BWT in der Praxis

Nina Luhmann (kleine Änderungen von Roland)

Sequenzanalyse-Praktikum

2./3. Mai 2017

Übersicht

1. Motivation
2. Wiederholung BWT
3. BWT in der Praxis

Motivation – Mapping

- sehr lange Referenzsequenz
  $\rightarrow$ viele mögliche Mappingpositionen für Reads
- Vorverarbeitung der Referenz, die schnelle Suche erlaubt
- Indizierung der Referenz mit BWT
  $\rightarrow$ effiziente Suche aller Mappingpositionen eines Reads
- Beispiele: BWA, Bowtie2

Motivation – Kompression

- große Datensätze $\rightarrow$ viel Speicherplatz
- Kompression von Daten notwendig
- bijektive Transformation: Wiederhestellung vom Original
- BWT: Sortiere zyklische Rotationen des Textes
  $\rightarrow$ gruppiere Zeichen nach ihrem Kontext
- Ergebnis ist Permutation des Originaltextes, die besser zu komprimieren ist das Original
- Beispiel: bzip2

Burrows- Wheeler Transformation

Gegeben ist ein Text $t = s\$ = \texttt{STETSTESTE\$}$.
Konstruiere die Matrix M aus zyklischen Rotationen von t...

S	T	E	T	S	T	E	S	T	E	$
T	E	T	S	T	E	S	T	E	$	S
E	T	S	T	E	S	T	E	$	S	T
T	S	T	E	S	T	E	$	S	T	E
S	T	E	S	T	E	$	S	T	E	T
T	E	S	T	E	$	S	T	E	T	S
E	S	T	E	$	S	T	E	T	S	T
S	T	E	$	S	T	E	T	S	T	E
T	E	$	S	T	E	T	S	T	E	S
E	$	S	T	E	T	S	T	E	S	T
$	S	T	E	T	S	T	E	S	T	E

Burrows-Wheeler Transformation

...und sortiere sie.

F										L
$	S	T	E	T	S	T	E	S	T	E
E	$	S	T	E	T	S	T	E	S	T
E	S	T	E	$	S	T	E	T	S	T
E	T	S	T	E	S	T	E	$	S	T
S	T	E	$	S	T	E	T	S	T	E
S	T	E	S	T	E	$	S	T	E	T
S	T	E	T	S	T	E	S	T	E	$
T	E	$	S	T	E	T	S	T	E	S
T	E	S	T	E	$	S	T	E	T	S
T	E	T	S	T	E	S	T	E	$	S
T	S	T	E	S	T	E	$	S	T	E

Burrows-Wheeler Transformation

$\Rightarrow bwt(t) = \texttt{ETTTET\$SSSE}$

	L
	E
	T
	T
	T
	E
	T
	$
	S
	S
	S
	E

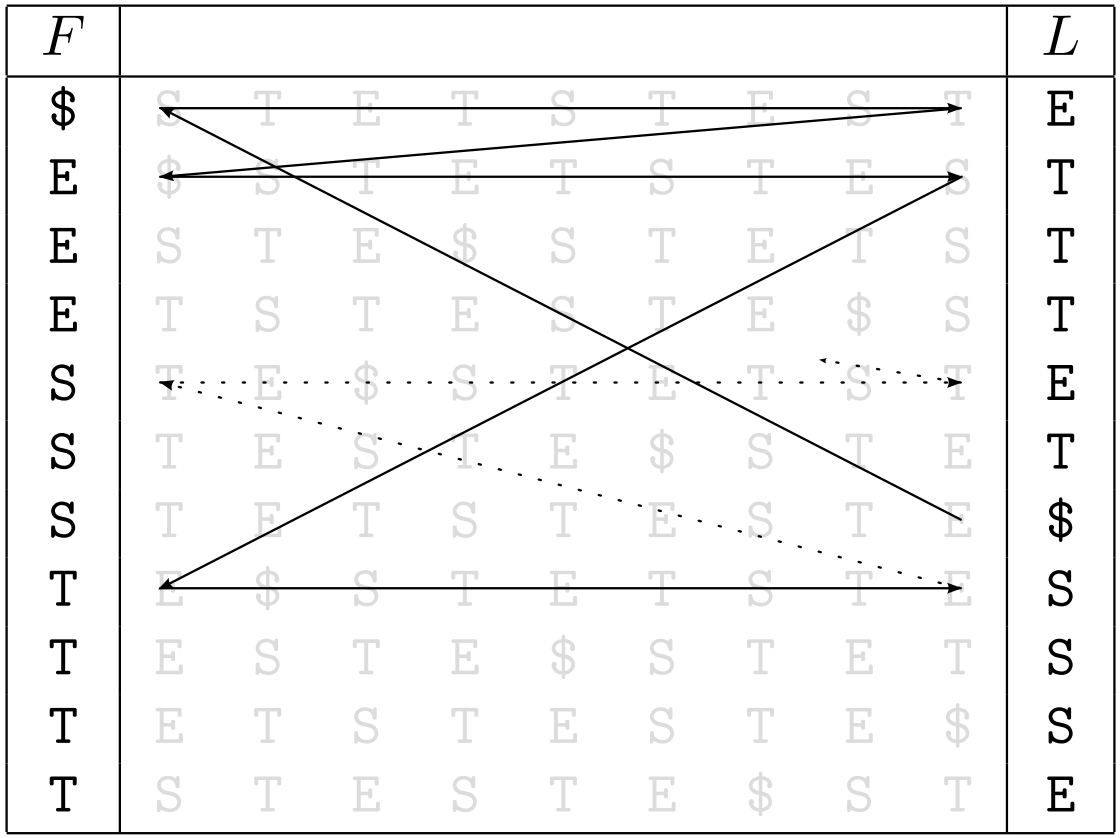
Burrows-Wheeler Rücktransformation

	L
	E
	T
	T
	T
	E
	T
	$
	S
	S
	S
	E

Burrows-Wheeler Rücktransformation

F										L
$	S	T	E	T	S	T	E	S	T	E
E	$	S	T	E	T	S	T	E	S	T
E	S	T	E	$	S	T	E	T	S	T
E	T	S	T	E	S	T	E	$	S	T
S	T	E	$	S	T	E	T	S	T	E
S	T	E	S	T	E	$	S	T	E	T
S	T	E	T	S	T	E	S	T	E	$
T	E	$	S	T	E	T	S	T	E	S
T	E	S	T	E	$	S	T	E	T	S
T	E	T	S	T	E	S	T	E	$	S
T	S	T	E	S	T	E	$	S	T	E

Burrows-Wheeler Rücktransformation



Read Mapping mit BWT

- Aufgabe: Finde **alle** Vorkommen eines kurzen Reads in einer laaaaaangen Referenz
- Idee: BWT der Referenz, damit Vorkommen zusammen gruppiert
- Bowtie2, BWA: FM-Index der Referenz
- FM-Index = BWT + Suffix Array + zusätzliche Tabellen
- wenig Speicherplatzverbrauch: 2,2 GB für menschliches Genom

Beispiel (exaktes) Mapping

$p = \texttt{TEST}$

F	$\cdots$	L
$		E
E		T
E		T
E		T
S		E
S		T
S		$
T		S
T		S
T		S
T		E

Beispiel (exaktes) Mapping

$p = \texttt{TEST}$

F	$\cdots$	L
$		E
E		T
E		T
E		T
S		E
S		T
S		$
T		S
T		S
T		S
T		E

F	$\cdots$	L
$		E
E		T
E		T
E		T
S		E
S		T
S		$
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T		E

Beispiel (exaktes) Mapping

$p = \texttt{TEST}$

F	$\cdots$	L
$		E
E		T
E		T
E		T
S		E
S		T
S		$
T		S
T		S
T		S
T		E

F	$\cdots$	L
$		E
E		T
E		T
E		T
S		E
S		T
S		$
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T		E

F	$\cdots$	L
$		E
E		T
E		T
E		T
S	$\longrightarrow$	E
S	$\longrightarrow$	T
S	$\longrightarrow$	$
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T		E

Beispiel (exaktes) Mapping

$p = \texttt{TEST}$

F	$\cdots$	L
$		E
E		T
E		T
E		T
S		E
S		T
S		$
T		S
T		S
T		S
T		E

F	$\cdots$	L
$		E
E		T
E		T
E		T
S		E
S		T
S		$
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T		E

F	$\cdots$	L
$		E
E		T
E		T
E		T
S	$\longrightarrow$	E
S	$\longrightarrow$	T
S	$\longrightarrow$	$
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T	$\longrightarrow$	S
T		E

F	$\cdots$	L
$		E
E		T
E	$\longrightarrow$	T
E		T
S	$\longrightarrow$	E
S		T
S		$
T	$\longrightarrow$	S
T		S
T		S
T		E

Kompression mit bzip2

1. Blocks of size 100.000–900.000
2. **Burrows–Wheeler transform (BWT)**
3. **Move to front (MTF)**
4. Run-length encoding (RLE)
5. **Huffman coding**
6. Selection between multiple Huffman tables
7. Unary base 1 encoding of Huffman table selection
8. Delta encoding (Δ) of Huffman code bit-lengths
9. Sparse bit array showing which symbols are used

Beispiel

$t = \texttt{STETSTESTE\$} \rightarrow 11 \text{ bytes} = 88 \text{ bit}$

Beispiel – BWT

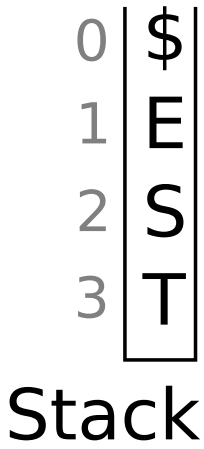
$t = \texttt{STETSTESTE\$} \rightarrow 11$ bytes $= 88$ bit

$bwt(s) = \texttt{ETTTET\$SSSE}$

Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt

E T T T E T $ S S S E

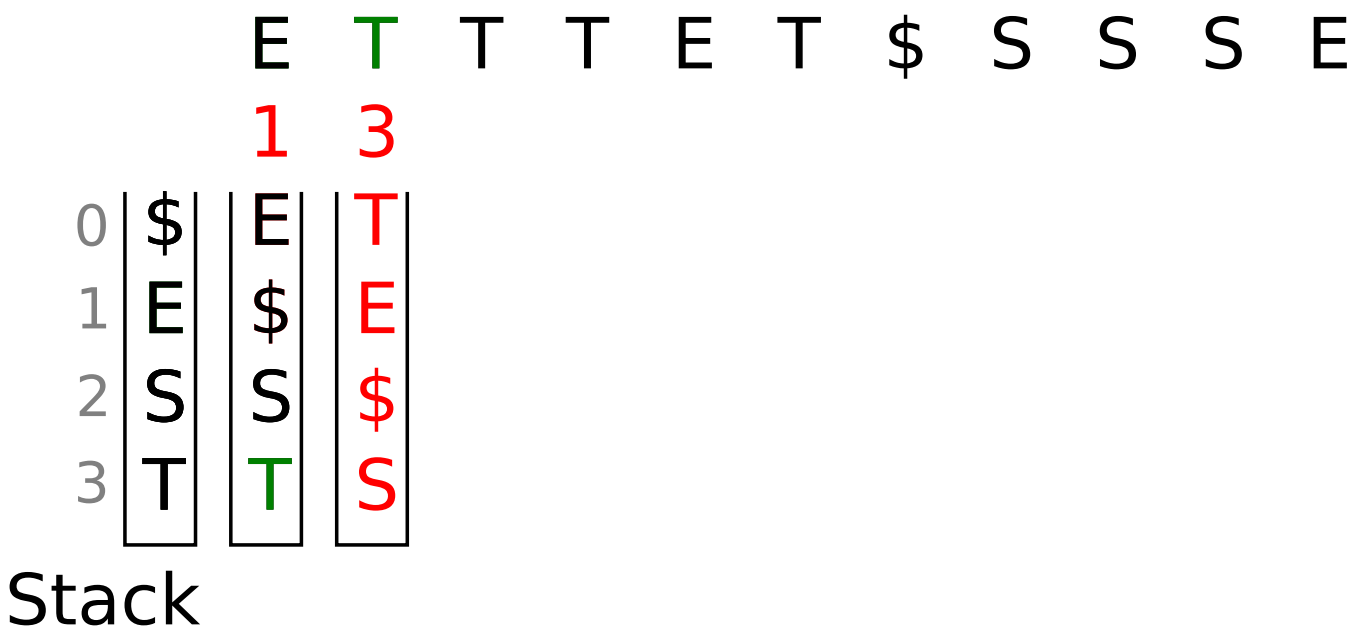


Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt

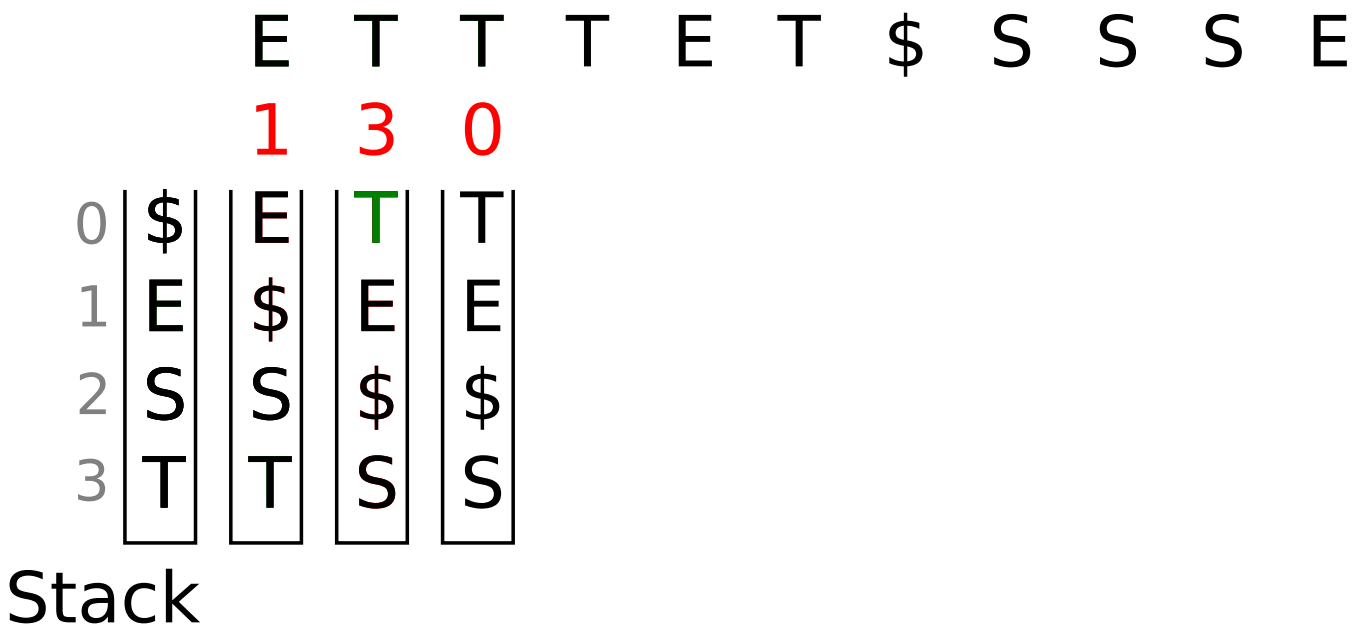
Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt



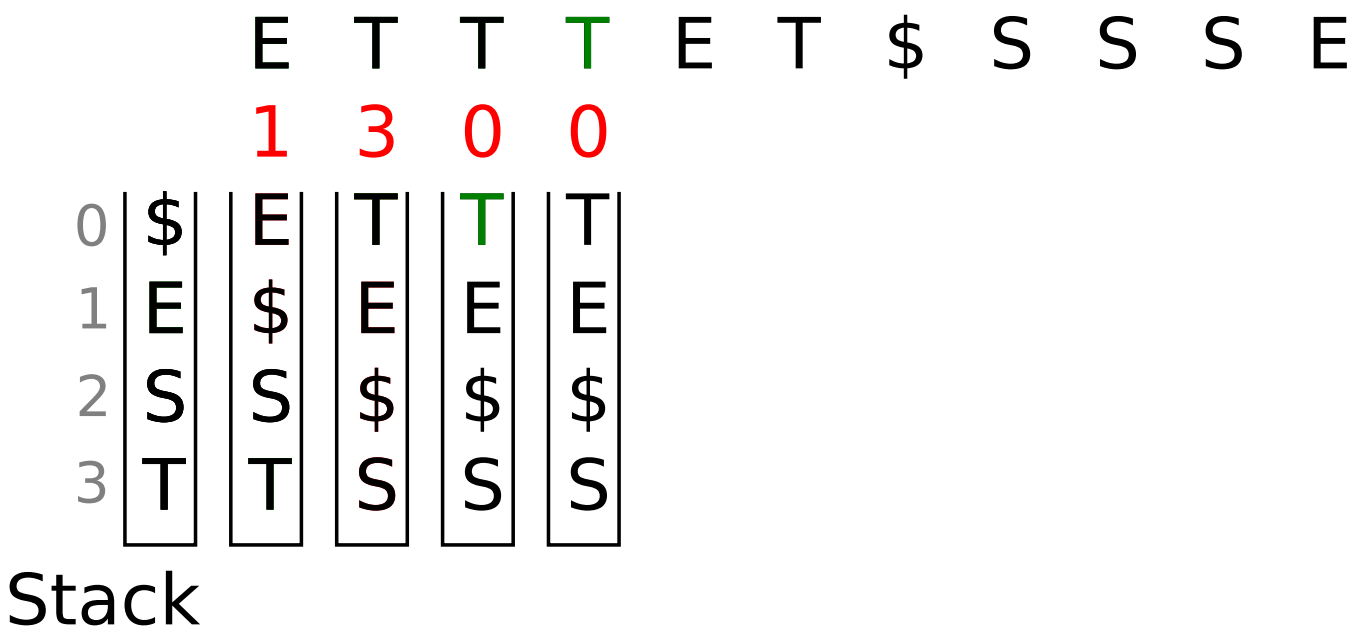
Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt



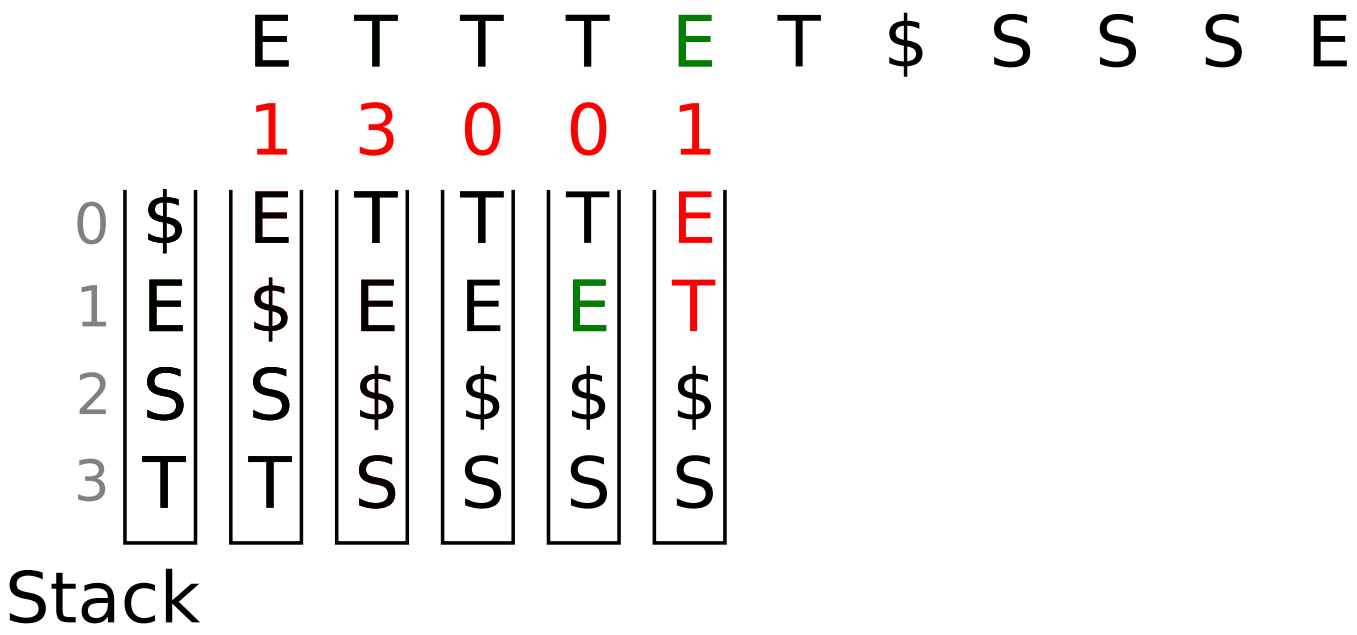
Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt



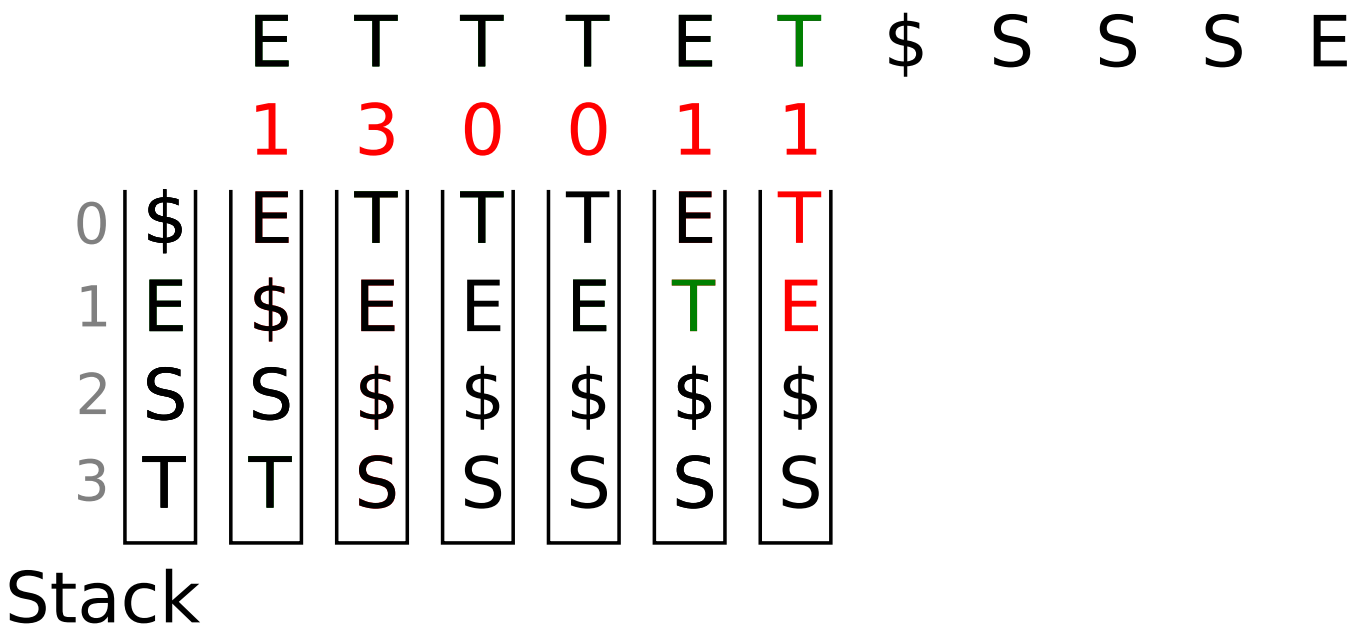
Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt



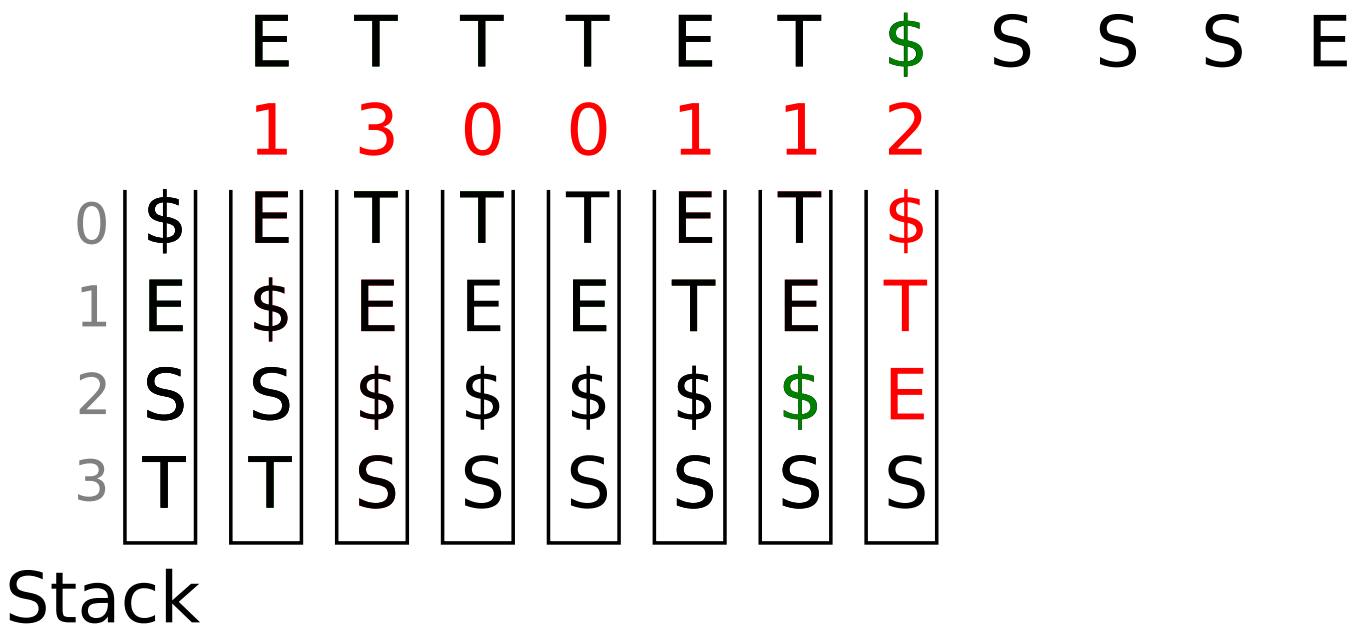
Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt



Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt



Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt

Beispiel – Move-to-front

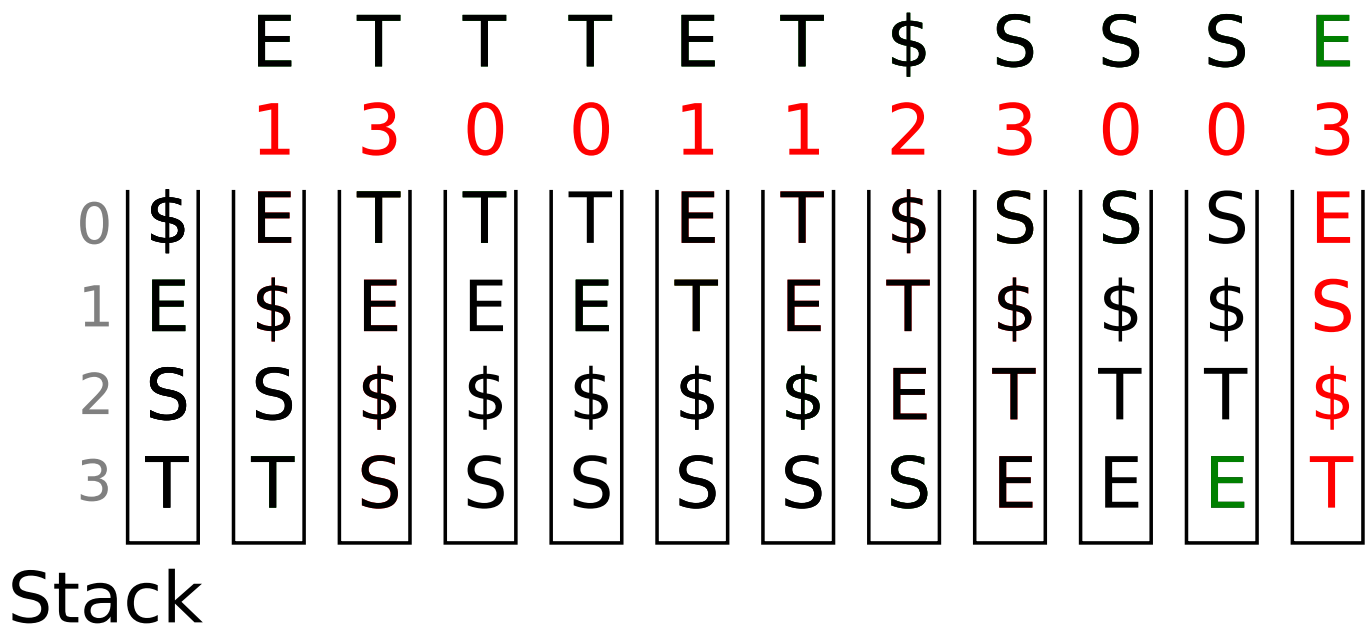
Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt

Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt

Beispiel – Move-to-front

Idee: Jedes Symbol wird durch seinen Index im Stack der zuletzt verwendeten Symbole ersetzt



Beispiel – Huffman Coding

$mtf = 1, 3, 0, 0, 1, 1, 2, 3, 0, 0, 3$
$stack = (E, S, \$, T)$
Idee: Kodierung von Zeichen in Binärform, so dass häufigere Zeichen kurze Codes und seltene Zeichen längere Codes erhalten (prefix-free codes)

Häufigkeiten:

0	4
1	3
2	1
3	3

Beispiel – Huffman Coding

$mtf = 1, 3, 0, 0, 1, 1, 2, 3, 0, 0, 3$

Alphabet	0	1	2	3

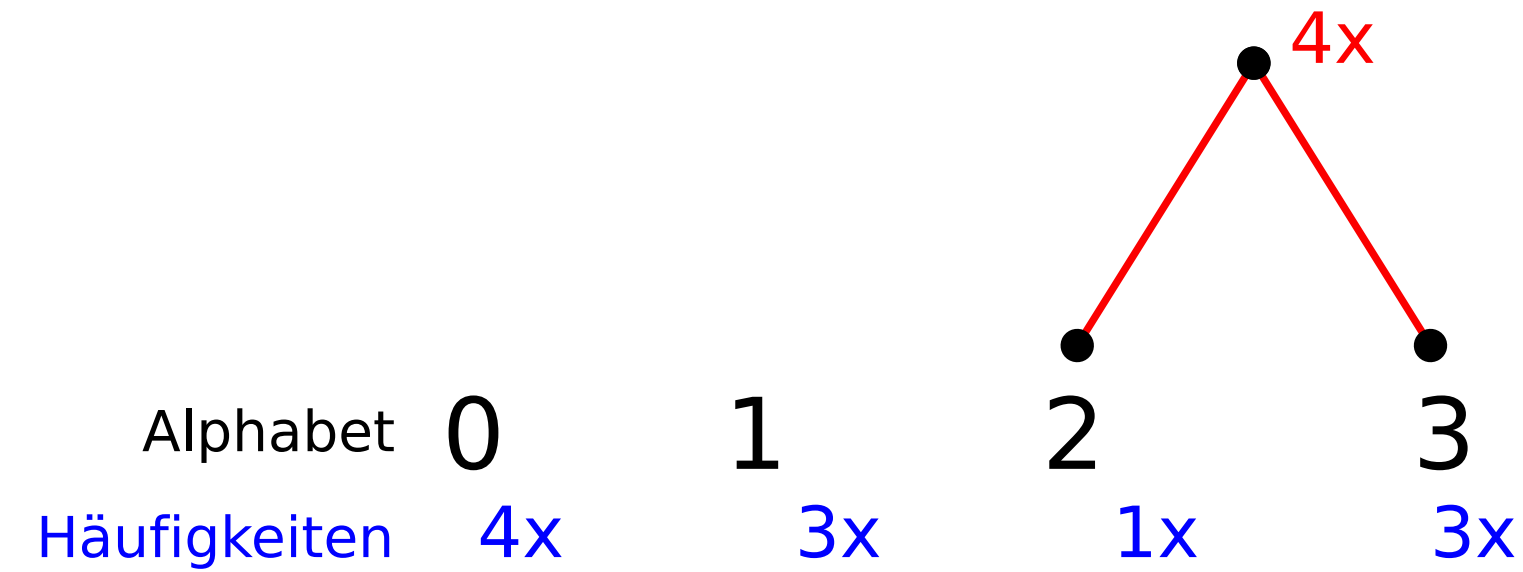
Beispiel – Huffman Coding

$mtf = 1, 3, 0, 0, 1, 1, 2, 3, 0, 0, 3$

Alphabet	0	1	2	3
Häufigkeiten	4x	3x	1x	3x

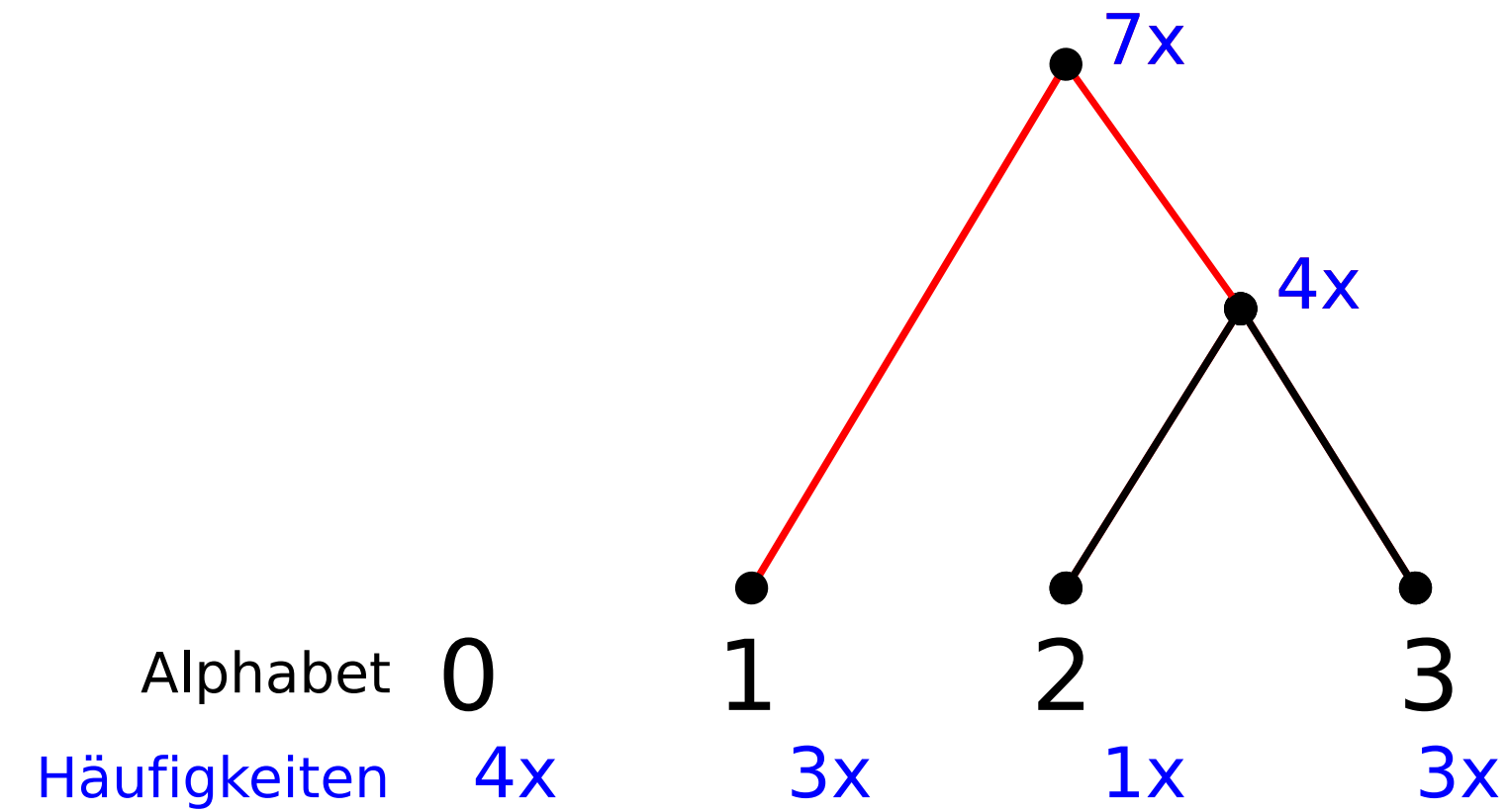
Beispiel – Huffman Coding

$mtf = 1, 3, 0, 0, 1, 1, 2, 3, 0, 0, 3$



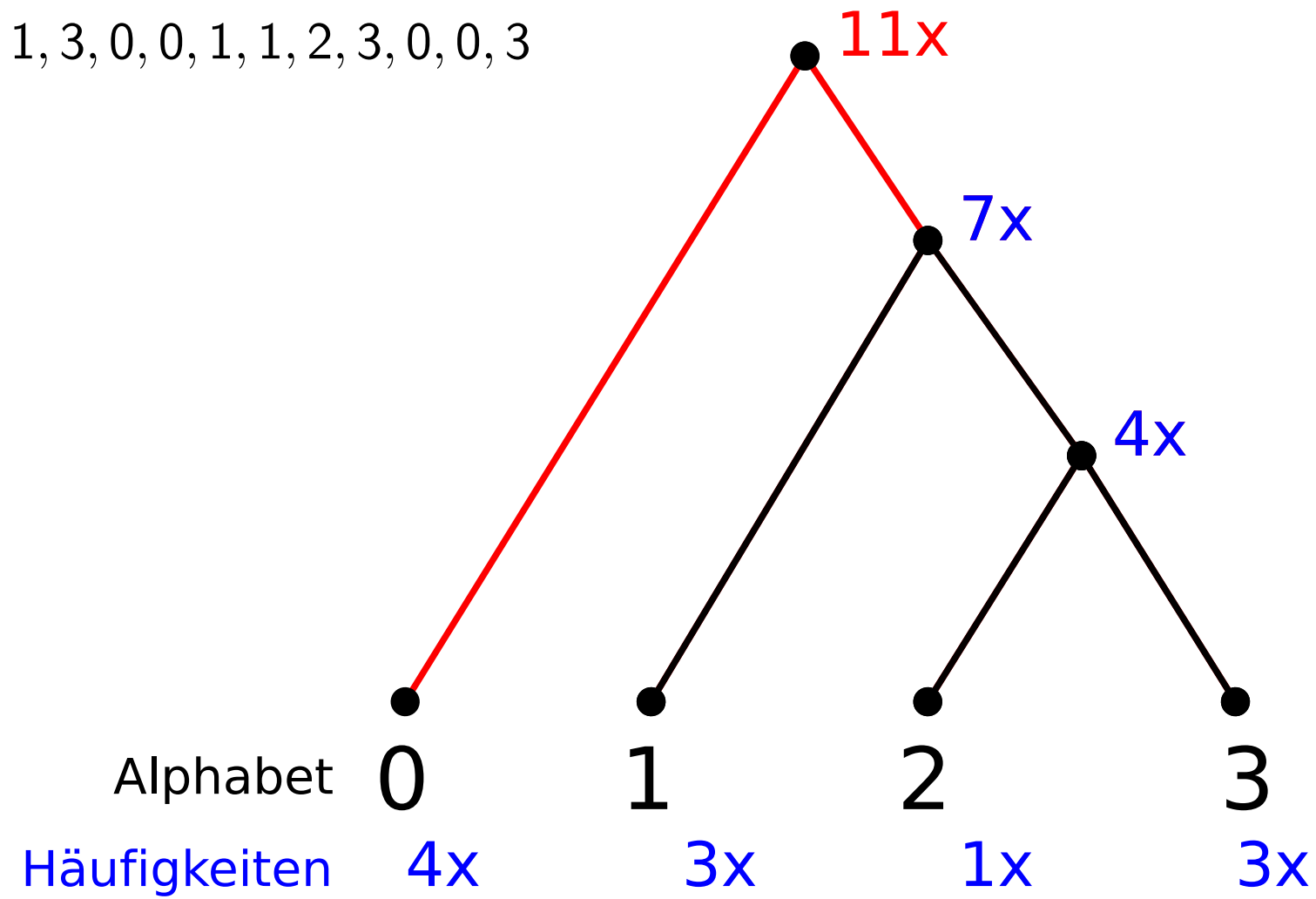
Beispiel – Huffman Coding

$mtf = 1, 3, 0, 0, 1, 1, 2, 3, 0, 0, 3$



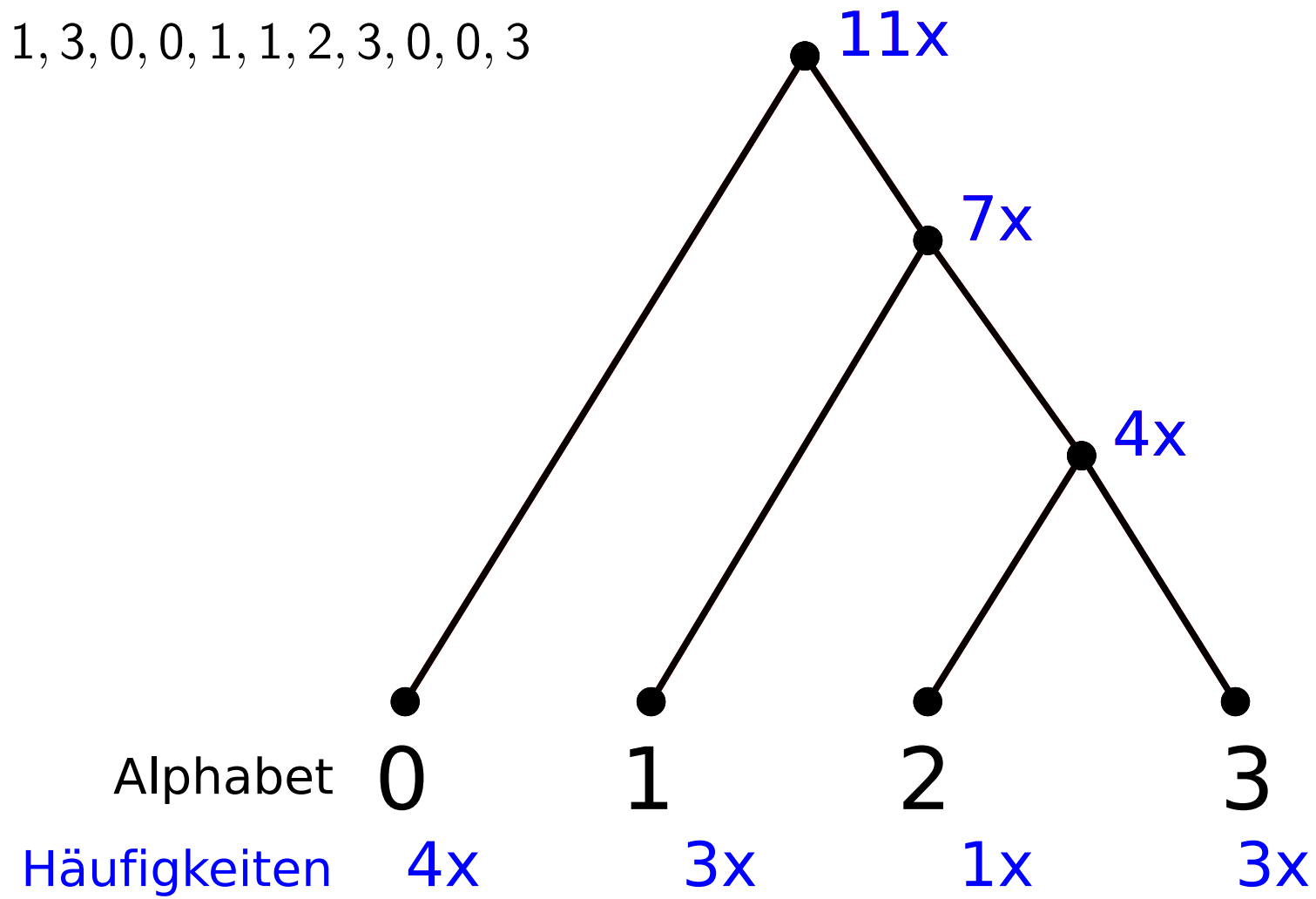
Beispiel – Huffman Coding

$mtf = 1, 3, 0, 0, 1, 1, 2, 3, 0, 0, 3$



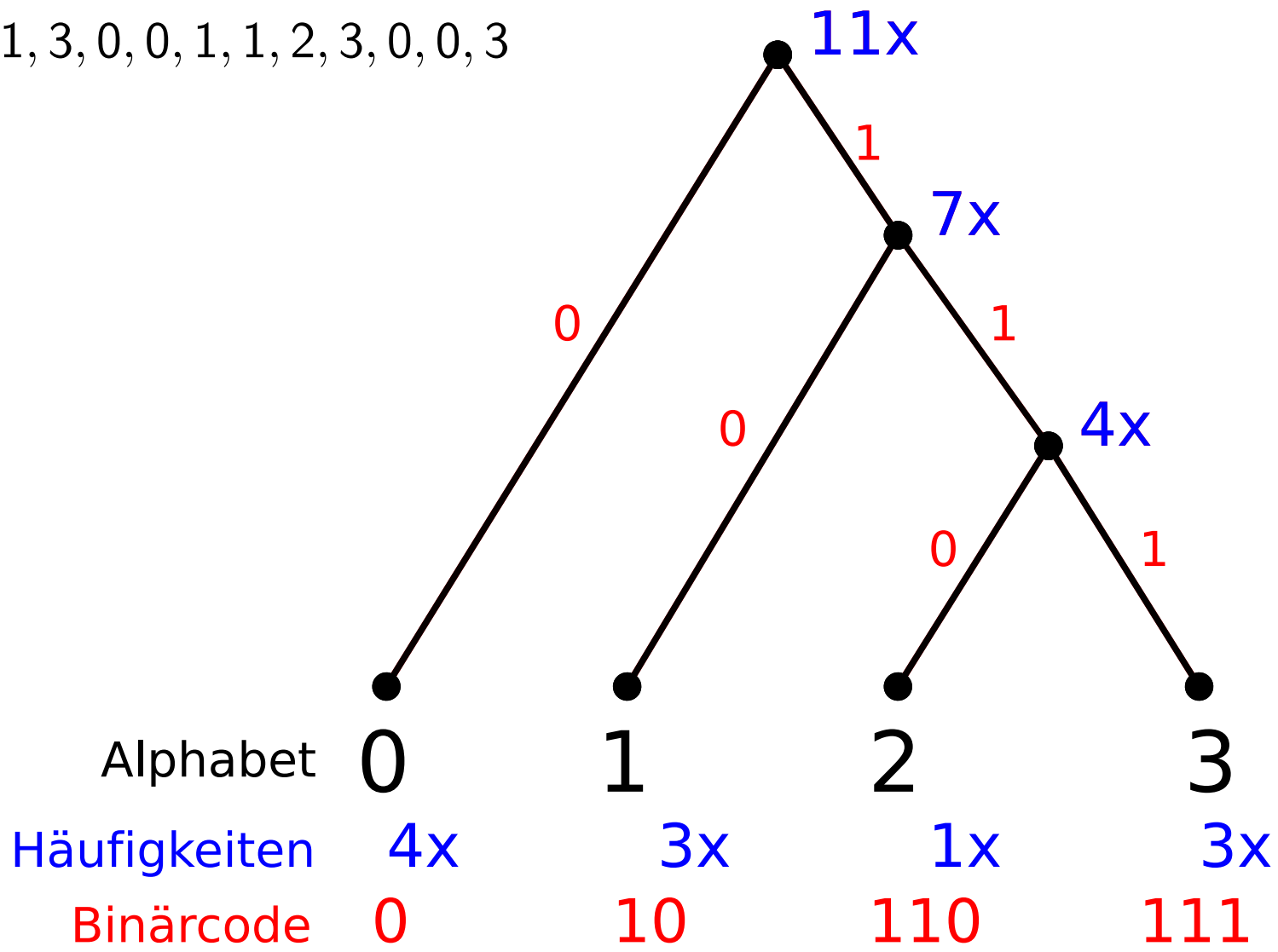
Beispiel – Huffman Coding

$mtf = 1, 3, 0, 0, 1, 1, 2, 3, 0, 0, 3$



Beispiel – Huffman Coding

$mtf = 1, 3, 0, 0, 1, 1, 2, 3, 0, 0, 3$



Beispiel - Huffman Coding

$t = \texttt{STETSTESTE\$} \rightarrow$ **88 bit**,
$mtf = 1, 3, 0, 0, 1, 1, 2, 3, 0, 0, 3$
$\Rightarrow 10111001010110111100111 \rightarrow$ **22 bit +**

	#	
0	4	0
1	3	10
2	1	110
3	3	111

Einfluss der Alphabetgröße auf bzip2?

1. Blocks of size 100.000–900.000
2. Burrows–Wheeler transform (BWT)
3. Move to front (MTF)
4. Run-length encoding (RLE)
5. Huffman coding
6. Selection between multiple Huffman tables
7. Unary base 1 encoding of Huffman table selection
8. Delta encoding (Δ) of Huffman code bit-lengths
9. Sparse bit array showing which symbols are used

Einfluss der Alphabetgröße auf bzip2?

1. Blocks of size 100.000–900.000
2. Burrows–Wheeler transform (BWT)
3. **Move to front (MTF)**
4. **Run-length encoding (RLE)**
5. **Huffman coding**
6. Selection between multiple Huffman tables
7. Unary base 1 encoding of Huffman table selection
8. Delta encoding (Δ) of Huffman code bit-lengths
9. Sparse bit array showing which symbols are used

Zusammenfassung

- BWT für Textkomprimierung
- Typischer Schritt vor der Move-to-front Codierung
- Teil des bzip2 Algorithmus
- Übung: Für welche Texte funktioniert das gut, für welche schlechter?
- BWT als Vorverarbeitung im Read mapping
- Übung: Spielerei mit Bowtie2

Quellen

- Skript zur Vorlesung “Sequenzanalyse”, Technische Fakultät, Universität Bielefeld, Wintersemester 2014/15
- M. Burrows and D. Wheeler: “A block sorting lossless data compression algorithm.”, Technical Report 124, Digital Equipment Corporation, 1994
- “bzip2”, https://en.wikipedia.org/wiki/Bzip2, Stand: 15.04.2016
- D.A. Huffman, “Method for the Construction of Minimum-Redundancy Codes“, Proceedings of the I.R.E., pp 1098–1102, 1952