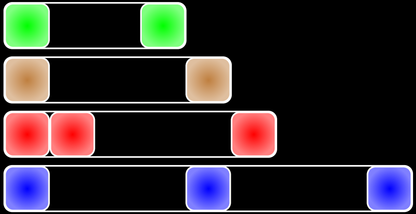
対数的な幅を持つ疎行列圧縮のNP完全性

坂内英夫[1]　後藤啓介[2]　神田峻介[2]　クップルドミニク[1,3]

[1]: 東京医科歯科大学, [2]: 株式会社リーガルオンテクノロジーズ, [3]: 山梨大学

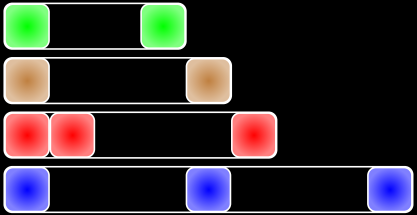


夏のLAシンポジウム2024 [15]

問題の設定

入力

- n 個の1次元ポリオミノ
- ポリオミノには隙間があってもよい



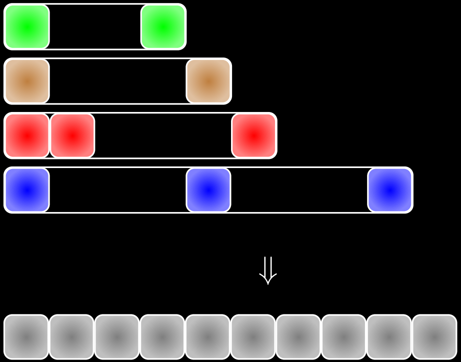
目標

- ポリオミノを1次元結合ポリオミノにする
- 隙間を埋めることができるが、埋めたブロックは重なってはならない
- 目的: 最短の結合ポリオミノを構築

問題の設定

補題
隙間のない結合ポリオミノは解である

証明.
ブロックは重ならないから □



決定問題

MINLENGTH すべてのポリオミノを長さ k の結合ポリオミノに結合できるか?

MAXSHIFT すべてのポリオミノの最初のブロックが最初の列に配置された場合、最大シフトが k 以下の結合ポリオミノを作成できるか?

しかし、MAXSHIFT はすでに Sparse Matrix Compression (SMC) problem として研究されていた！

Garey+'79 は $k \geq 2$ の場合に SMC が NP 困難であることを示した。
Garey+'79 の既存研究を以下の段階に紹介する：

- SMC の定義
- 3-COLORING の定義
- 3-COLORING から SMC への帰着

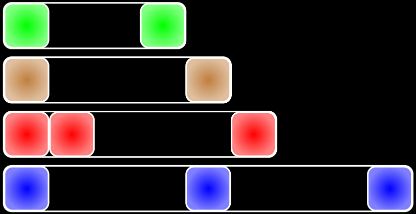
問題 (SMC, [Garey+'79, Chapter A4.2, Problem SR13])

入力：
- $n \times \ell$ 行列 $A[1..n][1..\ell]$、n 行 ℓ 列として、すべての $i \in [1..n]$, $j \in [1..\ell]$ に対して $A[i][j] \in \{0,1\}$ の要素を持つ
- 整数 $k \in [0..\ell \cdot (n-1)]$

目標: 以下の2つが存在できるかどうかを調べる:
- 整数配列 $C[1..\ell + k]$、$\forall i \in [1..\ell + k]$ について $C[i] \in [0..n]$
- シフト関数 $s : [1..n] \to [0..k]$ が存在し、$\forall i \in [1..n], \forall j \in [1..\ell]$ について $A[i][j] = 1 \Leftrightarrow C[s(i) + j] = i$ が成り立つ
- $A[0][j] = 0 \ \forall \ j$ であると仮定し、$C[i] = 0$ を設定することで、この要素が未割り当てであることをモデル化する

応用：
- 行列の圧縮 [Ziegler'77]
- 探索トライの実装 [Tarjan,Yao'79]
- ブルームフィルタ [Chang,Wu'91]

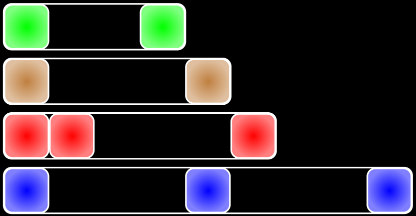
ポリオミノから行列へ



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

- $k = 2$ に対して、 3-COLORING を MAXSHIFT に帰着できる [Garey'79]

ポリオミノから行列へ



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

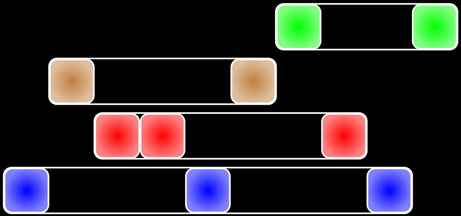
解答：

- $s = [6, 1, 2, 0]$
- $C = [4, 2, 3, 3, 4, 2, 1, 3, 4, 1]$

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

- $k = 2$ に対して、 3-COLORING を MAXSHIFT に帰着できる [Garey'79]

ポリオミノから行列へ



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

解答：

- $s = [6, 1, 2, 0]$
- $C = [4, 2, 3, 3, 4, 2, 1, 3, 4, 1]$

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

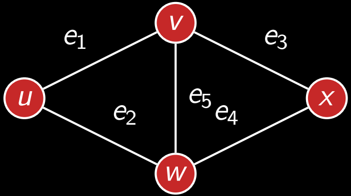
- $k = 2$ に対して、 3-COLORING を MAXSHIFT に帰着できる [Garey'79]

3-COLORING

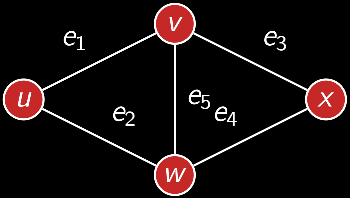
- 入力:
  - グラフ $G = (V, E)$
  - $n = |V|$
  - $m = |E|$
- 目的:
  - 各頂点 v に色 c を割り当てる
  - ただし $c \in \{$ 赤, 緑, 青 $\}$、かつ、v の隣接する頂点と v の色が異なる

例

- 頂点 $V = \{u, v, w, x\}$
- 辺 $E = \{e_1, \ldots, e_4\}$
- 最初はすべての頂点が赤で塗られているが、これは解ではない



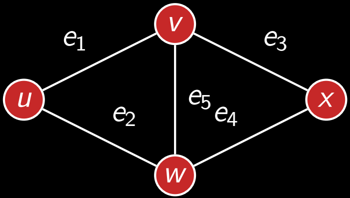
3-COLORING to MAXSHIFT



- 行数が n、列数が $3m$ の行列 A を構築する
- 3つの列ごとを辺に、各行を頂点に対応させる
- 目的：Aに対してMAXSHIFTを解く

$$
A = \begin{array}{c|ccccccccccccccc}
 & \multicolumn{3}{c}{\underline{e_1}} & \multicolumn{3}{c}{\underline{e_2}} & \multicolumn{3}{c}{\underline{e_3}} & \multicolumn{3}{c}{\underline{e_4}} & \multicolumn{3}{c}{\underline{e_5}} \\
u & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
w & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
x & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

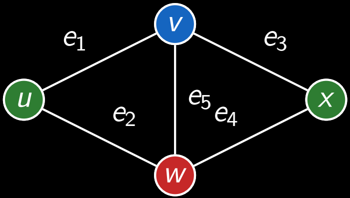
3-COLORING to MAXSHIFT



- uとxの行を1シフトし、vの行を2シフトする
- 各列に対して、最大1つの1を持つ行があるような行列Bを得る
- すべての行の最大シフトが2以内であるため、このグラフは3彩色可能である

$$
A = \begin{array}{c|ccc|ccc|ccc|ccc|ccc}
 & \multicolumn{3}{c}{e_1} & \multicolumn{3}{c}{e_2} & \multicolumn{3}{c}{e_3} & \multicolumn{3}{c}{e_4} & \multicolumn{3}{c}{e_5} \\
u & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
w & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
x & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}
\rightarrow
B = \begin{array}{c|ccc|ccc|ccc|ccc|ccc}
 & \multicolumn{3}{c}{e_1} & \multicolumn{3}{c}{e_2} & \multicolumn{3}{c}{e_3} & \multicolumn{3}{c}{e_4} & \multicolumn{3}{c}{e_5} \\
u & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
w & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}
$$

3-COLORING to MAXSHIFT



- Bは色の割り当てを符号化しており、次のようにグラフに描画できる
- シフトは色を表し、0：赤、1：緑、2：青
- 2つ以上のシフトが必要な場合、グラフは3彩色不可能である

$$
A = \begin{array}{c|ccccccccccccccc}
 & \multicolumn{3}{c}{e_1} & \multicolumn{3}{c}{e_2} & \multicolumn{3}{c}{e_3} & \multicolumn{3}{c}{e_4} & \multicolumn{3}{c}{e_5} \\
u & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
w & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
x & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{array}
\to
B = \begin{array}{c|ccccccccccccccc}
 & \multicolumn{3}{c}{e_1} & \multicolumn{3}{c}{e_2} & \multicolumn{3}{c}{e_3} & \multicolumn{3}{c}{e_4} & \multicolumn{3}{c}{e_5} \\
u & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
w & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}
$$

まとめ

- 3-COLORING は NP 困難であるから、$k \geq 2$ のとき MAXSHIFT も NP 困難である
- 証明の弱点：
  - ポリオミノの長さが制限されていない
  - 証明は MINLENGTH には適用できなさそう
- 新しいアイデア：ハミルトン路からの帰着

ハミルトン路から MINLENGTH へ

本研究 [Bannai+'24]:

- ハミルトン路から最短共通超文字列 (SCS) への帰着を利用 [Gallant+'80]
- ポリオミノを文字列と解釈することで帰着を応用
- 結果：幅 $\ell = \Omega(\lg n)$ のとき MINLENGTH と SMC が NP 困難であることを示す

RDHP: 制約付き有向ハミルトン経路問題

問題 (Gallant+'80)

入力:

- 有向グラフ $G = (V, E)$
- 指定された始点 $s \in V$ と終点 $t \in V$ であり、
  - s は入次数が 0 であり,
  - t は出次数が 0 であり、
  - t 以外のすべての頂点は出次数が 1 より大きい

目的: s から t までの経路が存在し、各頂点をちょうど一度通るかどうかを答える

- RDHPはNP完全である [Gallant+'80]

SCS：最短共通超文字列

問題

入力:

- n 個の文字列の集合 $\mathcal{T}$（各文字列の長さは ℓ）
- 整数 $k \geq \ell$

目標：文字列 $S[1..k]$（長さ k）が存在し，すべての $T \in \mathcal{T}$ について，T が S に現れるかどうかを確認する

- SCS は，RDHP からの帰着により NP 困難である [Gallant+'80]

定理 (Gallant+'80)

任意の整数 $\ell \geq 3$ に対し、すべての入力文字列が長さ ℓ である場合でも、SCS は NP 困難であることが示される。また、入力文字列集合 $\mathcal{T}$ 内の各文字が最大 1 回しか出現しないとしても、この問題はNP困難である。

証明.

RDHP を SCS に帰着で証明できる（スライドがあるが、割愛) □

MINLENGTH への帰着を行うために、SCS を以下のように変更：

- 入力文字列集合 $\mathcal{T}$ はポリオミノの集合と整数 $k \geq 0$ とする
- i-番目のポリオミノは、行列 A のように、バイナリ文字列 $\in \{0, i\}^*$に写像される
- 文字 0 は任意の文字とマッチする
- 目的：長さ k の文字列 $S[1..k]$が存在し、$\forall T \in \mathcal{T}$、$T$ が S の部分文字列にマッチするかどうかを確認

証明の概略

障害：

- 文字列 S にマッチする正規文字列 T がすべてのマッチした位置をカバーする
- その一方で、ワイルドカード 0 はより短い表現を許可する
- 縮小のため、そのゆとりを取り除きたい

アイデア：

- ビットレベルで作業し、ビット否定を使用する（$\bar{1} = 0, \bar{0} = 1$）

文字列の由来は？

- リマインダー：入力はグラフ $G = (V, E)$
- 目的：各頂点 $v \in V$ を二進文字列としてモデル化する

目的のため、以下の命令で写像 $\psi : V \to \{0, 1\}^*$ を定義する：

1. v の二進表現を b とすると、 b を $\mathcal{O}(\lg n)$ ビットで表現できる ($|V| = n$)
2. Thue–Morse ϕ 変換を b に適用する ($\phi(0) = 01$ および $\phi(1) = 10$)
3. 魔法のビット定数 $\alpha \in \{0, 1\}^*$ を b に先頭に追加する

例

$x \in V$	$b(x)$	$\phi(b(x))$	$\psi(x) = \alpha \cdot \phi(b(x))$	$\overline{\psi(x)} = \overline{\alpha \cdot \phi(b(x))}$
v_1	00000	01010101	01000011 01010101	10111100 10101010
v_2	00001	01010110	01000011 01010110	10111100 10101001
v_3	00010	01011001	01000011 01011001	10111100 10100110
v_4	00011	01011010	01000011 01011010	10111100 10100101

なぜ ϕ か？

- ϕ は 0 と 1 のビットを均等に分配するから
- $\Rightarrow$ α が不均等な分布を持つ場合、$\phi(b)$ は α と一致しない

α はどのように機能するのか？

ブロックの配置

$\alpha = 01000011$ を選ぶ。

例

- $b = 1110$とすると、vのビット表現はこれになります
- $\phi(b) = 10101001$
- $\psi(b) = \alpha \cdot \phi(b) = 01000011 \cdot 10101001$

補題

αと重なる他の文字列を見つけることはできません。ただし、同じ位置から開始する$\bar{\alpha}$を除く。

$\alpha =$ 0 1 0 0 0 0 1 1

ブロックの配置

$\alpha = 01000011$ を選ぶ。

例

- $b = 1110$とすると、vのビット表現はこれになります
- $\phi(b) = 10101001$
- $\psi(b) = \alpha \cdot \phi(b) = 01000011 \cdot 10101001$

補題

αと重なる他の文字列を見つけることはできません。ただし、同じ位置から開始する$\bar{\alpha}$を除く。

$$
\begin{array}{rcccccccccc}
 & \alpha = & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
 & & & & & & & & \updownarrow & \\
\alpha = & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 &
\end{array}
$$

ブロックの配置

$\alpha = 01000011$ を選ぶ。

例

- $b = 1110$とすると、vのビット表現はこれになります
- $\phi(b) = 10101001$
- $\psi(b) = \alpha \cdot \phi(b) = 01000011 \cdot 10101001$

補題

αと重なる他の文字列を見つけることはできません。ただし、同じ位置から開始する$\bar{\alpha}$を除く。

$$\alpha = \begin{array}{cccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

$$\bar{\alpha} = \begin{array}{cccccccc} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{array}$$

ブロックの配置

$\alpha = 01000011$ を選ぶ。

例

- $b = 1110$とすると、vのビット表現はこれになります
- $\phi(b) = 10101001$
- $\psi(b) = \alpha \cdot \phi(b) = 01000011 \cdot 10101001$

補題

αと重なる他の文字列を見つけることはできません。ただし、同じ位置から開始する$\bar{\alpha}$を除く。

$$
\begin{array}{rcccccccccc}
\alpha = & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & & \\
 & & & & & & & \updownarrow & \updownarrow & & \\
\bar{\alpha} = & & & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

ブロックの配置

$\alpha = 01000011$ を選ぶ。

例

- $b = 1110$とすると、vのビット表現はこれになります
- $\phi(b) = 10101001$
- $\psi(b) = \alpha \cdot \phi(b) = 01000011 \cdot 10101001$

補題

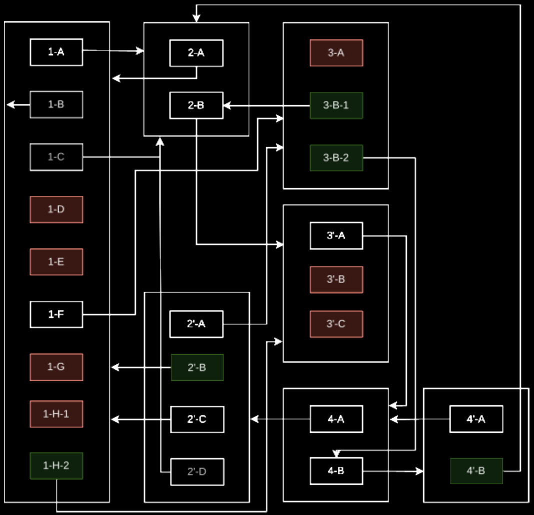
αと重なる他の文字列を見つけることはできません。ただし、同じ位置から開始する$\bar{\alpha}$を除く。

$$
\begin{array}{rccccccccccc}
\alpha = & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & & & \\
 & & & & & & & \updownarrow & \updownarrow & & & \\
\bar{\alpha} = & & & & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

鬼は細部に宿る

以下のように続ける（概念）：

- 文字列の組み合わせが複雑になる
- 4つの主要なケースと多数のサブケースの大規模なケース分析



証明の要点

RDHP から SCS への Gallant+'80 の帰着：

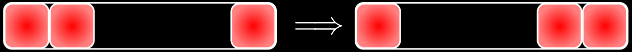
- 各頂点に長さ3の文字列を作成する
- ハミルトン路が存在することと、最短共通超文字列が特定の長さであることが同値であることを示す

本研究の帰着：

- 頂点を ψ で $\Theta(\lg n)$ ビットの長さの頂点署名で拡張する
- 同じ証明ステップに従うが、ワイルドカード '0' とのマッチングのために追加のケース分析が必要

未解決問題

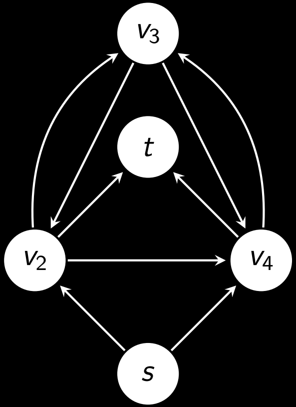
- 幅 $\ell \in \omega(\lg\lg n) \cap o(\lg n)$ の場合はどうなるか？
- 解が一意である条件は？
  - 隙間がないポリオミノの場合は明らかに一意ではない！
  - ポリオミノはジグソーパズルのようにユニークなペア接続子を持つことができる
- 拡張：水平反転や循環シフトを許可する



ご清聴ありがとうございました

RDHP から SCS の帰着

- $V = \{v_1 = s, v_2, v_3, v_4, v_5 = v_n = t\}$ 頂点集合
- $n = 5$ 超点数
- $m = 9$ 変数
- 目的： s から t までのハミルトン路を計算
- 答えは：$s \to v_2 \to v_3 \to v_4 \to t$



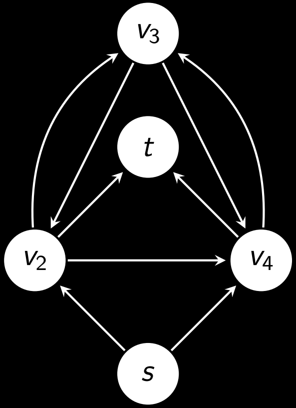
定義

各ノード v_i に対して $i \in \{1, \ldots, n-1\}$、

- $\delta(i)$ をノード v_i の出次数を表す
- $c(i, j)$ を v_i の $((j \bmod \delta(i)) + 1)$ 番目の子ノードに対応するノード番号とする。
- 子供$(i) := \{v_{c(i,1)}, \ldots v_{c(i,\delta(i))}\}$ は v_i の子ノードの集合を表す

例：

- $\delta(1) = 2$, $\delta(2) = 3$, $\delta(3) = 2$, $\delta(4) = 2$, $\delta(5) = 0$.
- 子供$(1) = \{v_2, v_4\}$, 子供$(2) = \{v_3, v_4, t\}$, 子供$(3) = \{v_2, v_4\}$, 子供$(4) = \{v_3, t\}$, 子供$(5) = \emptyset$



A と B の定義

- アルファベット $\Sigma = \{\ddagger, \#, \$\} \cup (V \setminus \{s\}) \cup W$ を定義する
- ただし、 $W = \{w_1, \ldots, w_{n-1}\}$ は新しい文字
- $\ddagger$ と \$ は入力文字列集合内で一度だけ現れ、超文字列を開始および終了する

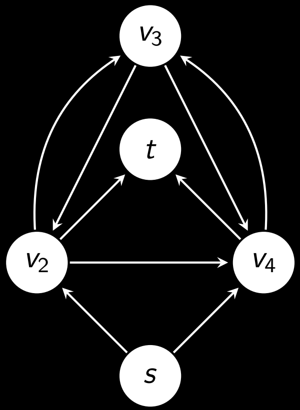
各 $v_{c(i,j)} \in$ 子供(i) に対して、文字列

- $A_{i,j} := w_i v_{c(i,j)} w_i \in \Sigma^3$、
- $B_{i,j} := v_{c(i,j)} w_i v_{c(i,j+1)} \in \Sigma^3$、
- $\mathcal{A}_i := \{A_{i,j} \mid v_{c(i,j)} \in \text{子供}(i)\} \cup \{B_{i,j} \mid v_{c(i,j)} \in \text{子供}(i)\}$

を定義する

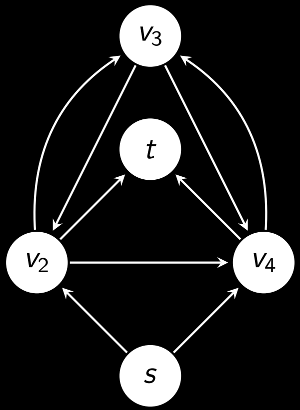
A と B の例

1. $\mathcal{A}_1$ は以下の文字列から成り立つ
   - $A_{1,1} = w_1 v_{c(1,1)} w_1 = w_1 v_2 w_1$
   - $A_{1,2} = w_1 v_{c(1,2)} w_1 = w_1 v_4 w_1$
   - $B_{1,1} = v_{c(1,1)} w_1 v_{c(1,2)} = v_2 w_1 v_4$
   - $B_{1,2} = v_{c(1,2)} w_1 v_{c(1,1)} = v_4 w_1 v_2$
2. $\mathcal{A}_2$ は以下の文字列から成り立つ
   - $A_{2,1} = w_2 v_{c(2,1)} w_2 = w_2 v_3 w_2$
   - $A_{2,2} = w_2 v_{c(2,2)} w_2 = w_2 v_4 w_2$
   - $A_{2,3} = w_2 v_{c(2,3)} w_2 = w_2 t w_2$
   - $B_{2,1} := v_{c(2,1)} w_2 v_{c(2,2)} = v_3 w_2 v_4$
   - $B_{2,2} := v_{c(2,2)} w_2 v_{c(2,3)} = v_4 w_2 t$
   - $B_{2,3} := v_{c(2,3)} w_2 v_{c(2,1)} = t w_2 v_3$



A と B の例

1. $\mathcal{A}_3$ は以下の文字列から成り立つ
   - $A_{3,1} = w_3 v_{c(3,1)} w_3 = w_3 v_2 w_3$
   - $A_{3,2} = w_3 v_{c(3,2)} w_3 = w_3 v_4 w_3$
   - $B_{3,1} := v_{c(3,1)} w_3 v_{c(3,2)} = v_2 w_3 v_4$
   - $B_{3,2} := v_{c(3,2)} w_3 v_{c(3,1)} = v_4 w_3 v_2$
2. $\mathcal{A}_4$ は以下の文字列から成り立つ
   - $A_{4,1} = w_4 v_{c(4,1)} w_4 = w_4 v_3 w_4$
   - $A_{4,2} = w_4 v_{c(4,2)} w_4 = w_4 t w_4$
   - $B_{4,1} := v_{c(4,1)} w_4 v_{c(4,2)} = v_3 w_4 t$
   - $B_{4,2} := v_{c(4,2)} w_4 v_{c(4,1)} = t w_4 v_3$



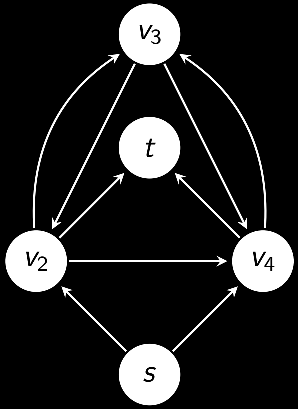
C の定義

- $C_1 := \ddagger\#w_1$、
- $C_i := v_i\#w_i \;\; \forall\, i \in \{2, \ldots, n-1\}$、
- $C_n := v_n\#\$ \in \Sigma^3$
- $\mathcal{C} := \{C_1, \ldots, C_n\}$

とする

例：

- $\mathcal{C} = \{\ddagger\#w_1, v_2\#w_2, v_3\#w_3, v_4\#w_4, t\#\$\}$



$\bowtie$ の定義

定義

2 つの文字列 X と Y に対して、X と $Y[\ell+1..]$ の連結を $X \bowtie Y$ と定義する。ここで、$Y[1..\ell]$ は X の接尾辞である最長の接頭辞である。

思い出し：

- $A_{i,j} := w_i v_{c(i,j)} w_i \in \Sigma^3$
- $B_{i,j} := v_{c(i,j)} w_i v_{c(i,j+1)} \in \Sigma^3$

例：

- $A_{i,j} \bowtie B_{i,j} = w_i v_{c(i,j)} w_i v_{c(i,j+1)} \in \Sigma^4$
- $B_{i,j} \bowtie A_{i,j+1} = v_{c(i,j)} w_i v_{c(i,j+1)} w_i \in \Sigma^4$

ハミルトン路と超文字列の関係

定理

G がsからtまでのハミルトン路を持つことと、文字列の集合 $\mathcal{T} = \mathcal{C} \cup \bigcup_{i=1}^{n-1} \mathcal{A}_i \subset \Sigma^3$ が長さ $2m + 3n$ の超文字列を持つことが同値であること

例：

- 定理によると、$\mathcal{T}$ は長さ $2m + 3n = 33$ の超文字列 S を持つ ($n = 5$, $m = 9$)
- 超文字列 S は、次のスライドのようにすべての頂点に対しる文字列 S_i で構築される（S_tを除く）
- 方針は、ハミルトン路のk番目の辺は(v_i, v_j)だとすると、
  - S_k は $A_{i,\cdot}$ から初めて、
  - $B_{i,j}$ で終わる、
  - すべて $A_{i,\cdot}$ と $B_{i,\cdot}$ を利用する

長い計算 1/2

$$
\begin{aligned}
S_1 &= A_{1,1} \bowtie B_{1,1} \bowtie A_{1,2} \bowtie B_{1,2} \\
&= w_1 v_2 w_1 \bowtie v_2 w_1 v_4 \bowtie w_1 v_4 w_1 \bowtie v_4 w_1 v_2 \\
&= w_1 v_2 w_1 v_4 w_1 v_2
\end{aligned}
$$

$$
\begin{aligned}
S_2 &= A_{2,1} \bowtie B_{2,1} \bowtie A_{2,2} \bowtie B_{2,2} \bowtie A_{2,3} \bowtie B_{2,3} \\
&= w_2 v_3 w_2 \bowtie v_3 w_2 v_4 \bowtie w_2 v_4 w_2 \bowtie v_4 w_2 t \bowtie w_2 t w_2 \bowtie t w_2 v_3 \\
&= w_2 v_3 w_2 v_4 w_2 t w_2 v_3
\end{aligned}
$$

$$
\begin{aligned}
S_3 &= A_{3,2} \bowtie B_{3,2} \bowtie A_{3,1} \bowtie B_{3,1} \\
&= w_3 v_4 w_3 \bowtie v_4 w_3 v_2 \bowtie w_3 v_2 w_3 \bowtie v_2 w_3 v_4 \\
&= w_3 v_4 w_3 v_2 w_3 v_4
\end{aligned}
$$

長い計算 2/2

最後に、文字列 S は C_i と S_i を接着して構築される。

$$
\begin{aligned}
S := & C_1 \bowtie S_1 \bowtie C_2 \bowtie S_2 \bowtie C_3 \bowtie S_3 \bowtie C_4 \bowtie S_4 \bowtie C_5 \\
= & \ddagger \# w_1 \bowtie w_1 v_2 w_1 v_4 w_1 v_2 \bowtie v_2 \# w_2 \bowtie w_2 v_3 w_2 v_4 w_2 t w_2 v_3 \\
& \bowtie v_3 \# w_3 \bowtie w_3 v_4 w_3 v_2 w_3 v_4 \bowtie v_4 \# w_4 \bowtie w_4 t w_4 v_3 w_4 t \bowtie t \# \$ \\
= & \ddagger \# w_1 v_2 w_1 v_4 w_1 v_2 \# w_2 v_3 w_2 v_4 w_2 t w_2 v_3 \# w_3 v_4 w_3 v_2 w_3 v_4 \# w_4 t w_4 v_3 w_4 t \# \$
\end{aligned}
$$

確かに、S の長さは 33 である。