

形式仕様言語 CafeOBJ を用いたテキストエディタの仕様記述

3 K - 7

来間 啓伸

二木 厚吉†

(株)日立製作所 システム開発研究所

†北陸先端科学技術大学院大学

1. はじめに

形式仕様言語を使ったソフトウェア開発技術は、基本的な理論や言語が整備されるにしたがって、実用化の段階へと向かいつつある。

CafeOBJ 言語は、等号論理、書き換え論理、隠蔽代数などの複数の論理に基づく代数型の形式仕様言語であり、仕様の直接実行が可能であるとともに、検証や変形のための確かな基盤を持つことが特徴である。CafeOBJ 言語処理系や編集、検証支援系などのツールも整備されてきている [1]。

我々は CafeOBJ 言語のソフトウェア開発への適用性を実証することを目的として、現実のソフトウェアを対象とした仕様記述実験を進めている。本報告では、テキストエディタ Emacs の編集機能の仕様記述について述べる。

Emacs は最も広く使われている実用性の高いエディタであり、対話型のソフトウェアとしての特徴も備えている。多くのコマンドはマニュアルを見ただけでは理解できないほど複雑であり、仕様を厳密に記述するのは難しい。

実用的な仕様記述では、複雑な仕様を構造化して記述できなければならない。そのために、同型射に基づく断片的部分仕様の結合方式を導入した。

2. 仕様の構造化

自明でない複雑さや規模の仕様を記述するためには、問題を把握しやすい形に分割し構造化することが有効である。形式仕様言語は仕様の意味を厳密に与え、仕様の理解や変更における誤りを少なくするが、仕様を構造化して記述することは、人間にとって理解しやすく変更しやすい仕様を得る上では不可欠である。

さらに、形式仕様では仕様の構造化法を数学的に表現することが可能であり、より効果的に構造化法を利用できる。例えば形式仕様言語 Z では、状態空間の間の対応関係（恒等関係）と演算の論理結合によって部分仕様を結合する方法 [2] が知られており、Z による仕様記述の特徴となっている。この方法によれば、1つの対象の異なる側面を各々断片的な部分仕様として記述し、それらを結合することにより対象全体の仕様を得ることができる。部分仕様の結合は Z 言語の枠内で行なわれるので、部分仕様間の対応関係を与えれば、双方の側面を満たす仕様が機

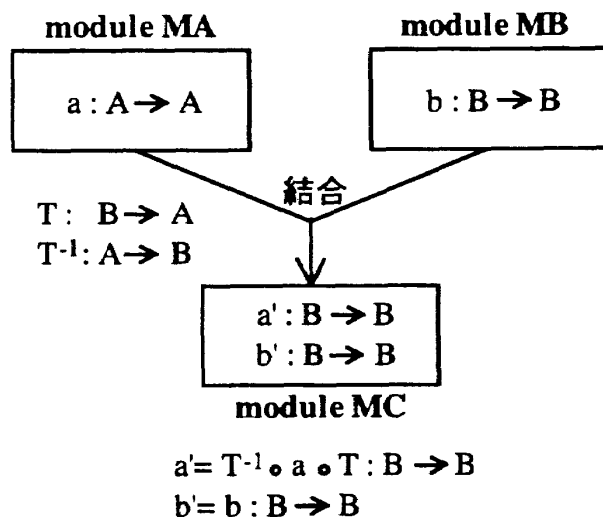


図 1: モジュールの結合

械的に得られることになる。

しかし、この方法は仕様を集合論と述語論理によって表現する Z 言語の特性に依存しており、代数型の形式仕様言語である CafeOBJ 言語にそのまま適用することはできない。ここでは、部分仕様を表す抽象データ型の間の同型性に着目し、同型射によって部分仕様を結合する、より一般的な枠組みの下に仕様を構造化することを考える。

CafeOBJ 言語では仕様はモジュールの集まりによって表されるので、部分仕様の結合はモジュールに記述された演算子の間の対応関係を与えることによって行なうことができる。この概要を図 1 に示す。この例で、モジュール MA ではソート A をドメインおよびコドメインとする演算子 a が、モジュール MB ではソート B をドメインおよびコドメインとする演算子 b がそれぞれ定義されているとする。ソート A のキャリア集合とソート B のキャリア集合の間に同型射が存在する時、ソート A のキャリア集合からソート B のキャリア集合への変換関数 T とその逆関数を使うことにより、MA と MB を結合したモジュール MC を構成することができる。この時、MC の演算子 a' 、 b' をソート B をドメインおよびコドメインとする演算子とすると、 a を図のように T とその逆関数ではさむことにより、演算子 a の性質をソート B 上で得ることができる。

この構造化法は、部分仕様を表す抽象データ型の同型性を前提としている。したがって、適用範囲は、各部分仕様を表す抽象データ型が 1つの抽象データ型に集約できる場合、言い換えれば各部分仕様は 1

つの対象の異なる側面の断片的記述であるとみなせる場合に限られる。

3. 仕様記述の範囲

仕様記述の対象は、以下に述べる Emacs の編集機能とした。一方、ファイルの読み書きやカスタム化に関する機能など、外部とのインタフェースに関する要素、メタ機能に属する要素は、記述を閉じさせるため対象外とした。

Emacs は、画面、ウィンドウ、バッファ、マーク、リージョン、ポインタなどの要素から構成される。ポインタはバッファ内を上下（行間）左右（文字間）に移動し、文字の挿入／削除などの編集操作はポインタによって示される位置に行われる。また、バッファ中にマークを設定することができ、マークとポインタにはさまれた文字列をリージョンとして、リージョン単位の挿入、削除を行うことができる。バッファはウィンドウを通じて表示／編集されるが、一つのバッファを複数のウィンドウに表示することができ、各々のウィンドウはそれぞれ異なるポインタを持つ一方、一つのウィンドウを通じて行われた編集操作の結果は他のウィンドウにも反映される。画面には複数のウィンドウを表示することができ、その内の1つのウィンドウがアクティブなウィンドウとして編集操作を受け付ける。

4. 編集機能の仕様記述

上記のように、このエディタでは各要素が密接に関わり合うため、仕様を単純に分割することは難しい。むしろ、各要素は1つの対象をそれぞれ断片的に表しているのとらえることができる。一方、まとめて1つの仕様として記述するには大きく、理解性や変更容易性の点でも構造化は必要である。ここに前述の構造化法を用いると、各要素をそれぞれ部分仕様として単純化して記述する一方で、相互に関連する部分仕様を変換関数によって結合し、各部分仕様の独立性の高い構造化した仕様を得ることができる。

例えば、1つのバッファに対して複数のウィンドウが開いている状態では、バッファ内には各ウィンドウに対応する複数のポインタが存在し、バッファの編集によって位置が更新される。このような局面では、テキストを構成する文字列中の位置によってポインタの位置を表現すると、編集操作に対するポインタ位置の更新が明確になり、理解しやすい記述が得られる。これに対し、バッファのスクロール操作では、ポインタがウィンドウから外れてしまう場合にはポインタをウィンドウ内に移動する。このような局面では、ポインタの位置をウィンドウ内の座標によって表現すると、ウィンドウに対するポインタの位置が明確になり、理解しやすい記述が得られる。これらは1つの対象を表す2つの側面であり、各々を表す部分仕様の間と同型射を設定することができる。この同型射は2つの部分仕様を結びつけ、ポインタ位置を文字列中の位置とウィンドウ内の座標の間で変換することにより、前者と後者が共に満たされる仕様を構成する。同様の方法は、より複雑

に見える部分仕様の結合、例えば編集操作の結果ポインタがウィンドウから外れてしまう時には、ポインタがウィンドウ内に戻るようにテキストをスクロールする場合にも応用することができる。

このエディタの仕様記述では4箇所での構造化法を用い、38モジュールから仕様を構成した。記述量は CafeOBJ の等式数にして約500であった。また、このように構造化した仕様は、変換関数を介するため効率が大幅に低下するが、実行可能である。

一例として、20個の編集操作からなるテストデータを与えて仕様を直接実行した場合、約10000回の項書換えが発生した。CafeOBJ インタプリタ¹はこの項書換えを約25秒で行い、直接実行による仕様の検証の点でも有効であった。この結果、部分仕様の記述と実行による検証、部分仕様の結合による拡張によって、段階的に仕様を構成することができた。

5. おわりに

一般に、代数型形式仕様言語は意味論が明確であり、仕様の直接実行や検証支援の可能性など優れた特性を持つ一方、ソフトウェアの仕様をどのように記述するかという点については他の形式仕様言語に比較して十分な実証がなされておらず、その適用性には未知の部分が多い。

ここでは、Emacs の編集機能の仕様記述実験を通じて CafeOBJ 言語の記述力を実証した。この結果、同型射を用いて仕様を構造化することにより、記述対象の複雑度や大きさに十分対応可能であるだけでなく、このように記述した仕様に対しても直接実行が有効に機能することを確かめることができた。

仕様を形式的に記述することの利点の一つは、仕様の不足や矛盾を早い段階で発見し、修正できることにある。ここに述べた記述実験でも、特に部分的な機能の組合せによって複雑な効果が現れる部分の記述において、この特性は有効であった。

〔謝辞〕

本研究に関して多くの御指導、御討論をいただいた、CafeOBJ ワーキンググループメンバーの方々に感謝します。とりわけ、CafeOBJ 言語および言語処理系についてさまざまな御支援をいただいた(株)SRA の澤田寿実氏、石曾根信氏には深く感謝いたします。

この成果は情報処理振興事業協会 (IPA) が実施している「創造的ソフトウェア育成事業」の一環として行なわれたものである。

参考文献

- [1] K. Futatsugi et al.: An Overview of CAFE Specification Environment, Proceedings of the 1st ICFEM, pp.170-181 (1997)
- [2] D. Jackson: Structuring Z Specifications with View, ACM Transactions on Software Engineering and Methodology, Vol.4, No.4, pp.365-389 (1995)

¹CafeOBJ home page <http://www.ipa.go.jp/STC/CafeP>