ZDDを用いた分割統治法に基づくパス数え上げアルゴリズム

◎**前田 恵太**　岩崎 巧実　藤岡 祐太
塩田 拓海　斎藤 寿樹

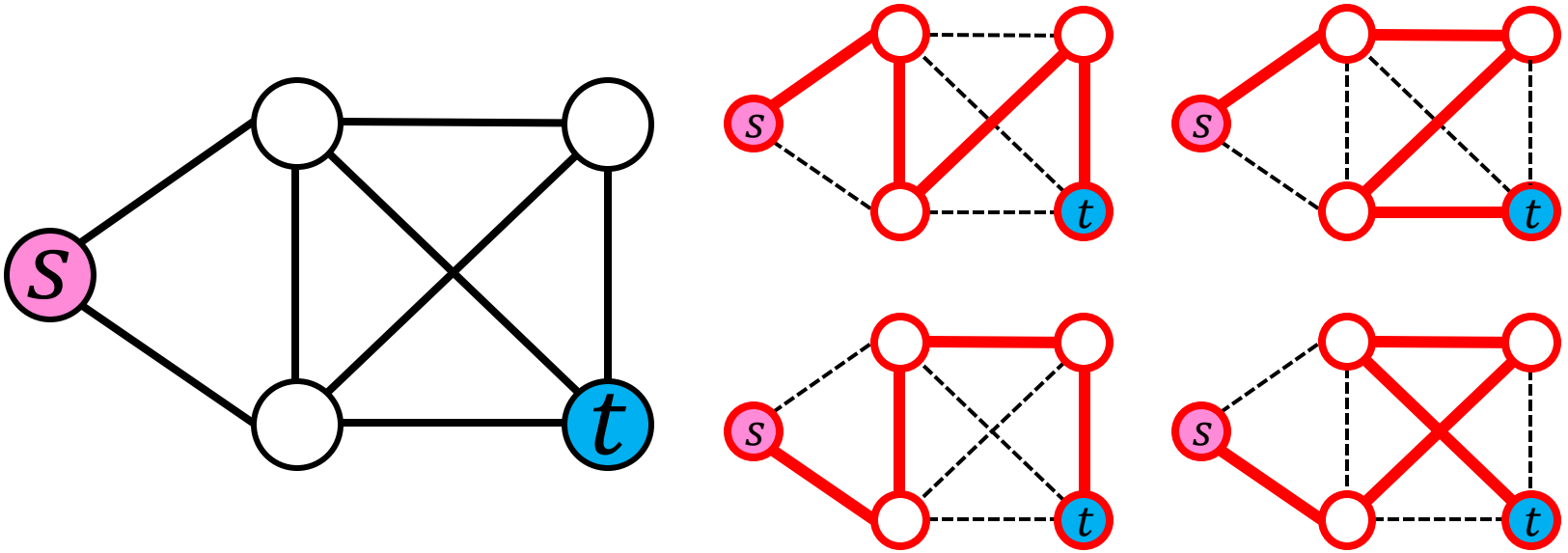
九州工業大学

パス数え上げ問題

入力：グラフ $G(V,E)$，頂点 s, t，自然数 ℓ

出力：長さ ℓ の s から t へのパスの本数

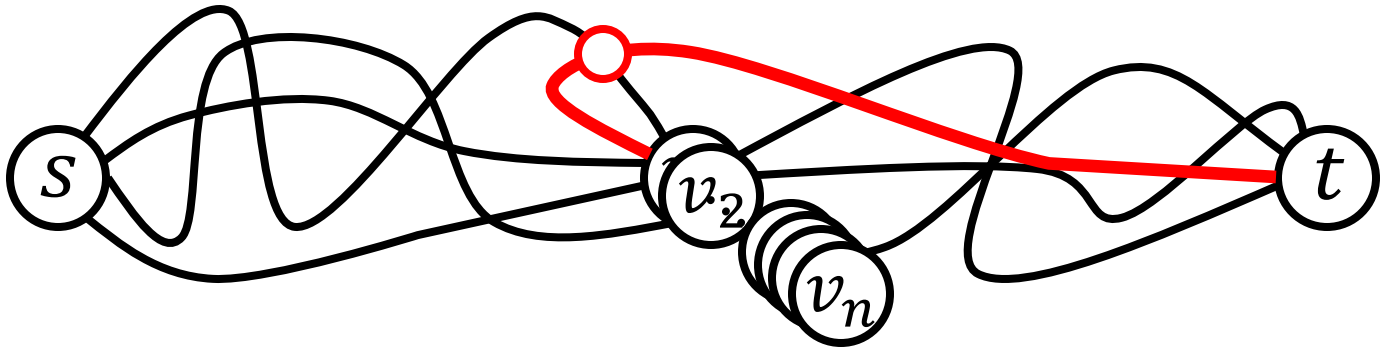
例えば…



$\ell = 4$ の $s-t$ パスの本数は 4

本研究におけるアプローチ

✓ 求めたい長さのパスを短く分割しつないでいく



$s-t$ パスの本数 $=$ $s-v_1$ パスの本数 $\times$ v_1-t パスの本数

$+$ $s-v_2$ パスの本数 $\times$ v_2-t パスの本数

$\cdots$

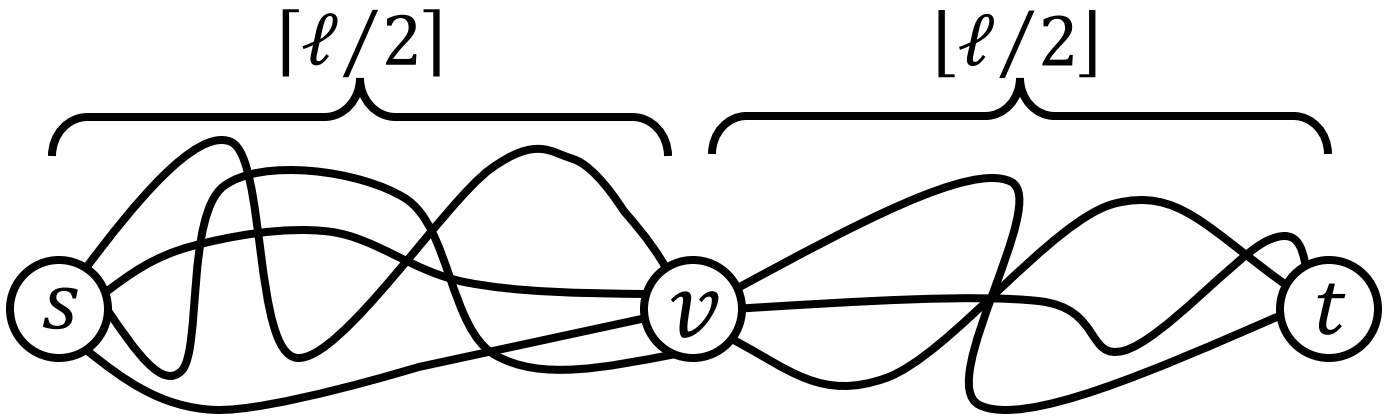
$+$ $s-v_n$ パスの本数 $\times$ v_n-t パスの本数

✓ 同じ頂点を含むと単純パスは形成されない

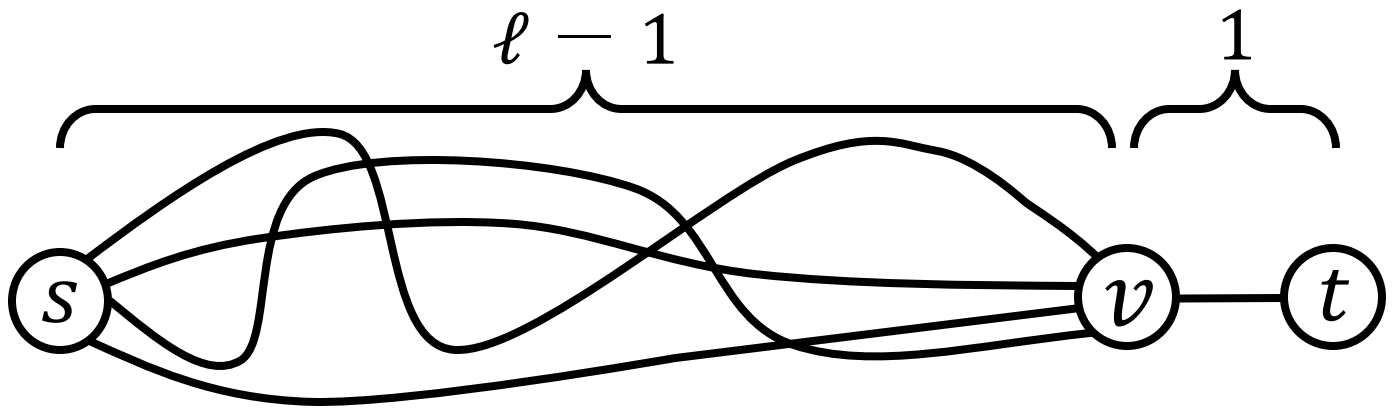
➡ 同じ頂点を含まないパスどうしをつなぐ

本研究におけるアプローチ

① **半分－半分に分割：** $\left(\left\lceil \frac{\ell}{2} \right\rceil, \left\lfloor \frac{\ell}{2} \right\rfloor\right)$



② **一辺ずつ分割：** $(\ell - 1, 1)$



本研究における成果

- ✓ **ZDD (**Zero-Suppressed Decision Diagram**)** を用いて，パスを短く分割してつなぐという考えに基づく数え上げアルゴリズムを提案

- ✓ 単純な mate-frontier 法 [Kawahara et al., 2017] による実装よりも高速に動作する

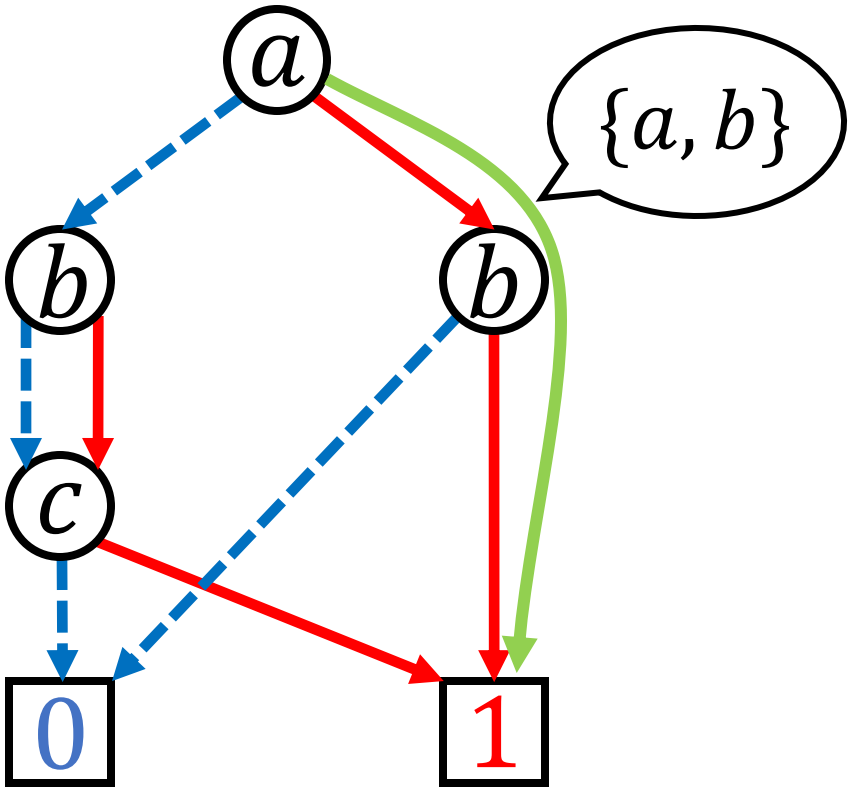
- ✓ ZDD における新たな演算「**素集合結合演算**」を考案・実装した

ZDD

→ 要素を選択する
⇢ 要素を選択しない

ZDD：組合せ集合を圧縮して表現するグラフ

[例] $S = \{\{c\}, \{ab\}, \{bc\}\}$ を表す ZDD

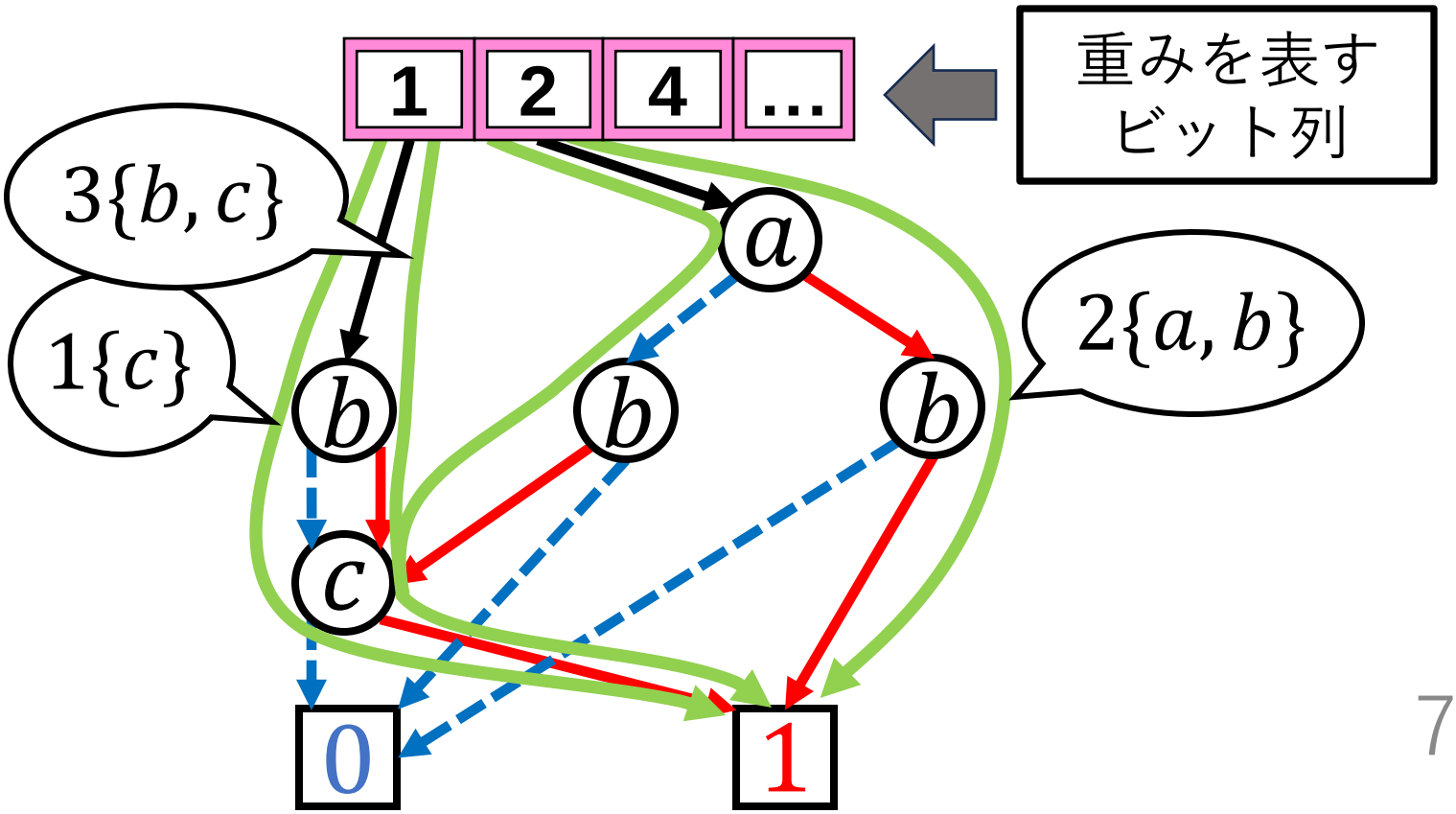


ZDD

要素を選択する
要素を選択しない

各組合せ集合に**重み**を持たせることができる

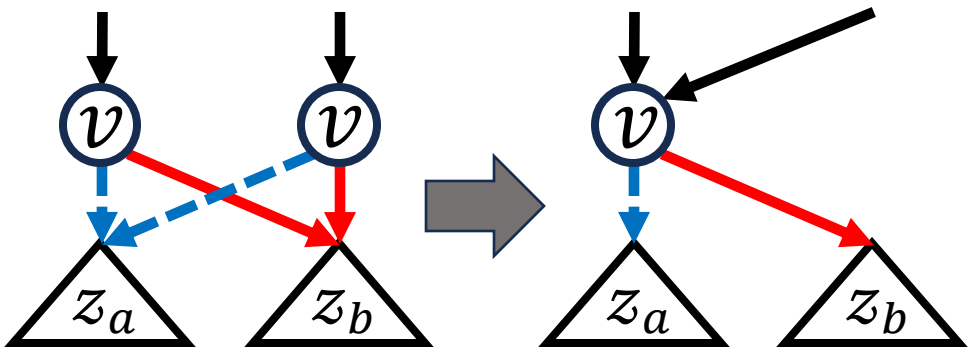
[例] $M = \{1\{c\}, 2\{ab\}, 3\{bc\}\}$ を表す ZDD



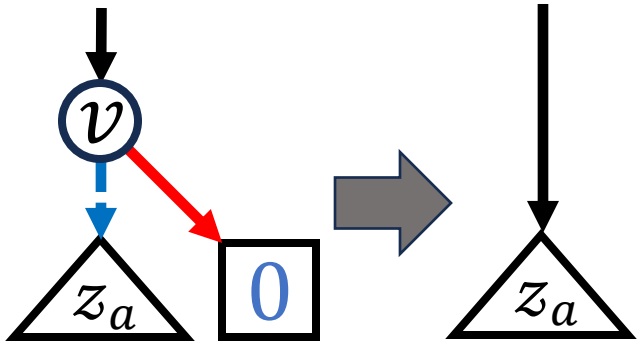
ZDD

要素を選択する
要素を選択しない

✓ 簡約化規則

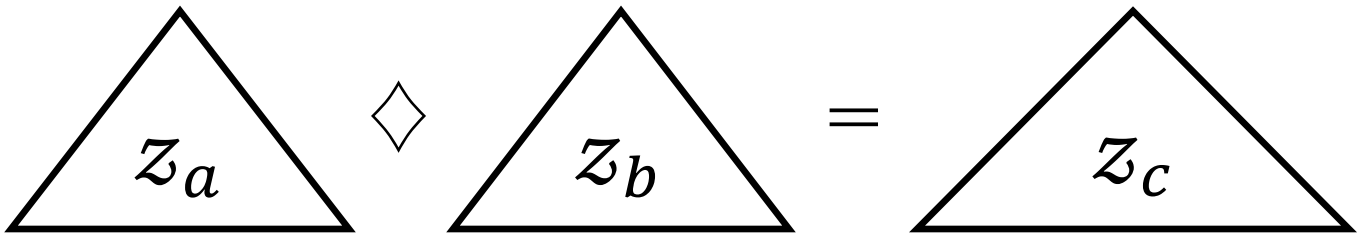


等価節点の共有



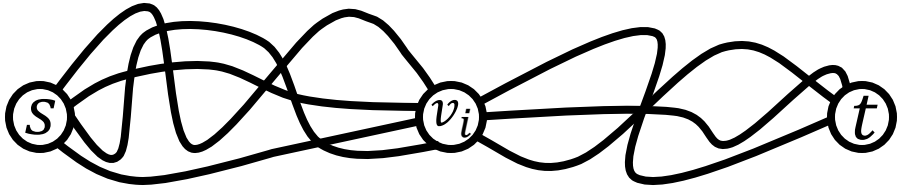
冗長節点の削除

✓ 集合どうしの演算を高速に行う演算体系

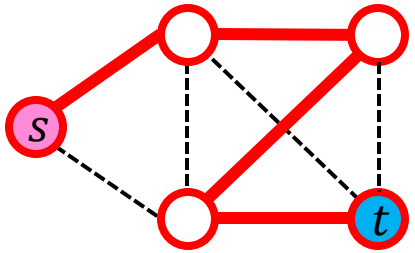


ただし ♢ ={∪,∩,\,/, ⋈, … }

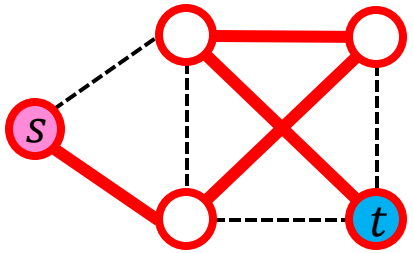
パスの表現方法



- ✓ 組合せ多重集合 $\boldsymbol{\mathcal{F}}(s,t,\ell)$：長さ ℓ の s–t パス
  組合せの要素：パスが通過する頂点
  重み w：各組合せの個数

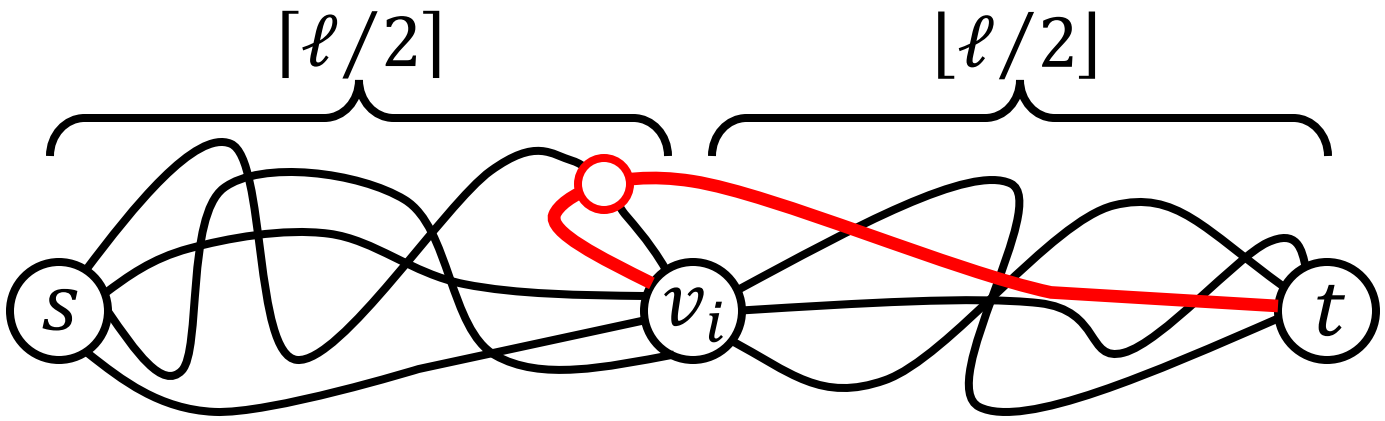


- ✓ 各組合せ多重集合を ZDD で表現
- ✓ ZDD の演算を用いてパスをつなげる



$$\bigcup_{v\in V\setminus\{s,t\}} \left(\boldsymbol{\mathcal{F}}(s,v_i,\ell-k) \diamond \boldsymbol{\mathcal{F}}(v_i,t,k) \right) = \boldsymbol{\mathcal{F}}(s,t,\ell)$$

$\left(\left\lceil \frac{\ell}{2} \right\rceil, \left\lfloor \frac{\ell}{2} \right\rfloor\right)$ アルゴリズム

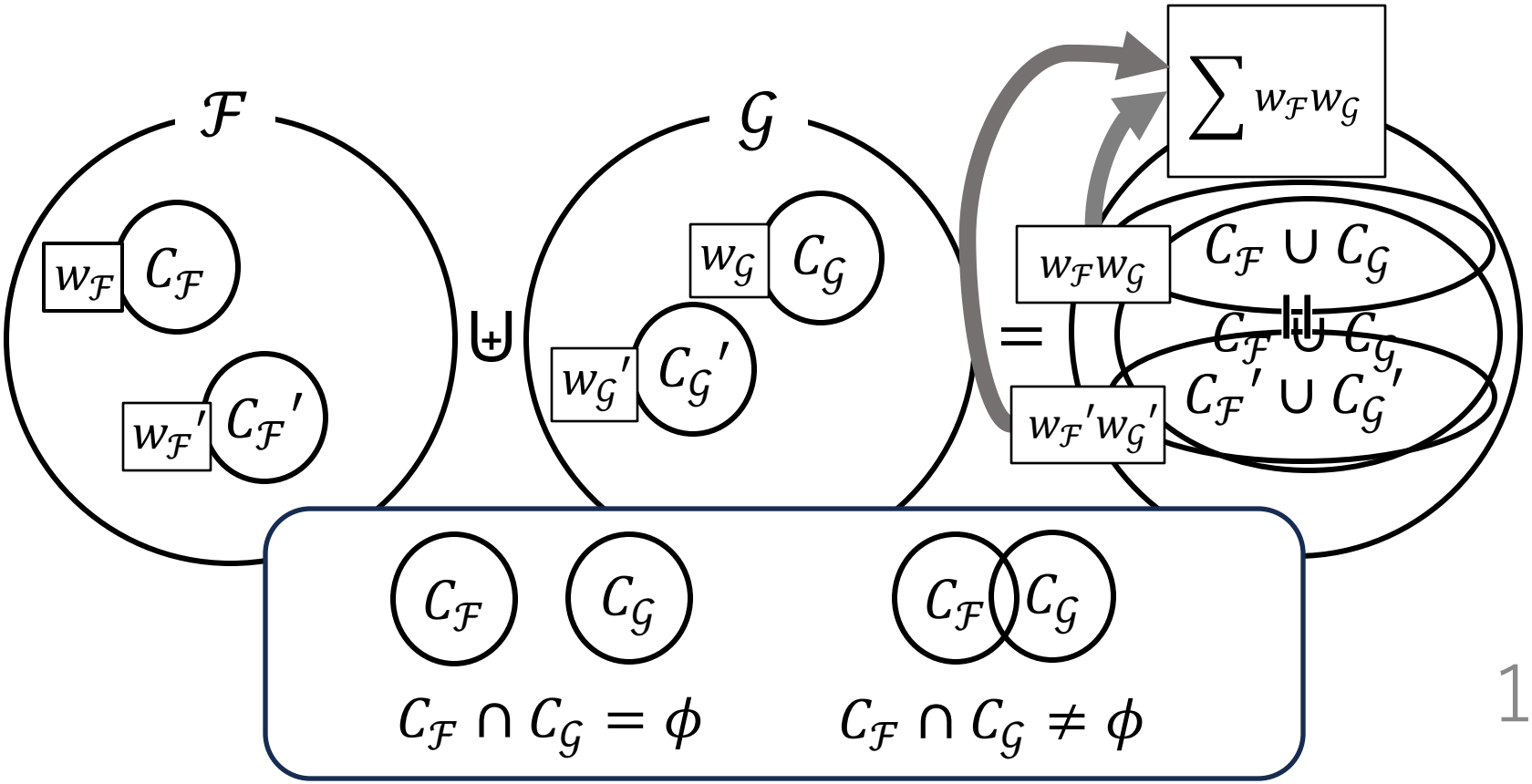


- ✓ $s-v_i$ パスにつなげても良い v_i-t パス
  - ➡ 頂点 v_i 以外に頂点の重なりがないパス

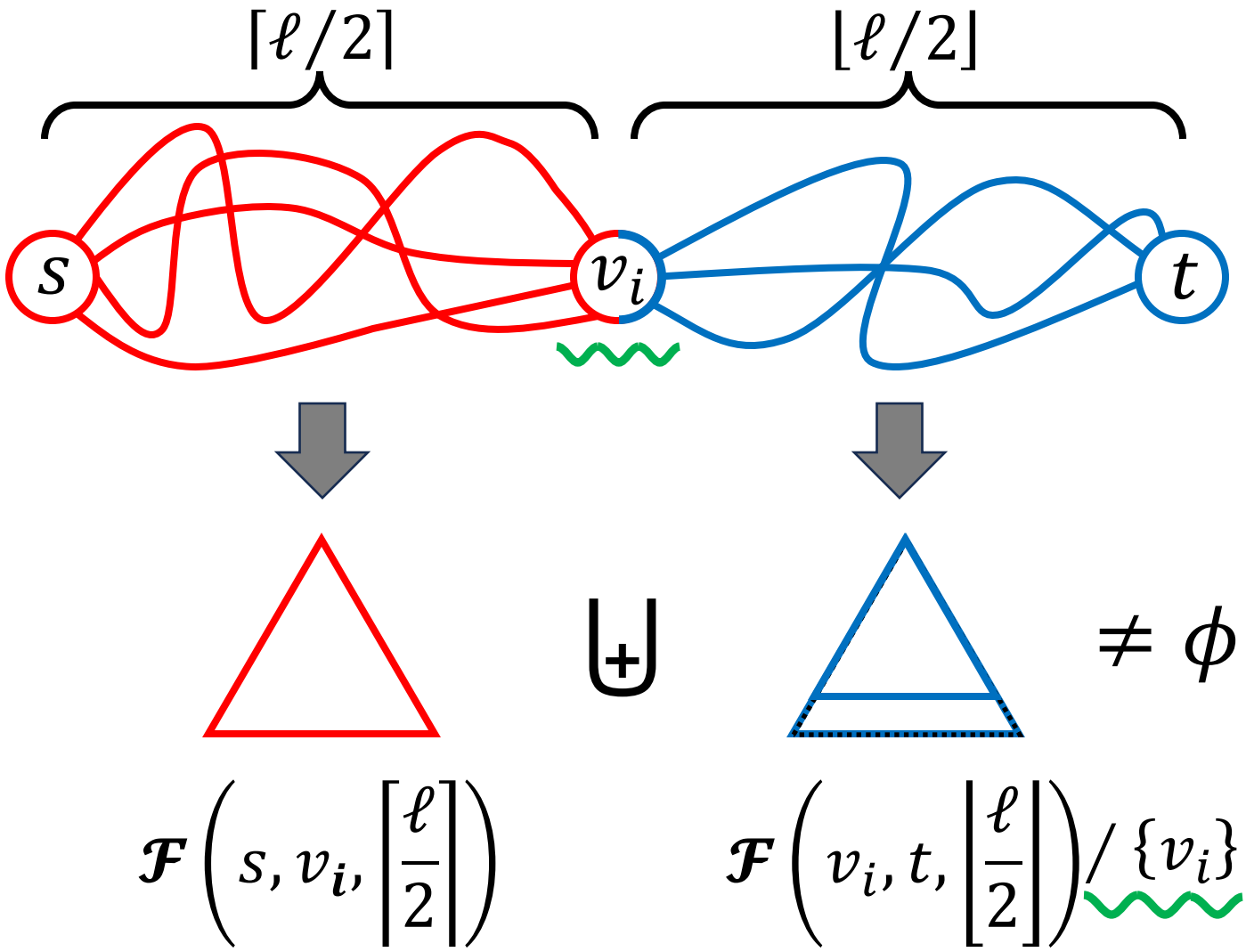
2つの組合せ多重集合に重なりがない時にのみ
集合どうしを結合する 素集合結合演算を提案

素集合結合演算

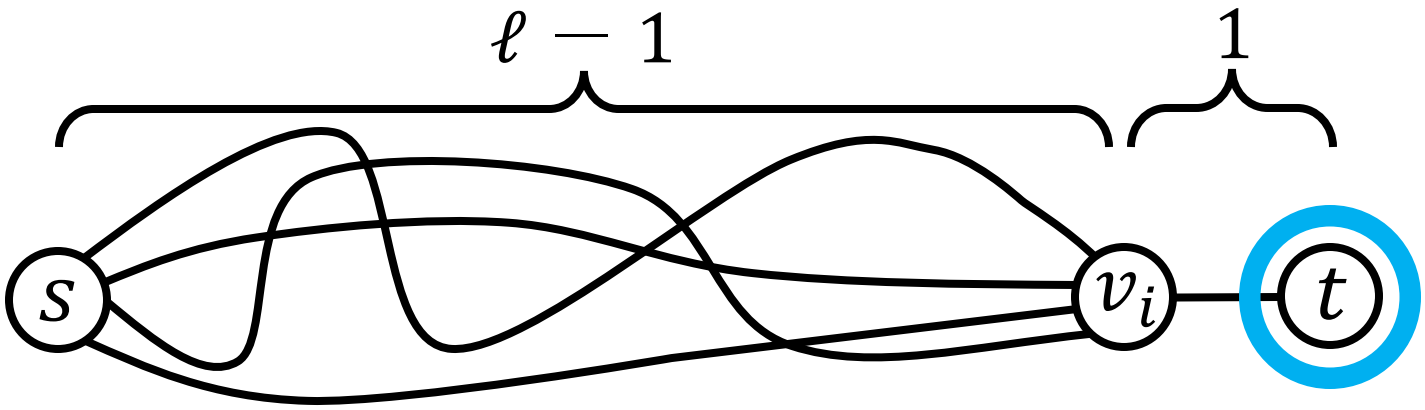
$$\mathcal{F} \uplus \mathcal{G} = \left\{ \left(\sum_{w_{\mathcal{F}} C_{\mathcal{F}} \in \mathcal{F},\, w_{\mathcal{G}} C_{\mathcal{G}} \in \mathcal{G}} w_{\mathcal{F}} w_{\mathcal{G}} \right) \left(C_{\mathcal{F}} \cup C_{\mathcal{G}} \right) \middle|\, C_{\mathcal{F}} \cap C_{\mathcal{G}} = \phi \right\}$$



$\left(\left\lceil \frac{\ell}{2} \right\rceil, \left\lfloor \frac{\ell}{2} \right\rfloor\right)$ アルゴリズム



$(\ell-1, 1)$アルゴリズム

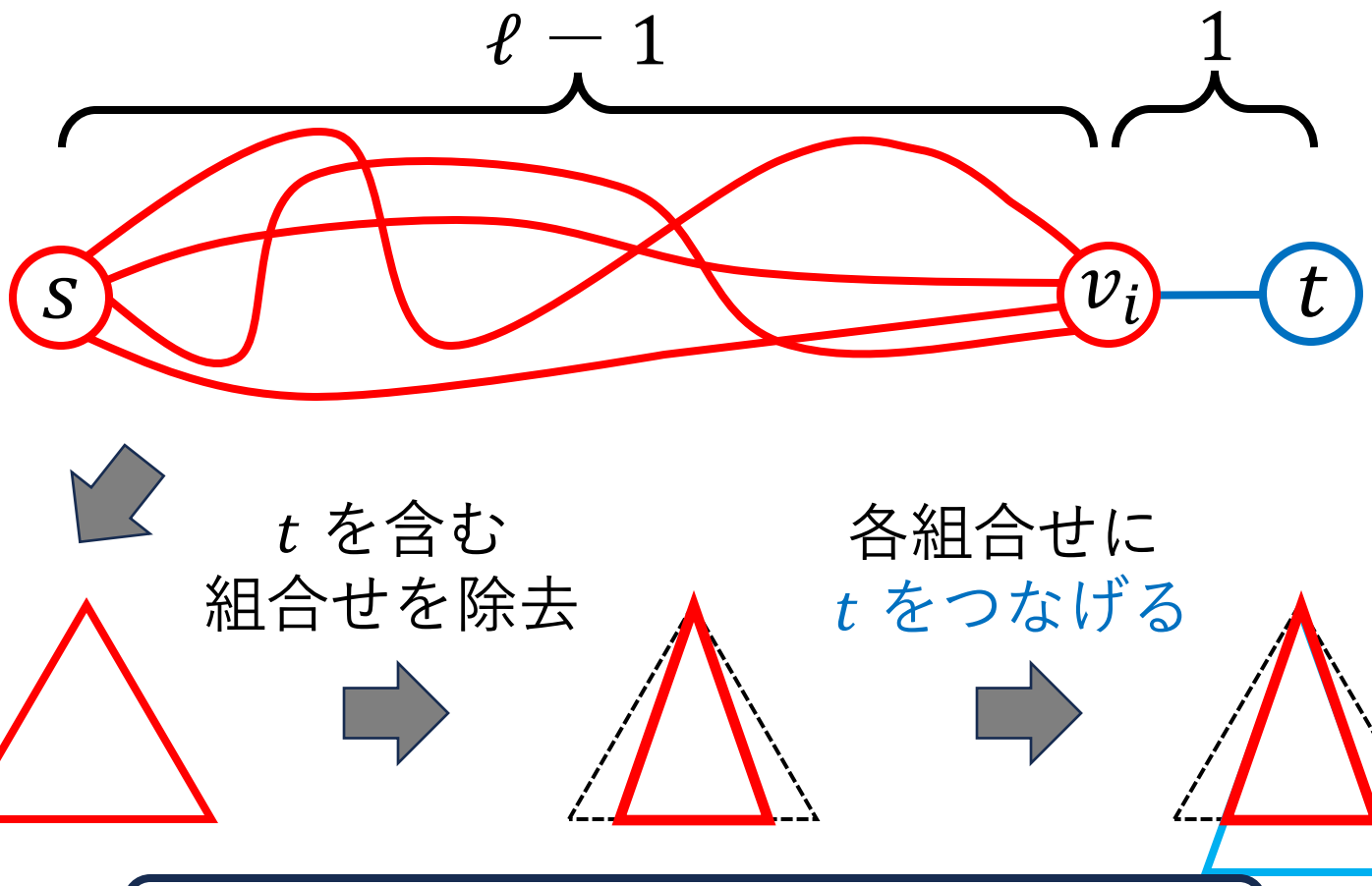


✓ $s-v_i$ パスにつなげても良い v_i-t パス

➡ 頂点 v_i 以外に頂点の重なりがないパス

頂点 t のみを見れば良い

$(\ell-1,1)$アルゴリズム



$\mathcal{F}(s, v_i, t$ ... $, t, \ell)$

素集合結合演算が不要！

計算機実験

目的：パス数え上げアルゴリズムの性能評価

入力：グラフ $G(V,E)$，パスの端点 s, t，自然数 ℓ が設定されたインスタンス 100 問 (提供：ICGCA)

制限時間：600秒／インスタンス

比較手法：mate-frontier 法 [Kawahara et al., 2017]

実験環境

CPU：Intel(R) Core(TM) i7-8565U @ 1.80GHz

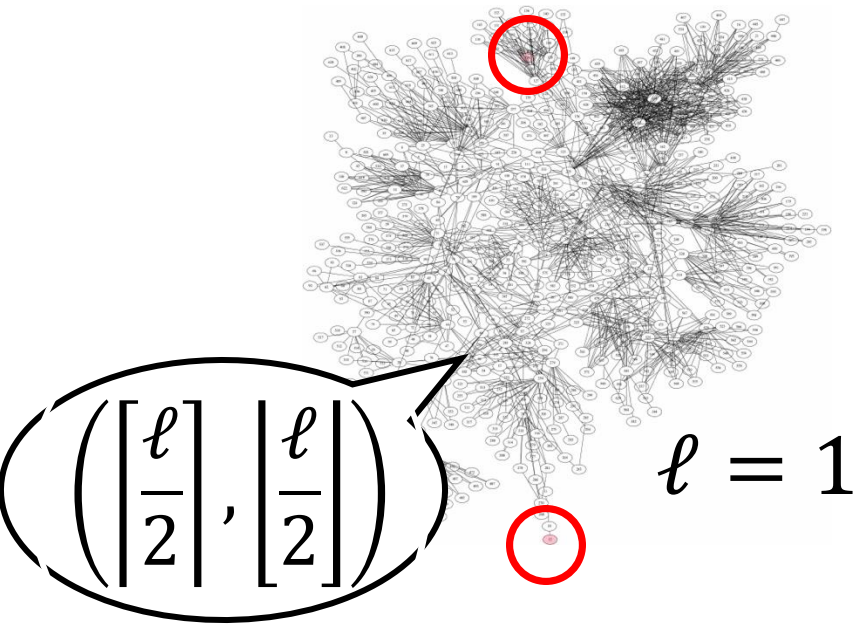
OS：Ubuntu 22.04.2 LTS　**メモリ**：8GB

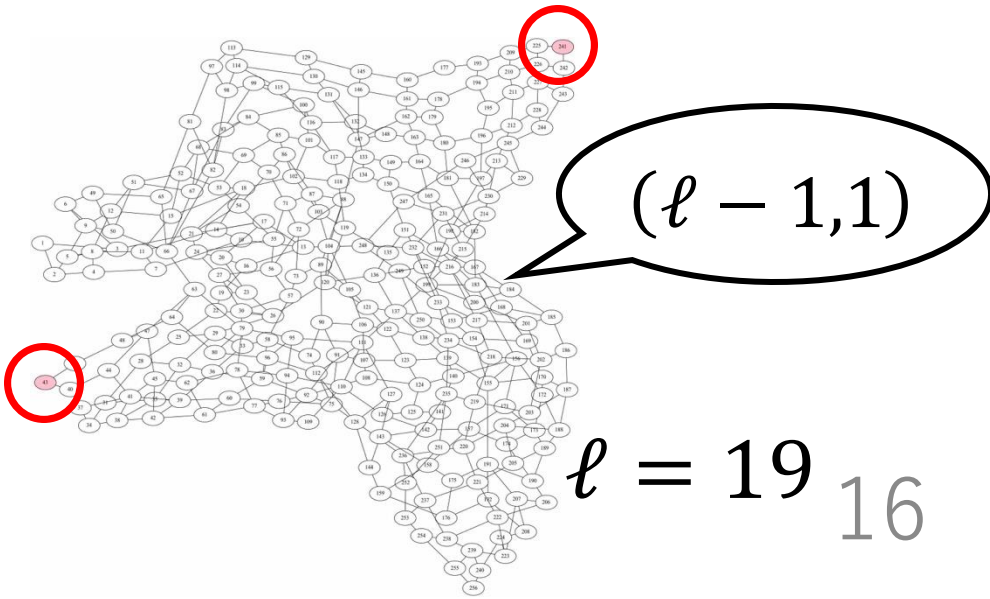
使用したライブラリ

提案法： SAPPOROBDD　mate-frontier 法：TdZdd

実験結果

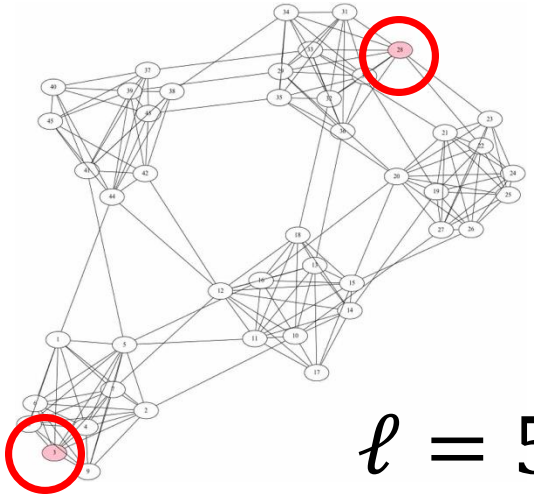
	$\left(\left\lceil \frac{\ell}{2} \right\rceil, \left\lfloor \frac{\ell}{2} \right\rfloor\right)$	$(\ell - 1, 1)$	**mate-frontier 法** [Kawahara et al., 2017]
解答数	**77**	**63**	**27**
当該手法のみ解けた問題数	**11**	**1**	**2**



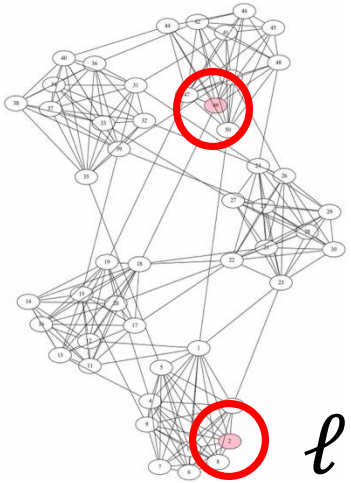


実験結果

	$\left(\left\lceil\frac{\ell}{2}\right\rceil, \left\lfloor\frac{\ell}{2}\right\rfloor\right)$	$(\ell-1, 1)$	**mate-frontier 法** [Kawahara et al., 2017]
解答数	**77**	**63**	**27**
当該手法のみ解けた問題数	**11**	**1**	**2**



$\ell = 50$



$\ell = 45$

mate-frontier法でのみ解けたインスタンス

まとめと今後の課題

- ✓ **提案手法とグラフの特性の関係性の把握**
  - ◎ 求める長さ　◎ 平面性
  - ◎ クリークの数と大きさ　◎ パス幅の大小
- ✓ **計算の高速化**

  パス分解に基づく頂点順序を導入する
- ✓ **素集合結合演算の他の問題への応用を検討**