

問題解決手法を用いたプログラム最適化の検討

3C-4

石川智浩

浅田保栄

春名修介

松下電器産業(株) 情報システム研究所

1. はじめに

コンパイラで実用化されている最適化手法 [1] が、局所的なものから大域的なものへと高度化するにつれ、形式的な処理に加えてプログラムの振舞いや、意味など、従来人工知能分野で取り組まれてきたテーマを取り入れることが必要になりつつある。

一方人工知能分野では、古くから自動プログラミングやプログラム理解などの研究がなされており、有力な手法のひとつとしてプランニングによる問題解決がある。近年、EBL, CBRなどのルールベースの学習、推論システムの研究が盛んになり、プランニングは基本的かつ、重要な要素として見直されつつある。

本システムは、C言語を対象としてプログラムの振舞いを問題解決の枠組を用いて表現し、プログラムの自動修正、最適化などの機能の実現を目指している。

2. 概要

本システムでは、C言語プログラムでの代入操作によって生じる、変数、定数の伝搬の最適化を検討する。

プログラムの構成要素は、文や演算子、組み込み関数などのプログラムを構成する基本要素と、連接、分岐、繰り返しと言った制御構造に分けられる。今回は基本ブロック(入口が先頭のみで、出口が末尾のみの順番に実行される連続した文の列)のみに問題を限定し、分岐、繰り返しなどの構造は扱わない。

問題解決の枠組を用いたプログラムの表現法について説明する。代入操作や局所変数宣言などの基本要素は、STRIPS-likeな、前提リスト、削除リスト、追加リストを持つ基本オペレータ [2] で表現される。これらは、それぞれオペレータが適用されるための条件、適用後に偽になる事実、真になる事実を表す。基本ブロックでのプログラムの振舞いは、個々の基本オペレータの連接(オペレータ・シーケンス)で表される。これをプランと呼ぶ。

コンパイラの最適化では、生成される値を静的に解析する手法が中心であるのに対し、本システムでは、プログラムをオペレータ・シーケンスに置き換えて、その振舞いを動的に追跡して最適化を行う点が異なる。問題解決手法の特徴は、初期状態から目標状態へ状態を遷移させるプランを、探索によって求めることである。これにより、与えられたプログラムの流れに必ずしもとらわれずに、プログラムの最適化を行うことができる。

3. システムの概要

Fig.1 にシステム構成を示す。トランスレータは、ソースプログラムとプランとの相互変換を行う。知識ベースには、プログラムの構成要素に関する知識が、基本オペレータとして蓄えられている。

Fig.2 に全体の処理の流れを示す。システムは、

- ・不完全なプログラム
- ・プログラムの入力状態
- ・プログラムの入出力仕様

の3種類のデータを要求する。

ソースプログラムはトランスレータでプランに変換された後、問題解決部に渡される。問題解決部では、まずプランナーによりプランのシミュレーションを行い、入力状態(初期状態)とプランから、出力状態を求める。次にプラン検証部で出力状態と出力仕様とを比較し、異なる部分が手掛かりにプランの修正情報を生成する。これをもとに、プラン修正部でプログラムの意図に合った最適なプランを作り、これをトランスレータでソースプログラムに変換して終了する。

現在、プランナーを中心として試作を行っており、次章で、問題解決部について説明する。

4. 問題解決部

問題解決部では、プログラムの振舞いをプランという形式で取り扱う。プランが最適化される様子を Fig.3 の2変数の swap 関数のプログラムを用いて説明する。この関数の機能は、Fig.4 に示すように、関数の実行前後で、2つの変数 p1, p2 の値が入れ替わっていることである。この例題は単純ではあるが、Fig.4 (b) を2つのゴールとするプランニング問題としてみた時、

- ・ゴールの干渉が起こって独立には解けない
- ・手段目標解析では解けない

と言った基本的な問題を含んでいる。

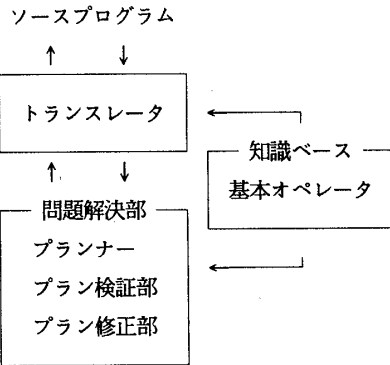


Fig.1 システム構成

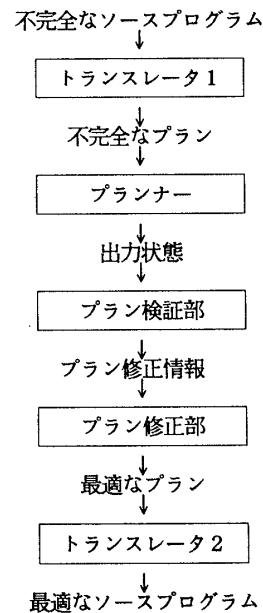


Fig.2 全体の流れ

```

swap(p1, p2)
int *p1, *p2;
{
    int w1, w2;
    w1 = *p1;
    w2 = *p2;
    *p1 = w2;
    *p2 = w1;
}
    
```

Fig.3 2変数の swap 関数の C プログラム

