B-003

リファインメントを考慮した形式的ソフトウェア合成アルゴリズム

Formal Software Synthesis Algorithm Considering Refinement

叶野 英俊† **織田 健**†
Hidetoshi Kano **Takeshi Oda**

1 はじめに

形式手法の 1 つに B Method があり、これはモデル、リファインメント、実装の 3 つからなる。我々はこれらをそれぞれ分解した組を部品とし、それを再利用することで高信頼ソフトウェアを低コストで合成する手法を提案しているが、リファインメントを考慮した手法は提案されていない。そこで本研究では、リファインメント段数や変数の詳細化順序の不一致などの問題を解決し、リファインメントを考慮した合成アルゴリズムを提案する。

2 研究背景

2.1 B Method

B Method は数学的基盤に基づき開発を行う形式手法の 1 つであり、仕様記述からコード生成までを支援する。これは仕様記述であるモデル、中間的記述のリファインメント、最も詳細な記述の実装からなる [1]。リファインメントは 0 段以上の任意の段数記述され、これによりモデルは段階的に詳細化される。リファインメントと実装には詳細化前後の変数の関係を示すリンク不変条件が記述され、モデルと実装の整合性が証明される。

2.2 先行研究

ソフトウェアの開発コスト増大や信頼性低下の解決策の 1 つとして、既存ソフトウェアの再利用が挙げられる。中村は既存ソフトウェアを分解しリポジトリに登録する部品生成手法と、部品再利用による新規ソフトウェア自動合成手法を提案し [2]、熊谷らは新規ソフトウェア合成の際、部品の結合可否を判定する手法を提案した [3]。

しかしこれらはモデルと実装のみを持つソフトウェアを対象とし、リファインメントは考慮されていない。

3 リファインメント考慮における課題

3.1 リファインメントを考慮した部品生成

中村の部品生成手法では、リンク不変条件からモデルと実装の変数を対応付けることで、モデルを分解した細分化モデルに対応する部品を生成する。しかしリファインメントを含むソフトウェアはモデルから実装へ直接変数の対応がとれない。よって、リファインメントを介して、対応する部品を生成できる手法が必要となる。

3.2 リファインメント段数の差異の吸収

生成される部品は元のソフトウェアと同じ段数のリファインメントを持つため、段数はそれぞれ異なる。また部品結合の際、リファインメントは各段で結合を行うため、図 1 のように段数の異なる 2 部品では結合できない段がある。段数の異なる部品が結合できないことは再利用性の観点で適切ではないため、リファインメントの段数の差異を吸収し、結合可能とする手法が必要となる。

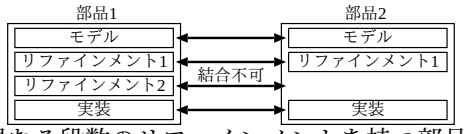

図 1: 異なる段数のリファインメントを持つ部品の結合

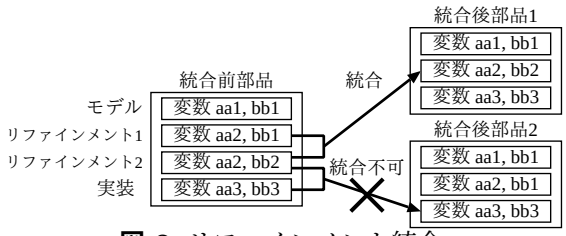

図 2: リファインメント統合

3.3 部品の結合可否の判定

ソフトウェア合成では要求を分解した細分化モデルのふるまいと一致する部品を検索するが、一致する部品は複数存在する。しかし、変数の型が不一致な部品同士は結合できないため、複数部品の中から結合可能部品群を取得する必要がある。従来は 2 部品間の変数の型の一致を総当たりで調べ、部品が結合可能かを判定していた。

しかしリファインメントを考慮すると 2 部品の型一致では不十分で、必要な全ての部品が同時に結合可能であることを示さねばならない。これを総当たりで行うのは計算コストの観点で現実的ではないため、必要な部品群を取得する現実的な手法が必要となる。

4 リファインメントを考慮したソフトウェア合成アルゴリズム

本章は 4.1 節でリファインメントの統合と追加という手法を提案し、4.2 節以降はそれを利用しリファインメントを考慮したソフトウェア合成アルゴリズムを提案する。

4.1 リファインメントの統合と追加

4.1.1 リファインメントの統合

リファインメントの統合では、隣接した 2 つのリファインメントや実装をまとめることで部品の段数を減らす。隣接するリファインメントのリンク不変条件をまとめて記述し、変数や操作はより詳細な記述を利用することでリファインメントを統合できる。ただし、統合は変数の詳細化順序を保持できるリファインメント間でのみ行える。図 2 において変数 aa1, aa2, aa3 と bb1, bb2, bb3 は変数が詳細化される過程である。リファインメント 1 と 2 を統合した統合後部品 1 の変数の詳細化順序は統合前部品と等しいが、リファインメント 2 と実装の統合では、変数 b1 の詳細化が bb1, bb3 となり、統合前部品と異なるため統合は行えない。また統合は不可逆であり、

†電気通信大学大学院情報理工学研究科情報学専攻

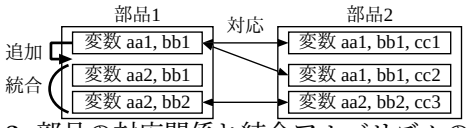

図 3: 部品の対応関係と結合アルゴリズムの例

統合後部品から統合前部品を生成することはできない。

4.1.2 リファインメントの追加

リファインメントの追加では、部品のリファインメントや実装を複製することで段数を増やす。複製であるため、部品が元々持つ変数の型の段のみを追加できる。図 2 の統合後部品 1 では、aa1 と bb1、aa2 と bb2、aa3 と bb3 の組のみを追加可能であり、aa1 と bb2 のような組は追加できない。また実装を複製しリファインメントとする場合、実装特有の記述である VALUE 節を削除し、IMPORT 節を INCLUDE 節に変更する必要がある。

4.2 部品生成

リファインメントを考慮した部品生成手法は、基本的に先行研究 [4] に準ずる。ただしモデルを元に実装を直接抽出するのではなく隣接する段それぞれで変数の対応をとり、段階的に抽出することで細分化モデルに対応する部品を抽出する。このときリファインメントは変数の一部のみを詳細化でき、詳細化されない変数のみを持つ細分化モデルにおいて、そのリファインメントから生成できる細分化リファインメントは空となる。その場合、空のリファインメントを削除し無意味な記述を減らす。

4.3 部品の結合可否判定と結合アルゴリズム

4.3.1 部品の結合可否判定

部品結合の可否を判定するため、2 部品に共通する変数を調べる。共通変数がない場合、2 部品は関係性がないため結合可能である。共通変数を持つ場合、共通変数の詳細化順序を比較する。詳細化順序が異なる時その 2 部品は結合不可能である。一致する時、2 部品の各段について相手の部品の変数の型が一致する段と対応をとる。全ての段で相手の段と対応が取れる時、部品は結合可能である。対応がとれない段がある時、その段が統合可能ならば結合でき、統合不可の時は結合できない。

4.3.2 結合アルゴリズム

結合可能な 2 部品の結合アルゴリズムを提案する。2 部品が共通変数を持たない場合、段数が等しい時は各段の結合だけでよい。段数が異なる場合、段数が一致するまでリファインメントを追加し、その後結合を行う。

共通変数を持つ場合、4.3.1 項と同様に段の対応をとる。対応が 1 対多の場合、1 のほうにリファインメントを追加して段数を合わせる。また対応関係のない段は 1 つ下の段と統合を行う。図 3 は段の対応と結合の例である。以上により部品の段数と変数の詳細化順序が一致するため、対応する段の結合により部品の結合が可能となる。

4.4 ソフトウェア合成

3.3 節より、総当たりでの部品選択は現実的ではない。よって総当たりによる判定ではなく、基準となる細分化モデルの部品に付け加える形で部品群を選択し、結合を行う。この手法は以下の 2 つの手順に分けられる。

4.4.1 変数のグループ化

部品の組み合わせは細分化モデルの数が多くなるほど爆発的に増加する。しかし関連を持たない細分化モデルもあるため、関連を持つ細分化モデルでグループを作ることで組み合わせを減らすことができる。グループは変数を基準とし、共通変数を持つ全ての細分化モデルを 1 つのグループとする。これにより、関連する細分化モデルのグループで独立して部品の結合可否を判定できる。

4.4.2 細分化モデルの複雑さに基づく部品選択と結合

4.4.1 項の各グループに対し結合可能部品を選択する。部品選択で取得できなかった部品は人手により記述する必要があるため、部品選択では人が記述しにくい複雑な部品を選択できることが望ましい。人が記述しにくい部品とは、制約が多く証明が困難な部品であると考えられるため、実装部品の制約条件の数を複雑さとする。細分化モデルに対し部品は複数存在するため、細分化実装の制約条件数の平均を細分化モデルの複雑さと定義する。

以下の手順で結合可能な部品を選択する。まず最も複雑な細分化モデルの部品の 1 つを基準として、次に複雑な細分化モデルの部品で結合可能なものを 4.3.1 項の判定によりすべて選択しそれぞれを結合する。次に結合した各部品を基準として、3 番目に複雑な細分化モデルの部品の中で結合可能なものをすべて選択し結合する。以上のように基準部品に付け加える形で再帰的な結合を行い、最終的な結合部品群の中で複雑さの合計値が最も高いものを選択する。最後に各グループで選択された結合部品を結合することで、要求ソフトウェアが合成される。

5 実験と考察

リファインメントを含む小規模なソフトウェアについて、リファインメントを考慮した実装抽出手法を適用したところ、細分化モデルとふるまいが等価なリファインメントを含む部品を抽出することができた。部品の結合アルゴリズムやソフトウェア合成は実験ができていないため、今後行っていく。また今回の実装抽出実験は小規模で制約条件の少ないソフトウェアを使用したため、制約がより複雑なソフトウェアにおける実験も必要である。

6 おわりに

本稿では、リファインメントを含むソフトウェアについて、部品生成や選択、結合アルゴリズムなどを提案した。今後は実験していない部分の検証やより大規模なソフトウェアにおいての手法の正当性を検証していきたい。

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,2014.

[3] 熊谷 恒, 織田 健. 形式的ソフトウェア合成手法における部品の充足を考慮した合成手順, 第 77 回情報処理学会全国大会講演論文集, vol.1 pp.351-352, 2015.

[4] 叶野 英俊, 織田 健. 形式的ソフトウェア部品生成のための実装抽出手法, 第 80 回情報処理学会全国大会講演論文集, vol.1 pp.219-220, 2018.