4L-07

形式的ソフトウェア合成手法における細分化モデルの可読性向上手法

結城 翔† 織田 健‡
電気通信大学情報理工学域 電気通信大学大学院情報学専攻

1 はじめに

近年、ソフトウェア開発の現場ではソフトウェアの大規模化や複雑化に伴い、開発コストの増大が問題になっている。それに対し我々は、既存ソフトウェアの部品再利用によるソフトウェア自動合成手法として MSSS 手法 [1] を提案している。MSSS 手法において必要な部品を再利用できず不足した部品は人が記述するが、文字列一致による細分化モデルの検索を可能にするために字面統一が施された式は、人には難解な表現になる。そこで、本研究では不足部品の式の可読性向上手法を提案する。

2 背景と目的

2.1 B Method

B Method[2] は数学の集合論に基づく形式手法の一種で、抽象的な最初の抽象機械「モデル」と最終的な「実装」、その間で段階的に詳細化された仕様である「リファインメント」から成る。まず無矛盾なモデルを記述し、それを段階的に詳細化することにより、最終的な実装の整合性が保証される。これらの記述には B 言語を用いる。

モデルはソフトウェアの仕様を数学的に、且つ抽象的に記述したものである。モデルはシステムが常に満たす必要がある不変条件を定義する INVARIANT 節、定数や集合に関して型等の制約を与える PROPERTIES 節、操作を記述する OPERATIONS 節やその中で各操作の事前条件を記述する PRE 節など、複数の節から構成される。実装は抽象的なモデルを C 言語などのソースコードに変換できる形にまで詳細化したものである。

2.2 MSSS 手法

MSSS 手法 [1] はソフトウェア部品の再利用の自動化を行う手法である。この手法は部品を整備する MSFC 生成とソフトウェアを合成する MSSS の 2 つから成る。

MSFC 生成はモデル細分化・実装抽出・部品登録の 3 つから成る。無矛盾性を証明済みのモデルと対応する実装を入力し、モデルを相互に非依存な代入文単位まで細分化 [3] した後、各細分化モデルに対応する細分化実装を抽出し、それらの組を部品としてリポジトリに登録する。MSSS 手法におけるモデル細分化は、部品の粒度を細かくし、部品の再利用性の向上と数学的に等価なまま文字列での部品検索を可能とする目的がある。しかし、部品検索を可能とするために字面統一が施された式は、人には難解な表現になる。

MSSS ではモデルとして与えられた要求を細分化し、それらをキーとして字面一致する部品を検索する。得られた候補から、データの詳細化方法などが共通する部品を選択・結合することで要求モデルを満たす実装を出力する。該当する部品が無い場合は不足部品を人が記述する。

2.3 研究の目的

本研究では、人には難解な表現である不足部品の式を人が読みやすい式に書き換えることを目的とする。

3 先行研究と課題

3.1 モデル細分化

モデル細分化は以下の工程で行われる。この中で本研究に関連する工程 (2〜4) に関して特に詳しく述べる。

1. 非決定的値生成の分離
2. **プリミティブ化**
3. **式構造統一化**
4. **暗黙の条件の抽出**
5. 操作分割
6. 制約条件抽出
7. 構文整列
8. 識別子置換

プリミティブ化では限られた演算子の集合であるプリミティブな演算子のみになるように式を書き換える。以下に書き換え規則の例を示す。

$$x \subset y \quad \Rightarrow \quad x \in \mathrm{POW}(y) \wedge (x \neq y)$$
$$x \bmod y \quad \Rightarrow \quad x - y \times x/y$$

式構造統一化では、規則に基づいて式を書き換えて数学的に等価である式の構造を統一化する。

暗黙の条件の抽出では、暗黙的に成り立つ条件を明示するために、B Method の開発環境である Atelier B の推論ルールを基に整備した式の推論規則 [4] をモデルの全ての条件式に適用する。例えば式 $a \leq b \wedge b < c$ からは式 $a < c$ が導出され、さらにプリミティブ化することで式 $a \leq c \wedge (a \neq c)$ が得られる。モデル細分化は一代入文単位で行われ、それぞれの代入文で用いられる変数のみで構成された条件式が抽出される。ここで変数 a, c からなる代入文に対応する前述の式 $a \leq c \wedge (a \neq c)$ は要求モデルには存在しない式となる。

3.2 書き換え履歴を利用した可読性向上

研究室内では、字面統一の変換履歴を活用する手法が提案されていた。この手法では細分化前の式に対する項の書き換えをスタックに保存する。プリミティブ化された式に対しスタックを参照しながら逆の規則を施すことで、元の式の再現を目指した。

3.3 課題

前節の手法では要求モデルの式 21 行に対し字面統一したモデルが 44 行となったが、手法適用後も 42 行あり大きな効果はなかった。また、元の式を再現できない場合の対応が未提案であった。

A Method for Improving Readability of Sliced Models in Formal Software Synthesis Methods

†Sho Yuki, The University of Electro-Communications, School of Informatics and Engineering

‡Takeshi Oda, The University of Electro-Communications, Graduate School of Information and Communication Engineering

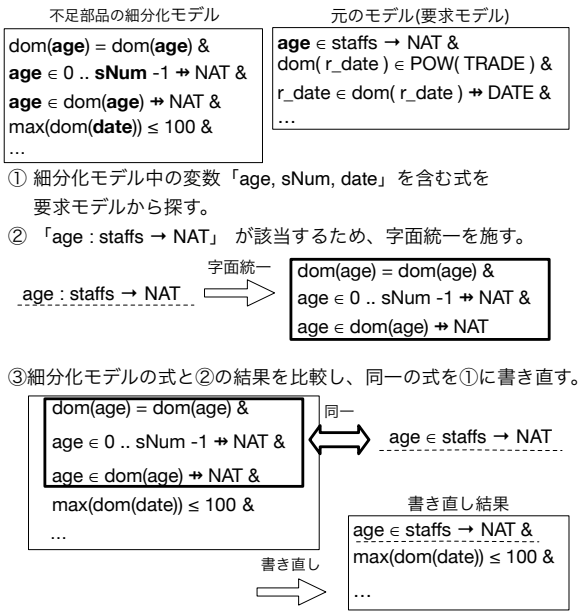

図 1: 字面統一後の変数と対応する制約条件

4 細分化モデルの可読性向上手法

4.1 手法の方針

可能な限り要求モデルの式を再現するために、字面統一後の変数に注目した制約条件抽出を行う (4.2 節)。また、完全な再現ができない式に関しては、適用後に式の行数が少なくなり、かつ簡明な表現になるような推論規則を整備し、その適用による式の書き換えを行う (4.3 節)。以上の操作により、なるべく要求モデルの式を用いつつ、少ない行数で数学的に等価なまま式の書き直しができることを、可読性向上の評価指標とする。

4.2 字面統一後の変数に注目した制約条件抽出

図 1 のように字面統一後の式に含まれる変数に注目し、その変数のみを使用している元のモデルの制約条件を細分化前の式の中から抽出する。それらに対し字面統一を施し、その結果導出された式群が全て細分化モデルの制約条件の式に含まれていた場合は、それらの式群にチェックをつけた後、字面統一前の制約条件を書き加える。

4.3 推論規則の適用による書き換え

3.1 節で説明した式の推論規則のうち、プリミティブ化ルールの逆である推論規則を整備する。また、4.2 節で説明した操作を行った際、細分化モデルの中でまだチェックのついていない式を 1 つ以上含む複数の式に対して、整備した推論規則を適用して書き加える。推論規則を適用した式にはチェックをつける。

4.4 可読性向上アルゴリズム

細分化モデルの式と 4.2 節及び 4.3 節の操作で書き加えられた式から、チェックのついている式を削除する。これを細分化モデルの可読性向上アルゴリズムとする。

5 実験

細分化モデルの可読性向上確認のため、従来の提案手法における実験で用いられていたものと同じモデルを用意し、4 章で説明した手法を適用した。実験はモデルの PROPERTIES 節、INVARIANT 節、PRE 節に対して行った。実験によって書き換えた式の例として、INVARIANT 節の実験結果を以下に示す。

$$\begin{cases} \mathrm{dom(r\_date)} = \mathrm{dom(r\_date)} \quad \& \\ \mathrm{r\_date} \in \mathrm{dom(r\_date)} \nrightarrow \mathrm{DATE} \quad \& \\ \mathrm{r\_date} \in \mathrm{dom(r\_date)} \nrightarrow 0..100 \quad \& \\ \mathrm{dom(r\_date)} \in \mathrm{POW(TRADE)} \quad \& \\ \mathrm{dom(r\_date)} \in \mathrm{POW(PERSON)} \quad \& \\ \mathrm{dom(r\_date)} \in \mathrm{POW(COMMODITY)} \quad \& \\ \mathrm{dom(r\_date)} \in \mathrm{POW(1..100)} \end{cases}$$

$$\Downarrow$$

$$\begin{cases} \mathrm{r\_date} \in \mathrm{dom(r\_date)} \rightarrow \mathrm{DATE} \quad \& \\ \mathrm{dom(r\_date)} \in \mathrm{POW(TRADE)} \end{cases}$$

この細分化モデルの INVARIANT 節の書き直し結果は、要求モデルの INVARIANT 節と完全には一致しなかったが、数学的に等価であった。また、従来手法の実験結果 3 行に対し、本手法では 2 行と行数が少なくなった。

6 考察

6.1 信頼性の保持

PROPERTIES 節と PRE 節は元のモデルの制約条件と完全に一致し、INVARIANT 節は元のモデルと数学的に等価な結果となった。このことから、本実験の範囲では書き換えにより信頼性が失われることはなかった。

6.2 可読性の向上

従来手法による実験結果と比較して、本実験の書き換え結果は信頼性を保持しながらより少ない行数となった。このことから、本手法は従来の提案手法より細分化モデルの可読性を向上させることができたと言える。

6.3 今後の課題

現時点では、可読性向上アルゴリズムの計算量爆発に関する検証が不十分である。また、今回の実験に使用した推論規則のみで、考慮すべき規則を全て網羅できているかの検証が、現時点では不十分である。

7 終わりに

本研究では細分化モデルの可読性向上手法を提案した。従来手法よりも可読性の向上性能は高くなった一方で、計算量爆発や推論規則の充足性に関する検証を十分に行い、さらに手法を改善することが今後の課題である。

参考文献

[1] 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, 2014
[2] 来間 啓伸. B メソッドにおける形式仕様記述. 近代科学社, 2007
[3] 高橋宏夢. 形式手法 B Method における高信頼細粒度部品の粒度の提案. 情報処理学会第 80 回全国大会講演論文集, vol.1 pp.217-218, (2018.03)
[4] 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, (2013.09)