DanceDD-C を用いた XCC に対するアルゴリズム

松本 吏司　原田 崇司

高知工科大学 情報学群

2025 年 1 月 14 日

研究背景

- 色付き厳密被覆問題は幅広い問題を比較的簡単に扱える
- 現実の問題を定式化すると，インスタンスが巨大化
- 厳密被覆問題に対してはインスタンスを圧縮し，その表現上で探索を行う高速なアルゴリズム ($\mathrm{D}^3\mathrm{X}$) が存在

色付き厳密被覆問題に対してはこのような手法が存在しない

$\mathrm{D}^3\mathrm{X}$ を拡張

研究背景

色付き厳密被覆問題で扱える問題の例

- スケジューリング
- ワードサーチ
- グラフ彩色
- s–t パス列挙

厳密被覆問題

厳密被覆問題 (Exact Cover, XC)

入力：集合 U と U の冪集合の部分集合 $\mathcal{F}$

出力：U の分割となる $\mathcal{F}$ の部分集合

集合 U の要素をアイテム，集合 $\mathcal{F}$ の要素をオプションと呼称
各アイテムを丁度一度被覆するオプションの組合せを求める問題

XC の例

$$U = \{a, b, c, d\},$$
$$\mathcal{F} = \{\{a\}, \{a, c, d\}, \{b, c\}, \{b, d\}, \{d\}\}.$$

$\{a\}, \{b, c\}, \{d\}$ の組合せが解
$\{a, c, d\}, \{b, c\}$ の組合せは c が重複するため解でない

色付き厳密被覆問題

色付き厳密被覆問題 (Exact Covering with Colors, XCC)

アイテム集合 U をプライマリアイテムの集合 U_p と，セカンダリアイテムの集合 U_s に分割

セカンダリアイテムには，新たな属性として色を付与

XCC は XC を拡張した問題であり，Knuth によって提案

各アイテムの制約

プライマリアイテム：丁度一度被覆しなければならない

セカンダリアイテム：高々一度被覆してよい

ただし，同じ色がつけられている場合は重複して被覆してよい

XCC の例

セカンダリアイテム s に色 c が付いていることを $s\!:\!c$ と表現

$$
\begin{aligned}
U_p &= \{a, b, c\},\ U_s = \{x, y\}, \\
\mathcal{F} &= \{\{a, x\!:\!A\}, \{a, b, x\!:\!B, y\}, \{a, c, x, y\!:\!A\}, \{b, x\!:\!A, y\!:\!B\}, \\
&\qquad \{b, c, x\!:\!B, y\!:\!A\}, \{c, x\!:\!B\}, \{c, y\}\}.
\end{aligned}
$$

$\{a, b, x\!:\!B, y\}, \{c, x\!:\!B\}$ の組合せが解

$\{a, x\!:\!A\}, \{b, c, x\!:\!B, y\!:\!A\}$ の組合せは x に付けられた色が異なるため解でない

Algorithm X

XC を解くバックトラックアルゴリズム

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}
\xrightarrow{1\text{を選択}}
\begin{array}{c} \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c} \begin{array}{ccc} b & c & d \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}
\xrightarrow{3\text{を選択}}
\begin{array}{c} \\ 5 \end{array}
\begin{array}{c} d \\ \begin{pmatrix} 1 \end{pmatrix} \end{array}
$$

Algorithm X

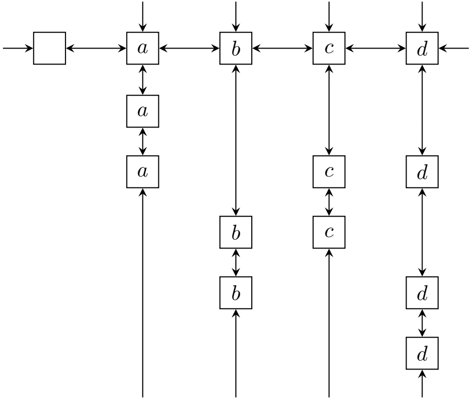
$$
\begin{array}{c|cccc}
 & a & b & c & d \\
\hline
1 & 1 & 0 & 0 & 0 \\
2 & 1 & 0 & 1 & 1 \\
3 & 0 & 1 & 1 & 0 \\
4 & 0 & 1 & 0 & 1 \\
5 & 0 & 0 & 0 & 1
\end{array}
\xrightarrow{1\text{を選択}}
\begin{array}{c|ccc}
 & b & c & d \\
\hline
3 & 1 & 1 & 0 \\
4 & 1 & 0 & 1 \\
5 & 0 & 0 & 1
\end{array}
\xrightarrow{4\text{を選択}}
\begin{array}{c|c}
 & c \\
\hline
- & -
\end{array}
$$

$$
\xrightarrow{2\text{を選択}}
\begin{array}{c|c}
 & b \\
\hline
- & -
\end{array}
$$

Dancing Links

双方向リンクを持つノードによって構成されるデータ構造
ノードの削除・復元を，リンクの付替えによって高速に処理
疎なインスタンスに対して省メモリ

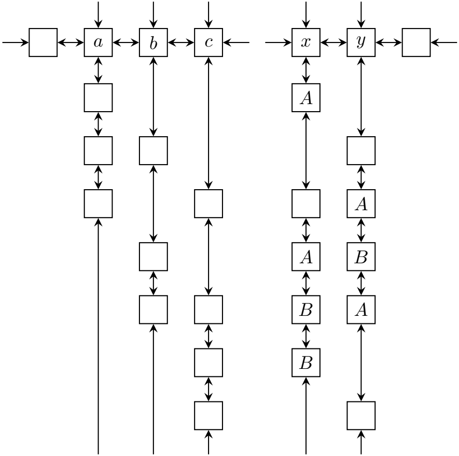
$$
\begin{array}{c c}
 & \begin{array}{cccc} a & b & c & d \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{pmatrix}
1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$



Algorithm C (XCC への拡張)

同じ色のセカンダリアイテムを持つオプションは削除しない
Dancing Links は，問題の拡張に合わせてヘッダ部を分割

$$\begin{array}{c c}
 & \begin{matrix} a & b & c & x & y \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} &
\begin{pmatrix}
1 & 0 & 0 & A & 0 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & A \\
0 & 1 & 0 & A & B \\
0 & 1 & 1 & B & A \\
0 & 0 & 1 & B & 0 \\
0 & 0 & 1 & 0 & 1
\end{pmatrix}
\end{array}$$



Algorithm C

$$
\begin{array}{c|c}
 & \begin{matrix} a & b & c & x & y \end{matrix} \\
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} &
\begin{pmatrix}
1 & 0 & 0 & A & 0 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & A \\
0 & 1 & 0 & A & B \\
0 & 1 & 1 & B & A \\
0 & 0 & 1 & B & 0 \\
0 & 0 & 1 & 0 & 1
\end{pmatrix}
\end{array}
\xrightarrow{1\text{を選択}}
\begin{array}{c|c}
 & \begin{matrix} b & c & y \end{matrix} \\
\begin{matrix} 4 \\ 7 \end{matrix} &
\begin{pmatrix}
1 & 0 & B \\
0 & 1 & 1
\end{pmatrix}
\end{array}
\xrightarrow{4\text{を選択}}
\begin{array}{c|c}
 & c \\
- & \begin{pmatrix} - \end{pmatrix}
\end{array}
$$

Algorithm C

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} a & b & c & x & y \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\begin{pmatrix}
1 & 0 & 0 & A & 0 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & A \\
0 & 1 & 0 & A & B \\
0 & 1 & 1 & B & A \\
0 & 0 & 1 & B & 0 \\
0 & 0 & 1 & 0 & 1
\end{pmatrix}
\end{array}
\xrightarrow{2\text{を選択}}
\begin{array}{cc}
 & \begin{array}{cc} c & x \end{array} \\
6 & \begin{pmatrix} 1 & B \end{pmatrix}
\end{array}
$$

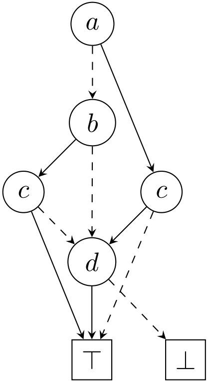
$$
\xrightarrow{3\text{を選択}}
\begin{array}{cc}
 & b \\
- & \begin{pmatrix} - \end{pmatrix}
\end{array}
$$

インスタンスの圧縮

ゼロサプレス型二分決定グラフ
(Zero-suppressed binary decision diagram, ZDD)

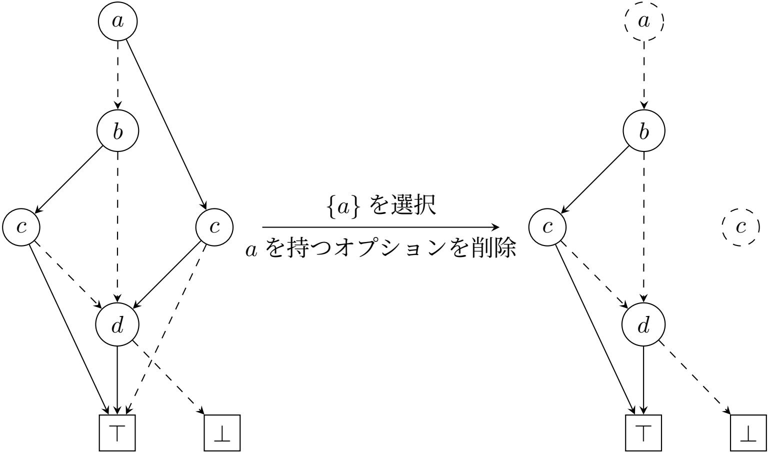
組合せ集合を簡潔に表現できるデータ構造
0 枝，1 枝の子を持つ非終端節点と，0-終端節点，1-終端節点で構成される

$$U = \{a, b, c, d\},$$
$$\mathcal{F} = \{\{a\}, \{a, c, d\}, \{b, c\}, \{b, d\}, \{d\}\}$$

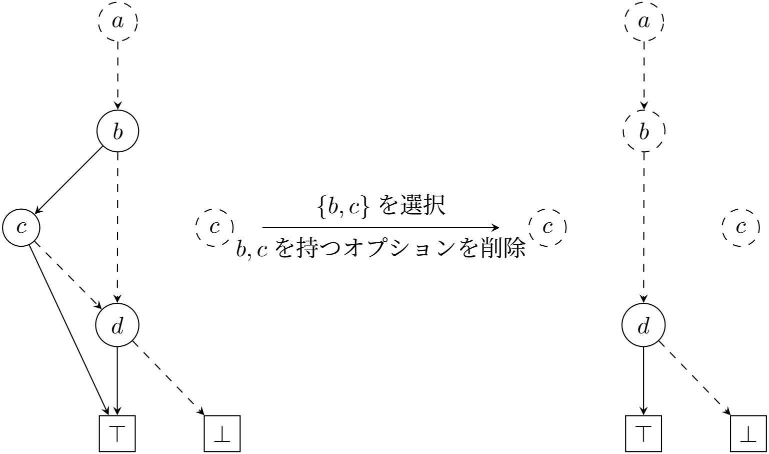


アルゴリズム D^{3}X Nishino et al.

D^{3}X は ZDD 上で探索を行うバックトラックアルゴリズム

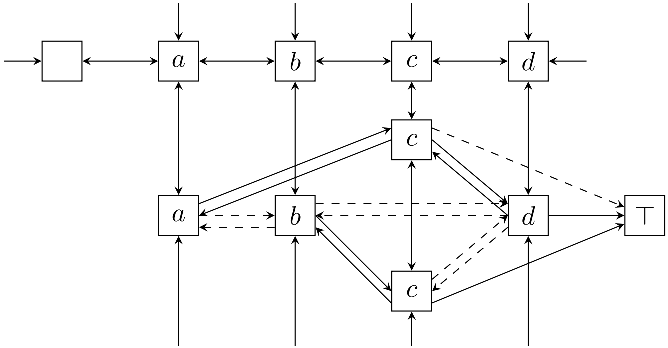


アルゴリズム D^{3}X Nishino et al.

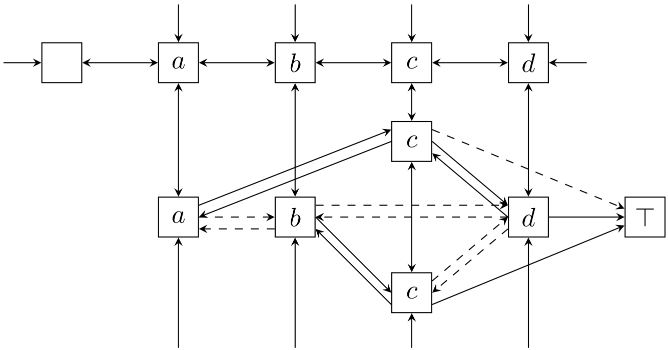


DanceDD Nishino et al.

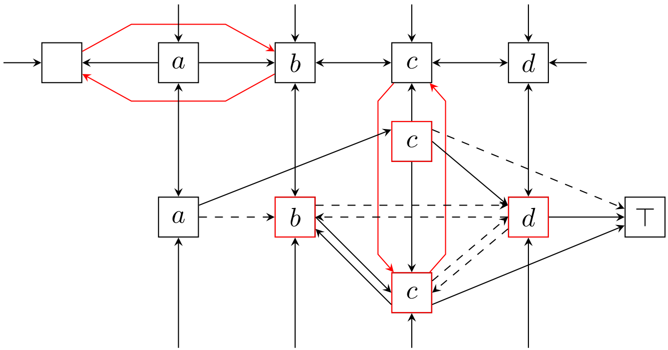
Dancing Links のオプションに対応する部分を ZDD で表現したデータ構造
巨大な入力に対しても効率よく処理



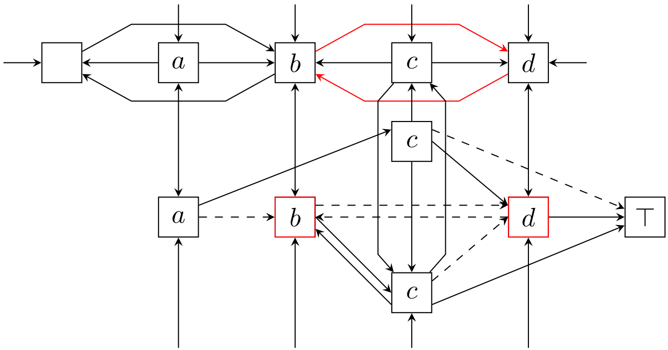
D^{3}X の適用例



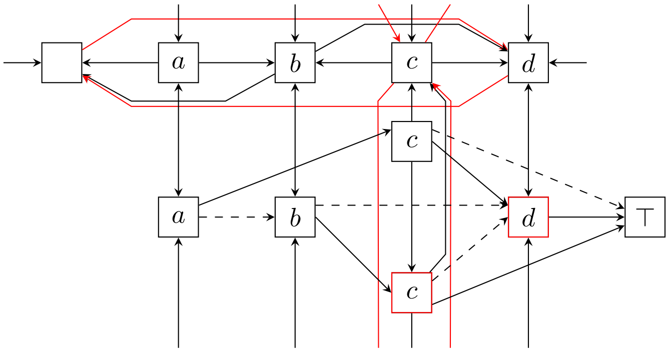
a を被覆



a, c を被覆



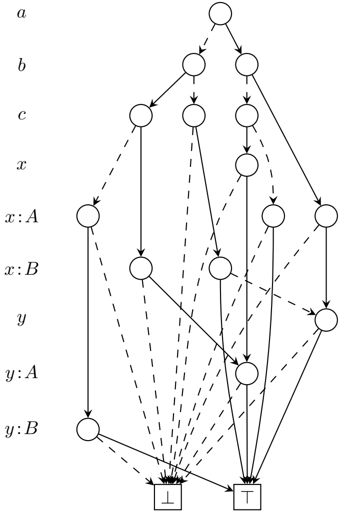
a, c, b を被覆



D^{3}X の XCC への拡張

色ごとに異なる変数を用意し，ZDD で表現

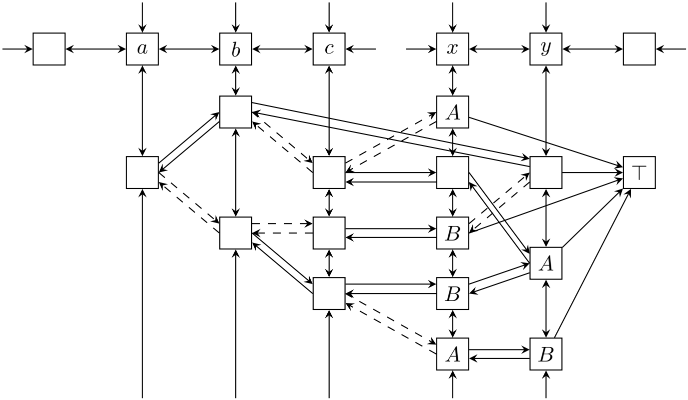
$$
\begin{aligned}
U_p &= \{a, b, c\},\ U_s = \{x, y\},\\
\mathcal{F} &= \{\{a, x{:}A\}, \{a, b, x{:}B, y\}, \{a, c, x, y{:}A\},\\
&\quad \{b, x{:}A, y{:}B\}, \{b, c, x{:}B, y{:}A\},\\
&\quad \{c, x{:}B\}, \{c, y\}\}
\end{aligned}
$$



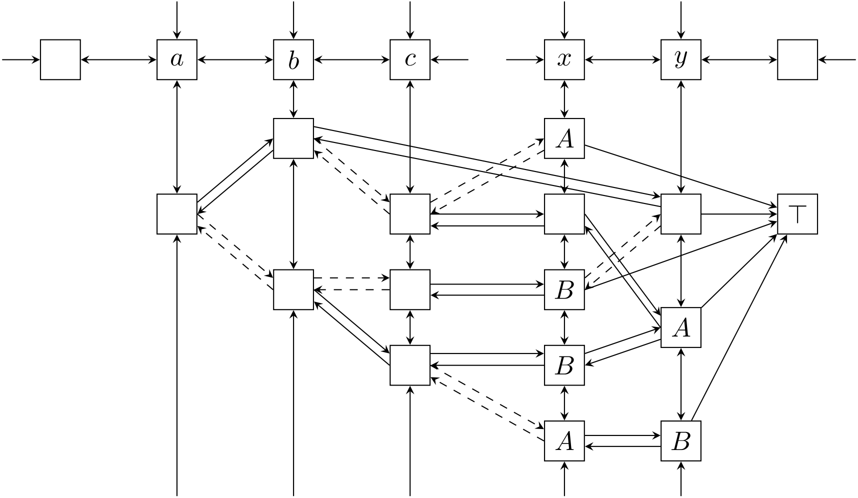
アルゴリズム D^{3}C

DanceDD-C

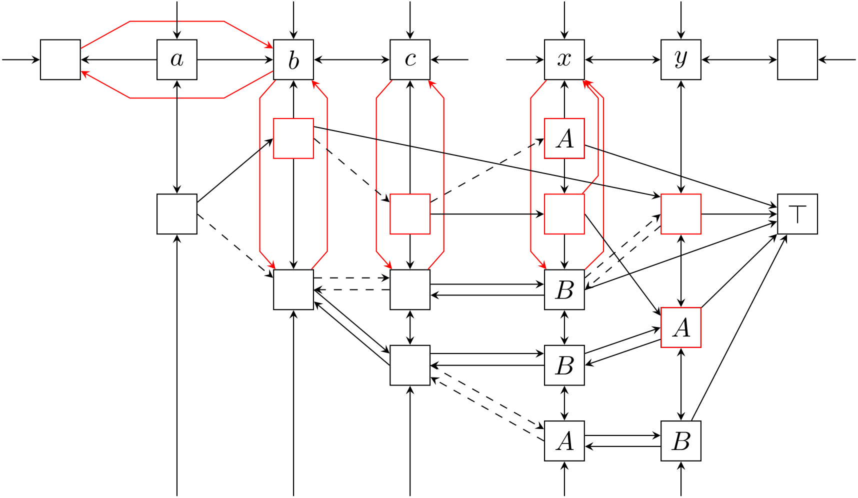
DanceDD を XCC に対応できるように拡張
Algorithm C の拡張と同様に，ヘッダ部を分割
複数の変数に別れたセカンダリアイテムを，同一のアイテムにまとめて表現



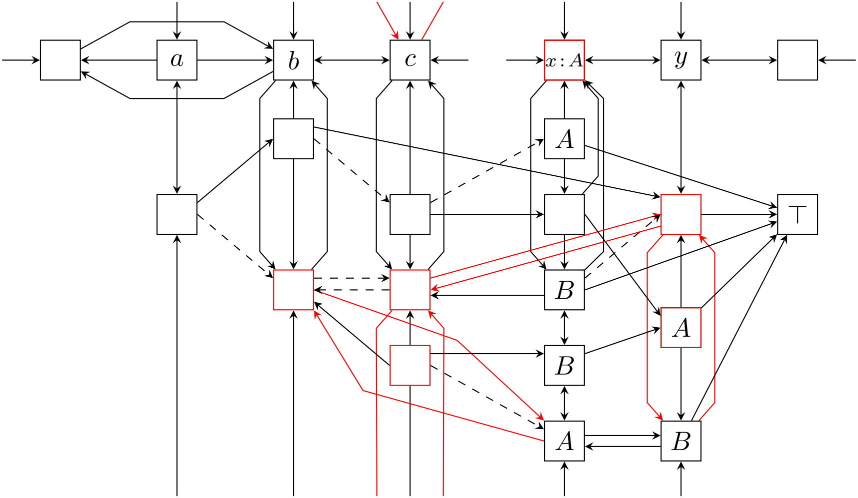
D^{3}C の適用例



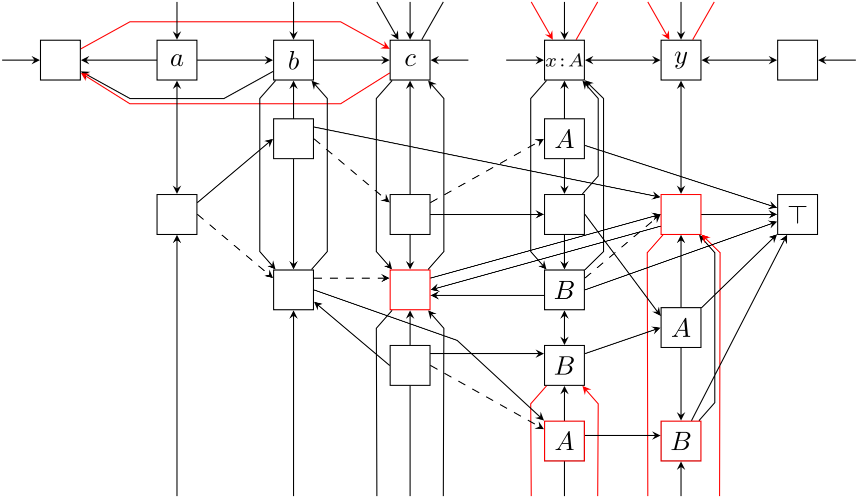
a を被覆



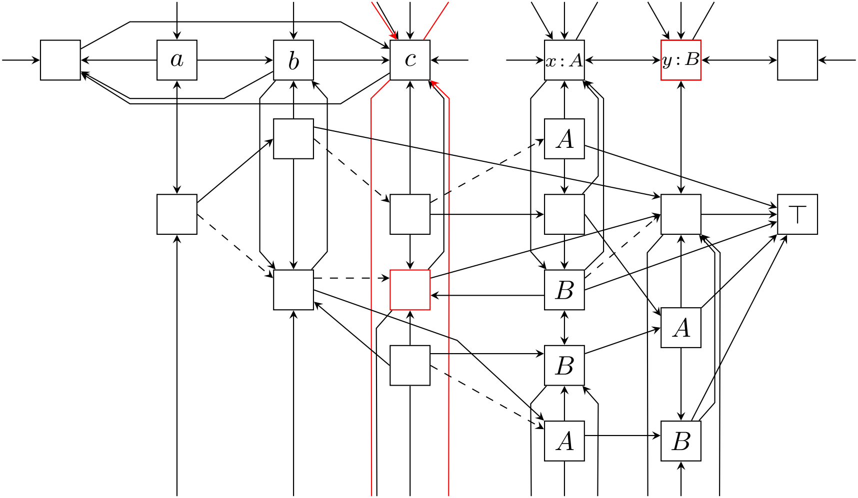
a, $x : A$ を被覆



a, $x:A$, b を被覆



a, $x:A$, b, $y:B$ を被覆



まとめと今後の課題

まとめ

- D^3X を拡張し，XCC に対応したデータ構造とアルゴリズムを提案
- 巨大な XCC インスタンスを圧縮して扱う

今後の課題

- 実装による Algorithm C との比較
- XCC で効率よく解ける問題の調査
- MCC (Covering with Multiplicities and Colors) への拡張
- SDD, ZSDD 等のデータ構造の検討