項書換え抽象機械における組み込み演算の処理*

3Ｈ－6

石曽根 僖[†1]　澤田 寿実[‡1]　緒方 和博[§2]
[1](株)SRA[¶]　[2]北陸先端科学技術大学院大学[‖]
e-mail: ishisone@sra.co.jp, sawada@sra.co.jp, ogata@jaist.ac.jp

1　はじめに

CafeOBJ [1] は順序ソートと等式論理に基づく代数型仕様記述言語である。この言語のインタプリタ [2] は、公理として記述された等式を、左辺から右辺への書換え規則とみなすことにより、与えられた項の簡約化を実行することができる。この処理系は LISP で実装されており、簡約化の他にもさまざまな機能を持つが書換えの速度はそれほど速くない。

そこで機能を簡約に特化し、高速に実行するために、順序ソートに基づく項書換え抽象機械 TRAM [3] をベースとしたコンパイラおよび実行系を実装中である。

2　項書換え抽象機械 TRAM

項書換え抽象機械 TRAM は順序ソートに基づく項書換え系の処理系である。書換え規則との高速な照合を実現するために、規則の左辺を木構造にエンコードし、書換え対象項をその木構造をたどるための抽象機械命令列で表す。抽象機械上でその命令を実行することで高速に書換え規則との照合が行なわれる (図 1)。

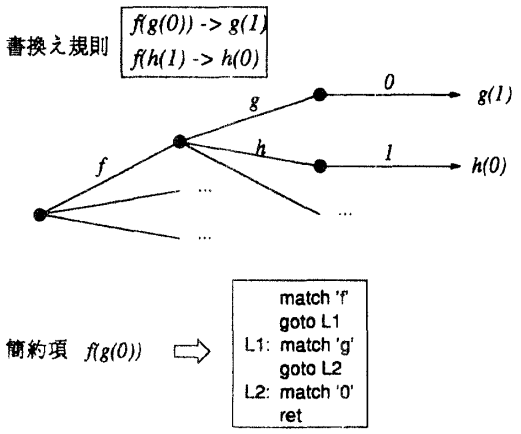

図 1: 項書換え抽象機械 TRAM 概念図

木構造の終端には、書換え規則の右辺に対応する抽象機械命令列がある。この命令列は完全なものではなく、再配置可能な参照解決前のものである。例えば右辺が変数を含んでいれば実行時に変数の参照の解決を行なわねばならない。この命令列をコピーして参照の解決などを行なうと書換えた項に対応する命令列になり、元の命令列をこれで置き換えることによって書換えが実行される。

書換えを高速に実行するため、規則に含まれる全ての関数記号 (含定数) は番号に変換され、TRAM 内部では全て番号で扱われる。

3　組み込みデータ型

CafeOBJ 処理系には整数、浮動小数、文字列といった特殊な組み込みのデータ型、およびそれらに対する各種の演算が用意されている。これらに関する書換えは処理系が実装されている LISP 処理系で処理される。これらのデータ型はしばしば利用されるものなので、コンパイラでも扱えるようにすることが望ましい。

しかし TRAM でこれらの組み込みデータ型をサポートする際には次のような問題がある。

- 組み込み型のデータの内部表現
- 組み込み型のデータ演算 (書換え)

4　データ内部表現

前述したように TRAM では全ての関数記号は内部的に番号を与えられ、それぞれについて引数などの情報を保持している。出現する関数記号がすべてあらかじめわかっていれば問題はないが、組み込み型のデータに関しては書換えのたびに新たな関数記号が出現する可能性がある。しかし書換え途中に新たに出現した関数記号を登録することは、速度的な面から採用することはできない。

そこで組み込み型のデータに関しては型毎に疑似関数記号を１つ割り当て、内部的にはその記号に対する番号で扱うこととする。これで他のデータと同様に扱うことができる。ただしこの場合、番号だけでは実際のデータがわからない。そこで組み込み型のデータの場合、対応する抽象機械命令の後に実際のデータを配置し、また照合用の木構造にも実際のデータ (へのポインタ) を追加する。そして組み込み型のデータの照合用に抽象機械命令を新たに追加し、これらの照合の際には関数記号だけではなく、実際のデータの照合も

* Implementing builtin operations in Term Rewriting Abstract Machine
† Makoto Ishisone
‡ Toshimi Sawada
§ Kazuhiro Ogata
¶ Software Research Associates, Inc.
‖ Japan Advanced Institute of Science and Technology

行なうようにする (図 2)。

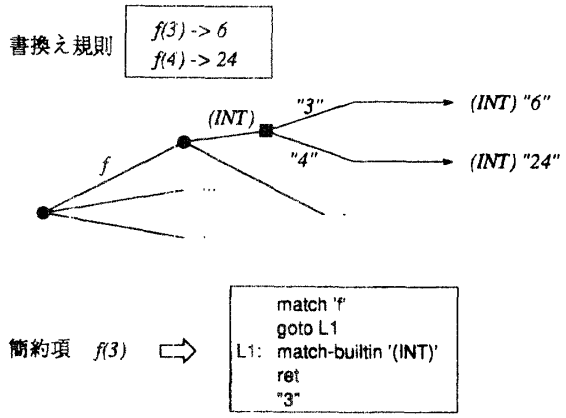

図 2: 組み込み型データ内部表現

5 組み込み演算

TRAM では書換えは書換え規則の右辺に対応する抽象機械命令列を実体化することで行なわれるが、組み込みのデータ型に関してはあらかじめ命令列を用意しておくことはできないし、そもそも組み込みのデータ型に関する演算を行なうことは項書換えに特化した TRAM の枠内では不可能である。

一方 CafeOBJ インタプリタでは、これらの演算はインタプリタの実装言語である LISP の関数を呼び出すことにより実行している。そこで TRAM でも同様の方式を採用する。すなわち、組み込みデータの演算に関してはあらかじめ演算を実行して結果に対応する命令列を生成する手続きを用意しておき、これを呼び出すこととする。組み込みのデータ型に関する演算は外部の手続きを呼び出さざるを得ないので、これは妥当な方法である。手続きには書換え規則左辺に出現する変数の束縛値のリストを渡す。

6 実装

TRAM 自体は C 言語で実装されており、この拡張機能も同様に C 言語で実装する。ただし、将来の組み込みのデータ型の追加や変更に対処できるよう、実装に柔軟性を持たせるために、基本的な機能のみを C 言語で実装し、TRAM に与える書換え規則にこれらの基本機能へのアクセスが記述できるようにする (図 3)。

さらに組み込みデータを扱う部分は TRAM 処理系本体とは切り離してモジュール化し、本体とのインターフェイスには汎用性を持たせることとする。インターフェイスは次のような基本機能を提供する。

```
X:String + (builtin-constant String "") → X
X:Int + Y:Int → (builtin-Int-add X Y)
```

図 3: 書換え規則への埋め込み

- 外部表現↔内部表現変換
  書換え規則に含まれる組み込みデータ表記と TRAM の内部表現との相互変換を行なう。
- 照合
  組み込みデータ型の照合の際に、実際のデータ内容の照合を行なう。
- 演算 (書換え)
  あらかじめ定義された各演算について、

これらのインターフェイスを備えた各モジュールを動的結合可能なライブラリ形式にすることによって必要に応じて本体とリンクできるようにする。

7 おわりに

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」の一環として行われたものである。プロジェクト自体の概要は [4] などを参照されたい。

参考文献

[1] Futatsugi, K. and Diaconescu, R., *CafeOBJ Report*, a technical report in preparation, Japan Advanced Institute for Science and Teleology, 1997

[2] Nakagawa, A.T., Sawada, T., and Futatsugi, K. *CafeOBJ Manual*, SRA, 1997; available at `ftp://www.sra.co.jp/pub/lang/CafeOBJ/Manual/manual.ps`

[3] Ogata, K., Ohhara, K. and Futatsugi, K., *TRAM: An Abstract Machine for Order-Sorted Conditional Term Rewriting Systems*. Proc. of the 8th Int. Conf. on Rewriting Techniques and Applications. LNCS 1232 Springer-Verlag. (1997) 335 - 338

[4] Nakagawa, A.T., "*Manipulating CafeOBJ on Networks*", in *Preprint for 12th Workshop on Algebraic Development Techniques*, Tarquinia, June 1997