理論計算機科学特論 (2026 年前学期)

計算複雑性の基礎

第 4 回

領域計算量：L，NL，PSPACE

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2026 年 4 月 28 日

最終更新：2026 年 4 月 27 日　12:39

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)

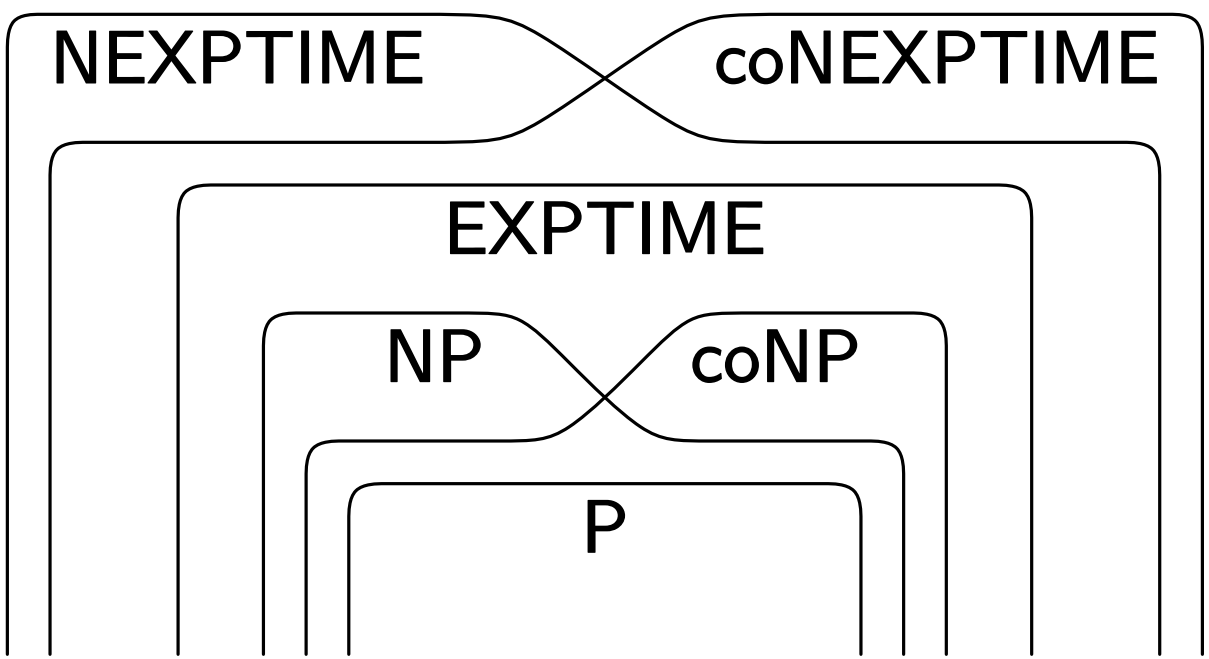
スケジュール (後半)

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

* 休み (授業のない日) (7/28)

復習：時間計算量

	決定性	非決定性	
多項式時間	P	NP,	coNP
指数時間	EXPTIME	NEXPTIME,	coNEXPTIME



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 ⇒ 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：非決定性と証拠

非決定性に対する別の解釈

Yes インスタンスに対しては，guess によって，
証拠 (certificate) を与える

$X = \{2, 4, 6, 9\}$ のとき

```
a = guess(X)    a = 6
b = guess(X)    b = 4
if a + b == 10:
  return "Yes"
else:
  return "No"
end
```

X が Yes インスタンス
であるための証拠

証拠を与えたあとは
決定性 (非決定性がない) で
証拠の **検証** をする
(verification)

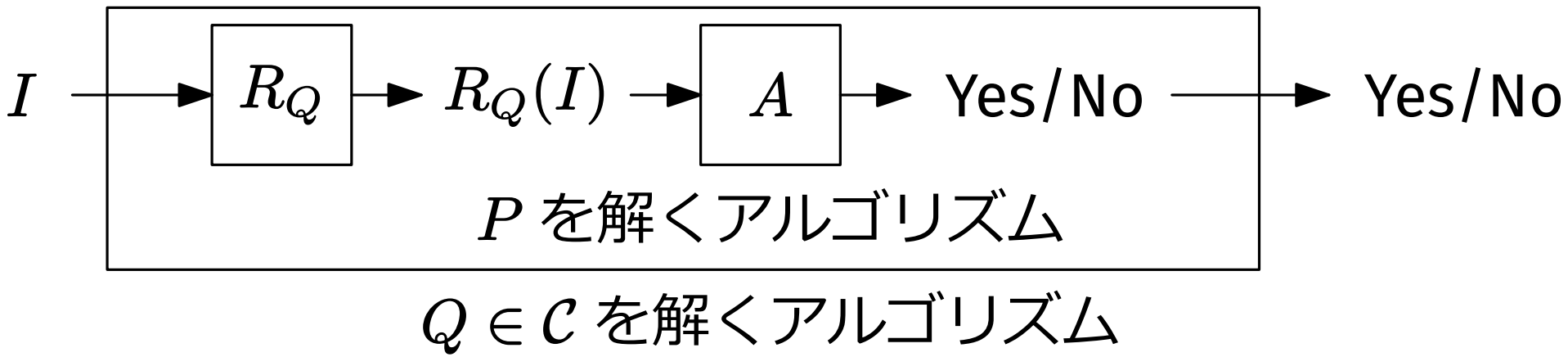
復習：困難性

クラス $\mathcal{C} \in \{\mathsf{NP}, \mathsf{coNP}, \mathsf{EXPTIME}, \mathsf{NEXPTIME}, \mathsf{coNEXPTIME}\}$

定義：困難性 (hardness)

問題 P が $\mathcal{C}$ **困難** ($\mathcal{C}$-hard) であるとは
$\mathcal{C}$ の任意の問題から P への多項式時間帰着が存在すること

直感：P は $\mathcal{C}$ のどの問題よりも難しいか，同程度に簡単



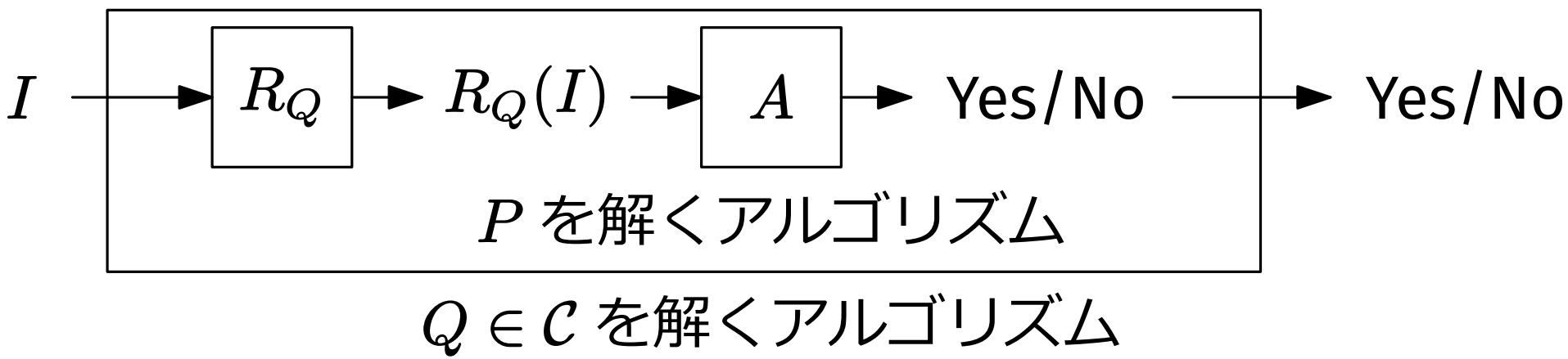
復習：困難性

クラス $\mathcal{C} \in \{\mathrm{NP}, \mathrm{coNP}, \mathrm{EXPTIME}, \mathrm{NEXPTIME}, \mathrm{coNEXPTIME}\}$

定義：困難性 (hardness)

問題 P が $\mathcal{C}$ **困難** ($\mathcal{C}$-hard) であるとは
$\mathcal{C}$ の任意の問題から P への多項式時間帰着が存在すること

直感：P は $\mathcal{C}$ のどの問題よりも難しいか，同程度に簡単



$\mathcal{C}$ には具体的なクラスを入れて，次の言い方をよく行う

- NP 困難，coNP 困難，EXPTIME 困難，NEXPTIME 困難，coNEXPTIME 困難

復習：完全性

クラス $\mathcal{C} \in \{\mathrm{NP}, \mathrm{coNP}, \mathrm{EXPTIME}, \mathrm{NEXPTIME}, \mathrm{coNEXPTIME}\}$

定義：完全性 (completeness)

問題 P が $\mathcal{C}$ **完全** ($\mathcal{C}$-complete) であるとは
P が $\mathcal{C}$ 困難であり，かつ，$P \in \mathcal{C}$ であること

直感：P は $\mathcal{C}$ の中でもっとも難しい問題 (の 1 つ)
($\mathcal{C}$ の中には，P より難しい問題がない)

$\mathcal{C}$ には具体的なクラスを入れて，次の言い方をよく行う

- NP 完全，coNP 完全，EXPTIME 完全，NEXPTIME 完全，coNEXPTIME 完全

復習：完全性

クラス $\mathcal{C} \in \{\mathrm{NP}, \mathrm{coNP}, \mathrm{EXPTIME}, \mathrm{NEXPTIME}, \mathrm{coNEXPTIME}\}$

定義：完全性 (completeness)

問題 P が $\mathcal{C}$ **完全** ($\mathcal{C}$-complete) であるとは
P が $\mathcal{C}$ 困難であり，かつ，$P \in \mathcal{C}$ であること

直感：P は $\mathcal{C}$ の中でもっとも難しい問題 (の 1 つ)
($\mathcal{C}$ の中には，P より難しい問題がない)

注意：言葉遣い

「○○困難/完全」はクラスを表す用語ではない
問題の性質を表す用語である

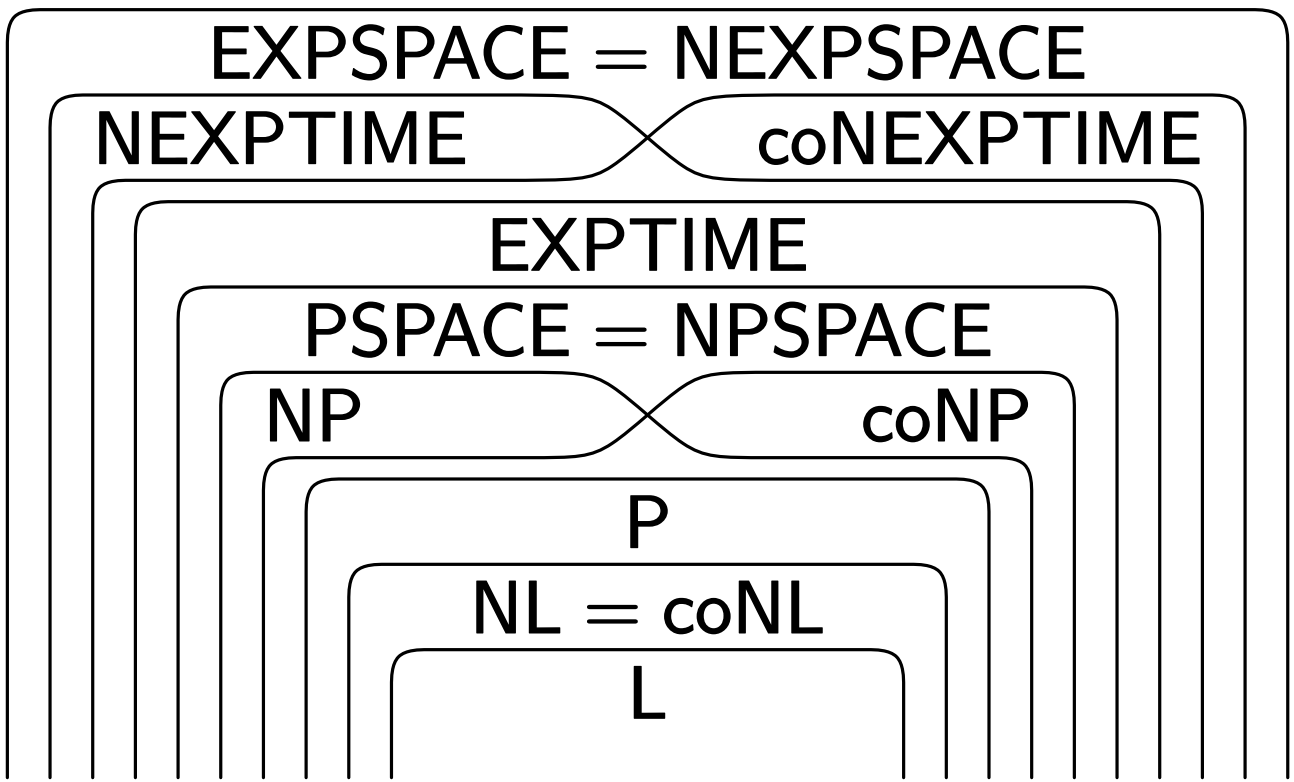
$\mathcal{C}$ には具体的なクラスを入れて，次の言い方をよく行う

- NP 完全，coNP 完全，EXPTIME 完全，NEXPTIME 完全，coNEXPTIME 完全

本日の目標

本日の目標

- 計算資源として **領域** (空間) を扱えるようになる
- 領域に関する計算複雑性クラスと帰着，完全性を扱えるようになる



本日の内容

1. **領域計算量と領域に関する複雑性**
2. 対数領域帰着と完全性

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)

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

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

非決定性アルゴリズムの領域計算量

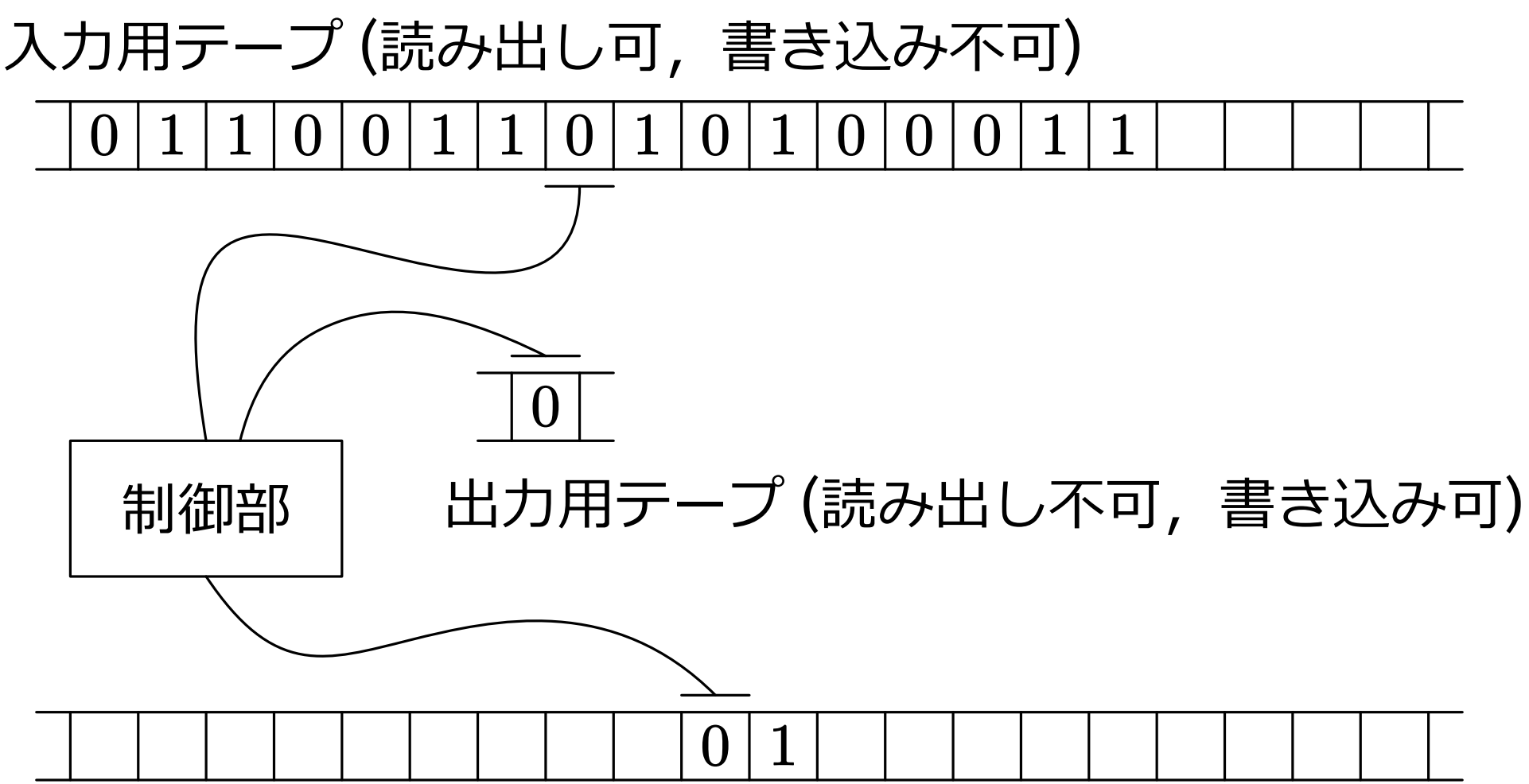
P は判定問題，A は P を解く非決定性アルゴリズム

定義：非決定性アルゴリズムの領域計算量

非決定性アルゴリズム A の **領域計算量** とは，
符号長 n 以下の入力 I と guess の出力に対して，
A が使う作業領域量の I と guess に関する最大値

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

(復習) チューリング機械：イメージ

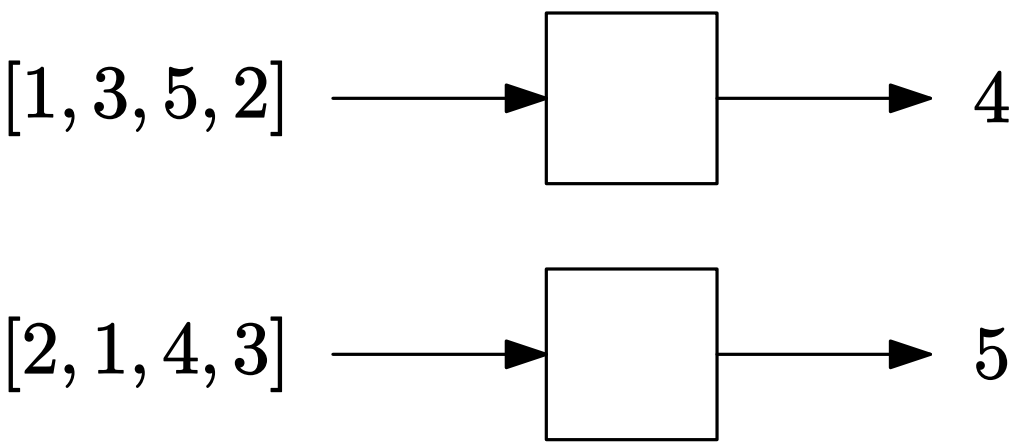


領域計算量：例

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

例：$m = 5$ のとき



注意：入力の符号長は？

領域計算量：例 (アルゴリズム 1)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

長さ m のビット配列 b を用意して，

i が入力にあったら，$b[i]=1$ とする

$[1,3,5,2]$

	1	2	3	4	5
b	0	0	0	0	0

領域計算量：例 (アルゴリズム 1)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

長さ m のビット配列 b を用意して，

i が入力にあったら，$b[i]=1$ とする

$[1,3,5,2]$

	1	2	3	4	5
b	1	0	0	0	0

領域計算量：例 (アルゴリズム 1)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

長さ m のビット配列 b を用意して，

i が入力にあったら，$b[i]=1$ とする

$[1,3,5,2]$

	1	2	3	4	5
b	1	0	1	0	0

領域計算量：例 (アルゴリズム 1)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

長さ m のビット配列 b を用意して，

i が入力にあったら，$b[i]=1$ とする

$[1,3,5,2]$

	1	2	3	4	5
b	1	0	1	0	1

領域計算量：例 (アルゴリズム 1)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

長さ m のビット配列 b を用意して，

i が入力にあったら，$b[i]=1$ とする

$[1,3,5,2]$

	1	2	3	4	5
b	1	1	1	0	1

領域計算量：例 (アルゴリズム 1)

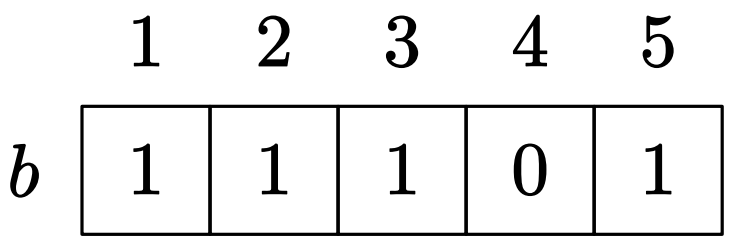
解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

長さ m のビット配列 b を用意して，

i が入力にあったら，$b[i]=1$ とする

$[1,3,5,2]$



$\rightsquigarrow$ 正答は 4

領域計算量：例 (アルゴリズム 1)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

長さ m のビット配列 b を用意して，

i が入力にあったら，$b[i]=1$ とする

$[1,3,5,2]$

	1	2	3	4	5
b	1	1	1	0	1

$\rightsquigarrow$ 正答は 4

アルゴリズム 1 の領域計算量 $=O(m)$

領域計算量：例 (アルゴリズム 2)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

$s = 1 + 2 + \cdots + m = \frac{1}{2}m(m+1)$ を計算して，

i が入力にあったら，s から i を引く

$[1, 3, 5, 2]$ s: 15

領域計算量：例 (アルゴリズム 2)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

$s = 1 + 2 + \cdots + m = \frac{1}{2}m(m+1)$ を計算して，

i が入力にあったら，s から i を引く

$[1, 3, 5, 2]$ $s\colon 15 \to 14$

領域計算量：例 (アルゴリズム 2)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

$s = 1 + 2 + \cdots + m = \frac{1}{2}m(m+1)$ を計算して，

i が入力にあったら，s から i を引く

$[1, 3, 5, 2]$

s: $15 \to 14 \to 11$

領域計算量：例 (アルゴリズム 2)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

$s = 1 + 2 + \cdots + m = \frac{1}{2}m(m+1)$ を計算して，

i が入力にあったら，s から i を引く

$[1, 3, 5, 2]$

s: $15 \to 14 \to 11 \to 6$

領域計算量：例 (アルゴリズム 2)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

$s = 1 + 2 + \cdots + m = \frac{1}{2}m(m+1)$ を計算して，

i が入力にあったら，s から i を引く

$[1, 3, 5, 2]$

s: $15 \to 14 \to 11 \to 6 \to 4$

$\rightsquigarrow$ 正答は 4

領域計算量：例 (アルゴリズム 2)

解きたい問題

入力： 1 から m までの整数の中の異なる $m-1$ 個
出力： 1 から m までの整数の中で入力に含まれないもの

$s = 1 + 2 + \cdots + m = \frac{1}{2}m(m+1)$ を計算して，
i が入力にあったら，s から i を引く

$[1, 3, 5, 2]$ $\quad s\colon 15 \to 14 \to 11 \to 6 \to 4$

$\rightsquigarrow$ 正答は 4

アルゴリズム 2 の領域計算量 $= O(\log m)$

多項式領域と指数領域

P は判定問題，A は P を解くアルゴリズム

定義：多項式領域アルゴリズム

A が **多項式領域アルゴリズム** であるとは，
A の領域計算量が n の**多項式**で上から抑えられること

つまり，ある $k > 0$ が存在して，$s_A(n) = O(n^k)$

定義：指数領域アルゴリズム

A が **指数領域アルゴリズム** であるとは，
A の領域計算量が n の**指数関数**で上から抑えられること

つまり，ある $k > 0$ が存在して，$s_A(n) = O(2^{n^k})$

対数領域

P は判定問題，A は P を解くアルゴリズム

定義：対数領域アルゴリズム

A が **対数領域アルゴリズム** であるとは，
A の領域計算量が n の**対数関数**で上から抑えられること

つまり，$s_A(n) = O(\log n)$

直感：定数個の変数しか使わないアルゴリズム

注：$O(\log^2 n)$，$O(\log^3 n)$，$\ldots$ は対数領域では許されない

クラス PSPACE，EXPSPACE，L

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

定義：クラス EXPSPACE

クラス EXPSPACE とは，
指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 L

クラス L とは，
対数領域アルゴリズムで解ける判定問題全体のこと

L は「logarithm (対数)」の意味

非決定性領域複雑性クラス

	非決定性なし	非決定性あり
指数領域	EXPSPACE	NEXPSPACE
多項式領域	PSPACE	NPSPACE
対数領域	L	NL

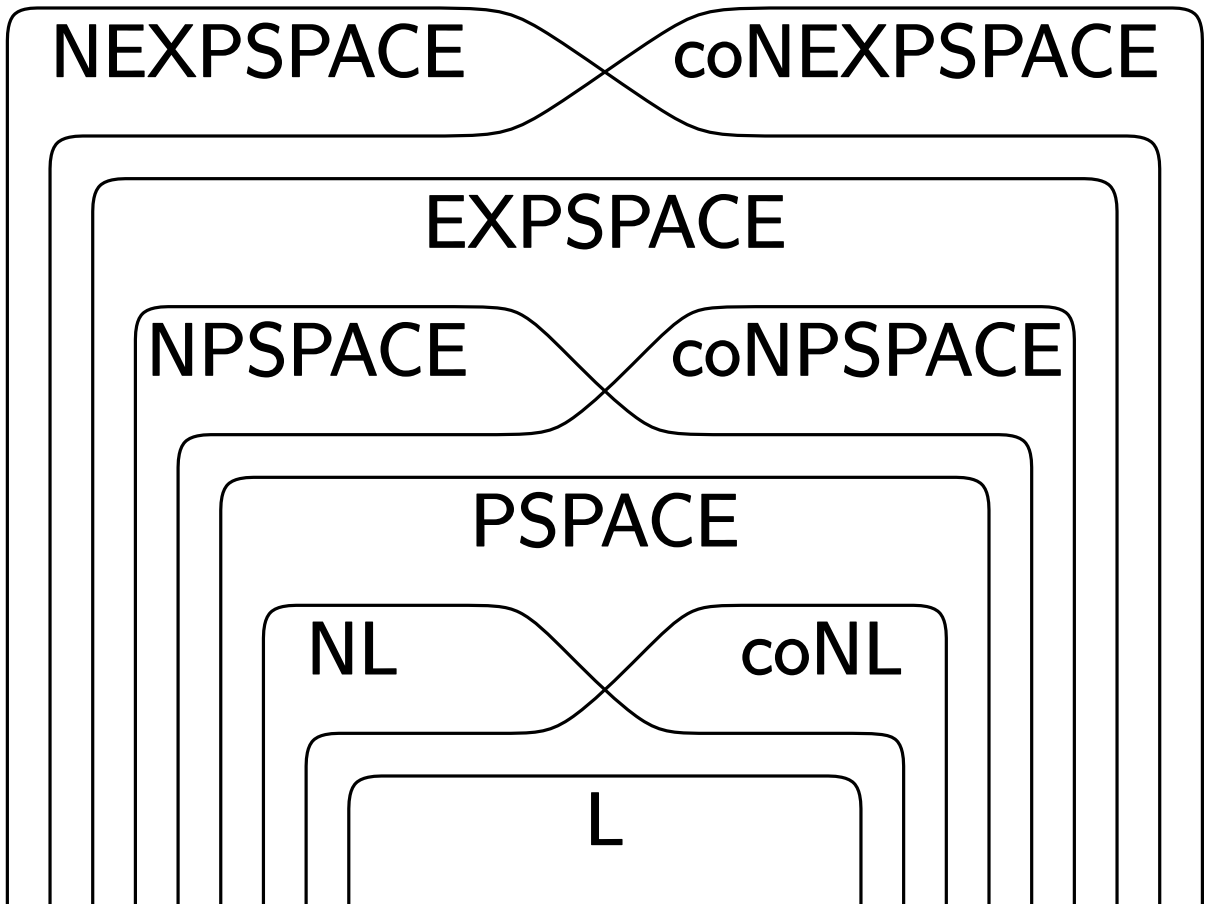
非決定性領域複雑性クラス

	非決定性なし	非決定性あり
指数領域	EXPSPACE	NEXPSPACE coNEXPSPACE
多項式領域	PSPACE	NPSPACE coNPSPACE
対数領域	L	NL coNL

領域複雑性クラスの間の関係 (1)

定義から次の包含関係が分かる

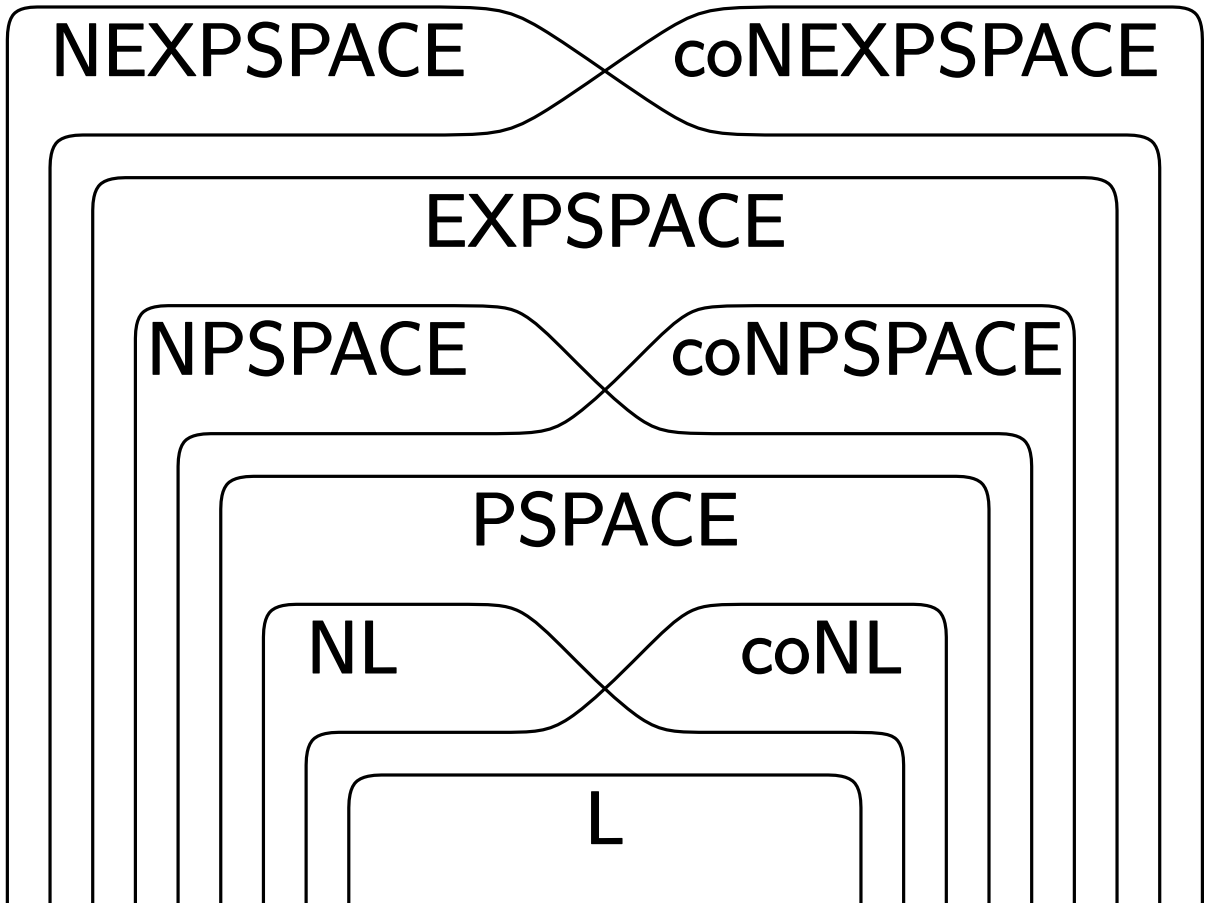
- $L \subseteq PSPACE \subseteq EXPSPACE$
- $NL \subseteq NPSPACE \subseteq NEXPSPACE$
- $coNL \subseteq coNPSPACE \subseteq coNEXPSPACE$



領域複雑性クラスの間の関係 (2)

定義から次の包含関係が分かる

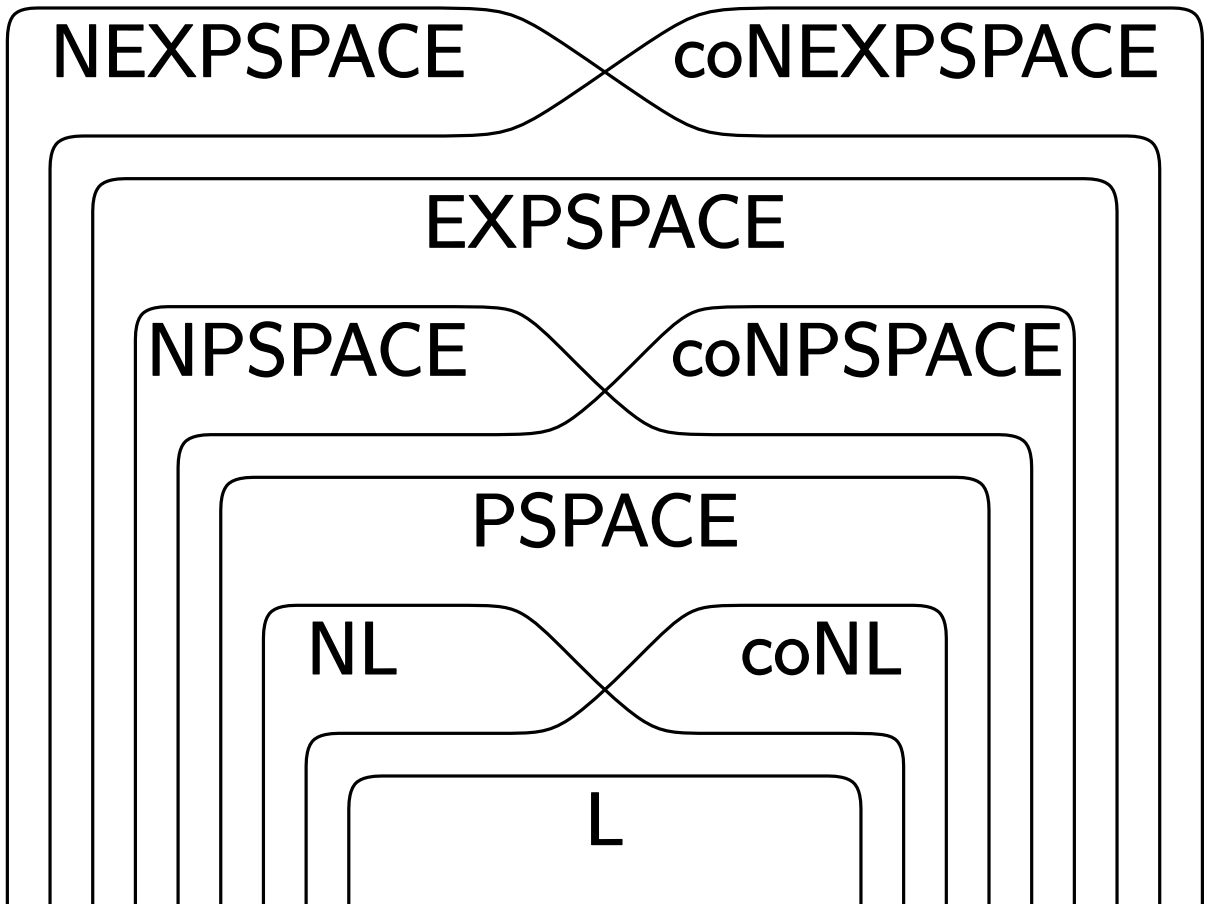
- EXPSPACE $\subseteq$ NEXPSPACE $\cap$ coNEXPSPACE
- PSPACE $\subseteq$ NPSPACE $\cap$ coNPSPACE
- L $\subseteq$ NL $\cap$ coNL



領域複雑性クラスの間の関係 (3)

実は次も正しい (次回以降の講義)

- NPSPACE ∪ coNPSPACE ⊆ EXPSPACE
- NL ∪ coNL ⊆ PSPACE



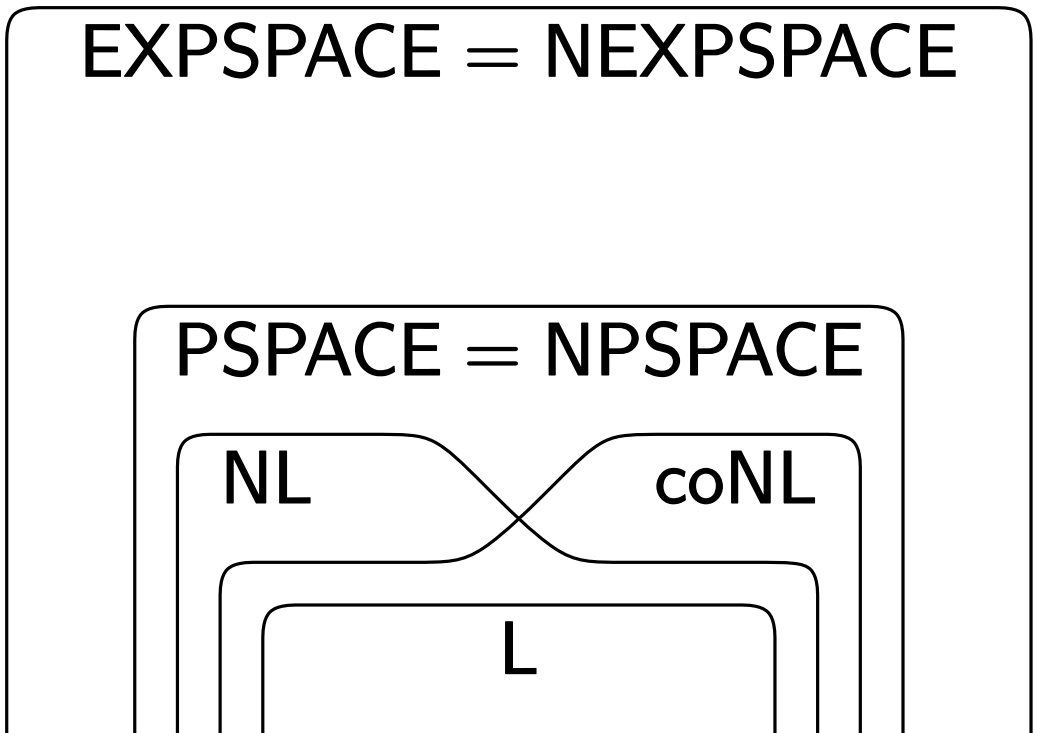
領域複雑性クラスの間の関係 (4)

実は次も正しい (次回以降の講義)

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

(Savitch '70)

注：$\mathrm{P} \stackrel{?}{=} \mathrm{NP}$，$\mathrm{EXPTIME} \stackrel{?}{=} \mathrm{NEXPTIME}$ は未解決



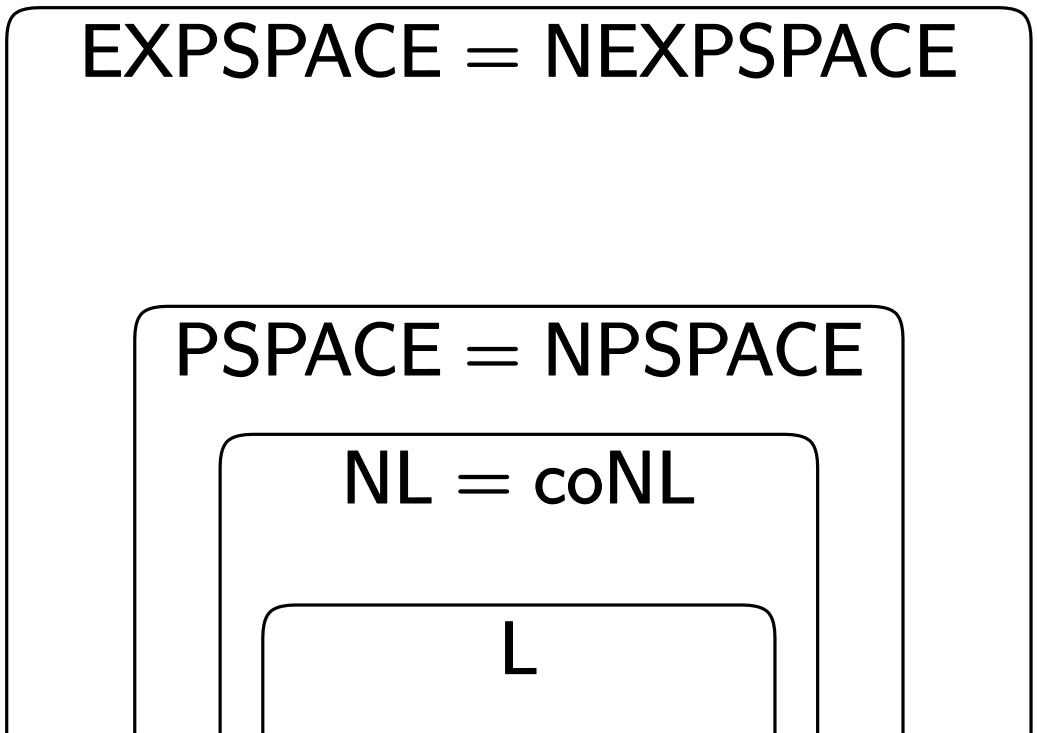
領域複雑性クラスの間の関係 (5)

実は次も正しい (次回以降の講義)

- NL = coNL

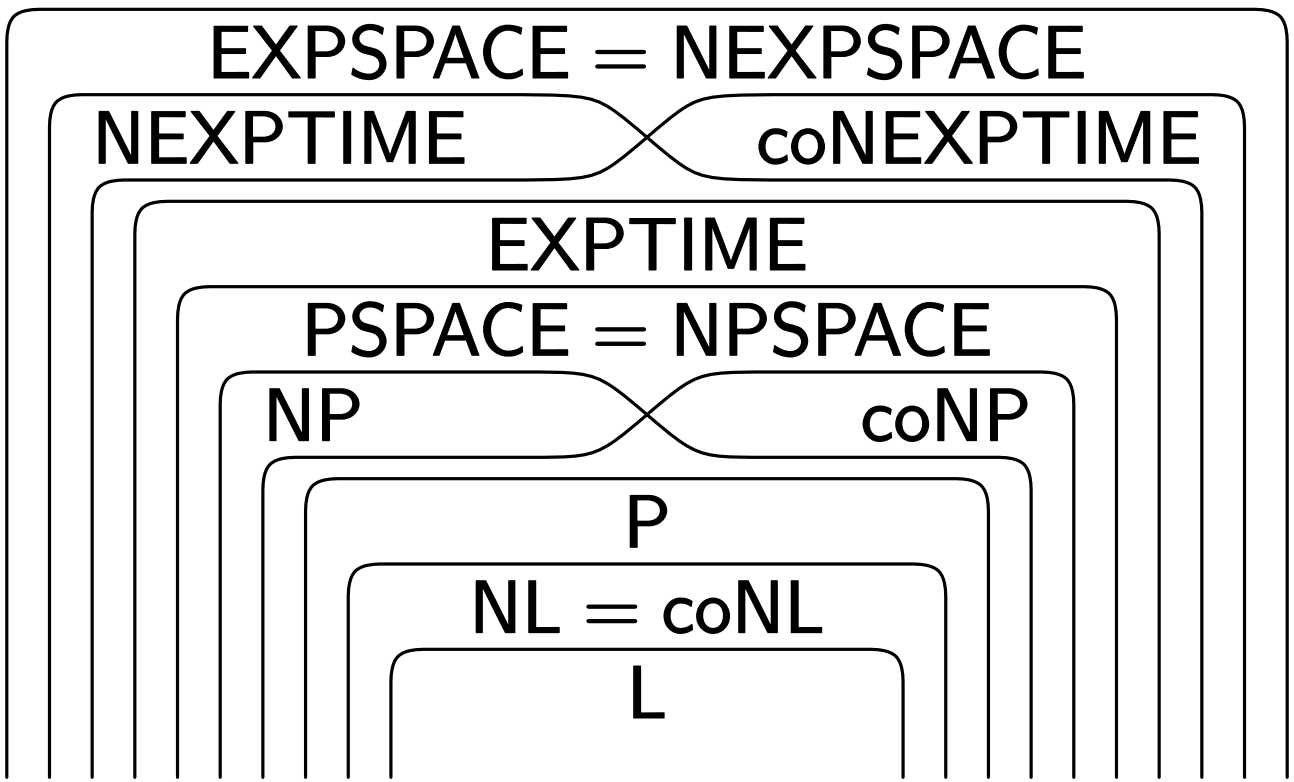
(Immerman '80, Szelepcsényi '80)

注：$L \stackrel{?}{=} NL$ は未解決



実は次も正しい (次回以降の講義)

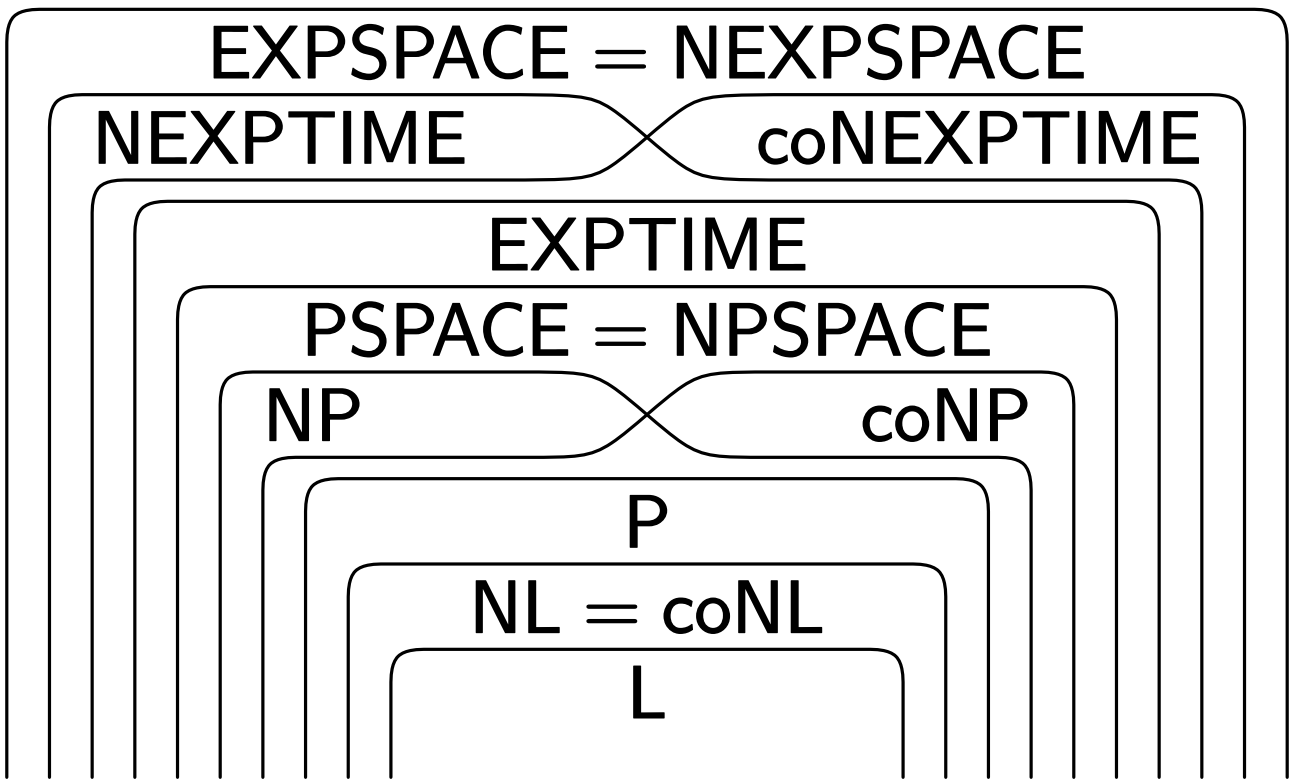
- $\mathsf{NL} \subseteq \mathsf{P}$
- $\mathsf{NP} \cup \mathsf{coNP} \subseteq \mathsf{PSPACE}$
- $\mathsf{NEXPTIME} \cup \mathsf{coNEXPTIME} \subseteq \mathsf{EXPSPACE}$



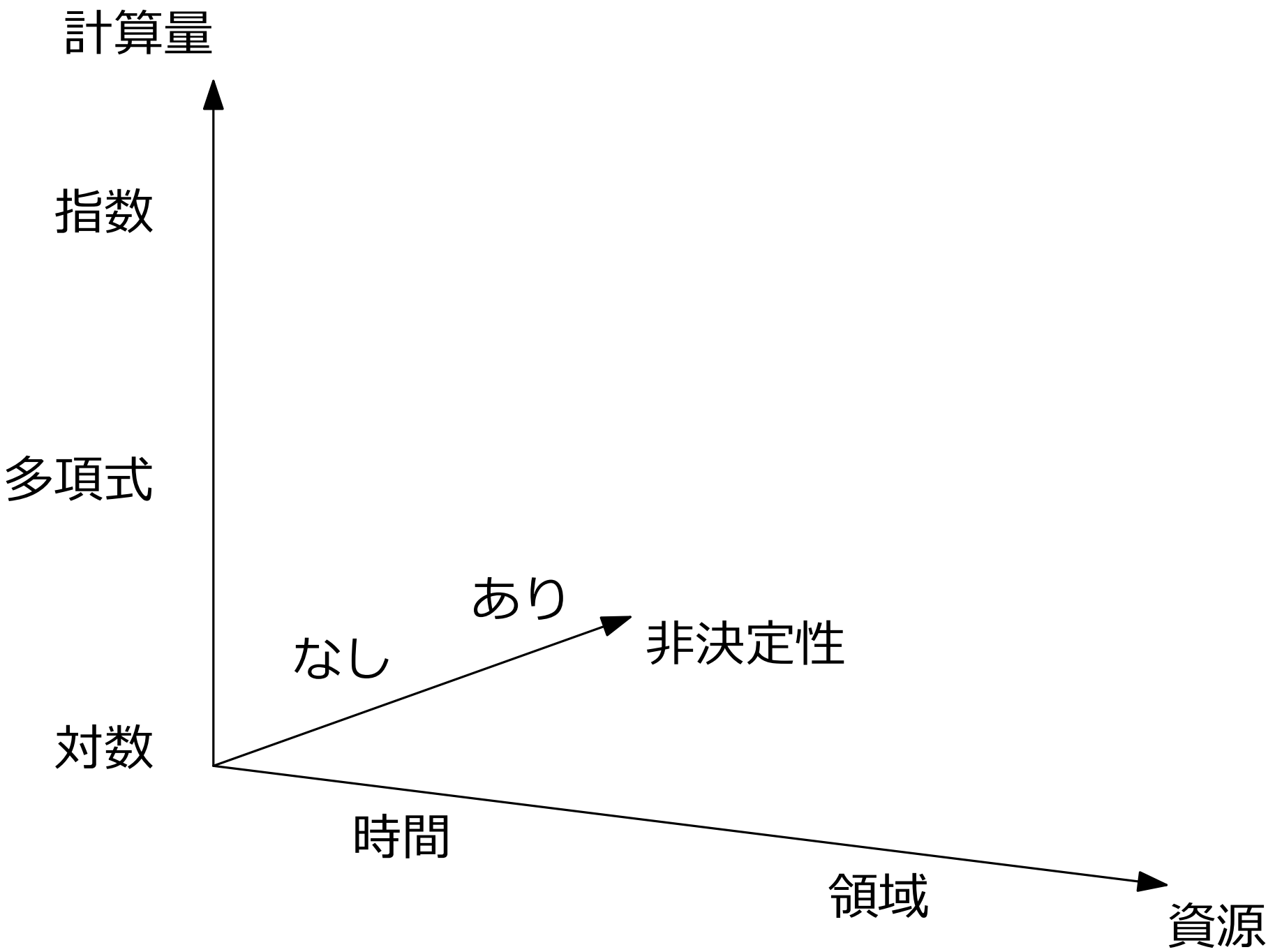
未解決問題

未解決問題

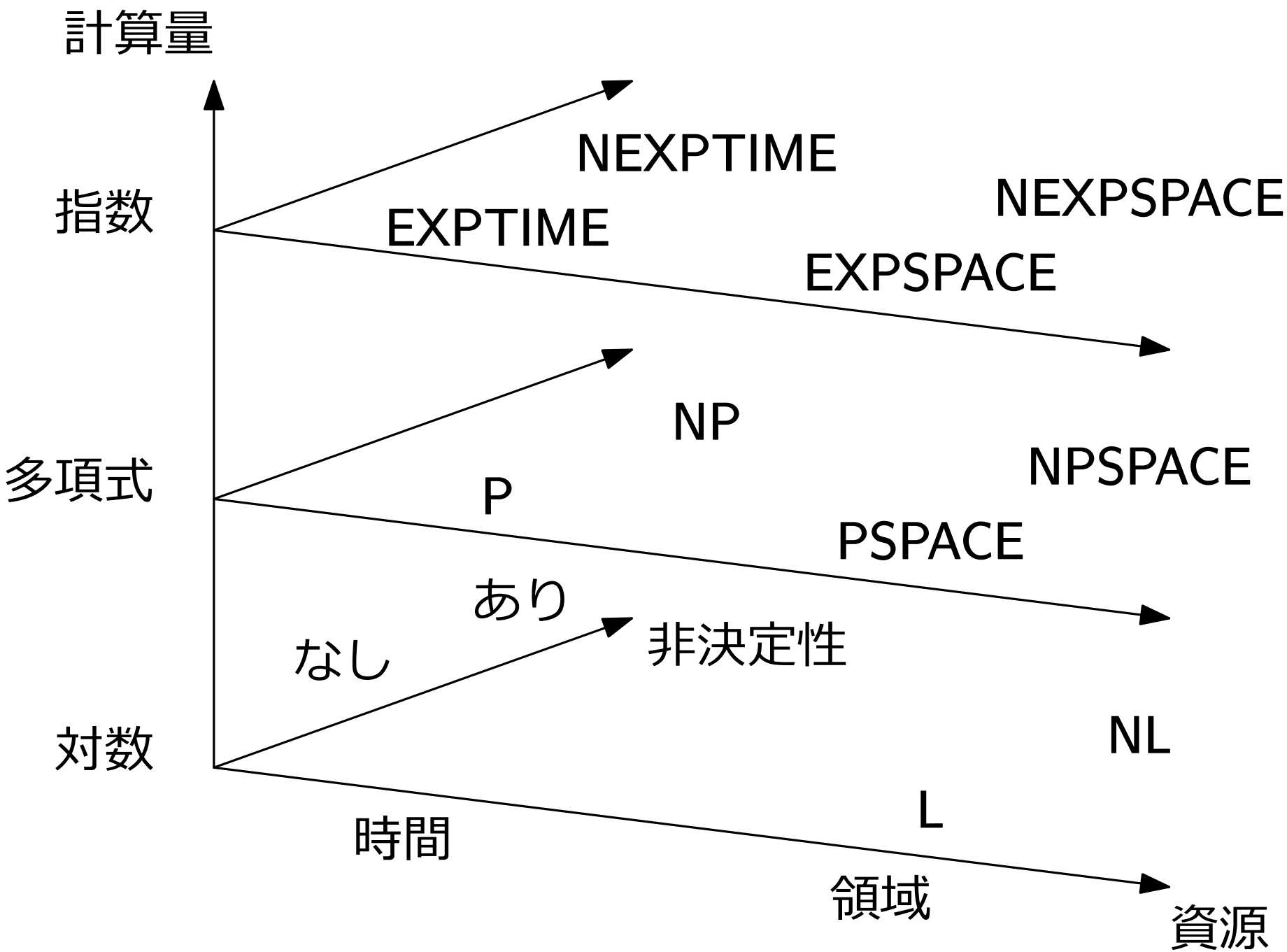
- $L \stackrel{?}{=} NL$
- $NP \stackrel{?}{=} PSPACE$
- $NEXPTIME \stackrel{?}{=} EXPSPACE$



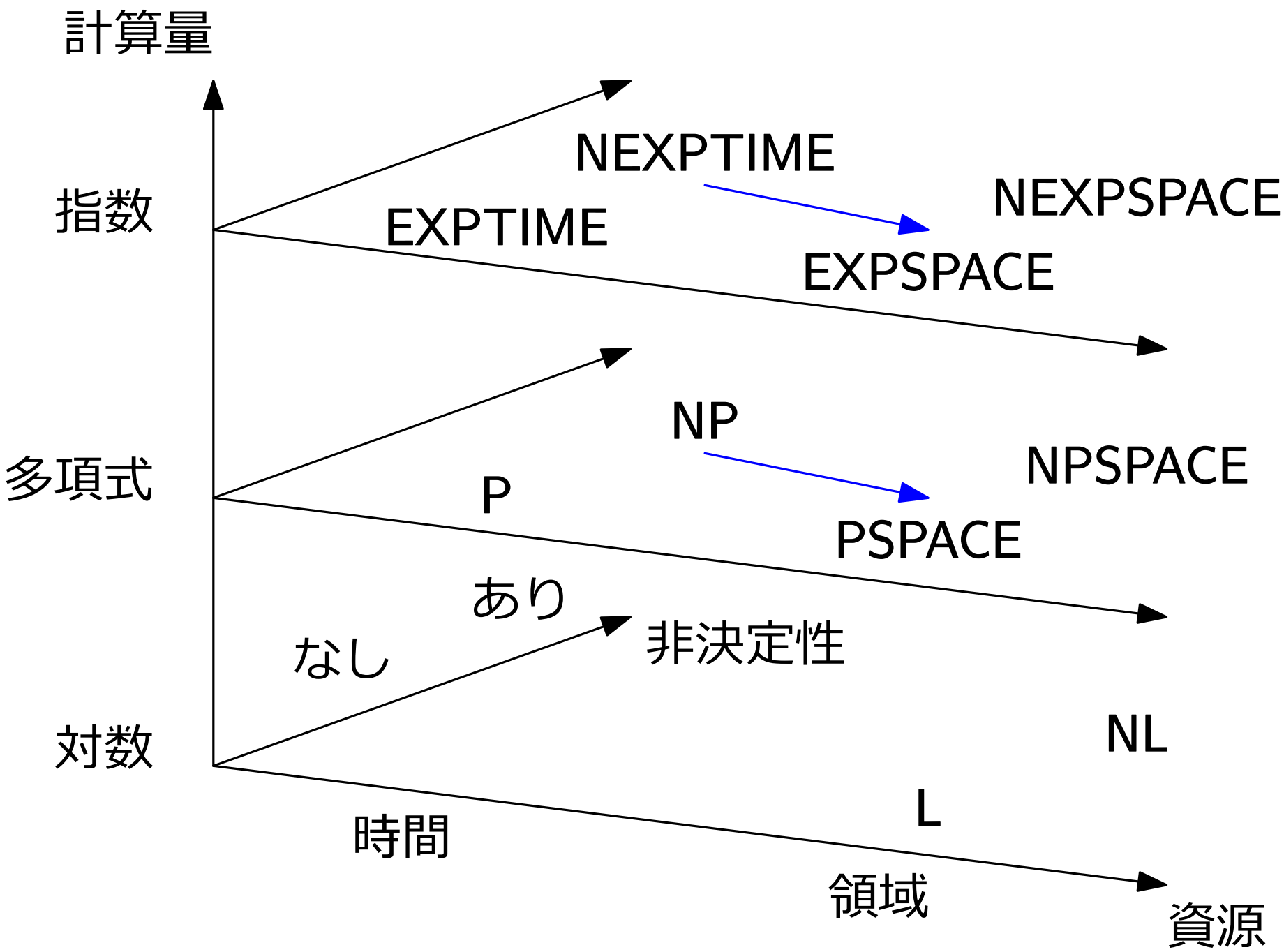
三次元的？ に書くと



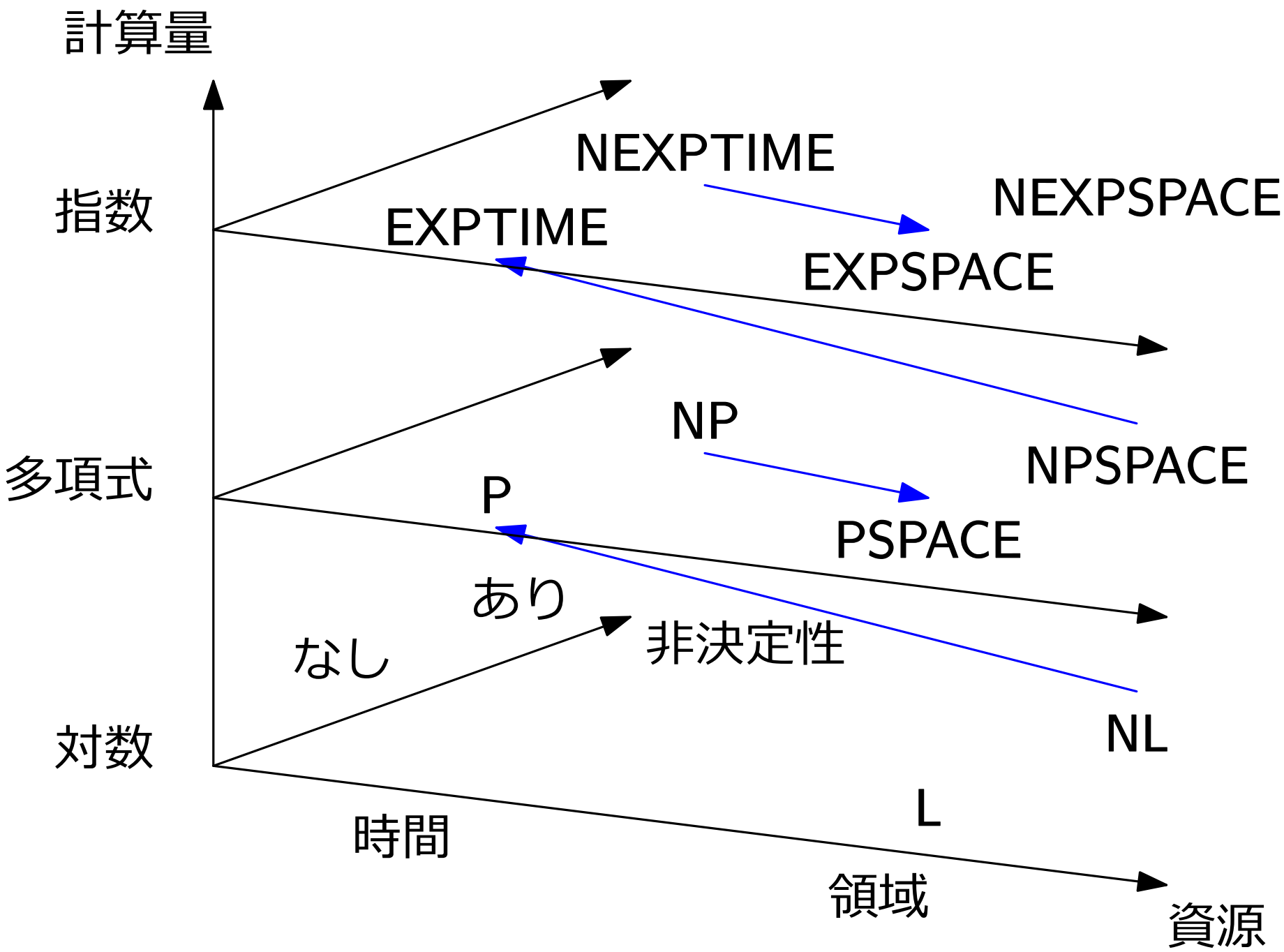
三次元的？ に書くと


計算量
指数
多項式
対数
時間
領域
資源
NEXPTIME
EXPTIME
NEXPSPACE
EXPSPACE
NP
P
NPSPACE
PSPACE
あり
なし
非決定性
NL
L

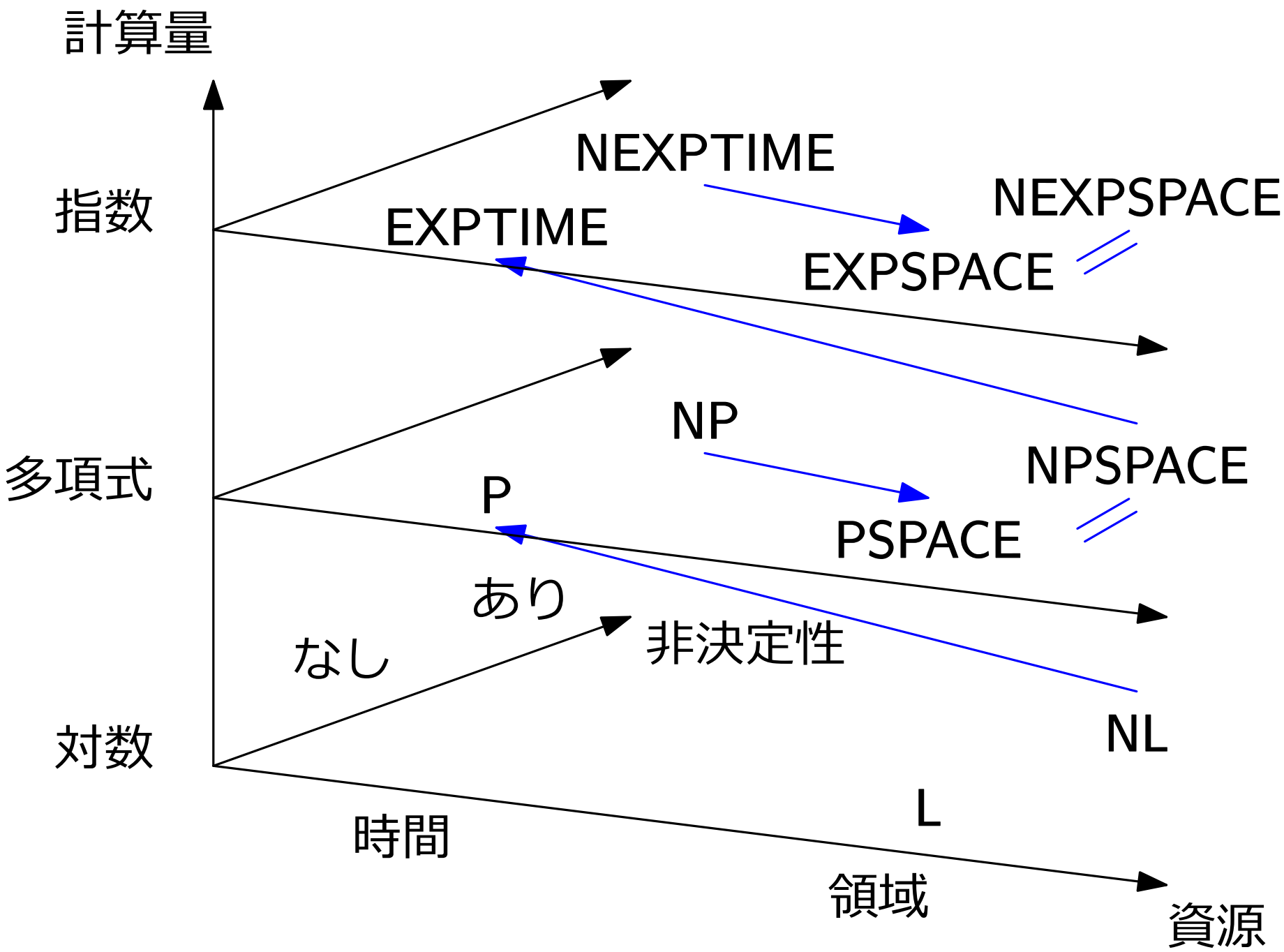
三次元的？ に書くと


計算量
指数
NEXPTIME
EXPTIME
NEXPSPACE
EXPSPACE
多項式
NP
P
NPSPACE
PSPACE
あり
なし
非決定性
対数
NL
L
時間
領域
資源

三次元的？ に書くと


計算量
指数
多項式
対数
NEXPTIME
EXPTIME
NEXPSPACE
EXPSPACE
NP
NPSPACE
P
PSPACE
あり
なし
非決定性
NL
L
時間
領域
資源

三次元的？ に書くと


計算量
指数
多項式
対数
時間
領域
資源
NEXPTIME
EXPTIME
NEXPSPACE
EXPSPACE
NP
P
NPSPACE
PSPACE
あり
なし
非決定性
NL
L

本日の内容

1. 領域計算量と領域に関する複雑性
2. **対数領域帰着と完全性**

完全性と困難性

クラス NP，coNP，EXPTIME と同様に，
クラス PSPACE，NPSPACE，coNPSPACE，EXPSPACE，
NEXPSPACE，coNEXPSPACE に対して，
困難問題や完全問題を定義できる

定義：PSPACE 困難性 (PSPACE-hardness)

問題 P が PSPACE **困難** であるとは
PSPACE の任意の問題から P への多項式時間帰着が存在すること

定義：PSPACE 完全性 (PSPACE-completeness)

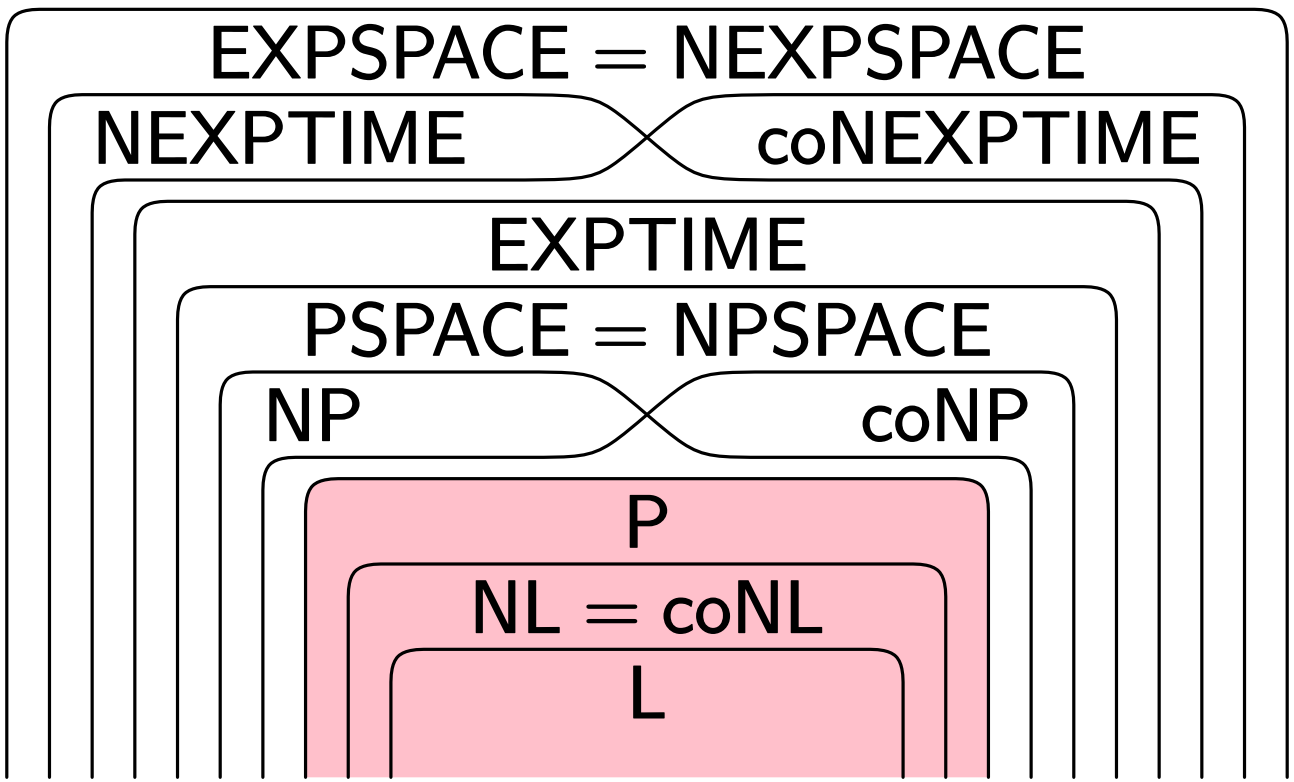
問題 P が PSPACE **完全** であるとは
P が PSPACE 困難であり，かつ，$P \in$ PSPACE であること

P に含まれるクラスにおける完全性

前回の反省

多項式時間帰着を使うと，P の問題を分類できない

- 理由：多項式時間帰着が強力すぎる
- 解決策：多項式時間帰着よりも弱い帰着を考える

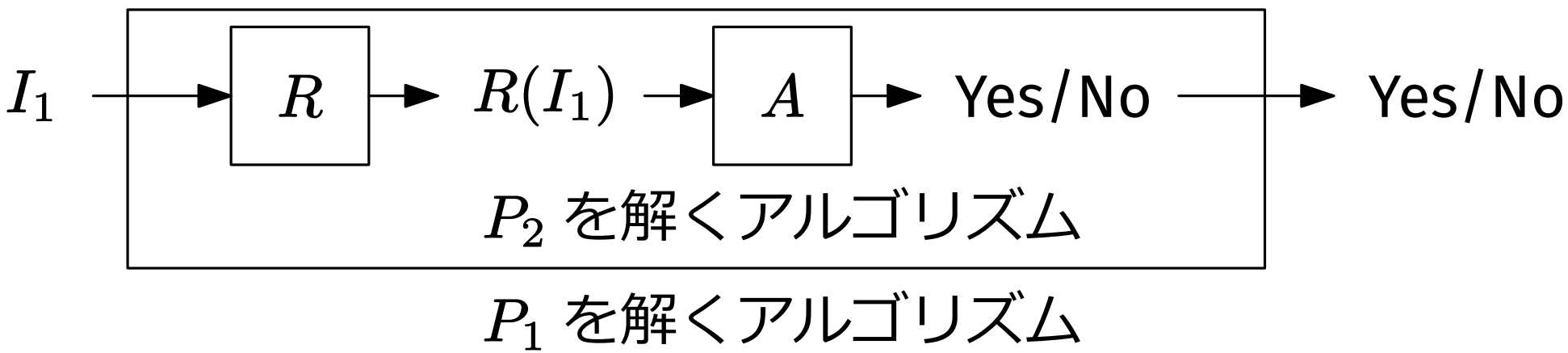


対数領域帰着

問題 P_1, P_2

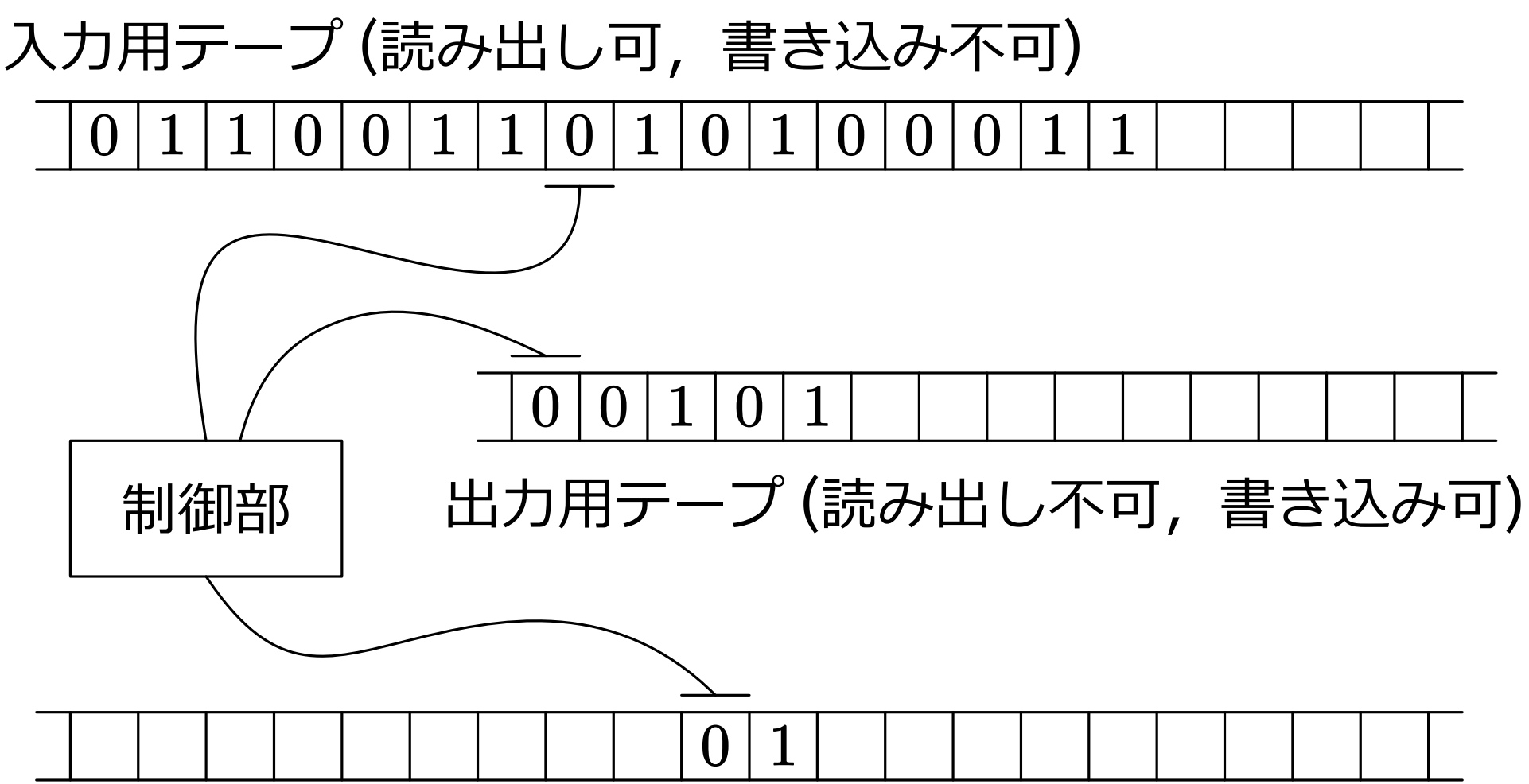
定義：対数領域帰着 (logspace reduction)

P_1 から P_2 への **対数領域帰着** とは,
P_1 から P_2 への帰着 R で,
R の領域計算量が対数であること



注：R が対数領域帰着 $\Rightarrow$ R が多項式時間帰着

注意：チューリング機械：イメージ



P 困難性と P 完全性

クラス P と NL における困難性と完全性は
対数領域帰着を使って定義する

定義：P 困難性 (P-hardness)

問題 P が P **困難** であるとは
P の任意の問題から P への対数領域帰着が存在すること

定義：P 完全性 (P-completeness)

問題 P が P **完全** であるとは
P が P 困難であり，かつ，$P \in \mathsf{P}$ であること

NL 困難性と NL 完全性

クラス P と NL における困難性と完全性は
対数領域帰着を使って定義する

定義：NL 困難性 (NL-hardness)

問題 P が NL **困難** であるとは
NL の任意の問題から P への対数領域帰着が存在すること

定義：NL 完全性 (NL-completeness)

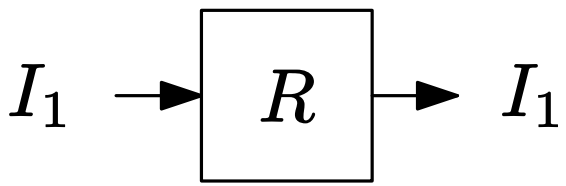
問題 P が NL **完全** であるとは
P が NL 困難であり，かつ，$P \in$ NL であること

対数領域帰着の性質：反射性と推移性

問題 P_1, P_2, P_3

性質：対数領域帰着の反射性と推移性

1. P_1 から P_1 への対数領域帰着が存在
2. P_1 から P_2 への対数領域帰着が存在，かつ，
   P_2 から P_3 への対数領域帰着が存在
   $\Rightarrow$ P_1 から P_3 への対数領域帰着が存在



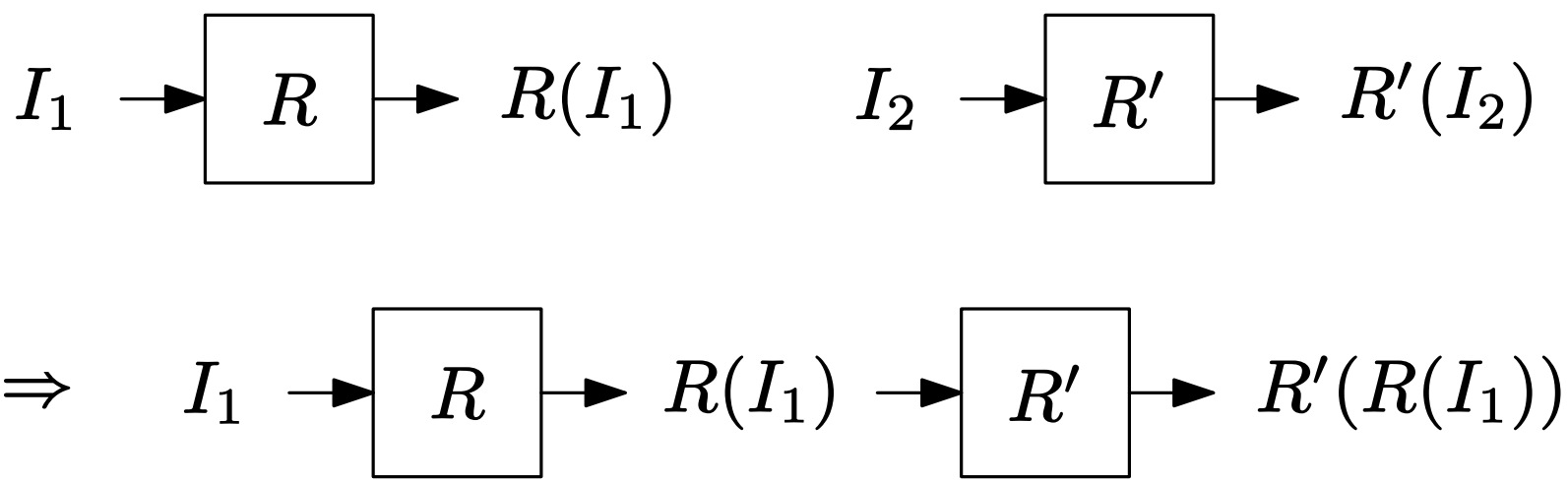

```
1: while (char := sys.stdin.read(1)):
2:    sys.stdout.write(char)
```

対数領域帰着の性質：反射性と推移性

問題 P_1, P_2, P_3

性質：対数領域帰着の反射性と推移性

1. P_1 から P_1 への対数領域帰着が存在
2. P_1 から P_2 への対数領域帰着が存在，かつ，
   P_2 から P_3 への対数領域帰着が存在
   $\Rightarrow$ P_1 から P_3 への対数領域帰着が存在

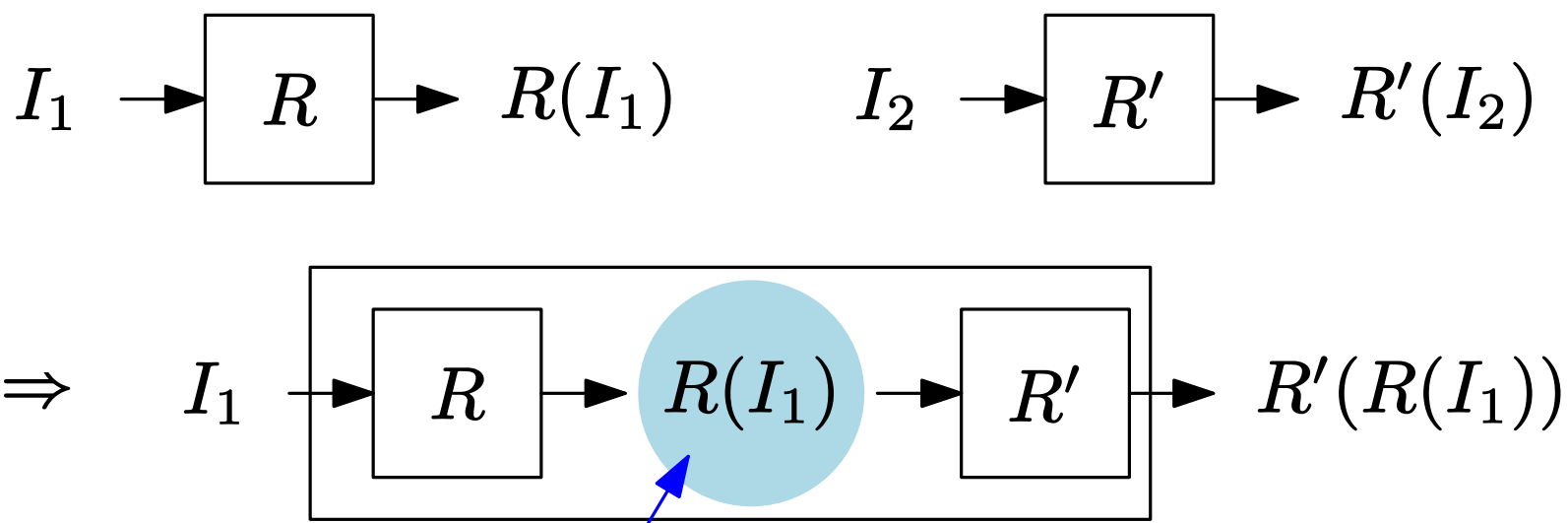


対数領域帰着の性質：反射性と推移性

問題 P_1, P_2, P_3

性質：対数領域帰着の反射性と推移性

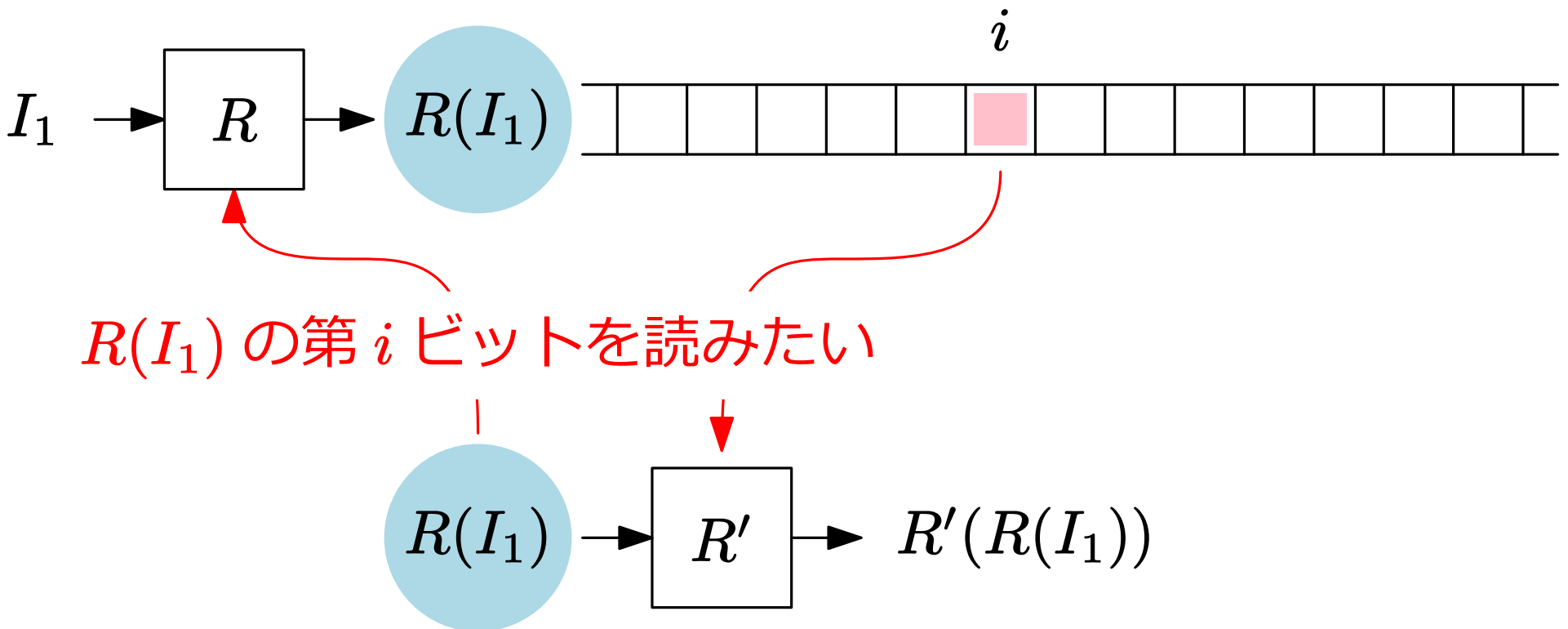
1. P_1 から P_1 への対数領域帰着が存在
2. P_1 から P_2 への対数領域帰着が存在，かつ，
   P_2 から P_3 への対数領域帰着が存在
   $\Rightarrow$ P_1 から P_3 への対数領域帰着が存在



推移性の証明 (1)

工夫

R' は $R(I_1)$ の必要なビットを，その都度 R からもらう



注：i を表す変数 は 作業領域に含まれる

推移性の証明 (2)

工夫の正当化：次を証明すれば十分

ある定数 k が存在して，$|R(I_1)| = O(|I_1|^k)$

$\therefore i$ を表す変数の符号長 $= O(\log |R(I_1)|) = O(\log |I_1|)$

推移性の証明 (2)

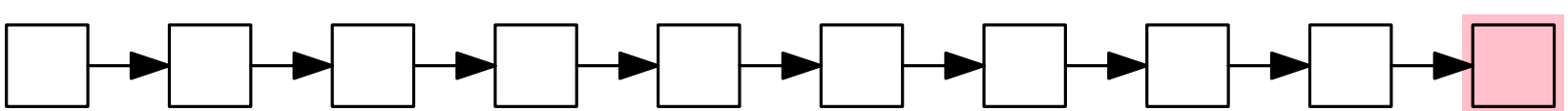
工夫の正当化：次を証明すれば十分

ある定数 k が存在して，$|R(I_1)| = O(|I_1|^k)$

$\therefore i$ を表す変数の符号長 $= O(\log |R(I_1)|) = O(\log |I_1|)$

証明：R が対数領域アルゴリズムであることを思い出す

- 時点状況の総数 $= O($作業領域の取りうる状態の総数$)$
  $= 2^{O(\log |I_1|)} = O(|I_1|$の多項式$)$
- 各時点状況では $R(I_1)$ の高々 1 ビットが書かれるので，
  $|R(I_1)| = O($時点状況の総数$) = O(|I_1|$の多項式$)$ □



困難性・完全性の性質 (1)

NP 困難性，NP 完全性と同様に，次の性質が成り立つ

性質：PSPACE 完全問題が非決定性多項式時間で解けたら

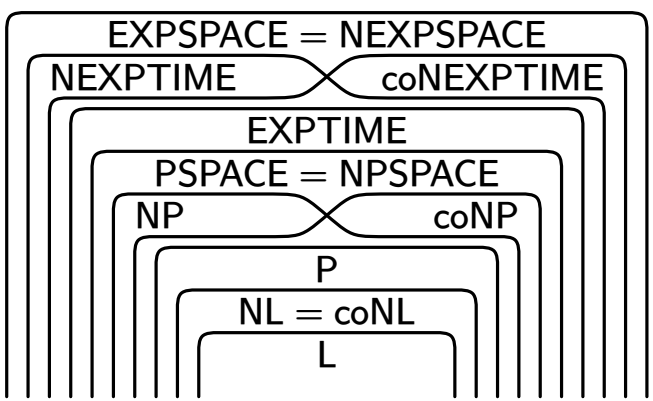
P が PSPACE 完全，かつ，$P \in \mathrm{NP} \Rightarrow \mathrm{NP} = \mathrm{PSPACE}$

性質：P 完全問題が非決定性対数領域で解けたら

P が P 完全，かつ，$P \in \mathrm{NL} \Rightarrow \mathrm{NL} = \mathrm{P}$

性質：NL 完全問題が対数領域で解けたら

P が NL 完全，かつ，$P \in \mathrm{L} \Rightarrow \mathrm{L} = \mathrm{NL}$



困難性・完全性の性質 (2)

NP 困難性，NP 完全性と同様に，次の性質が成り立つ

性質：PSPACE 困難問題から帰着ができたら

PSPACE 困難問題 P から Q への多項式時間帰着が存在
$\Rightarrow Q$ は PSPACE 困難

性質：P 困難問題から帰着ができたら

P 困難問題 P から Q への対数領域帰着が存在
$\Rightarrow Q$ は P 困難

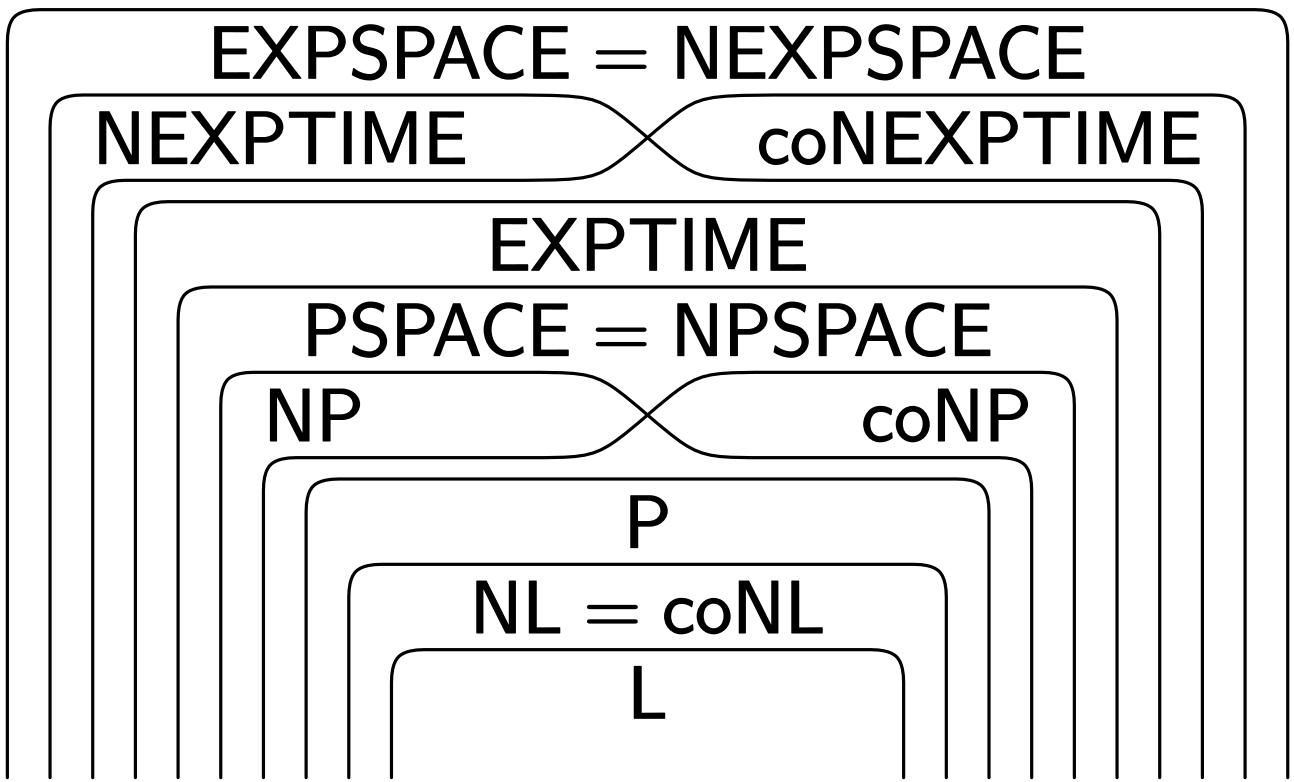
性質：NL 困難問題から帰着ができたら

NL 困難問題 P から Q への対数領域帰着が存在
$\Rightarrow Q$ は NL 困難

本日のまとめ

本日の目標

- 計算資源として **領域** (空間) を扱えるようになる
- 領域に関する計算複雑性クラスと帰着，完全性を扱えるようになる



次回の予告

次回

次の包含関係を証明する

- NP $\subseteq$ PSPACE
  - NEXPTIME $\subseteq$ EXPSPACE
- NL $\subseteq$ P
  - NPSPACE $\subseteq$ EXPTIME

次の命題を証明する (技法：水増し論法)

- P $=$ NP $\Rightarrow$ EXPTIME $=$ NEXPTIME

クイズ

Q

1.

2.

3.

4.

付録

1. 典型的な P 完全問題

素朴な疑問

P 完全問題，PSPACE 完全問題などを定義したが…

疑問

そもそも P 完全問題，PSPACE 完全問題などは
存在するのか？

回答

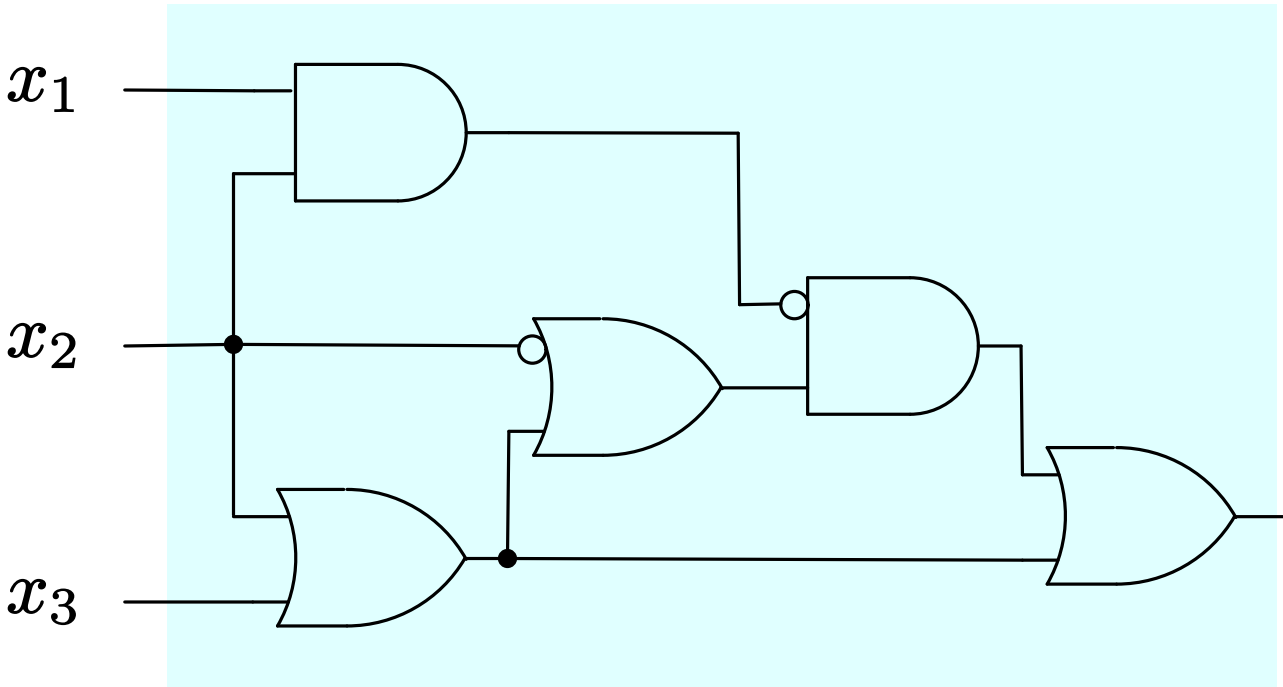
次は存在する

- NL 完全問題 (≡ coNL 完全問題)
- P 完全問題
- PSPACE 完全問題 (≡ NPSPACE 完全問題)
  (≡ coNPSPACE 完全問題)

典型的な P 完全問題を紹介する

(証明はしない)

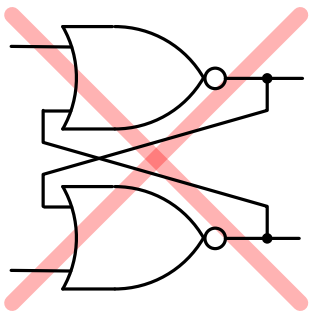
復習：論理回路 (Boolean circuit)



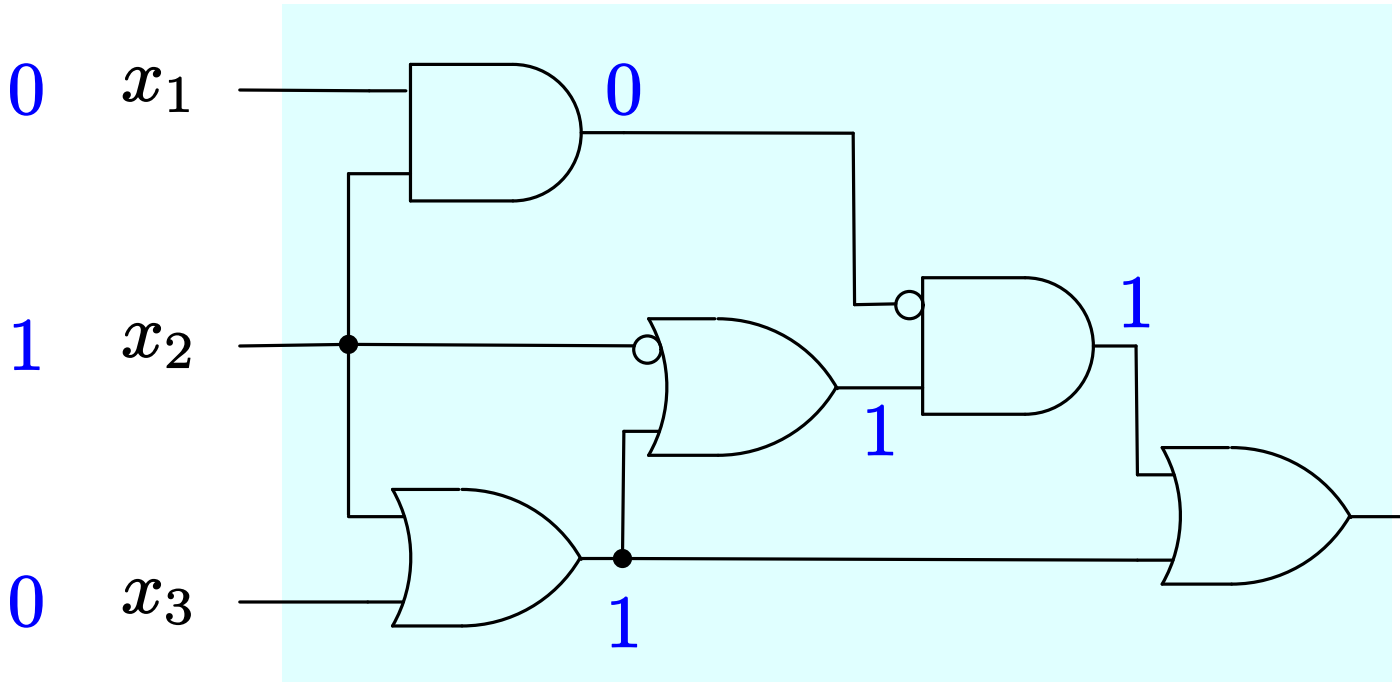
$y = C(x_1, x_2, x_3)$

入力　　C　　出力

注意：フィードバックはない



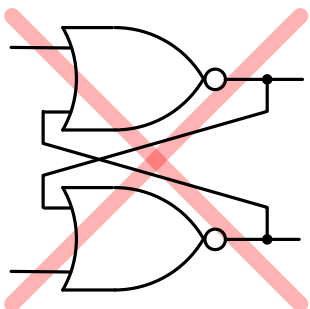
復習：論理回路 (Boolean circuit)



$$y = C(x_1, x_2, x_3)$$

$$1 = C(0, 1, 0)$$

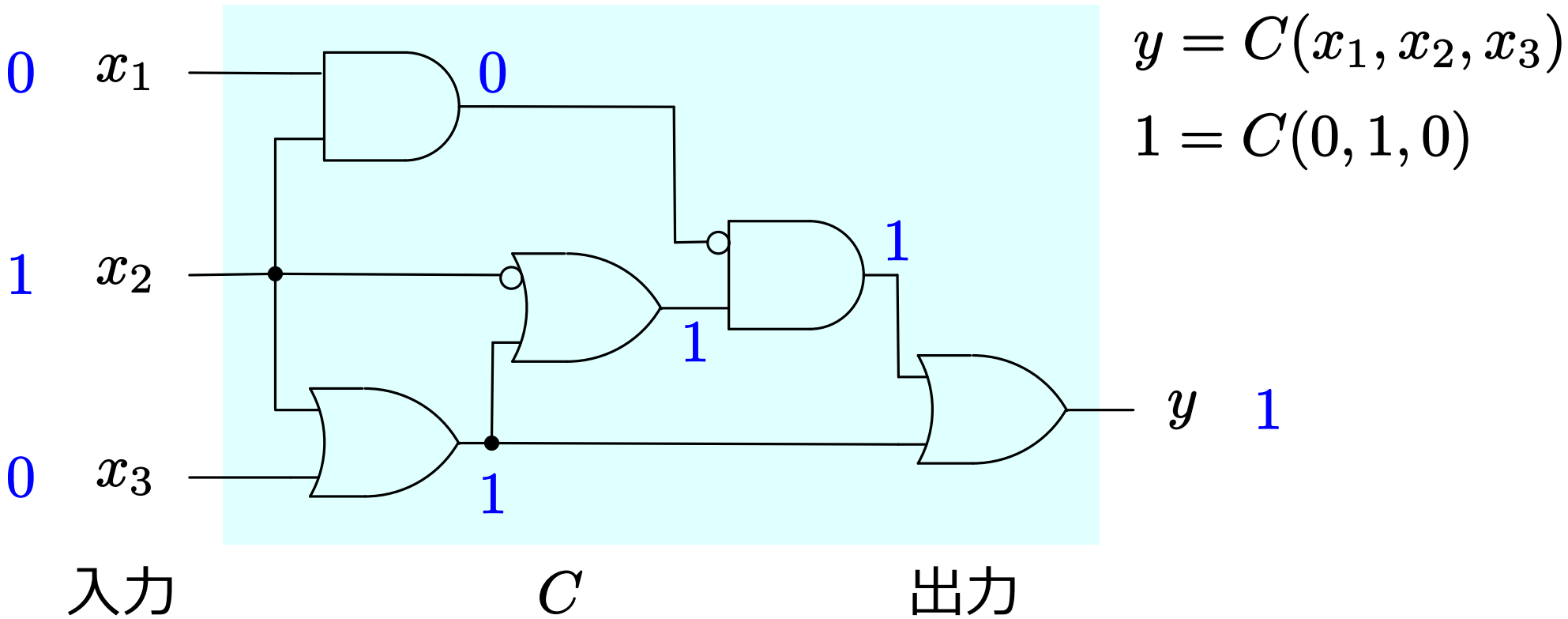
注意：フィードバックはない



論理回路評価問題

問題：論理回路評価問題 (circuit value problem)

入力： 論理回路 C，割当 α
出力： $C(\alpha) = 1 \Rightarrow$ Yes
$C(\alpha) = 0 \Rightarrow$ No



$y = C(x_1, x_2, x_3)$

$1 = C(0, 1, 0)$

論理回路評価問題

問題：論理回路評価問題 (circuit value problem)

入力： 論理回路 C，割当 α
出力： $C(\alpha) = 1 \Rightarrow$ Yes
$C(\alpha) = 0 \Rightarrow$ No

(論理回路値問題，論理回路判定問題と言うこともある)

定理

論理回路評価問題は P 完全

(Ladner '75)

注意：論理式評価問題は (おそらく) P 完全ではない

(Buss '87)