新 組み込みソフトへの数理的アプローチ
～形式仕様記述をどのように使うか～

第11回

プログラム検証とテスト
――CBMCを使ってソース・コードから仕様を検証する

藤倉 俊幸

1 ソース・コードから意図を読み取る

連載の前回（2010年2月号，pp.150-156）では，要求を構造的に表現して最終的にソース・コードまで降りられることを示した．今回からしばらくは，逆にソース・コードから上へ登って仕様や要求にたどり着く方法を検討する．

ここでやりたいのは，プログラムの目的に迫ることである．ソース・コードには誤りがあるかもしれないということを考慮した上で，ソース・コードをどのように解析すれば，それが何をやろうとしていたのか見えてくるのか．ソース・コードだけでは本当に無理な話である．しかし，ゼロの割り算とか配列オーバ・フローなど，そのソース・コードが何をやろうとしているのかと関係なく検証するツールは存在する．このようなツールの出力を利用して，計算式のリバースをしてみる．リバースによってプログラムの事後条件が見えてくる．

次に，テスト・ケースを作る手法を利用して事前条件を検討する．道具立てとして，プログラム検証ツールとSMTソルバ，ソース・コードそのものと，仕様というか動作条件の指定または検査の機構として`assert`文を使うことにする．関数の入り口と出口で使用すれば事前条件・事後条件の指定に利用できる．

ソース・コード・レベルの事前条件・事後条件の扱いにはダイクストラ（Dijkstra）の考え方が使用できる．もう少し粒度の大きいというか抽象度の高いところではメイヤー（Meyer）の契約による設計（Design By Contract）がある．今回は下からいくのでダイクストラの方法をリファレンスとして利用する．この方法は難しいことで有名だが，紙と鉛筆だけでやっていた昔と違って形式手法系のツールを利用すると何とかなる．

2 考え方

とりあえず簡単なC言語のサンプル・プログラムに対して仕様や要求にたどり着けるかどうかやってみたい．サンプル・プログラムを用意し，それをフリーのプログラム検証ツールに掛けてプログラムを論理構造に分解する．その論理構造から仕様が出てくるかどうか調べてみよう．

● 利用するツール

ソース・コードを扱うので，気軽にコンパイルや実行のできるような環境設定が必要である．最近は便利になってコンパイラやデバッガを無料で使えるようになった．Windowsであれば，米国Microsoft社のWebサイト(1)からVC++のExpress Edition（`vcsetup.exe`）をダウンロードしてくればよい．C言語環境を使用するだけなので，SQL Serverなどのオプションはすべて外してインストールする．

インストールが終了したらプロジェクトを作成する．テンプレートから「空のCLRプロジェクト」を選択し，プロジェクトができたらソース・コードをこれに登録する．いわゆるIDE（Integrated Development Environment；統合開発環境）というものはいきなりソース・コードを書けず，たいていこの二つのステップが必要になる．そして，このやり方が何通りもあって，バージョンによって微妙にやり方が変更されるので毎回無駄な時間を使うことになる．デバッガとリンクした使い方などをマウス・クリックだけで作業したい場合は我慢するしかない．Eclipseが普及したら，少なくともEclipseプラグイン間では統一されるかと思ったがやはり微妙に違う．VC++の場合は操作方法がいろいろあるだけでなく，ライブラリの設定などもあり面倒だが，ここを乗り越えるとデバッガ連携やコード・