



# 帰納論理プログラミングの 論理的基礎

---

山本 章博

京都大学 情報学研究科

<http://www.i.kyoto-u.ac.jp/~akihiro>

[mail: akihiro@ i.kyoto-u.ac.jp](mailto:akihiro@i.kyoto-u.ac.jp)



# 目次

---

- 1 . 帰納論理プログラミングとは
  - 帰納とは
  - 論理プログラミング
- 2 . 計算論的な学習
  - 概念と仮説
  - 学習のモデル化に必要な項目
  - 完全提示からの極限同定
- 3 . 精密化の利用
  - 正データからの学習
  - Model Inference System (負データを用いた学習)
- 4 . 最近の研究動向



# ポイント

---

- “帰納”という推論操作を計算論・論理的に定式化するための枠組み
- “精密化”による論理的な仮説生成
- “帰納に潜む数学・論理”から  
“数学・論理に潜む帰納”

# 1. 帰納論理プログラミングとは

## ■ 帰納論理プログラミング

(Inductive Logic Programming)

= 帰納的な推論 + 論理プログラミング

帰納論理プログラミング(ILP)とは、具体的な観測事例から、それを一般的に説明する規則性を論理プログラムの形で構成する手法を対象とする研究分野である。ILPは機械学習や知識のマイニングなどの帰納推論への応用を目的としているが、観測事例を仕様、規則性をプログラムと考えれば、論理プログラムの合成と捉えることもできる。

帰納論理プログラミング

計算論的学習

# 帰納(induction)とは

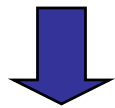
【**帰納**】 (induction) : 個々の具体的事実から  
**一般的な**命題ないし法則を導き出すこと。  
⇔ **演繹** (deduction)

- **機械学習**(machine learning)
  - 訓練例を与えることにより, 一般的な法則を構成して, 未知の事例に対処する.
- **知識発見**(knowledge discovery)
  - 実験結果や収集したデータ(観測事実)を説明できる法則(知識)を発見する

# プログラミングの観点から

- 規則・命題・法則 = プログラム
- プログラムが必要な利用者が、直接にプログラミングをすることなく、**訓練例・観測事実**に相当する**データ**からプログラムを構成する。[Rivest改]

$E_1, E_2, E_3, \dots$  : 訓練例・観測事実



$P_1, P_2, P_3, \dots$  : プログラム



# フロンティア

---

- G. Plotkin (1971)
  - **論理実証主義**(Hintikka and Hilpen)による発見の定式化
  
- E. Shapiro (1981)
  - **批判的合理主義**(Hume, Popper)による発見の定式化

# 論理プログラミング(1)

- 論理プログラミング:

プログラミング言語 Prolog の原理

- 論理プログラム = 確定節の有限集合

- 確定節 :  $A \leftarrow B_1, B_2, \dots, B_n$

$A, B_1, B_2, \dots, B_n$  は原子論理式 (atomic formula)

- Prolog言語では  $A :- B_1, B_2, \dots, B_n$ .

## 例 (自然数の計算)

$\text{plus}(s(X), Y, s(Z)) \leftarrow \text{plus}(X, Y, Z)$

$\text{plus}(0, X, X) \leftarrow$



# 論理プログラミング(2)

## 例(構造体計算)

$\text{app}([A|X], Y, [A|Z]) \leftarrow \text{app}(X, Y, Z)$

$\text{app}([], X, X) \leftarrow$

$\text{mem}(A, [A | X]) \leftarrow$

$\text{mem}(A, [B | X]) \leftarrow \text{mem}(A, X)$

## 例(データベース計算)

$\text{gf}(X, Y) \leftarrow \text{pt}(X, Z), \text{pt}(Z, Y), \text{ml}(X)$

$\text{pt}(\text{bob}, \text{alice}) \leftarrow$

$\text{pt}(\text{alice}, \text{catherine}) \leftarrow$

$\text{ml}(\text{bob}) \leftarrow$

# 論理プログラミング(3)

- 導出原理(融合原理, resolution)によるゴール節からのプログラムの実行

← plus(s(s(0)), s(0), Z)

$X_1 = s(0), Y_1 = s(0), Z = s(Z_1)$

plus(s(X<sub>1</sub>), Y<sub>1</sub>, s(Z<sub>1</sub>)) ← plus(X<sub>1</sub>, Y<sub>1</sub>, Z<sub>1</sub>)

← plus(s(0), s(0), Z<sub>1</sub>)

$X_2 = 0, Y_2 = s(0), Z_1 = s(Z_2)$

plus(s(X<sub>2</sub>), Y<sub>2</sub>, s(Z<sub>2</sub>)) ← plus(X<sub>1</sub>, Y<sub>1</sub>, Z<sub>2</sub>)

← plus(0, s(0), Z<sub>2</sub>)

$X_3 = s(0), Z_2 = s(0)$

plus(0, X<sub>3</sub>, X<sub>3</sub>) ←

$Z = s(Z_1) = s(s(Z_2)) = s(s(s(0)))$

# 論理プログラミング(4)

## ■ 解の検証

← plus(s(s(0)), s(0), s(s(s(0))))

plus(s(X<sub>1</sub>), Y<sub>1</sub>, s(Z<sub>1</sub>)) ← plus(X<sub>1</sub>, Y<sub>1</sub>, Z<sub>1</sub>)

← plus(s(0), s(0), s(s(0)))

plus(s(X<sub>2</sub>), Y<sub>2</sub>, s(Z<sub>2</sub>)) ← plus(X<sub>1</sub>, Y<sub>1</sub>, Z<sub>2</sub>)

← plus(0, s(0), s(0))

plus(0, X<sub>3</sub>, X<sub>3</sub>) ←

←

# 最小Herbrandモデル(1)

## ■ 論理プログラム $P$ の最小Herbrandモデル

$M(P) = P$  のHerbrandモデルで最小

$= \{ A \mid A \text{ は基礎原子論理式で } P \models A \}$

$= \{ A \mid A \text{ は基礎原子論理式で} \\ \leftarrow A \text{ からの導出が成功する} \}$

## 例

$P_{\text{plus}} : \text{plus}(s(X), Y, s(Z)) \leftarrow \text{plus}(X, Y, Z)$   
 $\text{plus}(0, X, X) \leftarrow$

$M(P_{\text{plus}}) = \{ \text{plus}(0,0,0), \text{plus}(0,s(0),s(0)), \dots,$   
 $\text{plus}(s(0),0,s(0)), \text{plus}(s(0),s(0),s(s(0))), \dots,$   
 $\text{plus}(s(s(0)),0,s(s(0))), \dots \}$

# 最小Herbrandモデル(2)

- 帰納推論の場合, 一つの述語に注目することが多い.
- 注目している述語に関する成功集合

$$M(P, p) = \{ p(t) \mid p(t) \text{ は基礎原子論理式で} \\ \leftarrow p(t) \text{ からの導出が成功する} \}$$

## 例

$$P_{\text{times}} : \text{times}(s(X), Y, Z) \leftarrow \text{times}(X, Y, W), \text{plus}(W, Y, Z) \\ \text{times}(0, X, 0) \leftarrow$$

$$M(P_{\text{times}}) = \{ \text{times}(0,0,0) , \text{times}(0,s(0),0), \dots, \\ \text{times}(s(0),0,0), \text{times}(s(0),s(0),s(0)), \dots, \\ \text{times}(s(s(0)),s(0),s(s(0))), \dots \}$$

# 論理プログラムの機械学習

HB : 基礎原子論理式全体の集合

$M$  : HBの部分集合で  $M = M(P)$

$P$  : 未知の論理プログラム  $P$

- $M(P)$ の要素 :

- ➡ 正例(正の訓練例, 正の観測事実)

- HB -  $M(P)$ の要素

- ➡ 負例(負の訓練例, 負の観測事実)

- 既知の論理プログラム  $B$

- ➡ 背景知識

未知の論理プログラム  $P$  を構成する .

# プログラムの成功集合

## 例 HBの部分集合

$$M = \{ p(0,0,0) , p(0,s(0),s(0)), \dots, \\ p(s(0),0,s(0)), p(s(0),s(0),s(s(0))), \dots, \\ p(s(s(0)),0,s(s(0))), \dots \}$$

- $M$  の要素を正例,
- HB -  $M$  の要素を負例,  
としたときに,  $M = M(P)$  なる論理プログラム

$$P : \begin{cases} p(s(X), Y, s(Z)) \leftarrow p(X, Y, Z) \\ p(0, X, X) \leftarrow \end{cases}$$

を構成する



# 議論を厳密にするために

---

- HBの部分集合  $M$  を(利用者の意図として)“選ぶ”とはどういうことか？
- 無限集合かも知れない  $M$  の要素や, その補集合の要素を観測例(訓練例)として与える, とはどういうことか？
- 目標のプログラムが“正しく学習できた”とはどういうことか？





## 2. 計算論的学習によるILP

---

### ■ 計算論的学習

#### 学習を計算論的に定式化・解析

- 極限同定(identification in the limit)
- EXACT学習(質問学習)
- PAC学習(probably approximately correct)

### ■ 情報論的・統計的学習

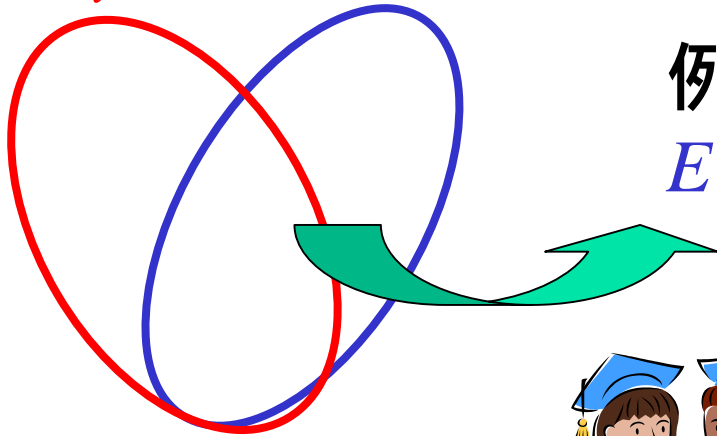
- ニューラルネットワーク
- ベイジアンネットワーク
- サポートベクトルマシン
- EM

# 計算論的な学習モデル



教師  
概念  $M = M(P)$   
に関する例示

$M(H_i)$   $M = M(P)$



例(データ)

$E_1, E_2, E_3, \dots$



仮説(など)

$H_1, H_2, H_3, \dots$



学習機械  
例からを仮説を  
導出する  
アルゴリズム

# 概念と仮説

- $U$  : 訓練例(観測事実)の表現からなる空間
  - 論理プログラミングの場合は  $HB$
- 概念 :  $U$  の部分集合
  - 論理プログラミングの場合は  $M \subseteq HB$
- 仮説 : 個々の概念を表現する方法
  - 論理プログラミングの場合は  $M = M(P)$
- 許容性 : 個々の概念は一つ以上の仮説で表現されている
  - 概念は, 仮説をパラメータとした集合



# 例

---

概念 :  $M(P) =$

$$\{ p(0,0,0) , p(0,s(0),s(0)), \dots, \\ p(s(0),0,s(0)), p(s(0),s(0),s(s(0))), \dots, \\ p(s(s(0)),0,s(s(0))), \dots \}$$

仮説 :  $P = p(s(X), Y, s(Z)) \leftarrow p(X, Y, Z)$   
 $p(0, X, X) \leftarrow$

# 学習のモデル化に必要な項目

- (対象) 概念空間  $C(H)$ 
  - 学習させたい(したい)概念を含むクラス
  - 未知の目標または「正しい理論」がある
- (表現) 仮説空間  $H$ 
  - 概念の表現全体の集合
  - 個々の概念はある方法で表現されている
- (入力) 訓練例・観測事実など
- (方式) 推論の方法(近似アルゴリズム)
- (出力) 仮説など
- (評価) 推論の正当性
  - 教師の意図と学習者の意図の一致

# 例と注

- $H = \{P_1, P_2, P_3, \dots\}$

$$P_1 = p(X, Y, Z) \leftarrow$$

$$P_2 = p(0, Y, Z) \leftarrow$$

...

$$P_{1576} = \begin{cases} p(0, Y, Z) \leftarrow \\ p(s(0), Y, s(0)) \leftarrow \end{cases}$$

...

$$P_{2893} = \begin{cases} p(0, Y, Y) \leftarrow \\ p(s(X), Y, s(Z)) \leftarrow p(X, Y, Z) \end{cases}$$

...

- $M(P_i) = M(P_j)$  であることも認める

# 比較：最尤推定

手元にあるサイコロの1の目の出る確率は  
いくらか？

概念空間	二項分布 $B(p, i)$ 全体
仮説空間 (概念の表現の空間)	1の目の出る確率 $0 \leq p \leq 1$
例の提示方法	試行
近似アルゴリズム	相対頻度 (最尤推定) $p^{\wedge} = k / n$
推論の正当性	統計的一致性

# 極限同定モデルでの例の提示



$E_1, E_2, E_3, \dots$



$H_1, H_2, H_3, \dots$

$M$  : 概念空間の要素である集合

■  $M$ に関する 正例 :  $M$  の要素

負例 :  $HB - M$  の要素

■ 正例と負例は  $\langle A, + \rangle$  ,  $\langle A, - \rangle$  のように区別して表現

■  $M$ に関する例の提示 : 正例と負例の無限列

完全提示 :  $M$ に関する全ての正例と全ての負例が少なくとも1回出現する列



# 枚挙と生成テストによる極限同定

- 仮説空間  $H$  内の論理プログラムの枚挙を固定

$P_1, P_2, \dots, P_n, \dots$

for  $n = 1$  forever

$E_n = \langle A_n, b_n \rangle$  を受取る ( $b_n = +$  または  $-$ )

$k = 1, H = P_1$

while ( $0 \leq \exists j \leq n$

$(E_j = \langle A_j, + \rangle$  かつ  $H \not\models A_j$ ) または

$(E_j = \langle A_j, - \rangle$  かつ  $H \models A_j$ ))

$H = P_k; k ++$

output  $H_n = H$



# 例

---

$$P : p(s(X), Y, s(Z)) \leftarrow p(X, Y, Z) \\ p(0, X, X) \leftarrow$$

$$E_1 = \langle p(s(0), s(0), s(s(0))), + \rangle \quad H_1 = p(X, Y, Z) \leftarrow$$

$$E_2 = \langle p(s(0), 0, 0), - \rangle$$

$$H_2 = \begin{cases} p(0, Y, Z) \leftarrow \\ p(s(0), Y, s(0)) \leftarrow \end{cases}$$

$$E_3 = \langle p(0, 0, 0), + \rangle$$

$$H_3 = H_2$$

...

...

$$E_{318} = \langle p(s(0), s(0), 0), - \rangle$$

$$H_{318} = \begin{cases} p(0, Y, Y) \leftarrow \\ p(s(X), Y, s(Z)) \leftarrow p(X, Y, Z) \end{cases}$$

...

...

# 極限同定 (identification in the limit) [Gold]



$E_1, E_2, E_3, \dots$



$H_1, H_2, H_3, \dots$

- 学習アルゴリズム  $IIM$  が概念  $M$  を**極限において (BC)同定**する

$\Leftrightarrow M(P)$ の任意の完全提示に対して

$$\exists N \forall n > N H_n = P \text{ かつ } M(P) = M.$$

- 学習アルゴリズム  $IIM$  が概念クラス  $C$  を**極限において同定**する

$\Leftrightarrow C$ 中の各概念を極限において同定する



# 背景知識を用いる例

---

$B : \text{app}([A \mid X ], Y, [A \mid Z]) \leftarrow \text{app}(X, Y, Z)$   
 $\text{app}([ ], Y, Y) \leftarrow$

$P : \text{r}([ ], [ ]) \leftarrow$

$\text{r}([A \mid X], Y) \leftarrow \text{r}(X, Z), \text{app}(Z, [A], Y)$

$\langle \text{r}([a, b], [b, a]), + \rangle, \langle \text{r}([a, b], [b]), - \rangle,$   
 $\langle \text{r}([a], [a]), + \rangle, \langle \text{r}([a, b, c], [c, b, a]), + \rangle,$   
 $\langle \text{r}([ ], [a]), - \rangle, \dots$

# 完全提示からの極限同定

概念空間 $C(H)$ (推定したい概念のクラス)	$H$ の要素の最小 Herbrandモデル
仮説空間 $H$ (概念の表現の集合)	ある性質を満たす論 理プログラムの集合
例の提示方法	完全提示
近似アルゴリズム(推論)	生成テスト
推論の正当性 (大域的な振舞)	極限同定

# 生成テストと論理プログラム

- 仮説空間  $H$  中の全ての論理プログラム  $P$  と任意の基礎原子論理式  $A$  に対して  
 $P \models A$  かどうか有限時間で判定可能であれば、概念クラス  $C(H)$  は枚挙と生成テストにより、極限同定可能である。

(注)  $P \models A$  の判定をPrologの実行を用いて行うとは限らない。

# 線形プログラム(1)

## ■ 線形論理プログラム $P$ :

$P$ 中の各節  $A \leftarrow B_1, B_2, \dots, B_n$  に対して

1.  $A$ 中の総記号数  $\geq B_i$ 中の総記号数
2. 各変数  $X$  について

$A$ 中の  $X$  の出現回数  $\geq B_i$ 中の  $X$  の出現回数

## 例

$\text{even}(0) \leftarrow$

$\text{even}(s(s(X))) \leftarrow \text{even}(X)$

$\text{plus}(s(X), Y, s(Z)) \leftarrow \text{plus}(X, Y, Z)$

$\text{plus}(0, X, X) \leftarrow$



# 線形プログラム(2)

---

## 例の続き

$\text{app}([A \mid X], Y, [A \mid Z]) \leftarrow \text{app}(X, Y, Z)$

$\text{app}([], X, X) \leftarrow$

$\text{mem}(A, [A \mid X]) \leftarrow$

$\text{mem}(A, [B \mid X]) \leftarrow \text{mem}(A, X)$

$\text{s-rev}([], X, X) \leftarrow$

$\text{s-rev}([A \mid X], Y, Z) \leftarrow \text{s-rev}(X, [A \mid Y], Z)$



# MIS(枚挙版)

- $B \wedge H \vdash E_j$  の証明の段数は  $n$  段に制限可
- $B \wedge H \not\vdash E_j$  の証明の段数を関数  $h$  によって制限する ( $h$ -easy)

for  $n = 1$  forever

$f_n = \langle E_n, b_n \rangle$  を受取る

$k = 1, H = S_1$

while (  $0 \leq \exists j \leq n$

( $f_j = \langle E_j, + \rangle$  かつ  $B \wedge H \not\vdash_{h(E_j)} E_j$ ) または  
( $f_j = \langle E_j, - \rangle$  かつ  $B \wedge H \vdash_n E_j$  ) )

$H = S_k; k ++$

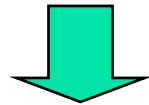
output  $H_n = H$

# 論理プログラムと $h$ -easiness

- 論理プログラムを対象にする場合

$B \wedge H \vdash A_n$  かどうかをSLD反駁で判定

$\Rightarrow h$  : SLD導出の無限ループ判定



線形性を仮定すれば,  $h$  は構成可能

# 極限同定モデルでは

- 概念クラス  $C$  が極限同定可能であったとしても、現在の仮説  $H_i$  が正しい仮説 ( $M = M(H_i)$ ) であるかどうかは、学習機械  $IIM$  にはわからない。
  - 質問(EXACT)学習[Shapiro, Angluin]
  - 確率的近似(PAC)学習[Valiant]
- 仮説の生成の速さについて考慮していない
  - 多項式時間学習

# 論理を積極的に利用した学習へ

- 枚挙と生成テストによる極限同定学習において、論理を使っている点は  $H \models A_j$  であるかどうかの判定だけである。
- 論理をより積極的に用いた学習
  - 学習における論理性
  - 固定した枚挙に頼らない柔軟な学習
  - 完全提示を求めない高機能な学習
  - 仮説生成の高速化



## 3. 精密化を利用したILP

---

### 精密化(refinement)

- Shapiro がMIS(Model Inference System)のために導入
- その後の帰納論理プログラミングの理論やシステムにおいて、様々なバージョンが陽に暗に利用される。
- Plotkinによる帰納論理プログラミングの核である反単一化も精密化に基づいている



# 正データ(正提示)

---

$M$  : 概念空間の要素である集合

- $M$ に関する正例 :  $M$ の要素
- $M$ に関する例の**正提示(正データ)** :  
 $M$ に関する全ての正例だけからなり, 全ての正例少なくとも1回出現する列
- 完全データの場合と同様に極限同定可能性を定義する.

# 正データからの極限同定(1)

Th. [Gold] 概念空間  $C(H)$  は、全ての有限集合と無限集合を一つ以上含めば正データから極限同定可能でない。

- 単位節プログラム：有限個の原子論理式からなる論理プログラム

$$P : \left\{ \begin{array}{l} A_1 \leftarrow \\ A_2 \leftarrow \\ \dots \\ A_n \leftarrow \end{array} \right.$$

$H_{\text{UNIT}}$  : 単位節プログラム全体

$C(H_{\text{UNIT}})$  は正データから極限同定可能でない。

# 原子論理式と正データ学習

- $H_{\text{ATOM}} = \{ A \leftarrow : A \text{は原子論理式} \}$   
: 単位節一つだけからなる論理プログラム全体

Th. [Lassez et al.]  $C(H_{\text{ATOM}})$ は正データから極限  
同定可能

例  $p(X, Y, s(Z)) \leftarrow \in H_{\text{ATOM}}$

$M(p(X, Y, s(Z)) \leftarrow ) =$

$\{ p(0,0,s(0)), p(s(0),0,s(0)), p(s(s(0)),0,s(0)), \dots$   
 $p(0,s(0),s(0)), \dots, p(0,0,s(s(0))), \dots \}$



# 原子論理式の極限同定

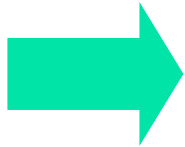
概念空間 $C(H)$ (推定したい概念のクラス)	$C(H_{\text{ATOM}})$ : 原子論理式 の最小Herbrandモデル 全体
仮説空間 $H$ (概念の表現の集合)	原子論理式全体 $\{A_1, A_2, \dots\}$
例の提示方法	<b>正提示</b>
近似アルゴリズム(推論)	<b>反単一化</b>
推論の正当性 (大域的な振舞)	<b>極限同定</b>

# 反代入例(anti-instance)

- 部分項を変数化
- 異なる部分項には, 異なる変数
  - 一つの部分項には, 複数の変数も可

$p(X, s(0), s(s(0)))$

$p(X, s(Y), s(s(0)))$



$p(X, s(Y), s(s(Y)))$

$p(X, Y, s(Z))$

- 通常は, 逆代入ではなく, 逆代入による結果に興味がある.

# 最小共通汎化(lca)

$A, B$  : 項または原子論理式

- $A, B$  の **共通汎化** (common anti-instance):  
 $L\theta = A, L\sigma = B$  となる項または原子論理式  $L$
- $A, B$  の **最小共通汎化**  $L$  (least common anti-instance, lca) :
  1.  $A$  と  $B$  の共通汎化であって,
  2.  $A$  と  $B$  の任意の共通汎化  $M$  に対して  $M\xi = L$

**Th[Plotkin]** 項または同じ述語記号の原子論理式  $A, B$  に対して, 最小共通汎化は常に存在して, それを求めるアルゴリズムが存在する.

# 共通汎化の別名

- 共通汎化を**反単一化**(anti-unification)ともよばれる。最小共通汎化を求めるアルゴリズムを**反単一化アルゴリズム**とよぶ。
- 原子論式以外の論理式に対しても汎化を定義するとき、汎化は **generalization**, 最小共通汎化は **least general generalization** という語を用いる。
  - Plotkinは単一化(unification)と相対的な演算として**反単一化**を考案したが、least general generalizationとよんだ

# 反単一化アルゴリズム

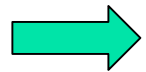
$$\langle p(0, s(0), s(s(0))), p(s(0), 0, s(0)) \rangle$$

$$\longrightarrow \langle p(0, s(0), s(s(0))), p(s(0), 0, s(0)) \rangle$$

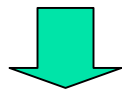
$$\longrightarrow \langle p(X_{0,s(0)}, s(0), s(s(0))), p(X_{0,s(0)}, 0, s(0)) \rangle$$

$$\longrightarrow \langle p(X_{0,s(0)}, X_{s(0),0}, s(s(0))), p(X_{0,s(0)}, X_{s(0),0}, s(0)) \rangle$$

$$\longrightarrow \langle p(X_{0,s(0)}, X_{s(0),0}, s(s(0))), p(X_{0,s(0)}, X_{s(0),0}, s(0)) \rangle$$



$$\langle p(X_{0,s(0)}, X_{s(0),0}, s(X_{s(0),0})), p(X_{0,s(0)}, X_{s(0),0}, s(X_{s(0),0})) \rangle$$



$$p(X, Y, s(Y))$$

# 原子論理式に対する推論規則

## ■ 原子論理式に対する推論規則

$$\frac{A}{A[X := f(X_1, \dots, X_n)]}$$

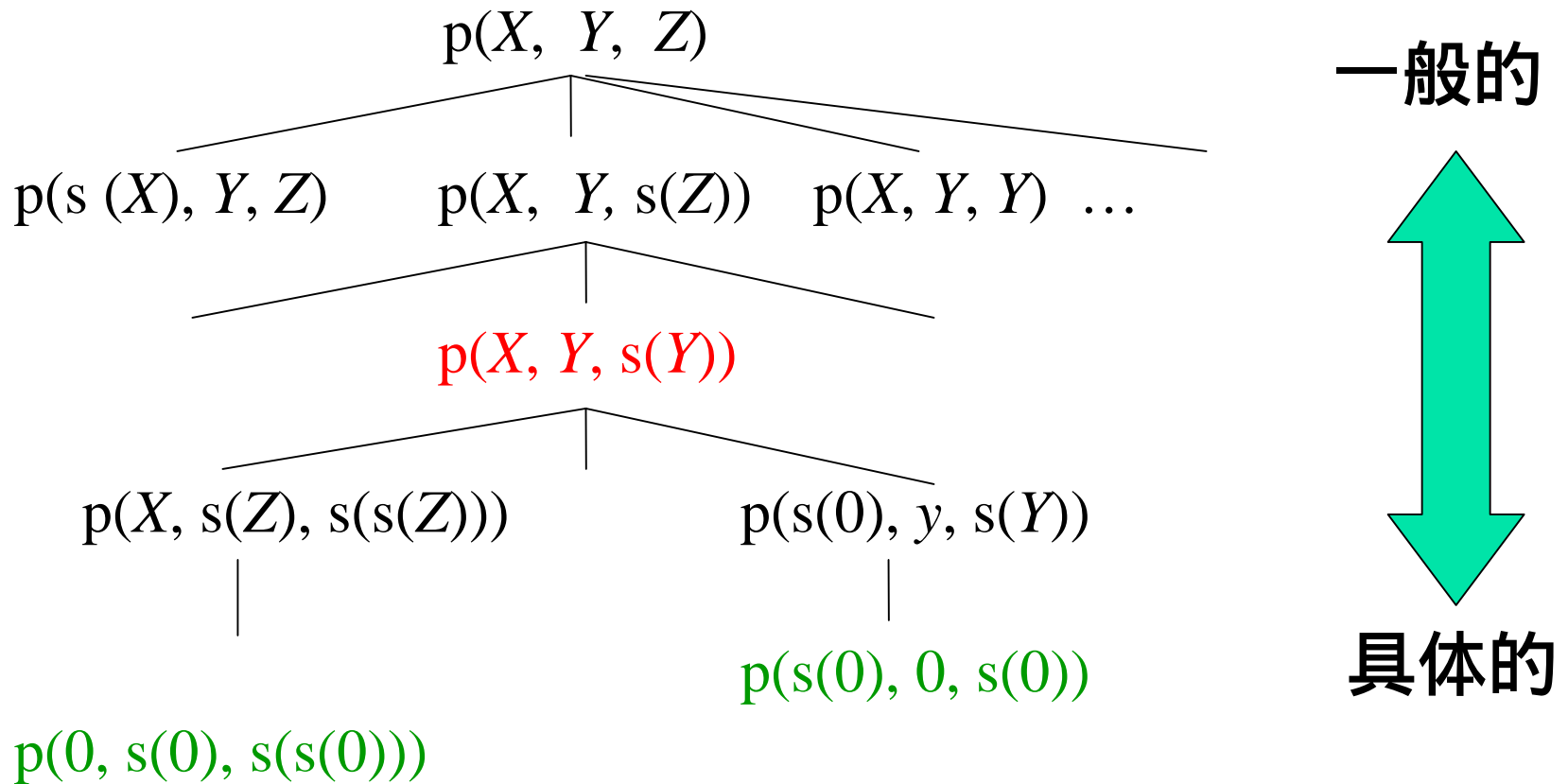
$$\frac{A}{A[X := Y]}$$

- $n \geq 0$
- $X, Y$  は  $A$  に出現する変数
- $X_1, \dots, X_n$  は  $A$  には出現しない相異なる変数

- 述語記号  $p$  の任意の原子論理式(の変種, variant)は上の二つの規則によって,  $p(X_1, \dots, X_n)$  から導出可能

# 反単一化と意味

- 2つの原始論理式の演繹的最小近似



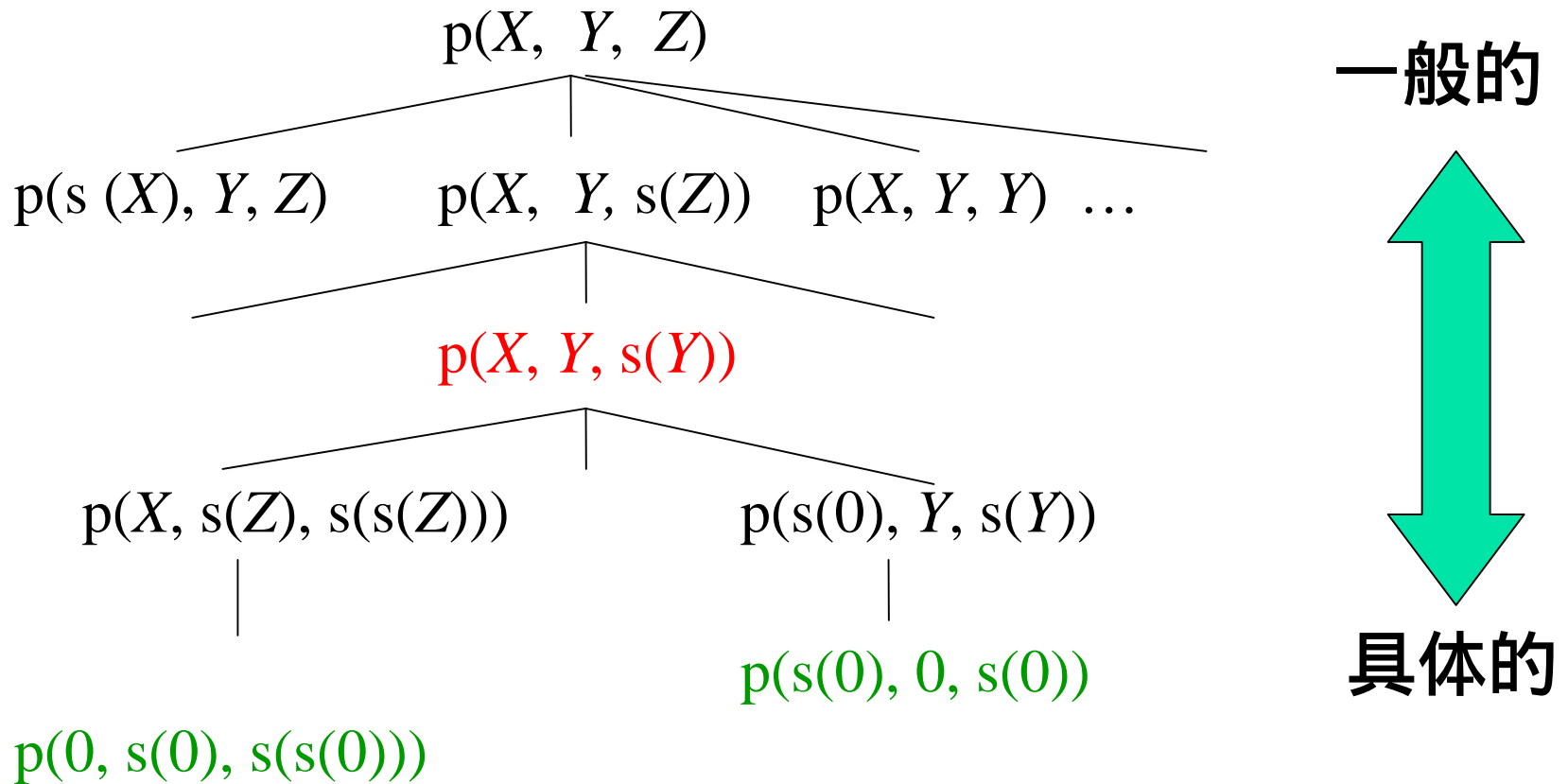
# 原子論理式の極限同定

$C(H_{\text{ATOM}})$ : 原子論理式の 最小Herbrandモデル全体	二項分布 $B(p, i)$ 全体
原子論理式全体 $\{A_1, A_2, \dots\}$	1の目の出る確率 $0 \leq p \leq 1$
正提示	試行
反単一化	相対頻度 (最尤推定) $p^{\wedge} = k / n$
極限同定	統計的一致性



# 有限の厚さ (finite thickness)

- 一つの基礎原子論理式に対して, それを含むような概念  $L(A)$  は有限個しかない





# 正データからの学習(2)

Th. [Angluin] もし概念空間  $C(H)$  が **有限の厚さ**を持つならば,  $C(H)$  は正データから極限同定可能である.

# 応用例

## ■ 半構造化文書群からの共通パターン抽出

[Yamamoto et al. 01]

The diagram illustrates the extraction of common patterns from semi-structured weather data. It shows four overlapping screenshots of a Yahoo! Weather page for Fukuoka, with various elements highlighted by yellow and blue boxes and labeled with 'V' and 'I' identifiers.

**Yellow boxes (Leftmost screenshot):**

- Yahoo!天気情報
- トップ > 福岡県 > 福岡(福岡県 福岡地方)
- 今日・明日の天気
- 8月15日(水)
- 最高気温(度) 35
- 暗れ
- 降水確率(%) --- 10 20 10
- 時間帯(時) 0-6 6-12 12-18 18-24
- 警報注意報
- 生活指数

**Blue boxes (Rightmost screenshot):**

- V(0) > V(1)地方
- 気情報
- 今日・明日の天気
- V(5)
- 最高気温(度) V(9)
- 風: V(10)
- 波: V(11)m
- 降水確率(%) --- V(12) V(13) V(14)
- 時間帯(時) 0-6 6-12 12-18 18-24
- 警報注意報
- 生活指数

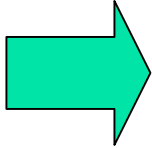
**Other elements:**

- 関連天気情報
- 天気図
- アメダス画像
- Yahoo!関連コンテンツ
- Yahoo!関連カテゴリ

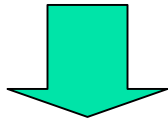
A large green arrow points from the leftmost screenshot to the rightmost one, indicating the direction of pattern extraction.

# 半構造化データ

a	b
c	d



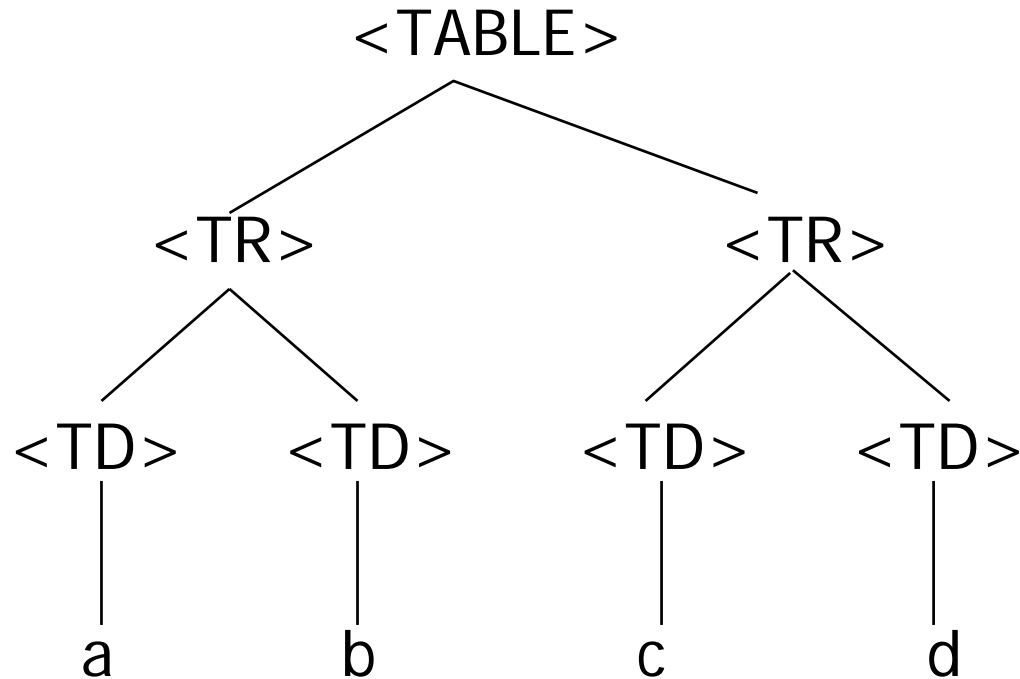
```
<TABLE>  
  <TR>  
    <TD>a</TD> <TD>b</TD>  
  </TR>  
  <TR>  
    <TD>c </TD> <TD>d</TD>  
  </TR>  
</TABLE>
```



```
TABLE(TR(TD(a)TD(b))TR(TD(c)TD(d)))
```

# DOM

TABLE(TR(TD(a)TD(b))TR(TD(c)TD(d)))



# 反単一化への帰着

- タグの子の数は、タグによって定まっている。
  - 対象とする表の行数、列数は一定
  - 兄弟タグは**コンマ**で分離

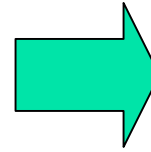
```
TABLE(TR(TD(a), TD(b)), TR(TD(c), TD(d)))
```

- 文字列部分(PCDATAに相当)は記号1個と仮定する。
  - 文字列の構造までは考慮しない
- 共通パターンを表すために**変数**を用いる。

# 変数による共通パターン表現

a	b
c	d

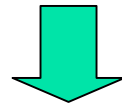
e	f
g	h



X	Y
Z	W

TABLE(TR(TD(a), TD(b)), TR(TD(c), TD(d)))

TABLE(TR(TD(e), TD(f)), TR(TD(g), TD(h)))

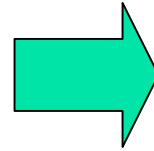


TABLE(TR(TD(X), TD(Y)), TR(TD(Z), TD(W)))

# 変数による共通パターン表現

a	b
a	d

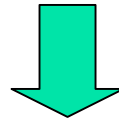
e	b
e	h



X	b
X	W

TABLE(TR(TD(a), TD(b)), TR(TD(a), TD(d)))

TABLE(TR(TD(e), TD(b)), TR(TD(e), TD(h)))



TABLE(TR(TD(X), TD(b)), TR(TD(X), TD(W)))

$X_{a,e}$

$X_{a,e}$

$X_{d,h}$



# Unit Programの極限同定

- 概念空間

$$C(H_{\text{ATOM}})^k = \{ M(A_1) \cup \dots \cup M(A_n) \}$$

:  $A_i$  は基礎原子論理式かつ  $n \leq k$  }

は正データから極限同定可能

- Ramsey's TheoremやHigman's Theoremを使う

# 有限の弾力性

- 概念空間  $C(H)$  が**有限の弾力性**をもつ

⇔ HBの要素の無限列

$$A_1, A_2, \dots, A_n, \dots$$

と  $C(H)$  中の概念の無限列

$$M(P_1), M(P_2), \dots, M(P_n), \dots$$

で、常に

$$\{A_1, A_2, \dots, A_n\} \subseteq M(P_n) \text{ かつ } A_{n+1} \notin M(P_{n+1})$$

であるようなものを構成することができない

- 有限の厚さ  $\Rightarrow$  有限の弾力性

# 正データからの学習(3)

Th. [Wright] もし概念空間  $C(H)$  が**有限の弾力性を持つならば**,  $C(H)$  (中の各概念)は正データから極限同定可能である.

Th. [Wright] もし概念空間  $C(H)$  が**有限の弾力性を持つならば**,

$$C(H)^k = \{ C_1 \cup \dots \cup C_n : C_i \in C(H) \text{ かつ } n \leq k \}$$

は正データから極限同定可能である.

# 節の最小共通汎化

Th[Plotkin]節(確定節)の最小共通汎化は常に存在して、それを求めるアルゴリズムが存在する.

- $A \leftarrow B_1, B_2, \dots, B_n$  と  $A' \leftarrow B'_1, B'_2, \dots, B'_m$  の最小共通汎化:

$$A \leftarrow B_1, \dots, B_1, B_2, \dots, B_2, B_3, \dots, B_n$$

$$A \leftarrow B'_1, \dots, B'_1, B'_2, \dots, B'_2, B'_3, \dots, B'_n$$

に原子論理式としての反単一化を適用し、その結果を宿約する.



# RDBからの知識発見[Plotkin1971]

ID	Class	Size	Color	Animal
A	Dangerous	Small	Black	Bear
B	Dangerous	Medium	Black	Bear
C	Dangerous	Large	Black	Dog
D	Safe	Small	Black	Cat
E	Safe	Medium	Black	Horse
F	Dangerous	Large	Black	Horse
G	Dangerous	Large	Brown	Horse

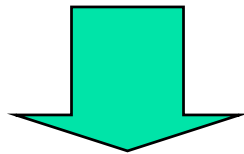


---

class(a, dangerous) ←

size(a, small), color(a, black), animal(a, bear)

...



*H* : cls(*X*, dng) ← clr(*X*, blk), ani(*X*, bear)

cls(*X*, dng) ← sz(*X*, large)

cls(*d*, sf) ← sz(*d*, small), clr(*d*, blk), ani(*d*, cat)

cls(*e*, sf) ← sz(*e*, small), clr(*e*, blk), ani(*e*, bear)

# 包摂関係

- 包摂(subsumption)関係  $P \geq Q$  :  
論理プログラム間の導出可能性による順序

$$\frac{P}{P \cup \{ A \leftarrow B_1, B_2, \dots, B_n, p(X_1, \dots, X_n) \}} \quad \frac{P \cup C}{P}$$

$$\frac{P}{P \cup C[X := f(X_1, \dots, X_n)]} \quad \frac{P}{P \cup C[X := Y]}$$

ただし  $C : A \leftarrow B_1, B_2, \dots, B_n$

$X_1, \dots, X_n$  は  $C$  には出現しない相異なる変数

# 包摂関係と最小Herbrandモデル

- 包摂関係は最小Herbrandモデルの包含関係に関する順序を導く(単調性)

$$P \geq Q \quad M(P) \supseteq M(Q)$$

- $\geq$ に関して最大のプログラム:

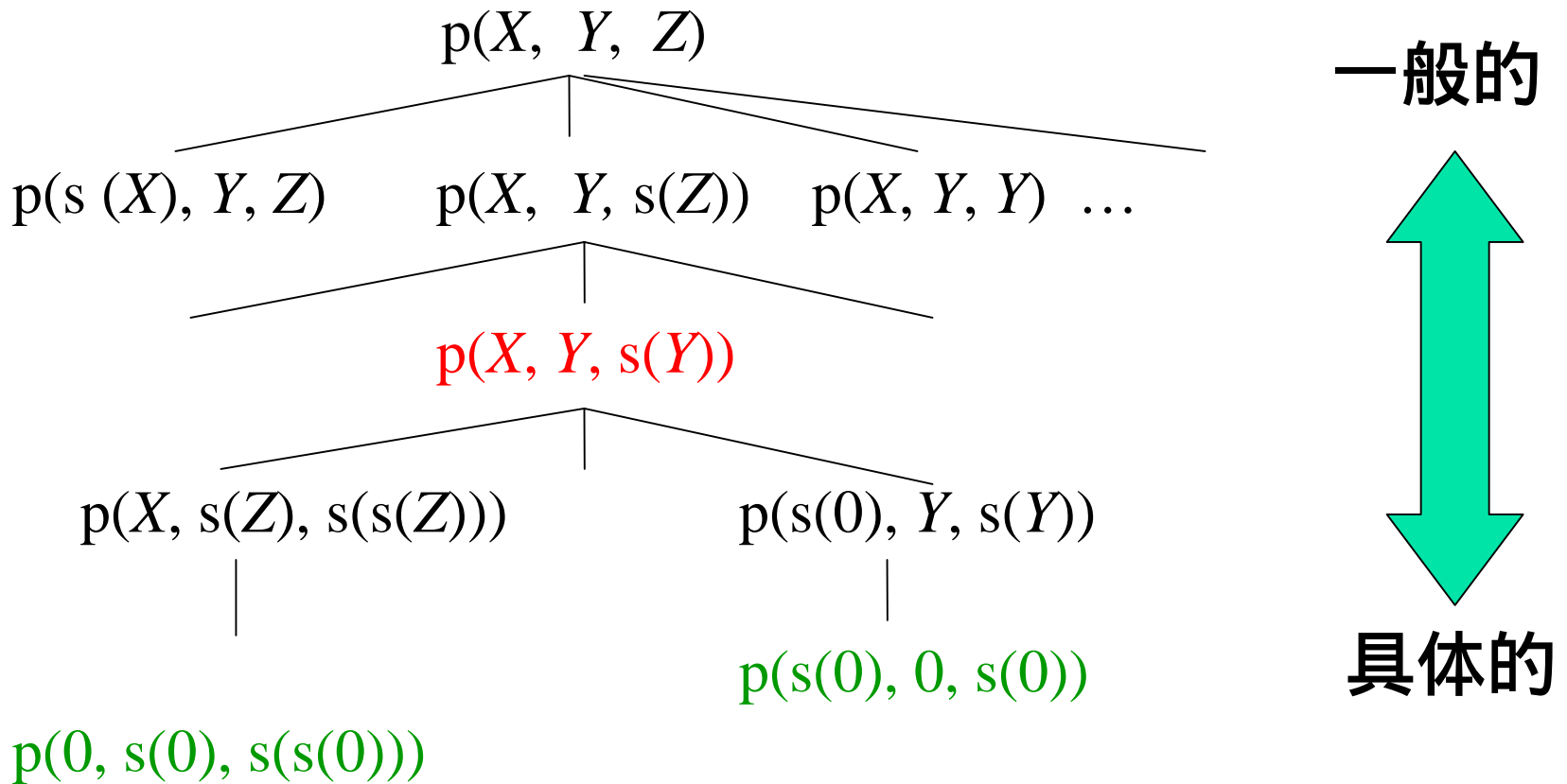
$$\left\{ \begin{array}{l} p_1(X_1, \dots, X_{n_1}) \leftarrow \\ p_2(X_1, \dots, X_{n_2}) \leftarrow \\ p_3(X_1, \dots, X_{n_3}) \leftarrow \\ \dots \end{array} \right.$$

- 推論規則による論理プログラムの導出を帰納推論では**精密化**とよぶ



# 反単一化と意味

- 2つの原始論理式の演繹的最小近似



# Model Inference System

- Model Inference System[Shapiro]  
節の包摂関係を利用した負データからの学習

$$A \leftarrow B_1, B_2, \dots, B_n$$

---

$$A \leftarrow B_1, B_2, \dots, B_n, p(X_1, \dots, X_n)$$

$$C$$

---

$$C[X := f(X_1, \dots, X_n)]$$

$$C$$

---

$$C[X := Y]$$

# 負例による仮説の具体化

- $H \vdash E$  ならば  $H \wedge C \vdash E$

➡ 正例を説明するだけなら、冗長な節が含まれていても気にしなくてよい

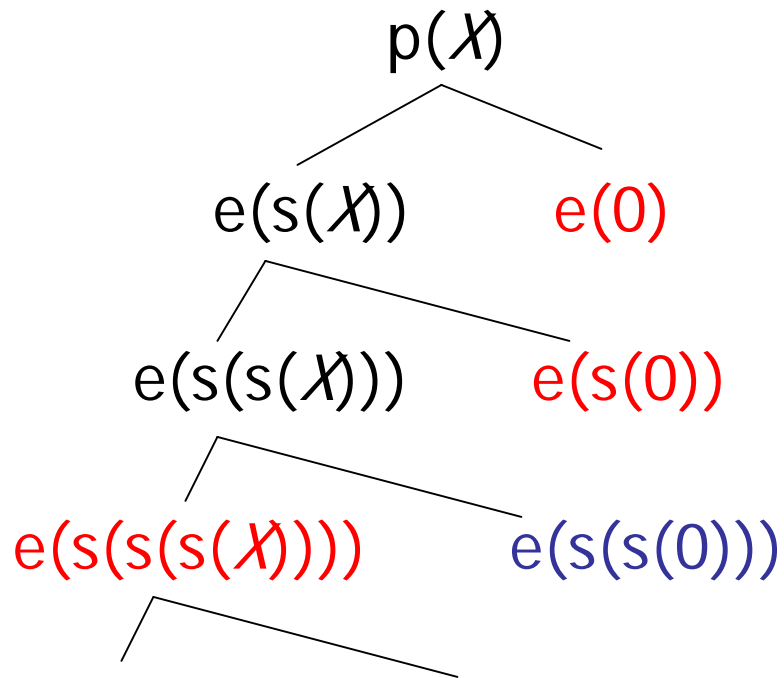
- 節  $C$  が  $D$  を包摂するならば、

$$H' \wedge D \vdash E \Rightarrow H' \wedge C \vdash E$$

➡  $D$  を用いて負例が説明できるのならば、 $C$  を使っても説明できてしまう

# 負の例による具体化

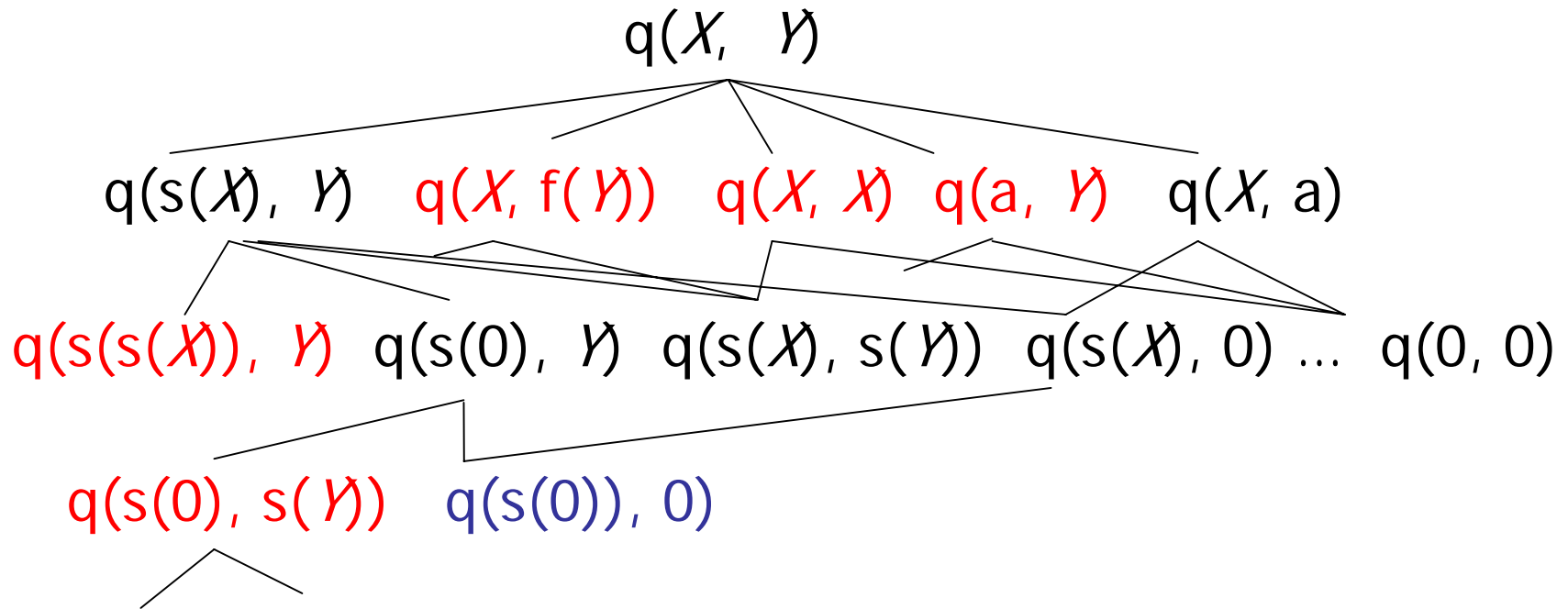
$\langle e(s(s(0))), - \rangle$



$$H = e(0) \wedge e(s(0)) \wedge e(s(s(s(X))))$$

# 負の例による具体化

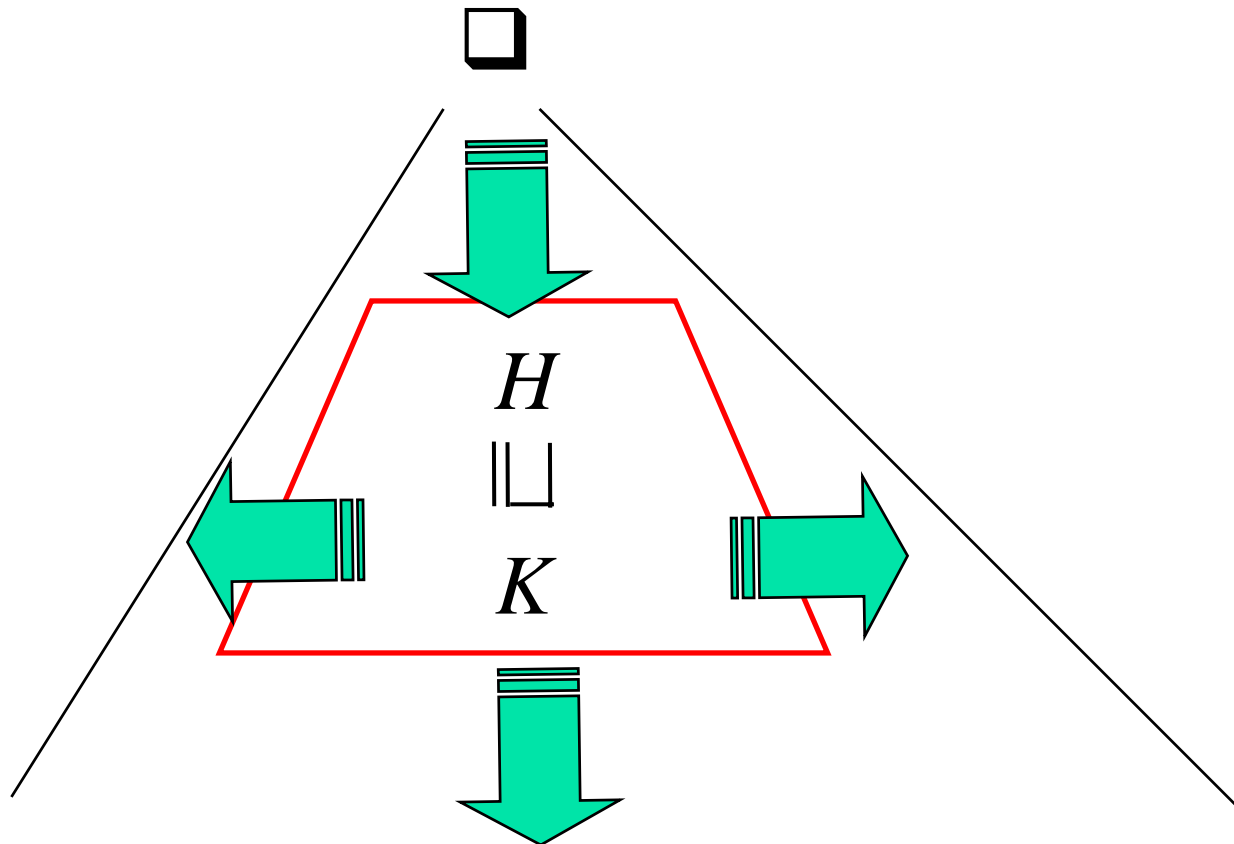
$\langle q(s(0), 0), - \rangle$



$$H = q(X, s(Y)) \wedge q(X, X) \\ \wedge q(0, Y) \wedge q(s(s(X)), Y) \wedge q(s(0), s(Y))$$

# MISの仮説探索方法

- **節全体**からなる仮説空間を包摂関係と記号数を用いて構造化



# 例

$$H_{i-1} = e(X) \leftarrow f_{i-1} = \langle e(s(s(s(0)))) , - \rangle$$

$$L_1 = e(0) \leftarrow \wedge e(X) \leftarrow$$

$$L_2 = L_1 \\ \wedge e(s(0)) \leftarrow \wedge e(s(X)) \leftarrow$$

$$L_3 = L_2 \\ \wedge e(s(s(0))) \leftarrow \wedge e(s(s(X))) \leftarrow \\ \wedge e(s(0)) \leftarrow e(0) \wedge e(s(X)) \leftarrow e(0) \\ \wedge e(s(0)) \leftarrow e(X) \wedge e(s(X)) \leftarrow e(X)$$

$$L_4 = L_3 \\ \wedge e(s(s(s(0)))) \leftarrow \wedge e(s(s(s(X)))) \leftarrow \\ \wedge \dots \\ \wedge e(s(s(0))) \leftarrow e(X) \wedge e(s(s(X))) \leftarrow e(X) \wedge \dots$$

# 質問学習モデル



教師  
概念 $H$ に関する  
例示

例 + 判定  
 $E_1, E_2, E_3, \dots$



仮説 + 質問  
 $H_1, H_2, H_3, \dots$



学習機械  
例からを仮説を  
導出する  
アルゴリズム



# 多項式時間質問学習 [Angluin(1988)]

学習アルゴリズム  $M$  は  
等価性質問  $h_{i-1} = L?$  を用い、  
もしnoなら多項式個の反例

$w_{i1}, \dots, w_{im} \in h_{i-1} \Delta L$  を受け取り  
多項式時間で仮説  $h_i$  を生成し、  
最終的に  $h_n = L$  を出力する

$L$  のパラメータと反例の最大長の多項式

# 漸増戦略(1)

$H = \emptyset$

while ( $H \neq H_*$ ) do

次の反例  $E$  をもらう

for each  $C \in H$  do

$\text{lca}(C, E)$ を用いて所属性質問をする

( $H_*$  に $\text{lca}(C, E)$ を包摂する節があるか)

if yes for some  $C$  then

$H$  中の $C$ を $\text{lca}(C, D)$ で置き換え

else

$H$  に  $E$  を追加



# 例

---

$$H_* = p(X, Y, f(Y)) \wedge p(f(a), b, f(a)) \quad H =$$

$$1 \quad p(b, a, f(a))$$

$$H = p(b, a, f(a))$$

$$2 \quad p(f(a), b, f(a))$$

$$? \text{lca}(p(b, a, f(a)), p(f(a), b, f(a))) = p(X, Y, f(a))$$

“no”  $H = p(b, a, f(a)) \wedge p(f(a), b, f(a))$

$$3 \quad p(a, f(a), f(f(a)))$$

$$? \text{lca}(p(b, a, f(a)), p(a, f(a), f(f(a)))) = p(X, Y, f(Y))$$

“yes”

$$? \text{lca}(p(f(a), b, f(a)), p(a, f(a), f(f(a)))) = p(X, Y, f(Z))$$

“no”

$$H = p(X, Y, f(Y)) \wedge p(f(a), b, f(a))$$



# 欠落節

---

Missing( $E, H$ )

未知の論理プログラム  $H_*$  を用いて  $H_* \vdash E$  を証明する際に用いた節  $C$  が包摂する節  $D$  で  $H \not\vdash D$  であるもの

# 漸増戦略(2)

$H = \emptyset$

while ( $H \neq H_*$ ) do begin

次の反例  $E$  をもらう

$D := \text{Missing}(E, H)$

*/\*  $H \not\subseteq D$  かつ  $C \in H_*$  かつ  $C \geq D$  \*/*

for each  $C \in H$  do

$H_*$  に  $\text{lca}(C, E)$  を包摂する節があるか?

if yes for some  $C$  then

$H$  中の  $C$  を  $\text{lca}(C, D)$  で置き換え

else

$H$  に  $E$  を追加



## 4 . 最近の研究動向

---

- 極限同定と $\Sigma_2$ 論理式
- 正データ学習と計算代数
- 情報論的学習・統計的学習と論理プログラミング
- データマイニングと論理プログラミング

# 極限同定と $\Sigma_2$ 論理式

- 極限同定学習における生成テストのプロセスは $\Sigma_2$ 論理式で表現できる

$$\exists H (\text{hyp} (H) \wedge \forall E (\text{expl} (E) \rightarrow \text{cst}(H, E)))$$

hyp : 仮説空間の定義述語

expl : 概念の定義述語

cst : 無矛盾性の判定

- 極限数学[Nakata and Hayashi]
- RichProlog[Martin et al.]
- 排中律の階層[Akama et al.]

# 極限同定と互除法

概念空間	$C(H_{\text{ATOM}})$	自然数 $m$ の 倍数
概念の表現 (仮説空間)	原始論理式 $A$	$m$ そのもの
例の提示方法	正データ	正データ
近似アルゴリズム ( <b>推論</b> )	反単一化	互除法
<b>推論の正当性</b> (大域的な振舞)	<b>極限同定</b>	<b>自然数の最小 値の存在</b>



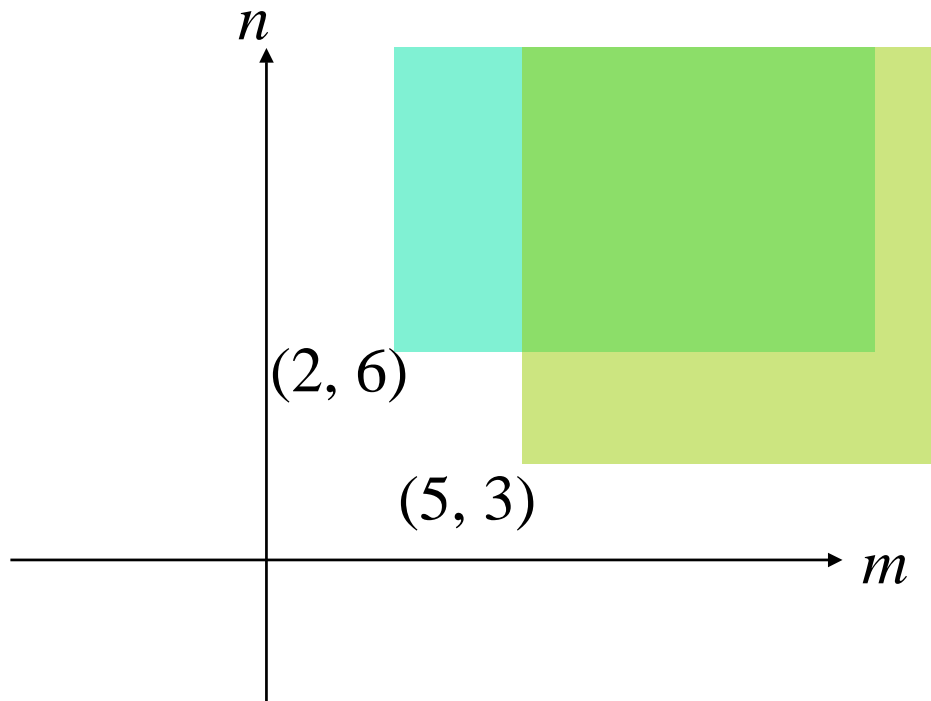
# 正データ学習と計算代数

概念空間	$C(H_{\text{ATOM}})$	多項式環の イデアル全体
概念の表現 (仮説空間)	原始論理式 $A$	イデアルの 有限基底
例の提示方法	正データ	正データ
近似アルゴリズム ( <b>推論</b> )	反単一化	Buchberger アルゴリズム
<b>推論の正当性</b> (大域的な振舞)	<b>極限同定</b>	<b>ネータ性</b>

# Hilbertの基底定理の本質

- 単項式イデアルは有限生成(Dicksonの補題)

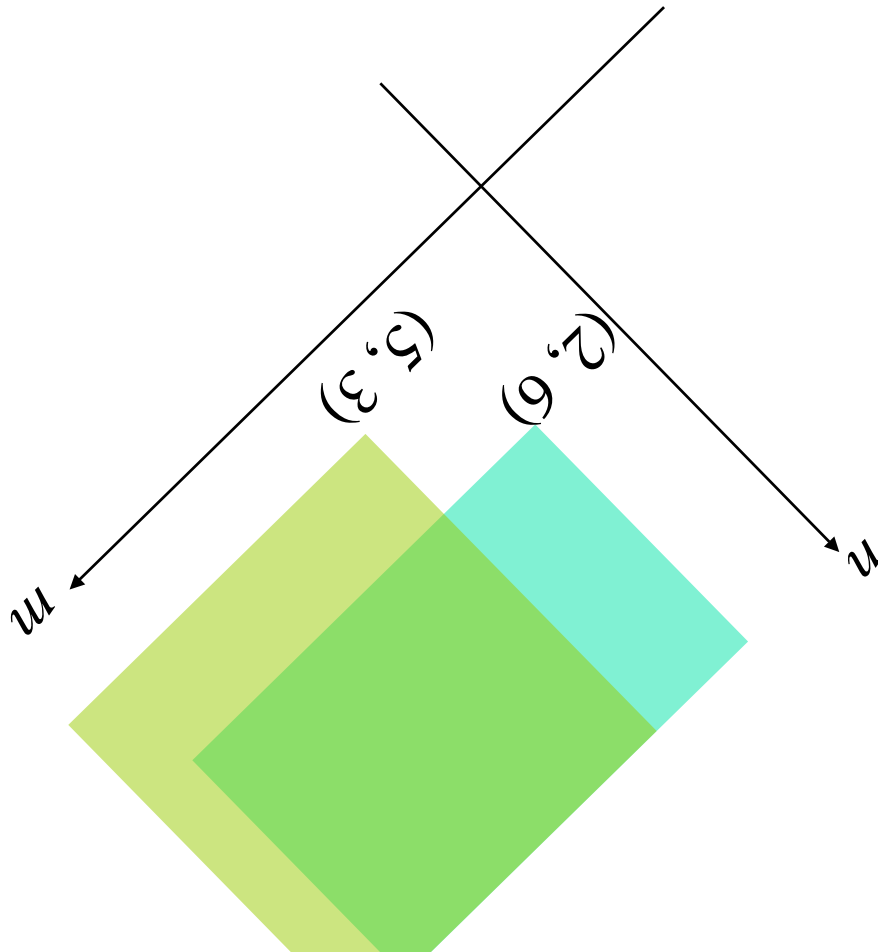
$$x^m y^n z^k \Rightarrow (m, n, k)$$



# Hilbertの基底定理の本質

- 単項式イデアルは有限生成(Dicksonの補題)

$$x^m y^n z^k \Rightarrow (m, n, k)$$





# 「計算」と「学習」の歴史

## 基底定理

Hilbert 決定問題

1930s Church 「計算」概念提唱

計算論

TM から 計算機へ

1960s Gold 計算論的学習

1970s Angluin 正データ学習

1980s Valiant 効率的学習

グレブナ基底

Mathematica

# 反単一化再考

- 任意の原始論理式が, 2つの基礎原始論理式の“差”で表せる

$M(A)$ には基底  $\{A_1, A_2\}$  がある.

$$\langle p(0, s(0), s(s(0))), p(s(0), 0, s(0)) \rangle$$

$$\longrightarrow \langle p(0, s(0), s(s(0))), p(s(0), 0, s(0)) \rangle$$

$$\longrightarrow \langle p(X_{0,s(0)}, s(0), s(s(0))), p(X_{0,s(0)}, 0, s(0)) \rangle$$

$$\longrightarrow \langle p(X_{0,s(0)}, X_{s(0),0}, s(s(0))), p(X_{0,s(0)}, X_{s(0),0}, s(0)) \rangle$$

$$\longrightarrow \langle p(X_{0,s(0)}, X_{s(0),0}, s(s(0))), p(X_{0,s(0)}, X_{s(0),0}, s(0)) \rangle$$

$$\longrightarrow p(X, Y, s(Y))$$

# 正データからの学習(4)

Th. [Angluin] 概念空間  $C(H)$  が正データから極限同定可能であるための必要十分条件は、 $C(H)$  中の各概念  $M(P)$  の有限証拠集合 (finite-telltale)  $T(P)$  が存在し、表現  $P$  を与えると  $T(P)$  の要素を枚挙する手続きが存在することである

概念  $M(P)$  の有限証拠集合 (finite-telltale)  $T(P)$  :

$T(P) \subseteq M(P)$  かつ

$M(P) \neq M(P')$  であるどのよう  $C(H)$  中の概念  $M(P')$  に対しても  $T(P) \subseteq M(P') \subset M(P)$  とはならない.



# 情報論的学習と論理プログラミング

---

- パラメータ推定と論理プログラミング

PRISM [Sato]

- ベイジアンネットと論理プログラミング [de Raed]

- SVMと論理プログラミング

などなど



# 実働システムについて

---

- Aleph 帰納論理プログラミング・システム  
YAP (Yet Another Prolog) のApplication の一つ  
<http://www.ncc.up.pt/~vsc/Yap/>  
SWI Prologでも動作
- PRISM 統計的学習システム(東工大 佐藤研)  
<http://sato-www.cs.titech.ac.jp/ja/research/>  
B-Prologで動作
  - 論理プログラムの分布意味論に基づく





# 国際会議は

---

- ILP (Inductive Logic Programming)
- ALT (Algorithmic Learning Theory)
- COLT (Computational Learning Theory)

などが有名ですが...

# Learning with Logics and

# Logics for Learning

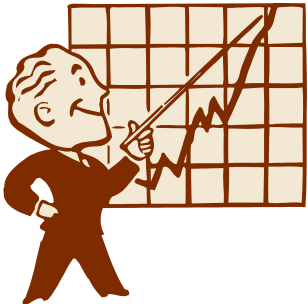
---

collocated with

the 19th Annual Conference of JSAI

KitaKyuhusu (Kokura)

13-14 June 2005





# 参考文献

---

## フロンティア

- G. Plotkin : Automatic Method of Inductive Inference, Edinburgh University, 1971
  - 一部の証明などを省略した論文  
A Further Note on Inductive Generalization が  
Machine Intelligence 6, Edinburgh University Press (1971)に収録
- E.Y.Shapiro, Inductive Inference of Theories From Facts, Technical Report 192, Department of Computer Science, Yale University, 1981.
  - J.-L. Lassez and G. Plotkin (eds.) Computational Logic, The MIT Press, pp.~199--254, 1991 に収録
  - 有川(訳): 知識の帰納的推論, 共立出版, 1986.



## 専門書

- S.-H. Nienhuys-Cheng and R. de Wolf : *Foundations of Inductive Logic Programming* (LNAI 1228), Springer, 1997.
- P. D. Laird: *Learning from Good and Bad Data*, Kluwer Academic Publishers, 1988.
  - 横森貴(訳) 例からの学習 --計算論的学習理論--, オーム社, 1992.
- 大須賀・佐伯(編) 知識の獲得と学習, オーム社, 1987.
- 榊原, 小林, 横森 : 計算論的学習, 培風館, 2001.
- S. Jain, D. Osherson J. S. Royer, and A. Sharma : *Systems That Learn* – 2nd Edition, MIT Press.
- 古川・植野・尾崎 : 帰納論理プログラミング, 共立出版, 2001.



## 解説

- 山本：帰納論理プログラミングの基礎理論, 人工知能学会誌, 12(5), 13-22(1997).
- H. Arimura and A. Yamamoto : Inductive Logic Programming : From Logic of Discovery to Machine Learning, IEICE Trans. Inf. and Syst. E83-D(1), 10-18 (2000).
- 有村, 平田:一階論理式の学習と帰納論理プログラミング,人工知能学会誌14, 790--799, (1999).
- 有村, 平田, 山本: 帰納論理プログラミングと証明補完, bit別冊「発見科学とデータマイニング」, 34-44 (2000).



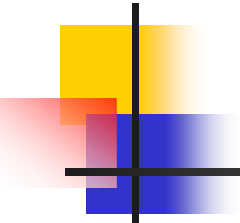
---

## 関連論文

- D. Gold: Language Identification in the Limit, *Information and Control* **10**, 447—474, 1967.
- D. Angluin: Inductive Inference of Formal Languages from Positive Data, *Information and Control* **45**, 117—135, 1980.
- J.-L.Lassez, M. J. Maher, and K. Marriott: Unification Revisited, in J. Minker. (ed): *Foundations of Deductive Databases and Logic Programming*, 587—626, Morgan-Kaufman, 1988.

## 最新の話題

- S. Hayashi : Mathematics Based on Learning, *Proc. of the 13th International Workshop on Algorithmic Learning Theory (LNAI 2533)*, 7—21, Springer-Verlag, 2002.
- E. Martin et al. : Learning in Logic with RichProlog, *Proc. of the 18th International Conference on Logic Programming (LNCS 2401)*, 239—254, Springer, 2002.

- 
- 
- Y. Akama, S. Berardi, U. Kohlenbach, S. Hayashi : An Arithmetical hierarchy of the laws of excluded middle and related principles, *Proc. of IEEE Symp. Logic in Computer Science*, pp.192—201, 2004.
  - F. Stephan and Y. Ventsov : Learning Algebraic Structures from Text, *Theoretical Computer Science* **268**, 221—273, 2001.
  - T. Sato : Parameterized Logic Programs where Computing Meets Learning, *Proc. of FLOPS2001* (LNCS 2024), pp.40-60, 2001.
  - L. de Raedt and K. Kersting: Probabilistic Inductive Logic Programming, *Proc. of the 15th International Workshop on Algorithmic Learning Theory* (LNAI 3244), 19—36, Springer-Verlag, 2002.