B-010

形式仕様記述の細分化における未記述制約抽出手法の提案
Extraction of Implied Constraints on Formal Specification Slicing

森下 匡平† 織田 健†
Kyohei Morishita Takeshi Oda

1 序論

ソフトウェア開発の複雑化大・規模化に伴い、開発コストの低減や信頼性の確保などが課題となっている。我々は、人的コストの低減および信頼性向上を可能とするソフトウェア部品再利用と高信頼なソフトウェア開発を目的とした形式手法の一つである B-method により、仕様記述を細分化し同等の操作を行う既存のソフトウェア部品を検索・自動合成することで新しいソフトウェアを生成する手法を提案している。細分化前の仕様記述が持つ制約を維持し、細分化後の仕様記述の無矛盾性を保持する事が必要不可欠であるが、未記述制約の抽出が不十分である。本研究では、仕様内に暗黙的に存在する制約を推論により抽出する事で仕様の細分化における無矛盾性の保持を図る。

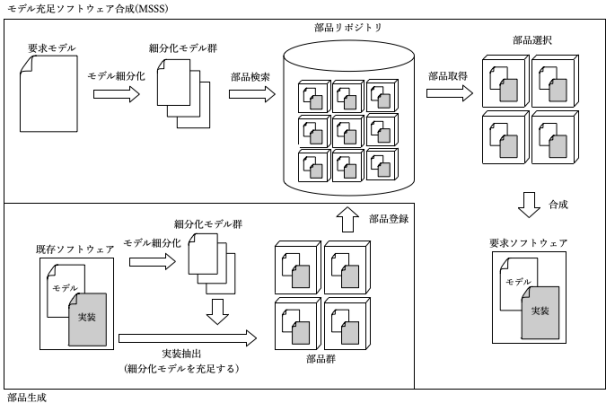

図 1: MSSS 手法概説図

2 研究背景

2.1 B-method

B-method は高信頼なソフトウェア開発を目的とした形式手法の一つであり、集合論に基づいた仕様記述であるモデルと、段階的詳細化に従いモデルを徐々に具体化した実装からなる。[1] モデルと実装は数学的に記述されているため、モデルの無矛盾性およびモデルと実装の間の整合性を数学的証明により確保することでができる。

モデルは大きく分けて “制約 (制約条件)” と “操作” の２つの記述からできている。操作はモデルの動作を表しており、制約はその操作に関連する条件を記述している。

証明はモデルが制約を常に満足するかを示すことで遂行される。

2.2 モデル充足ソフトウェア合成手法 (MSSS 手法)

B-method にソフトウェア部品再利用の概念を取り入れ、少ない人的コストで高信頼なソフトウェアを合成することを目的とした手法である [2]。B-method によって開発された既存のソフトウェアを細分化することで得られる細分化モデルと実装のセットを部品とする。要求仕様を細分化し得られた細分化モデルと等価な機能を持つ部品を検索・合成することで新規ソフトウェアを生成する。

モデルの細分化は操作単位での分割と必要制約の抽出からなる。細分化によってモデルの持つ性質を損失しないよう、細分化後のモデルは細分化前のモデルの制約を維持し、また無矛盾性を保証する必要がある。

2.3 先行研究について触れる

細分化では、細分化前のモデルの制約を維持するとともに細分化後のモデルの無矛盾性を保証する必要がある。モデル内には未記述の制約が暗黙的に存在し、これを推論によって全て書き出した上で細分化を行うことで細分化前のモデルが意図する動作を細分化後のモデルも行うようにする。

しかし、暗黙的に存在する未記述の制約は無数にあるため膨大な数の推論ルールを用意する必要があり現実的でなく、また、推論結果を元の制約に加えるために推論可能な組み合わせが際限なく増えてしまう。

これに対し三鍋は、制約を表す条件式に用いられる演算子を制限することで必要な推論ルールを低減する “プリミティブ化” を行なったのちに推論する手法を提案した [3]。プリミティブ化では全ての条件式を、高機能な演算子を制限した低機能な演算子のセット “プリミティブな演算子” のみの条件式に書き換える。また、プリミティブ化した条件式には二重否定などの冗長な部分が存在してしまうため、これを解消する (冗長性を排除する) “簡約化” をプリミティブ化と推論の過程に設けた。

2.4 先行研究の課題と本研究の目的

先行研究ではプリミティブな演算子への書き換えルールや推論ルールが十分に整備されていないことから、暗黙的に存在する未記述制約が十分に抽出できなかった。提案されていたプリミティブ化は、高機能な演算子を許した式表現を低機能な演算子で表現し直すため、条件式が複雑化してしまう。例として、プリミティブ化により多くの関数は部分関数を混えた式に一度書き換えられるが、部分関数もプリミティブな演算子で書き換えられるため、元の式は非常に複雑な式になってしまう。推論材料である条件式の複雑化に伴い、推論コストが増加し、暗黙的に存在する未記述制約が正しく導出できない問題や停止性の問題が生じている。推論に関わる定義不足および推論ルールの未整備により、推論できる条件式が少なく必要な制約を十分に明示できない。

本研究ではこれらの問題を解決することを目的とする

†電気通信大学大学院情報理工学研究科情報学専攻

3　未記述制約抽出手法

3.1　課題解決の指針

プリミティブ化による条件式の複雑化に伴う問題を緩和するため、プリミティブ化前にも推論を行う。プリミティブ化前の推論はプリミティブ化による複雑化がされていない条件式を対象とするため、推論ルールは簡潔なものとなる。推論ルールの複雑さを排除し、取り扱えるルールを拡張することで、制約抽出の取りこぼしを防ぐ。また、プリミティブ化前に暗黙的に存在する未記述制約を可能な限り抽出しておくことで、プリミティブ化後に必要な複雑な条件式による推論の低減を図る。

推論に関わる定義および推論ルールについては、必要な制約を抽出できるよう整備する。

3.2　推論における定義とルールの整備

プリミティブ化前の推論では、高機能な演算子による条件式の表現が許されており、条件式の組み合わせによる推論結果の爆発を防ぐため、推論が収束するように推論ルールを設ける。複数の要素を持つ条件式から、その要素数より少ない要素らの関係が暗黙的に存在し導出できるとき、これを条件式に書き加える。　プリミティブ化後の推論は、後述のプリミティブ化の定義に従いプリミティブな演算子で表現された全ての条件式に対して推論を行い、導出される推論結果を条件式を書き加える。

推論ルールは定理証明器 Atelier B を参照する。推論は推論した条件式を含め、全ての式の組み合わせに対して行う。ただし、一度推論した条件式および既に制約に存在している条件式は書き加えない。

3.3　簡約化

先行研究における簡約化はプリミティブ化された条件式を対象としていたが、本手法ではプリミティブ化されていない条件式にも簡約化を行うため、簡約化ルールの拡張を行う。

3.4　プリミティブ化

プリミティブ化による条件式の複雑化を緩和するため、プリミティブな演算子を再定義する。

4　実験

従来手法では抽出することができなかった暗黙的に存在する未記述の制約を、提案手法が抽出可能であり、かつ従来手法で抽出できたものについても不備なく抽出可能であることを実験を行った。

5　考察

実験により提案手法が従来手法では抽出できなかった暗黙的に存在する未記述の制約を抽出可能であることが確認できた。プリミティブ化前の推論では推論を収束させるため、推論によって新たに導出される条件式中の要素数が減少するようへ推論ルールを設けた。これにより推論による条件式数の爆発による停止性の問題は改善された。　今後の課題として、プリミティブ化後の推論の停止性についての検証が求められる。

参考文献

[1] 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.
[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
[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, (2013.09)