非有界個のセッションの下で計算論的に健全な記号的匿名性

川本 裕輔[1 2]

with Hubert Comon-Lundh[3], 萩谷 昌己[1], 櫻田 英樹[4]

[1] 東京大学大学院情報理工学系研究科
[2] 日本学術振興会特別研究員
[3] ENS Cachan
[4] NTT コミュニケーション科学基礎研究所

暗号プロトコルの検証手法

	形式的手法	計算論的手法
参加者 & 攻撃者	記号的プロセス	確率的多項式時間Turing機械
メッセージ	Dolev-Yao項（記号表現）	ビット列
攻撃者の操作	いくつかの関数記号	任意の確率的多項式時間計算

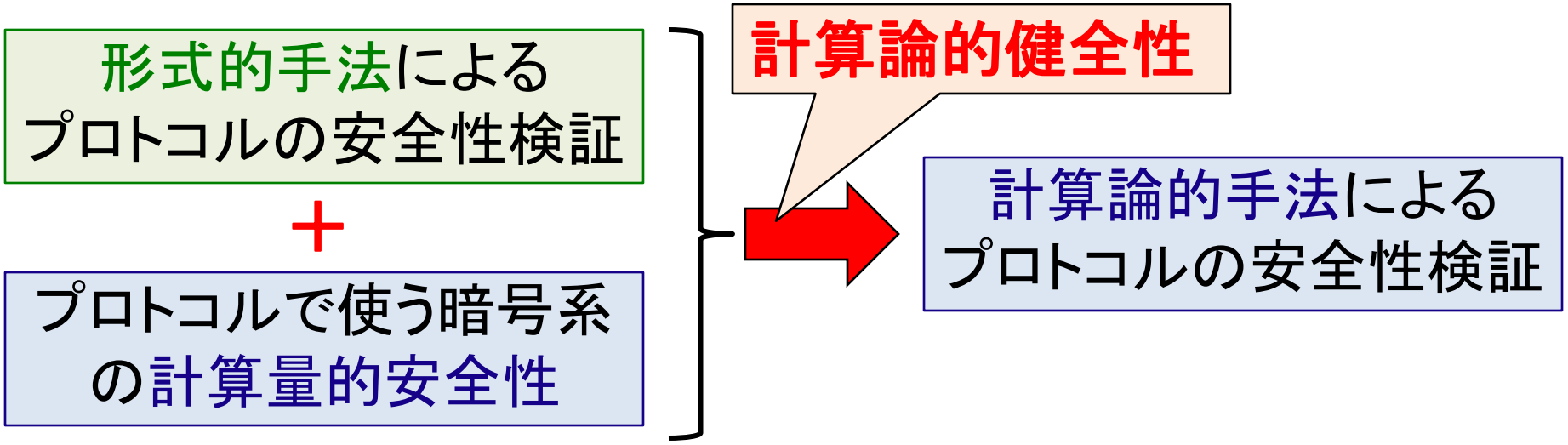
暗号プロトコルの検証手法

	形式的手法	計算論的手法
参加者 & 攻撃者	記号的プロセス	確率的多項式時間Turing機械
メッセージ	Dolev-Yao項（記号表現）	ビット列
攻撃者の操作	いくつかの関数記号	任意の確率的多項式時間計算
安全性の仮定	理想的な暗号の安全性	計算量理論に基づく暗号の安全性
プロトコル検証	現実世界に忠実でない 単純で自動化しやすい	現実世界に忠実である 複雑で間違えやすい

（以後のスライドでは，**緑色**は形式的手法，**青色**は計算論的手法）

形式的手法の計算論的健全性

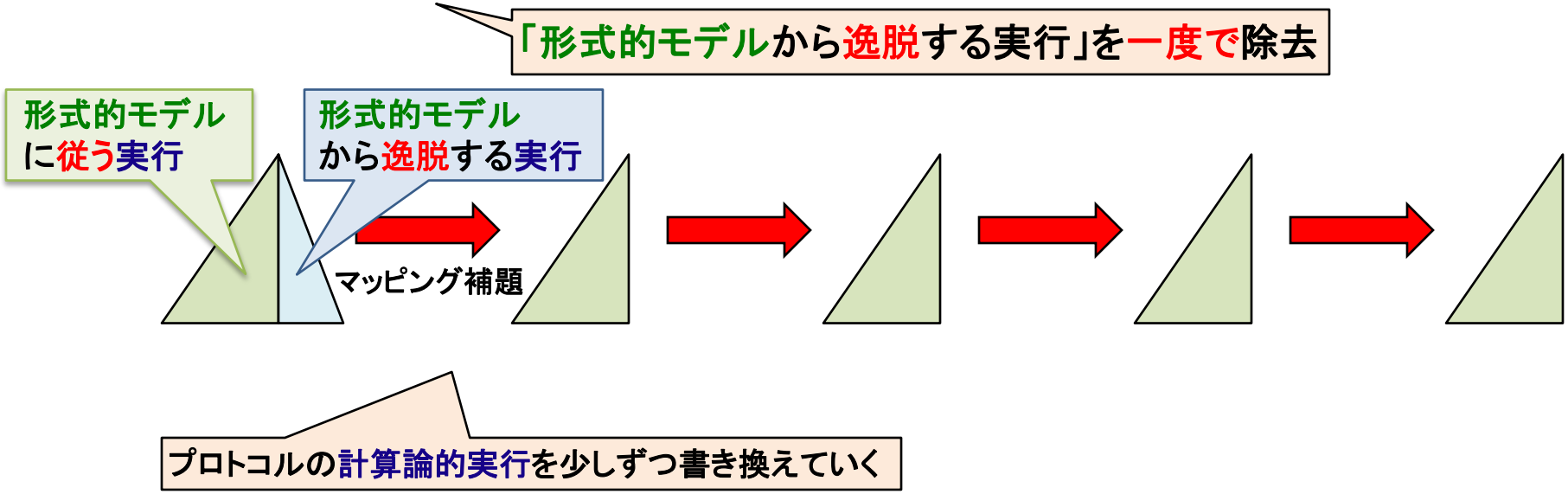
[Abadi,Rogaway'00], [Baudet,Cortier,Kremer'05], [Comon,Cortier'08] ……



- 計算論的健全性は形式的手法の既存技術を活用する上で重要.

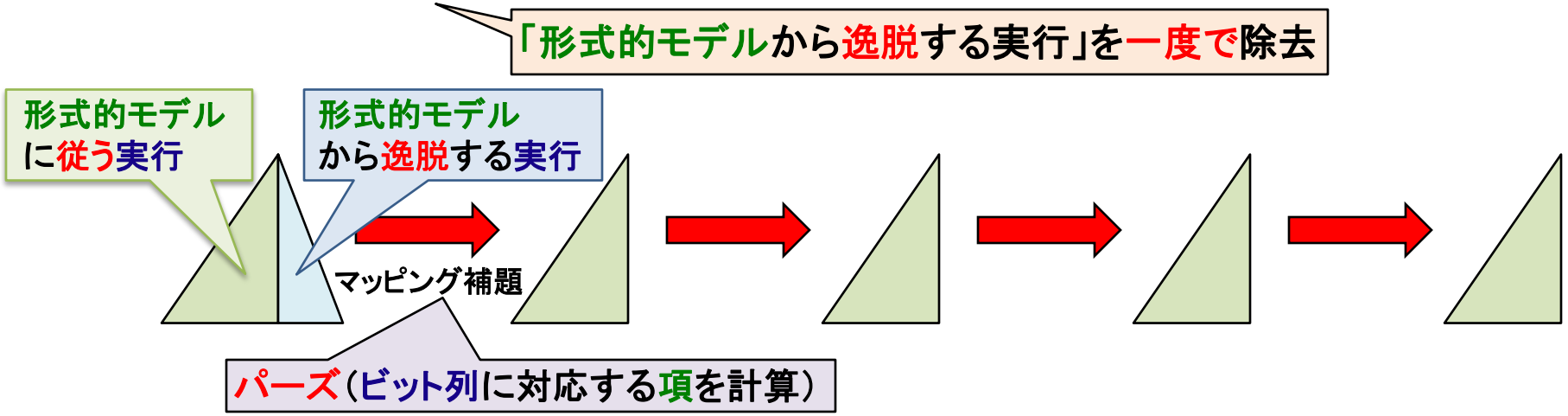
計算論的健全性の先行研究の問題点

計算論的健全性の証明 [Comon, Coriter'08]



計算論的健全性の先行研究の問題点

計算論的健全性の証明 [Comon, Coriter'08]



- **暗号文**: 復号して平文を取り出すと，対応する項を計算できる．
- **ハッシュ値**: 元のメッセージを取り出せないため，対応する項を計算できない．

[Comon, Coriter'08]の証明技法では，ハッシュ関数を扱えない

本研究の成果

- ハッシュ関数と公開鍵暗号と電子署名(リング署名)を扱える **applied pi-calculus**の観測同値の計算論的健全性を証明

暗号プロトコルの安全性検証に広く使われている形式的手法のひとつ [Abadi, Fournet'01]

匿名性の表現に有用

  - 先行研究は対称鍵暗号の場合[Comon, Coriter'08]だけ

  - 新たな証明技法を導入し，より多くの暗号系を扱えるようになった
    - 先行研究では，能動的攻撃者の下で計算論的健全性を証明するには，多項式時間計算可能なパーズ(ビット列から項を得る操作)が必要だったため，(ランダムオラクル以外の)ハッシュ関数を扱えなかった.
    - 本研究の証明では，多項式時間計算可能なパーズを必要としない新たな証明技法を導入することで，ハッシュ関数も扱えるようになった.

本研究の成果

- ハッシュ関数と公開鍵暗号と電子署名(リング署名)を扱える **applied pi-calculus**の観測同値の計算論的健全性を証明

  計算論的モデルでは多項式個

  - 能動的攻撃者と非有界個のセッション
    - 先行研究は対称鍵暗号の場合[Comon, Coriter'08]だけ
  - 新たな証明技法を導入し，より多くの暗号系を扱えるようになった.
    - 先行研究では，能動的攻撃者の下で計算論的健全性を証明するには，多項式時間計算可能なパーズ(ビット列から項を得る操作)が必要だったため，(ランダムオラクル以外の)ハッシュ関数を扱えなかった.
    - 本研究の証明では，多項式時間計算可能なパーズを必要としない新たな証明技法を導入することで，ハッシュ関数も扱えるようになった.

概要

- ハッシュ関数を扱える**applied pi-calculus**
- **applied pi-calculus**の計算論的健全性
- 先行研究[Comon,Cortier'08]との比較
- 先行研究[Cortier et.al'06]との比較
- まとめ

計算論的ハッシュ関数の安全性仮定

- 原像計算困難性（preimage-resistance）

$\Pr\left[n \xrightarrow{\text{ランダム}} \boxed{\text{攻撃者}} \xrightarrow{m} : n = \mathrm{h}(m) \right]$ が無視できる

- 衝突困難性（collision-resistance）

$\Pr\left[\boxed{\text{攻撃者}} \xrightarrow{m_1, m_2} : m_1 \neq m_2 \ \&\ \mathrm{h}(m_1) = \mathrm{h}(m_2) \right]$ が無視できる

- 非展性（non-malleability）を仮定しない.

➡ 例えば, $\mathrm{h}(m_1)$ と $\mathrm{h}(m_2)$ から $\mathrm{h}(\langle m_1, m_2 \rangle)$ が生成できるかもしれない.

形式的体系

applied pi-calculus

- Dolev-Yao項　メッセージ

u ::= n | y | $<u, u>$ | h(u) | …

乱数を表す名前

攻撃者のメッセージを表す変数

メッセージuのハッシュ値

他に，暗号文，電子署名など

- 書換規則
  - $u_1, u_2 \rightarrow <u_1, u_2>$
  - $<u_1, u_2> \rightarrow u_i$　　(i = 1, 2)
  - ⋮　他に，暗号文や電子署名に関する規則
  - 任意の導出 $u_1, u_2, \ldots, u_n \rightarrow^* u'$ に対して，

    h(u_1), h(u_2), … , h(u_n) → h(u')　非展性を仮定しないことに対応

    例： h(u_1), h(u_2) → h($<u_1, u_2>$)

形式的体系

applied pi-calculus

- Dolev-Yao項　メッセージ

u ::= n | y | $<u, u>$ | h(u) | …

乱数を表す名前

攻撃者のメッセージを表す変数

メッセージuのハッシュ値

他に，暗号文，電子署名など

- 書換規則
  - $u_1, u_2 \rightarrow <u_1, u_2>$
  - $<u_1, u_2> \rightarrow u_i$　　(i = 1, 2)
  - ⋮　他に，暗号文や電子署名に関する規則
  - 任意の導出 $u_1, u_2, \ldots, u_n \rightarrow^* u'$ に対して，

    h(u_1), h(u_2), … , h(u_n) → h(u')

非展性を仮定しないことに対応

「applied pi-calculusで安全性を証明したら，計算論的モデルでも安全」を示すためには，applied pi-calculus における攻撃者を十分強くする必要がある．

形式的体系

applied pi-calculus

- プロセス

プロトコルの振る舞いを記述

P, Q ::=	$\mathrm{c}(x).P$	input
|	$\overline{\mathrm{c}}(u).P$	output
|	0	terminated process
|	$P \parallel Q$	parallel composition
|	$!P$	replication
|	$(\nu\alpha)P$	restriction
|	$\mathtt{if}\ \Phi\ \mathtt{then}\ P\ \mathtt{else}\ Q$	condition

[Comon & Coriter'08] と異なる

- 条件

Φ, Φ' ::=	$M(u)$	well-formedness
|	$EQ(u, u')$	equality
|	$\neg\, \Phi$	negation
|	$\Phi \wedge \Phi'$	conjunction

概要

- ハッシュ関数を扱えるapplied pi-calculus
- **applied pi-calculus**の計算論的健全性
- 先行研究[Comon,Cortier'08]との比較
- 先行研究[Cortier et.al'06]との比較
- まとめ

applied pi-calculusの計算論的健全性

[Comon, Cortier'08], [本研究]

applied pi-calculusによるプロトコルの匿名性検証

観測同値を証明

$$\Pi_0 \sim \Pi_1$$

Aさんが候補1に投票
Bさんが候補2に投票

Aさんが候補2に投票
Bさんが候補1に投票

定義

$$(\Pi_0 \| A)\downarrow_c \Leftrightarrow (\Pi_1 \| A)\downarrow_c$$

攻撃者のプロセス

攻撃者の観測
（メッセージの送受信）

計算論的手法によるプロトコルの匿名性検証

計算量的識別不能性を証明

$$[[\Pi_0]] \approx [[\Pi_1]]$$

Aさんが候補1に投票
Bさんが候補2に投票

Aさんが候補2に投票
Bさんが候補1に投票

定義

$\Pr[\, [[\Pi_b]]^\tau \| A(r) \rightarrow b' : b' = b \,] - \frac{1}{2}$
が無視できる

applied pi-calculusの計算論的健全性

[Comon, Cortier'08], [本研究]

applied pi-calculusによるプロトコルの匿名性検証

観測同値を証明

$$\Pi_0 \sim \Pi_1$$

Aさんが候補1に投票
Bさんが候補2に投票

Aさんが候補2に投票
Bさんが候補1に投票

計算論的健全性

計算論的手法によるプロトコルの匿名性検証

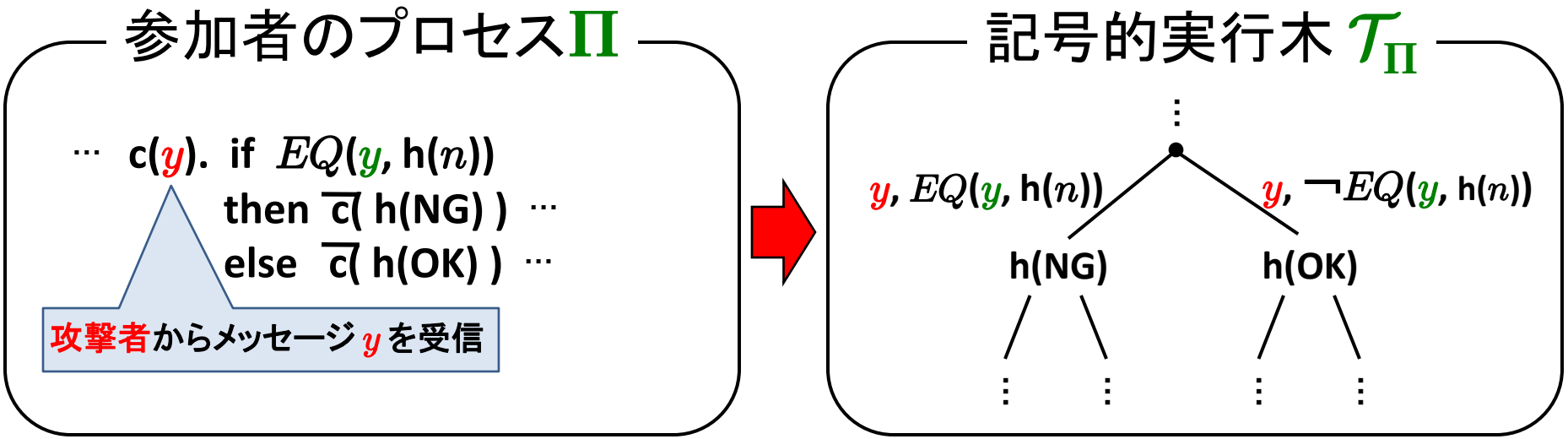
計算量的識別不能性を証明

$$[[\Pi_0]] \approx [[\Pi_1]]$$

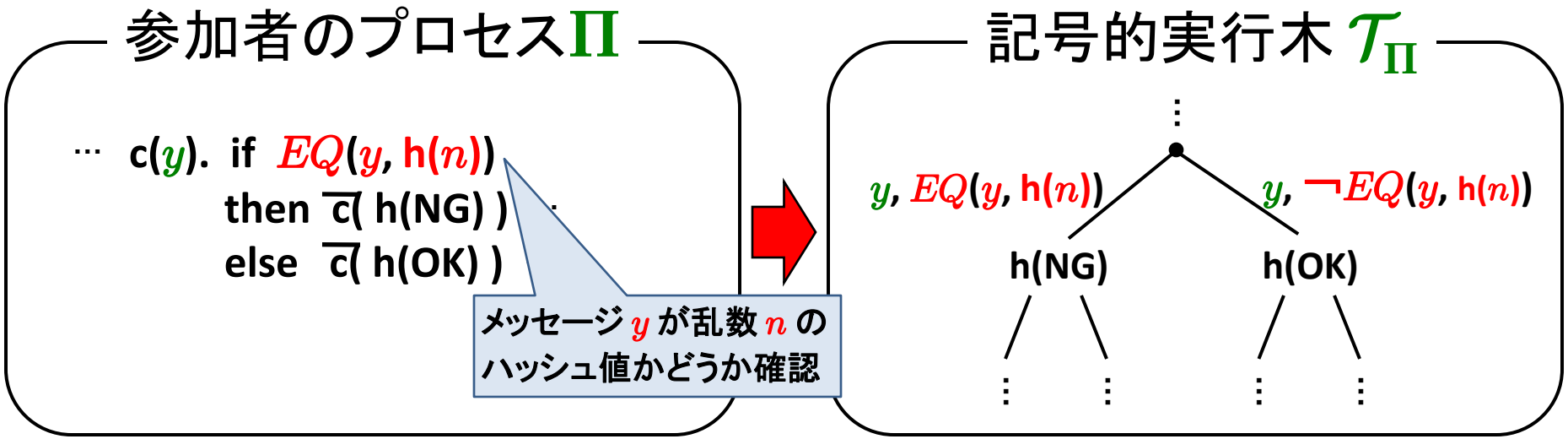
Aさんが候補1に投票
Bさんが候補2に投票

Aさんが候補2に投票
Bさんが候補1に投票

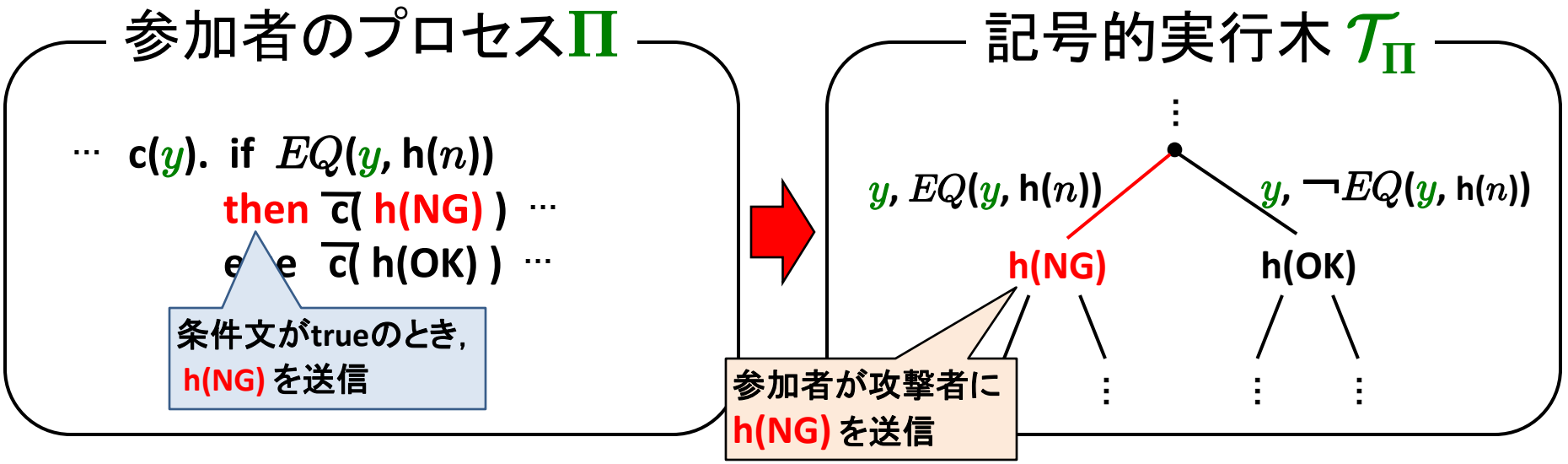
計算論的健全性の証明の道具 [本研究]



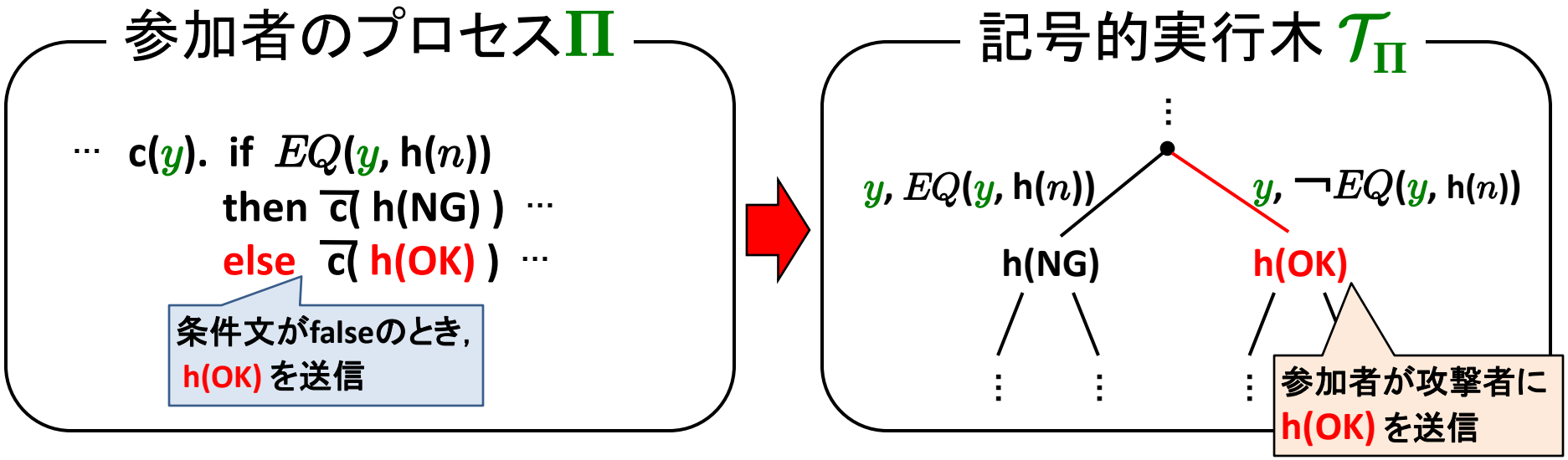
計算論的健全性の証明の道具 [本研究]



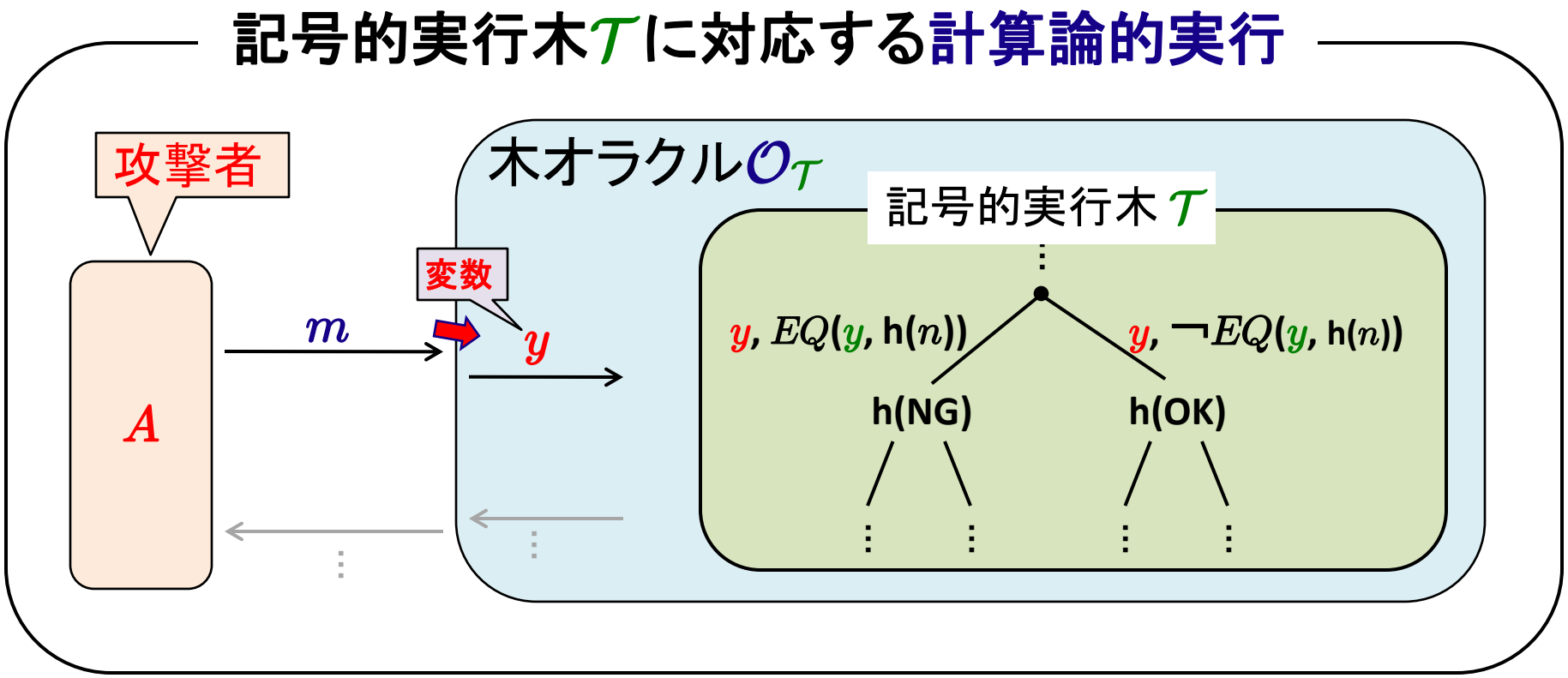
計算論的健全性の証明の道具 [本研究]



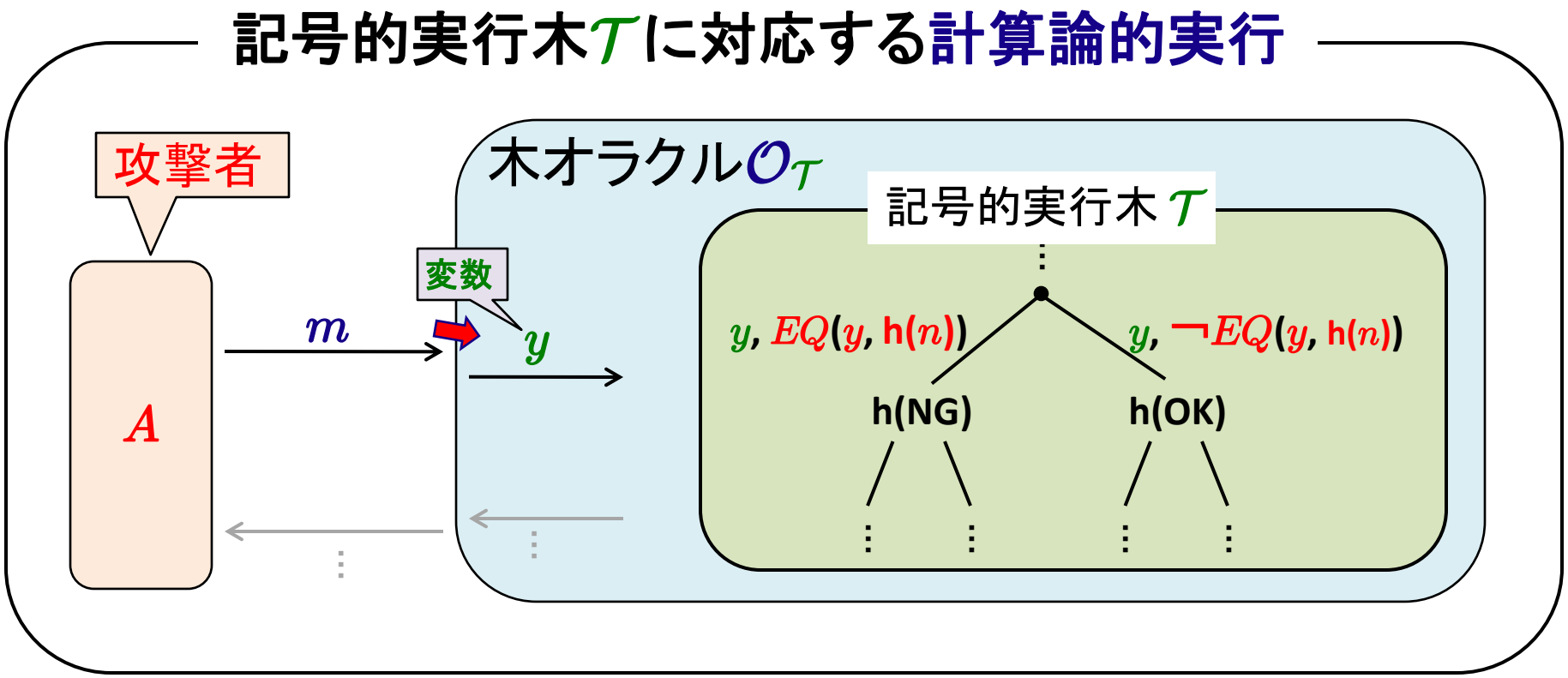
計算論的健全性の証明の道具 [本研究]



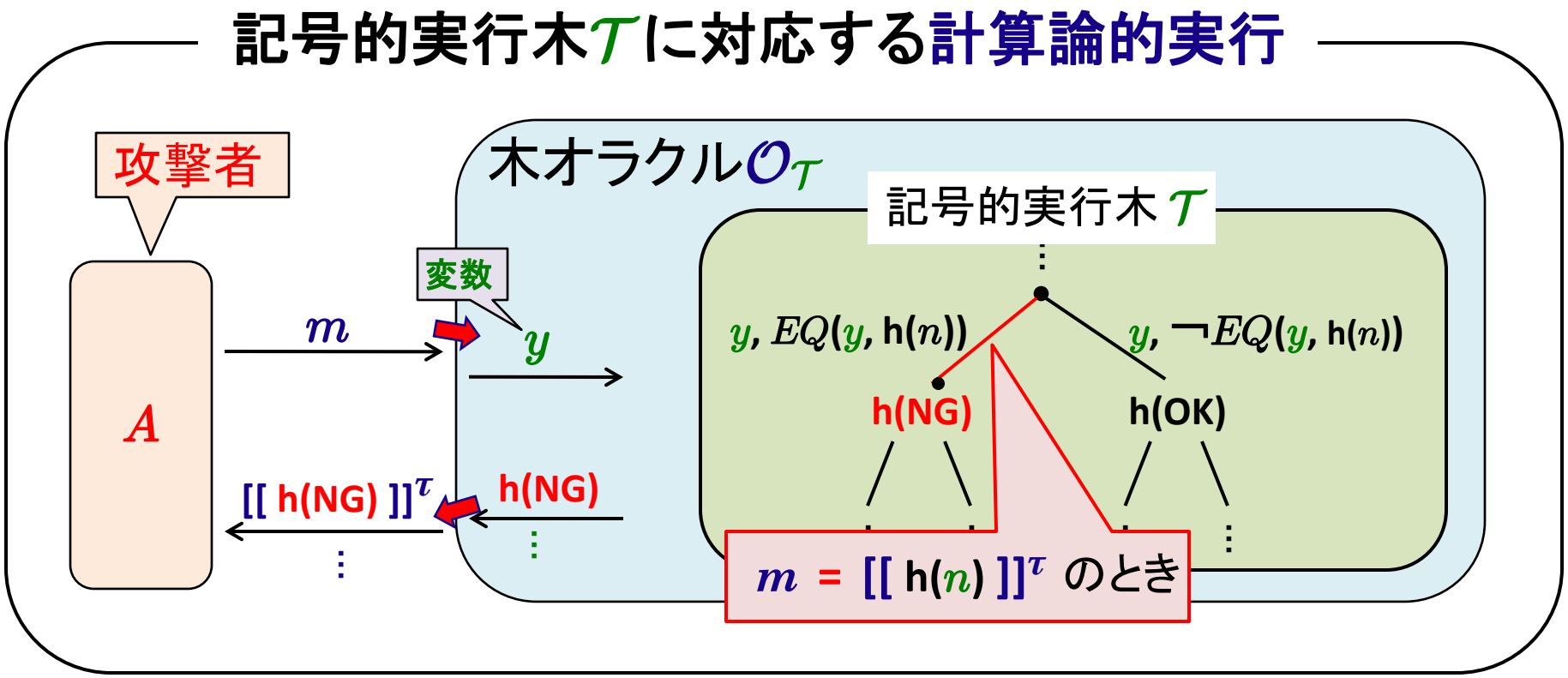
計算論的健全性の証明の道具 [本研究]



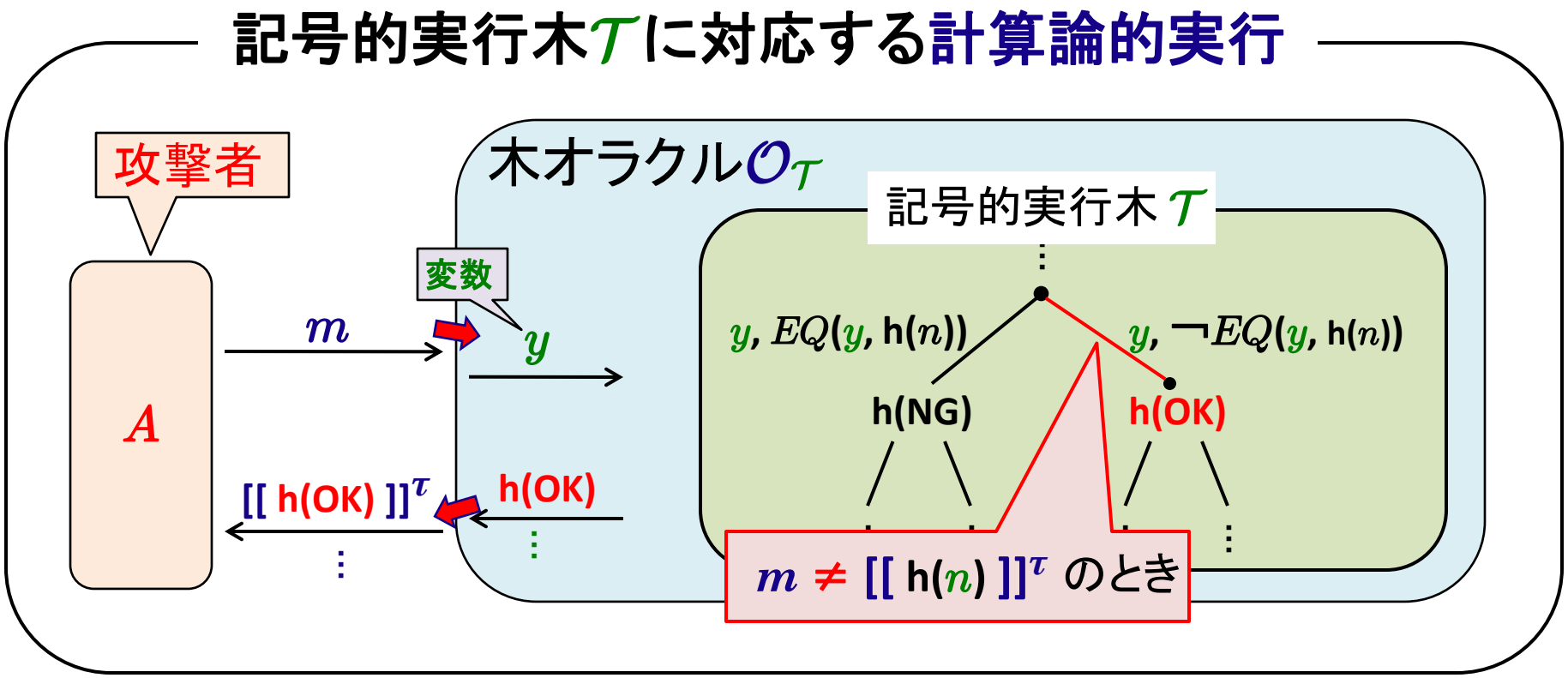
計算論的健全性の証明の道具 [本研究]



計算論的健全性の証明の道具 [本研究]



計算論的健全性の証明の道具 [本研究]

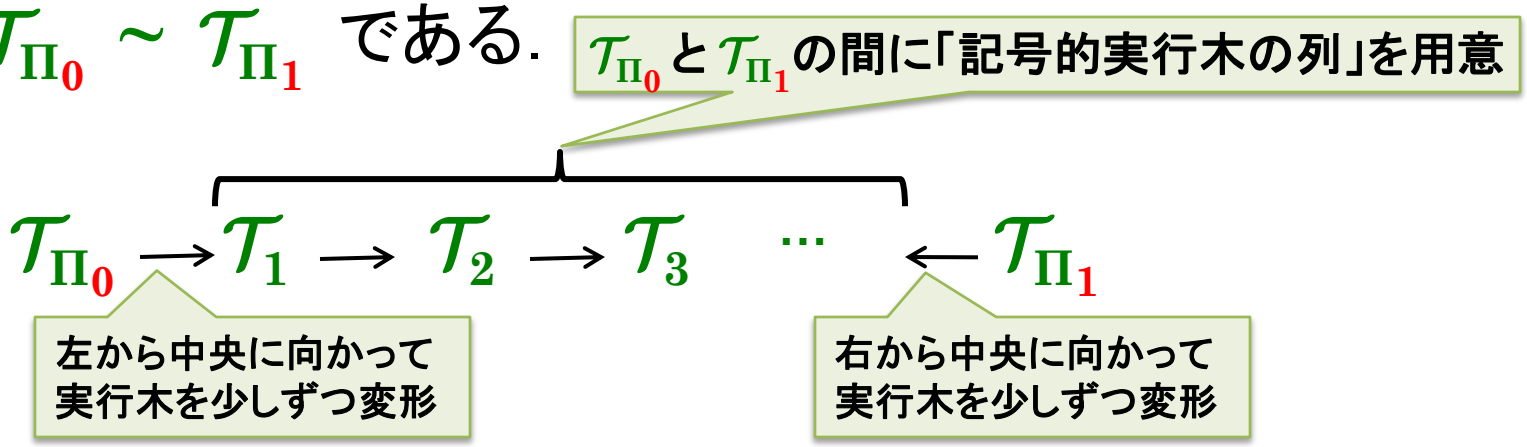


計算論的健全性の証明の概要 [本研究]

- $\Pi_0 \sim \Pi_1$ を仮定する.
  - 観測同値
- すると, $\mathcal{T}_{\Pi_0} \sim \mathcal{T}_{\Pi_1}$ である.
  - 木の構造が等しい

計算論的健全性の証明の概要 [本研究]

- $\Pi_0 \sim \Pi_1$ を仮定する.
- すると, $\mathcal{T}_{\Pi_0} \sim \mathcal{T}_{\Pi_1}$ である.



計算論的健全性の証明の概要 [本研究]

- $\Pi_0 \sim \Pi_1$ を仮定する.

木の構造が等しい

- すると, $\mathcal{T}_{\Pi_0} \sim \mathcal{T}_{\Pi_1}$ である.

$$\mathcal{T}_{\Pi_0} \sim \mathcal{T}_1 \sim \mathcal{T}_2 \sim \mathcal{T}_3 \sim \cdots \sim \mathcal{T}_{\Pi_1}$$

- このとき,

$$[[\Pi_0]] = \mathcal{O}_{\mathcal{T}_{\Pi_0}} \approx \mathcal{O}_{\mathcal{T}_1} \approx \mathcal{O}_{\mathcal{T}_2} \approx \mathcal{O}_{\mathcal{T}_3} \approx \cdots \approx \mathcal{O}_{\mathcal{T}_{\Pi_1}} = [[\Pi_1]]$$

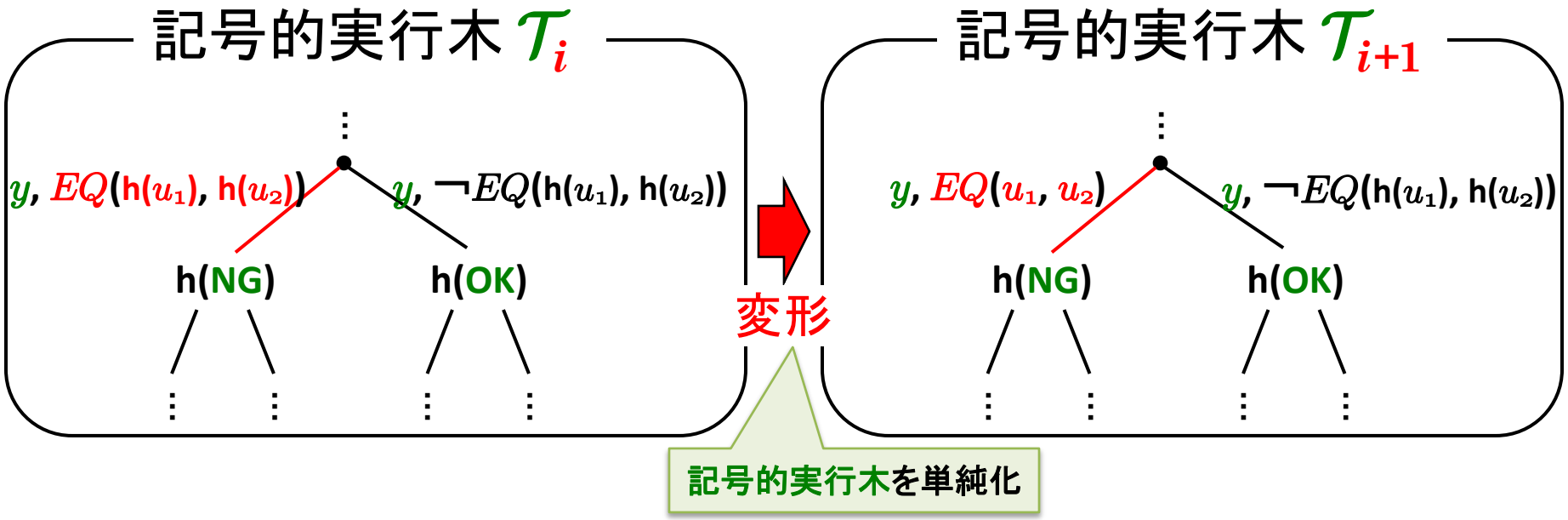
を示したい.

Π_0の計算論的実行 $[[\Pi_0]]$ と全く同じに振舞う.

Π_1の計算論的実行 $[[\Pi_1]]$ と全く同じに振舞う.

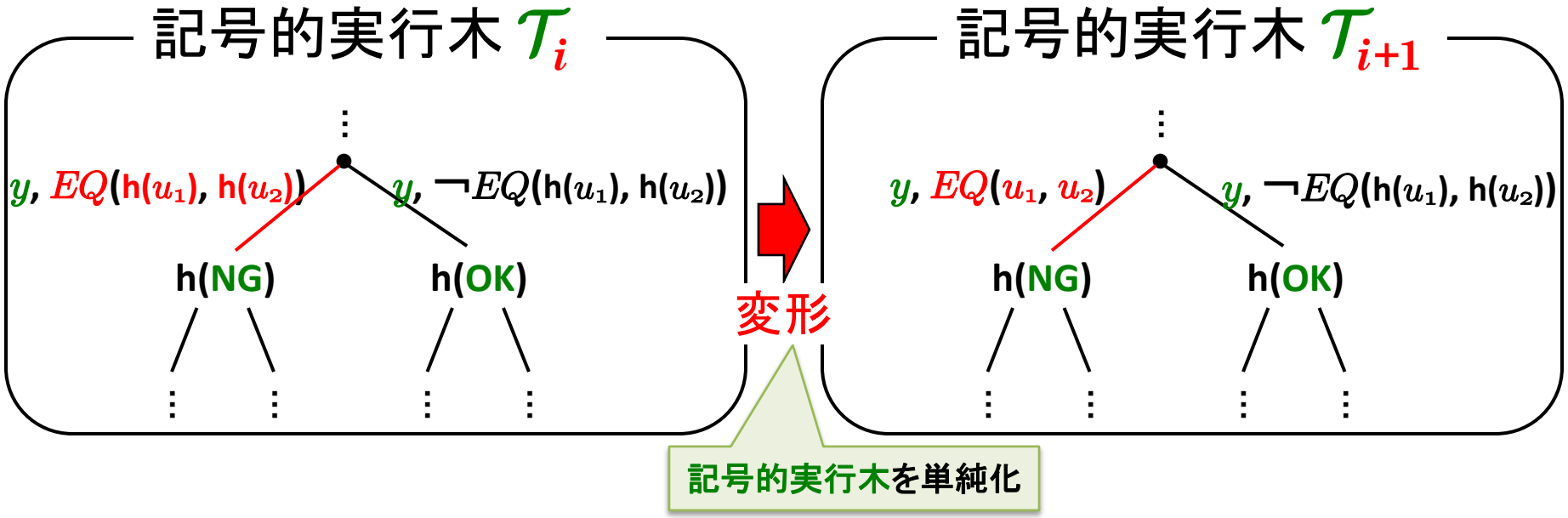
計算論的健全性の証明の概要 [本研究]

変形例1



計算論的健全性の証明の概要 [本研究]

変形例1の前後の識別不能性を示す



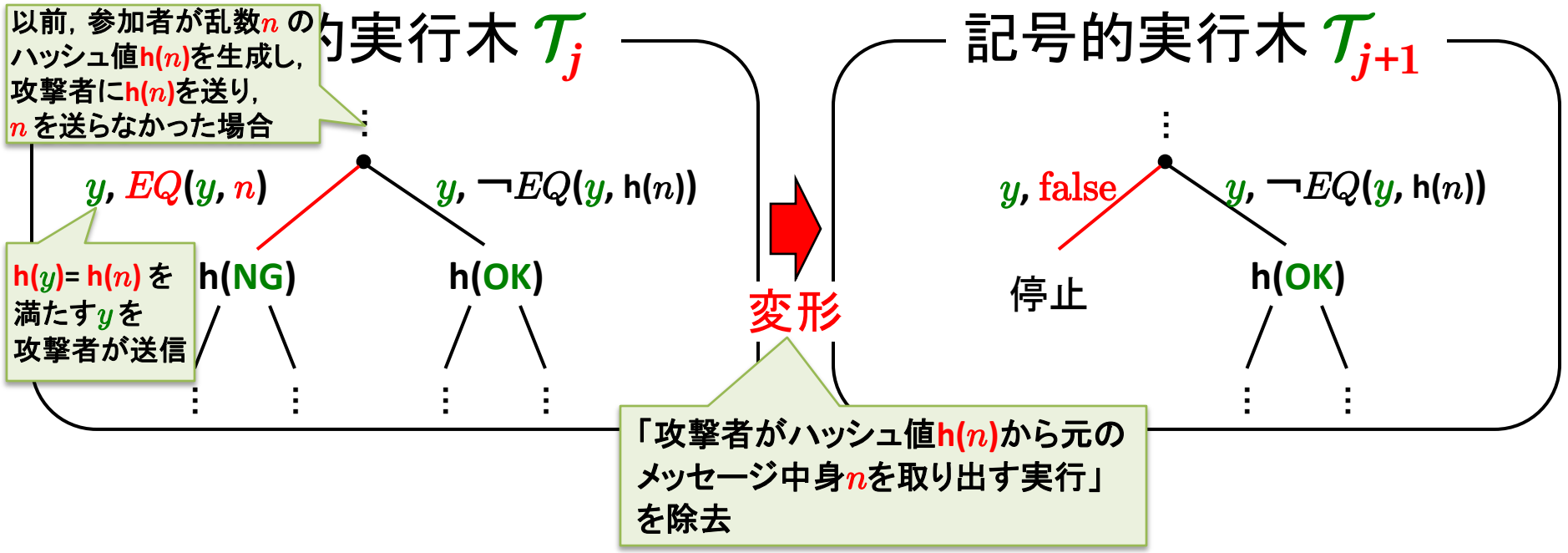
変形前 $\mathcal{O}_{\mathcal{T}_i}$ ≈ $\mathcal{O}_{\mathcal{T}_{i+1}}$ 変形後

ハッシュ関数の衝突困難性

圧倒的な確率で,
[[h(u_1)]] = [[h(u_2)]]
ならば[[u_1]] = [[u_2]]

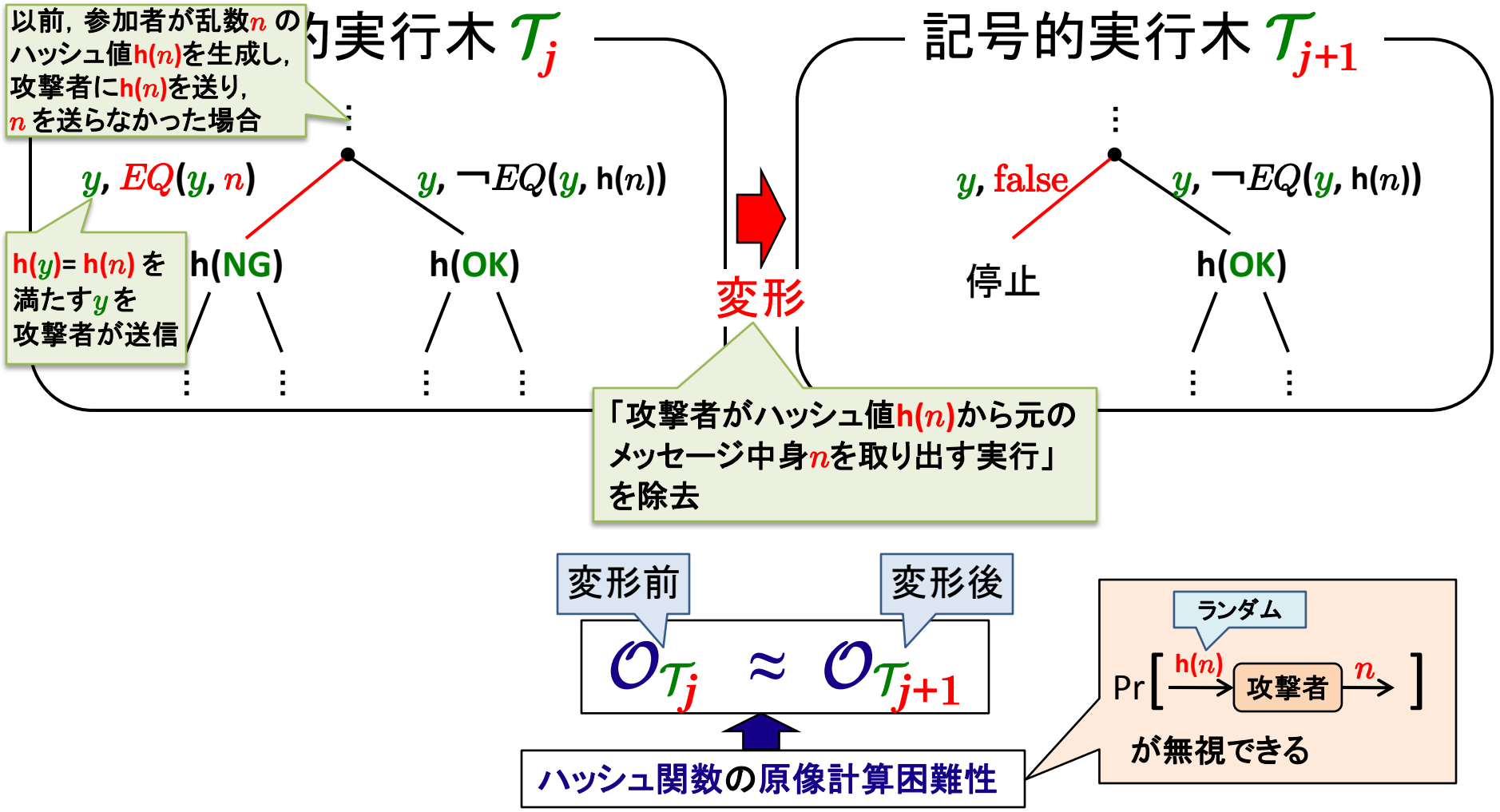
計算論的健全性の証明の概要 [本研究]

変形例2



計算論的健全性の証明の概要 [本研究]

変形例2の前後の識別不能性を示す



概要

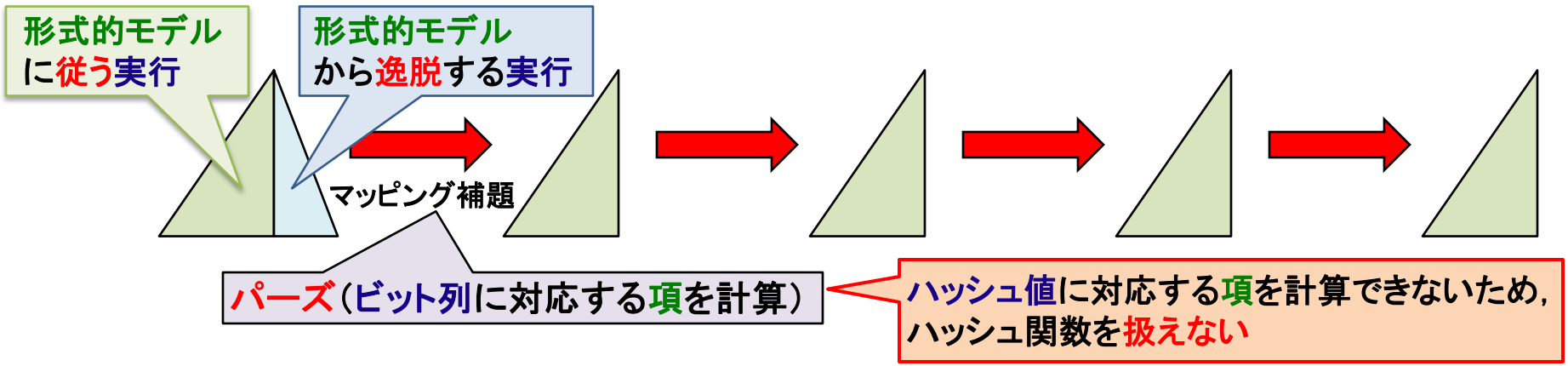
- ハッシュ関数を扱えるapplied pi-calculus
- applied pi-calculusの計算論的健全性
- 先行研究[Comon,Cortier'08]との比較
- 先行研究
- まとめ

• **対称鍵暗号のみ**
• **計算論的健全性の証明では**パーズ**を用いる**

計算論的健全性の証明技法の比較

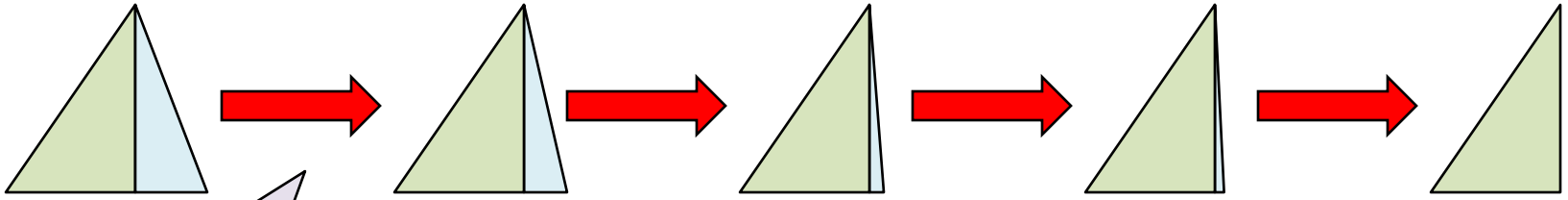
[Comon, Coriter'08]における証明

「形式的モデルから逸脱する実行」を一度で除去



本研究における証明

「形式的モデルから逸脱する実行」を少しずつ除去



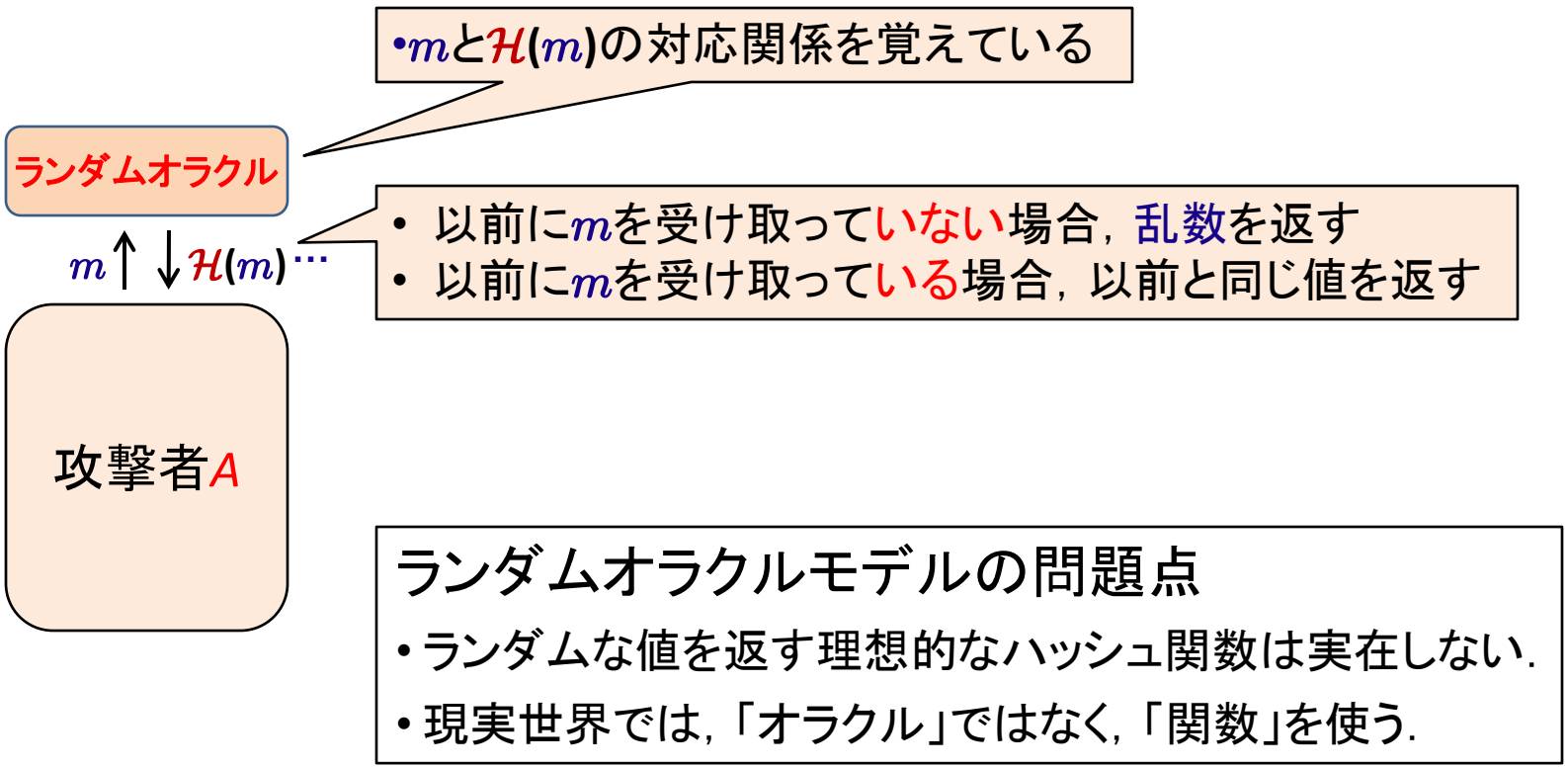
概要

- ハッシュ関数を扱えるapplied pi-calculus
- applied pi-calculusの計算論的健全性
- 先行研究[Comon,Cortier'08]との比較
- 先行研究[Cortier et.al'06]との比較
- まとめ

・ランダムオラクルモデル（強い仮定）
・導出不能性の計算論的健全性（弱い結果）

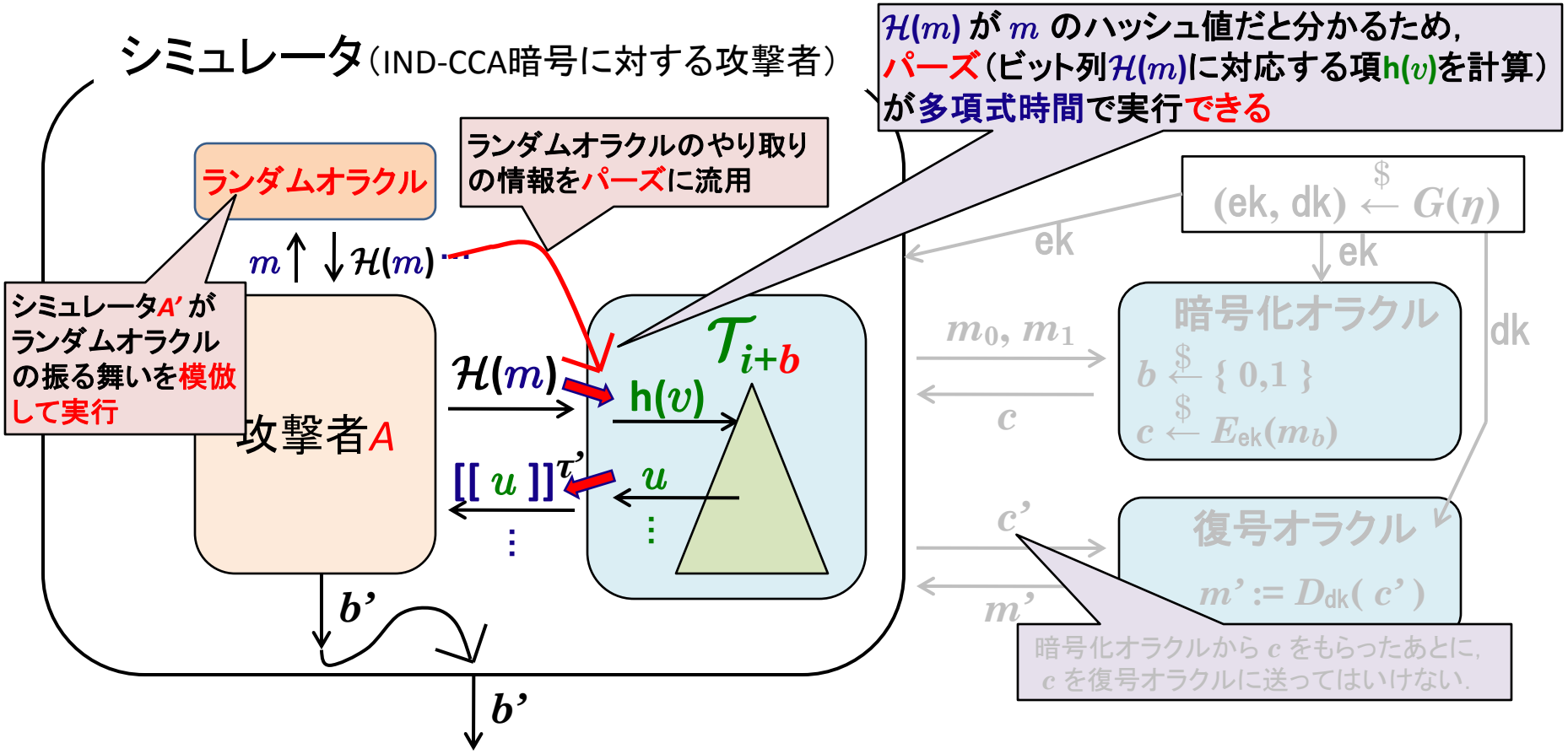
[Cortier et.al'06]の証明手法

- ランダムオラクルモデルとは？



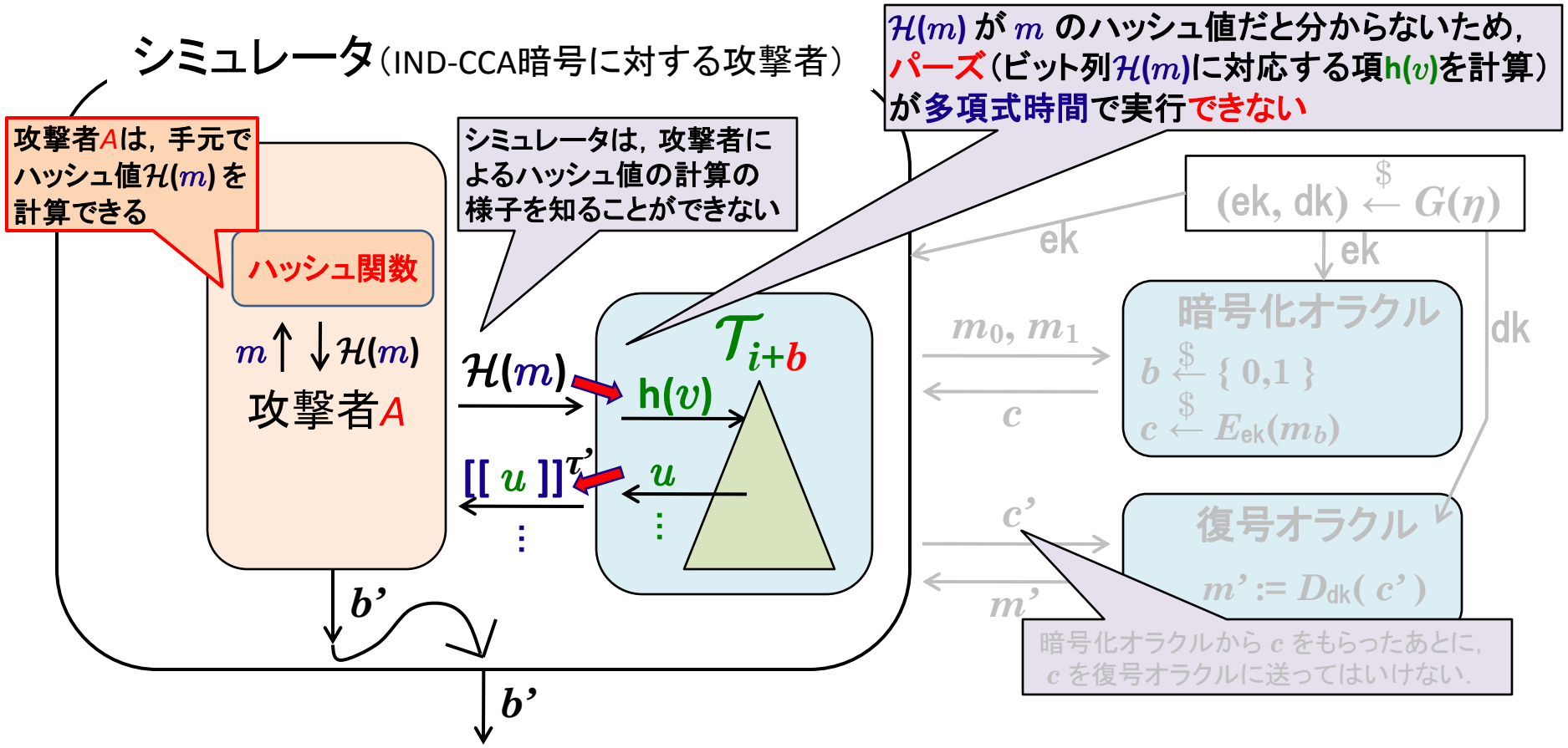
[Cortier et.al'06]の証明手法

- 証明の中でパーズを行う.（ [Comon, Cortier'08]等と同様）



[Cortier et.al'06]の証明手法の問題点

- ランダムオラクルの代わりに，現実のハッシュ関数を使う場合，[Cortier et.al'06]の証明手法を使えない．



概要

- ハッシュ関数を扱えるapplied pi-calculus
- applied pi-calculusの計算論的健全性
- 先行研究[Comon,Cortier'08]との比較
- 先行研究[Cortier et.al'06]との比較
- まとめ

本研究のまとめ

- ハッシュ関数と公開鍵暗号と電子署名(リング署名)を扱える **applied pi-calculus**の観測同値の計算論的健全性を証明
  - 能動的攻撃者と非有界個のセッション
  - パーズを用いない新たな証明技法を導入することにより,
    (ランダムオラクル以外の)ハッシュ関数も扱えるようになった.

- 今後の課題: 排他的論理和(XOR)

BPWモデルでは, パーズを用いるため,
ハッシュ関数と排他的論理和を扱えない.
([Backes, Pfitzmann'05])

ご清聴ありがとうございました.