

自動演繹と現代社会

数学の未解決問題に AI が挑戦する時代になりました。その核心技術は機械学習に基づく仮説生成と、それを立証する自動演繹の二つです。私達は「自動演繹」の基盤理論の研究をしています。

自動演繹の原理

証明の自動化とは、証明を計算によって実現することを意味します。現代の自動演繹技法は「項書換え」と呼ばれる計算モデルの理論に立脚し、証明に必要な計算規則を自動生成して証明しています。

項書換えの例

• 書換え系

$$\begin{array}{llll}
 0 + 0 \rightarrow 0 & 1 + 0 \rightarrow 1 & \dots & 9 + 0 \rightarrow 9 & 0 : x \rightarrow x \\
 0 + 1 \rightarrow 1 & 1 + 1 \rightarrow 2 & \dots & 9 + 1 \rightarrow 1 : 0 & x + (y : z) \rightarrow y : (x + z) \\
 \dots & \dots & \dots & \dots & (x : y) + z \rightarrow x : (y + z) \\
 0 + 9 \rightarrow 9 & 1 + 9 \rightarrow 1 : 0 & \dots & 9 + 9 \rightarrow 1 : 8 & x : (y : z) \rightarrow (x + y) : z
 \end{array}$$

• 書換え

$$\begin{aligned}
 (2 : 3) + (7 : 7) &\rightarrow 7 : ((2 : 3) + 7) \rightarrow 7 : (2 : (3 + 7)) \rightarrow 7 : (2 : (1 : 0)) \\
 &\rightarrow 7 : ((2 + 1) : 0) \rightarrow 7 : (3 : 0) \rightarrow (7 + 3) : 0 \rightarrow (1 : 0) : 0
 \end{aligned}$$

研究は数独から

5	3		7				
6			1	9	5		
	9	8					6
8			6				3
4			8		3		1
7			2				6
	6					2	8
			4	1	9		5
			8		?	7	9

方程式を解く研究

$$sum(n) = 10 \implies n = 4$$

$$rev(xs) = [1, 2, 3] \implies xs = [3, 2, 1]$$

$$quicksort([3, x, 2]) = [1, 2, y] \implies \begin{cases} x = 1 \\ y = 3 \end{cases}$$

$$\begin{array}{ccc}
 \begin{matrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{matrix} & \implies & \begin{matrix} 2 & 9 & 4 \\ 7 & 5 & 3 \\ 6 & 1 & 8 \end{matrix} \\
 \text{魔方陣} & &
 \end{array}$$

プログラム解析の研究

$$quick\ sort \rightarrow \text{停止性解析} \rightarrow YES$$

$$quick\ sort \rightarrow \text{計算量解析} \rightarrow O(n^2)$$

$$merge\ sort \rightarrow \text{計算量解析} \rightarrow O(n \log n)$$

理論探査の自動化研究

数学

入力

$$\begin{aligned}
 1 \cdot x &= x \\
 x^{-1} \cdot x &= 1 \\
 (x \cdot y) \cdot z &= x \cdot (y \cdot z)
 \end{aligned}$$

出力

$$\begin{aligned}
 x \cdot 1 &= x \\
 (x \cdot y)^{-1} &= y^{-1} \cdot x^{-1} \\
 (x^{-1})^{-1} &= x \\
 \dots
 \end{aligned}$$

関数型言語

入力

$$\begin{aligned}
 [] ++ ys &= ys \\
 (x : xs) ++ ys &= x : (xs ++ ys) \\
 rev([]) &= [] \\
 rev(x : xs) &= rev(xs) ++ [x]
 \end{aligned}$$

出力

$$\begin{aligned}
 xs ++ [] &= xs \\
 rev(xs ++ ys) &= rev(ys) ++ rev(xs) \\
 rev(rev(ys)) &= ys \\
 \dots
 \end{aligned}$$

手続き型言語

入力

```

def sum(x):
  y = 0
  while x > 0:
    y = x + y
    x = x - 1
  return y
    
```

出力

$$sum(n) = \frac{n(n+1)}{2}$$

ロジックパズル

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

魔方陣

出力

$$\begin{aligned}
 x_5 &= 5 \\
 x_1 + x_9 &= 10 \\
 x_6 + x_8 &= 2x_1 \quad !!! \\
 \dots
 \end{aligned}$$

定理自動証明の研究

次の証明を自動生成するためにはどのような技術が必要でしょうか？

Theorem

$$\sum_{k=0}^n k = \frac{n(n+1)}{2} \text{ for all natural numbers } n.$$

Proof.

We use induction on n .

- If $n = 0$ then the claim is trivial.
- If $n > 0$ then

$$\sum_{k=0}^n k = n + \sum_{k=0}^{n-1} k \stackrel{\text{IH}}{=} n + \frac{(n-1)(n-1+1)}{2} = \frac{n(n+1)}{2}$$

研究への誘い

理論分野に属する研究テーマですが、数学・論理学・プログラミングのどれか一つでも好きであれば、修士研究には十分です。特に以下のトピックにに興味があれば楽しく研究できると思います。

プログラミング言語、証明支援系、数式処理、論理パズル、オートマトン、ラムダ計算、代数、グレブナ基底

修了までに身につく技能

- 数式を読み書きする能力
- 教科書・専門書を読む能力
- 数式とプログラムの対応関係の理解