1K-01

小規模なモデルを対象とした形式的ソフトウェア合成システムの構築

田中 涼介[†]
電気通信大学情報理工学域

檜垣 廉[‡]
電気通信大学大学院情報学専攻

織田 健[†††]
電気通信大学大学院情報学専攻

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や信頼性の低下に対し、我々は形式手法を用いた既存ソフトウェアから部品を生成、再利用することでソフトウェアの合成を行う MSSS 手法 [1] を提案している。本研究では、先行研究のソフトウェア合成手法の不足を補って具体化し、手法の実装の第一歩として単一の抽象機械でモデル化されるソフトウェアのみを対象に構築したモデル充足ソフトウェア合成システムについて述べる。

2 研究背景

2.1 B Method

B Method[2] は形式手法の 1 つで、集合論と一階述語論理に基づいた仕様記述言語を用いて抽象的な仕様を表すモデルと、これを段階的に詳細化したリファインメントや実装を記述することでソフトウェアを開発する手法である。モデルやリファインメント、実装は状態を変化させる操作や常に満たす条件などからなり、それぞれの無矛盾性と詳細化の整合性を定理証明により証明することで信頼性の高いソフトウェアを開発できる。また、他のモデルを利用することでモジュール構造を構成し、複雑なソフトウェアを開発できる。

2.2 MSSS 手法

MSSS 手法は、B Method によって信頼性を保証されたソフトウェア部品を生成し、再利用することで仕様を満たすソフトウェアを合成する手法である。以下では部品の生成については割愛し、ソフトウェア合成 (図 1) に必要な工程について述べる。

2.2.1 モデル細分化

モデル細分化では、要求モデルの操作を原則 1 代入文単位になるよう条件分岐や同時代入部分を分割し、各代入文に関係する条件・宣言等を合わせて細分化モデルとする。項書換えを応用した式の等価変換や、可換要素の数学的構造によるソート、識別子の付け替えによって、数学的な意味が同じ記述が文字列上でもできるだけ一致するようにする字面統一を行う。この字面統一を含む全く同じモデル細分化を部品生成時にも行うことで、文字列一致による部品検索が可能となり、計算コストを削減できる。各工程の大まかなアルゴリズムは三鍋 [3] によって提案され、その後高橋 [4] によって部品の結合に配慮した粒度に変更された。

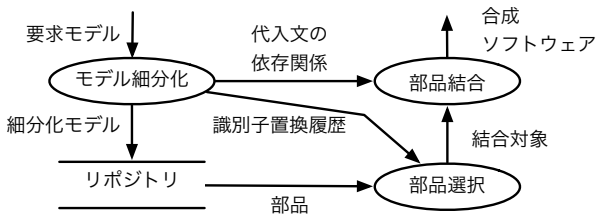

図 1: MSSS のデータフロー

2.2.2 部品リポジトリと検索手法

合成に必要な部品を検索する際には、要求を細分化した部品とリポジトリ内の部品の仕様について、パラメタ制約とプロパティ制約、不変条件、代入が一致し、操作の事前条件が含意の関係を満たすときに取得部品とする。これを含意検索という。研究室内では関係データモデルを用いて、要求との一致を見る要素と事前条件をそれぞれ部品と関連付けるデータベースと、内部結合を用いて含意検索を行うアルゴリズムが提案されていた。

2.2.3 部品選択・結合手法

部品を選択する際には、2 部品で共に用いられる変数の型と詳細化の方法の一致を判定することで、矛盾のない部品を選択しやすくする [4]。リファインメントの数を揃えて結合する際、この判定は結合と並行して行う必要があり、計算コストを下げるために細分化モデルに優先順位をつけて選択を試行する手法が提案された [5]。また、要求モデルにおける変数の参照から代入文の依存関係を推定し、実装の代入文の順序を定め、結合する。

3 先行研究の課題と研究目的

ソフトウェア合成に必要な上述の工程は全てのアルゴリズムが厳密に定まっているとはいえず、プログラムとして実装するには不十分である。本研究はそのような曖昧な手順を補い、各工程を連携させることでソフトウェア合成システムを構築することを目的とする。

4 モデル充足ソフトウェア合成システム

本章では、従来手法の不足を補って構築したソフトウェア合成システムについて述べる。このシステムはモジュール構造を持たない要求モデルを対象とする。

4.1 初期化の扱い

初期化と操作は共に代入文で構成されるが、部品における初期化の扱いは定まっていなかった。ここでは初期化を操作の 1 つとみなし、操作と同様に細分化や結合を適用することとした。

4.2 システムのモジュール構成

図 2 にシステムのモジュール構成を示す。取得した部品をオブジェクトとして扱い、これに対する操作を行うという方針で実装を行った。`msss` は種々のモジュー

Construction of the Model Satisfiable Software Synthesis for a Small Scale Model

[†]Ryosuke Tanaka, The University of Electro-Communications, School of Informatics and Engineering

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

[†††]Takeshi Oda, The University of Electro-Communications, Graduate School of Informatics and Engineering

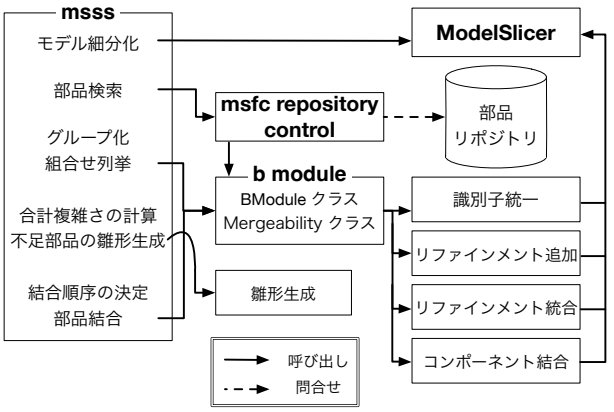

図 2: MSSS のモジュール構造

ルを呼び出すことでソフトウェア合成を実行する。初めに ModelSlicer を用いて要求モデルを細分化し、細分化モデルから `msfc repository control` が部品を取得、オブジェクト化する。部品を表すクラスは `b module` に BModule クラスとして定義されている。このクラスは部品で使われている変数や詳細化の方法などの情報を持っており、これらの情報を元に部品の組合せの探索や結合可否判定を行う。部品が不足した場合は取得部品から情報を集め、人が記述するための部品の雛形を生成する。最後に BModule クラスのメソッドを用いて部品を結合し、合成ソフトウェアとする。

4.3 ModelSlicer

ModelSlicer は Standard ML で記述したモデル細分化を行うプログラムである。従来手法のモデル細分化から、型情報の利用や可換演算への規則の書き換えの効率化等のアルゴリズムの改善を行い、他の工程との整合を図り分割の基準を代入文により変更される変数単位に変更した。また、図 1 のように、細分化の途中で得られる代入文の依存関係や識別子置換履歴を後工程に渡すためのフォーマットを決定した。

4.4 部品リポジトリ

部品リポジトリは、検索のキーとなる情報については先行研究と同様に保持するが、部品の内容を拡張し、それぞれの部品は 0 以上のリファインメントと 1 つの実装、それぞれに対する変数の詳細化等に関する付加情報、さらに利用するモジュールの情報を持たせた。

4.5 部品選択に関するプログラム

先行研究において部品選択は、既に選択した部品と結合可能なものを選択することを繰り返すことで行うため部品結合と同時に進めていたが、結合可否判定は部品の付加情報のみを用いて行えるため、選択のための結合可否判定と実際の部品結合は分離した。また、変数の詳細化方法が同一であるかの判定方法を厳密に定めた。

5 実験

実装の妥当性を検証するための実験を行った。まず、リファインメントを含んだ B Method によるソフトウェアを用意し、そのモデルを細分化した細分化モデルを元に、リファインメントを 0 段または 1 段持つ部品を作成した。これらの部品をリポジトリに登録し、細分化前のモデルを要求モデルとして構築したシステムを実行した。

その結果、ライブラリを用いる部品の結合において手法の問題点を発見した。この点を修正して再度実行すると、用意したものと同等のソフトウェアが合成できた。

6 考察

6.1 実装の妥当性

実験の結果、証明できない証明責務が生じることはなく、本実験の範囲ではソフトウェアの合成に成功した。要求モデルが部品の元となったソフトウェアのものと同一であるため部品の汎用性について検証できたとはいえないが、モデル細分化や検索、及び結合については検証することができた。また、部品選択についてはリポジトリの部品の数を増やしてさらに検証を行う必要がある。

6.2 リファインメントの追加

リファインメントの数を揃えるために行うリファインメントの追加は従来、モデルやリファインメント、実装のどれに対しても適用できるとされていたが、ライブラリを使っている実装には適用できないことが実験中に判明したため、暫定的に実装には適用しないよう変更した。

6.3 型制約に関する付加情報

従来手法における型制約は実装等から抽出したものを用いていたが、異なる部品において元は同じ変数でも部品に含まれる他の変数によって型制約が変わることがあるため、実装等に型推論を行い、その結果を型制約とするのが望ましいと考える。

6.4 今後の課題

ライブラリを用いる部品について適用範囲が限られているため、結合手法を新たに考案する必要がある。また、モジュール構造を持つソフトウェアの合成に対応することは課題である。今後はこれらの課題に対応し、計算コストや部品の充足性についても実用性の向上を目指す。

7 おわりに

本研究ではソフトウェア部品に対する処理を中心に構築したモデル充足ソフトウェア合成システムについて述べた。合成をプログラムとして実行できるようになったが、より複雑なソフトウェアによる検証やモジュール構造への対応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, 2013
[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] 三鍋孝介, 織田健. 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)
[4] 高橋宏夢, 織田健. 形式手法 B Method の細粒度部品の結合による高信頼ソフトウェアの合成, 第 17 回情報科学技術フォーラム論文集, vol.1 pp.137-138, (2018.09)
[5] 叶野英俊, 織田健. 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, 第 18 回情報科学技術フォーラム論文集, vol.1 pp.129-130, (2019.09)