# Regnant: 分割可能所有権と篩型を用いた Java 検証器

小林 亮太[*1]　John Toman[*2]　五十嵐 淳[*1]　末永 幸平[*1]
[*1]京都大学　[*2]Certora

## 背景: ConSORT [Toman et al. '20]

分割可能所有権と篩型を用いた，ポインタのある手続型プログラムの検証器．

$\tau\ \mathbf{ref}^r$

- $\tau$型の値を指す参照型
- $r \in [0,1]$ ... 所有権
  - $r = 1$ ... mutable
  - $0 < r < 1$ ... Immutable かつ， mutable な参照のエイリアスになっていない
  - $r = 0$ ... immutable かつ， mutable な参照のエイリアスになっている可能性がある

let 式で所有権が割り振られる

```
{
  let x = mkref 2 in   // x : {ν : int | ν = 2} ref^1
  let y = x in {       // x : {ν : int | ⊤} ref^0, y : {ν : int | ν = 2} ref^1
    y := 5;            // x : {ν : int | ⊤} ref^0, y : {ν : int | ν = 5} ref^1
    alias(x = y);      // x : {ν : int | ν = 5} ref^0.5, y : {ν : int | ν = 5} ref^0.5
    assert(*x = 5)
  }
}
```

alias 文で所有権の再割り振りを行う

## 動機

- ConSORT は独自言語である ConSORT プログラムしか扱えない
  - Java といった広く使われている言語も扱いたい

## 今後の課題

- 更新されない変数を immutable に定義
- 関数の引数の最適化
- alias 文の自動挿入

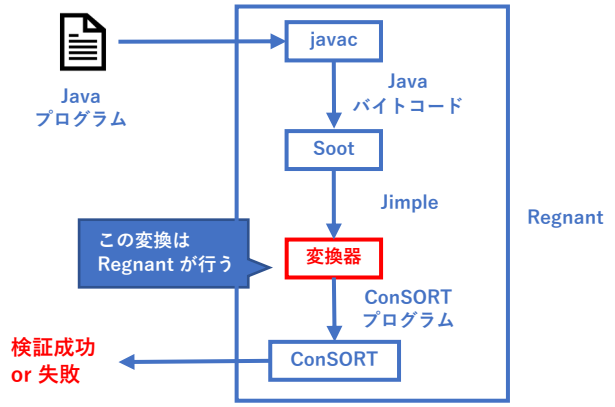
## 本研究

### 本研究の内容

- 分割可能所有権と篩型を用いた検証器 ConSORT を用いた Java プログラムの検証
- 本研究では，Java プログラムを ConSORT プログラムに変換し，それを検証器 ConSORT の入力とすることで検証結果を得る Java プログラム検証器 Regnant を開発した
  - ConSORT プログラムにはループ構文が存在しないので変換方法が非自明

### Regnant の構造

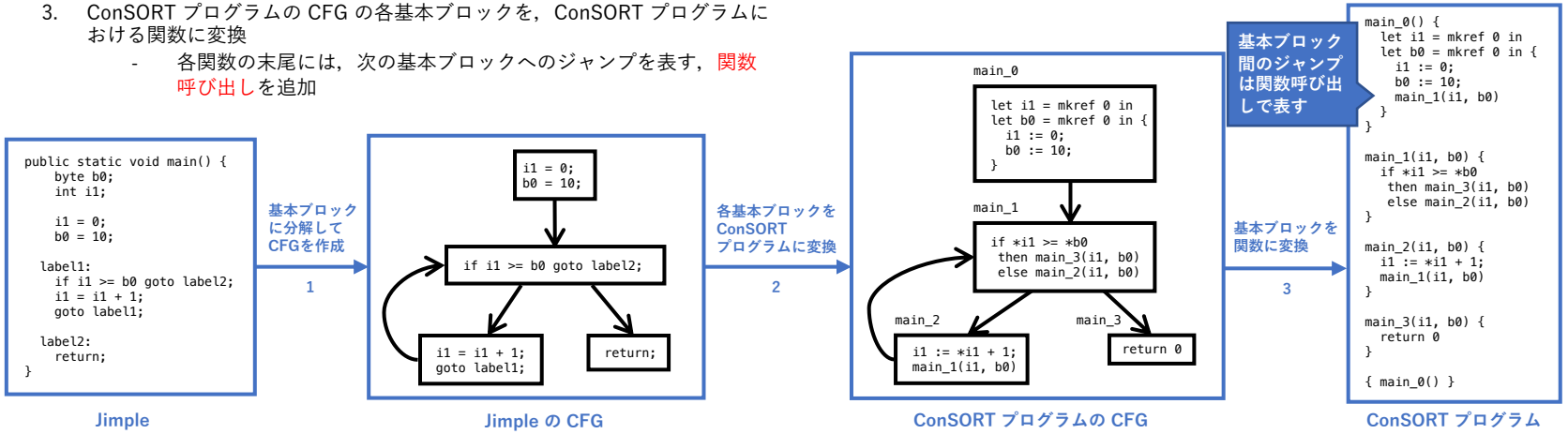
- Java バイトコードを Java の最適化フレームワーク Soot によって中間表現 Jimple に変換し，それを ConSORT プログラムに変換する




Regnant の構造

### 変換のアプローチ

1. Jimple コードを基本ブロックに分解し，CFG を作成
   - Jimple の各命令は前後の命令やジャンプ先の命令の情報を持っている
2. Jimple の CFG の各基本ブロックの命令・式を，ConSORT プログラムにおける命令・式に変換することで ConSORT プログラムの CFG に変換
3. ConSORT プログラムの CFG の各基本ブロックを，ConSORT プログラムにおける関数に変換
   - 各関数の末尾には，次の基本ブロックへのジャンプを表す，関数呼び出しを追加




Jimple から ConSORT プログラムへの変換例