プログラムを質的・量的に比較する

室屋 晃子

(京都大学 数理解析研究所)

日本ソフトウェア科学会第40回大会（東京大学）, 14 September 2023

立ち位置

- ソフトウェア科学
  - プログラミング言語理論
    - プログラム意味論：プログラムの「意味」の数学的な表現

・プログラムの入出力の対応がどのような数学的関数で表せるか？

・プログラムの入力と出力との間に成り立つ性質は何か？どう示すか？

例：入力が長さ n の配列ならば出力は長さ n の昇順の配列

・プログラムはどのような過程で実行されるか？

    - 関数型プログラムの実行過程の数学的な表現（= モデリング）

プログラムの比較

- プログラムのある種の変更には、比較がつきもの

  - リファクタリング
  - 最適化

  - 同じ実行結果で
  - より読みやすく
  - より低コストで

プログラムの比較

- プログラムのある種の変更には、比較がつきもの

  - リファクタリング
  - 最適化

  - 同じ実行結果で
  - より読みやすく
  - より低コストで

- 変更の成功 = 比較の成立

- 誰が、どうやって保証するのか？

  - プログラマ
  - 処理系開発者
  - **（システム）**

  - テスト
  - **数学的証明**

プログラムの比較

- プログラムのある種の変更には、比較がつきもの

  - リファクタリング
  - 最適化

  - 同じ実行結果で
  - より読みやすく
  - より低コストで

- 変更の成功 = 比較の成立

- 誰が、どうやって保証するのか？

  - プログラマ
  - 処理系開発者
  - **（システム）**

  - テスト
  - **数学的証明**

比較の成立を証明する
簡便な方法を提供したい

プログラムの質的な比較

- 2つのプログラムが同じ実行結果を持つか？
- （理論的な）証明手法
  - applicative/environmental bisimulation [Abramsky ’90] [Koutavas+ ’11]
  - logical relation [Plotkin ’73] [Statman ’85]
  - 様々な（関数型）プログラムに対応

プログラムの質的な比較

- 2つのプログラムが同じ実行結果を持つか？
- 自動化可能な証明手法
  - algorithmic game semantics [Abramsky ’01] [Jaber ’20]
    - シンプルな型を持つ関数型プログラムに対応
  - simulation discovery [Fedyukovich+ ’15]
    - Cプログラムを対象、SMT-based
  - regression verification [Godlin+ ’09]
    - ほとんど同一なプログラムを対象

プログラムの質的・量的な比較

- 2つのプログラムが同じ実行結果を持ち、かつ**片方がより低コスト**か？
- （理論的な）証明手法
  - (time) improvement [Moran+ ’99]
  - space improvement [Schmidt-Schauß+ ’17]
  - 純粋な（必要呼び）関数型プログラムに対応

プログラムの質的・量的な比較

- 2つのプログラムが同じ実行結果を持ち、かつ**片方がより低コスト**か？
- （自動化可能な？）証明手法
  - preorder-constrained simulation [M., Sanada & Urabe, unpublished yet]
  - 一部の副作用を持つ関数型プログラムに対応

プログラムの質的・量的な比較の例

- $P \preceq Q \iff$ Pがある値nに評価されるとき、

  Qはより少ない手数でnに評価される

- $P \preceq Q$ となる例：

  - $P \equiv 1 \times 3 + 2 \times 3 \to 3 + 2 \times 3 \to 3 + 6 \to 9$
  - $Q \equiv (1 + 2) \times 3 \to 3 \times 3 \to 9$

プログラムの質的・量的な比較の例

- $P \preceq Q \iff$ Pがある値nに評価されるとき、

  Qはより少ない手数でnに評価される

- $P \preceq Q$ となる例：

  - $P \equiv \mathrm{let}\ x = 3\ \mathrm{in}\ x + x \to 3 + 3 \to 6$
  - $Q \equiv 3 + 3 \to 6$

プログラムの質的・量的な比較の例

- $P \preceq Q \iff$ Pがある値nに評価されるとき、

  Qはより少ない手数でnに評価される

- $P \preceq Q$ となる例：

  - P ≡ let rec sum’ n =

    if flip () **then** error **else** (**if** n > 0 **then** n + sum’ (n - 1) **else** n)

    in sum’ 5

  - Q ≡ let rec sum n =

    if flip () **then** n **else** (**if** n > 0 **then** n + sum (n - 1) **else** n)

    in sum 5

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ Pがある値nに評価されるとき、

  Qはより少ない手数でnに評価される

- 「評価する」：プログラムの段階的な書き換え
  - $P \equiv 1 \times 3 + 2 \times 3 \to 3 + 2 \times 3 \to 3 + 6 \to 9$
  - $Q \equiv (1 + 2) \times 3 \to 3 \times 3 \to 9$

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ Pがある値nに評価されるとき、

  Qはより少ない手数でnに評価される

- 「評価する」：~~プログラムの段階的な書き換え~~　オートマトンの状態遷移
  - P ：$p1 \to p2 \to p3 \to p4 \xrightarrow{9} p!$
  - Q ：$q1 \to q2 \to q3 \xrightarrow{9} q!$

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ p1からある値nを通ってp!に着くとき、

  q1からより少ない遷移でnを通ってq!に着く

- 「評価する」：~~プログラムの段階的な書き換え~~　オートマトンの状態遷移
  - P ：$p1 \to p2 \to p3 \to p4 \xrightarrow{9} p!$
  - Q ：$q1 \to q2 \to q3 \xrightarrow{9} q!$

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ p1からある値nを通ってp!に着くとき、

  q1からより少ない遷移でnを通ってq!に着く

  $\Longleftarrow$ $(p1,q1) \in R$ となる *preorder-constrained simulation* R が存在する

- 「評価する」：~~プログラムの段階的な書き換え~~　オートマトンの状態遷移
  - P ：p1 → p2 → p3 → p4 $\xrightarrow{9}$ p!
  - Q ：q1 → q2 → q3 $\xrightarrow{9}$ q!

R = (p1,q1), (p!,q!)

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ p1から ある値nを通ってp!に着くとき、

  q1からより少ない遷移でnを通ってq!に着く

  $\Longleftarrow$ (p1,q1) ∈ R となる *preorder-constrained simulation* R が存在する

- 「評価する」：~~プログラムの段階的な書き換え~~　オートマトンの状態遷移
  - P ：$p1 \rightarrow p2 \xrightarrow{L} p31 \rightarrow p41 \xrightarrow{3} p!$

    $\quad {}_R\searrow p32 \rightarrow p42 \nearrow_4$
  - Q ：$q1 \xrightarrow{L} q21 \rightarrow q31 \rightarrow q41 \xrightarrow{3} q!$

    $\quad {}_R\searrow q22 \rightarrow q32 \rightarrow q42 \nearrow_4$

R = (p1,q1), (p31,q31), (p32,q32), (p!,q!)

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ p1からある値nを通ってp!に着くとき、

  q1からより少ない遷移でnを通ってq!に着く

  $\Longleftarrow$ $(p1,q1) \in R$ となる *preorder-constrained simulation* R が存在する

- 「評価する」：~~プログラムの段階的な書き換え~~　オートマトンの状態遷移
  - P ：p1 → p2 → p3 → p4 $\xrightarrow{9}$ p!
  - Q ：q1 → q2 → q3 $\xrightarrow{9}$ q!

R = (p1,q1), (p!,q!)

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ p1からある値nを通ってp!に着くとき、

  q1からより少ない遷移でnを通ってq!に着く

  $\Longleftarrow$ $(p1,q1) \in R$ となる a *preorder-constrained simulation* R が存在する

  $\iff$ p1q1 から始めて a *two-player reachability game* G(R) に勝利する

  - p1q1から始めてそれぞれ矢印に沿って進み、Pの進みかたをQが真似できればQの勝ち

- 「評価する」：~~プログラムの段階的な書き換え~~　オートマトンの状態遷移

  - P ：$p1 \to p2 \to p3 \to p4 \xrightarrow{9} p!$
  - Q ：$q1 \to q2 \to q3 \xrightarrow{9} q!$

R = (p1,q1), (p!,q!)

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ p1からある値nを通ってp!に着くとき、

  q1からより少ない遷移でnを通ってq!に着く

  $\Longleftarrow$ $(p1,q1) \in R$ となる a *preorder-constrained simulation* R が存在する

  $\iff$ p1q1 から始めて a *two-player reachability game* G(R) に勝利する

  - p1q1から始めてそれぞれ矢印に沿って進み、Pの進みかたをQが真似できればQの勝ち

- G(R)の「譜面」

  - p1q1 → p2q1 → p2q2 → p3q2 → p3q3 → p4q3 → p4q3 → p!q3 → p!q! → WIN
    ↘ p3q2 → p4q2 → p4q2 → p!q2 ↗

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ p1からある値nを通ってp!に着くとき、

  q1からより少ない遷移でnを通ってq!に着く

  $\Longleftarrow$ $(p1,q1) \in R$ となる a *preorder-constrained simulation* R が存在する

  $\iff$ p1q1 から始めて a *two-player reachability game* G(R) に勝利する

  $\iff$ p1q1から始めてG(R)の譜面において WIN に辿り着く

- G(R)の「譜面」

  - p1q1 → p2q1 → p2q2 → p3q2 → p3q3 → p4q3 → p4q3 → p!q3 → p!q! → WIN

    ↘ p3q2 → p4q2 → p4q2 → p!q2 ↗

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \iff$ p1からある値nを通ってp!に着くとき、

  q1からより少ない遷移でnを通ってq!に着く

  $\Longleftarrow$ $(p1,q1) \in R$ となる a *preorder-constrained simulation* R が存在する

  $\iff$ p1q1 から始めて a *two-player reachability game* G(R) に勝利する

  $\iff$ p1q1から始めてG(R)の譜面において WIN に辿り着く

- G(R)の「譜面」

  有向グラフの到達性問題に帰着！

  - p1q1 → p2q1 → p2q2 → p3q2 → p3q3 → p4q3 → p4q3 → p!q3 → p!q! → WIN

    ↘ p3q2 → p4q2 → p4q2 → p!q2 ↗

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \Longleftarrow$ 有向グラフにおける WIN への到達性

  - P, Q の評価過程の状態総数に対して、多項式時間で解ける

    プログラムの大きさでは抑えられない

  - P, Q に対して有向グラフが完全に分かっている必要がある

    つまり、P, Q の評価過程を全て求める必要がある

    全パターンを網羅するテストと同等の労力

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \Longleftarrow$ 有向グラフにおける WIN への到達性
- $P \preceq Q$ となる例：
  - $P \equiv$ let rec sum' n =

```
    if flip () then error else (if n > 0 then n + sum' (n - 1) else n)
in sum' 5
```

  - $Q \equiv$ let rec sum n =

```
    if flip () then n else (if n > 0 then n + sum (n - 1) else n)
in sum 5
```

時間計算量はsum, sum' の引数に対して線形に増加

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \Longleftarrow$ 有向グラフにおける WIN への到達性
- $P \preceq Q$ となる例：

時間計算量はfib, fib’ の引数に対して指数的に増加

  - P ≡ let rec fib’ n =

```
    if flip () then error else (if n > 0 then fib’ (n-2) + fib’ (n - 1) else n)
in fib’ 5
```

  - Q ≡ let rec fib n =

```
    if flip () then n else (if n > 0 then fib (n-2) + fib (n - 1) else n)
in fib 5
```

Preorder-constrained simulationによる比較の証明

- $P \preceq Q \Longleftarrow$ 有向グラフにおける WIN への到達性

  - P, Q の評価過程の状態総数に対して、多項式時間で解ける

    プログラムの大きさでは抑えられない

  - P, Q に対して有向グラフが完全に分かっている必要がある

    つまり、P, Q の評価過程を全て求める必要がある

    全パターンを網羅するテストと同等の労力

プログラムの質的・量的な比較

- 2つのプログラムが同じ実行結果を持ち、かつ**片方がより低コスト**か？
- （自動化可能な？）証明手法
  - preorder-constrained simulation [M., Sanada & Urabe, unpublished]
  - 一部の副作用を持つ関数型プログラムに対応
  - 全パターンを網羅するテストと同等の労力がかかる