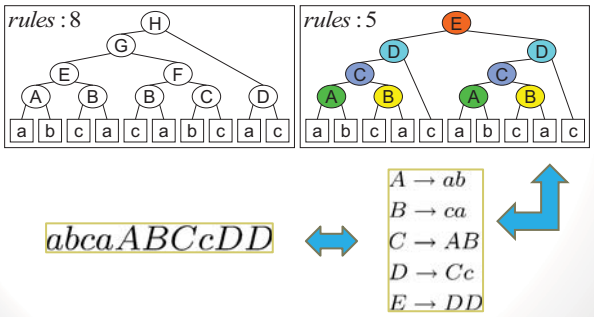
検索可能な文法圧縮の実現

― 文字列間類似度の高速計算 ―

坂本比呂志
九州工業大学大学院情報工学研究院
科学技術振興機構

データ圧縮としてのCFG

- 唯一の文字列を生成する文脈自由文法(CFG)

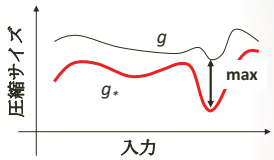


$$abcaABCcDD \Leftrightarrow \begin{cases} A \to ab \\ B \to ca \\ C \to AB \\ D \to Cc \\ E \to DD \end{cases}$$

CFG圧縮の最適化問題

- 圧縮アルゴリズム A の近似率

$$= \max_{w \in \Sigma^*} \left\{ \frac{|g_A(w)|}{|g_*(w)|} \right\}$$



- O(log n)-approximable by:
  - Rytter (2002), *Theor. Comput. Sci.*
  - Sakamoto (2003), *J. Discrete Algorithms.*
  - Charikar, Lehman, et al.(2005), *IEEE Trans.*
  - Sakamoto, Maruyama, et al.(2009), *IEICE Trans.*

文字列間類似度との関連

- 正規圧縮距離(Normalized Compression Distance, *NCD*)

$$NCD = \frac{\max\{C(xy) - C(x), C(yx) - C(y)\}}{\max\{C(x), C(y)\}}$$

二つの文字列xyとxの圧縮サイズで類似度を定義

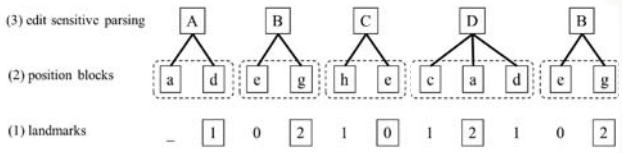
- 移動付き編集距離(approximating *Edit Distance with Move*)

編集距離の変種
- Move a substring in O(1) time
- Delete a substring in O(1) time

EX. $S = a^n b^m c^n d^m e^n$, $S = a^n d^m c^n b^m e^n$

Edit Sensitive Parsing (ESP)

- 極大共通部分文字列による構文木(ESP)



- Alphabet Reduction

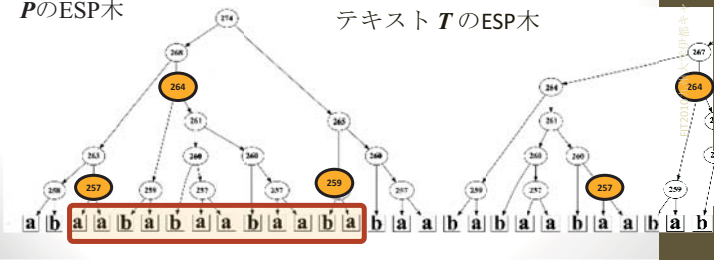
(1)string in binary	a 000	d 011	e 100	g 110	h 111	e 100	c 010	a 000	d 011	e 100	g 110
(2) labels	_	001	000	011	001	000	011	010	001	000	011
(3) labels as integers	_	1	0	3	1	0	3	2	1	0	3
(4) final labels & landmarks	_	1	0	2	1	0	1	2	1	0	2

ESPの特性

パターンの高々log|P|個の構文木を
テキストの構文木に埋め込む問題に帰着

PのESP木

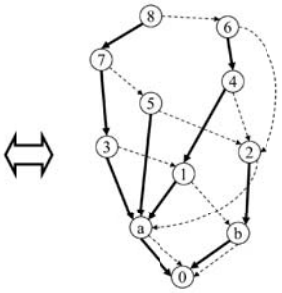
テキスト T のESP木



Naïve CFGとDAG表現

- 互いに等価な生成規則とDAGの関係

$$
\begin{aligned}
X_1 &\rightarrow ab \\
X_2 &\rightarrow ba \\
X_3 &\rightarrow aX_1 \\
X_4 &\rightarrow X_1X_2 \\
X_5 &\rightarrow aX_2 \\
X_6 &\rightarrow X_4X_2 \\
X_7 &\rightarrow X_3X_5 \\
X_8 &\rightarrow X_7X_6
\end{aligned}
$$

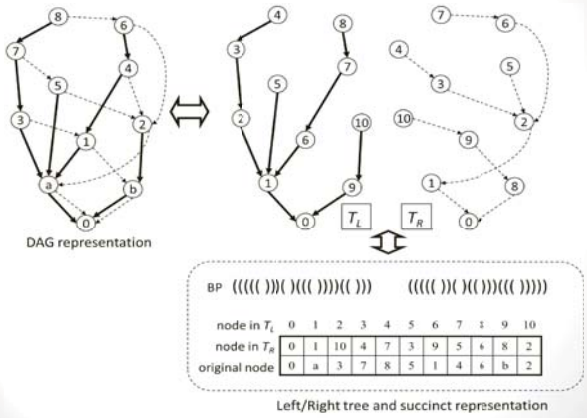


Chomsky normal form CFG G

DAG representation of G with a virtual sink

FIT2010九州大学伊都キャンパス 2010/9/8

DAGの分解と簡潔表現



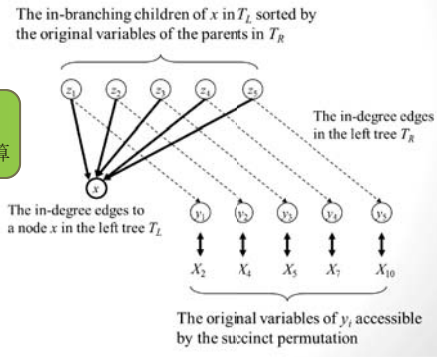
Left/Right tree and succinct representation

FIT2010九州大学伊都キャンパス 2010/9/8

辞書の削減

- 文字の置換で使用する辞書を圧縮済データで模倣

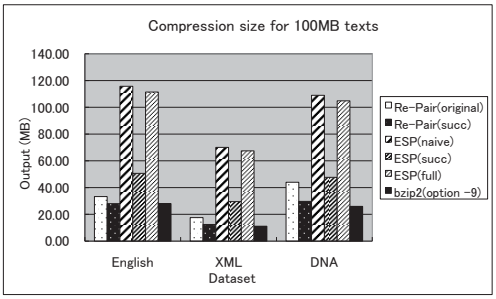
$Z \rightarrow XY$
に対する $f(XY)=Z$ の計算



FIT2010九州大学伊都キャンパス 2010/9/8

実験結果：サイズ

- 圧縮データと索引付き圧縮データの比較



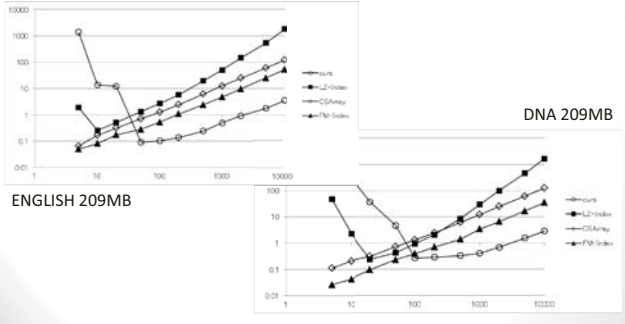
$2(1+\varepsilon)v \log v$ bit-space

$O(\frac{1}{\varepsilon}(m \log v + occ_c(\log m \log n)))$ time to count occurrences

FIT2010九州大学伊都キャンパス 2010/9/8

実験結果：検索時間

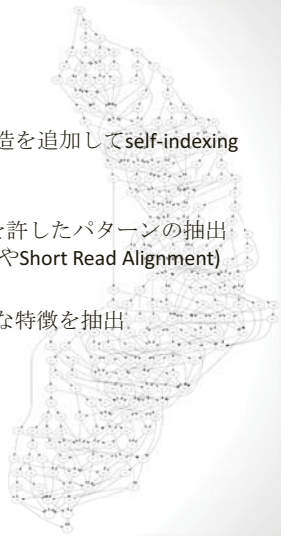
- パターン長に対する出現回数の計算時間(パターンの選択はランダム)



FIT2010九州大学伊都キャンパス 2010/9/8

今後の計画

- 出現位置を返すためのデータ構造を追加してself-indexingとして完成させる
- Exact Matchではなく Miss Matchを許したパターンの抽出へ拡張→(文書間の類似性の発見やShort Read Alignment)
- 巨大な文書集合全体から部分的な特徴を抽出



FIT2010九州大学伊都キャンパス 2010/9/8