

逆順の系列集合を表す SeqBDDの構築

立命館大学 M1
青木 洋士

立命館大学
山下 茂

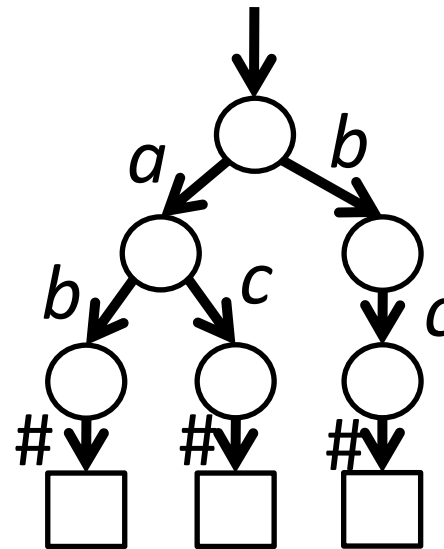
北海道大学
湊 真一

目次

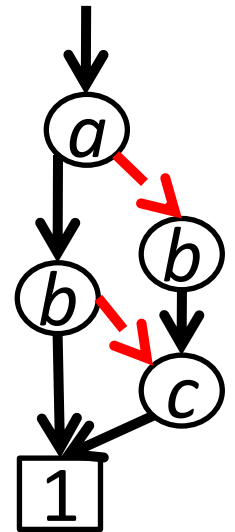
- トライとSeqBDD
- 接頭辞・接尾辞による検索
- 逆順の系列集合の構築
 - 単純手法
 - 提案手法
- 実験
- 応用
 - SeqBDDによる連想配列のキーと値の入替え
 - 1つのワイルドカードによる検索
- まとめ

トライとSeqBDD

- **接頭辞を共有**して系列集合を表現
- SeqBDDは接尾辞も共有
- 接頭辞を指定して検索
 - 検索系列長に比例



トライ

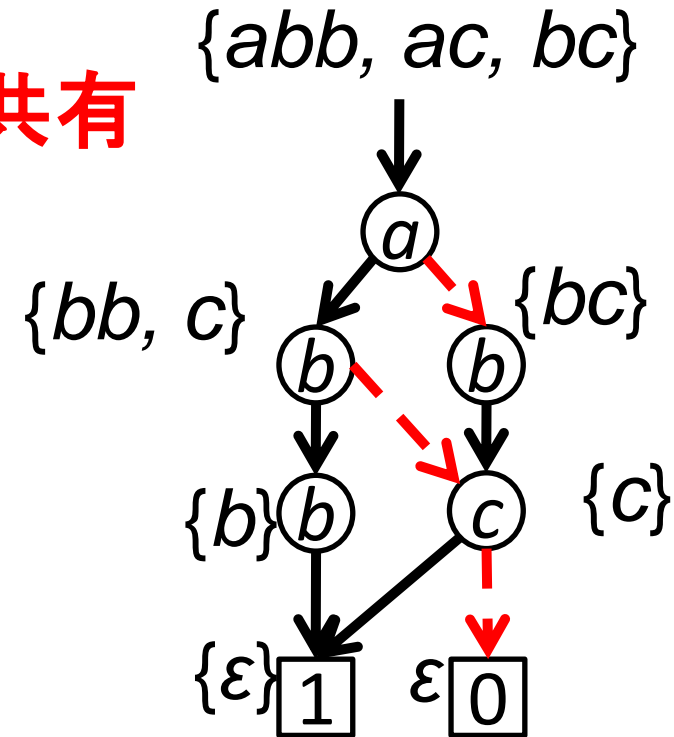


SeqBDD

$\{ab, ac, bc\}$

SeqBDD

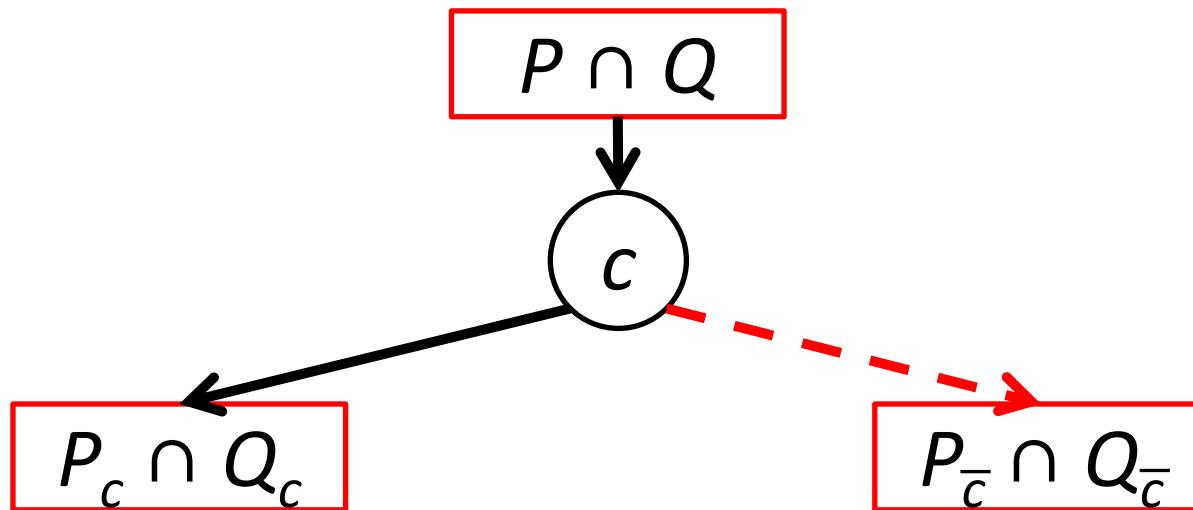
- 系列集合間で**共通な節点を共有**
 - 各節点が系列集合に対応
- 枝のたどり方
 - 1-枝は節点の文字を含む
 - 0-枝は節点の文字を含まない
- 系列集合を**一意に表現**
 - 0-枝側は、辞書順に文字が出現



SeqBDD同士の集合演算

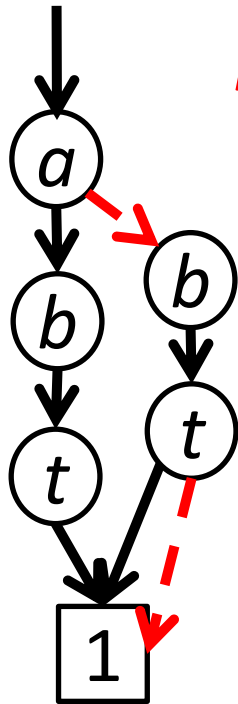
- **先頭が文字 c であるかどうか**で系列を分類
 - 再帰的に展開
 - 演算結果をキャッシュし、再利用

効率の良い演算



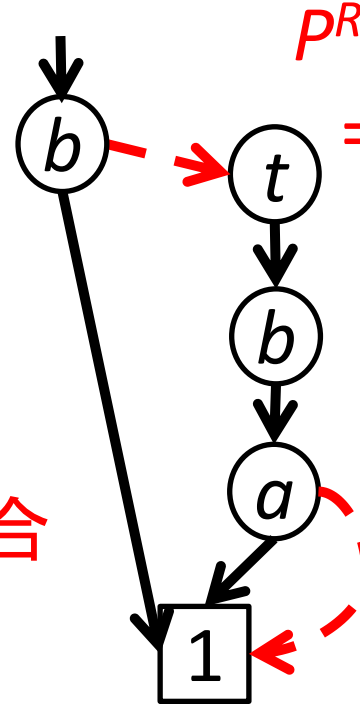
接頭辞・接尾辞による検索

$P = \{abt, bt, b\}$



$P.\text{psearch}(a)$
 $= \{abt\}$

$P^R = \{tba, tb, b\}$

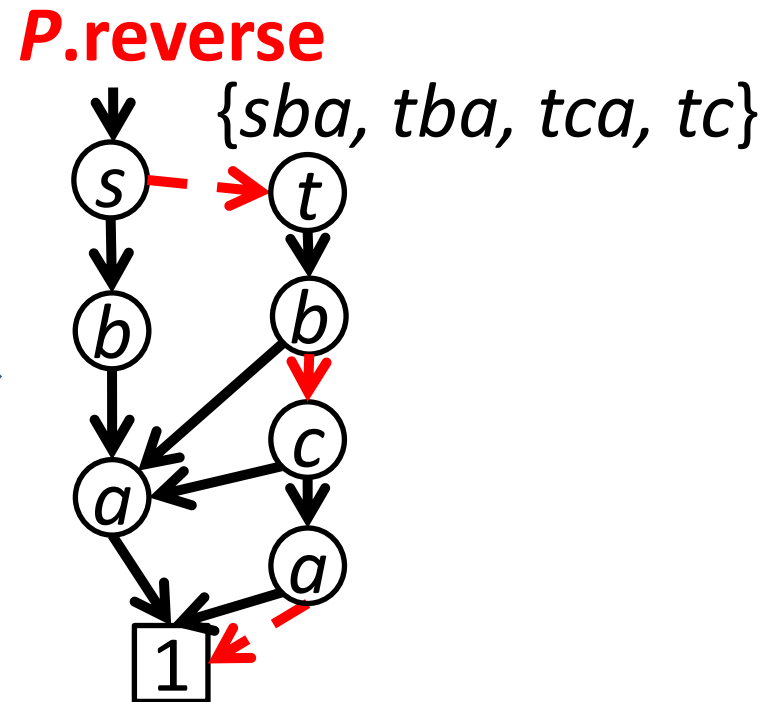
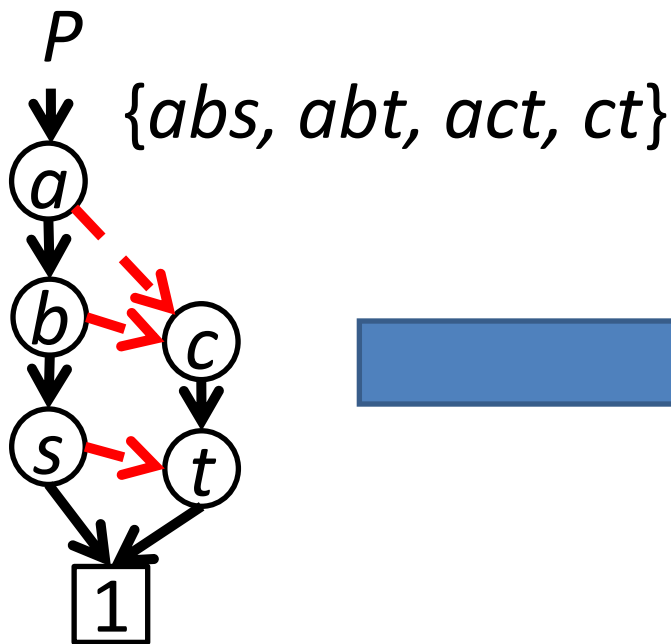


$P^R.\text{psearch}(t)$
 $= \{tba, tb\}$

逆順の系列集合

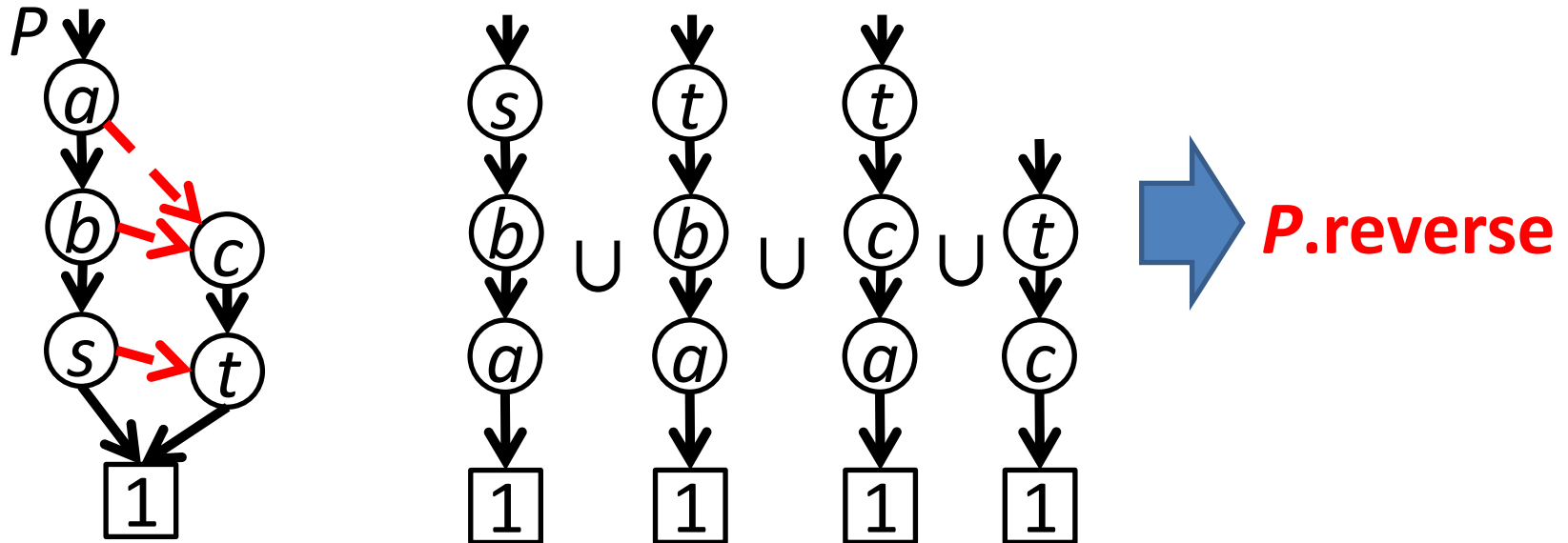
問題定義

- $P.\text{reverse} = \{s \mid s^R \in P\}$
 - s は系列、 s^R は s の逆順の系列



単純な手法

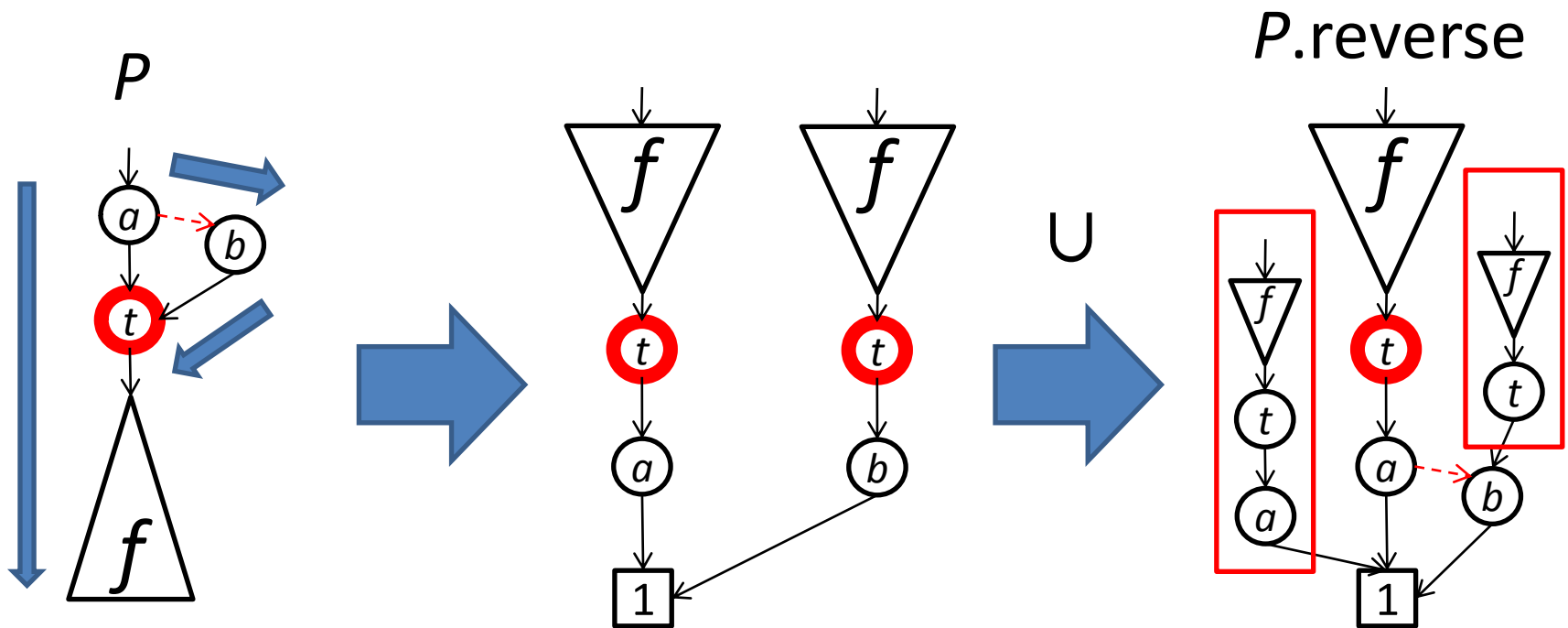
- 再帰的に処理
 - 系列を1つずつ取り出しながら構築
- 共有されている接尾辞を**何度も処理**



$\{abs, abt, act, ct\}$

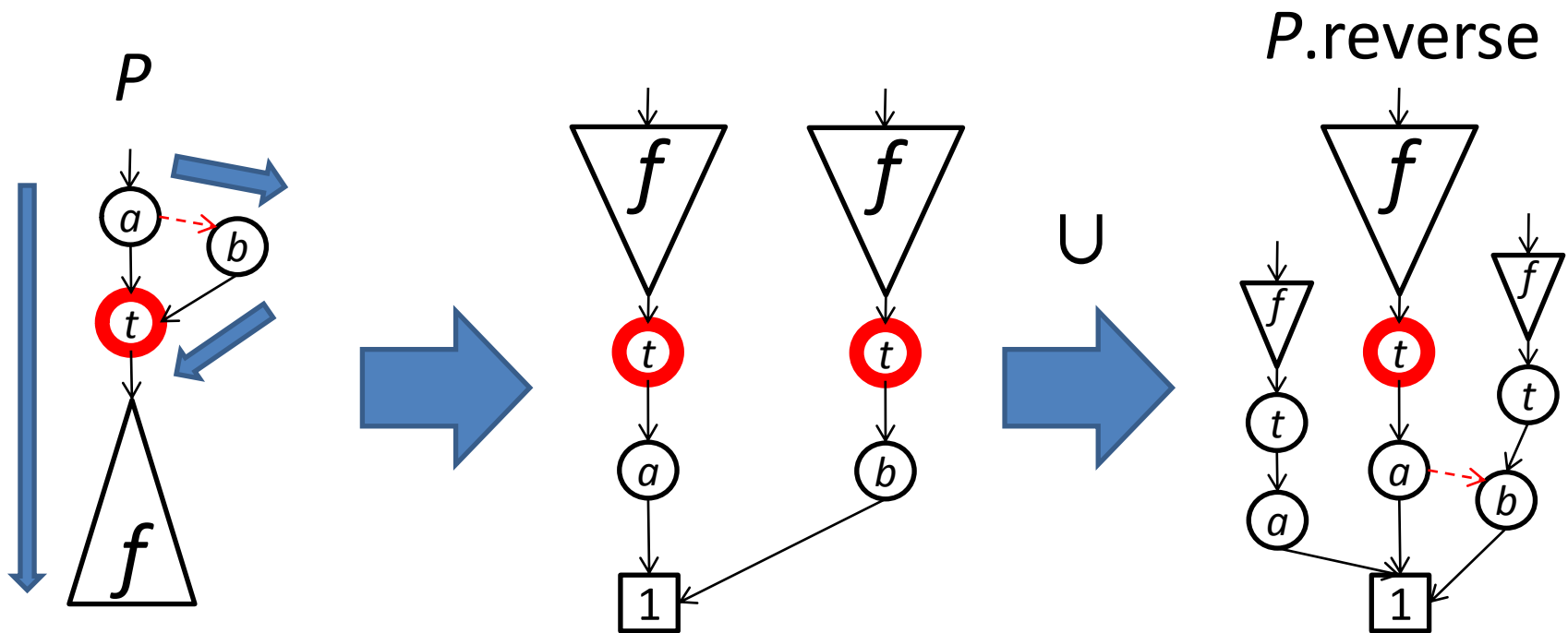
再帰的手法の特徴 (1/2)

- 共有されている接尾辞を何度も通る
- 不要な節点を生成



再帰的手法の特徴 (2/2)

- 元節点と対応する節点が変化
→メモキャッシュの対応付けが困難

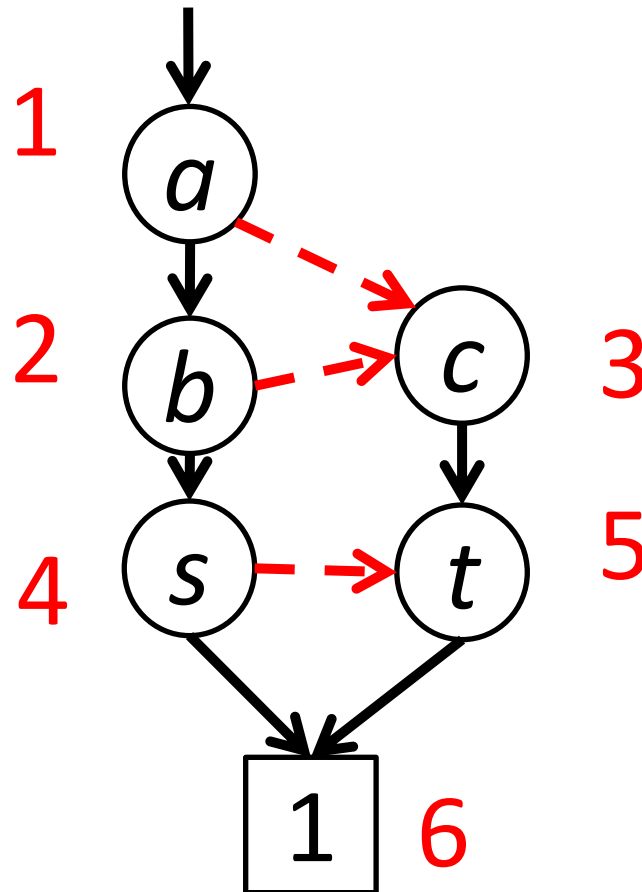


提案手法

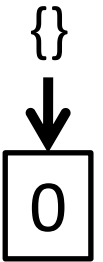
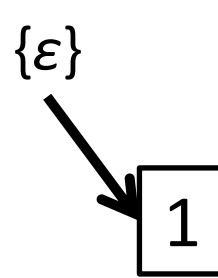
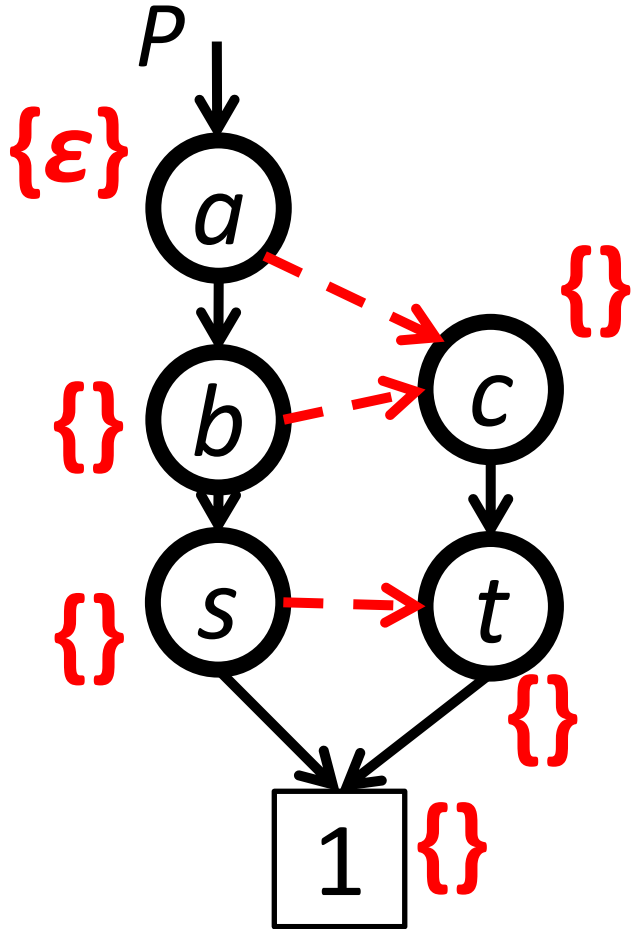
- トポロジカル順序で節点进行处理
 - 節点をトポロジカルソートする
 - 根から1-終端節点まで部分解を遷移
- 共有されている接尾辞を**一度だけ処理**

節点をトポロジカルソート

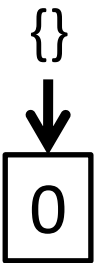
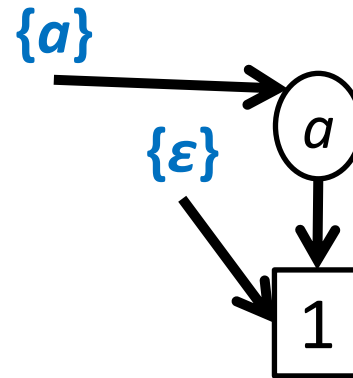
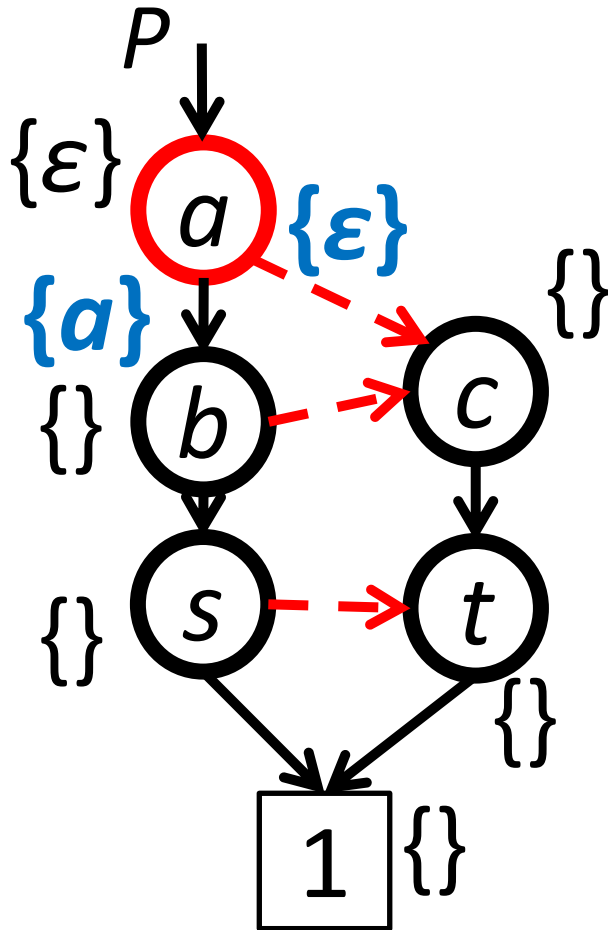
$P = \{abs, abt, act, ct\}$



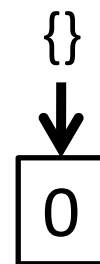
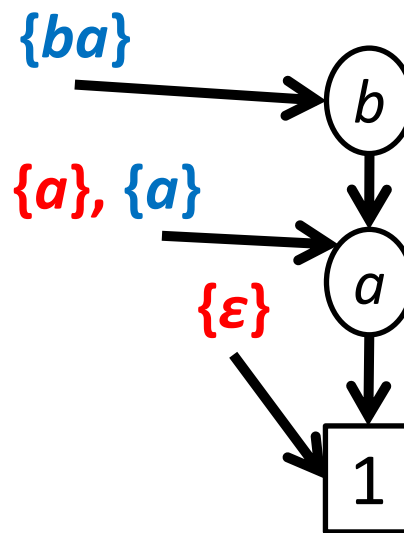
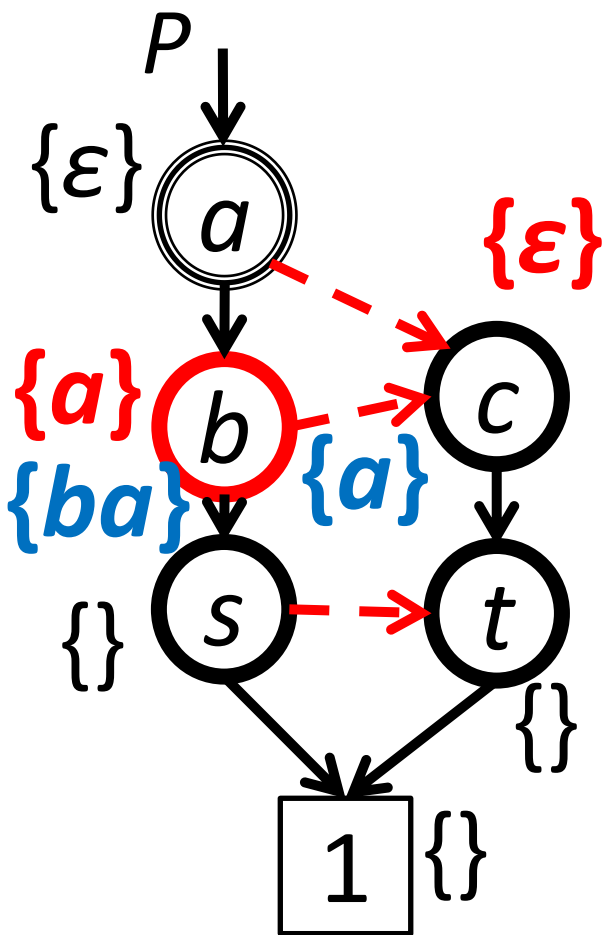
根に $\{\varepsilon\}$ 、他の節点に $\{\}$ を対応付け



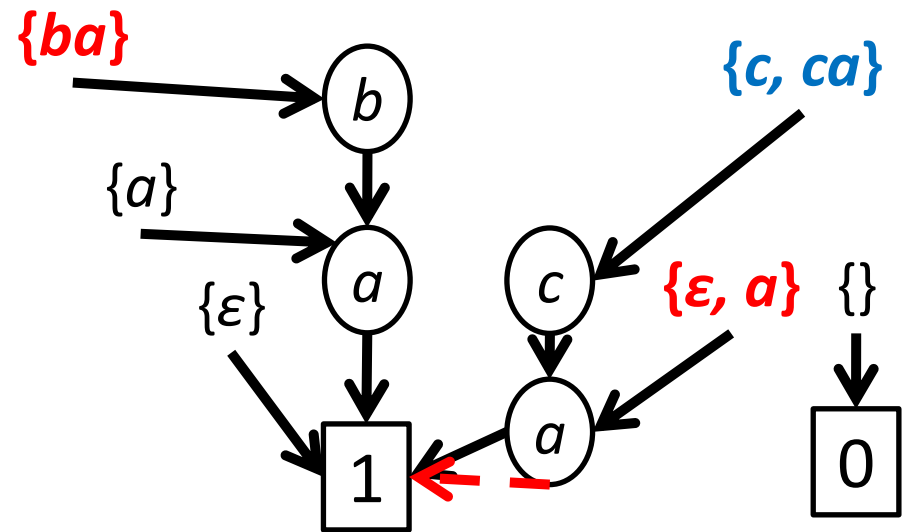
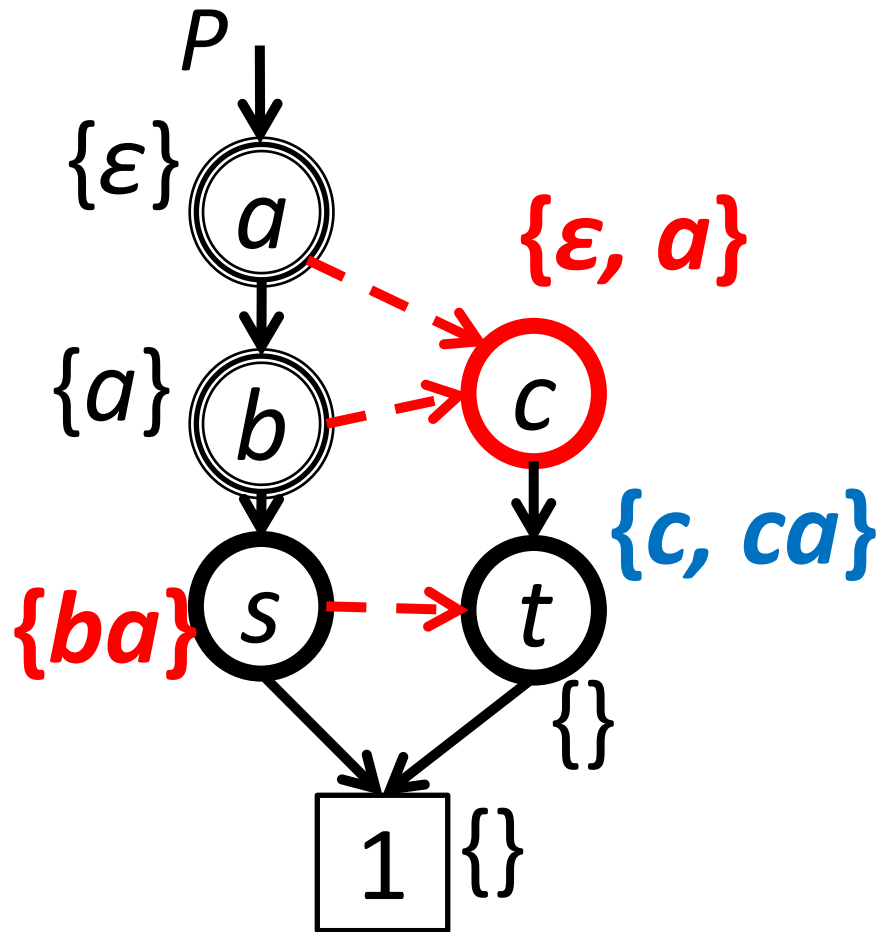
文字 a に対応する節点を展開



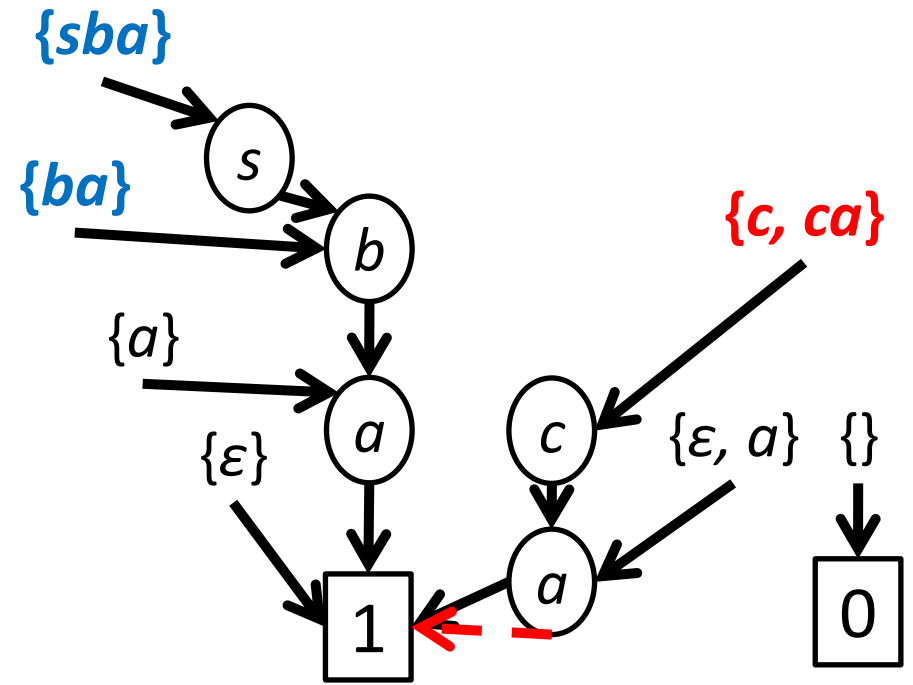
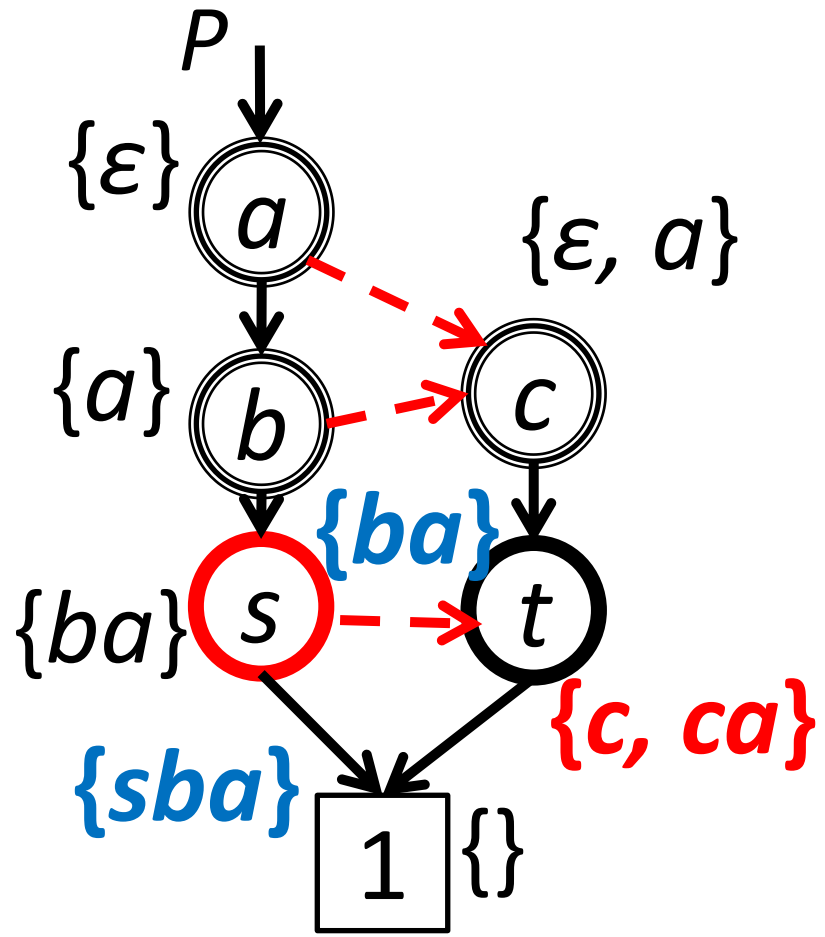
文字 b に対応する節点を展開



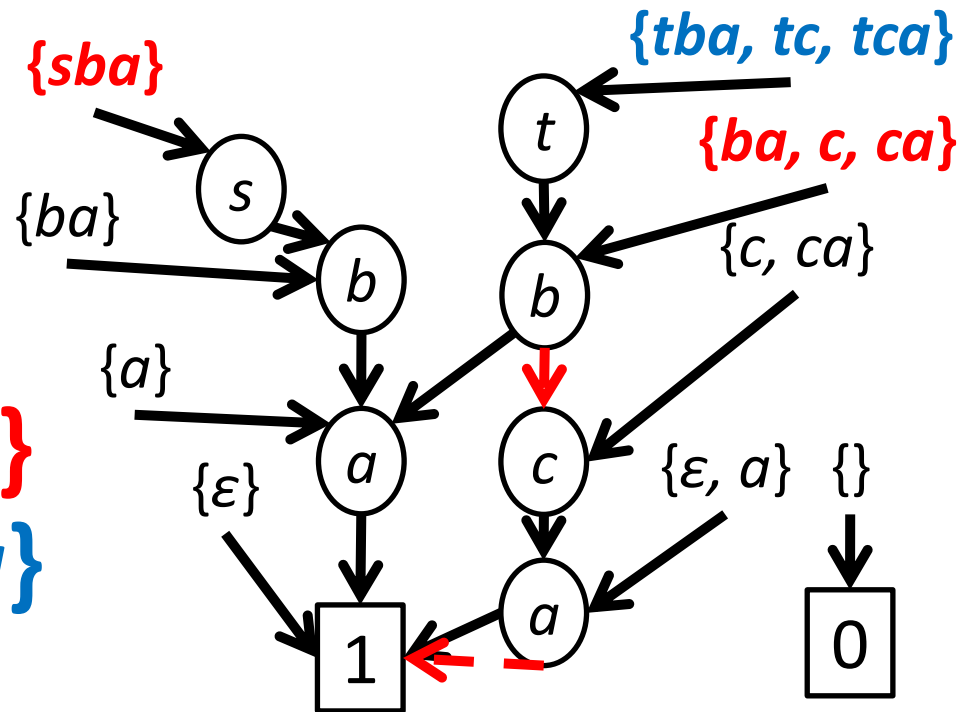
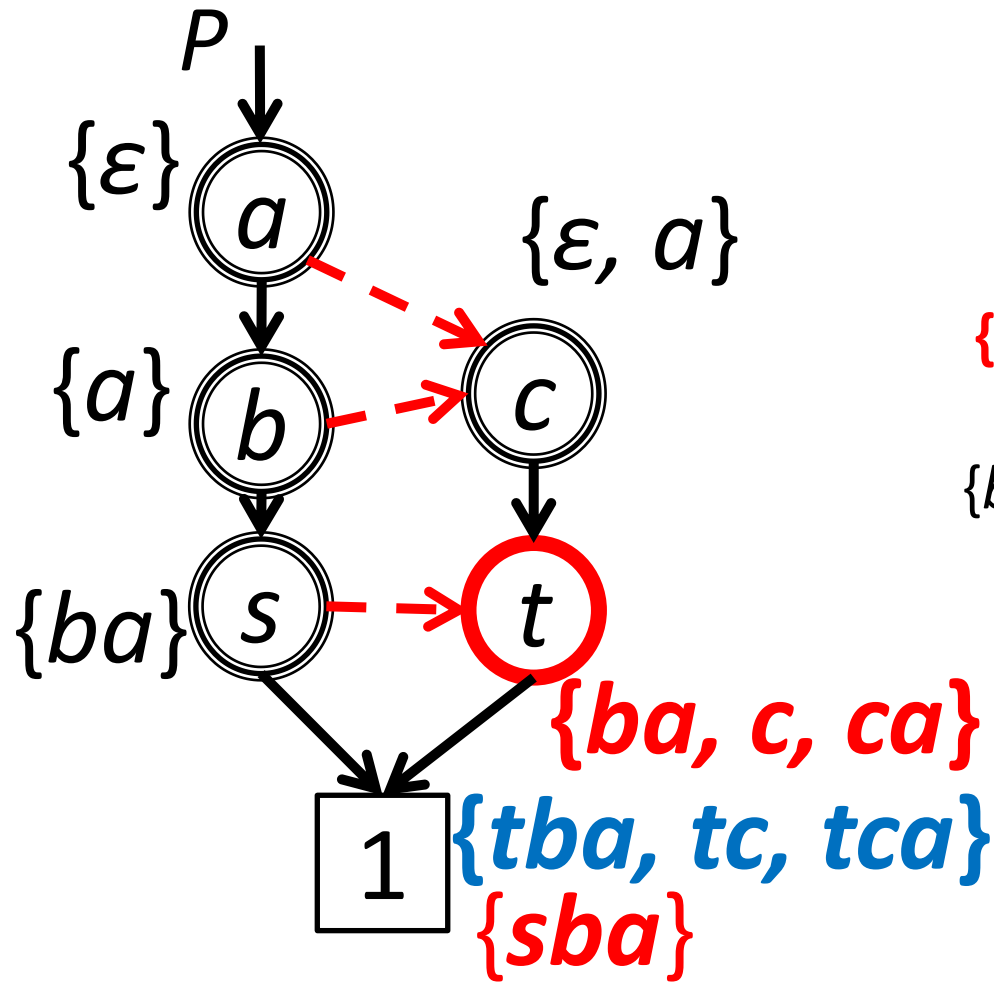
文字cに対応する節点を展開



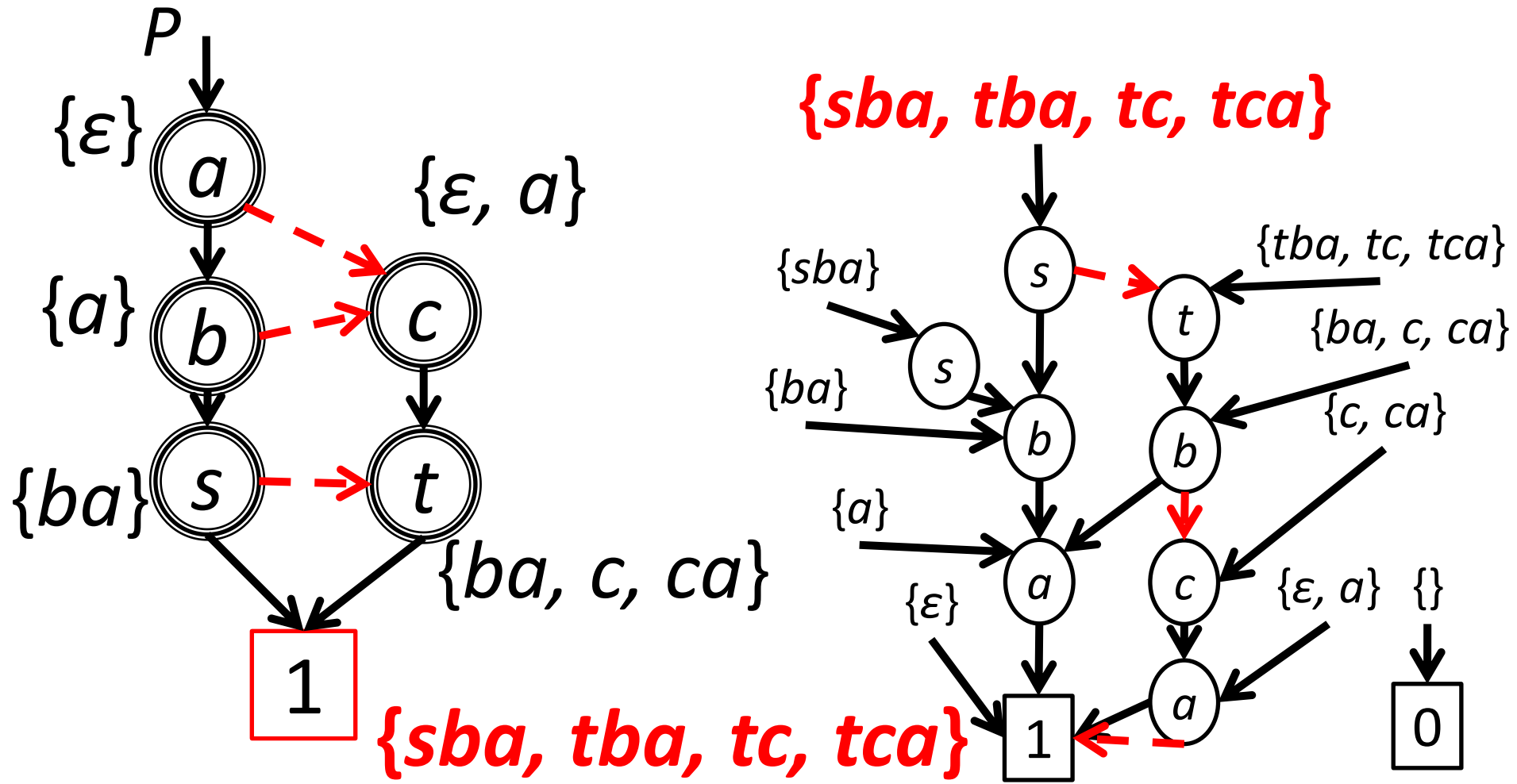
文字sに対応する節点を展開



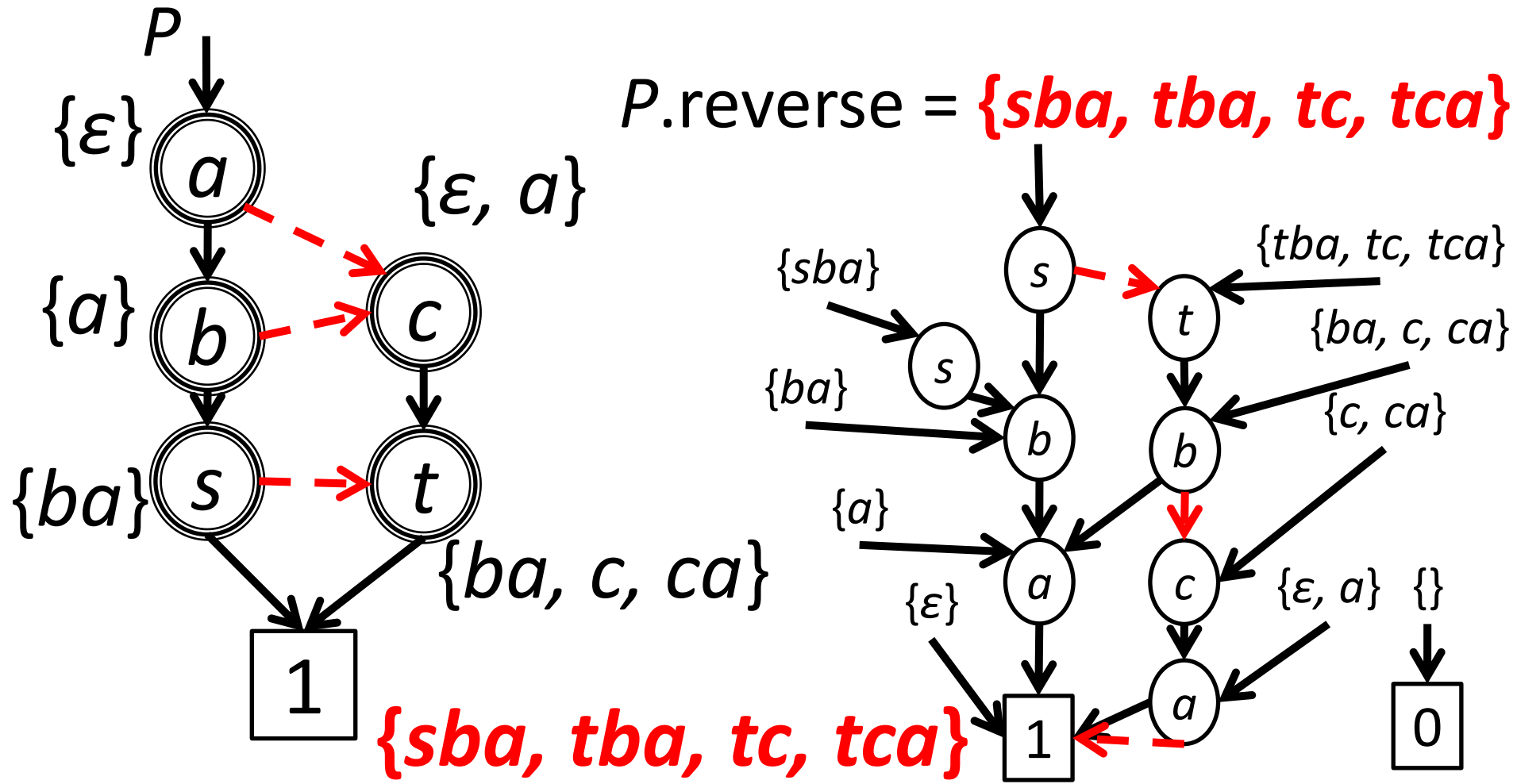
文字 t に対応する節点を展開



全ての節点の展開が完了



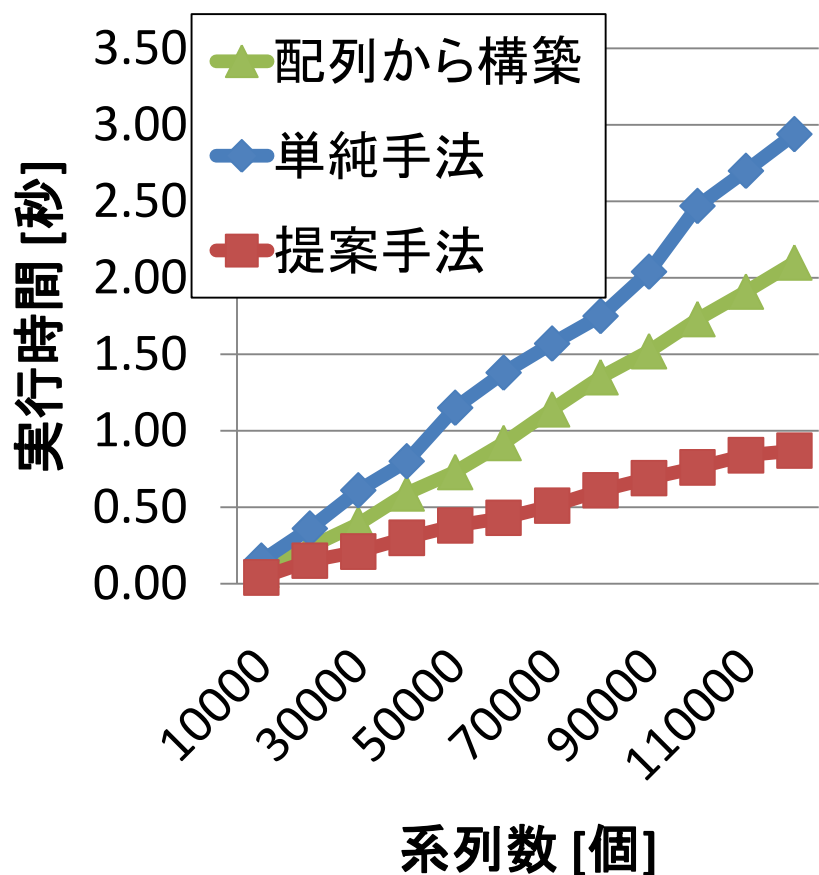
1-終端節点にて逆順の系列集合が対応



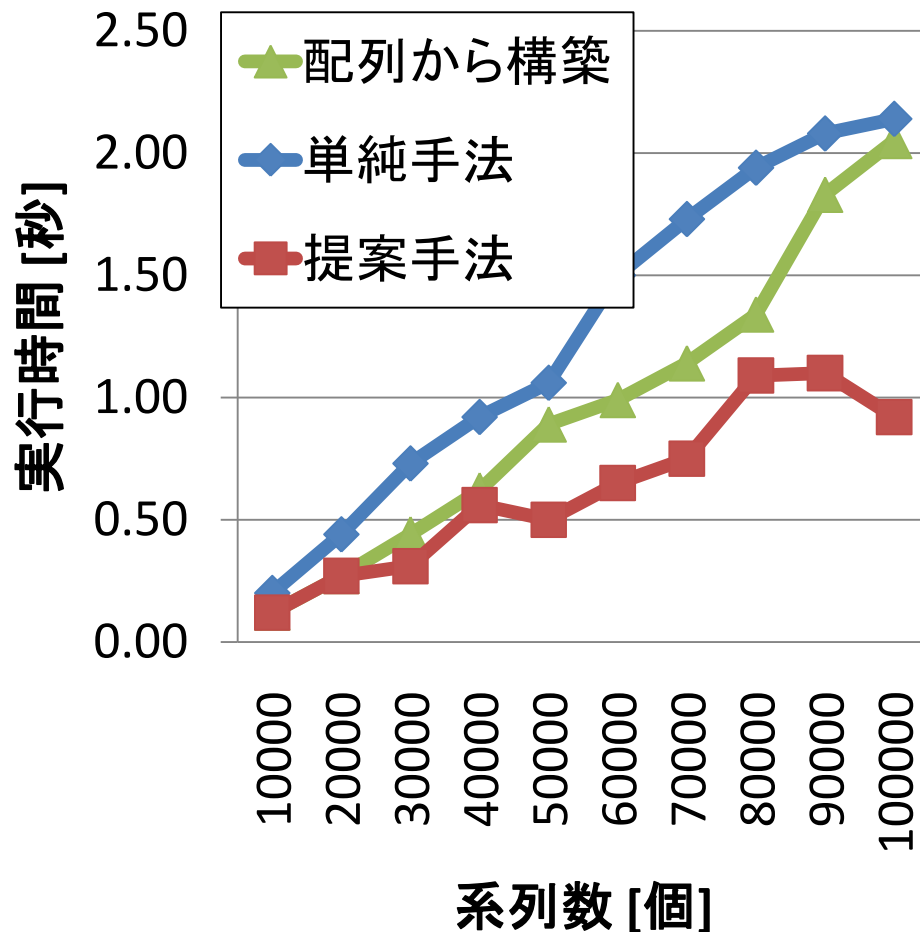
実験

- 3つの手法で実験
 - 配列で与えられた系列集合から**構築**
 - 再帰的なアルゴリズムによる**単純手法**
 - **提案手法**
- データ
 - 日本の住所、ヒトゲノム、ランダムの3種類
 - **系列数を変化**させた
 - ランダムデータは、**系列長も変化**させた

現実のデータでの実験

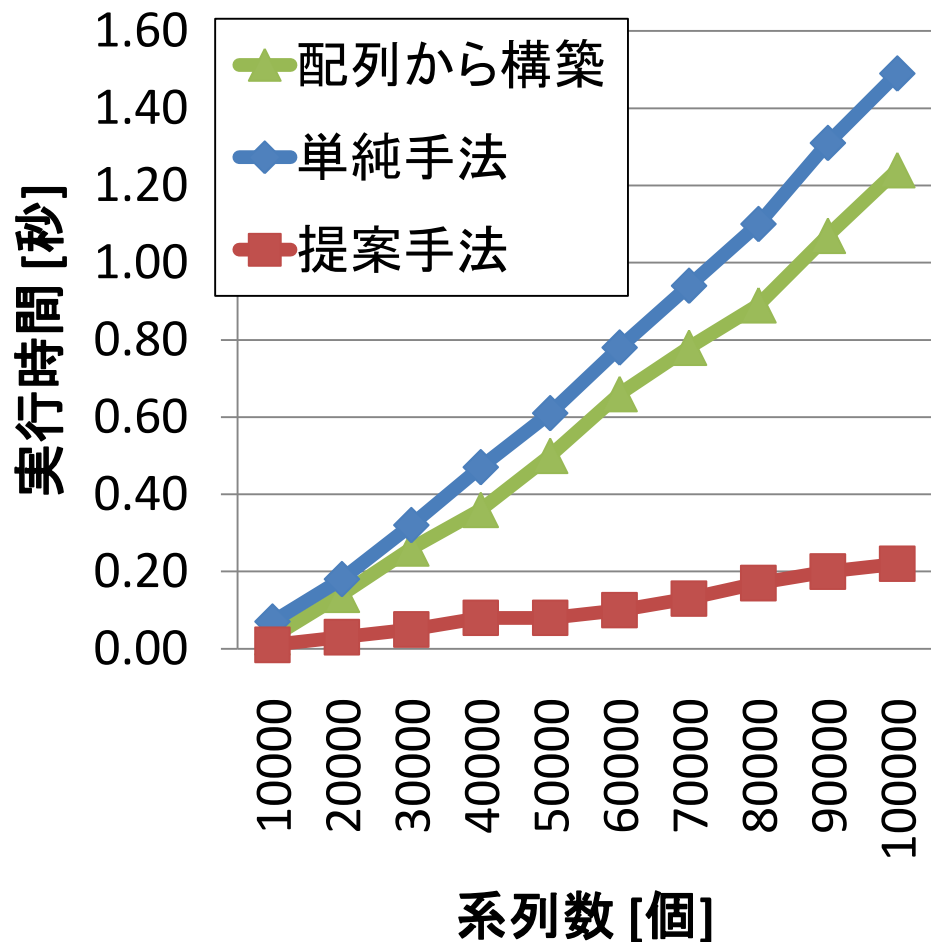


日本の住所

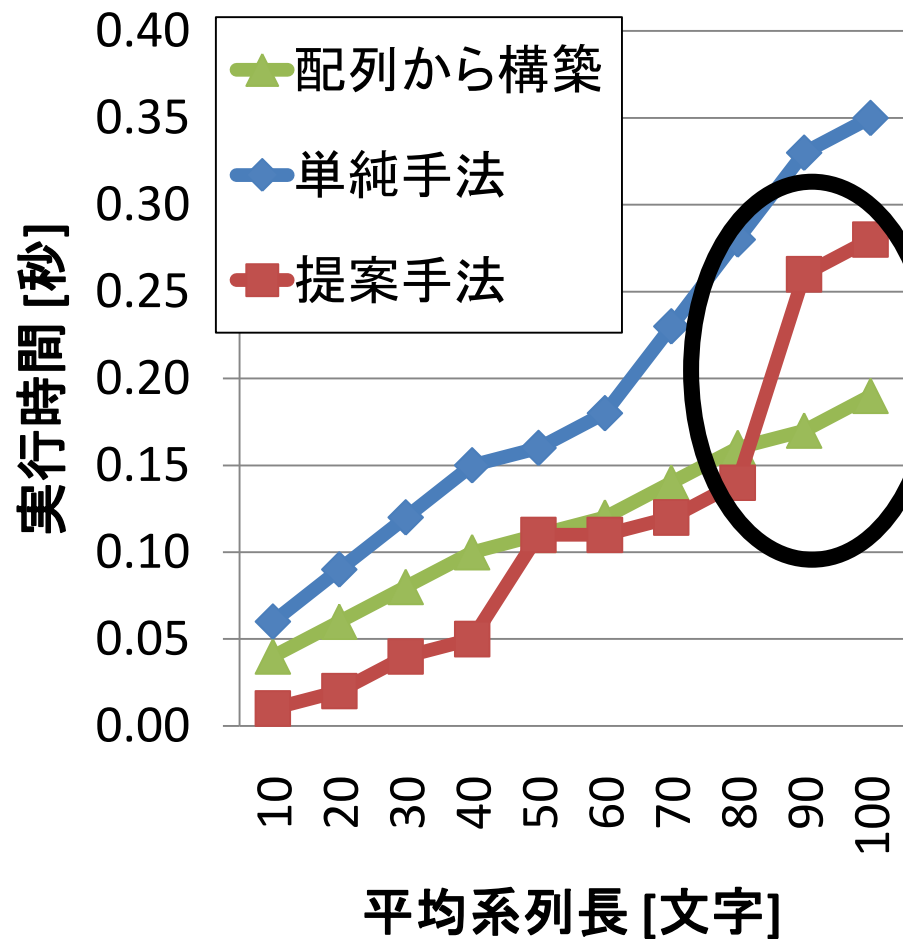


ヒトゲノム

ランダムデータでの実験



系列数を変化



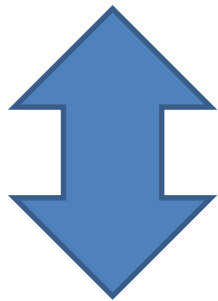
系列長を変化

逆順の系列集合の応用

- SeqBDDによる連想配列の**キーと値の入替え**
- 1つのワイルドカードによる検索
 - 接頭辞と接尾辞の検索結果の積集合

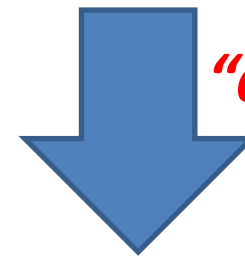
$\{ (ab, yz), (ac, yx) \}$

$\{ abc, ab, bc \}$



**キーと値の
入替え**

$\{ (yz, ab), (yz, ac) \}$



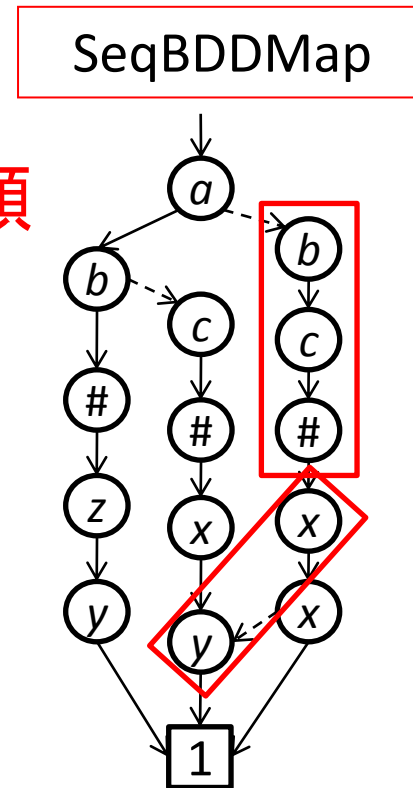
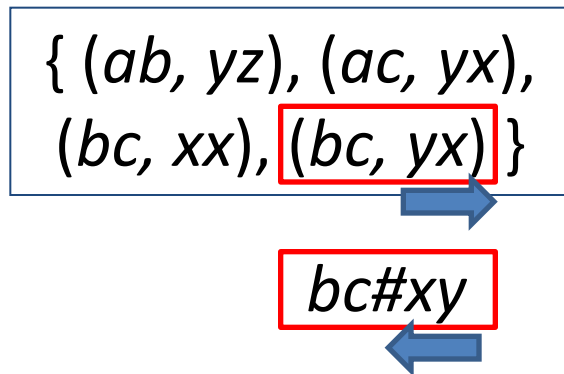
“a*c”で検索

$\{ abc \}$

SeqBDDによる連想配列

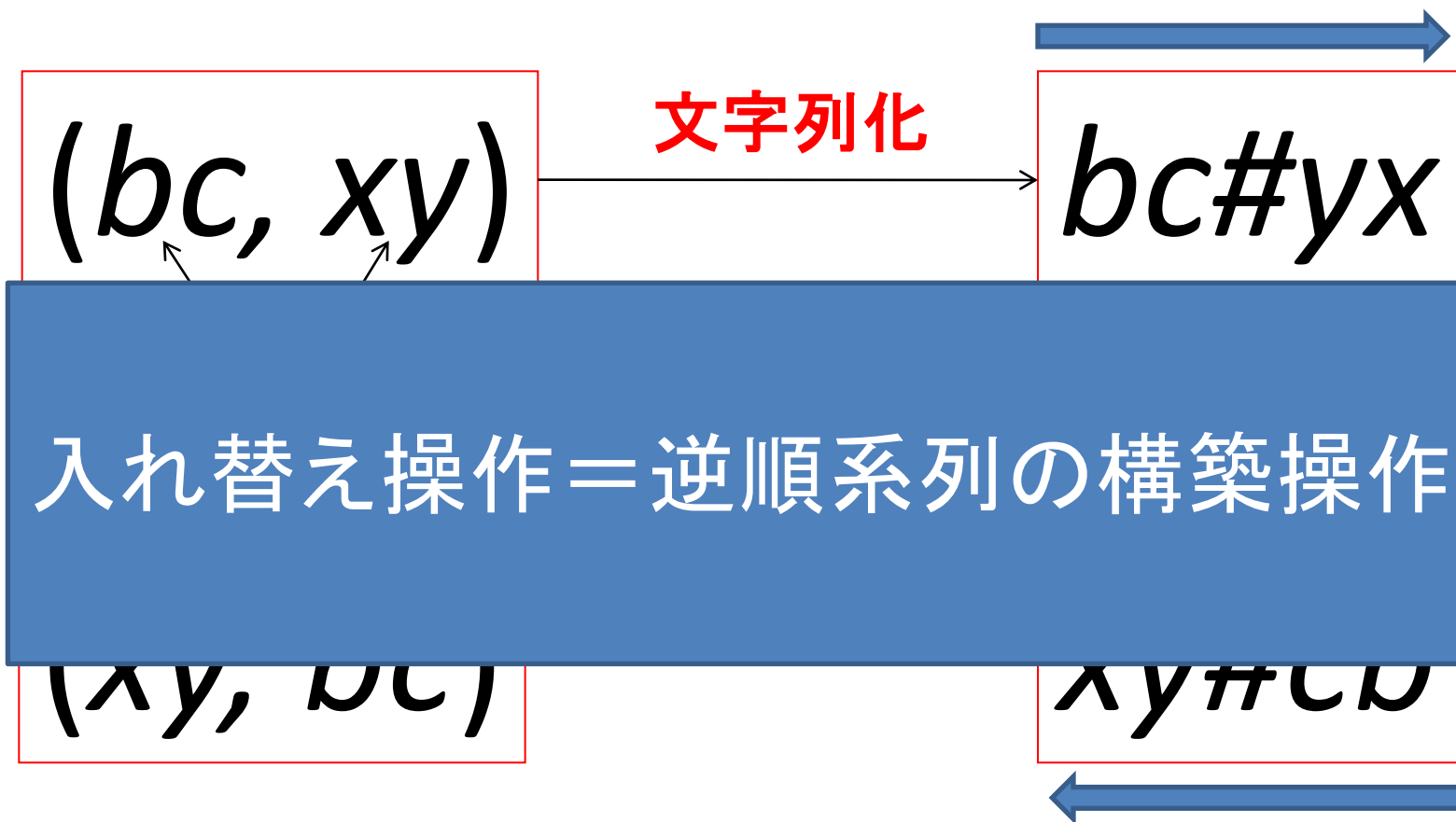
- SeqBDDMap
- キーと値のペア (k, v) を連結
- 値を逆順にする

$(k, v) \Rightarrow k\#v^R$ 文字列 v の逆順



SeqBDDMapのキーと値の入替え

- (bc, xy) のキーと値の入れ替えを考える



1つのワイルドカードによる検索

- Single-Wild-Card Query
 - ワイルドカードを1つだけ許す検索
 - 接頭辞と接尾辞の検索の積集合

$P.swcq(a*t)$

$= P.psearch(a) \cap P^R.psearch(t^R).reverse$

$P = \{abs, abt, act, ct\}$ ならば、

$= \{abs, abt, act\} \cap \{tba, tc, tca\}.reverse$

$= \{abs, abt, act\} \cap \{abt, ct, act\}$

$= \{abt, act\}$

まとめ

- SeqBDDの逆順の系列集合の構築
 - 節点をトポロジカル順序で処理する手法
 - 接尾辞の共有部分を**一度だけ処理**
 - 単純な再帰的手法と比べて効率が良い
- 応用
 - 連想配列のキーと値の入替え
 - 1つのワイルドカードによる検索