

可逆プログラミング言語の引数渡し機構の拡張

1. 研究分野

可逆計算

—可逆プログラミング言語

2. 目的

Janus を用いた可逆プログラミングが容易になるようにする。

3. 背景

・Janus には、プロシージャ呼び出しの実引数は互いに異なる変数もしくは配列変数でなければならないという制限。

・R 言語は、可逆性の保証はコンパイル後の低レベル可逆プログラミング言語の可逆性に依っている。

4. アプローチ

・大域変数・局所変数を用いて、プロシージャ呼び出しの引数に添字付き配列変数を渡す。また、Janus に同一の配列変数や変数を渡すことができるようにすることで、配列変数の代入に同一の配列変数が用いられるようにするような拡張。

・環境・記憶域モデルを用いた形式的意味論を用いて、拡張された Janus が可逆であることの証明。

・代入の行われる配列変数が右辺式にも現れる文も実行できるように、Janus の構文の制約を緩めてその部分の意味論を拡張。

・文献[39]の Janus に、大域変数を導入し、同一の変数をプロシージャに複数個渡せるようにする。これらの構文に対応する推論規則を追加するような Janus の意味論の拡張。

・変数と値の対応関係しか管理できない状態モデルの代わりに、左辺値をうまくあつかえる環境・記憶域モデルを用いる。

・環境の形を制限し、記憶域の更新に可逆な更新のみを用いるといった制約を与えることで、推論規則の可逆性を保証。

5. 結果

既存の文献にあった Janus をアプローチにあった拡張・証明をすることで大域変数を持つ可逆プログラミング言語 Janus のプロシージャの実引数に

- ・大域変数
- ・添字付き配列変数の参照
- ・同一の参照を持つ変数
- ・添字付き配列変数

を渡せるようにした.

6. 有用性

- ・プロシージャ呼び出しを行うプログラムをより簡潔に記述できる.
- ・代入を部分的に効率化することができる.
- ・Janus を用いた可逆プログラミングが容易になる

7. 限界・短所

提案する推論規則を可逆なランタイムシステムにおいてより効率良く実現すること.

8. 次に何を読めばいいか？

[1] Kazutaka Matsuda and Meng Wang: Sparcl: A Language for Partially-Invertible Computation. Proceedings of the ACM on Programming Languages, Volume 4 (ICFP): 118:1-118:31 (2020).

<https://doi.org/10.1145/3409000>

[2] Mogensen T.Æ. (2020) Hermes: A Language for Light-Weight Encryption. In: Lanese I., Rawski M. (eds) Reversible Computation. RC 2020. Lecture Notes in Computer Science, vol 12227. Springer, Cham.

https://doi.org/10.1007/978-3-030-52482-1_5