

炭化水素列挙 RTA を つまみ食い

2025/10/12

すうがく徒のつとどい8 @ 大阪大学

すむーずぷりんちゃん🍰 @mat_der_D

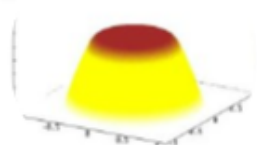
自己紹介

- 名前: すむーずぷりんちゃん 🍮
- Twitter(現X) ID: @mat_der_D
- 略歴:
 - ~'20年3月: 京都大学在籍
...専門は流体力学の数値計算
 - '20年4月~: 東京で社会人
 - '24年10月~: 関西に帰還
- つどい発表歴 (関西...つどい):
 - 第11回 ('18年): クソ問のすゝめ
 - 第12回 ('19年): 私の趣味数学

← すむーずぷりんちゃん 🍮
24万 件のポスト

🔍

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \sum_{i=1}^3 u_i \frac{\partial}{\partial x_i}$$



プロフィールを編集

すむーずぷりんちゃん 🍮
@mat_der_D

Q. で始まるツイートはアンケート付です。流体力学/数値解析/数学/物理学/Python/Fortran/Julia/Rust/C#/作問/ユーフォニアム/ENTP/spm6th,8th,9th,11th

smooth-pudding.hatenablog.com 📅 2014年9月からXを利用しています

581 フォロー中 2,685 フォロワー

炭化水素？列挙？RTA？

- **炭化水素**: 炭素いくつかと水素いくつかがついた分子
- **列挙**: 一つひとつ数え上げる
- **RTA**: Real Time Attack = 短時間クリアを目指す遊び

つまり...

炭化水素列挙 RTA
= 炭素と水素からなる分子を
なるべく短時間で
全部数え上げる遊び

なぜそんなことを？

先行研究『炭素と水素による構造式一覧』

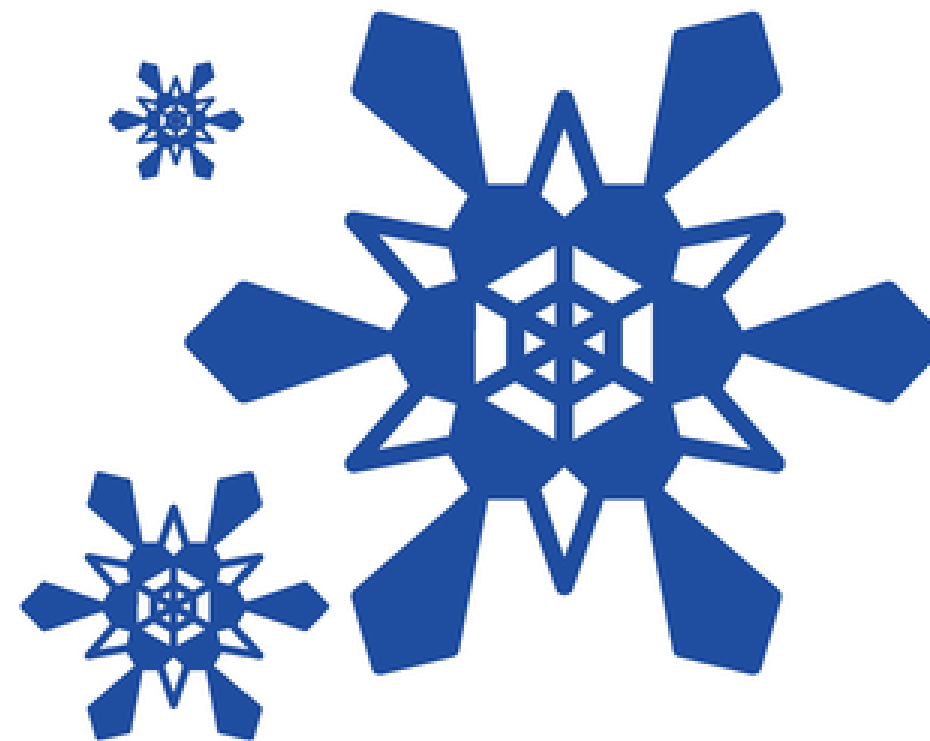
- 2024年5月、技術書典16で出会う
- 炭化水素を頑張って列挙しているらしい
- Python でめっちゃ頑張っている
- ほな Rust 使って高速化でもしてみよか
- たーのしー！(洗脳完了)

The Chemical Structures consist of Carbon and Hydrogen

炭素と水素による 構造式一覧

改訂第2版

著者 :@_rhodium_



宣伝コーナー

技術書典19にて右の電子書籍を販売します。

- イベント期間: 2025/11/15-30
(イベント終了後も販売予定)
- 場所: オンライン
- 価格: 1000円 (予定)

今日はこの本の一部を抜粋して講演します。



炭化水素列挙
RTA

すむーずぷりんちゃん 著

今日話すこと・前提知識

アブストラクトの以下の範囲:

2 講演で扱うこと

以下のトピックのうちいくつかを, 時間の許す範囲で紹介します. 途中, 行列を使った議論が出てくるので, 線形代数のごく基礎的な内容については既知とします.

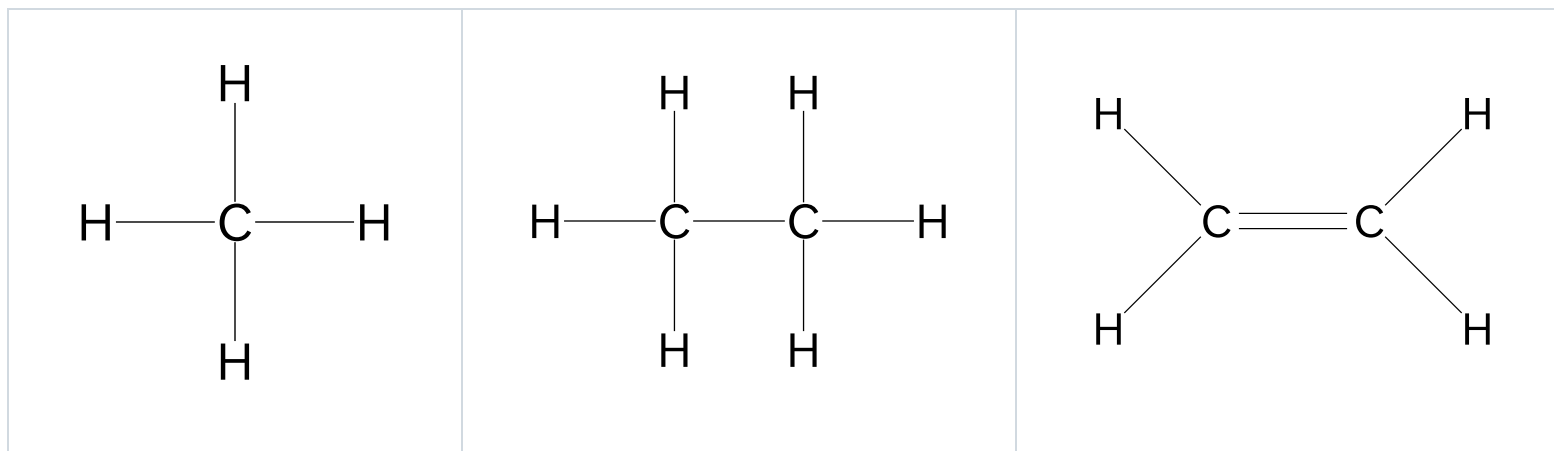
- グラフ理論に入門し, 炭化水素列挙をグラフの問題に焼き直す
- 基本的な探索アルゴリズムを用いて列挙する方法を考える
- 枝刈りを利用して探索範囲を絞る
- ~~● 不変量をうまく見つけて探索範囲を絞る~~
- ~~● 分子構造の対称性を活用して探索範囲を絞る~~

問題設定

炭素と水素の基礎知識

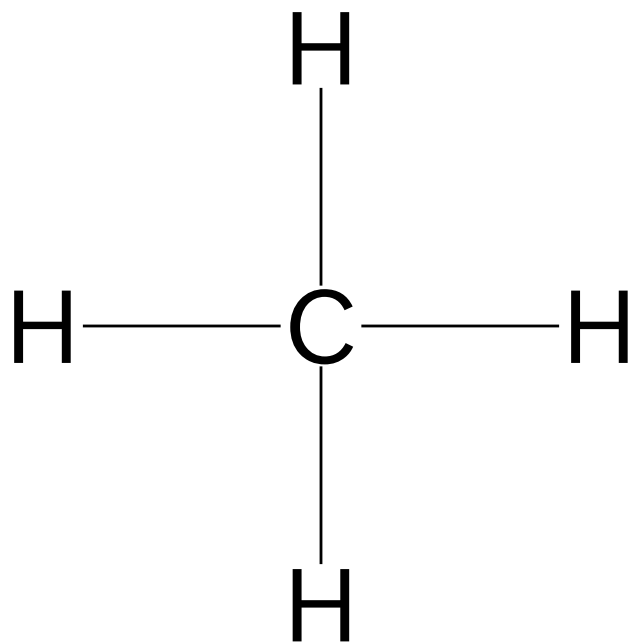
- 水素は元素番号 1 の原子。記号は H。手が1本ある。
- 炭素は元素番号 12 の原子。記号は C。手が4本ある。
- 違う原子の手同士はつなぐことができる。
- 自分自身の手同士はつなげない。

炭化水素の例:



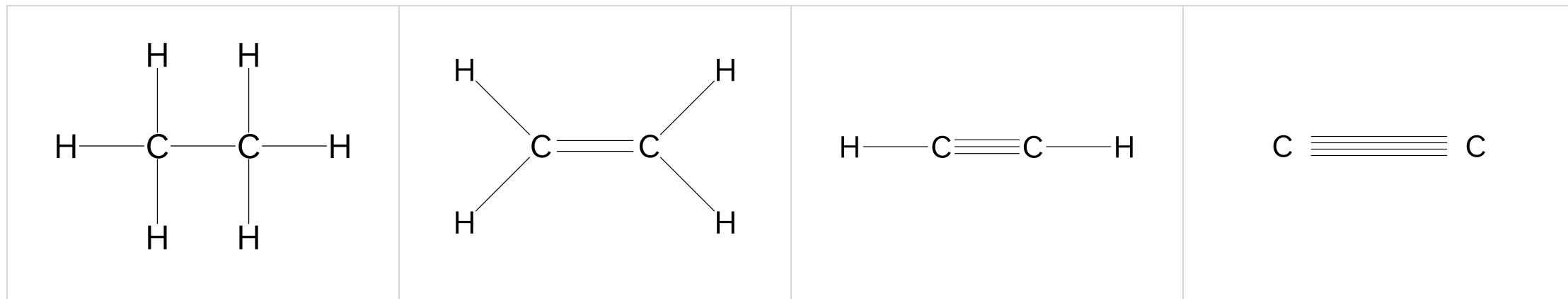
炭化水素を手で数えてみる

炭素1個 → 1種類



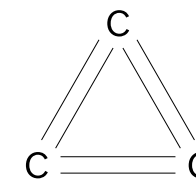
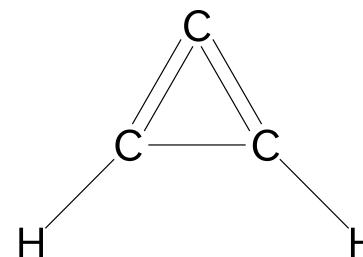
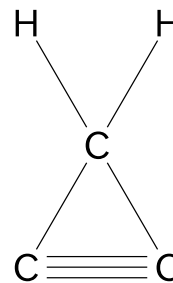
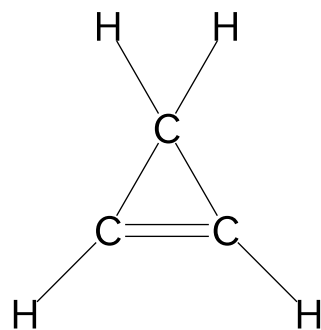
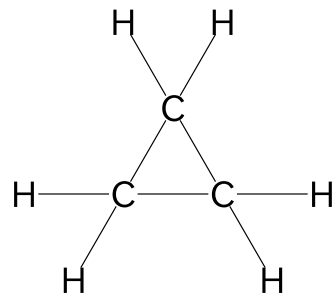
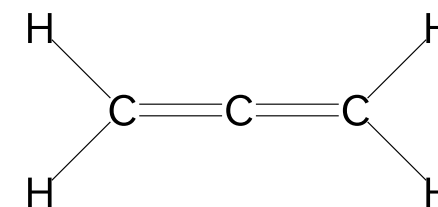
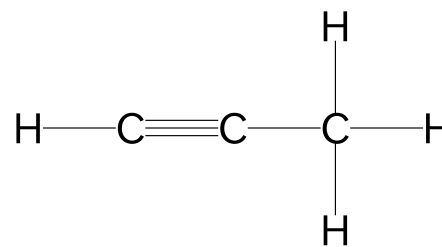
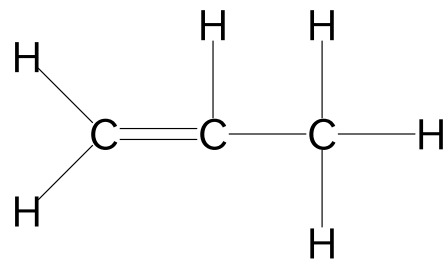
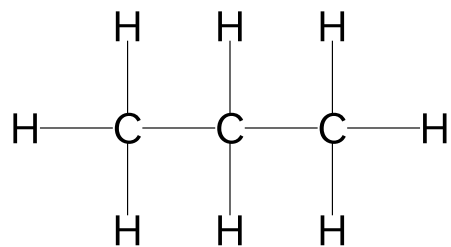
炭化水素を手で数えてみる

炭素2個 → 4種類



炭化水素を手で数えてみる

炭素3個 → 9種類



炭化水素を手で数えてみる

- 炭素4個 → 37種類
- 炭素5個 → 146種類
- 炭素6個 → 772種類
- 炭素7個 → 4,449種類
- ...

流石に手作業は無理。コンピューターを使う。

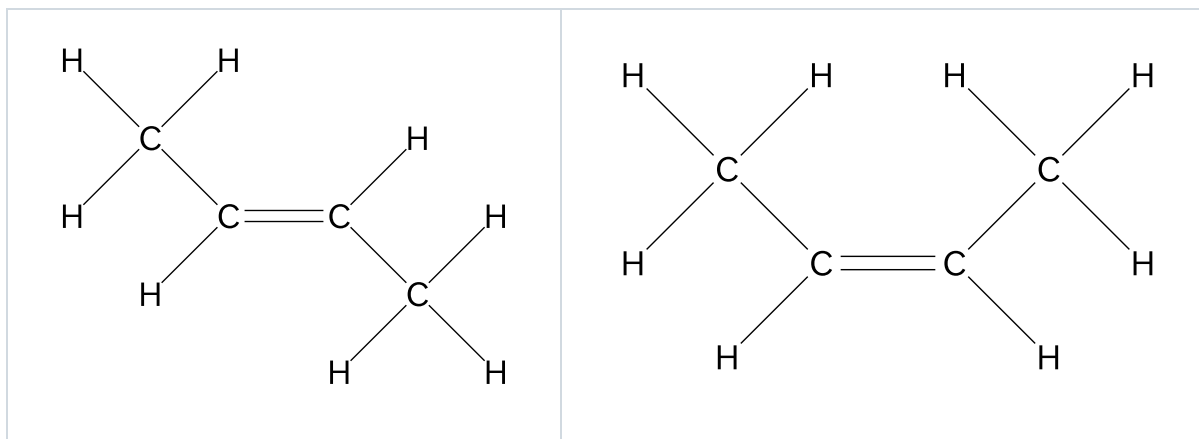
??? 「RTAは人力でやるものでは？」

🍰 「いえ、このゲームはコンピューターがプレイヤーなんです！(迫真)」

RTA のレギュレーション

- 与えられた個数(2～10)の炭素を含む分子をすべて列挙する
- 何らかの方法で分子構造を復元できるのであれば、データ構造は問わない
- 分子の立体構造は区別しない
- 分子が実際に合成できるかは考えない

立体構造が異なるが同一視する例:

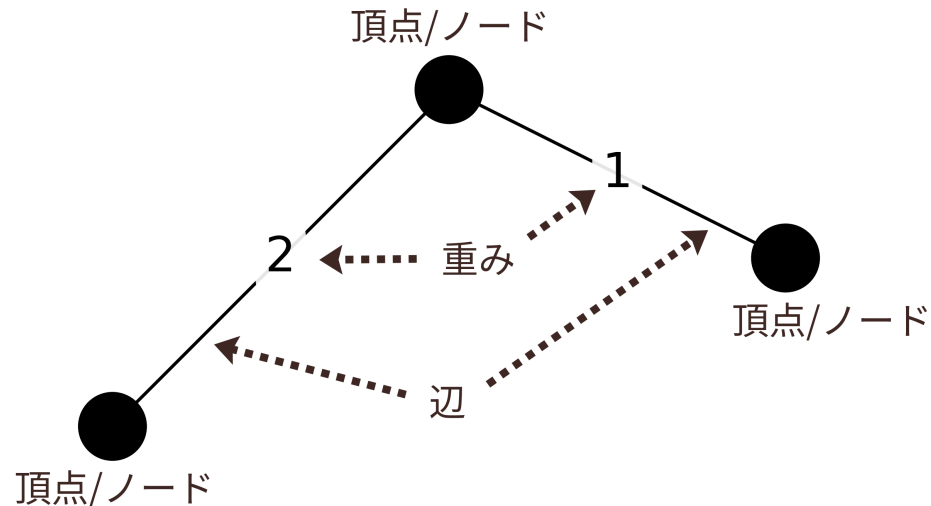


グラフ理論入門

グラフとは？

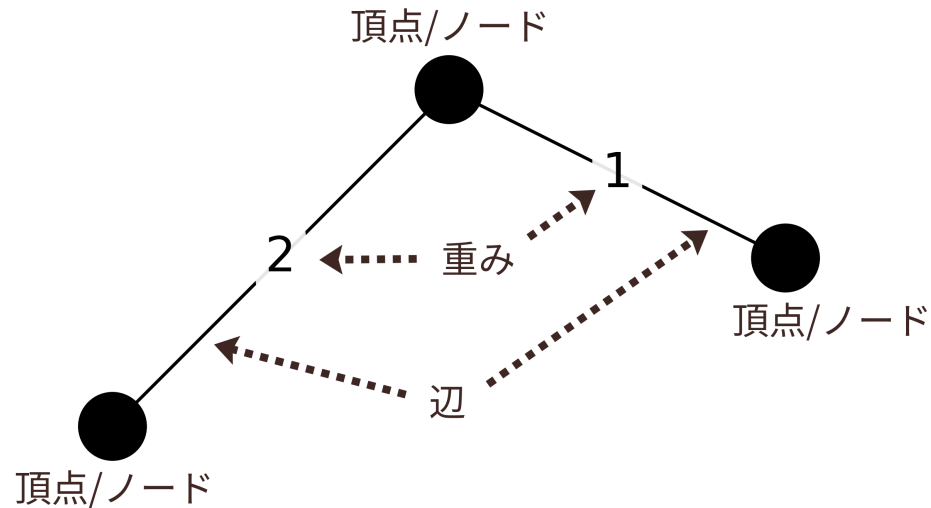
- 頂点 (ノード) と辺 (エッジ) からなる対象
- 辺に向きはない (無向グラフ)
- 各辺には**重み**という整数値が設定されている (重み付きグラフ)

この講演では重み付き無向グラフのみを扱い、単にグラフと呼ぶ。



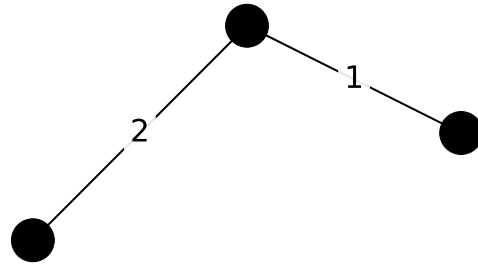
グラフ用語の導入

- (ある頂点の)**次数**: その頂点につながっている辺の重みの合計
- **連結**: グラフがひとつつながりであること
- **隣接行列**: つながり情報を持った行列 (後述)

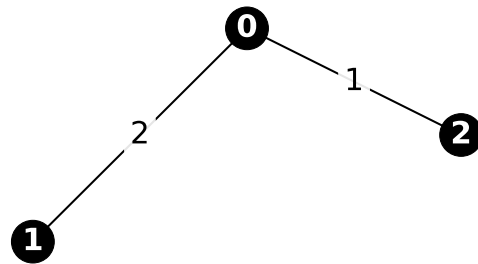


隣接行列

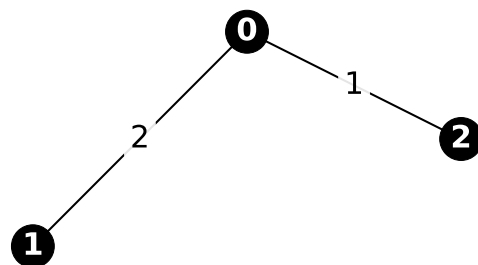
以下のグラフが与えられたとする。



各頂点に 0, 1, 2 の数字を割り振る。



隣接行列



重みの情報を表で表現してみる。

	頂点 0	頂点 1	頂点 2
頂点 0		2	1
頂点 1	2		
頂点 2	1		

隣接行列

	頂点0	頂点1	頂点2
頂点0		2	1
頂点1	2		
頂点2	1		

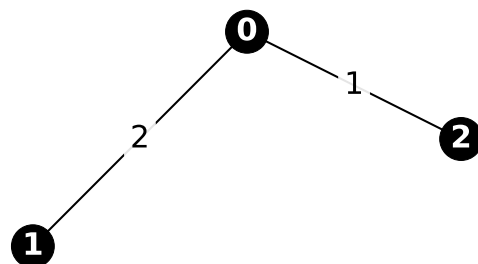
簡略化してみる。

$$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

これが隣接行列。

隣接行列と次数

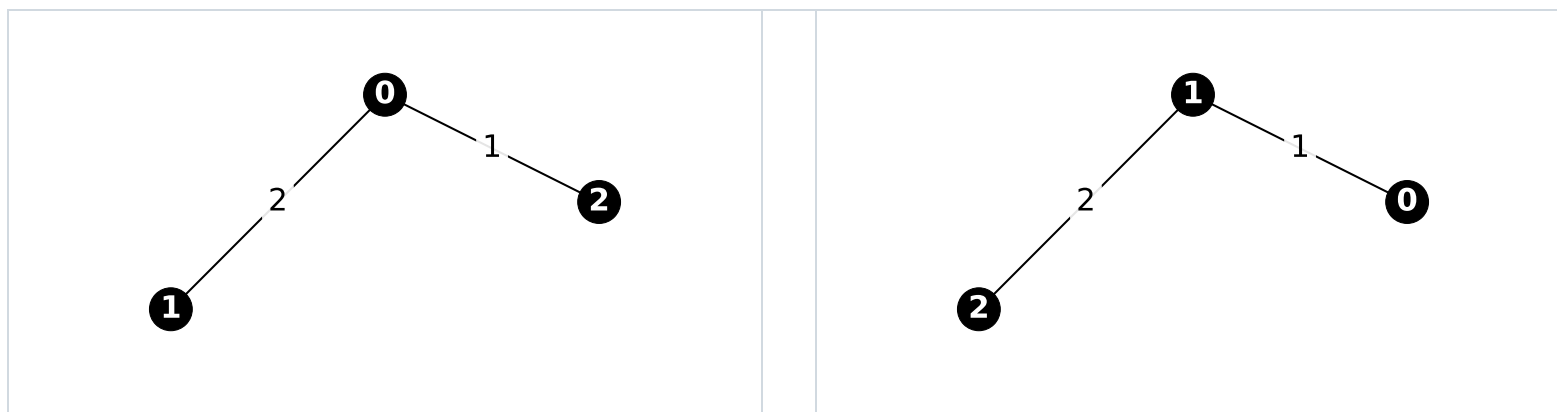
隣接行列を用いると、次数はその頂点に対応する行 (または列) の和になる。



$$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

同値な隣接行列

頂点の番号の付け方が変わると、隣接行列も変わることがある。
2つの隣接行列が同じグラフに対応するとき、**同値**であるということにする。



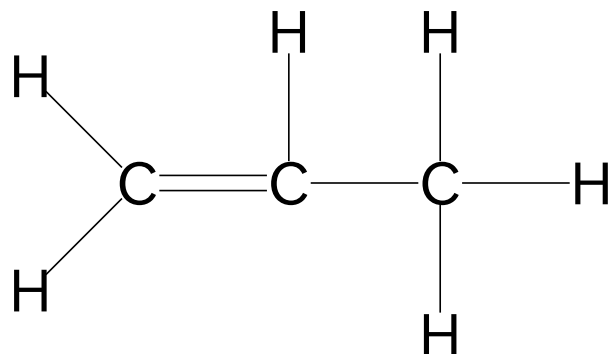
$$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix}$$

炭化水素とグラフの対応

以下のステップで炭化水素をグラフに変換できる。(復元も可能)

1. 水素をもぎ取る (手の数を考えれば復元可能)
2. 炭素を頂点に、手の繋がりの本数を重みにする



炭化水素の列挙をグラフの問題として考える

以下を満たすグラフを**炭化水素**と呼ぶことにする。

- 連結
- すべての頂点の次数が4以下

すると、元の問題は以下のように表現できる。

問題. N を正の整数とする。サイズが N の隣接行列であって、炭化水素であるものをすべて求めよ。ただし同値なものは同一視する。

→ 行列を数える問題になったので、コンピューターで扱いやすい！

状況の整理

候補はどれくらいあるか

炭素の数が N 個なら、隣接行列の要素は N^2 個。ただし

- 炭化水素は必ず対称行列
- 対角成分はすべて 0

→ 独立に動かせるのは $\frac{1}{2}(N^2 - N) = \frac{N(N-1)}{2}$ 個。

例. $N = 4$ なら、独立な成分は 6 個。

$$\begin{pmatrix} 0 & x_1 & x_2 & x_3 \\ x_1 & 0 & x_4 & x_5 \\ x_2 & x_4 & 0 & x_6 \\ x_3 & x_5 & x_6 & 0 \end{pmatrix}$$

候補はどれくらいあるか

各成分は 0, 1, 2, 3 のいずれかを取る。(唯一の例外は C_2)

→ 候補の数は $4^{N(N-1)/2}$ 種類。

→ $N = 10$ の場合は

$$4^{N(N-1)/2} = 1,237,940,039,285,380,274,899,124,224 \approx 10^{27}$$

→ すべてを調べるのは現実的ではない。

無駄な候補

実際には、 $4^{N(N-1)/2}$ 種類の候補の中には無駄なものが多く含まれる。

例えば

- 成分が大きすぎると、いずれかの頂点の次数が 4 を超えてしまう。
- 連結でなくなってしまう。特に 0 が多いと孤立した頂点が出ることも。
- グラフが同じもの、つまり同値な隣接行列が存在する。

なんとかして候補を絞りたい。

候補を絞る

探索を2ステップに分解する。

【Step1】

すべての重みが1であるような炭化水素のみをすべて列挙する。

【Step2】

Step1 で列挙した各々の炭化水素について、どの頂点も次数が 4 を超えない範囲で、重みを増やしていく。

この分解のメリット:

- Step1 の候補は $2^{N(N-1)/2}$ 種類となり、大幅に少なくなる (まだ多いが)
- Step2 では次数が4を超える候補をバッサリ削れる。

候補を絞る

定義. すべての辺の重みが1の炭化水素を**飽和炭化水素**と呼ぶ。

Step1 は以下のように言い換えられる。

【Step1】

飽和炭化水素をすべて列挙する。

この講演では、この Step1 を高速化する方法を考えていく。

Step2 については書籍を参照。

探索アルゴリズム

深さ優先探索

問. n を正の偶数とする。開き括弧 "(" および閉じ括弧 ")" の2種類の記号を考える。各項がこれらのいずれかである記号の列 $s_1, s_2, \dots, s_n$ であって、以下の条件を満たすものをすべて列挙せよ。

- "(" と連続する部分を消去する操作を繰り返すと、列全体を消去できる。

例 ($n = 6$).

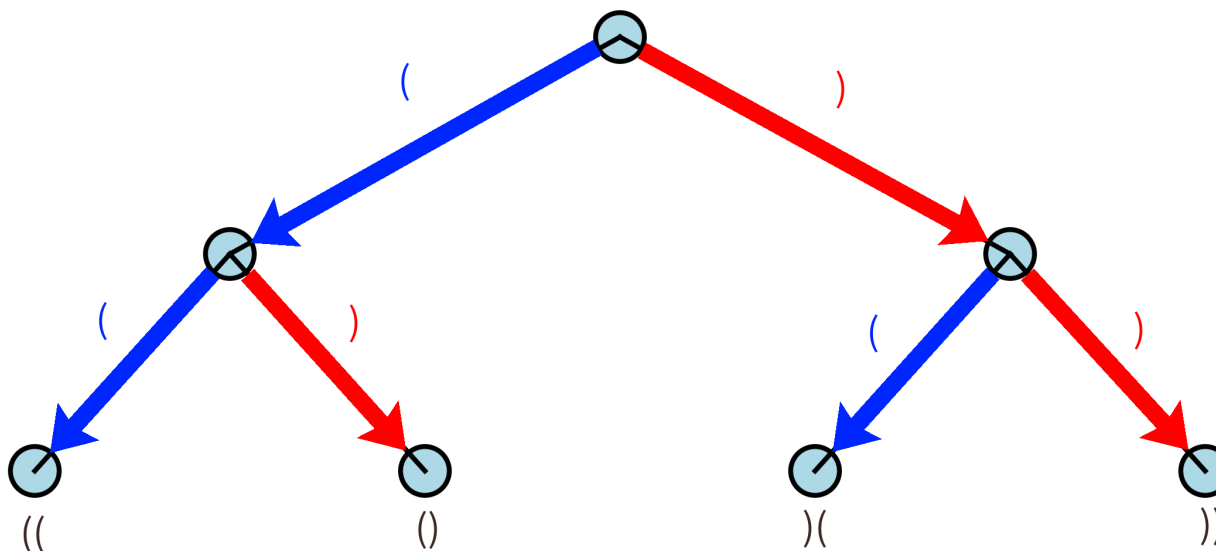
- 満たす例: `()()()` `((()))` `((()()))`
- 満たさない例: `)()()(` `((()))(` `((()((`

深さ優先探索

$n = 2$ の場合で記号列を全列挙すると、以下の4通りが考えられる。

((())())

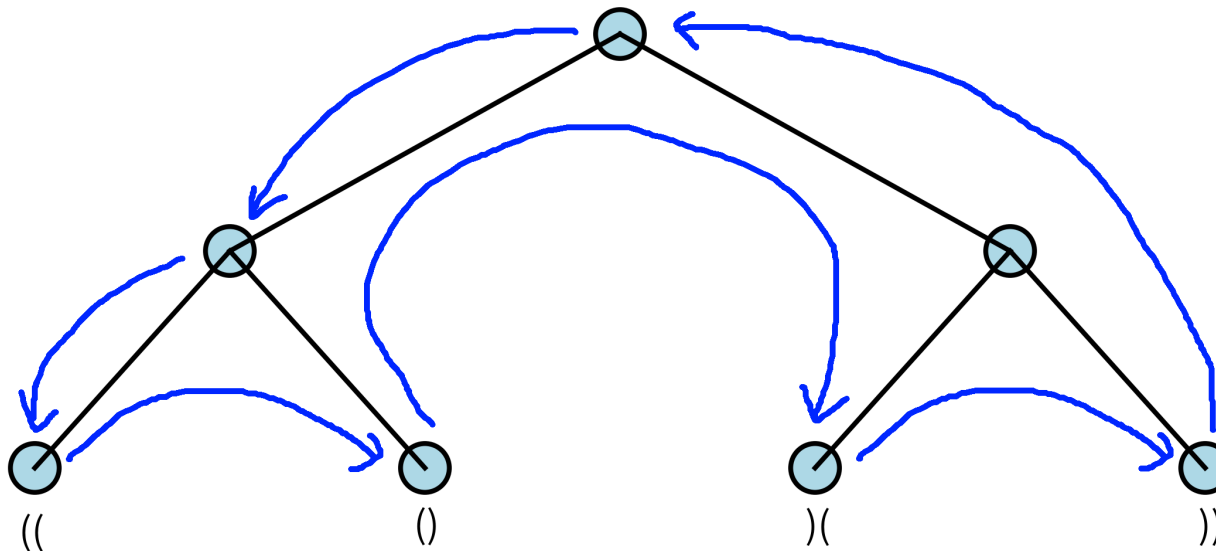
これは以下の図式 (二分木) を上からたどるすべての道筋として表現できる。



深さ優先探索

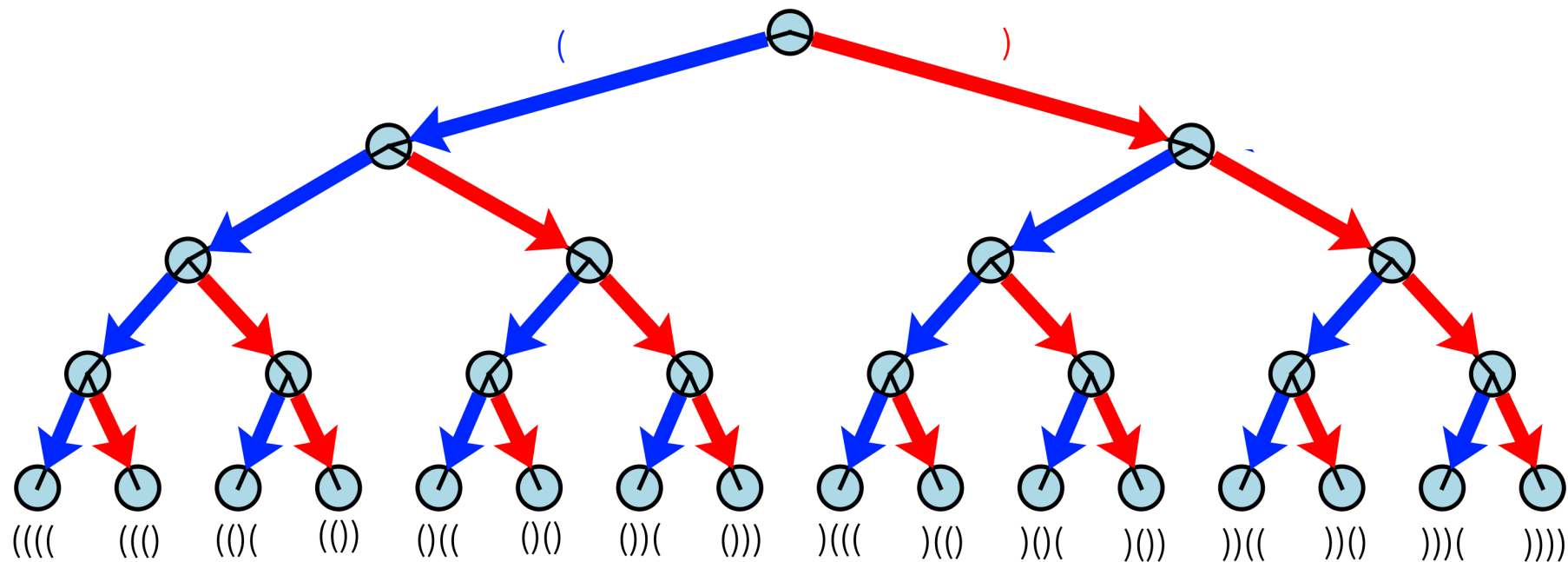
((→ () →)(→)) とたどる順序は、以下のように表現できる。

「最深部まで一気に進み、深いところから順に探索する」というこの探索を、**深さ優先探索 (Depth-First Search)** という。



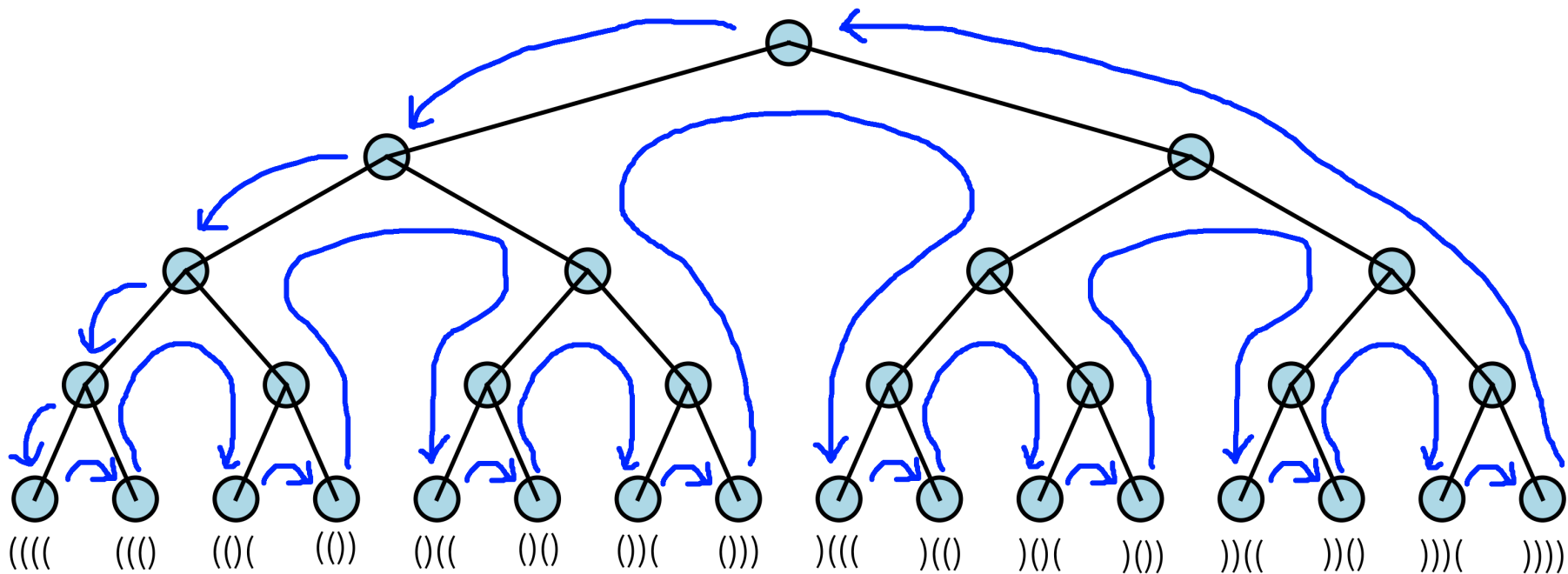
深さ優先探索

$n = 4$ の場合でも同様に表現できる。



深さ優先探索

$n = 4$ の場合でも同様に表現できる。



枝刈り

よく観察してみると、以下の条件があることが分かる。

- 開き括弧 "(" と閉じ括弧 ")" の個数は同じでなければならない。
- 左からそれぞれの個数を数えて、閉じ括弧のほうが多い箇所があるとNG。

例.

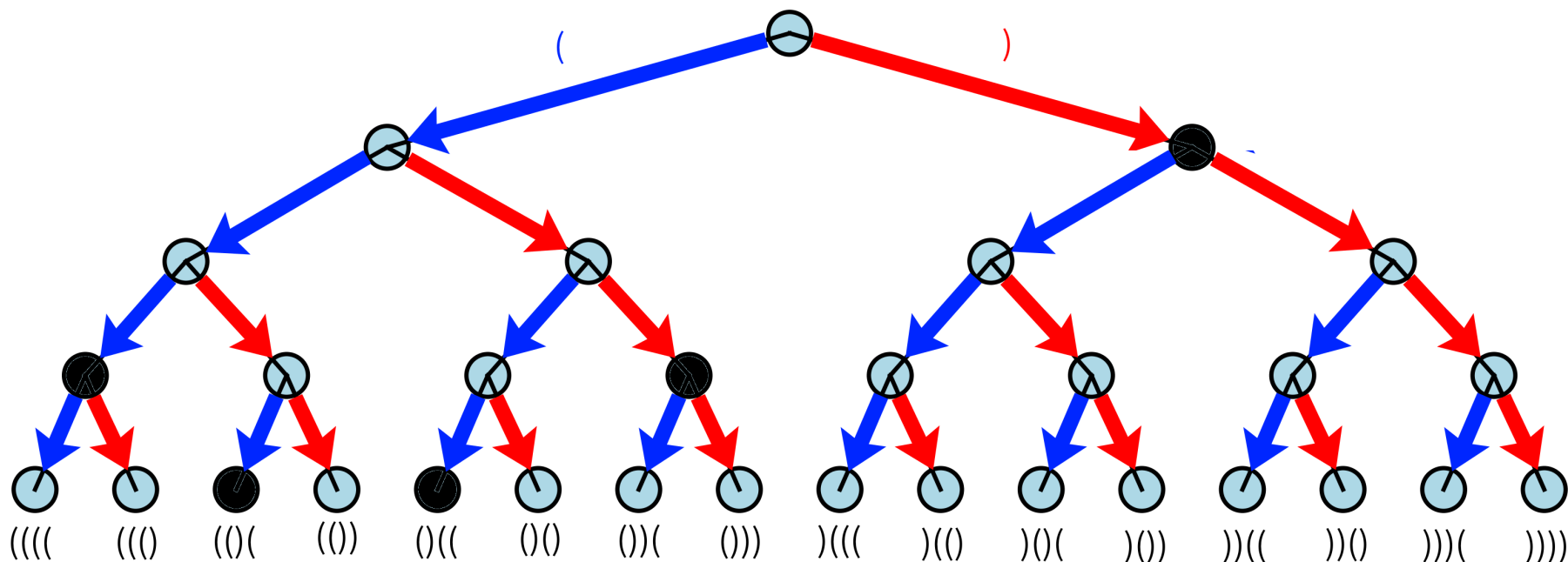
- `()()()(` :
開き括弧が5個、閉じ括弧が3個なのでNG
- `((()))(` :
5個目の位置 `((()))` の時点で、開き括弧が2個、閉じ括弧が3個なのでNG

枝刈り

これらを利用して、一部の探索をスキップすることができる。

このような探索範囲のスキップを**枝刈り**と呼ぶ。

例 ($n = 4$). 以下の黒丸を通る道筋はNGなので、その先は探索不要。

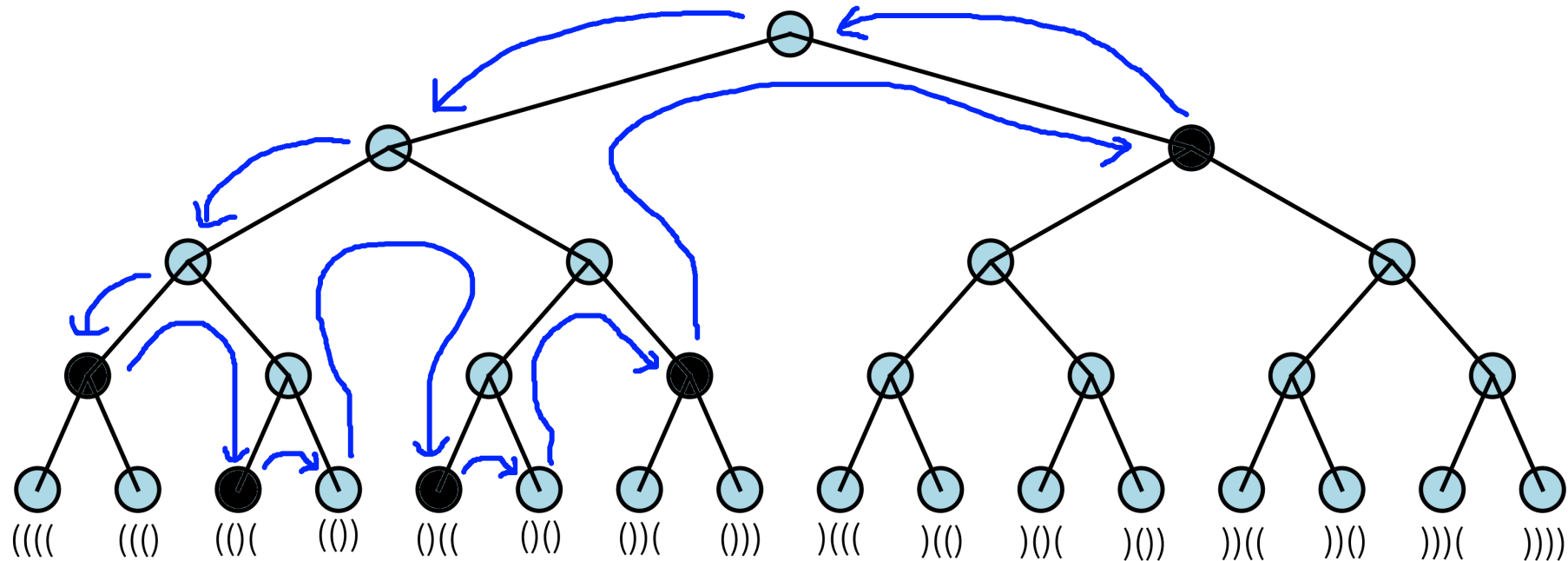


枝刈り

これらを利用して、一部の探索をスキップすることができる。

このような探索範囲のスキップを**枝刈り**と呼ぶ。

例 ($n = 4$). 以下の黒丸を通る道筋はNGなので、その先は探索不要。



飽和炭化水素の探索アルゴリズム

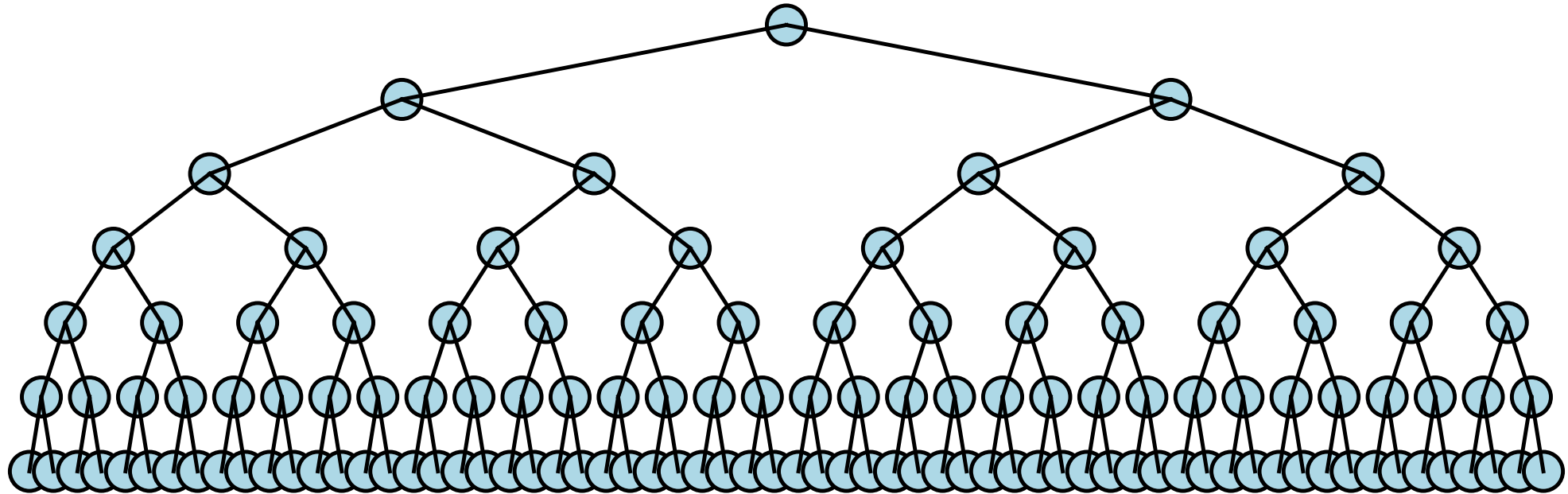
探索の図式化

- 飽和炭化水素の独立な変数は $\frac{N(N-1)}{2}$ 個 (上三角部分)。
- 各成分は 0 or 1 の値を取る。
→ 二分木で表現できる。

$$\begin{pmatrix} 0 & x_1 & x_2 & \dots & x_N \\ x_1 & 0 & x_{N+1} & \dots & x_{2N-1} \\ x_2 & x_{N+1} & 0 & \dots & x_{3N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_N & x_{2N-1} & x_{3N-3} & \dots & 0 \end{pmatrix}$$

探索の図式化

例 ($N = 4$). $\frac{4 \cdot (4-1)}{2} = 6$ 回の枝分かれがある。



探索の効率化

- 依然として探索範囲が $2^{N(N-1)/2}$ もあり、大きすぎる。
 - うまく枝刈りしたい。
 - 正解が満たすべき必要条件を考える。
- 省きたいもの
 - 連結じゃないもの
 - 次数が4を超えるもの
 - すでに見つけたものと同値なもの

探索の効率化

- 連結じゃないものを省きたい
 - 一番最後の枝分かれて連結になる場合がある。
 - 途中段階で省く条件が複雑。
- 次数が4を超えるものを省きたい
 - 途中段階で既に次数が 4 を超えるなら省ける。 **枝刈りチャンス！**
- すでに見つけたものと同値なものを省きたい
 - 同値なものたちのうち、標準的な代表 (標準形) を選びたい。
 - うまい条件を定めて、満たすものが必ず存在するようにしたい。
 - もし、探索の途中段階で標準形から外れるとわかれば **枝刈りチャンス！**

次数による枝刈り

隣接行列の一部分を取り出す。

$$\begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ * & \dots & * & 0 & x_k & x_{k+1} & \dots & x_{k+M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

- 探索の途中段階では、一部の要素のみが確定で、ほかは未定。
- $x_k, \dots, x_{k+\ell}$ のみが確定しているとき、それらの和が4を超えるなら、未定部分をどう定めても必ず次数は4を超える
→ 枝刈り！

飽和炭化水素の標準形

同値な隣接行列 = 頂点の番号のつけかえで一致する

- ある特殊な番号の付け方でのみ満たす条件を考えたい
- 頂点の間に何らかの順序を定めて、整列した番号付けにする
- 頂点の数値化を考えて、その数値が整列するような条件にする

飽和炭化水素の標準形

数値化案1: 次数

- 頂点 0 の次数が最大、頂点 ($N - 1$) の次数が最小になるようにする。
- 必ずこの番号付けは存在する。
- 同じ次数の部分については、複数の番号付けの候補がある。
- 次数による枝刈りを自動的に行える。

飽和炭化水素の標準形

数値化案2: 隣接行列の行を二進法の桁とみなして数値化

$$(a_0 \quad a_1 \quad \dots \quad a_{N-1}) \rightarrow \sum_{n=0}^{N-1} a_n \cdot 2^n = a_0 + a_1 \cdot 2 + \dots + a_{N-1} \cdot 2^{N-1}$$

- 頂点 0 で最大、頂点 $(N - 1)$ で最小になるようにする。
- この数値は番号の付け方に依存して決まる。
 - 整列させようとして、番号を入れ替える間にも数値が変わる。
 - 整列したものが存在するかどうか非自明。

飽和炭化水素の標準形

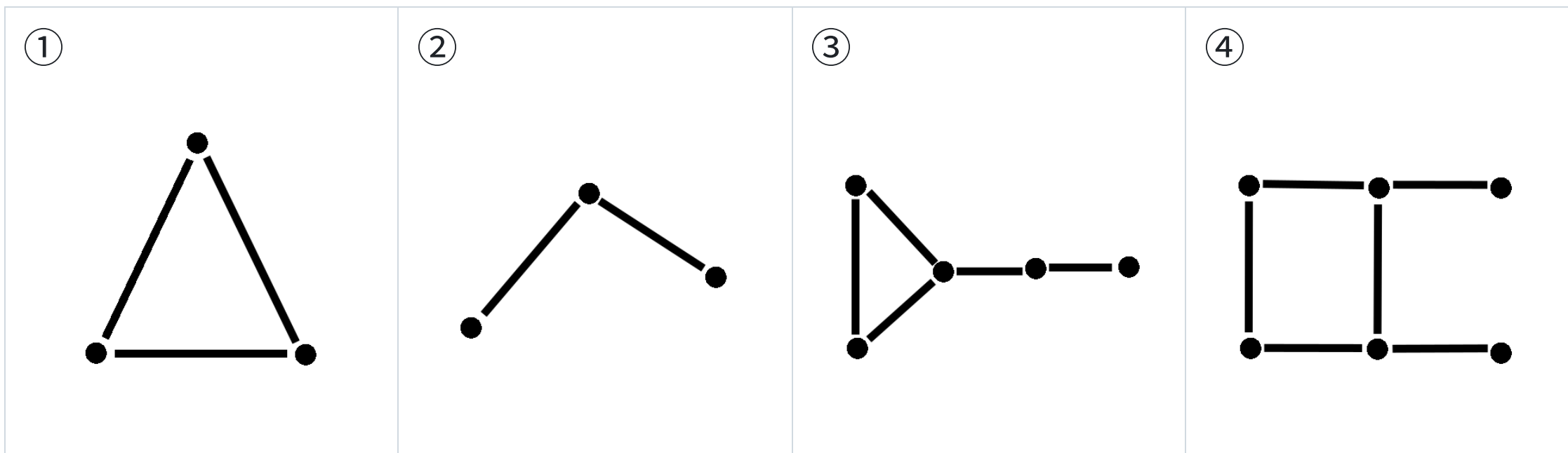
数値化案3: 案1と案2のハイブリッド

$$(a_0 \quad a_1 \quad \dots \quad a_{N-1}) \rightarrow \underbrace{\sum_{n=0}^{N-1} a_n \cdot 2^n}_{\text{案2の数値}} + \underbrace{\sum_{n=0}^{N-1} a_n \cdot 2^N}_{(\text{次数}) \cdot 2^N}$$

- 案1を優先し、決まらない部分は案2で整列する方法。
- やはり番号付けに依存して決まるので、整列されたものが存在するかは非自明。
- 数値化案1よりは限定できる。
- 次数を使った枝刈りも自動的にできる。
- 実験的に $N \leq 10$ では存在が確かめられたので採用。

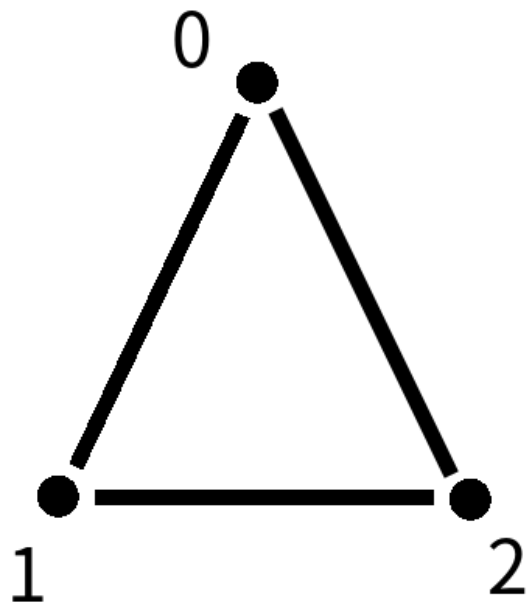
飽和炭化水素の標準形

例題. 以下の各飽和炭化水素に対応する、標準形の隣接行列を求めよ。
(重みはすべて1なので省略)



飽和炭化水素の標準形

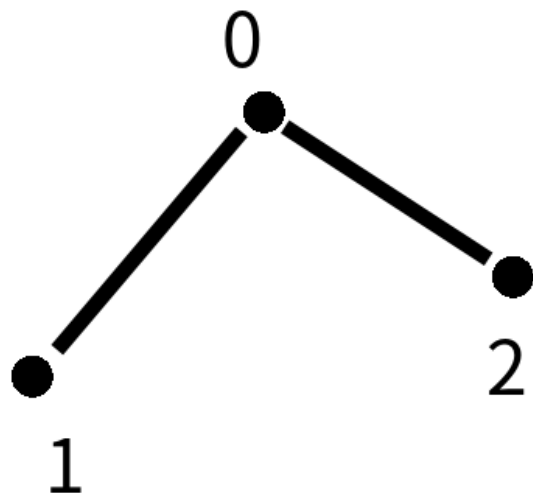
解答. ①



$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

飽和炭化水素の標準形

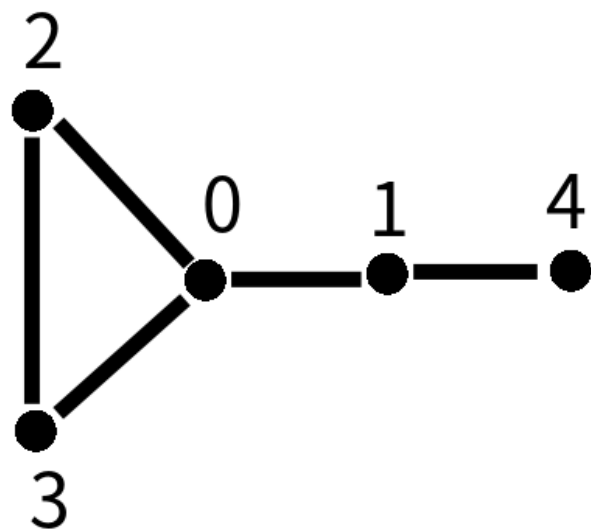
解答. ②



$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

飽和炭化水素の標準形

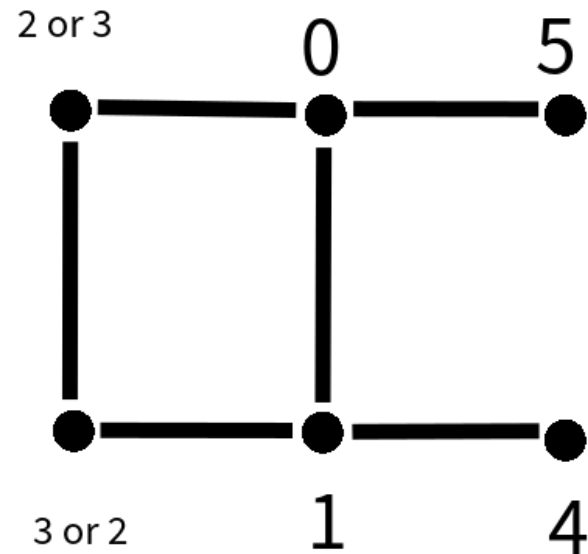
解答. ③



$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

飽和炭化水素の標準形

解答. ④



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ or } \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

飽和炭化水素の標準形

予想. (V, E) を連結な無向グラフとする。 $N = \#V$ とおく。各 $v \in V$ に対し、 v の次数を $d(v)$ とする。ある番号付け $\phi: V \rightarrow \{0, \dots, N-1\}$ であって、以下を満たすものが存在する。

- 各 $v \in V$ に対して $f(v) = \sum_{v' \in V} \chi(v, v')(2^{\phi(v')} + 2^N)$ とするとき、
$$f \circ \phi^{-1}(0) \geq f \circ \phi^{-1}(1) \geq \dots \geq f \circ \phi^{-1}(N-1)$$

が成り立つ。ただし

$$\chi(v, v') = \begin{cases} 1 & \text{for } (v, v') \in E \\ 0 & \text{for } (v, v') \notin E \end{cases}$$

もし証明できたら教えてください。

標準形を使った枝刈り

- 上半三角部分が確定している行は、その行の左端の部分も確定している。
- 未定の要素を 0 とみなして計算した行の値は、確定が増えても減ることはない。
 - 最後の確定行の値以上である、という条件は常に満たす必要がある。
 - **探索の途中で枝刈りチャンス！**

例. $N = 5$ で x_5 まで確定した場合と x_7 まで確定した場合

$$\begin{pmatrix} 0 & x_1 & x_2 & x_3 & x_4 \\ x_1 & 0 & x_5 & ? & ? \\ x_2 & x_5 & 0 & ? & ? \\ x_3 & ? & ? & 0 & ? \\ x_4 & ? & ? & ? & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & x_1 & x_2 & x_3 & x_4 \\ x_1 & 0 & x_5 & x_6 & x_7 \\ x_2 & x_5 & 0 & ? & ? \\ x_3 & x_6 & ? & 0 & ? \\ x_4 & x_7 & ? & ? & 0 \end{pmatrix}$$

同値なものの削減

- 標準形が存在することは分かったが、一意ではない。
→ 同値なものを削減する処理は依然として必要。
- 見つかったものの任意の番号の付け替えを作って保持しておく。
- 新たな候補が保持したものに含まれていればボツ。

計算時間の測定

先行研究よりある程度速くなった。

ケース名	N=2~7	N=2~8	N=2~9	N=2~10
先行研究	-	2分半	17分	96時間43分
枝刈り後	1.69 秒	1分48秒	-	-

ただし先行研究といろいろな条件が異なる。

- 言語の違い (Python vs Rust)
- マシンの違い
- アルゴリズムの違い (枝刈り以外では先行研究のほうがかなり工夫している)
→ まだまだ高速化の余地がある。

まとめ

まとめ

- 炭化水素はグラフと見なせる。
 - 隣接行列を列挙する問題にすれば、コンピュータで扱いやすい。
- 深さ優先探索 + 枝刈りで、高速に列挙できる。
 - 標準形をうまく設定して、探索範囲を削減。
- 証明できない予想が生えたので、誰か証明してほしい。

紹介できなかった高速化手法

この講演で紹介したのはごく一部。

『炭化水素列挙 RTA』では、以下の工夫を行っている。

- Rust のビルドオプションのチューニング
- 隣接行列のビット表現
- 不変量を用いて対称性を下げる工夫
- 行列同士の同値判定の高速化
 - 対称性の情報の流用
 - 同値な行列の生成の高速化
- 並列化

すべての高速化を適用した場合

書籍で紹介するすべての高速化を適用すると、劇的に速くなる。

ケース名	N=2~7	N=2~8	N=2~9	N=2~10
先行研究	-	2分半	17分	96時間43分
枝刈り後	1.69 秒	1分48秒	-	-
最終結果	-	-	-	0.25秒

約140万倍の高速化！

宣伝コーナー (再)

技術書典19にて右の電子書籍を販売します。

- イベント期間: 2025/11/15-30
(イベント終了後も販売予定)
- 場所: オンライン
- 価格: 1000円 (予定)

ぜひお買い求めください！



炭化水素列挙
RTA

すむーずぷりんちゃん 著

FIN