理論計算機科学特論 (2026 年前学期)

計算複雑性の基礎

第 8 回

Savitch の定理：PSPACE = NPSPACE

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2026 年 6 月 2 日

最終更新：2026 年 6 月 9 日　22:56

スケジュール (前半)

1. 計算理論の復習 (4/7)
2. 時間計算量：P, NP, coNP (4/14)
3. 帰着と完全性：NP 完全 (4/21)
4. 領域計算量：L, NL, PSPACE (4/28)

$*$ 休み (祝日) (5/5)

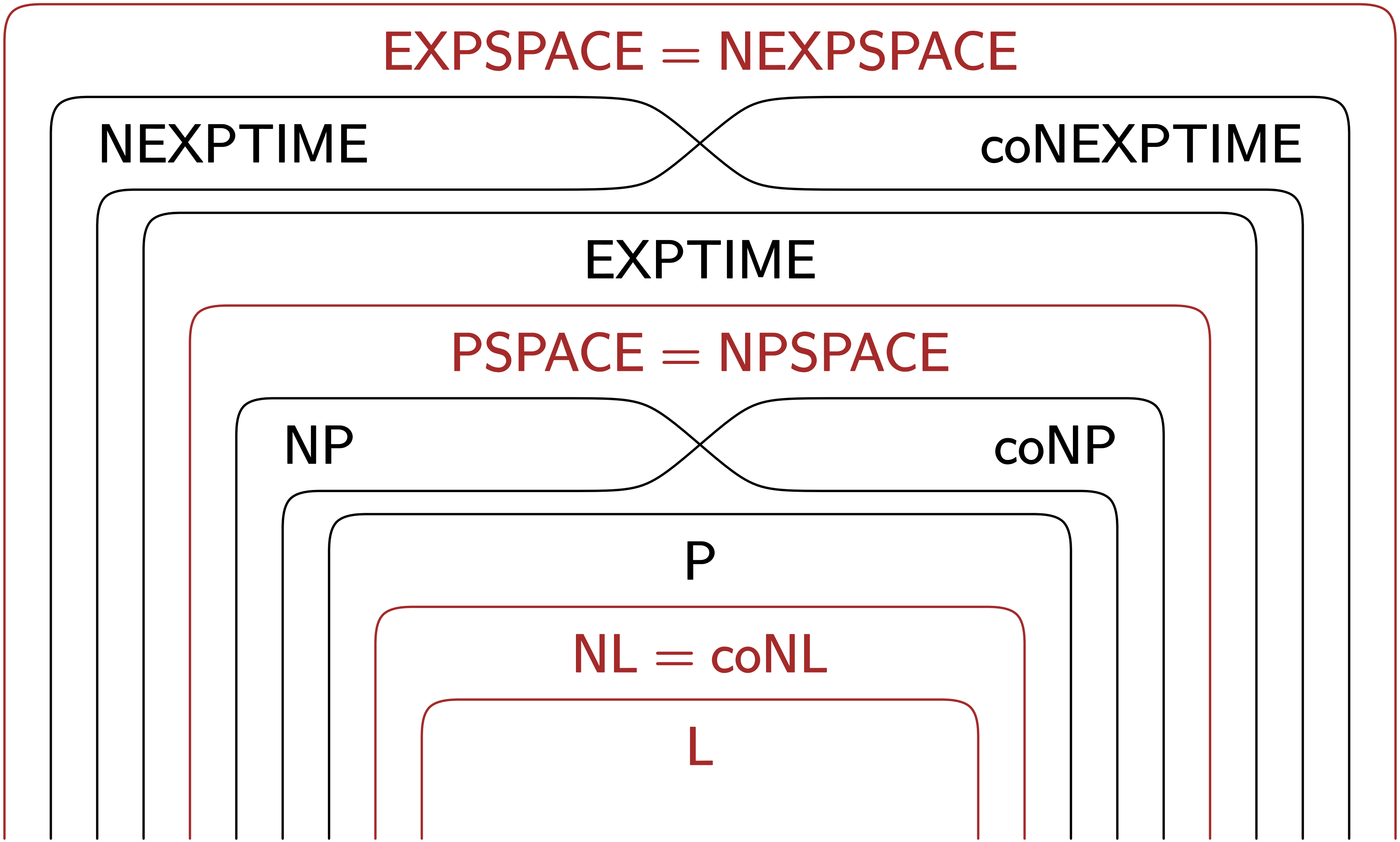
5. 時間と領域の関係：$\mathrm{P} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$ (5/12)
6. 階層定理：$\mathrm{P} \neq \mathrm{EXPTIME}$ (5/19)
7. Ladner の定理：$\mathrm{NP} - \mathrm{P} = \mathrm{NPC} \Rightarrow \mathrm{P} = \mathrm{NP}$ (5/26)

スケジュール (後半)

8. **Savitch の定理：** PSPACE = NPSPACE (6/2)
9. Immerman-Szelepcsényi の定理：NL = coNL (6/9)
10. 多項式階層：$\mathrm{P} = \mathrm{NP} \Rightarrow \mathrm{P} = \mathrm{PH}$ (6/16)
11. 交代性計算：AP = PSPACE (6/23)
12. 確率的計算：$\mathrm{P} \subseteq \mathrm{BPP} \subseteq \mathrm{PP}$, $\mathrm{NP} \subseteq \mathrm{PP}$ (6/30)
13. 対話証明系 (1)：$\mathrm{NP} \subseteq \mathrm{MA} \subseteq \mathrm{AM}$ (7/7)
14. 対話証明系 (2)：$\mathrm{IP} \subseteq \mathrm{PSPACE}$ (7/14)
15. 対話証明系 (3)：$\mathrm{PSPACE} \subseteq \mathrm{IP}$ (7/21)

$*$ 休み (授業のない日) (7/28)

復習：時間計算量と領域計算量



黒は時間複雑性クラス，茶は領域複雑性クラス

復習：アルゴリズムの領域計算量

P は判定問題，A は P を解くアルゴリズム

定義：アルゴリズムの領域計算量

アルゴリズム A の **領域計算量** (space complexity) とは，
符号長 n 以下の入力 I に対して A が使う作業領域量の
I に関する最大値 (単位：ビット)

$$s_A(n) = \max_{I\colon |I| \leq n} (A(I) \text{ が使う作業領域量 (ビット)})$$

注：これを A の **最悪領域計算量** ということがある
「領域」ではなく「空間」ということもある

ポイント

領域計算量は n の関数

復習：PSPACE と NPSPACE

定義：クラス PSPACE

クラス PSPACE とは,
多項式領域アルゴリズムで解ける判定問題全体のこと

領域計算量 $s_A(n)$ が n の多項式であるアルゴリズム A

定義：クラス NPSPACE

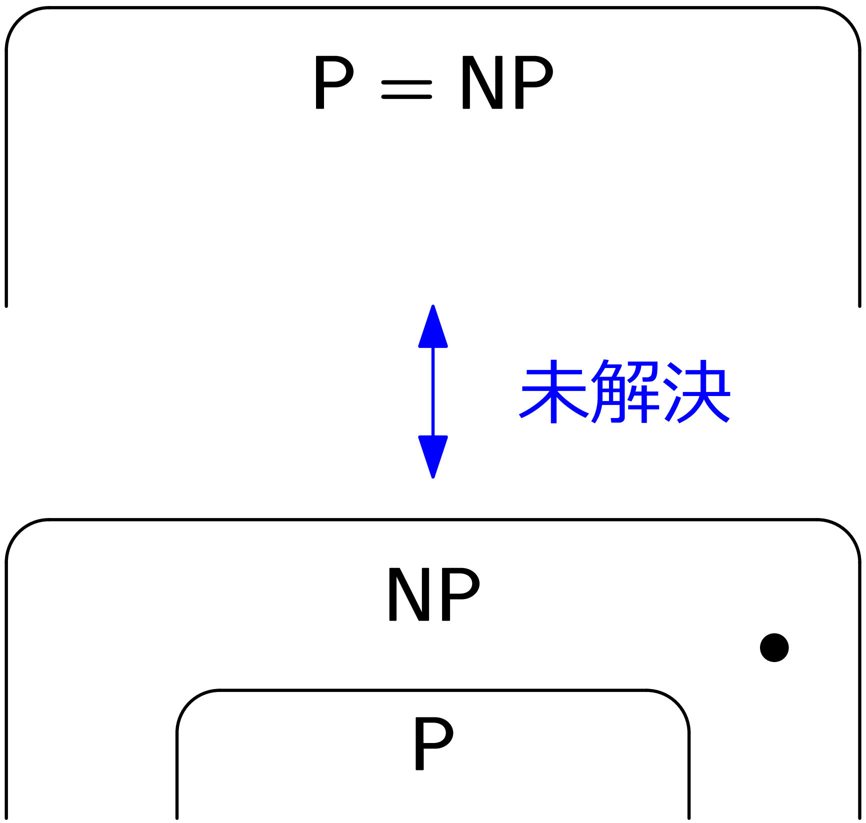
クラス NPSPACE とは,
非決定性のある多項式領域アルゴリズムで解ける
判定問題全体のこと

定義より：PSPACE ⊆ NPSPACE

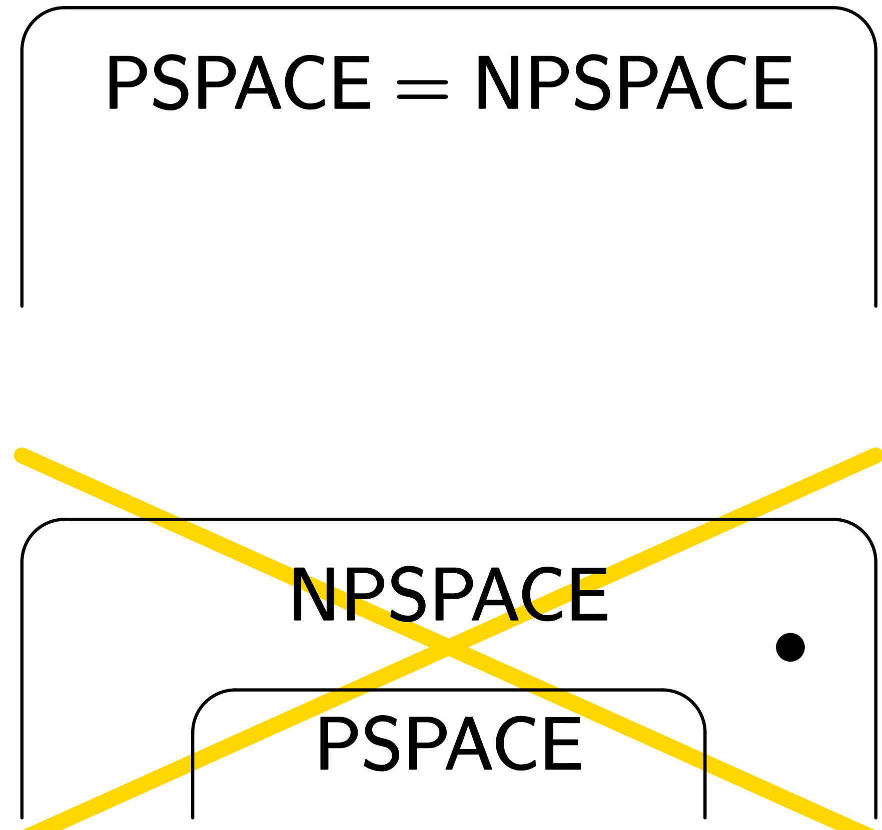
Savitch の定理 ('70)

PSPACE = NPSPACE

時間に対しては



領域に対しては



Savitch の定理：証明方針

証明の流れ

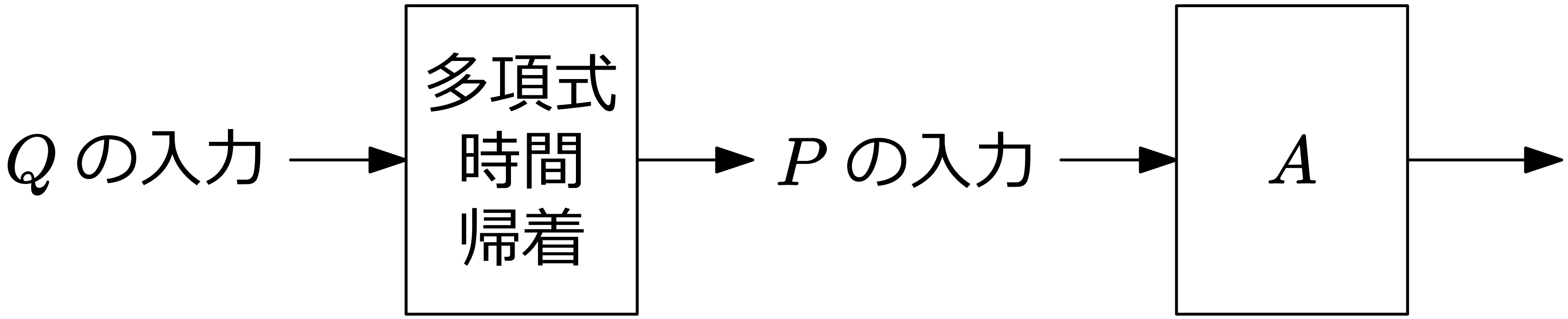
1. ある NPSPACE 完全問題 P を考える
2. P を解く多項式領域アルゴリズム A を作る

Savitch の定理：証明方針

証明の流れ

1. ある NPSPACE 完全問題 P を考える
2. P を解く多項式領域アルゴリズム A を作る

任意の問題 $Q \in$ NPSPACE に対して

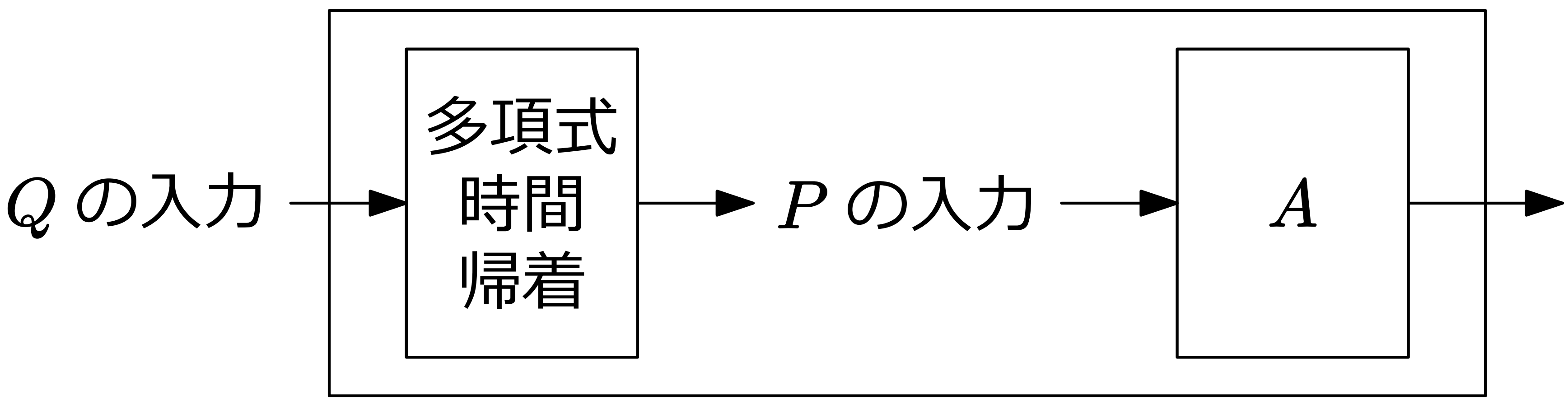


Savitch の定理：証明方針

証明の流れ

1. ある NPSPACE 完全問題 P を考える
2. P を解く多項式領域アルゴリズム A を作る

任意の問題 $Q \in$ NPSPACE に対して



Q を解く多項式領域アルゴリズム

$\therefore \quad Q \in$ PSPACE

本日の内容

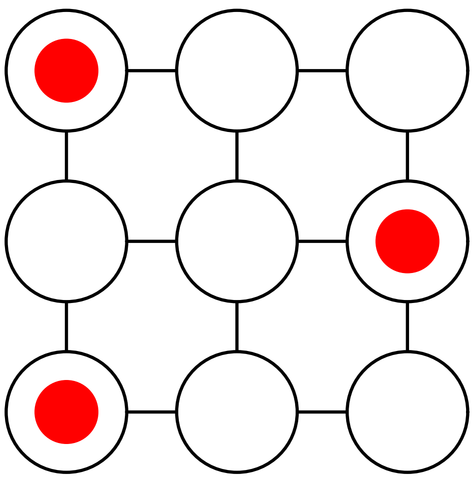
1. NPSPACE **完全問題の例**
2. Savitch の定理：証明
3. Savitch の定理：補足
4. Savitch の定理の適用例

独立集合遷移問題

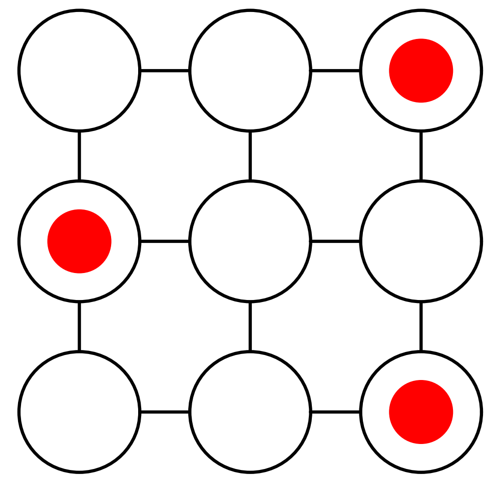
独立集合遷移問題（スライド版）

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



I_s



I_t

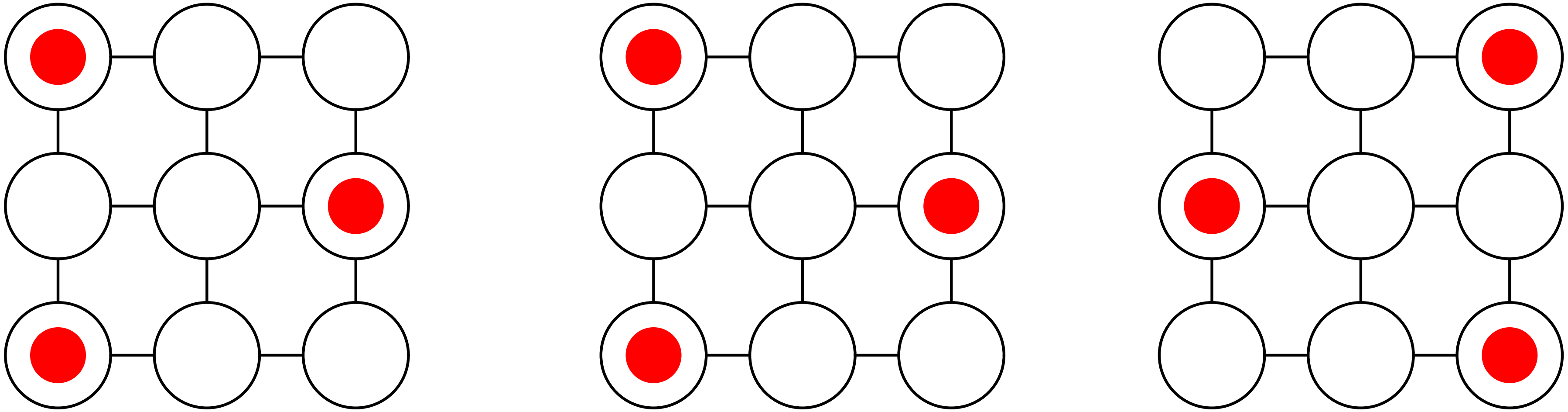
$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

独立集合遷移問題

独立集合遷移問題（スライド版）

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



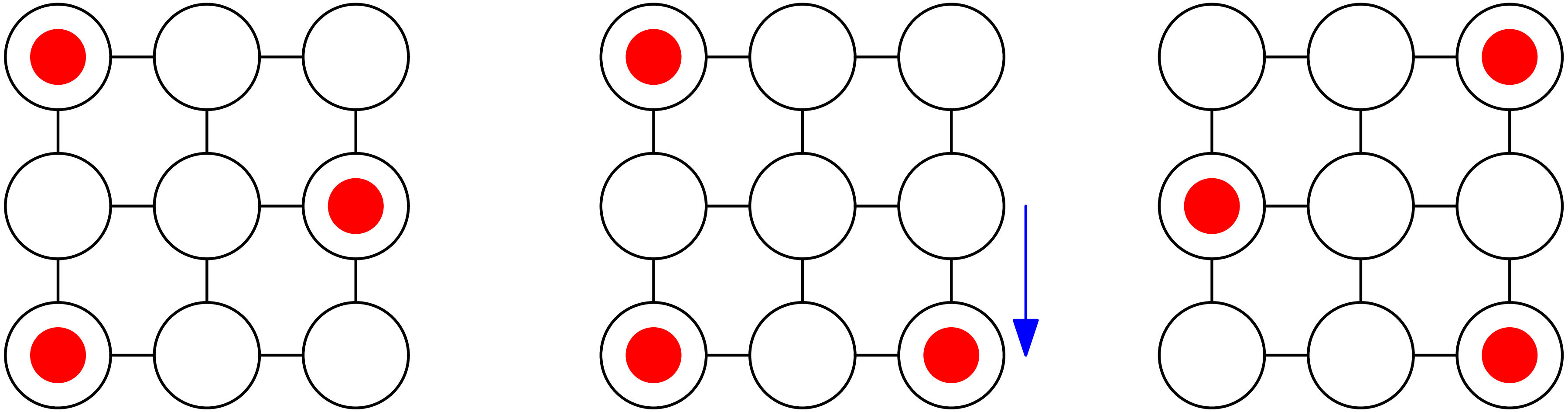
I_s I_t

$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

独立集合遷移問題（スライド版）

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



I_s I_t

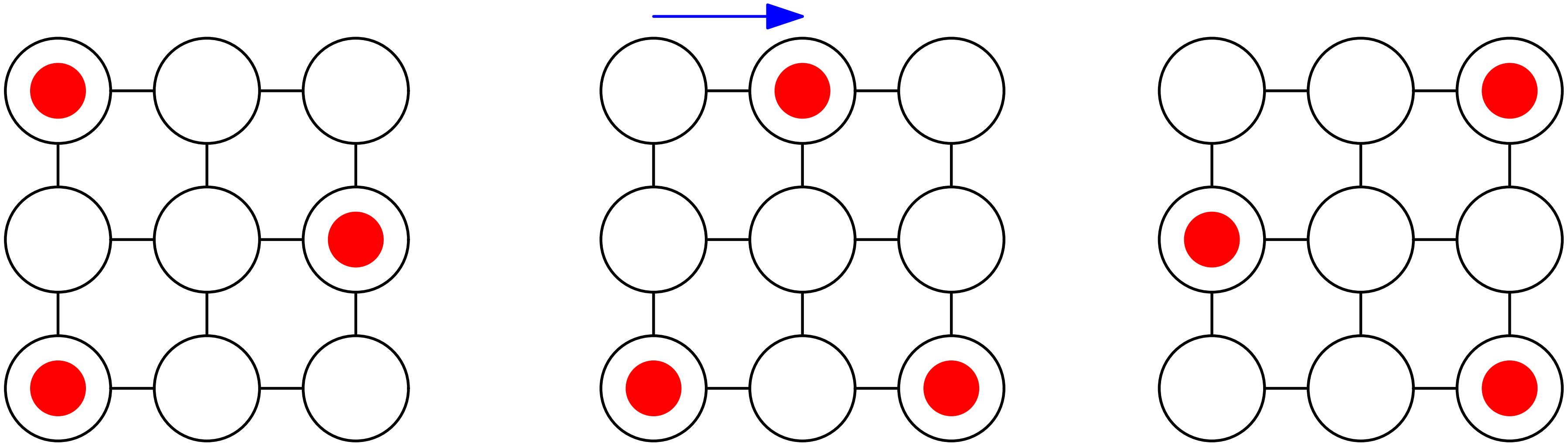
$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

独立集合遷移問題

独立集合遷移問題（スライド版）

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



I_s I_t

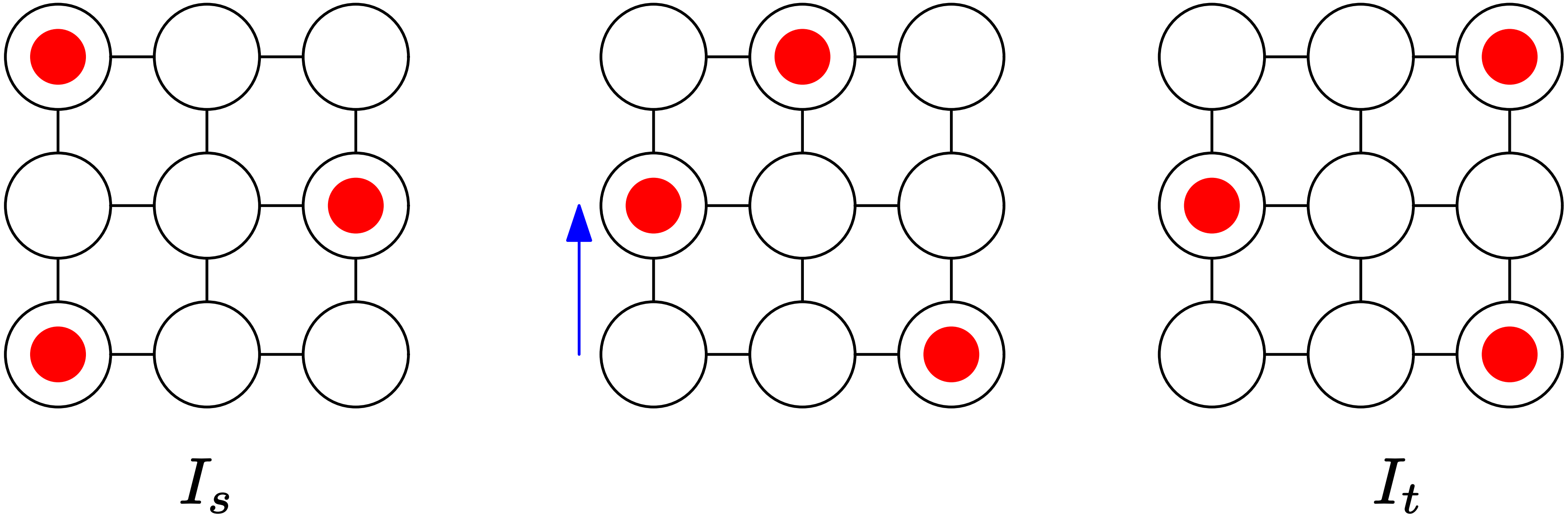
$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

独立集合遷移問題

独立集合遷移問題 (スライド版)

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



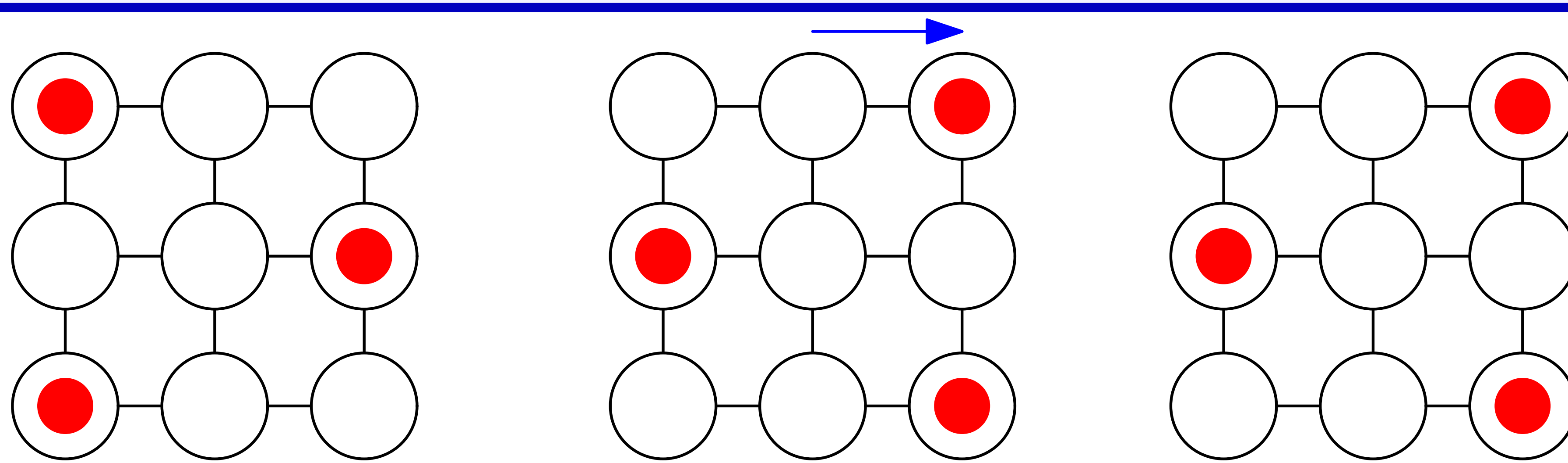
$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

独立集合遷移問題

独立集合遷移問題 (スライド版)

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No



I_s I_t

$I \subseteq V$ が $G = (V, E)$ の独立集合：$u, v \in I \Rightarrow \{u, v\} \notin E$

独立集合遷移問題は NPSPACE 完全

独立集合遷移問題（スライド版）

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$

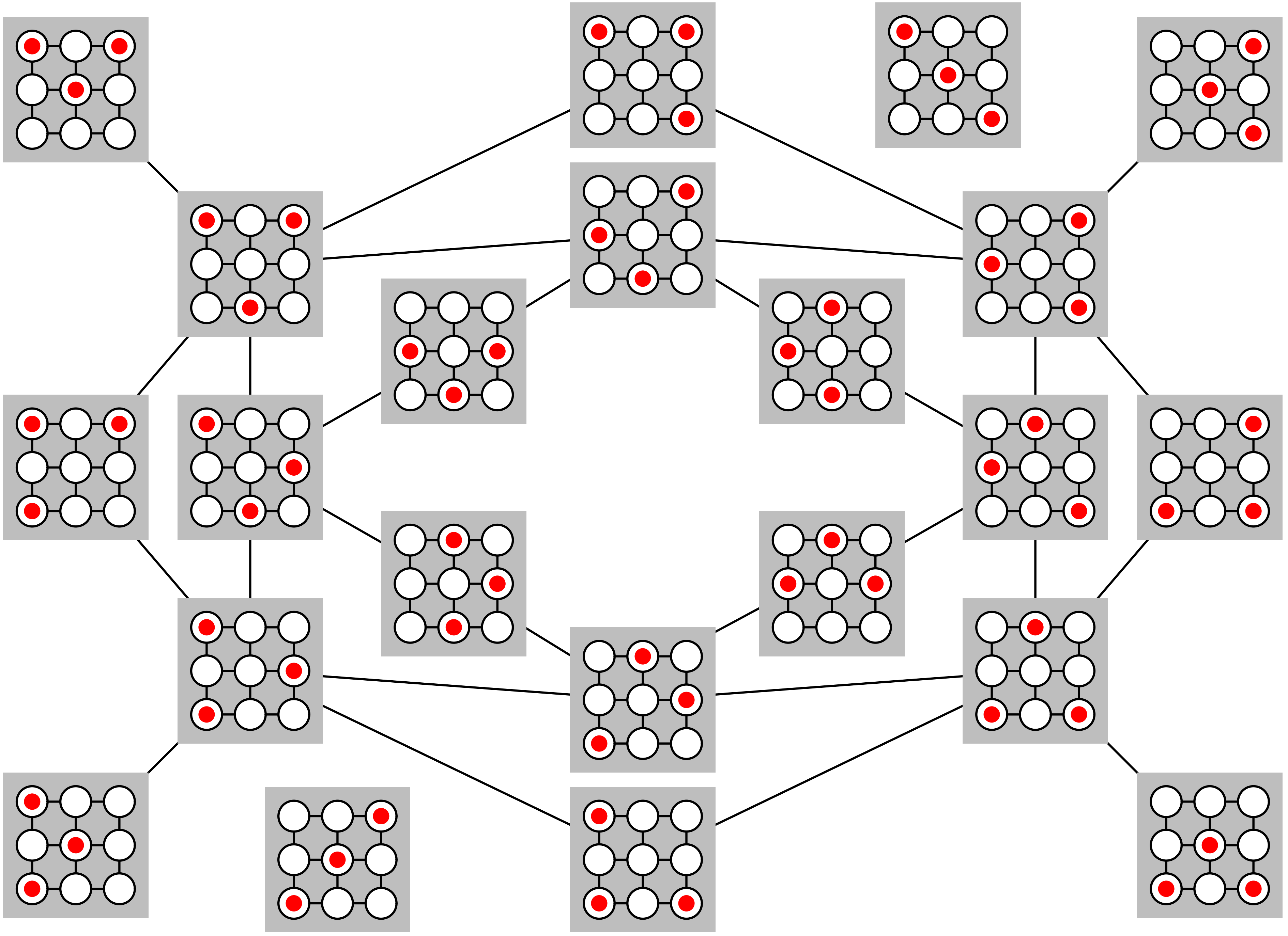
出力： I_s から I_t に遷移可能である $\Rightarrow$ Yes
I_s から I_t に遷移可能ではない $\Rightarrow$ No

事実

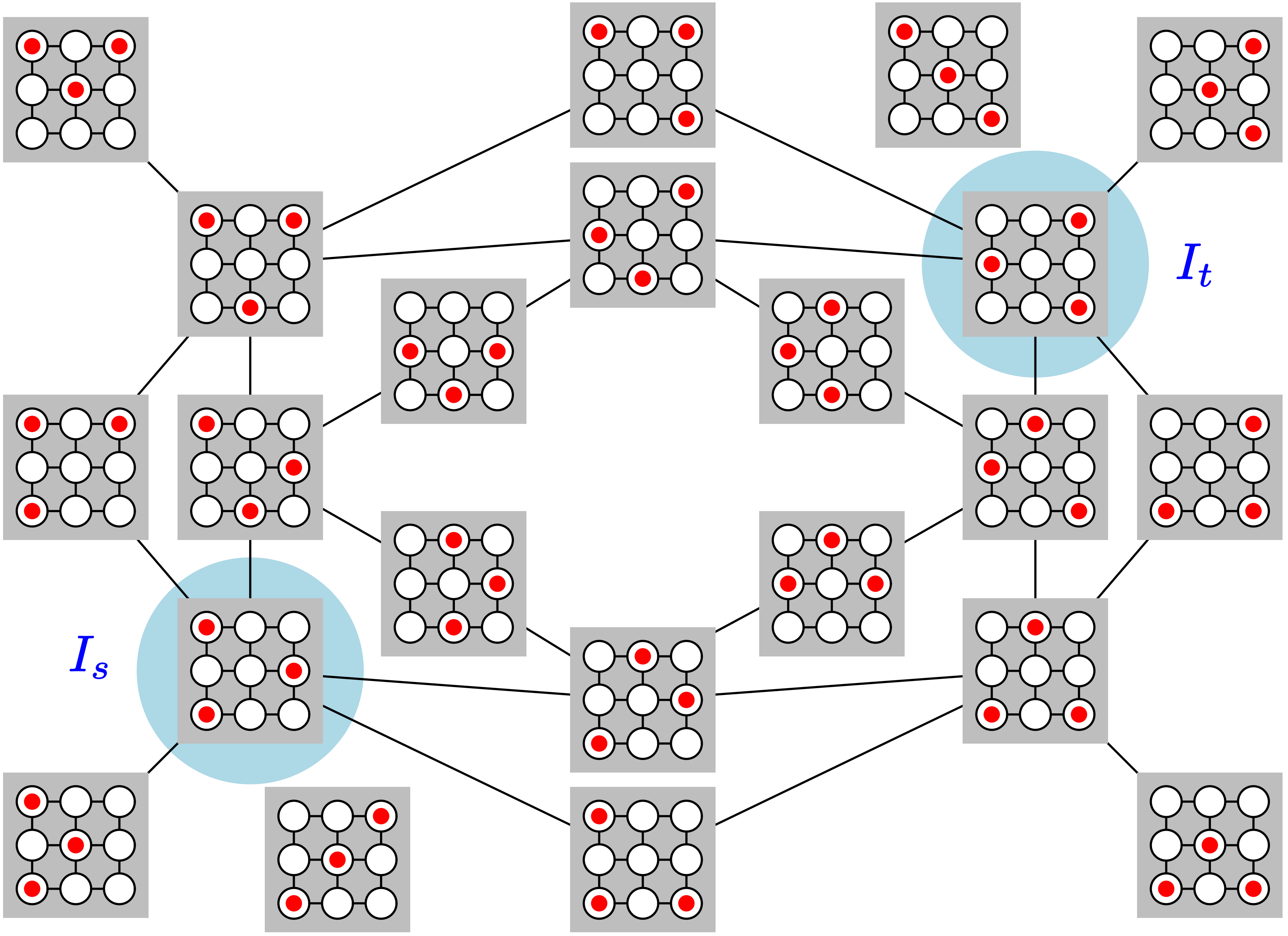
- 独立集合遷移問題は NPSPACE 完全　(Demaine, Hearn ’05)

以下，独立集合遷移問題 $\in$ NPSPACE を証明する

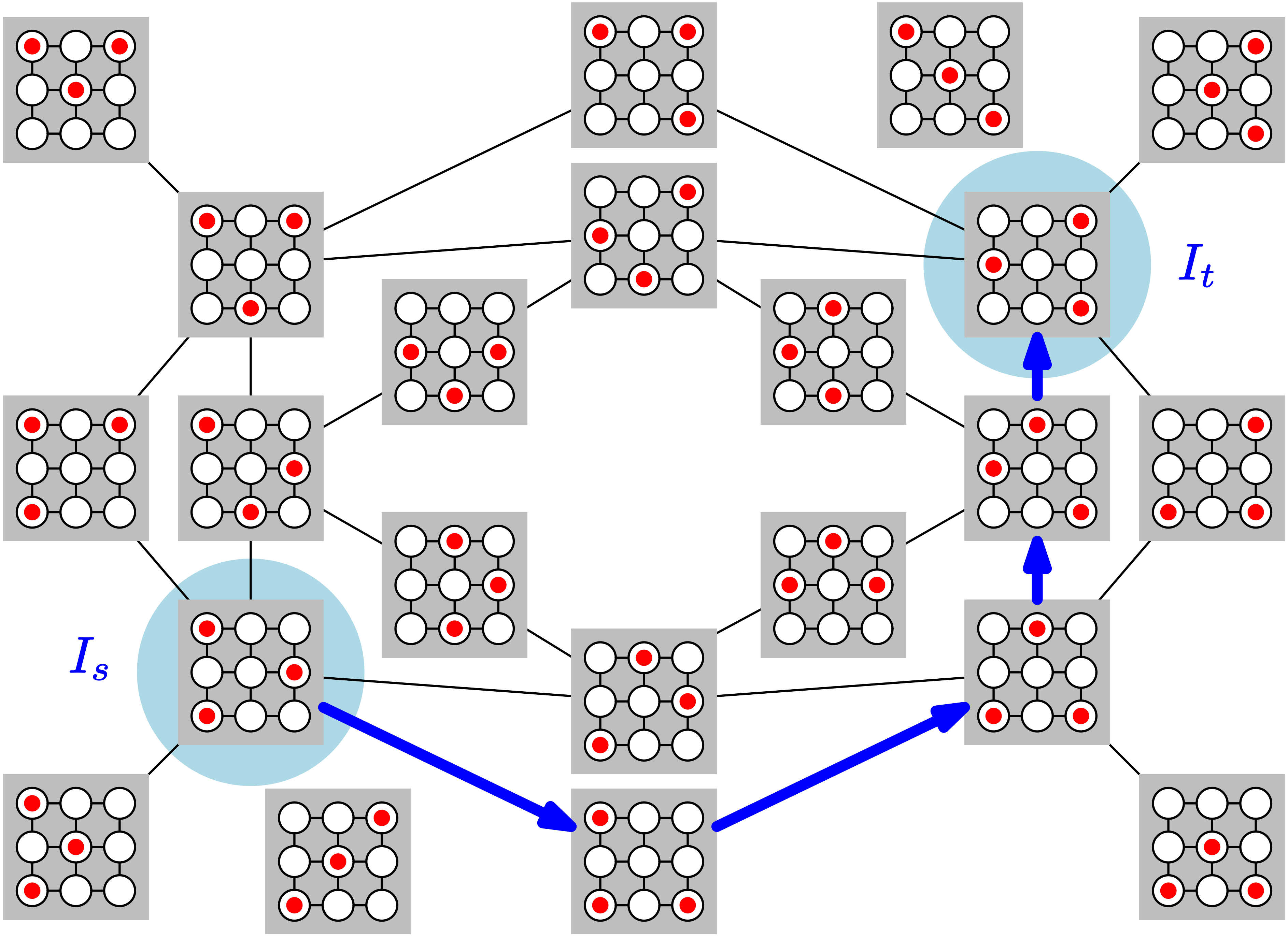
独立集合遷移問題：配位グラフ



独立集合遷移問題：配位グラフ



独立集合遷移問題：配位グラフ



独立集合遷移問題 $\in$ NPSPACE

アルゴリズム (入力：$G=(V,E), I_s, I_t$)

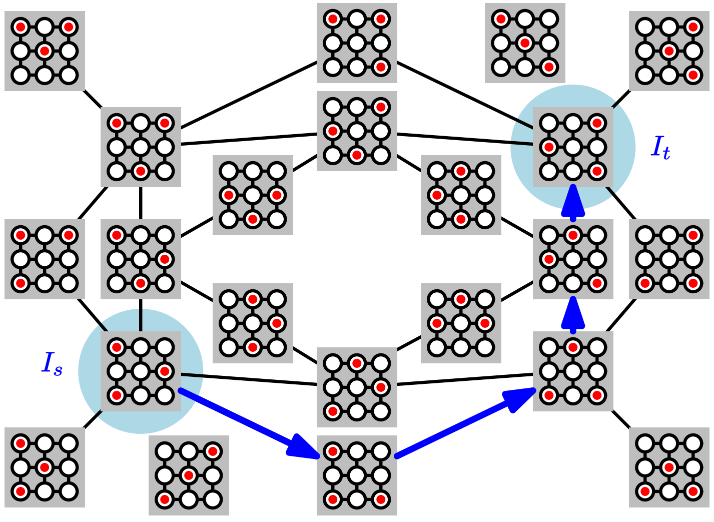
1. $I := I_s$
2. $I = I_t$ となるか $2^{|V|}$ 回反復するまで次を行う
   - I の次に訪れる独立集合を guess し，新たに I とする
3. $I = I_t$ となったら Yes，そうでなければ No

作業領域で覚えるもの

- I
- 反復回数

$\therefore$ 作業領域量 $= O(|V|)$

$\therefore$ 独立集合遷移問題 $\in$ NPSPACE



補足 (復習)：STCON ∈ NL

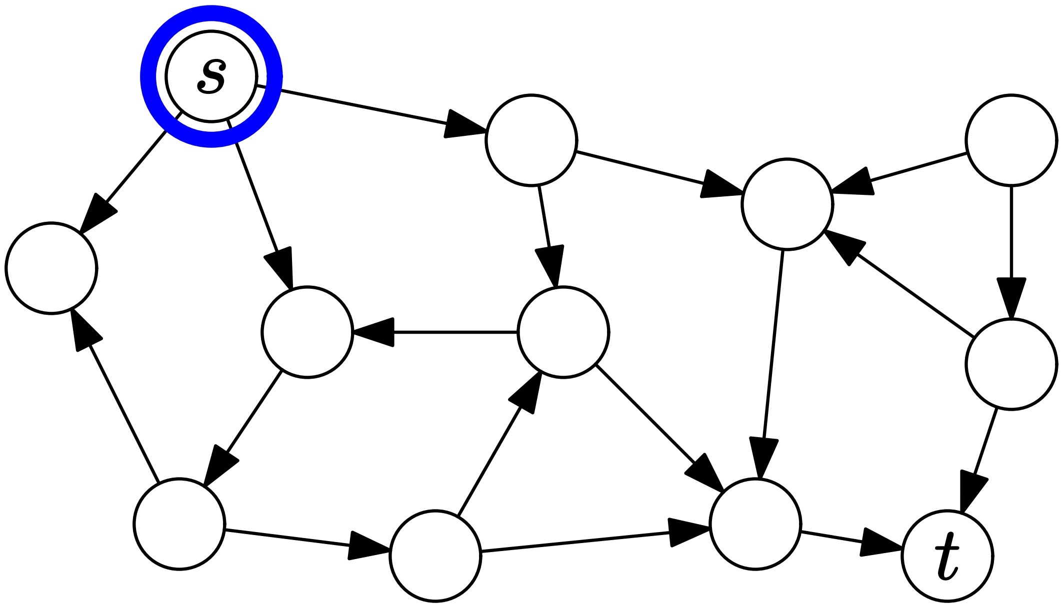
有向グラフ連結性判定問題 (STCON)

入力： 有向グラフ $G = (V, A)$, 2 頂点 $s, t \in V$

出力： s から t へ至る経路が存在する $\Rightarrow$ Yes

s から t へ至る経路が存在しない $\Rightarrow$ No

第 5 回にて：STCON ∈ NL

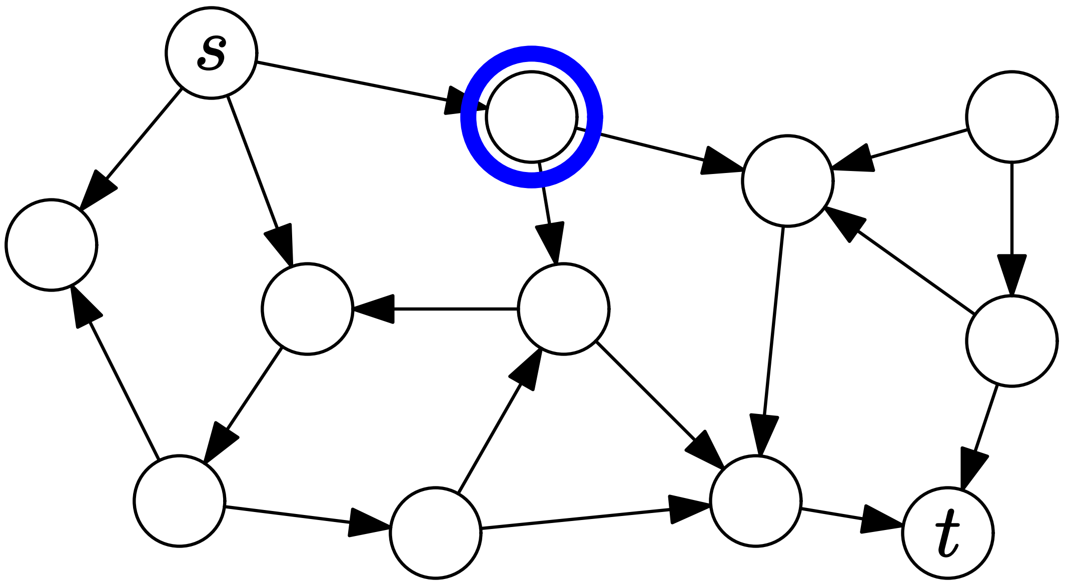


補足 (復習)：STCON ∈ NL

有向グラフ連結性判定問題 (STCON)

入力： 有向グラフ $G = (V, A)$, 2 頂点 $s, t \in V$
出力： s から t へ至る経路が存在する $\Rightarrow$ Yes
s から t へ至る経路が存在しない $\Rightarrow$ No

第 5 回にて：STCON ∈ NL



補足 (復習)：STCON ∈ NL

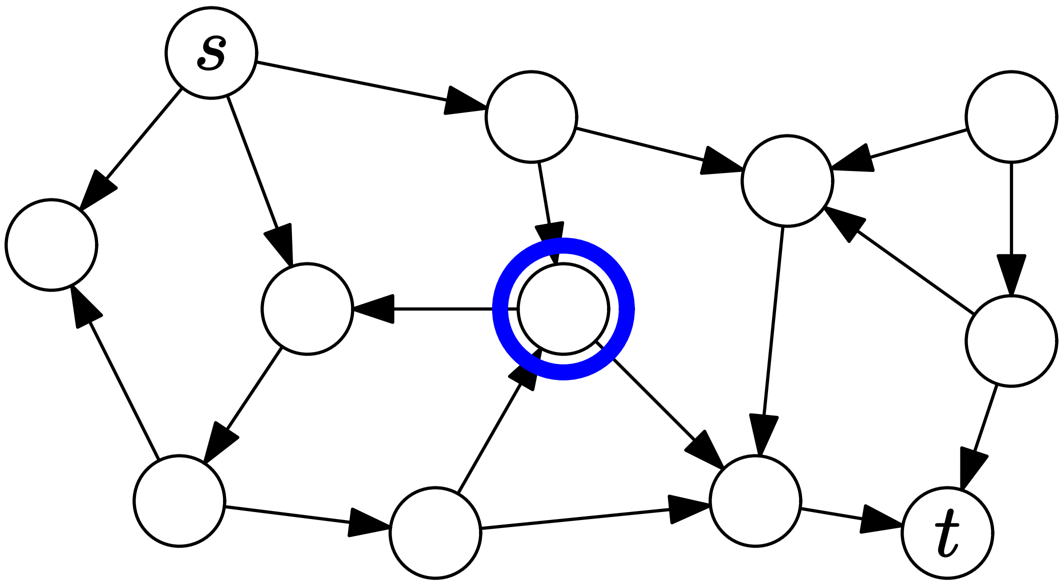
有向グラフ連結性判定問題 (STCON)

入力： 有向グラフ $G = (V, A)$, 2 頂点 $s, t \in V$

出力： s から t へ至る経路が存在する $\Rightarrow$ Yes

s から t へ至る経路が存在しない $\Rightarrow$ No

第 5 回にて：STCON ∈ NL

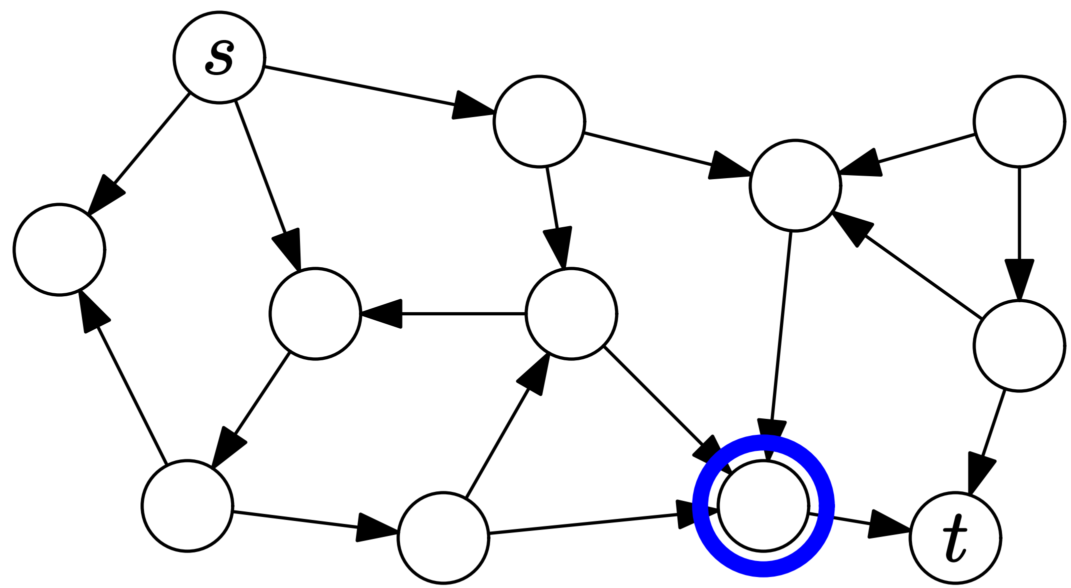


補足 (復習)：STCON ∈ NL

有向グラフ連結性判定問題 (STCON)

入力： 有向グラフ $G = (V, A)$, 2 頂点 $s, t \in V$
出力： s から t へ至る経路が存在する $\Rightarrow$ Yes
s から t へ至る経路が存在しない $\Rightarrow$ No

第 5 回にて：STCON ∈ NL

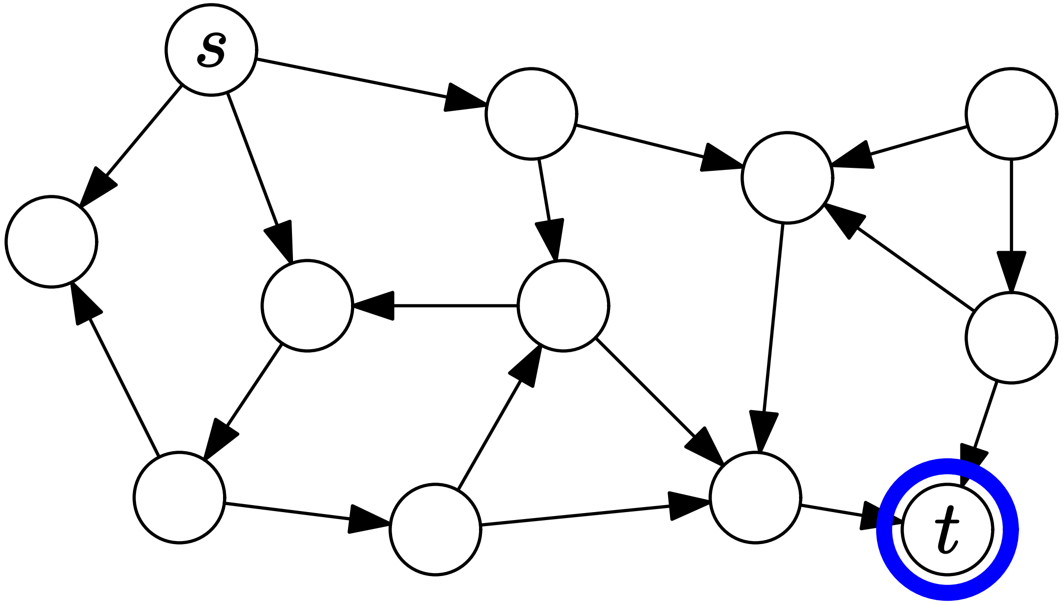


補足 (復習)：STCON ∈ NL

有向グラフ連結性判定問題 (STCON)

入力： 有向グラフ $G = (V, A)$, 2 頂点 $s, t \in V$
出力： s から t へ至る経路が存在する ⇒ Yes
s から t へ至る経路が存在しない ⇒ No

第 5 回にて：STCON ∈ NL



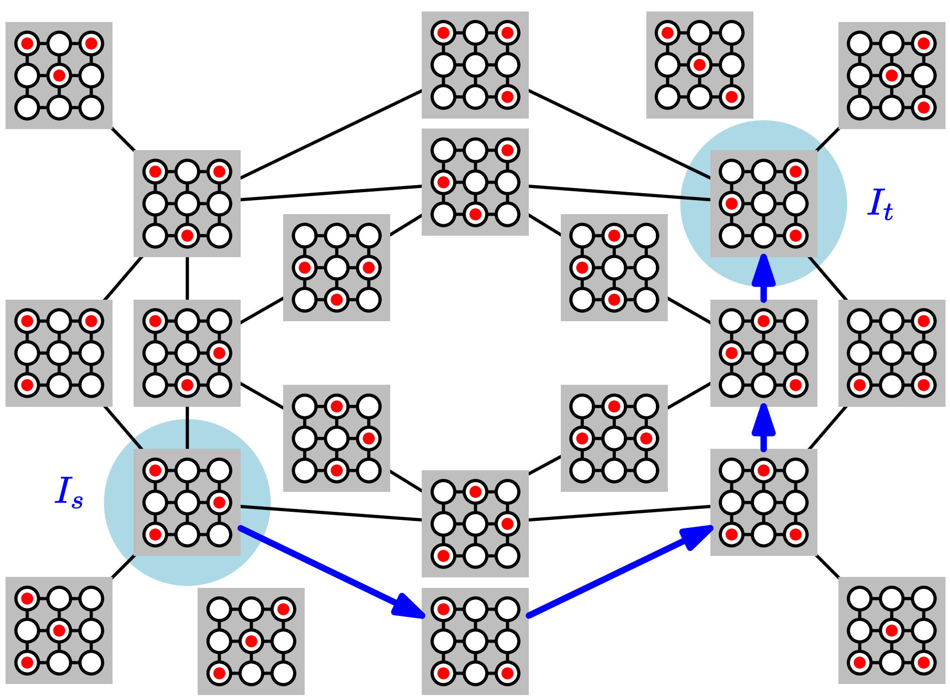
本日の内容

1. NPSPACE 完全問題の例
2. **Savitch の定理：証明**
3. Savitch の定理：補足
4. Savitch の定理の適用例

独立集合遷移問題 ∈ PSPACE

いまから行うこと

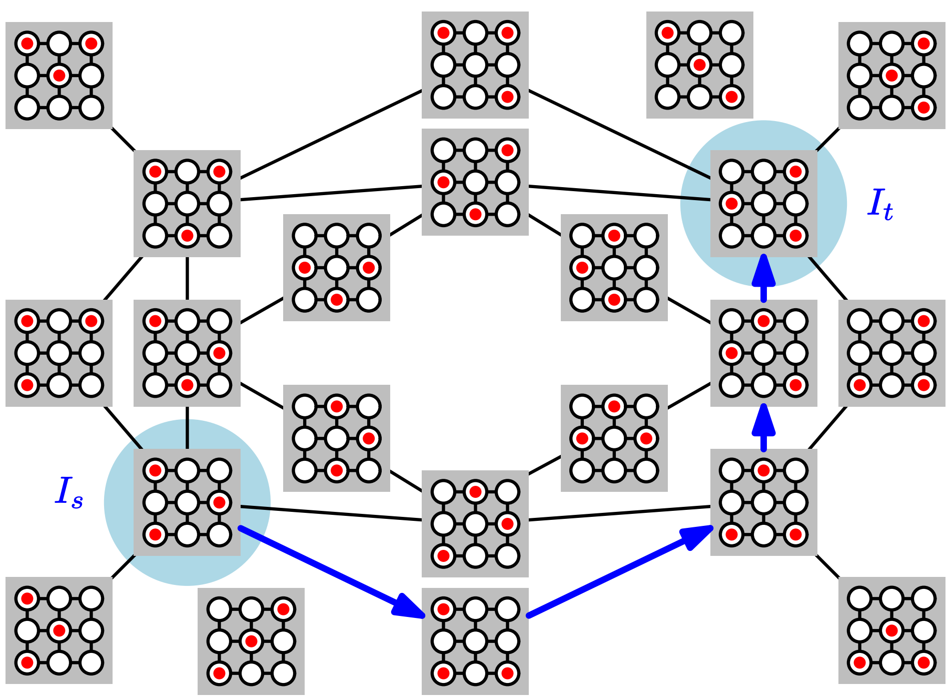
独立集合遷移問題に対する多項式領域アルゴリズムを作る



アイディア1：ステップ数で場合分け

アルゴリズム (ひな形)

1. $k = 0, 1, \ldots, 2^{|V|}$ に対して，次を繰り返し
   - I_s から I_t に k ステップで遷移可能 ⇒ Yes を出力
2. ここまでたどり着いたら，No を出力



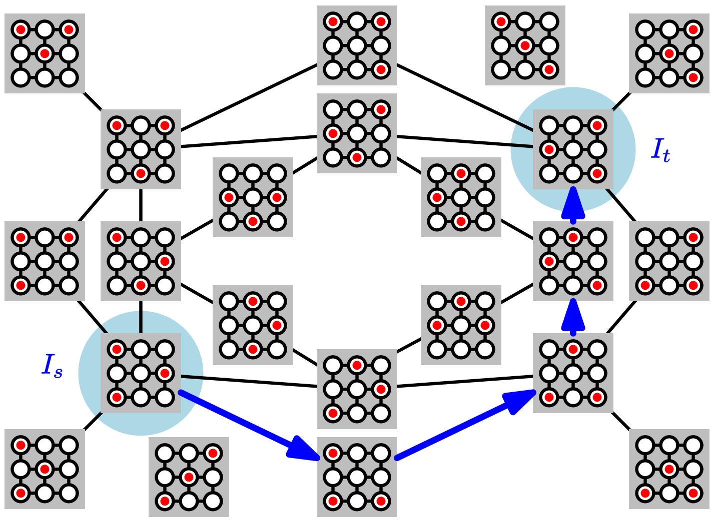
ステップ数限定独立集合遷移問題

つまり，次の問題が多項式領域で解ければ十分

ステップ数限定独立集合遷移問題

入力： 無向グラフ $G = (V, E)$,
G の独立集合 2 つ $I_s, I_t \subseteq V$,
非負整数 $k \geq 0$

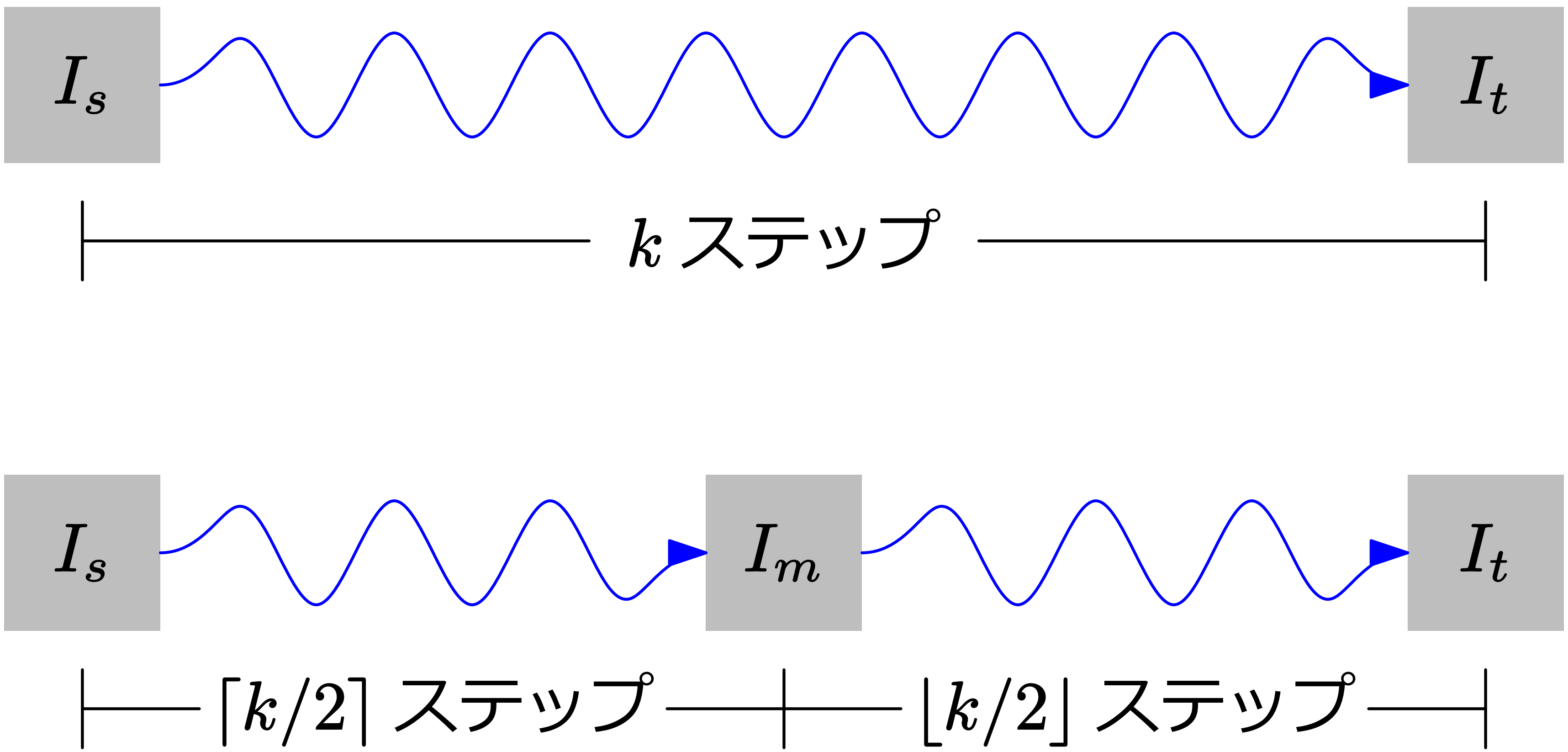
出力： I_s から I_t に k ステップ以内で遷移可能 ⇒ Yes
そうではない ⇒ No



アイディア 2：再帰

ステップ数限定独立集合遷移問題：解法のアイディア

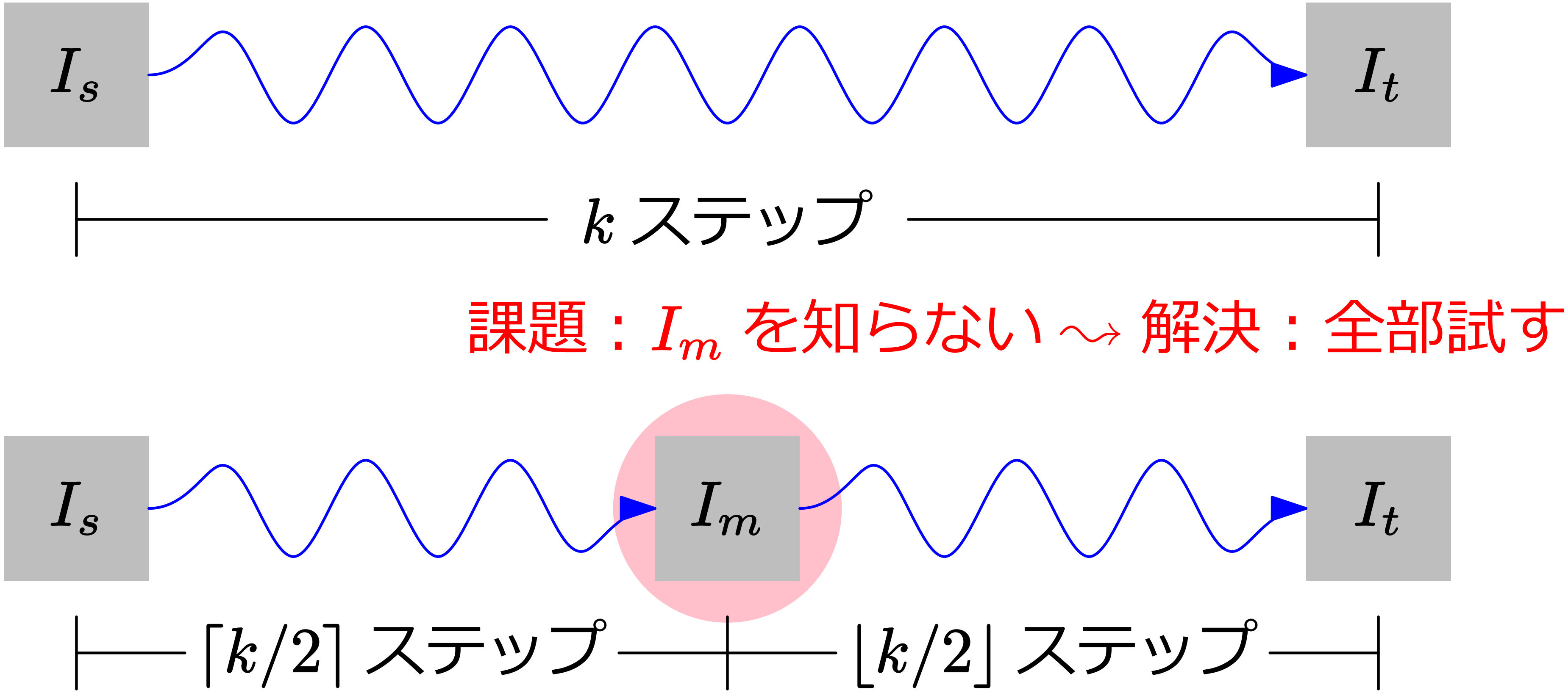
k ステップの遷移を
$\lceil k/2 \rceil$ ステップの遷移と $\lfloor k/2 \rfloor$ ステップの遷移に分ける



アイディア 2：再帰

ステップ数限定独立集合遷移問題：解法のアイディア

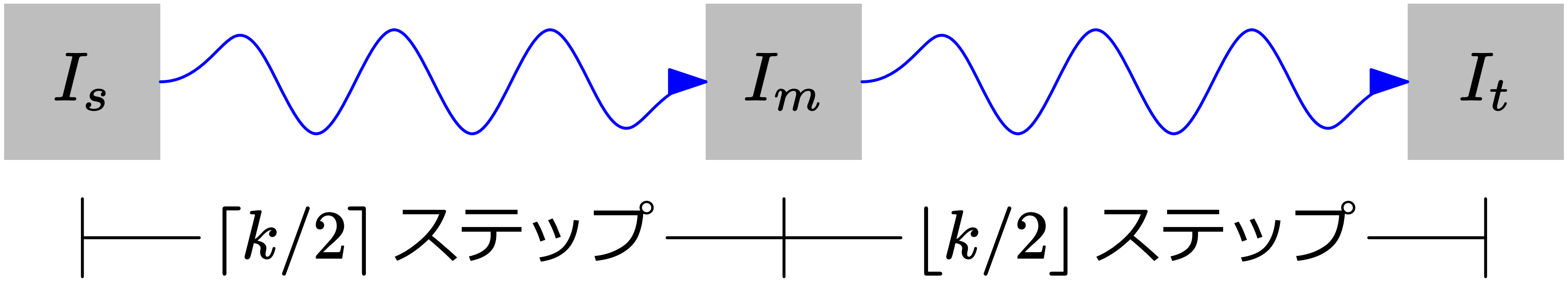
k ステップの遷移を
$\lceil k/2 \rceil$ ステップの遷移と $\lfloor k/2 \rfloor$ ステップの遷移に分ける



アルゴリズム

アルゴリズム $A(I_s, I_t, k)$

1. $k \leq 1$ のとき，解くのは簡単
2. $k \geq 2$ のとき，
   G のすべての独立集合 I_m に対して，次を実行
   (a) $A(I_s, I_m, \lceil k/2 \rceil)$ を実行
   (b) $A(I_m, I_t, \lfloor k/2 \rfloor)$ を実行
   (c) 両者が Yes を出力 $\Rightarrow$ Yes を出力して終了
3. ここまで来たら，No を出力して終了

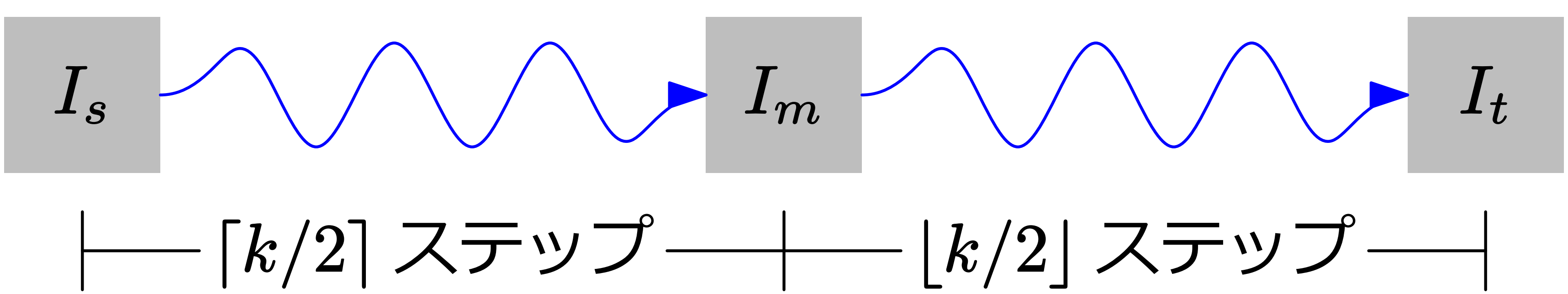


アルゴリズム

アルゴリズム $A(I_s, I_t, k)$

1. $k \leq 1$ のとき，解くのは簡単
2. $k \geq 2$ のとき，
   G のすべての独立集合 I_m に対して，次を実行
   (a) $A(I_s, I_m, \lceil k/2 \rceil)$ を実行
   (b) $A(I_m, I_t, \lfloor k/2 \rfloor)$ を実行
   (c) 両者が Yes を出力 $\Rightarrow$ Yes を出力して終了
3. ここまで来たら，No を出力して終了

これが正しい出力をすることは 今までの議論から分かる



アルゴリズム：領域計算量

アルゴリズム $A(I_s, I_t, k)$

領域計算量を $s(k)$ とする

1. $k \leq 1$ のとき，解くのは簡単
2. $k \geq 2$ のとき，
   G のすべての独立集合 I_m に対して，次を実行
   (a) $A(I_s, I_m, \lceil k/2 \rceil)$ を実行
   (b) $A(I_m, I_t, \lfloor k/2 \rfloor)$ を実行
   (c) 両者が Yes を出力 $\Rightarrow$ Yes を出力して終了
3. ここまで来たら，No を出力して終了

$s(0) = s(1) = O(|V|)$， $s(k) = O(|V|) + s(\lceil k/2 \rceil)$ if $k \geq 2$

アルゴリズム：領域計算量

アルゴリズム $A(I_s, I_t, k)$

領域計算量を $s(k)$ とする

1. $k \leq 1$ のとき，解くのは簡単
2. $k \geq 2$ のとき，
   G のすべての独立集合 I_m に対して，次を実行
   (a) $A(I_s, I_m, \lceil k/2 \rceil)$ を実行
   (b) $A(I_m, I_t, \lfloor k/2 \rfloor)$ を実行
   (c) 両者が Yes を出力 ⇒ Yes を出力して終了
3. ここまで来たら，No を出力して終了

$s(0) = s(1) = O(|V|)$， $s(k) = O(|V|) + s(\lceil k/2 \rceil)$ if $k \geq 2$

$$s(2^{|V|}) = O(|V|) + s(\lceil 2^{|V|}/2 \rceil)$$
$$= \underbrace{O(|V|) + O(|V|) + \cdots + O(|V|)}_{O(|V|) \text{ 個}} = O(|V|^2)$$

Savitch の定理：まとめ

ここまでのまとめ

- 独立集合遷移問題は NPSPACE 完全
- 独立集合遷移問題 $\in$ PSPACE

結論：Savitch の定理 ('70)

PSPACE $=$ NPSPACE

Savitch の定理：まとめ

ここまでのまとめ

- 独立集合遷移問題は NPSPACE 完全
- 独立集合遷移問題 $\in$ PSPACE

結論：Savitch の定理 ('70)

PSPACE $=$ NPSPACE

Savitch の定理から ただちに 得られること

- NPSPACE 完全問題は PSPACE 完全
- NPSPACE $=$ coNPSPACE

本日の内容

1. NPSPACE 完全問題の例
2. Savitch の定理：証明
3. **Savitch の定理：補足**
4. Savitch の定理の適用例

Savitch の定理 (一般形)

教科書には，次の形で Savitch の定理が書かれている

Savitch の定理 (一般形) ('70)

多テープ・チューリング機械において，
任意の関数 $s\colon \mathbb{N} \to \mathbb{N}$ に対して (ただし，$s(n) \geq \log n$)，

$$\mathsf{NSPACE}(s(n)) \subseteq \mathsf{SPACE}(s(n)^2)$$

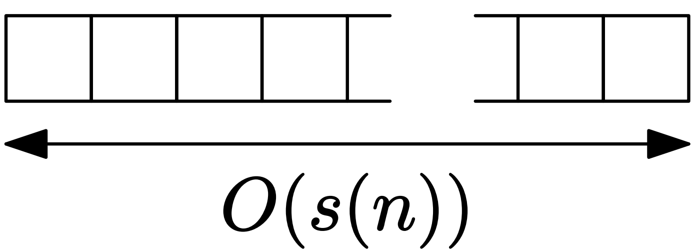
- $\mathsf{SPACE}(f(n)) =$ 領域計算量 $O(f(n))$ の (非決定性がない) アルゴリズムで解ける判定問題のクラス
- $\mathsf{NSPACE}(f(n)) =$ 領域計算量 $O(f(n))$ の非決定性があるアルゴリズムで解ける判定問題のクラス

Savitch の定理 (一般形) ('70)

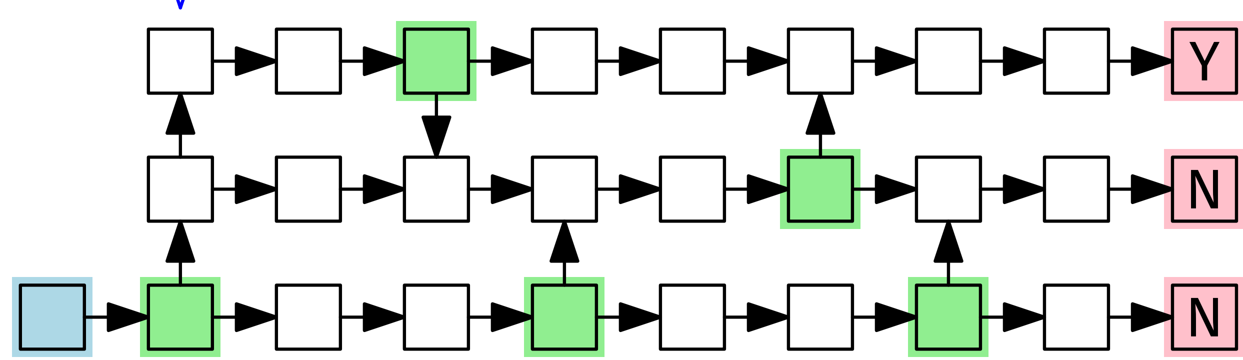
多テープ・チューリング機械において,
任意の関数 $s\colon \mathbb{N} \to \mathbb{N}$ に対して (ただし，$s(n) \geq \log n$),

$$\mathsf{NSPACE}(s(n)) \subseteq \mathsf{SPACE}(s(n)^2)$$

証明の考え方：独立集合遷移問題 $\in$ PSPACE の証明とほぼ同じ



入力を固定したとき
時点状況の総数 $= O(2^{O(s(n))})$



一般形から，特に次も正しいことが分かる

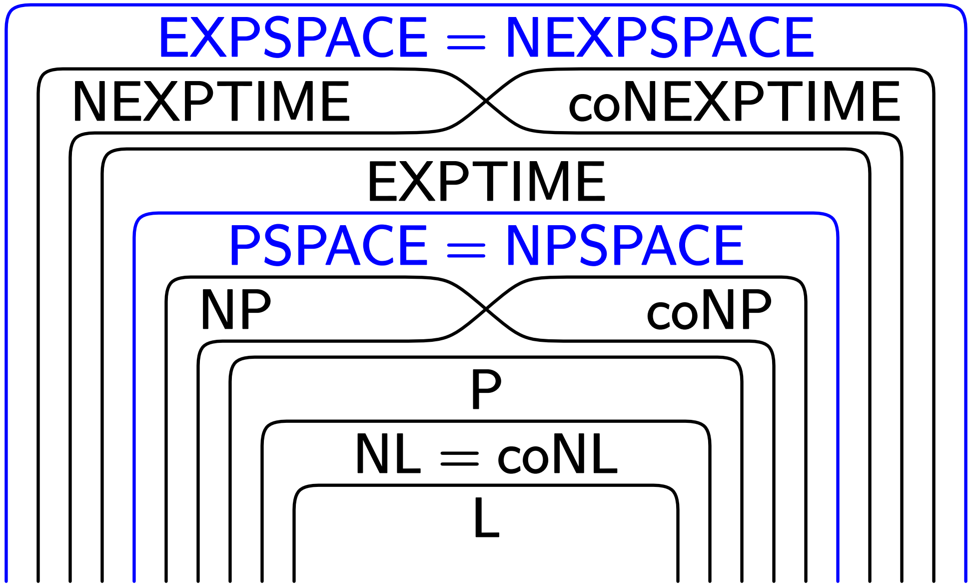
性質：Savitch の定理の系

EXPSPACE = NEXPSPACE

これは，

「PSPACE = NPSPACE」に水増し論法を適用しても得られる

(2 ページ後に行う)



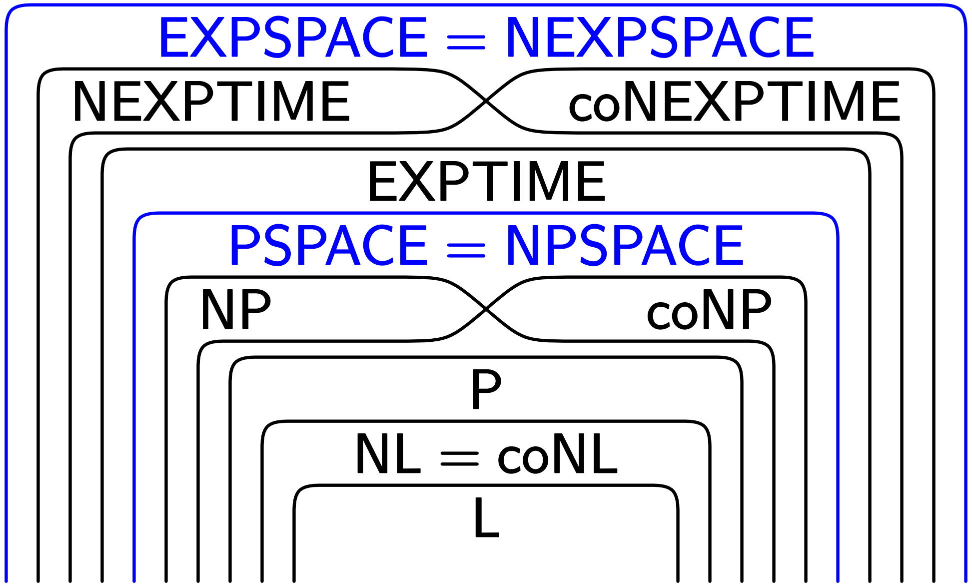
L $=$? NL

一般形から次は証明できず，実際，次は未解決である

未解決問題

$\mathrm{L} \stackrel{?}{=} \mathrm{NL}$

注：L, NL の領域計算量 $= O(\log n)$



EXPSPACE = NEXPSPACE：証明 (1)

証明：NEXPSPACE ⊆ EXPSPACE を示せば十分

- $P \in$ NEXPSPACE とする
- A は P を解く非決定性指数領域アルゴリズムだとする
- A の領域計算量を $O(2^{|I|^k})$ とする (k は定数)

$$P \text{ の入力 } I \longrightarrow \boxed{A} \longrightarrow \text{Yes/No}$$

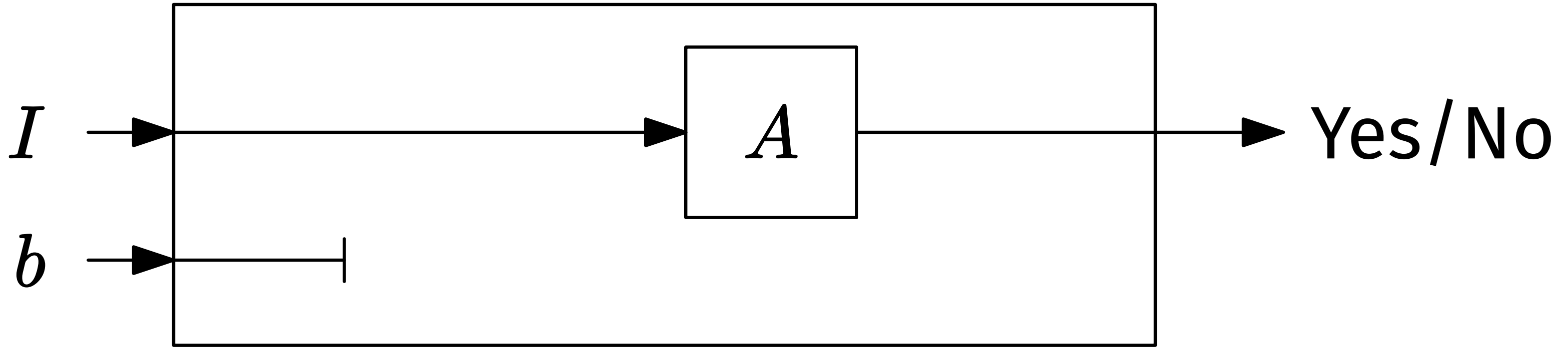
次の問題「水増し P」を考える

入力： P の入力 I，長さ $2^{|I|^k}$ のビット列 b

出力： I が P の Yes インスタンス ⇒ Yes
I が P の No インスタンス ⇒ No

- A を使って，「水増し P」を解くアルゴリズム A' が次のように作れる

1: b を無視して，I だけを取り出す
2: $A(I)$ を実行して，その出力をそのまま出力する



水増し P を解くアルゴリズム A'

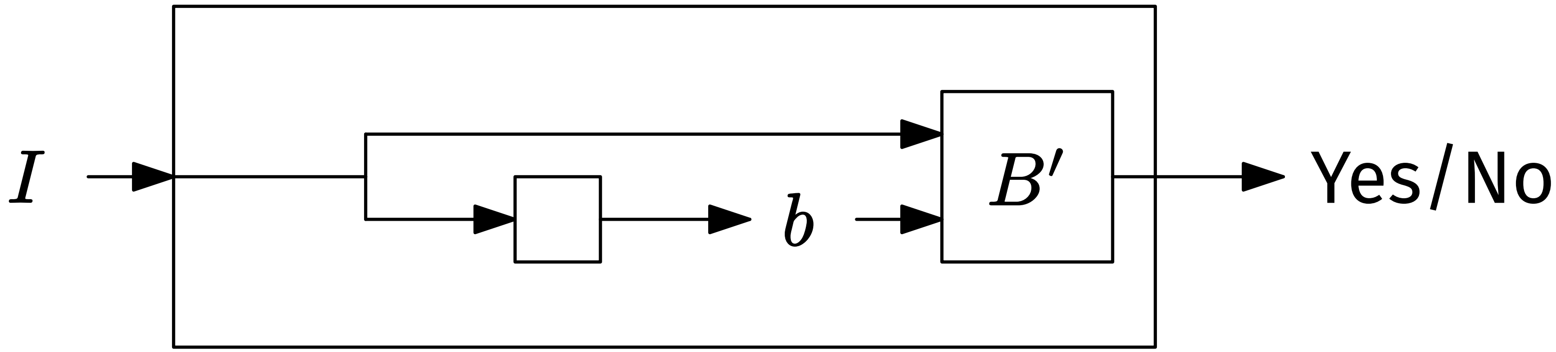
- 入力の符号長 $n = O(|I| + |b|) = O(|I| + 2^{|I|^k}) = O(2^{|I|^k})$
- A の領域計算量 $= O(2^{|I|^k}) = O(n)$
- $\therefore A'$ は非決定性多項式領域アルゴリズム
- $\therefore$ 水増し $P \in$ NPSPACE

EXPSPACE = NEXPSPACE：証明 (3)

- Savitch の定理より，水増し $P \in$ PSPACE
- $\therefore$ 水増し P を解く多項式領域アルゴリズム B' が存在

- B' を使って，元の問題 P を解くアルゴリズム B を作る

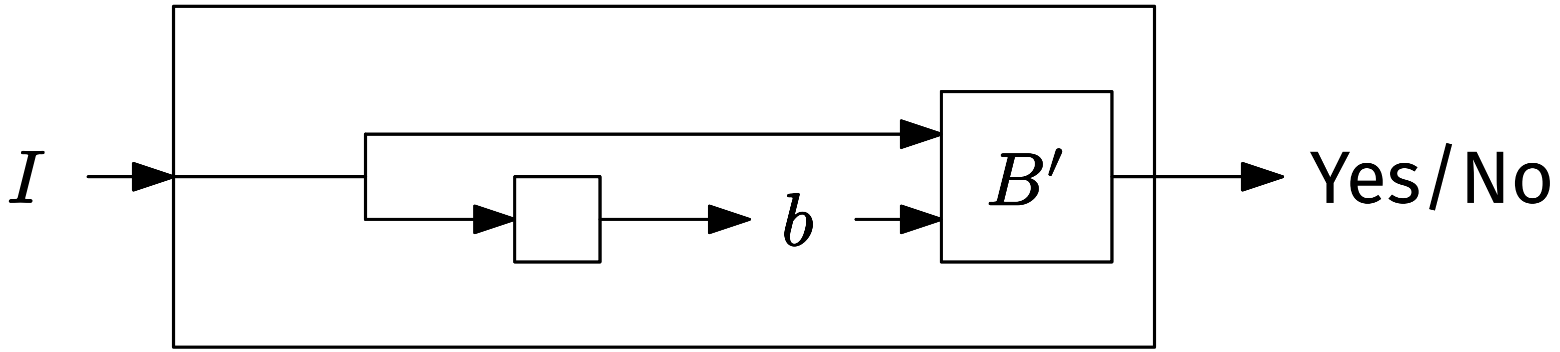
1: 入力 I から $2^{|I|^k}$ を計算する
2: 長さ $2^{|I|^k}$ のビット列 b を作る
3: $B'(I, b)$ を実行し，その出力をそのまま出力とする



P を解くアルゴリズム B

- 注：ステップ 1, 2 は $O(2^{|I|^k})$ 領域で実行可能

- B' の領域計算量 $= O(|I'|^{k'})$ とすると
  B の領域計算量 $= O((2^{|I|^k})^{k'}) = O(2^{k'|I|^k})$
- $\therefore P \in$ EXPSPACE
- $\therefore$ NEXPSPACE $\subseteq$ EXPSPACE $\square$



P を解くアルゴリズム B

本日の内容

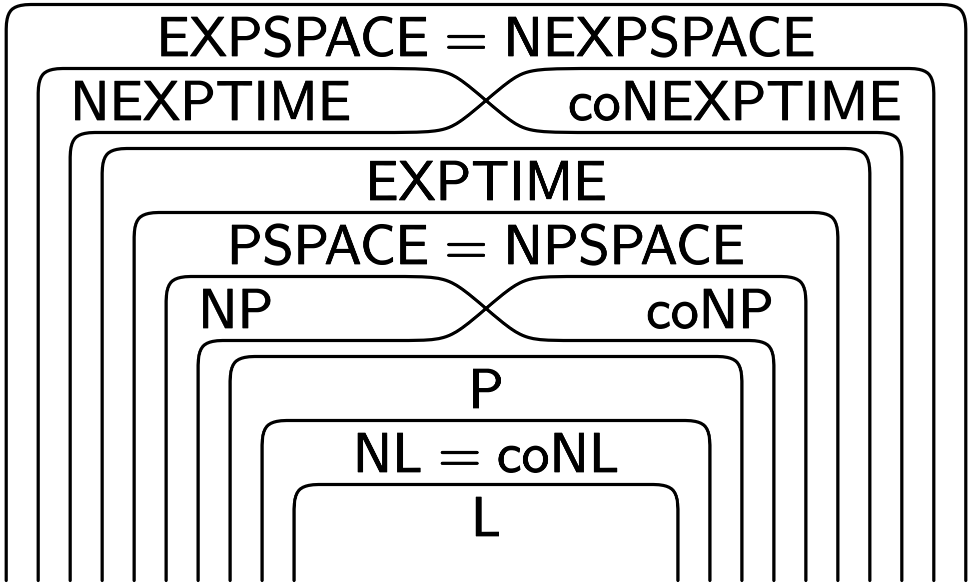
1. NPSPACE 完全問題の例
2. Savitch の定理：証明
3. Savitch の定理：補足
4. **Savitch の定理の適用例**

適用例：PSPACE の所属性

ある問題 P が PSPACE に属することを証明するには？

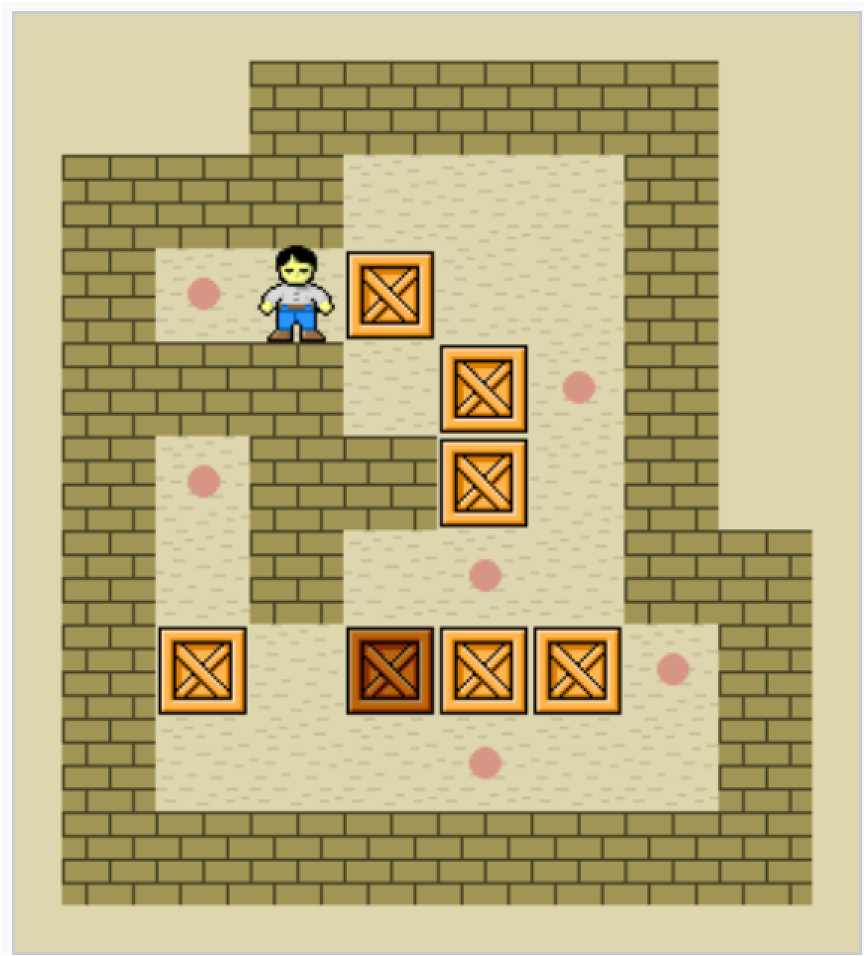
- **素朴な方法**：
  P を解く多項式領域アルゴリズムを作る
  $(\Rightarrow P \in \mathrm{PSPACE})$

- **より簡単な方法**：
  P を解く非決定性多項式領域アルゴリズムを作る
  $(\Rightarrow P \in \mathrm{NPSPACE} = \mathrm{PSPACE})$

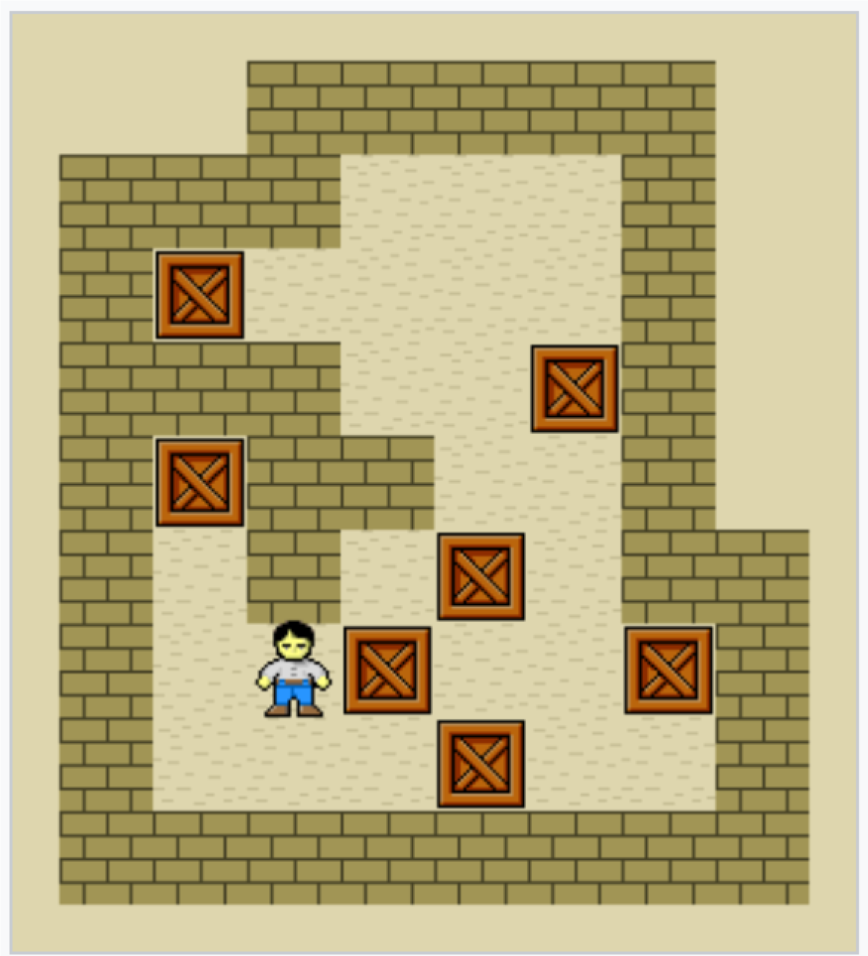


例：倉庫番

(シンキングラビット ’82)



盤面



解けた状況

https://commons.wikimedia.org/wiki/File:Sokoban_ani.gif

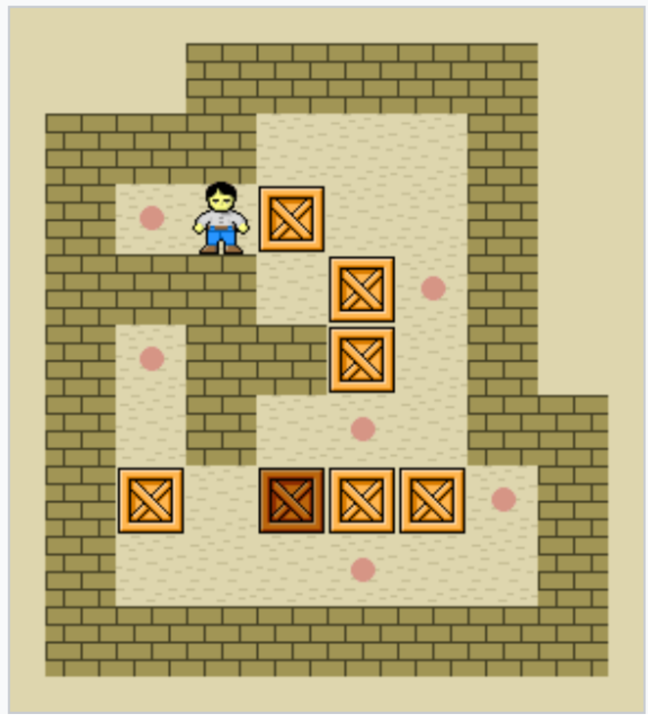
倉庫番：問題として

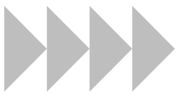
問題：倉庫番

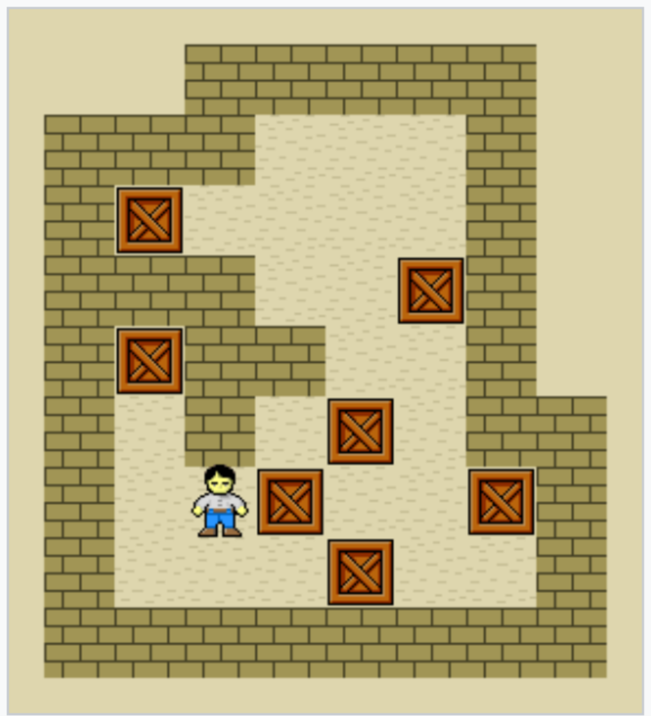
入力： 倉庫番の盤面 B

出力： B が解ける $\Rightarrow$ Yes

B が解けない $\Rightarrow$ No







https://commons.wikimedia.org/wiki/File:Sokoban_ani.gif

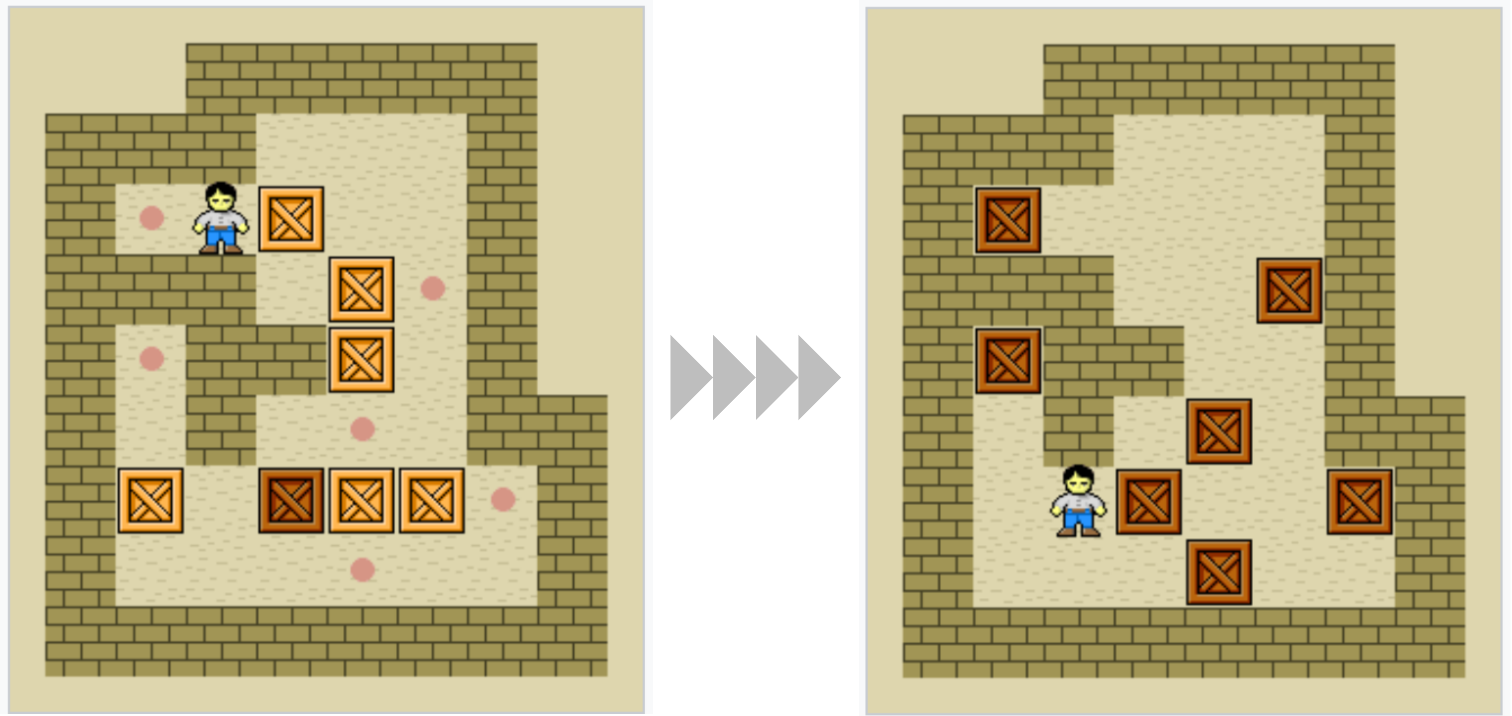
倉庫番：問題として

問題：倉庫番

入力： 倉庫番の盤面 B

出力： B が解ける $\Rightarrow$ Yes

B が解けない $\Rightarrow$ No



注：各マスは

- 空き
- 箱
- 人

のどれか

https://commons.wikimedia.org/wiki/File:Sokoban_ani.gif

性質

倉庫番 $\in$ PSPACE

証明：倉庫番 $\in$ NPSPACE を示す

$\therefore$ Savitch の定理より，倉庫番 $\in$ NPSPACE $=$ PSPACE $\square$

倉庫番 $\in$ PSPACE

性質

倉庫番 $\in$ PSPACE

証明：倉庫番 $\in$ NPSPACE を示す

アルゴリズム (入力：B)

1. $T := B$
2. T が解けた状態となるか 3^n 回反復するまで次を行う
   - T の次に訪れる盤面を guess し，新たに T とする
3. T が解けた状態となったら Yes，そうでなければ No

(ただし，$n =$ 盤面 B のマス目の総数)

これは，非決定性のある多項式領域アルゴリズムである

$\therefore$ Savitch の定理より，倉庫番 $\in$ NPSPACE $=$ PSPACE $\square$

倉庫番：知られていること

倉庫番という問題について知られていること

倉庫番は PSPACE 完全 (Culberson ’98)

関連するパズルの計算複雑性・未解決問題について

- Erik Demaine の招待講演 (JCDCG^3 ’22)
  - YouTube で視聴可能

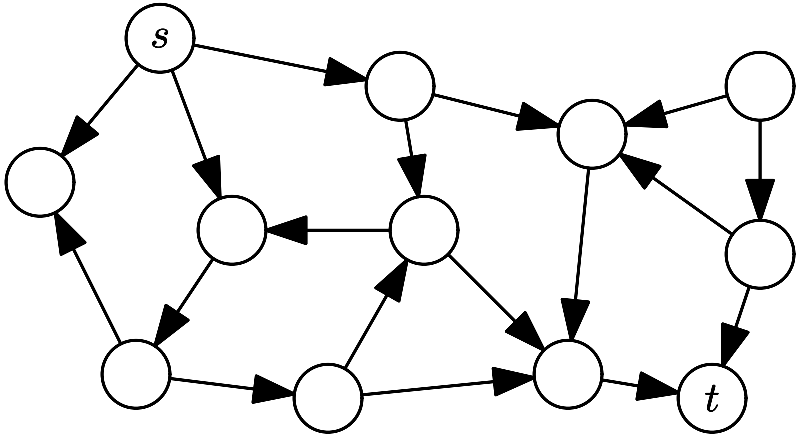
    `https://www.youtube.com/watch?v=IIe0WHB1loY`

PSPACE 完全問題の気分 1

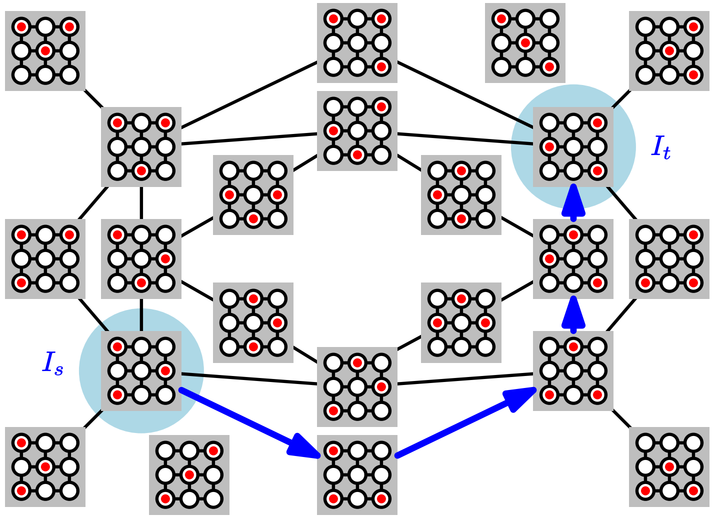
気分

「圧縮された STCON」は PSPACE 完全になりがち

STCON は NL 完全



独立集合遷移問題は PSPACE 完全

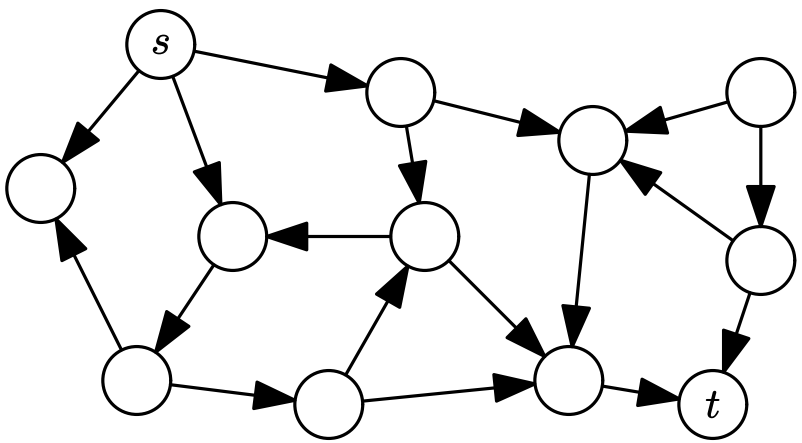


PSPACE 完全問題の気分 1

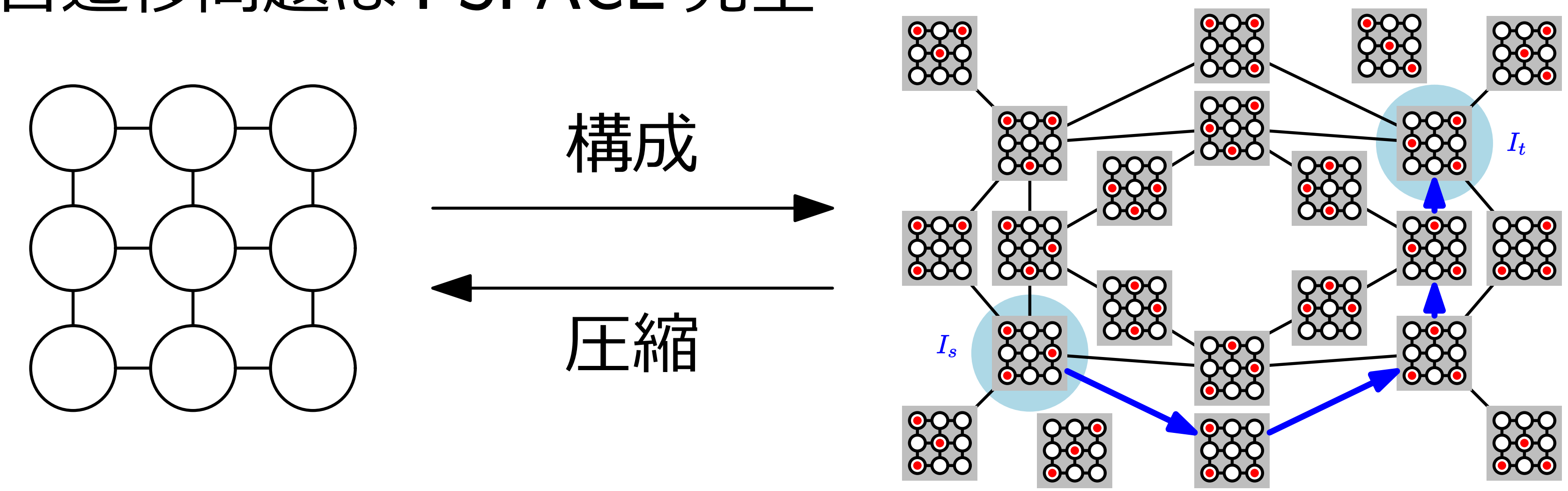
気分

「圧縮された STCON」は PSPACE 完全になりがち

STCON は NL 完全



独立集合遷移問題は PSPACE 完全



注：USTCON

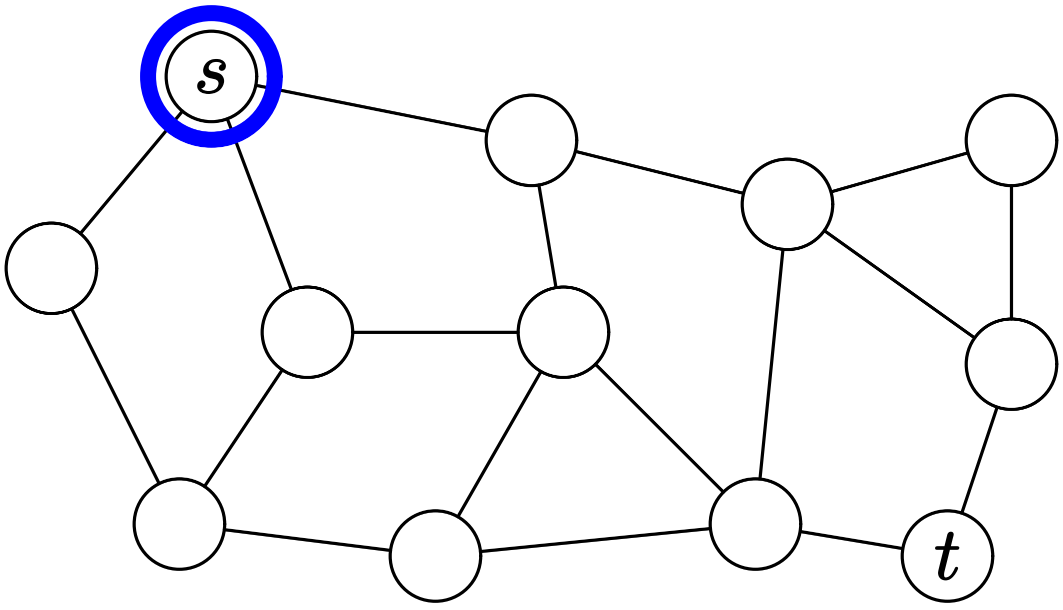
無向グラフ連結性判定問題 (USTCON)

入力： 無向グラフ $G = (V, E)$, 2 頂点 $s, t \in V$

出力： s から t へ至る経路が存在する $\Rightarrow$ Yes

s から t へ至る経路が存在しない $\Rightarrow$ No

- 事実：USTCON は L 完全 (Reingold '08)
- 未解決：L $\stackrel{?}{=}$ NL

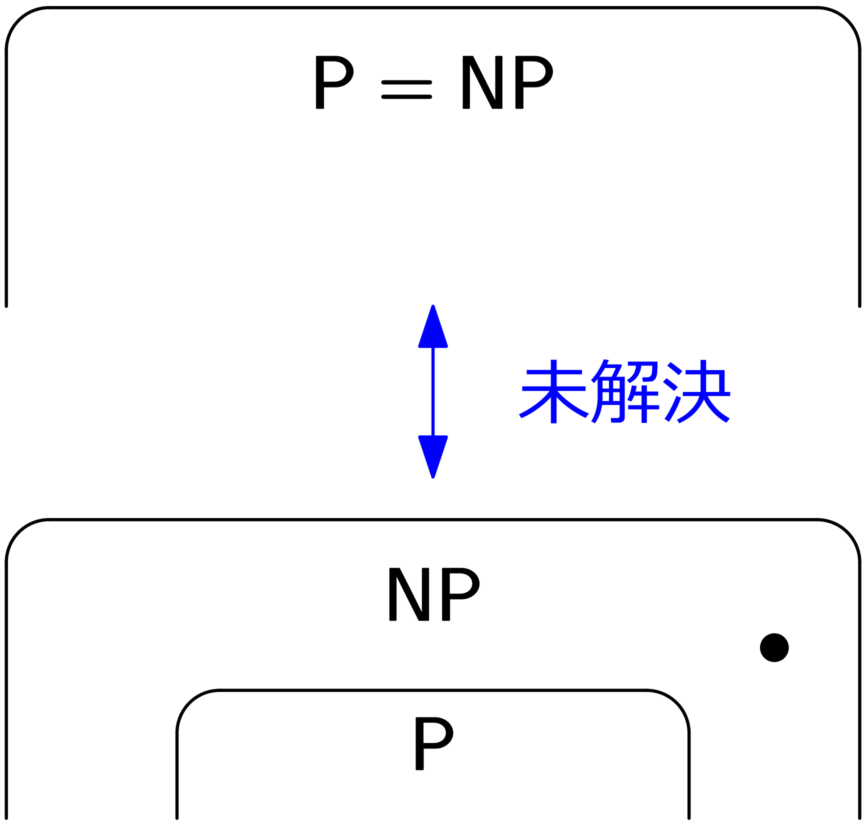


本日のまとめ

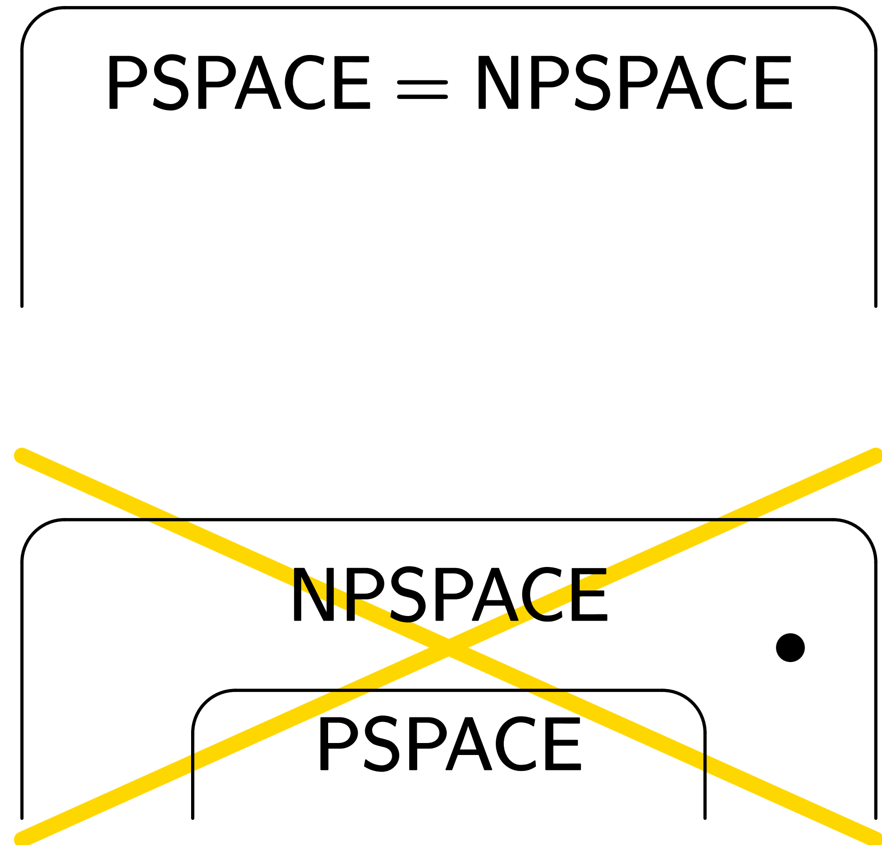
Savitch の定理 ('70)

PSPACE = NPSPACE

時間に対しては



領域に対しては

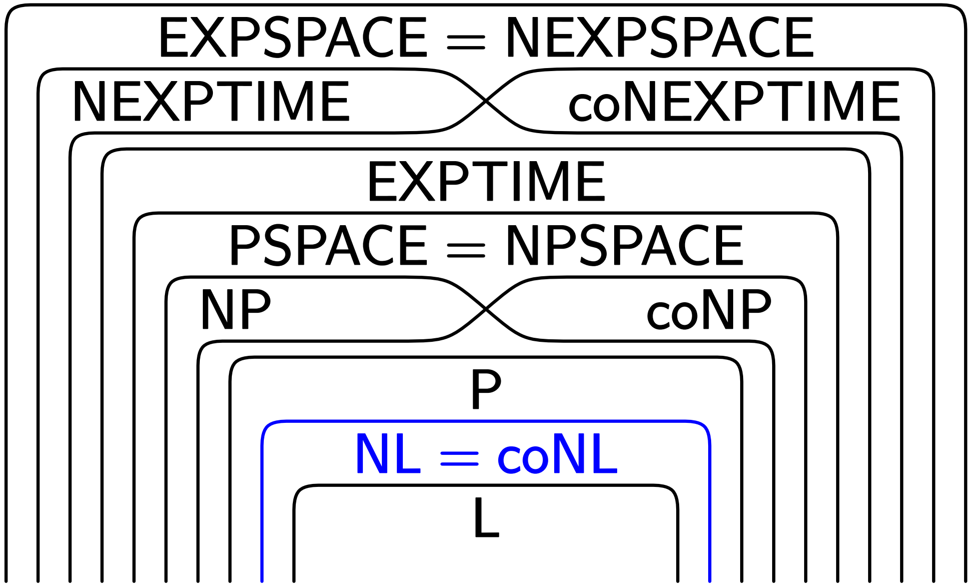


次回の予告

目標

次の「Immerman–Szelepcsényi の定理」を証明する

- $\mathsf{NL} = \mathsf{coNL}$



クイズ

Q

1.

2.

3.

4.