理論計算機科学特論 (2026 年前学期)

計算複雑性の基礎

第 9 回

Immerman-Szelepcsényi の定理：NL = coNL

岡本 吉央　(電気通信大学)
okamotoy@uec.ac.jp

2026 年 6 月 9 日

最終更新：2026 年 6 月 9 日　22:57

スケジュール (前半)

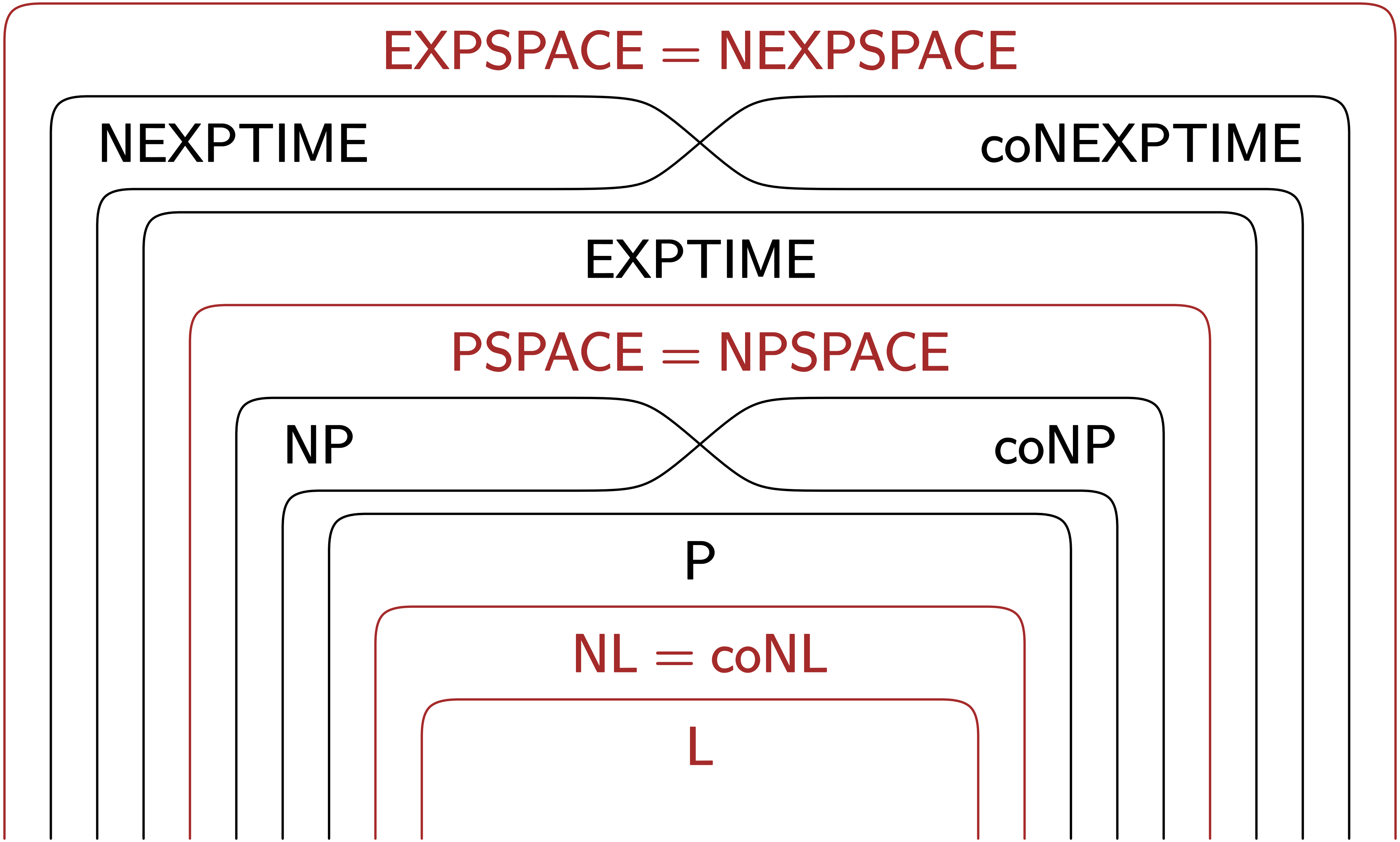
1.	計算理論の復習	(4/7)
2.	時間計算量：P, NP, coNP	(4/14)
3.	帰着と完全性：NP 完全	(4/21)
4.	領域計算量：L, NL, PSPACE	(4/28)
$*$	休み (祝日)	(5/5)
5.	時間と領域の関係：$\mathrm{P} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$	(5/12)
6.	階層定理：$\mathrm{P} \neq \mathrm{EXPTIME}$	(5/19)
7.	Ladner の定理：$\mathrm{NP} - \mathrm{P} = \mathrm{NPC} \Rightarrow \mathrm{P} = \mathrm{NP}$	(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. 交代性計算：AP = PSPACE (6/16)
11. 多項式階層：P = NP ⇒ P = PH (6/23)
12. 確率的計算：P ⊆ BPP ⊆ PP, NP ⊆ PP (6/30)
13. 対話証明系 (1)：NP ⊆ MA ⊆ AM (7/7)
14. 対話証明系 (2)：IP ⊆ PSPACE (7/14)
15. 対話証明系 (3)：PSPACE ⊆ 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：非決定性アルゴリズム

定義 (非形式)：非決定性アルゴリズム

判定問題 P を **解く非決定性アルゴリズム** とは,
任意の入力 I に対して，次を行うもの

- どんな guess をしても，必ず停止する
- I が Yes インスタンス ⇒ うまく guess をすると Yes を出力
- I が No インスタンス ⇒ どんな guess をしても No を出力

$X = \{2, 4, 6, 9\}$

$X = \{1, 2, 3, 6\}$

```
a = guess(X)
b = guess(X)
if a + b == 10:
  return "Yes"
else:
  return "No"
end
```

復習：非決定性アルゴリズム

定義 (非形式)：非決定性アルゴリズム

判定問題 P を **解く非決定性アルゴリズム** とは,
任意の入力 I に対して，次を行うもの

非対称！

- どんな guess をしても，必ず停止する
- I が Yes インスタンス ⇒ $\boxed{\text{うまく guess をすると}}$ Yes を出力
- I が No インスタンス ⇒ $\boxed{\text{どんな guess をしても}}$ No を出力

$\mathsf{X} = \{2, 4, 6, 9\}$

$\mathsf{X} = \{1, 2, 3, 6\}$

```
a = guess(X)
b = guess(X)
if a + b == 10:
  return "Yes"
else:
  return "No"
end
```

復習：補問題

定義：補問題 (complement)

問題 P の **補問題** は，次の問題 $\overline{P}$

入力： P の入力 I

出力： I が P の No インスタンス $\Rightarrow$ Yes

I が P の Yes インスタンス $\Rightarrow$ No

つまり，

I が P の No インスタンス $\Leftrightarrow$ I が $\overline{P}$ の Yes インスタンス

I が P の Yes インスタンス $\Leftrightarrow$ I が $\overline{P}$ の No インスタンス

また，$\overline{\overline{P}} = P$

復習：補問題

定義：補問題 (complement)

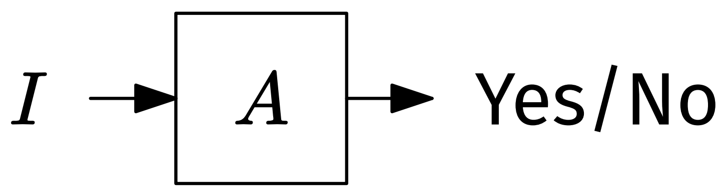
問題 P の **補問題** は，次の問題 $\overline{P}$

入力： P の入力 I

出力： I が P の No インスタンス $\Rightarrow$ Yes

I が P の Yes インスタンス $\Rightarrow$ No

P を解く決定性アルゴリズム A があれば，$\overline{P}$ もすぐ解ける



P を解く決定性アルゴリズム

$\therefore P \in \mathrm{P} \Rightarrow \overline{P} \in \mathrm{P}$ (逆も成立)

復習：補問題

定義：補問題 (complement)

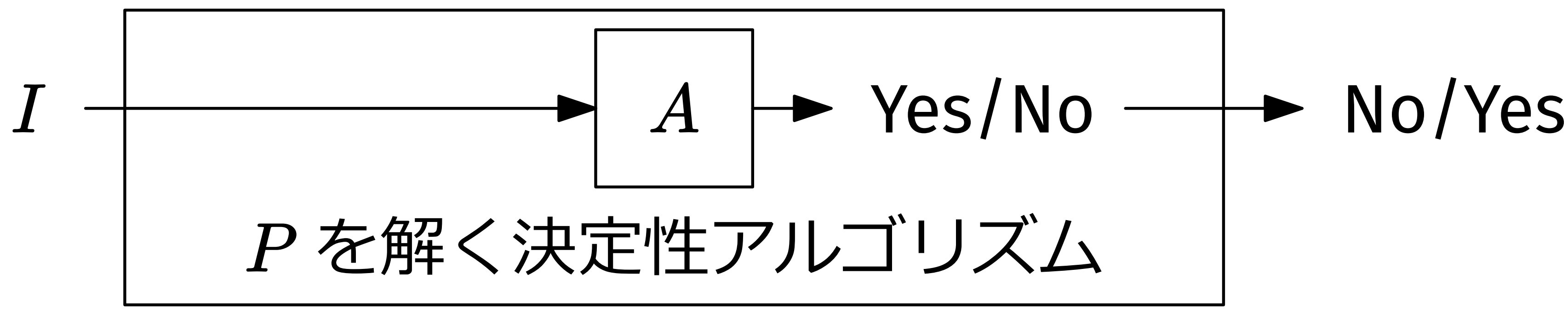
問題 P の **補問題** は，次の問題 $\overline{P}$

入力： P の入力 I

出力： I が P の No インスタンス $\Rightarrow$ Yes

I が P の Yes インスタンス $\Rightarrow$ No

P を解く決定性アルゴリズム A があれば，$\overline{P}$ もすぐ解ける



$\overline{P}$ を解く決定性アルゴリズム

$\therefore P \in \mathrm{P} \Rightarrow \overline{P} \in \mathrm{P}$ (逆も成立)

復習：NL と coNL

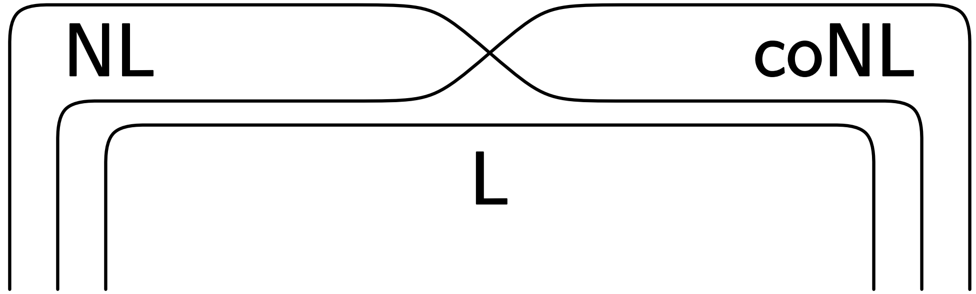
定義：クラス NL

クラス NL とは，
非決定性を持つ対数領域アルゴリズムで解ける
判定問題全体のこと

定義：クラス coNL

クラス coNL とは，
補問題が NL に属するような判定問題全体のこと

つまり， $P \in \mathsf{NL} \Leftrightarrow \overline{P} \in \mathsf{coNL}$



Immerman–Szelepcsényi の定理 ('87)

NL = coNL

読み方

- Immerman : **イ**マーマン
- Szelepcsényi : **セ**レプチェーニ

Immerman–Szelepcsényi の定理 ('87)

NL = coNL

領域に対しては

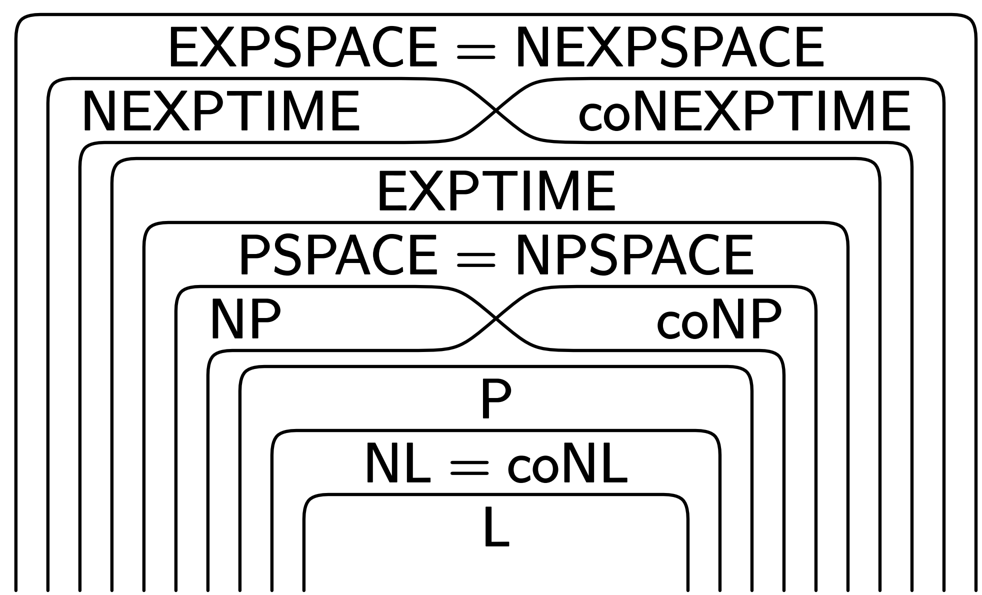
- NEXPSPACE = coNEXPSPACE (= EXPSPACE)
- NPSPACE = coNPSPACE (= PSPACE)

Savitch

- NL = coNL

時間に対しては

- NEXPTIME $\stackrel{?}{=}$ coNEXPTIME
- NP $\stackrel{?}{=}$ coNP



NL = coNL：証明の方針

証明の流れ

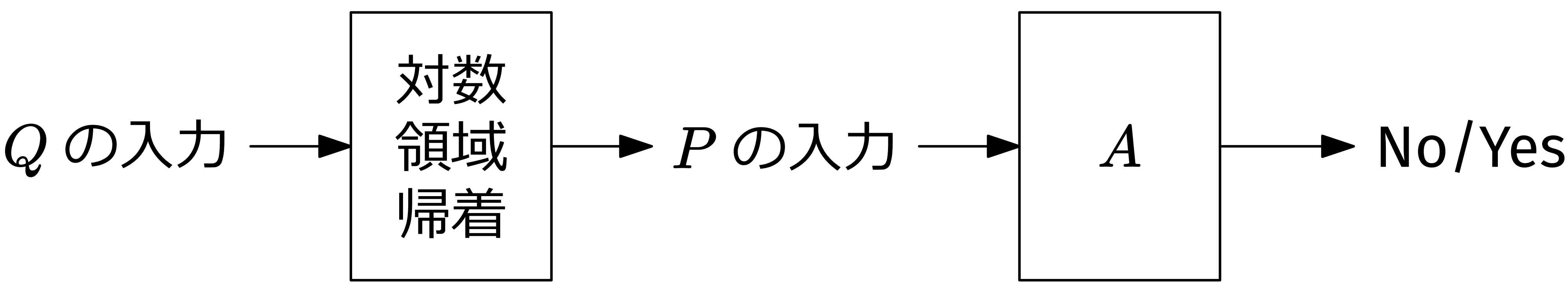
1. ある NL 完全問題 P を考える
2. $\overline{P}$ を解く非決定性対数領域アルゴリズム A を作る

NL = coNL：証明の方針

証明の流れ

1. ある NL 完全問題 P を考える
2. $\overline{P}$ を解く非決定性対数領域アルゴリズム A を作る

任意の問題 $Q \in \mathrm{NL}$ に対して

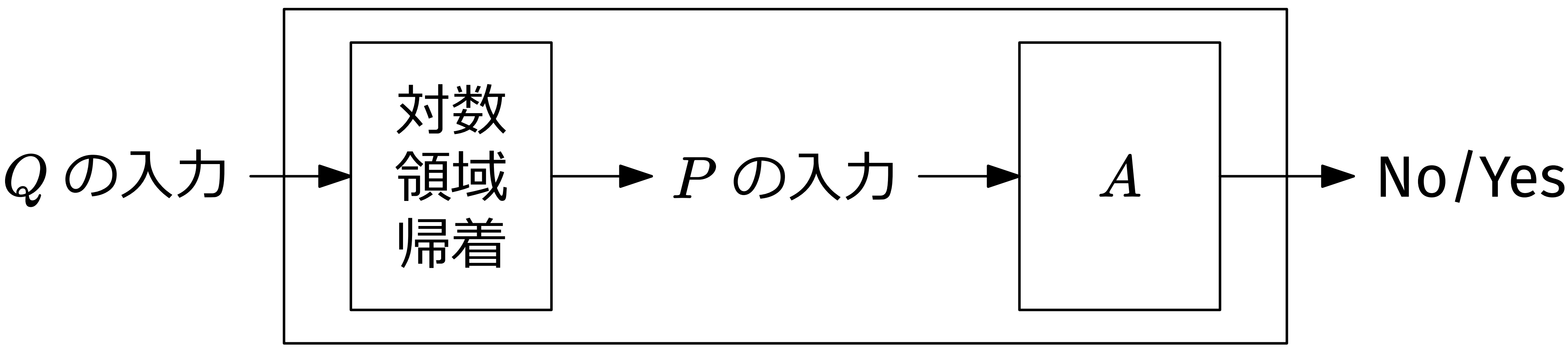


NL = coNL：証明の方針

証明の流れ

1. ある NL 完全問題 P を考える
2. $\overline{P}$ を解く非決定性対数領域アルゴリズム A を作る

任意の問題 $Q \in$ NL に対して



$\overline{Q}$ を解く非決定性対数領域アルゴリズム

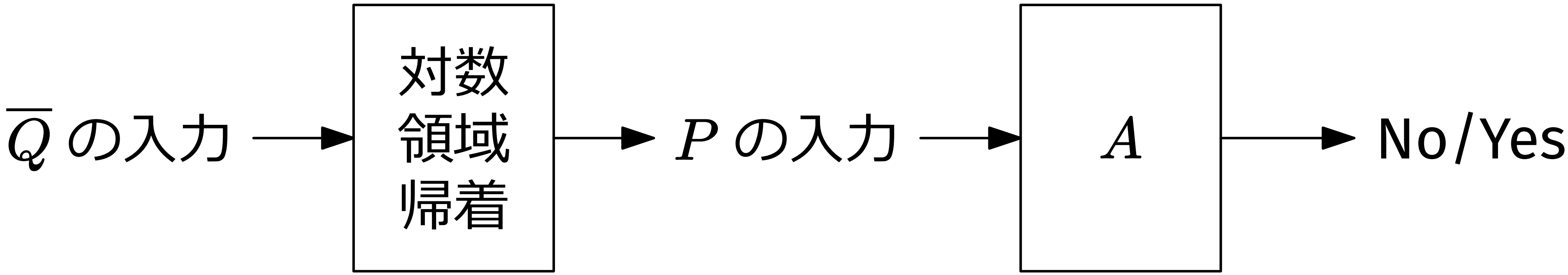
$\therefore \quad \overline{Q} \in$ NL (すなわち，$Q \in$ coNL)

NL = coNL：証明の方針

証明の流れ

1. ある NL 完全問題 P を考える
2. $\overline{P}$ を解く非決定性対数領域アルゴリズム A を作る

任意の問題 $Q \in$ coNL に対して，$\overline{Q} \in$ NL なので

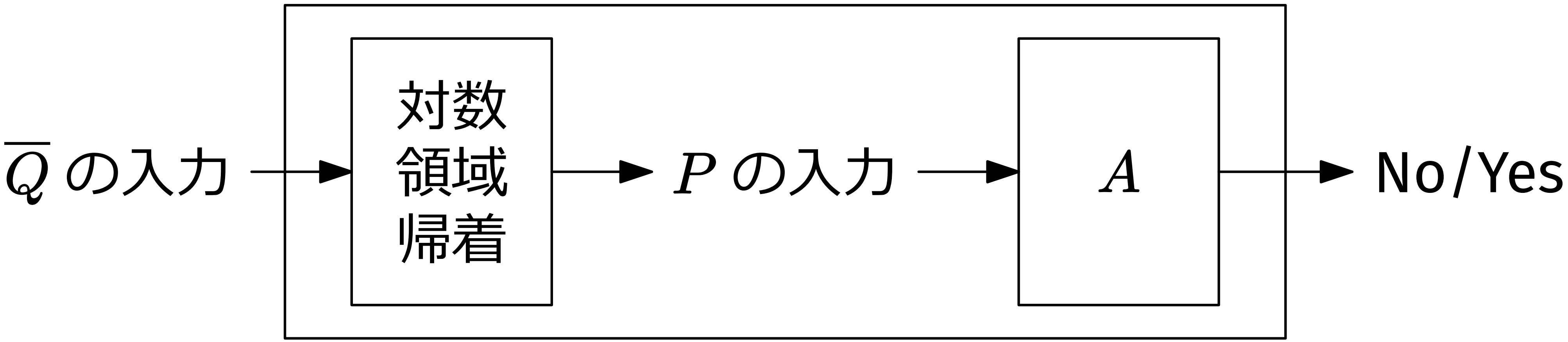


NL = coNL：証明の方針

証明の流れ

1. ある NL 完全問題 P を考える
2. $\overline{P}$ を解く非決定性対数領域アルゴリズム A を作る

任意の問題 $Q \in$ coNL に対して，$\overline{Q} \in$ NL なので



$\overline{\overline{Q}}$ を解く非決定性対数領域アルゴリズム

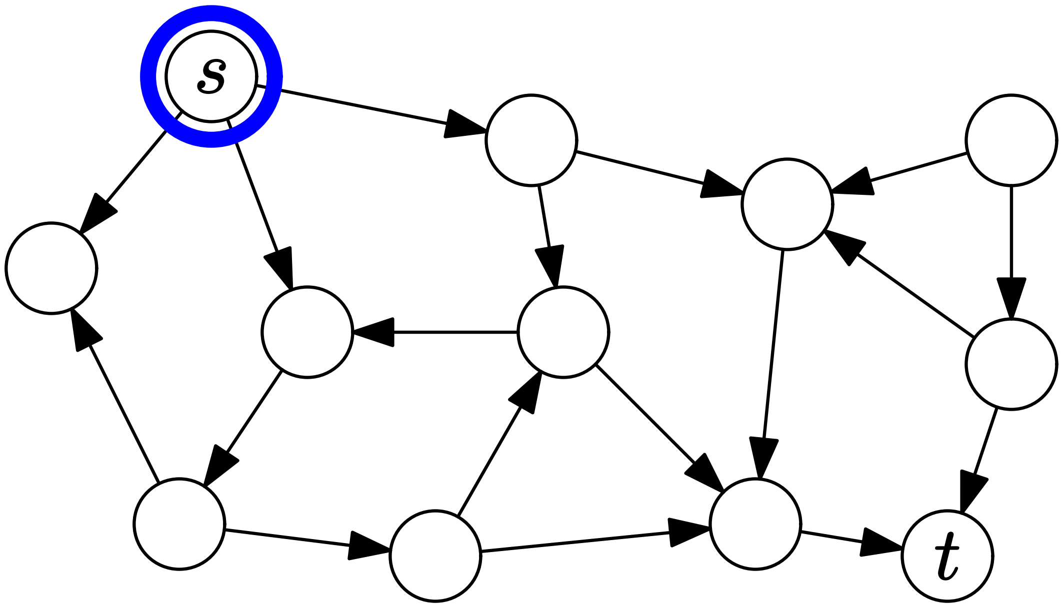
$\therefore \quad Q \in$ NL

復習：STCON は NL 完全

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



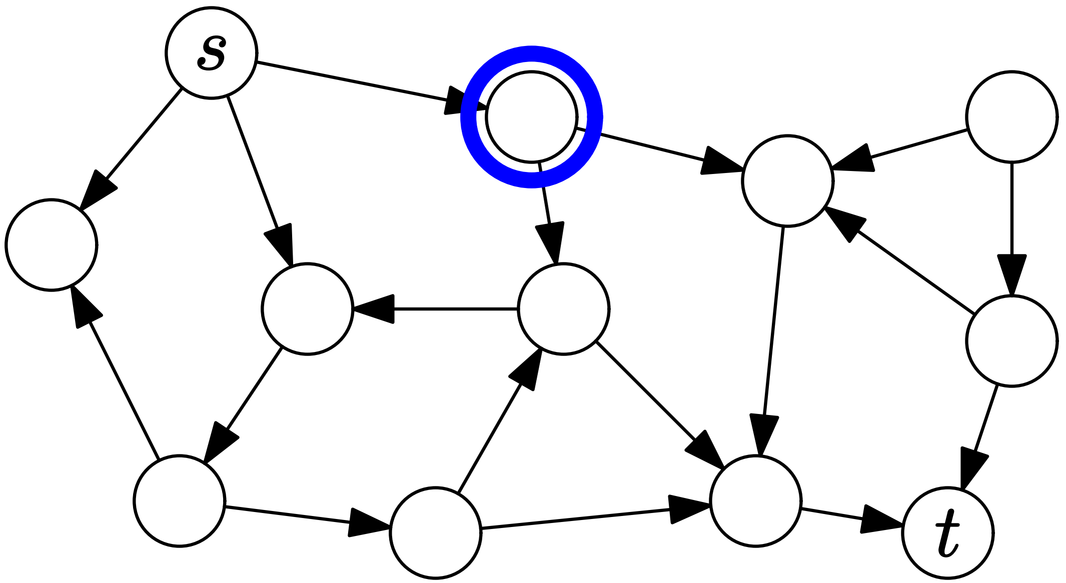
復習：STCON は NL 完全

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



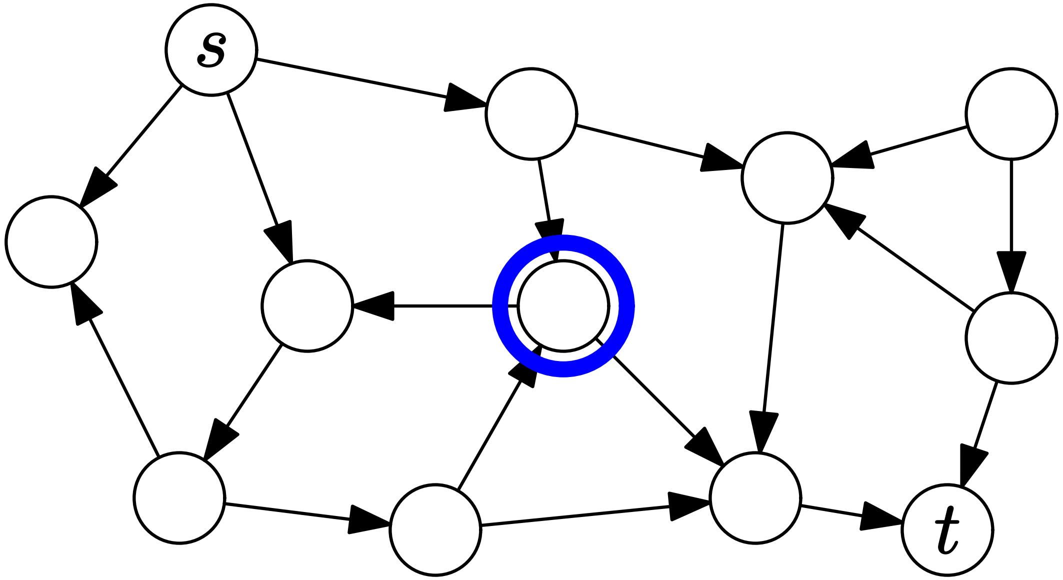
復習：STCON は NL 完全

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



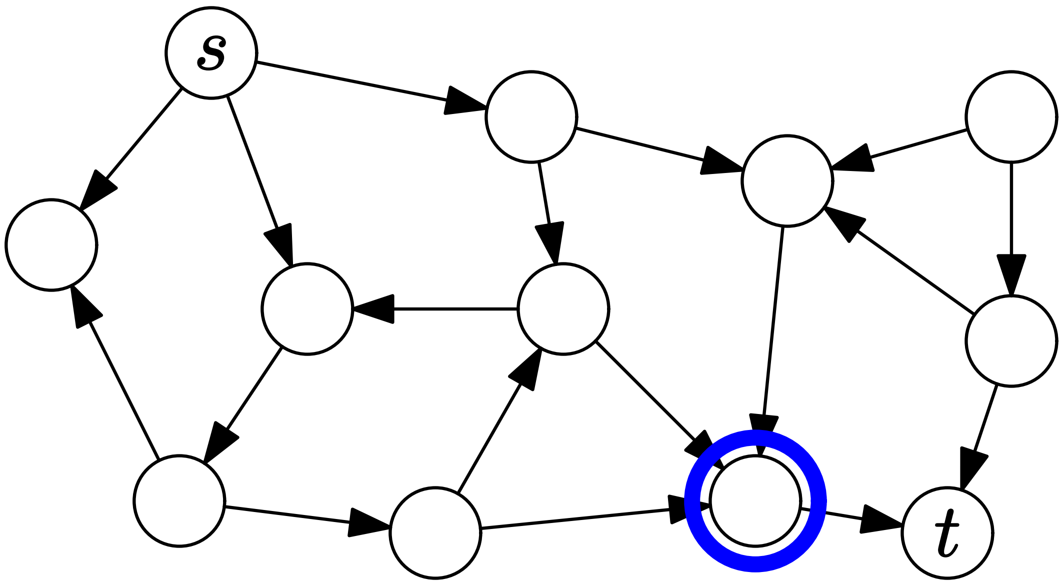
復習：STCON は NL 完全

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

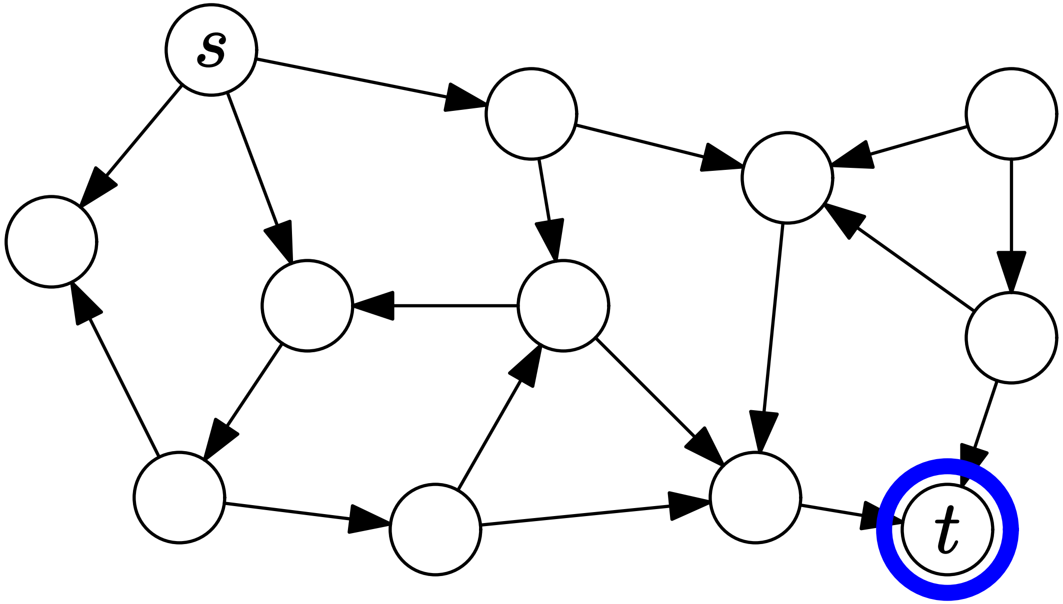
s から t へ至る経路が存在しない $\Rightarrow$ No



復習：STCON は NL 完全

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



本日の内容

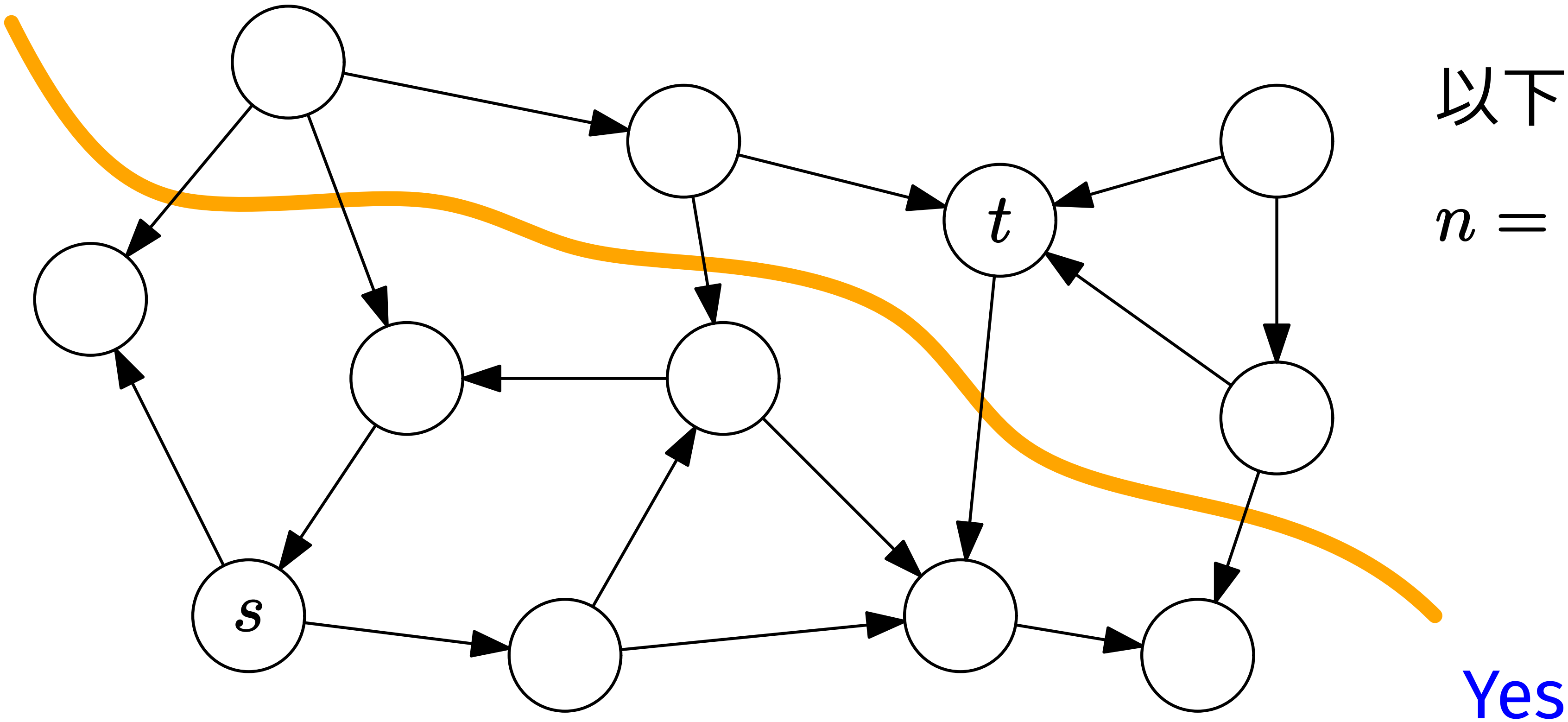
1. **Immerman–Szelepcsényi の定理：証明の考え方**
2. Immerman–Szelepcsényi の定理：アルゴリズム
3. Immerman–Szelepcsényi の定理：補足

STCON の補問題

$\overline{\text{STCON}}$：STCON の補問題

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

出力： s から t へ至る経路が存在しない $\Rightarrow$ Yes
s から t へ至る経路が存在する $\Rightarrow$ No



以下，

$n = |V|$ とする

$\overline{\text{STCON}}$：作りたいアルゴリズム

$\overline{\text{STCON}}$：STCON の補問題

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

作りたいアルゴリズム が 満たすべき性質

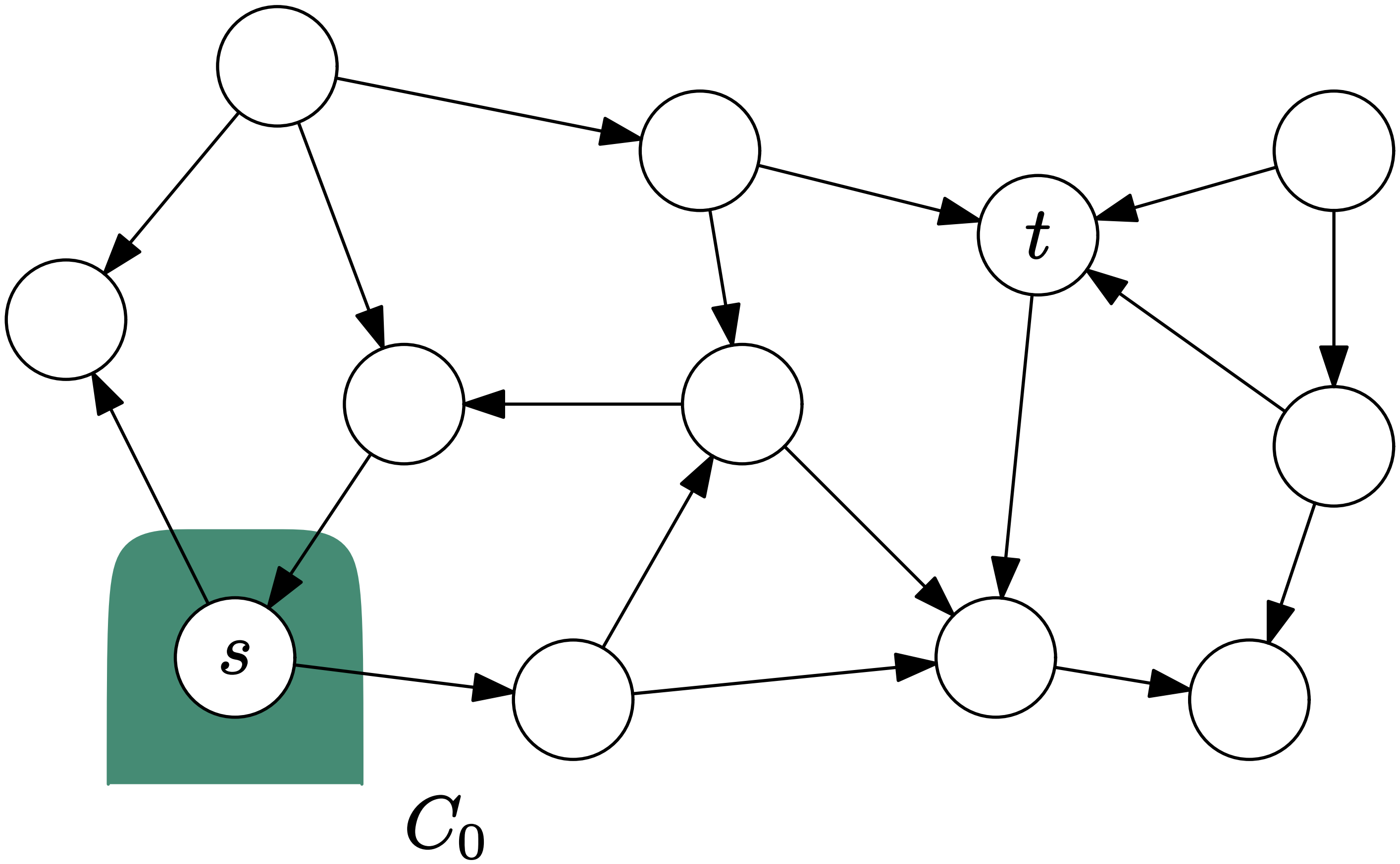
1. $O(\log n)$ 領域で動く，非決定性を持つ $(n=|V|)$
2. G において，s から t へ至る経路が存在しない
$\Rightarrow$ ある guess に対して Yes を出力
3. G において，s から t へ至る経路が存在する
$\Rightarrow$ どんな guess に対しても No を出力

NL = coNL：アイディア 1

アイディア 1

各非負整数 $i = 0, 1, \ldots, n$ に対して，次を考える

$$C_i = \{v \in V \mid s \text{ から } v \text{ に至る長さ } i \text{ 以下の経路が存在}\}$$

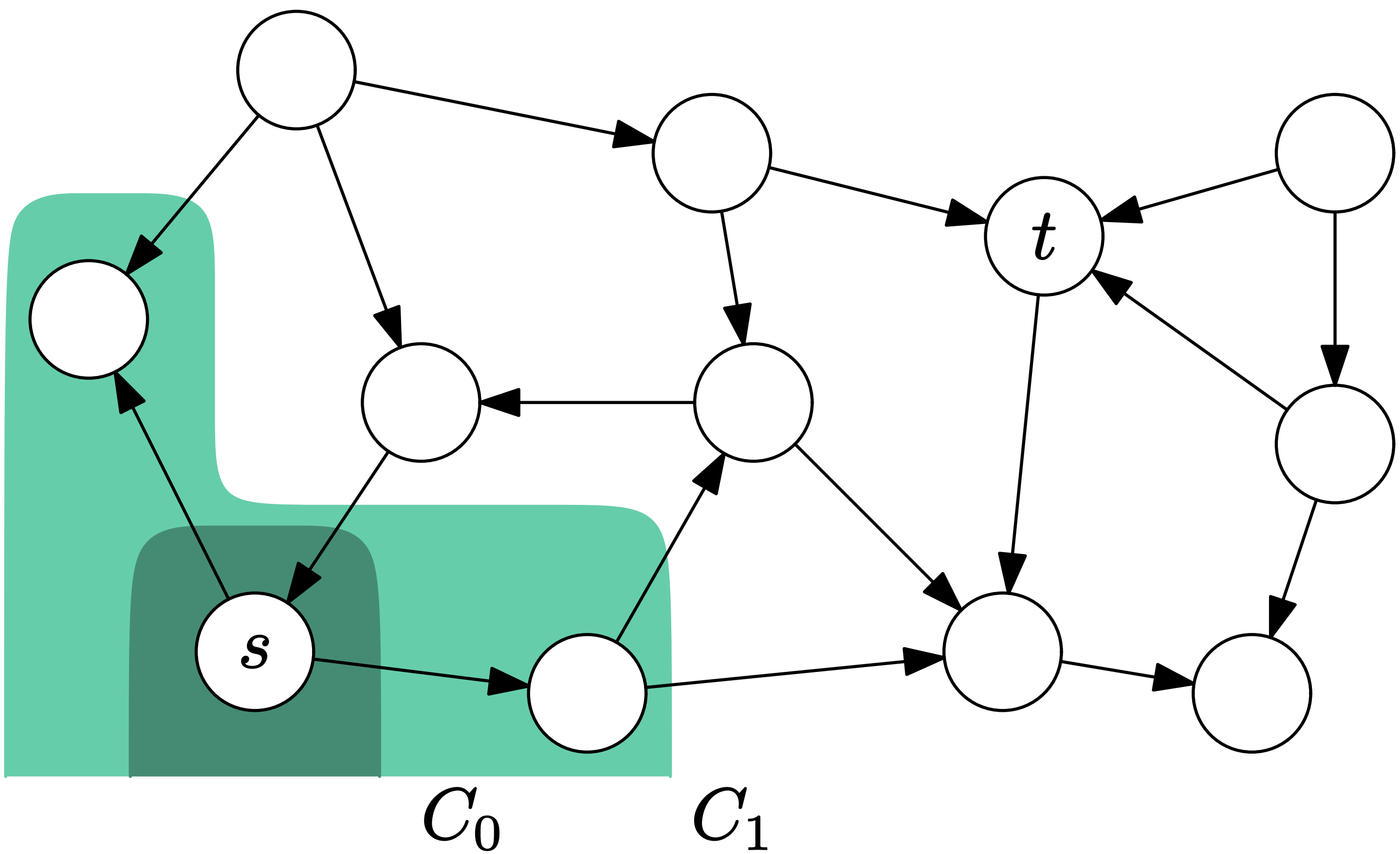


NL = coNL：アイディア 1

アイディア 1

各非負整数 $i = 0, 1, \ldots, n$ に対して，次を考える

$$C_i = \{v \in V \mid s \text{ から } v \text{ に至る長さ } i \text{ 以下の経路が存在}\}$$

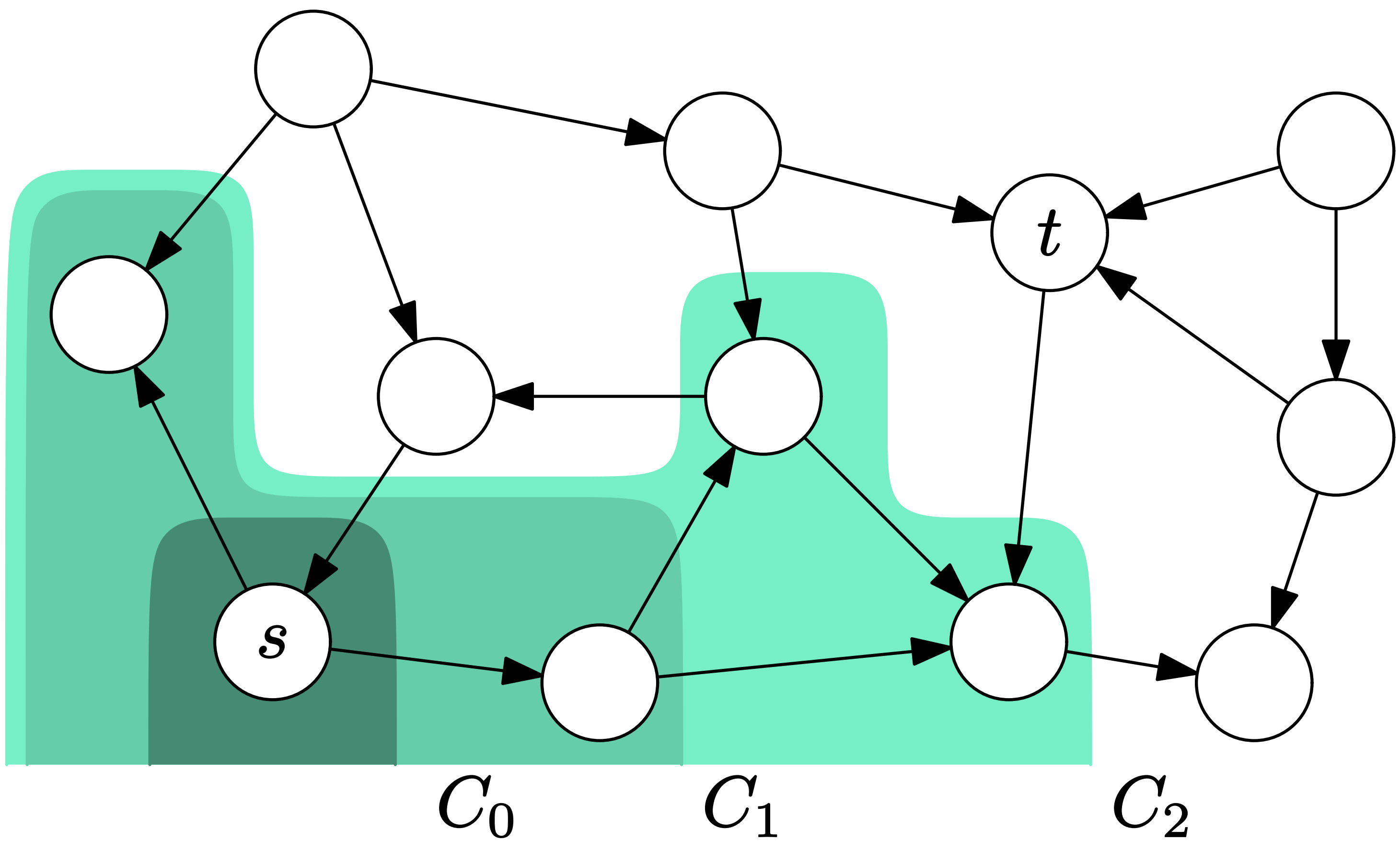


NL = coNL：アイディア 1

アイディア 1

各非負整数 $i = 0, 1, \ldots, n$ に対して，次を考える

$$C_i = \{v \in V \mid s \text{ から } v \text{ に至る長さ } i \text{ 以下の経路が存在}\}$$

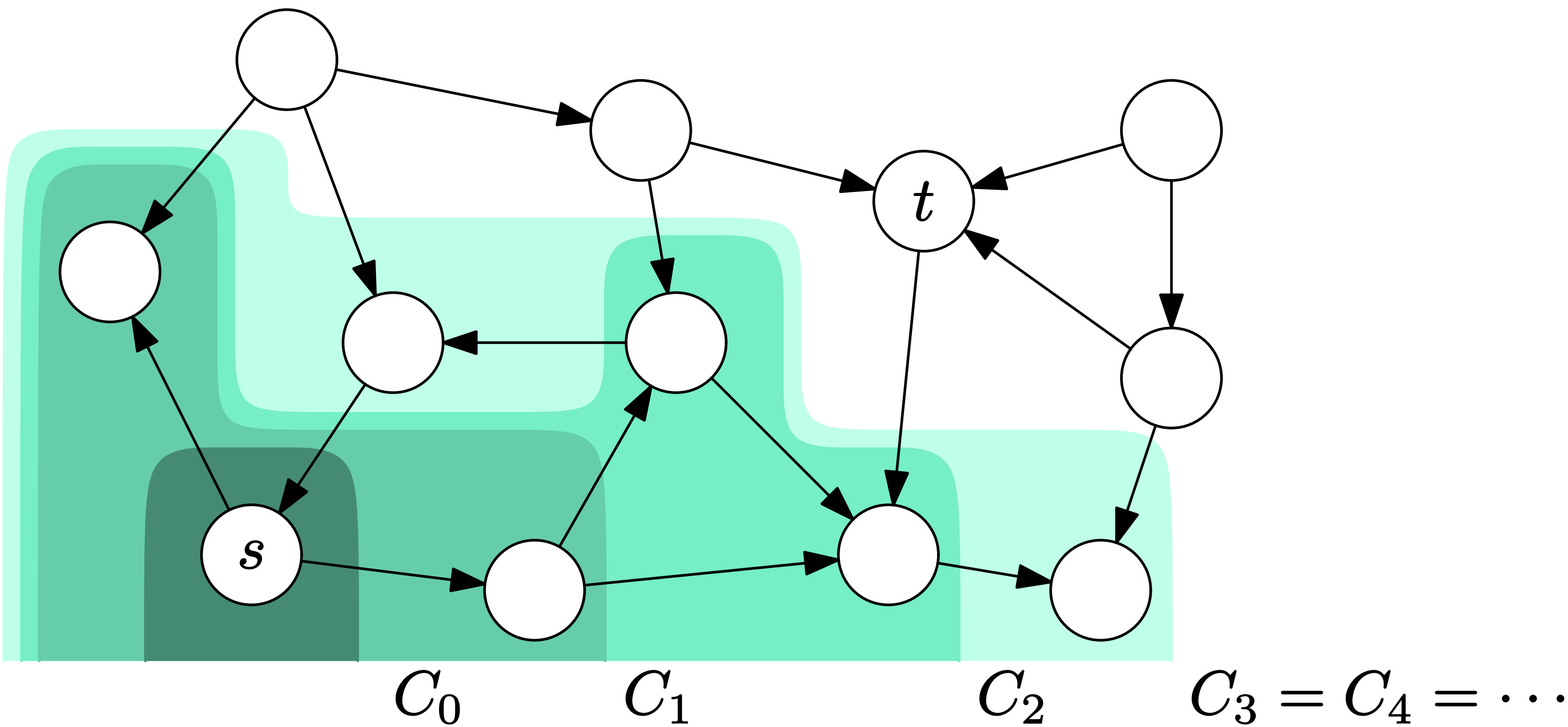


NL = coNL：アイディア 1

アイディア 1

各非負整数 $i = 0, 1, \ldots, n$ に対して，次を考える

$$C_i = \{v \in V \mid s \text{ から } v \text{ に至る長さ } i \text{ 以下の経路が存在}\}$$

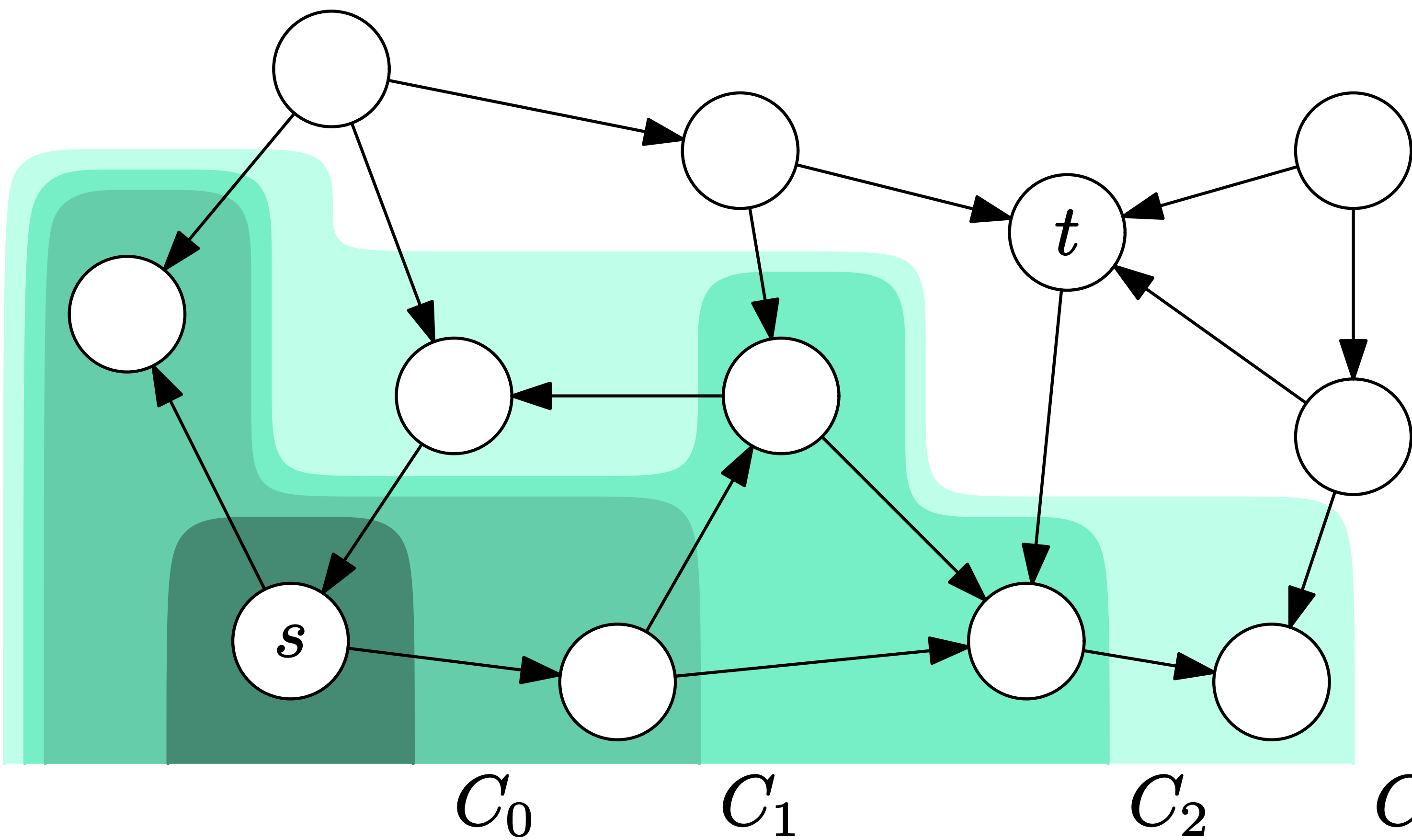


NL = coNL：アイディア 1

アイディア 1

各非負整数 $i = 0, 1, \ldots, n$ に対して，次を考える

$$C_i = \{v \in V \mid s \text{ から } v \text{ に至る長さ } i \text{ 以下の経路が存在}\}$$



アルゴリズムの目標：

$t \notin C_n$

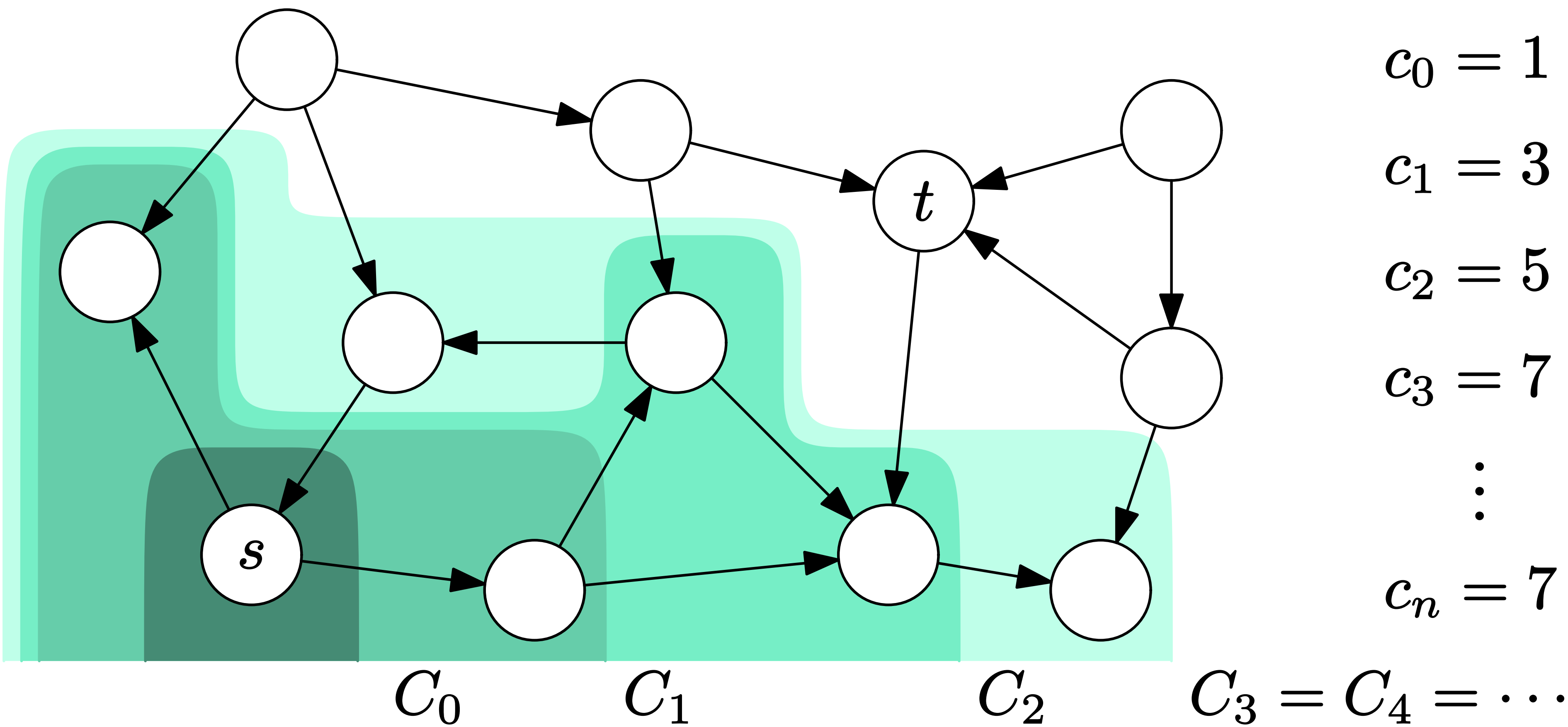
を示す

NL = coNL：アイディア 2

アイディア 2

各 $i = 0, 1, \ldots, n$ に対して，次のみを作業領域に保持する

$$c_i = |C_i|$$



$c_0 = 1$

$c_1 = 3$

$c_2 = 5$

$c_3 = 7$

$\vdots$

$c_n = 7$

NL = coNL : アルゴリズムのひな型

アルゴリズム

1. $c_0 = 1$
2. $i = 1, \ldots, n$ に対して順に，c_{i-1} から c_i を計算
3. c_n から $t \notin C_n$ を判定

NL = coNL：アルゴリズムのひな型

アルゴリズム

1. $c_0 = 1$
2. $i = 1, \ldots, n$ に対して順に，$\boxed{c_{i-1} \text{ から } c_i \text{ を計算}}$
3. $\boxed{c_n \text{ から } t \notin C_n \text{ を判定}}$

この 2 つを

非決定性対数領域アルゴリズムで

行う

アルゴリズム

1. $c_0 = 1$
2. $i = 1, \ldots, n$ に対して順に, $\boxed{c_{i-1} \text{ から } c_i \text{ を計算}}$
3. $\boxed{c_n \text{ から } t \notin C_n \text{ を判定}}$

部分問題 2

部分問題 1

この 2 つを

非決定性対数領域アルゴリズムで

行う

NL = coNL：アルゴリズムのひな型

アルゴリズム

1. $c_0 = 1$
2. $i = 1, \ldots, n$ に対して順に，$\boxed{c_{i-1} \text{ から } c_i \text{ を計算}}$
3. $\boxed{c_n \text{ から } t \notin C_n \text{ を判定}}$

部分問題 2

部分問題 1

この 2 つを

非決定性対数領域アルゴリズムで

行う

ポイント

部分問題 2 を解くこと

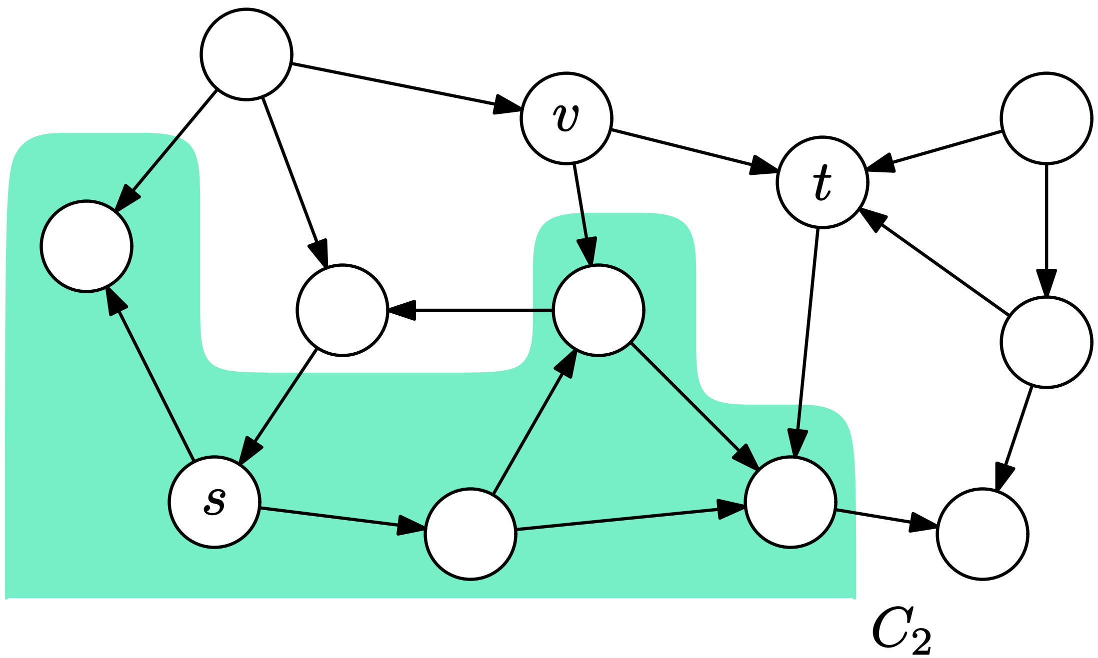
(帰納的数え上げ inductive counting と呼ばれる手法)

部分問題 1

部分問題 1：c_i から $v \notin C_i$

入力： 有向グラフ $G = (V, A)$，2 頂点 $s, t \in V$，
頂点 $v \in V$，添字 i，C_i の要素数 c_i

出力： $v \notin C_i \Rightarrow$ Yes
$v \in C_i \Rightarrow$ No

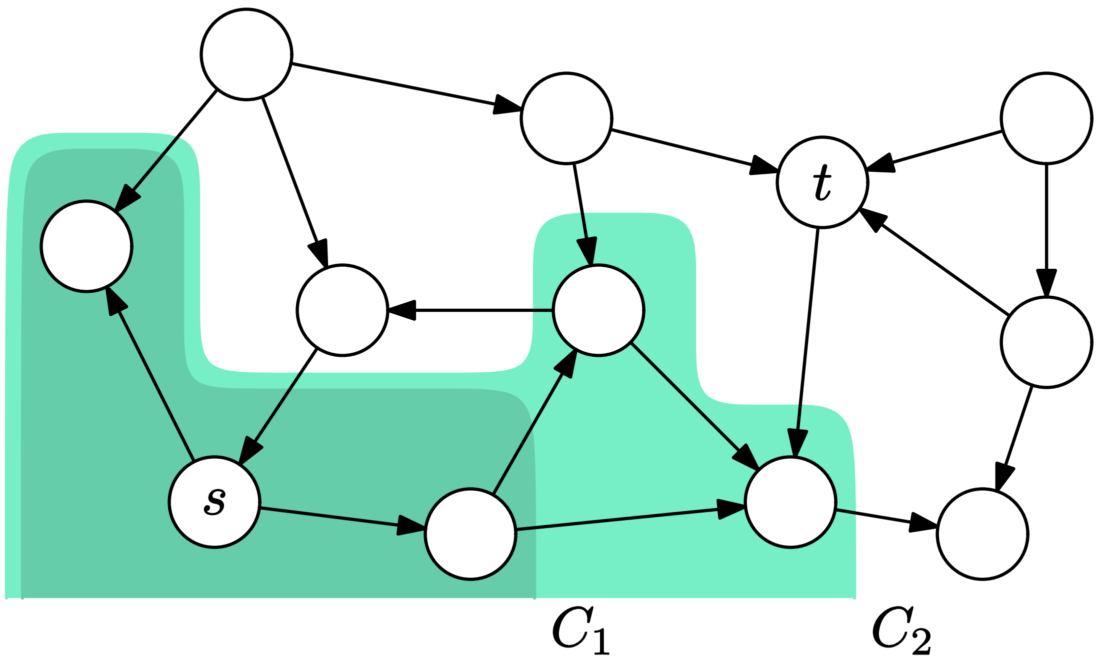


部分問題 2

部分問題 2：c_{i-1} から c_i

入力： 有向グラフ $G = (V, A)$，2 頂点 $s, t \in V$，
添字 i，c_{i-1}

出力： c_i



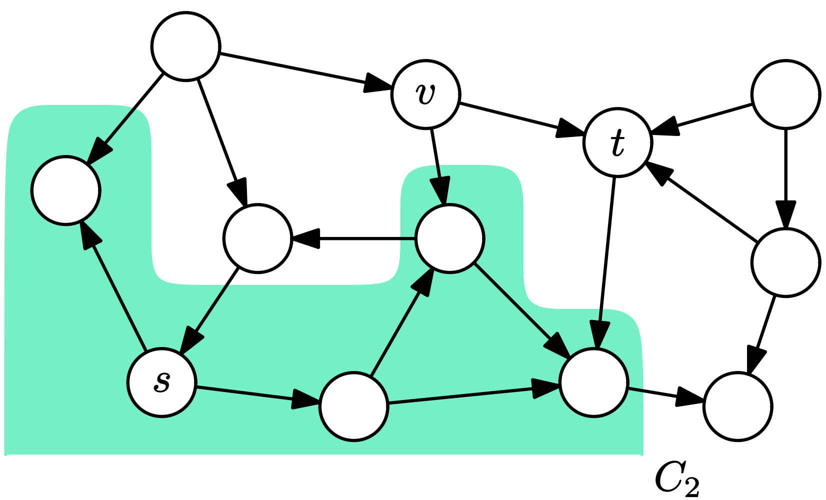
本日の内容

1. Immerman–Szelepcsényi の定理：証明の考え方
2. **Immerman–Szelepcsényi の定理：アルゴリズム**
3. Immerman–Szelepcsényi の定理：補足

部分問題 1：アルゴリズム

<u>アルゴリズム</u>

1. $d := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - s から u へ至る長さ i 以下の経路を guess して，検証
   - $u \in C_i \Rightarrow d := d + 1$
   - $u = v$ のときの結果を覚えておく
3. $d \neq c_i$ または $v \in C_i$ と判定 $\Rightarrow$ No を出力
4. そうでない $\Rightarrow$ Yes を出力



部分問題 1：c_i から $v \notin C_i$

入力： 有向グラフ $G = (V, A)$，2 頂点 $s, t \in V$，頂点 $v \in V$，添字 i，C_i の要素数 c_i

出力： $v \notin C_i \Rightarrow$ Yes
$v \in C_i \Rightarrow$ No

部分問題 1：領域計算量

アルゴリズム

1. $d := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - s から u へ至る長さ i 以下の経路を guess して，検証
   - $u \in C_i \Rightarrow d := d + 1$
   - $u = v$ のときの結果を覚えておく
3. $d \neq c_i$ または $v \in C_i$ と判定 $\Rightarrow$ No を出力
4. そうでない $\Rightarrow$ Yes を出力

用いる領域 (変数)

• d	$O(\log n)$ ビット
• $u = v$ のときの結果	$O(1)$ ビット
• 経路の guess と検証	$O(\log n)$ ビット

$\therefore$ このアルゴリズムの領域計算量 $= O(\log n)$ ビット

部分問題 1：正当性 (Yes インスタンス)

アルゴリズム

1. $d := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - s から u へ至る長さ i 以下の経路を guess して，検証
   - $u \in C_i \Rightarrow d := d + 1$
   - $u = v$ のときの結果を覚えておく
3. $d \neq c_i$ または $v \in C_i$ と判定 $\Rightarrow$ No を出力
4. そうでない $\Rightarrow$ Yes を出力

$v \notin C_i$ である入力に対して

- $u \in C_i$ となるすべての u に対して，正しい guess を行えば，$d = c_i$ が成り立つ
- $\therefore$ そのとき，このアルゴリズムは Yes を出力する

部分問題 1：正当性 (No インスタンス)

アルゴリズム

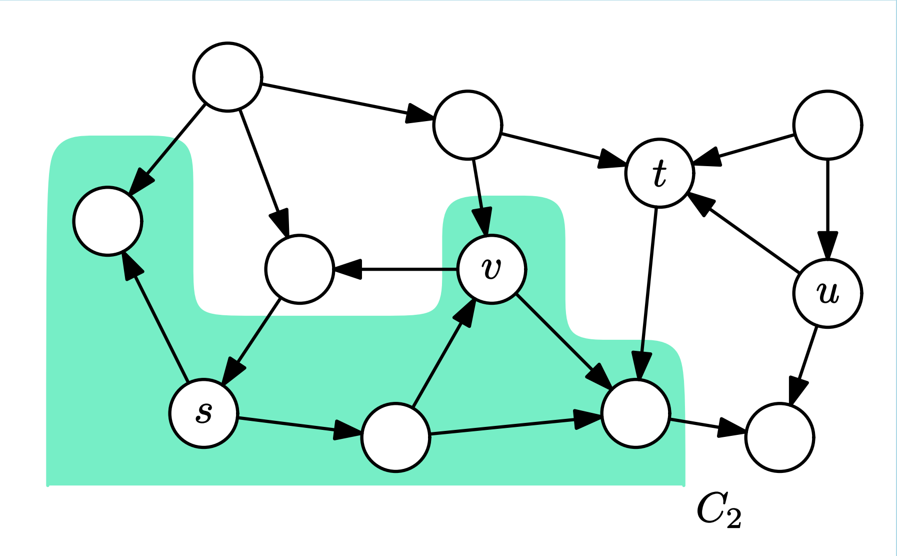
1. $d := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - s から u へ至る長さ i 以下の経路を guess して，検証
   - $u \in C_i \Rightarrow d := d + 1$
   - $u = v$ のときの結果を覚えておく
3. $d \neq c_i$ または $v \in C_i$ と判定 $\Rightarrow$ No を出力
4. そうでない $\Rightarrow$ Yes を出力

$v \in C_i$ である入力に対して，Yes を出力したと仮定する

- アルゴリズムは $d = c_i$ かつ $v \notin C_i$ と判定
- v に対する経路の guess は間違っていて，
  $u \notin C_i$ を満たすある頂点 u に対する経路の guess も間違っている
- 矛盾

部分問題 1：正当性 (No インスタンス)

<u>アルゴリズム</u>



1. $d := 0$
2. すべての頂点 u に対して (ラベ
   - s から u へ至る長さ i 以下の
   - $u \in C_i \Rightarrow d := d + 1$
   - $u = v$ のときの結果を覚えて
3. $d \neq c_i$ または $v \in C_i$ と判定 $\Rightarrow$ No を出力
4. そうでない $\Rightarrow$ Yes を出力

$v \in C_i$ である入力に対して，Yes を出力したと仮定する

- アルゴリズムは $d = c_i$ かつ $v \notin C_i$ と判定
- v に対する経路の guess は間違っていて，
  <u>$u \notin C_i$ を満たすある頂点 u に対する経路の guess も間違っている</u> これは不可能
- 矛盾

部分問題 1：まとめ

アルゴリズム

1. $d := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - s から u へ至る長さ i 以下の経路を guess して，検証
   - $u \in C_i \Rightarrow d := d + 1$
   - $u = v$ のときの結果を覚えておく
3. $d \neq c_i$ または $v \in C_i$ と判定 $\Rightarrow$ No を出力
4. そうでない $\Rightarrow$ Yes を出力

$\therefore$ このアルゴリズムは部分問題 1 を解く

非決定性対数領域アルゴリズムである

部分問題 1：c_i から $v \notin C_i$

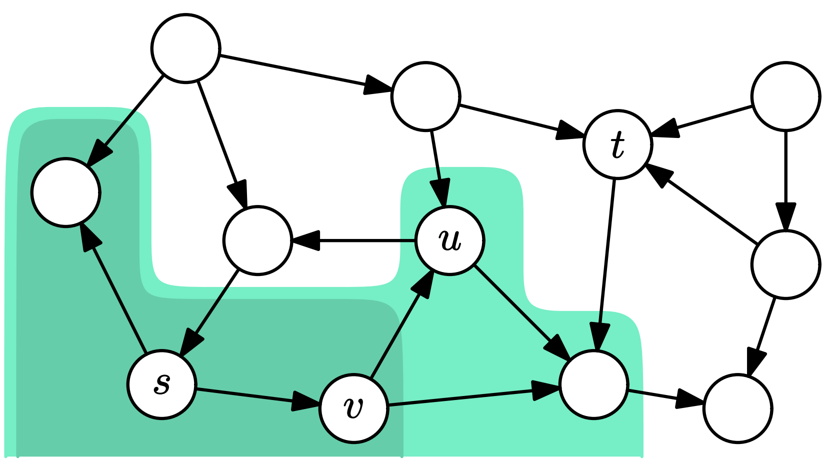
入力： 有向グラフ $G = (V, A)$，2 頂点 $s, t \in V$，
頂点 $v \in V$，添字 i，C_i の要素数 c_i

出力： $v \notin C_i \Rightarrow$ Yes
$v \in C_i \Rightarrow$ No

部分問題 2：アルゴリズム

アルゴリズム

1. $c_i := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - $u \in C_i \Rightarrow c_i = c_i + 1$
3. c_i を出力



部分問題 2：c_{i-1} から c_i

入力： 有向グラフ $G = (V, A)$，2 頂点 $s, t \in V$，
添字 i，c_{i-1}

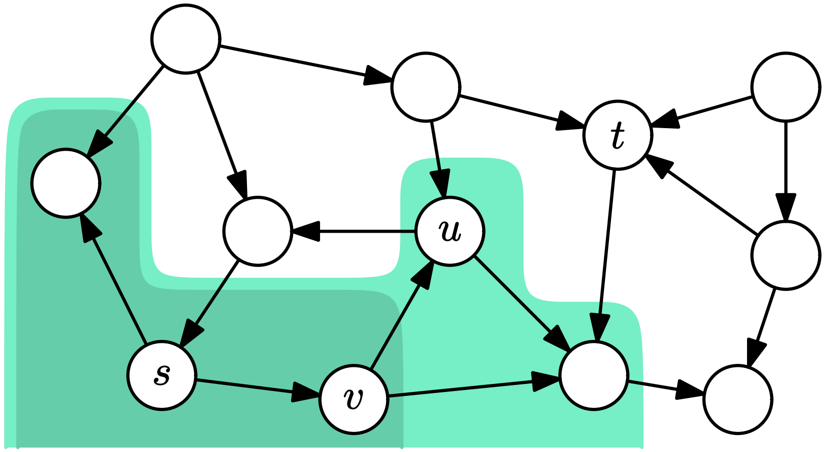
出力： c_i

部分問題 2：アルゴリズム

アルゴリズム

1. $c_i := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   2-1. $d := 0$
   2-2. すべての頂点 v に対して (ラベルの小さい順に)
   - $v \in C_{i-1} \Rightarrow d := d + 1$
   - $v \in C_{i-1}$ かつ「$u = v$ または $(v, u) \in A$」$\Rightarrow$ フラグ ON

   2-3. $c_{i-1} = d$ かつフラグ ON $\Rightarrow c_i := c_i + 1$
3. c_i を出力



部分問題 2：c_{i-1} から c_i

入力： 有向グラフ $G = (V, A)$，2 頂点 $s, t \in V$，添字 i，c_{i-1}

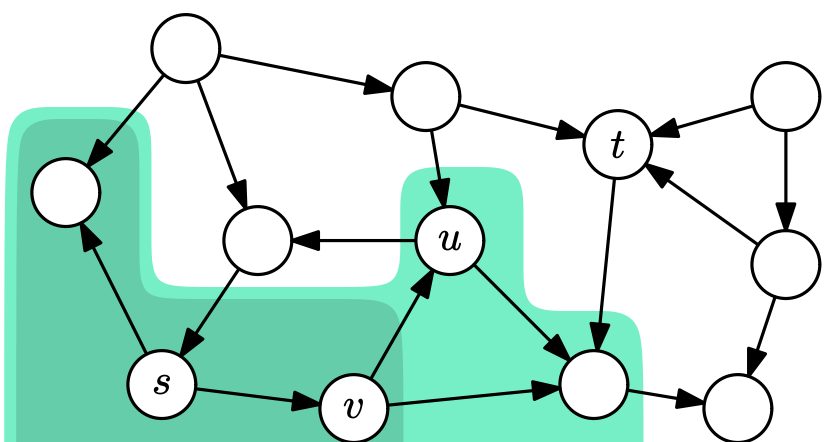
出力： c_i

部分問題 2：アルゴリズム

アルゴリズム

長さ $i-1$ 以下の経路を guess して検証

1. $c_i := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - 2-1. $d := 0$
   - 2-2. すべての頂点 v に対して (ラベルの小さい順に)
     - $\boxed{v \in C_{i-1}} \Rightarrow d := d + 1$
     - $v \in C_{i-1}$ かつ「$u = v$ または $(v, u) \in A$」$\Rightarrow$ フラグ ON
   - 2-3. $c_{i-1} = d$ かつフラグ ON $\Rightarrow c_i := c_i + 1$
3. c_i を出力



部分問題 2：c_{i-1} から c_i

入力： 有向グラフ $G = (V, A)$，2 頂点 $s, t \in V$，添字 i，c_{i-1}

出力： c_i

部分問題 2：領域計算量

アルゴリズム

1. $c_i := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - 2-1. $d := 0$
   - 2-2. すべての頂点 v に対して (ラベルの小さい順に)
     - $v \in C_{i-1} \Rightarrow d := d + 1$
     - $v \in C_{i-1}$ かつ「$u = v$ または $(v, u) \in A$」$\Rightarrow$ フラグ ON
   - 2-3. $c_{i-1} = d$ かつフラグ ON $\Rightarrow c_i := c_i + 1$
3. c_i を出力

用いる領域 (変数)

- c_i, d $O(\log n)$ ビット
- フラグ $O(1)$ ビット
- 経路の guess と検証 $O(\log n)$ ビット

$\therefore$ このアルゴリズムの領域計算量 $= O(\log n)$ ビット

部分問題 2：guess が正しいとき

アルゴリズム

1. $c_i := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - 2-1. $d := 0$
   - 2-2. すべての頂点 v に対して (ラベルの小さい順に)
     - $v \in C_{i-1} \Rightarrow d := d + 1$
     - $v \in C_{i-1}$ かつ「$u = v$ または $(v, u) \in A$」$\Rightarrow$ フラグ ON
   - 2-3. $c_{i-1} = d$ かつフラグ ON $\Rightarrow c_i := c_i + 1$
3. c_i を出力

$v \in C_{i-1}$ を満たす v に対して，2-2 で正しく guess する

- $\Rightarrow$ 正しい c_i を出力する

部分問題 2：guess が正しくないとき

<u>アルゴリズム</u>

1. $c_i := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - 2-1. $d := 0$
   - 2-2. すべての頂点 v に対して (ラベルの小さい順に)
     - $v \in C_{i-1} \Rightarrow d := d + 1$
     - $v \in C_{i-1}$ かつ「$u = v$ または $(v, u) \in A$」$\Rightarrow$ フラグ ON
   - 2-3. $c_{i-1} = d$ かつフラグ ON $\Rightarrow c_i := c_i + 1$
3. c_i を出力

guess が正しくないとき

- 本当は $v \in C_{i-1}$ であるのに，$v \notin C_{i-1}$ と判定してしまう
- $\therefore c_{i-1} = d$ とならない
- このとき，No を出力して，全体を停止するよう変更

$\therefore$ アルゴリズムが出力を行うとき，出力 c_i は必ず正しい

部分問題 2：まとめ

<u>アルゴリズム</u>

1. $c_i := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - 2-1. $d := 0$
   - 2-2. すべての頂点 v に対して (ラベルの小さい順に)
     - $v \in C_{i-1} \Rightarrow d := d + 1$
     - $v \in C_{i-1}$ かつ「$u = v$ または $(v, u) \in A$」$\Rightarrow$ フラグ ON
   - 2-3. $c_{i-1} \neq d \Rightarrow$ No を出力して，アルゴリズム全体を終了
     $c_{i-1} = d$ かつフラグ ON $\Rightarrow c_i := c_i + 1$
3. c_i を出力

変更部分

$\therefore$ このアルゴリズムは，出力を行うとき，
正しい c_i を出力する非決定性対数領域アルゴリズムである

NL = coNL：アルゴリズムの全体像

<u>アルゴリズム</u>

1. $c_0 = 1$
2. $i = 1, \ldots, n$ に対して順に，$\boxed{c_{i-1} \text{ から } c_i \text{ を計算}}$
3. $\boxed{c_n \text{ から } t \notin C_n \text{ を判定}}$

1. $d := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - s から u へ至る長さ i 以下の経路を guess して，検証
   - $u \in C_n \Rightarrow d := d + 1$
   - $u = t$ のときの結果を覚えておく
3. $d \neq c_n$ または $t \in C_i$ と判定 $\Rightarrow$ No を出力
4. そうでない $\Rightarrow$ Yes を出力

1. $c_i := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)

   2-1. $d := 0$

   2-2. すべての頂点 v に対して (ラベルの小さい順に)
   - $v \in C_{i-1} \Rightarrow d := d + 1$
   - $v \in C_{i-1}$ かつ「$u = v$ または $(v, u) \in A$」$\Rightarrow$ フラグ ON

   2-3. $c_{i-1} \neq d \Rightarrow$ No を出力して，アルゴリズム全体を終了
   $c_{i-1} = d$ かつフラグ ON $\Rightarrow c_i := c_i + 1$
3. c_i を出力

NL = coNL：アルゴリズムの全体像

アルゴリズム

1. $c_0 = 1$
2. $i = 1, \ldots, n$ に対して順に，$\boxed{c_{i-1} \text{ から } c_i \text{ を計算}}$
3. $\boxed{c_n \text{ から } t \notin C_n \text{ を判定}}$

---

1. $d := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - s から u へ至る長さ i 以下の経路を guess して，検証
   - $u \in C_n \Rightarrow d := d + 1$
   - $u = t$ のときの結果を覚えておく
3. $d \neq c_n$ または $t \in C_i$ と判定 ⇒ No を出力
4. そうでない ⇒ Yes を出力

---

1. $c_i := 0$
2. すべての頂点 u に対して (ラベルの小さい順に)
   - 2-1. $d := 0$
   - 2-2. すべての頂点 v に対して (ラベルの小さい順に)
     - $v \in C_{i-1} \Rightarrow d := d + 1$
     - $v \in C_{i-1}$ かつ「$u = v$ または $(v, u) \in A$」⇒ フラグ ON
   - 2-3. $c_{i-1} \neq d \Rightarrow$ No を出力して，アルゴリズム全体を終了
     $c_{i-1} = d$ かつフラグ ON $\Rightarrow c_i := c_i + 1$
3. c_i を出力

定理

$\overline{\text{STCON}} \in$ NL (すなわち，STCON $\in$ coNL)

Immerman–Szelepcsényi の定理 ('87)

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

証明の流れ

- (STCON が NL 完全であることを示す)
- $\overline{\text{STCON}} \in \mathsf{NL}$ を示す
  - そのために，帰納的数え上げを用いる

本日の内容

1. Immerman–Szelepcsényi の定理：証明の考え方
2. Immerman–Szelepcsényi の定理：アルゴリズム
3. **Immerman–Szelepcsényi の定理：補足**

I-Sz の定理：一般形

Immerman–Szelepcsényi の定理 ('87)

NL $=$ coNL

Immerman–Szelepcsényi の定理に水増し論法を適用すると，次が導ける

性質

$s\colon \mathbb{N} \to \mathbb{N}$ が領域構成可能であり，$s(n) \geq \lceil \log_2 n \rceil \Rightarrow$ 多テープ・チューリング機械において

$$\mathrm{NSPACE}(s(n)) = \mathrm{coNSPACE}(s(n))$$

領域構成可能関数の定義は，第 6 回を参照

NSPACE$(s(n))$ の定義は，第 8 回を参照

性質

$s \colon \mathbb{N} \to \mathbb{N}$ が領域構成可能であり，$s(n) \geq \lceil \log_2 n \rceil \Rightarrow$ 多テープ・チューリング機械において

$$\mathsf{NSPACE}(s(n)) = \mathsf{coNSPACE}(s(n))$$

例えば，次は正しい

- $\mathsf{NSPACE}(\log n) = \mathsf{coNSPACE}(\log n)$
- $\mathsf{NSPACE}(\log^2 n) = \mathsf{coNSPACE}(\log^2 n)$
- $\mathsf{NSPACE}(n) = \mathsf{coNSPACE}(n)$
- $\mathsf{NSPACE}(n^2) = \mathsf{coNSPACE}(n^2)$

I-Sz の定理：一般形 (続)

性質

$s\colon \mathbb{N} \to \mathbb{N}$ が領域構成可能であり，$s(n) \geq \lceil \log_2 n \rceil \Rightarrow$ 多テープ・チューリング機械において

$$\mathrm{NSPACE}(s(n)) = \mathrm{coNSPACE}(s(n))$$

例えば，次は正しい

- $\mathrm{NSPACE}(\log n) = \mathrm{coNSPACE}(\log n)$
- $\mathrm{NSPACE}(\log^2 n) = \mathrm{coNSPACE}(\log^2 n)$
- $\boxed{\mathrm{NSPACE}(n) = \mathrm{coNSPACE}(n)}$ 第 2 LBA 問題の肯定的解決
- $\mathrm{NSPACE}(n^2) = \mathrm{coNSPACE}(n^2)$

歴史：LBA 問題

LBA 問題 (黒田の問題) (Kuroda '64)

第 1 LBA 問題 以下の等式は成り立つか？ (未解決)

$$\mathrm{SPACE}(n) \stackrel{?}{=} \mathrm{NSPACE}(n)$$

第 2 LBA 問題 以下の等式は成り立つか？ (解決済)

$$\mathrm{NSPACE}(n) \stackrel{?}{=} \mathrm{coNSPACE}(n)$$

LBA $=$ Linear Bounded Automaton

(線形有界オートマトン，線形限定オートマトン，
線形拘束オートマトン，線形境界つきオートマトン)

領域計算量 $O(n)$ のチューリング機械の別名

歴史：LBA 問題

LBA 問題 (黒田の問題) (Kuroda ’64)

第 1 LBA 問題 以下の等式は成り立つか？ (未解決)

$$\mathsf{SPACE}(n) \stackrel{?}{=} \mathsf{NSPACE}(n)$$

第 2 LBA 問題 以下の等式は成り立つか？ (解決済)

$$\mathsf{NSPACE}(n) \stackrel{?}{=} \mathsf{coNSPACE}(n)$$

LBA = Linear Bounded Automaton
(線形有界オートマトン，線形限定オートマトン，
線形拘束オートマトン，線形境界つきオートマトン)
領域計算量 $O(n)$ のチューリング機械の別名

Immerman–Szelepcsényi の定理の一般形
⇒ 第 2 LBA 問題の「=」は成立

歴史：LBA 問題 (続)

LBA 問題 (黒田の問題) (Kuroda '64)

第 1 LBA 問題 以下の等式は成り立つか？ (未解決)

$$\mathsf{SPACE}(n) \stackrel{?}{=} \mathsf{NSPACE}(n)$$

第 2 LBA 問題 以下の等式は成り立つか？ (解決済)

$$\mathsf{NSPACE}(n) \stackrel{?}{=} \mathsf{coNSPACE}(n)$$

すぐにわかること

- $\mathsf{SPACE}(n) = \mathsf{NSPACE}(n) \Rightarrow \mathsf{NSPACE}(n) = \mathsf{coNSPACE}(n)$
  第 1 LBA 問題の肯定的解決　第 2 LBA 問題の肯定的解決
- つまり
  $\mathsf{NSPACE}(n) \neq \mathsf{coNSPACE}(n) \Rightarrow \mathsf{SPACE}(n) \neq \mathsf{NSPACE}(n)$
  第 2 LBA 問題の否定的解決　第 1 LBA 問題の否定的解決

形式言語とチョムスキー階層

言語	文法	機械
正規言語	正規文法	有限状態オートマトン
⇓		
文脈自由言語	文脈自由文法	プッシュダウン・オートマトン
⇓		
文脈依存言語	文脈依存文法	LBA
⇓		
帰納的可算言語	句構造文法	チューリング機械
	チョムスキー階層 (Chomsky hierarchy)	

形式言語とチョムスキー階層

言語	文法	機械
正規言語	正規文法	有限状態オートマトン 非決定性 = 決定性
$\Downarrow$		
文脈自由言語	文脈自由文法	プッシュダウン・オートマトン 非決定性 ($\neq$ 決定性)
$\Downarrow$		
文脈依存言語	文脈依存文法	LBA 非決定性 ($\stackrel{?}{=}$ 決定性)
$\Downarrow$		
帰納的可算言語	句構造文法	チューリング機械 非決定性 = 決定性

形式言語とチョムスキー階層

言語	文法	機械
正規言語	正規文法	有限状態オートマトン 非決定性 = 決定性
$\Downarrow$		
文脈自由言語	文脈自由文法	プッシュダウン・オートマトン 非決定性 ($\neq$ 決定性)
$\Downarrow$		
文脈依存言語	文脈依存文法	LBA　　第 1 LBA 問題 $\boxed{\text{非決定性} \ (\stackrel{?}{=} \text{決定性})}$
$\Downarrow$		
帰納的可算言語	句構造文法	チューリング機械 非決定性 = 決定性

形式言語とチョムスキー階層

言語 補集合で	文法	機械
正規言語	正規文法	有限状態オートマトン 非決定性 $=$ 決定性
$\Downarrow$ 閉じている		
文脈自由言語	文脈自由文法	プッシュダウン・オートマトン 非決定性 ($\neq$ 決定性)
$\Downarrow$ 閉じていない		
文脈依存言語	文脈依存文法	第 1 LBA 問題 LBA $\boxed{\text{非決定性 } (\stackrel{?}{=} \text{ 決定性})}$
$\Downarrow$ $\boxed{\text{閉じている}}$	第 2 LBA 問題	
帰納的可算言語	句構造文法	チューリング機械 非決定性 $=$ 決定性
閉じている		

黒田 成幸

くろだ・しげゆき (1934–2009)

- 言語学者
- UCSD 名誉教授，東北大学名誉教授
- 東大理学部数学科 →
  東大文学部 (学士入学)
- MIT 博士課程の指導教員は
  チョムスキー

https://dss-sites.ucsd.edu/kuroda/

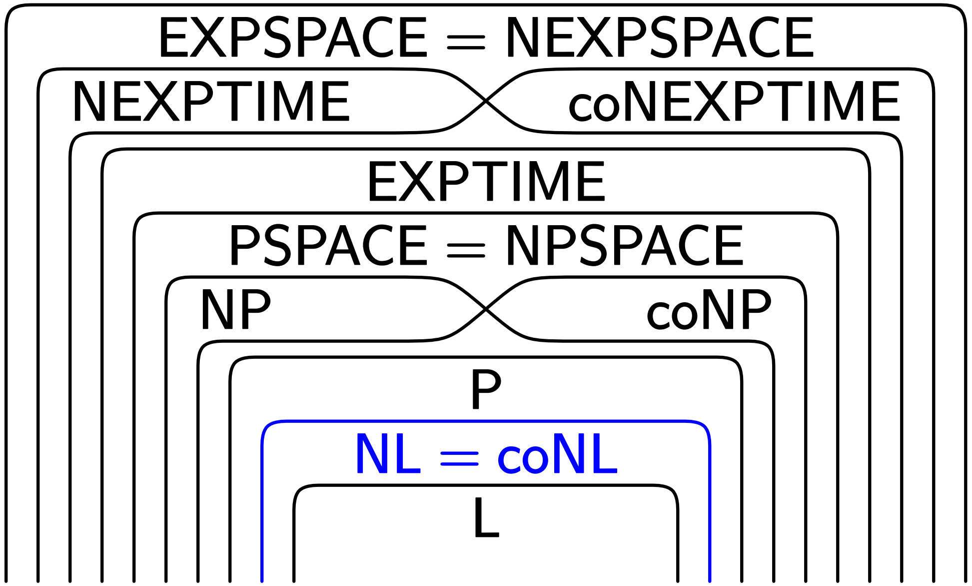
本日のまとめ

目標

次の「Immerman–Szelepcsényi の定理」を証明する

- NL $=$ coNL

証明手法：帰納的数え上げ (inductive counting)



次回の予告

内容

- **交代性計算** という計算モデルを導入する
- AP = PSPACE を証明する
- PSPACE 完全問題として別のものを紹介する
  - QBF (限定論理式問題，量化論理式問題)
- (EXPTIME などのクラスの完全問題も紹介する)

クイズ

Q

1.

2.

3.

4.