Decoding . . . . .	77
5.1.2	Substring Decoding Algorithm . . . . .	79
5.1.3	Space Requirement . . . . .	79
5.2	Supporting Appends . . . . .	80
5.2.1	Data Structures . . . . .	81
5.2.2	Substring Decoding Algorithm . . . . .	81
5.2.3	Construction Time . . . . .	81
5.2.4	Space Requirement . . . . .	81
5.3	Application to Full-Text Indexing . . . . .	82
5.3.1	The Burrows-Wheeler Transform . . . . .	82
5.3.2	The Wavelet Tree . . . . .	83
<b>6</b>	<b>A Compressed Text Index on Secondary Memory</b>	<b>85</b>
6.1	An Entropy-Compressed Rank Dictionary on Secondary Memory . . . . .	87
6.2	An Entropy-Bounded Data Structure for Secondary Memory . . . . .	87
6.3	A Compressed Secondary Memory Structure . . . . .	89
6.3.1	Counting . . . . .	89
6.3.2	Locating . . . . .	93
6.3.3	Extracting . . . . .	93
6.4	Experiments . . . . .	94
6.5	LCSA Construction in Secondary Memory . . . . .	95

6.5.1	Compressing the Differential Suffix Array . . . . .	98
6.5.2	Compressing the Dictionary . . . . .	102
<b>7</b>	<b>Rank/Select on Dynamic Compressed Sequences</b>	<b>105</b>
7.1	Collection of Searchable Partial Sums with Indels . . . . .	106
7.2	Uncompressed Dynamic Rank-Select Structures . . . . .	110
7.3	Compressed Dynamic Rank-Select Structures . . . . .	114
7.4	Applications . . . . .	116
<b>8</b>	<b>Conclusions</b>	<b>118</b>
8.1	Contribution of this Thesis . . . . .	118
8.1.1	Compressed Text Indexes: From Theory to Practice . . . . .	118
8.1.2	Locally Compressed Suffix Arrays . . . . .	119
8.1.3	Statistical Encoding of Sequences . . . . .	120
8.1.4	A Compressed Text Index on Secondary Memory . . . . .	120
8.1.5	Rank/Select on Dynamic Compressed Sequences . . . . .	121
8.2	Further Work . . . . .	121
<b>Appendix:</b>		
<b>A</b>	<b>API for Text Indexes</b>	<b>123</b>
	<b>Bibliography</b>	<b>125</b>

# List of Figures

2.1	Algorithm to count using an FM-index. . . . .	18
2.2	Algorithm to locate using an FM-index. . . . .	19
2.3	Algorithm to count using a CSA. . . . .	20
3.1	Comparison of different popcount methods to solve <i>rank</i> . . . . .	28
3.2	Comparison of different approaches to solve <i>rank</i> . . . . .	29
3.3	Comparison of alternatives to solve <i>select</i> by binary search. . . . .	31
3.4	Comparison of different space overheads for <i>select</i> based on binary search. . . . .	32
3.5	Pseudocode for <i>select</i> ( $B, \ell$ ) . . . . .	34
3.6	Comparison of Clark's <i>select</i> and two binary searches. . . . .	35
3.7	Pseudocode for <i>selectNext</i> ( $B, i$ ) . . . . .	36
3.8	Comparison of different alternatives to solve <i>selectNext</i> . . . . .	37
3.9	Space-time tradeoffs for locating occurrences. . . . .	50
3.10	Space-time tradeoffs for extracting text symbols. . . . .	52
4.1	Algorithm to compress $A'$ using $\Psi$ in $O(n)$ time. . . . .	60
4.2	Algorithm to compress the dictionary $R$ and to update $C$ in $O(n)$ time. . . . .	63
4.3	Compression achieved per pass using $\text{RP}\Psi_0\text{SP}$ . . . . .	68
4.4	Simulating a suffix array to binary search and locate the occurrences. . . . .	72
4.5	Time to extract a portion of the suffix array. . . . .	74
6.1	Block propagation over the wavelet tree. . . . .	90
6.2	Algorithm to obtain the number of occurrences inside a disk block. . . . .	91
6.3	Compression ratio achieved. . . . .	94
6.4	Counting cost vs. space requirement . . . . .	96
6.5	Locating cost vs. space requirement . . . . .	97
6.6	Operations generated by a replacement. . . . .	101

# List of Tables

1.1	Contribution per research problem . . . . .	5
3.1	General statistics for our indexed texts. . . . .	45
3.2	Ideal compressibility of our indexed texts. . . . .	46
3.3	Real compressibility of our indexed texts. . . . .	46
3.4	Parameters used for the different indexes in our experiments. . . . .	47
3.5	Time and peak of main memory usage required to build the various indexes. . . . .	47
3.6	Experiments on the counting of pattern occurrences. . . . .	48
3.7	Number of searched patterns and total number of located occurrences. . . . .	49
3.8	<i>Locate</i> time required by plain SA . . . . .	51
3.9	The most promising indexes. . . . .	53
4.1	Compression ratio obtained using different values of $s$ for RP $\Psi$ SP. . . . .	69
4.2	Compression obtained using different values of $\delta$ for RP $\Psi$ SP. . . . .	69
4.3	Index size and build time using Re-Pair and its $\Psi$ -based approximations. . . . .	70
4.4	<i>Locate</i> time required by the LCSA and the LZ-index. . . . .	75
6.1	Different sizes and times obtained to answer <i>rank</i> . . . . .	88
6.2	Different sizes and times obtained to answer <i>count</i> . . . . .	93
6.3	Message types and meanings used by the secondary memory construction. . . . .	100