

プログラム可視化システムにおける動作解析手法

石崎あゆみ, 内田好昭, 直田繁樹, 市川 至, 毛利友治

富士通株式会社

1. はじめに

プログラム可視化システム〔1〕は、プログラムの構造・動作を、抽象的な図形を用いて画面上にアニメーションにより表示するシステムである。可視化対象プログラムの構造・実行状態の解析を行い、ユーザにより定義される図形構造に描画命令を送ることによって、画面上に表示を行う〔2〕。

本論文では、プログラム可視化システムにおける動作解析手法について述べる。

2. 動作解析系

プログラム可視化システムにおける動作解析系は、プログラムの実行動作を解析し、画面上の図形に対する描画命令を生成する。プログラムの世界と図形の世界を対応づけるものとして、可視化指示がシステムに与えられる。この指示に従って、システムはプログラムの実行状態の変化から、図形の状態変化を引き起こす。

可視化指示には、プログラム中での状態変化列と図形の状態変化の対応を与える可視化命令、プログラムの要素と図形の要素との対応を与える写像定義がある。

動作解析系は、可視化指示を用いて、プログラムの世界での状態・構造を解析して図形の世界の状態・構造に変換する。

以下、動作解析系に用いた手法について述べる。

3. 動作解析のための方法

プログラムの実行状態をアニメーションとして表示するには、プログラムの動作列を対応する図形の動作に変換することが必要となる。従って、何らかの方法で実行中のプログラムの状態についての情報を取り出し、その状態を解析することにより、図形に対して操作を生成する必要がある。

(1) プログラムの状態の抽出

プログラムの状態を抽出する方法として、プログラム中に可視化のためのメソッドを埋め込む方法、インタプリタを用いる方法、通過した点から実行状態を取り出す方法等が考えられる。

可視化用メソッド埋め込み方式は、ソースプログラム中に可視化用メソッドを直接埋め

込み、プログラムの実行が埋め込み地点に到達してメソッドが起動されると、直接図形に対して操作が送られる方法である。この方法は、実現が比較的簡単であるが、プログラムの状態を保存するのが困難であり、細かい動作に対する可視化が難しいという欠点がある。

インタプリタ方式は、可視化専用のインタプリタを作り、その上でプログラムを実行する方法である。この方法は、プログラムの細かい動作を取り出せるという利点があるが、その反面、不必要な動作までとりだしてしまう。また、対象言語用のインタプリタを新たに作らなければならない場合がある。

トレーサ埋め込み方式は、上の2つの中間的方式である。プログラム中に可視化指示に従ってトレーサを埋め込み、実行中トレーサを通過する時、実行状態についての情報を取り出せるようにする。この方法では、インタプリタのように不必要な情報まで取り出してしまふことはない。

可視化に必要な情報を得るには、プログラム全体を解析する必要はなく、その一部分から情報を抽出すればよい。

以上の検討から、トレーサ埋め込み方式を採用する。

(2) プログラムの状態変化の保存

プログラムを実行すると、トレーサから実行状態についての情報が得られ、その情報と可視化命令に従って、システムは可視化のための条件をチェックし、条件が満たされると図形構造に描画操作を送る。ただし、論理プログラムの場合、バックトラックの制御を考慮にいれなければならない。プログラムでバックトラックが起こった時は、画面上の図形の表示状態ももとに戻す必要がある。これに対処するには、バックトラックが起きたことを検出し、プログラムが以前の状態に戻ったことを通知する方法が考えられる。

バックトラックが発生したことをトレーサによって検出し、表示画面を以前の状態に戻すために、実行中たどった状態の履歴を保存しておく必要がある。

この機構を実現するため、履歴管理部を用意し、プログラムの状態履歴を管理する。状態を保存するためにプッシュダウンオートマトン(PDA)のスタックを用いる。

複数のPDAによって分散して状態履歴が記憶され、履歴管理部によってこれらのPDAが管理されており、トレーサからメッセー

A Program Status Analyzer for A Program Visualization System
A. Ishizaki, Y. Uchida, S. Sugata, I. Ichikawa and T. Mohri
FUJITSU LIMITED.

ジは分配される。

PDAは、可視化命令に従って決定された状態遷移表を持っており、メッセージが送られると、状態遷移表に従って自分の状態を変化させ、前の状態をスタック上に保存する。この保存された状態によって、バックトラックが起きたというメッセージを受けた場合にも、以前の状態に戻すことができる。状態遷移の結果、PDAが最終状態に達した場合は、可視化命令にしたがって可視化の条件を判断し、図形に対する描画メッセージを発行する。

(3) 図形要素との対応

PDAからのメッセージ内容は、プログラムの要素を用いて記述されている。そこで、メッセージ中のプログラム要素を、図形要素に写像する必要がある。

図形構造では、図形オブジェクトの生成・削除が行われ、構造が動的に変化する。また、構造の複写を認めるため、複数の同一の構造が存在することもあり、プログラムの状態変化を直接的に図形構造に伝えたと対応できない場合があり得る。そこで、図形オブジェクトを管理する分配器を作り、分配器を通して図形構造にメッセージを送ることにする。

プログラム要素と図形要素との対応は、写像定義によって定義され、これを管理するため、分配器にmap tableを用意した。図形構造で図形オブジェクトを作成した場合は、分配器のmap tableにオブジェクトを登録する。分配器は、PDAからのメッセージを受け取ると、map tableを用いて写像を行い、図形要素に対する描画操作に変換して図形構造に送る。これにより、図形構造の動的な変化に対応することが可能となった。

4. システムの構成

以上の検討の結果、プログラム動作解析系の構成として、図1のような構成を考えた。各部の役割は、以下とする。

(1) トレーサ

プログラムの実行状態を抽出するため、可視化命令に従って可視化対象のプログラムに埋め込まれる。プログラムが実行されトレーサに処理が到達すると、トレーサよりプログラムの実行状態についての情報が得られる。

(2) 履歴管理部

トレーサから得られた情報を管理する。履歴管理部は、プッシュダウンオートマトンを管理しており、トレーサからの情報を保存する。

プッシュダウンオートマトンは、可視化命令に基づいた状態遷移表を持っており、受けるメッセージによって自分の状態を変化させる。最終状態になると、分配器にメッセージを送る。

(3) 分配器

分配器は、図形構造への橋渡しを行う。図形オブジェクトの管理をするmap tableを持ち、メッセージを受けると、map tableを用いて写像を行い、対応する図形オブジェクトに動作をさせるメッセージを送る。

5. おわりに

以上、我々のプログラム可視化システムにおけるプログラム動作解析手法について述べた。現在、本手法を用いた動作解析系を試作中である。

謝辞

本研究は、第5世代コンピュータプロジェクトの一環として行われた。御支援いただいた方々に、深く感謝いたします。

参考文献

- [1] 市川他：「プログラム可視化システムの概要」，情報処理学会第38回全国大会(1989)。
- [2] 神子他：「プログラム可視化システムにおける動作表示手法」，情報処理学会第38回全国大会(1989)。

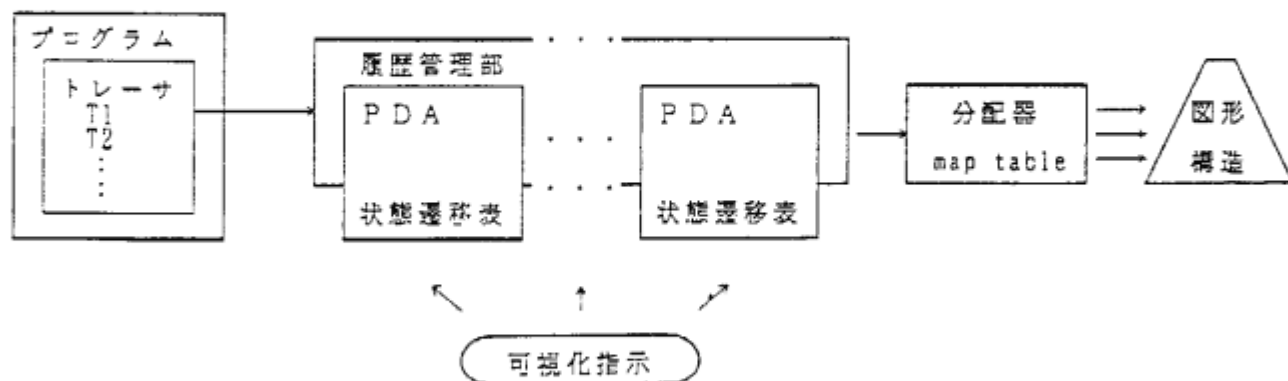


図1 プログラム動作解析系の構成図