

双方向変換網における操作変換を用いた競合解決手法

仲野 祐希[†] 日高 宗一郎[‡]
 法政大学 情報科学部 コンピュータ科学科[§]

1 導入

双方向変換とは、2つもしくはそれ以上の情報源の間で変換を介して一貫性を維持する仕組みのことである。多数の情報源の間で同期を取る場合、図1のように自然な双方向変換の結合方法として、(a)逆方向変換である *put* が直列に並ぶ場合、(b)*put* が向き合う場合の2通りの場合が考えられる。このうち、(b)の場合は、競合が不可避である。先行研究 [1] では、操作変換を応用したアルゴリズムを適用することで2つのビューの更新内容をどちらも考慮したデータ更新ができる双方向変換を実現しているが、3つ以上の更新に現実的な操作変換で対処するには操作変換の適用方法の制御が求められる。本研究では3つ以上のビューの更新内容を考慮した双方向変換を実現することを目的とする。

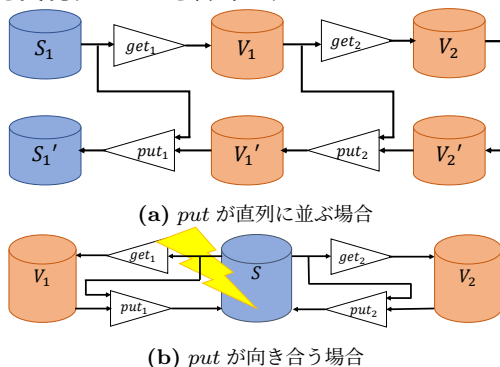


図 1: 多数の情報源の間での双方向変換の例

1.1 操作変換

文字列を対象とした操作変換 [2] では、挿入、削除といった操作を引数とする変換関数 T により、互いの文字列が一致するような編集操作を算出することで、共同編集文書の一貫性を保つことができる。操作変換の主な特性として、データを更新する主体の数に応じて C_1 と、 C_2 がある。 C_1 では2つの場合に、 C_2 では3つ以上の場合も、適用順によらず編集結果が等しくなる。図2は、共同編集ソフトウェアで起こる操作変換の過程を例示し

たものであり、本研究で利用する特性 C_1 に対応する。

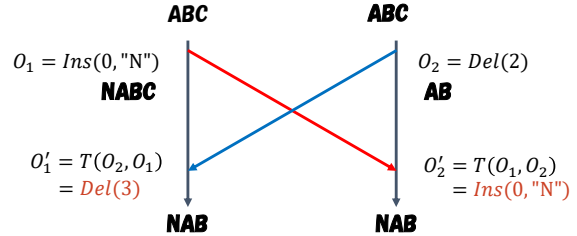


図 2: 特性 C_1 に対応する例

2 関連研究

本研究では、木構造について C_1 が満たされていることが Coq[3] を用いて数学的に証明済である Sinchuk 他による研究 [2] を操作変換の実装として利用する。 C_2 に関して、Sinchuk 他は、極めて特殊な状況を対象とすることから満たす必要はないとして、証明は省略している。

本研究の先行研究として、土生による研究 [1] が挙げられる。一般的な双方向変換では、ソースとビューは1対1の関係であるが、ソースとビューが1対2の場合に対応するため、複製されたデータをマージする能力を備えた原子的なコンビネータ Dup を持つ双方向変換言語 X を応用している。Dup は順方向変換ではデータの複製を、逆方向変換ではマージを行う。土生はこの Dup を拡張することにより、順方向に対応した *mget*、逆方向に対応した *mput* という新しい関数を定義・実装している。*mget* はソース S を複製してビューを2つ生成する。*mput* は2つのビューを入力とし、マージする。*mput* の特徴として、入力される複製のビューに更新が発生した場合には、必ずビュー間の競合解決を行う必要がある。操作変換を利用することにより、2つのビューの更新内容が異なる場合においても、どちらの更新も考慮したデータ伝播が可能となる。実装の実行結果から、X 言語の特性 GETPUTGET, PUTGETPUT, 操作変換の特性 C_1 を満たすことが確認されている。

3 本研究の特徴

操作変換の特性 C_1 のみが満たされている場合に、三者以上がデータを更新すると、編集結果は操作の適用順に依存してしまう。この問題に対して、(1) 特性 C_2 を満たす (2) 何らかの方法で適用順を決定するという2つ

Conflict Resolution with Operational Transformation in Bidirectional Transformation Network
[†] Yuki Nakano [‡] Soichiro Hidaka
[§] Department of Computer Science, Faculty of Computer and Information Sciences, Hosei University

のアプローチが考えられる。Randolph 他 [4] は, C_2 の証明がしばしば管理不能ほどに複雑であるため困難であること, 特性 C_1, C_2 をともに満たすテキストバッファの変換関数を実装することは不可能であることを示唆している。

以上の理由から, 本研究では (2) のアプローチを選択し, ブロックチェーンを適用順決定の手法として用いることを提案する。簡単な適用順決定の方法として, 操作が行われた際にタイムスタンプを保持し, これに従った順番で変換を適用する方法が考えられる。しかしながら, 複数の操作が同時に行われ, 発行されたタイムスタンプが等しい場合, 適用順決定ができない。ブロックチェーンを利用する場合, ブロックチェーン全体の合意を可能とする, 様々なコンセンサスアルゴリズムが存在し, 適用順決定に利用できる。二者が同時に 2 つの異なる文字を共同編集文書の同じ位置に挿入する場合, 特性 C_1 を満たすためには, 実装では一貫してどちらかの更新を優先することが必要だと考えられる。現に, Sinchuk 他 [6] の木の操作変換ではどちらの更新を優先するのかプログラマが真偽値で決定しており, 本研究ではブロックチェーンの性質を用いて更新の優先度を決定する。

3.1 コンセンサスアルゴリズムとブロックチェーン基盤ソフトウェアの選定

ブロックチェーンで採用されているコンセンサスアルゴリズムは, ブロックを接続するために一定の難易度の暗号パズルを解く作業であるマイニングを用いるものとして大別される。ビットコイン [5] 等で採用されている Proof-of-Work(PoW) と, Ethereum[6] で採用されている Proof-of-Stake(PoS) では, マイニングが必要となる。解が別々に見つかった場合は一時的な分岐となるが, 任意の分岐を選択し, 次のブロックへの接続を試みる。これを継続し, 結果的に最も長くなったチェーンを正当なチェーンとする。PoW を改良した PoS は, 多くのデジタル通貨を持つものほどブロックの接続権が得られやすい点で PoW と異なるが, ブロックへの接続は競争的に行われるため正当なチェーンが確率的に決定される点は PoW と同じである。一方, Raft[7] は, Hyperledger Fabric[8] においてトランザクションの順序付けを担当する Orderer ノードで, トランザクションの順序を決定する際に推奨されている手法であり, ブロックへの接続は決定的に行われる。複数の操作が同時に行われた際の適用順決定の手法として利用することを考えると, 確率的な振舞いが含まれる場合, 共同編集ソフトウェアで同じように操作したとしても, 共有される文字列が異なることが想定され, 実験の再現

性が担保できない。従って本研究では, Raft を採用して実装が公開されている Hyperledger Fabric をブロックチェーン基盤として利用する。

3.2 ブロックチェーンによる改ざん検知

多数の情報源の間での双方向変換として, 共同編集ソフトウェアを考えた場合, 更新を伝播する際には複数の双方向変換を繋ぐ部分でネットワークを経由する場合があることから, 操作を行った人側の操作と操作を受け取った人側の操作が異なるといった, 改ざんの余地がある。ブロックチェーンを利用することで, こうした改ざんの検知が期待できる。

4 評価実験

XML 形式のファイルを用意する。三者以上により, 同時刻に同じ XML 要素に対し更新を行った場合, 検討した操作変換が期待する結果と本研究の提案手法により得られた結果が一致しているかどうかを評価対象とする。また, ブロックチェーンを適用順決定の手法として用いた場合, データの改ざんを検知できるかどうかという点についても評価を行う。ここでのデータの改ざんは, 3 節で述べた状況を指す。

5 まとめ

本稿では, 3 つ以上のビューの更新内容を考慮した双方向変換を実現するにあたり, 複数の操作が同時に行われた際の適用順決定の手法として, ブロックチェーンを用いることを提案した。多数の情報源の間での双方向変換として共同編集ソフトウェアを考え, 評価実験を行う予定である。

参考文献

- [1] Mikiya Habu and Soichiro Hidaka. Conflict resolution for data updates by multiple bidirectional transformations. *SFDI2021 (to appear)*, 8 2021.
- [2] Sinchuk et al. Verified operational transformation for trees. In *International Conference on Interactive Theorem Proving*, pp. 358–373. Springer, 2016.
- [3] Institut National de Recherche en Informatique et en Automatique. The coq proof assistant. <https://coq.inria.fr/>.
- [4] Randolph et al. On consistency of operational transformation approach. *Electronic Proceedings in Theoretical Computer Science*, Vol. 107, p. 45–59, Feb 2013.
- [5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. <https://bitcoin.org/bitcoin.pdf>.
- [6] Ethereum. <https://ethereum.org/>.
- [7] D.Ongaro and J.Ousterhout. In search of an understandable consensus algorithm. In *USENIX ATC 14*, pp. 305–319, June 2014.
- [8] Elli Androulaki, et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proc of EuroSys'18*, p. 15, 2018.