「フカシギの数え方」― 組合せ爆発に立ち向かう最先端アルゴリズム技術

湊　真一

北海道大学 / JST ERATO

自己紹介

- **湊　真一（みなと しんいち）**

  北海道大学 大学院 情報科学研究科 教授

  JST　ERATO湊離散構造処理系プロジェクト　研究総括

  - 石川県出身　金沢大学附属高校卒
  - 1990年　京都大学 大学院 修士課程 修了
    （1995年　博士(工学)）
  - 1990年～2004年　NTT研究所　研究員
  - 1997年　米国スタンフォード大学　客員研究員(1年間)
  - 2004年～　北海道大学 情報科学研究科 助教授
  - 2010年～　同 教授

（北大に着任して10年目）

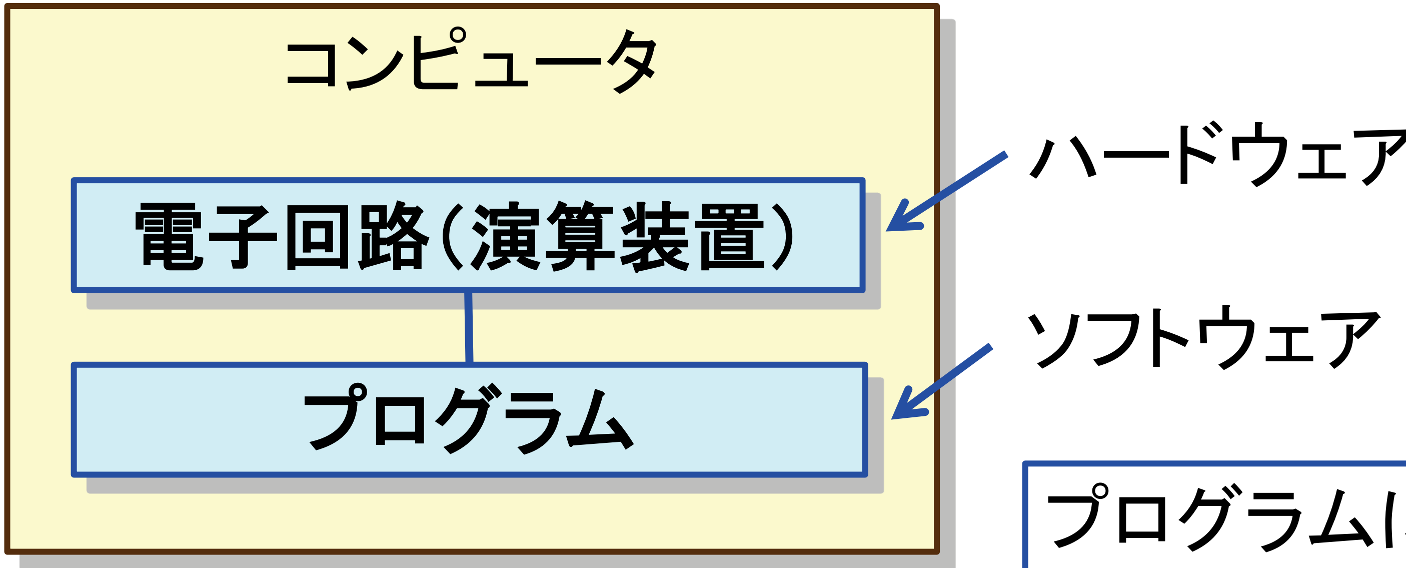
本講演の内容

- アルゴリズム技術とは
  - 組合せ爆発のすごさとアルゴリズム技術の重要性
- ERATO湊離散構造処理系プロジェクト
  - 論理と集合を表現するBDD/ZDD技術
  - 圧縮・列挙・索引化・演算処理
- 日本科学未来館「フカシギの数え方」
  - 「おねえさんの問題」とその計算法
  - 実社会の様々な問題への応用

本講演の内容

- アルゴリズム技術とは
  - 組合せ爆発のすごさとアルゴリズム技術の重要性
- ERATO湊離散構造処理系プロジェクト
  - 論理と集合を表現するBDD/ZDD技術
  - 圧縮・列挙・索引化・演算処理
- 日本科学未来館「フカシギの数え方」
  - 「おねえさんの問題」とその計算法
  - 実社会の様々な問題への応用

アルゴリズム技術とは



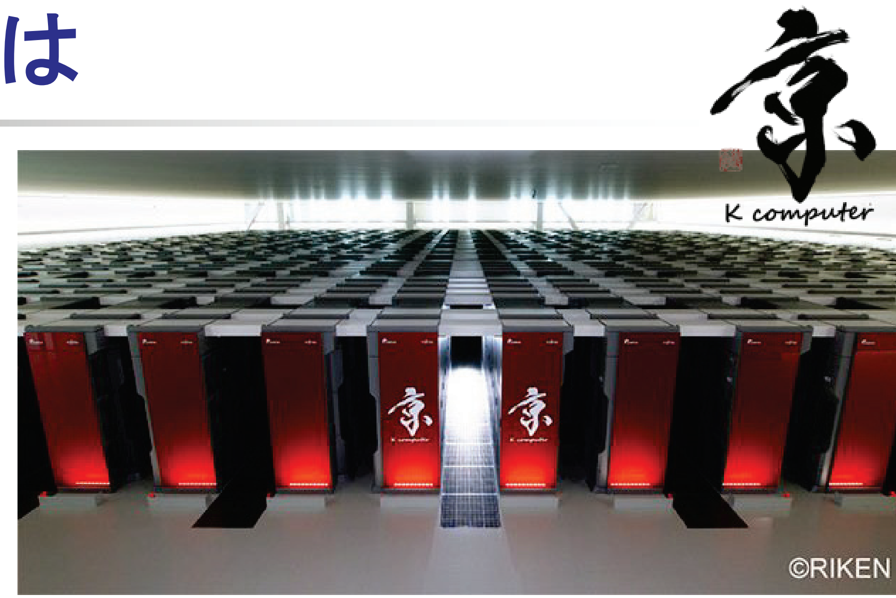
プログラムに書かれた計算手順・戦略のことを**「アルゴリズム」**という。

- コンピュータで様々な計算を行う際に、アルゴリズムの工夫によって、何十倍、何百倍も計算時間が違ってくることがある。

計算時間の影響

- 速いか遅いかの違いだけと軽く考えてはいけません。
  → とても重要！
  - 3秒で終わるなら対話的に仕事ができるが、3分かかるとなると、ずっと座って待ってはいられない。
    （web検索に1回3分かかったら今のインターネットはない）
  - 新作ゲームの設計がクリスマスに間に合うかどうかで、メーカーの売り上げが大きく影響を受ける。
    企業の存続/倒産が左右されることも！
    （従業員や家族の人生にまで影響が及ぶ）
  - ある計算に100年かかるとしたら、その人にとっては解けないのと同じ （人の寿命は有限）

計算時間を短縮するには



- スーパーコンピュータを使う？
  - 神戸に建設されたスパコン「京」は64万個以上の演算装置を並べて毎秒１京回（兆の１万倍）計算できる。
- ただし誰もが使えるものではない
  - 開発費：約1000億円、　運用費：約80億円/年（うち電気代20億円/年）→ 国家的/世界的に重要な問題を解くために使われる。
- **アルゴリズム技術**
  - 与えられた問題を解くために、できるだけ少ない基本演算回数で済まそうとする工夫
  - 同じコンピュータでも、何百倍も速くなることがある。
  - または、もっと安価で低消費電力なコンピュータで済む。

計算時間がかかる問題とは

- 超巨大なデータを扱う計算
  - 全国のコンビニの購買データは、年に100万件以上
  - ヒトゲノム（人間の遺伝情報）は、3億文字以上の長さ
  - インターネットのwebページの総数は、（推定）1兆ページ以上
- データはそれほど巨大でないが、
  膨大な数の場合分けが必要な計算
  - 囲碁・将棋の勝ち負けパタンの分類
  - 数独のヒントの配置
  - カーナビの最短経路の探索

身近な問題

- クーポン券が10枚あります。
  1000円の商品を買いたいのですが、おつりは出ません。
  ぴったりの組合せはあるでしょうか？

130　230　280　390　250　80　50　440　190　310

身近な問題

- クーポン券が10枚あります。
  1000円の商品を買いたいのですが、おつりは出ません。
  ぴったりの組合せはあるでしょうか？

130　230　280　390

250　80

50　440　190　310

身近な問題

- クーポン券が10枚あります。
  1000円の商品を買いたいのですが、おつりは出ません。
  ぴったりの組合せはあるでしょうか？

130　230　280　390

250　80

50　440　190　310

身近な問題

- クーポン券が10枚あります。
  1000円の商品を買いたいのですが、おつりは出ません。
  ぴったりの組合せはあるでしょうか？

130　230　280　390

250　80

50　440　190　310

身近な問題

- クーポン券が10枚あります。
  1000円の商品を買いたいのですが、おつりは出ません。
  ぴったりの組合せはあるでしょうか？

130 230 280 390

250 80

50 440 190 310

身近な問題

- クーポン券が10枚あります。
  1000円の商品を買いたいのですが、おつりは出ません。
  ぴったりの組合せはあるでしょうか？

130　230　280　390

80

250　440　190

50　310

身近な問題

- クーポン券が10枚あります。
1000円の商品を買いたいのですが、おつりは出ません。
ぴったりの組合せはあるでしょうか？
- **答えは全部で10通りあります。**

130　230　280　390　250　80　50　440　190　310

身近な問題

- クーポン券が10枚あります。
1000円の商品を買いたいのですが、おつりは出ません。
ぴったりの組合せはあるでしょうか？
- **では、980円の組合せは作れるでしょうか？**

130	230	280	390	250
80	50	440	190	310

身近な問題

- クーポン券が10枚あります。
  1000円の商品を買いたいのですが、おつりは出ません。
  ぴったりの組合せはあるでしょうか？
- **では、980円の組合せは作れるでしょうか？　Yes！**

130　230　280　390

250　80

50　440　190　310

身近な問題

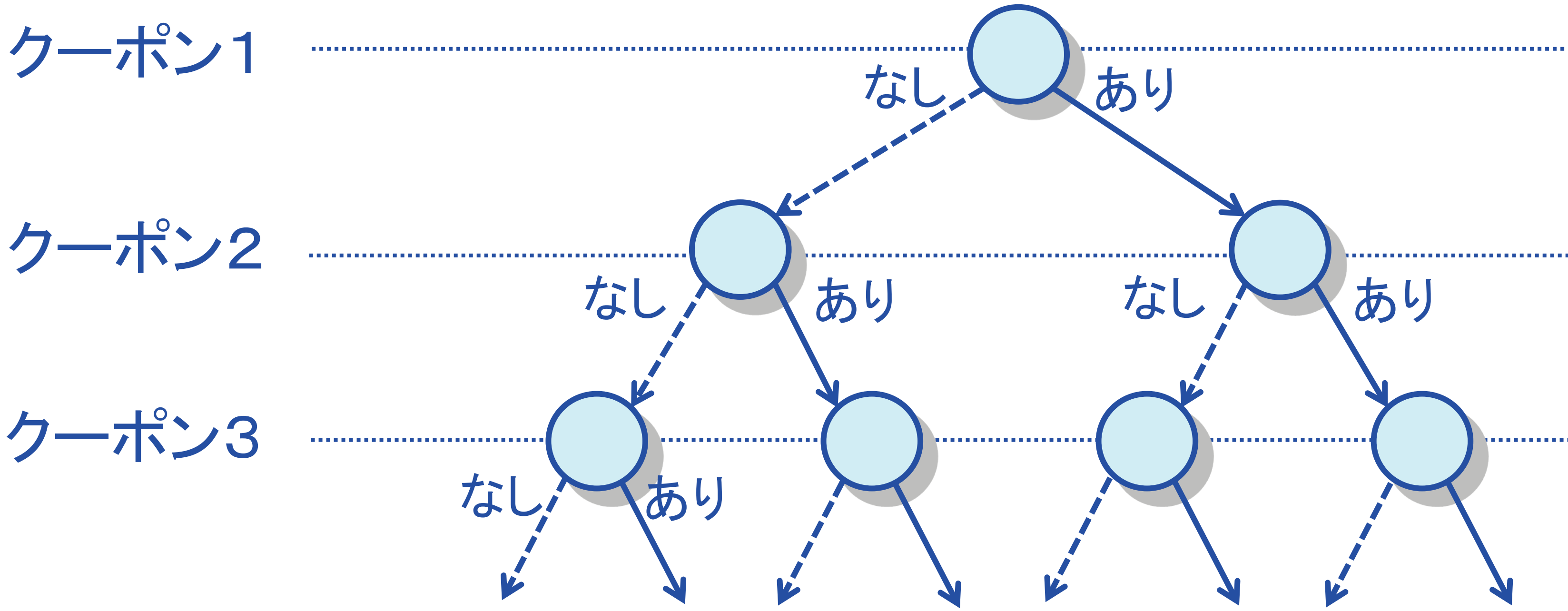
- クーポン券が10枚あります。
  1000円の商品を買いたいのですが、おつりは出ません。
  ぴったりの組合せはあるでしょうか？
- **そもそも組合せ方は全部で何通り？**

130　230　280　390

250　80

50　440　190　310

二分木の考え方



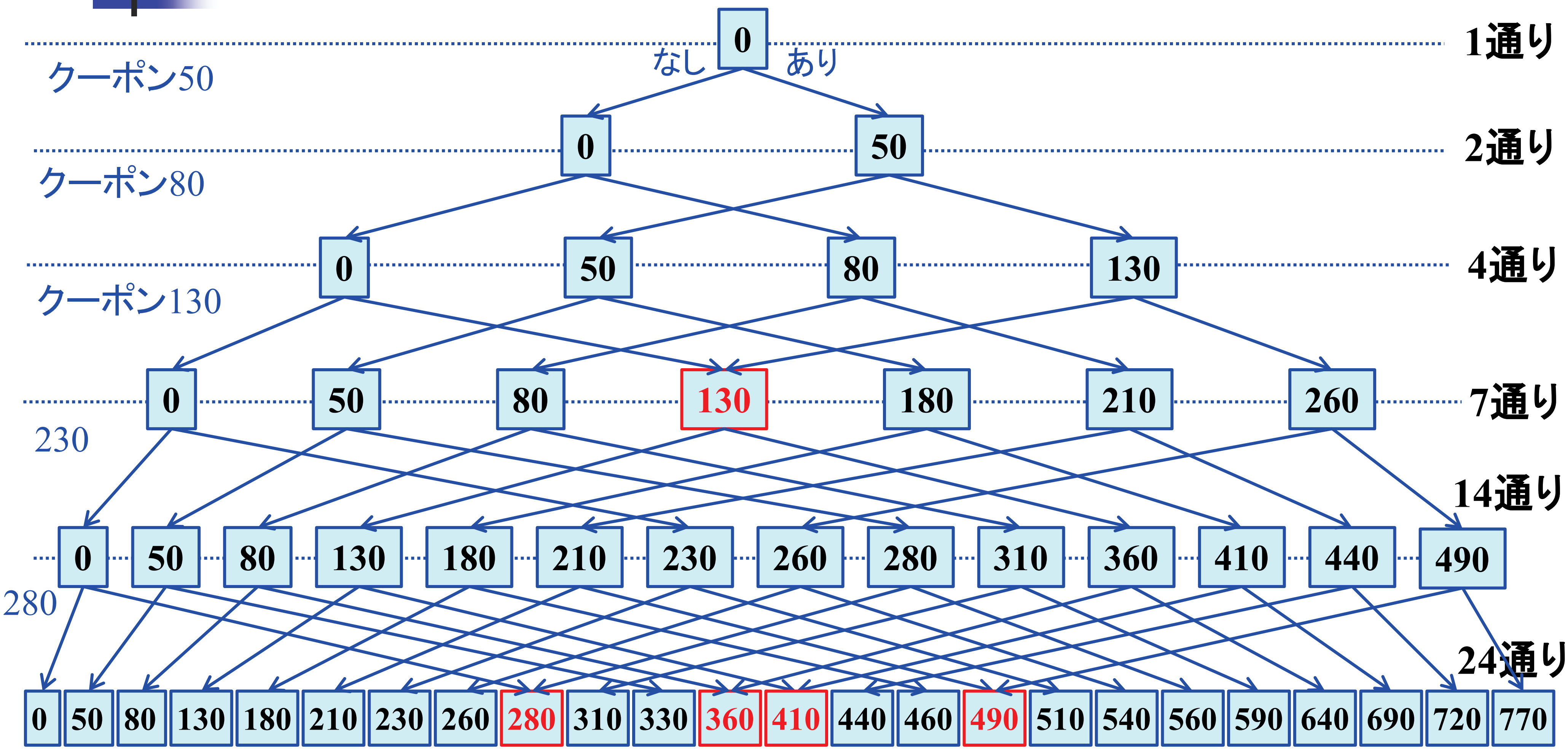
- クーポンが1つ増えるたびに組合せの総数は2倍になる。
  - クーポンの枚数を n とすると、組合せの総数は 2^n 通り
  - n=10 のときは全部で1024通り
（ちょうど1000円になるのは、その中の10通りだけ）

2のn乗の脅威（→指数爆発）

n	2^n
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1,024

n	2^n
10	1,024
20	1,048,576
30	1,073,741,824
40	1,099,511,627,776
50	1,125,899,906,842,624
60	1,152,921,504,606,846,976
70	1,180,591,620,717,411,303,424
80	1,208,925,819,614,629,174,706,176
90	1,237,940,039,285,380,274,899,124,224
100	1,267,650,600,228,229,401,496,703,205,376

アルゴリズムの工夫



最後まで行っても198通りしか出て来ない　（→ 1024通り調べなくても済む！）

アルゴリズム技術の重要性

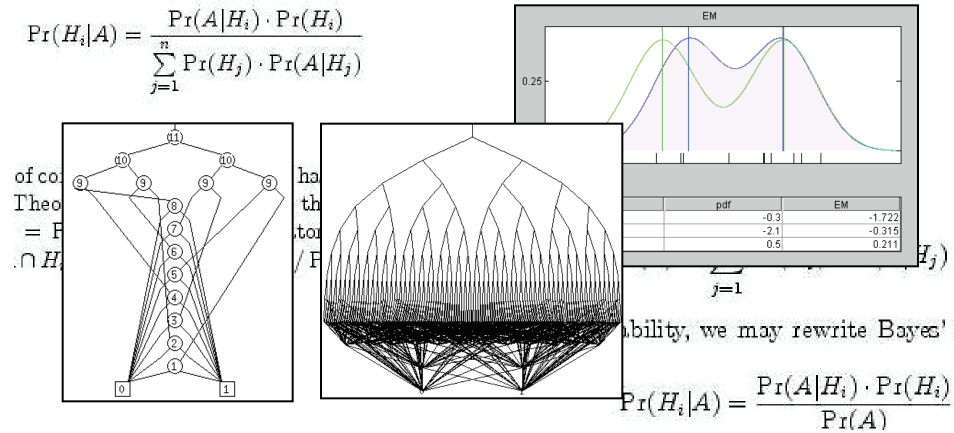
- 今回の例では、1024通りが198通りに減るので、
  アルゴリズムの工夫で約5倍の高速化
  → 問題が大規模になるほど高速化の倍率が大きくなる。
  （クーポンの枚数が20枚になると、効果が数十倍、数百倍になる例も）
- スーパーコンピュータ開発のような、コンピュータそのものの高速化は、問題の規模に関係なく一定の倍数だけ速くなる。
- アルゴリズムの改良は、問題の規模が大きくなるほど高速化の倍率が高くなることが多い。→劇的な効果
  - 特別な装置を使わないので費用もかからない。
  - ただし、どんな問題でも高速化できるとは限らない。
    実用の場面での最適なアルゴリズムの設計は簡単ではない。

本講演の内容

- アルゴリズム技術とは
  - 組合せ爆発のすごさとアルゴリズム技術の重要性
- ERATO湊離散構造処理系プロジェクト
  - 論理と集合を表現するBDD/ZDD技術
  - 圧縮・列挙・索引化・演算処理
- 日本科学未来館「フカシギの数え方」
  - 「おねえさんの問題」とその計算法
  - 実社会の様々な問題への応用

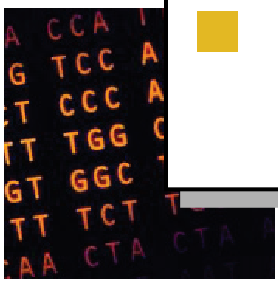
背景：離散構造とその処理

- 離散構造とは
  - **離散数学および計算機科学の基礎をなす数学的構造**
  - 集合理論、記号論理、帰納的証明、グラフ理論、組合せ論、確率論などを含む



- およそ計算機が扱うあらゆる問題は、単純な基本演算を要素とする離散構造の処理に帰着される。
  - 最終的には、**膨大な数の場合分け処理**を要することが多い。
- 離散構造の処理は、計算機の様々な応用分野に共通する基盤技術
  - 典型的な応用分野：　システム設計自動化、大規模システム故障解析、制約充足問題、データマイニングと知識発見、機械学習と自動分類、生命情報科学、web情報解析、etc.
  - **現代情報化社会に対する波及効果は極めて大。**

本プロジェクトの基本構想



- 様々な工学的応用を持つ基盤技術として「離散構造処理系」に着目し、研究開発を行う

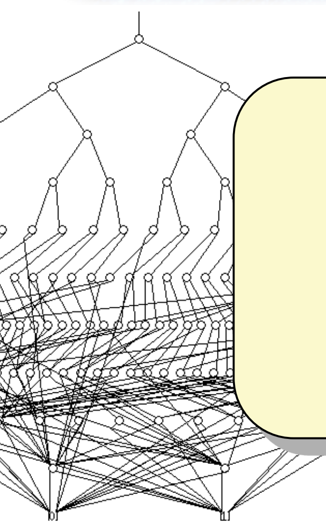
システム設計自動化
データマイニングと知識発見
大規模システム故障解析
機械学習と自動分類
制約充足問題
生命情報科学
web情報解析

工学的応用
→ 社会への影響大

本研究の対象領域

離散構造処理系

性能向上
(10倍～100倍以上)

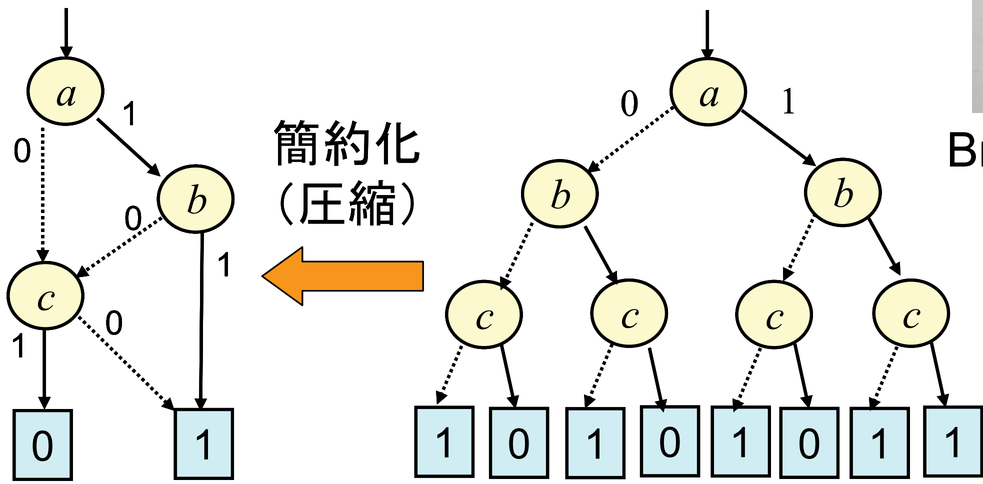


集合理論
記号論理
帰納的証明
組合せ論
グラフ理論
確率論

数学的概念構造

BDD（二分決定グラフ）

離散構造の最も基本的なモデルである**「論理関数」**の処理技法

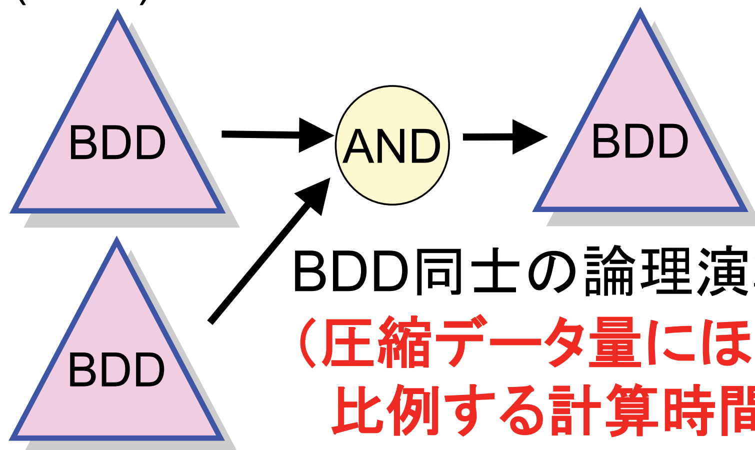


（BDD）　　（場合分け二分木）

- 場合分け二分木グラフを簡約化（データ圧縮）
- **多くの実用的な論理データをコンパクトかつ一意に表現。**（数十～数百倍以上の圧縮率が得られる例も）

Bryant (CMU)

1986年に画期的なBDD演算**（Apply演算アルゴリズム）**を提案。以後急速にBDD技術が発達。（長期間、情報科学の全分野での最多引用文献となった）

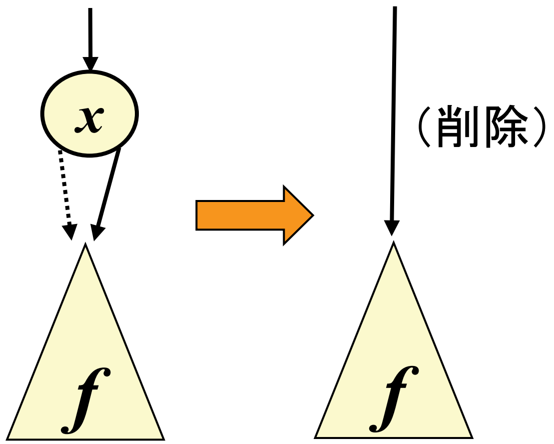


BDD同士の論理演算
（圧縮データ量にほぼ比例する計算時間）

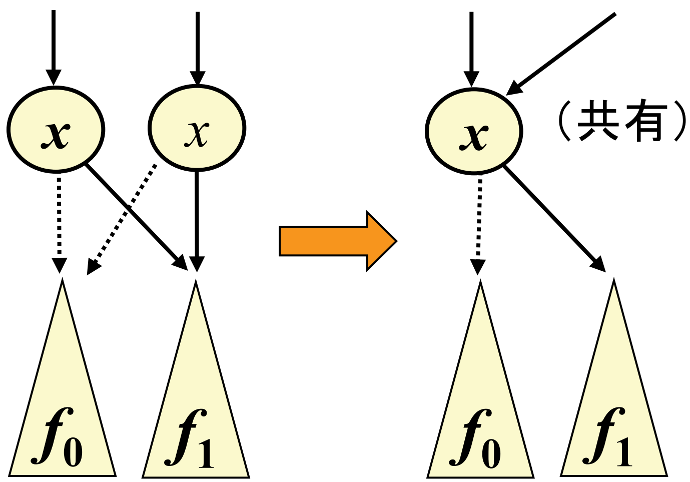
近年のPC主記憶の大規模化により、BDDの適用範囲が拡大（特に2000年以降）

BDDの簡約化規則

(a)



(b)

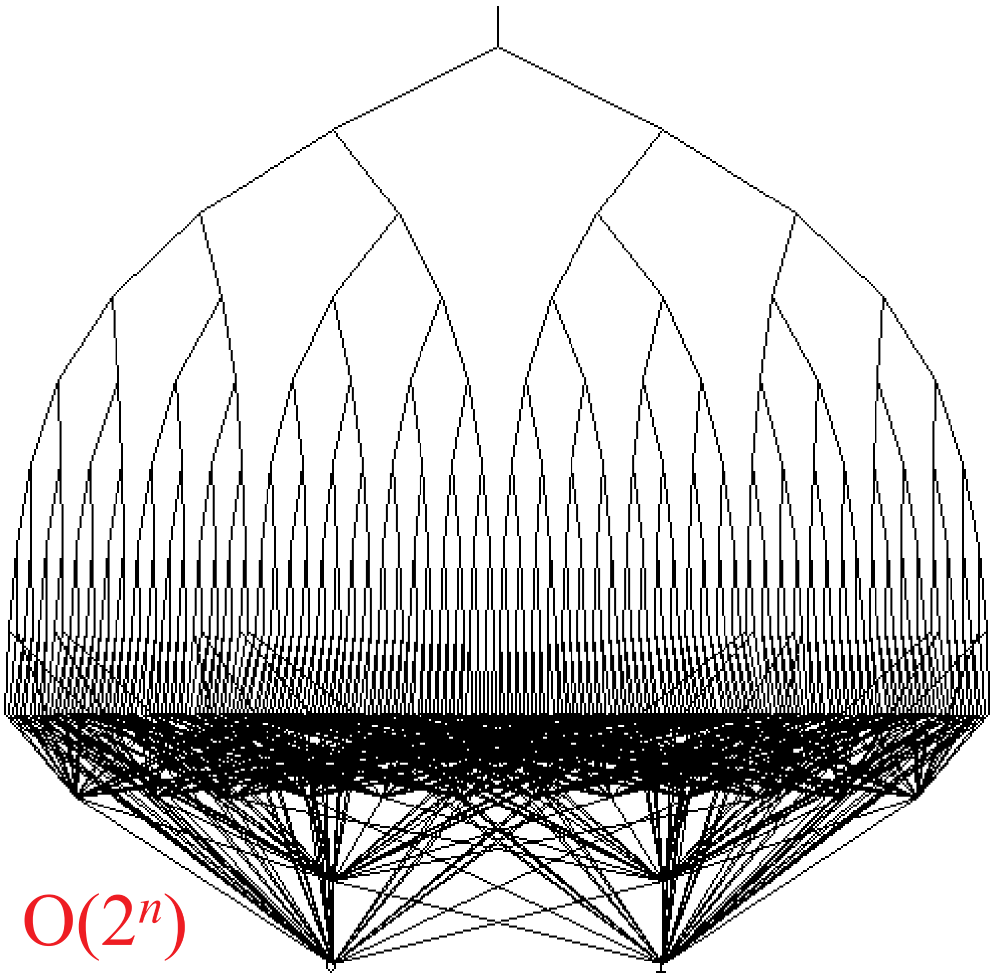


(a) 冗長な節点を全て削除
(b) 等価な節点を全て共有

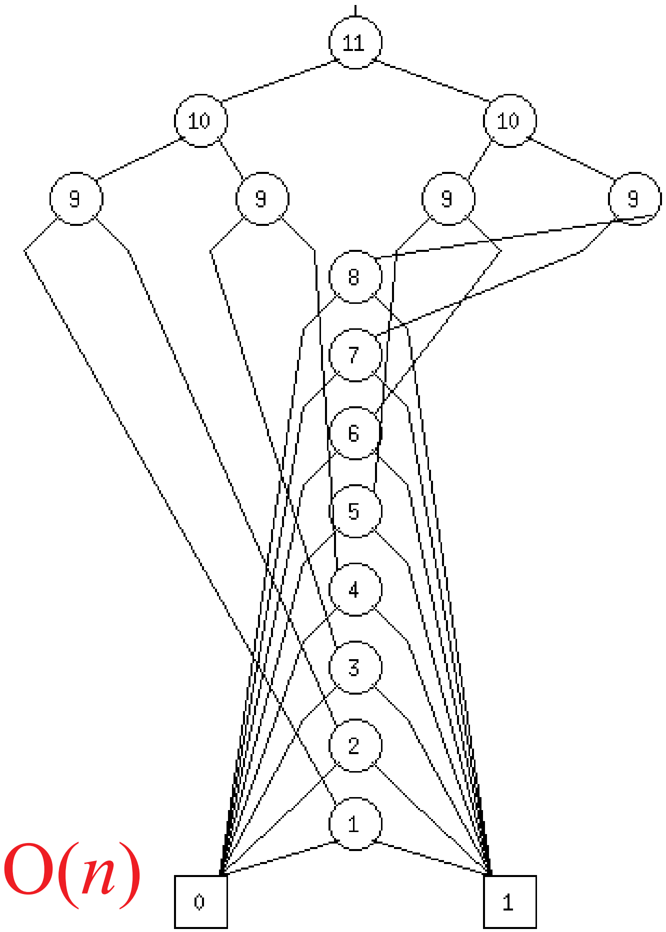
既約なBDDが得られる

BDD簡約化の効果

- 特定の問題では、指数関数的な圧縮効果が得られる。
  - 例題に依存するが、多くの実用的な問題で効果がある。



$O(2^n)$



$O(n)$

論理関数と組合せ集合

a	b	c	F	
0	0	0	0	
1	0	0	0	
0	1	0	0	
1	1	0	1	→ ab
0	0	1	1	→ c
1	0	1	1	→ ac
0	1	1	0	
1	1	1	0	

論理関数:

$$F = (a\ b \sim c) \vee (\sim b\ c)$$

組合せ集合:

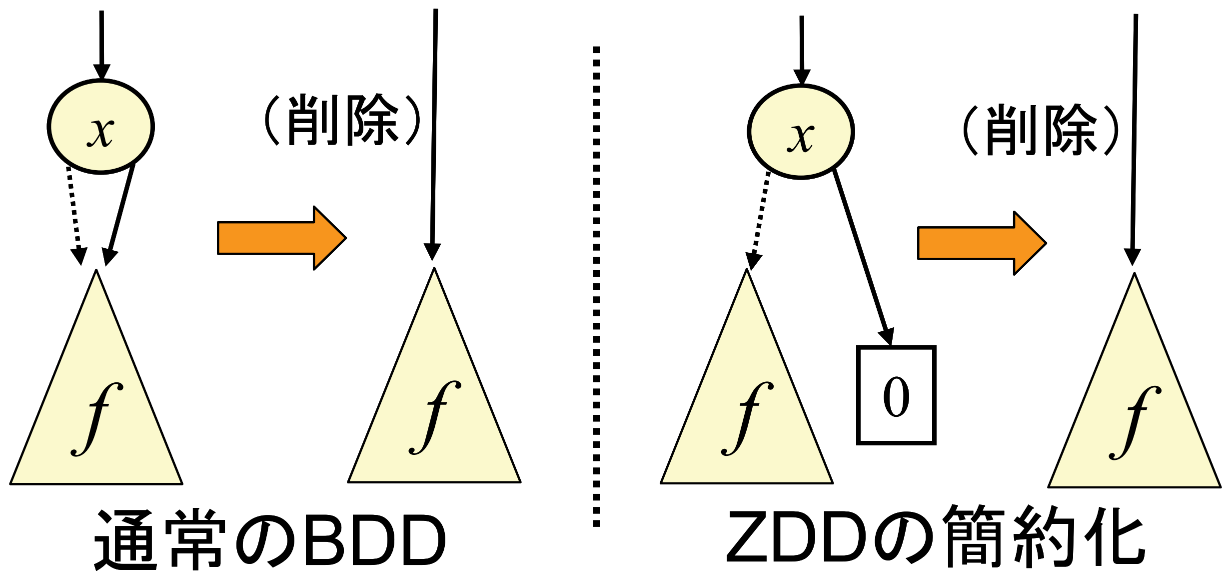
$$F = \{ab, ac, c\}$$

(買い物客の購入品)

- 組合せ集合と論理関数の演算は対応関係がある。
  - Union of sets → logical OR
  - Intersection of sets → logical AND
  - Complement set → logical NOT

ZDD(ゼロサプレス型BDD)による集合の表現

- 「組合せ集合」を効率的に表現するためのBDDの改良
  - 湊が世界で初めて考案し命名(1993年)
  - 通常と異なる簡約化規則を考案。
  - 疎な集合の族を扱う場合に著しい効果が得られる。
    (例: 商店の陳列アイテム数に比べて1顧客の購入点数は極めて少ない。)



- ZDDはBDDの改良技術として現在、世界的に広く使われている。
  - 最近では、データマイニング分野に応用されて、画期的な有効性が示されている。(数百倍のデータ圧縮率・数十倍の処理高速化)
  - 他にも応用例は増えつつある。

ZDDの応用

- 元々はLSI設計の論理式(CNF/DNF)の簡単化に利用
  - 膨大な項数の論理式を高速に因数分解する方法[Minato96]
  - 算術式の因数分解法[Minato97]
- データマイニングへの応用
  - ZDDを用いた頻出パタンマイニング[Minato2005]
  - LCM over ZDDs アルゴリズム[Minato-Uno2008]
  - 時分割データベースからのパタン変化の検出[Minato2010]
- グラフに関する種々の問題への応用
  - 最大クリーク問題、彩色問題、カバー問題等[Coudert97]
  - ZDDを用いた超高速パス列挙アルゴリズム[Knuth2009]
  - 電力ネットワークの制御、避難所配置問題、通信NW、他

ZDD応用例：頻出アイテム集合抽出問題

- データマイニングの最も基本的な問題
  - 最小出現頻度α以上のレコードに含まれるアイテム組合せの部分集合を抽出・列挙する問題

レコード番号	アイテム集合
1	a b c
2	a b
3	a b c
4	b c
5	a b
6	a b c
7	c
8	a b c
9	a b c
10	a b
11	b c

最小頻度 α = 10 → { b }

最小頻度 α = 8 → { ab, a, b, c }

最小頻度 α = 7 → { ab, bc, a, b, c }

最小頻度 α = 5 → { abc, ab, bc, ac, a, b, c }

最小頻度 α = 1 → { abc, ab, bc, ac, a, b, c }

LCM-ZDD 法による高速化

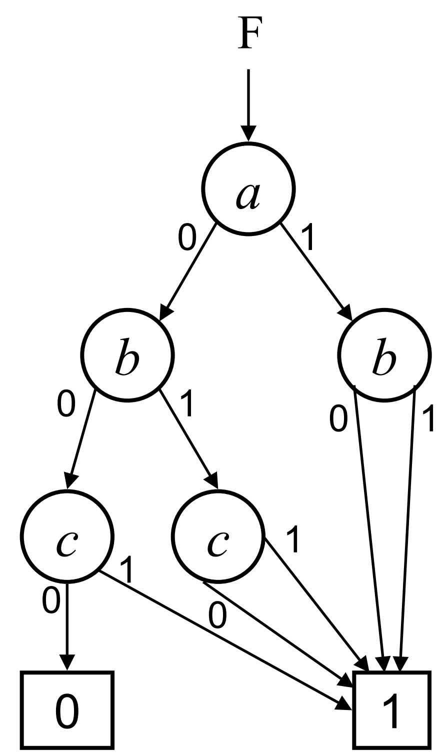
- 計算結果の頻出アイテム集合を、メモリ上に圧縮してZDDで表現し、そのポインタのみを返す。
  - 計算結果をファイルに出力しない。

レコード番号	アイテム集合
1	$a\ b\ c$
2	$a\ b$
3	$a\ b\ c$
4	$b\ c$
5	$a\ b$
6	$a\ b\ c$
7	c
8	$a\ b\ c$
9	$a\ b\ c$
10	$a\ b$
11	$b\ c$

LCM-ZDD法

(最小頻度 $\alpha = 7$)

$\{ ab, bc, a, b, c \}$



solutions

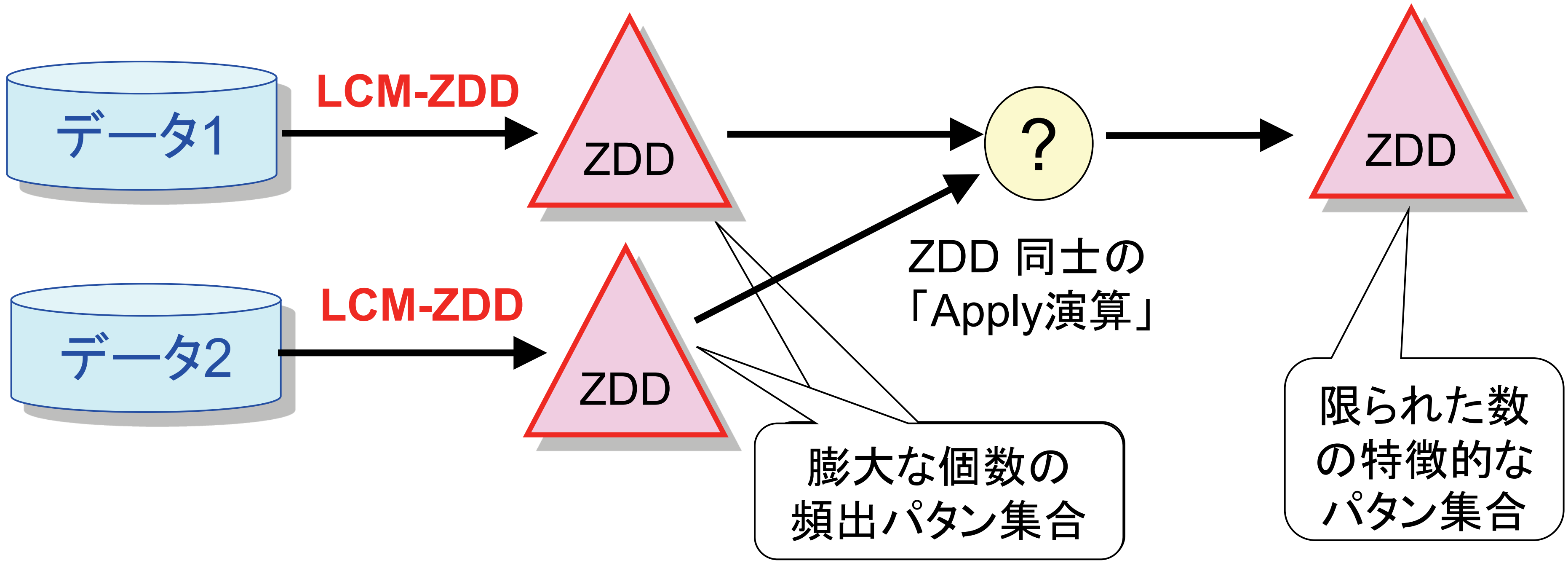
LCM-ZDD

Original LCM

Table 2. Co CM o ith the pre

Dataset name: min. support	#Frequent itemsets	LCMoverZBDDs |ZBDD|	LCMoverZBDDs Time(s)	LCM-count Time(s)	LCM-dump Time(s)	ZBDD-growth Time(s)
mushroom: 1,000	123,287	760	0.50	0.49	0.64	1.78
500	1,442,504	2,254	1.32	1.30	3.29	3.49
300	5,259,786	4,412	2.25	2.22	9.96	5.11
200	18,094,822	6,383	3.21	3.13	31.63	6.24
100	66,076,586	11,584	5.06	4.87	114.21	6.72
70	153,336,056	14,307	7.16	7.08	277.15	6.97
50	198,169,866	17,830	8.17	7.86	357.27	6.39
T10I4D100K: 100	27,533	8,482	0.85	0.85	0.86	209.82
50	53,386	16,872	0.97	0.92	0.98	242.31
20	129,876	58,413	1.13	1.08	1.20	290.78
10	411,366	173,422	1.55	1.36	1.64	332.22
5	1,923,260	628,491	2.86	2.08	3.54	370.54
3	6,169,854	1,576,184	5.20	3.15	8.14	386.72
2	19,561,715	3,270,977	9.68	5.09	22.66	384.60
BMS-WebView-1: 50	8,192	3,415	0.11	0.11	0.12	29.46
40	48,544	10,755	0.18	0.18	0.22	48.54
36	461,522	28,964	0.49	0.42	0.98	67.16
35	1,177,608	38,164	0.80	0.69	2.24	73.64
34	4,849,466	49,377	1.30	1.07	8.58	83.36
33	69,417,074	59,119	3.53	3.13	144.98	91.62
32	1,531,980,298	71,574	31.90	29.73	3,843.06	92.47
chess: 1,000	29,442,849	53,338	197.58	197.10	248.18	1,500.78
connect: 40,000	23,981,184	3,067	5.42	5.40	49.21	212.84
pumsb: 32,000	7,733,322	5,443	60.65	60.42	75.29	4,189.09
BMS-WebView-2: 5	26,946,004	353,091	4.84	3.62	51.28	118.01

ZDDを用いたパタン集合演算

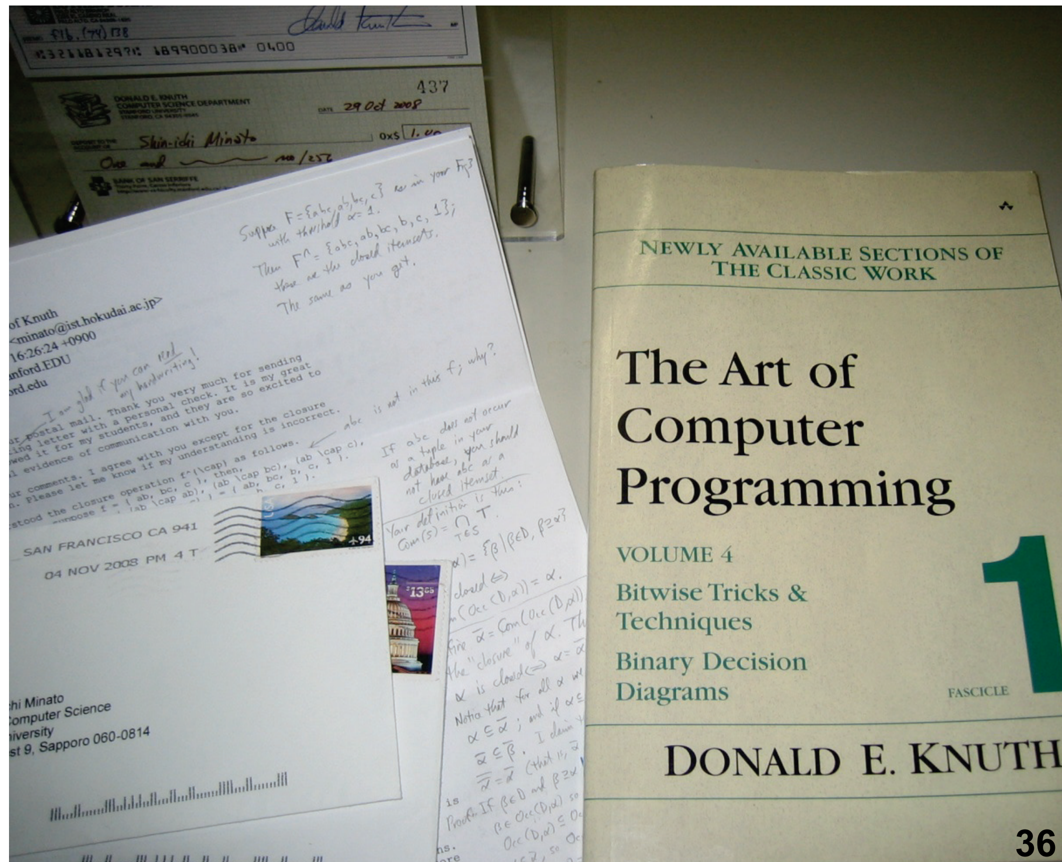


- 数十億ものパタンを含む集合を圧縮して表現し、ZDDの集合演算を使って効率よく絞込みを行える。
  - 従来の明示的な表現方法では，意味のある解析処理を現実的な時間で行うことは不可能

KnuthVの名著とBDD/ZDD

- Knuthの世界的名著「The Art of Computer Programming」の最新巻（Vol.4, Fascicle 1, 2009）で、BDDが取り上げられ、その中でZDDが30ページ以上に渡り詳しく解説された。

- **日本人の研究成果が、このシリーズに項目として詳細に掲載されるのは初めて。**
- Knuth氏本人から、ZDD考案者として校正作業への協力を依頼する長文のメールと手紙を受領。
- 2010年5月には湊がKnuth邸を訪問し、プロジェクトの方向性について意見交換を行った。



本研究プロジェクトの対象領域

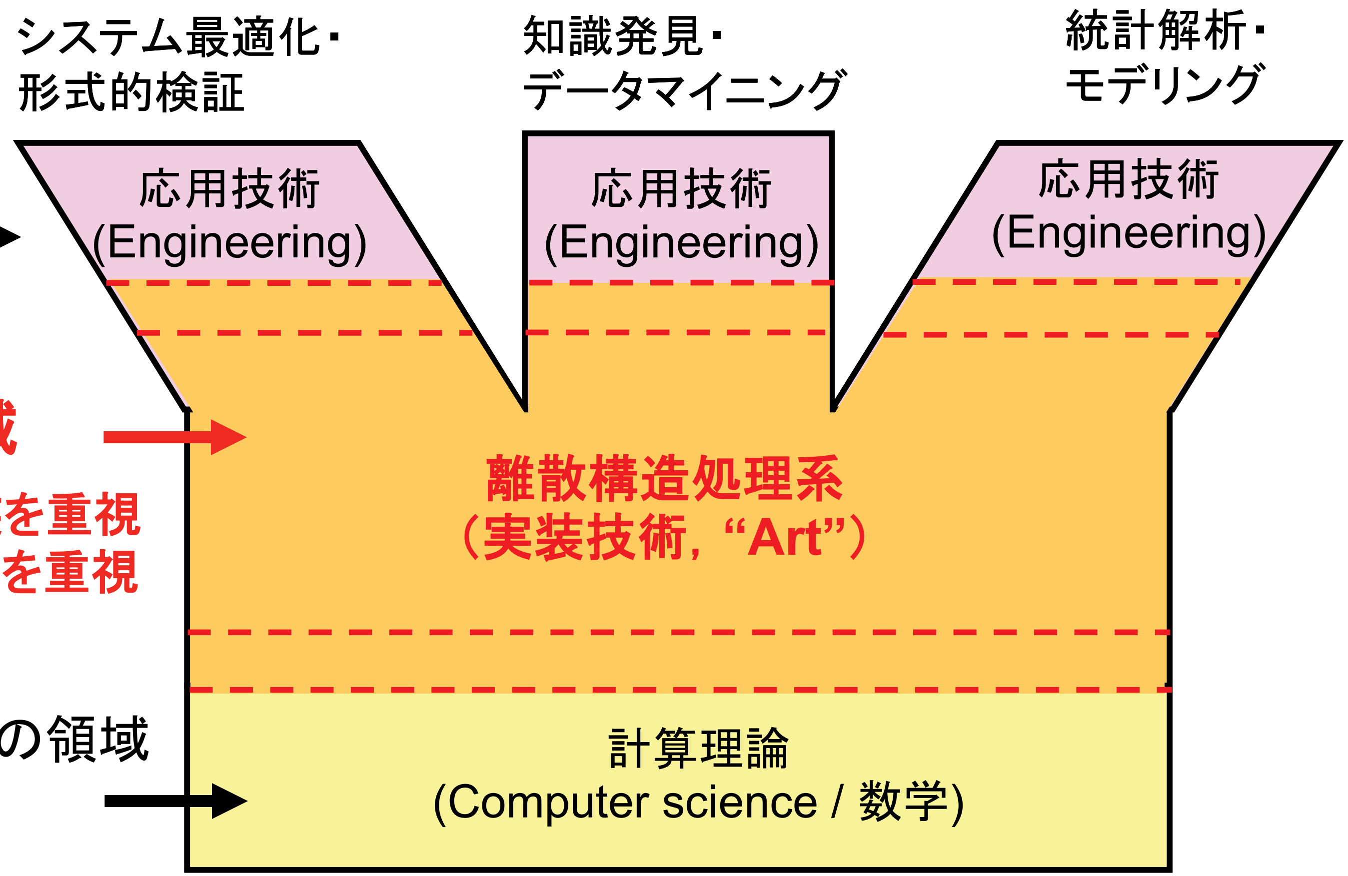
BDD/ZDD技術の新しい切り口として、様々な離散構造を**統合的に演算処理**する技法を体系化し、**分野横断的かつ大規模な**実問題を高速に処理するための技術基盤を構築する。

個別の工学的応用に
特化した技術領域

本研究で扱う技術領域

- 概念・理論だけでなく処理系実装を重視
- 技術基盤としての簡潔さ・汎用性を重視

分野横断的な計算理論の領域
（概念的・理論的）



研究実施場所とメンバ

- 札幌メインオフィス（北大・情報科学研究科）
  - ポスドク研究員6名
  - 事務スタッフ3名、技術員2名、RA3名
  - 北大研究者（GCOE他）と連携

- 東京サテラボ（東工大）
  - GL：産総研・津田宏治氏
  - ポスドク研究員2名、RA1名
  - 東工大GCOE・東京地区研究者と連携

- 大阪サテラボ（梅田駅北口）
  - GL：阪大・鷲尾隆 教授
  - ポスドク1名、
    技術員1名、RA5名
  - 関西地区研究者と連携

各地区は高品質TV会議システムで常時接続

フルタイムの若手研究者は極力、札幌に常駐させて、活性化・相乗効果を図る。

ERATO

北海道大学@札幌

本講演の内容

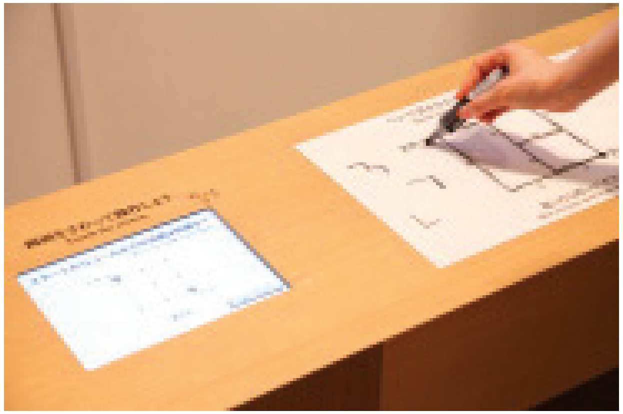
- アルゴリズム技術とは
  - 組合せ爆発のすごさとアルゴリズム技術の重要性
- ERATO湊離散構造処理系プロジェクト
  - 論理と集合を表現するBDD/ZDD技術
  - 圧縮・列挙・索引化・演算処理
- 日本科学未来館「フカシギの数え方」
  - 「おねえさんの問題」とその計算法
  - 実社会の様々な問題への応用

日本科学未来館での成果展示

- 日本科学未来館（東京・お台場）で我々の研究プロジェクトの展示**「フカシギの数え方」**を開催
  - 2012年8月1日～2013年2月25日（期間中、夏休み冬休み1回ずつ）
  - 小中高生・一般市民向けに研究内容をわかりやすく展示
- 好評につき会期延長(~4/25まで)
  - 期間中に23万人が来場
  - 6月以降、北大で再展示

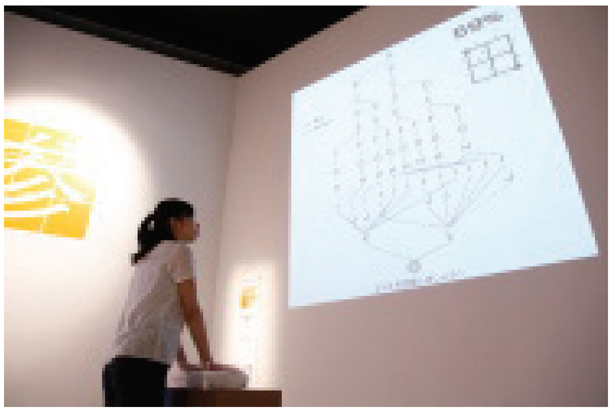
Miraikan

展示の工夫



インタラクティブ展示
（ペンでなぞってみる）

巨大な数表
（視野の広さで実感）



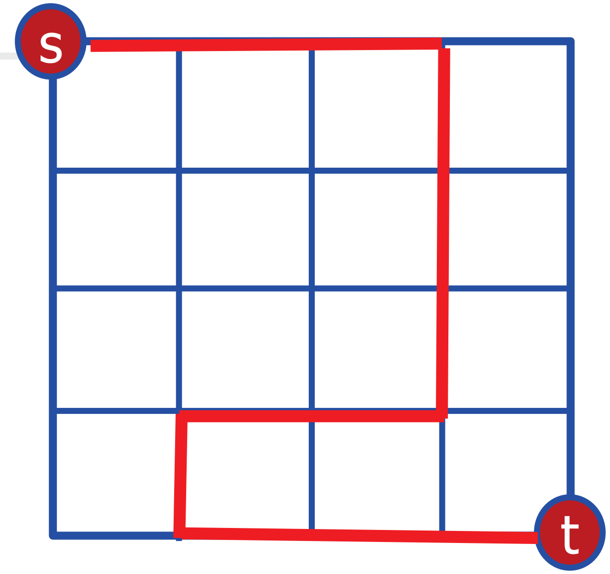
体感展示
（体重をかけて圧縮）

顔写真と説明文
（研究者が語りかける）

アニメーション動画の反響

- 「フカシギの数え方」で、本プロジェクトの企画・監修によるアニメーション動画を製作
- **YouTube再生回数が138万回以上**（ニコ動でも40万回以上）
  - サイエンス系コンテンツでは極めて異例の大ヒットに

「おねえさんの問題」の豆知識



- 最短経路の数え上げは簡単
(→ ${}_{2n}C_n$; 高校で習う問題)
- 遠回りを許すと突然難しくなる。
  - 簡単な計算式は見つかっていない。
  - Knuthの教科書のZDDの章の演習問題になっている
→ ムダなく数え上げるアルゴリズムが示されている
  - アニメ動画で、おねえさんが25万年かかった計算結果の数字が出ているため、「公式を教えて」というコメント多数。
→「どうも公式ないらしいよ」
→「ZDDというのを使うらしいよ」

この問題の数え上げ世界記録

- 本プロジェクトの研究チームが**21×21の数え上げに成功**
  - Knuthのアルゴリズムを、メモリ効率が良くなるようにさらに改善
  - 18×18まではすべての解を索引化したZDDを生成できる
  - 19以降は、ZDDは輪切りにした横一列だけ生成しながら、解の個数だけをカウント。
  - さらに細かい工夫を加えて、数え上げに成功
- Integer Sequences（数列のオンライン百科事典サイト）に登録（2012年9月19日）
  - これまでは19×19までだった。一気に2段階更新。
  - 登録しようとしたら、9/14の時点でInteger Sequenceに「おねえさん動画」がリンクされていた。（YouTubeを見たフィンランドの人が提案して認められたらしい）

This site is supported by donations to The OEIS Foundation.

Integer Sequences

Search Hints

(Greetings from The On-Line Encyclopedia of Integer Sequences!)

A007764 Number of nonintersecting (or self-avoiding) rook paths joining opposite corners of an n X n grid. 13

1, 2, 12, 184, 8512, 1262816, 575780564, 789360053252, 3266598486981642, 41044208702632496804, 1568758030464750013214100, 182413291514248049241470885236, 64528039343270018963357185158482118 (list; graph; refs; listen; history; text; internal format)

OFFSET 0,2

COMMENTS
The length of the path varies.
Daniel Forgues, Jan 03 2011 (Start) For n=14, there exists at least one Hamiltonian path from (0,0) to (14,14). For which n do we have at least one Hamiltonian path?
Lattice graphs have their values located at the corners of grid cells. Lattice graph edges join the corners of grid cells.
Grid graphs have their values located at the centers of grid cells. Grid graph edges join the centers of grid cells.
An (m+1) X (n+1) square lattice constitutes the cell corners (coordinates (0,0) to (m,n)) of an m X n square grid.
Number of self-avoiding walks from (0,0) to (n,n), n >= 0, of a (n+1) X (n+1) square lattice.
Since rooks move from centers to centers of adjacent grid cells, should the definition say? "Number of nonintersecting (or self-avoiding) rook paths joining opposite corner cells of an (n+1) X (n+1) grid." (End)

REFERENCES
S. R. Finch, Mathematical Constants, Cambridge, 2003, pp. 331-339.
D. E. Knuth, 'Things A Computer Scientist Rarely Talks About,' CSLI Publications, Stanford, CA, 2001, pages 27-28.
Netnews group rec.puzzles, Frequently Asked Questions (FAQ) file. (Science Section).
D. E. Knuth, The Art of Computer Programming, Section 7.1.4.

LINKS
I. Jensen and H. Iwashita, Table of n, a(n) for n = 0..21 (I. Jensen computed terms 0 to 19)
H. Iwashita, J. Kawahara, and S. Minato, ZDD-Based Computation of the Number of Paths in a Graph
Doi, Maeda, Nagatomo, Niiyama, Sanson, Suzuki, et al, Time with class! Let's count! [Youtube-animation demonstrating this sequence. In Japanese with English translation]
M. Bousquet-Melou, A. J. Guttmann and I. Jensen, Self-avoiding walks crossing a square
S. R. Finch, Self-Avoiding-Walk Connective Constants
I. Jensen, Series Expansions for Self-Avoiding Walks
OEIS Wiki, Self-avoiding walks
Eric Weisstein's World of Mathematics, Self-Avoiding Walk

0	1
1	2
2	12
3	184
4	8512
5	1262816
6	575780564
7	789360053252
8	3266598486981642
9	41044208702632496804
10	1568758030464750013214100
11	182413291514248049241470885236
12	64528039343270018963357185158482118
13	69450664761521361664274701548907358996488
14	227449714676812739631826459327989863387613323440
15	2266745568862672746374567396713098934866324885408319028
16	68745445609149931587631563132489232824587945968099457285419306
17	6344814611237963971310297540795524400449443986866480693646369387855336
18	1782112840842065129893384946652325275167838065704767655931452474605826692782532
19	1523344971704879993080742810319229690899454255323294555776029866737355060592877569255844
20	3962892199823037560207299517133362502106339705739463771515237113377010682364035706704472064940398
21	31374751050137102720420538137382214513103312193698723653061351991346433379389385793965576992246021316463868

これまでの
記録:19×19

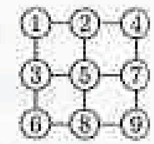
新記録:21×21
（107桁）

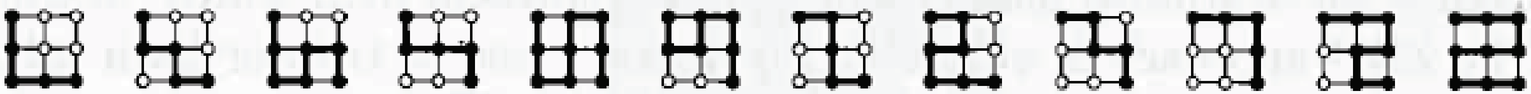
世界記録のその後の展開

- 「サ、ツギハ 22×22ネ」
  - グラフの形をおねえさん問題に限定すれば、まだまだ行けそう
    → 特別チームを作ってアルゴリズム開発を続行
- 正月休みに一般市民プログラマからメールが届く
  - 「高速な解き方を思いついたので見てもらえませんか？」
  - 休日に何度か北大に来訪。プロも驚くアイデアを提案
  - 「アマチュアプログラマ」の肩書で共著者に入ってもらう
- 2013年2月：ノルウェーの大学チームが
  一気にn=24まで記録更新していた。
- 2013年4月：本プロジェクトの総力を結集して
  n=25までの計算に成功。世界記録を奪還。
  - 主記憶1.5TBの計算機を2～3日占有して計算。
  - この記録は当分破られないだろうと予想（たぶん）

Knuthの教科書の記述

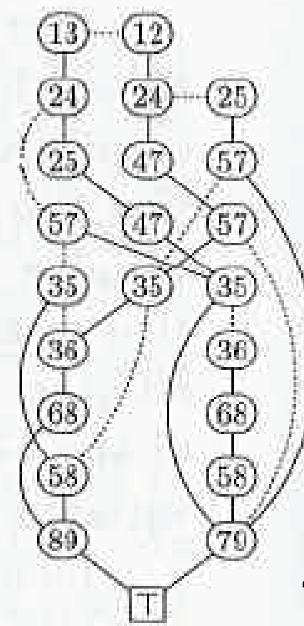
We can also use ZDDs to represent simple paths in an *undirected* graph. For example, there are 12 ways to go from the upper left corner of a 3×3 grid to the lower right corner, without visiting any point twice:





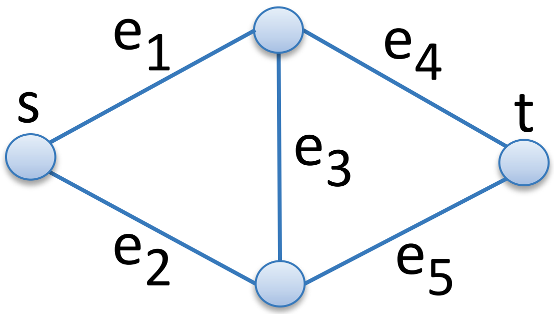
(132)

These paths can be represented by the ZDD shown at the right, which characterizes all sets of suitable edges. For example, we get the first path by taking the HI branches at (13), (36), (68), and (89) of the ZDD. (As in Fig. 28, this diagram has been simplified by omitting all of the uninteresting LO branches that merely go to $\boxed{\perp}$.) Of course this ZDD isn't a truly great way to represent (132), because that family of paths has only 12 members. But on the larger grid $P_8 \square P_8$, the number of simple paths from corner to corner turns out to be 789,360,053,252; and they can all be represented by a ZDD that has at most 33580 nodes. Exercise 225 explains how to construct such a ZDD quickly.



A similar algorithm, discussed in exercise 226, constructs a ZDD that represents all *cycles* of a given graph. With a ZDD of size 22275, we can deduce that $P_8 \square P_8$ has exactly 603,841,648,931 simple cycles.

Knuth によるアルゴリズム Simpath

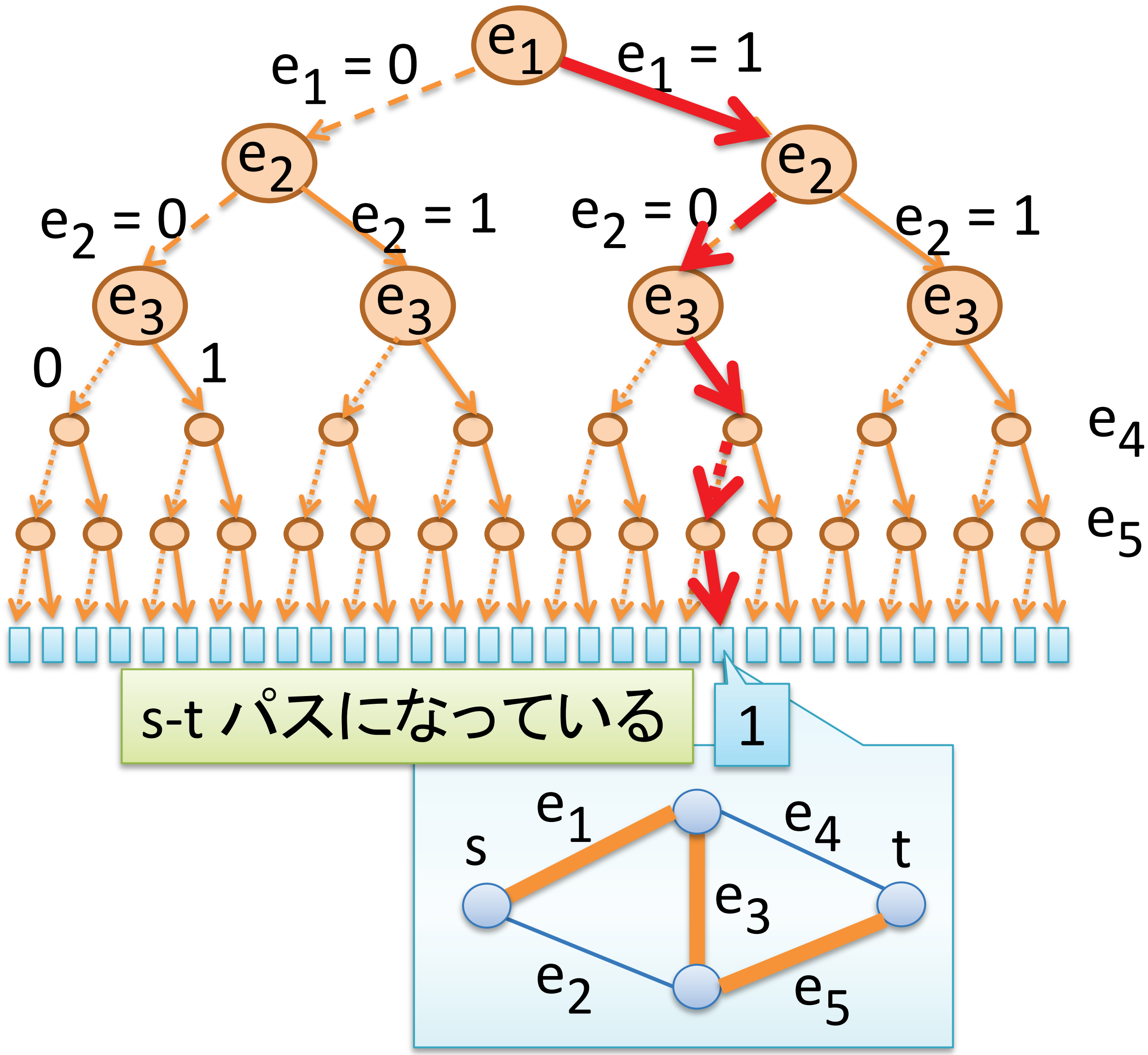


1. 枝に順番を付ける
（例えば、s から幅優先）
（もっと良い方法もあり）

2. 2分木を構築

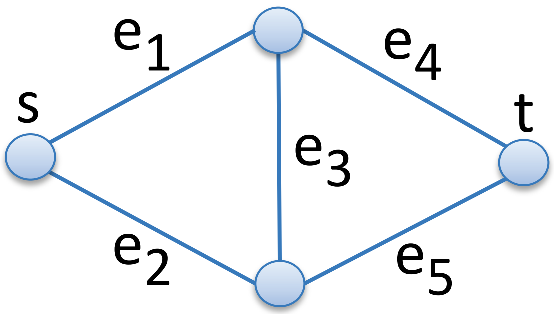
枝 $e_1, e_2, \ldots$ の順に処理

各枝変数 e_i に対し、
$e_i = 0$ or 1 を決めていく



$e_i = 1$ である枝が s-t パスになっているか？

Knuth によるアルゴリズム Simpath

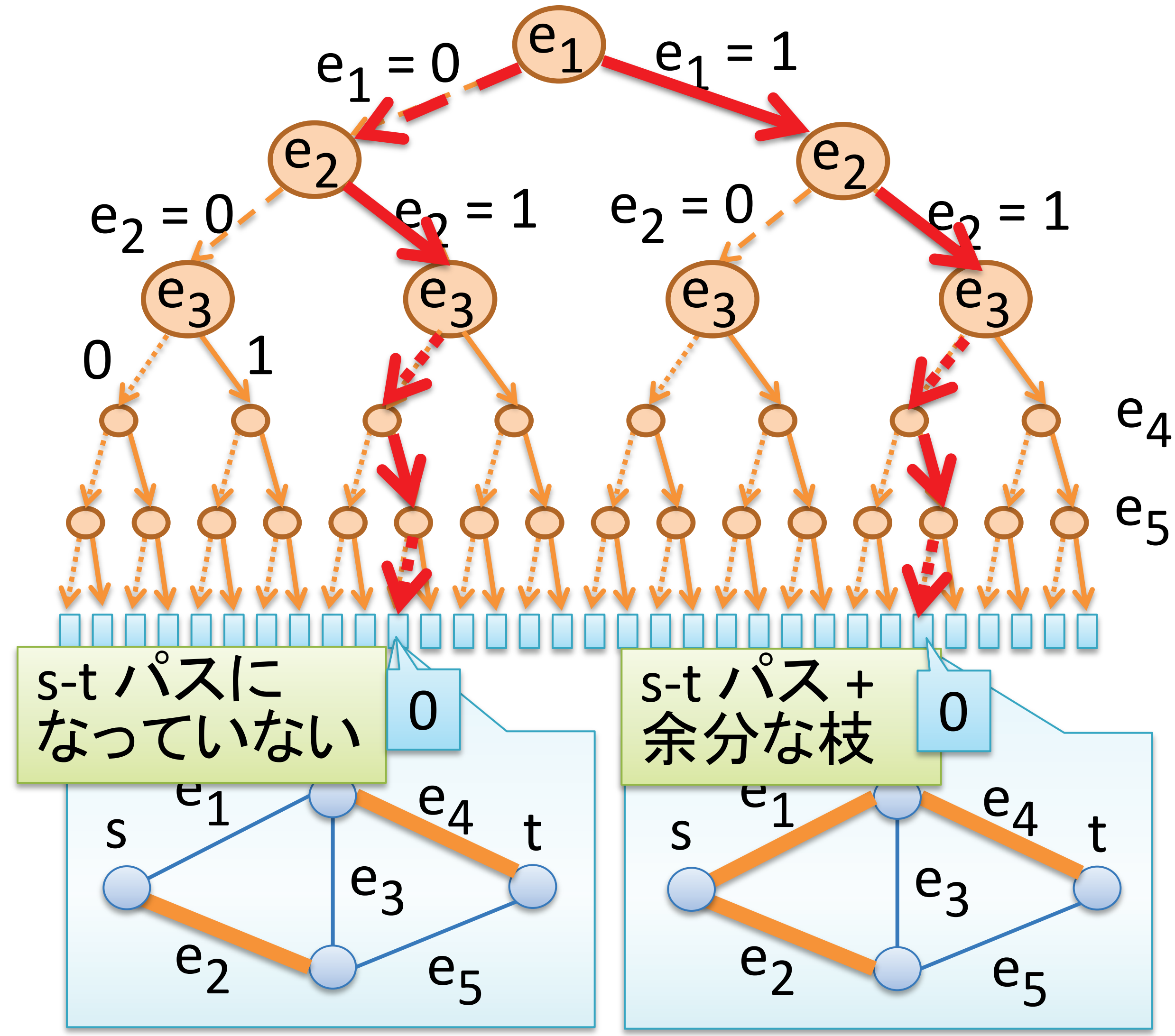


1. 枝に順番を付ける
（例えば、s から幅優先）
（もっと良い方法もあり）

2. 2分木を構築

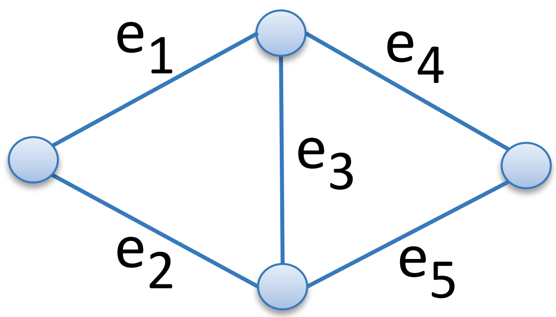
枝 $e_1, e_2, \ldots$ の順に処理

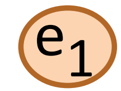
各枝変数 e_i に対し、
$e_i = 0$ or 1 を決めていく



$e_i = 1$ である枝が s-t パスになっているか？

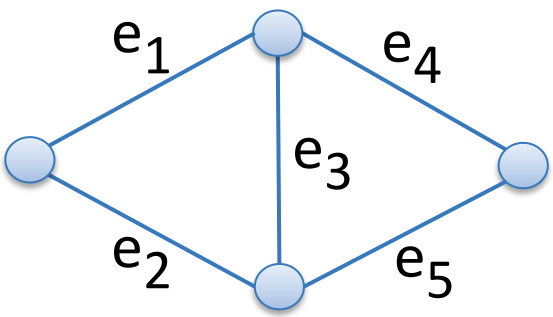
Simpathアルゴリズムの概略

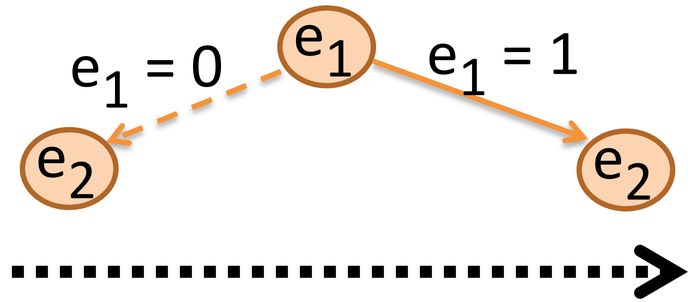




- 上位の枝から順に、枝の有無で
  場合分けしながら、幅優先で二分木を構築して行く
  - 途中で制約違反が見つかれば0で終端し、以降は分岐させない
  - 最後まで枝を決めて制約を満たすならば、1で終端

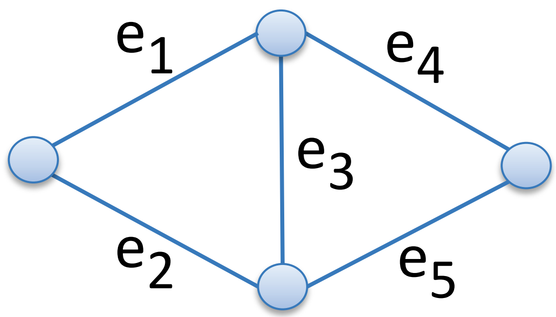
Simpathアルゴリズムの概略

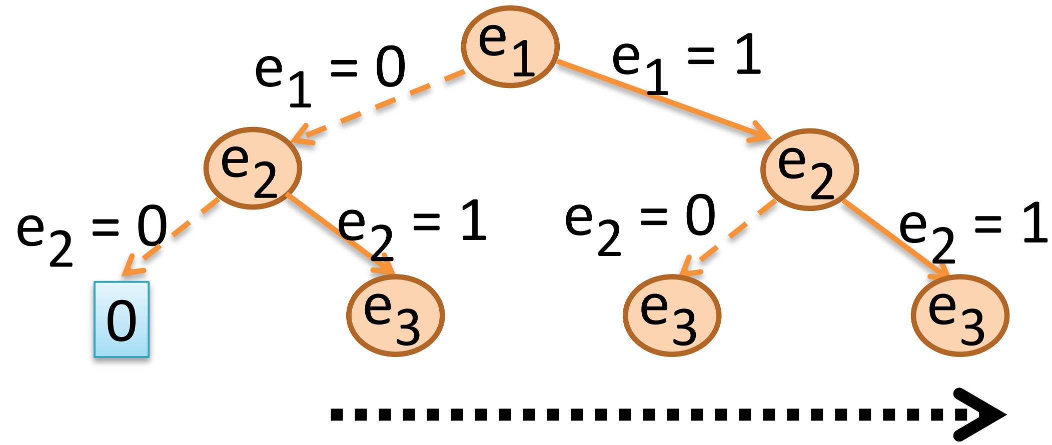




- 上位の枝から順に、枝の有無で場合分けしながら、幅優先で二分木を構築して行く
  - 途中で制約違反が見つかれば0で終端し、以降は分岐させない
  - 最後まで枝を決めて制約を満たすならば、1で終端

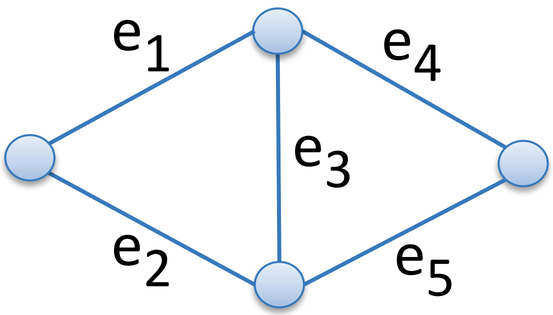
Simpathアルゴリズムの概略

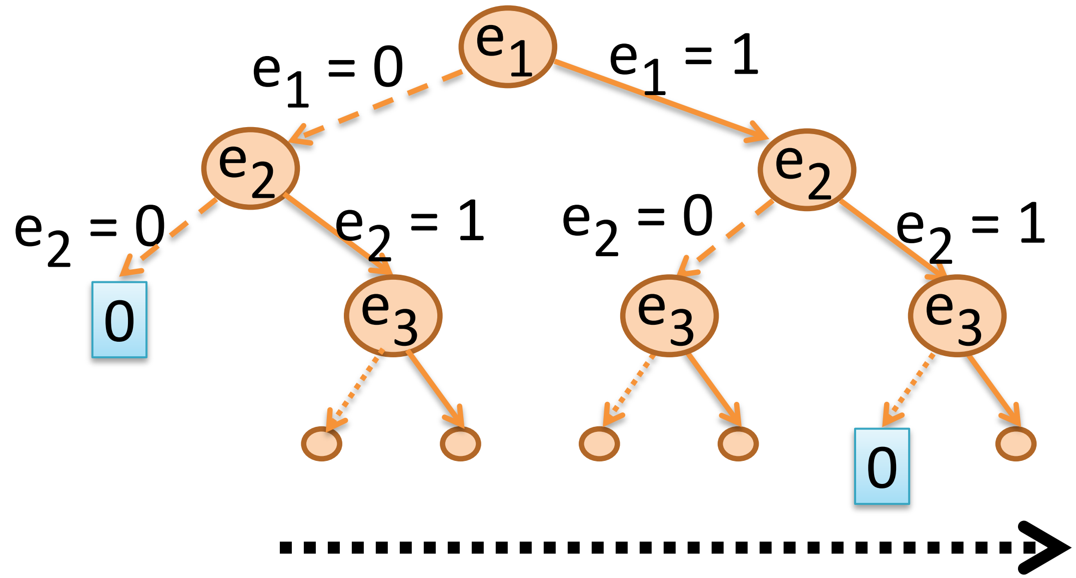




- 上位の枝から順に、枝の有無で
  場合分けしながら、幅優先で二分木を構築して行く
  - 途中で制約違反が見つかれば0で終端し、以降は分岐させない
  - 最後まで枝を決めて制約を満たすならば、1で終端

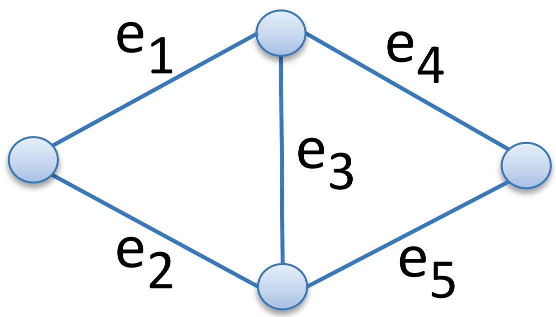
Simpathアルゴリズムの概略

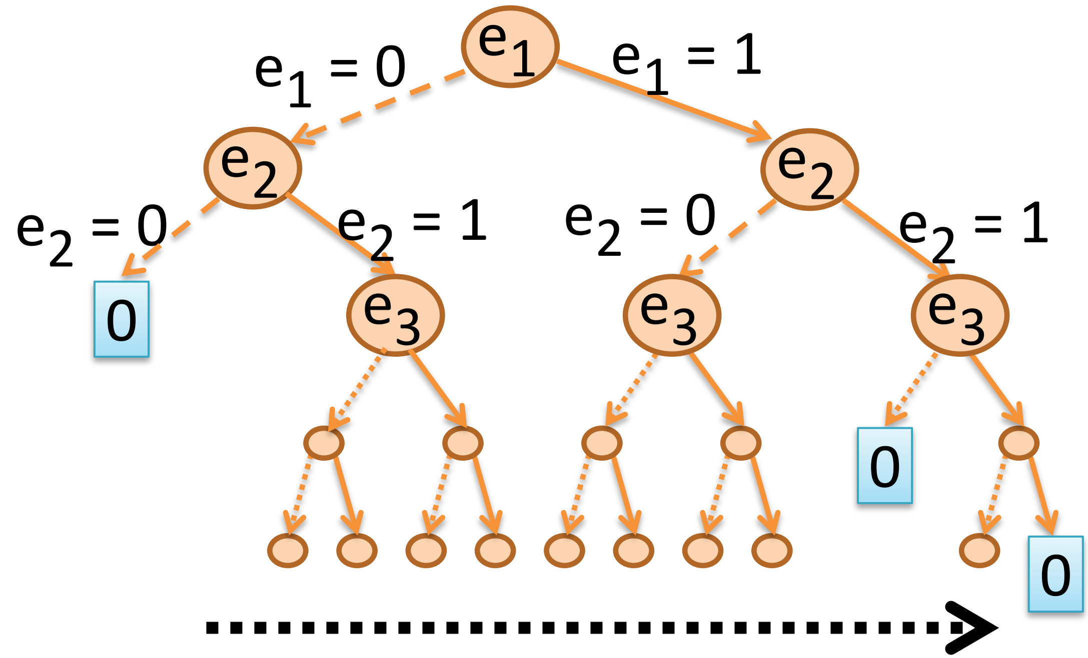




- 上位の枝から順に、枝の有無で
  場合分けしながら、幅優先で二分木を構築して行く
  - 途中で制約違反が見つかれば0で終端し、以降は分岐させない
  - 最後まで枝を決めて制約を満たすならば、1で終端

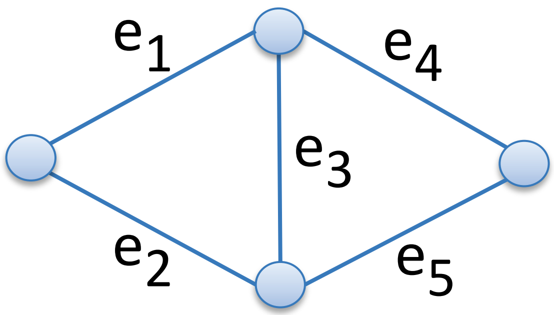
Simpathアルゴリズムの概略

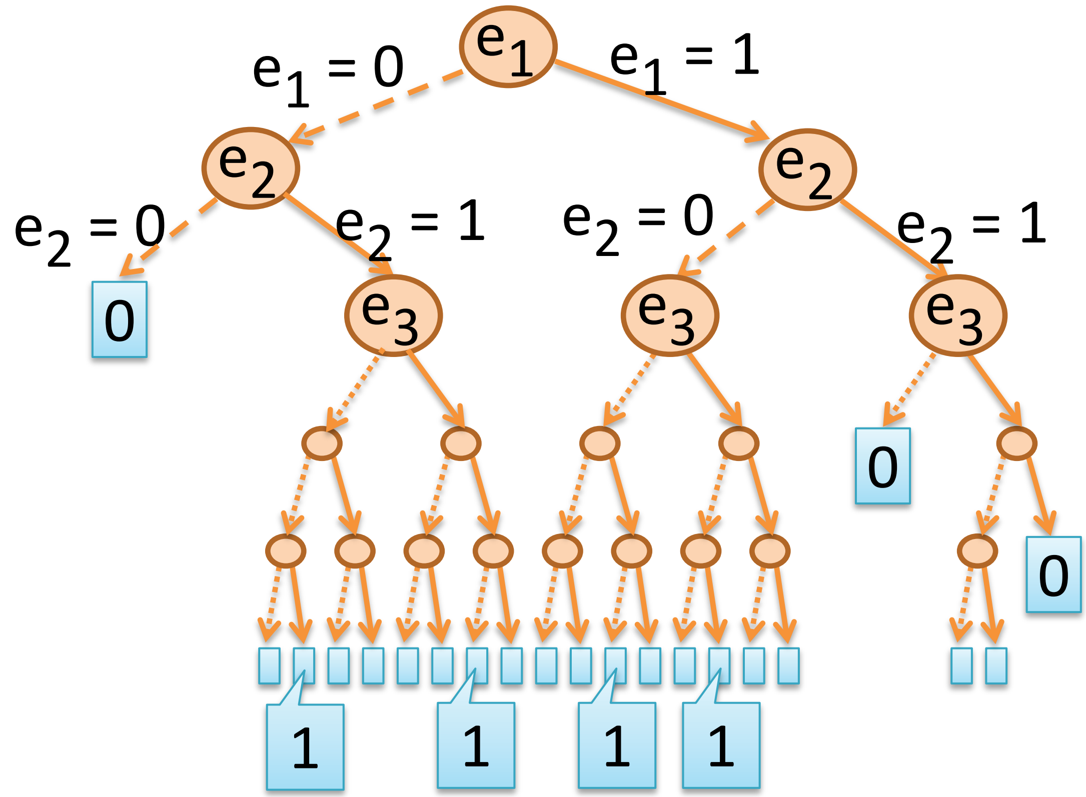




- 上位の枝から順に、枝の有無で
  場合分けしながら、幅優先で二分木を構築して行く
  - 途中で制約違反が見つかれば0で終端し、以降は分岐させない
  - 最後まで枝を決めて制約を満たすならば、1で終端

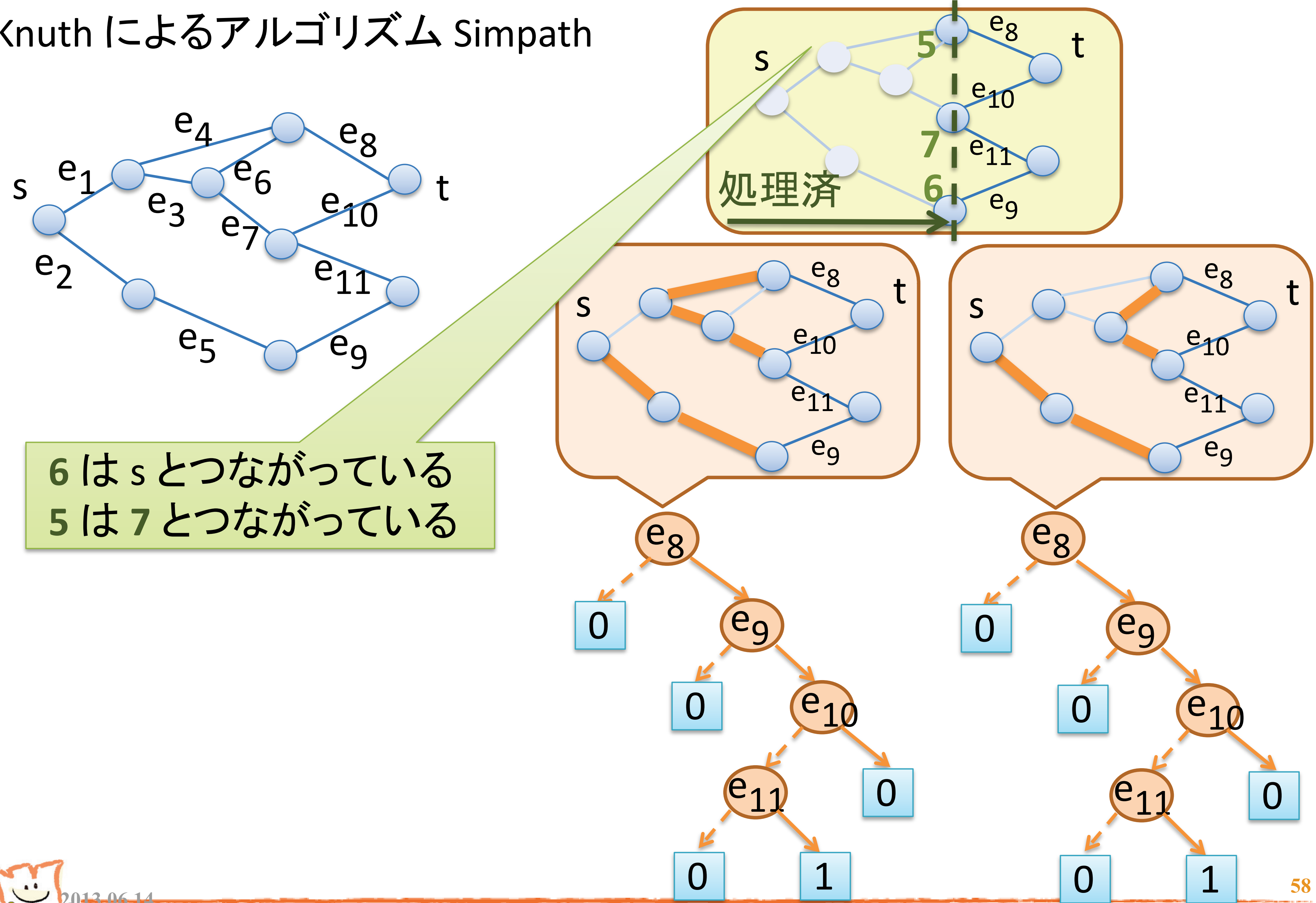
Simpathアルゴリズムの概略





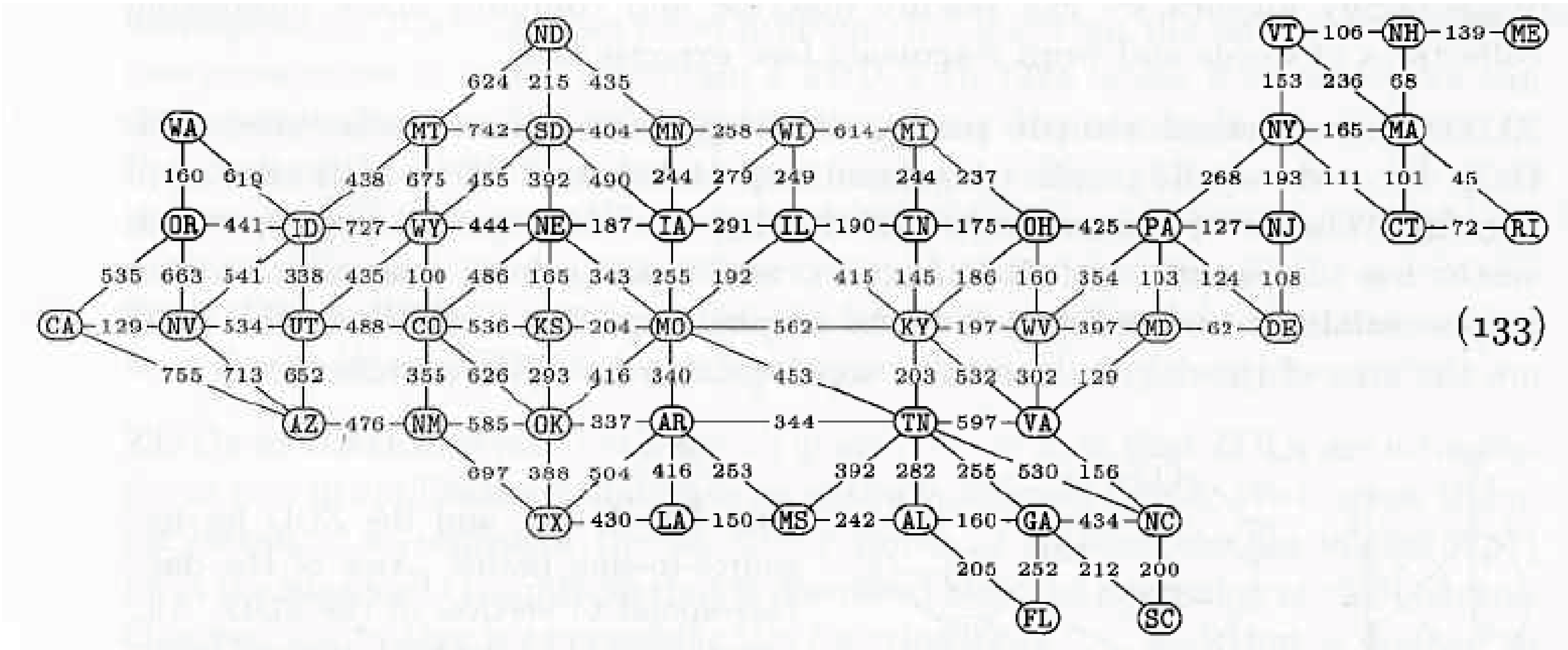
- 上位の枝から順に、枝の有無で場合分けしながら、幅優先で二分木を構築して行く
  - 途中で制約違反が見つかれば0で終端し、以降は分岐させない
  - 最後まで枝を決めて制約を満たすならば、1で終端

Knuth によるアルゴリズム Simpath



6 は s とつながっている
5 は 7 とつながっている

“simpath” for US map in Knuth-book

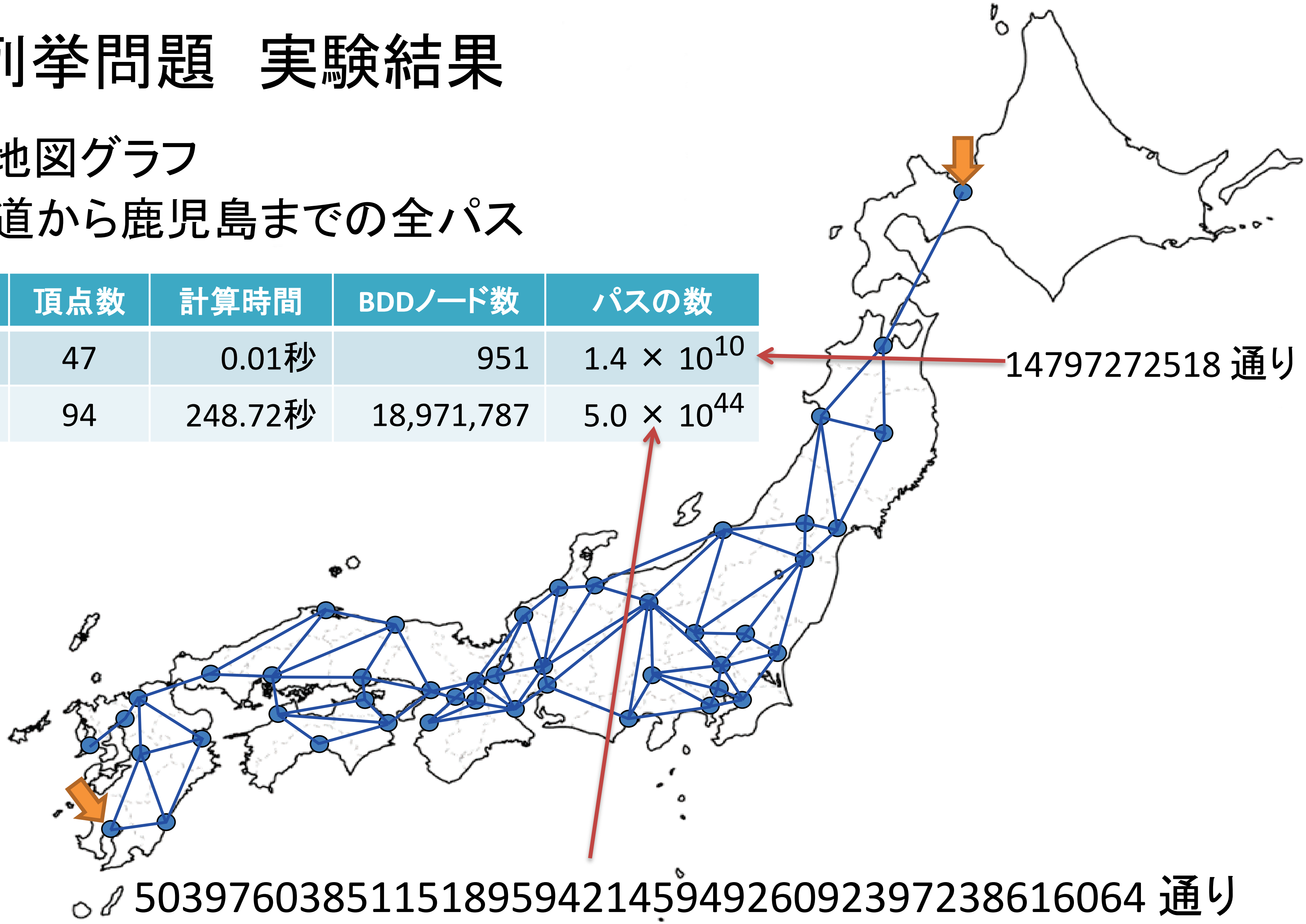


パス列挙問題　実験結果

日本地図グラフ
北海道から鹿児島までの全パス

	頂点数	計算時間	BDDノード数	パスの数
日本地図	47	0.01秒	951	1.4×10^{10}
2重化	94	248.72秒	18,971,787	5.0×10^{44}

14797272518 通り



5039760385115189594214594926092397238616064 通り
(= 503正9760澗3851溝1518穣9594杼2145垓9492京6092兆3972億3861万6064)

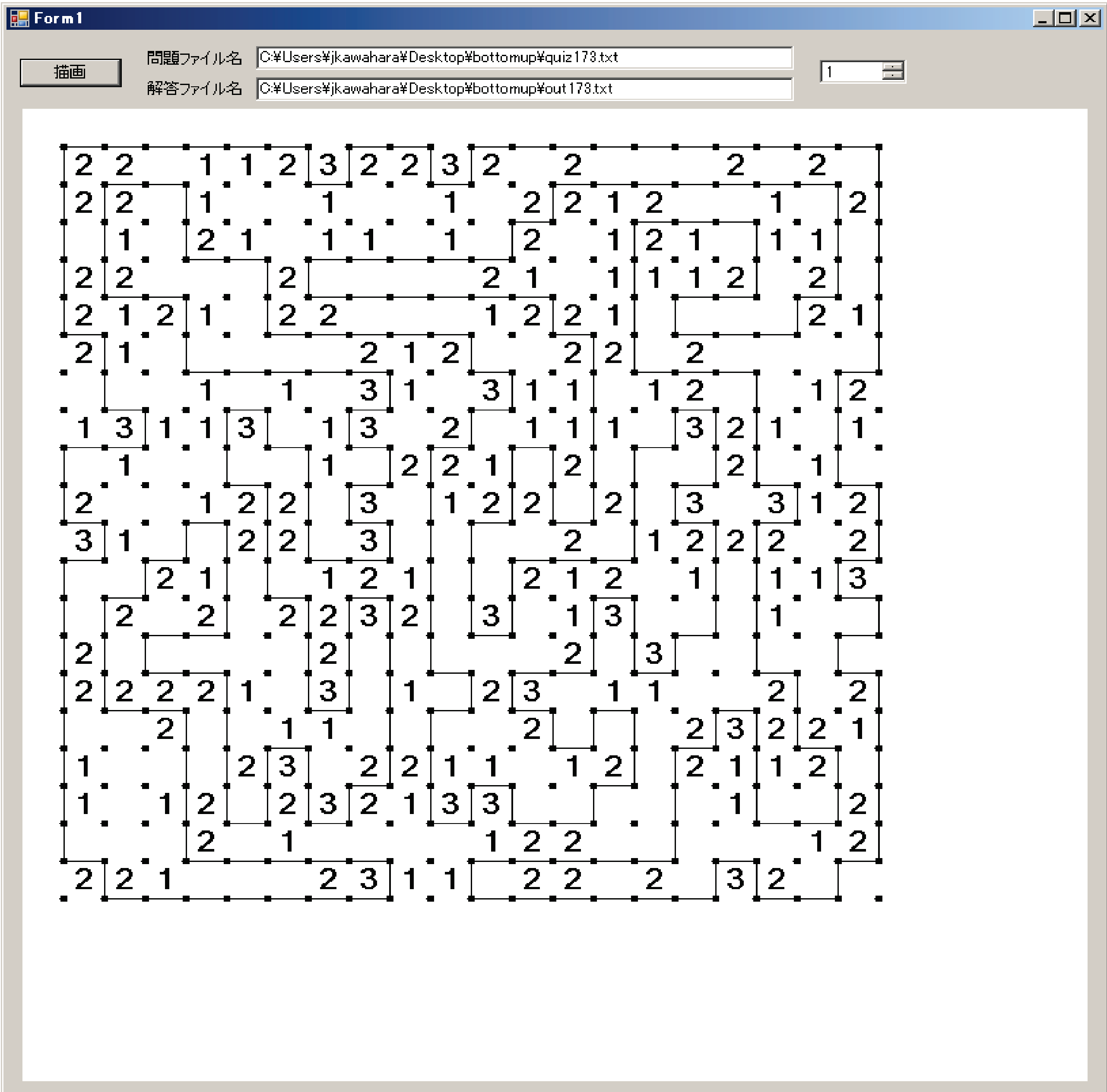
「おねえさん問題」の一般化

- パス列挙のバリエーション
  - パス列挙 → サイクル列挙 （Knuth本に演習問題あり）
  - 無向グラフ → 有向グラフでも可能（これも演習問題）
  - 起点終点が複数ペアの場合でも可能（→非交差配線問題）
- ERATOで議論しているうちに、他にも様々なグラフ列挙問題に適用できることが分かってきた
  - 部分木列挙、大域木/林の列挙、カットセット列挙、グラフのk分割問題、連結確率の計算、完全/不完全マッチング列挙、etc.

  → 実社会の様々な問題への応用が可能
- Tutte多項式の計算法[Sekine-Imai95]との類似性
  - 理論的な広がり

$$T(x,y) = \sum_{A \subseteq E} (x-1)^{\rho(E)-\rho(A)} (y-1)^{|A|-\rho(A)}$$

スリザーリンクパズル（条件付きサイクル）の列挙



電力網への応用（研究成果プレスリリース）

- 林 泰弘 教授（早稲田大）との共同研究
  - 電力系スマートグリッド業界のリーダ的存在（経産省スマートハウス標準化検討会座長、他多数の要職）
  - 電力網最適化の研究で1990年代より湊と協力関係
- 大震災後、より緊急性の高い研究課題に
  - 今後長期的に不足する電力を自然エネルギーで補うために必須の電力網解析・制御技術を支援

情報科学の研究者集団として我が国の苦境を克服するためできる限り貢献したい。

→ ERATOプロジェクトでの取り組みを加速

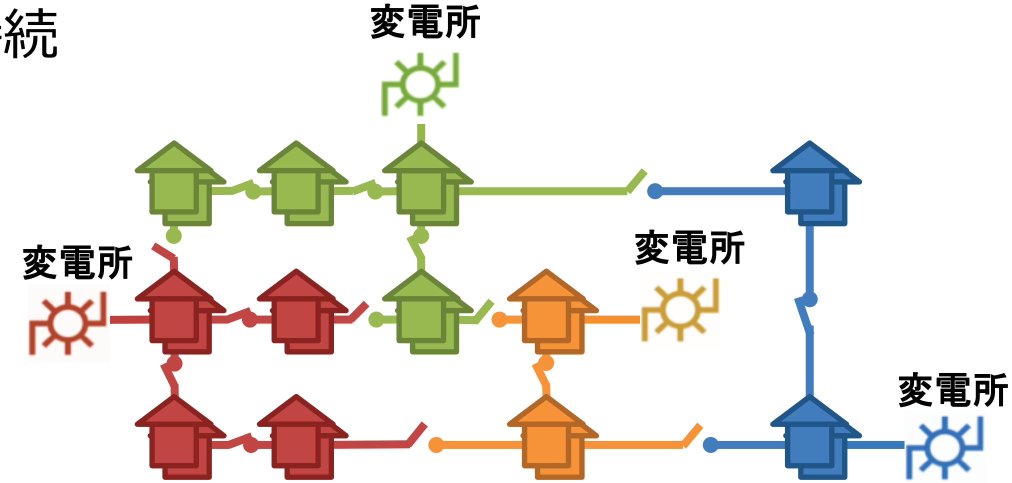
電力網のスイッチ制御

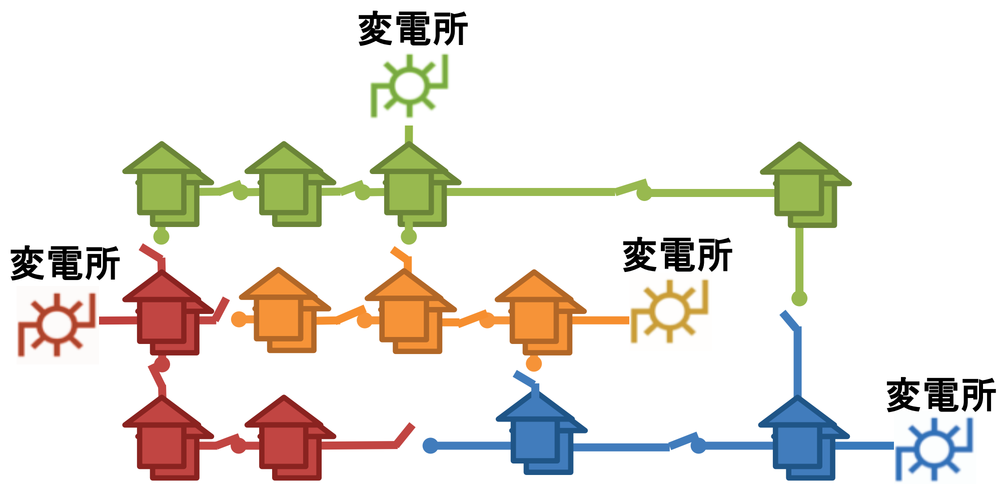
・各区域が必ず１ヶ所の変電所設備に接続
・停電しない
・異なる変電所系統をつながない
・電流が多過ぎると電線が焼ける
・遠くに伝えると電圧が下がる

膨大な数の構成が存在

右のような13スイッチの小さな例題でも
8192通りの組合せの中から
210通りを探し出す問題

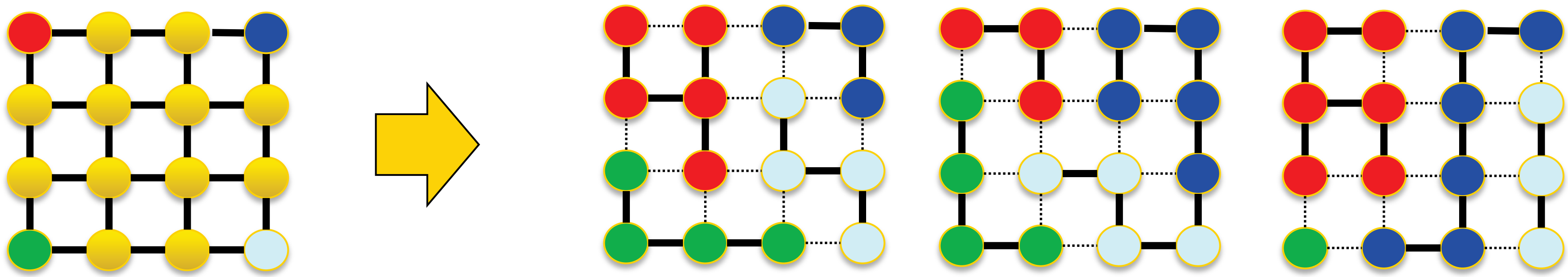
標準的な配電網の例では
468個のスイッチがあり、
10の140乗の組合せから
探し出す問題に





電力網のスイッチ制御

・ZDDを使った新しい高速アルゴリズムを開発。
・標準的な電力網モデル（スイッチ468個）でグラフ的制約と電気的制約を共に満たす解を全て求め、圧縮して表現することに成功
・電力の損失を最小にする組合せを求めることにも成功



圧縮データ:約110万個（779MB）　　実行時間:数十分

解の個数:約10^{63}（**2136那由他8201阿僧祇3834恒河沙8532極9116載8261正2214澗8049溝560穣9817杼8392垓4438京5235兆3981億8952万1540）通り**

電力網のほかにも様々な応用がある

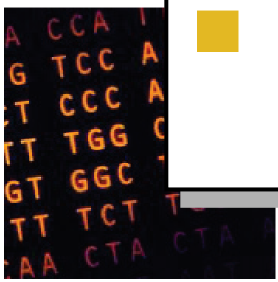
・避難所の配置問題　・道路、鉄道、ガス、水道、通信
・選挙の区割り問題　・建物のフロアプラン

「おねえさん問題」の工学的応用

社会的に重要な様々な実問題に関係

- 有向/無向グラフのパス列挙：
  - 地理情報システム
  - 大規模システムの依存関係の解析、フローチャートの解析
  - ナンバーリンク、スリザーリンク等のパズル
  - 文字列の連接可能性の列挙
- グラフk分割問題：
  - 電力網の配電区割りの列挙
  - 避難所の配置問題
  - 選挙区割り問題
- 地理情報、電力網、物流網のような社会インフラの構造は，平面グラフや格子グラフに近い形であることが多い。
  → 圧縮列挙・高速化の効果が極めて高くなる傾向がある。

本プロジェクトの基本構想



- 様々な工学的応用を持つ基盤技術として
「離散構造処理系」に着目し、研究開発を行う

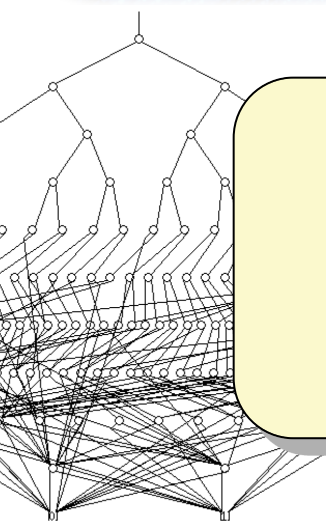
システム設計自動化
データマイニングと知識発見
大規模システム故障解析
機械学習と自動分類
制約充足問題
生命情報科学
web情報解析

工学的応用
→ 社会への
影響大

本研究の
対象領域

性能向上
(10倍～
100倍以上)

離散構造処理系



集合理論
記号論理
帰納的証明
組合せ論
グラフ理論
確率論

数学的
概念構造

まとめ

- アルゴリズム技術の社会的な重要性
  - コンピュータの計算速度が10～100倍変わると、世の中の人々の生活に大きな影響がある。
  - コンピュータ自体の高速化も大事だが、アルゴリズムの改良も非常に重要
    → 大規模な問題ほど高速化の効果が大きい
    → 特別な装置が不要、安価、省エネルギー
- 離散構造処理系プロジェクト
  - 膨大な個数の場合分けを圧縮して超高速に処理。
  - 「論理」や「集合」は、コンピュータのあらゆる問題に関係している。
  - BDD/ZDDの技術は、社会の様々な問題に幅広く応用されている。
- 「フカシギの数え方」
  - 組合せ爆発のすごさと、アルゴリズム技術の威力
  - 一見、単純な問題が様々な実社会の問題につながっている。