

# 廣川研: Theory of Computing

Assoc. Prof. Nao Hirokawa

廣川 直 准教授

# 研究分野: 項書換え (Area: Term Rewriting)

- プログラミング言語 (Programming Languages)

# 研究分野: 項書換え (Area: Term Rewriting)

- プログラミング言語 (Programming Languages)
- 自動証明 (Automated Theorem Proving)

# プログラミング言語

(Programming Languages)

## クイックソート (C 言語)

```
void qsort(int *a, int n)
{
    if (n <= 0) return;
    int i = 0, j = n - 1, x = a[n / 2];
    do {
        while (a[i] < x) i++;
        while (x < a[j]) j--;
        if (i <= j) swap(a, i++, j--);
    } while (i <= j);
    qsort(a, j);
    qsort(a + i, n - i);
}
```

# プログラミング言語

(Programming Languages)

## クイックソート (Haskell)

```
qsort []          = []
qsort (x : xs) =
  qsort [ y | y <- xs, x < y ] ++ [x] ++
  qsort [ y | y <- xs, x >= y ]
```

# プログラミング言語

(Programming Languages)

## クイックソート (Haskell)

```
qsort []          = []
qsort (x : xs) =
  qsort [ y | y <- xs, x < y ] ++ [x] ++
  qsort [ y | y <- xs, x >= y ]
```

- 言語機能 (パターンマッチ, ラムダ, 遅延評価, ...)

# プログラミング言語

(Programming Languages)

## クイックソート (Haskell)

```
qsort []          = []
qsort (x : xs) =
  qsort [ y | y <- xs, x < y ] ++ [x] ++
  qsort [ y | y <- xs, x >= y ]
```

- 言語機能 (パターンマッチ, ラムダ, 遅延評価, ...)
- プログラム変換 (最適化, 構文解析, ...)

# プログラミング言語

(Programming Languages)

## クイックソート (Haskell)

```
qsort []          = []
qsort (x : xs) =
  qsort [ y | y <- xs, x < y ] ++ [x] ++
  qsort [ y | y <- xs, x >= y ]
```

- 言語機能 (パターンマッチ, ラムダ, 遅延評価, ...)
- プログラム変換 (最適化, 構文解析, ...)
- プログラム解析 (停止性, 計算量解析, ...)



# 計算量解析 (Complexity Analysis)



最新の技術で、どれくらい正確に自動で解析できる？

# 計算量解析 (Complexity Analysis)



最新の技術で、どれくらい正確に自動で解析できる？

- バブルソート:  $O(n^2)$

# 計算量解析 (Complexity Analysis)



最新の技術で、どれくらい正確に自動で解析できる？

- バブルソート:  $O(n^2)$
- マージソート:  $O(n^2)$

本当は  $O(n \log n)$

# 計算量解析 (Complexity Analysis)



最新の技術で、どれくらい正確に自動で解析できる？

- バブルソート:  $O(n^2)$
- マージソート:  $O(n^2)$  本当は  $O(n \log n)$
- クイックソート: 手も足も出ない 本当は  $O(n^2)$

# 計算量解析 (Complexity Analysis)



最新の技術で、どれくらい正確に自動で解析できる？

- バブルソート:  $O(n^2)$
- マージソート:  $O(n^2)$  本当は  $O(n \log n)$
- クイックソート: **手も足も出ない** 本当は  $O(n^2)$
- ユークリッドの互除法:  $O(n^2)$  本当は  $O(n)$

# 自動証明 (Automated Reasoning)

$$\left\{ \begin{array}{l} x^2y = 1 \\ xy^2 = x \end{array} \right\}$$

1 実数解  $(x, y)$  は常に  $x^4 - y^4 = 0$  を満たす？

# 自動証明 (Automated Reasoning)

$$\left\{ \begin{array}{l} x^2y = 1 \\ xy^2 = x \end{array} \right\}$$

① 実数解  $(x, y)$  は常に  $x^4 - y^4 = 0$  を満たす？

YES! 補題  $y^2 = 1, x^2 = y$  に気がつけば簡単

# 自動証明 (Automated Reasoning)

$$\left\{ \begin{array}{l} x^2y = 1 \\ xy^2 = x \end{array} \right\}$$

① 実数解  $(x, y)$  は常に  $x^4 - y^4 = 0$  を満たす?

YES! 補題  $y^2 = 1, x^2 = y$  に気がつけば簡単

②  $x^5 - y^5 = 0$  は?



# 自動証明 (Automated Reasoning)

$$\left\{ \begin{array}{l} x^2y = 1 \\ xy^2 = x \end{array} \right\}$$

① 実数解  $(x, y)$  は常に  $x^4 - y^4 = 0$  を満たす?

YES! 補題  $y^2 = 1, x^2 = y$  に気がつけば簡単

②  $x^5 - y^5 = 0$  は?

NO! 反例  $(x, y) = (-1, 1)$

# 自動証明 (Automated Reasoning)

$$\left\{ \begin{array}{l} x^2y = 1 \\ xy^2 = x \end{array} \right\}$$

① 実数解  $(x, y)$  は常に  $x^4 - y^4 = 0$  を満たす?

YES! 補題  $y^2 = 1, x^2 = y$  に気がつけば簡単

②  $x^5 - y^5 = 0$  は?

NO! 反例  $(x, y) = (-1, 1)$

■ 自動証明 (同値変形, 補題発見, ...)

# 自動証明 (Automated Reasoning)

$$\left\{ \begin{array}{l} x^2y = 1 \\ xy^2 = x \end{array} \right\}$$

① 実数解  $(x, y)$  は常に  $x^4 - y^4 = 0$  を満たす？

YES! 補題  $y^2 = 1, x^2 = y$  に気がつけば簡単

②  $x^5 - y^5 = 0$  は？

NO! 反例  $(x, y) = (-1, 1)$

- 自動証明 (同値変形, 補題発見, ...)
- 自動反証 (反例生成, 完備化, ...)

# 方程式を解く (Solving Equations)

文字列の方程式  $ax = yb$  の解は無限にある：

$$(x, y) = (b, a), (ab, aa), (bb, ab), (aab, aaa), \dots$$

一般解は？

# 方程式を解く (Solving Equations)

文字列の方程式  $ax = yb$  の解は無限にある：

$$(x, y) = (b, a), (ab, aa), (bb, ab), (aab, aaa), \dots$$

一般解は？

$$(x, y) = (zb, az) \quad z \text{ は任意の文字列}$$

# 方程式を解く (Solving Equations)

文字列の方程式  $ax = yb$  の解は無限にある：

$$(x, y) = (b, a), (ab, aa), (bb, ab), (aab, aaa), \dots$$

一般解は？

$$(x, y) = (zb, az) \quad z \text{ は任意の文字列}$$

- 解の存在条件、一般解形式の存在条件

# 方程式を解く (Solving Equations)

文字列の方程式  $ax = yb$  の解は無限にある：

$$(x, y) = (b, a), (ab, aa), (bb, ab), (aab, aaa), \dots$$

一般解は？

$$(x, y) = (zb, az) \quad z \text{ は任意の文字列}$$

- 解の存在条件、一般解形式の存在条件
- 解を実際に計算する手法

# 方程式を解く (Solving Equations)

文字列の方程式  $ax = yb$  の解は無限にある：

$$(x, y) = (b, a), (ab, aa), (bb, ab), (aab, aaa), \dots$$

一般解は？

$$(x, y) = (zb, az) \quad z \text{ は任意の文字列}$$

- 解の存在条件、一般解形式の存在条件
- 解を実際に計算する手法
- プログラムの方程式を解くには？

$$\text{qsort}([3, x, 1]) = [y, 3, 4]$$



# 組ひも理論 (Braid Theory)

## NOTATION

a:

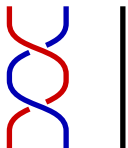


b:

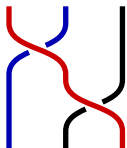


## EXAMPLE

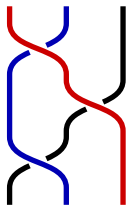
aa:



ab:



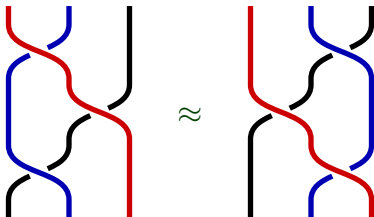
aba:



# 組ひもの等価性

## DEFINITION

$$\mathcal{B} = \{aba \approx bab\}$$



Q

$aababab \approx_{\mathcal{B}} ababab$ ?

# L-System

Koch's curve

$$F \rightarrow F+F--F+F \quad \delta = 60^\circ$$

# L-System

Koch's curve

$$F \rightarrow F+F--F+F \quad \delta = 60^\circ$$

$$F \Rightarrow F+F--F+F$$

$$\Rightarrow F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F$$

$$\Rightarrow \dots$$

# L-System

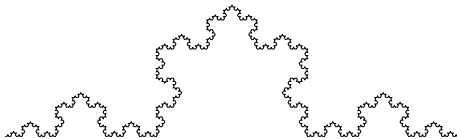
Koch's curve

$$F \rightarrow F+F--F+F \quad \delta = 60^\circ$$

$$F \Rightarrow F+F--F+F$$

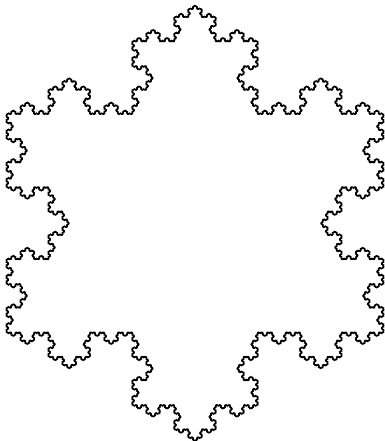
$$\Rightarrow F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F$$

$$\Rightarrow \dots$$



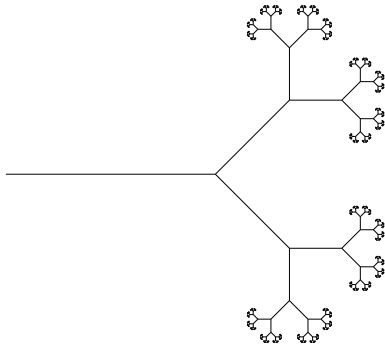
# Koch's Snowflake

$S \rightarrow F--F--F$      $F \rightarrow F+F--F+F$      $\delta = 60^\circ$



# Pythagoras Tree

$A \rightarrow B[+A]-A$      $B \rightarrow BB$      $\delta = 45$



# Fractal Plant

$S \rightarrow ++++++++X$

$F \rightarrow FF$

$X \rightarrow F-[[X]+X]+F[+FX]-X$

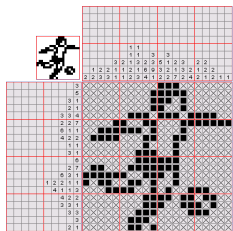
$\delta = 25$





# 1年目

- 4 —
- 5 —
- 6 —
- 7 パズルを (自動で) 解く
- 8 テーマ決め
- 9
- 10 研究
- 11 ∴
- 12 ∴
- 1 ∴
- 2 ∴
- 3 国内学会



(C) Juraj Simlovic, CC BY-SA 3.0

## 2年目

4 花見

5 ∴

6 ∴

7 国際ツールコンペティション

8 中間審査

9 国内学会

10 ∴

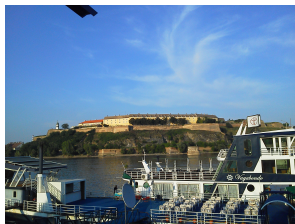
11 ∴

12 ∴

1 国際会議投稿 ...

2 修了審査

3



# 研究室セミナー (Seminars)

■ 研究室セミナー

週 1 回

# 研究室セミナー (Seminars)

- 研究室セミナー 週 1 回
- 個別ミーティング 週 0.5 ~ 1 回

# 研究室セミナー (Seminars)

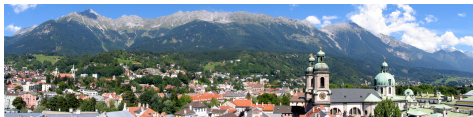
- 研究室セミナー 週 1 回
- 個別ミーティング 週 0.5 ~ 1 回
- 輪講 (小川研・寺内研合同) 週 1 回

# 研究室セミナー (Seminars)

- 研究室セミナー 週 1 回
- 個別ミーティング 週 0.5 ~ 1 回
- 輪講 (小川研・寺内研合同) 週 1 回
- 合同セミナー 月 2 回 (不定期)

# 研究協力 (Research Collaborations)

## ■ University of Innsbruck (Austria)



(C) Pahu, CC BY-SA 3.0

# 研究協力 (Research Collaborations)

## ■ University of Innsbruck (Austria)



(C) Pahu, CC BY-SA 3.0

## ■ LORIA (France)



(C) François Bernardin, CC BY 3.0



# 廣川研 (Hirokawa Laboratory)

- 研究室の場所: I-54
- <http://www.jaist.ac.jp/~hirokawa/>

以下に興味のある人は楽しく研究できると思います

- プログラミング言語
- ロジックパズル
- 数式処理システムや自動証明