

# 再帰の基本

## 1 導入

この講義で最重要なのは、再帰を「関数が自分を呼ぶ書き方」でなく、「問題を同じ形の小さい問題へ縮める考え方」として見ることである。

再帰で誤りやすいのは、停止条件を忘れてたり、何が小さくなっているかが曖昧なまま呼び出すことである。

この講義では、停止条件と再帰呼び出しの役割を分けて見る。

## 2 用語と定義

再帰とは、ある問題を、それより小さい同種の問題に分けて解く方法である。

停止条件とは、それ以上分けずにすぐ答えられる場合である。

## 3 方針

まず「どこまで小さくなれば直接答えられるか」を決める。そのあと、「大きい問題をどう1段階小さくするか」を決める。

## 4 直感的な説明

階段を1段ずつ下りるように、問題を少しずつ小さくしていき、一番下まで着いたら答えを戻していくのが再帰である。

## 5 厳密な説明

### 5.1 1. 停止条件

たとえば階乗  $n!$  なら

$$1! = 1$$

を停止条件にできる。

### 5.2 2. 再帰呼び出し

$$n! = n \cdot (n - 1)!$$

と書けるので、大きな問題  $n!$  を小さい問題  $(n - 1)!$  へ渡せる。

### 5.3 3. 正しく終わるための条件

再帰呼び出しごとに、問題の大きさが確実に減っていなければならない。ここが曖昧だと無限に呼び出して  
しまう。

## 6 見分け方

- 同じ形の小問題へ分けられるなら、再帰を疑う。
- 停止条件が1行で書けないなら、まだ設計が曖昧な可能性がある。

## 7 最終形

再帰 = 停止条件 + 小さい同種問題への帰着

## 8 一言でいうと

- 再帰では、関数の書き方より「何が小さくなっているか」を見ることが重要だ。