

認知科学的アプローチに基づくプログラム理解システムについて¹

2N-8

佐藤 徳幸^{†2} 神保 至^{†3} 中島 歩[†]情報処理振興事業協会(IPA)[†] 東京電機大学理工学部経営工学科[†]

1 はじめに

プログラムの理解支援をコンピュータ上で行なわせる手法としては、人間をモデリングした認知科学に基づく方法・ソフトウェア工学に立脚した変数解析等が考えられる。しかしながら、コンピュータによるプログラム理解の方法論は確立できていないのが現状である。

本稿では、プログラムの理解をコンピュータ上で行なわせるにあたって認知科学にもとづくアプローチが有効であると考え、C言語のプログラムソースコードからその仕様書を導出する実現方法を述べる。

2 認知科学的アプローチに基づくプログラム理解

プログラムの目的はコンピュータに操作の伝達をすることであるのに対して、仕様書の目的は人間にプログラムの説明をすることである。

そこで、プログラムの要点をつかみそれを自然言語に翻訳し表現するためには、操作の記述を説明に置き換えるための何らかの推論機構が必要である。我々は、この推論機構に認知科学的アプローチをとることにした。理解ということ振り返って考えてみると、理解とは「意味づけて知る、創造的に知る、そして意義づけて知る」ことである。すなわちこれらは「こころ」の働きである。よって、「心的活動を理解しようとする試み」である認知科学を基盤にしたアプローチを採用することはコンピュータによって理解させることへの近道であると考えられる。

プログラム理解システムの実装は次のように行なう。まず、理解を行なうべきC言語プログラムを正規化する。そして、用意したプログラム知識と、ボトムアップ推論とトップダウン推論を共応的に組み合わせながらマッチングによって同定し、プログラムの意味の内部表現に変換する。最後にその内部表現を自然言語で表し仕様書を導出する。

次小節以降では、プログラム理解手法を処理手順を

追って説明する。

2.1 正規化処理

実際上のプログラムについて考えてみると、全く同じ機能を実現するにしてもそのコーディング・スタイルは殆んど無限に考えられる。さらに、複数の人間によってなされた度重なる修正や機能拡張により、1つのプログラム中に多種多様なコーディング・スタイルが使用されることも多い。しかし、考えられるパターンをすべて事前に用意しておくことは不可能である。したがって、プログラム知識とのマッチングによるプログラム理解のためには、対象となるプログラムをマッチングによる同定の前にできる限り標準的な形式に変換しておくことが得策であろう。この処理を我々は正規化と呼ぶ。

ここで行なう正規化処理は、プログラムの持つ意味やアルゴリズムを変更することなく、構造的・構文的な面から行なう。具体的には、関係演算子の整列化・カンマ演算子の置き換え・for文とwhile文の統一・定数宣言の排除等を行なう[2]。

2.2 推論処理

一般に、曖昧で不完全な知識を取り扱うには、演繹的推論の枠を越えた高次推論が要求される。プログラムの理解に於いては主題が不明瞭な場合が多い[3, pp.155-156]ので、高次推論が必要とされる。

ところで、人間がプログラムを解釈する場面を考えてみると、まずプログラムの意味上のchunkごとに解釈し、そのchunkの並びから全体の機能を導き出している。また逆に、導き出した全体の機能からプログラムのchunkの機能を推測している。すなわち、ボトムアップ推論とトップダウン推論とを共時的・協応的に利用していると考えられる。ボトムアップ推論は生成的であり、トップダウン推論は検証的であることから、これを組み合わせて推論を行なうこの方法はきわめて合理的であるといえる。

我々の推論機構では、知識フレームを用いてこのようなボトムアップ推論とトップダウン推論を交互に用いる推論方法をモデル化している。

特にプログラム理解においてトップダウン推論を行なうことは次の点で有効である。それは、予想に現れるものだけを当てはめ予測にない処理を割愛するトッ

¹A Program Understanding System based on Cognitive Science

SATOU Noriyuki (Email: noriyuki@stc.ipa.go.jp)
Information-technology Promotion Agency, Japan, (IPA)
3-1-38 Shibakouen, Minato-ku, Tokyo 105, Japan

²(株)日本コンピュータ研究所より出向中

³(株)三菱総合研究所より出向中

ブダウン推論を採用することにより、プログラムの本質と関係のない例外処理等は省いて、プログラムの本質的な意味を汲み取ることがはじめて可能となるからである。

推論機構は基本的に、次の2つのコンポーネントの集合として構築する。

- 意味表現への変換機構
- 日本文変換機構

2.2.1 意味表現への変換機構

本節では、意味表現の実装方法と意味表現への変換機構について述べる。

ボトムアップ推論とトップダウン推論により協応的に推論を進めるのであるが、それはある時点において分析した情報はそれ以前の知識情報と統合し、その知識情報が次の情報分析に影響を与えて行き、そしてその結果が期待するパターンと接合するまで繰り返すということである。

よって、ボトムアップ推論とトップダウン推論により協応的に進めて行く我々のプログラム理解システムのためには、推論方法と知識の持ち方の両方にそのための仕組みが内在していなければならない。

プログラムの意味階層を考えてみると、次のように最上位から最下位に向けて階層化をすることが出来ると考えられる [3, pp.158]。

問題解決概念レベル 代表的問題解決技法の概念や、その抽象的アルゴリズムの概念

抽象データ処理アルゴリズムレベル 代表的データ処理アルゴリズムに関する実現手続きの知識

基本データ処理技法レベル 基本的データ処理技法に関する知識

基本データ操作技法レベル 基本的データ操作や制御に関する知識

以上により、我々はおおむね4水準の階層をなす記憶構造を持ったプログラム知識構造を知識フレームとして実装することを考えている。

次に、推論方法について述べる。

少なくとも手続き型言語のプログラムの意味は「いくつかのデータ操作の集合」の「連鎖」として解釈できる。そこで、まずデータ操作のひとつかたまりをブロックとして切り分けるが、これは理解を行なうプログラムから、基本データ操作技法レベルの知識のプログラミングパターンと同定したものを順に切り取って行くことで行なう。

次に、ソースコードとのマッチングに成功した単数又は複数の基本データ操作技法レベルから上位概念知

識フレームへを参照して単数又は複数の基本データ処理技法レベルを連想することが出来る。これがボトムアップ推論である。

さらに、連想した基本データ処理技法レベルの下位概念知識フレームスロットにより、次に現れる基本データ操作処理技法レベルのコーディングを予測することが出来る。また、予測した基本データ操作技法レベルのもつプログラミングパターンスロットの内容とマッチングをとることにより、その予測が正しいかどうかを検証する。この検証により実証できたものだけを基本データ処理技法レベルの候補として残す。これがトップダウン推論である。

同様に、このボトムアップ推論とトップダウン推論を抽象データ処理アルゴリズムレベルと問題解決概念レベルでも繰り返す行なう。

こうして、最も上位の意味のレベルまで到達したものをそのプログラムの持つ意味として採用する。

2.2.2 日本文変換機構

このコンポーネントは、これまでにコンピュータによる理解によって生成した内部表現を人間が理解できる形の自然言語に変換する機構である。

前段階までに決定した各プログラムの意味レベルごとのフレームを意味レベルごとに整理した形で出力する。

3 おわりに

本稿では、認知科学的アプローチに基づくプログラム理解システムについて述べた。現在我々の考えているプログラム理解システムでは、命令コード自体には着目しているが、コメント情報・関数名や変数名の持つ意味といったものを考慮していない。これでは機能の仮説を立てるための重要なヒントを捨てているに外ならない。

コメント情報の解釈は自然言語処理の技術を応用して、関数名や変数名の持つ意味の解釈は辞書をあらかじめ用意しておくことによって、これらの持つ情報を汲み取れるのではないかと考えている。

参考文献

- [1] 竹下亨. “ソフトウェアの保守・再開発と再利用”. 共立出版, 1993.
- [2] 神谷始, 中島歩, 上野晴樹. “プログラム理解能力向上のための正規化法について”. 情報処理学会第48回全国大会, 2巻, pp. 191-192, Apr 1994.
- [3] 大須賀節男, 上野 晴樹等. “知的プログラミング”. オーム社, 1993.