二分決定グラフに基づく大規模ハイパーグラフの極小横断列挙

戸田貴久[1,2]　湊真一[2,1]

JST 湊ERATOプロジェクト[1]
北海道大学大学院情報科学研究科[2]

２０１３年７月２６日　第３回CSPSAT2研究会

発表の概要

- ハイパーグラフの極小横断列挙
  - 基本概念と問題定義
  - 既存研究
- データ構造ZDD
- 提案法
  - ZDDを用いた極小横断の列挙
- 計算機実験
  - 提案手法の性能評価
- 発表のまとめと今後の展開

基礎概念と問題定義

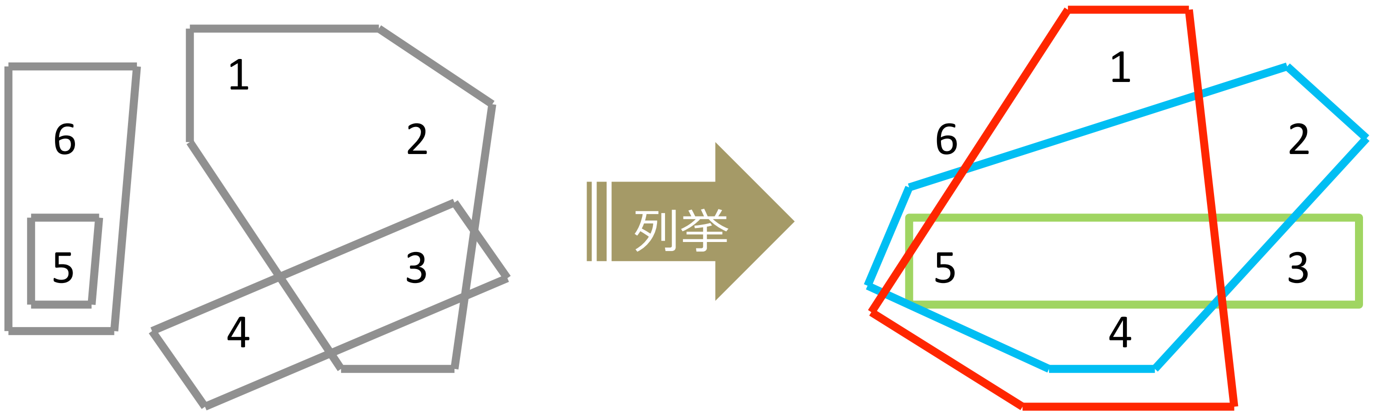
- ハイパーグラフ $H=(V, E)$
  - V: 台集合
  - E: Vの部分集合の集まり
    E の元はハイパーエッジ
- Eの横断（ヒッティング集合）
  - V の部分集合で、 Eのすべてのハイパーエッジと交差するもの

ハイパーグラフ極小横断の列挙

入力　ハイパーグラフ

出力　すべての極小横断

例）



発表の概要

- ハイパーグラフの極小横断列挙
  - 基本概念と問題定義
  - 既存研究
- データ構造ZDD
- 提案法
- 計算機実験
- 発表のまとめ

さまざまな分野への応用

データマイニング，論理，人工知能，・・・

決定問題	計算問題
Monotone Dual	Monotone Dualization
co-IMSAT, co-SIMSAT	Maximal frequent sets, Minimal infrequent sets generation
co-Additional World	Horn envelope
FD-RELATION EQUIVALENCE	Model-based diagnosis

論理関数の基礎概念

x1	x2	f
0	0	0
0	1	1
1	0	1
1	1	1

双対 $\Leftrightarrow$

x1	x2	f^d
1	1	1
1	0	0
0	1	0
0	0	0

- 論理関数 f の双対論理関数

  $f^d(x_1,\ldots,x_n) := \overline{f}(\overline{x}_1,\ldots,\overline{x}_n)$

  例） $f(x) = x1 \vee x2$,　$f^d(x) = x1 \wedge x2$

  $=(\overline{x}1 \vee x2) \wedge (x1 \vee \overline{x}2) \wedge (x1 \vee x2)$

- リテラル：変数あるいはその否定

  節：リテラルの論理和

  CNF：節の論理積として の論理関数の表記

  主節：論理関数によって含意される節のうち、
  どのリテラルも除去不可なもの

  主CNF：すべての主節からなるCNF

論理関数の双対化

- Dual

入力　論理関数 f と g のCNFs φ と ψ

出力　f と g は互いに双対か？

- Dualization

入力　論理関数 f のCNF φ

出力　双対論理関数 f^d の主CNF ψ

⇒　充足可能性問題を含むので一般に計算困難

単調な論理関数の双対化

- 単調な論理関数

  $u \leq v$ ならば $f(u) \leq f(v)$ を満たす論理関数

- f が単調 $\leftrightarrow$ f は定数または否定記号なしで論理和と論理積だけで表記可能

- Monotone Dual

  入力　単調な論理関数 f と g のCNFs φ と ψ

  出力　f と g は互いに双対か？

- Monotone Dualization

  入力　単調な論理関数 f のCNF φ

  出力　双対論理関数 f^d の主CNF ψ

既存結果と未解決問題

Algorithm (Fredman and Khachiyan '96)

Monotone Dualは$N^{o(\log N)}$で解くことができる。
ただし、N は入力CNFサイズの和とする。

Corollary

Monotone Dualizationは$N^{o(\log N)}$で解くことができる。ただし、N は入力と出力のCNFサイズの和とする。

未解決問題： 多項式時間で解くことができるか？

（補足）co-Monotone Dualが（準）多項式可解 $\leftrightarrow$ Monotone Dualizationが（準）多項式可解

極小横断の列挙との関係

入力

$\Phi = (x1 \lor x2 \lor x3) \land (x3 \lor x4) \land (x5 \lor x6)\ ) \land x5$

双対化

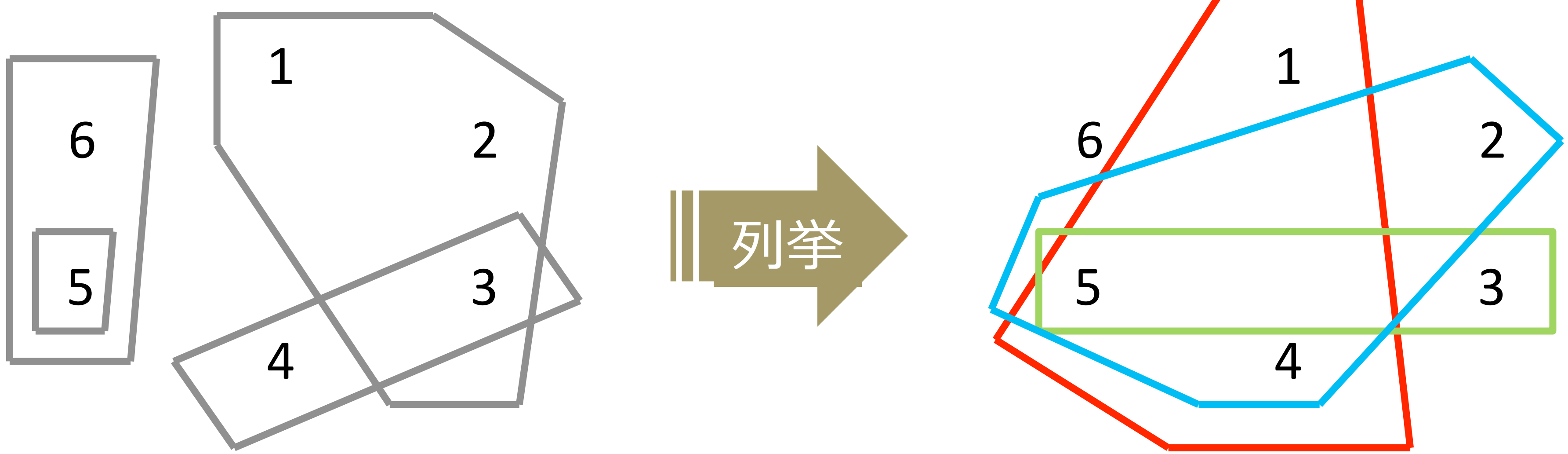
$(x1 \land x2 \land x3) \lor (x3 \land x4) \lor (x5 \land x6) \lor x5$

形式変換

出力

$\Psi = (x3 \lor x5) \land (x1 \lor x4 \lor x5) \land (x2 \lor x4 \lor x5)$

TRAS-ENUM-complete



極小横断の列挙は、
さまざまな計算問題に形をかえ現れる。

Trans-Hyp-complete	Trans-Enum-complete
Monotone Dual	Monotone Dualization
co-IMSAT, co-SIMSAT	Maximal frequent sets, Minimal infrequent sets generation
co-Additional World	Horn envelope
FD-RELATION EQUIVALENCE	Model-based diagnosis

既存アルゴリズム

ベルジュアルゴリズム型	山登りアルゴリズム型	逆探索型	ZDD型
Kavvadias-Stravropoulos (‘99)	Hérbert-Bretto-Crémilleux (‘07)	村上・宇野 (‘13)	Knuth (‘09)
Bailey-Manoukian-Ramamohanarao (‘03)			
Dong-Li (‘05)			

計算機実験で優れた性能を達成

TAOCP Vol.4a の練習問題
性能不明

Dong-Li法

入力の集合族 $F = \{U_1,\cdots, U_m\}$に対して

$F_0 := \emptyset \longrightarrow \mathrm{Tr}(F_0) := \emptyset$

$F_1 := \{U_1\} \longrightarrow$ Tr(F_1

… …

$F_i := \{U_1,\cdots, U_i\} \longrightarrow \mathrm{Tr}(F_i)$

$F_{i+1} := \{U_1,\cdots, U_i, U_{i+1}\} \longrightarrow \mathrm{Tr}(F_{i+1})$

Tr(F_i)とU_{i+1}からTr(F_{i+1})作成

(i) $S \in \mathrm{Tr}(F_i)$でU_{i+1}にも交差するならば$\mathrm{Tr}(F_{i+1}) := \mathrm{Tr}(F_{i+1}) \cup \{S\}$

(ii) そうでないとき、$S \cup \{e\}$ が極小となるすべての $e \in U_{i+1}$ を $\mathrm{Tr}(F_{i+1}) := \mathrm{Tr}(F_{i+1}) \cup \{S \cup \{e\}\}$

極小性判定のコスト高い

Tr(F_i)を記憶する必要あり

既存アルゴリズム

ベルジュアルゴリズム型	山登りアルゴリズム型	逆探索型	ZDD型
Kavvadias-Stravropoulos (‘99)	Hérbert-Bretto-Crémilleux (‘07)	村上・宇野 (‘13)	Knuth (‘09)
Bailey-Manoukian-Ramamohanarao (‘03)			
Dong-Li (‘05)			

計算機実験で優れた性能を達成

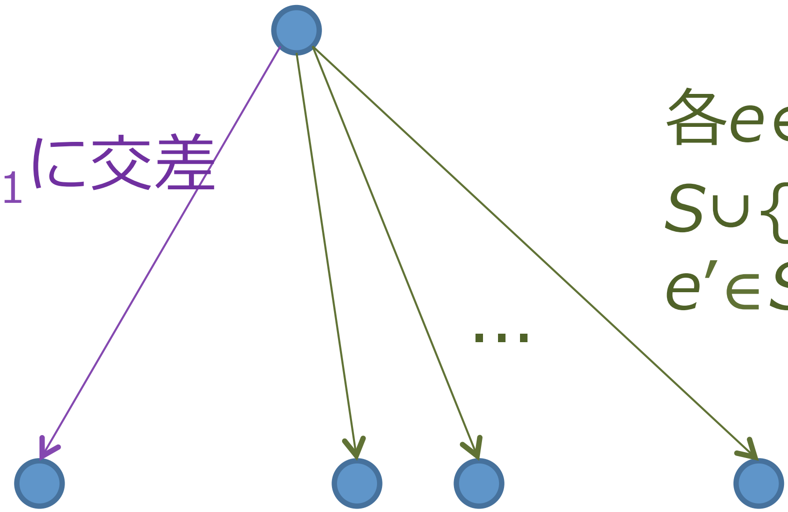
TAOCP Vol.4a の練習問題
しかし、性能不明！

Kavvadias-Stravropoulos法

集合Sは$F_i := \{U_1, \cdots, U_i\}$に対する極小横断

深さ優先探索

(S, i)



SはU_{i+1}に交差のとき

各$e \in U_{i+1}$に対して
$S \cup \{e\} - \{e'\}$が横断となる
$e' \in S$は存在しないとき

…

Tr(F_i)を記憶する必要なし
だが極小性判定のコスト依然高い

$(S, i+1)$ $(S \cup \{e\}, i+1)$

Sを1だけ拡大して$F_{i+1} := \{U_1, \cdots, U_i, U_{i+1}\}$に対する極小横断となるものたち

既存アルゴリズム

ベルジュアルゴリズム型	山登りアルゴリズム型	逆探索型	ZDD型
Kavvadias-Stravropoulos (‘99)	Hérbert-Bretto-Crémilleux (‘07)	村上・宇野 (‘13)	Knuth (‘09)
Bailey-Manoukian-Ramamohanarao (‘03)			
Dong-Li (‘05)			

計算機実験で優れた性能を達成

TAOCP Vol.4a の練習問題
性能不明

村上・宇野法（逆探索版）

入力U_iの集合族 $F = \{U_1, \cdots, U_m\}$

DFS版は割愛します。

交差できない集合ない

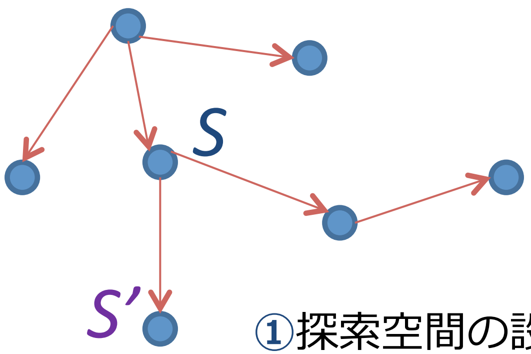
各頂点にクリティカルハイパーエッジある

極小性条件

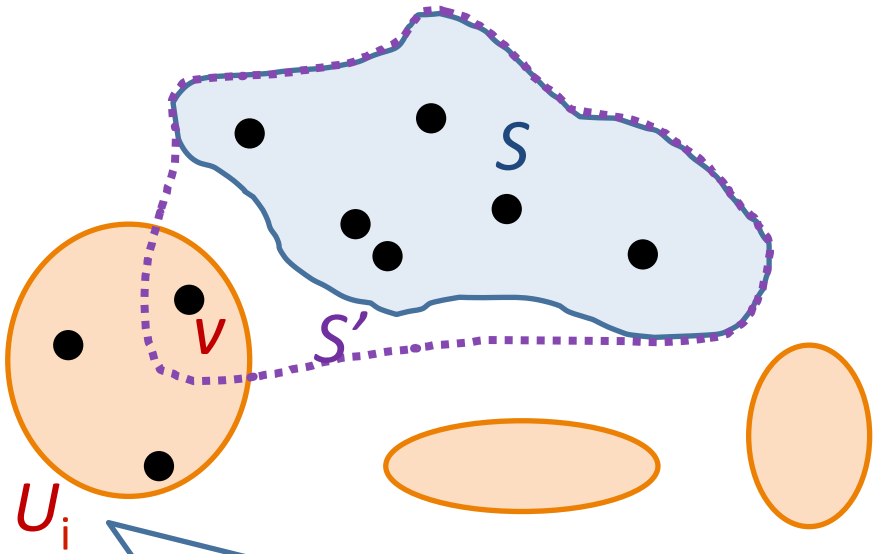
Sが極小横断 ↔ uncov(S) = ∅ かつ crit(v, S) ≠ ∅ ($\forall v \in S$)

ただし、uncov(S) := $\{U_i: S \cap U_i = \emptyset\}$、crit($v$, S) := $\{U_i: S \cap U_i = \{v\}\}$

逆探索の基本アプローチ



①探索空間の設定
②親子関係定義
③根から探索



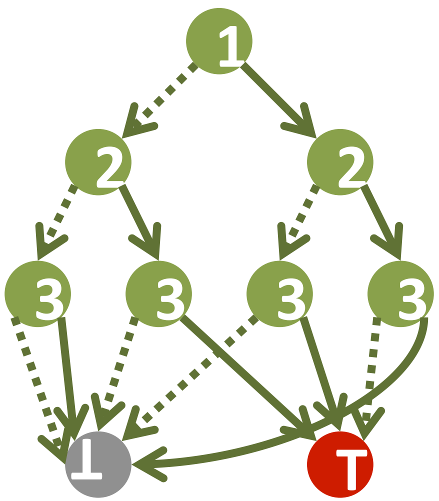
交差しない集合のうち、最小インデックスのものを選ぶ

発表の概要

- ハイパーグラフの極小横断列挙
- データ構造ZDD
- 提案法
- 計算機実験
- 発表のまとめと今後の展開

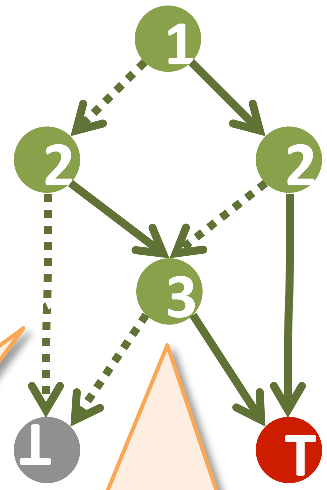
集合族のためのデータ構造

{{1,2}, {1,3}, {2,3}}の
二分木



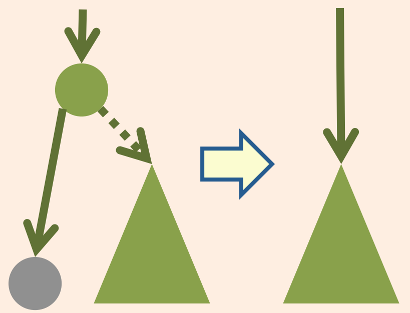
圧縮

ZDD
(**Z**ero-suppr

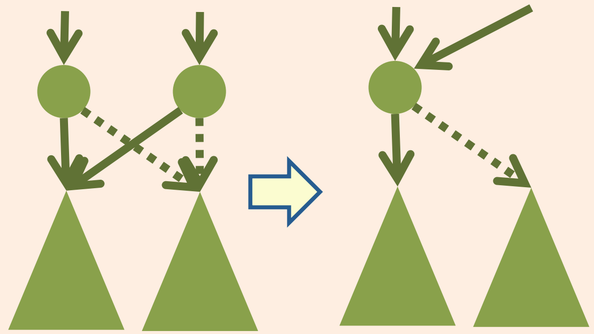


一意形！

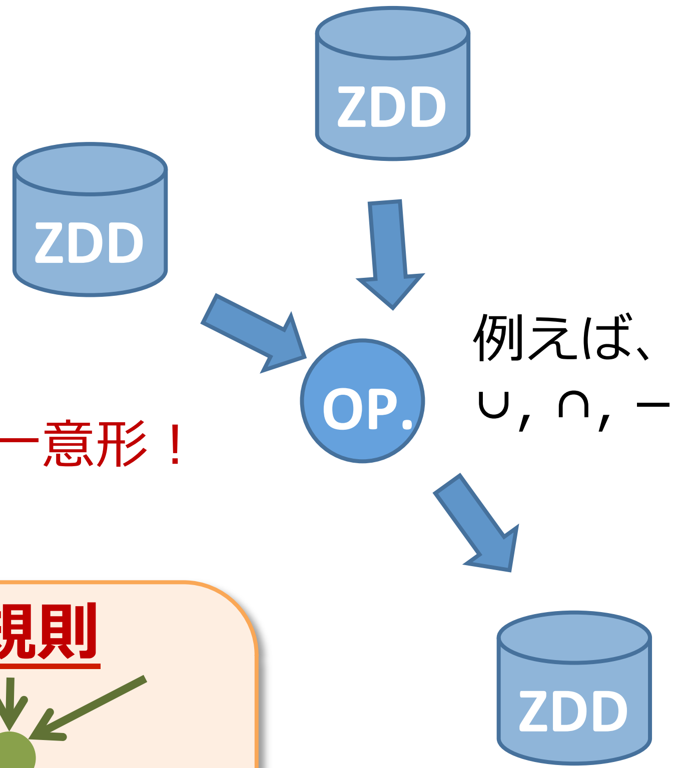
節点削除規則



節点共有規則



ZDDの効率的演算



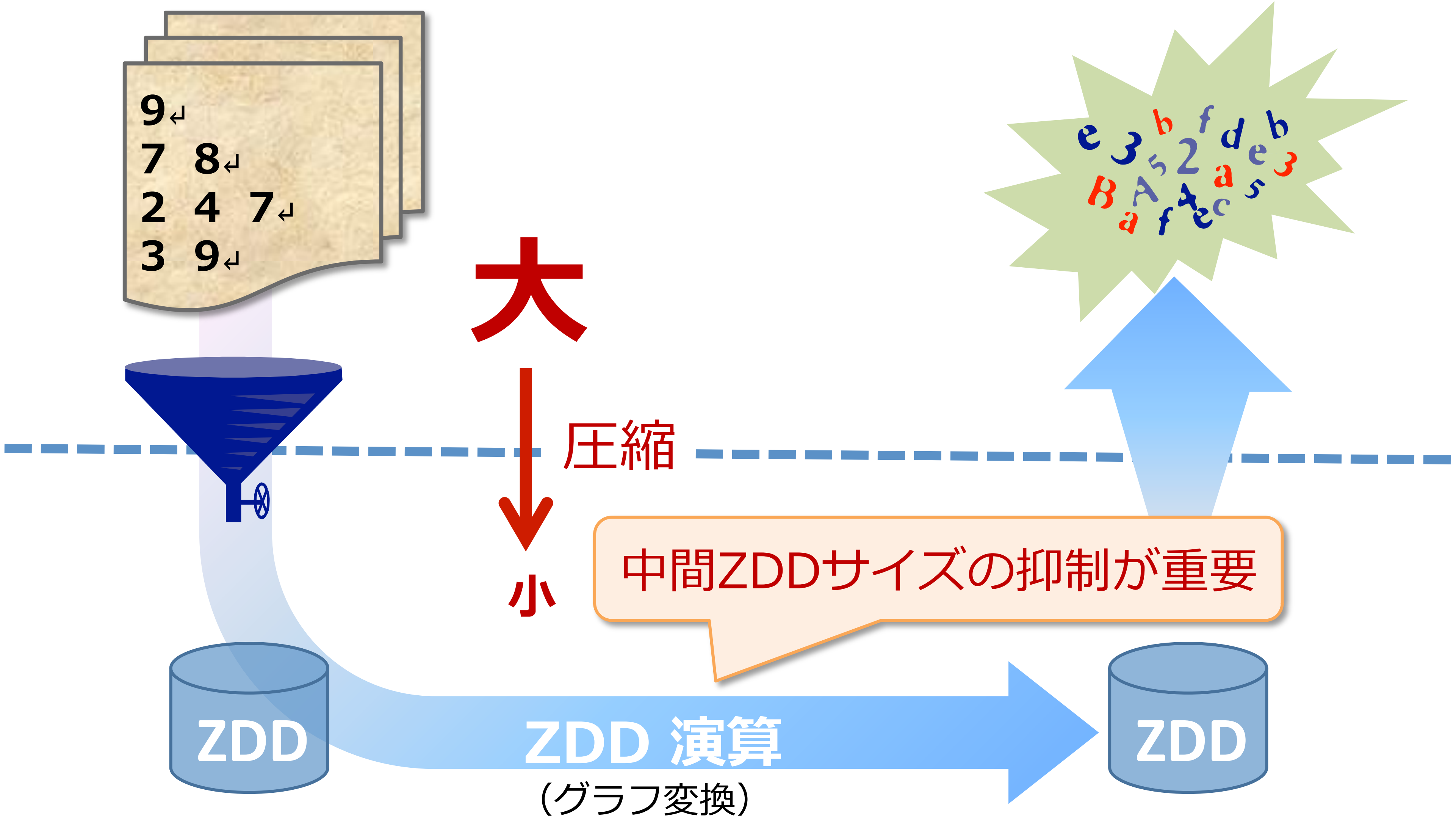
例えば、
∪, ∩, −, など

多くの実用的な演算は
入力ZDDサイズに比例
する時間で計算できる。

ZDDに基づく計算のアプローチ

[入力]　各行がハイパーエッジに対応する巨大サイズのファイル

[出力]



発表の概要

- ハイパーグラフの極小横断列挙
- データ構造ZDD
- 提案法
- 計算機実験
- 発表のまとめと今後の展開

提案法の概要

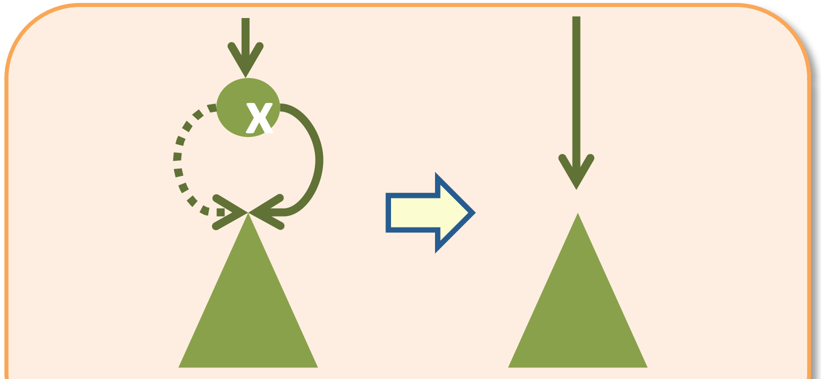
1) **圧縮部**　入力集合族をZDDに圧縮

2) **HIT部**　ZDDからすべての横断を表すBDDを構築

3) **MIN部**　BDDから極小集合だけからなるZDDを構築

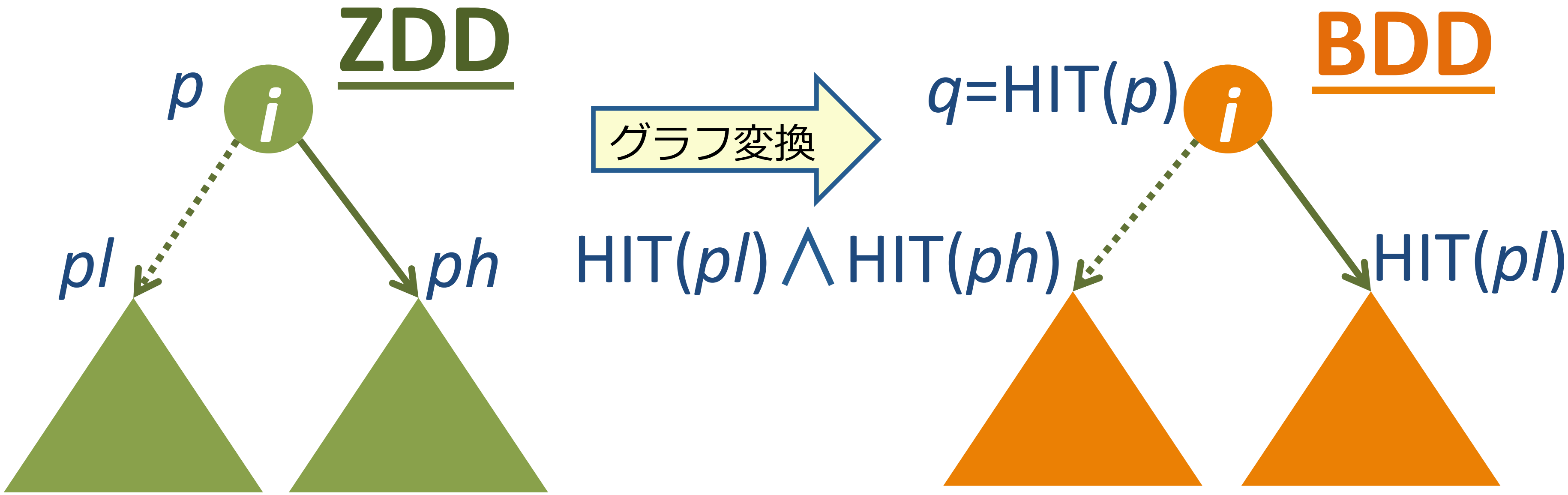
4) **解凍部**　ZDDを解凍し集合族を出力

BDD は節点削除規則を除いて
ZDDと同じデータ構造



BDDの節点削除規則

再帰関数HIT：ZDDの根pを受け取り、すべての横断を表すBDDの根qを返す



ただし、$p=\bot$のとき$q=\top$を返す。$p=\top$のとき$q=\bot$を返す。

CNFの節集合
（制約の集まり）

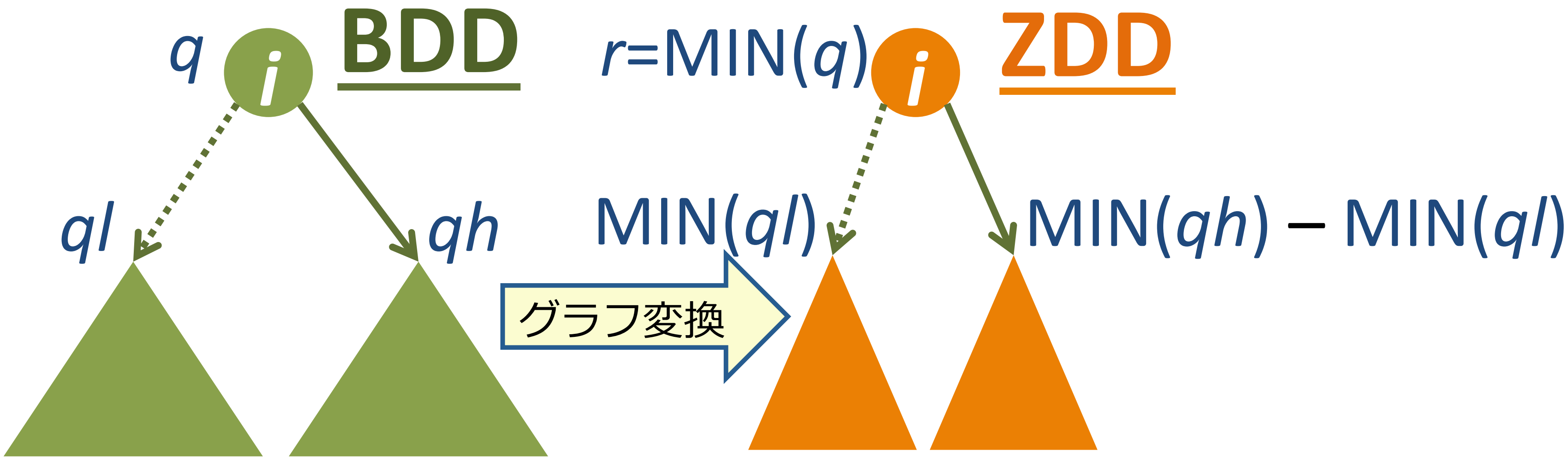
対応する論理関数
（制約を満たす解集合を表現）

BDDを直接構築するのは難しい！

計算される論理関数は単調である。
なぜなら、バイナリベクトルuと集合$U:=\{i:u_i=1\}$との対応により、

$U\subseteq U'$のときUが横断ならばU'もまた横断 $\leftrightarrow$ $u\leq u'$ならば$f(u)\leq f(u')$

$\boxed{\text{再帰関数}\textbf{MIN}}$：BDDの根$q$を受け取り、極小集合からなるZDDの根$r$を返す



ただし、$q=\perp$のとき$r=\perp$ を返す。$q=\top$のとき$r=\top$を返す。

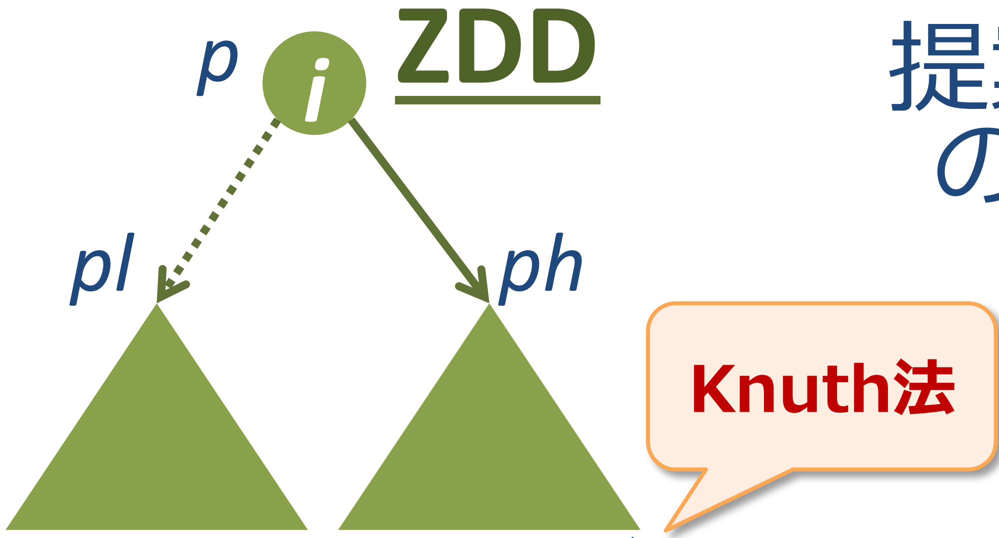
単調な論理関数
（解集合）

同じ論理関数の主項の集まり

（注意）一般の論理関数では正しく動作しないが、HITの後に使うとＯＫ！

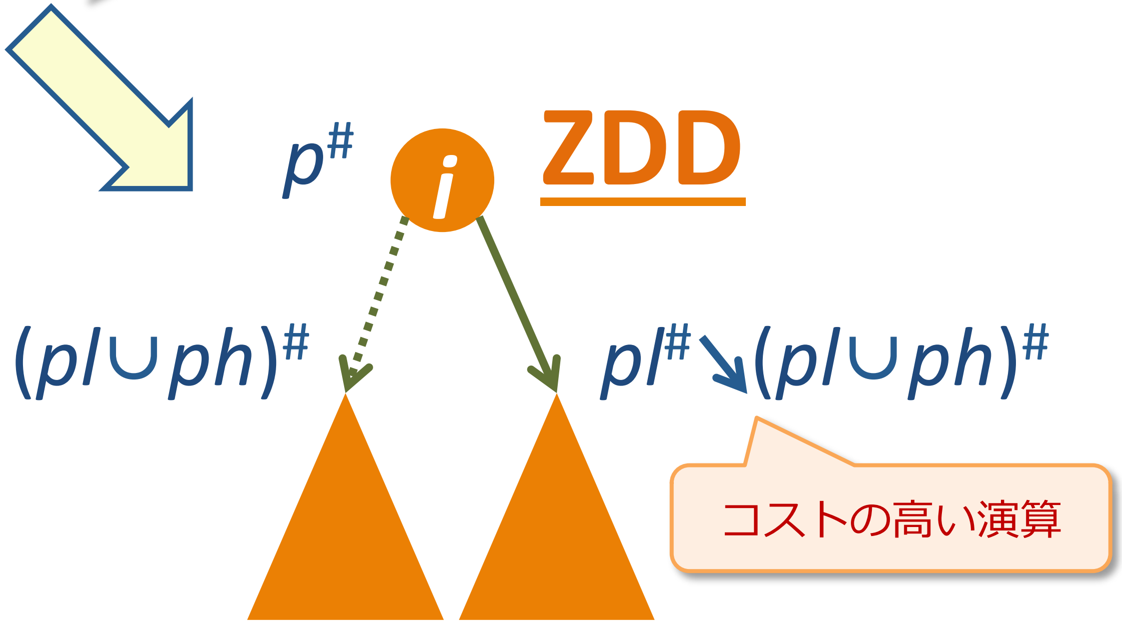
理論的な未解決問題（Knuth先生のTAOCP Vol.4a p.674）

単調論理関数fに対してO(|Z(PI(f))|)=O(|B(f)|)が成り立つか？



提案法とKnuth法の違いは何か？

①ZDDだけを用いる。
②途中で横断すべてを求めないで、直接極小横断を求めている。
③それにより我々の極小化演算を使えず、単純な差分以上の処理が必要！



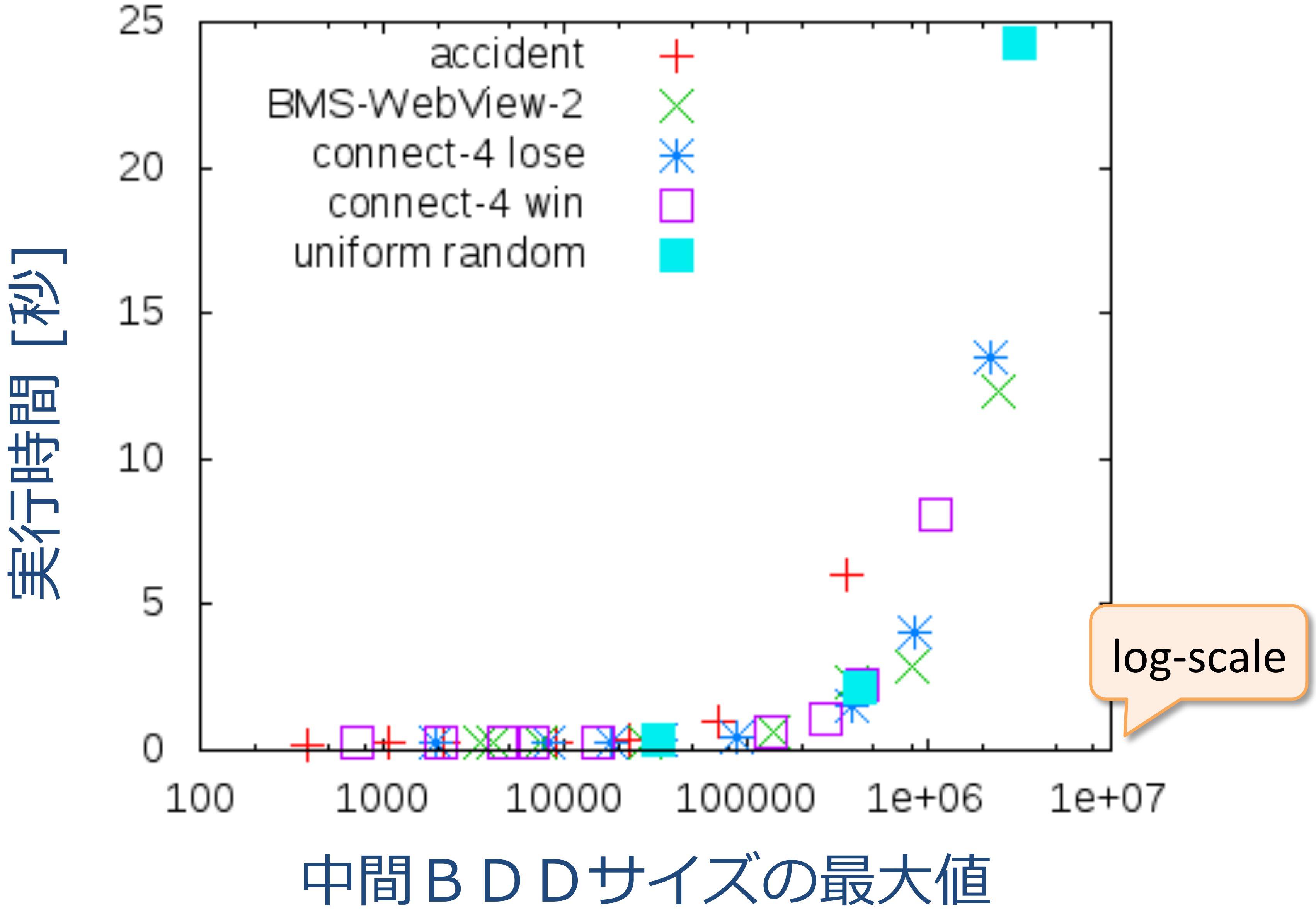
Reference

Knuth, D.: The Art of Computer Programming, vol. 4. Addison-Wesley Professional, New Jersey (2011), pp.669–670

発表の概要

- ハイパーグラフの極小横断列挙
- データ構造ZDD
- 提案法
- 計算機実験
  - 実験１：提案法の性能を左右する因子
  - 実験２：既存手法との性能比較
  - 実験のまとめ
- 発表のまとめと今後の展開

(1) HIT部とMIN部を合わせた時間



発表の概要

- ハイパーグラフの極小横断列挙
- データ構造ZDD
- 提案法
- 計算機実験
  - 実験１：提案法の性能を左右する因子
  - 実験２：既存手法との性能比較
- 発表のまとめと今後の展開

(2) アルゴリズムの比較

プログラム

- **Toda:** 提案法(圧縮 + HIT部 + MIN部 + 解凍部)
- **Knuth:** *TAOCP* Vol.4aで与えられた方法（我々が実装）
- **MU-0, MU-D:** 村上・宇野法（彼らのＨＰで公開）

入力データ

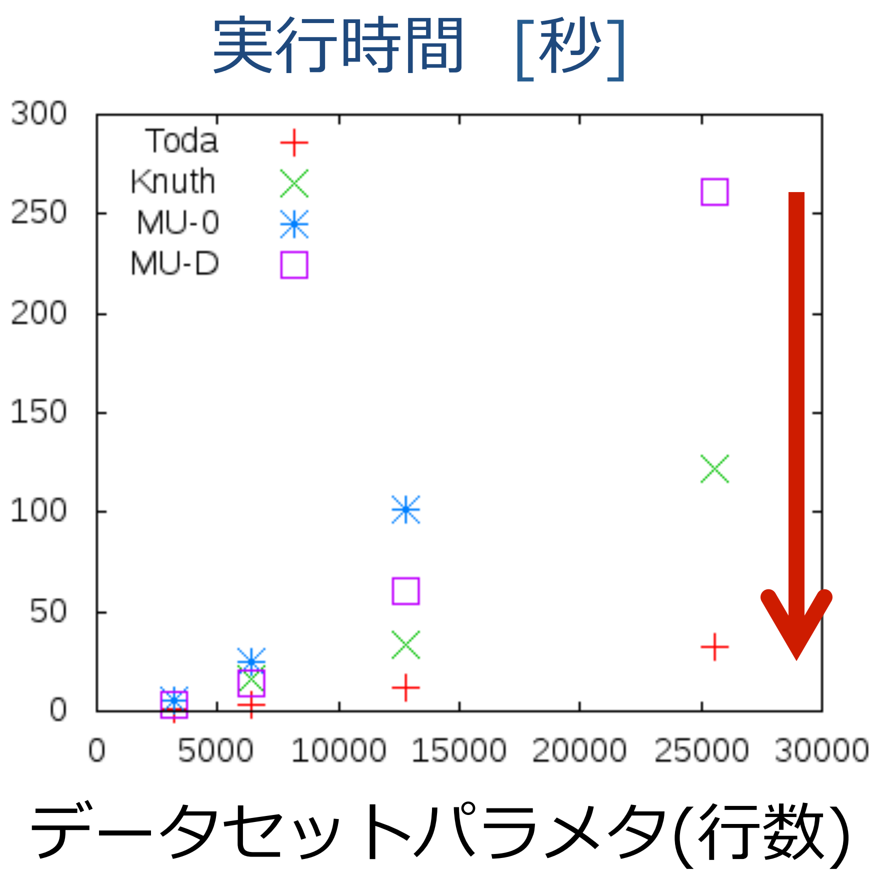
- １０種類合計９０個データセット
  （既存研究でよく使用されている）

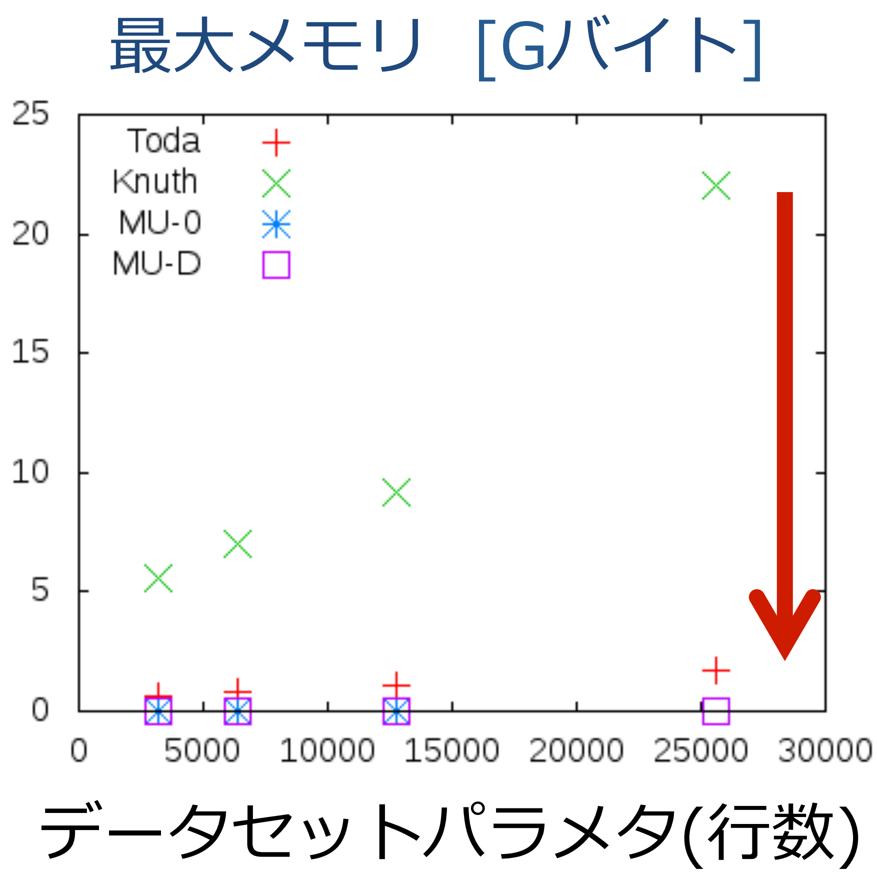
制限時間

- 1000 秒

connect-4 win

パズルのデータセット： ボードゲームconnect-4の先手必勝局面からなる





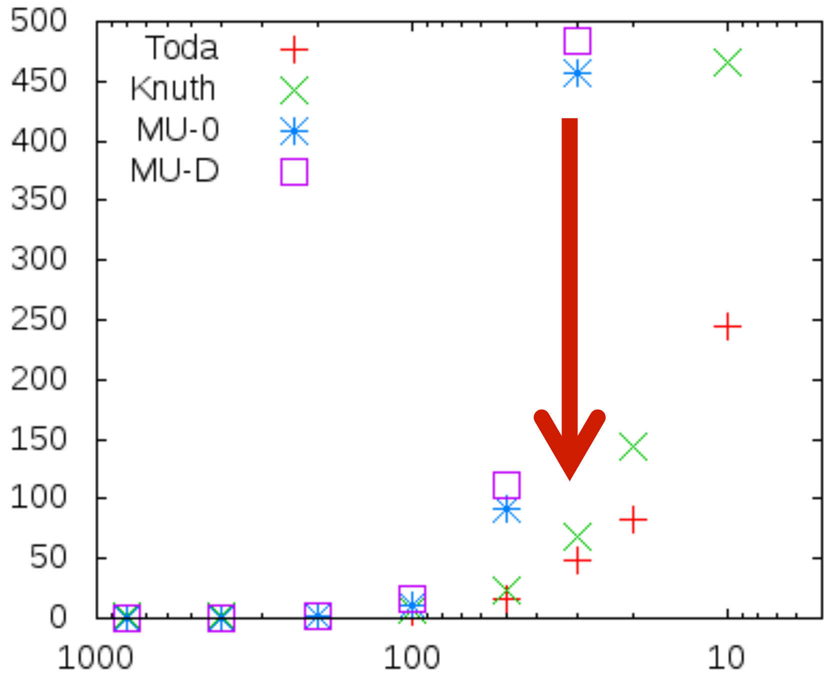
データセットの入手先

Hypergraph Dualization Repository (2013), http://research.nii.ac.jp/~uno/dualization.html

BMS-Web-View2

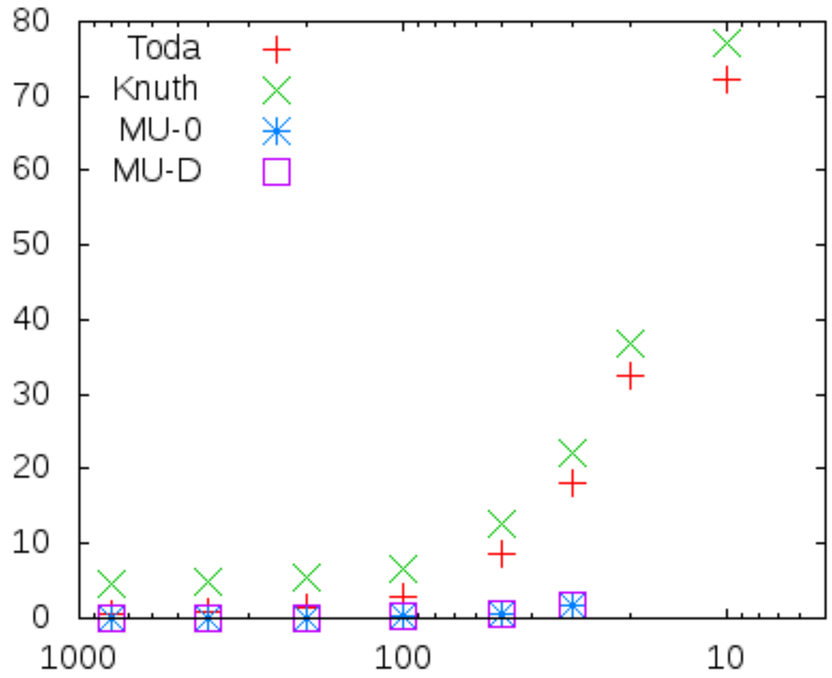
現実のデータセット：極大頻出集合から極小頻出集合の計算に対応

実行時間 [秒]



データセットパラメタ(閾値)

最大メモリ [Gバイト]



データセットパラメタ(閾値)

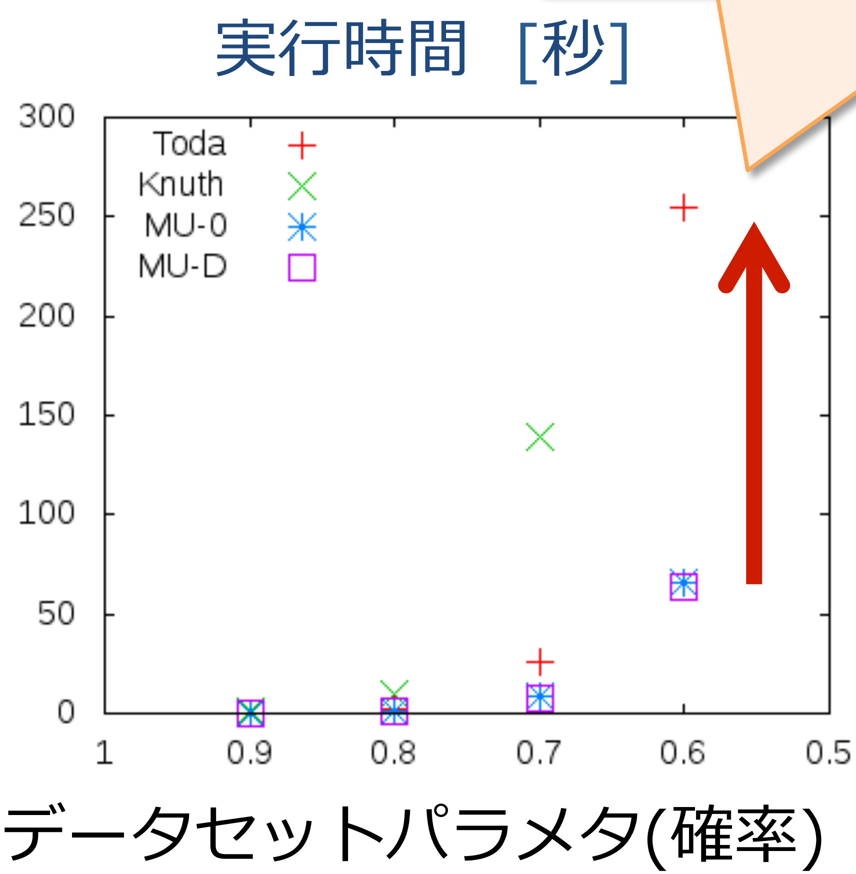
データセットの入手先

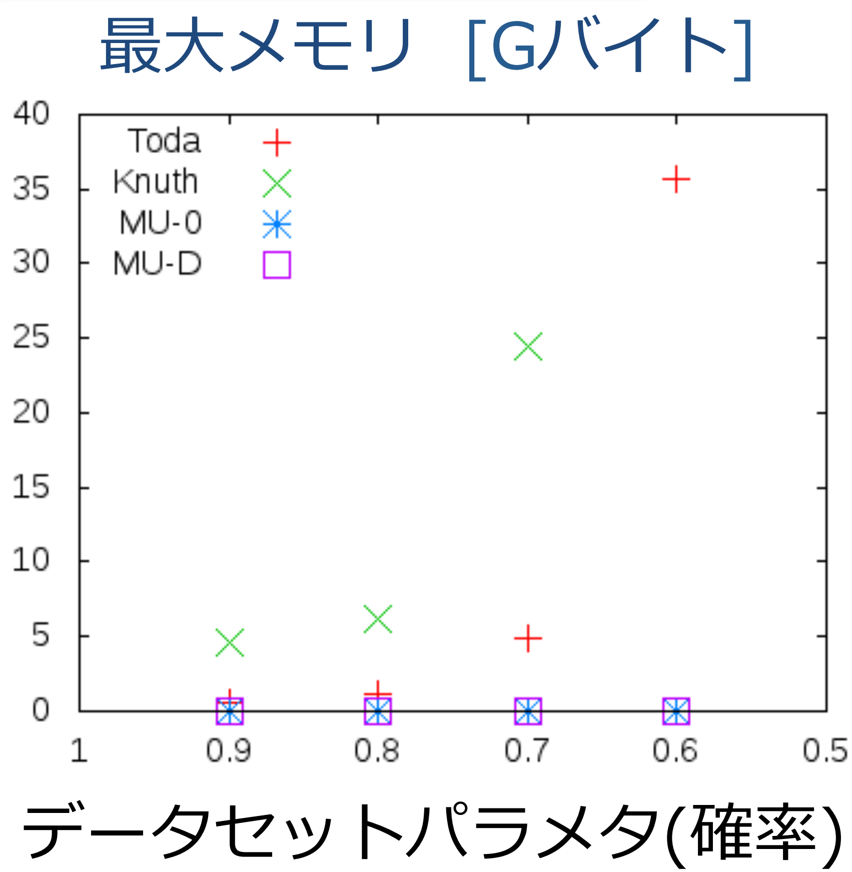
Hypergraph Dualization Repository (2013), http://research.nii.ac.jp/~uno/dualization.html

Uniform Random

ランダム生成したデータセット

中間BDDサイズ＝入力ZDDサイズの1378倍！





データセットの入手先

Hypergraph Dualization Repository (2013), http://research.nii.ac.jp/~uno/dualization.html

実験のまとめ

- 提案法：中間BDDサイズ が性能左右
- ほとんどの入力データにおいて、
  Knuth法や村上・宇野法よりも
  提案法はかなり速い。
- ランダムなデータセットなど苦手なものもある。では、何が苦手/得意か？
- 提案法およびKnuth法はメモリ使用量大

発表の概要

- ハイパーグラフの極小横断列挙
- データ構造ZDD
- 提案法
- 計算機実験
- 発表のまとめと今後の展開

発表のまとめ

- ハイパーグラフの極小横断列挙
  - 計算機科学に多くの応用
  - 実用上高速に動作するアルゴリズム開発盛ん
- ZDDに基づく計算アプローチ
- 提案法
  - Knuth法の亜種
    - 従来法とはまったく異なるパラダイム
  - 基本アイディア
    - すべての横断列挙は無謀に思われるが、BDDでコンパクトに表現できる上、効率的な極小化演算が可能
    - 適切なデータ表現の選択：
制約の集まりはZDDで表現、解集合はBDDで表現
- 計算機実験
  - 実験したほとんどのデータで提案法は従来法より著しく速い。
  - 大規模データのときメモリ使用量大きい（多くの従来法はそのようなデータを現実的な時間内に処理できなかったのでそれほど大きな欠点ではない）。

提案法の実装公開しています⇒　http://kuma-san.net/htcbdd.html