Process Mining without case ids: making sense of unlabeled event logs

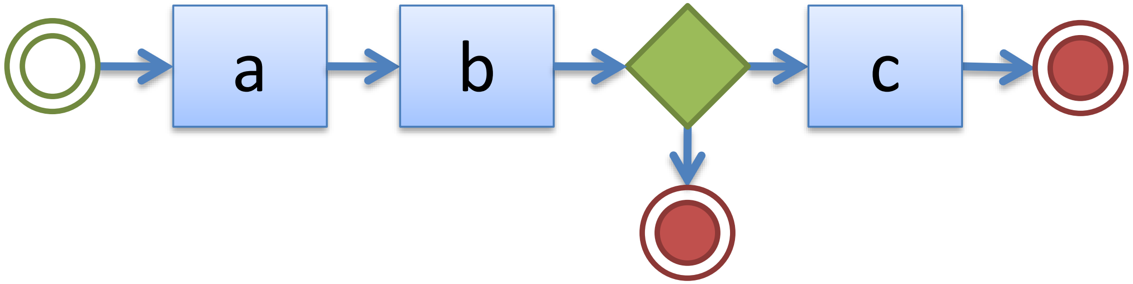
Diogo R. Ferreira

IST – Technical University of Lisbon

Portugal

University of Twente, November 11, 2010

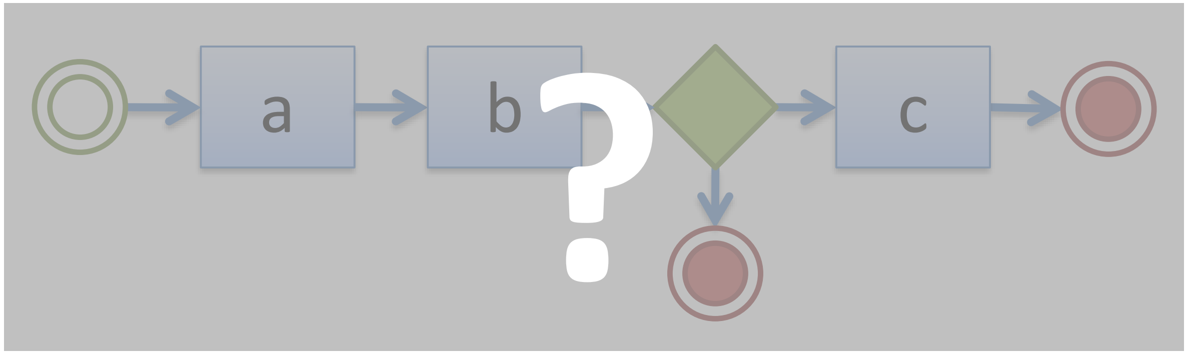
Introduction



Log

case id	*task id*
1	a
1	b
2	a
3	a
2	b
3	b
2	c
4	a
…	…

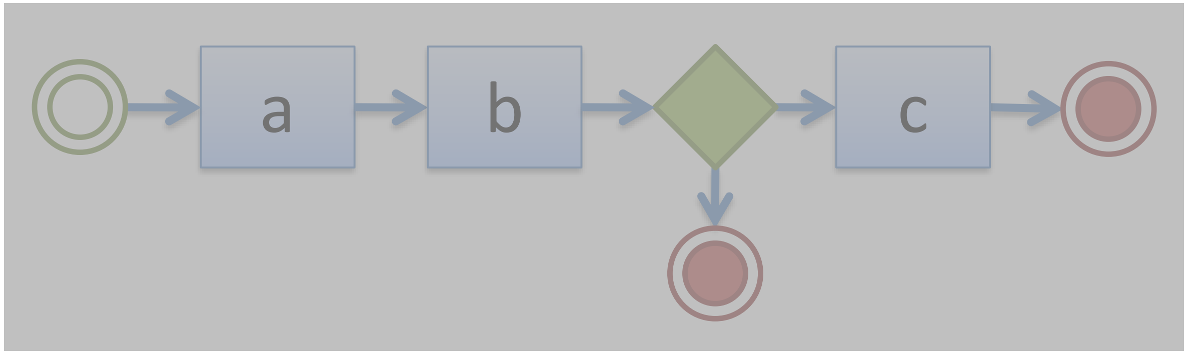
Introduction



Log

case id	*task id*
1	a
1	b
2	a
3	a
2	b
3	b
2	c
4	a
…	…

Introduction



Log

case id	*task id*
1	a
1	b
2	a
3	a
2	b
3	b
2	c
4	a
...	...

?

Sequence partitioning

a b a a b b c a c b

Sequence partitioning

a	**b**	**a**	**a**	**b**	**b**	**c**	**a**	**c**	**b**
a	b						a		b
		a	a	b	b	c		c	

Sequence partitioning

a	b	a	a	b	b	c	a	c	b
a	b						a		b
		a	a	b	b	c		c	

2*ab + 2*abc

Sequence partitioning

a	**b**	**a**	**a**	**b**	**b**	**c**	**a**	**c**	**b**
a		a		b	b	c		c	
	b		a				a		b

2*abc + 1*ba + 1*ab

Sequence partitioning

a	**b**	**a**	**a**	**b**	**b**	**c**	**a**	**c**	**b**
			a			c	a	c	
a				b					
	b	a			b				b

2*ac + 1*ab + 1*ba + 2*b

Sequence partitioning

- Problem

partition of a sequence into
a minimal number of patterns

- Restrictions
  - – patterns with no repeated symbols ~~aba~~
  - – patterns with length of at least 2 symbols ~~b~~
  - – patterns with at least 2 repetitions ~~1*ab~~

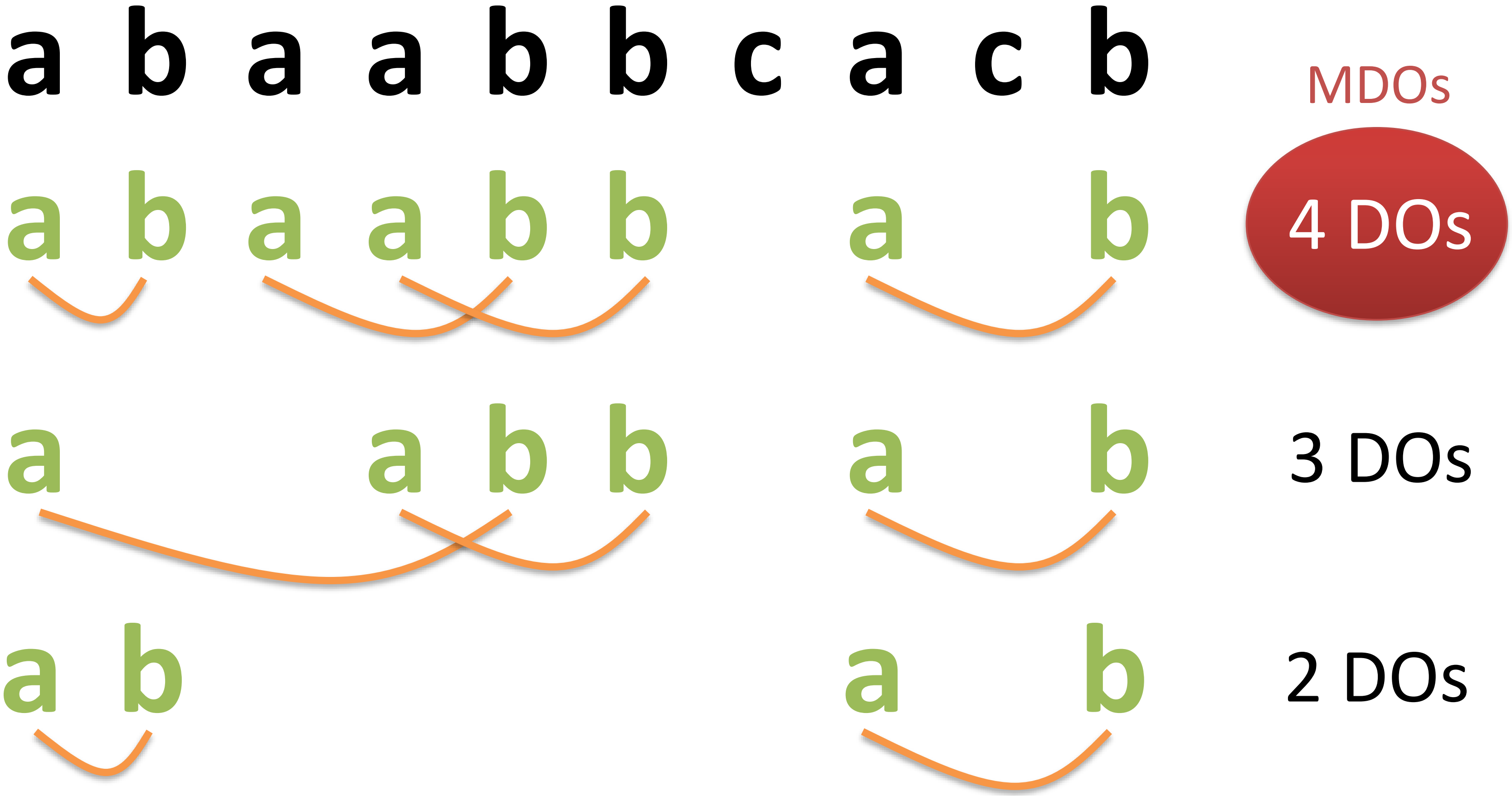
Sequence partitioning

- Approach
  1. get all admissible patterns in the sequence
  2. get all possible occurrences for each pattern
  3. choose a set of *disjoint occurrences* that cover the sequence

- Tools
  - a special data structure (*trie*) for steps 1 and 2
  - Knuth's algorithm X for step 3

Disjoint Occurrences (DOs)

- DOs of pattern **ab**



a b a a b b c a c b

a b a a b b a b

a a b b a b

a b a b

MDOs

4 DOs

3 DOs

2 DOs

Disjoint Occurrences (DOs)

- Questions:
  - which patterns are there in the sequence?
  - how many DOs of each pattern?

- Answer:
  - build the trie

The trie

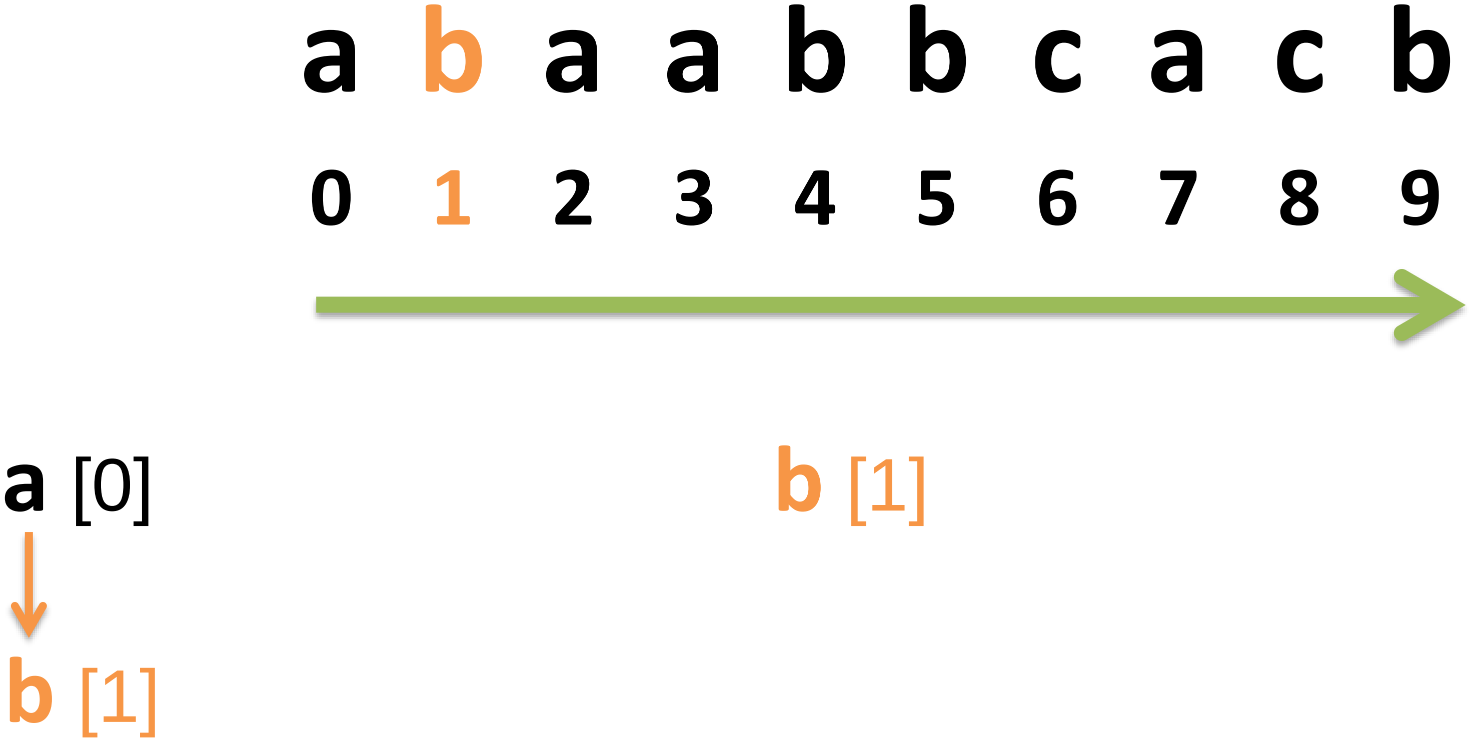
a	**b**	**a**	**a**	**b**	**b**	**c**	**a**	**c**	**b**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

The trie

a	b	a	a	b	b	c	a	c	b
0	1	2	3	4	5	6	7	8	9

a [0]

The trie



a b a a b b c a c b

0 1 2 3 4 5 6 7 8 9

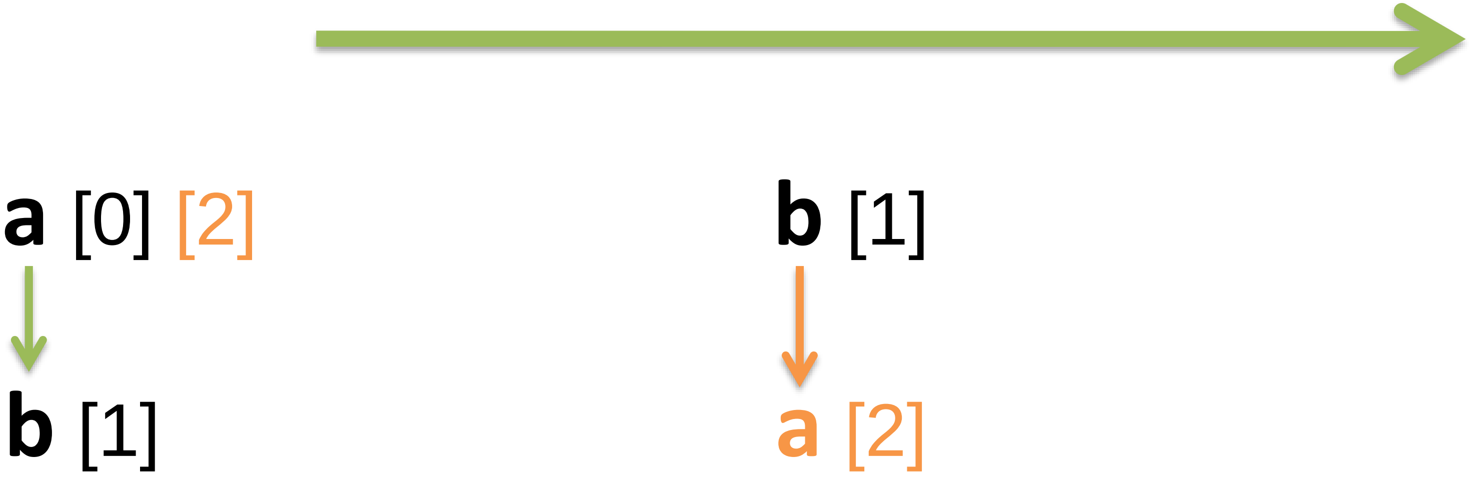
a [0]

b [1]

b [1]

The trie

a	b	a	a	b	b	c	a	c	b
0	1	2	3	4	5	6	7	8	9



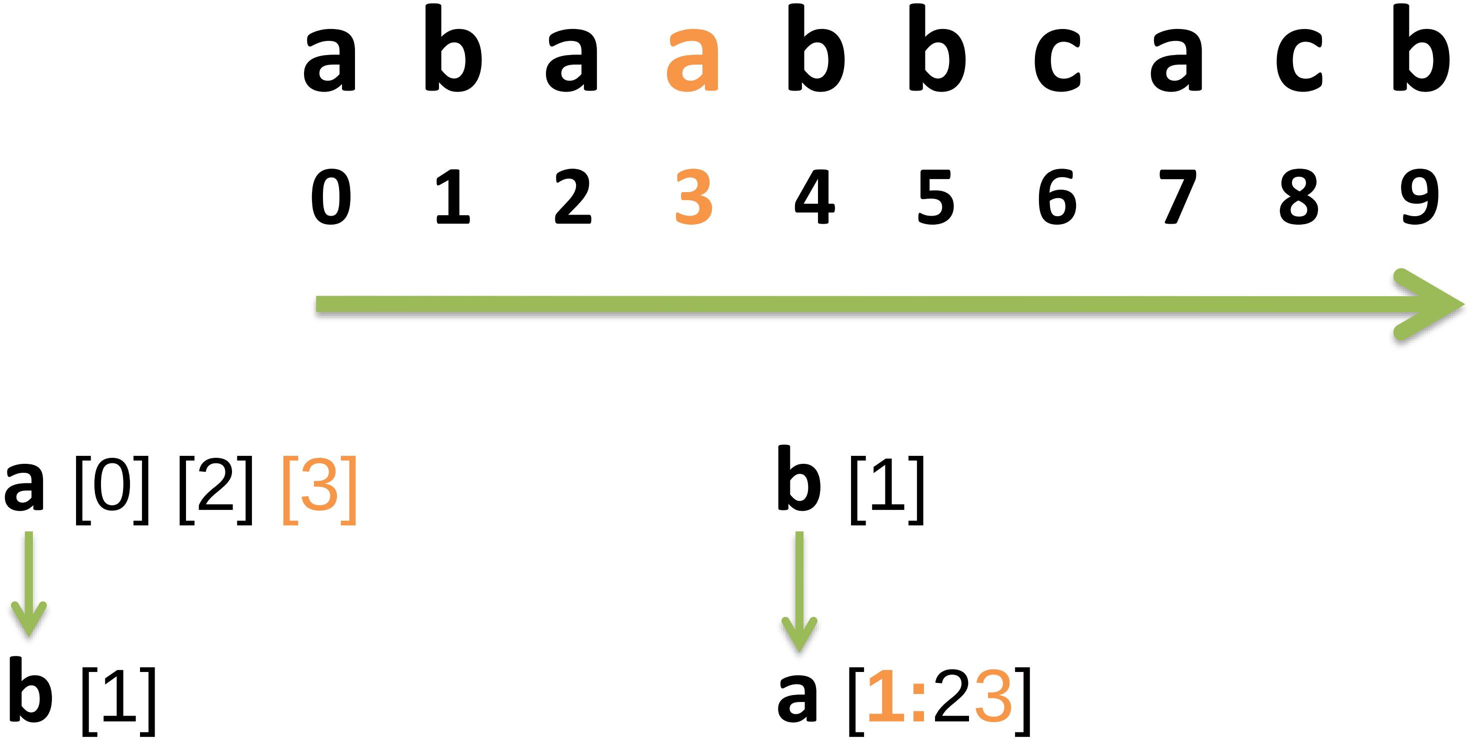
a [0] [2]

b [1]

b [1]

a [2]

The trie



a	b	a	a	b	b	c	a	c	b
0	1	2	3	4	5	6	7	8	9

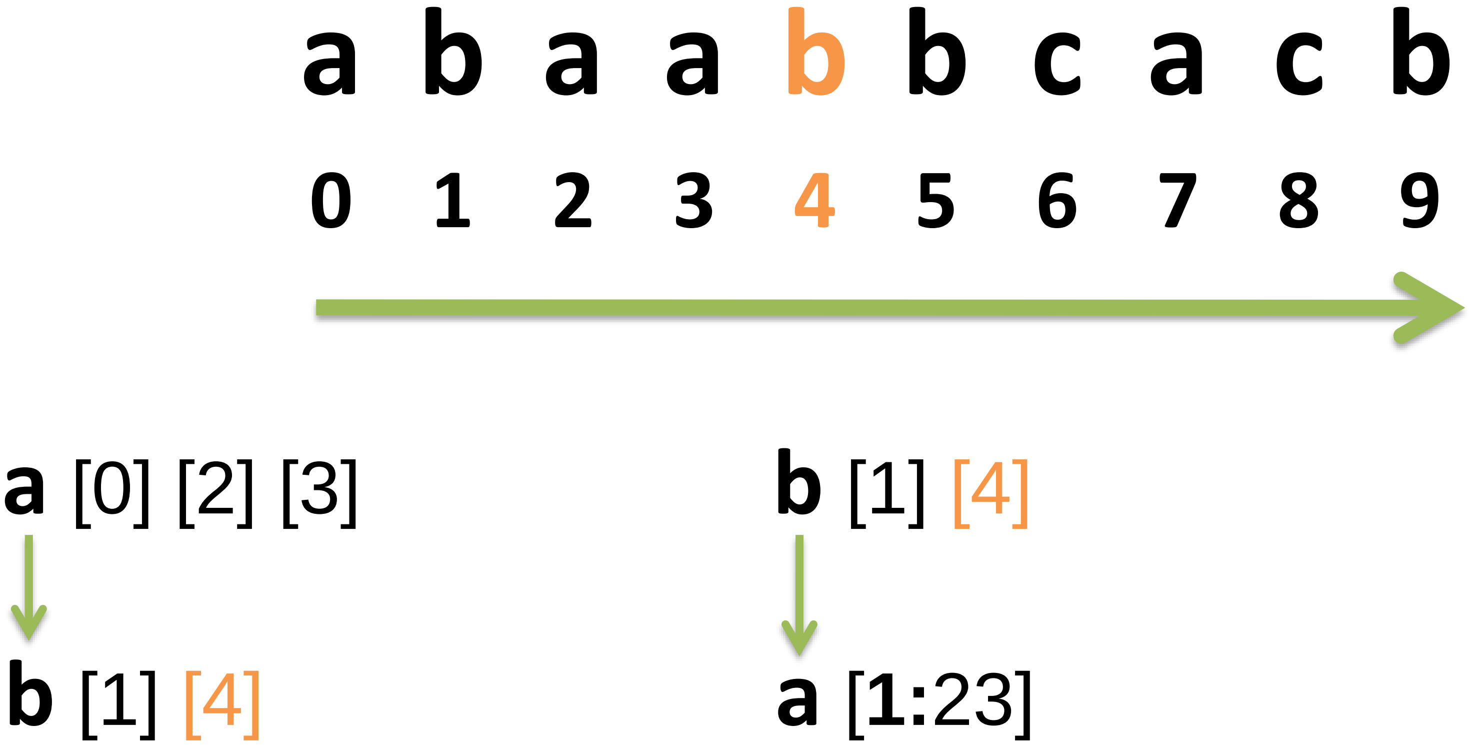
a [0] [2] [3]

b [1]

b [1]

a [1:23]

The trie



a b a a b b c a c b

0 1 2 3 4 5 6 7 8 9

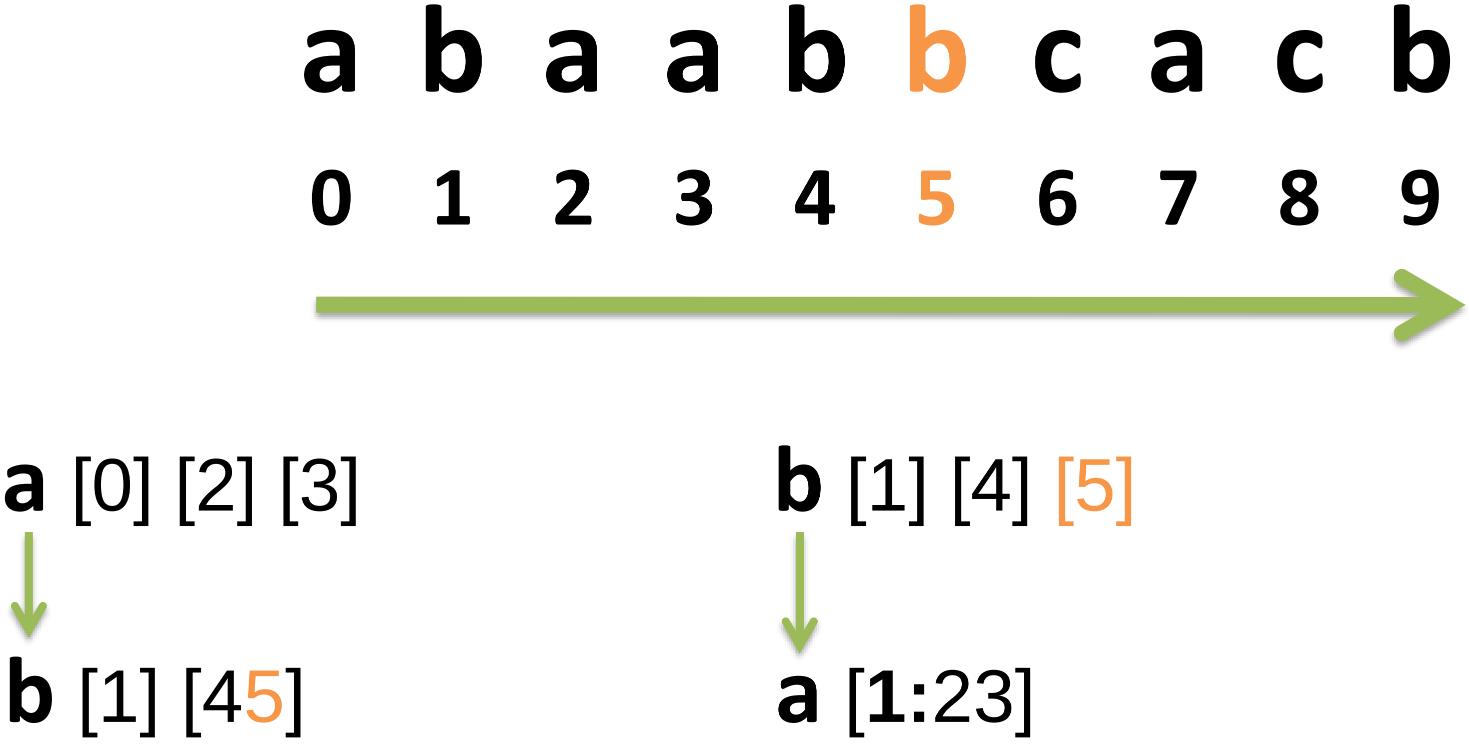
a [0] [2] [3]

b [1] [4]

b [1] [4]

a [**1:**23]

The trie



a	b	a	a	b	b	c	a	c	b
0	1	2	3	4	5	6	7	8	9

a [0] [2] [3]

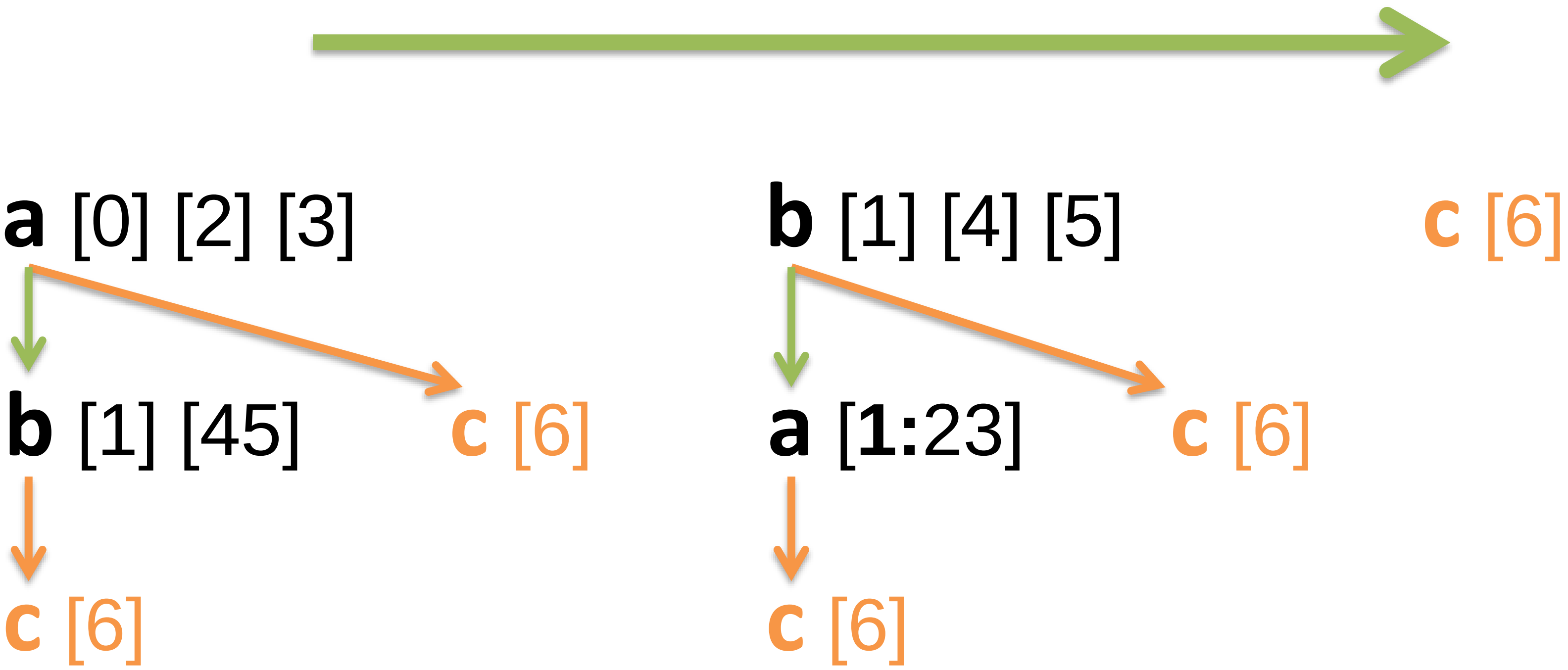
b [1] [45]

b [1] [4] [5]

a [**1:**23]

The trie

a	**b**	**a**	**a**	**b**	**b**	**c**	**a**	**c**	**b**
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**



a [0] [2] [3]

b [1] [45]

c [6]

c [6]

b [1] [4] [5]

a [**1:**23]

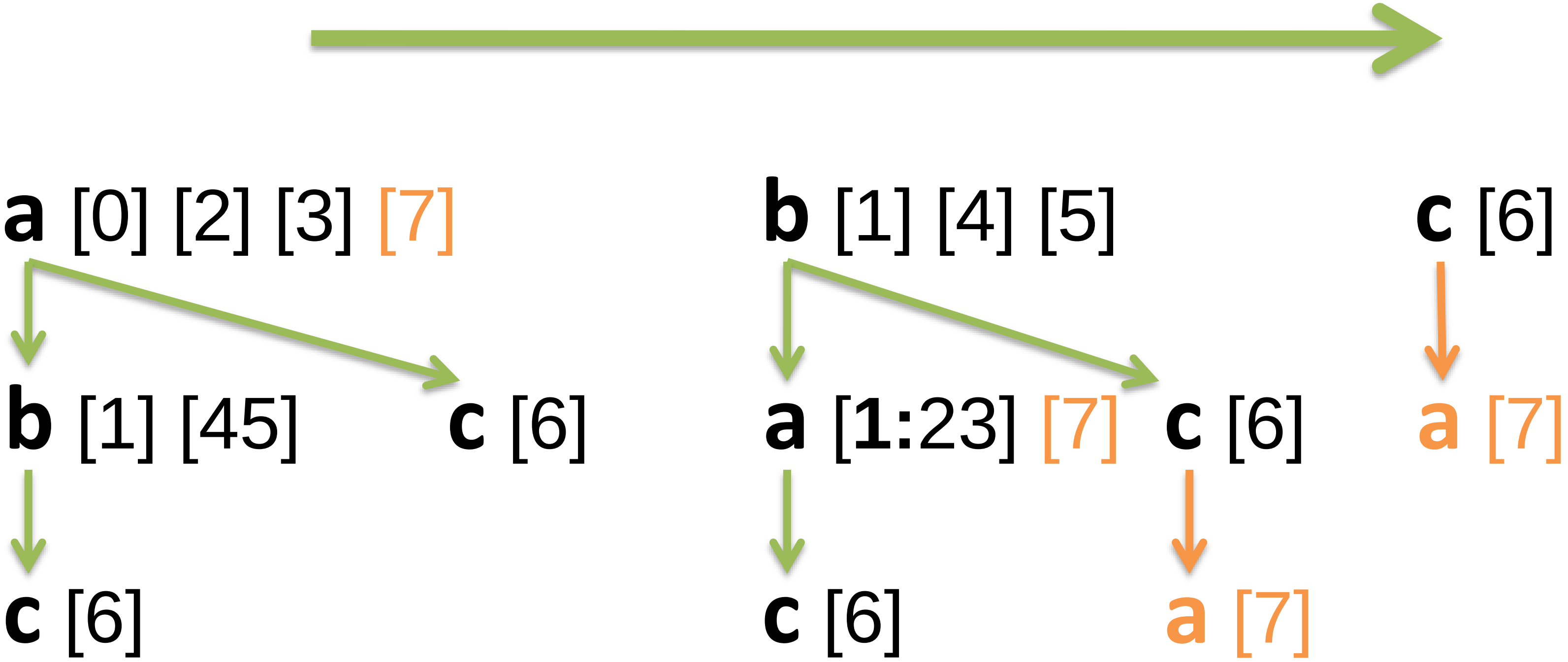
c [6]

c [6]

c [6]

The trie

a	b	a	a	b	b	c	a	c	b
0	1	2	3	4	5	6	7	8	9



a [0] [2] [3] [7]

b [1] [45]

c [6]

c [6]

b [1] [4] [5]

a [1:23] [7]

c [6]

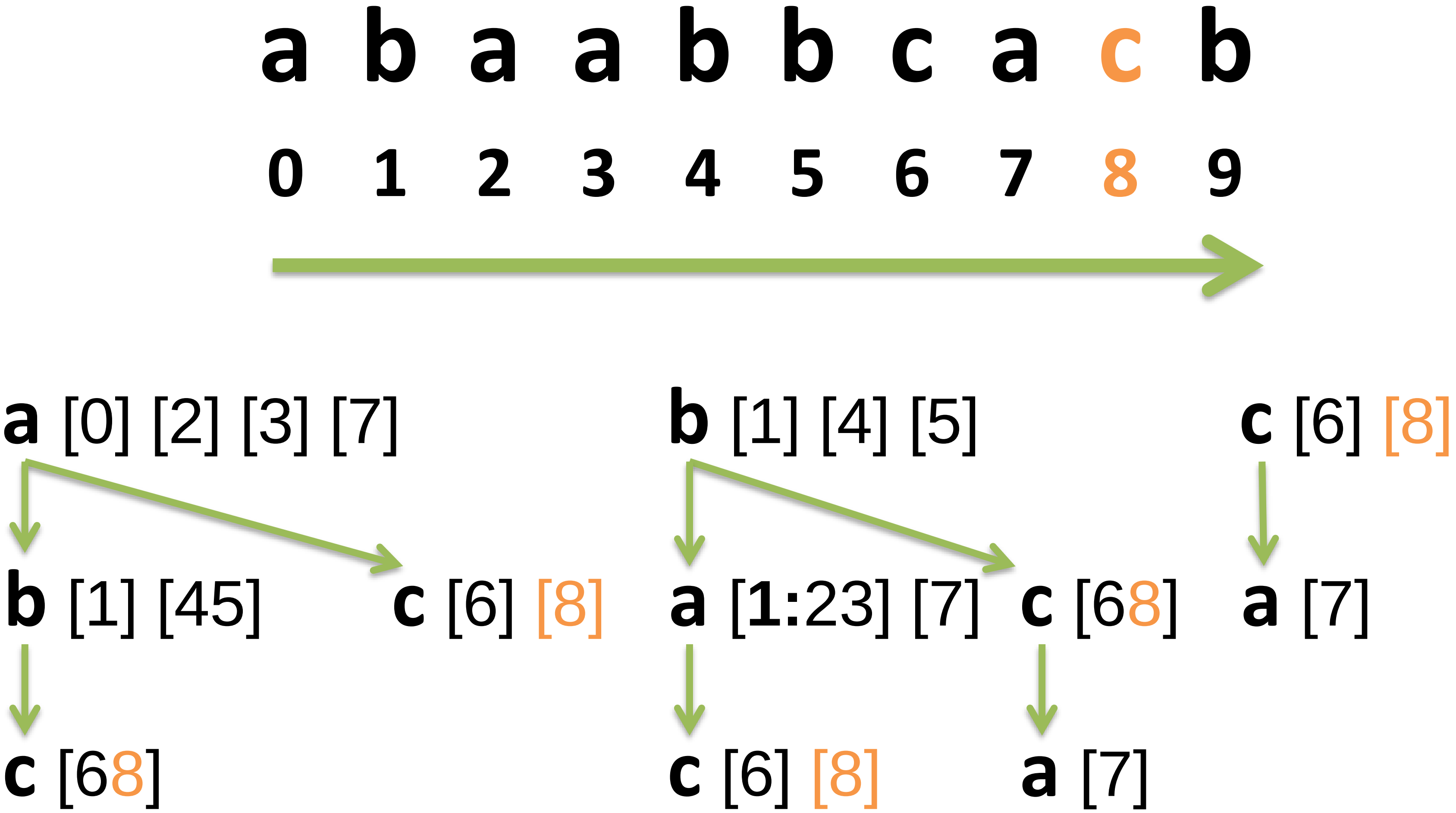
c [6]

a [7]

c [6]

a [7]

The trie



a b a a b b c a c b

0 1 2 3 4 5 6 7 8 9

a [0] [2] [3] [7]

b [1] [4] [5]

c [6] [8]

b [1] [45]

c [6] [8]

a [**1:**23] [7]

c [68]

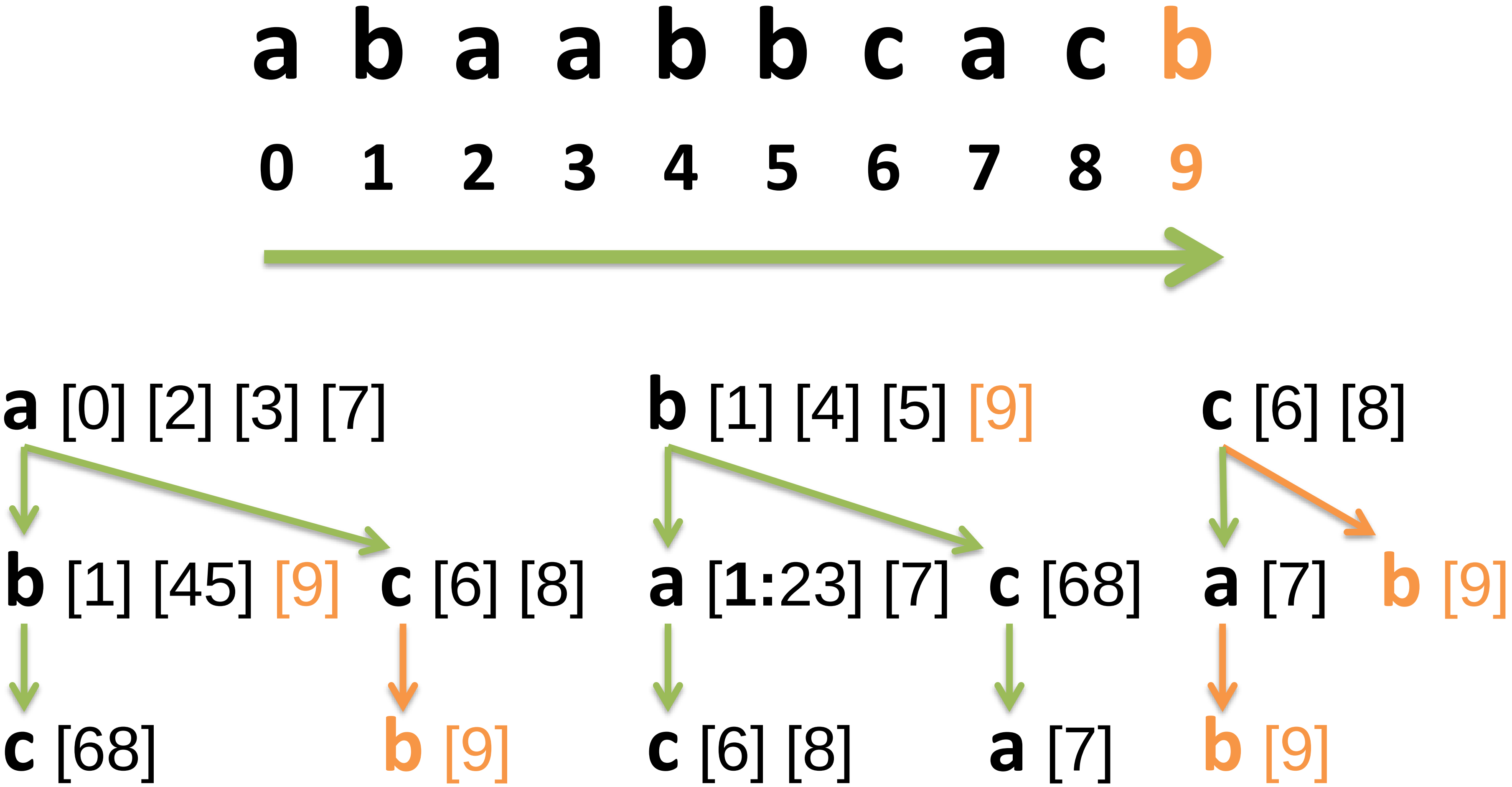
a [7]

c [68]

c [6] [8]

a [7]

The trie



a b a a b b c a c b

0 1 2 3 4 5 6 7 8 9

a [0] [2] [3] [7]

b [1] [45] [9]

c [68]

c [6] [8]

b [9]

b [1] [4] [5] [9]

a [**1:**23] [7]

c [6] [8]

c [68]

a [7]

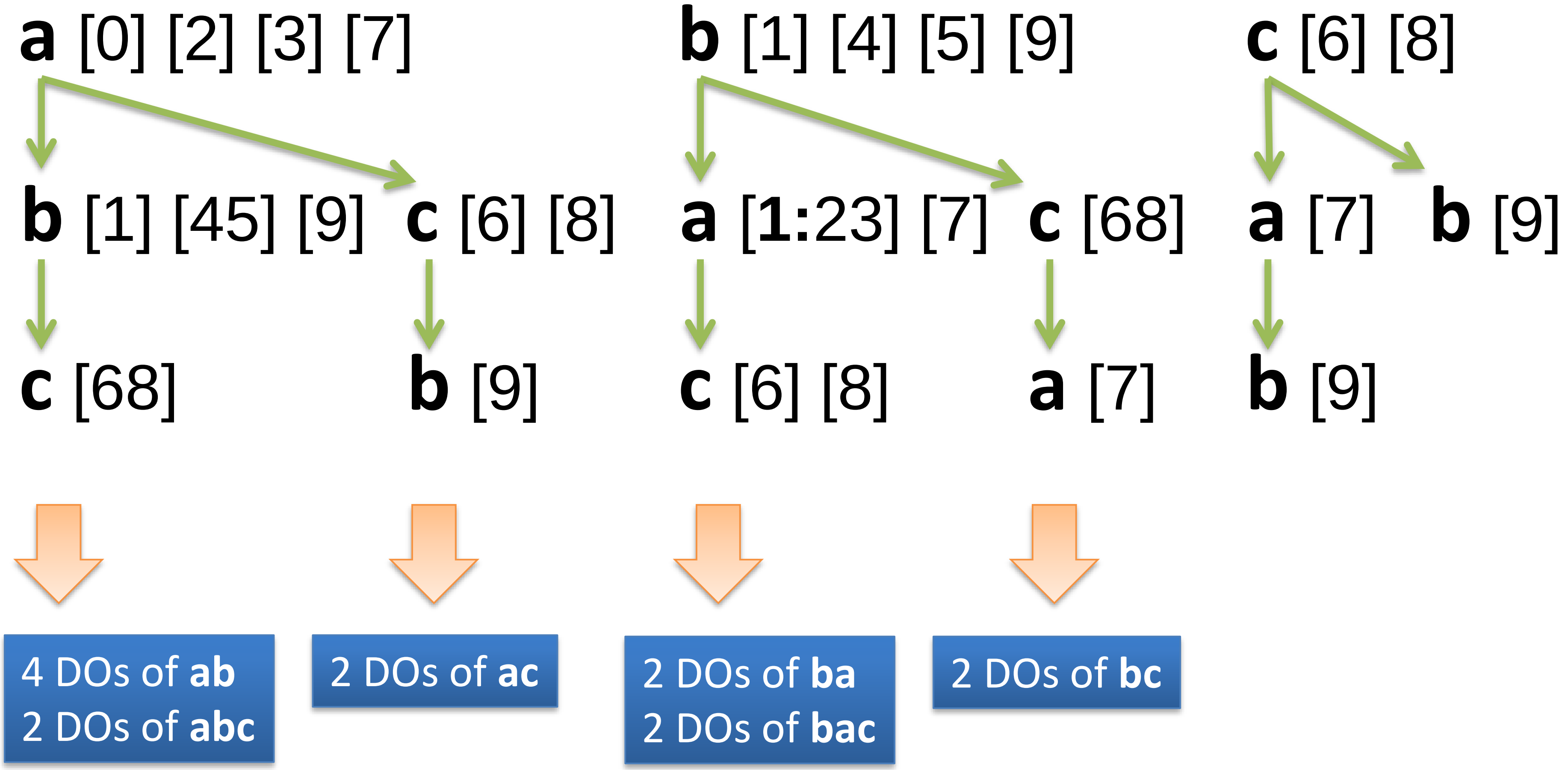
c [6] [8]

a [7]

b [9]

b [9]

The trie



a [0] [2] [3] [7]

b [1] [4] [5] [9]

c [6] [8]

b [1] [45] [9]

c [6] [8]

a [**1:**23] [7]

c [68]

a [7]

b [9]

c [68]

b [9]

c [6] [8]

a [7]

b [9]

4 DOs of **ab**
2 DOs of **abc**

2 DOs of **ac**

2 DOs of **ba**
2 DOs of **bac**

2 DOs of **bc**

Sequence Partitioning

- Question:
  - which set of DOs covers the sequence?

- Answer:
  - Knuth's algorithm X

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	1	1	1	1	0	1	0	1	4
	1	1	1	1	1	1	0	0	0	0	3
	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	1	1	0	1	0	1	1	0	1	0	2
	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	0	1	0	0	1	0	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	1	1	1	1	0	1	0	1	4
	1	1	1	1	1	1	0	0	0	0	3
	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	1	1	0	1	0	1	1	0	1	0	2
	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	0	1	0	0	1	0	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	1	1	1	1	0	1	0	1	4
	1	1	1	1	1	1	0	0	0	0	3
	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	1	1	0	1	0	1	1	0	1	0	2
	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	0	1	0	0	1	0	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	1	1	1	1	0	1	0	1	4
	1	1	1	1	1	1	0	0	0	0	3
	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	1	1	0	1	0	1	1	0	1	0	2
	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	0	1	0	0	1	0	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	1	1	1	1	0	1	0	1	4
	1	1	1	1	1	1	0	0	0	0	3
	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	1	1	0	1	0	1	1	0	1	0	2
	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	0	1	0	0	1	0	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	1	1	1	1	0	1	0	1	4
	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~3~~
	~~1~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~0~~	~~1~~	~~2~~
	…	…	…	…	…	…	…	…	…	…	
abc	~~1~~	~~1~~	~~0~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~1~~	~~0~~	~~2~~
	~~0~~	~~0~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~1~~	~~0~~	~~2~~
	…	…	…	…	…	…	…	…	…	…	
bc	~~0~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	~~1~~	~~0~~	~~1~~	~~0~~	~~2~~
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**4**
	…	…	…	…	…	…	…	…	…	…	
abc	…	…	…	…	…	…	…	…	…	…	
bc	…	…	…	…	…	…	…	…	…	…	

- No row covers **2*c**
  - only the following row would work but it does not exist:

c	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**2**

- Therefore, no solution by taking the first row!

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	1	1	1	1	0	1	0	1	4
	1	1	1	1	1	1	0	0	0	0	3
	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	1	1	0	1	0	1	1	0	1	0	2
	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	0	1	0	0	1	0	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	1	1	1	1	0	1	0	1	4
	1	1	1	1	1	1	0	0	0	0	3
	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	1	1	0	1	0	1	1	0	1	0	2
	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	0	1	0	0	1	0	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	1	1	1	1	0	1	0	1	4
	1	1	1	1	1	1	0	0	0	0	3
	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	1	1	0	1	0	1	1	0	1	0	2
	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	0	1	0	0	1	0	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~1~~	~~0~~	~~1~~	~~4~~
	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~0~~	~~0~~	~~0~~	~~0~~	~~3~~
	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	~~1~~	~~1~~	~~0~~	~~1~~	~~0~~	~~1~~	~~1~~	~~0~~	~~1~~	~~0~~	~~2~~
	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	~~0~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	~~1~~	~~0~~	~~1~~	~~0~~	~~2~~
	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	…	…	…	…	…	…	…	…	…	…	

Knuth's algorithm X

	a	b	a	a	b	b	c	a	c	b	
ab	1	1	0	0	0	0	0	1	0	1	2
	…	…	…	…	…	…	…	…	…	…	
abc	0	0	1	1	1	1	1	0	1	0	2
	…	…	…	…	…	…	…	…	…	…	
bc	…	…	…	…	…	…	…	…	…	…	

2*ab + 2*abc

Using only MDOs

- Hypothesis

use only MDOs!?

- Positive side
  - drastically reduces the number of rows
  - search becomes much faster
- Negative side
  - may not find all (or even any) solutions

Using only MDOs

a b a a b b c a c b

- All subsets of DOs
  - 58 rows
  - 2 solutions
    - 2*ab + 2*abc
    - 2*ab + 2*bac

- Only MDOs
  - 25 rows
  - 0 solutions

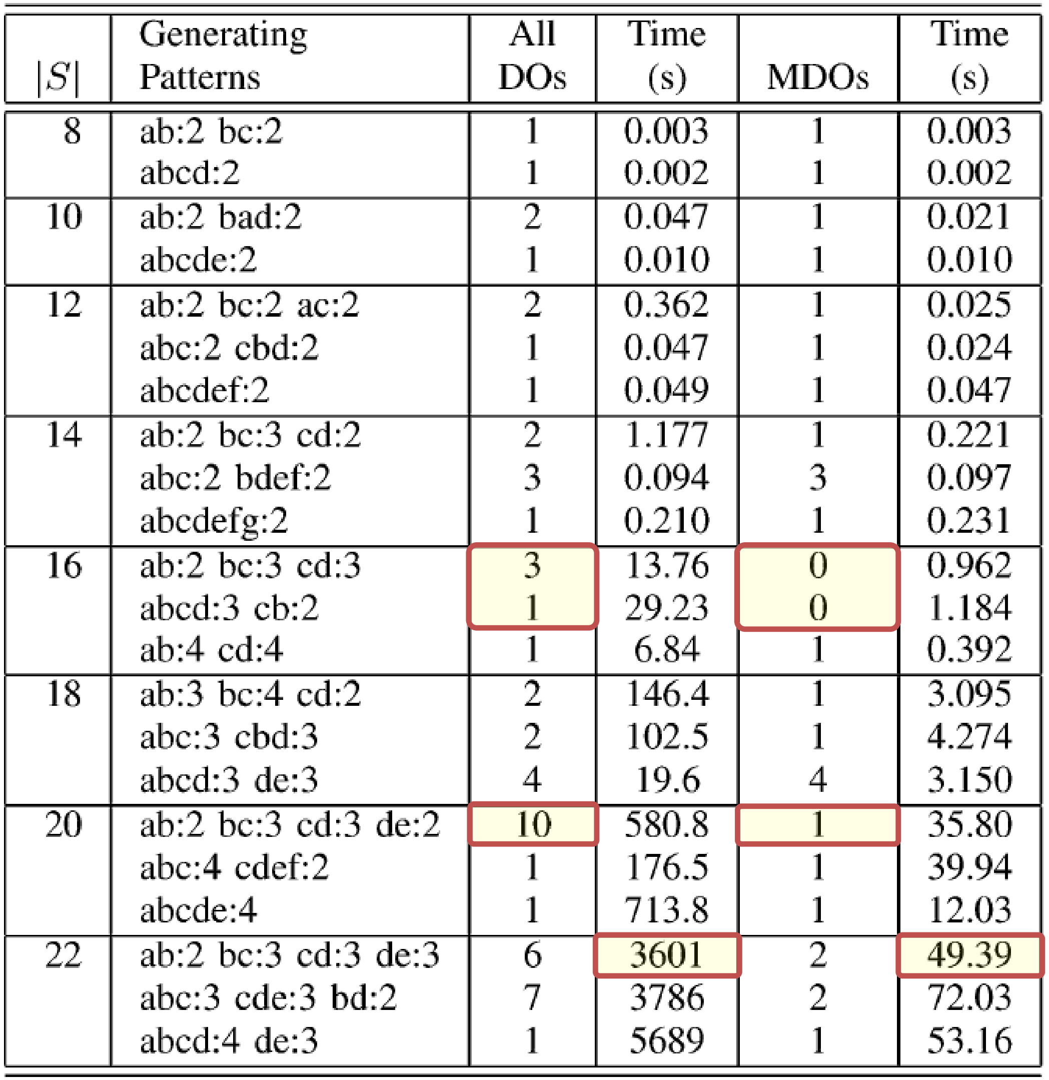
Using only MDOs

a b a a b b c a c b a a

- All subsets of DOs
  - 124 rows
  - 2 solutions
    - 2*ac + 4*ba
    - 2*ca + 4*ab

- Only MDOs
  - 35 rows
  - 2 solutions
    - 2*ac + 4*ba
    - 2*ca + 4*ab

Test runs



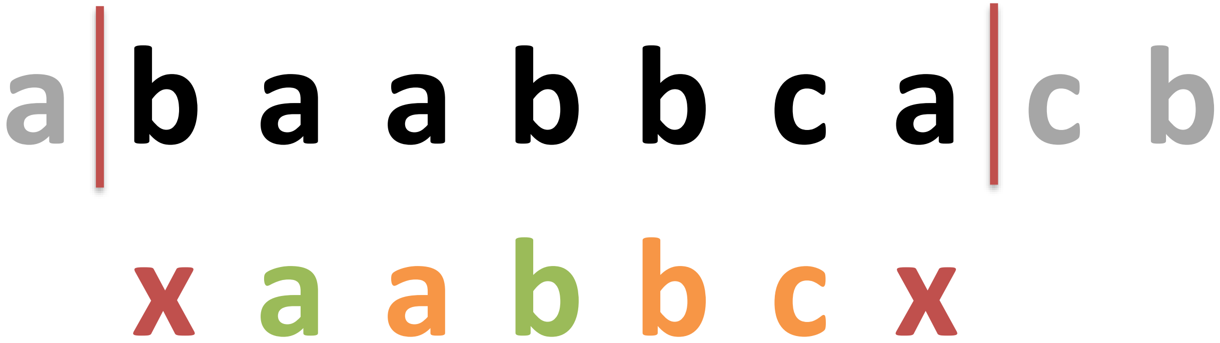
$\|S\|$	Generating Patterns	All DOs	Time (s)	MDOs	Time (s)
8	ab:2 bc:2	1	0.003	1	0.003
	abcd:2	1	0.002	1	0.002
10	ab:2 bad:2	2	0.047	1	0.021
	abcde:2	1	0.010	1	0.010
12	ab:2 bc:2 ac:2	2	0.362	1	0.025
	abc:2 cbd:2	1	0.047	1	0.024
	abcdef:2	1	0.049	1	0.047
14	ab:2 bc:3 cd:2	2	1.177	1	0.221
	abc:2 bdef:2	3	0.094	3	0.097
	abcdefg:2	1	0.210	1	0.231
16	ab:2 bc:3 cd:3	3	13.76	0	0.962
	abcd:3 cb:2	1	29.23	0	1.184
	ab:4 cd:4	1	6.84	1	0.392
18	ab:3 bc:4 cd:2	2	146.4	1	3.095
	abc:3 cbd:3	2	102.5	1	4.274
	abcd:3 de:3	4	19.6	4	3.150
20	ab:2 bc:3 cd:3 de:2	10	580.8	1	35.80
	abc:4 cdef:2	1	176.5	1	39.94
	abcde:4	1	713.8	1	12.03
22	ab:2 bc:3 cd:3 de:3	6	3601	2	49.39
	abc:3 cde:3 bd:2	7	3786	2	72.03
	abcd:4 de:3	1	5689	1	53.16

Conclusion

- Non-issues
  - multiple solutions (keep only minimal ones)
  - loops (body of loop in separate pattern)
  - parallelism (more patterns will appear)
- Issues
  - no. of rows (choices for the DOs of patterns)
  - run-time (use MDOs, but...)
  - truncated sequences (use fringes)

Conclusion

- Truncated sequences



a b a a b b c a c b

x a a b b c x

  – admit that *F* symbols cannot be covered (fringe)
  – adapt algorithm X to cover $|S| - F$

More info…

- Michal Walicki, Diogo R. Ferreira, *Mining Sequences for Patterns with Non-Repeating Symbols*, IEEE Congress on Evolutionary Computation 2010 (IEEE World Congress on Computational Intelligence), pp. 3269-3276, Barcelona, Spain, July 18-23, 2010
- Diogo R. Ferreira, Daniel Gillblad, *Discovering Process Models from Unlabelled Event Logs*, Proceedings of the 7th International Conference on Business Process Management (BPM 2009), LNCS 5701, pp. 143-158, Springer, 2009

Thank you!

Questions?