

定理証明器 J-Bob の類型証明ステップをまとめるための機能拡張

本嶋 琉乃 202103341

1 はじめに

定理証明器とは、数学的または論理的な定理や主張を自動的に証明するためのシステムであり、与えられた仮説と定義に基づいて、論理的に正しい一連の証明手順を自動、または半自動で導出する。本研究では、The Little Prover [1] に登場する定理証明器 J-Bob において、類型証明ステップを必要とするパターンを短縮するための拡張機能を検討し、可読性の向上を目指す。

2 背景

J-Bob は、[1] に登場する定理を証明するための定理証明器であり、簡素な定理証明支援系である。一方、繰り返しが多い証明ステップが存在し、証明ステップが冗長になるパターンがある。if の持ち上げを表すコードでは、以下ようになる。

```
dethm.memb?/remb1:
((1)
 (if-same (equal x1 '?))
 (memb?
  (if (equal x1 '?))
    (remb '())
    (cons x1 (remb '())))))
((1 A 1)
 (if-nest-A
  (equal x1 '?))
 (remb '())
 (cons x1 (remb '()))))
((1 E 1)
 (if-nest-E
  (equal x1 '?))
 (remb '())
 (cons x1 (remb '()))))
```

このように if の持ち上げは if-same, if-nest-A, if-nest-E の 3 つの公理を記述する必要があり、冗長である。これを 1 つの証明ステップで処理可能とするため、定理 if-lifting として定義し、証明ステップの短縮を図る。

3 dethm の機能拡張

J-Bob では、階層的な証明ステップを記述する際にフォーカス（記述した証明ステップを適用する文）へ至るパスを入れることで、公理を呼び出して対象のフォーカスに適用することが出来る。これを利用し、定理 if-lifting を定義する。

```
(dethm if-lifting (x y z)
 (and
  (() (if-same x y))
  (() (if-nest-A x y z))
  (() (if-nest-E x y z))))
```

if-lifting を定義するにあたって、and を全域なものに変更する必要がある（引数として任意の値が渡された場合でも、必ず t または nil を返す必要がある）ため、次のように定義した。

```
(define (and x y)
 (if x
  ;; x が 't の場合のみ y を評価
  (if y 't 'nil)
  ;; x が 'nil なら即座に 'nil を返す
  'nil))
```

if-lifting の実装において、以下 2 つの課題が挙げられる。

1. パスの動的生成

階層的な証明ステップにおいて、公理を呼び出して対象のフォーカスに適用させるため、フォーカスパスを動的に更新する必要がある。

2. rewrite/steps との結合

rewrite/steps は、揭示された証明のステップに従って主張を書き換える関数である。証明が完了するか、主張が変化しなくなるまで書き換えを繰り返す。if-lifting を証明ステップで利用するには、rewrite/steps を修正し、自動的に if-same などの公理を展開する仕組みが必要である。よって、既存の rewrite/steps に統合する方法を確立する必要

がある。

実装を試験的に適用した例を以下に挙げる。

```
((1)
  (if-lifting (equal x1 '?))
    (remb '())
    (cons x1 (remb '()))))
```

このように、複数の公理適用を一つのステップにまとめることで、階層的な証明ステップを記述する際の可読性の向上が実現できる。

4 議論

本研究の成果は、J-Bob の証明ステップの短縮に向けた基盤的な取り組みであり、他の証明環境にも応用可能な示唆を与えるものである。

研究の過程では、if-lifting を設計する際に、複数の条件を効率的に評価するための関数 `and` を全域的に再定義した。この全域的な `and` によって、引数が任意の値であっても正しい結果を返す信頼性が実現され、if-lifting の基盤を構築することができた。しかし、現時点では if-lifting の実装が未完成であり、動作を確認するには至っていない。

さらに、if-lifting の適用範囲を検証するためには、より多くのテストケースを用意する必要がある。現在の段階では、限定的な条件下での動作を想定しているが、複雑な条件分岐や異なる証明パターンに対しても柔軟に対応できるかを検証する必要がある。

5 まとめと今後の課題

本研究では、定理証明器 J-Bob における類型証明ステップの可読性の向上を目的として、if の持ち上げに関する拡張定理 if-lifting の設計を試みた。この拡張機能は、if-same, if-nest-A, if-nest-E という 3 つの公理を 1 つの証明ステップに統合し、繰り返しの多い証明ステップを短縮することを目指している。このアプローチにより、特に階層的な証明において、記述量の削減による可読性の向上、記述ミスの防止が期待される今後、if-lifting の実装を完成させ、証明ステップの短縮だけでなく、さらなる汎用性を持つ拡張機能としての開発を目指したい。

最後に、課題として特にフォーカスへ至るパスを動的に生成し、証明対象が複雑な場合にも正確に管理する仕組みの構築が挙げられる。現在の実装では、階層的な条

件分岐に十分対応できておらず、この点が大きな障壁となっている。また、rewrite/steps をはじめとする既存の証明ステップ処理と if-lifting を統合し、証明プロセスの中で自動的に展開・適用される仕組みを実現することが求められる。この統合により、より多くの証明例への適用可能性が広がると考えられる。

参考文献

- [1] Daniel P.Friedman and Carl Eastlund, 中野圭介 (訳), 定理証明手習い:”The Little Prover”, ラムダノート株式会社, 2017.