

アルゴリズム知識とプラン知識を組み合わせたプログラム理解法*

7K-3

大熊 義嗣†

上野 晴樹‡

東京電機大学理工学部経営工学科§

概要

我々は、すでにPascalプログラムを対象とする教育向き知的プログラミング環境INTELLTUTORシステムを開発し、その有用性も評価している。INTELLTUTORの中心をなすサブシステムはプログラム理解システムALPUSであるが、今回C言語プログラムを取り扱えるようにシステムの汎用化を行うのに対応して、理解システムALPUSも見直しを行い、アルゴリズム知識とプラン知識を組み合わせた、より汎用かつ柔軟性の高い理解システムの実現を試みた。新システムでは、アルゴリズム知識を表現した階層的手続きグラフ(HPG)と、プラン知識の組み合わせをプログラム知識とする。

本稿では、このプログラム知識を用いて、複数の論理ミスを含む学生のCプログラムを理解し、バグを検出し、訂正を指摘する為の推論方法を議論する。

1. はじめに

現在のプログラム理解システムALPUSでは、アルゴリズムに基づいた知識(HPG)により、推論を行っているので複雑なプログラムが理解できる。しかし、プロセスにまたがるようなコードを、理解することは困難である。

また、プラン知識を用いている推論を行っているPROUSTシステム[1]は、簡単なプログラムしか理解することはできない。

そこで、より柔軟なプログラム理解を行うためのプログラム知識として、アルゴリズムに基づく知識(HPG)とプログラムの断片知識であるプラン知識をプログラム知識として、推論を行う方法を示す。

2. INTELLTUTORのシステム構成

ALPUS(Algorithm-Based Program Understanding System)は、アルゴリズムの知識に基づくプログラム理解システムであり、知的プログラミング環境INTELLTUTOR(Intelligent Tutoring System as Programming Environment)のサブシステムである。INTELLTUTORは、学生や初心者プログラマを助けるようにはたらく機構の実現を目標としたシステムである。

また、知的エディタ、プログラム理解システム、および知的CAIを含む統合型知的プログラミング環境である。また、このシステムは、4つの知識ベース、1つのデータベース、および3つのソフトウェアモジュールから構成されている。

知識ベースは、構文グラフを中心とした構文則に関する知識を管理する知識ベース、アルゴリズム知識を中心としたプログラムの意味に関する知識を管理する知識ベース、学生の知識と理解状態を管理する知識ベース、および教授知識を管理する知識ベースからなる。

データベースは、マニュアルやプログラム例をハイパーテキストとして管理する機構である。

ソフトウェアモジュールは、学生のプログラム入力を助けるためのガイデッドエディタGUIDE、プログラムに含まれる複数の論理ミスを検出し訂正の助言を行うプログラム理解システムALPUSと、プログラミング知識を教授するための知的CAIシステムTUTORとで構成されている。

3. 推論方法

3.1 人の推論方法

どのような目的で作成されたかをまえて知っているプログラムを人が、理解しようとしたとき、どのように推論をおこなうのだろうか。

まず、プログラム内でキーワードを探索し、そのキーワードによりプログラムの解析をはじめ、プログラムの意図を理解しようとする。

* Program Understanding by Combining Algorithm Knowledge and Plan Knowledge

† OOKUMA Yoshitugu

‡ UENO Haruki

§ Graduate School of Science and Engineering, Tokyo Denki Univ.

1. キーワードの探索

- ・変数宣言
- ・ループ・繰り返し
- ・関数呼出し・手続き呼出し 等

2. キーワードによるプログラム・ブロックの意図の理解

見つかったキーワードにより、その（キーワードを）プログラム・ブロックがどのような意図を持ってかかれたのかを理解する。

- ・グローバル変数宣言
- ・ローカル変数宣言
- ・変数の初期化
- ・変数の値の交換 等

3. ブロックがどのような制御構造になっているのかの理解

このように推論していると考えられる。

3.2 ALPUSにおける推論方法

ALPUSにおいて、プログラムの意図を理解するためにはアルゴリズムに基づく知識とプログラムの断片知識が必要としている。

よって、プログラムの制御構造を知るにはアルゴリズムに基づく知識が用い、キーワードの並びによるプログラムの意図を知るにはプログラムの断片知識を用いる。

このとき、知識をフレームシステムにより表現し、アルゴリズムに基づく知識は、HPGとしてプログラムの断片知識は、プランとして表現される。

1. HPGとプランによるフレームシステムにソースコードをインプットする。
2. フレームの付加手続きにより、プログラムの断片（プログラム・ブロック）を照合（パターン・マッチング）
3. フレームのインスタンスを生成
4. インスタンスの構造とHPGの構造を照合（パターン・マッチング）

3.3 アルゴリズムに基づく知識のみを用いた推論とプログラムの断片知識も用いた推論の違い

アルゴリズムに基づいた知識を用いた推論ではアルゴリズムに基づいてコード化されているプログラムならば理解は可能であるが、アルゴリズムに基づいてはいるがプロセスにコードがまたがってしまっているようなコードを理解することは不可能である。なぜなら、アルゴリズムに基づく知識ではコードとアルゴリズムが密接に結び付いており、知識はプロセス内で知識ベースに納められているからである。

しかし、プログラムの断片知識であるプランともとも結びつけられることにより柔軟にプログラム推論ができると思われる。

4. おわりに

本稿では、プログラム理解をプログラムを制御構造からみた知識(HPG)と、プログラムが断片的な知識の集合とみた知識から行う方法を示している。

今後の課題として以下の点を示す。

1. プログラムの断片知識の獲得
例題（クイック・ソート）が用いるすべてのプログラム断片知識の獲得
2. 知識のフレーム構造
HPGとプランの知識を効率よく用いることのできるフレーム構造の研究

参考文献

- [1] Jonson, W.E. and Soloway, E., "PROUST: Knowledge-Based Support Tool for Code Understanding", *Artificial Intelligence* 15, pp.72-122, 1980
- [2] 上野 晴樹, 知識工学入門, オーム社, 1985
- [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, no.1, pp.1280-1296, 1987.