

代数的プログラミング環境

河口信夫

酒井正彦

坂部俊樹

稲垣康善

名古屋大学工学部

1 はじめに

代数的プログラミング、すなわち等式によるプログラミングは、参照透明性というデバuggラムや再利用に非常に有用な性質を持ち、また代数的に意味論も明確であり検証の自動化もある程度進めることができる。このような代数的プログラミングの利点は、わかりやすく間違いの少ないプログラミングの作成を助け、ソフトウェアの生産性を向上させることにある。代数的プログラミングは等式を項書換え規則とみなした項書換え系によって進むため、人間がその計算過程を直観的に理解することは困難である。そのため、代数的プログラミングの作成を支援するプログラミングミンタ環境が必要となる。近年計算機のプログラミングミンタには多くの提案がなされている[1]。本稿では代数的プログラミングミンタに必要と考えられるエディタ、ブラウザ、デバッグなどの機能や、その問題点について検討を行なった代数的プログラミングミンタ環境の構想を示す。

2 代数的プログラミングミンタ環境の機能

記号計算を基本とする代数的プログラミングにおいては、今まで手続き型言語や、オブジェクト指向言語とは違ったプログラミングミンタールが必要となる。代数的プログラミング作成の支援のために基本的なツールとして以下のものが挙げられる。

- ・ エディタ、ブラウザ：代数的プログラミングの記述の支援。
- ・ 検証系：代数的プログラミングの検証を行なう。
- ・ 直接実行系：代数的プログラミングの実行を行なう。
- ・ デバッグ：代数的プログラミングのデバッグを行なう。

これらのツールはすべてウインドウベースで、マウスなどのポインタで統一された操作を用いる。これら全てを合わせたプログラミングミンタ環境を以後「統合環境」と呼ぶ。

2.1 代数的プログラミングの形式

統合環境で代数的プログラミングを作成する場合、当然環境にあった言語を必要とする。近年、代数的言語はいくつか研究されており、OBJやCOLD-K[3]、PLUSS[4]などがある。また、我々もすでに代数的プログラミング直接実行系 Cimpl[2]や、項書換え系 pureTIs などを実現してきた。本研究ではこれらの言語の性質を採り入れ、統合環境で処理しやすい代数的プログラミング言語を設計する。ラムのモジュール化、階層化、バリエーション、またオブジェクト指向の考え方より、一つのソートをクリックできるようにとらえ、継承の機能を探り入れた言語を検討している。

2.2 エディタ、ブラウザ

エディタ、ブラウザは代数的プログラミングの階層化とデータベース化を支援する。検証系、実行系はいつでもエディタから起動でき、代数的プログラミングの参照透明性を利用して、ポリアブストラクティングを援助する。このエディタでは関数のアリティや、ソートのチェックなどはインタラクティブに行なわれ、プログラミング作成者の負担を最小限に抑える。すでに作成してあるプログラムを再利用したり、変更したりするのはエディタ中のブラウザを用いる。ブラウザとは、プログラム部品データベースからユーザの希望に合ったプログラムの階層関係や、再帰関係や、継承関係を視覚的に表示することが可能であり、ウインドウ上で

ポインタを操作することにより効率的に移動、検索を行なうことができる。このエディタ、ブラウザにより、階層化された代数的プログラミングの作成が支援され効率的なプログラミング作成が行なえる。

2.3 検証系、直接実行系

完成したプログラムの検証を行なうためのシステムとして Knuth-Bendix アルゴリズムを用いた、簡単な検証系 KBint を実現している。この検証系により様々なプログラムの検証を行なっている。統合環境では検証系はエディタと結合して、プログラム部品の検証を行ないポリアブストラクティングを援助したり、プログラム全体の整合性を確認するのに用いられる。項書換え系の処理系として我々はすでに Cimpl などを実現してきた。これらを用いて階層化した代数的プログラミングを実行するためには、代数的プログラミングを非階層化する必要があり、この非階層化については現在検討中である。

2.4 デバッグ

デバッグはプログラム開発においてエディタに次ぐ重要な位置を占めている。プログラミングの実行中の形式を視覚的に未構築で表現したり、書き換える様子を色の変化やマウスで表示したりして、手書き型言語のデバッグにある機能に加え、項書換え系特有の機能を提供して、プログラミング作成者の負担を抑える。実行系とは密に結合して、ステックのことの書換えや、ルール指定の書換え、ルールによるブレークポイント、関数記号によるブレークポイントなどが指定できる。

3 まとめ

代数的プログラミングのための統合環境を提案した。統合環境に適切な代数的言語を設計して、そのエディタ、ブラウザ、デバッグの持つべき機能を提案した。なお、これらの代数的プログラミングミンタ環境の機能のうち、代数的言語の subset に対する検証系、直接実行系はすでに実現されている。また、デバッグに必要な項のグラフィック表示も実現している。それ以外の機能は代数的プログラミング言語の設計と合わせて検討中である。

謝辞

日頃御指導賜わる平田富夫助教授、杉野花津江助手、直井徹助手、結城祥治助手をはじめ、いつも御討論を交わしてくださる稲垣研究室、阿草研究室の皆様には感謝致します。

参考文献

- [1] 高橋：関数型プログラムの作成支援システム、情報処理。Vol.29, No.8, pp.872-880(1988).
- [2] 酒井、坂部、稲垣：抽象データ型の代数的仕様の直接実行系 Cimpl、信学技報、COMP-86-67(1987).
- [3] H.B.M. Jonkers: An Introduction to COLD-K. LNCS394, pp.139-205(1989).
- [4] M.Bodit, M.C.Gaudel, A.Mauboussin: How to Make Algebraic Specifications More Understandable. LNCS394, pp.31-65(1989).