Compression

- Compression
- Lempel-Ziv
- Re-Pair and Grammars

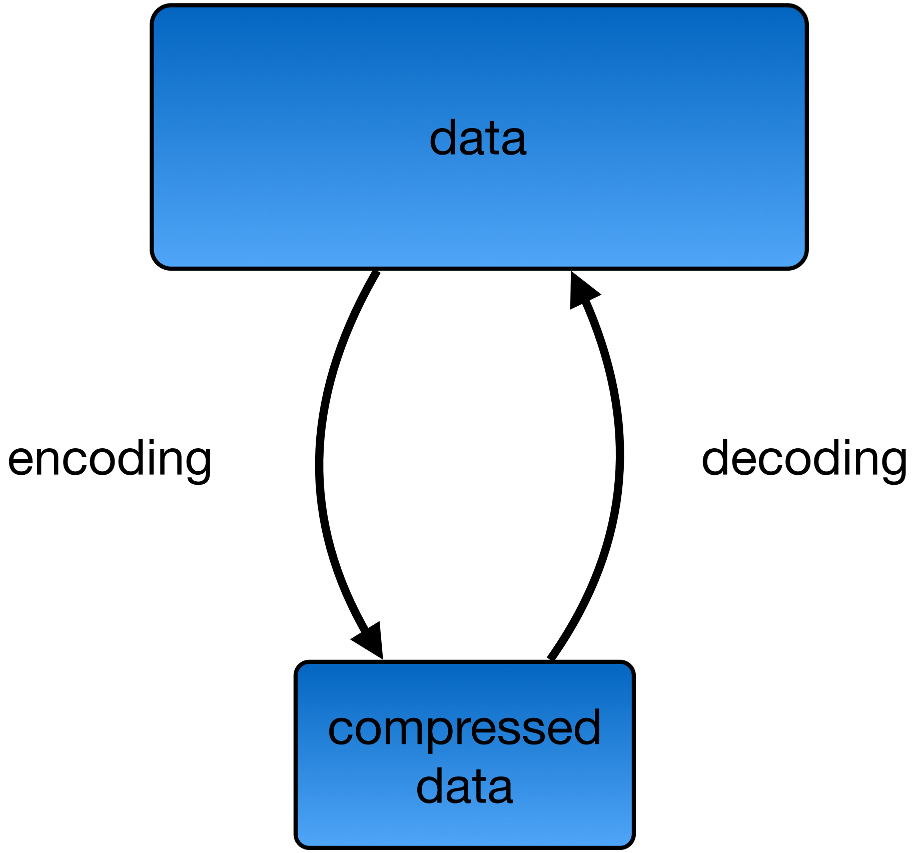
Philip Bille

Compression

- Compression
- Lempel-Ziv
- Re-Pair and Grammars

Compression

- Encoding and decoding.
- Lossless and lossy
- Compressed computation.



Compression

- Statistical compression.
    - Huffman, arithmetic encoding, Burrows-Wheeler, PPM, ...
- Dictionary compression.
    - Lempel-Ziv 77, Lempel-Ziv 78, Lempel-Ziv-Welch, …
- Grammar based schemes.
    - Re-Pair, sequitur, greedy, bisection, …
- Kolmogorov compression.
    - Ultimate compression scheme.

- Transformation techniques.
    - Differencing, Burrows-Wheeler, run-length encoding, Fourier transform, …

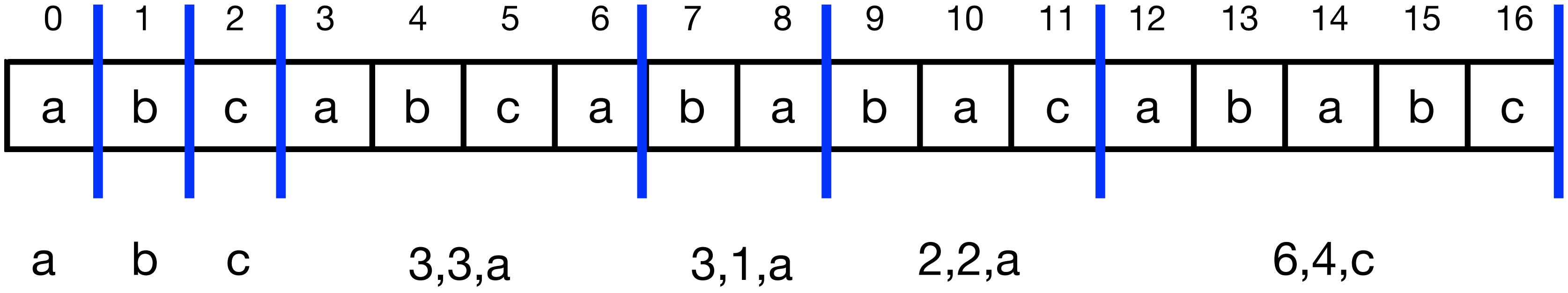
Compression

- Compression
- Lempel-Ziv
- Re-Pair and Grammars

Lempel-Ziv 77

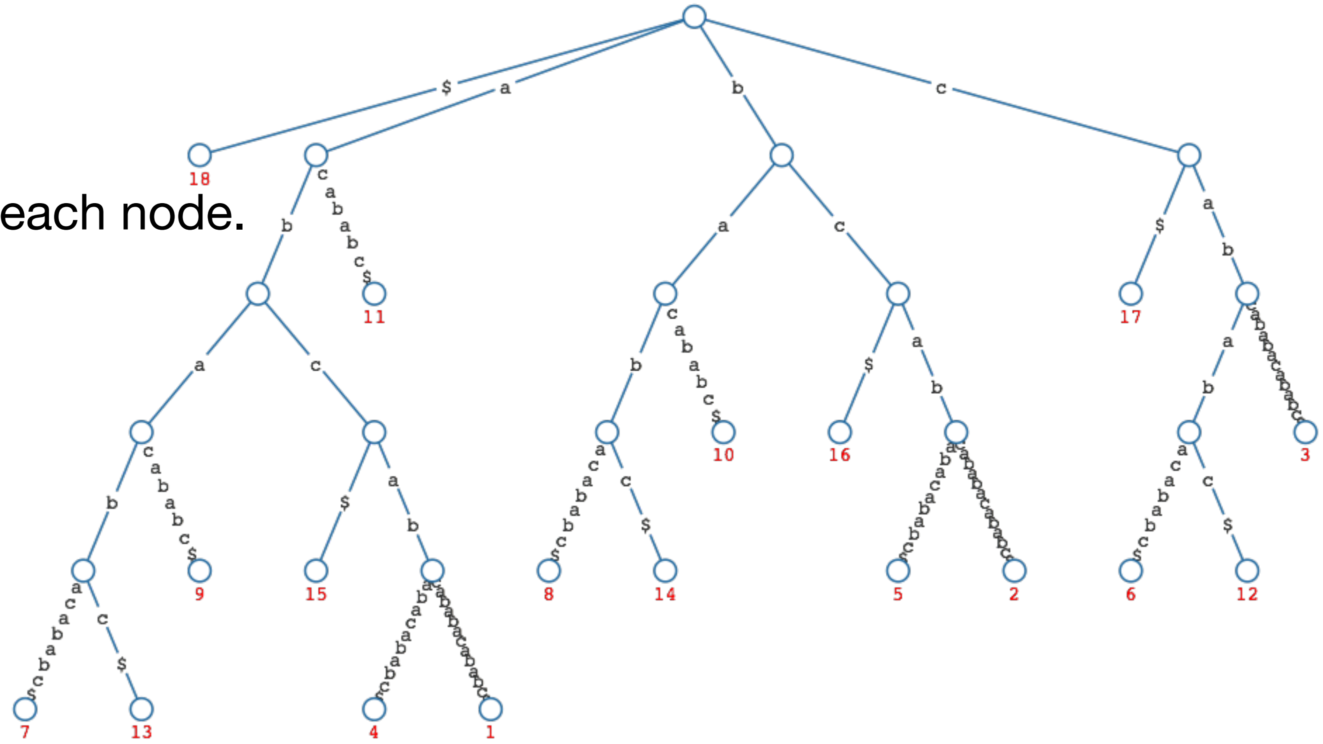
- Encoding.
  - Parse from left-to-right into phrases.
  - Select longest substring starting before current position + 1 character.
  - Encode phrases by (previous occ, length, single character) or single character.

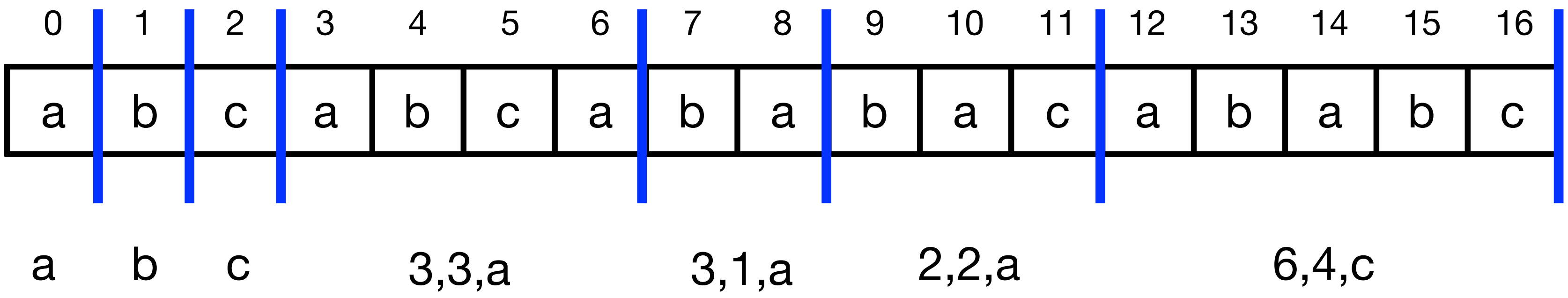
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	b	c	a	b	c	a	b	a	b	a	c	a	b	a	b	c



Lempel-Ziv 77

- Encoding.
    - Build suffix tree
    - Store smallest leaf below each node.
    - Greedy left-to-right parse.
- Time. O(n)



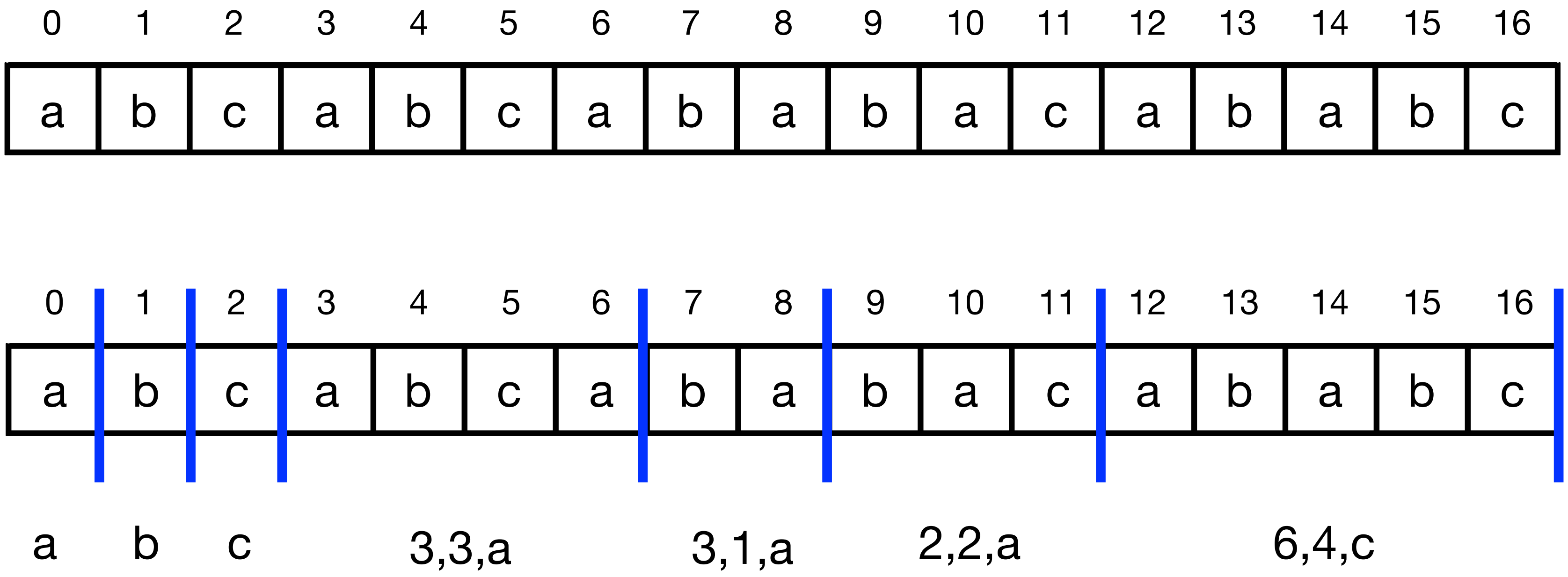


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	b	c	a	b	c	a	b	a	b	a	c	a	b	a	b	c

a b c 3,3,a 3,1,a 2,2,a 6,4,c

Lempel-Ziv 77

- Decoding. Read and decode left-to-right.
- Time. O(n)

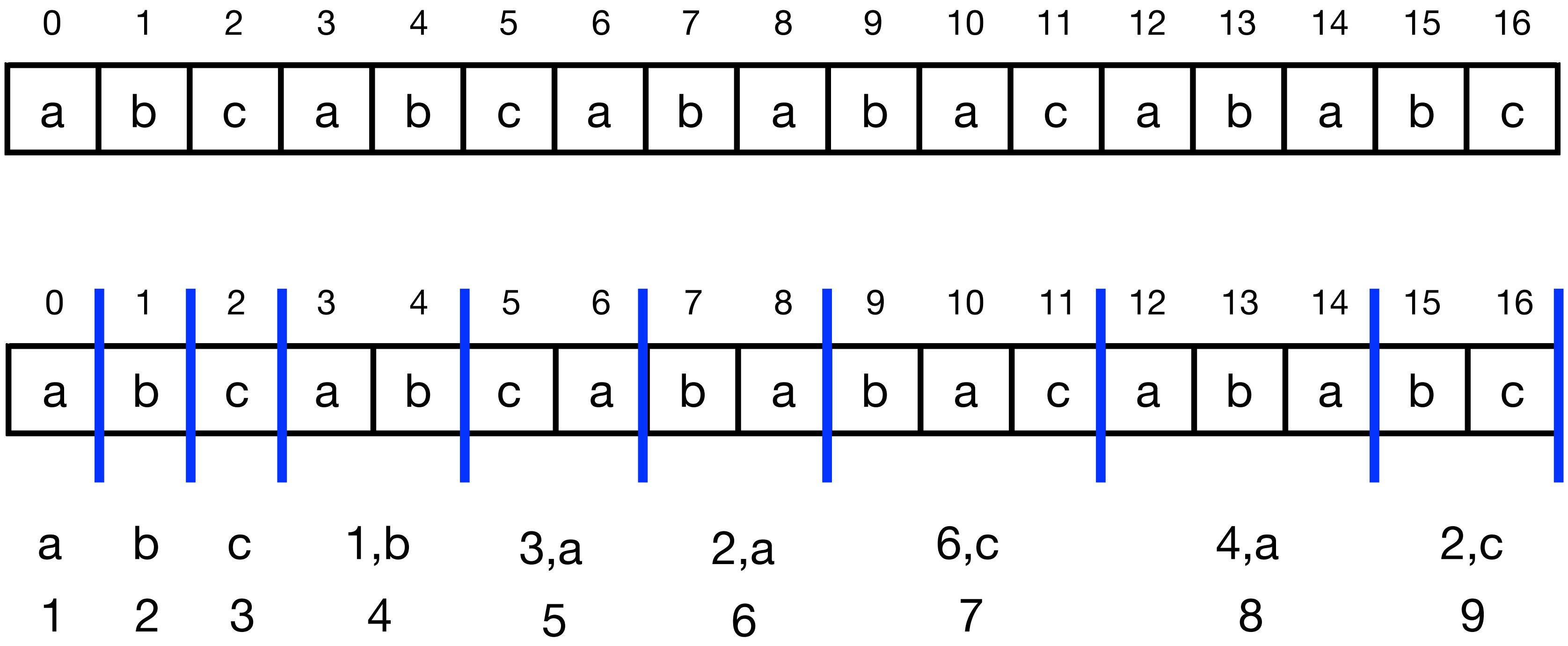


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	b	c	a	b	c	a	b	a	b	a	c	a	b	a	b	c

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	b	c	a	b	c	a	b	a	b	a	c	a	b	a	b	c

Lempel-Ziv 78

- Encoding.
  - Parse from left-to-right into phrases.
  - Select longest phrase seen before + a single character.
  - Encode phrases (previous phrase, character) or



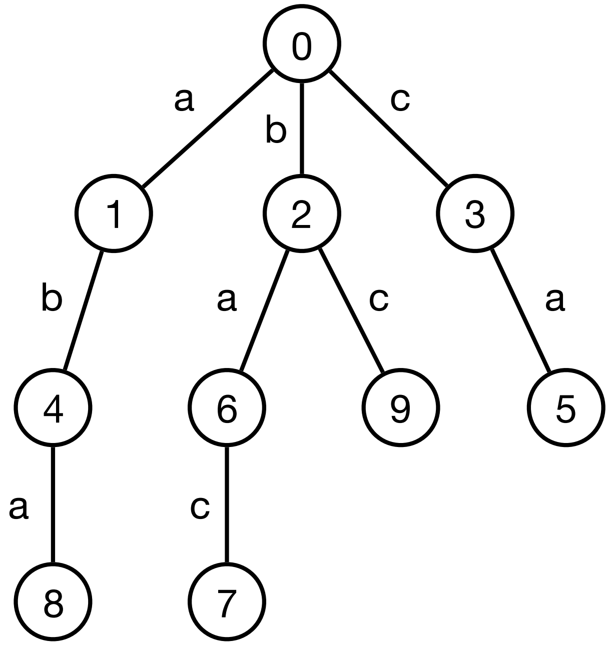
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	b	c	a	b	c	a	b	a	b	a	c	a	b	a	b	c

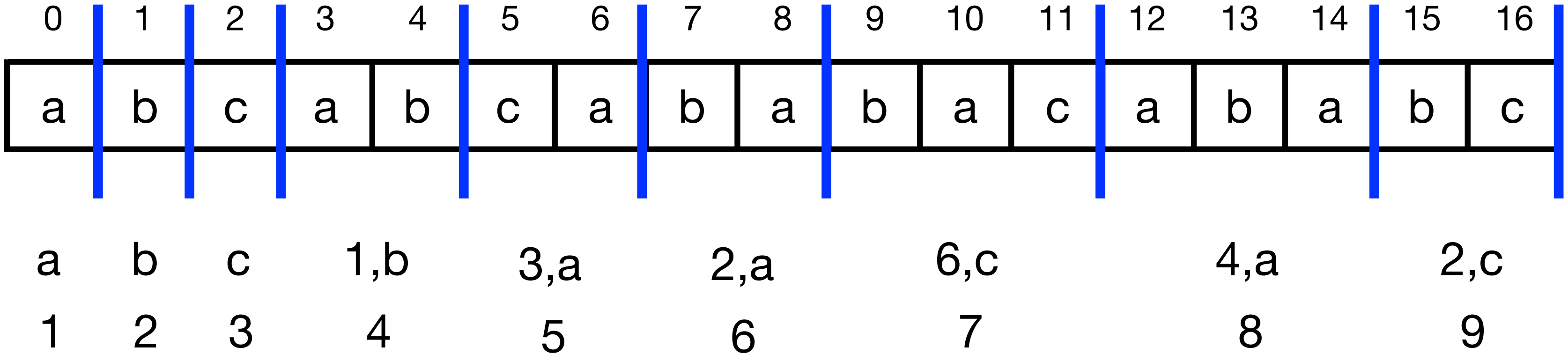
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	b	c	a	b	c	a	b	a	b	a	c	a	b	a	b	c

a	b	c	1,b	3,a	2,a	6,c	4,a	2,c
1	2	3	4	5	6	7	8	9

Lempel-Ziv 78

- Encoding.
    - Dynamically build and traverse the LZ78 trie.
- Time. O(n)



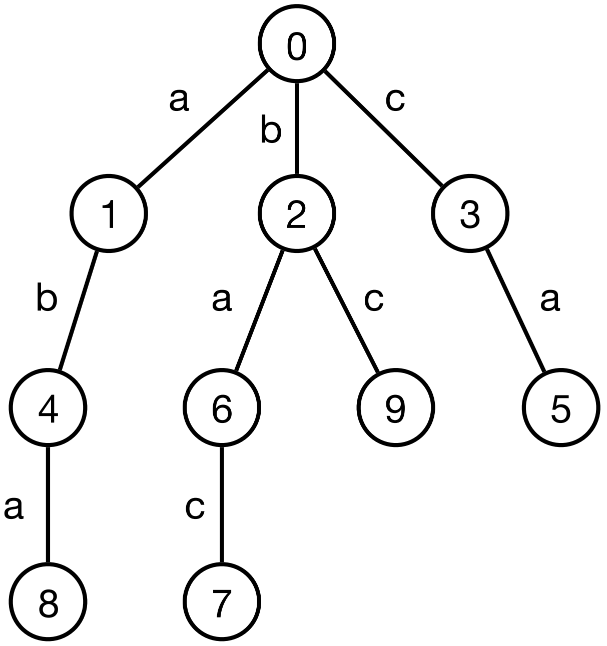


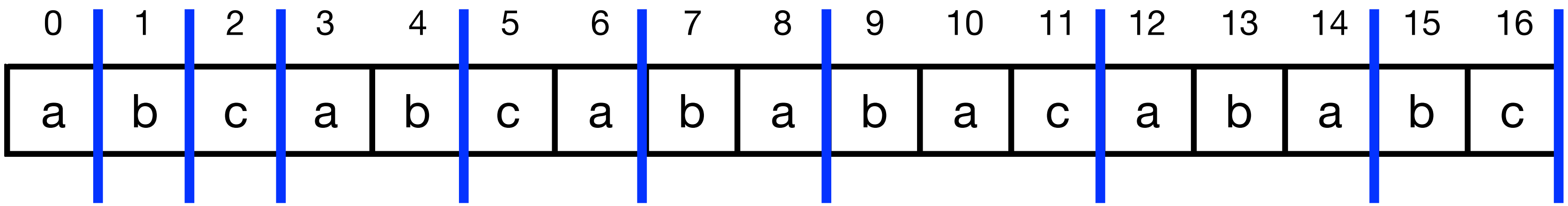
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	b	c	a	b	c	a	b	a	b	a	c	a	b	a	b	c

a	b	c	1,b	3,a	2,a	6,c	4,a	2,c
1	2	3	4	5	6	7	8	9

Lempel-Ziv 78

- Decoding. Read and decode left-to-right.
- Time. O(n)





0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	b	c	a	b	c	a	b	a	b	a	c	a	b	a	b	c

a	b	c	1,b	3,a	2,a	6,c	4,a	2,c
1	2	3	4	5	6	7	8	9

Compression

- Compression
- Lempel-Ziv
- Re-Pair and Grammars

Re-Pair Compression

- Recursive-pairing compression [Larsson and Moffat 2000].
  - Start with string S.
  - Replace a most frequent pair ab by new character X_i. Output rule $X_i \rightarrow ab$.
  - Repeat until we have a single pair.
- Decoding. Unfold rules top-down.

X_9	
X_8X_6	$X_9 \rightarrow X_8X_6$
$X_3X_7X_6$	$X_8 \rightarrow X_3X_7$
$X_3X_4X_5X_6$	$X_7 \rightarrow X_4X_5$
$X_3X_4X_5X_1X_2$	$X_6 \rightarrow X_1X_2$
$X_3X_4acX_1X_2$	$X_5 \rightarrow ac$
$X_3X_1X_1acX_1X_2$	$X_4 \rightarrow X_1X_1$
$X_2X_2X_1X_1acX_1X_2$	$X_3 \rightarrow X_2X_2$
$X_1cX_1cX_1X_1acX_1X_1c$	$X_2 \rightarrow X_1c$
abcabcababacababc	$X_1 \rightarrow ab$

Grammar Compression

- Grammar compression. Encode string S as an grammar G that generates S.
- Parse tree. Unfolded set of rules.

$X_{12} \rightarrow X_{11}X_9$

$X_{11} \rightarrow X_6X_{10}$

$X_{10} \rightarrow X_7X_8$

$X_9 \rightarrow X_4X_5$

$X_8 \rightarrow X_1X_3$

$X_6 \rightarrow X_5X_5$

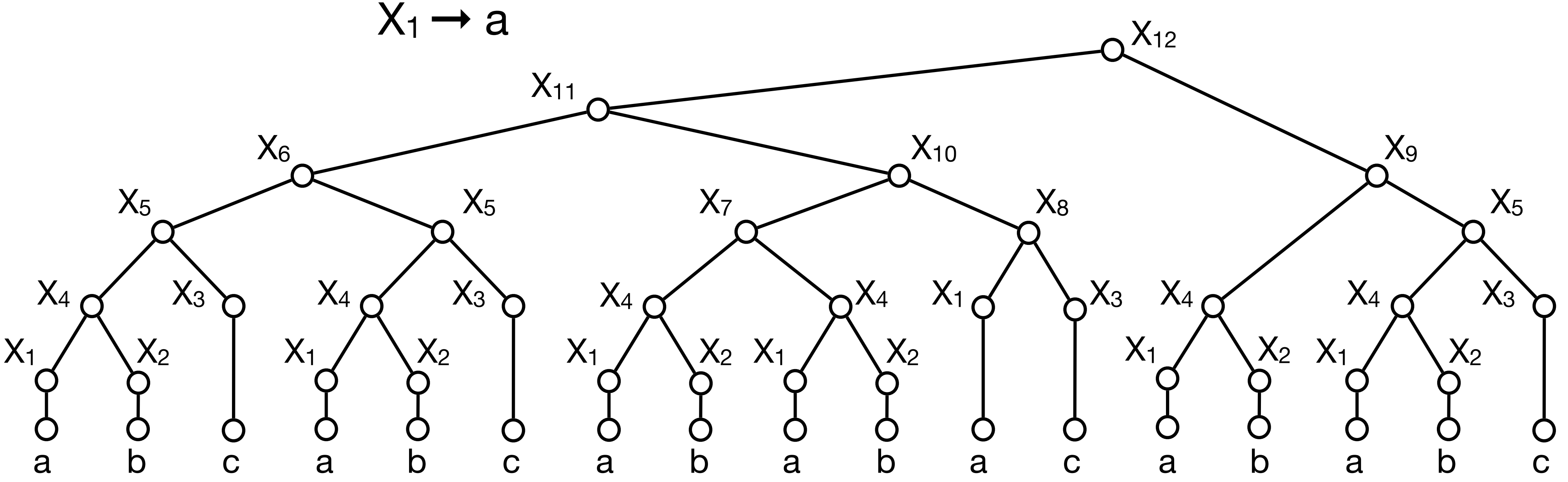
$X_5 \rightarrow X_4X_3$

$X_4 \rightarrow X_1X_2$

$X_3 \rightarrow c$

$X_2 \rightarrow b$

$X_1 \rightarrow a$


X12
X11
X6
X10
X9
X5
X5
X7
X8
X5
X4
X3
X4
X3
X4
X4
X1
X3
X4
X4
X3
X1
X2
X1
X2
X1
X2
X1
X2
X1
X2
X1
X2
a
b
c
a
b
c
a
b
a
b
a
c
a
b
a
b
c