5D−6

リバースエンジニアリングのためのプログラム理解システムプロトタイプ「PRIPARE」(2)ー推論処理と知識ベースー

神保 至†[1]　佐藤徳幸†[2]

†情報処理振興事業協会(IPA)
技術センター

1 はじめに

近年、ソフトウェアのリバースエンジニアリングが注目を浴びているが、従来のリバースエンジニアリング・ツールは、対象プログラム内で扱われるデータに着目したもの、すなわちプログラムの構造的な分析を行なうものが主であった。

これに対して我々は、プログラミング教育を目的とした、アルゴリズムに基づくプログラム理解手法[1]をリバースエンジニアリングに応用し、プログラムの意味的な理解を行なうことを目標に研究を進めてきた。

(1)に引き続いて、本稿(2)では、事務処理用のC言語プログラムのソースコードからモジュール仕様書レベルの情報を導出する、プログラム理解システムプロトタイプ「PRIPARE」の推論処理と知識ベースについて述べる。

2 PRIPAREの知識ベース

プログラムの意味を理解(推論)するためには、人間が持っているようなそれ相応の知識を体系化し、知識ベースとして構築しておくことが重要である。

本プロトタイプでは、プログラミングに関する知識と業務に関する知識とに分離して知識ベースを構築した。前者はプログラムのパターンや断片的なアルゴリズムを表現した、理解対象となるドメインに独立な知識である。一方、後者は社員管理／在庫管理などの各業務の処理手順を表現した、対象ドメインに依存したドメイン固有の知識である。

このように、ドメインに独立な知識とドメイン固有の知識とを分離することにより、拡張性の高い知識ベースを構築することができる。また、それぞれの知識を階層的に分類し表現することで、効率的な知識管理を行なえるように設計した。

また、これらの知識の他に、プログラミング言語(ここではC言語)に関する知識と、システムに依存する知識も必要となる。C言語知識はC言語の構文規則を表現した知識であり、システム知識はシステムコールや標準ライブラリ関数などに関する知識である。

以上の各知識間の分類・階層構造を表したものを図1に示す。

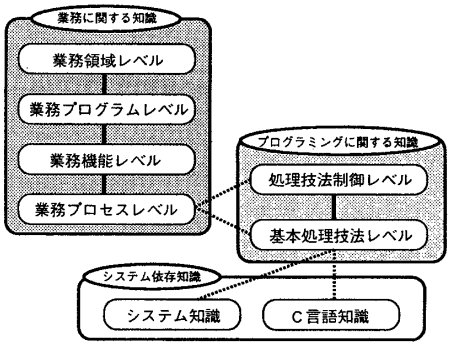

図 1: 知識ベースの構造

本プロトタイプの知識ベースは、社員管理を行う200ステップ程度のC言語プログラム20本を基に、汎用フレーム型知識工学環境ZERO[2]を利用して構築した。構築に当たっては、基になるプログラムに対して業務分析・プログラミングパターン解析を行ない、上述の分類・階層構造にしたがって、各レベルの知識を複数のフレームとして表現した。

階層化した各レベルの知識は、1つの管理用フレーム(ルートフレーム)と、これによって管理される複数のクラスフレームで構成されている。これらのクラスフレームには、中間言語コードで表現されたプログラミングパターンあるいは下位フレームのシーケンスが記述されており、後述する推論サブシステムのパターンマッチングに利用される。

各クラスフレームは、各レベルのルートフレーム／インスタンスフレームとのISA関係、および隣接した上位・下位レベルのクラスフレームとのPART-OF関係

A Program Understanding System Prototype for Software Reverse Engineering 'PRIPARE' - Inference and Knowledge Base

†JIMBO Itaru (e-mail: jimbo@stc.ipa.go.jp)
Software Technology Center,
Information-technology Promotion Agency, Japan
3-1-38 Shiba-kouen Minato-ku TOKYO 105 ,JAPAN
[1](株)三菱総合研究所より出向中
[2](株)日本コンピュータ研究所より出向中

を保持している。ISA階層関係は、知識ベース内でフレームを管理するために利用される。一方、PART-OF階層関係は、そのフレームが表現する処理の大きさの分類や処理手順を表現するために利用される(図2)。

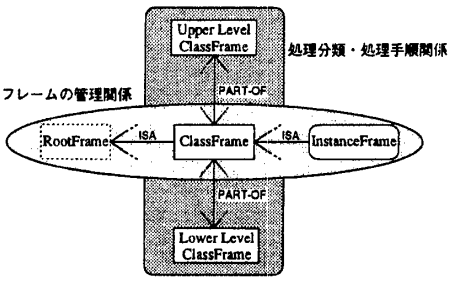

図 2: フレーム間の階層関係

3 PRIPAREの推論サブシステム

本サブシステムの推論処理は、正規化サブシステムで生成された中間言語リストと、プログラミング／業務に関する知識とのパターンマッチングが中心となる。パターンマッチングに成功した場合には、クラスフレームからインスタンスフレームを生成し、最終的には、業務領域レベルのインスタンスフレームを頂点とするフレームツリーが推論結果となる。

ここでの推論過程は、知識ベースのクラスフレームのツリー構造(PART-OF階層関係)の中から、対象プログラムの意味を最もよく表すサブツリーを探し出す探索過程と見ることができる。推論開始に当たって何も手がかりがない場合、この探索空間は知識ベース全体に広がってしまうため、本サブシステムでは最初に、ユーザから与えられるドメイン情報に基づいて、探索空間すなわちクラスフレームツリーの絞り込みを行なっている。

次にこのクラスフレームツリーから、1つの仮説(仮説サブツリーと呼ぶ)を生成する。仮説サブツリーは、PART-OF階層関係で関連付けられている各レベルの複数のクラスフレームで構成される。この仮説サブツリーのリーフノードは、基本処理技法レベルのクラスフレームとなり、このフレームには中間言語コードで表現されたプログラミングパターンが記述されている。

推論エンジンは、このプログラミングパターンと対象プログラムの中間言語リストとのマッチングにより、仮説の検証を行なう。この際、入力値の意味をユーザから与えてもらうことにより、仮説の正しさを検証する。マッチングに成功した場合は、そのクラスフレームからインスタンスフレームを生成する。

この操作を繰り返して、対象プログラムの中間言語リストが残らずパターンにマッチすれば、仮説サブツリーを上位方向に遡ってインスタンスフレームツリー全体を完成させる。また、パターンに一致しない場合にはその仮説サブツリーを廃棄し、新たな仮説サブツリーを基に検証する。すべての仮説サブツリーを試しても解が見つからない場合は、全体として最もよく対象プログラムの意味を表していると評価される仮説サブツリーを近似解とする。

推論結果は、生成されたインスタンスフレームツリーとして表現され、仕様生成サブシステムがこれらのフレームの内容を統合することにより、仕様情報を導出する。

4 おわりに

本稿では、リバースエンジニアリングのためのプログラム理解システムプロトタイプ「PRIPARE」の推論処理と知識ベースについて述べた。

ここで述べたような、知識とのパターンマッチングによる推論では、知識ベース内の知識数(フレーム数)が増大するにしたがって、推論速度は次第に低下することが予測される。これに対処するためには、知識ベースのさらなる階層化や効率的なフレーム探索アルゴリズムの採用、あるいはフレーム同士の協調による推論手法などを考慮する必要がある。

なお、本システムの詳細については[3]を参照されたい。

謝辞

本研究は、東京電機大学理工学部経営工学科上野研究室で開発された、プログラム理解システムALPUSを基盤として、実用的なプログラム理解システムの構築を目指して進められた。本研究プロジェクトを進めるにあたり、貴重な御意見・御指導を頂いた、東京電機大学理工学部の上野教授、コンサルティング委員／ワーキング委員の方々、IPAの古宮特別研究員、並びに関係者の皆様に深く感謝致します。

参考文献

[1] 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. 10, pp. 1280–1296, Oct 1987.

[2] 今井健志, 伊藤治之, 吉村貞徳, 上野晴樹. 汎用フレーム・システムZERO−その概要とユーザ・インタフェースについて. 電子情報通信学会 人工知能と知識処理研究会, No. AI-87-22, pp. 35–42, 1987.

[3] 佐藤徳幸, 神保至, 他. プログラム理解システムとその応用に関する調査研究報告書. 7技-163, 情報処理振興事業協会 技術センター, 1996.