# 理論計算機科学特論 (2026 年前学期)

## 計算複雑性の基礎

## 第 3 回

# 帰着と完全性：NP 完全

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2026 年 4 月 21 日

最終更新：2026 年 4 月 22 日　10:00

# 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)



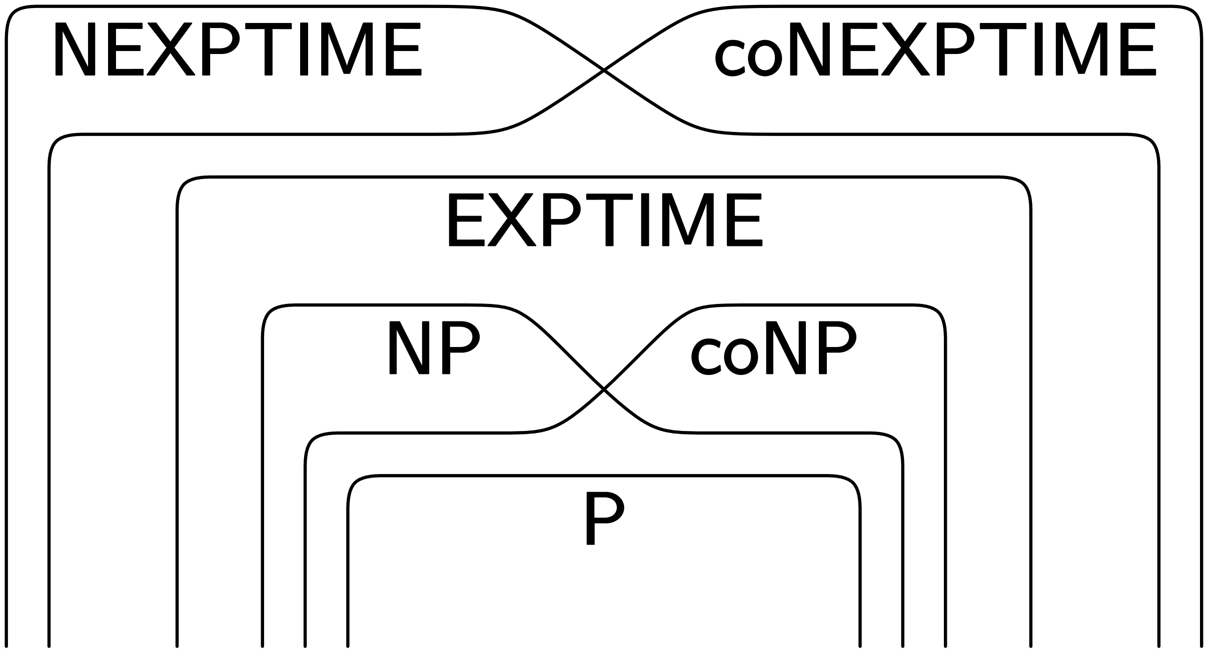
8. Savitch の定理：PSPACE = NPSPACE (6/2)
9. Immerman-Szlepcsényi の定理：NL = coNL (6/9)
10. 多項式階層：P = NP ⇒ P = PH (6/16)
11. 交代性計算：AP = PSPACE (6/23)
12. 確率的計算：P ⊆ BPP ⊆ PP, NP ⊆ PP (6/30)
13. 対話証明系 (1)：NP ⊆ MA ⊆ AM (7/7)
14. 対話証明系 (2)：IP ⊆ PSPACE (7/14)
15. 対話証明系 (3)：PSPACE ⊆ IP (7/21)

* 休み (授業のない日) (7/28)

# 復習：時間計算量



| | 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 $\Rightarrow$ うまく guess をすると Yes を出力
- $I$ が No インスタンス $\Rightarrow$ 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：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)
b = guess(X)
if a + b == 10:
  return "Yes"
else:
  return "No"
end
```

a = 6
b = 4

X が Yes インスタンス
であるための証拠

# 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：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

# 復習：補問題

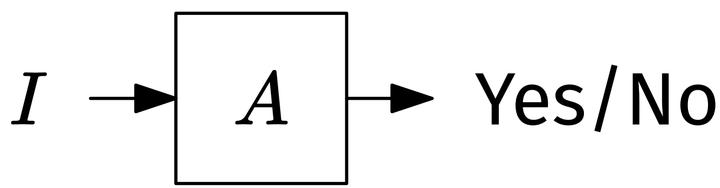


**定義：補問題 (complement)**

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



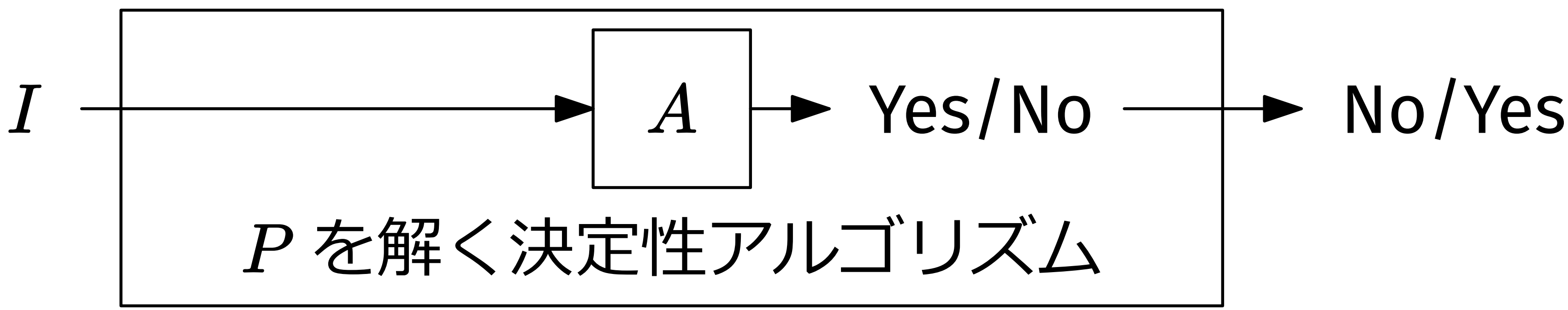
**定義：補問題 (complement)**

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



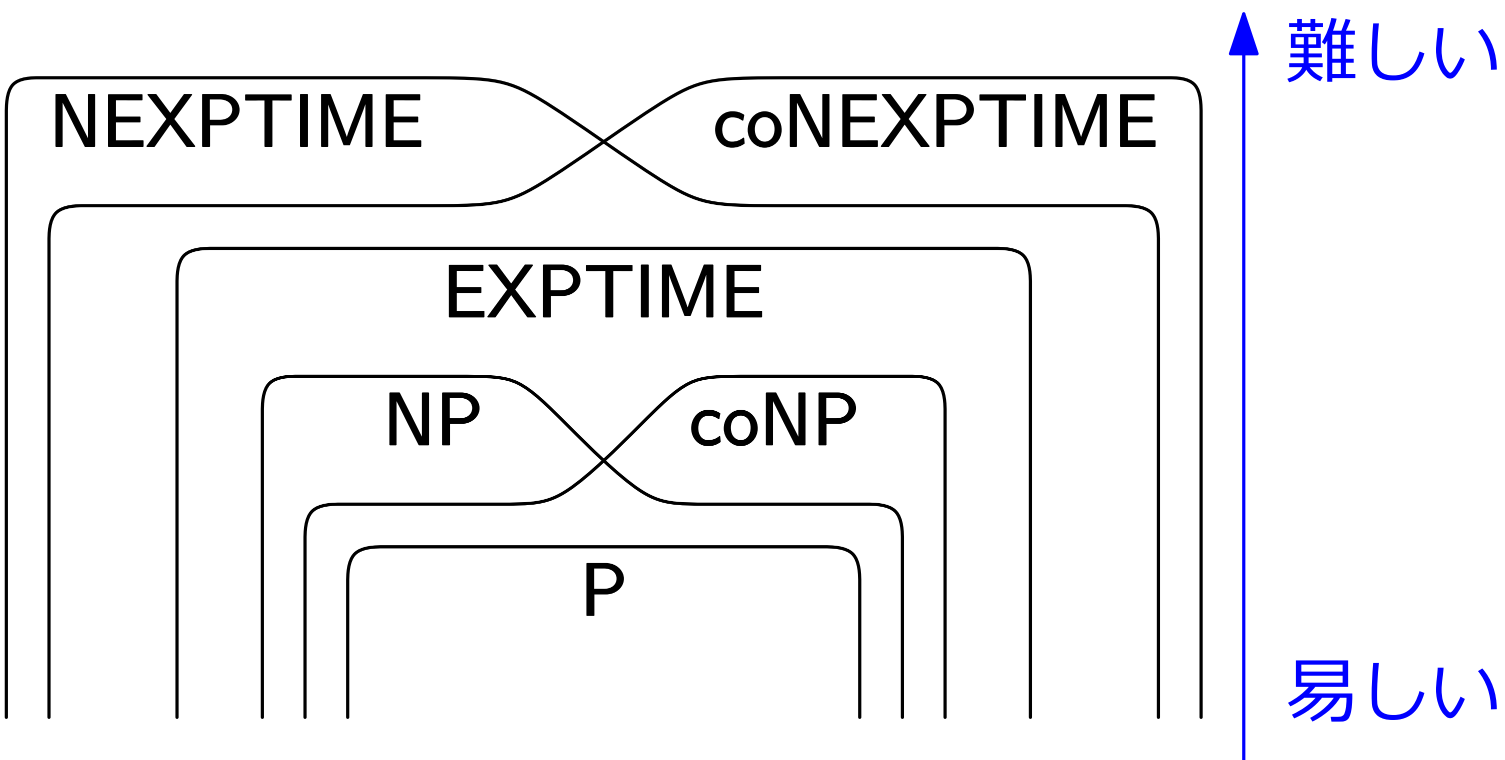

$\overline{P}$ を解く決定性アルゴリズム

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

## 本日の目標

- **帰着** を使って問題の難しさを比較できる
- **完全性** を通して，複雑性クラスの難しさを述べられる

特に，「同じクラスの中の問題どうし」を比較したい



# 本日の内容



1. **帰着とその性質**
2. 困難性と完全性
3. 完全問題の例

# 問題の難しさを比較したい



**問題 $P_1$**

**入力：** $I_1$
**出力：** Yes/No

**問題 $P_2$**

**入力：** $I_2$
**出力：** Yes/No

「$P_1$ が $P_2$ より難しくない」ということを言いたい

(「$P_1$ が $P_2$ より簡単か，同程度に難しい」ということ)

# 問題の難しさを比較したい



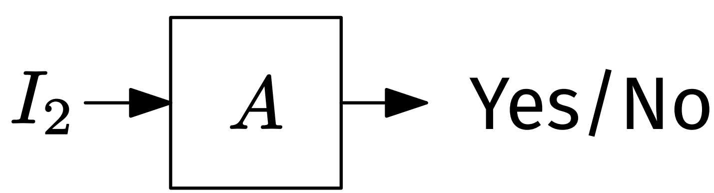
**問題 $P_1$**

**入力：** $I_1$
**出力：** Yes/No

**問題 $P_2$**

**入力：** $I_2$
**出力：** Yes/No

「$P_1$ が $P_2$ より難しくない」ということを言いたい
(「$P_1$ が $P_2$ より簡単か，同程度に難しい」ということ)




$P_2$ を解くアルゴリズム

# 問題の難しさを比較したい



**問題 $P_1$**

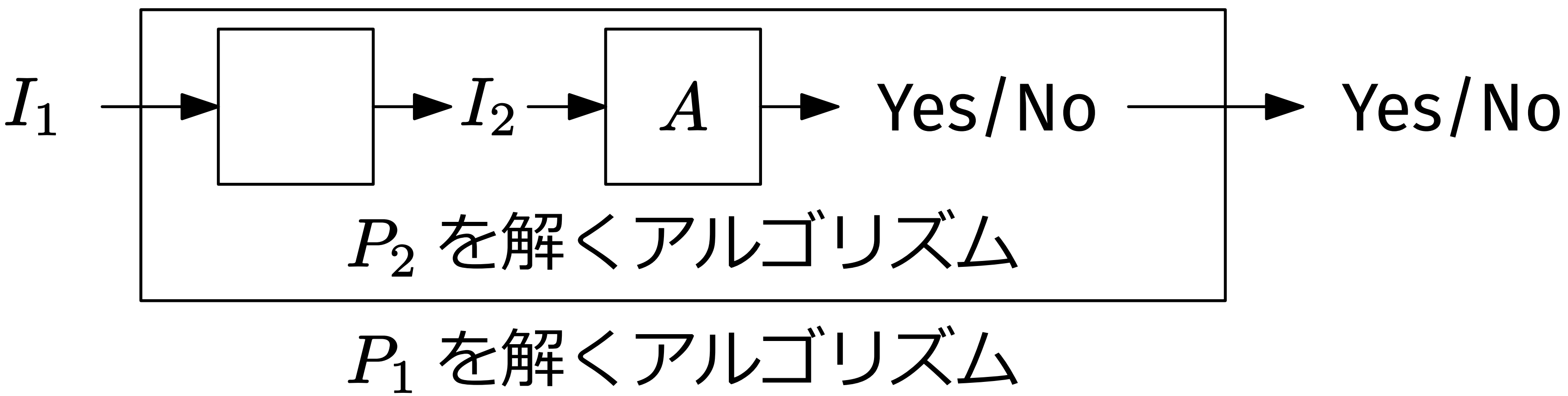
**入力：** $I_1$
**出力：** Yes/No

**問題 $P_2$**

**入力：** $I_2$
**出力：** Yes/No

「$P_1$ が $P_2$ より難しくない」ということを言いたい
(「$P_1$ が $P_2$ より簡単か，同程度に難しい」ということ)

$\leadsto$「$P_2$ を解くアルゴリズムを使って，$P_1$ が解ける」ことを示す



# 問題の難しさを比較したい

**問題 $P_1$**

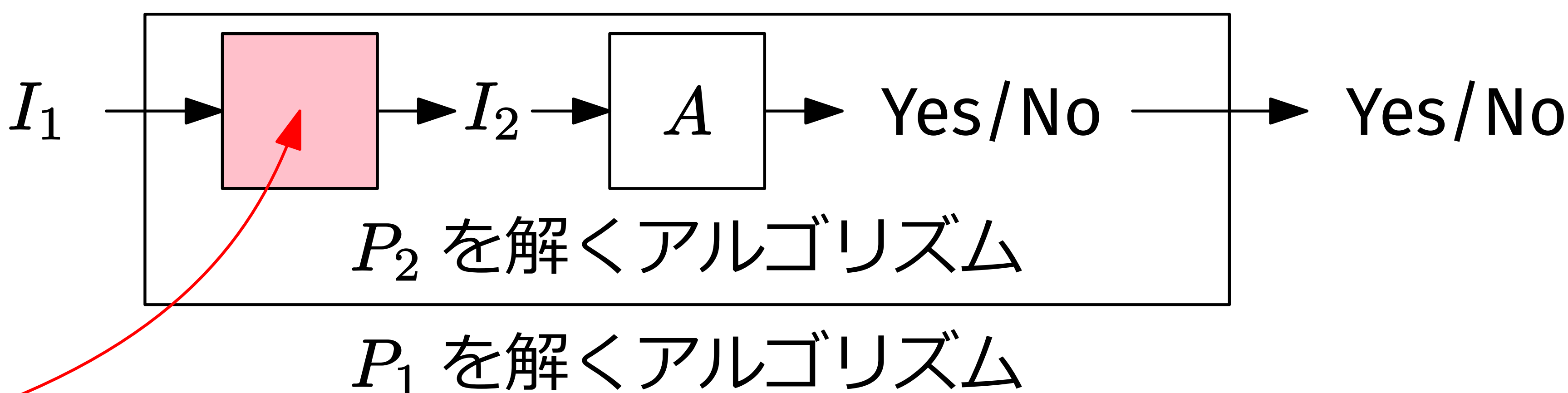
**入力：** $I_1$
**出力：** Yes/No

**問題 $P_2$**

**入力：** $I_2$
**出力：** Yes/No

「$P_1$ が $P_2$ より難しくない」ということを言いたい
(「$P_1$ が $P_2$ より簡単か，同程度に難しい」ということ)

$\leadsto$「$P_2$ を解くアルゴリズムを使って，$P_1$ が解ける」ことを示す



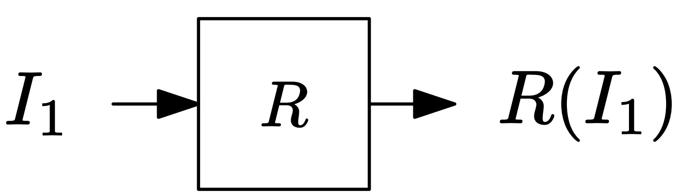

帰着，還元 (reduction)　　「$P_1$ を $P_2$ に帰着する」

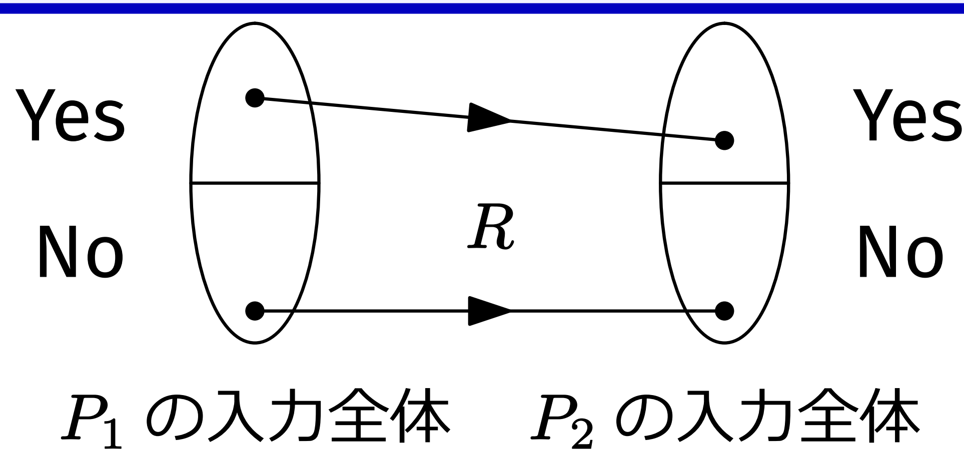
問題 $P_1, P_2$ の任意の入力 $I_1, I_2$

**定義 : 多対一帰着 (たたいいちきちゃく, many-one reduction)**

$P_1$ から $P_2$ への **多対一帰着** とは,
$P_1$ の入力 $I_1$ を $P_2$ の入力に変換するアルゴリズム $R$ で
次を満たすもの

- $I_1$ が $P_1$ の Yes インスタンス
  $\Rightarrow R(I_1)$ が $P_2$ の Yes インスタンス
- $I_1$ が $P_1$ の No インスタンス
  $\Rightarrow R(I_2)$ が $P_2$ の No インスタンス



# 多対一帰着

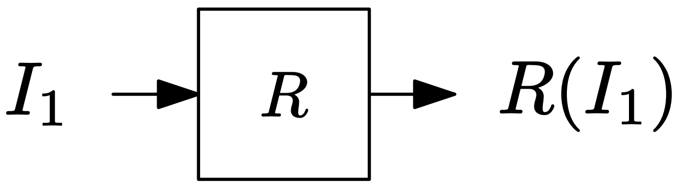


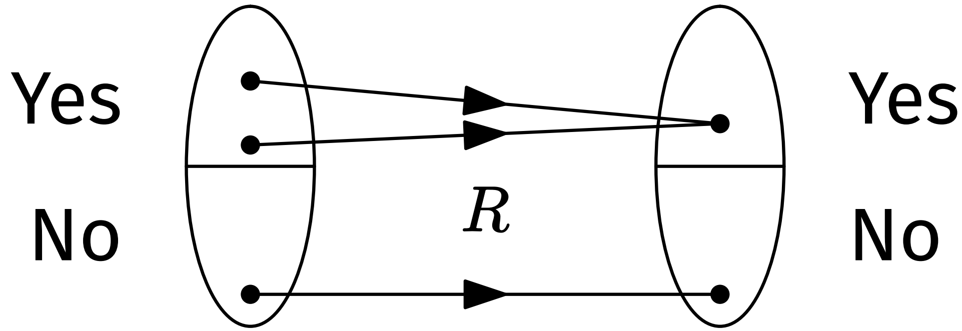
問題 $P_1, P_2$ の任意の入力 $I_1, I_2$

**定義：多対一帰着 (たたいいちきちゃく, many-one reduction)**

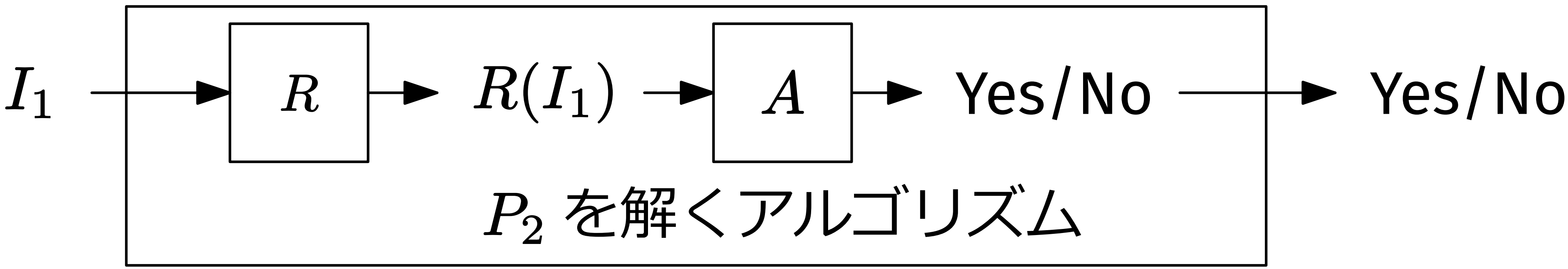
$P_1$ から $P_2$ への **多対一帰着** とは,
$P_1$ の入力 $I_1$ を $P_2$ の入力に変換するアルゴリズム $R$ で
次を満たすもの

- $I_1$ が $P_1$ の Yes インスタンス
  $\Rightarrow R(I_1)$ が $P_2$ の Yes インスタンス
- $I_1$ が $P_1$ の No インスタンス
  $\Rightarrow R(I_2)$ が $P_2$ の No インスタンス



# 多対一帰着 (続)




$P_1$ を解くアルゴリズム

$\therefore P_2$ が解ける $\Rightarrow P_1$ が解ける

直感的な意味：$P_1$ は $P_2$ よりも難しくない

**定義 : 多対一帰着 (たたいいちきちゃく, many-one reduction)**

$P_1$ から $P_2$ への **多対一帰着** とは,
$P_1$ の入力 $I_1$ を $P_2$ の入力に変換するアルゴリズム $R$ で
次を満たすもの

- $I_1$ が $P_1$ の Yes インスタンス
  $\Rightarrow R(I_1)$ が $P_2$ の Yes インスタンス
- $I_1$ が $P_1$ の No インスタンス
  $\Rightarrow R(I_2)$ が $P_2$ の No インスタンス

# 約束：帰着といったら…



**以後…**

この授業では，多対一帰着を単に **帰着** と呼ぶ

注意：計算理論では，いろいろな帰着が用いられる

- 多対一帰着
- チューリング帰着
- 真理値表帰着
- …

その中でも，多対一帰着がよく使われる

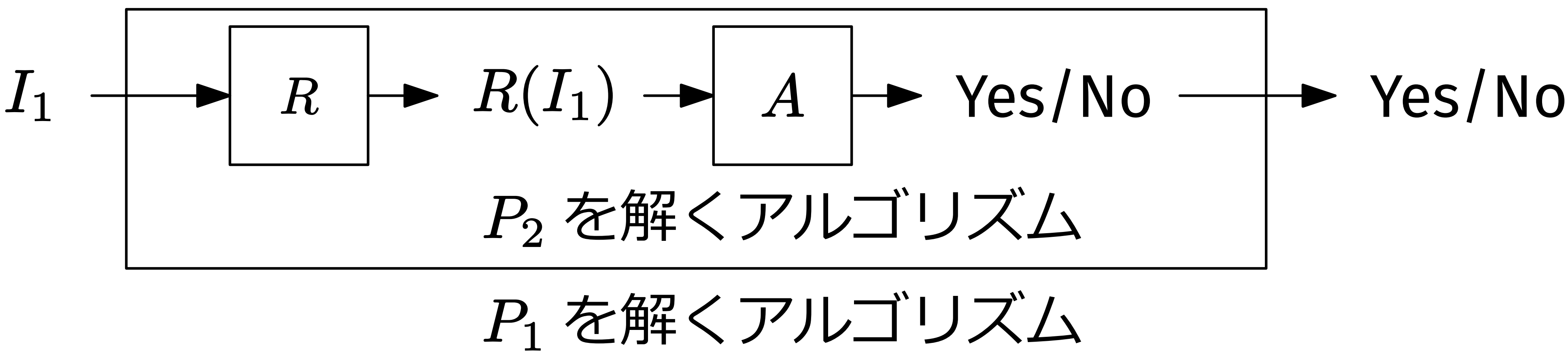
# 多項式時間帰着



問題 $P_1, P_2$

**定義：多項式時間帰着**

$P_1$ から $P_2$ への **多項式時間帰着** とは,
$P_1$ から $P_2$ への帰着 $R$ で,
$R$ の時間計算量が多項式であること



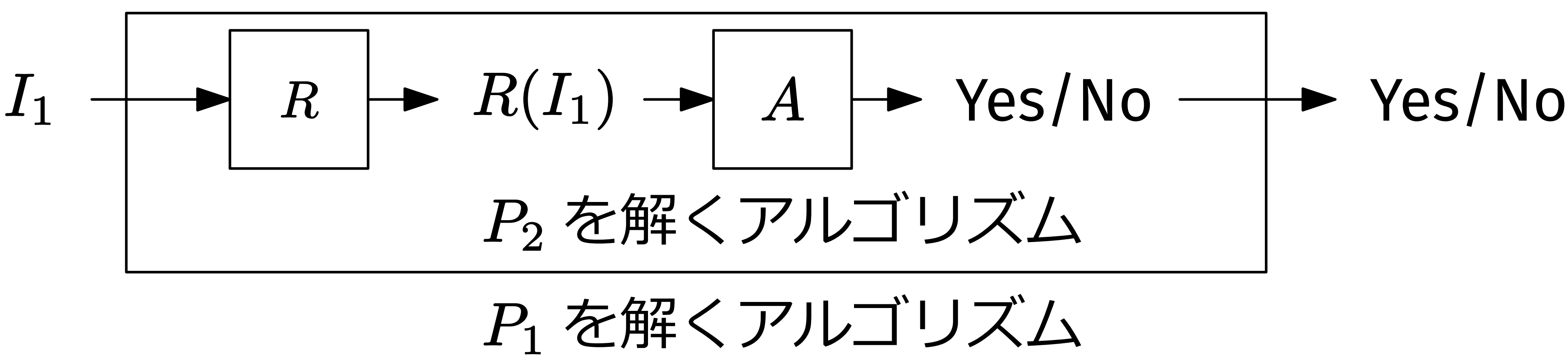
# 多項式時間帰着と複雑性クラス



問題 $P_1, P_2$

**性質：多項式時間帰着と多項式時間アルゴリズム**

$P_1$ から $P_2$ への多項式時間帰着が存在
かつ，$P_2 \in \mathrm{P}$
$\Rightarrow P_1 \in \mathrm{P}$




注：$R$ の時間計算量 $= O(|I_1|^k) \Rightarrow |R(I_1)| = O(|I_1|^k)$

# 多項式時間帰着と複雑性クラス

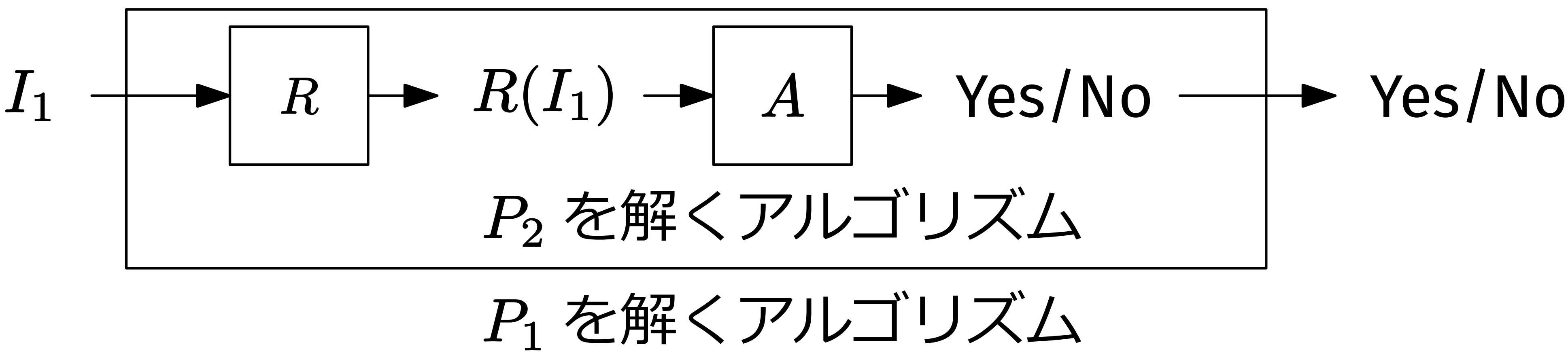


問題 $P_1, P_2$

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

**性質：多項式時間帰着と複雑性クラス**

$P_1$ から $P_2$ への多項式時間帰着が存在
かつ，$P_2 \in \mathcal{C}$ $\Rightarrow P_1 \in \mathcal{C}$



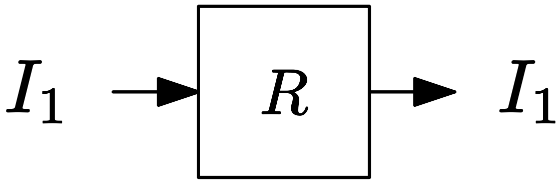

注：$R$ の時間計算量 $= O(|I_1|^k) \Rightarrow |R(I_1)| = O(|I_1|^k)$

# 多項式時間帰着の性質：反射性と推移性 

問題 $P_1, P_2, P_3$

**性質：多項式時間帰着の反射性と推移性**

1. $P_1$ から $P_1$ への多項式時間帰着が存在
2. $P_1$ から $P_2$ への多項式時間帰着が存在，かつ，
   $P_2$ から $P_3$ への多項式時間帰着が存在
   $\Rightarrow$ $P_1$ から $P_3$ への多項式時間帰着が存在



# 多項式時間帰着の性質：反射性と推移性 

問題 $P_1, P_2, P_3$

**性質：多項式時間帰着の反射性と推移性**

1. $P_1$ から $P_1$ への多項式時間帰着が存在
2. $P_1$ から $P_2$ への多項式時間帰着が存在，かつ，
   $P_2$ から $P_3$ への多項式時間帰着が存在
   $\Rightarrow$ $P_1$ から $P_3$ への多項式時間帰着が存在

$$I_1 \longrightarrow \boxed{R} \longrightarrow R(I_1) \qquad I_2 \longrightarrow \boxed{R'} \longrightarrow R'(I_2)$$

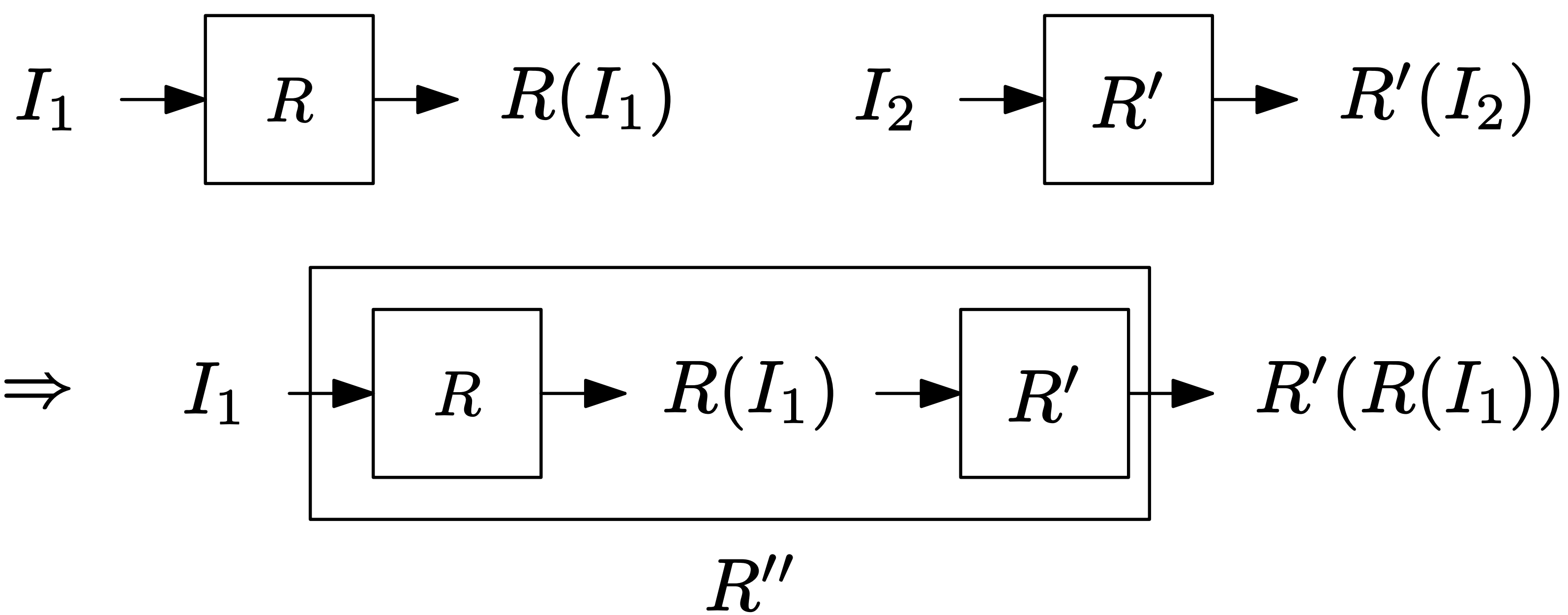
$$\Rightarrow \quad I_1 \longrightarrow \boxed{R} \longrightarrow R(I_1) \longrightarrow \boxed{R'} \longrightarrow R'(R(I_1))$$

# 多項式時間帰着の性質：反射性と推移性 

問題 $P_1, P_2, P_3$

**性質：多項式時間帰着の反射性と推移性**

1. $P_1$ から $P_1$ への多項式時間帰着が存在
2. $P_1$ から $P_2$ への多項式時間帰着が存在，かつ，
   $P_2$ から $P_3$ への多項式時間帰着が存在
   $\Rightarrow$ $P_1$ から $P_3$ への多項式時間帰着が存在



# 多項式時間帰着：反対称性の不成立



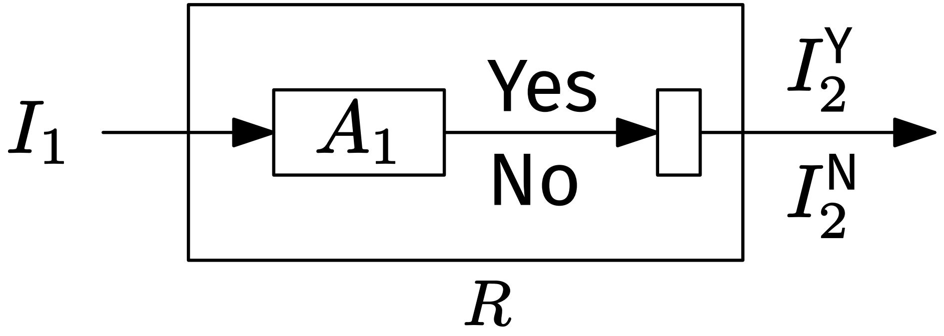
問題 $P_1, P_2$ (非自明：Yes インスタンスと No インスタンスを持つ)

**性質：多項式時間多対一帰着とクラス P**

$P_1 \in \mathsf{P} \Rightarrow P_1$ から $P_2$ への多項式時間帰着が存在

証明：仮定より，$P_1$ に多項式時間アルゴリズムが存在

- $A_1$ を $P_1$ を解く多項式時間アルゴリズムとする
- $I_2^{\mathsf{Y}}$ を $P_2$ の Yes インスタンスの 1 つ，
  $I_2^{\mathsf{N}}$ を $P_2$ の No インスタンスの 1 つとする
- 多項式時間帰着 $R$ を次のように構成する
  (要確認：これが本当に多項式時間帰着であること) □



# 多項式時間帰着：反対称性の不成立



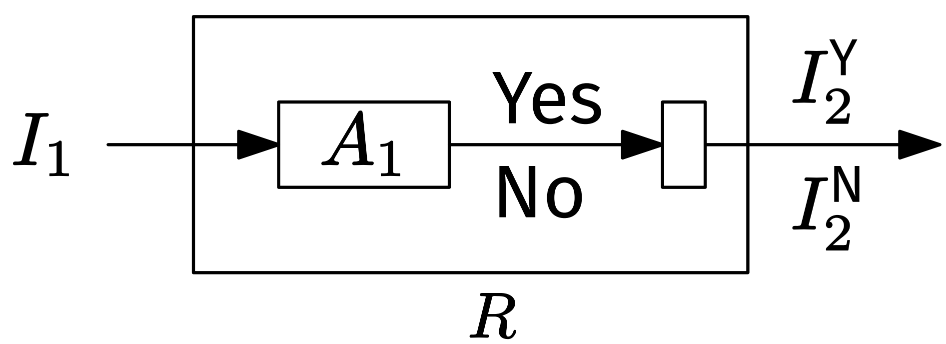
問題 $P_1, P_2$ (非自明：Yes インスタンスと No インスタンスを持つ)

**性質：多項式時間多対一帰着とクラス P**

$P_1 \in \mathsf{P} \Rightarrow P_1$ から $P_2$ への多項式時間帰着が存在

証明：仮定より，$P_1$ に多項式時間アルゴリズムが存在

- $A_1$ を $P_1$ を解く多項式時間アルゴリズムとする
- $I_2^{\mathsf{Y}}$ を $P_2$ の Yes インスタンスの 1 つ，
  $I_2^{\mathsf{N}}$ を $P_2$ の No インスタンスの 1 つとする
- 多項式時間帰着 $R$ を次のように構成する
  (要確認：これが本当に多項式時間帰着であること) □




if $A_1(I_1) ==$ Yes then
  return $I_2^{\mathsf{Y}}$
if $A_1(I_1) ==$ No then
  return $I_2^{\mathsf{N}}$

# 多項式時間帰着：反対称性の不成立 (続) 

問題 $P_1, P_2$ (非自明：Yes インスタンスと No インスタンスを持つ)

**性質：多項式時間多対一帰着とクラス P**

$P_1 \in \mathrm{P} \Rightarrow P_1$ から $P_2$ への多項式時間帰着が存在

**系：反対称性の不成立**

$P_1, P_2 \in \mathrm{P} \Rightarrow$
$P_1$ から $P_2$ への多項式時間帰着が存在，かつ，
$P_2$ から $P_1$ への多項式時間帰着が存在

つまり，P に属する任意の (非自明な) 問題の難しさは同程度？

# 本日の内容



1. 帰着とその性質
2. **困難性と完全性**
3. 完全問題の例

# 困難性

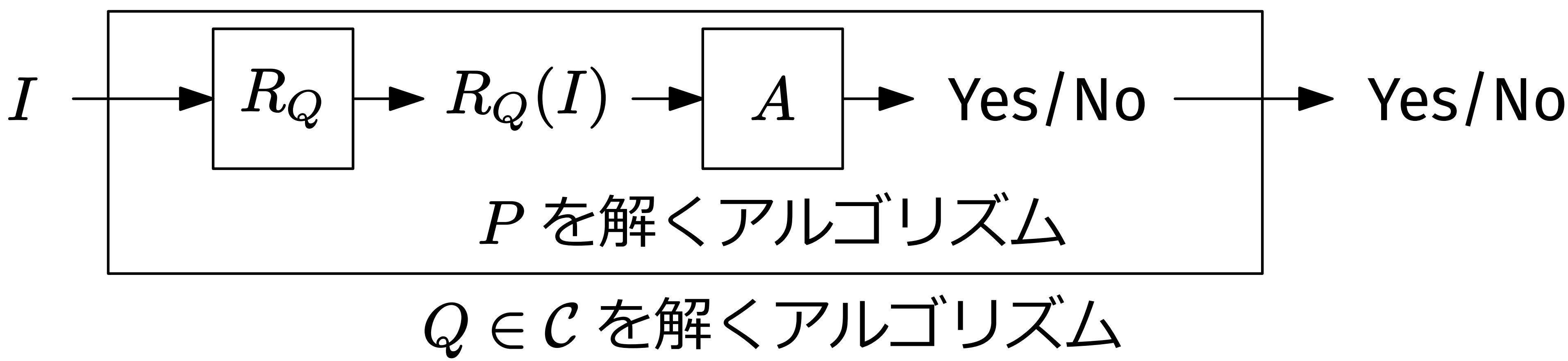


クラス $\mathcal{C} \in \{\mathrm{NP}, \mathrm{coNP}, \mathrm{EXPTIME}, \mathrm{NEXPTIME}, \mathrm{coNEXPTIME}\}$

**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

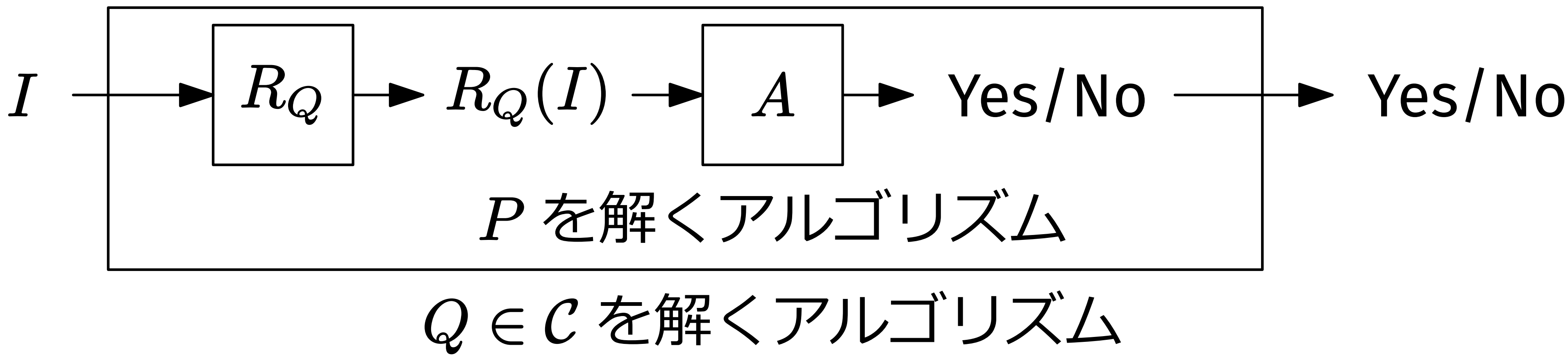


クラス $\mathcal{C} \in \{\mathrm{NP}, \mathrm{coNP}, \mathrm{EXPTIME}, \mathrm{NEXPTIME}, \mathrm{coNEXPTIME}\}$

**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

<u>直感</u>：$P$ 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

# 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

# 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**

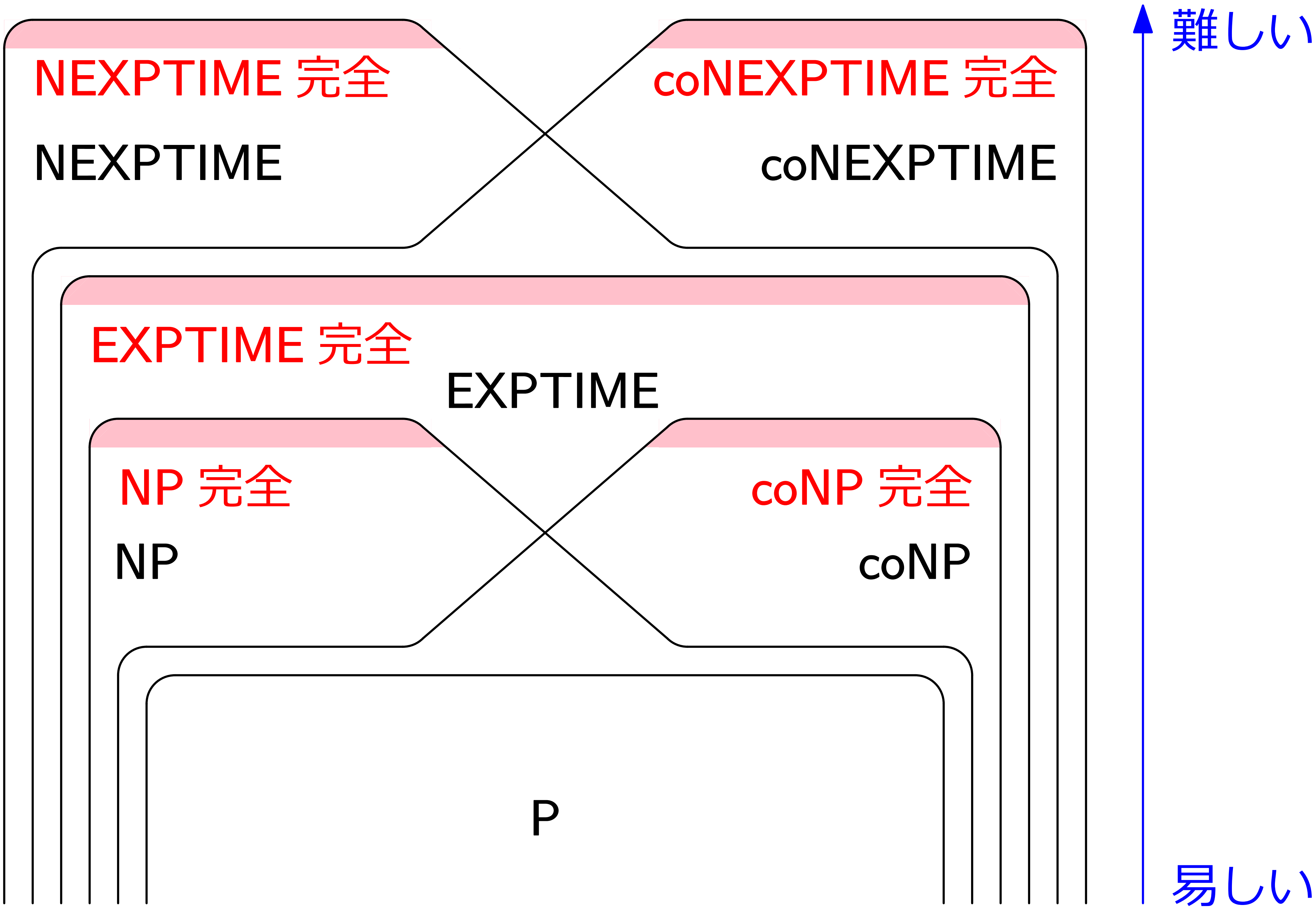
「○○困難/完全」はクラスを表す用語ではない
問題の性質を表す用語である

$\mathcal{C}$ 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

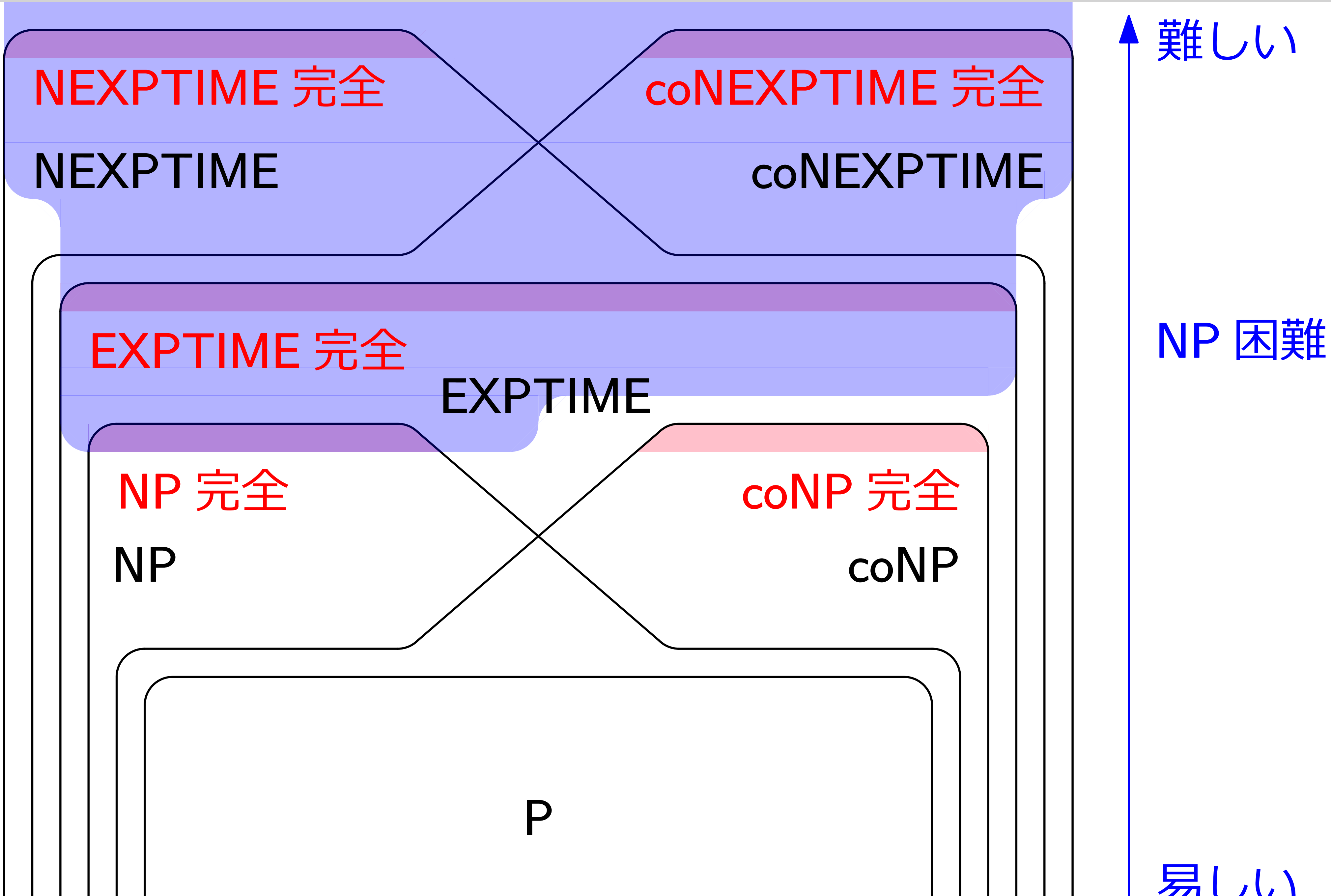
# 完全問題：イメージ




NEXPTIME 完全
coNEXPTIME 完全
NEXPTIME
coNEXPTIME
EXPTIME 完全
EXPTIME
NP 完全
coNP 完全
NP
coNP
P
難しい
易しい

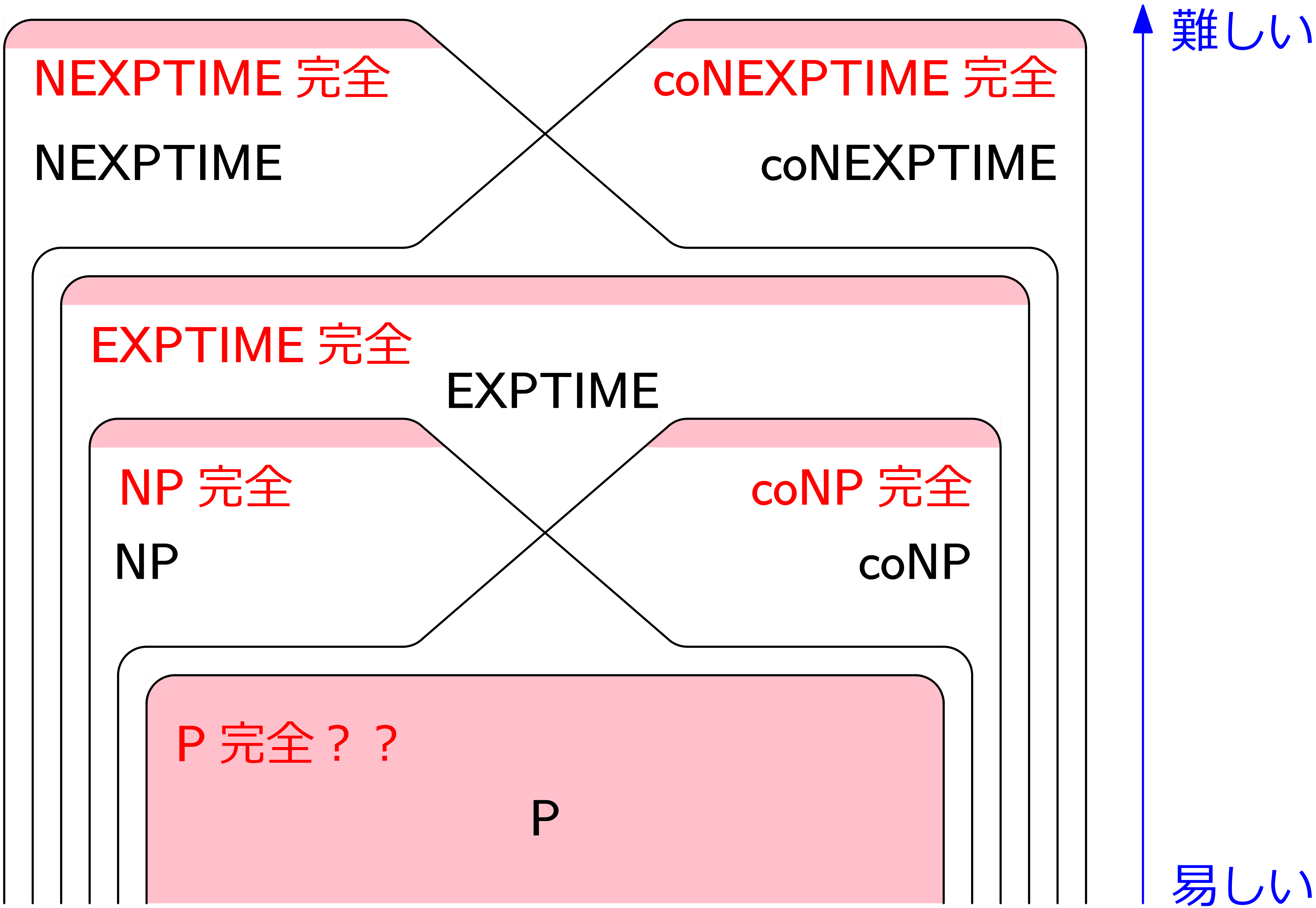
# 完全問題：イメージ




NEXPTIME 完全
coNEXPTIME 完全
NEXPTIME
coNEXPTIME
EXPTIME 完全
EXPTIME
NP 完全
coNP 完全
NP
coNP
P
難しい
NP 困難
易しい

# 完全問題：イメージ



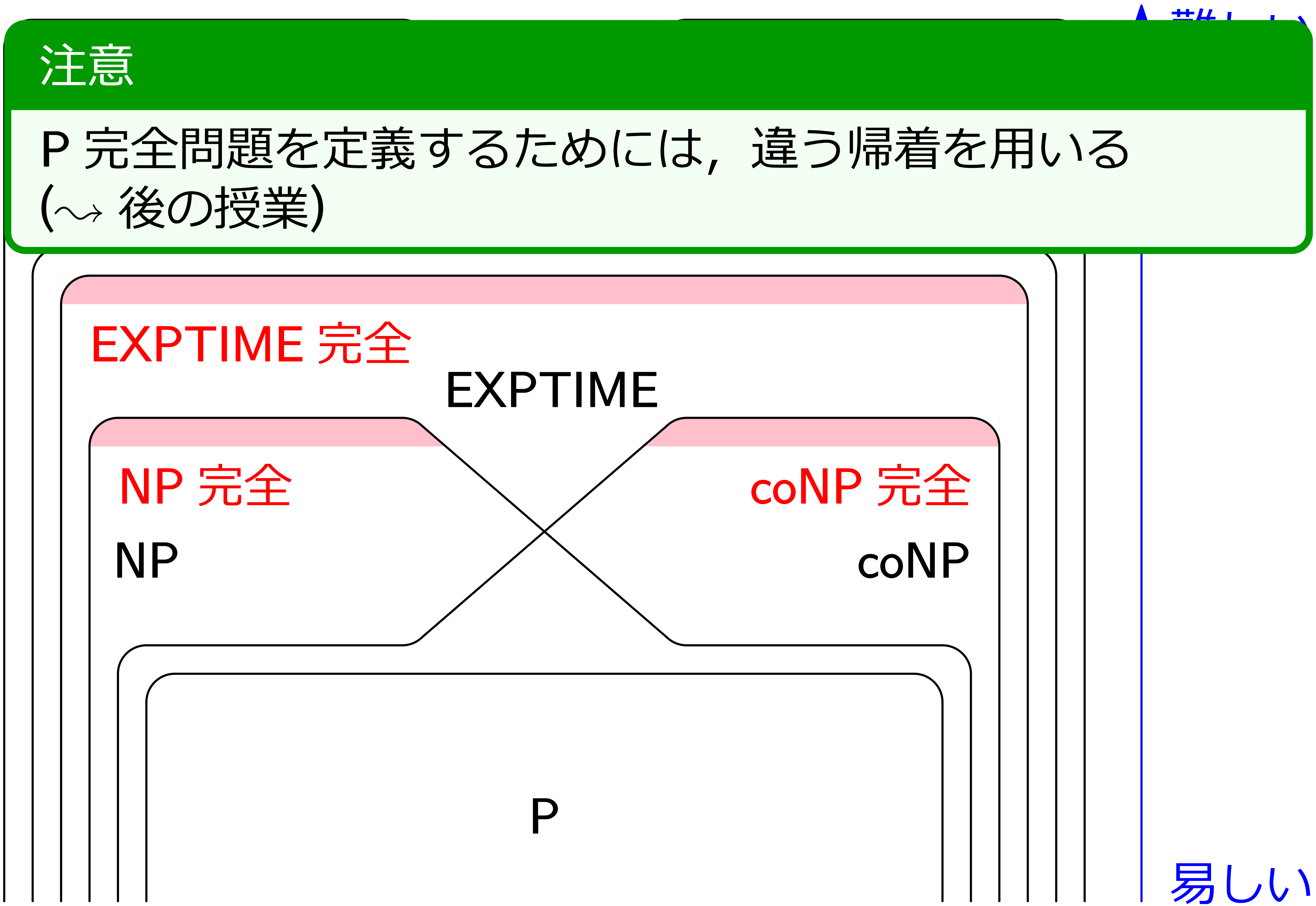


NEXPTIME 完全
coNEXPTIME 完全
NEXPTIME
coNEXPTIME
EXPTIME 完全
EXPTIME
NP 完全
coNP 完全
NP
coNP
P 完全？？
P
難しい
易しい

# 完全問題：イメージ

**注意**

P 完全問題を定義するためには，違う帰着を用いる
($\rightsquigarrow$ 後の授業)



# 完全問題の性質 (1)



問題 $P$

**性質：NP 完全問題が多項式時間で解けたら**

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

帰結：NP 完全問題が 1 つでも多項式時間で解けたら
$\mathrm{P} = \mathrm{NP}$ が正しいと言える

# 完全問題の性質 (1)



問題 $P$

**性質：NP 完全問題が多項式時間で解けたら**

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

<u>証明</u>：$\mathrm{P} \subseteq \mathrm{NP}$ なので，$\mathrm{NP} \subseteq \mathrm{P}$ を示せばよい

- $Q \in \mathrm{NP}$ とする

- $\therefore Q \in \mathrm{P}$

$\square$

# 完全問題の性質 (1)



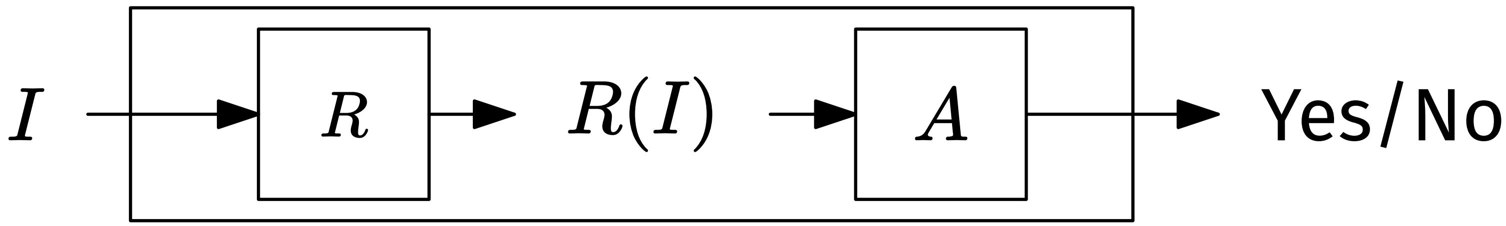
問題 $P$

**性質：NP 完全問題が多項式時間で解けたら**

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

証明：$\mathrm{P} \subseteq \mathrm{NP}$ なので，$\mathrm{NP} \subseteq \mathrm{P}$ を示せばよい

- $Q \in \mathrm{NP}$ とする
- $P$ は NP 完全なので，
  $Q$ から $P$ への多項式時間帰着 $R$ が存在
- $P \in \mathrm{P}$ なので，$P$ を解く多項式時間アルゴリズム $A$ が存在




$Q$ を解く多項式時間アルゴリズム

- $\therefore Q \in \mathrm{P}$ $\square$

# 完全問題の性質 (1)：他のクラスも同様 

問題 $P$

**性質：coNP 完全問題が多項式時間で解けたら**

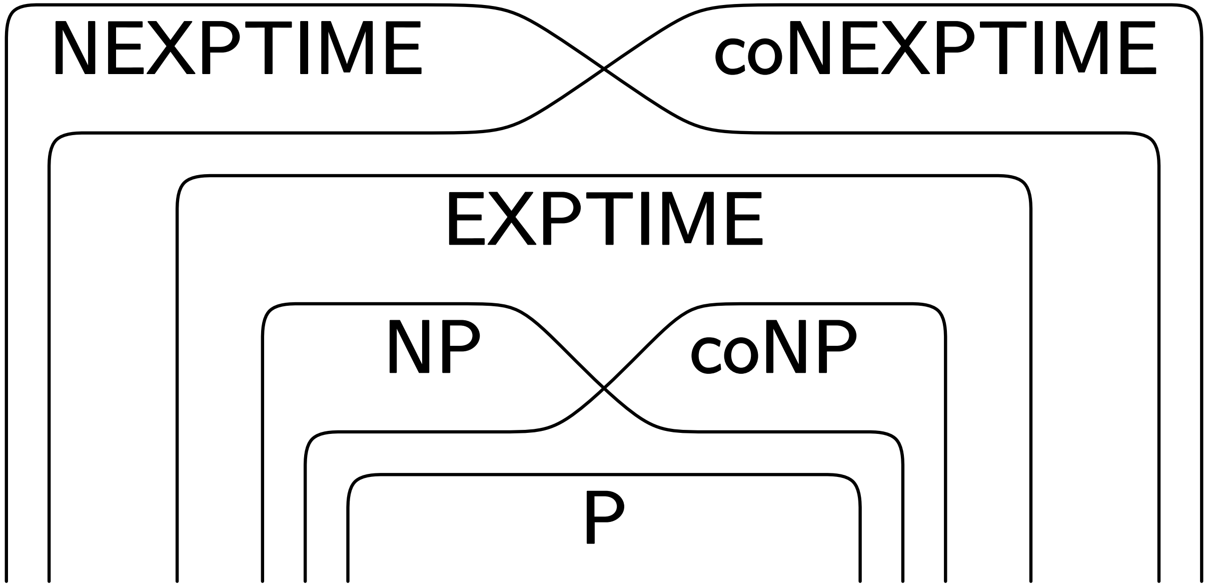
$P$ が coNP 完全，かつ，$P \in \mathsf{P} \Rightarrow \mathsf{P} = \mathsf{coNP}$

**性質：EXPTIME 完全問題が非決定性多項式時間で⋯**

$P$ が EXPTIME 完全，かつ，$P \in \mathsf{NP} \Rightarrow \mathsf{NP} = \mathsf{EXPTIME}$

など，⋯

証明は同様



# 完全問題の性質 (2)



問題 $P, Q$

**性質：NP 困難問題から帰着ができたら⋯**

$P$ が NP 困難，$P$ から $Q$ への多項式時間帰着が存在
$\Rightarrow Q$ は NP 困難

帰結：問題 $Q$ が NP 困難であることを証明するには，
NP 困難問題 $1$ つを帰着すれば十分

# 完全問題の性質 (2)



問題 $P, Q$

**性質：NP 困難問題から帰着ができたら…**

$P$ が NP 困難，$P$ から $Q$ への多項式時間帰着が存在
$\Rightarrow Q$ は NP 困難

証明：任意の問題 $P' \in$ NP を考える

- $\therefore P'$ から $Q$ への多項式時間帰着が存在 $\square$

# 完全問題の性質 (2)



問題 $P, Q$

**性質：NP 困難問題から帰着ができたら⋯**

$P$ が NP 困難，$P$ から $Q$ への多項式時間帰着が存在
$\Rightarrow Q$ は NP 困難

証明：任意の問題 $P' \in \mathrm{NP}$ を考える

- $R$ を $P$ から $Q$ への多項式時間帰着とする
- $P$ は NP 困難なので，
  $P'$ から $P$ への多項式時間帰着 $R'$ が存在

$$I' \longrightarrow \boxed{R'} \longrightarrow R'(I') \longrightarrow \boxed{R} \longrightarrow R(R'(I'))$$

問題 $P'$ の入力　問題 $P$ の入力　問題 $Q$ の入力

- $\therefore P'$ から $Q$ への多項式時間帰着が存在 $\square$

# 完全問題の性質 (2)



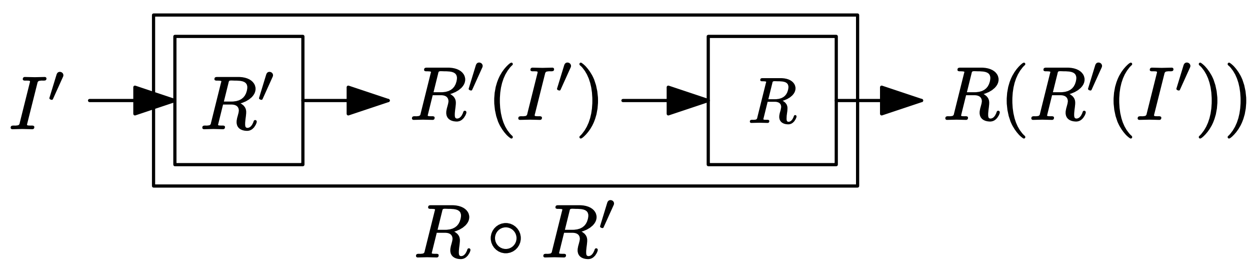
問題 $P, Q$

**性質：NP 困難問題から帰着ができたら…**

$P$ が NP 困難，$P$ から $Q$ への多項式時間帰着が存在
$\Rightarrow Q$ は NP 困難

<u>証明</u>：任意の問題 $P' \in$ NP を考える

- $R$ を $P$ から $Q$ への多項式時間帰着とする
- $P$ は NP 困難なので，
  $P'$ から $P$ への多項式時間帰着 $R'$ が存在




- $R \circ R'$ は $P'$ から $Q$ への多項式時間帰着である
- $\therefore P'$ から $Q$ への多項式時間帰着が存在 $\square$

# 完全問題の性質 (2)：他のクラスも同様



問題 $P, Q$

**性質：coNP 困難問題から帰着ができたら⋯**

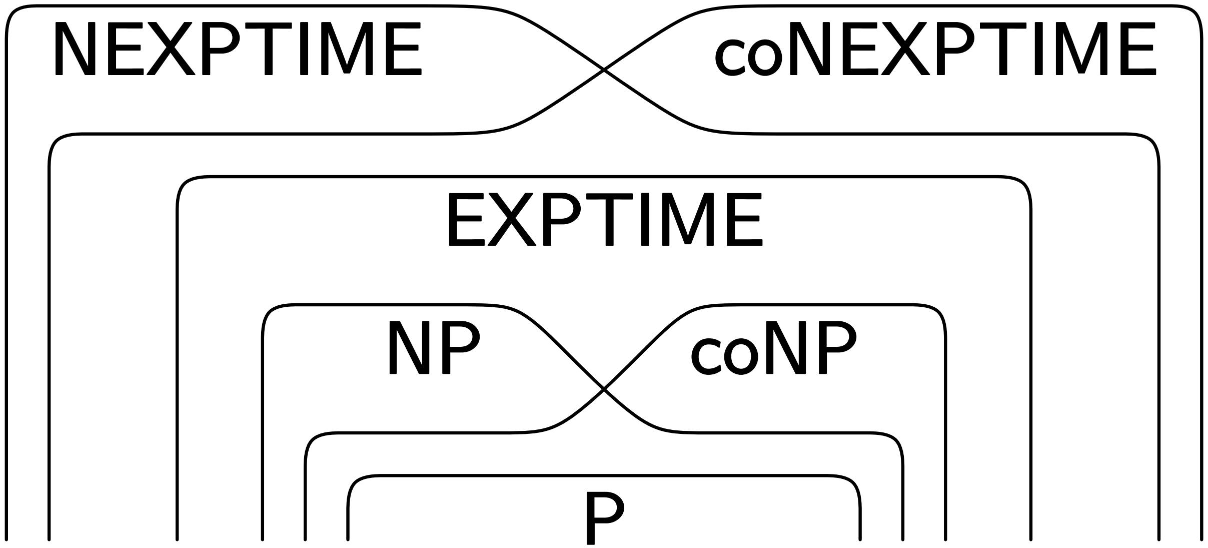
$P$ が coNP 困難，$P$ から $Q$ への多項式時間帰着が存在
$\Rightarrow Q$ は coNP 困難

**性質：EXPTIME 困難問題から帰着ができたら⋯**

$P$ が EXPTIME 困難，$P$ から $Q$ への多項式時間帰着が存在
$\Rightarrow Q$ は EXPTIME 困難

など，⋯

証明は同様



# 本日の内容



1. 帰着とその性質
2. 困難性と完全性
3. **完全問題の例**

# 素朴な疑問



NP 完全問題，EXPTIME 完全問題などを定義したが···

**疑問**

そもそも NP 完全問題，EXPTIME 完全問題などは存在するのか？

# 素朴な疑問



NP 完全問題，EXPTIME 完全問題などを定義したが…

**疑問**

そもそも NP 完全問題，EXPTIME 完全問題などは存在するのか？

**回答**

次は存在する

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

典型的な NP 完全問題，coNP 完全問題を紹介する

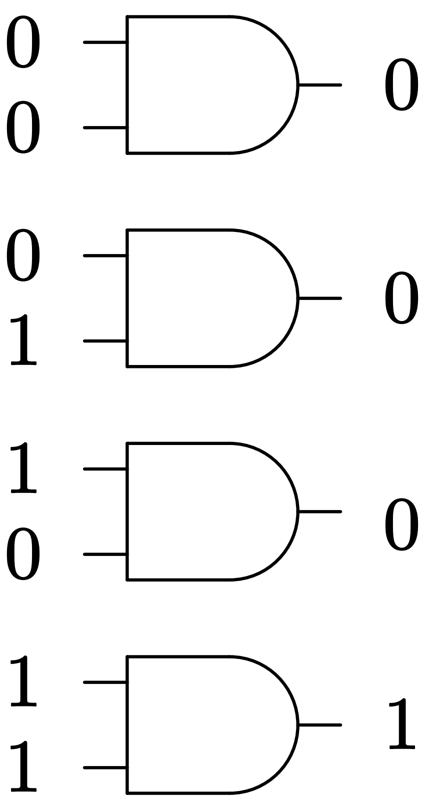
(証明はしない)

$\leadsto$ 論理回路，論理式

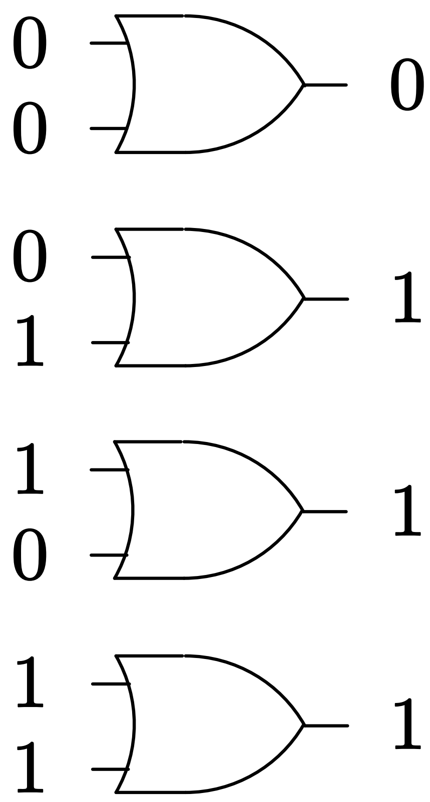
# 復習：論理素子 (logic gate)



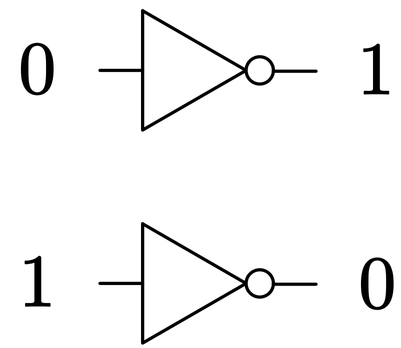
AND 素子
(AND gate)




OR 素子
(OR gate)



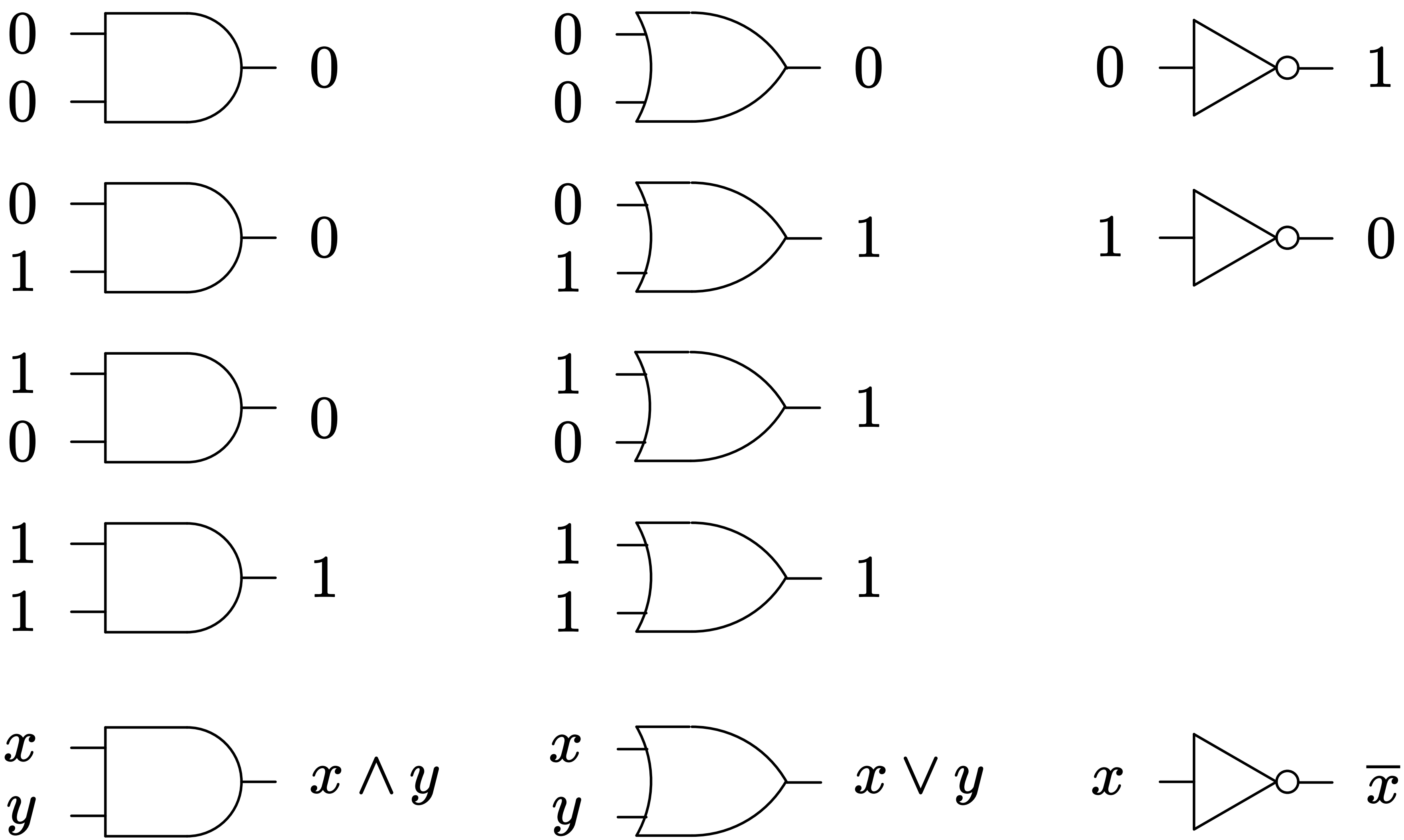

NOT 素子
(NOT gate)



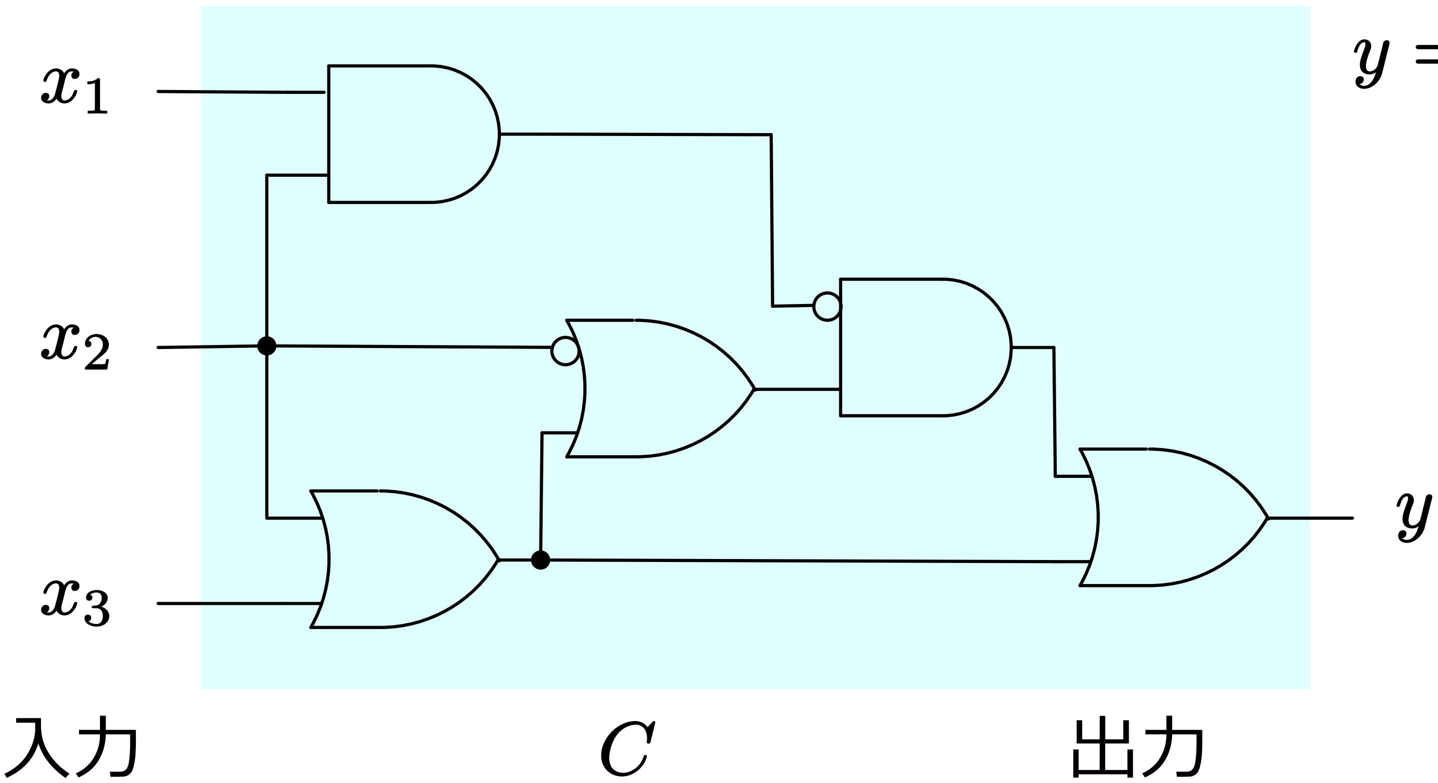
AND 素子
(AND gate)

OR 素子
(OR gate)

NOT 素子
(NOT gate)

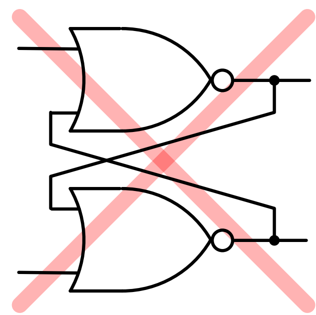


# 復習：論理回路 (Boolean circuit)

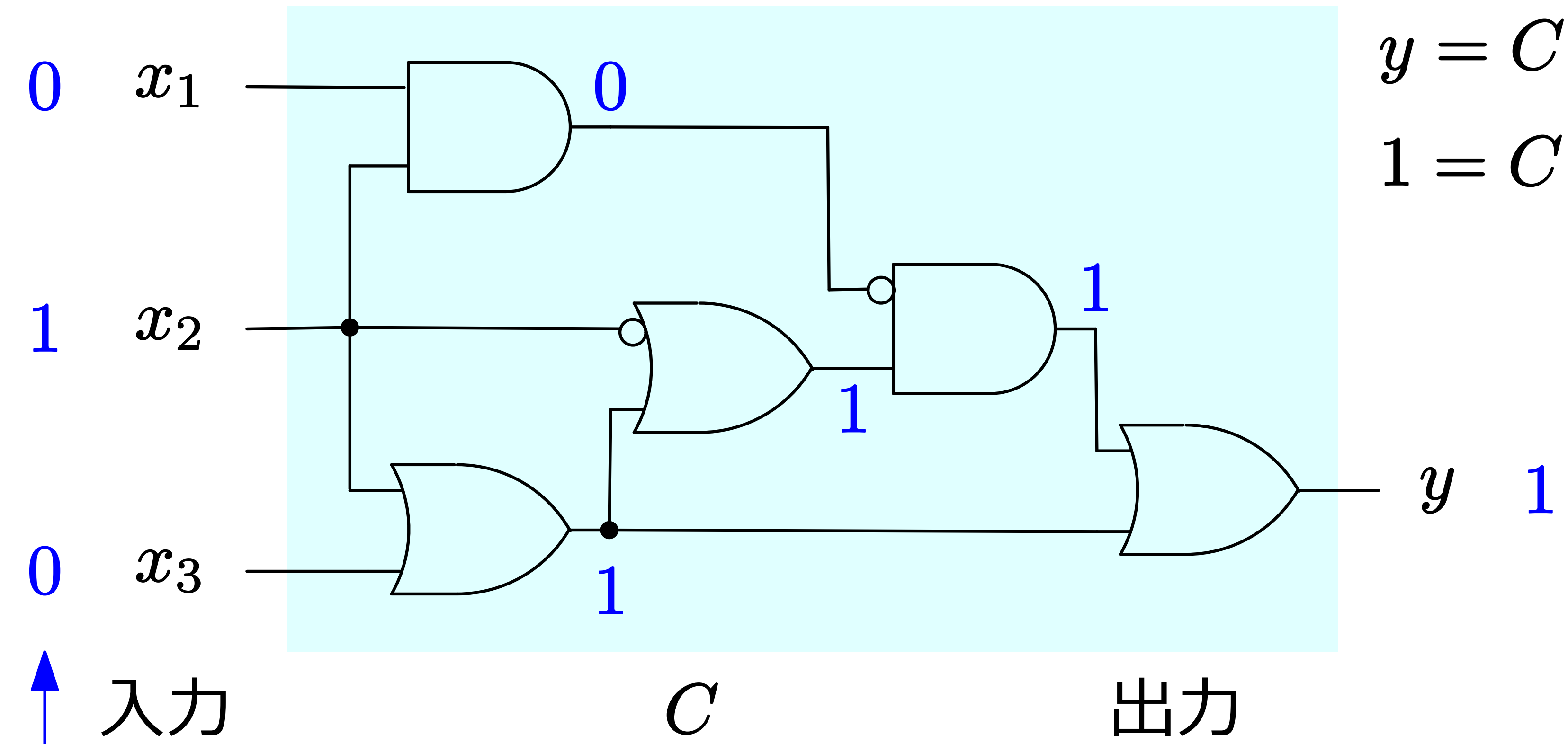



$y = C(x_1, x_2, x_3)$

注意：フィードバックはない



# 復習：論理回路 (Boolean circuit)

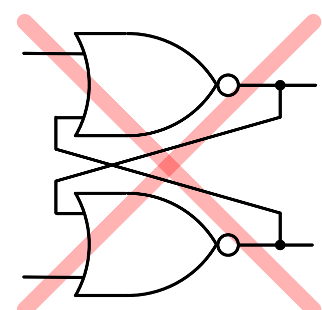


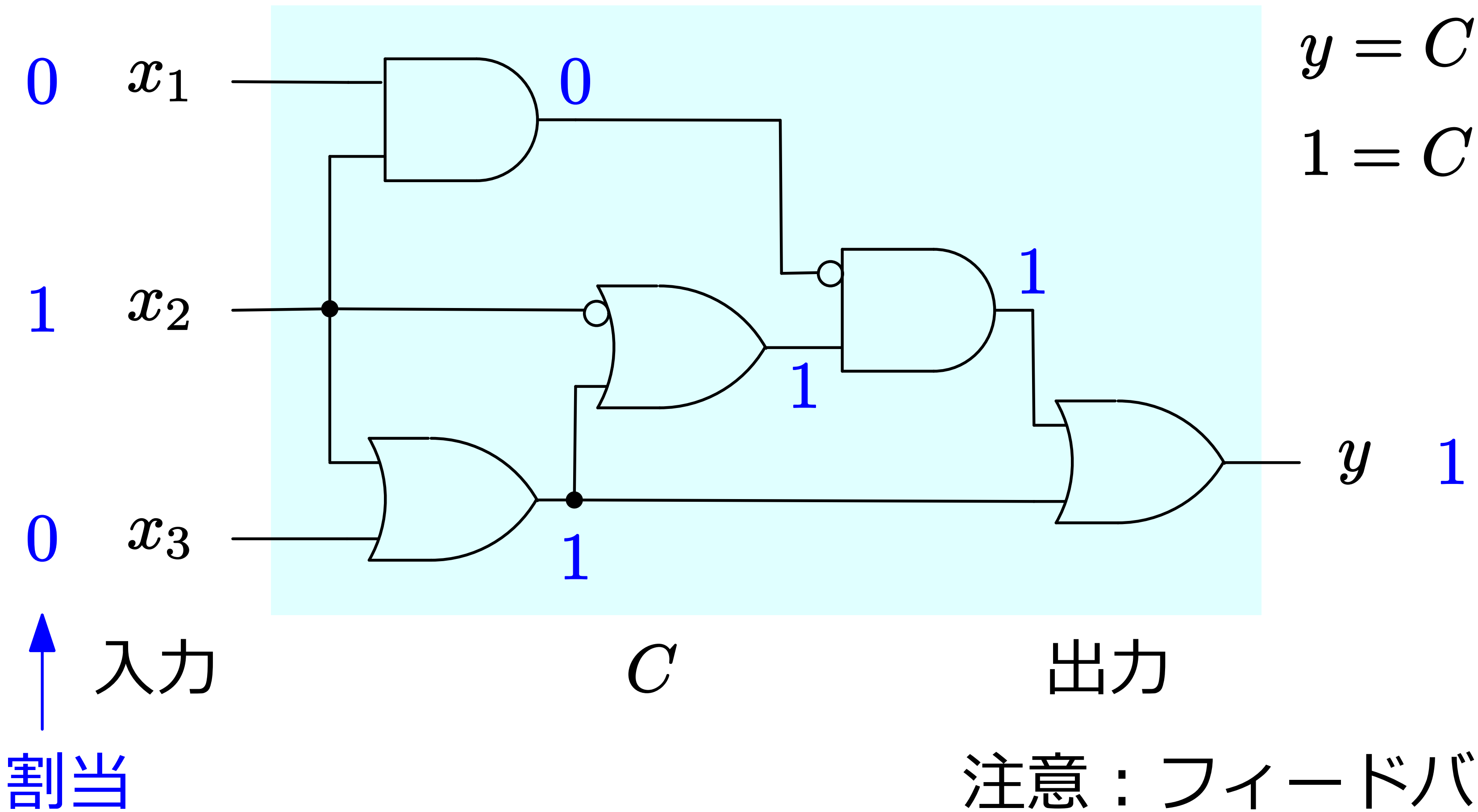

割当

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

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

注意：フィードバックはない






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

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

注意：フィードバックはない

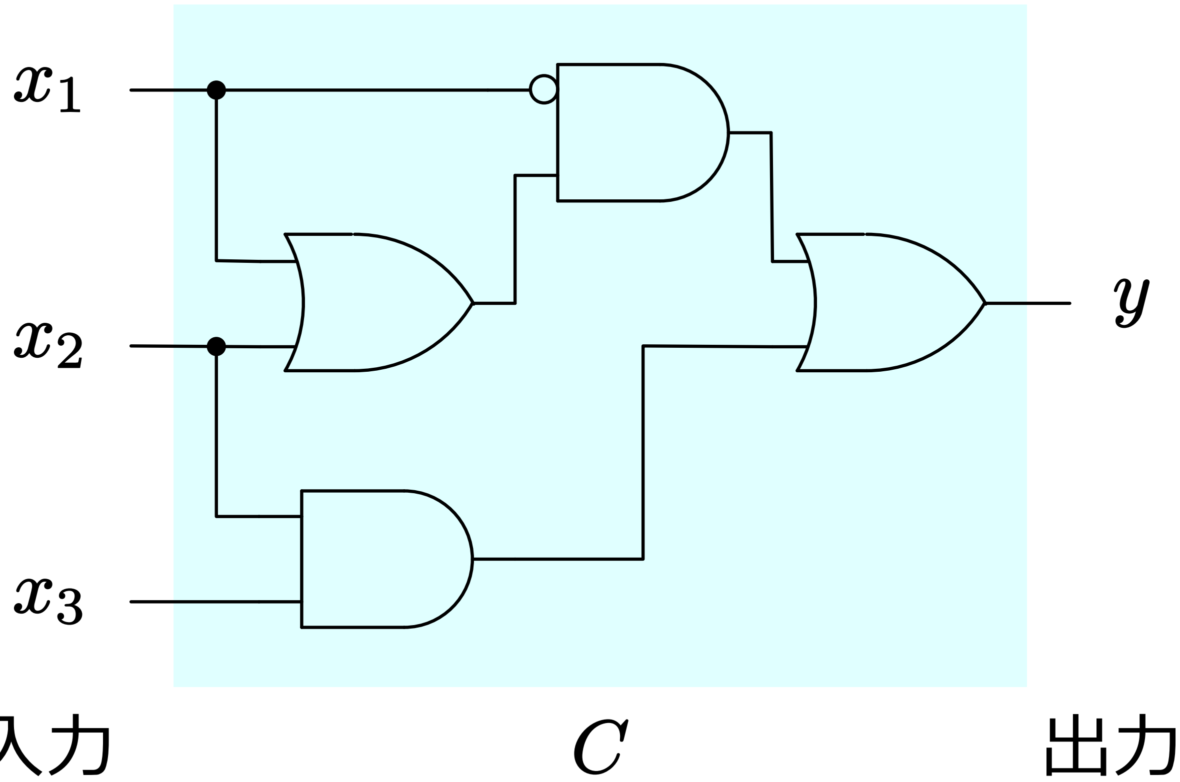
**定義：充足割当 (satisfying assignment)**

論理回路 $C(x)$ に対する割当 $\alpha$ が **充足割当** であるとは $C(\alpha) = 1$ であること

$$\varphi(x_1, x_2, x_3) = (\overline{x_1} \wedge (x_1 \vee x_2)) \vee (x_2 \wedge x_3)$$

# 復習：論理式 (Boolean formula) 

$$\varphi(x_1, x_2, x_3) = (\overline{x_1} \wedge (x_1 \vee x_2)) \vee (x_2 \wedge x_3)$$




論理式から得られる論理回路では,
各素子の出力が枝分かれをしない

**定義 (非形式)：連言標準形**

**連言標準形** の論理式であるとは,
それが「OR の AND」で書かれた論理式のこと

$$(x_1 \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor x_4 \lor x_5) \land (x_4 \lor \overline{x_5}) \land \cdots$$

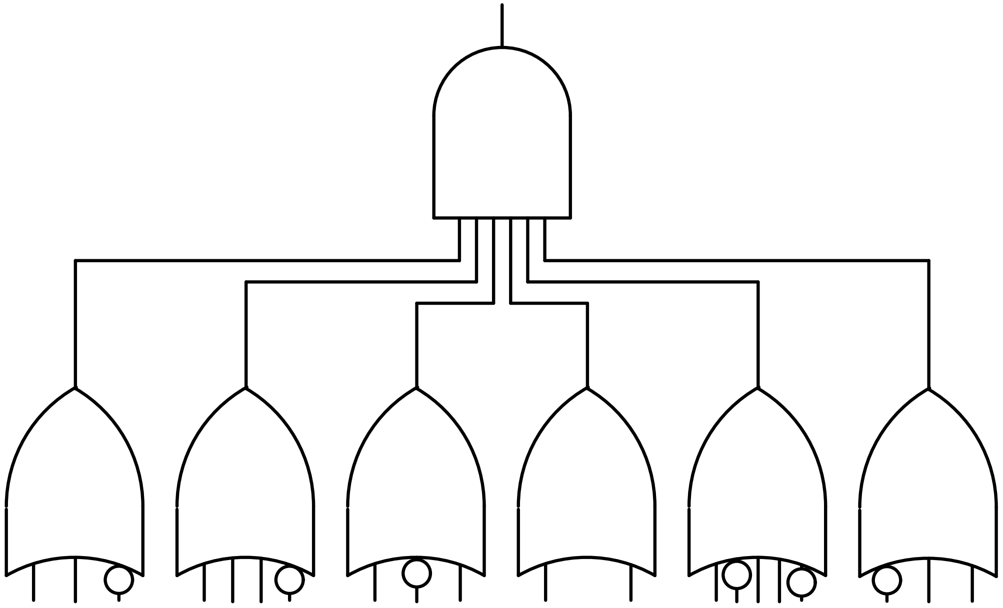
# 復習：連言標準形 (conjunctive normal form) 

**定義 (非形式)：連言標準形**

**連言標準形** の論理式であるとは,
それが「OR の AND」で書かれた論理式のこと

$$\underline{(x_1 \vee x_2 \vee \overline{x_3})} \wedge (x_1 \vee \overline{x_2} \vee x_4 \vee x_5) \wedge (x_4 \vee \overline{x_5}) \wedge \cdots$$

$$\leftrightarrow (x_1 \vee x_2) \vee \overline{x_3}$$



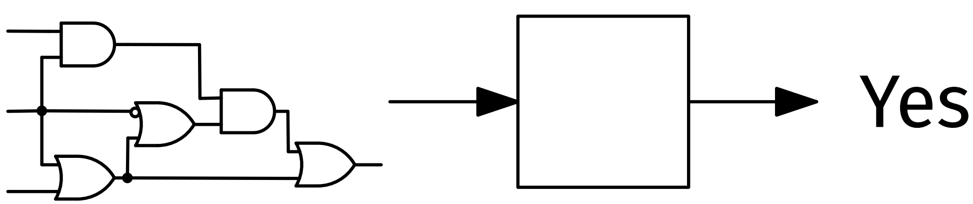
**論理回路の充足可能性問題 (CIRCUIT-SAT)**

**入力：** 論理回路 $C$

**出力：** $C$ が充足割当を持つ $\Rightarrow$ Yes

$C$ が充足割当を持たない $\Rightarrow$ No

充足可能性問題 = Satisfiability Problem




**性質：CIRCUIT-SAT の NP 完全性**

CIRCUIT-SAT は NP 完全

(Cook ’71; Levin ’73)

**論理式の充足可能性問題 (FORMULA-SAT)**

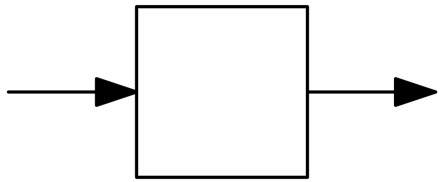
**入力：** 論理式 $\varphi$

**出力：** $\varphi$ が充足割当を持つ $\Rightarrow$ Yes

$\varphi$ が充足割当を持たない $\Rightarrow$ No

充足可能性問題 $=$ Satisfiability Problem

$$(\overline{x_1} \wedge (x_1 \vee x_2)) \vee (x_2 \vee x_3)$$



Yes

**性質：FORMULA-SAT の NP 完全性**

FORMULA-SAT は NP 完全

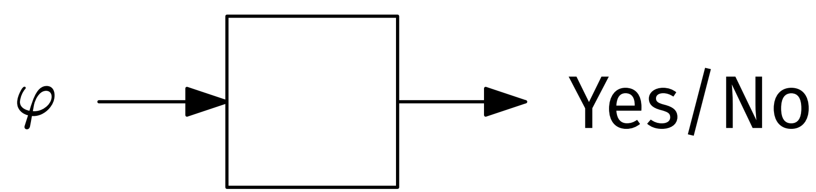
(Cook ’71; Levin ’73)

# 連言標準形論理式の充足可能性問題 

**連言標準形論理式の充足可能性問題 (CNF-SAT)**

**入力：** 連言標準形の論理式 $\varphi$
**出力：** $\varphi$ が充足割当を持つ $\Rightarrow$ Yes
$\varphi$ が充足割当を持たない $\Rightarrow$ No

単に「SAT」といったら，普通は CNF-SAT を指す




**性質：CNF-SAT の NP 完全性**

CNF-SAT は NP 完全

(Cook ’71; Levin ’73)

# 補足：CNF → FORMULA → CIRCUIT



CNF-SAT が NP 完全であることから，
FORMULA-SAT と CIRCUIT-SAT の NP 完全性がすぐ分かる

任意の問題 $P \in \mathrm{NP}$

| | 帰着 | |
|---|---|---|
| $P$ の Yes インスタンス | → | CNF-SAT の Yes インスタンス |
| $P$ の No インスタンス | → | CNF-SAT の No インスタンス |

# 補足：CNF → FORMULA → CIRCUIT



CNF-SAT が NP 完全であることから，

FORMULA-SAT と CIRCUIT-SAT の NP 完全性がすぐ分かる

任意の問題 $P \in$ NP

$P$ の Yes インスタンス → 

FORMULA-SAT の Yes⋯

CNF-SAT の Yes インスタンス

帰着

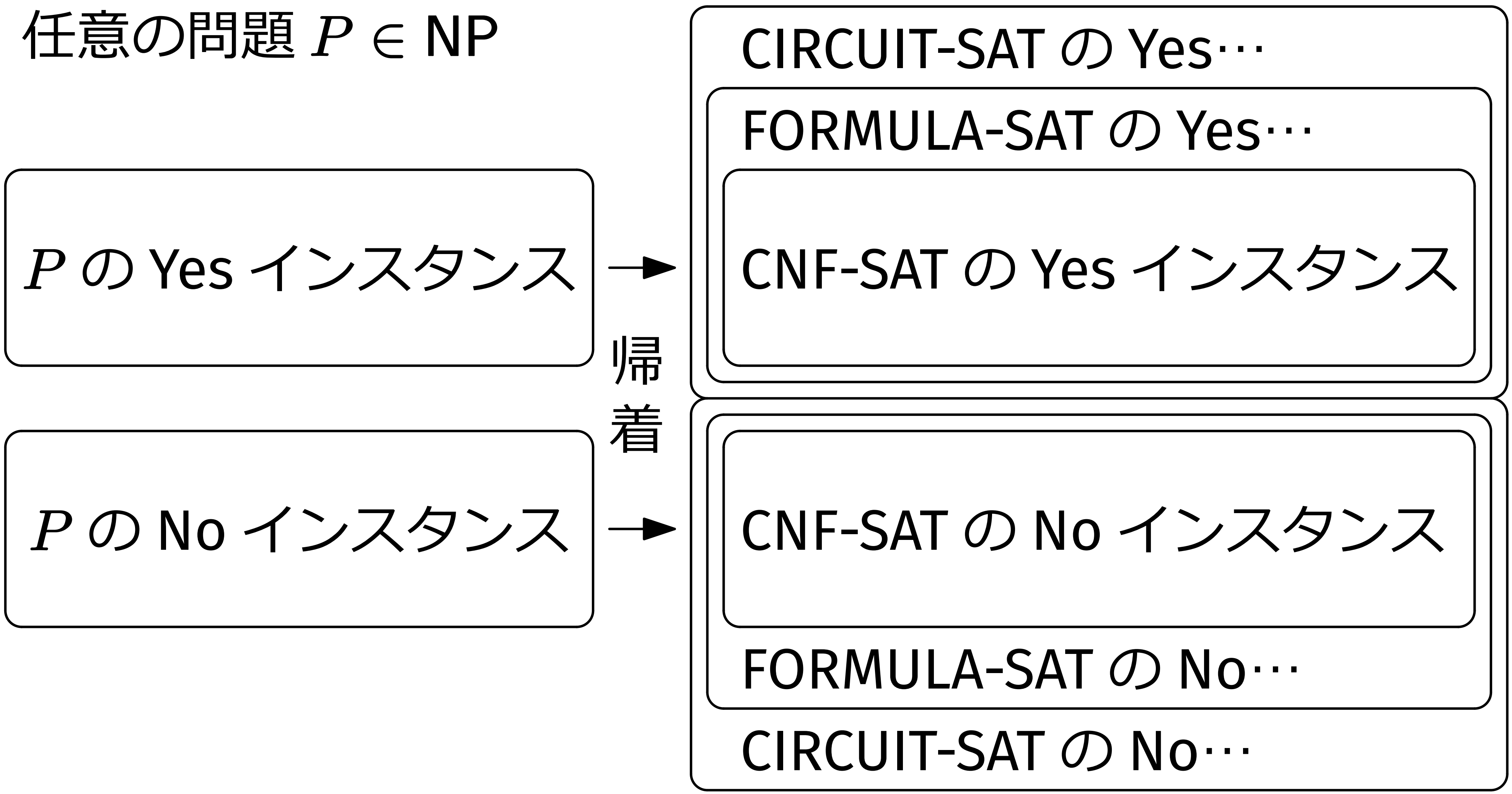
$P$ の No インスタンス → 

CNF-SAT の No インスタンス

FORMULA-SAT の No⋯

# 補足：CNF → FORMULA → CIRCUIT

CNF-SAT が NP 完全であることから，
FORMULA-SAT と CIRCUIT-SAT の NP 完全性がすぐ分かる



# 連言標準形論理式の充足不能性問題



**連言標準形論理式の充足不能性問題 (CNF-UNSAT)**

**入力：** 連言標準形の論理式 $\varphi$

**出力：** $\varphi$ が充足割当を持たない $\Rightarrow$ Yes

$\varphi$ が充足割当を持つ $\Rightarrow$ No

充足不能性問題 = Unsatisfiability Problem

**性質：CNF-UNSAT の coNP 完全性**

CNF-UNSAT は coNP 完全

(Cook ’71; Levin ’73)

CIRCUIT-UNSAT，FORMULA-UNSAT も同様に定義できて，

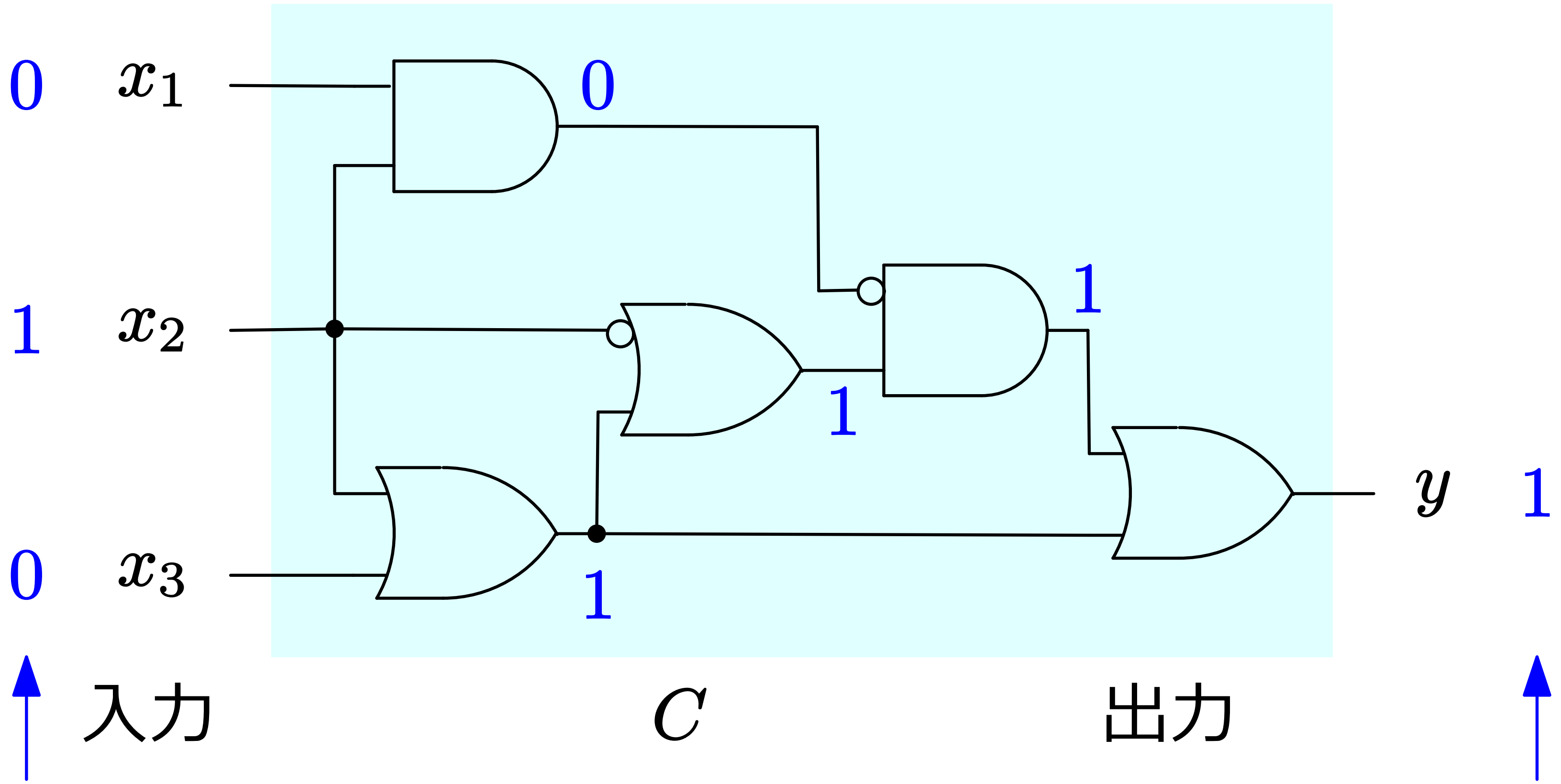
CIRCUIT-UNSAT，FORMULA-UNSAT も coNP 完全

# CIRCUIT-SAT $\in$ NP



**CIRCUIT-SAT が NP 完全だと証明するには**

1. CIRCUIT-SAT $\in$ NP を証明する
2. CIRCUIT-SAT が NP 困難だと証明する



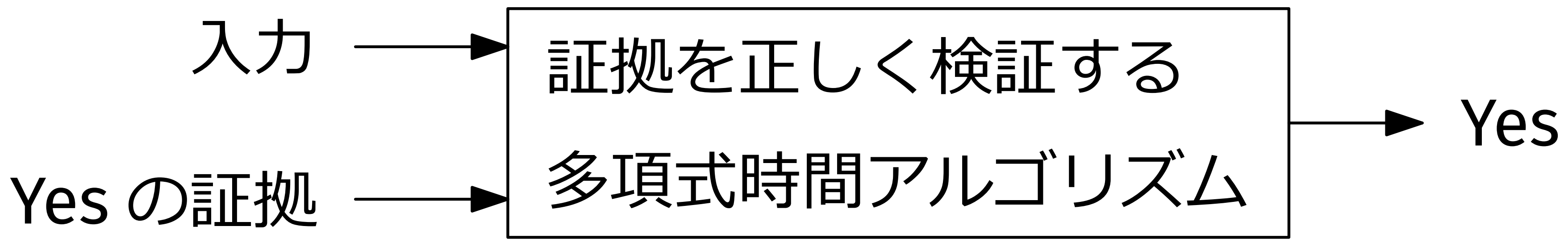

割当を guess する

割当から 論理回路の出力は多項式時間で得られる

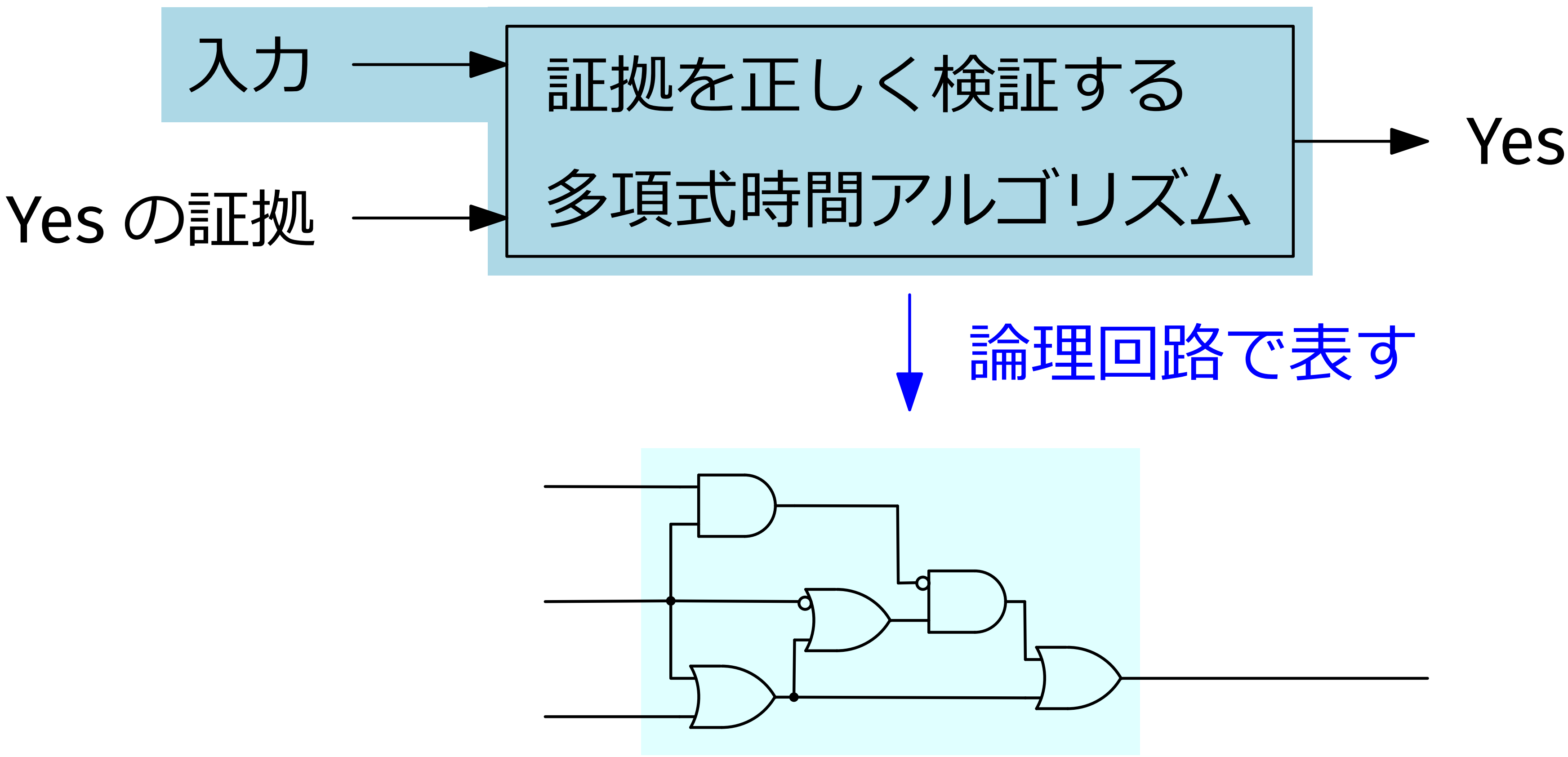
# CIRCUIT-SAT は NP 困難：方針のみ



任意の問題 $P \in \mathrm{NP}$



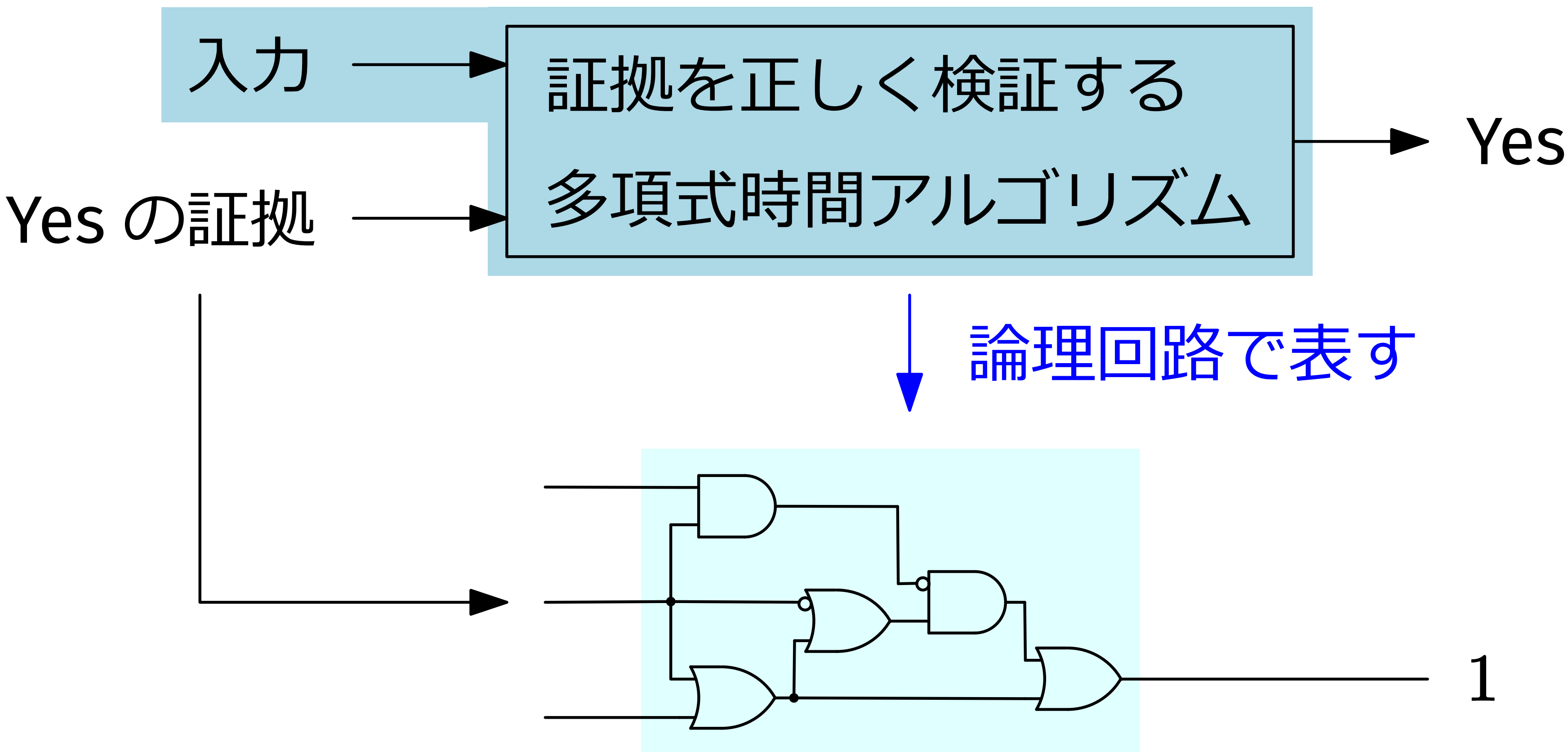
任意の問題 $P \in \mathrm{NP}$



# CIRCUIT-SAT は NP 困難：方針のみ



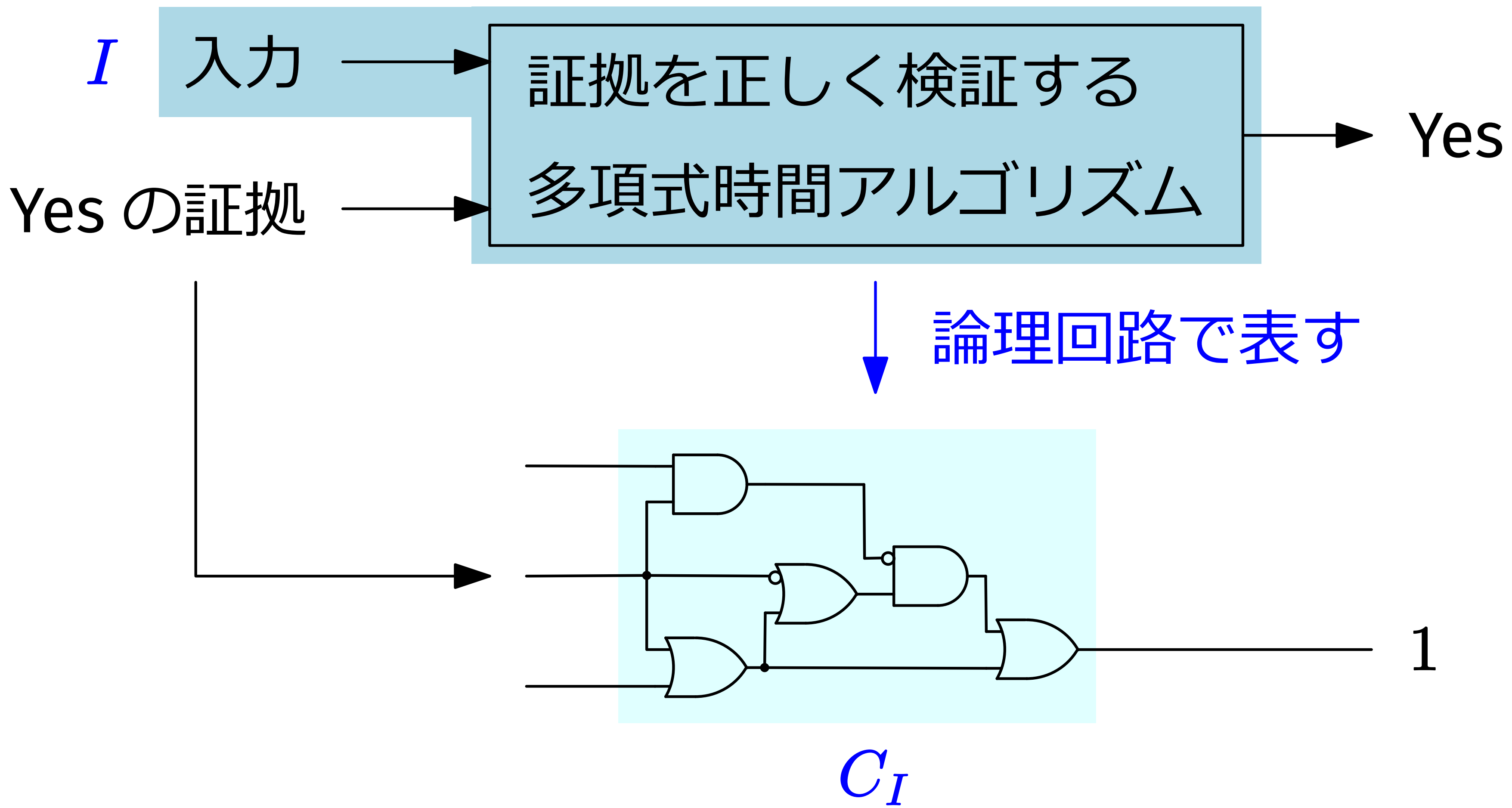
任意の問題 $P \in$ NP

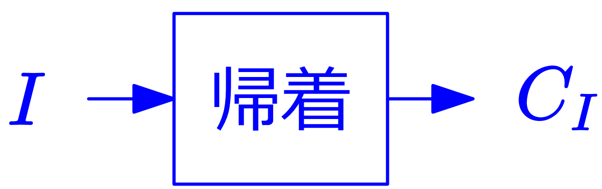


# CIRCUIT-SAT は NP 困難：方針のみ



任意の問題 $P \in \mathrm{NP}$

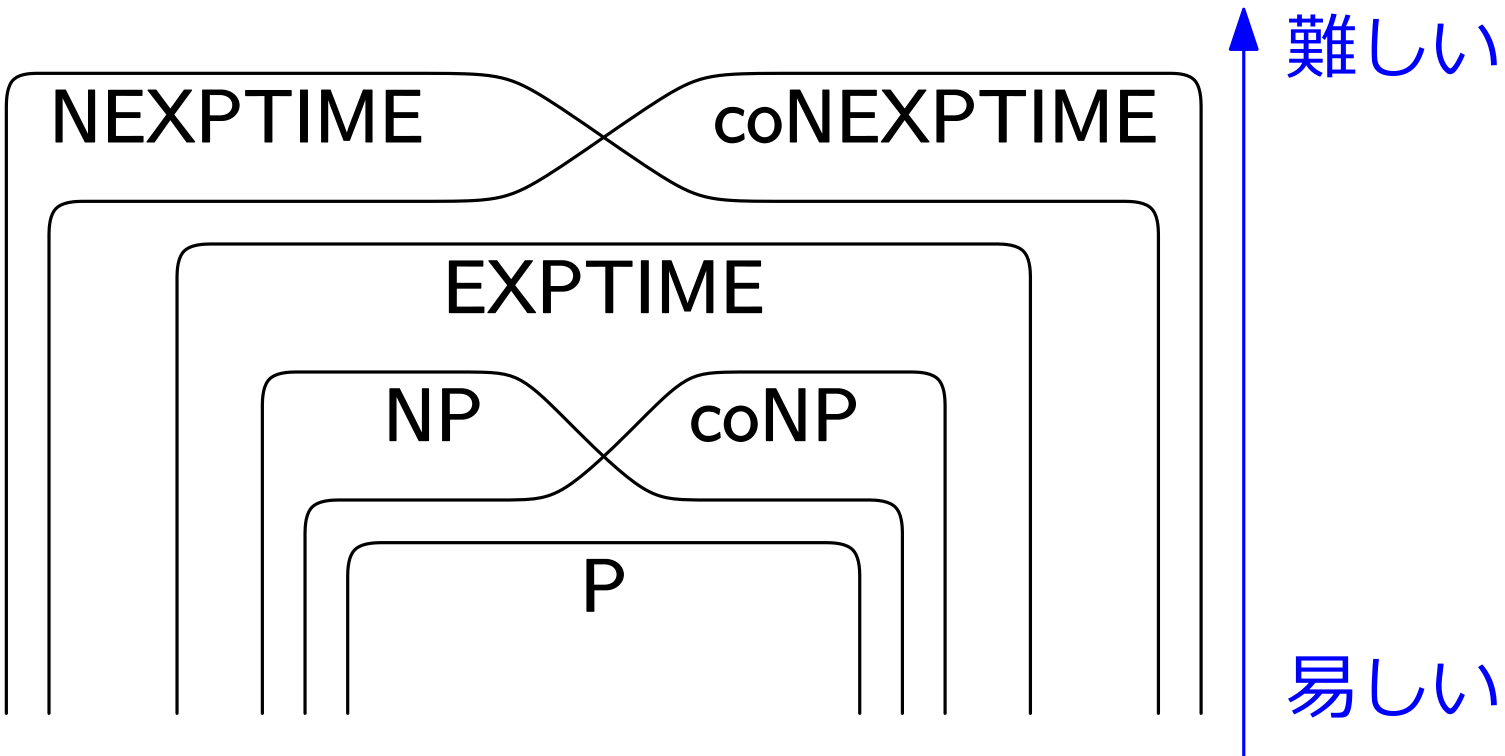


# 本日のまとめ

## 本日の重要概念

- 多対一帰着 (あるいは帰着)，多項式時間帰着
- 困難性，完全性
- 充足可能性問題 (NP 完全)



# 次回の予告



**次回**

- 資源として **領域** (つまり空間) を扱う
- 領域に関する計算複雑性と帰着を定義する

計算複雑性クラス

- PSPACE，NPSPACE，L，NL

帰着と完全性

- 対数領域帰着
- P 完全性，NL 完全性

# クイズ



クイズはここに手書きで伝える

Q

1.

2.

3.

4.