理論計算機科学特論 (2026 年前学期)

計算複雑性の基礎

第 13 回

対話証明系 (1) : NP ⊆ MA ⊆ AM

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2026 年 7 月 7 日

最終更新：2026 年 7 月 6 日　13:11

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 $\Rightarrow$ P = PH (6/23)
12. 確率的計算：PP, RP, BPP, ZPP (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)

対話証明系ってなに？

(interactive proof system)

アーサー (Arthur)

王様

検証者 (verifier)

マーリン (Merlin)

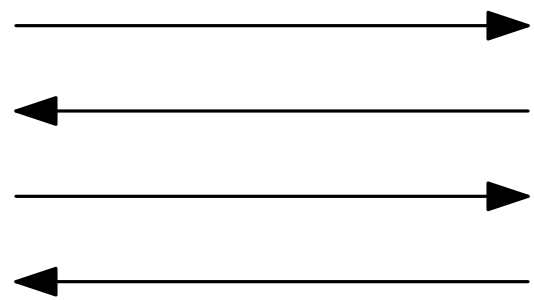
魔法使い

証明者 (prover)

対話証明系ってなに？

(interactive proof system)

メッセージ



アーサー (Arthur)

王様

検証者 (verifier)

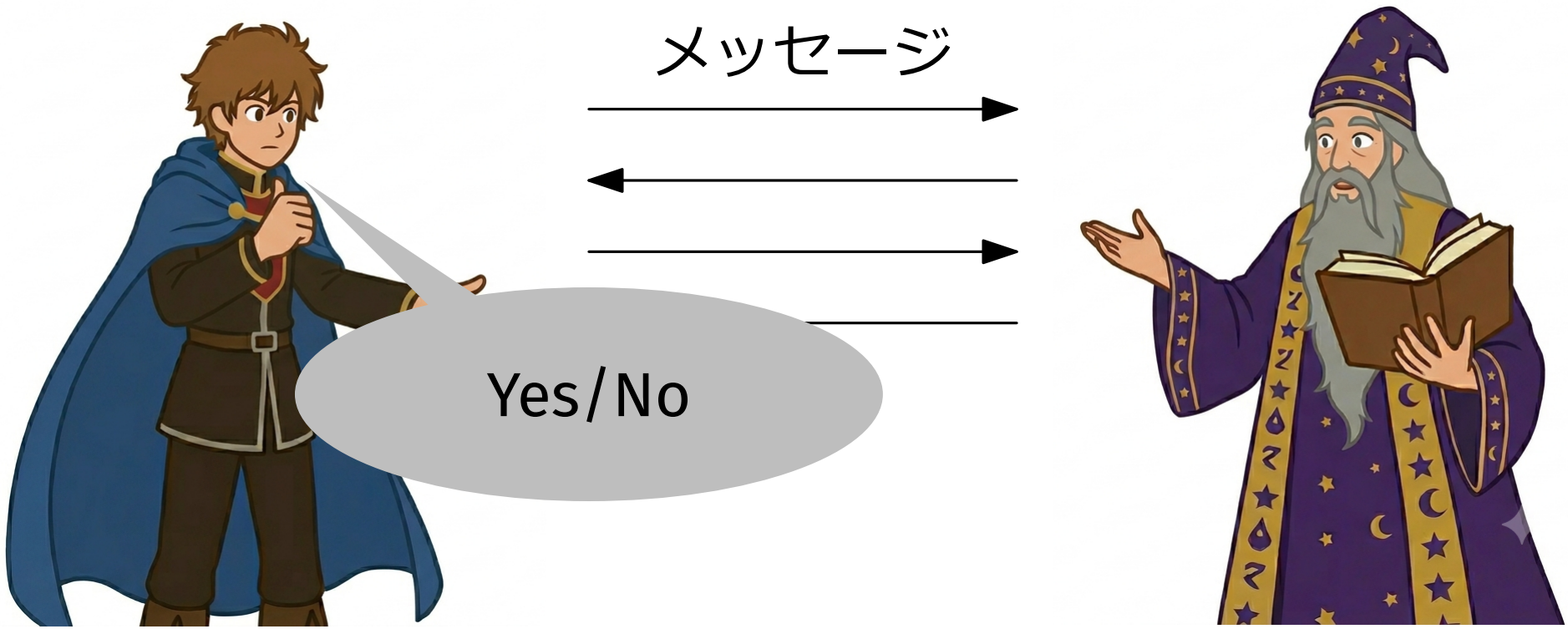
マーリン (Merlin)

魔法使い

証明者 (prover)

対話証明系ってなに？

(interactive proof system)



アーサー (Arthur)

王様

検証者 (verifier)

マーリン (Merlin)

魔法使い

証明者 (prover)

対話証明系ってなに？(続)

判定問題 P を解きたい

入力 I

計算能力：低い

計算能力：高い

信頼できない

対話証明系ってなに？(続)

判定問題 P を解きたい

入力 I

正しく Yes/No を
言いたい

アーサーに Yes と
言わせたい

計算能力：低い

計算能力：高い

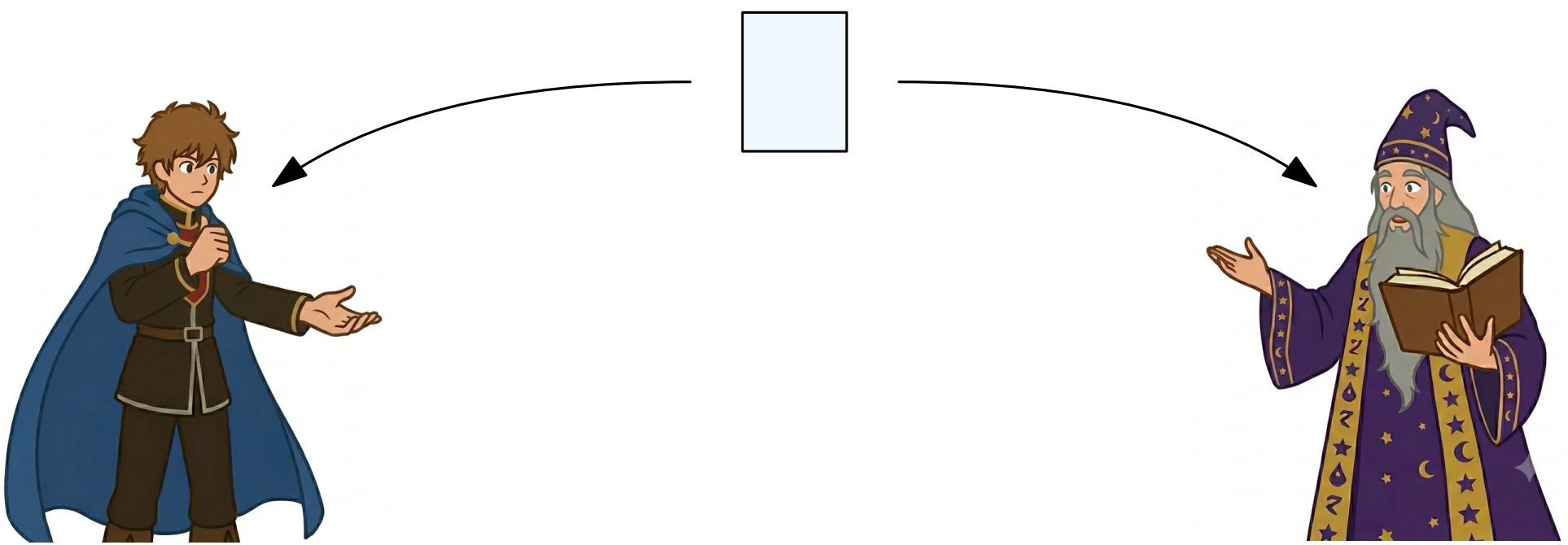
信頼できない

対話証明系：例

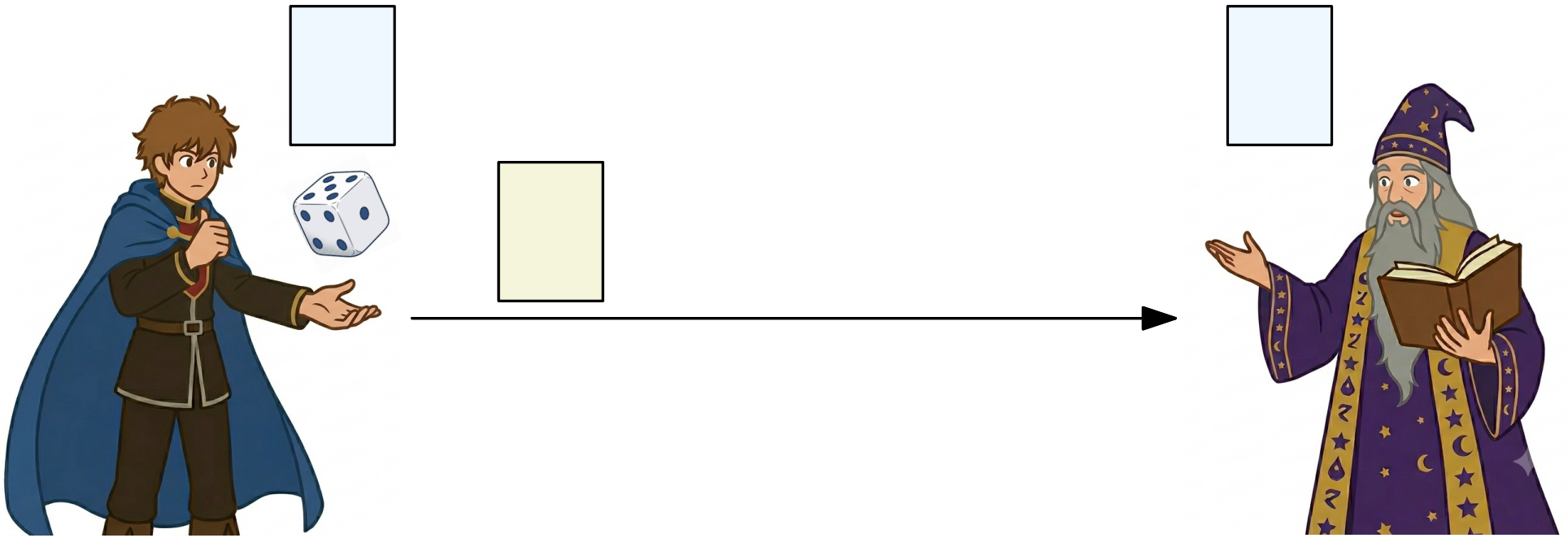
白 200 色問題

入力： 異なる 200 色の白カードの中の 1 枚
(アーサーは見分けられないが，
マーリンは見分けられる)

出力： そのカードの色が#ffffff ⇒ Yes
そうでない ⇒ No

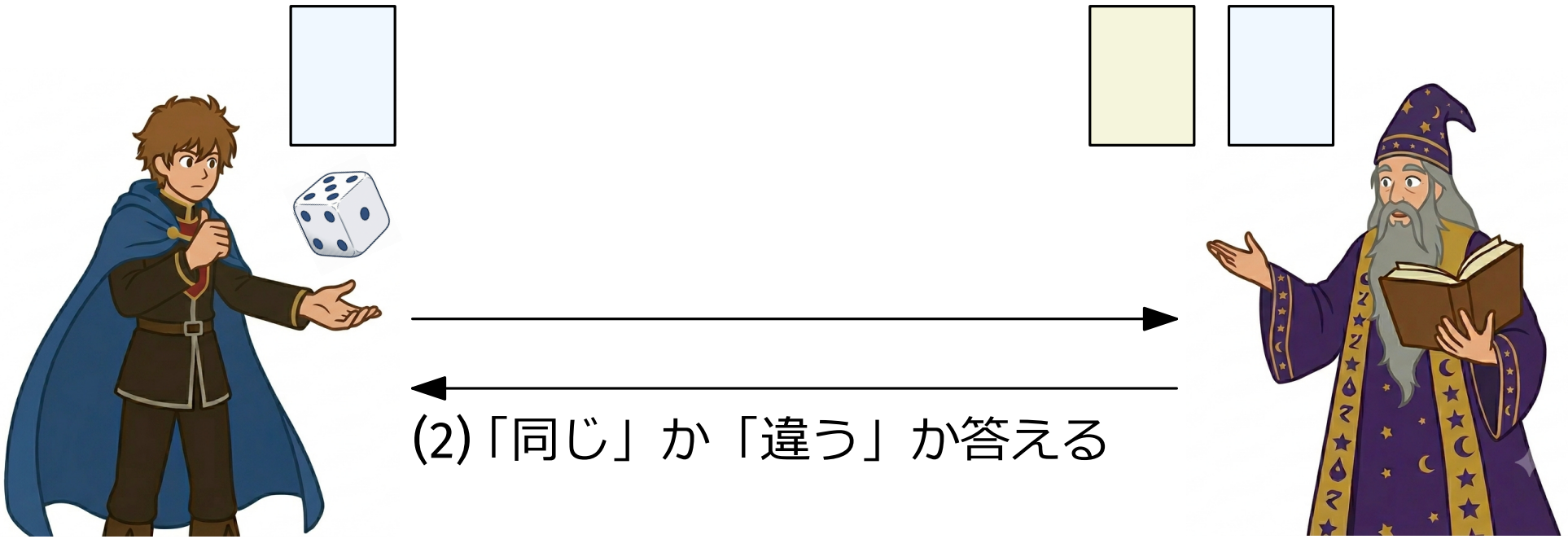


対話証明系：例 (続き 1)



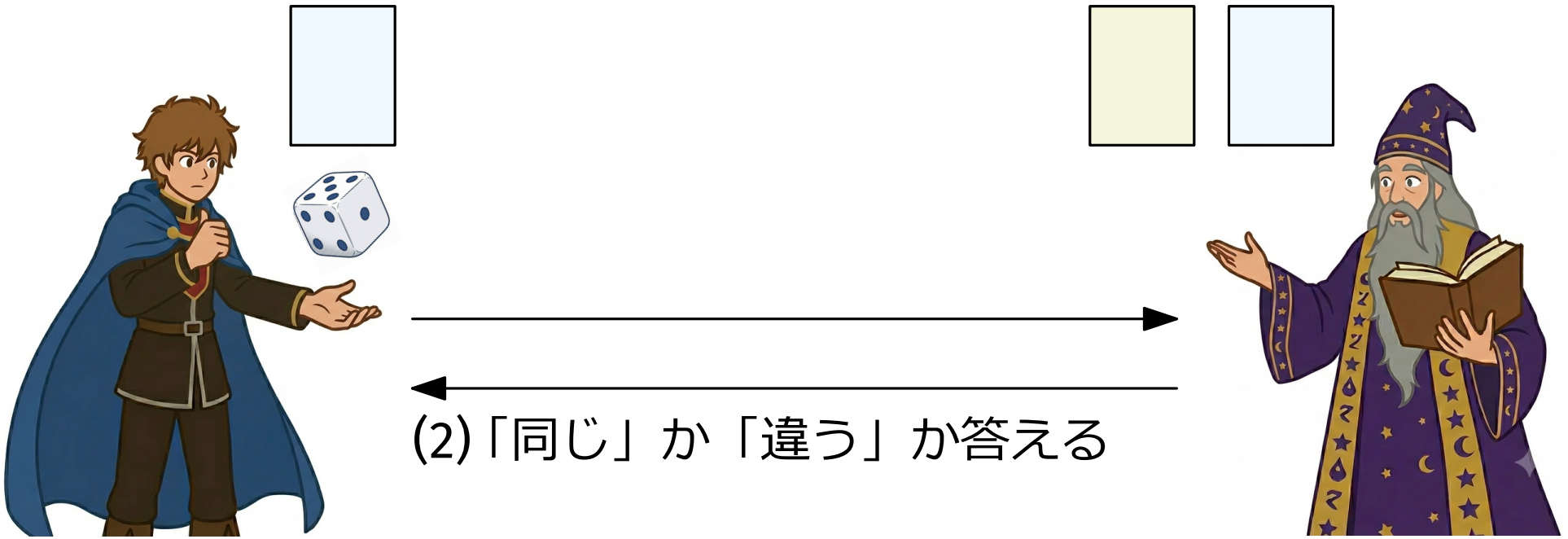
(1) コインを投げて
確率 1/2 で#ffffff のカードを
確率 1/2 で入力カードをそのまま
マーリンに渡す
(どちらを渡したか覚えておく)

対話証明系：例 (続き 1)



(1) コインを投げて
確率 $1/2$ で#ffffff のカードを
確率 $1/2$ で入力カードをそのまま
マーリンに渡す
(どちらを渡したか覚えておく)

対話証明系：例 (続き 1)



(3) 次に従って，Yes/No を判定する

		マーリンの答え 同じ	違う
アーサーが渡したカード	#ffffff	Yes	No
	入力そのもの	No	Yes

対話証明系：例 (続き 2)

入力が Yes インスタンス (入力 $=$ `#ffffff`) $\Rightarrow$

$\Pr$(アーサーが Yes と言う) $= 1/2$

		マーリンの答え: 同じ	違う
アーサーが渡したカード	`#ffffff`	Yes	No
	入力そのもの	No	Yes

対話証明系：例 (続き 2)

入力が Yes インスタンス (入力 $=$ #ffffff) $\Rightarrow$

$\Pr(\text{アーサーが Yes と言う}) = 1/2$

入力が No インスタンス (入力 $\neq$ #ffffff) $\Rightarrow$

$\Pr(\text{アーサーが No と言う}) = 1$

$\therefore$ アーサーは (マーリンの力を借りた) RP のアルゴリズム

		マーリンの答え	
		同じ	違う
アーサーが渡したカード	#ffffff	Yes	No
	入力そのもの	No	Yes

対話証明系：別の例

先生 (検証者)

生徒 (証明者)

対話証明系：別の例

質問 →

← 回答

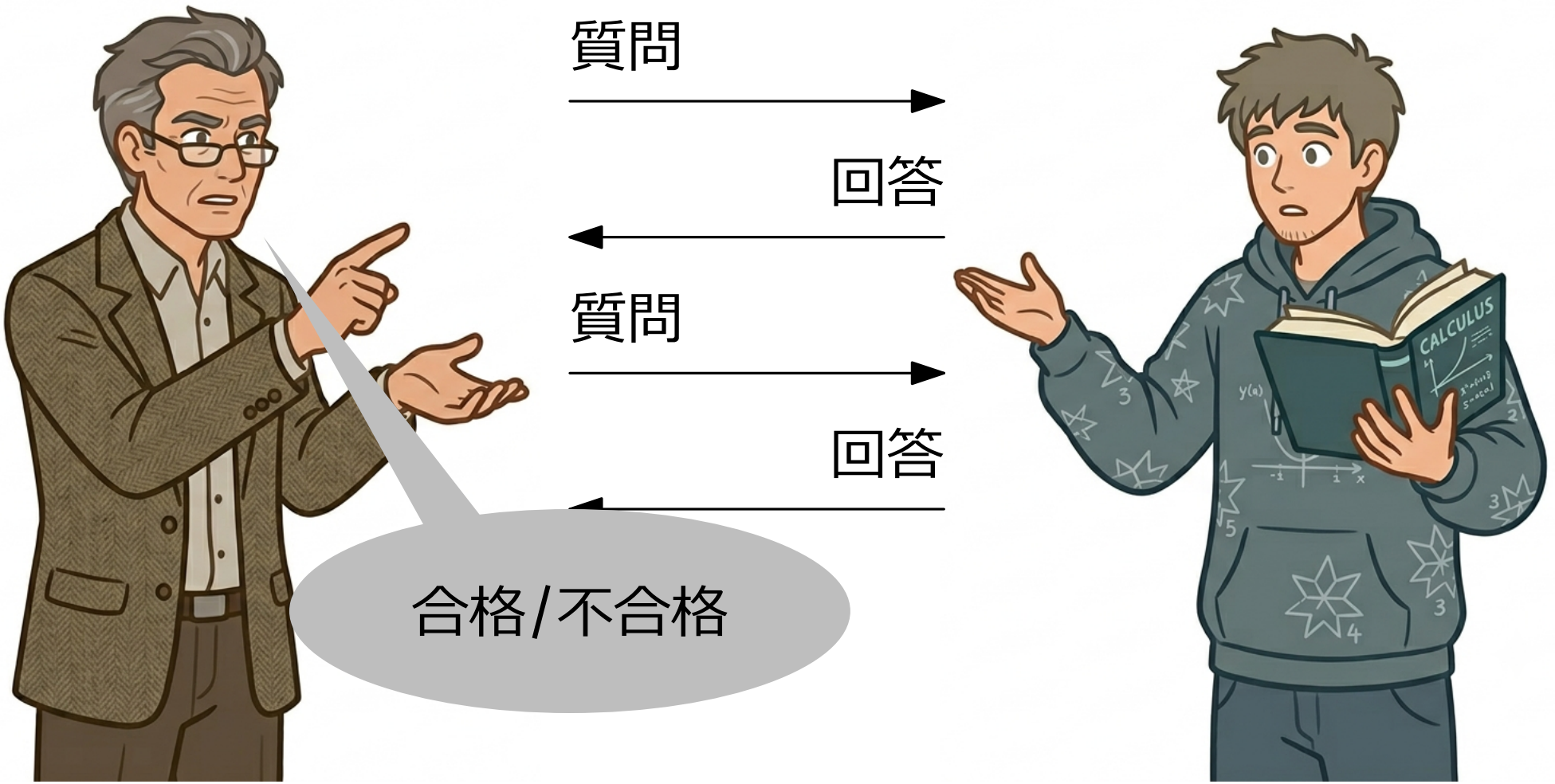
質問 →

← 回答

先生 (検証者)

生徒 (証明者)

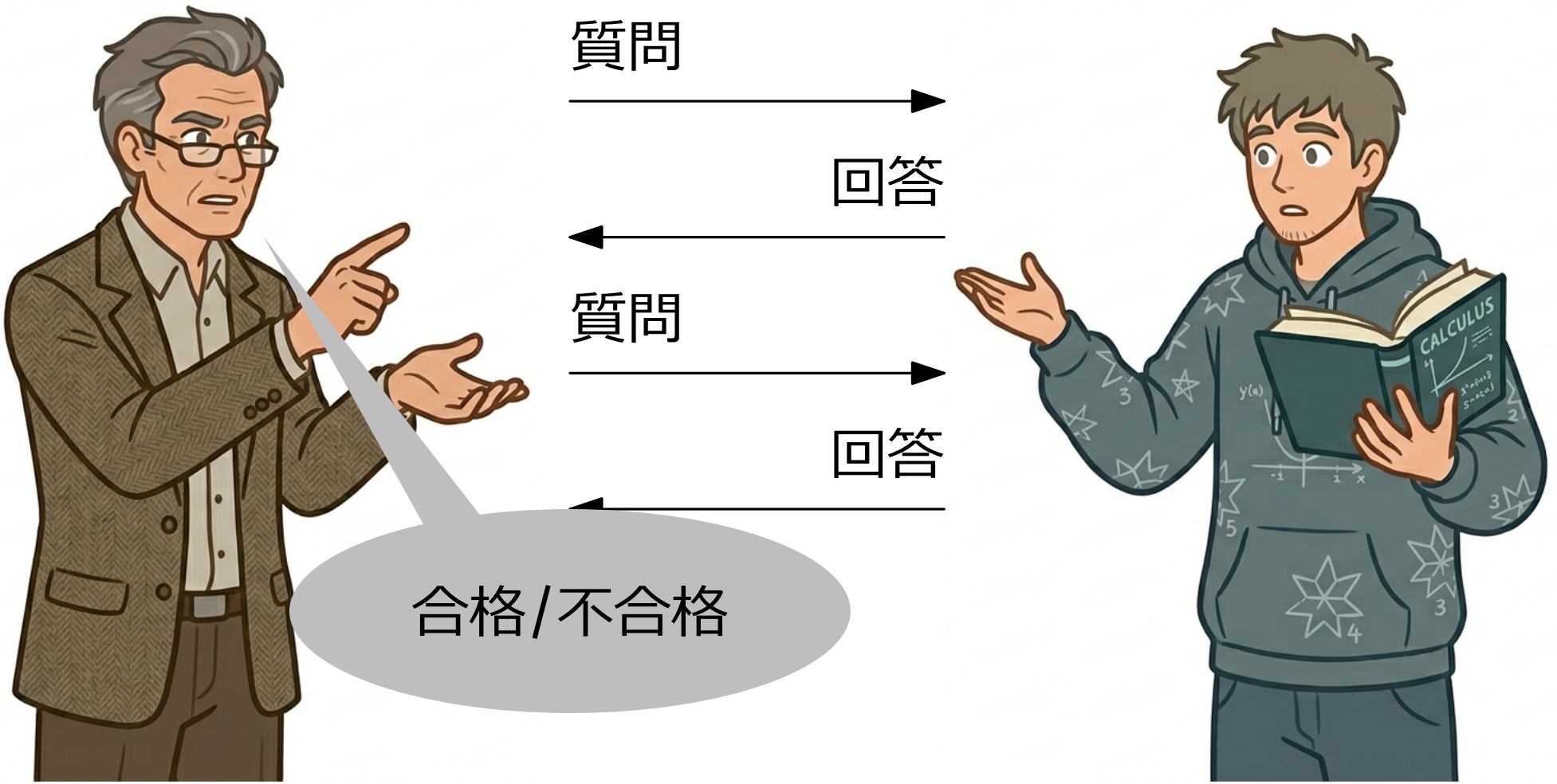
対話証明系：別の例



先生 (検証者)

生徒 (証明者)

対話証明系：別の例



先生 (検証者)

正しく合格/不合格を
判定したい

生徒 (証明者)

先生に「合格」と
言わせたい

本日の内容

内容

- 対話証明系の計算モデルを導入する
- 複雑性クラス AM$[k]$，MA$[k]$，IP$[k]$ を導入する
- いままで登場した計算複雑性クラスとの関係を調べる

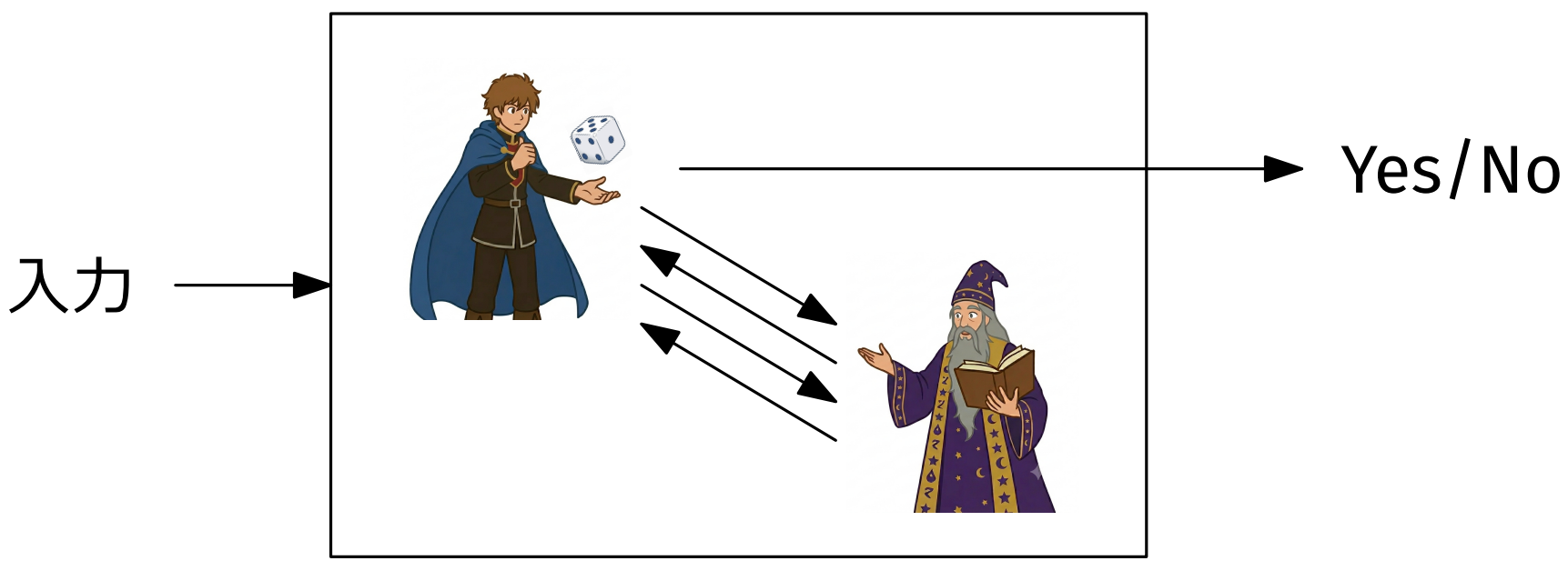
本日の内容

1. **対話証明系**
2. 複雑性クラスの関係

対話証明系：設定

対話証明系の設定

- 登場人物
  検証者　多項式時間アルゴリズム＋乱数使用
  証明者　なんでも解ける (計算不能問題も解ける)
- 入力は両方に与えられる
- 出力は検証者が行う

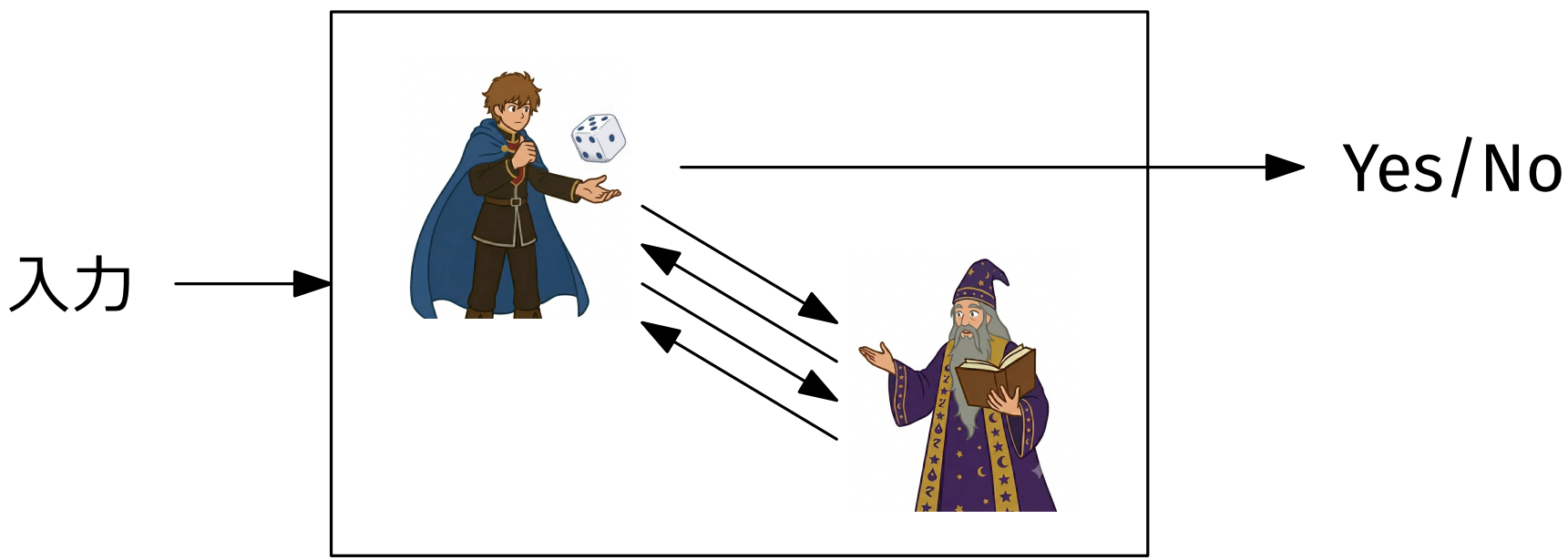


対話証明系：プロトコル

定義 (非形式)：プロトコル

対話証明系における **プロトコル** (protocol) とは，
検証者と証明者が行う動作を定めたもの

検証者・証明者はプロトコルから逸脱しない



対話証明系によって問題を解くとは

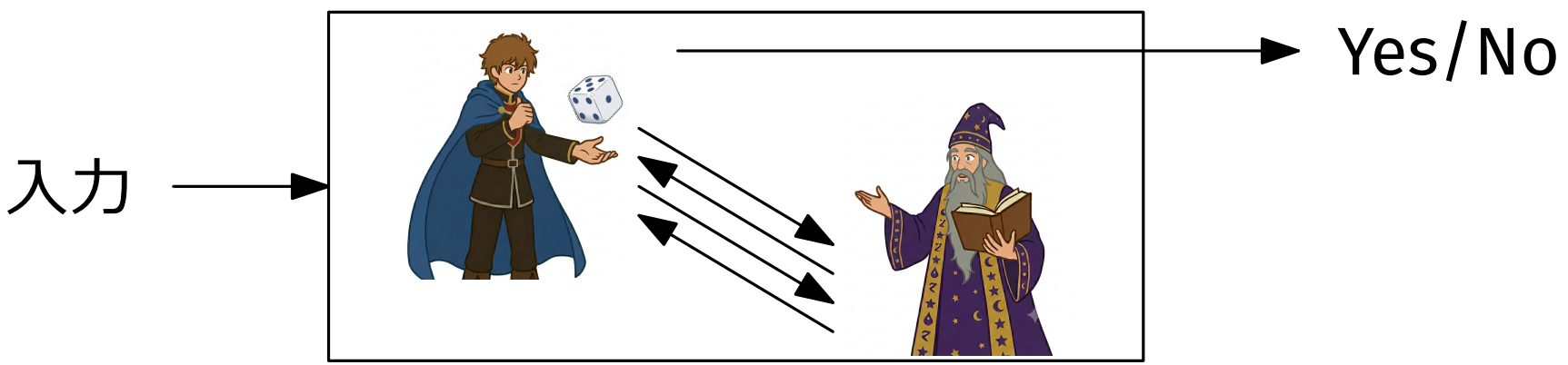
プロトコル A が判定問題 P を解くとは

- 入力が P の Yes インスタンス ⇒
  証明者がうまく動作することで，次を満たす

$$\Pr_r(\text{検証者が Yes を出力}) \geq \frac{2}{3}$$

- 入力が P の No インスタンス ⇒
  証明者がどのように動作しても，次を満たす

$$\Pr_r(\text{検証者が No を出力}) \geq \frac{2}{3}$$



対話証明系によって問題を解くとは

プロトコル A が判定問題 P を解くとは

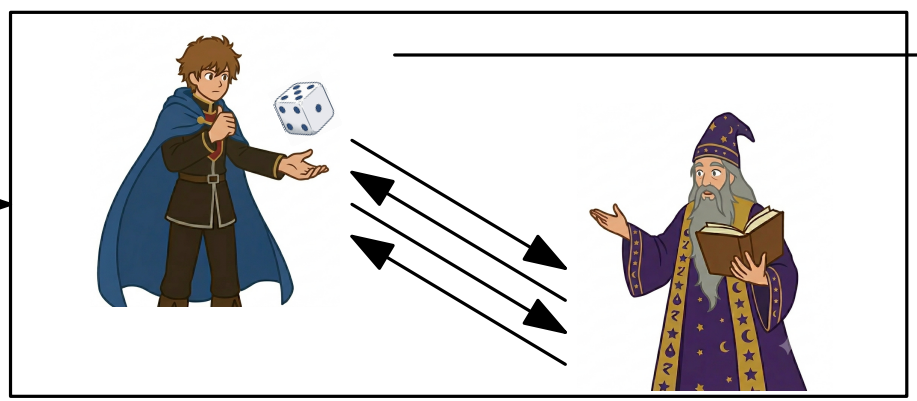
- 入力が P の Yes インスタンス ⇒
  証明者がうまく動作することで，次を満たす

$$\Pr_r(\text{検証者が Yes を出力}) \geq \frac{2}{3}$$

- 入力が P の No インスタンス ⇒
  証明者がどのように動作しても，次を満たす

$$\Pr_r(\text{検証者が No を出力}) \geq \frac{2}{3}$$

検証者の乱数列 r



複雑性クラス AM$[k]$

正整数 k

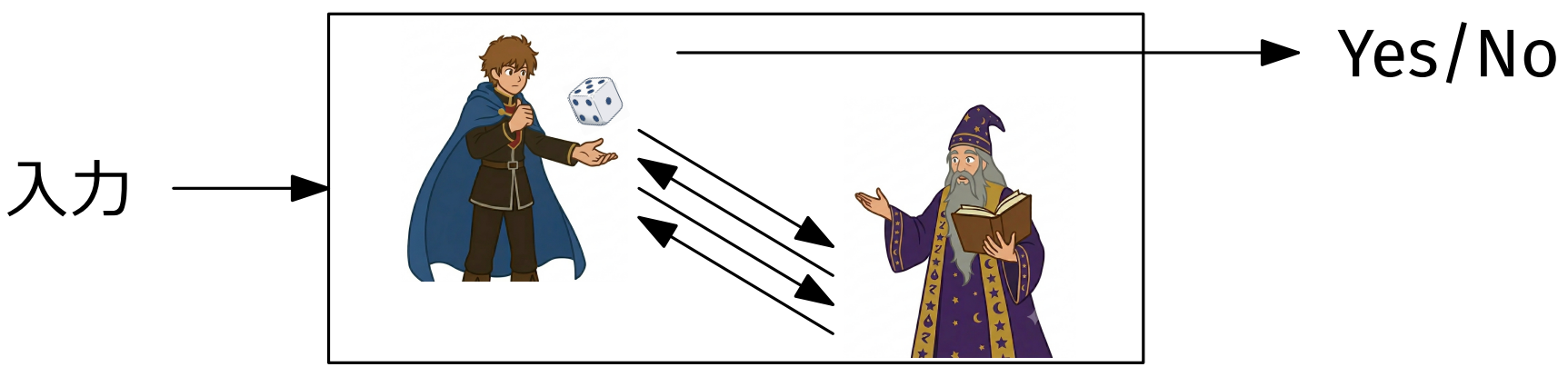
複雑性クラス AM$[k]$ (Babai '85)

クラス AM$[k]$ は次を満たすプロトコルで解ける問題全体

- プロトコルは 検証者 の動作から始まる
- 検証者と証明者が順に計 k ラウンド の動作を行う
- 検証者の乱数は証明者から見える (public coin)

AM = Arthur-Merlin

AM$[4]$ のプロトコル



複雑性クラス MA$[k]$

正整数 k

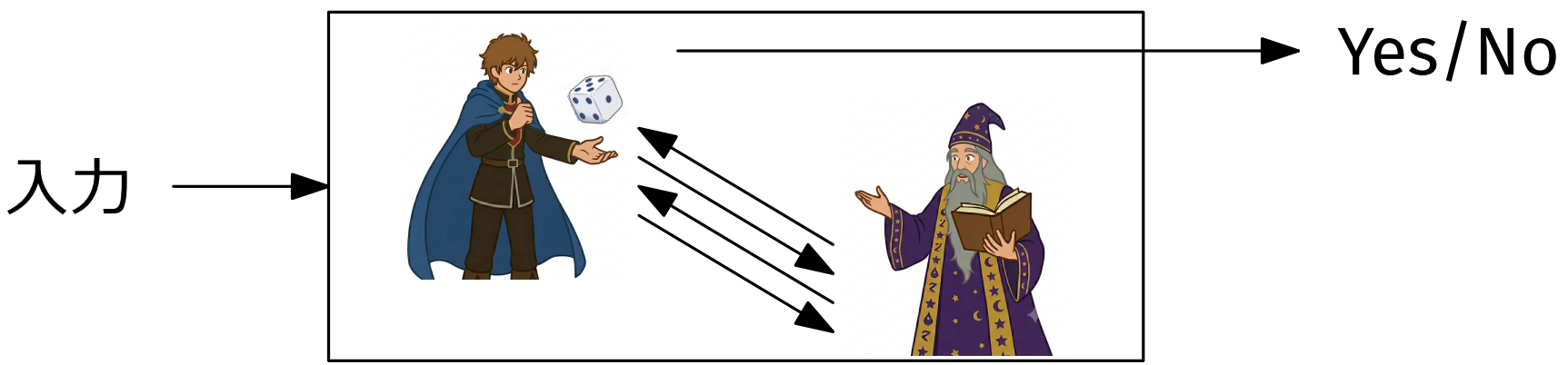
複雑性クラス MA$[k]$ (Babai '85)

クラス MA$[k]$ は次を満たすプロトコルで解ける問題全体

- プロトコルは 証明者 の動作から始まる
- 検証者と証明者が順に計 k ラウンド の動作を行う
- 検証者の乱数は証明者から見える (public coin)

MA = Merlin-Arthur

MA$[4]$ のプロトコル



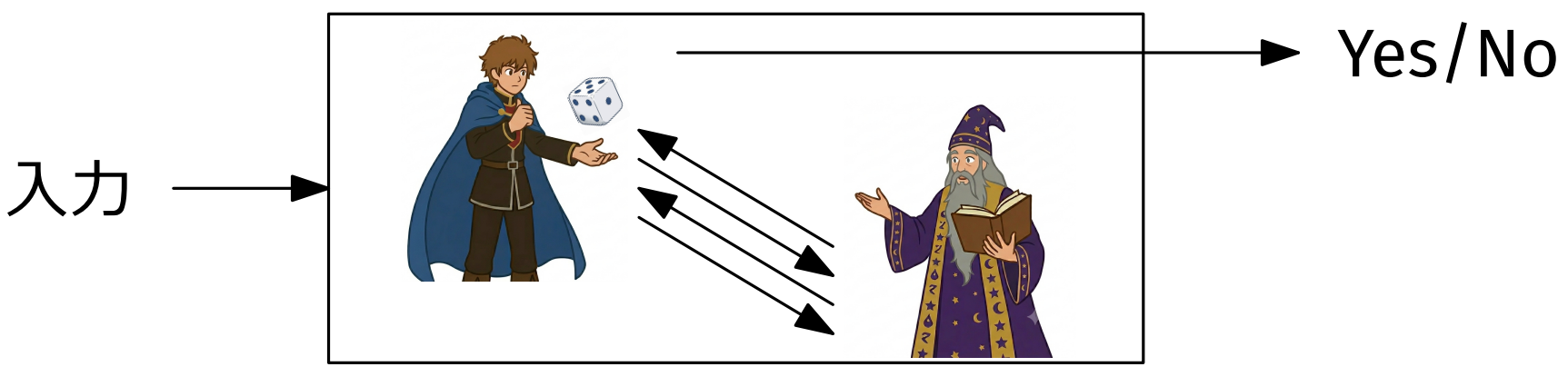
注：ラウンドの数え方

証明者のラウンド１回

- 何かを計算して，検証者にメッセージを送る

検証者のラウンド１回

- 乱数を使って，何かを (多項式時間で) 計算して，
  証明者にメッセージを送る
- 注：出力をするときに，乱数を使わなかったら，
  それはラウンドに数えない



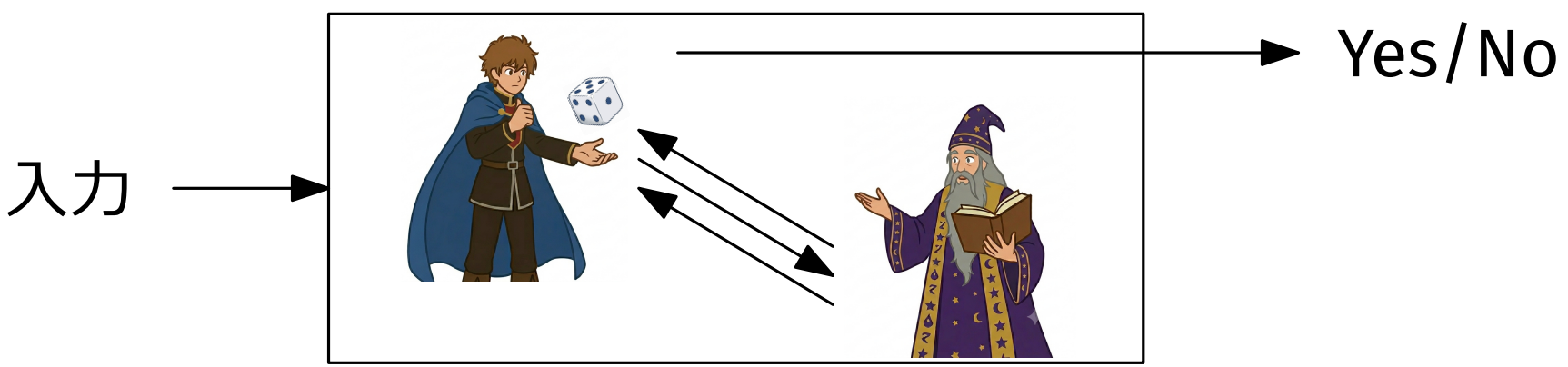
注：ラウンドの数え方

証明者のラウンド１回

- 何かを計算して，検証者にメッセージを送る

検証者のラウンド１回

- 乱数を使って，何かを (多項式時間で) 計算して，
  証明者にメッセージを送る
- 注：出力をするときに，乱数を使わなかったら，
  それはラウンドに数えない



複雑性クラス IP$[k]$

正整数 k

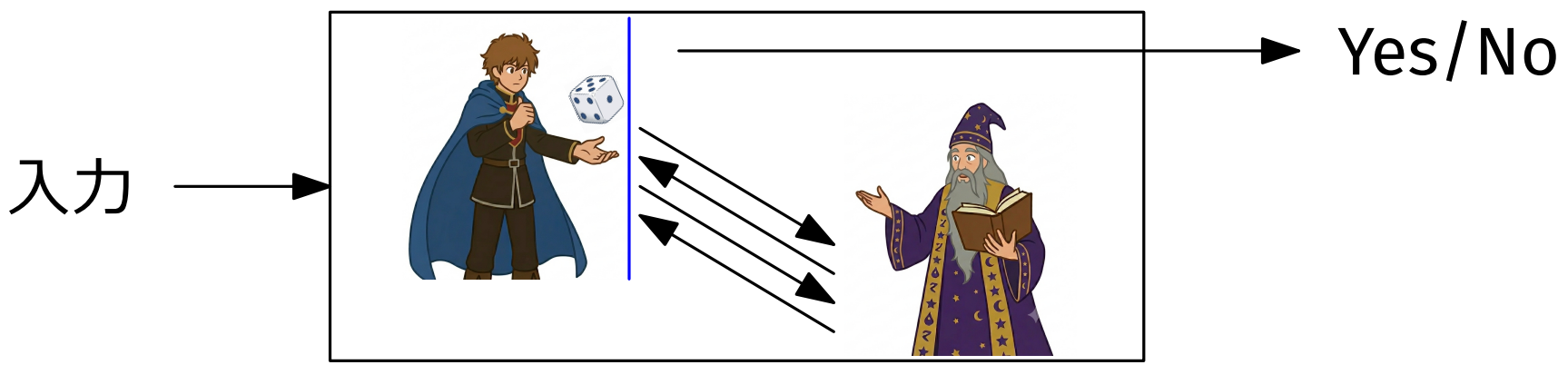
複雑性クラス IP$[k]$ (Goldwasser, Micali, Rackoff '89)

クラス IP$[k]$ は次を満たすプロトコルで解ける問題全体

- プロトコルは 検証者 の動作から始まる
- 検証者と証明者が順に計 k ラウンド の動作を行う
- 検証者の乱数は証明者から 見えない (private coin)

IP = Interactive Proof

IP$[4]$ のプロトコル



AM$[k]$, MA$[k]$, IP$[k]$

正整数 k はラウンド数

	はじめるのは	乱数は証明者から
AM$[k]$	検証者	見える
MA$[k]$	証明者	見える
IP$[k]$	検証者	見えない

AM$[k]$, MA$[k]$, IP$[k]$

正整数 k はラウンド数

	はじめるのは	乱数は証明者から
AM$[k]$	検証者	見える
MA$[k]$	証明者	見える
IP$[k]$	検証者	見えない

定義からすぐに分かる包含関係

- AM$[k] \subseteq$ AM$[k+1]$, MA$[k] \subseteq$ MA$[k+1]$, IP$[k] \subseteq$ IP$[k+1]$
- AM$[k] \subseteq$ MA$[k+1]$, MA$[k] \subseteq$ AM$[k+1]$, AM$[k] \subseteq$ IP$[k]$

AM$[k]$, MA$[k]$, IP$[k]$

正整数 k はラウンド数

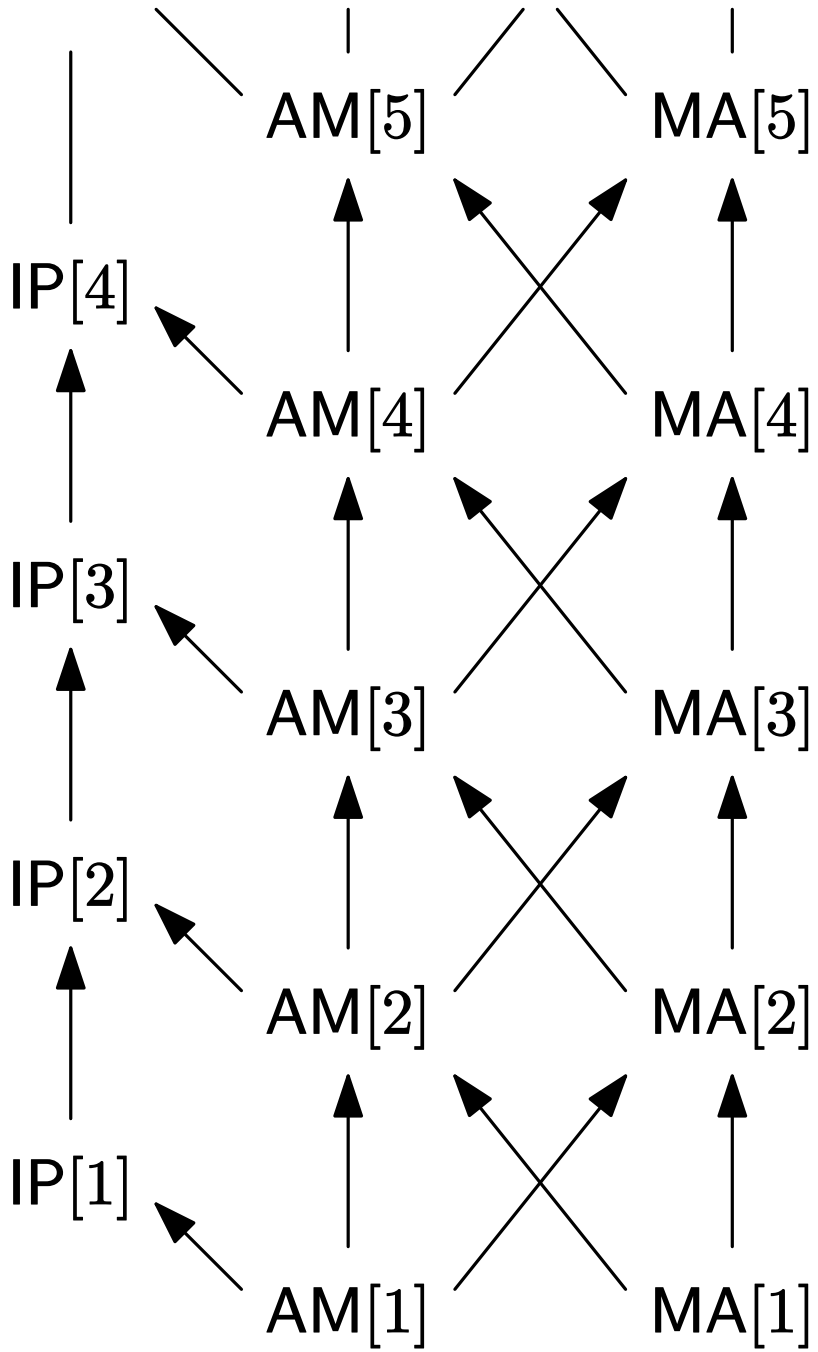
	はじめるのは	乱数は証明者から
AM$[k]$	検証者	見える
MA$[k]$	証明者	見える
IP$[k]$	検証者	見えない

乱数が証明者に見えないと
証明者は嘘をつきにくい
(検証者は成功しやすい)

定義からすぐに分かる包含関係

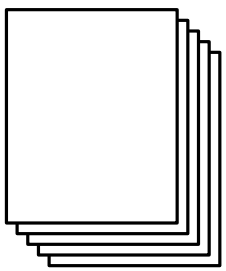
- AM$[k] \subseteq$ AM$[k+1]$, MA$[k] \subseteq$ MA$[k+1]$, IP$[k] \subseteq$ IP$[k+1]$
- AM$[k] \subseteq$ MA$[k+1]$, MA$[k] \subseteq$ AM$[k+1]$, AM$[k] \subseteq$ IP$[k]$

複雑性クラスの関係：ここまでの話



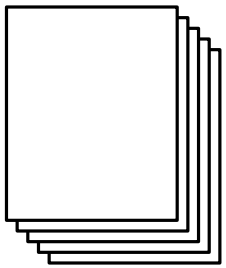
先生と生徒の例：MA[2]

1. レポートを提出する



先生と生徒の例：MA[2]

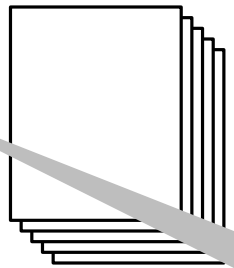
1. レポートを提出する



2. サイコロを振って，
   読むページを決める

先生と生徒の例：MA[2]

1. レポートを提出する



2. サイコロを振って，
   読むページを決める

合格/不合格

先生と生徒の例：AM[2]

問1	問2	問3	問4
問5	問6	問7	問8

先生と生徒の例：AM[2]

1. サイコロで解く問題を決める

問1	問2	問3	問4
問5	問6	問7	問8

先生と生徒の例：AM[2]

1. サイコロで解く問題を決める

問1	問2	問3	問4
問5	問6	問7	問8

2. 解答を提出する

先生と生徒の例：AM[2]

1. サイコロで解く問題を決める

問1　問2　問3　問4

合格/不合格

5　6　問7　問8

2. 解答を提出する

先生と生徒の例：IP[2]

先生と生徒の例：IP[2]

1. サイコロを振って問題を作り，送る

$$\begin{cases} x+y=1 \\ x-y=3 \end{cases}$$

$x=2$

$y=-1$

先生と生徒の例：IP[2]

1. サイコロを振って問題を作り，送る

$$\begin{cases} x+y=1 \\ x-y=3 \end{cases}$$

$x=2$

$y=-1$

2. 解答を提出する

先生と生徒の例：IP[2]

1. サイコロを振って問題を作り，送る

$$\begin{cases} x+y=1 \\ x-y=3 \end{cases}$$

$x=2$

$y=-1$

2. 解答を提出する

合格/不合格

本日の内容

1. 対話証明系
2. **複雑性クラスの関係**

複雑性クラスの関係：知られていること

簡単なもの (今から証明も紹介)

- $\mathsf{MA}[1] = \mathsf{NP}$
- $\mathsf{AM}[1] = \mathsf{IP}[1] = \mathsf{BPP}$
- $\mathsf{MA}[k] \subseteq \mathsf{AM}[k]$ (for $k \geq 2$)

難しいもの (事実だけ紹介)

- $\mathsf{AM}[k] = \mathsf{AM}[k+1] = \mathsf{MA}[k+1]$ (for $k \geq 2$) (Babai '85)
- $\mathsf{AM}[k] = \mathsf{AM}[2k]$ (for $k \geq 2$) (Babai, Moran '88)
- $\mathsf{IP}[k] \subseteq \mathsf{AM}[k+2]$ (Goldwasser, Sipser '87)

複雑性クラスの関係：知られていること

簡単なもの (今から証明も紹介)

- $\mathsf{MA}[1] = \mathsf{NP}$
- $\mathsf{AM}[1] = \mathsf{IP}[1] = \mathsf{BPP}$
- $\mathsf{MA}[k] \subseteq \mathsf{AM}[k]$ (for $k \geq 2$)

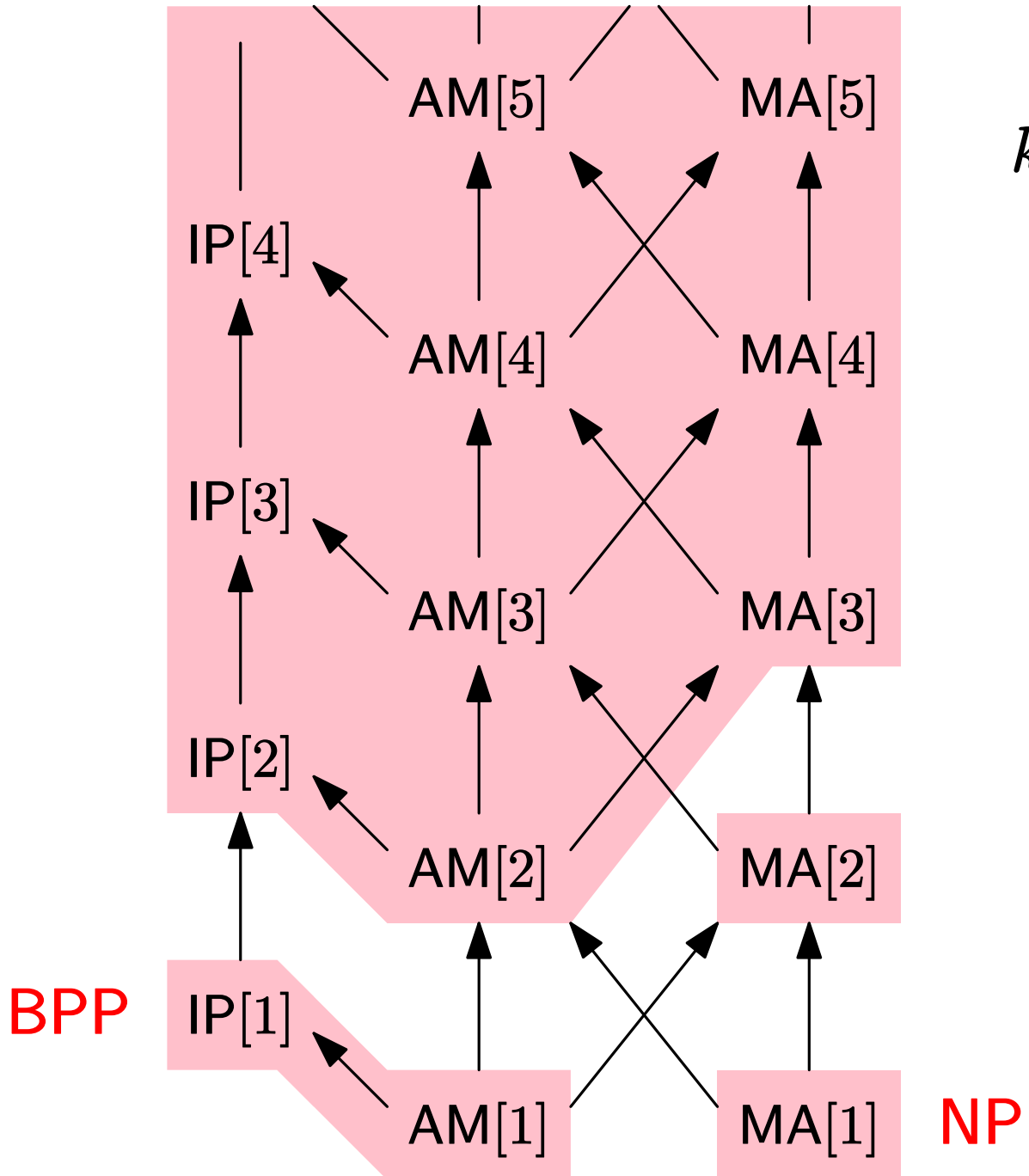
難しいもの (事実だけ紹介)

- $\mathsf{AM}[k] = \mathsf{AM}[k+1] = \mathsf{MA}[k+1]$ (for $k \geq 2$) (Babai '85)
- $\mathsf{AM}[k] = \mathsf{AM}[2k]$ (for $k \geq 2$) (Babai, Moran '88)
- $\mathsf{IP}[k] \subseteq \mathsf{AM}[k+2]$ (Goldwasser, Sipser '87)

特に，定数 $k \geq 2$ に対して

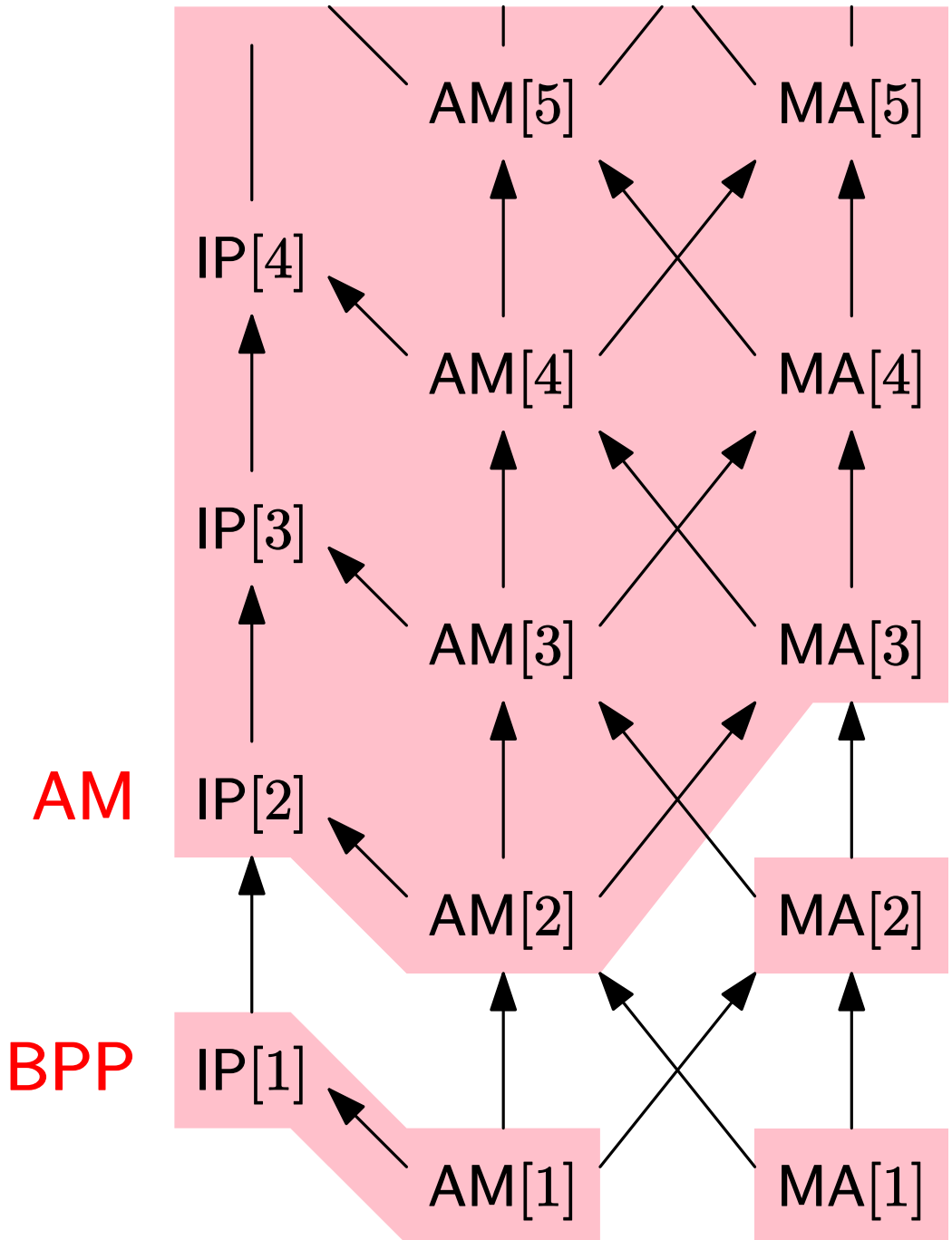
- $\mathsf{AM}[2] = \mathsf{AM}[k] = \mathsf{MA}[k+1] \subseteq \mathsf{AM}[k+2] = \mathsf{AM}[2]$
- $\mathsf{IP}[2] \subseteq \mathsf{IP}[k] \subseteq \mathsf{AM}[k+2] = \mathsf{AM}[2] \subseteq \mathsf{IP}[2]$

複雑性クラスの関係：崩壊



k が定数である限り

複雑性クラスの関係：崩壊



定義

- AM := AM[2]
- MA := MA[2]

性質：まとめ

- NP ⊆ MA ⊆ AM
- BPP ⊆ MA ⊆ AM

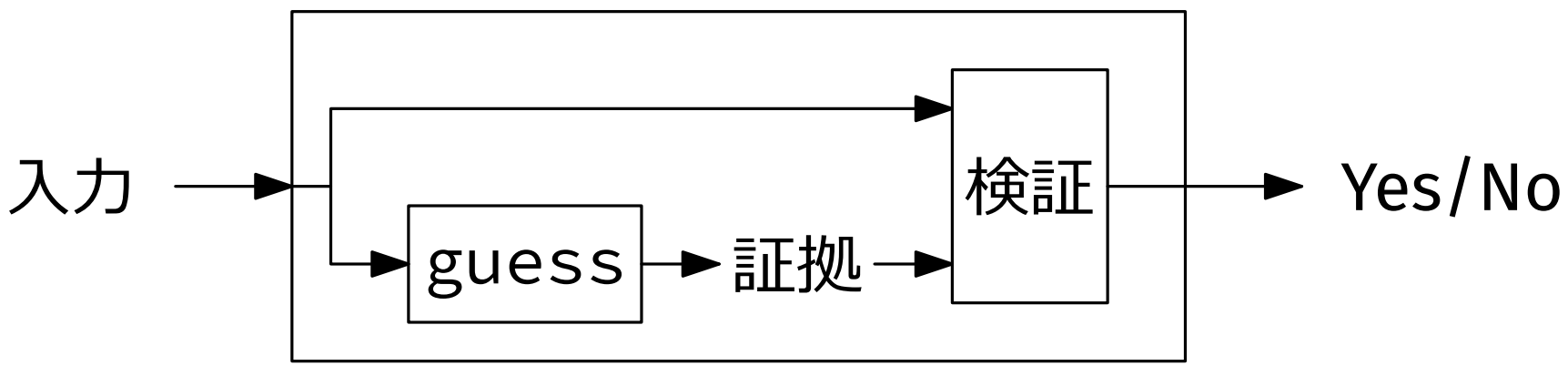
「=」の成立は未解決

NP = MA[1]

性質

NP = MA[1]

証明：NP に対する「証拠・検証」の視点を使う □

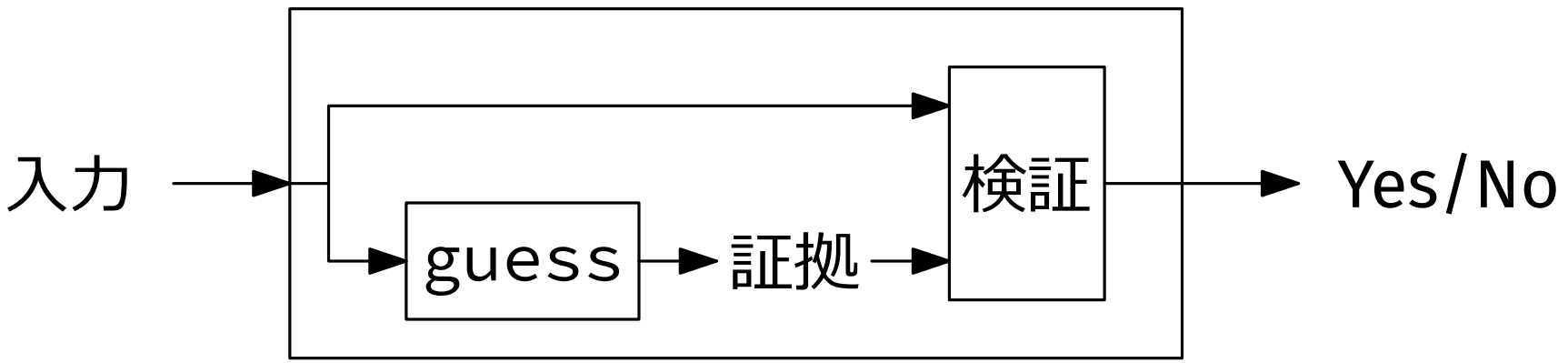


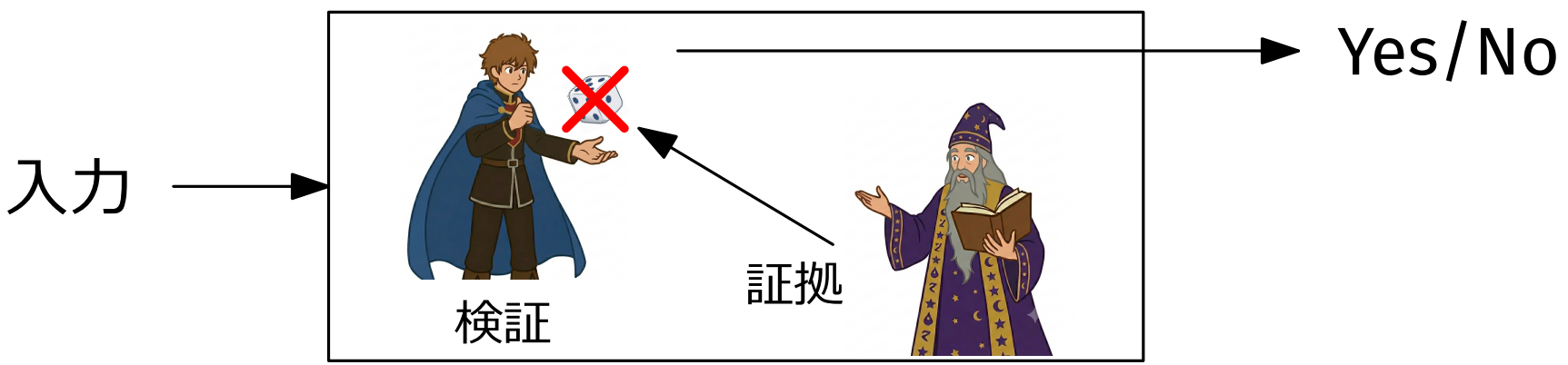
NP = MA[1]

性質

NP = MA[1]

証明：NP に対する「証拠・検証」の視点を使う □





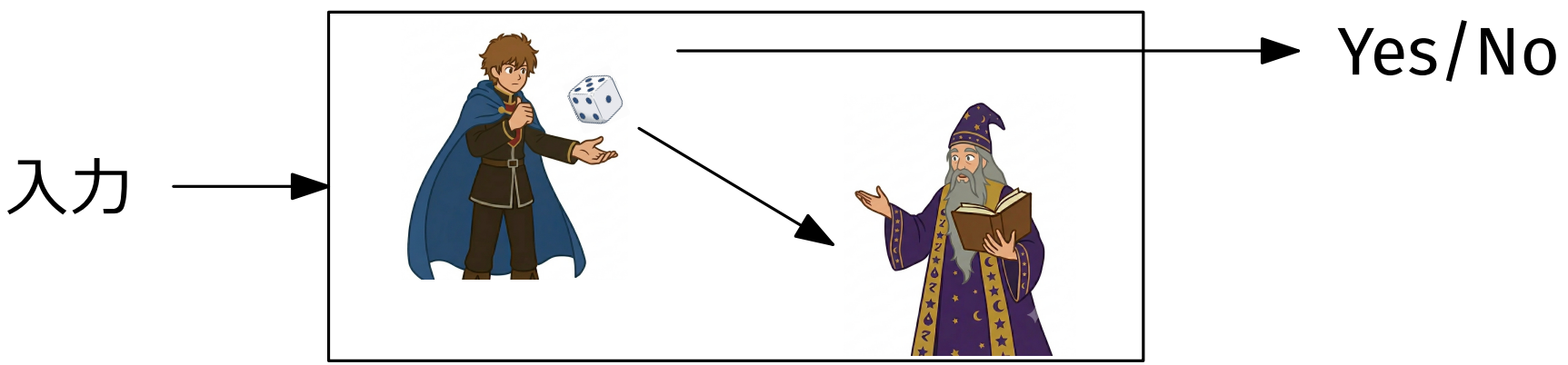
$\mathsf{BPP} = \mathsf{AM}[1] = \mathsf{IP}[1]$

性質

$\mathsf{BPP} = \mathsf{AM}[1] = \mathsf{IP}[1]$

証明：検証者は証明者のメッセージを使わずに判定する

- $\therefore$ どれも，検証者のみが多項式時間で
  成功確率 $\geq 2/3$ を実現する □



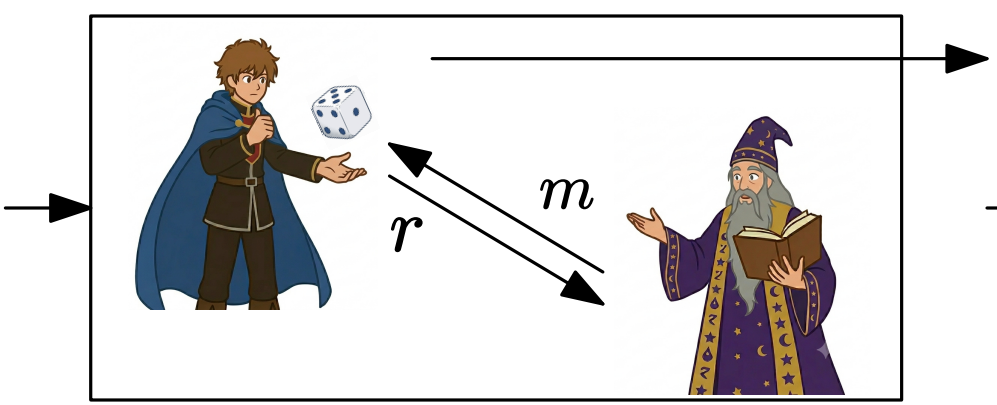
$\mathsf{MA}[k] \subseteq \mathsf{AM}[k]$ (1)

性質 (Babai '85)

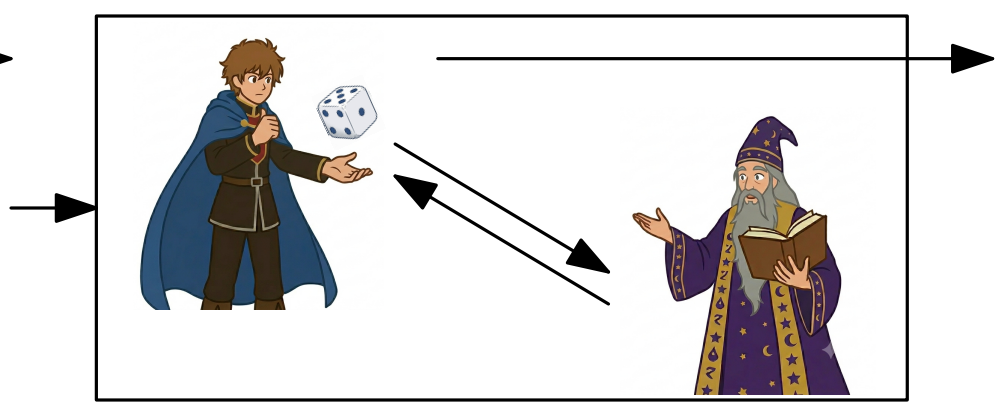
任意の $k \geq 2$ に対して，$\mathsf{MA}[k] \subseteq \mathsf{AM}[k]$

証明 ($k = 2$ の場合)：$P \in \mathsf{MA}$ とする，すなわち

- Yes インスタンス $\Rightarrow$
  $\exists$ 証明者のメッセージ m : $\Pr_r$(検証者が Yes を出力) $\geq 2/3$
- No インスタンス $\Rightarrow$
  $\forall$ 証明者のメッセージ m : $\Pr_r$(検証者が No を出力) $\geq 2/3$



$\mathsf{MA}[2]$ ($= \mathsf{MA}$)



$\mathsf{AM}[2]$ ($= \mathsf{AM}$)

$\mathsf{MA}[k] \subseteq \mathsf{AM}[k]$ (2)

証明の考え方

MA プロトコル (成功確率 $\geq 2/3$)

↓ 確率増幅

MA プロトコル (成功確率 $\geq 1 - \varepsilon$)

↓ 動作順の反転

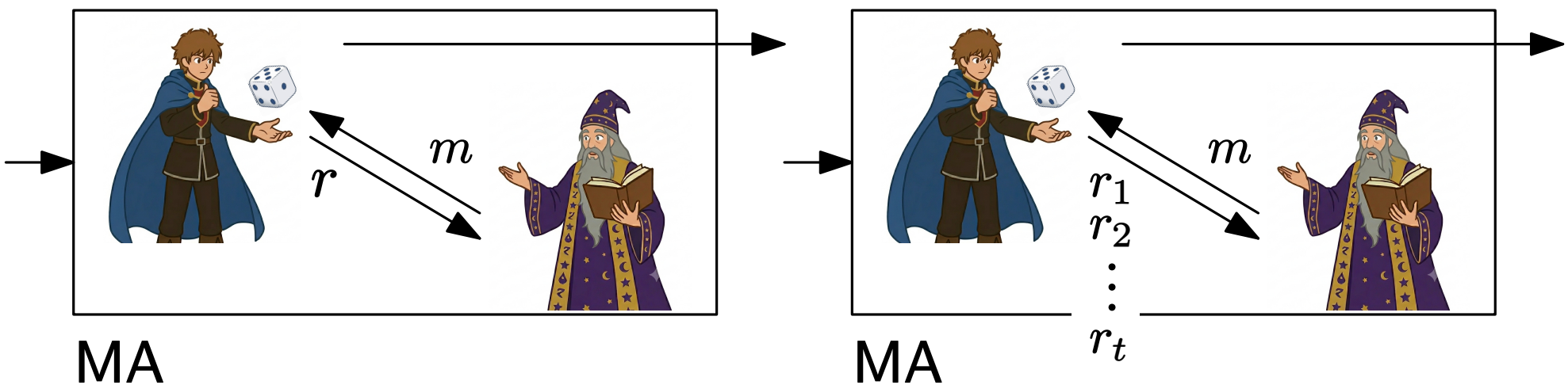
AM プロトコル (成功確率 $\geq 2/3$)

$\mathrm{MA}[k] \subseteq \mathrm{AM}[k]$ (3)

復習：BPP と確率増幅

$P \in \mathrm{BPP} \Rightarrow$ 任意の多項式 p に対して，
次を満たす確率的アルゴリズム A が存在

- P の任意の入力 I に対して，
  - 必ず多項式時間で停止する
  - $\Pr_r(A\text{の出力が正しい}) \geq 1 - \frac{2}{3}\left(\frac{8}{9}\right)^{p(|I|)}$

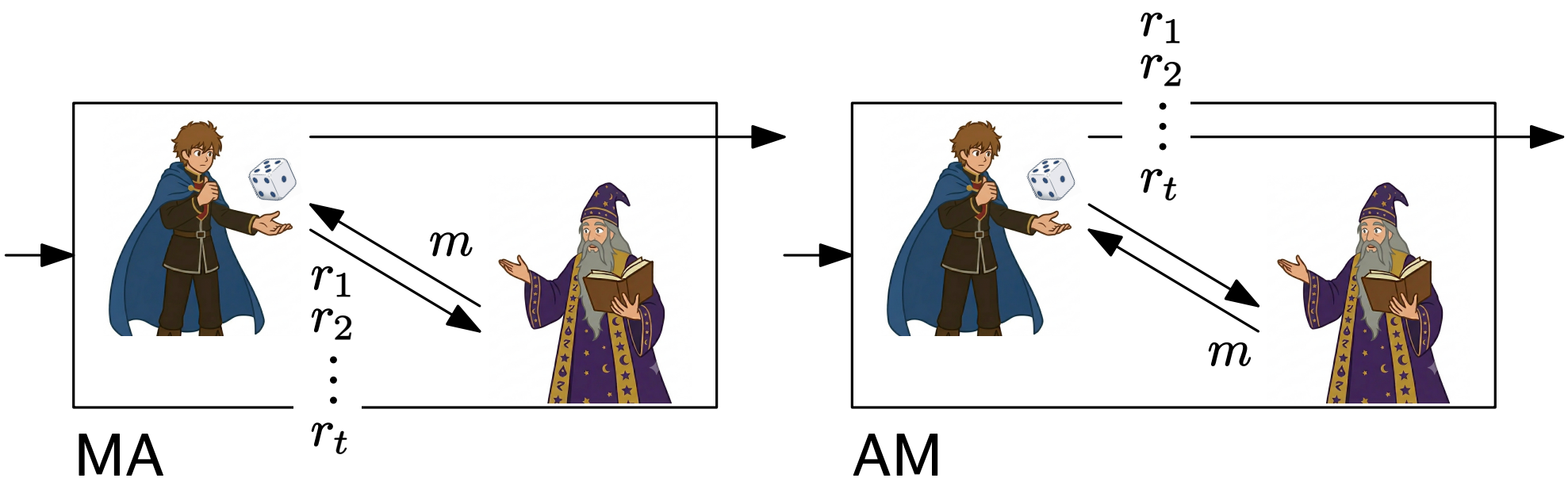


$\mathrm{MA}[k] \subseteq \mathrm{AM}[k]$ (4)

AM プロトコル

1. 検証者：ランダム文字列 $r_1, \ldots, r_t$ を作り，証明者に送る
2. 証明者：メッセージ m を作り，検証者に送る
3. 検証者：m と r_i を使って元の MA プロトコルのとおりに Yes/No の判定を行い $(i = 1, \ldots, t)$，t 個の判定結果の多数決で，Yes/No を出力する

t が多項式 $\Rightarrow$ 検証者は多項式時間アルゴリズム



MA$[k] \subseteq$ AM$[k]$ (5)

AM プロトコル

1. 検証者：ランダム文字列 $r_1, \ldots, r_t$ を作り，証明者に送る
2. 証明者：メッセージ m を作り，検証者に送る
3. 検証者：m と r_i を使って元の MA プロトコルのとおりに Yes/No の判定を行い $(i = 1, \ldots, t)$，t 個の判定結果の多数決で，Yes/No を出力する

Yes インスタンス $\Rightarrow$

- m を「元の MA プロトコルで証明者が送るもの」とする
- $\therefore \Pr(\text{成功}) \geq \dfrac{2}{3}$

MA[k] $\subseteq$ AM[k] (6)

AM プロトコル

1. 検証者：ランダム文字列 $r_1, \ldots, r_t$ を作り，証明者に送る
2. 証明者：メッセージ m を作り，検証者に送る
3. 検証者：m と r_i を使って元の MA プロトコルのとおりに Yes/No の判定を行い ($i = 1, \ldots, t$)，t 個の判定結果の多数決で，Yes/No を出力する

No インスタンス $\Rightarrow$

- m をメッセージとする
  (メッセージ長 $|m|$ は多項式で一定と仮定)
- $\Pr(\exists m:\ \text{失敗}) \leq \sum_m \Pr(m \text{ で失敗}) \leq 2^{|m|} \frac{2}{3} \left(\frac{8}{9}\right)^t \leq \frac{1}{3}$
- $\therefore \Pr(\forall m:\ \text{成功}) \geq \frac{2}{3}$

$t = 6|m|$

□

$\mathsf{MA}[k] \subseteq \mathsf{AM}[k]$ (7)

一般の $k \geq 3$ に対して

m

r

m'

r'

m''

r''

$\mathsf{MA}[k] \subseteq \mathsf{AM}[k]$ (7)

一般の $k \geq 3$ に対して

$r_1, r_2, \ldots, r_t$

m

m'

r'

m''

r''

$\mathsf{MA}[k] \subseteq \mathsf{AM}[k]$ (7)

一般の $k \geq 3$ に対して

$r_1, r_2, \ldots, r_t$

m, m'

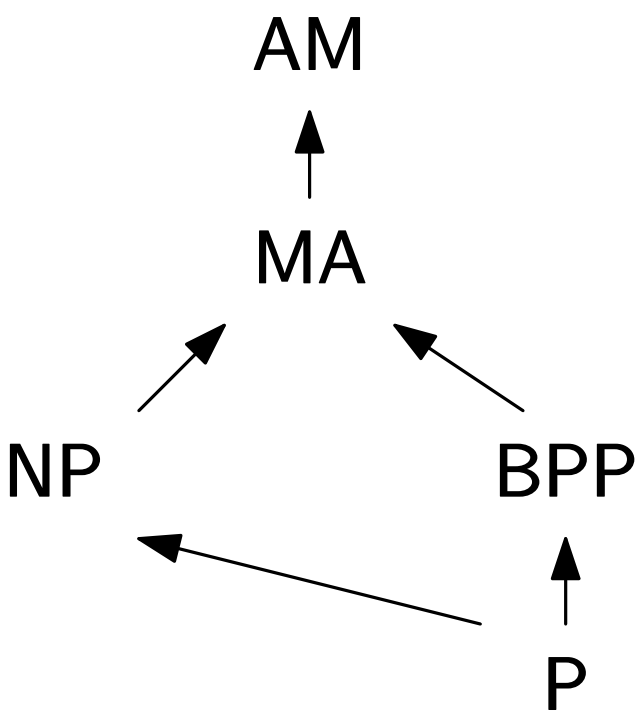
r'

m''

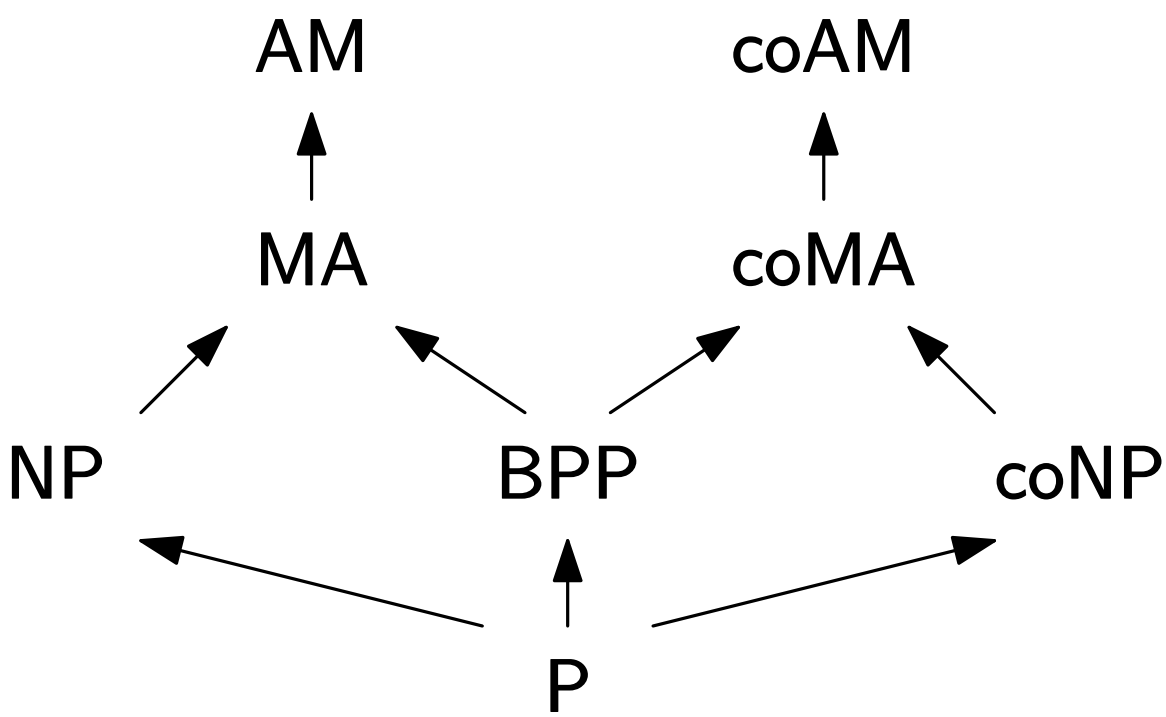
r''

つまり，$\mathsf{MA}[k] \subseteq \mathsf{AM}[k-1] \subseteq \mathsf{AM}[k]$ □

複雑性クラスの関係 (続)



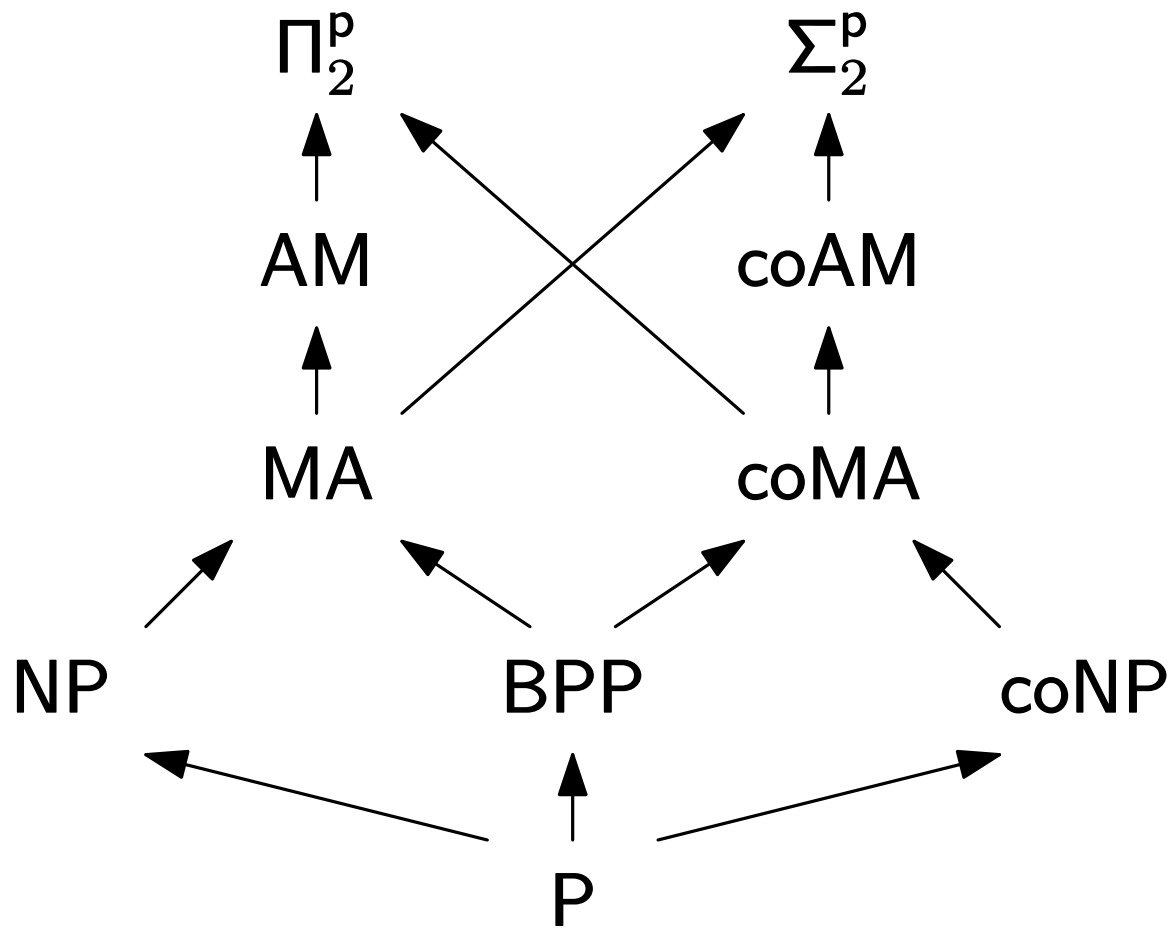
複雑性クラスの関係 (続)



複雑性クラスの関係 (続)

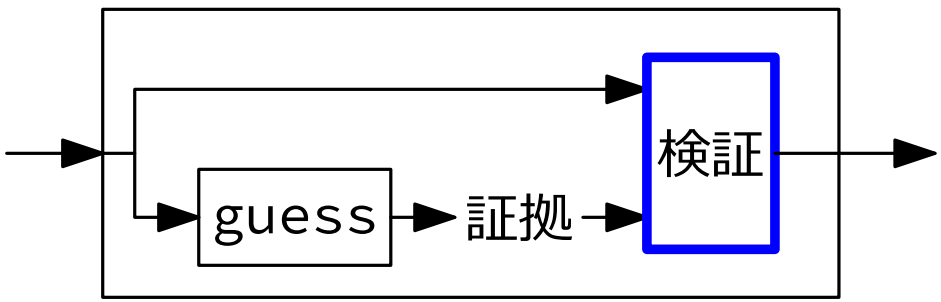
Sipser-Lautemann の定理 ($\mathrm{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$) の証明と同様に，次も正しいことが分かる

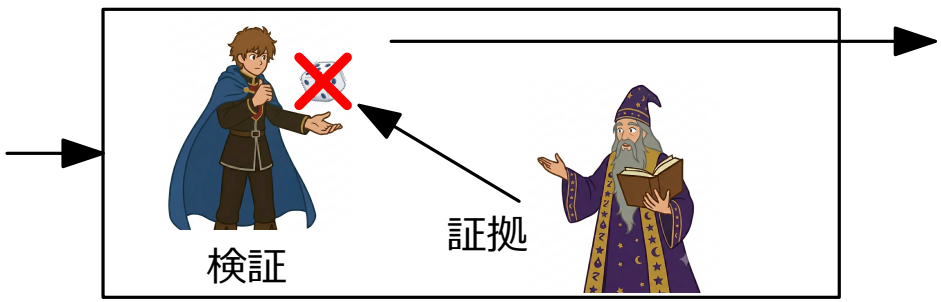
- $\mathrm{MA} \subseteq \Sigma_2^p \cap \Pi_2^p$
- $\mathrm{AM} \subseteq \Pi_2^p$



補足：MA は NP の確率版

NP　＝　MA[1]

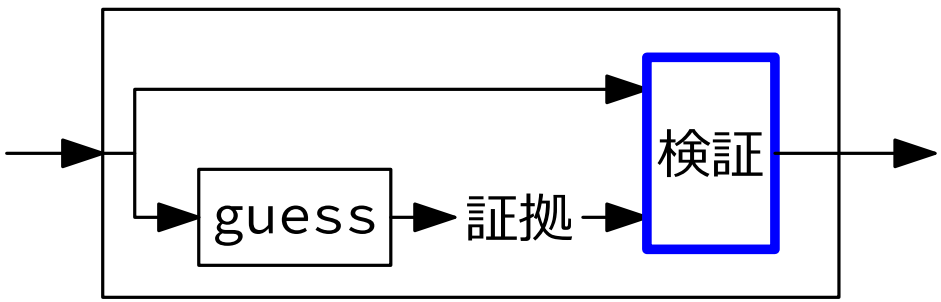


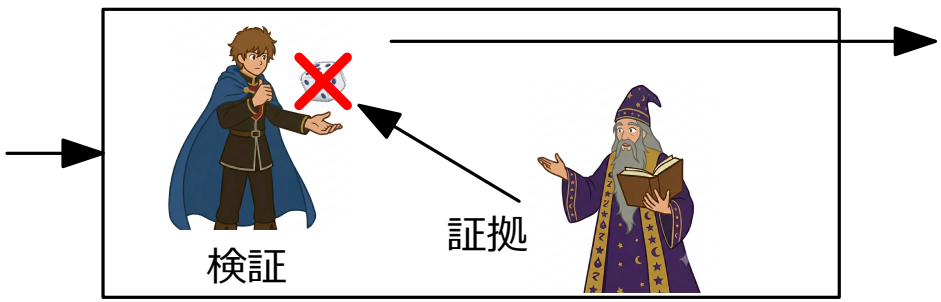


P のアルゴリズム

補足：MA は NP の確率版

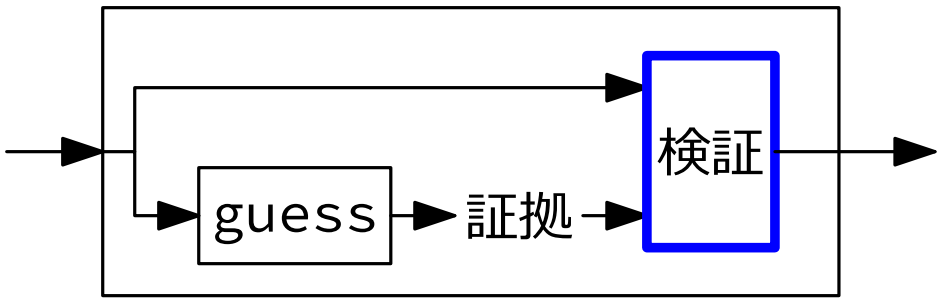
NP = MA[1]

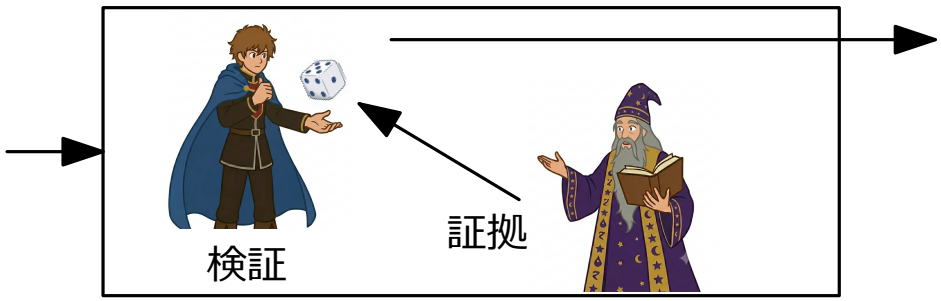




P のアルゴリズム

NP の確率版 = MA (= MA[2])





BPP のアルゴリズム

本日のまとめ

内容

- 対話証明系の計算モデルを導入する
- 複雑性クラス AM$[k]$，MA$[k]$，IP$[k]$ を導入する
- いままで登場した計算複雑性クラスとの関係を調べる

次回と次々回の予告

内容

- 次を証明する

$$\mathsf{IP} = \mathsf{PSPACE}$$

- 技法：代数的手法

ここで，$\mathsf{IP} = \bigcup_{k=1}^{\infty} \mathsf{IP}[n^k]$　(ただし，n は入力の符号長)

クイズ

Q

1.

2.

3.

4.