

# Foundations of Reversible Computation

## 1. 研究分野

可逆コンピューティング  
ーサーベイ

## 2. 目的

可逆コンピューティングについてのサーベイ論文.

## 3. 背景

可逆コンピューティングはエネルギー効率の良いコンピュータや, シミュレーション, ロボット工学, デバッグなどの幅広い応用分野を持つ.

参考)

・ エネルギー消費 :

Landauer, R.: Irreversibility and heat generated in the computing process. IBM J. Res. Dev. 5, 183–191 (1961)

・ シミュレーション :

Carothers, C.D., Perumalla, K.S., Fujimoto, R.: Efficient optimistic parallel simulations using reverse computation. ACM Trans. Model. Comput. Simul. 9(3), 224–253 (1999)

・ ロボット工学 :

Laursen, J.S., Schultz, U.P., Ellekilde, L.: Automatic error recovery in robot assembly operations using reverse execution. In: Intelligent Robots and Systems, IROS 2015, pp. 1785–1792. IEEE (2015)

・ デバッグ :

McNellis, J., Mola, J., Sykes, K.: Timetraveldebugging: rootcausing bugs in commercial scale software. CppCon talk (2017). [https://www.youtube.com/watch?v=1YJTg\\_A914](https://www.youtube.com/watch?v=1YJTg_A914)

## 4. アプローチ

可逆コンピューティングについて, 特に2015~2019年の結果をまとめる.

## 5. 結果

### 圏論

可逆計算の圏論を用いたモデル化について  
調査中

### 可逆言語

可逆言語は、ハードウェアとアプリケーションの橋渡しを行う、可逆コンピュータ開発の中心的役割を持つ。R-Coreは構造化可逆言語で単純な設計であるが、可逆性を持つ。R-Whileは基礎的な研究や教育に利用しやすい可逆言語である。CoreFunは型付けされた可逆関数型言語である。JouleとROOPLは可逆オブジェクト指向言語である。

**R-Core** : Gluck, R., Yokoyama, T.: A minimalist's reversible while language. IEICE Trans. Inf. Syst. E100-D(5), 1026–1034 (2017)

**R-While** : Gluck, R., Yokoyama, T.: A linear-time self-interpreter of a reversible imperative language. Comput. Soft. 33(3), 108–128 (2016)

**CoreFun** : Jacobsen, P.A.H., Kaarsgaard, R., Thomsen, M.K.: CoreFun: a typed functional reversible core language. In: Kari, J., Ulidowski, I. (eds.) RC 2018. LNCS, vol. 11106, pp. 304–321. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-99498-7\\_21](https://doi.org/10.1007/978-3-319-99498-7_21)

**Joule** : Schultz, U.P., Axelsen, H.B.: Elements of a reversible object-oriented language. In: Devitt, S., Lanese, I. (eds.) RC 2016. LNCS, vol. 9720, pp. 153–159. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-40578-0\\_10](https://doi.org/10.1007/978-3-319-40578-0_10)

**ROOPL** : Haulund, T., Mogensen, T.Æ., Gluck, R.: Implementing reversible object-oriented language features on reversible machines. In: Phillips, I., Rahaman, H. (eds.) RC 2017. LNCS, vol. 10301, pp. 66–73. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-59936-6\\_5](https://doi.org/10.1007/978-3-319-59936-6_5)

可逆言語は可逆意味論を持つ(可逆意味論から成り立つ言語が可逆言語) :  
Gluck, R., Kaarsgaard, R., Yokoyama, T.: Reversible programs have reversible semantics. In: Reversibility in Programming, Languages, and Automata, RPLA 2019. Lecture Notes in Computer Science. Springer (2019, to appear)

可逆意味論の基礎 :

Gluck, R., Kaarsgaard, R.: A categorical foundation for structured reversible flowchart languages. In: Mathematical Foundations of Programming Semantics, MFPS 2018. Electronic Notes in Theoretical Computer Science, vol. 341, pp. 155–171. Elsevier (2018)  
Gluck, R., Kaarsgaard, R.: A categorical foundation for structured reversible flowchart languages: soundness and adequacy. Logical Methods Comput. Sci. 14(3) (2018)

動的メモリ管理 :

Mogensen, T.Æ.: Reversible garbage collection for reversible functional languages. New Gener. Compu. 36(3), 203–232 (2018)

レジスタ割り当て :

Mogensen, T.Æ.: RSSA: a reversible SSA form. In: Mazzara, M., Voronkov, A. (eds.) PSI 2015. LNCS, vol. 9609, pp. 203–217. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-41579-6\\_16](https://doi.org/10.1007/978-3-319-41579-6_16)

可逆データ型(可逆なデータ構造) :

Cservenka, M.H., Gluck, R., Haulund, T., Mogensen, T.Æ.: Data structures and dynamic memory management in reversible languages. In: Kari, J., Ulidowski, I. (eds.) RC 2018. LNCS, vol. 11106, pp. 269–285. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-99498-7\\_19](https://doi.org/10.1007/978-3-319-99498-7_19)

非可逆と可逆を合わせるハイブリットアプローチ：  
Schordan, M., Ooppelstrup, T., Thomsen, M.K., Gluck, R.: Reversible languages and incremental state saving in optimistic parallel discrete event simulation. In: Ulidowski, I., et al. (eds.) Reversible Computation. LNCS 12070, pp. 187–207. Springer, Cham (2020)

### **項書き換え**

式の項を別の項に置換する手法  
双方向変換  
調査中

### **膜コンピューティング**

生物細胞から新しい計算モデルを作成しようとするもの  
調査中

### **プロセス計算**

並行システムを形式的にモデリングしようとするもの  
調査中

### **ペトリネット**

離散分離システム  
調査中

### **オートマトン**

調査中

### **量子計算・量子機械学習**

調査中

6. 有用性

7. 限界・短所

8. 次に何を読めばいいか？