

# Ones of the Best are Better than One of the Best

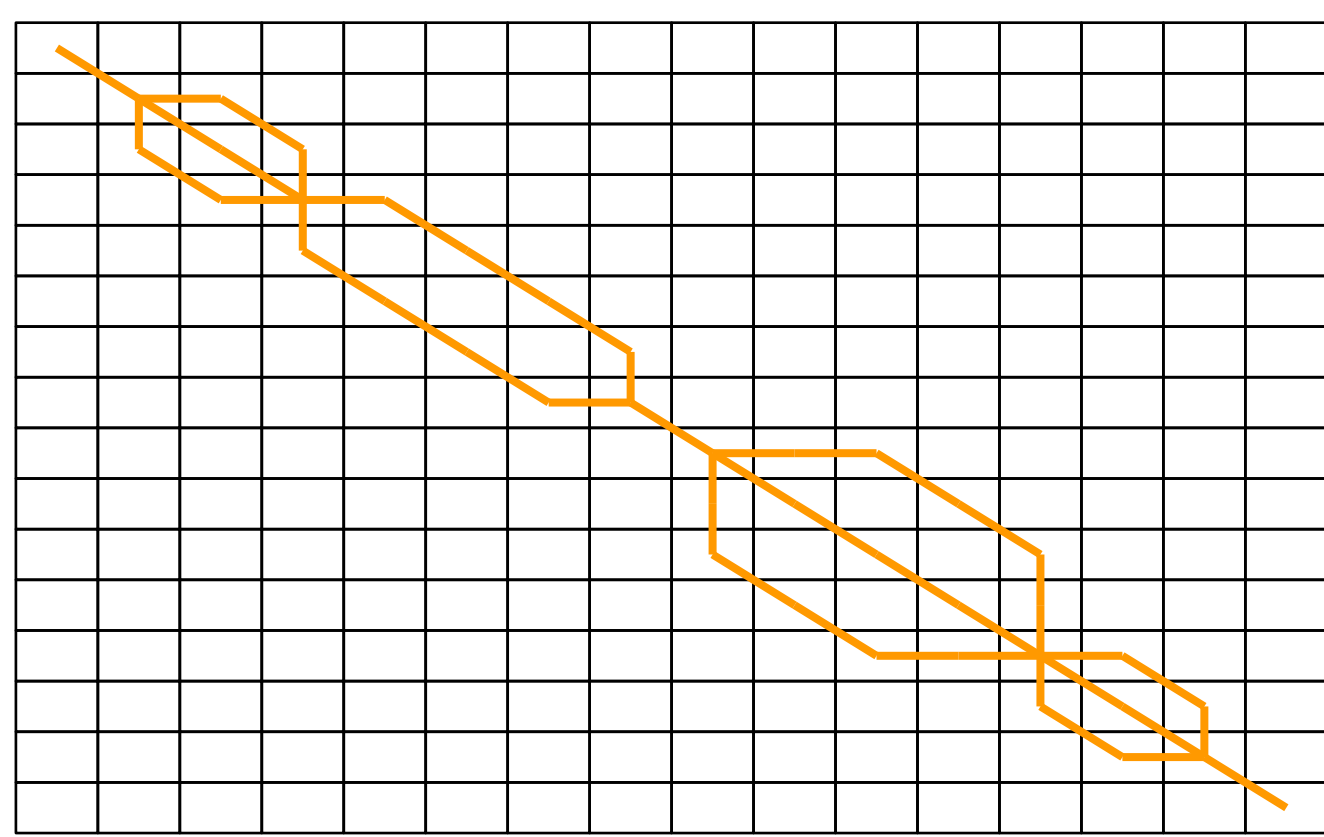
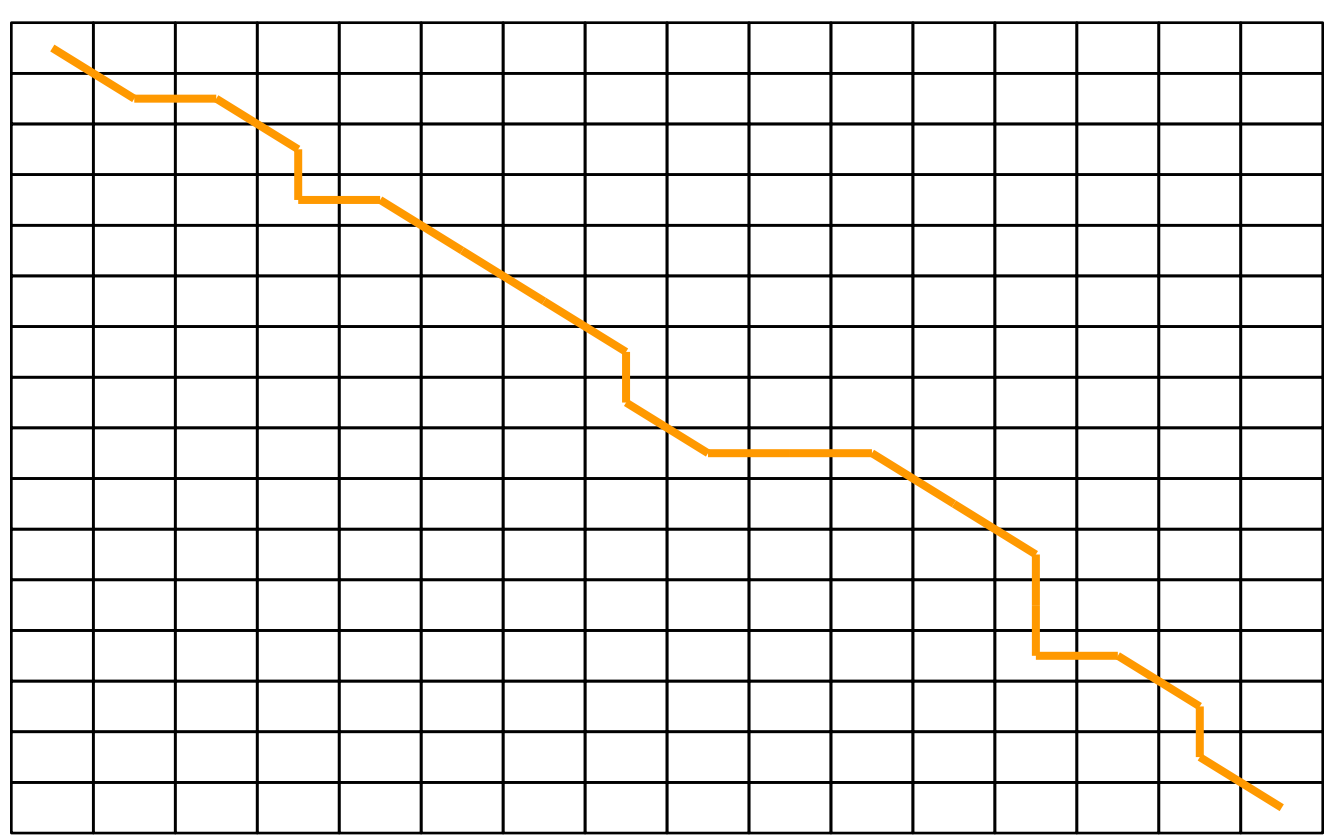
最適なものを実算した後その中から一つだけを取り出し処理を続行することがある。ここでは全ての最適な解をまとめて保持したまま効率よく計算する手法を提案する。

基本的な考え：シーケンスアラインメントを例として

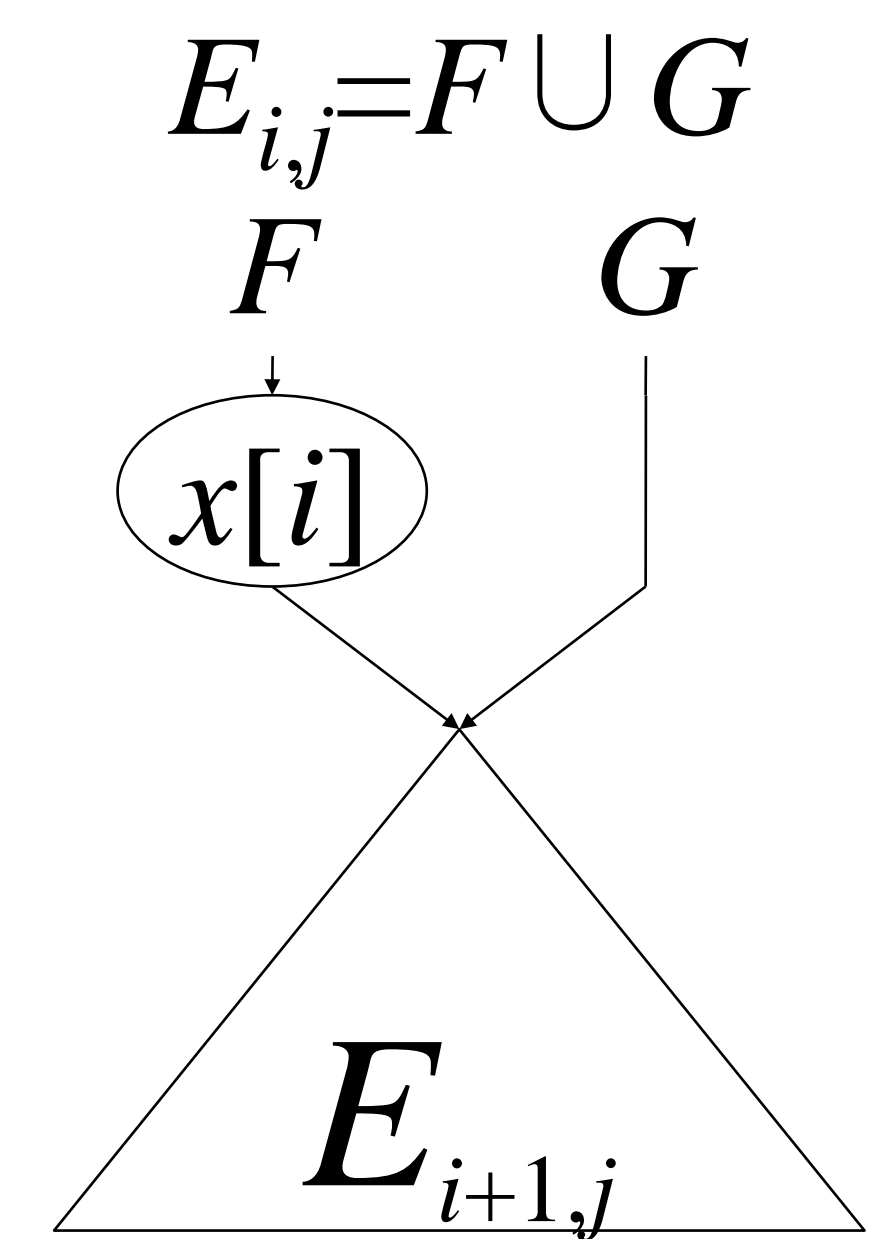
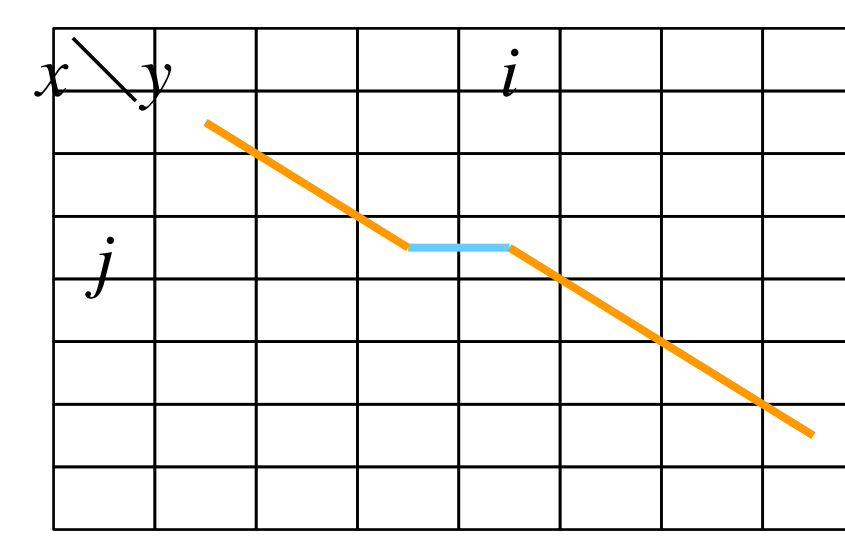
文字列同士のアラインメントを計算する手法としてよく知られているのは動的計画法による方法である。文字列 $x$ と $y$ が与えられた時に、 $|x| \times |y|$ のサイズのテーブルを作成しその上で動的計画法を実行する。編集距離が最小のアラインメントが複数得られるが、一般的にはその内の一つだけを取り出し使用する。一方、本手法では最適なアラインメントを全てSeqBDDの形式でコンパクトに保存し以降の処理もその圧縮された構造上で効率よく行えるようにする。

アラインメントは最小の編集距離を求める時に計算する重みを考慮した際の最短パスに対応する

そういった最短パスは実際には一つだけではなく同時に複数存在することがありDAGのようになる。

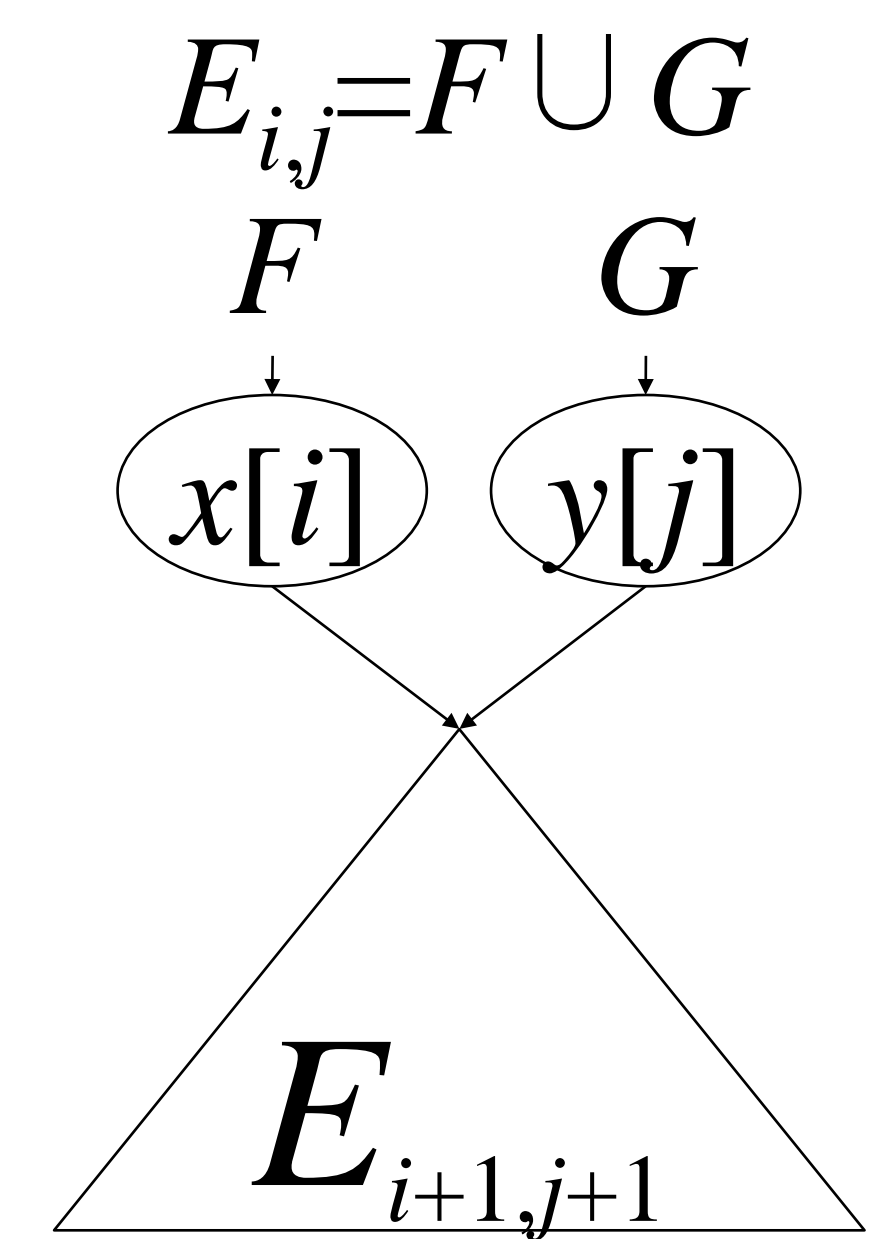
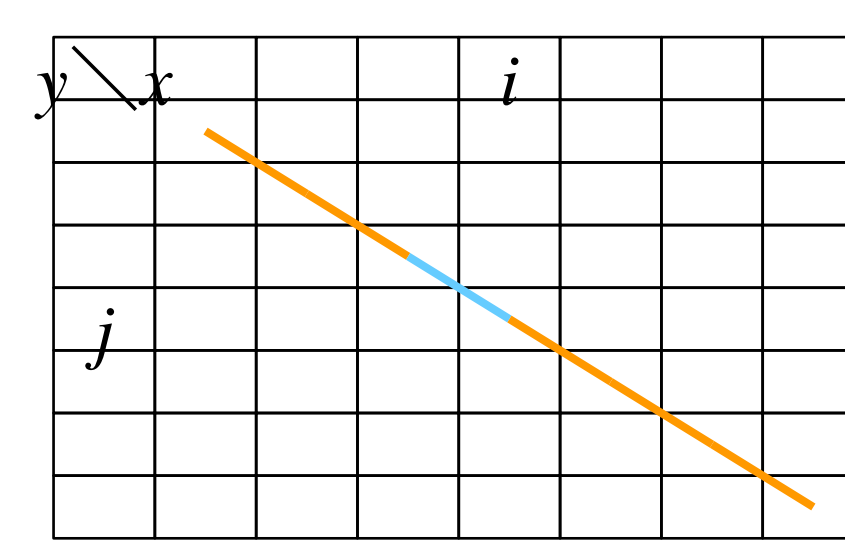


挿入/削除に対応するSeqBDD構築



$x[i]$ を含む  
or  $y[i]$ を  
含まない

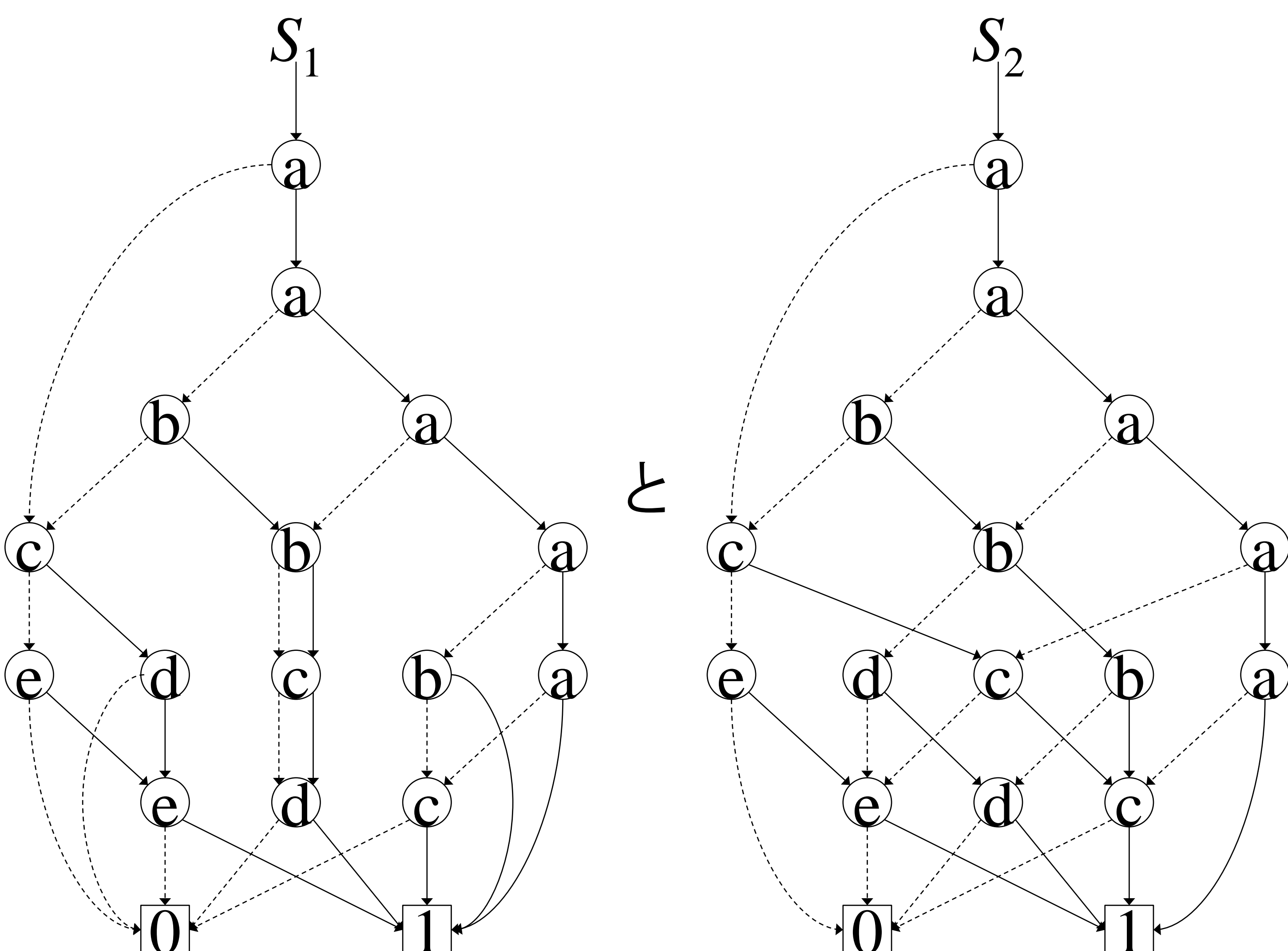
置換に対応するSeqBDD構築



$x[i]$ を含む  
or  
 $y[i]$ を含む

グラフ同士への拡張

昨年の本ワークショップで発表された、SeqBDD上で動的計画法を行い編集距離が最小になる文字列集合を発見するアルゴリズム(西野正彬, SeqBDD上での動的計画法の検討)を適用することで上記の文字列対文字列のアルゴリズムを文字列集合対文字列集合へ拡張することができる。2つのSeqBDDが与えられた時に、その節点数×節点数のサイズのテーブルを作成し動的計画法を実行する。



入力がただの文字列集合の場合

出力されるSeqBDDが含むのは、二つの集合間における編集距離で最も近い文字列同士の全てのアラインメントとなる。各アラインメントはSeqBDD中で元のペアに含まれる文字列との編集距離の和が最小となるような文字列として表される。

入力がアラインメントの場合

出力されるSeqBDDが含むのは、入に含まれる既に計算された最適アラインメント同士の更に最適アラインメントとなる。このとき気をつけるべきなのは3本以上の文字列に対する真に最適なアラインメントにはなっていないということである。

まとめ

SeqBDDを用いて全最適アラインメントに対応する文字列を格納する手法を提案した。同様のテクニックは遺伝子配列アセンブリや文字列圧縮といった複数の可能性を同時に試すことで性能向上が見込まれるアプリケーションに適用可能と考えている。複数の解をSeqBDD形式で格納した後どのような処理が可能かも今後の課題である。