理論計算機科学特論 (2026 年前学期)

計算複雑性の基礎

第 2 回

時間計算量：P, NP, coNP

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2026 年 4 月 14 日

最終更新：2026 年 4 月 15 日　11:53

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：$P \subseteq PSPACE \subseteq EXPTIME$ (5/12)
6. 階層定理：$P \neq EXPTIME$ (5/19)
7. Ladner の定理：$NP - P = NPC \Rightarrow P = NP$ (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：計算複雑性における課題

課題 1

計算における **資源** とは何か？

$\rightsquigarrow$ 時間，領域，非決定性，ランダム性，インタラクション

課題 2

資源どうしに **関係** はあるか？

$\rightsquigarrow$ 計算複雑性クラス

注：この 2 つだけが課題であるわけではない

復習：判定問題とインスタンス

この講義で扱うのは たいてい **判定問題** である

定義 (非形式)：判定問題 (decision problem)

判定問題 とは，次の形で書けるもののこと

入力： I
出力： I が性質 $\mathcal{P}$ を満たす $\Rightarrow$ Yes
I が性質 $\mathcal{P}$ を満たさない $\Rightarrow$ No

「性質 $\mathcal{P}$ を判定する問題」という言い方をすることがある

定義 (非形式)：インスタンス (instance)

判定問題 P の **インスタンス** とは，P の入力 1 つのこと

- **Yes インスタンス** とは，出力が Yes であるインスタンス
- **No インスタンス** とは，出力が No であるインスタンス

復習：アルゴリズム

定義 (非形式)：アルゴリズム (algorithm)

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

- I が性質 $\mathcal{P}$ を満たす　　$\Rightarrow$ Yes を出力
- I が性質 $\mathcal{P}$ を満たさない $\Rightarrow$ No を出力

定義 (非形式)：判定問題 (decision problem)

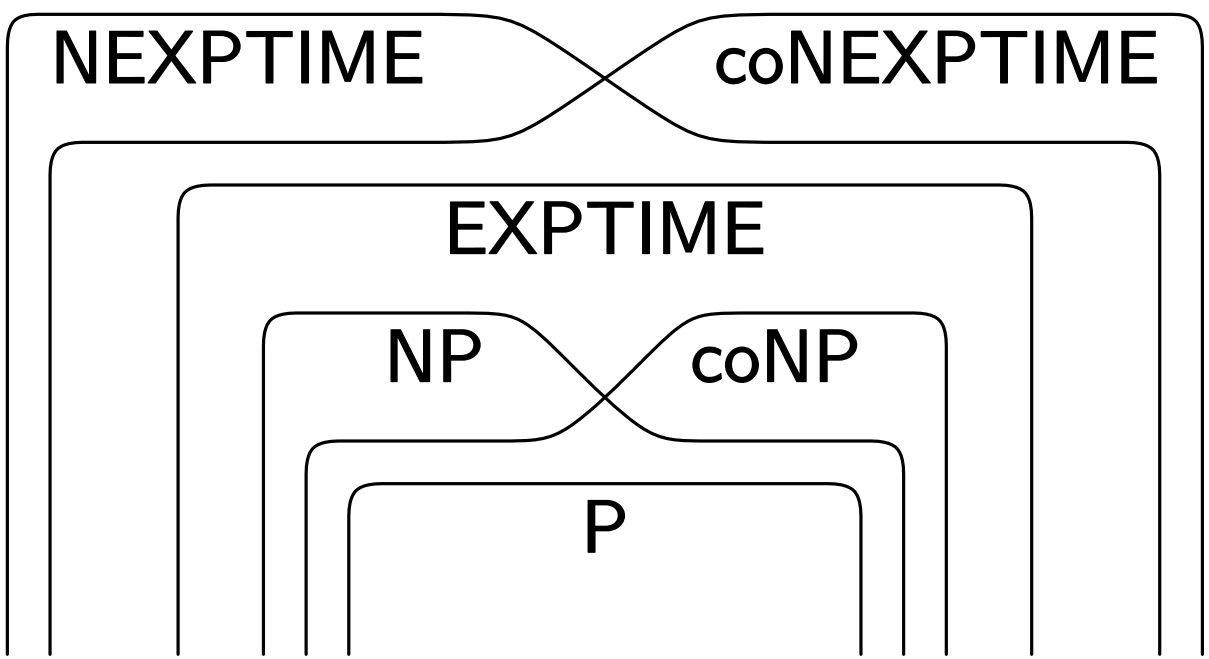
判定問題 とは，次の形で書けるもののこと

入力： I
出力： I が性質 $\mathcal{P}$ を満たす　　$\Rightarrow$ Yes
　　I が性質 $\mathcal{P}$ を満たさない $\Rightarrow$ No

本日の目標

本日の目標

- **時間** を資源としてみることができるようになる
- **非決定性** を持つ計算モデルを扱えるようになる
- 時間に関する **計算複雑性クラス** の間の包含関係を正しく導出できる



本日の内容

1. **時間計算量：P, EXPTIME**
2. 非決定性の計算モデル
3. 非決定性モデルの時間計算量：NP, NEXPTIME
4. 補クラス：coNP, coNEXPTIME

アルゴリズムの時間計算量

P は判定問題，A は P を解くアルゴリズム

定義：アルゴリズムの時間計算量

アルゴリズム A の **時間計算量** (time complexity) とは，
符号長 n 以下の入力 I に対して A が行う単位操作数の
I に関する最大値

$$t_A(n) = \max_{I:\ |I| \leq n} (A(I) \text{ にかかる単位操作数})$$

注：これを A の **最悪時間計算量** ということがある

普通のアルゴリズムの授業で扱う「計算量」と同じ

アルゴリズムの時間計算量

P は判定問題，A は P を解くアルゴリズム

定義：アルゴリズムの時間計算量

アルゴリズム A の **時間計算量** (time complexity) とは，
符号長 n 以下の入力 I に対して A が行う単位操作数の
I に関する最大値

$$t_A(n) = \max_{I:\ |I| \leq n} (A(I) \text{ にかかる単位操作数})$$

注：これを A の **最悪時間計算量** ということがある

普通のアルゴリズムの授業で扱う「計算量」と同じ

ポイント

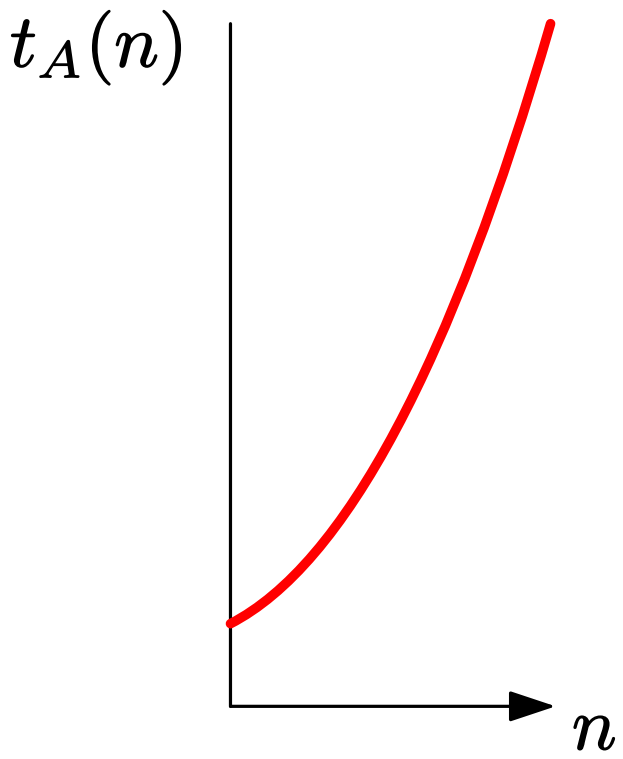
時間計算量は n の関数

アルゴリズムの時間計算量：イメージ図

入力：正整数 k

(符号長 $n = O(\log k)$)

```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<k) {
4:    s=s+i;
5: }
6: return s;
```



定義：アルゴリズムの時間計算量

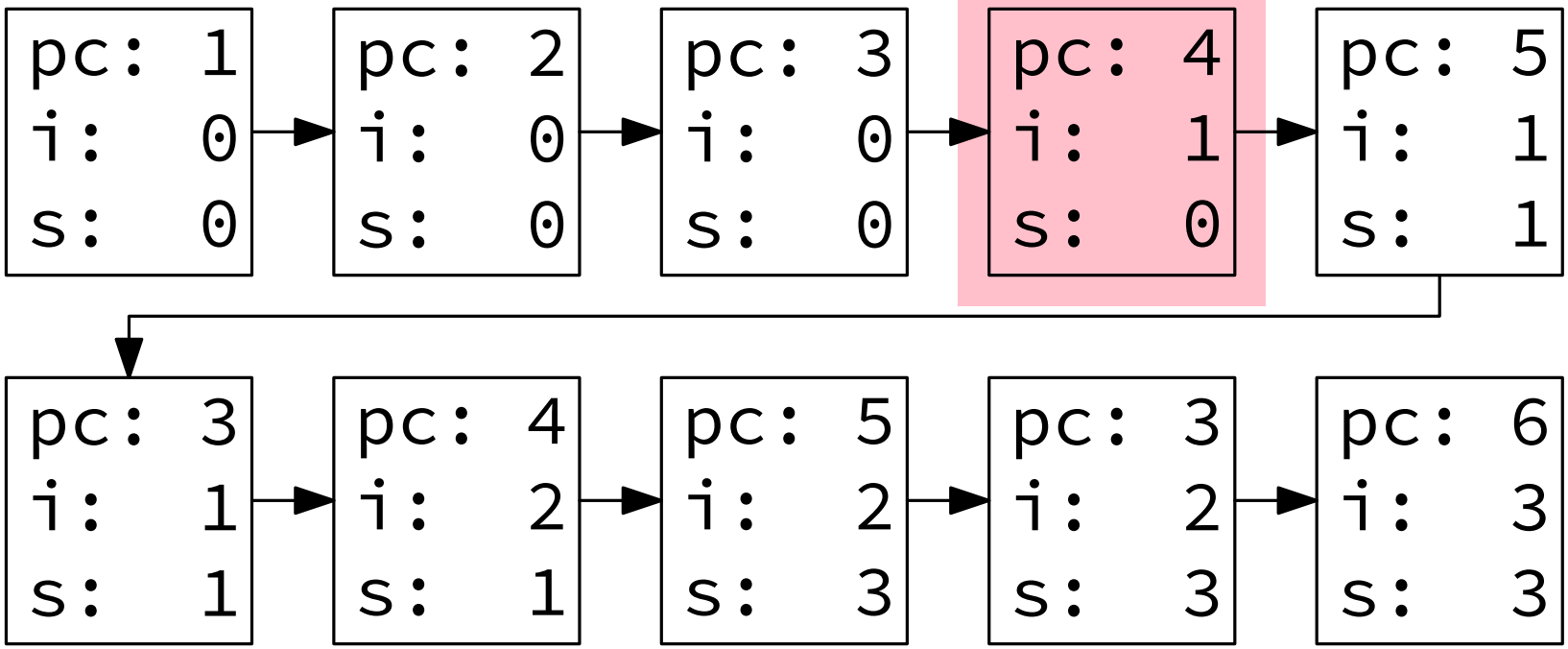
アルゴリズム A の **時間計算量** (time complexity) とは，
符号長 n 以下の入力 I に対して A が行うステップ総数の
I に関する最大値

$$t_A(n) = \max_{I:\ |I| \le n} (A(I) \text{ にかかるステップ数})$$

アルゴリズムの実行と時点状況：例

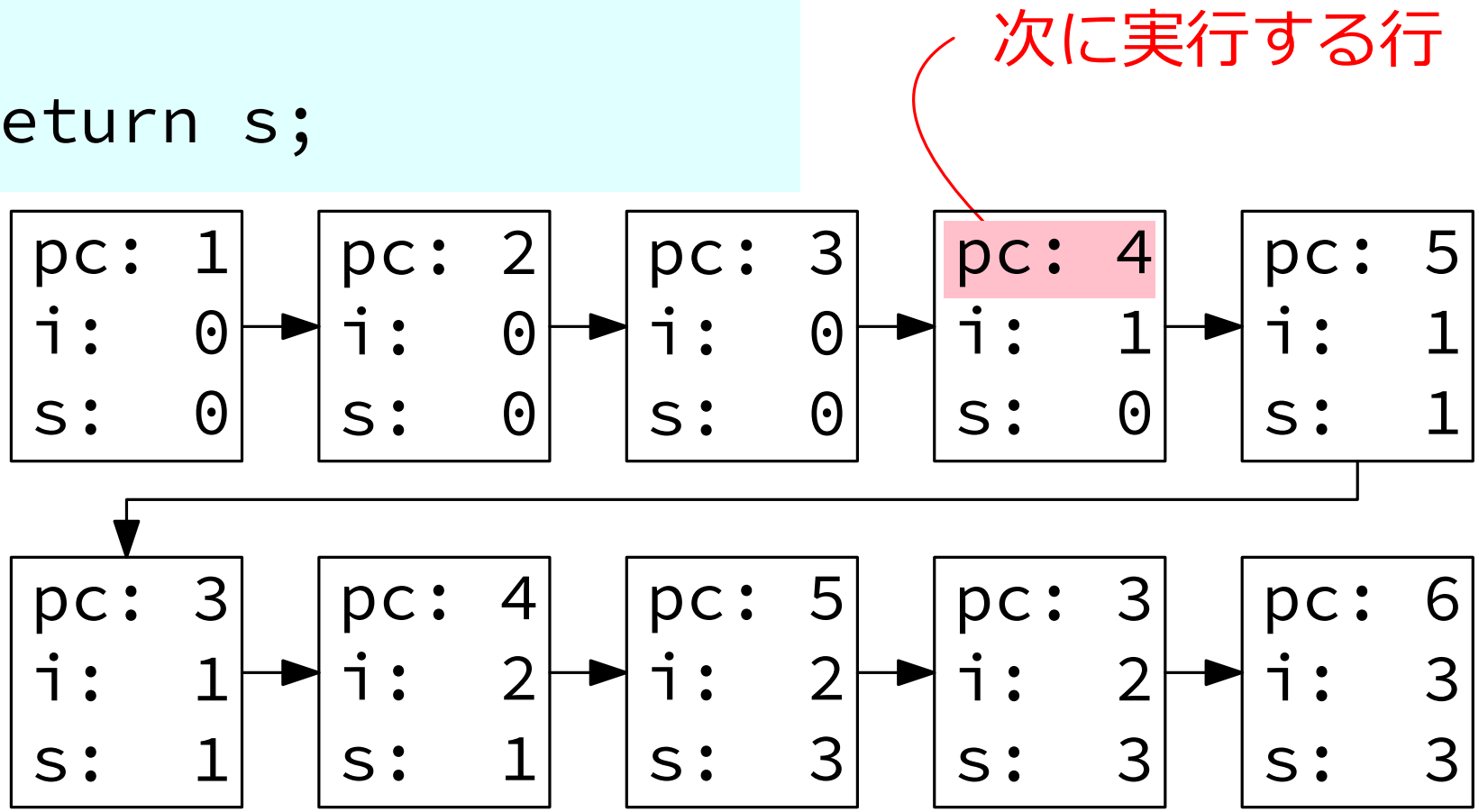

```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```

時点状況 (configuration)



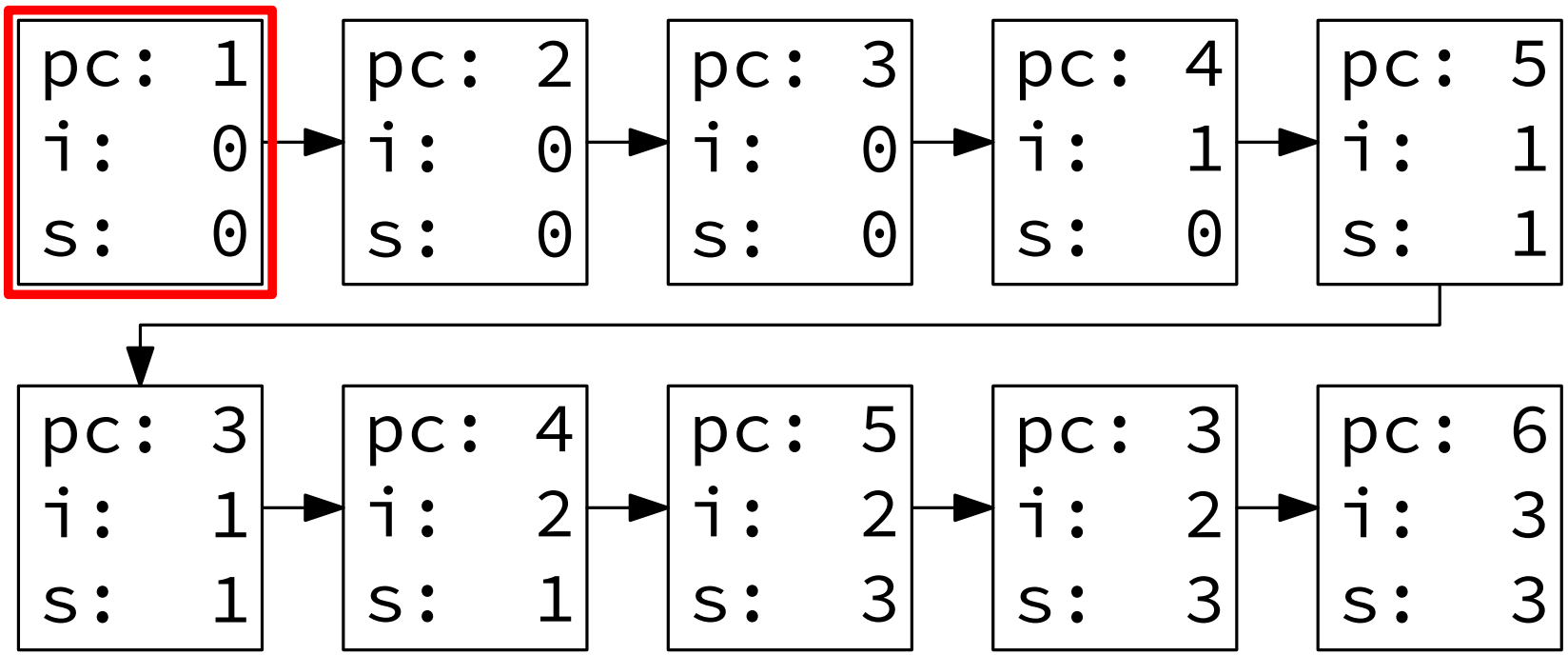
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```



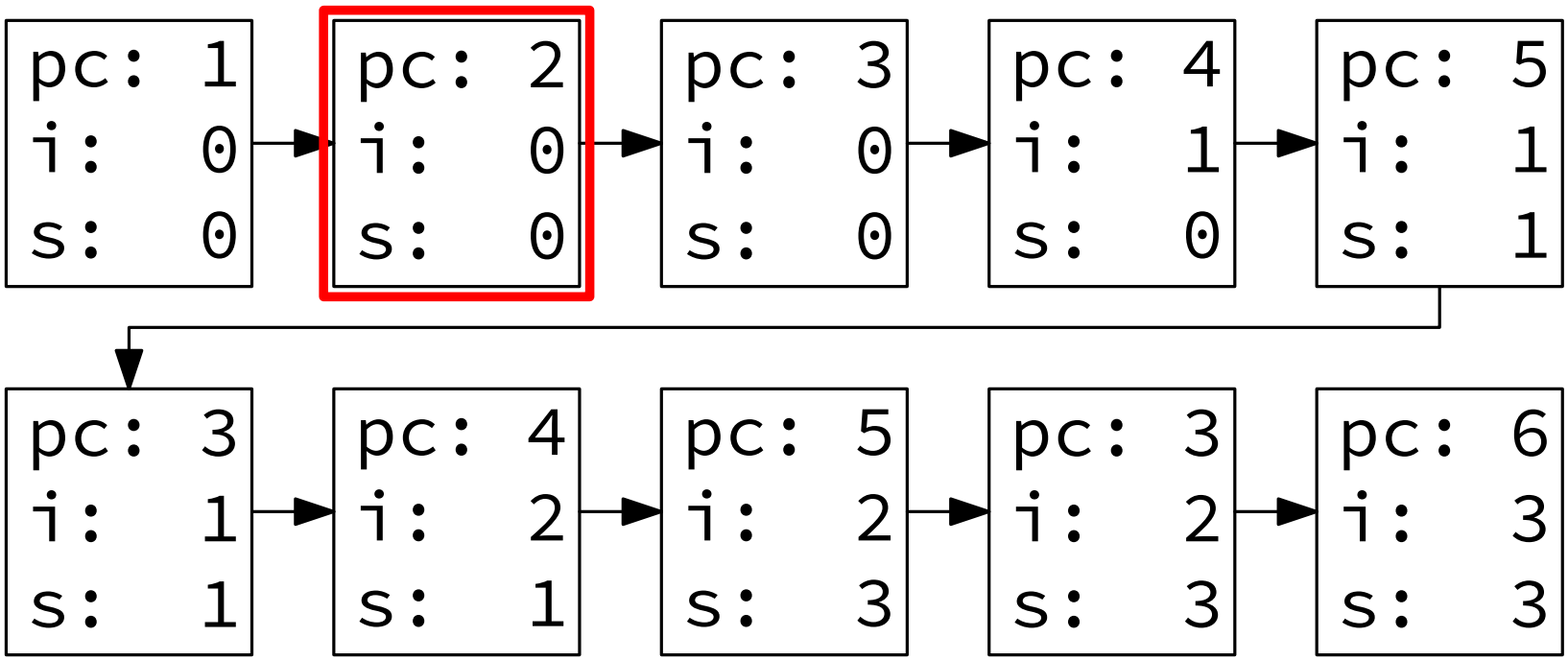
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```



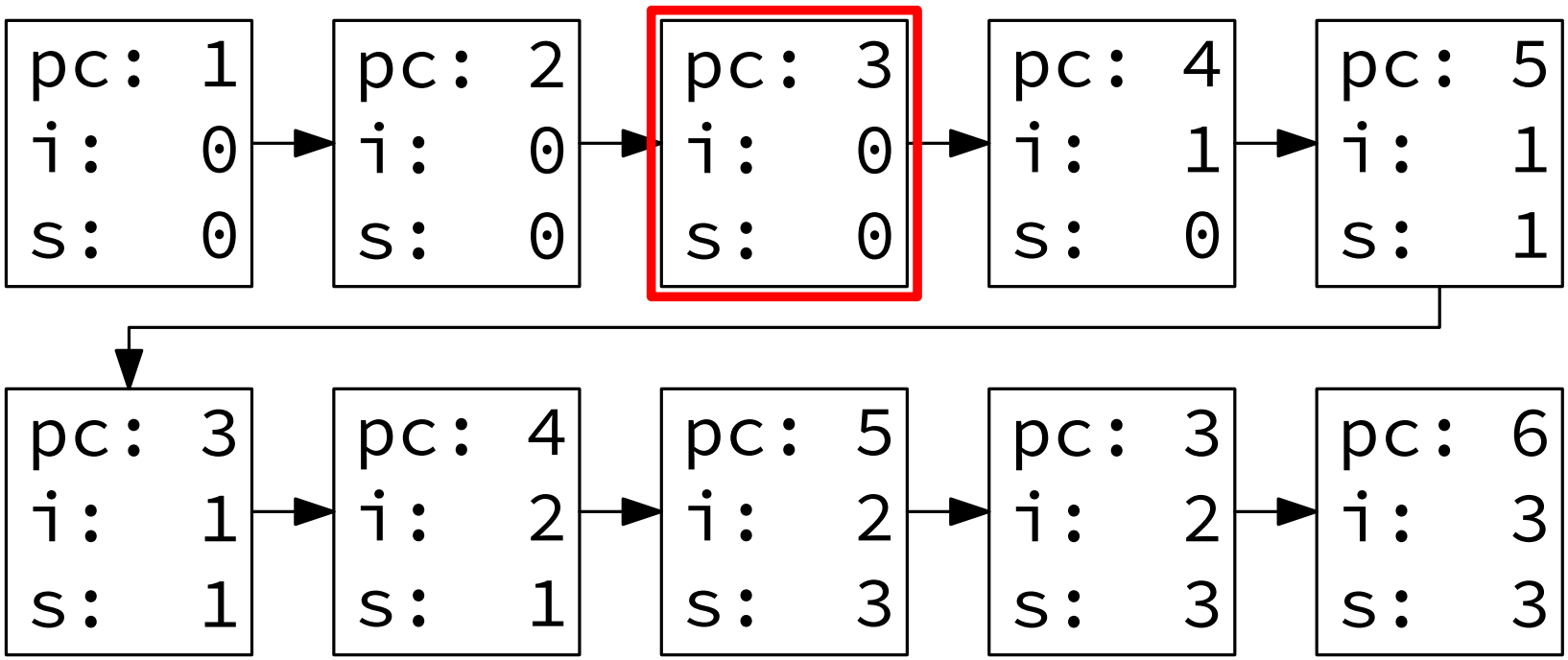
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```



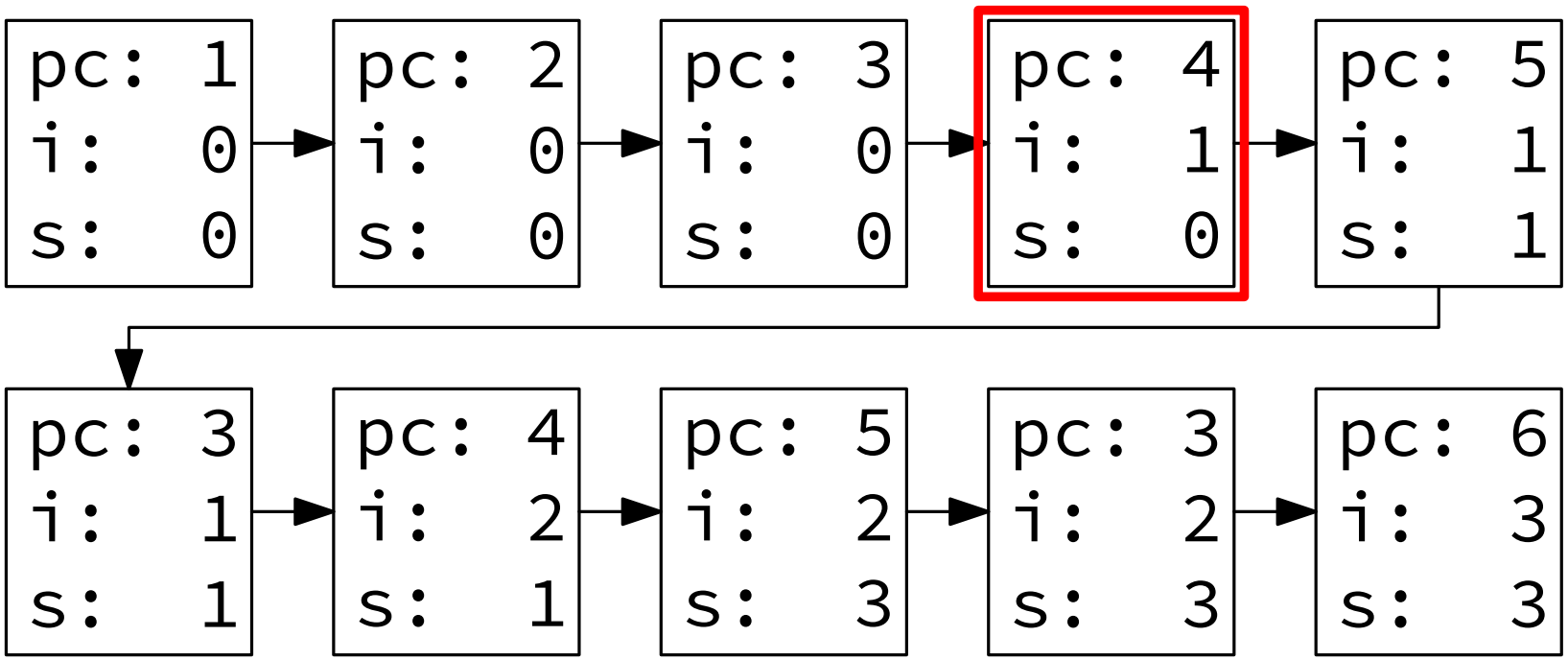
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```



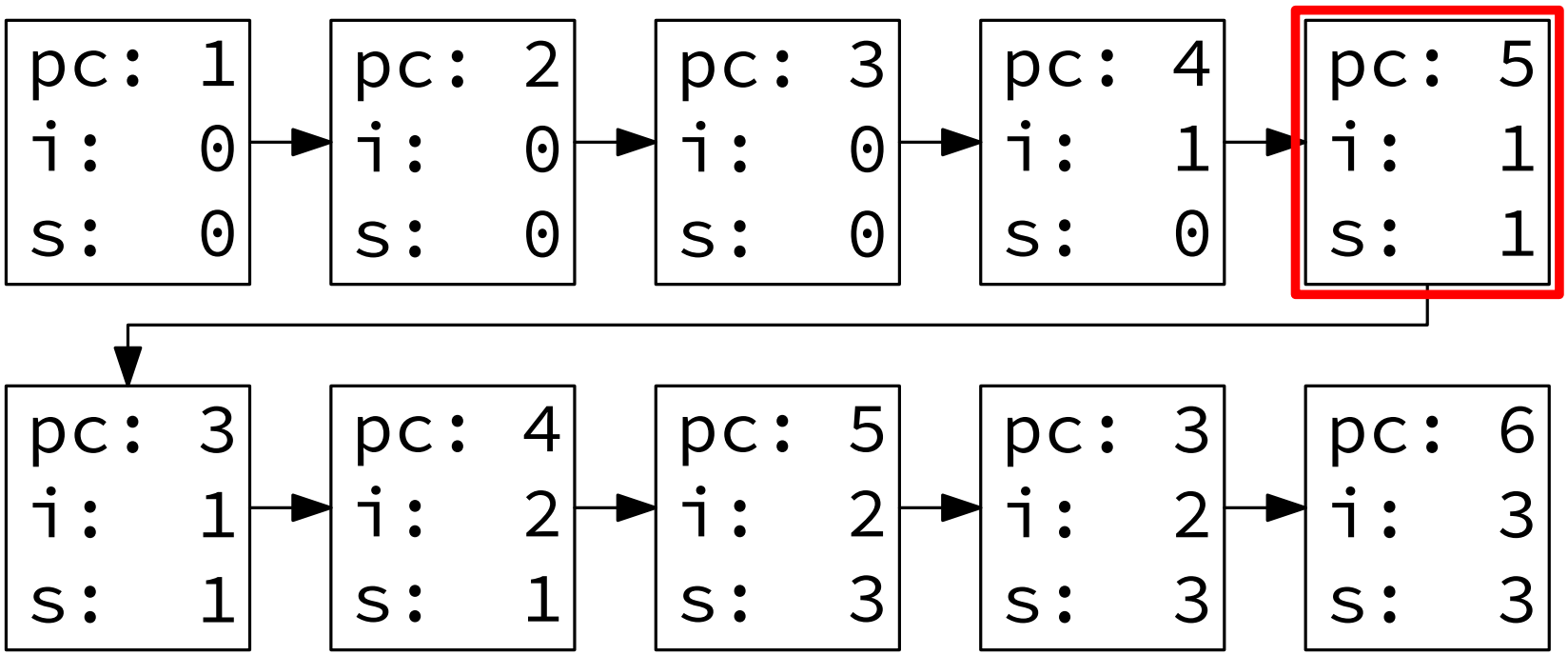
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```



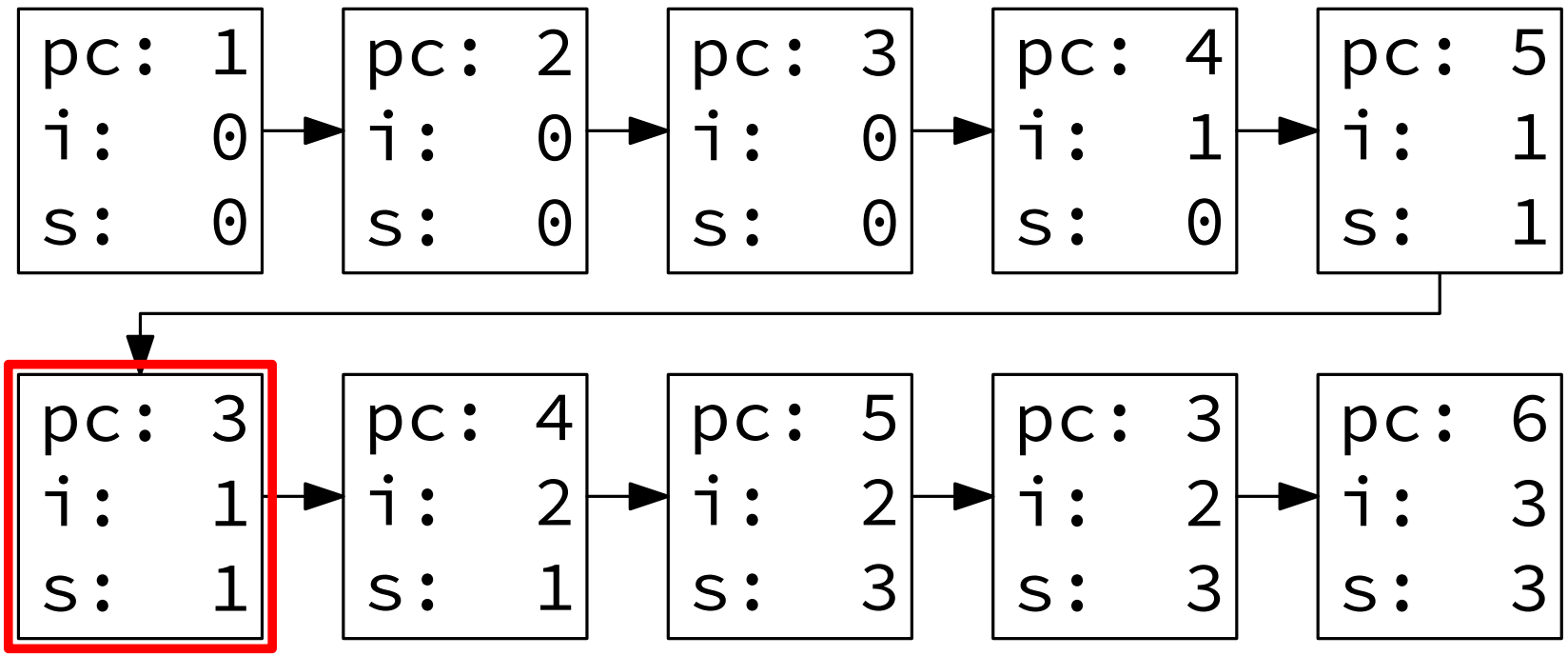
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```



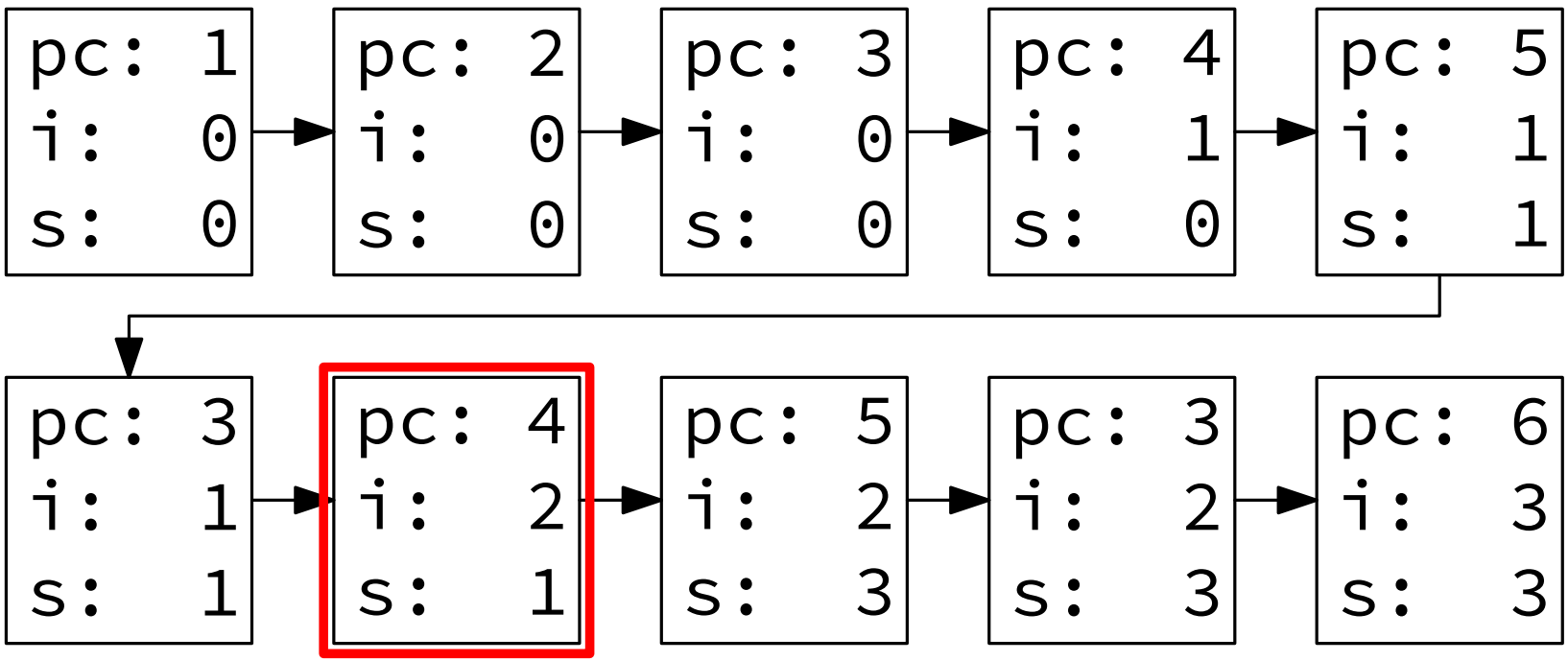
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```



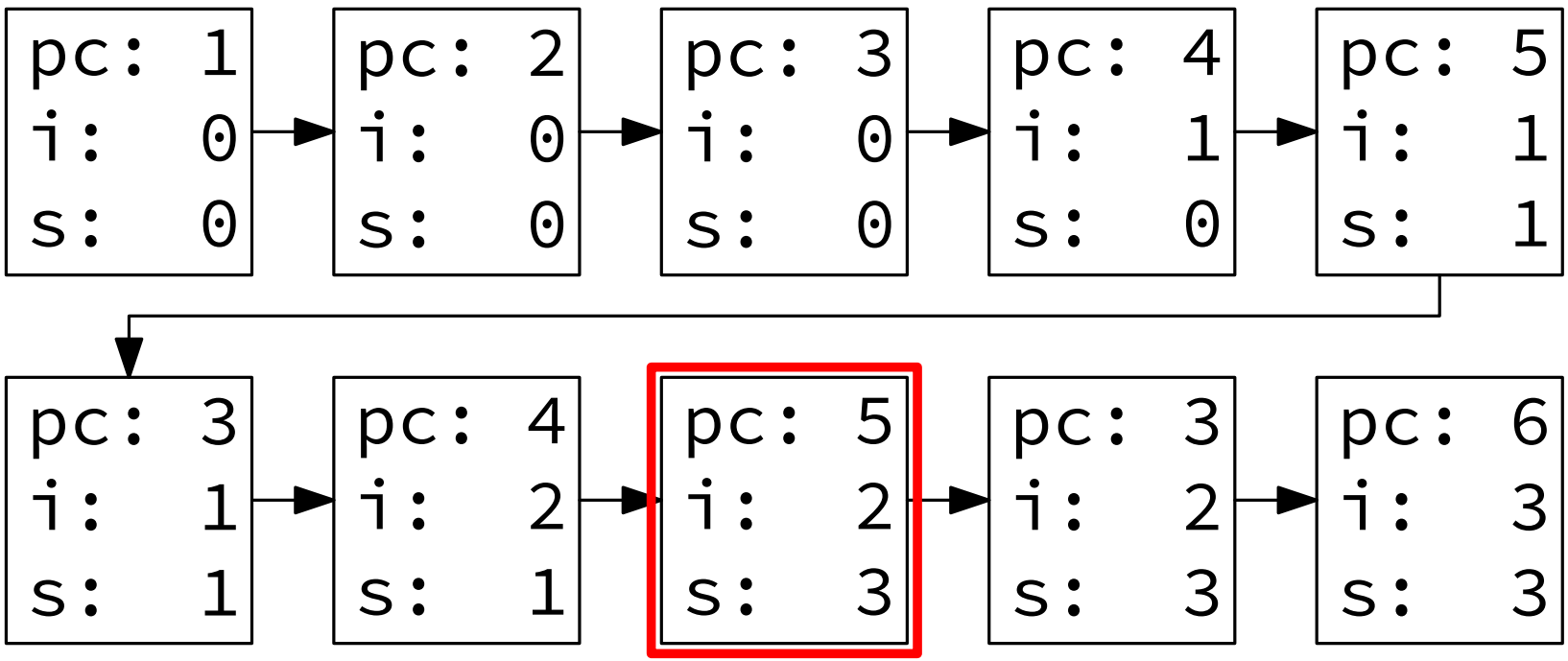
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```



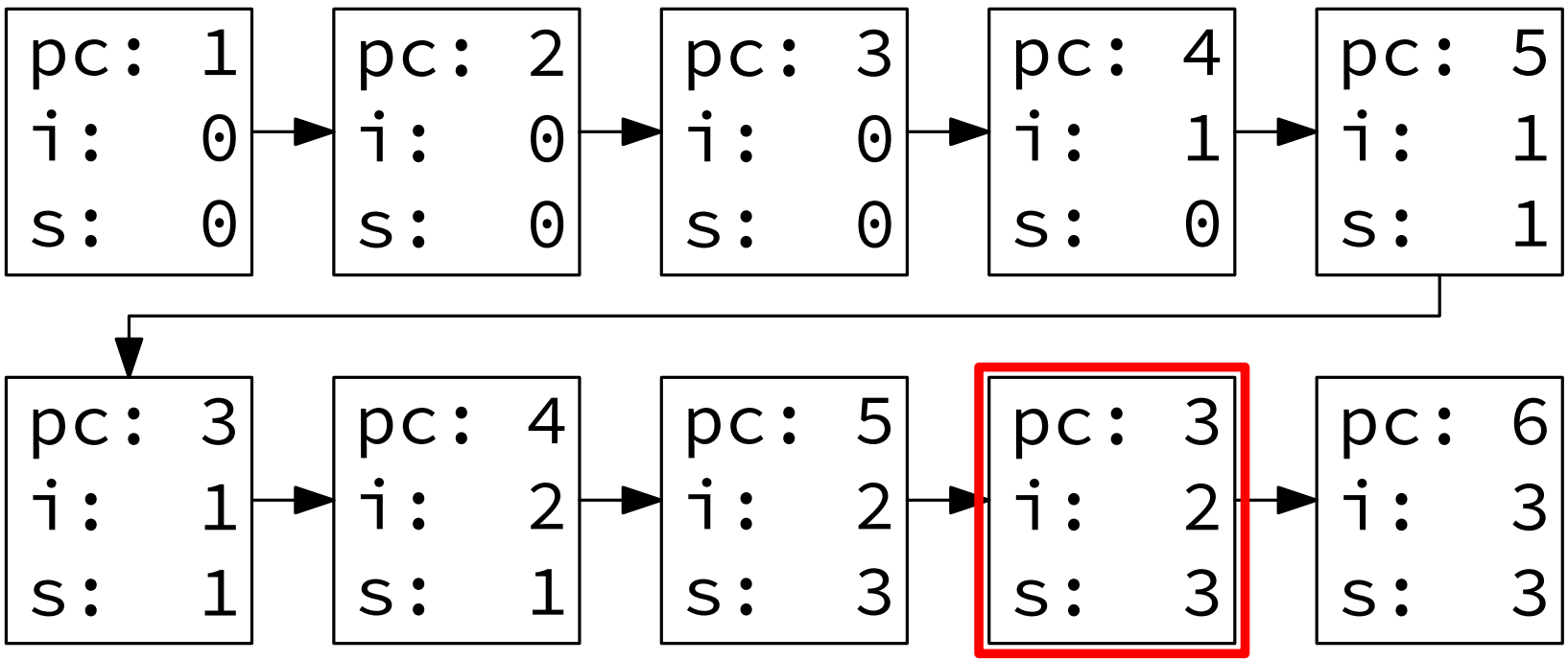
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```



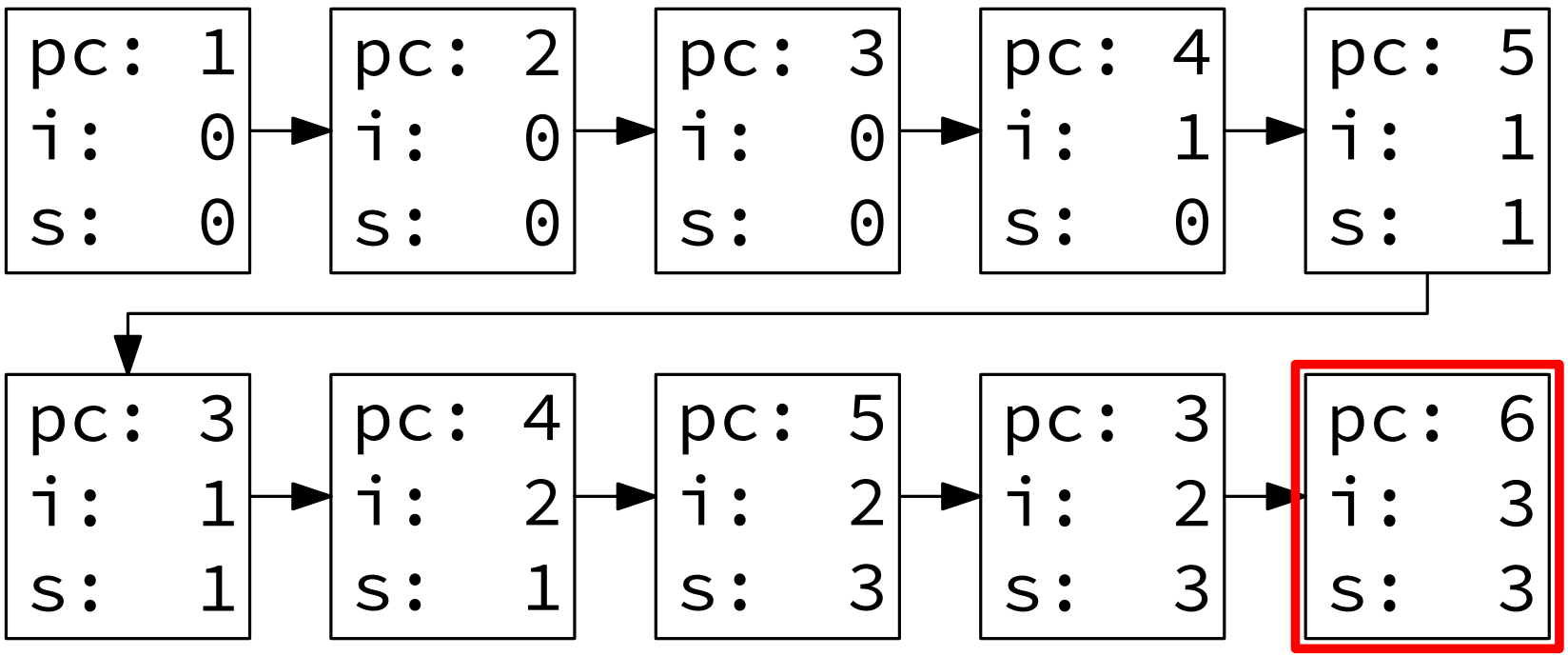
アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```

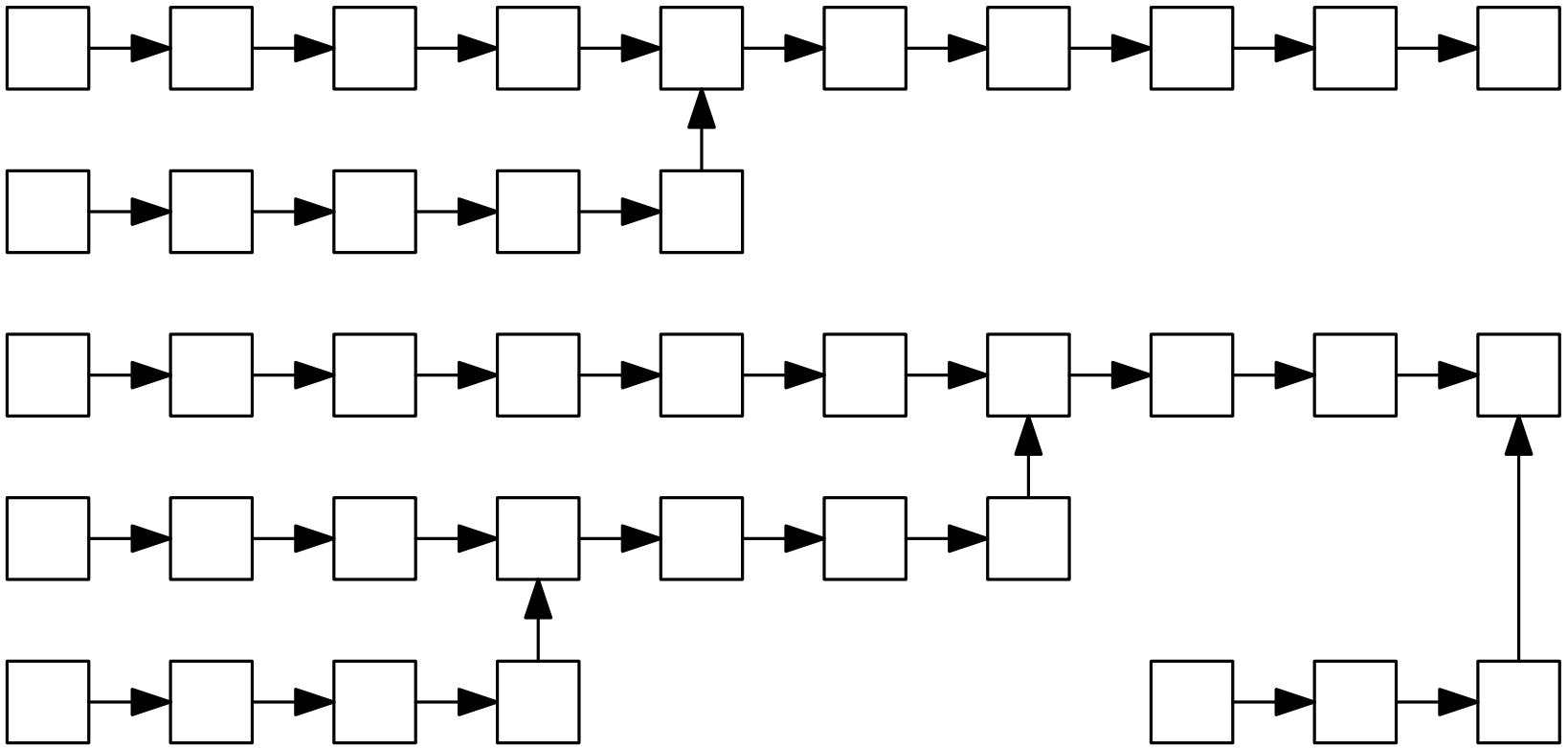


アルゴリズムの実行と時点状況：例


```
1: int i=0;
2: int s=0;
3: for(i=1; i++; i<3) {
4:    s=s+i;
5: }
6: return s;
```

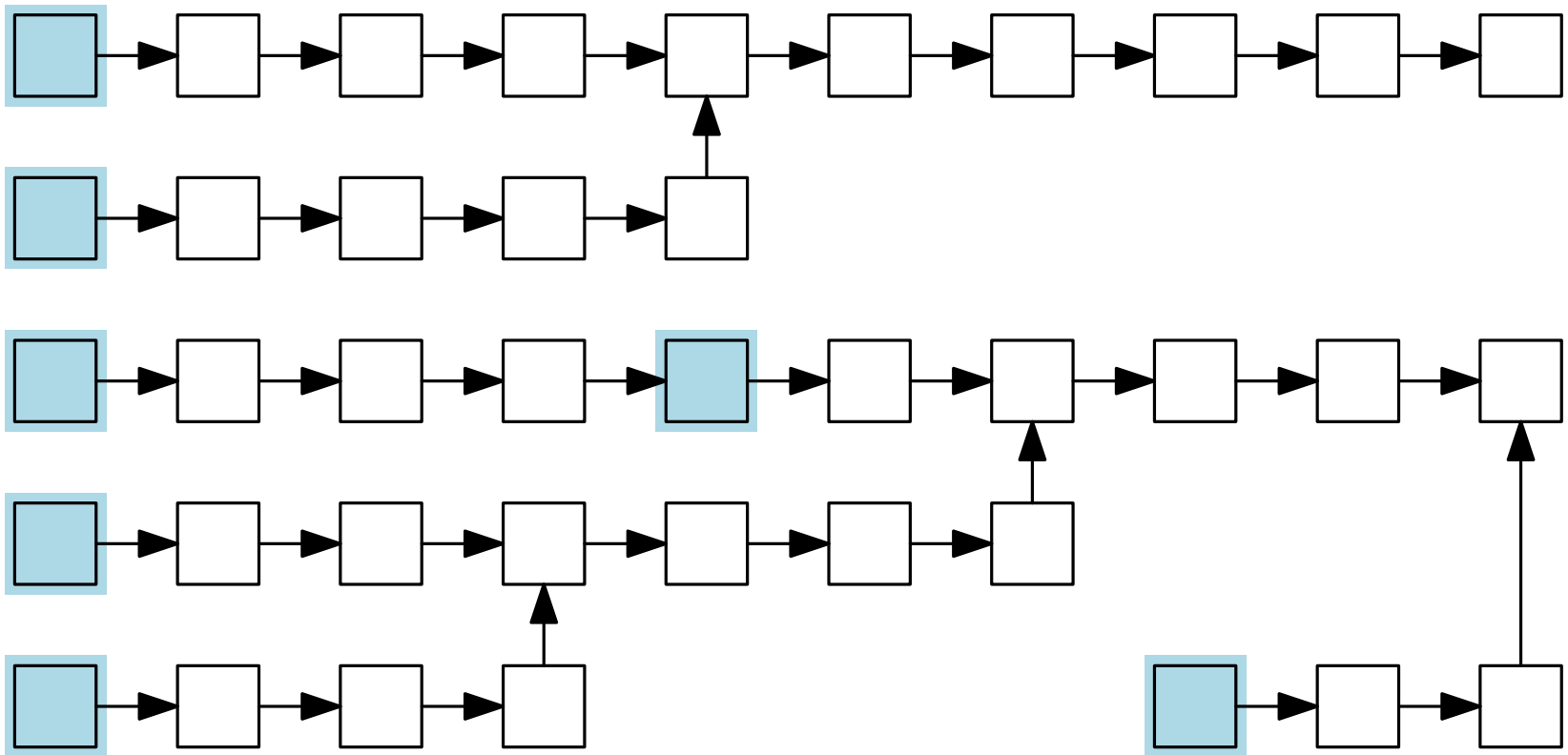


アルゴリズムの実行と時点状況



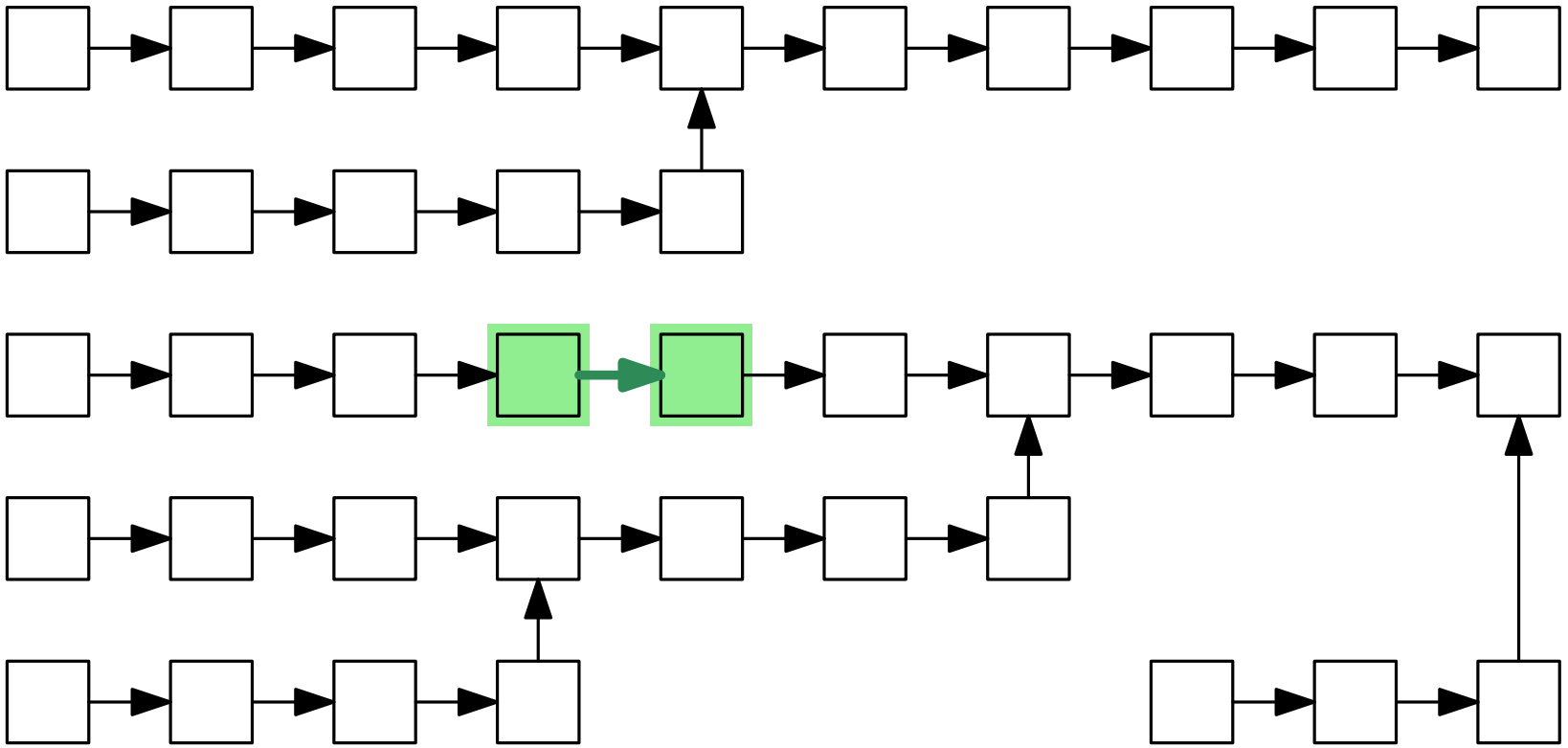
- アルゴリズムからこの図が一意に決まる (無限かもしれない)
- 入力によって **開始状況** が異なる
- 各状況に対して次の状況があれば一意に決まる
- 次の状況がないところは **停止状況** である

アルゴリズムの実行と時点状況



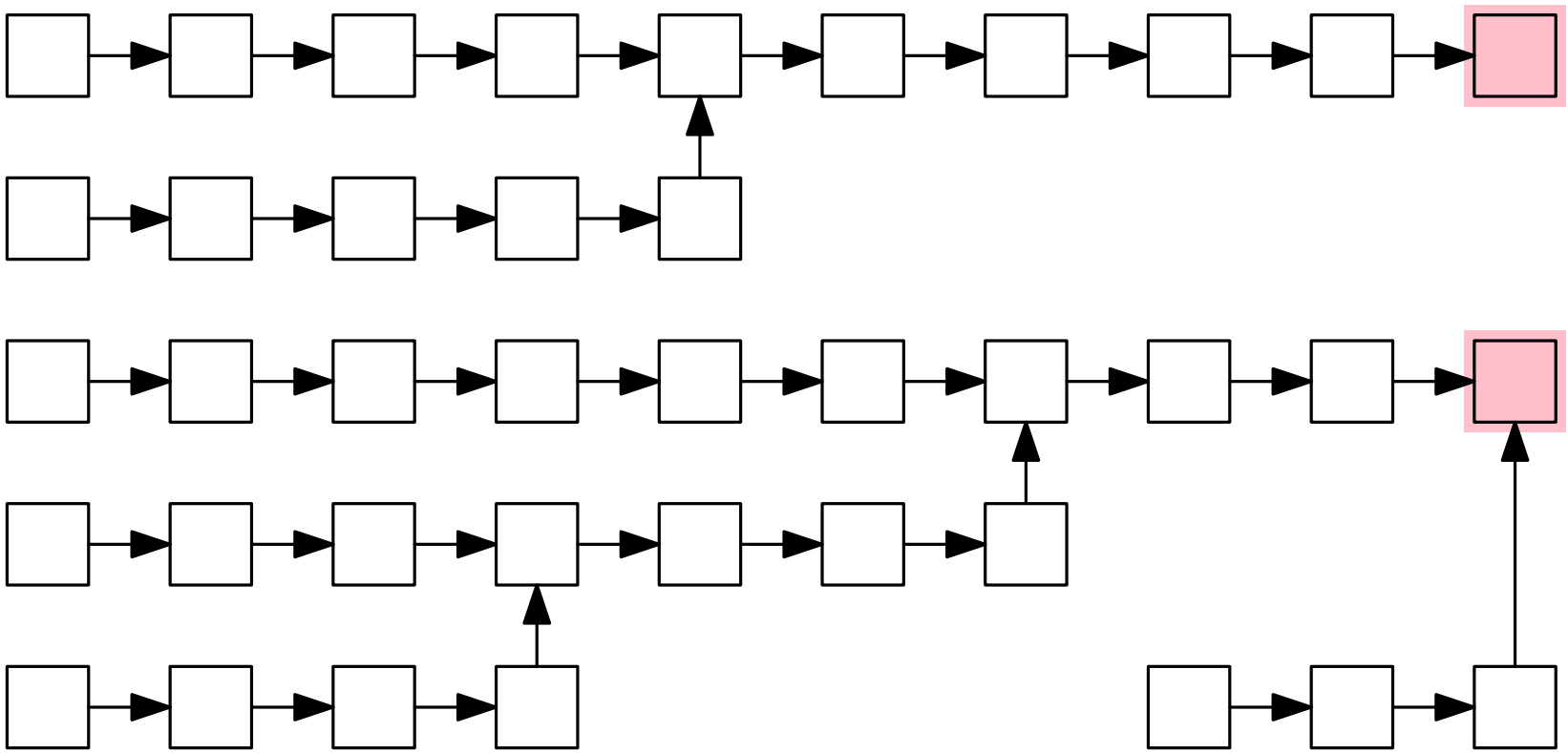
- アルゴリズムからこの図が一意に決まる (無限かもしれない)
- 入力によって **開始状況** が異なる
- 各状況に対して次の状況があれば一意に決まる
- 次の状況がないところは **停止状況** である

アルゴリズムの実行と時点状況



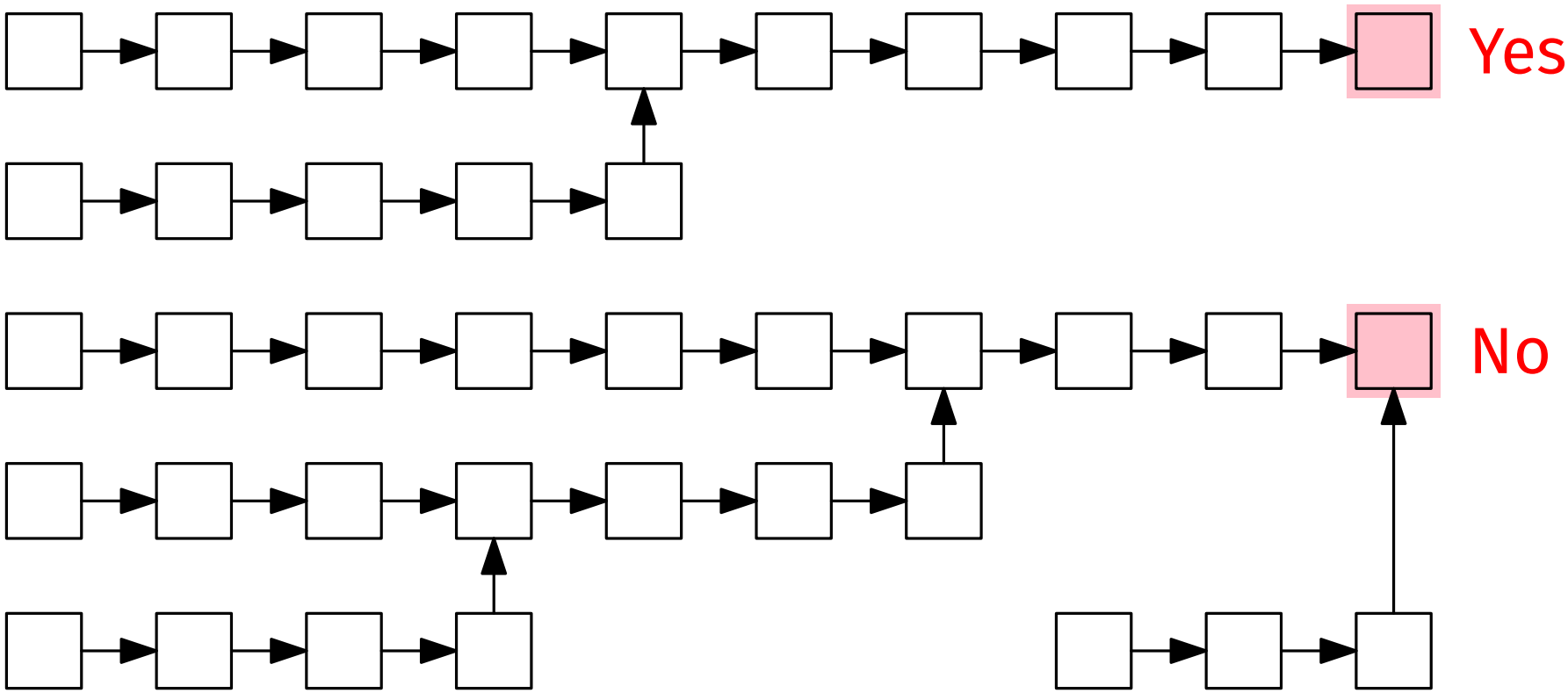
- アルゴリズムからこの図が一意に決まる (無限かもしれない)
- 入力によって **開始状況** が異なる
- 各状況に対して次の状況があれば一意に決まる
- 次の状況がないところは **停止状況** である

アルゴリズムの実行と時点状況



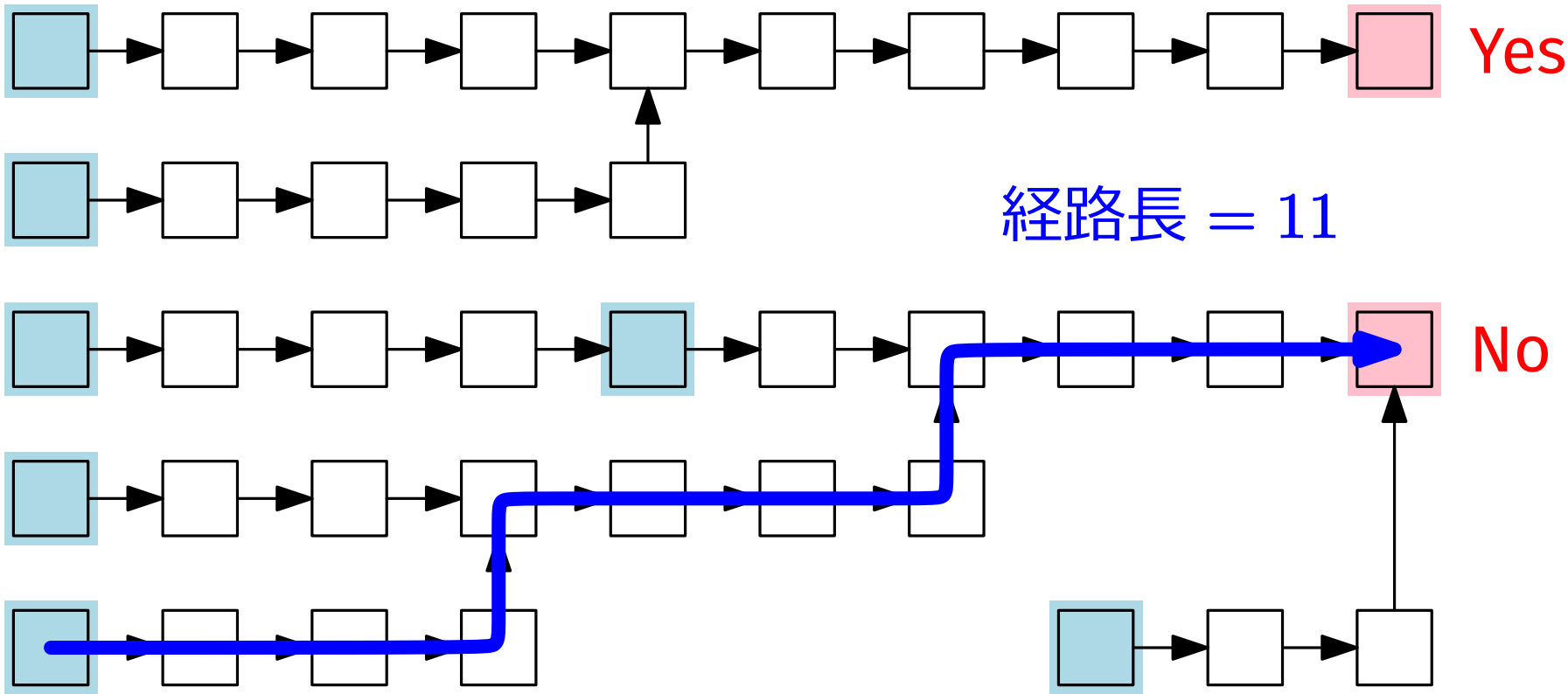
- アルゴリズムからこの図が一意に決まる (無限かもしれない)
- 入力によって **開始状況** が異なる
- 各状況に対して次の状況があれば一意に決まる
- 次の状況がないところは **停止状況** である

アルゴリズムの実行と時点状況



- アルゴリズムからこの図が一意に決まる (無限かもしれない)
- 入力によって **開始状況** が異なる
- 各状況に対して次の状況があれば一意に決まる
- 次の状況がないところは **停止状況** である

アルゴリズムの実行と時点状況



- アルゴリズムからこの図が一意に決まる (無限かもしれない)
- 入力によって **開始状況** が異なる
- 各状況に対して次の状況があれば一意に決まる
- 次の状況がないところは **停止状況** である

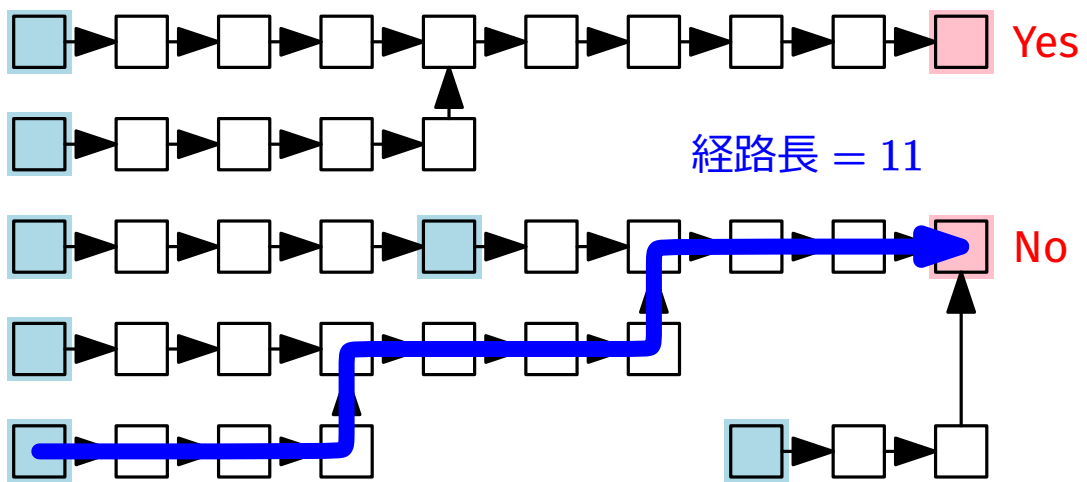
アルゴリズムの時間計算量：別の視点

P は判定問題，A は P を解くアルゴリズム

同値な定義：アルゴリズムの時間計算量

アルゴリズム A の **時間計算量** (time complexity) とは，
符号長 n 以下の入力 I に対して，
I に対応する開始時点から停止時点に至る経路長の
I に関する最大値

仮定：アルゴリズムの各行は定数個のステップしか含まない



多項式時間と指数時間

P は判定問題，A は P を解くアルゴリズム

定義：多項式時間アルゴリズム

A が **多項式時間アルゴリズム** であるとは，
A の時間計算量が 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，$t_A(n) = O(n^k)$

例：$O(n)$，$O(n^2)$，$O(n^{100})$，$O(n \log n)$, ...

多項式時間と指数時間

P は判定問題，A は P を解くアルゴリズム

定義：多項式時間アルゴリズム

A が **多項式時間アルゴリズム** であるとは，
A の時間計算量が 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，$t_A(n) = O(n^k)$

例：$O(n)$，$O(n^2)$，$O(n^{100})$，$O(n \log n)$, ...

定義：指数時間アルゴリズム

A が **指数時間アルゴリズム** であるとは，
A の時間計算量が 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，$t_A(n) = O(2^{n^k})$

例：$O(2^n)$，$O(2^{n^3})$，$O(3^n)$，$O(2^{\sqrt{n}})$，$O(n^2)$, ...

注意：指数時間

注意

分野や文脈によって「指数時間」の定義が異なる

	この授業	他の文脈 1	他の文脈 2	他の文脈 3
$O(n^2)$	○	×	○	×
$O(2^{\sqrt{n}})$	○	○	○	×
$O(2^n)$	○	○	○	○
$O(2^{n^2})$	○	○	×	×
$O(2^{2^n})$	×	×	×	×

クラス P とクラス EXPTIME

定義：クラス P

クラス P とは，
多項式時間アルゴリズムで解ける判定問題全体のこと

クラス P を PTIME と書くこともある

P は「polynomial (多項式)」の意味

定義：クラス EXPTIME

クラス EXPTIME とは，
指数時間アルゴリズムで解ける判定問題全体のこと

(読み方：エクスプタイム，エックスピータイム，イーエックスピータイム)

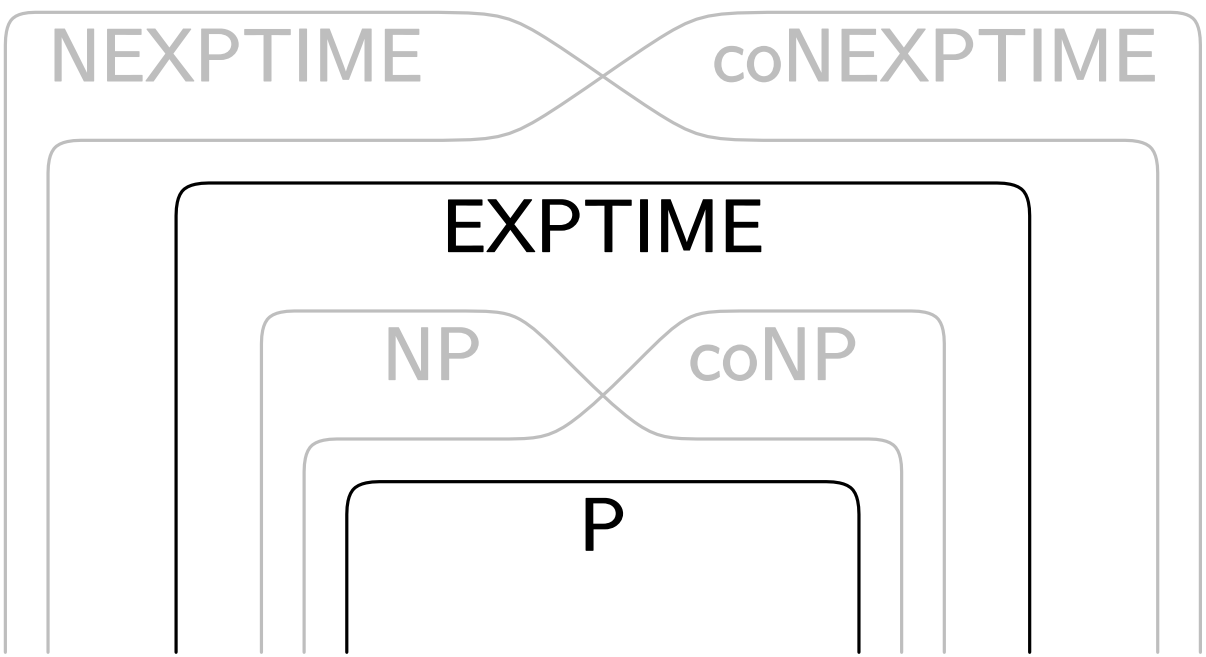
クラス EXPTIME を EXP と書くこともある

EXP は「exponential (指数的)」の意味

クラス間の関係 (1)

性質：定義から次が分かる

$P \subseteq EXPTIME$



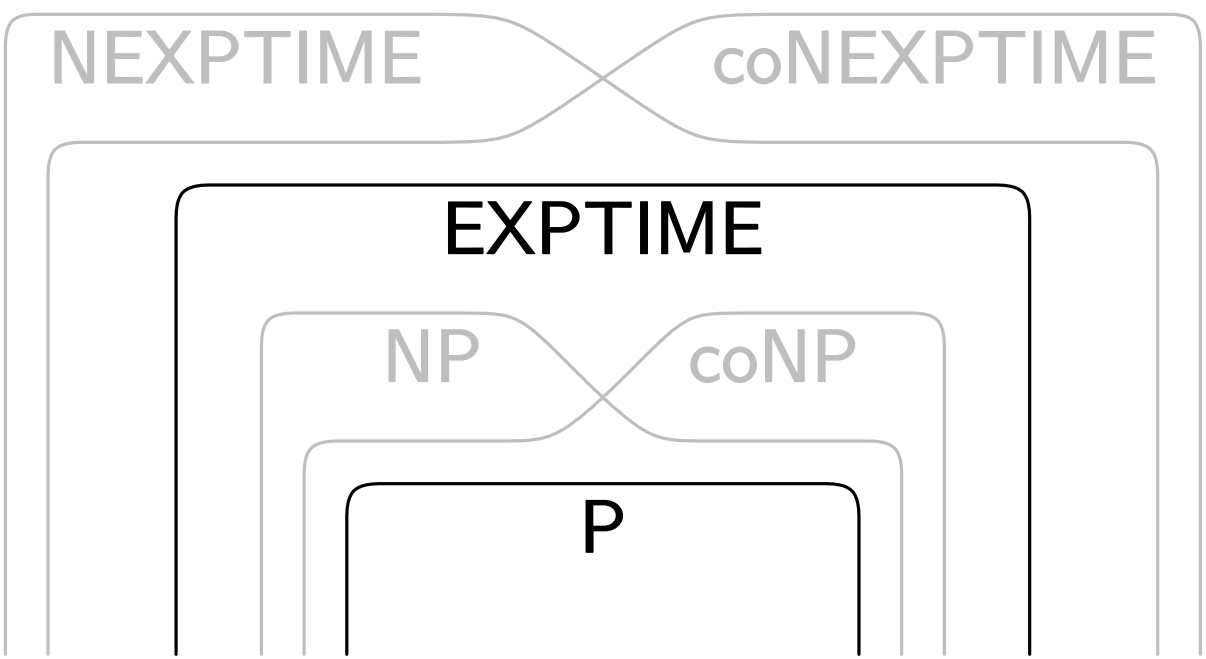
クラス間の関係 (1)

性質：定義から次が分かる

$P \subseteq EXPTIME$

予告：後の講義で次を証明する

$P \neq EXPTIME$



問題解決からの疑問

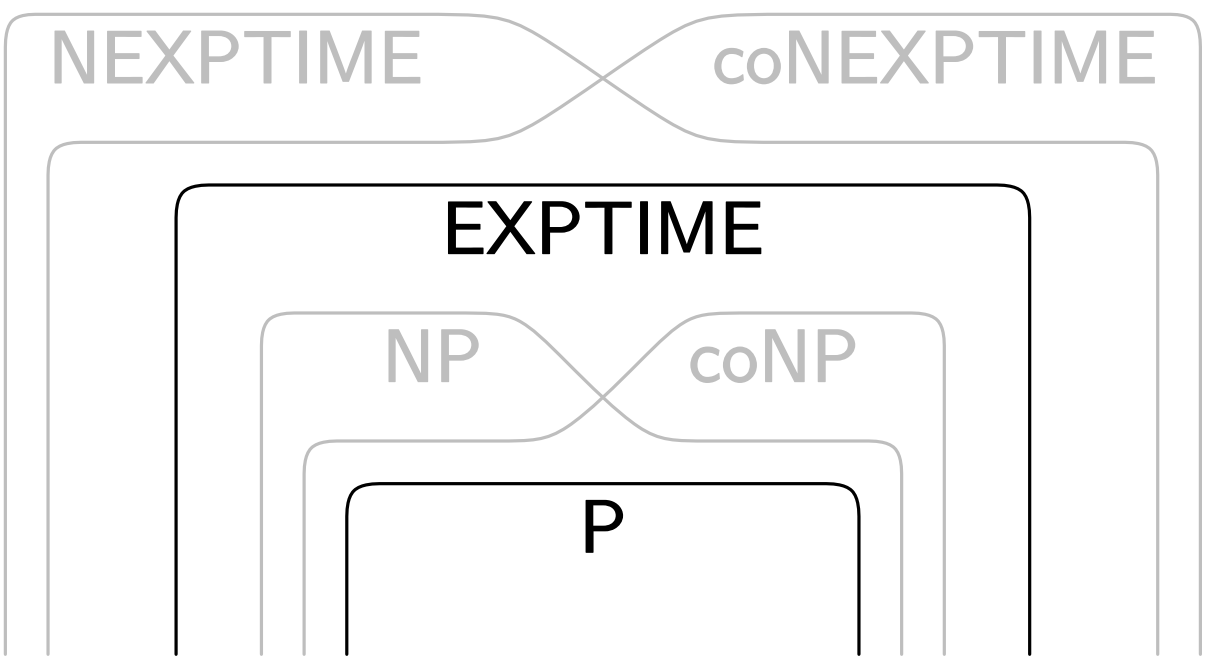
考えたいこと

現状

- 世の中には，指数時間で解けるが，多項式時間で解けるか分からない問題がたくさんある

疑問

- これらは「同程度の難しさ」を持つのか？



計算複雑性理論はこの疑問にある程度答えられる

問題解決からの疑問

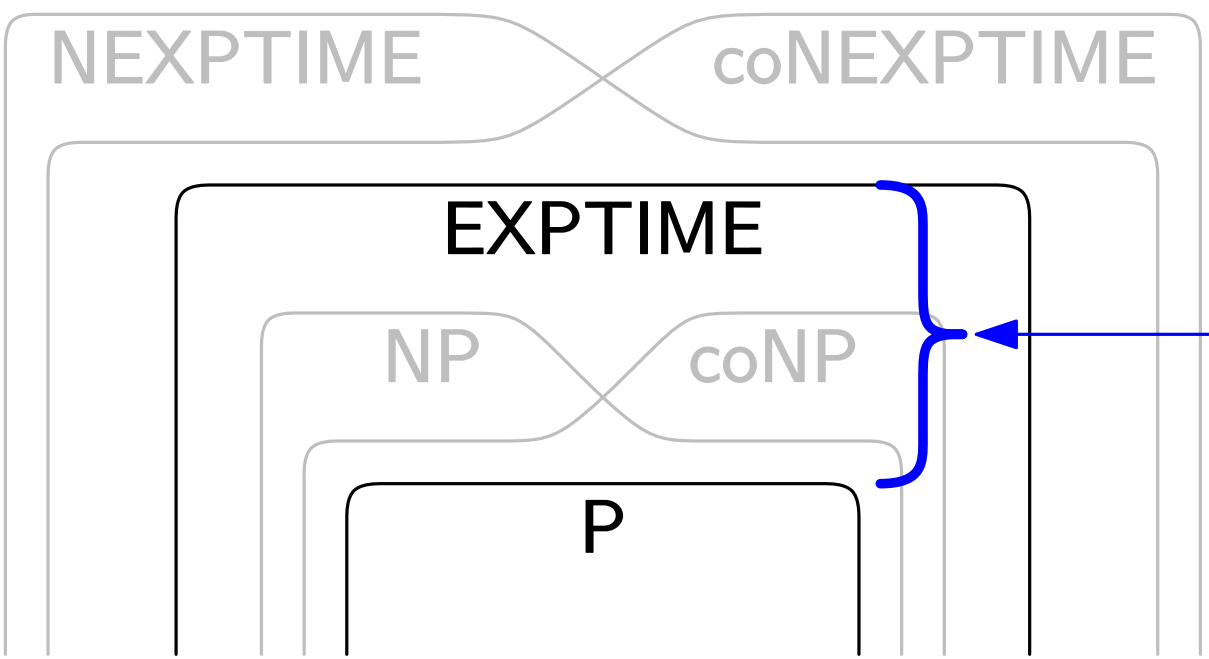
考えたいこと

現状

- 世の中には，指数時間で解けるが，多項式時間で解けるか分からない問題がたくさんある

疑問

- これらは「同程度の難しさ」を持つのか？



計算複雑性理論は
この疑問にある程度
答えられる

回答 1

P と EXPTIME の間に
無限の階層 がある

本日の内容

1. 時間計算量：P, EXPTIME
2. **非決定性の計算モデル**
3. 非決定性モデルの時間計算量：NP, NEXPTIME
4. 補クラス：coNP, coNEXPTIME

非決定性を持つ計算モデル

非決定性を持つプログラムでは

命令 guess を使える

- 有限集合 X に対して，guess(X) とすると，
  X の要素を 1 つ好きなように選べる
  - それにかかる時間 $= O(\log |X|)$ ステップ

注意：好きなように $\neq$ ランダムに

ポイント

- 非決定性は普通のプログラムで実現できない
- $\therefore$「非決定性の強さ」を知りたい

非決定性 = Non-determinism　(名詞)

非決定的 = Non-deterministic　(形容詞)

非決定性を持つ計算：例

この質問に正しく答えたい

集合 X $= \{2, 4, 6, 9\}$ の中に $a + b = 10$ となる $a, b \in X$ があるか？

決定性 (determinism)

```
foreach a in X:
  foreach b in X:
    if a + b == 10:
      return "Yes"
    end
  end
return "No"
```

非決定性

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

非決定性を持つ計算：例

この質問に正しく答えたい

集合 X $= \{2, 4, 6, 9\}$ の中に $a + b = 10$ となる $a, b \in X$ があるか？

決定性 (determinism)

```
foreach a in X:
  foreach b in X:
    if a + b == 10:
      return "Yes"
    end
  end
return "No"
```

非決定性

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

非決定性を使って解くとは？

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
```

非決定性を使って解くとは？

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

非対称！

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

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

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

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

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)

本日の内容

1. 時間計算量：P, EXPTIME
2. 非決定性の計算モデル
3. **非決定性モデルの時間計算量：NP, NEXPTIME**
4. 補クラス：coNP, coNEXPTIME

非決定性アルゴリズムの時間計算量

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時間計算量

非決定性アルゴリズム A の **時間計算量** とは，
符号長 n 以下の入力 I と guess の出力に対して，
A が行う単位操作数の I と guess に関する最大値

$$t_A(n) = \max_{\substack{I:\ |I| \leq n \\ \mathrm{guess}(\cdot)}} (A(I) \text{ にかかる単位操作数})$$

非決定性アルゴリズムの時間計算量

P は判定問題，A は P を解く非決定性アルゴリズム

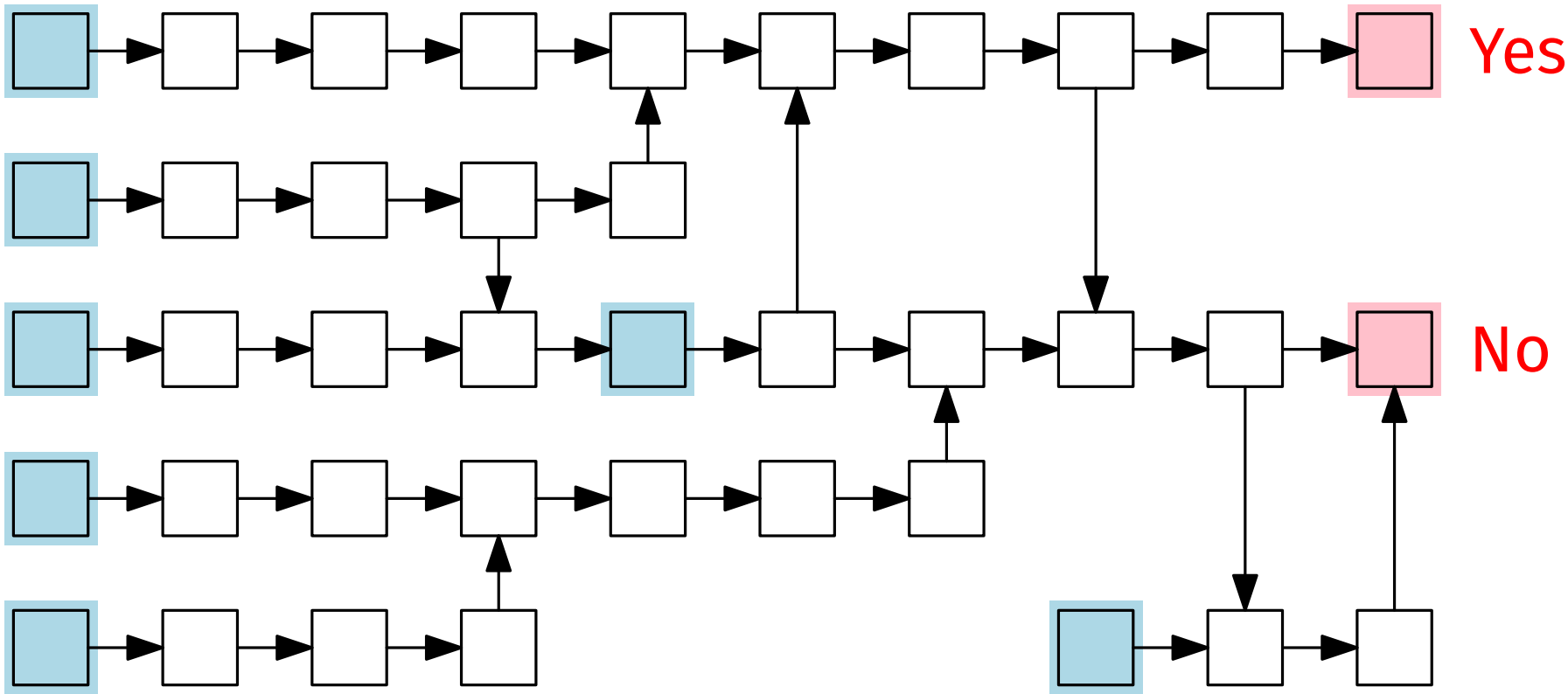
定義：非決定性アルゴリズムの時間計算量

非決定性アルゴリズム A の **時間計算量** とは，
符号長 n 以下の入力 I と guess の出力に対して，
A が行う単位操作数の I と guess に関する最大値

$$t_A(n) = \max_{\substack{I:\ |I| \leq n \\ \mathrm{guess}(\cdot)}} (A(I) \text{ にかかる単位操作数})$$

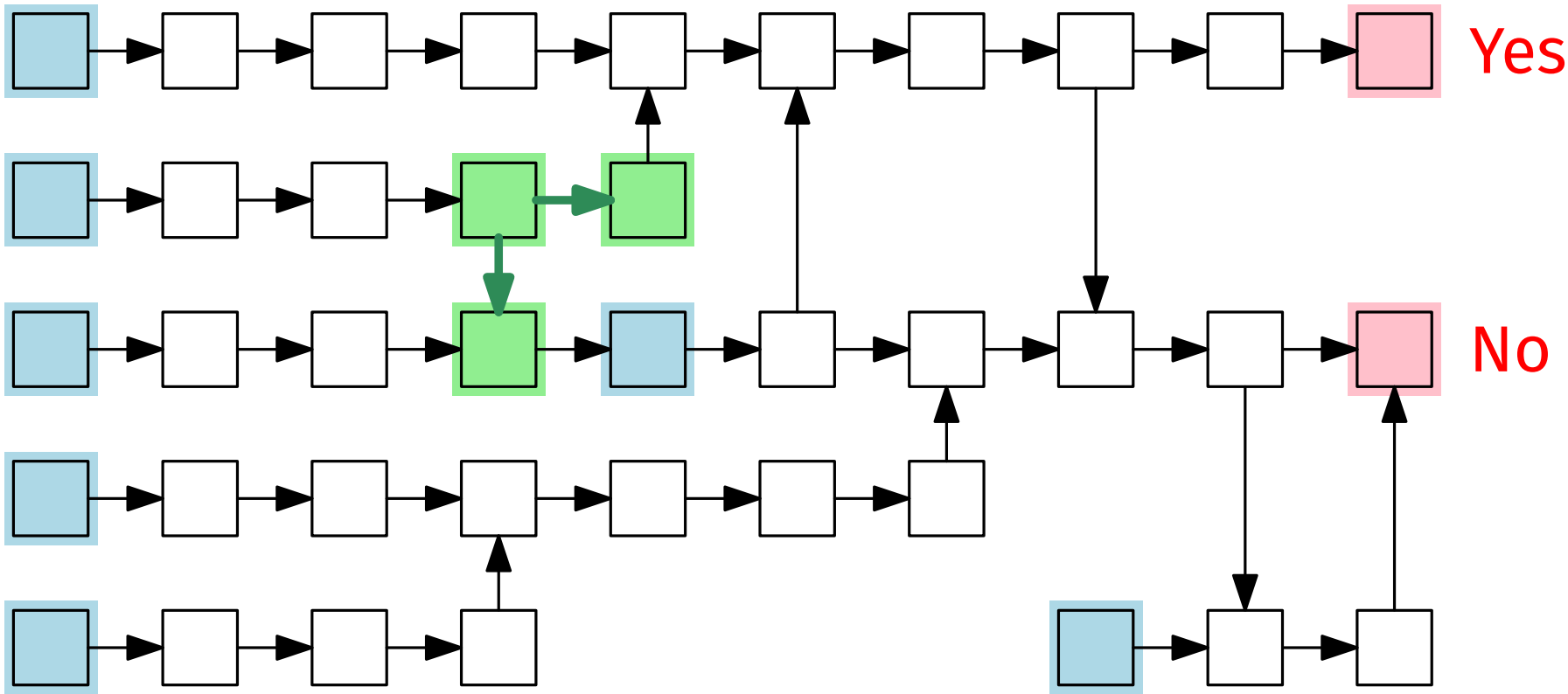
```
a = guess(X)
b = guess(X)
if a + b == 10:
  return "Yes"
else:
  return "No"
end
```

アルゴリズムの実行と時点状況



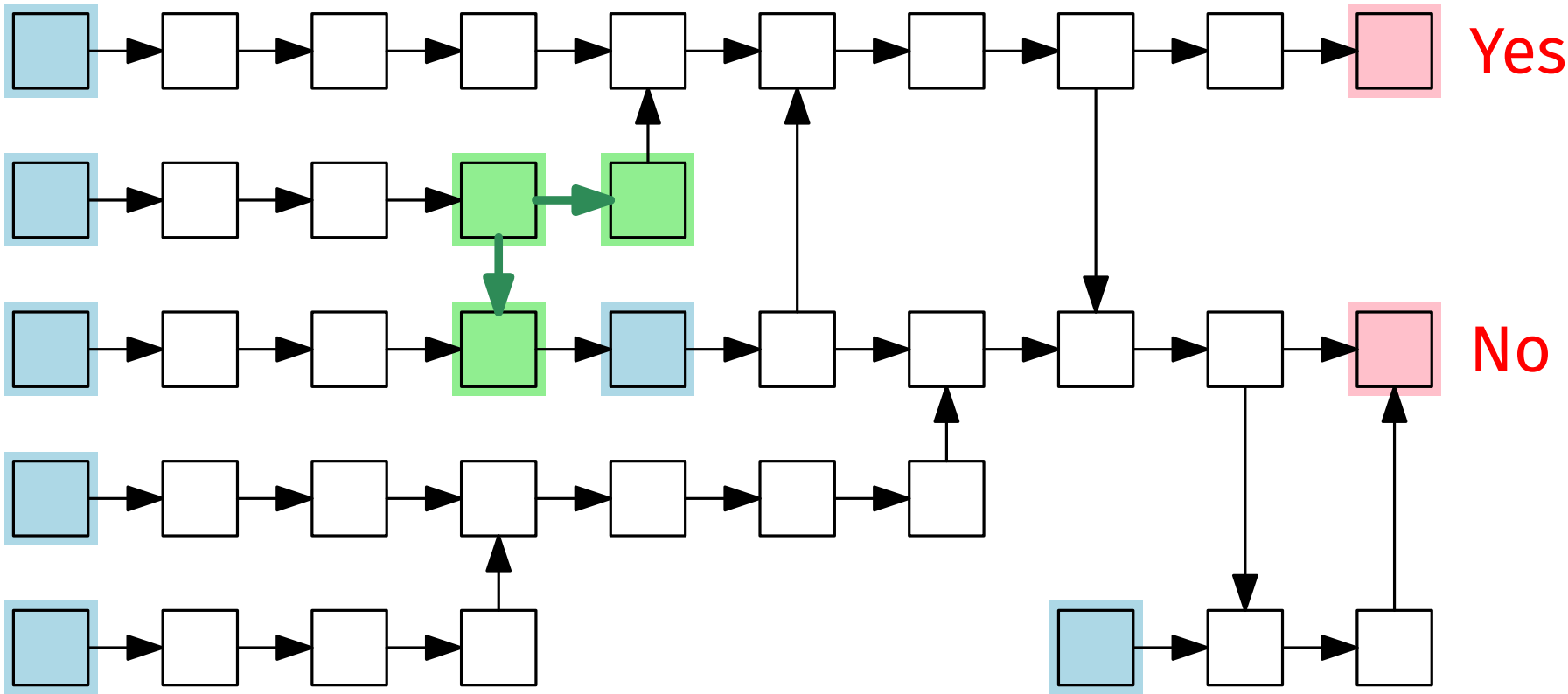
- 非決定性アルゴリズムでは,
  各状況に対して次の状況が一意に決まるとは限らない
  (それでも，高々 2 つとする)
- guess = 次の状況の選択

アルゴリズムの実行と時点状況

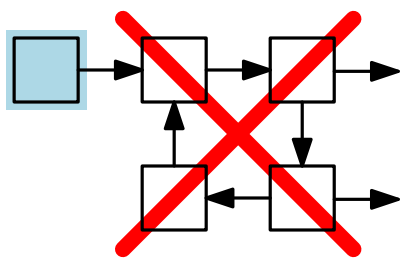


- 非決定性アルゴリズムでは,
  各状況に対して次の状況が一意に決まるとは限らない
  (それでも，高々 2 つとする)
- guess = 次の状況の選択

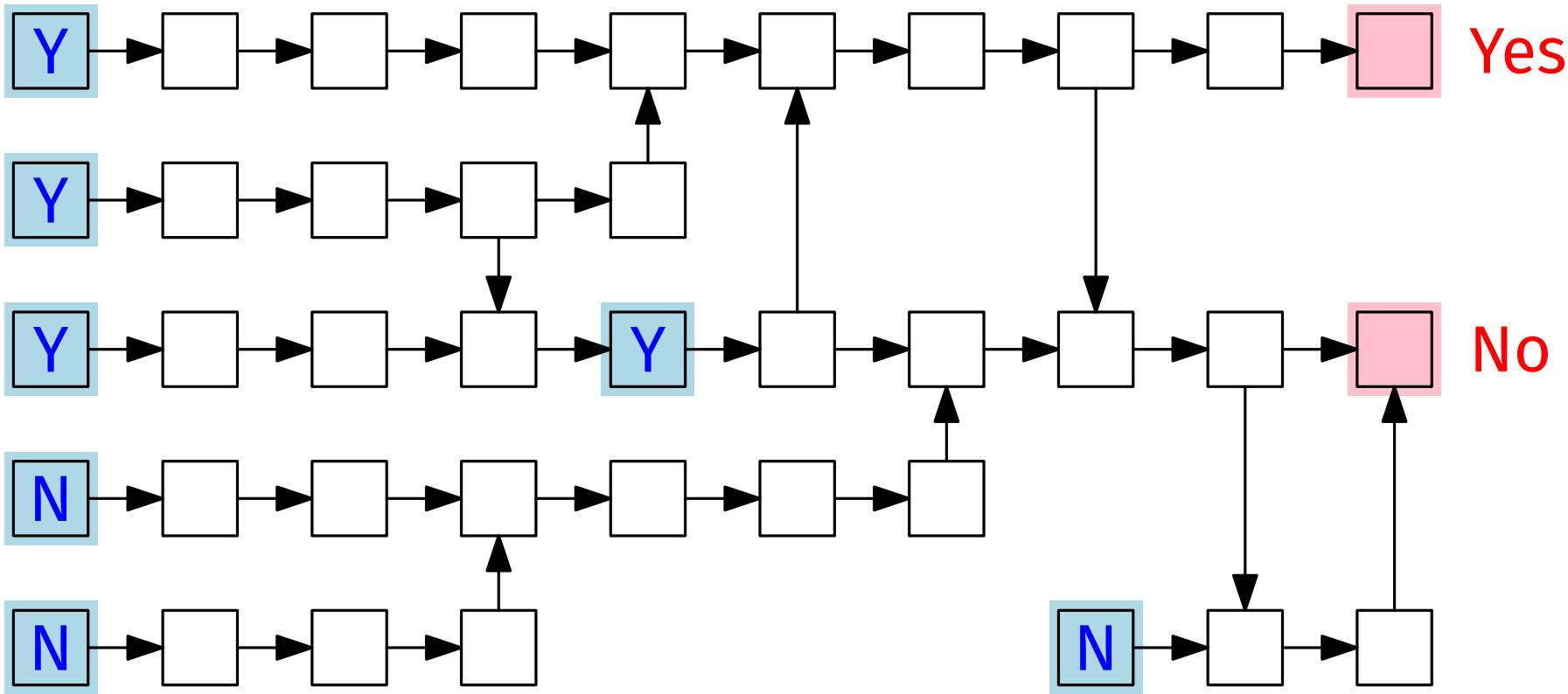
アルゴリズムの実行と時点状況



- 非決定性アルゴリズムでは,
  各状況に対して次の状況が一意に決まるとは限らない
  (それでも，高々 2 つとする)
- guess = 次の状況の選択

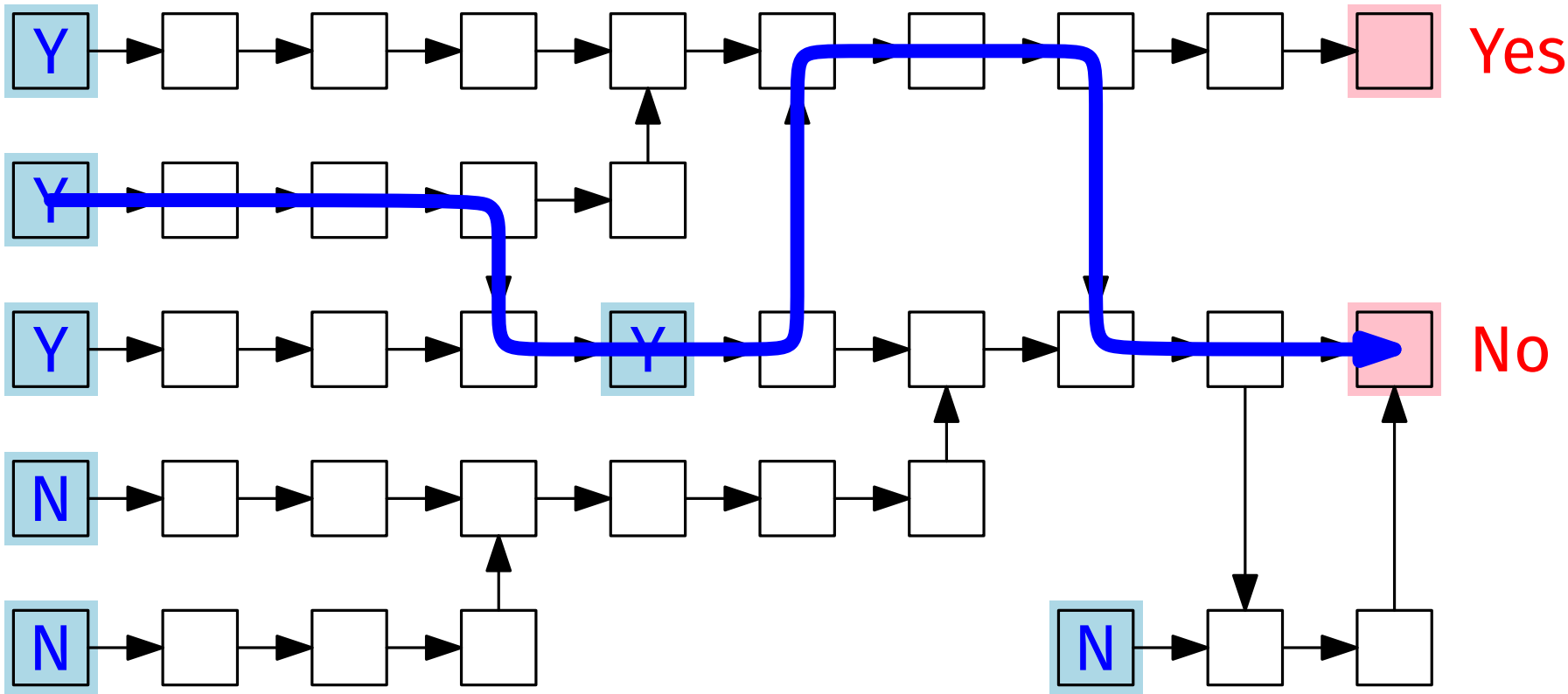


アルゴリズムの実行と時点状況



- 非決定性アルゴリズムでは,
  各状況に対して次の状況が一意に決まるとは限らない
  (それでも，高々 2 つとする)
- guess = 次の状況の選択

アルゴリズムの実行と時点状況



- 非決定性アルゴリズムでは,
  各状況に対して次の状況が一意に決まるとは限らない
  (それでも，高々 2 つとする)
- guess = 次の状況の選択

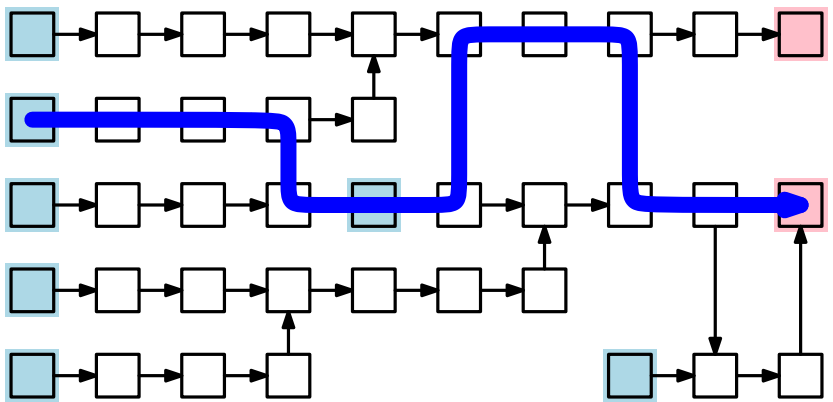
非決定性アルゴリズムの時間計算量 (2)

P は判定問題，A は P を解く非決定性アルゴリズム

同値な定義：非決定性アルゴリズムの時間計算量

アルゴリズム A の **時間計算量** (time complexity) とは，
符号長 n 以下の入力 I に対して，
I に対応する開始時点から停止時点に至る最大経路長の
I に関する最大値

仮定：アルゴリズムの各行は定数個のステップしか含まない



クラス NP とクラス NEXPTIME

定義：クラス NP

クラス NP とは，
多項式時間非決定性アルゴリズムで解ける
判定問題全体のこと

(読み方：エヌ・ピー)

クラス NP を NPTIME と書くこともある

N は「non-determinstic (非決定的)」の意味

定義：クラス NEXPTIME

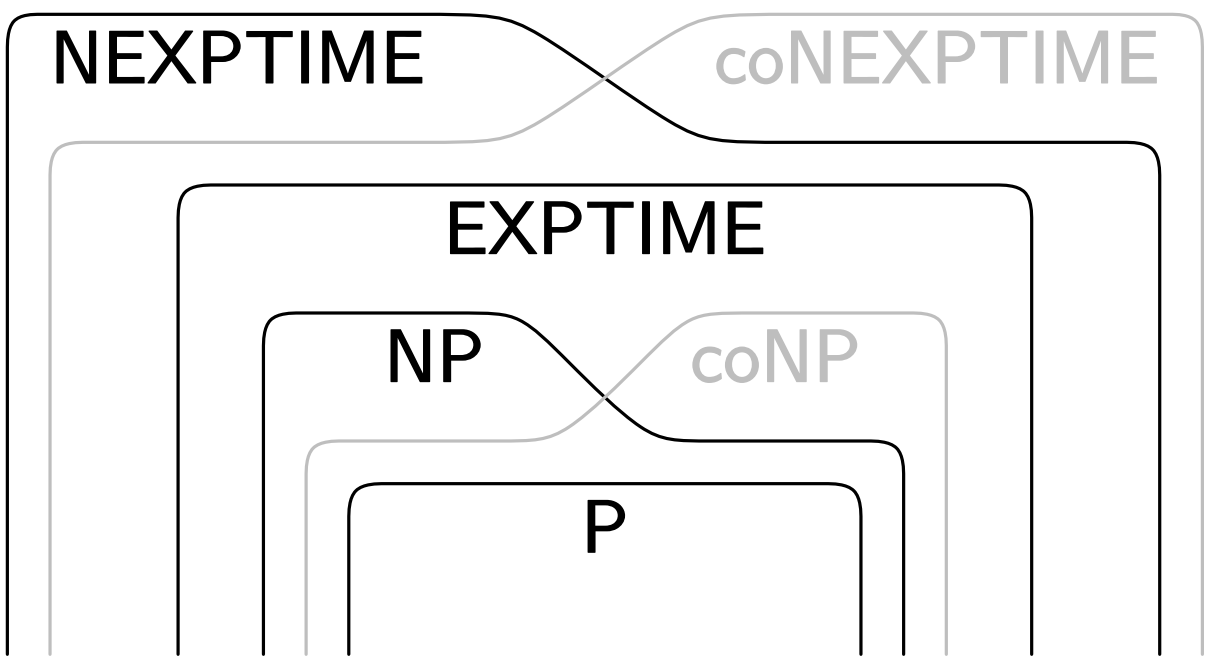
クラス NEXPTIME とは，
指数時間非決定性アルゴリズムで解ける
判定問題全体のこと

(読み方：エヌ・エクスプタイム，...)

クラス間の関係 (2)

性質：定義から次が分かる

$P \subseteq NP$，$EXPTIME \subseteq NEXPTIME$



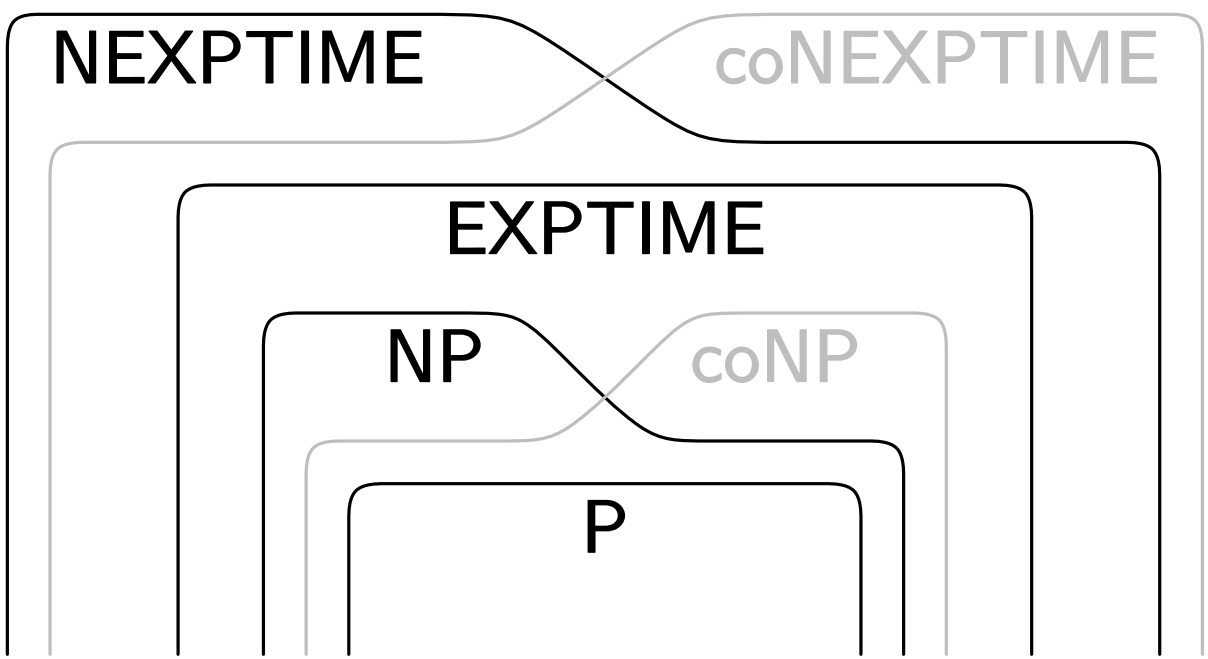
クラス間の関係 (2)

性質：定義から次が分かる

$P \subseteq NP$，$EXPTIME \subseteq NEXPTIME$

予告：後の講義で次を証明する

$NP \subseteq EXPTIME$

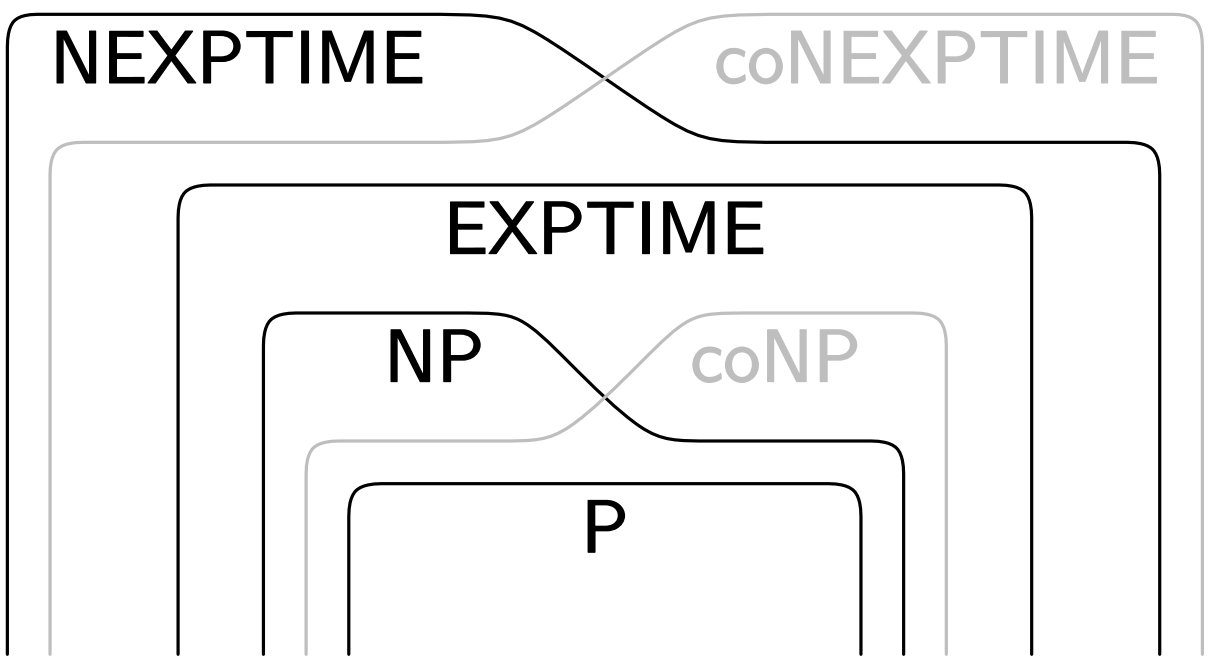


クラス間の関係 (2) 続

未解決問題

- $P \stackrel{?}{=} NP$
- $NP \stackrel{?}{=} EXPTIME$
- $EXPTIME \stackrel{?}{=} NEXPTIME$

注：$P \neq EXPTIME$ は正しい



多項式時間検証としての NP

別の (同値な) 定義：クラス NP

クラス NP とは,
証拠が与えられたとき，その検証を多項式時間で行える
判定問題全体のこと

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

a = 6
b = 4

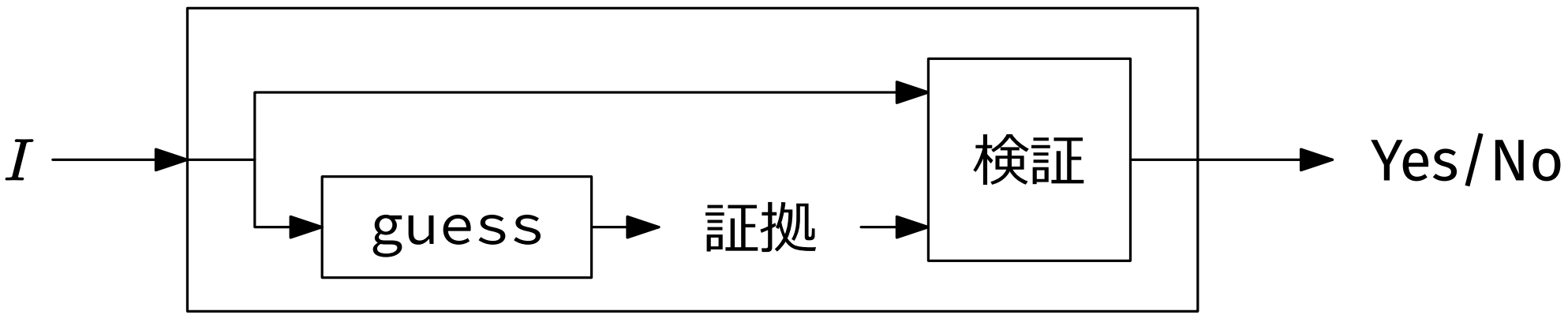
} X が Yes インスタンス であるための証拠

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)

多項式時間検証としての NP

別の (同値な) 定義：クラス NP

クラス NP とは,
証拠が与えられたとき，その検証を多項式時間で行える
判定問題全体のこと



注：検証を $|I|$ の多項式時間で行うためには

|証拠| $\leq |I|$ の多項式 でなくてはならない

本日の内容

1. 時間計算量：P, EXPTIME
2. 非決定性の計算モデル
3. 非決定性モデルの時間計算量：NP, NEXPTIME
4. **補クラス：**coNP, coNEXPTIME

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，$\overline{\overline{P}} = P$

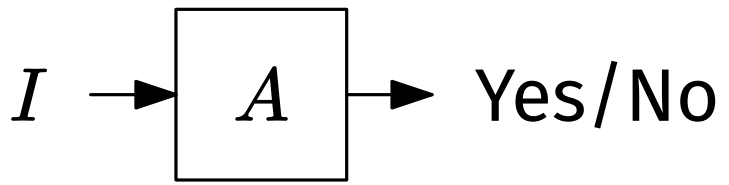
補問題

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

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

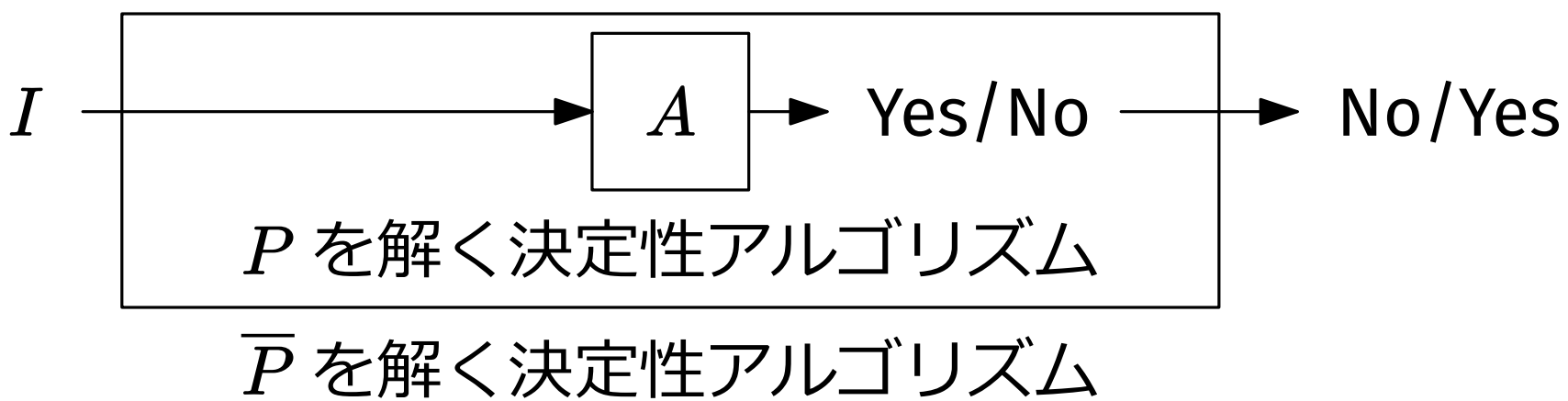
補問題

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



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

補問題と非決定性

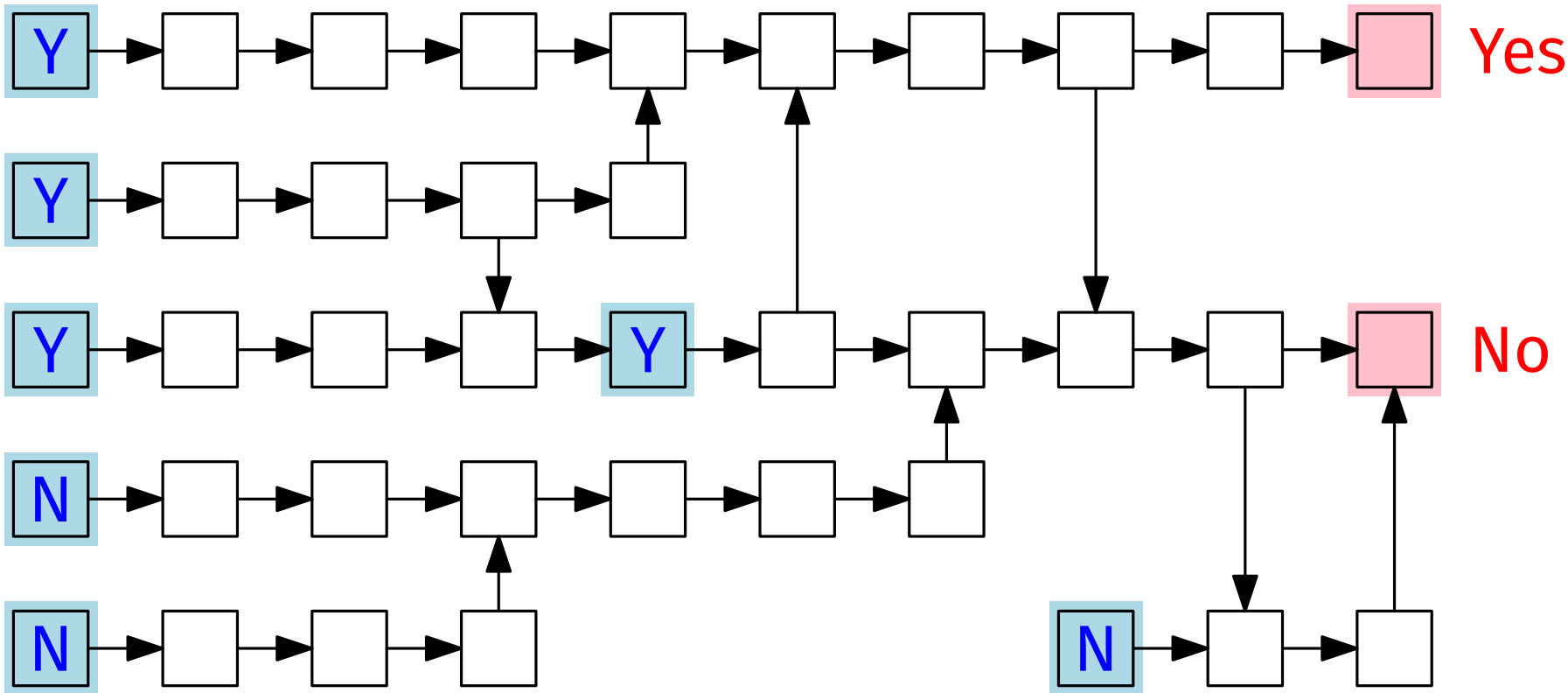
補問題と非決定性の定義から次が言える

判定問題 P の補問題 $\overline{P}$ を解く非決定性アルゴリズムは，
P の任意の入力 I に対して，次を行う

- どんな guess をしても，必ず停止する
- I が P の No インスタンス $\Rightarrow$ うまく guess すると Yes を出力
- I が P の Yes インスタンス $\Rightarrow$ どんな guess しても No を出力

	P では	$\overline{P}$ では
P の Yes インスタンス	うまい guess が必要	任意の guess を考慮
P の No インスタンス	任意の guess を考慮	うまい guess が必要

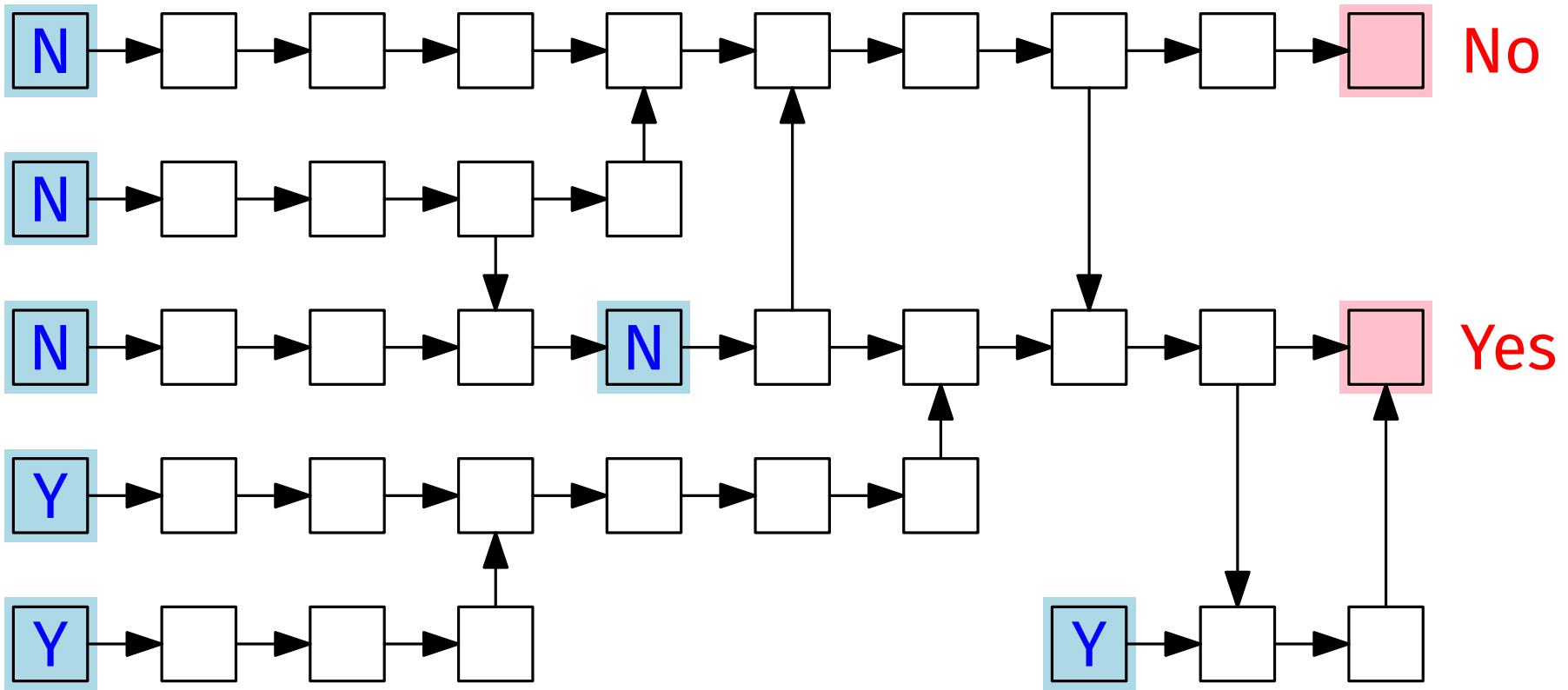
補問題と非決定性 (続)



- P を解く非決定性アルゴリズムの Yes/No を入れ換えれば $\overline{P}$ を解く非決定性アルゴリズムが得られるか？

  $\rightsquigarrow$ **得られるとは限らない！** ($P \in \mathrm{NP} \not\Rightarrow \overline{P} \in \mathrm{NP}$)

補問題と非決定性 (続)

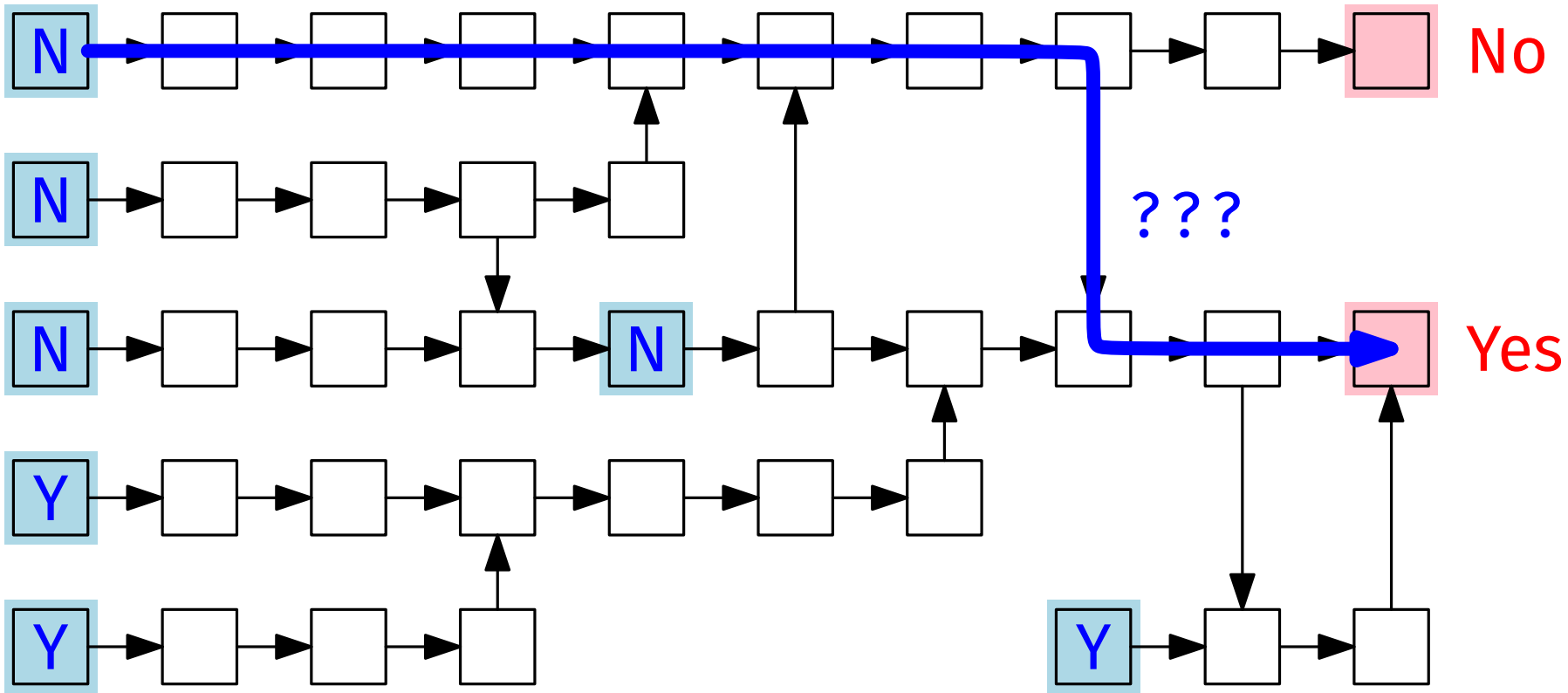


入れ換えたもの

- P を解く非決定性アルゴリズムの Yes/No を入れ換えれば $\overline{P}$ を解く非決定性アルゴリズムが得られるか？

  $\rightsquigarrow$ **得られるとは限らない！** ($P \in \mathrm{NP} \not\Rightarrow \overline{P} \in \mathrm{NP}$)

補問題と非決定性 (続)



入れ換えたもの

- P を解く非決定性アルゴリズムの Yes/No を入れ換えれば $\overline{P}$ を解く非決定性アルゴリズムが得られるか？

  $\rightsquigarrow$ **得られるとは限らない！** ($P \in \mathrm{NP} \not\Rightarrow \overline{P} \in \mathrm{NP}$)

クラス coNP とクラス coNEXPTIME

定義：クラス coNP

クラス coNP とは，
補問題が NP に属するような判定問題全体のこと

(読み方：コ・エヌ・ピー)

co は「complement (補)」の意味

定義：クラス coNEXPTIME

クラス coNEXPTIME とは，
補問題が NEXPTIME に属する判定問題全体のこと

(読み方：コ・エヌ・エクスプタイム，...)

クラス coNP とクラス coNEXPTIME

定義：クラス coNP

クラス coNP とは，
補問題が NP に属するような判定問題全体のこと

(読み方：コ・エヌ・ピー)

co は「complement (補)」の意味

定義：クラス coNEXPTIME

クラス coNEXPTIME とは，
補問題が NEXPTIME に属する判定問題全体のこと

(読み方：コ・エヌ・エクスプタイム，...)

性質：定義より

$P \in \mathrm{NP} \Leftrightarrow \overline{P} \in \mathrm{coNP}$

クラス coNP とクラス coNEXPTIME

定義：クラス coNP

クラス coNP とは，
補問題が NP に属するような判定問題全体のこと

(読み方：コ・エヌ・ピー)

co は「complement (補)」の意味

定義：クラス coNEXPTIME

クラス coNEXPTIME とは，
補問題が NEXPTIME に属する判定問題全体のこと

(読み方：コ・エヌ・エクスプタイム，...)

性質：定義より

$P \in \mathrm{NEXPTIME} \Leftrightarrow \overline{P} \in \mathrm{coNEXPTIME}$

例：素数判定

問題：素数判定

入力： 正整数 n
出力： n が素数である $\Rightarrow$ Yes
n が素数でない $\Rightarrow$ No

例：素数判定

問題：素数判定

入力： 正整数 n
出力： n が素数である $\Rightarrow$ Yes
n が素数でない $\Rightarrow$ No

問題：素数判定の補問題

入力： 正整数 n
出力： n が素数でない $\Rightarrow$ Yes
n が素数である $\Rightarrow$ No

n が素数でない $\Leftrightarrow$ 2 以上のある整数 a, b が存在して $n = ab$

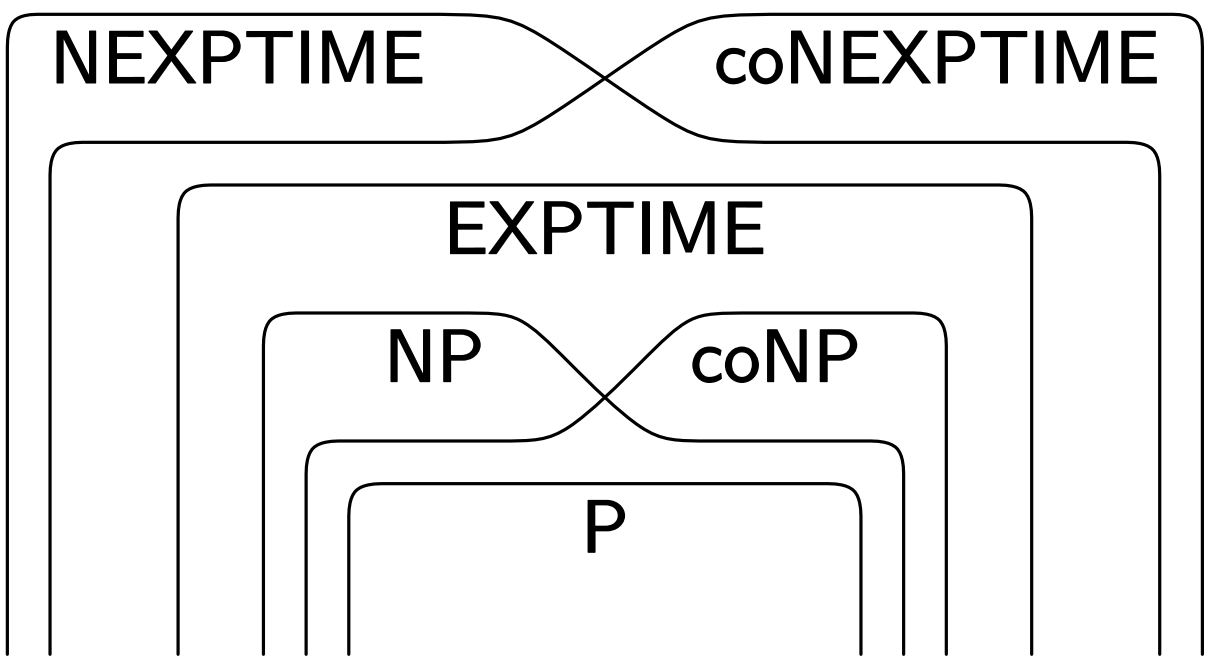
証拠

$\therefore$ 素数判定 $\in$ coNP

クラス間の関係 (3)

性質：いままでの議論から次が分かる

$P \subseteq coNP$，$EXPTIME \subseteq coNEXPTIME$



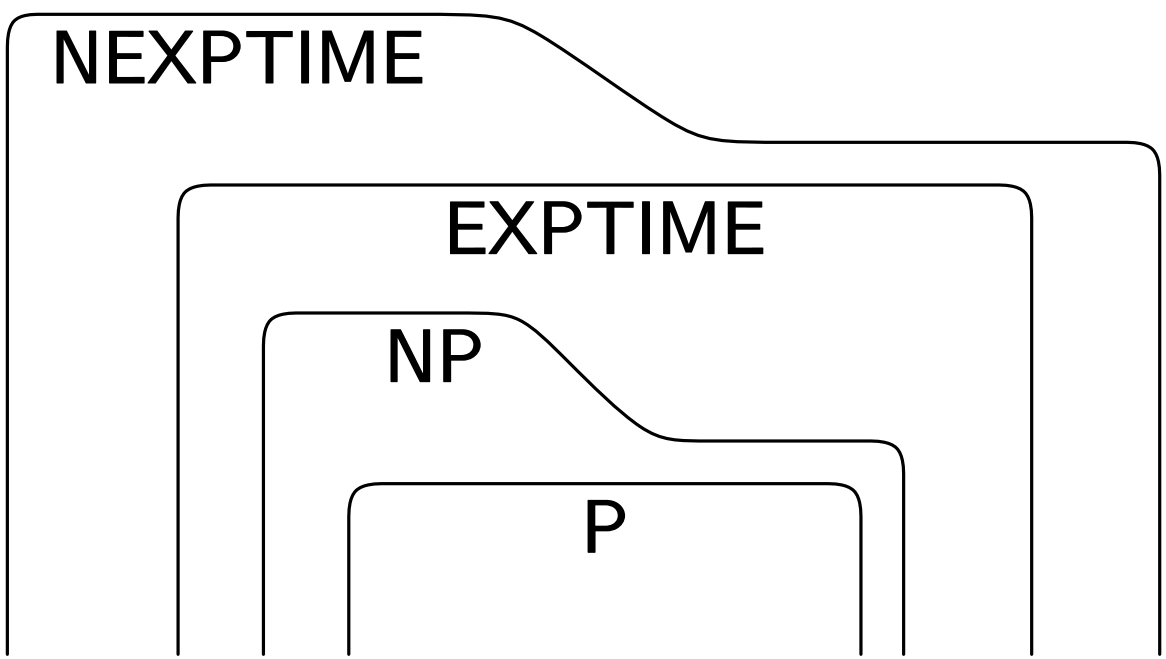
クラス間の関係 (3)

性質：いままでの議論から次が分かる

$\mathrm{P} \subseteq \mathrm{coNP}$，$\mathrm{EXPTIME} \subseteq \mathrm{coNEXPTIME}$

$\mathrm{P} \subseteq \mathrm{coNP}$ の証明：$P \in \mathrm{P}$ とする

- このとき，$\overline{P} \in \mathrm{P}$
- $\mathrm{P} \subseteq \mathrm{NP}$ より，$\overline{P} \in \mathrm{NP}$
- coNP の定義より，$P \in \mathrm{coNP}$ □



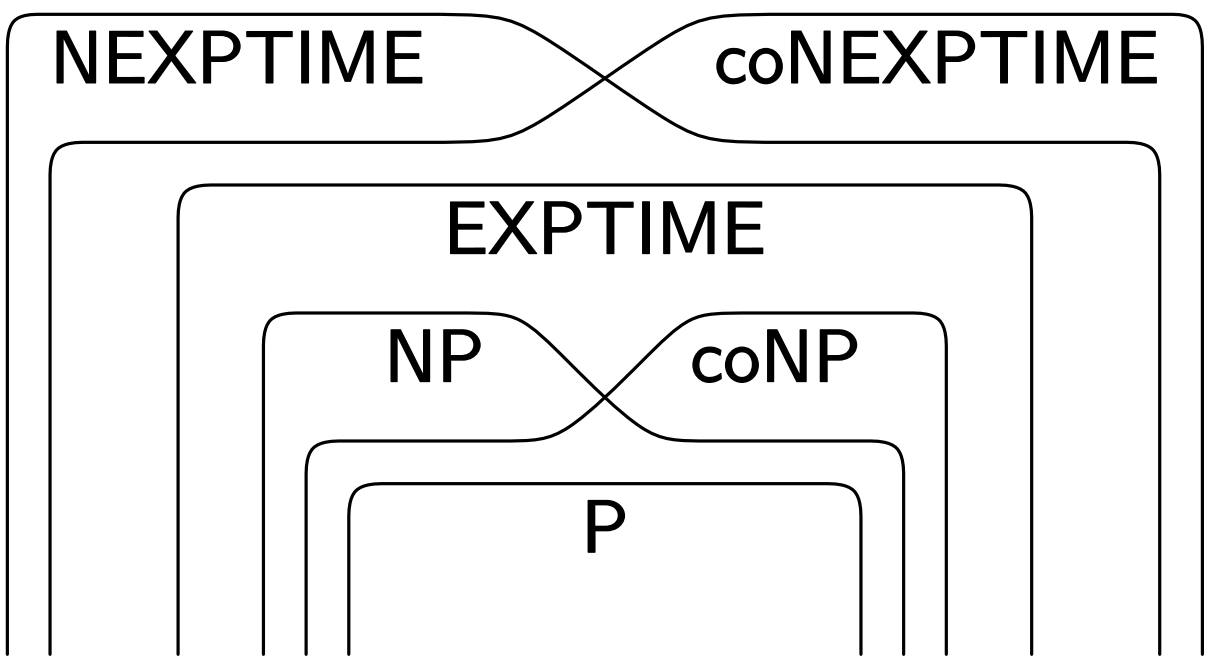
クラス間の関係 (3)

性質：いままでの議論から次が分かる

$\mathrm{P} \subseteq \mathrm{coNP}$，$\mathrm{EXPTIME} \subseteq \mathrm{coNEXPTIME}$

$\mathrm{P} \subseteq \mathrm{coNP}$ の証明：$P \in \mathrm{P}$ とする

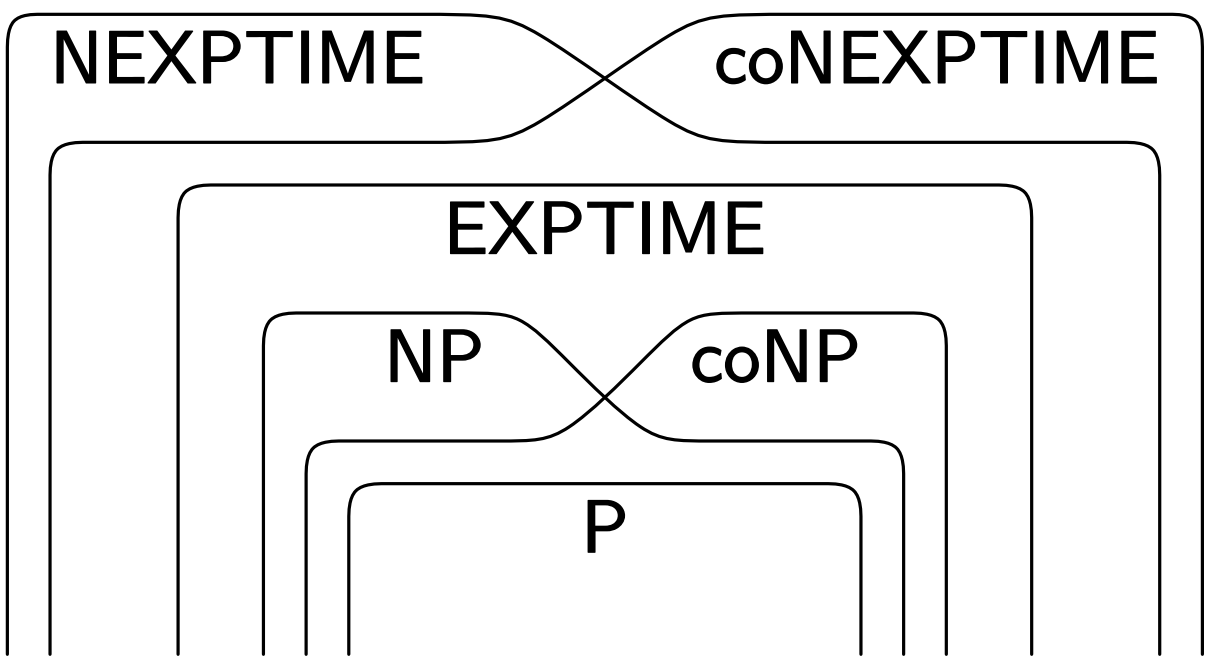
- このとき，$\overline{P} \in \mathrm{P}$
- $\mathrm{P} \subseteq \mathrm{NP}$ より，$\overline{P} \in \mathrm{NP}$
- coNP の定義より，$P \in \mathrm{coNP}$ □



クラス間の関係 (3) 続

未解決問題

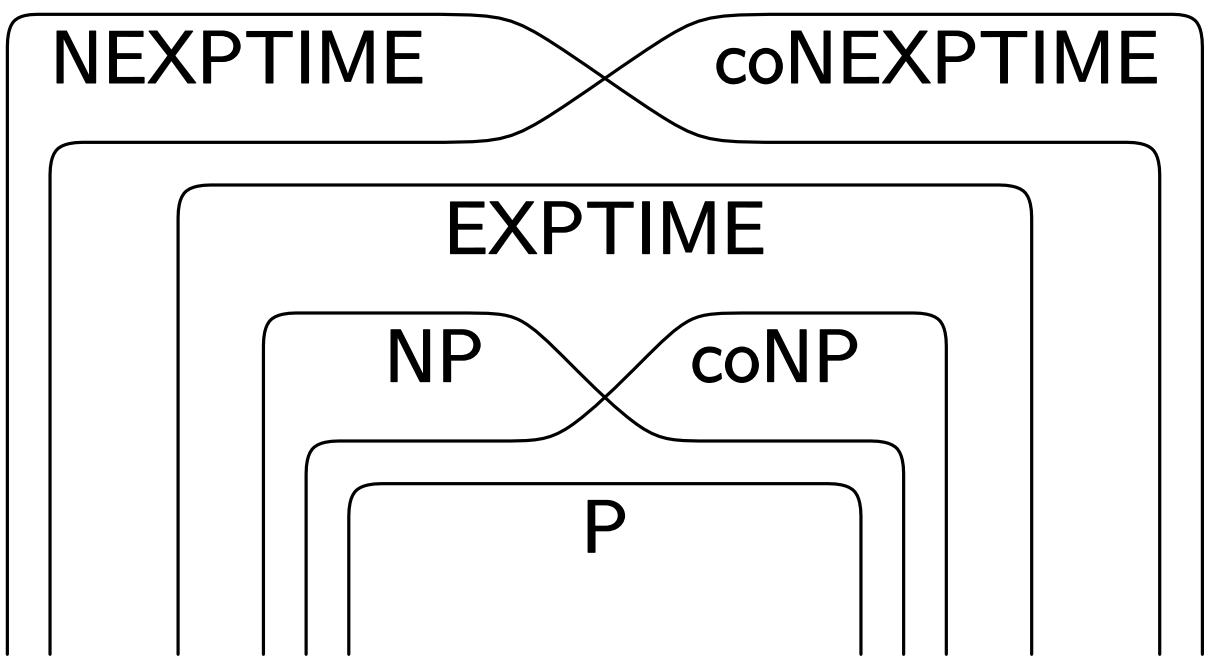
- $\mathrm{P} \stackrel{?}{=} \mathrm{coNP}$, $\mathrm{EXPTIME} \stackrel{?}{=} \mathrm{coNEXPTIME}$
- $\mathrm{NP} \stackrel{?}{=} \mathrm{coNP}$, $\mathrm{NEXPTIME} \stackrel{?}{=} \mathrm{coNEXPTIME}$
- $\mathrm{P} \stackrel{?}{=} \mathrm{NP} \cap \mathrm{coNP}$, $\mathrm{EXPTIME} \stackrel{?}{=} \mathrm{NEXPTIME} \cap \mathrm{coNEXPTIME}$



クラス間の関係 (3) 続

未解決問題

- $\mathrm{P} \stackrel{?}{=} \mathrm{coNP}$, $\mathrm{EXPTIME} \stackrel{?}{=} \mathrm{coNEXPTIME}$
- $\mathrm{NP} \stackrel{?}{=} \mathrm{coNP}$, $\mathrm{NEXPTIME} \stackrel{?}{=} \mathrm{coNEXPTIME}$
- $\mathrm{P} \stackrel{?}{=} \mathrm{NP} \cap \mathrm{coNP}$, $\mathrm{EXPTIME} \stackrel{?}{=} \mathrm{NEXPTIME} \cap \mathrm{coNEXPTIME}$



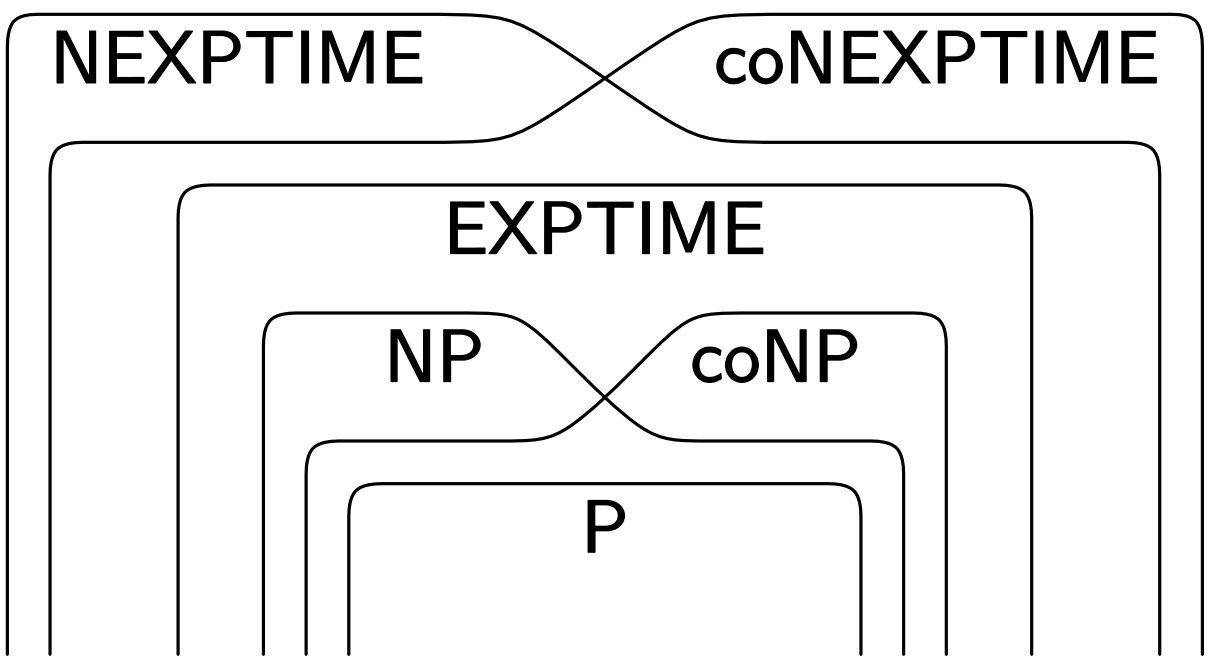
一般的に
非決定性のない
クラス $\mathcal{C}$ について
気になること

- $\mathcal{C} \stackrel{?}{=} \mathrm{N}\mathcal{C}$
- $\mathcal{C} \stackrel{?}{=} \mathrm{coN}\mathcal{C}$
- $\mathrm{N}\mathcal{C} \stackrel{?}{=} \mathrm{coN}\mathcal{C}$
- $\mathcal{C} \stackrel{?}{=} \mathrm{N}\mathcal{C} \cap \mathrm{coN}\mathcal{C}$

本日のまとめ

本日の目標

- **時間** を資源としてみることができるようになる
- **非決定性** を持つ計算モデルを扱えるようになる
- 時間に関する **計算複雑性クラス** の間の包含関係を正しく導出できる



次回以降

- NP $\subseteq$ EXPTIME, coNP $\subseteq$ EXPTIME
- P $\neq$ EXPTIME

次回の予告

次回

- 問題の難しさの比較法として **帰着** を扱う
- 各クラスで「最も難しい問題」(**完全問題**) を紹介する

重要概念

- 帰着
- NP 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.