4L-02

文字列一致による等価性判定のための形式仕様の正規化

檜垣 廉[†]
電気通信大学情報理工学域

織田 健[‡]
電気通信大学大学院情報学専攻

1 はじめに

近年のソフトウェアの大規模化と複雑化による信頼性や開発効率の低下に対し、形式手法と部品再利用が注目されている。我々は、形式手法 B Method の信頼性保証の枠組みを利用し、部品再利用により高信頼ソフトウェアを合成する MSSS 手法を提案している。この手法では、自然言語の曖昧性を除いた形式的な部品検索を行うことで、信頼性の高いソフトウェアを開発できる。本研究では、文字列一致によってソフトウェアの検索を行うための形式仕様の正規化手法を提案する。

2 背景と研究の目的

2.1 部品再利用

ソフトウェア部品の再利用は、開発の効率化に加えソフトウェアの信頼性確保にも有用である。しかし、必要な部品や使用する部品の仕様は曖昧性を排除しにくい自然言語によって説明されている。

2.2 形式手法 B Method

B Method は、集合論と一階述語論理に基づいた仕様記述と、抽象仕様から実装への段階的詳細化が特徴の形式手法である [1]。詳細化の各段階で定理証明器により仕様の無矛盾性と詳細化の整合性を証明する。仕様は不変条件や操作等からなり、B 言語で記述される。本研究では B Method の開発環境として Atelier B を用いた。

2.3 MSSS 手法

MSSS 手法は、B Method の信頼性保証の枠組みを用いて高信頼部品の生成と要求仕様からのソフトウェアの合成を行う手法である [2]。要求仕様を満たす部品の検索は、計算量削減のため定理証明を使わず文字列一致によって行う。三鍋は、リポジトリへの部品の登録前や部品検索前に数学的に等価な仕様が文字列上でも一致するように字面を統一する手法を提案している [3]。

2.4 研究の目的

三鍋の手法では、書き換え規則の不足により、数学的に等価な仕様を文字列上で一致させられない場合が多かった。また、書き換えの規則が何らかの尺度において不足しているかどうか確かめる方法も示されていなかった。本研究では、MSSS 手法の字面統一に改良を施し、より幅広い表現を正規形に統一する手法を提案する。

Standardization of Formal Specification for Mathematical Equivalence Determination by String Matching

[†]Ren Higaki, The University of Electro-Communications, School of Informatics and Engineering

[‡]Takeshi Oda, The University of Electoro-Communications, Graduate School of Information and Communication Engineering

3 従来の字面統一

3.1 従来手法の概要

三鍋の手法では、字面統一は以下の工程からなる [3]。

プリミティブ化 高機能な演算子を用いた演算をより低機能な演算の組み合わせに書き換える。

簡約化 プリミティブ化と推論の途中で適宜冗長な表現を削除する。書き換えのルールはプリミティブな演算子のみを対象とする。

推論 暗黙の条件を導出し、条件に書き加える。

構文の整列 可換な演算子のオペランドを並び替える。

変数名の置換 変数名をその出現順により一意に定める。

3.2 従来手法の課題

従来手法の字面統一には以下のような課題がある。

代入文の書き換えの欠落 書き換え対象は式にとどまり、代入演算子を跨ぐ表現は対象外だった。

属する集合と型の同一視 B 言語では、型を表す集合に属す条件により変数等の型を示す。条件式では最初に型を示す必要があるが、従来手法では条件式の整列で型が特定できない順番になることがある。

プリミティブな演算子の妥当性 全ての演算子についてプリミティブか否か網羅的に示されていない。

可換な演算子への対応 加算や乗算等の可換な演算のオペランドを構文木上で順序のない集合として扱っているため、減算のように連続した際に 2 個目以降のオペランドが可換になる演算に対応できない。

算術演算の書き換え規則の不足 整数や実数の算術演算の書き換え規則がほぼ存在しない。

合流性の保証 同じ式に対し複数の書き換え規則が適用できる時、どちらを適用しても最終的な結果が同じになるような性質 (合流性) が保証されていない。

4 形式仕様の正規化手法

4.1 手法の概要

本手法では、従来手法の字面統一の流れを踏襲しつつ、前段に代入文の書き換えと型推論を加え、プリミティブ化と簡約化の部分を改良する。従来手法では、プリミティブ化と推論の途中に適宜簡約化を行っていたが、本手法ではプリミティブ化後に簡約化を式が書き換えにより変化しなくなるまで繰り返し行う。

4.2 代入文の書き換え規則

最初に、図 1 のように従来は未対応であった代入演算子を跨ぐ表現を書き換える。式の書き換えより先に行うのは、代入文と式の書き換え規則の合流性の保証を別々に行うためである。

4.3 型推論の導入

次に型推論を行う。B 言語の型は、基本型と型の直積型と型の冪集合型からなる。全ての変数や定数に型情報

$$fn(xx) := yy$$
$$\Rightarrow\ fn := fn \lessdot \{xx \mapsto yy\}$$
(上書き演算子)

図 1: 代入演算子を跨ぐ表現に対する書き換え

$$\begin{cases} x \times y/x \longrightarrow y & (1) \\ x \times (y+z) \longrightarrow x \times y + x \times z & (2) \end{cases}$$

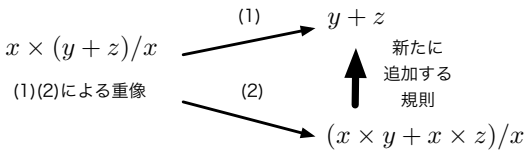

図 2: Knuth-Bendix の完備化手続きの例

を与えることで、型について常に定理証明器で証明できるように型情報を条件式で先に提示できる。また、集合かスカラーかによって異なる演算を意味する演算子に関する書き換え規則にも対応できるようになる。

4.4 プリミティブ化

プリミティブ化では、予め定めたプリミティブな演算子のみを用いるように、数学的な意味を保ったまま式を書き換え、演算の表現の幅を狭くする。従来手法に対して、全ての演算子を網羅するように規則を追加し、プリミティブな論理演算子から含意を表す $\Rightarrow$ を除くなどの変更を加える。この演算子はプリミティブ化後は他の論理演算子を組み合わせて表されるが、B 言語の構文規則上全称量化時に必要になるため、書き換えに影響しないよう全称量化子と 組み合わせて用いる文法上の記号として扱うことにした。さらに、除算や減算等の 2 個目以降のオペランドの可換性を既に対応済みの可換な演算子の可換性に帰着させるような書き換えをプリミティブ化時に行う。例えば、$a - b - c$ のようなスカラーの減算は $a + (-b) + (-c)$ のように単項マイナスと加算の組み合わせに変換し、$a, (-b), (-c)$ を同等に扱う。

4.5 簡約化

簡約化では、数学的な意味を保ちつつ冗長な表現を削除するような書き換えを行う。書き換え規則は、書き換え後の式構造が予め定めた正規形になるように新たな規則を追加したものを用いた。

4.6 規則の追加と合流性の保証

プリミティブ化・簡約化で用いる新しい書き換え規則には、Knuth-Bendix の完備化手続きを施す。この手続きでは、既存の 2 つの書き換え規則からどちらも適用可能な形である重像を生成し、それぞれの適用後の結果が合流しないならば結果の間を繋ぐ規則を追加する (図 2)。これを繰り返すことで、合流性の保証された書き換え系が得られる。

5 実験

字面統一の性能向上確認のため、予め無矛盾性を証明した数学的に等価で字面上異なる形式仕様を複数用意し、従来手法のプリミティブ化と簡約化を施した結果と、本手法による書き換えの結果を比較した。実験によって書き換えられた代入文の例を図 3 に示す。代入文 (1) は電子決済の割引と税金を考慮した支払額、代入文 (2) は学

$$pay := pay \lessdot \{order \mapsto (price - discount) \times tax\} \quad (1)$$
$$\Downarrow \text{書き換え}$$
同じ文構造
$$\begin{cases} pay := pay \lessdot \{order \mapsto price \times tax + (-discount) \times tax\} \\ att\_pt := att\_pt \lessdot \{student \mapsto class \times pt + (-abs) \times pt\} \end{cases}$$
$$\Uparrow \text{書き換え}$$
$$att\_pt(student) := class \times pt - abs \times pt \quad (2)$$

図 3: 実験によって書き換えられた代入文の例

校成績の出席点と、全く異なる対象を扱っているが、数学的には等価であり、書き換えにより同じ構造の代入文になった。この後変数名を置換すると字面が等しくなる。書き換え後は定理証明器を用いて無矛盾性の証明を行い、信頼性が保たれるか確認した。

6 考察

6.1 字面統一と信頼性保持

本研究で扱った範囲である、従来手法におけるプリミティブ化と簡約化の部分は、特に操作において適用範囲が大きく広がり、改善が見られた。書き換え規則自体の改良の他、簡約化を変化しなくなるまで繰り返し行うという適用方針も理由の一つだと思われる。今回は推論は行わなかったため、暗黙の条件が必要とされるような検索には現状では対応できない。構文の整列には手を加えられなかったため、可換な演算子のオペランドが入れ替わった状態で対応する変数が同じ位置にならなかったことがあった。新たに証明責務が生じることはあったが、全て証明可能であった。実験の範囲内では、書き換えによって信頼性が損なわれることはなかった。

6.2 条件付き書き換え規則の合流性保証

本研究では、簡単のため合流性の保証は無条件で行える書き換え規則のみを考慮した。今後は、特定の条件下において可能になる書き換え規則への対応が必要である。

6.3 局所変数の識別子の重複への対応

B 言語では他の変数と名前の重複する局所変数を宣言できる。現状では変数名を用いて同一の変数を探しているため、重複すると適切な型推論や書き換えが行えない可能性がある。現時点で考えられる対応策は、手法適用前に変数名を重複しないように置換する方法である。

7 終わりに

本研究では文字列一致による仕様の等価性判定のための正規化手法を提案した。同様の手法で一般的な言語における文字列一致による等価性判定への応用が期待できる。書き換えの適用範囲は大きく広がったが、適切な構文のソートが今後の主たる課題である。

参考文献

[1] 来間啓伸. B メソッドによる形式仕様記述. 近代科学社, 2007.

[2] 中村丈洋. B Method における部品再利用によるソフトウェア合成と高信頼ソフトウェア部品の整備. 電気通信大学 電気通信学研究科 博士 (工学) 学位論文. 2013.

[3] 三鍋孝介. 文字列一致による数学的等価性判定可能なモデル分割アルゴリズム. 第 12 回情報科学技術フォーラム論文集 vol.1 pp.271-272. 2013.