B-008

形式的ソフトウェア合成手法における計算コストを考慮した細分化モデル可読性向上手法

The Method for Improving the Readability of Sliced Models in Formal Software Synthesis Method with Consideration of Computational Cost

結城 翔[†] **Sho Yuki**　　**織田 健**[†] **Takeshi Oda**

1 はじめに

ソフトウェアの大規模化や複雑化に伴う開発コストの増大に対処するため、形式手法が注目されている。我々は形式手法の一つである B Method を用いた、既存ソフトウェアの部品再利用による形式的ソフトウェア合成手法（MSSS 手法）を提案している [1]。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、不足部品の式の可読性向上手法が提案されたが [2]、この提案は計算コストの無考慮などの課題があった。よって本稿では計算コストを考慮した細分化モデル可読性向上手法を提案する。

2 背景と目的

2.1 B Method と MSSS 手法

B Method[3] は集合論と一階述語論理に基づく形式手法の一種で、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を保証する。

MSSS 手法 [1] は B Method を用いた、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、適切な部品を選択・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。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。

2.2 不足部品の可読性向上手法

不足部品は字面統一が施され難解な表現で書かれている為、不足部品の可読性向上手法を提案してきた [2]。この手法では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などを行った。図 1 はこの手法の大まかな流れである。

1. 字面統一後の変数に注目した制約条件抽出 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す。
2. 等式による式の書き換え 要求モデルと細分化モデルに含まれる等式を抽出し、その等式を構成する定数・変数・集合（以降これらを項と呼ぶ）を含む細分化モデルの式に対して、等価な項により式を書き換え細分化モデルに追記する。
3. 推論規則の適用による書き換え 字面統一の際に用いられる式の推論規則をもとに、可読性向上のための推論規則を整備する。その後細分化モデルの式群に対して整備した推論規則を適用する。

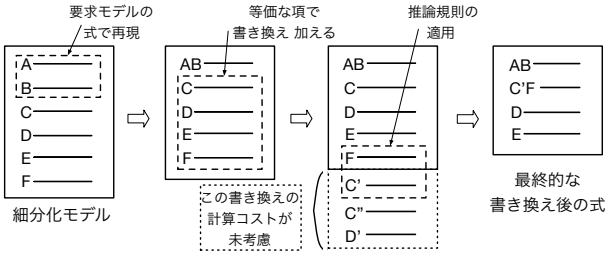

図 1: 従来手法の流れ

2.3 先行研究における課題

2.2 節の 2. の操作では、等式で結ばれた項によって既存の式を書き換えた式を追記しているが、要求モデルに記述された等式やその中で用いられている項の種類が増えることにより、この手順を行う計算コストが爆発的に増加しないかどうかの検証が不十分であった。加えて、書き換えに用いる等式の両辺がそれぞれ多項式になっている場合、等価であることを調べる際のパターンマッチに膨大な時間がかかってしまうことも考えられる。

2.4 研究目的

2.2 節の先行研究は不完全で、計算コストの無考慮などの課題があった。そこで本研究では、細分化モデル中の等式の数、項の種類と手法中で追記される式の数との間に成り立つ関係を導出する。これにより従来手法の計算コストを考慮し、計算コストが現実的ではなくなる場合に対して新たな工程を提案する。

3 計算コストを考慮した可読性向上手法

3.1 提案手法の概要

書き換え回数を抑えた上で効果的に可読性を向上させるために、まず書き換え手順の中間結果となる式の最大数を予想する。その最大数がプログラムの現実的な計算コストの範疇を超える場合に各項の重みづけをすることで、式の書き換え回数に差をつけるための指標を設ける工程を提案する。各項の重みづけの計算は以下の仮定に基づいて行われる。

- 等式の右辺より左辺の方が項の重要度が高い
- 左辺の項同士の重要度の差は、それぞれが右辺に持つ項の重要度の差に準じる。
- 項のうちリテラルの重要度は低い

[†]電気通信大学大学院情報理工学研究科情報学専攻

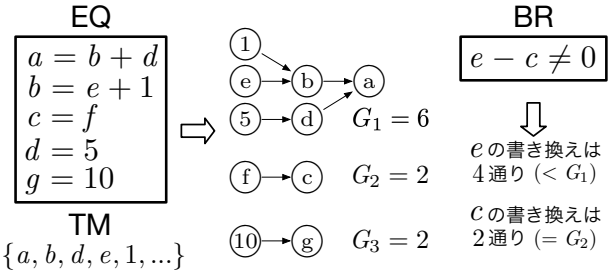

図 2: 有向グラフの例

1 つ目の仮定は、等式を記述するときにまず左辺に主要な表現を置き、右辺でそれを説明するパラメータを置くことが多いからである。2 つ目の仮定は 1 つ目の仮定に基づく。3 つ目の仮定は、定数による項の書き換えは優先度が低いからである。

3.2 式の最大数計算

表 1 は、以降に出てくる用語とその説明である。計算に用いる有向グラフは図 2 のように TM の要素をノードとして持ち、EQ の等式で結ばれた項のノード同士が、右辺の項から左辺の項に向かうエッジで結ばれている。独立した各グラフの i 番目のグラフの位数を G_i とおく。次に TM の要素が n 個用いられている BR の式が EQ を参照して書き換えられる式の数を考える。式中の TM の各要素はその項をノードとして持つ有向グラフの位数 G 通りの書き換えが可能である。今回は式の最大数を考慮する為、グラフの位数はすべてのグラフの中で最大の位数 $\max G_i$ とする。この場合、$\max {G_i}^n$ 通りの書き換えが考えられる。BR の j 番目の式に用いられている TM の要素数を BR_{j_n} とおけば、全ての式の書き換え結果の最大数 F_{max} は次のように表される。

$$F_{max} = \sum_{j=1}^{|\mathrm{BR}|} \max {G_i}^{\mathrm{BR}_{j\_n}}$$

3.3 有向グラフを利用した項の価数計算方法

3.2 節の有向グラフを用いて、3.1 節の仮定に基づき次の手順で価数計算を行う。

1. グラフの始点のノードの価数を 1 (リテラルの場合 0) とする。
2. エッジの価数を、そのエッジの始点ノードの価数 +1 とする。
3. 始点以外のノードの価数は、そのノードを終点とするエッジの価数の合計とする。
4. 手順 2, 3 を繰り返し行う。

以上の計算による各ノードの価数の大きさを、そのノードに対応する項の価数とする。図 3 は価数の計算例である。F_{max} が閾値を超える場合はこの価数をもとに、各項を用いている BR の要素に対し EQ を利用した書き換えを何回施した式までを用いるか決定する。

表 1: 用語の説明

用語	説明
EQ	式の書き換えに用いる等式の集合
TM	EQ 内で用いられている項の集合
BR	EQ 内の等式による書き換え対象の式の集合

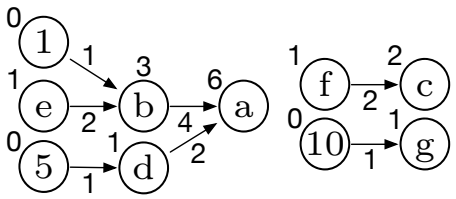

図 3: 価数の計算例

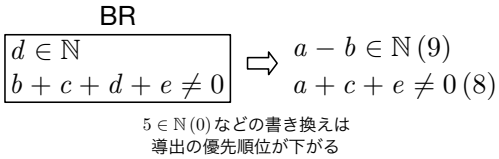

図 4: 価数を参照した式の書き換え例

3.4 項の価数を利用した式の書き換え

例えば簡単のために、等式の書き換えの上限が 2 回であるとする。ここでは「BR の各項を書き換える際、“ 書き換え後の項の価数の合計 − 書き換え前の項の価数の合計 ” がより大きくなる書き換えから優先的に行う」という規則を仮定し書き換えを行うと、結果は図 4 のようになる。ただし、書き換え後の式の右に書かれた数字は、書き換え後の項の価数の合計である。例えば BR の $b + c + d + e$ を書き換える際には、価数 1 の項 d を価数 6 の項 a と価数 3 の項 b で書き換えると最も書き換え前後の価数差が大きくなるため、図 2 の EQ の式 $a = b + d$ を変形して $d = a - b$ とし、書き換えることで $a + c + e$ を得る。この式の価数の合計は $6 + 2 + 1 = 8$ となる。このように、書き換え回数の上限が定められていても重要度が高い式を優先的に導出することができる。

4 考察

本提案ではプログラムの現実的な計算コストの範疇となる F_{max} の具体的な閾値を設けていないため、今後はこの検証が不可欠である。また各項の価数と EQ を利用した書き換え回数の間に成り立つ関係式を定義することも必要である。現時点ではドント式の活用を考えている。

5 終わりに

本稿では細分化モデル中の等式の数と項の種類、およびそれらを参照することにより手法中で追記される式の数との間に成り立つ関係を導出した。また、計算コストが現実的ではなくなる場合に対して、項の価数計算による重み付けを提案することで、計算コストを考慮した細分化モデルの可読性向上手法を提案した。今後はパターンマッチの計算コストへの対応を行う。

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] 結城 翔, 織田 健. 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. 情報処理学会第 85 回全国大会講演論文集, vol.1 pp.275-276, (2023.03)

[3] 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