アルゴリズムの知識に基づく プログラム理解システムＡＬＰＵＳ

6Ｄ－8

村山　浩　　関本　理佳　　中島　歩　　日向野　渡　　上野　晴樹
東京電機大学　理工学部

１．はじめに

現在当研究室では、初心者用知的プログラミング環境INTELLITUTORの開発を行っている[1,2]。プログラミング環境を知的にしようとすると、プログラムの意味理解の能力が不可欠となる。

本論文では、INTELLITUTORにおいてプログラム理解の部分を担当しているサブシステムALPUS(Algorithm-Based Program Understanding System)について報告する。

２．ＡＬＰＵＳが扱うプログラムの意味と知識

まず、ＡＬＰＵＳで取り扱う理解の対象であるプログラムの意味と知識について述べる。詳しくは参考文献[3]を参照されたい。

プログラムの意味はプログラミング知識と対応しており、これらはお互いに表裏一体の関係にあるものと考えられる。一般に、意味はいろいろなレベルで捉えることができるので、先ずプログラムの意味のレベルについて考えてみる。プログラムにはいくつかの意味レベルが存在するが、これは抽象度が高いほど問題解決の目的、意図や概念に近づき、低いほど具体的操作に近づく。これは最上位から最下位へ、

１）問題解決概念レベル
２）抽象データ処理アルゴリズムレベル
３）データ処理技法レベル
４）基本データ操作レベル

の４つに階層化されるものとする。これらはプログラミング知識のレベルにも対応している。従って、プログラム理解では、プログラミング知識を利用して、４つのレベルに対応する意味を推定することが基本的な仕事となる。ＡＬＰＵＳでは、上記の４つのレベルのうち最上位を除く３つのレベル、特に中間の２つのレベルに焦点が当てられている。

３．ALPUSにおけるプログラム理解の方針

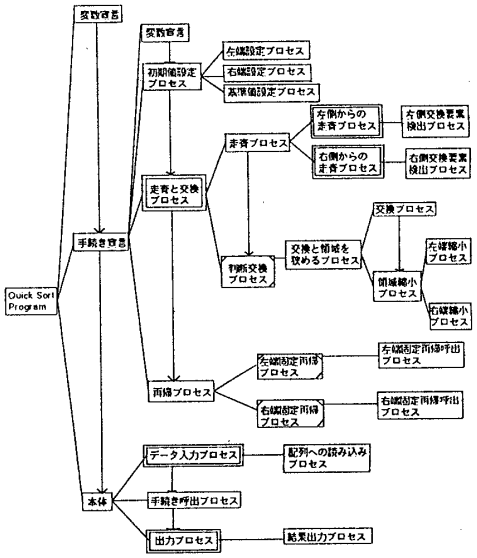
ALPUSにおけるプログラム理解は、アルゴリズムの知識を中心としているが、基本的には認知科学的な考え方に基づいている[4]。ここでいう認知科学的とは、人間のやり方を参考にしてモデル化するという意味である。人間は、与えられたプログラムを読む（理解する）時、そのプログラムコードの中から何等かの手がかりをつかもうとするはずである。プログラム名、変数名、コメント文等は手がかりを与えるが、プログラム文やプログラムの形（構造）は、より具体的な情報になると考えられる。このようないくつかの情報から、アルゴリズムや実現技法を推論し、またエラーを含んでいる場合はプログラマの意図を推定し、これらのものをテンプレートとして、プログラムコードをテンプレートマッチングによって理解しようとするものと考えられる。

我々はこのような観点から、アルゴリズムをHPGグラフと呼ぶ階層的手続きグラフで知識ベース化し、モデル推論によってミスを含むプログラムを理解しようと考えている。

４．HPGグラフによる知識表現と理解

４．１HPGグラフ

アルゴリズムはデータ構造に対する操作のシーケンスであると考えられる。これは複数のデータ処理のチャンク（まとまり）の組合せになる。このチャンクのことをプロセスと呼ぶことにすると、アルゴリズムはプロセスのシーケンスとして表現できる。



図－１　クイックソート法のHPG

A Knowledge of Algorithm Based Program Understanding System ALPUS
Hiroshi MURAYAMA　Rika SEKIMOTO　Ayumi NAKAJIMA　Wataru HIGANO　Haruki UENO
Tokyo Denki Univ.

また、各プロセスはさらに小さな副プロセスの組合せと考えることができるので、プロセスの階層グラフとしてアルゴリズムを表現することができる。これを我々は、HPG(Hierarchical Procedure Gragh) と呼んでいる。図－１にクイックソートの一般的な実現方法に基づいて書いたHPGを示す。

４．２プログラミング技法

HPGの各プロセスでは、それを実現する方法がいくつか存在するのが普通である。このような方法をプログラミング技法として持ち、プログラム理解に利用している。ALPUSでは、正しい方法だけでなく、初心者がよく犯すような誤った方法も持ち、これにより誤ったプログラムからプログラマの意図を理解している。

ここでは、プログラム技法を２節で示したプログラミング知識の分類にしたがって、次のように分類する。

１）基本データ操作

これは、プログラムコードにおとす際の方法である。具体例として図－２に代入の例(VARIABLE1←VARIABLE2)を示す。このほかには例えば、繰り返しであればWHILEやREPEATなどの方法があり、また条件判断においてはIFやCASEなどの方法がある。

Ａ）代入：標準パターン

```
VARIABLE1 := VARIABLE2
```

Ｂ）代入：エラーパターン

```
VARIABLE2 := VARIABLE1
```

図－２　基本データ操作の例

２）データ処理技法

これは特定のアルゴリズムによらず、一般的に広く使われるデータ処理の書き方を言う。この技法は、基本データ操作の組合せで表現される。具体的な例として、交換を図－３に示す。このほか個数の計数法、配列の中の最大値の求め方などが挙げられる。

Ａ）交換：標準パターン

```
WORK      := VARIABLE1
VARIABLE1 := VARIABLE2
VARIABLE  := WORK
```

Ｂ）交換：実行可能パターン

```
WORK1     := VARIABLE1
WORK2     := VARIABLE2
VARIABLE1 := WORK2
VARIABLE2 := WORK1
```

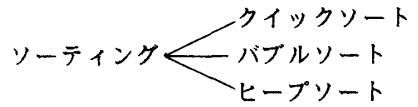
Ｃ）交換：エラーパターン

```
VARIABLE1 := VARIABLE2
VARIABLE2 := VARIABLE1
```

図－３　データ処理技法の例

３）抽象データ処理技法

こらは、データ処理技法よりも抽象度が高いために、データ処理技法では表現できないプロセスを実現する技法である。例えばソーティングにおいてはクイックソートやバブルソートやヒープソートなどがある（図－４）。また、クイックソートの副プロセスである分割プロセス（ある値より大きいものと小さいものとに分割するプロセス）では、交換法及び挿入法とでも呼ぶべき方法がある。



図－４　抽象データ処理技法

４．３プログラム理解の方法

ALPUSにおけるプログラム理解の方法は、与えられたプログラムコードから得られるいくつかの証拠に基づいて、抽象データ処理技法の中から適切なものを選択し、インスタンスを生成していく。これにより、図－１のような特定のプログラムにあったHPGグラフが生成される。次に、このHPGグラフに基づき、データ処理技法や基本データ操作の知識を用いて、プログラムコードとのマッチングをとりながらインスタンスを生成していく。完全にインスタンスが生成されたとき、完全に理解できたと考える。一部生成が失敗したとき、その部分が理解できなかったものとする。但し、理解できなかったものでも、いくつかの可能性を提示することによって修正のヒントに利用できるようにしてある。このように、理解力は知識の量と質、及び推論の能力によって決まるが、完全なものではない。

５．おわりに

現在本研究においては、対象問題を初心者が学ぶような典型的な解き方が存在する問題（クイックソート）に絞って研究を進めている。クイックソートのアルゴリズムの知識を知識ベース化することによって、初心者プログラマが作成したクイックソートのプログラム（但し、構文エラーはないものとする）のほとんどを理解することができるようになった。今後は、対象問題を広げていくことにより、本システムがどのくらい有効であるか評価していく予定である。

参考文献

[1]上野晴樹、知的プログラミング支援システムINTELLITUTORについて－背景と開発思想－、情報処理学会知識工学と人工知能37-5,1984

[2]中島慶人、上野晴樹、アルゴリズムの知識を利用した階層的手続きグラフによるプログラム理解、人工知能学会知識ベースシステム研究会資料、AI88-31,1988

[3]上野晴樹、知的プログラミング環境－プログラム理解を中心に－、情報処理、vol.28,no10,pp1280-1296,1987

[4]上野晴樹、プログラム理解システムALPUSの考え方と方法、人工知能学会研究会資料、SIG-KBS-8903-3