Software and Reversible Systems: A Survey of Recent Activities

1. 研究分野

可逆コンピューティング
ー可逆ソフトウェア
ーサーベイ

2. 目的

可逆コンピューティングにおけるソフトウェアについて，その近年の研究成果についてまとめる(2020)

3. 背景

エネルギー効率最適化を動機として始められた可逆コンピュータの研究は，従来のソフトウエアシステムに信頼性，強靭性及びスケーラビリティをもたらした.

4. アプローチ

この論文では，COST Action IC1405 on Reversible Computation グループの近年の研究成果について主にまとめる．内容には可逆ソフトウェアの近年の動向とbehavioural types、回復、並行、デバッグソフトウェア及びオブジェクト指向システムの近年の研究成果が含まれる.

5. 結果

Behavioural Types(行動型)

行動型：並行または分散計算に用いられる型理論．行動型により，コンポーネントが他のコンポーネントと正しく相互作用できるかどうかを自動的に判断できる(例えば通信など).

・*Hu ̈ttel, H., et al.: Foundations of session types and behavioural contracts. ACM Comput. Surv. 49(1), 3:1–3:36 (2016)*

行動型には、
・binary session types：2つのコンポーネント間の契約
・multiparty session types：複数のコンポーネント間の振る舞い
2つの種類がある.

可逆と binary session types のための monitored semantics の研究が行われた.

・*Mezzina, C.A., P ́erez, J.A.: Reversible semantics in session-based concurrency. In: Proceedings of the 17th Italian Conference on Theoretical Computer 2016, Volume 1720 of CEUR Workshop Proceedings, pp. 221–226 (2016). CEUR-WS.org*

・*Mezzina, C.A., P ́erez, J.A.: Reversible sessions using monitors. In: Proceedings of the Ninth Workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software, PLACES 2016. EPTCS, vol. 211, pp. 56–64 (2016)*
・*Mezzina, C.A., P ́erez, J.A.: Reversibility in session-based concurrency: a fresh look. J. Log. Algebr. Methods Program. 90, 2–30 (2017)*

multiparty session types の例として，あるシステム全体の通信方法を型で規定する global types がある．global types 及び local types を拡張して，可逆的に，既に実行されたプロトコルを追跡できる.
・*Mezzina, C.A., P ́erez, J.A.: Causally consistent reversible choreographies: a monitors-as-memories approach. In: Vanhoof, W., Pientka, B. (eds.) Proceedings of the 19th International Symposium on Principles and Practice of Declarative Programming, pp. 127–138. ACM (2017)*

いつでも，通過したチェックポイントにロールバックできる multiparty session types
・*Castellani, I., Dezani-Ciancaglini, M., Giannini, P.: Concurrent reversible sessions. In: Meyer, R., Nestmann, U. (eds.) International Conference on Concurrency Theory, CONCUR 2017. LIPIcs, vol. 85, pp. 30:1–30:17. Schloss Dagstuhl - Leibniz- Zentrum fuer Informatik (2017)*
・*Dezani-Ciancaglini, M., Giannini, P.: Reversible multiparty sessions with check- points. In: Gebler, D., Peters, K. (eds.) Proceedings Combined 23rd International Workshop on Expressiveness in Concurrency and 13th Workshop on Structural Operational Semantics, EXPRESS/SOS 2016. EPTCS, vol. 222, pp. 60–74 (2016)*

Behaviourl contracts：クライアントとサーバの通信時に，通信が従う規律
Behaviourl contracts の可逆計算を利用した拡張
・*Barbanera, F., Lanese, I., de'Liguoro, U.: A theory of retractable and speculative contracts. Sci. Comput. Program. 167, 25–50 (2018)*

Recovery(回復)
分散システム(マイクロサービスアーキテクチャ)は，開放的で，スケーラブルで，長期運用が求められエラーに寛大？(tolerant to faults)である必要があり，設計が困難．実際，開発は予期しない動作が発生することも踏まえ行われている.
→分散システムをフレキシブルで強靭にするためには，回復(recovery)と適応(adaptation)メカニズムを導入する必要がある
→現在はこのメカニズムはプログラムに組み込まれることが多い
→デバッグなどが困難に...

↓これを解決するための可逆計算を用いたアプローチ

回復と適応の特定を行う抽象的なアプローチ
・*Cassar, I., Francalanza, A., Mezzina, C.A., Tuosto, E.: Reliability and fault- tolerance by choreographic design. In: Francalanza, A., Pace, G.J. (eds.) Proceedings Second International Workshop on Pre- and Post-Deployment Verification Techniques, PrePost@iFM 2017. EPTCS, vol. 254, pp. 69–80 (2017)*

・*Francalanza, A., Mezzina, C.A., Tuosto, E.: Reversible choreographies via moni- toring in Erlang. In: Bonomi, S., Rivi`ere, E. (eds.) DAIS 2018. LNCS, vol. 10853, pp. 75–92. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93767-0 6*

回復のための静的分析

・*Neykova, R., Yoshida, N.: Let it recover: multiparty protocol-induced recovery. In: 26th International Conference on Compiler Construction, pp. 98–108. ACM (2017)*

システムのロールバック

・*Giachino, E., Lanese, I., Mezzina, C.A., Tiezzi, F.: Causal-consistent rollback in a tuple-based language. J. Log. Algebr. Methods Program. 88, 99–120 (2017)*

・*Vassor, M., Stefani, J.-B.: Checkpoint/Rollback vs causally-consistent reversibility. In: Kari, J., Ulidowski, I. (eds.) RC 2018. LNCS, vol. 11106, pp. 286–303. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99498-7 20*

・*Elnozahy, E.N., Zwaenepoel, W.: Manetho: transparent rollback-recovery with low overhead, limited rollback, and fast output commit. IEEE Trans. Comput. 41(5), 526–531 (1992)*

・*Lanese, I., Mezzina, C.A., Schmitt, A., Stefani, J.-B.: Controlling reversibility in higher-order pi. In: Katoen, J.-P., K ¨onig, B. (eds.) CONCUR 2011. LNCS, vol. 6901, pp. 297–311. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3- 642-23217-6 20*

Reversibility and Object-Oriented Languages(可逆オブジェクト指向言語)

可逆言語におけるオブジェクト指向は Joule で最初に実現された．その後 ROOPL が開発された．これらではオブジェクト指向で一般的である，カプセル化，継承，メソッド，仮想関数を可逆的に実装できる．

・*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*

・*Joule 拡張の JouleR Schultz, U.P.: Reversible object-oriented programming with region-based memory management. In: Kari, J., Ulidowski, I. (eds.) RC 2018. LNCS, vol. 11106, pp. 322–328. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99498-7 22*

・*ROOPL Haulund, T., Mogensen, T.Æ., Glu ¨ck, 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*

これらの言語により，factory desing pattern や iterator design pattern などが可逆的に実装可能である．もちろん，抽象データ型などのオブジェクト指向の特徴も．

・*Schultz, U.P.: Reversible object-oriented programming with region-based memory management. In: Kari, J., Ulidowski, I. (eds.) RC 2018. LNCS, vol. 11106, pp. 322–328. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99498-7 22*

オブジェクト指向言語においてはメモリ管理が求められる．

→Knuth の Buddy Memory algorithm に基づく可逆ヒープメモリマネージャによる ROOPL　の拡張．

・*Buddy Memory algorithm Knuth, D.E.: The Art of Computer Programming: Fundamental Algorithms. Addison-Wesley, Boston (1998)*

・*可逆ヒープメモリマネージャ Cservenka, M.H., Glu ¨ck, 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*

可逆 Janus self-interpreter で抽象構文木が表現されているように，複雑なデータ構造も，整数と配列のみを含む単純な型システムで表現できる.

・*Yokoyama, T., Glu ¨ck, R.: A reversible programming language and its invertible self-interpreter. In: Partial Evaluation and Program Manipulation, Proceedings, pp. 144–153. ACM (2007)*

可逆言語はデータを上書きできず，exit predicate (fi など)が必要なため，従来のアルゴリズムとデータ構造は，可逆言語で計算可能なものに考え直す必要がある．しかし，可逆オブジェクト指向による抽象データ型は，その作業を従来に比べ容易にする可能性がある.

・*Glu ¨ck, R., Yokoyama, T.: A linear-time self-interpreter of a reversible imperative language. Comput. Softw. 33(3), 108–128 (2016)*

・*Glu ¨ck, R., Yokoyama, T.: Constructing a binary tree from its traversals by reversible recursion and iteration. Inf. Process. Lett. 147, 32–37 (2019)*

・*Yokoyama, T., Axelsen, H.B., Glu ¨ck, R.: Towards a reversible functional language. In: De Vos, A., Wille, R. (eds.) RC 2011. LNCS, vol. 7165, pp. 14–29. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29517-1 2*

Reversing Imperative Concurrent Programs

不可逆的な言語に可逆性を追加する研究が行われている.

・*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)*

・*Perumalla, K.: Introduction to Reversible Computing. CRC Press, Boca Raton (2014)*

・*Schordan, M., Jefferson, D., Barnes, P., Oppelstrup, T., Quinlan, D.: Reverse code generation for parallel discrete event simulation. In: Krivine, J., Stefani, J.-B. (eds.) RC 2015. LNCS, vol. 9138, pp. 95–110. Springer, Cham (2015). https://doi.org/ 10.1007/978-3-319-20860-2 6*

・*Schordan, M., Oppelstrup, T., Jefferson, D., Barnes Jr., P.D., Quinlan, D.J.: Auto- matic generation of reversible C++ code and its performance in a scalable kinetic Monte-Carlo application. In: SIGSIM-PADS 2016 (2016)*

・*Vulov, G., Hou, C., Vuduc, R.W., Fujimoto, R., Quinlan, D.J., Jefferson, D.R.: The backstroke framework for source level reverse computation applied to parallel discrete event simulation. In: WSC 2011 (2011)*

可逆性はプログラムのデバッグに役立つ.

・*Chen, S.-K., Fuchs, W.K., Chung, J.-Y.: Reversible debugging using program instrumentation. IEEE Trans. Softw. Eng. 27, 715–727 (2001)*

・*Engblom, J.: A review of reverse debugging. In: System, Software, SoC and Silicon Debug, pp. 1–6. IEEE (2012)*

・*Giachino, E., Lanese, I., Mezzina, C.A.: Causal-consistent reversible debugging. In: Gnesi, S., Rensink, A. (eds.) FASE 2014. LNCS, vol. 8411, pp. 370–384. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54804-8 26*
・*Lanese, I., Nishida, N., Palacios, A., Vidal, G.: CauDEr website. https://github. com/mistupv/cauder*
・*Lanese, I., Nishida, N., Palacios, A., Vidal, G.: CauDEr: a causal-consistent reversible debugger for Erlang. In: Gallagher, J.P., Sulzmann, M. (eds.) FLOPS 2018. LNCS, vol. 10818, pp. 247–263. Springer, Cham (2018). https://doi.org/10. 1007/978-3-319-90686-7 16*
・*Undo Software: Undodb. Commercial reversible debugger. http://undo-software. com/*
・*Hoey, J., Ulidowski, I.: Reversible imperative parallel programs and debugging. In: Thomsen, M.K., Soeken, M. (eds.) RC 2019. LNCS, vol. 11497, pp. 108–127. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21500-2 7*

Reversible Debugger for Message Passing Systems

Message Passing Systems(クラウドシステムなどの分散システム)において，因果一貫性は重要な性質である(例えば，データベースをバックアップ・複製する場合，全てのデータが一致していないと困る)．しかし，コードの誤りによるバグから，この性質が保たれないことがある．可逆的なアプローチから，バグの存在を見つけ出すことに成功した．

・*Danos, V., Krivine, J.: Reversible communicating systems. In: Gardner, P., Yoshida, N. (eds.) CONCUR 2004. LNCS, vol. 3170, pp. 292–307. Springer, Hei- delberg (2004). https://doi.org/10.1007/978-3-540-28644-8 19*
・*Giachino, E., Lanese, I., Mezzina, C.A.: Causal-consistent reversible debugging. In: Gnesi, S., Rensink, A. (eds.) FASE 2014. LNCS, vol. 8411, pp. 370–384. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54804-8 26*

Control Theory

可逆的な計算を用いた，セルラーネットワークのアンテナ位置選定システム

・*Siljak, H., Psara, K., Philippou, A.: Distributed antenna selection for massive MIMO using reversing Petri nets. IEEE Wirel. Commun. Lett. 8(5), 1427–1430 (2019)*

障害回復処理

・*Siljak, H., Psara, K., Philippou, A.: Reversing Petri nets for resource management in wireless networks (2019, Manuscript in preparation)*

波の時間反転

・*Siljak, H., de Rosny, J., Fink, M.: Reversible hardware for acoustic wave time reversal. IEEE Commun. Mag. 58(1), 55–61 (2020)*
・*Fink, M.: Time reversal of ultrasonic fields. I. Basic principles. IEEE Trans. Ultra- son. Ferroelectr. Freq. Control 39(5), 555–566 (1992)*

波の現象をシミュレーションするために可逆セルオートマトンが用いられている．

・*Bagnoli, F., Rechtman, R., El Yacoubi, S.: Control of cellular automata. Phys. Rev. E 86(6), 066201 (2012)*
・*Bagnoli, F., Siljak, H.: Control of reversible cellular automata (2019, Manuscript in preparation)*

6. 有用性

7. 限界・短所

ソフトウェア開発における可逆性の活用は，まだ初期段階である．

8. 次に何を読めばいいか？