

「条件つき採録」についての回答書

「情報処理学会論文誌 プログラミング」に投稿しました論文に対して丁寧な査読をいただきありがとうございました。また、有用なコメントいただき感謝いたします。

論文番号：PRO2021-1-(7)

論文題目：再帰的ブロック構造を持つ並列プログラムに対する可逆実行環境

著者：池田崇志、結縁祥治

採録の条件に対する回答およびそれに伴う論文の修正について回答いたします。（査読報告の条件2については、修正点のリストに含めます。）

条件1に対する回答：

本論文の手法は、査読報告書にあるようにプログラムの逆解釈ではなく、プログラムを変換して順方向実行を逆に辿ることを可能にします。したがって、逆実行という言葉を使わず、抽象機械が順方向モードと逆方向モードという2つのモードにおいて、それぞれ命令を（順方向に）実行するというように修正しました。現在の実行では、初期状態から順方向モードで終了するまで実行し、順方向での実行で得られた値スタックとラベルスタックを入力として、逆方向モードで終了するまで実行する、としました。このために、「3.1 抽象機械の実行モード」という節を「3.2 順方向モードの実行環境」の前に追加して、抽象機械の2つのモードについて説明しました。モードという考え方に基づいて、順方向モード (3.4.2) と逆方向モード (3.4.3) に分けて振舞い定義を記述しています。

「逆実行」、「逆から実行」、「逆方向の実行」は、「逆方向モードの実行」というように記法を修正しました。修正に対する詳細は以下のリストにあげます。

表記ゆれ

一般的な言葉遣い

- 「手続き呼び出し」「関数呼び出し」「再帰呼び出し」→「手続き呼出し」「関数呼出し」「再帰呼出し」
- 「および」→「及び」
- 「手続」→「手続き」
- 「毎」→「ごと」

上記の言葉遣いに基づいて以下の点を修正しました。

修正点

1. p1. 「手続き呼び出し, 関数呼び出しを含むように」 → 「手続き呼出し, 関数呼出しを含むように」
2. p2. 「個々のブロック毎に」 → 「個々のブロックごとに」
3. p2. 「本発表では, ... 再帰的な手続き呼び出しを導入する」 → 「本論文では, ... 再帰的な手続き呼出しを導入する」
4. p2. 「手続および関数」 → 「手続きおよび関数」
5. p2. 「手続き呼び出しのブロック, 関数呼び出しのブロック」 → 「手続き呼出しのブロック, 関数呼出しのブロック」
6. p2. 「・手続き呼び出し」 → 「・手続き呼出し」
7. p3. 「・関数呼出し」 → 「・関数呼出し」
8. p3. 「関数の呼び出しは呼び出し名と関数名」 → 「関数の呼び出しは呼出し名と関数名」
9. p3. 「並列ブロックのネスト構造, 手続き呼び出し及び関数呼出しの拡張」 → 「並列ブロックのネスト構造, 手続き呼出しおよび関数呼出しの拡張」
10. p5. 「3.1.4 手続き呼び出しおよび関数呼び出し」 → 「3.2.4 手続き呼出しおよび関数呼出し」
11. p5. 「手続き呼び出しの引数は」 → 「手続き呼出しの引数は」
12. p13. 「再帰呼び出しに従って」 → 「再帰呼出しに従って」
13. p13. 「発生した呼出パスを」 → 「発生した呼出しパスを」
14. p13. 「このことで, 手続呼出と関数呼出において」 → 「このことで, 手続き呼出しと関数呼出しにおいて」

補足コメントに対する回答

- p.3 図1中のフォントと本文中のフォントについて一致させるようにしました。
- オーバーヘッドについては、本論文では現時点では対応する逆方向実行の実現、目的とするデバッグのための情報保存のための原理的な制限であると考えています。特に、値スタックの圧縮については今後の課題としています。オーバーヘッドについての考え方を明確にするために、「4.3 オーバーヘッド」の節を追加し、「5. おわりに」の章の最後にオーバーヘッドの解決についてのコメントを追加しました。

詳細な修正点

(修正場所の指定は、6/4 提出の発表資料による)

1. (和文概要 6 行目) 「バイトコードを順方向および逆方向において並行実行する」→「バイトコードを順方向および逆方向の 2 つのモードで並行実行する」
2. (英文概要 8 行目) 「executes multiple abstract machines」→「concurrently executes multiple abstract machines in the forward and backward modes」
3. (p2 左 第 1 段落 下から 3 行目) 「可逆計算に基づいて... できるようになる」を削除
4. (p2 右 上から 2 行目) 「抽象機械は逐次的に」→「抽象機械は順方向と逆方向の 2 つのモードを持ち、各モードで逐次的に」
5. (p2 右 第 2 段落 最初) 「本発表では」→「本論文では」
6. (p2 右 第 2 段落 2 行目) 「再帰的なブロック構造を許すことによって動的な並列ブロックの逆方向実行を可能にする。」→「並列に実行されるブロックに再帰的なブロック構造を持つプログラムに対する逆方向の実行を実現する。」
7. (p2 右 2.1 節 第 1 段落) DV, DP, DF, RV を斜体に修正
8. (p3 図 1 中) 「 $B ::= E == E \ E > E$ 」→「 $B ::= E == E \mid E > E$ 」
9. (p3 図 1 見出し) 「definition of language」→「Language definition」
10. (p3 左 「手続き呼出し」の段落中) 「手続きの引数は変数一つのみとし」→「手続きの引数はただか変数一つのみとし」
11. (p3 左 「関数呼出し」の段落中 2 行目) 「関数は簡単のため引数一つとする」→「関数は簡単のため、ただか引数一つとする」
12. (p3 左 「関数呼出し」の段落中 3 行目) 「関数内では関数の名前を返り値とする」→「関数は、その関数名の変数の値を返す。」
13. (p3 右 最下行) 「イメージを示す」→「概略を示す」
14. (p3 図 3) 「前向き」→「順方向」 「逆向き」→「逆方向」
15. (p4 表 1 見出し) 「順方向の抽象機械命令セット」→「順方向モードの抽象機械命令セット」
16. (p4 表 1 見出し) 「instruction of abstract machine for forward」→「Instructions of abstract machine in the forward executio mode」
17. (p4 左) 「3.1 順方向実行環境」→「3.2 順方向モードの実行環境」
18. (p4 左) 「3.1.1 順方向実行環境の概要」→「3.2.1 順方向モード実行環境の概要」
19. (p4 左 下から 4 行目) 「名前付けを行い以降は」→「名前付けを行い、以降は」
20. (p4 右 3.1.3 第 1 段落 4 行目) 「生成する並列テーブルはこの par0 と par1 の番地を組にして保存し各ブロックの開始番地終了番地を保存している」→「並列テーブルはそれぞれ並列ブロックの開始と終了の番地であるこの par0 と par1 を組にして保存している」

21. (p5 右 表 2) 「逆方向の抽象機械命令セット」 → 「逆方向モードの抽象機械命令セット」
「instruction of abstract machine for backward」 → 「instruction of abstract machine in the backward mode」
22. (p5 左) 「3.1.5 抽象機械の順方向出力」 → 「3.2.5 抽象機械の順方向モードにおける出力」
23. (p5 右 3.1.5 第 1 段落) 「この値は逆方向に実行するときに」 → 「この値は逆方向モードで実行するときに」
24. (p5 右 3.1.5 第 1 段落) 「r.free は順方向で」 → 「r.free は順方向モードで」
25. (p5 右) 「3.2 逆方向実行環境の概要」 → 「3.3 逆方向モードの実行環境」
26. (p5 右 3.2 第 1 段落) 「逆方向の抽象命令セットを」 → 「逆方向モードの抽象命令セットを」
27. (p5 右 3.2 第 2 段落) 「逆方向の実行では」 → 「逆方向モードの実行では」
28. (p5 右 3.2 第 2 段落) 「順方向実行のバイトコードの抽象命令」 → 「バイトコードの抽象命令」
29. (p5 右 3.2 第 2 段落) 「抽象命令を個々に変換した逆方向実行のバイトコードを抽象機械に与え」 → 「抽象命令を個々に変換し」
30. (p5 右 3.2 第 2 段落 下から 2 行目) 「順方向バイトコード系列 s から逆方向バイトコード系列への…」 → 「順方向モードにおけるバイトコード系列 s から逆方向モードにおけるバイトコード系列への…」
31. (p5 右 3.2 第 3 段落) 「逆方向の実行に必要な情報は」 → 「逆方向モードの実行に必要な情報は」 段落)
32. (p6 右 3.2.3) 「3.2.3 手続き、関数の逆方向の振舞い」 → 「3.3.3 手続き、関数の逆方向モードの振舞い」
33. (p6 右 3.2.3 第 1 段落 1 行目) 「手続き、および関数は逆方向の命令では」 → 「手続き、および関数は逆方向モードの命令では」
34. (p6 右 3.2.3 第 1 段落 6 行目) 「リターン命令の逆命令は特に動作を必要としない」 → 「リターン命令は nop に変換される。」
35. (p6 右 3.2.4) 「3.2.4 並列ブロックの逆方向の振舞い」 → 「3.3.4 並列ブロックの逆方向モードでの振舞い」
36. (p7 左) 「順方向の振舞い定義」 → 「順方向モードの振舞い定義」
37. (p7 左 3.3.2 1 行目) 「抽象機械のバイトコード (b, o) の振舞い」 → 「抽象機械の順方向モードにおけるバイトコード」
38. (p8 右) 「3.3.3 逆方向振舞い定義」 「3.4.3 逆方向モードの振舞い定義」
39. (p8 右 3.3.3 1 行目) 「逆方向のためのバイトコード」 → 「逆方向モードにおけるバイトコード」

40. (p9 左 3.4 1 行目) 「図 2 を順方向実行のバイトコードに変換した…」 → 「図 2 を順方向モード実行のバイトコードに変換した…」
41. (p9 左 3.4 第 1 段落 3 行目) 「逆方向実行のバイトコード」 → 「逆方向モードにおけるバイトコード」
42. (p9 左 3.4 第 2 段落 2 行目) 「ラベルスタック, 値スタックを使って逆方向実行を進めていくと順方向実行のバイトコードの」 → 「ラベルスタック, 値スタックを使って逆方向モード実行を進めていくと順方向モード実行のバイトコードの」
43. (p9 右 図 11) 「図 11 プログラム例: 順方向実行のバイトコード」 → 「図 11 プログラム例: 順方向モードのバイトコード」
44. (p9 右 図 11) 「Fig.11 sample program: a byte code of forward execution 」 → 「Fig.11 Sample program: byte codes in the forward execution mode」
45. (p10 右 第 2 段落 1 行目) 「図 1 4 が生成した順方向実行の」 → 「図 1 4 が生成した順方向モード実行の」
46. (p10 右 第 2 段落 4 行目) 「…ことで順方向の実行を行う」 → 「…ことで順方向モードでの実行を行う」
47. (p11 左 第 3 段落 1 行目) 「図 17 の逆方向実行バイトコードと図 15, 図 16 の逆方向実行に必要な情報を用いて順方向の実行を逆方向に辿る」 → 「図 17 の逆方向モードのバイトコードと図 15, 図 16 の逆方向モードの実行に必要な情報を用いて順方向モードの実行を逆方向に辿る」
48. (p13 左 下から 3 行目) 「Erlang ではアクターモデルに基づいた分散した非同期の振舞いを行う。」 → 「Erlang はアクターモデルに基づいた分散した非同期の並行実行意味を持ち,」
49. (p13 おわりに 第 1 段落 8 行目) 「逆方向実行のバイトコード列は」 → 「逆方向実行を実現するバイトコード列は」
50. (p13 おわりに 第 1 段落 9 行目) 「対応するバイトコードに変換することによって得られる」 → 「対応する逆方向モードのバイトコード列に変換することによって得られる」
51. (p13 おわりに 第 1 段落 10 行目) 「バイトコードを設計するとともに」 → 「双方向のモードのバイトコードを設計するとともに」
52. (p14 おわりに 最終段落のあと) 以下の段落を追加

本手法をそのまま実システムに用いるにはオーバーヘッドがとても大きい。特に逆方向実行の実現のための空間的なオーバーヘッドが大きいことが問題である。これに対して、再帰の深さが深くなるごとに大きくなるパスの保存に必要なメモリを減らすために、パス情報を共通化する、もしくは圧縮することによって空間的オーバーヘッドを改善することが想定される。さらに、予め対象とする性質を定めることによって、逆方向モード実行の情報を限定して空間的なオーバーヘッドを減少させることは今後の課題である。

以上、修正の上で再提出いたします。再度、査読のほど、よろしくお願いいたします。