

| | |
|--------------|---|
| Title | シナリオからの並行プログラムの合成についての研究 |
| Author(s) | 渡邊, 裕 |
| Citation | |
| Issue Date | 1999-03 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/1286 |
| Rights | |
| Description | Supervisor:平石 邦彦, 情報科学研究科, 修士 |

シナリオからの並行プログラムの合成についての研究

渡邊 裕

北陸先端科学技術大学院大学 情報科学研究科

1999年2月15日

キーワード: シナリオ, 並行プログラム, 超逐次プログラミング, 依存関係, 同期命令.

複数の逐次プログラムが通信を繰り返しながら並行的に動作を行う並行プログラムでは, 逐次プログラムのようなデバッグ手法が確立されておらず, そのテスト, デバッグには多大の労力を要する. 特に, 並行プログラムでは, 生成される状態数がプログラムのサイズに対し指数関数的に増大する状態空間爆発の問題があり, 全状態を生成して検査するには膨大な時間を必要とする. したがって, 並行プログラムの作成段階からできるだけ誤りを含まないようにすべきである.

並行プログラムに特徴的な誤りとして, 並行動作の非決定性により生じるものがある. このような誤りは, 逐次プログラムのデバッグに比べ, 発見することが困難である場合が多い. そこで, あらかじめ期待する正しい動作に対応する並行プログラムの逐次的な実行過程 (これをシナリオと呼ぶ) を与え, それに基づいて並行プログラムを合成することにより, このような誤りの発生を防ぐ手法が内平らによって提案された. この手法が「超逐次プログラミング」である.

「超逐次プログラミング」では, 各命令にラベルを付けることにより, プログラムの実行系列を記号列として抽象的に取り扱う. このとき, 各命令間には共有変数の *read*, *write* など, 実行順序が結果に影響を与える依存性が存在するが, これは, ラベルの集合上の反射的かつ対称的な関係として与える. まず, 並行プログラムに対するシナリオから, それを有向グラフで表現したシナリオグラフを構築する. シナリオグラフの命令間に同期命令を挿入し, 各逐次プログラムに射影する. そして, 各同期命令について, それを除去しても依存関係に関するシナリオとの等価性を保つならば, すなわちその同期命令が冗長ならば, それを除去する. 最後にコード化を行うことにより, 期待通りに動作する並行プログラムを得る.

この手法では、冗長な同期命令の削除が試行錯誤的に行われてることから、効率的とは言えない。また、作成された並行プログラムの最適性についての議論も十分とは言えない。本研究では「超逐次プログラミング」の手法を改良し、冗長な同期命令をほとんど含まず、かつ、正しい動作をする並行プログラムの効率的な合成アルゴリズムを提案した。

提案アルゴリズムでは、シナリオはラベルの集合上の正規表現として与えられる。アルゴリズムは、従来手法と同様に、シナリオのブロック化、同期命令の挿入、シナリオの復元、射影、コード化の各手順からなる。

最初のブロック化とは、シナリオを制御構造を含むブロックと含まないブロックに分割することである。分割した各ブロックに対して同期命令の挿入を行う。同期命令は、シナリオによって与えられた半順序関係を保つように挿入される。挿入位置は、シナリオを各ラベルを節点に、ラベルのシナリオ上での順序関係を有向枝にもつ有向グラフで表現し、それに対し推移的閉包、依存関係のない有向枝の除去、推移的リダクションの各操作を順次適用していくことにより一意に求められる。挿入位置の一意性は、有向非循環グラフの推移的リダクションの一意性に関する結果を用いて証明できる。その後、全てのブロックを代入していくことによりシナリオを復元する。復元したシナリオを各逐次プログラムに対して射影し、さらに、各ラベルを対応する命令に置き換えることにより、並行プログラムが得られる。

提案アルゴリズムによって合成される並行プログラムは、制御構造を含まない場合、シナリオに基づいた正しい動作を行い、かつ、冗長な同期命令を含まない。また、シナリオによって与えられた半順序関係を満たす全ての実行系列を生成できるという意味での最適性も保証される。制御構造を含む場合についても正しい動作を行うことが証明できるが、最適性は必ずしも保証されない。これはループなどを1つのブロックとして取り扱っているためである。