厳密被覆問題を表す ZDD に対する前処理

1220305　榎　新 星　【 アルゴリズム研究室 】

1　はじめに

数独やスケジューリング問題などは厳密被覆問題として定式化して解ける．厳密被覆問題を効率よく解く方法として Knuth が提案した DLX があるが，現実の問題は入力サイズが膨大になり，入力サイズがネックとなって DLX でも解くことができないものがある．そこで，Nishino らは入力を簡潔なデータ構造である ZDD で表現し，ZDD に新たなリンクを張って問題を解く手法 D3X を提案した [1]．本稿では，この入力の ZDD サイズを小さくする手法を提案し，その有効性を確かめる．

2　厳密被覆問題

厳密被覆問題とはアイテムの集合 U と U の冪集合の部分集合を要素とするオプションの集合 S に対して，S の部分集合で $\forall s_1, s_2 \in S'(s_1 \cap s_2 = \emptyset) \wedge (\bigcup_{s \in S'} S = U)$ を満たす S' を求める問題である．例えば，

$$\begin{aligned} U &= \{a, b, c, d, e, f, g\}, \\ S &= \{\{b, e\}, \{a, c, d\}, \{b, d, e, f\}, \{a, c\}, \\ &\quad \{a, g\}, \{f, g\}, \{e, g\}\} \end{aligned} \tag{1}$$

に対しては，$S' = \{\{b, e\}, \{a, c, d\}, \{f, g\}\}$ が解となる．

Zero-suppressed Binary Decision Diagram (ZDD) は組合せ集合を簡潔に表現できるデータ構造である．図 1 の ZDD は (1) の S を表す ZDD である．

3　提案手法

あるアイテム i を含むオプション全てが，別のアイテム j を必ず含むとき，j は不要なアイテムであり，i を含まずに j を含むオプションも不要である．(1) ではアイテム c を含むオプションはアイテム a を含んでいるためアイテム a とオプション $\{a, g\}$ が不要である．この処理を行うアルゴリズムを Algorithm1 に示す．3 行目はアイテム i を含まない組合せを表現する ZDD を返す演算

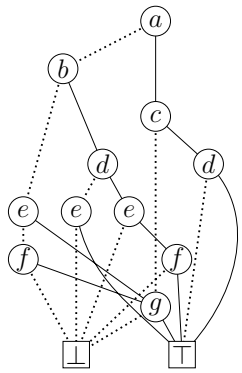

図 1　ZDD

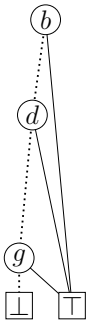

図 2　前処理後 ZDD

```
   Input  : ZDD z
   Output: Preprocessed ZDD z
1  while There is item to be deleted do
2  |  for i ∈ U(item set) do
3  |  |  z_1 ← z.OffSet(i);
4  |  |  for (j ∈ U) ∧ (j ≠ i) do
5  |  |  |  if z_1.OnSet(j) = ⊥ then
6  |  |  |  |  remove item i and update z;
7  |  |  |  |  break;
8  return z;
```

Algorithm 1: ZDD に対する前処理

表 1　前処理の時間と D3X の実行時間

インスタンス	前処理	前処理有	前処理無
burma14	5ms	2,852ms	2,844ms
grafo8373.100	434ms	**0ms**	483,778ms
grafo8224.100	22ms	**124,714ms**	199,406ms
Ion	11ms	**0ms**	1ms

である OffSet を使い，アイテム i を含まない組合せ集合を表現する z_1 を作成する．5 行目はアイテム j を含む組合せ集合を表現する ZDD を返す演算 OnSet を用い，アイテム i を含まず j を含む組合せ集合が空集合かを判定する．そのような組合せが無い場合は，i に関する不要なアイテムやオプションを削除する．Algorithm1 を図 2 の ZDD に適用するとアイテムが $\{b, d, g\}$ のみでオプションが $\{\{b\}, \{d\}, \{g\}\}$ である図 2 の ZDD になる．

4　計算機実験

提案手法の前処理を適用して得られた ZDD と元の ZDD に対して D3X を適用した際の実行時間を表 1 に示す．表 1 の 0ms は前処理の段階で解が無いことがわかったことを表す．表 1 から，前処理を行うことで一部の D3X の実行時間が大幅に減少することがわかる．

5　まとめ

本稿では厳密被覆問題を表現する ZDD をより小さくする前処理技法を提案し，その有効性を確かめた．

参考文献

[1] M. Nishino, N. Yasuda, and K. Nakamura,“Compressing Exact Cover Problems with Zero-suppressed Binary Decision Diagrams”, IJCAI-21, Aug. 2021, pp. 1996–2004