情報科学教育におけるアルゴリズム記述に関する研究

6P－2

宮本和典　永松正博　矢鳴虎夫
九州工業大学工学部

1．はじめに

情報科学で必要な離散数学やアルゴリズムの学習では、学習者がプログラミングを行うことで、理解がより深まる。また，教師にとっても，学習者の理解を助けるための教材としてのプログラムを作ることが必要になることも多い．しかし、実際のプログラミングを行なう場合，数学的な概念やアルゴリズムそのものの学習よりも、プログラムにおけるデータ構造の細部の設計に多くの時間を費やす。そこで、データ構造の細部の設計をできるだけ必要とせずに，離散数学やアルゴリズムの学習を行えるような教育用言語Tupleを提案する．

Tupleでは，基本的な型として，組（tup型），集合（set型），関数（fun型）を持つ．また，関数を要素とする集合，組を要素とする集合を要素とする組，集合を要素とする集合から組を要素とする集合への関数等，任意に組み合わされた複雑なデータ構造を容易に作ることができる．このため，組，集合，関数という概念を用いて記述されることの多い離散数学の概念やアルゴリズムを容易に，しかも，分かり易くプログラミングできる．さらに，これらの型を処理する演算子，組込み関数等を多く準備し，プログラムの記述性を高めている．

A Computer Programming for the Education of the Information Science
K. Miyamoto, M. Nagamatsu and T. Yanaru
Department of Electrical, Electronic and Computer Engineering
Kyushu Institute of Technology
1-1 Sensui, Tobata, Kitakyushu 804, Japan

集合をベースにしたPASCAL風な手続き言語として集合指向言語SOL[1]がある．Tupleでは標準的なC言語の構文を使用する．これは，現在，大学のプログラミング教育にはC言語が広く使われていることによる．また，Tupleでは，tup型，set型，fun型の基本的な型はC++のクラスで実現し，演算子や組込み関数はオーバーロードを用いて実現している．

Tupleを用いたプログラミングでは，tup，set，fun型をベースにしたデータ構造を作るが，通常，データ構造とアルゴリズムは一体となって設計されるべきであり，最適なデータ構造の設計が最もプログラム作成者が工夫する点である．この点に関しては，Tupleでは，tup，set，fun型だけでなく，本来のC++プログラミングで作ることのできるデータ構造はすべて作ることができるので，効率が要求されるデータに対しては，プログラム作成者独自のデータ構造を構築すればよい．さらに，初めは，tup，set，fun型を用いてプログラムを書き，必要な部分を独自のデータ構造で次第に置き換えてゆくことも可能である．

2．Tuple

Tupleとは，C++言語に組（tup型），集合（set型），関数（fun型）を組み込んだものである．

2．1 型

Tupleは次の3つの基本的なデータ型を持っている．

- tup型…組を表わすデータ型
- set型…集合を表わすデータ型
- fun型…関数を表わすデータ型

それぞれの要素としては任意のfloat型，文字

列型，tup 型，set 型，fun 型を取ることができる．

２．２ 演算子

Tuple で定義したデータ型には，それぞれの型を処理する演算子をオーバーロードを用いて定義しており，これらを使うことで離散数学やアルゴリズムで扱われる対象を容易に，しかも分かりやすくプログラムとして記述することができる．例えば，s1 と s2 を集合とすると，和集合，積集合，差集合は次のように表現できる．

s1＋s2 … s1 と s2 の和集合(s1∪s2)
s1＊s2 … s1 と s2 の積集合(s1∩s2)
s1－s2 … s1 と s2 の差集合(s1－s2)

２．３ 関係演算子

関係演算子についても操作できる．その関係が真の場合は TRUE(1)，偽の場合は FALSE(0)を返し，次のように表現できる．

s1 == s2 … s1＝s2
s1 != s2 … s1≠s2
s1 <= s2 , s1 >= s2 … s1⊆s2 , s1⊇s2
s1 < s2 , s1 > s2 … s1⊂s2 , s1⊃s2

２．４ 関数

Tuple での定義済み関数の例を次に示す．

make_set(x,exp1,e,exp2,y)
引数 x,y:tup,set,fun
e:要素型（eを定義する必要はない）
exp1,exp2：変数eを使用した式
返値 なし
機能 数学的にはx={exp1｜eはyの要素でexp2を満足}と同じ

例 xがset型の変数でy={“a”,”b”,”c”}のとき，make_set(x,tup_of(e,e),e,1,y)を実行すると x={(“a”,”a”),(“b”,”b”),(“c”,”c”)}になる．

for_each(ele,Exp2,Set2){<文>}for_end
ここで，Set2はtup,set,fun型の変数か式，eleは要素型の任意の型を取る変数で，この文の内部でのみ使用可能なもの．Exp2は条件を記述する式で，Set2の要素eleのうちで式Exp2を満たすものに対して<文>の部分を実行する．<文>の部分でSet2の内容を変更してもかまわない．また，eleはプログラム中で定義する必要はない．

例 整数集合xの要素の合計を計算する場合．

```
set x;
int sum=0;
for_each(ex,1,x){
  sum += ex;
}for_end
```

例 x，yの直積を求める場合．x,yの要素が何であるかを考える必要はない．

```
set x,y,z;
for_each(ex,1,x){
  for_each(ey,1,y){
    add_ele(tup_of(ex,ey),z);
  }for_end
}for_end
```

３．おわりに

情報科学において離散数学やアルゴリズムの学習を行うのに，C 言語の構文で簡単にプログラムを組むことのできる教育用言語 Tuple を提案した．特徴として，C++の処理系のある環境ならばどこでも使用可能である．現在，DOS，Windows95，Unix 上で同様のプログラミングが行える．また，現在，大学ではプログラミングの導入教育として C 言語が広く使われているので学習し易い．さらに，データ構造がアルゴリズムと関連して，独自のデータ構造が必要なときは，プログラム作成者が必要な部分を独自のデータ構造で置き換えてゆくことも可能である．

今後，さらに，数学的な記述が容易にできるように改良していきたい．

参考文献

1）重松，吉見，吉田：集合指向言語 SOL とその処理系の開発，情報処理学会論文誌，Vol.30, No.3, pp.357—365(1989)
2）B.ストラウストラップ：プログラミング言語 C++第２版，トッパン（１９９３）