仕様記述からプログラム・コードの自動生成，実機による実行まで

FM3マイコン基板で学ぶ Bメソッドによる形式手法

第3回 仕様が正しいことを証明する「対話証明」

船津 侑志，堀 武司，佐藤 晴彦

仕様を文章や表で書かず，コンピュータで解釈可能な形式で書き，それを証明するのが形式手法の考え方です．連載第3回目の今回は，記述した仕様が正しいことを証明する「対話証明」について解説していきます．

対話証明は，本連載で使用しているツール「Atelier B」では自動的に行うことができます．一部，自動的に行えない部分は手動で証明していきます．ツールの力を借りて，正しさを証明していきましょう．（編集部）

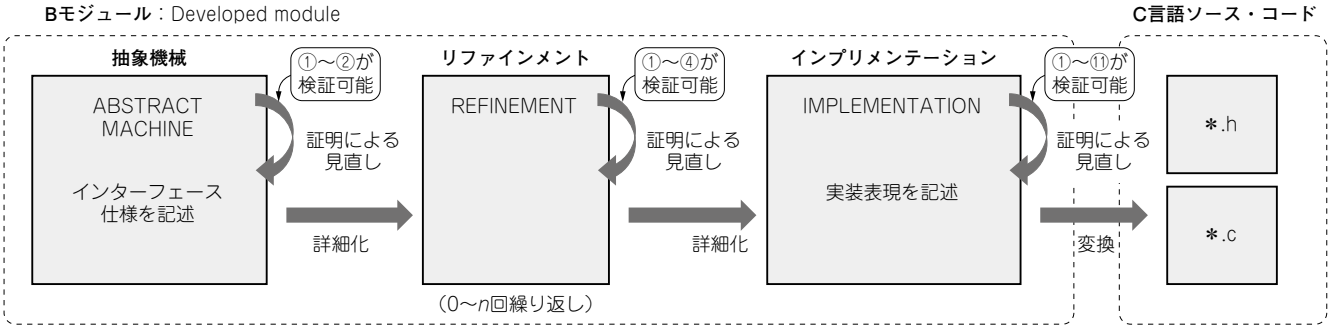
連載第2回目（2012年9月号）では，自動販売機を制御するプログラムを題材に，「抽象機械」の書き方についてご紹介しました．今回は記述した仕様が正しいことを証明する「定理証明」の実施方法について説明します．例として，前回作成した抽象機械を対象に，仕様の整合性の検証方法を解説します．

なお，説明の中で，証明に関連する用語が出てきます．これらの用語に慣れていない方は，コラム1に用語の説明を掲載しましたので，そちらをご覧いただくと，本文が理解しやすくなると思います．

1 復習：Bメソッドにおける仕様とは

Bメソッドによるソフトウェア開発では，抽象機械からリファインメントを経てインプリメンテーションに至る各段階において，こうしたさまざまな検証を証明として厳密行うことによって，ソフトウェア全体の整合性・一貫性を保つことが可能です．「Bメソッドのコンポーネントの種類と開発手順」と，各コンポーネントにおいて検証可能な項目を連載の第1回および第2回に掲載した図を元に書き加えたものです．**図1**に示します．

図1の検証項目のうち，抽象機械では項目①および②を



■検証可能な項目の例

① 初期化実行後に，不変条件が満たされていること
② 不変条件を満たした状態で操作を実行した場合，実行後も不変条件は満たされること
③抽象機械（または一段階前のリファインメント）の操作で記述した，分岐条件と一般化代入の組み合わせに反する一般化代入が存在しないこと
④初期化の実行および操作の実行により，抽象機械（または一段階前のリファインメント）で定義した変数とのリンク不変条件が満たされなくなるケースが発生しないこと
⑤制約条件を満たす，定数・変数の値が存在すること
⑥他モジュールの操作呼び出しの際に，事前条件が満たされないような呼び出しが発生しないこと
⑦モジュール内のローカルな操作の実装において，その仕様として記述した分岐条件と一般化代入の組み合わせに反する一般化代入が存在しないこと
⑧操作の出力パラメータの設定漏れが存在しないこと
⑨ローカル変数のオーバーフロー・アンダーフローが発生しないこと
⑩配列の境界外へのアクセスが発生しないこと
⑪ゼロ除算が発生しないこと

抽象機械で検証できる（今回はここ）
リファインメントで検証できる
インプリメンテーションで検証できる

図1　抽象機械，リファインメント，インプリメンテーションではいろいろな項目を検証できる

KEYWORD——形式手法，Bメソッド，Atelier B，定理証明