An Application of Stream Compression

— speeding up of data transmission —

Hiroshi Sakamoto

KYUTECH (Kyushu Institute of Technology)

joint work with

S.Maruyama (PFI)

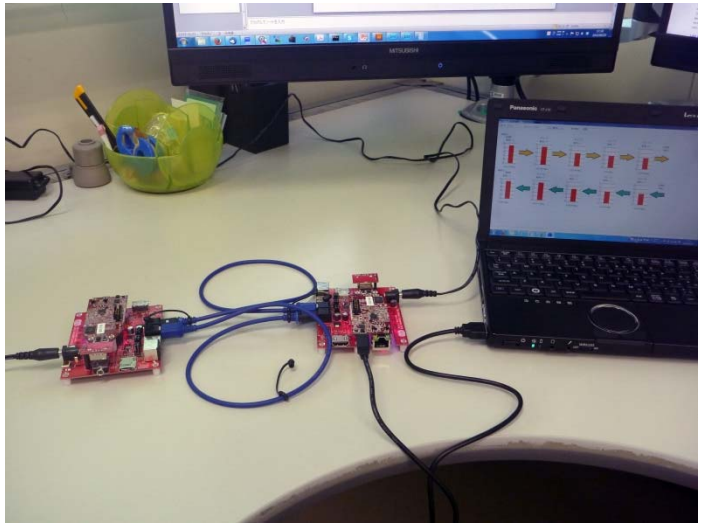
Y.Tabei (JST ERATO)

T.Kida (Hokkaido University)

K.Sadakane (NII)

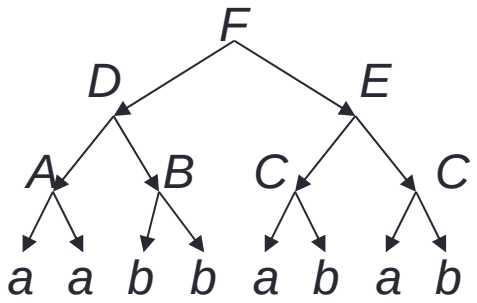
M.Takeda (Kyushu University)

S.Yamagiwa (University of Tsukuba)

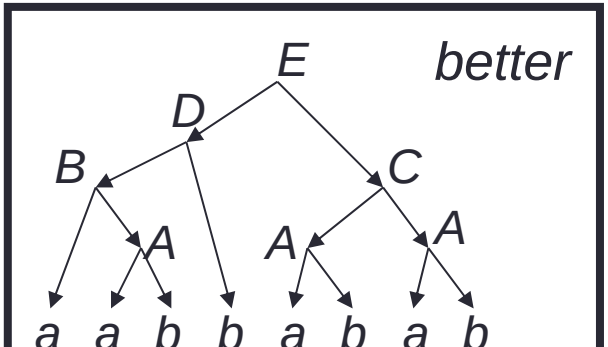


Introduction to SLP

- Grammar compression: CFG generating a single string

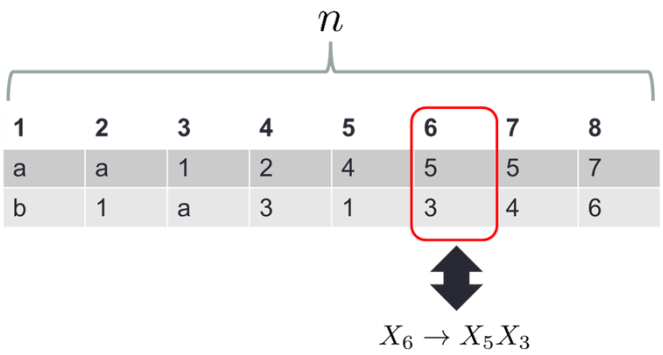


size(G)=2×6=12



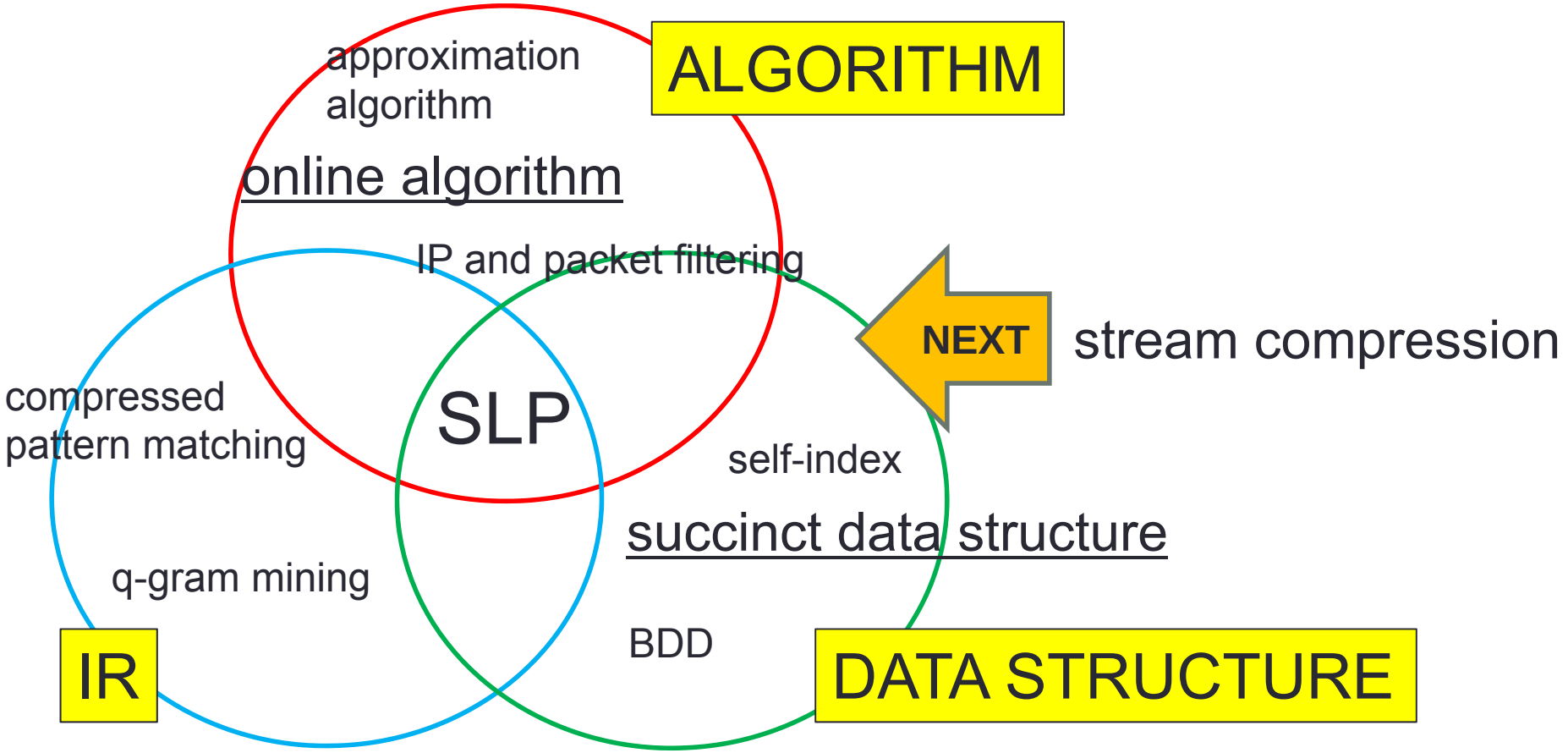
size(G)=2×5=10

- SLP: a canonical form of grammar compression
  - Straight Line Program (SLP) $X_k \to X_i X_j \ \ (k > i, j)$
  - A naïve representation by array requires $2n \log n$ bits



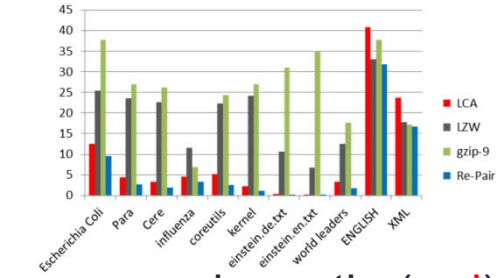
SLP World

- Wide relation to many researches

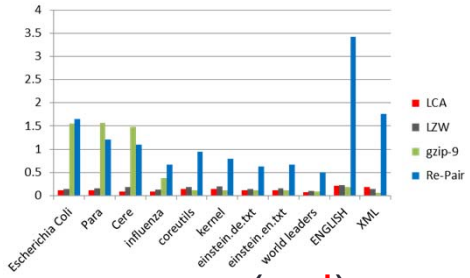


Why is SLP important?

- First: small space and simple algorithm

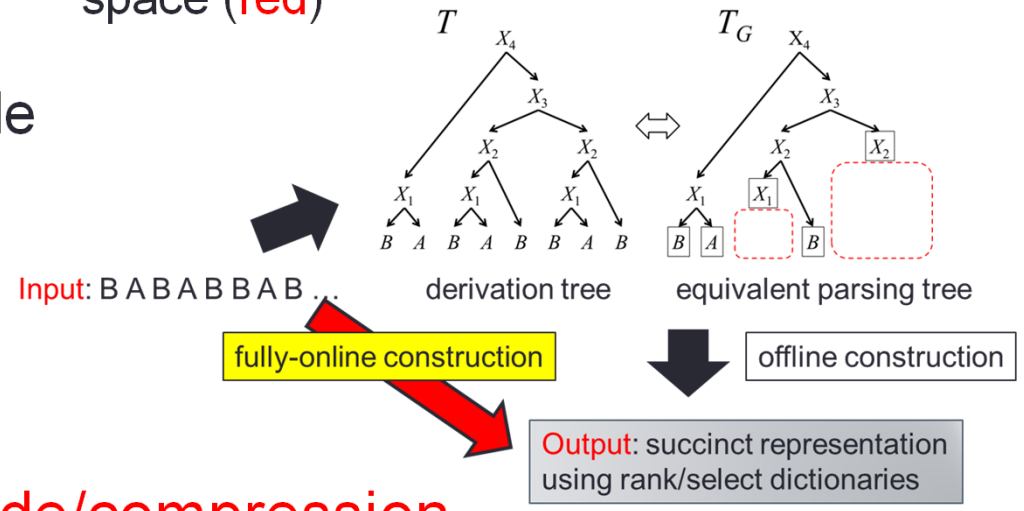


compression ratio (red)



space (red)

- Second: online constructable



- NEXT: Real-time Stream de/compression
  for fast data transmission

Recent Results on SLP

- [Tabei et al., CPM'13]
  Lower bound: The information-theoretic lower bound of SLP(n) in

$$\log((n-1)!) + 2n + o(n) \text{ bits}$$

- [Maruyama et al., SPIRE'13]
  Upper bound: Fully-online construction of SLP(n) in

$$n \log n + 2n + o(n) \text{ bits}$$

SLP(n): the set of SLPs with n characters

n

1	2	3	4	5	6	7	8
a	a	1	2	4	5	5	7
b	1	a	3	1	3	4	6

Lower bound: idea

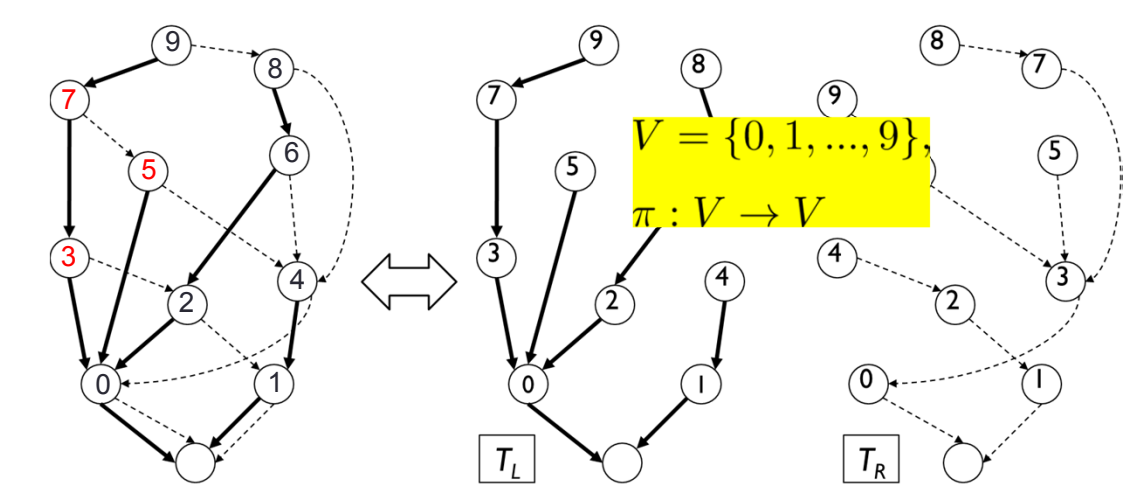
- The information-theoretic lower bound is $\log |SLP(n)|$
- How to count?
- Idea: spanning tree decomposition of SLP

SLP by array

0	1	2	3	4	5	6	7	8	9
		0	0	1	0	2	3	6	7
		1	2	0	4	4	5	4	8

SLP by DAG

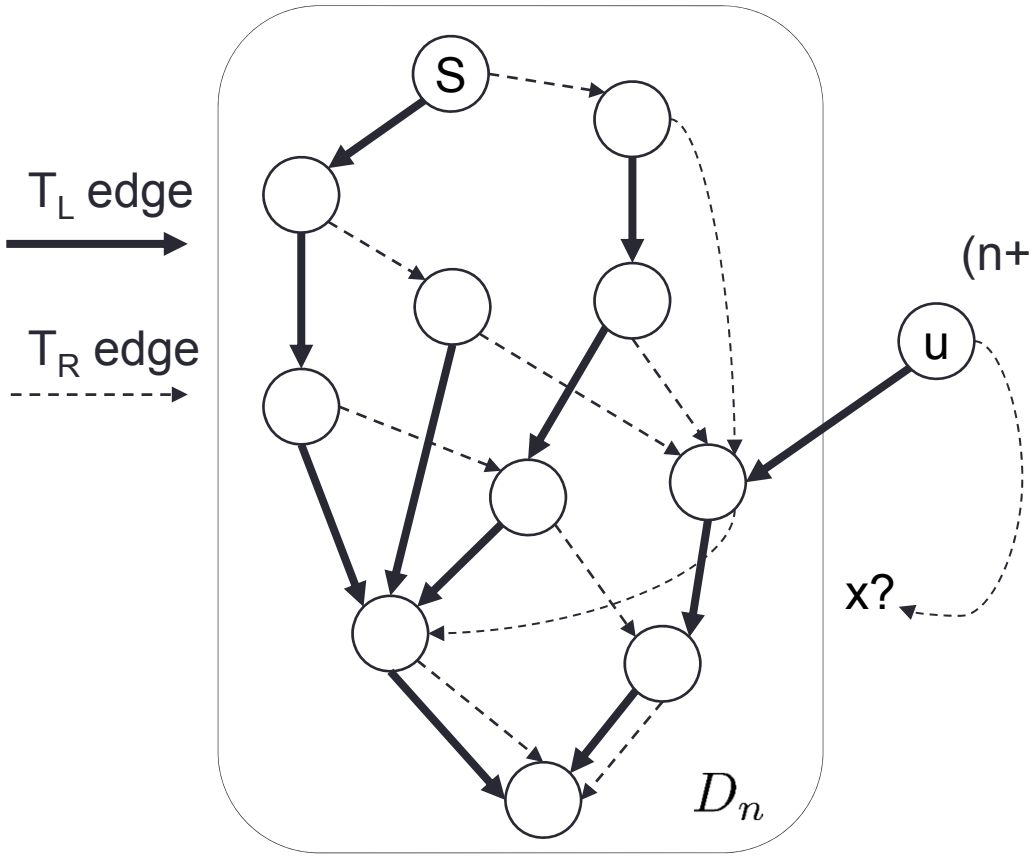
How many admissible π?



single source/sink dag

Lower bound: refinement of DAG

Induction on size n



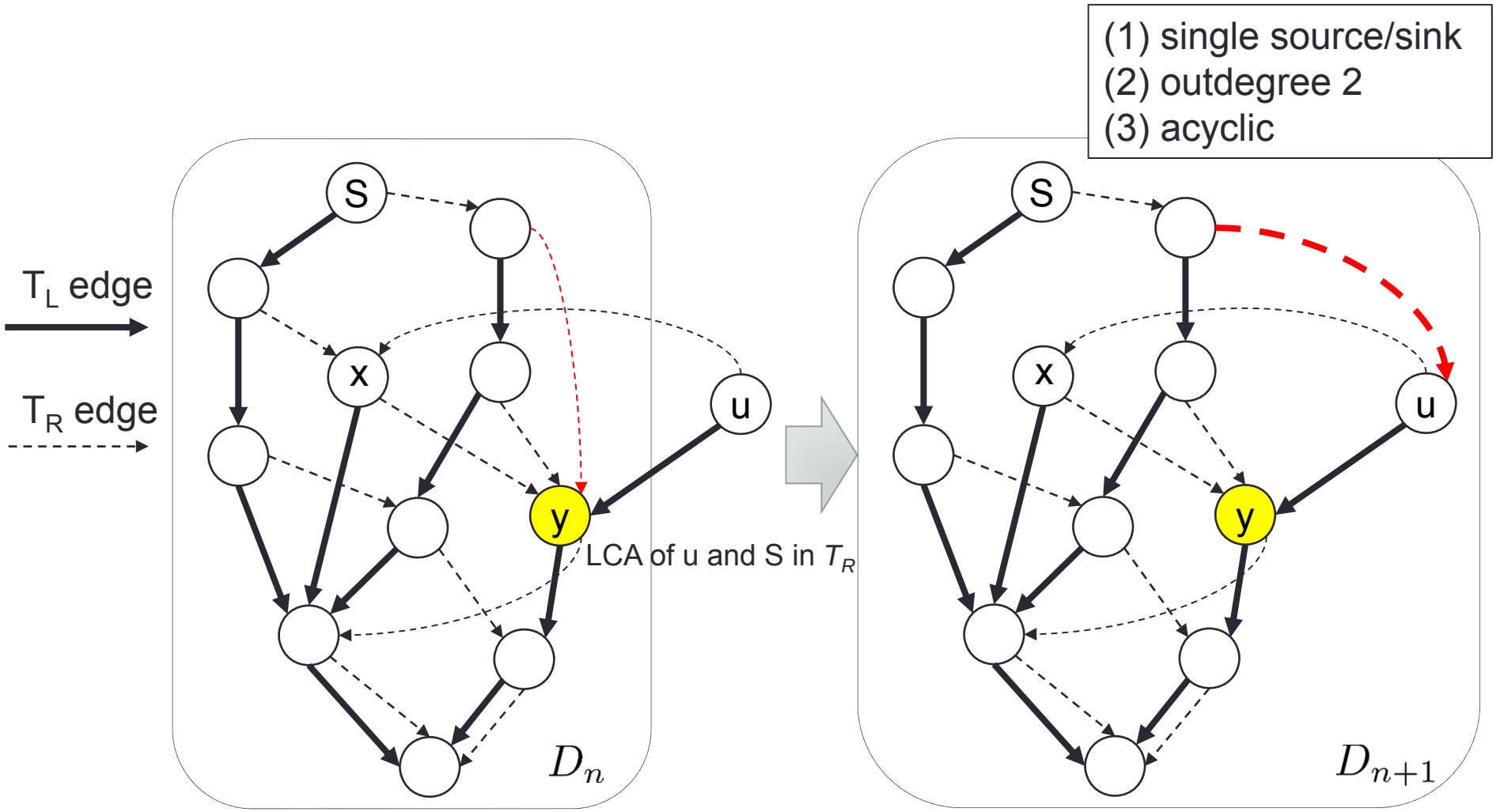
(n+1)-th node with fixed left edge

D_{n+1}

Admissible DAG for the right edge?

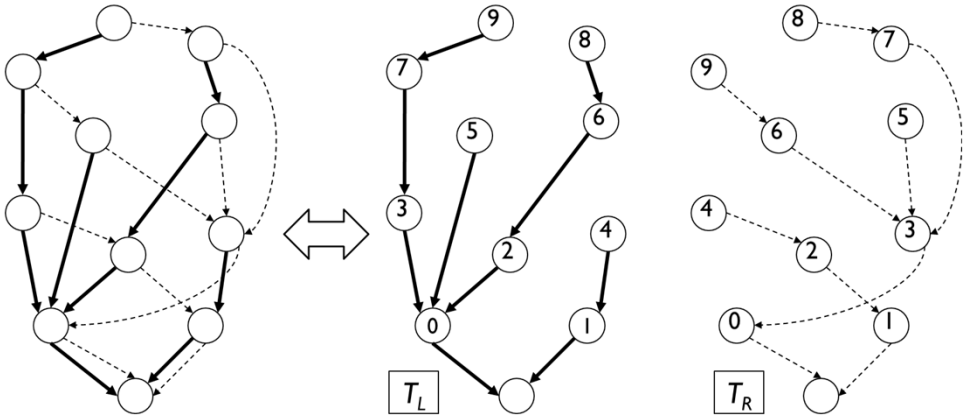
(1) single source/sink
(2) outdegree 2
(3) acyclic

Lower bound: example of refinement



Lower bound: result

- $S(n, T_L)$: subset of $SLP(n)$ with fixed T_L
- $|S(n, T_L)| \geq (n-1)!$
- $\#T_L = 2^{2n}$
- $|SLP(n)| = 2^{2n}(n-1)!$

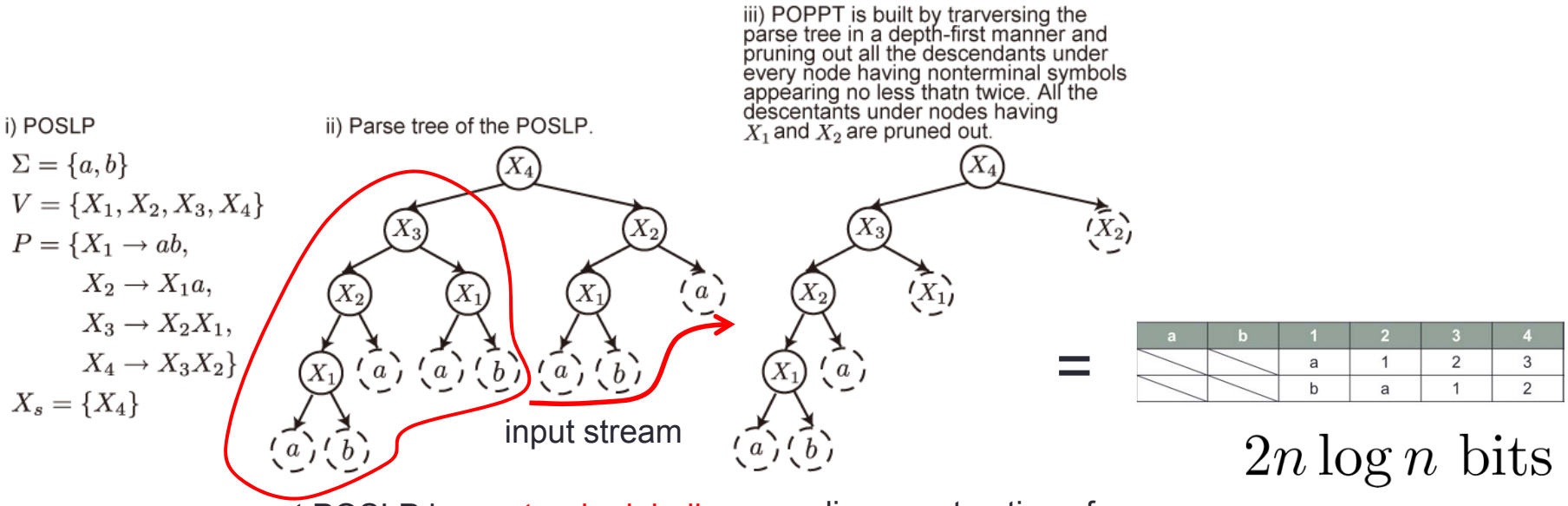


$n \log n + n + o(n)$ bits lower bound

next question: Can we get such a small representation actually?
→ our next result is YES by online

Upper bound: idea

- [Maruyama et al., Algorithms 2012]
  Post order partial parse tree (POPPT)



current POSLP by post order labeling of internal nodes

online construction of POPPT equivalent POSLP by removing frequent subtrees

Upper bound: fully-online construction

- [Maruyama et al., SPIRE'13]
  Succinct representation by rank/select

i) POSLP

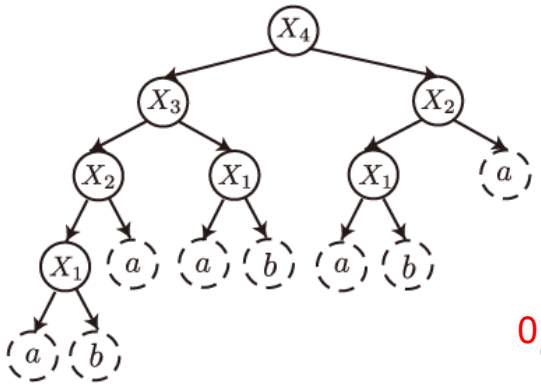
$\Sigma = \{a, b\}$

$V = \{X_1, X_2, X_3, X_4\}$

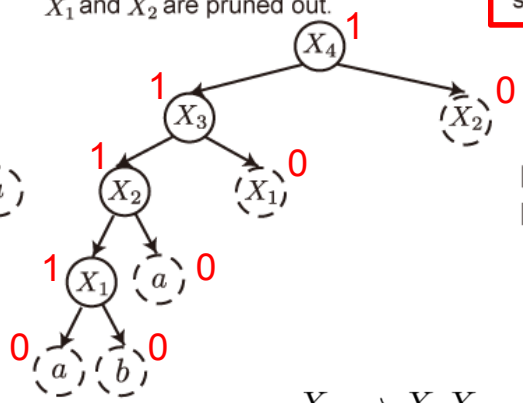
$P = \{X_1 \to ab,$
$X_2 \to X_1 a,$
$X_3 \to X_2 X_1,$
$X_4 \to X_3 X_2\}$

$X_s = \{X_4\}$

ii) Parse tree of the POSLP.



iii) POPPT is built by trarversing the parse tree in a depth-first manner and pruning out all the descendants under every node having nonterminal symbols appearing no less thatn twice. All the descentants under nodes having X_1 and X_2 are pruned out.



iv) Succinct representation of the POPPT. B is a bit string built by traversing the POPPT and putting bit '0' for a leaf and bit '1' for aninternal node. L is a label sequence of leaves.

12345678910
B:0010101011
L:abaX$_1$X$_2$

decompressed by

$X_k \to X_i X_j,$

e.g. the left child i is decompressed by

$$i = \begin{cases} rank_1(B, q_\ell), & \text{if } B[q_\ell] = 1, \\ L[select_0(B, q_\ell)]), & \text{otherwise} \end{cases}$$

where $q_\ell = left\_child(B, select_0(B, k))$

Application to Network Acceleration

- Our challenge: fast data transmission by compressor on FPGA