7N－6

プログラム理解能力向上のためのプログラムの正規化法について

神谷　始　　中島　歩　　上野　晴樹
東京電機大学

1．はじめに

当研究室では，プログラミング支援環境INTELLITURORの研究開発を行っている．INTELLITURORはGUIDE, ALPUSおよびTUTORの3モジュールから構成されている．ALPUSは，アルゴリズムの知識を中心に，プログラミング技法，バグの知識を用いて，プログラムの論理ミスを検出し，学習者の意図を推論し，訂正の助言を行うためのメッセージを出力する．現在ALPUSは，XEROX 1121 ,SUNワークステーション上で稼働している．本稿では，実際の学習者のプログラムを理解するために解決しなければならない問題点を調査した結果から判明した問題点を指摘し，その問題点の一部を改善するために，新しく追加した正規化機能（プログラム構造の単純化，定数宣言の排除，型宣言の排除，レコード型宣言の排除）の方法について述べる．

2．ＡＬＰＵＳの概要

ALPUSでは4種の知識が使われている．第1は，アルゴリズムの知識である．アルゴリズムは，複数のデータ処理の組合せであり，プロセスのシーケンスとして表現できる．これを，我々は，H-PG（Hierarchical Procedure Grah）で表現している．第2は，プログラミング技法の知識であり，HPGの各プロセスを表すノードに意味ネットとして付加される．第3は，バグの知識であり，認知科学実験で得られたバグと意図との関係を示す情報が関連するプロセスまたは技法にリンクされている．第4は，言語知識であり，プログラム言語に関する知識である．ALPUSはこれらの知識を使って，与えられたプログラムとのパターンマッチによってプログラムを理解し，バグを検出し，訂正の助言を行なう．ALPUSにおけるプログラム理解の手順は，図1に示す3つのステップによって達成される．第1ステップは，ALPUSの持っている知識とのマッチングが取り易い形に，学習者の意図を壊すことなく，学習者のプログラムを変換する処理である．これを，我々は，正規化処理と呼んでいる．正規化は，学習者の書く多様なスタイルのプログラムから少しでも多様性を減らし，プログラム理解の柔軟性を高める為に行なわれる．第2ステップは，学習者のプログラム中で使用した変数とALPUSの知識ベースにある変数の知識とを一致させる処理で，これを，変数同定と呼んでいる．第3ステップは，正規化されたプログラムの各セグメントとHPGの各プロセスとを対応付けをする．これを，プロセス同定と呼んでいる．これが，完全に対応がとれたとき，理解できたとし，そうでない場合，理解できなかったとする．

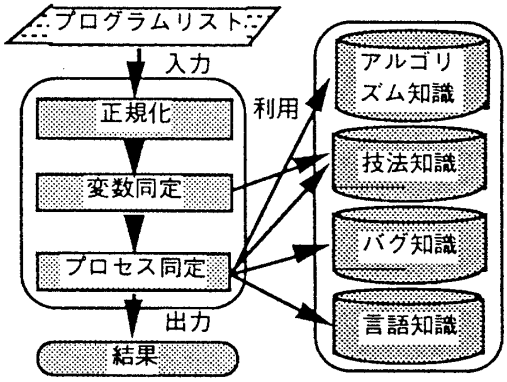

図1　ALPUSの処理手順

3．ALPUSの問題点

現在のALPUSシステムが，実際の学習者のプログラムを理解するために解決しなければならない問題点を調査した．被験者は，PASCAL言語をある程度学習した，情報工学系の大学の3，4年生を対象とした．この調査から判明した問題点と，その改善すべき機能を，図2に示す．これより，現在のシステムの問題点の５５％までが，正規化

Method of Normalization of Program for Higher Flexible Program Comprehension
HajimeKamiya
Tokyo Denki Uiniversity
Isizaka,Hatoyama,Saitama,350-03,Japan

機能の拡張によって改善できるということがわかった．

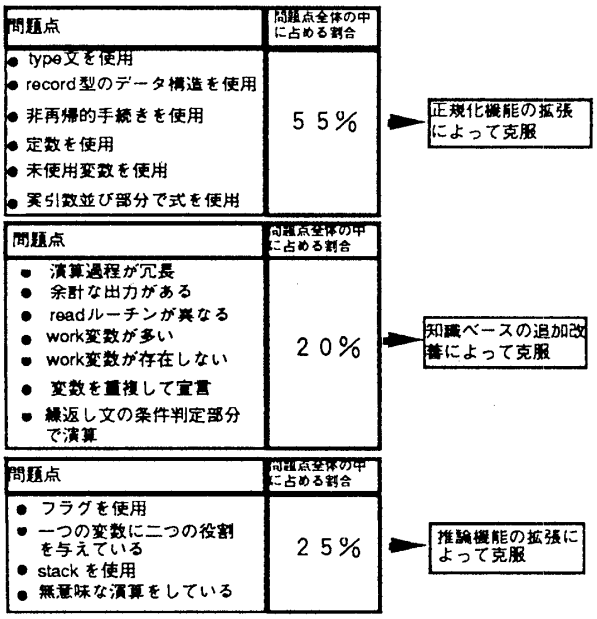

図２　調査から判明した問題点と改善箇所

４．新しい正規化機能

正規化は，図３のような手順で行なわれる．第１にプログラムコードの内部表現化，第２に，関係演算子の整列化，第３に，プログラム構造の単純化，第４に，型，定数宣言の排除，第５にRECORD宣言の排除の順に行なわれる．

図３　正規化の処理手順

４．１．内部表現化

プログラムの各文，各宣言の単位でプログラムコードに対する付加情報を付ける作業をする．付けられる情報には，行番号，文，宣言の種類，文の深さ（変数などの有効範囲などがわかる），の３種類である．

４．２．関係演算子の整列化

関係演算子の大小関係の向きを揃える作業をする．

４．３．プログラム構造の単純化

プログラムに再帰的でない手続きが存在するとき，その手続きを呼び出している手続き呼び出し文の位置に，その手続きの実行部を挿入することである．ただし，挿入することで生じる引数，変数に関する矛盾を解消するために，値引数は代入文化し，変数引数は，実引数変数を変数として使用してやることで，また，定数，型，変数宣言は，その各識別名の先頭に手続き名を付けて挿入先の各宣言部に挿入してやることで解決する．この作業で再帰的でない手続きをプログラムからなくすことが出来る．

４．４．型，定数宣言の排除

型宣言の排除では，型宣言で定義された型名が，プログラム中に存在したとき，型名をその型表記に戻し，定数宣言の排除では，定数宣言で定義された定数名をその定数に戻す作業をする．

４．５．RECORD型宣言の排除

構造化型のデータ構造であるRECORD型の変数を単純化型のデータ構造に変換することである．これを行なうには，RECORD型宣言の識別名とそのレコード変数名とを合成した名前の変数を新たにレコード変数の型名と同じ型で変数宣言してやることで，排除できる．

５．おわりに

本稿では，実際の学習者のプログラムを理解するために解決しなければならない問題点を指摘し，その問題点の一部を改善するために，新しく追加した正規化の方法について述べた．正規化機能があればプログラムの多様性をある程度吸収できることがわかる．しかしながら，正規化だけでは限界があり，他の機能の改善，追加等のさらなるシステムの改良が必要である．

参考文献

[1] Haruki Uemo : Integrated IntelligentProgramming Environment for Learing Programming , IEICE Trans. on Information and System , Vol.E77-D, No.1

[2]Haruki Ueno,and Ayumi Nakajima: INTELLITUTOR : Intelligent Programming Environment for Novices, NATO ARW on "Cognitive Models and Intelligent Environments for Learning Programming", Genova(1992).