

Program Slicing

Mark Weiser

IEEE Trans SE 10-4, 352-357, July 1984.

Introduction

- プログラムスライス：
データフローと制御フローを用いたプログラムの分解
- 自動的にプログラムの射影が作られる
 - プログラムの抽象化や手続き化は手動
- テスト、並列実行、デバッグなどの用途
- 熟練プログラマーはスライスを無意識に求めている

定義

- 出入り口各1つのフローグラフ
- Trajectory: $(n1, s1)(n2, s2)(n3, s3), \dots$ 実行トレース系列
- Projection: i のところのみ、 V の変数に関するのみの実行トレース系列
- Slice: S
 - 元のプログラム P から0個以上の文を削除してできる実行可能プログラム
 - P が入力 I に対して止まるなら S も I に対して止まり、 S と同じ projectionを持つ

スライス発見法

- 最小文数のスライスを求められない(停止問題に帰着)
- データフローに着目して、conservative なスライスを求める
- 直接影響を与える変数集合 $R_C^0(n)$
- データフローによってスライスに含まれる文集合 S_C^0
- 制御フローによってスライスに含まれる文集合 S_C

手続き間スライス

- 呼び出し文で、影響を受ける変数をパラメータ変換して、呼び出された手続きの最後から新たな基準で
- 呼び出された文の最初から、呼んだ文へ、パラメータ変換して、新たな基準で

分離コンパイル

- システムコールなど内容がわからない手続き呼び出し、呼び出され
- 最悪のケースを想定。全ての引数が定義される...

スライス計算の例

- 500から900行の簡単な19個のコンパイラのプログラム
- VAX11/780で36時間

並列実行

- スライス毎に別のプロセッサで計算
- 結果を合成すれば高速計算できる

関連研究

まとめ

- 自動的に小さな同じ振る舞いをするプログラムを生成
- 理解しやすい、デバッグしやすい
- 問題点
 - 計算量
 - 小さくないケース