Intents on the Extended UTxO Ledger

Polina Vinogradova[1][0000−0003−3271−3841] and Michael Peyton Jones[2][0000−0003−0602−1657]

[1] Input Output Global, Canada firstname.lastname@iohk.com
[2] Input Output Global, United Kingdom
firstname.lastname@iohk.com

Keywords: blockchain · ledger · distributed exchange · UTxO · extended UTxO · intents

UTxO Ledgers. A blockchain ledger is a distributed program that tracks the transfer of cryptocurrency assets, and its state is updated by applying users' transactions. A UTxO ledger (first introduced in BitCoin [7]) is a style of ledger which, at its core, contains a data structure called a UTxO set. Each entry in the UTxO has a unique identifier. This entry also stores an amount of (possibly heterogeneous [3]) assets, and a specification of the constraints on the kinds of transactions that are permitted to remove (spend) this entry from the set, e.g. a transaction signed by a specific public key. Entries in the UTxO set can never be altered, only added or removed.

EUTxO Smart Contracts and Formal Verification. An extended UTxO (EUTxO) ledger [4] allows UTxO entry spending permissions to be expressed via user-defined code with boolean output, called a *smart contract*, These boolean functions can be combined to create sophisticated programs that are executed on the ledger, e.g. games, distributed exchanges, or auctions. In the EUTxO model, all data required to execute a smart contract is included *in the transaction itself*. This allows contracts to provide users with robust formal guarantees independent of the ledger state. For example, a contract can guarantee that when an exchange offer is accepted, the transaction makes specific payouts to each party based on an exchange rate that is *fixed at the time of transaction construction*. Formal models of on-chain EUTxO contract execution can be used to specify and prove these kinds of contract properties [8, 2].

Transaction Inflexibility. The downside of this model is the inflexibility of transaction and contract construction. For example, a user may want to allow a transaction to perform an exchange at a rate within a certain *range*, or may not care about who the exchange is performed with (we call this *counterparty irrelevance*). Construction of these types of underspecified transactions is not supported in the EUTxO model. Here we discuss possible ways to address this by changing the ledger program processing blocks and transactions while maintaining certain existing formal guarantees, such as predictable script validation outcomes and fees. We discuss an existing proposal for processing a specific class of underspecified transactions (Babel fees), then generalize the underlying validation zones framework to other use cases.

Babel Fees. The *Babel fees* update to the EUTxO model [5] allows users to perform swaps of a pre-specified quantities of assets at a pre-determined rate,

but with counterparty irrelevance. This is done by modifying the atomic unit of validation (in blocks, etc.) from a singleton transaction to a list of transactions. This list, called a *validation zone*, forms a complete specification of what changes must be made to the UTxO set. The way in which a transaction is under-specified in this case is that it contains "missing assets". Such assets are usually combined with other assets available to anyone that is willing to construct a transaction that supplies the missing assets to form a complete validation zone. While the flexibility offered by this change is limited, the general approach is the basis for a variety of applications called *intents*.

Intents and Validation Zones. There is existing research into intents [6], however, ours focuses on implementing them on the EUTxO ledger. The goal of intents is to reduce reliance on fully specifying transactions and on the use of on-chain contracts in favour of partially specified transactions that allow some flexibility in how their intended actions will be fulfilled on-chain. We propose to do this via validation zones. In the EUTxO model, processing a valid transaction requires updating the UTxO set. With intents, processing a validation zone requires not only updating the UTxO set, but also updating a temporary structure `Intents`. A zone is only valid if every transaction makes a valid update, and `Intents` is empty after all transactions are applied. The exact type and update procedure for this structure depends on the details of the intent being implemented, e.g. Babel fees requires it to record all (still) missing assets. Research into building an `Intents` structure that can support arbitrary kinds of intents is ongoing work.

Off-chain Communication. It is impossible to determine whether an intent in an intent-containing transaction will be fulfilled until some user produces a transaction fulfilling it. So, to avoid a possibility of a DDoS of the blockchain network, unfulfilled intent-containing transactions must be communicated off-chain. Intents, however, aim to minimize the required off-chain communication. In particular, our Babel fees design ensures that there is no further action required from the user constructing and submitting an intents-containing transaction, e.g. signing the transaction fulfilling the intent, which is required in the Algorand atomic groups design [1]. Off-chain distribution schemes and decisions-making about what intents a user might choose to fulfill are important questions for future research, but are outside the scope of this work.

Applications. In addition to the Babel fees use case, we would like to support more lax exchange offers via intents, e.g. where not all the offered assets have to be traded, and not necessarily at a pre-determined rate. Another interesting use case that may be addressed via the intents framework is light clients. A light client (LC) may not be aware of the current ledger state, and may need a service provider (SP) to construct a transaction for them according to some specification. The LC must be able to check that the transaction meets the specification, but be unable to submit it themselves, thereby avoiding paying a fee to the SP. This can be accomplished with an *intent to spend an output*.

References

1. Algorand Team: Algorand Developer Documentation. https://developer.algorand.org/docs/ (2021)
2. Chakravarty, M.M.T., Chapman, J., MacKenzie, K., Melkonian, O., Müller, J., Jones, M.P., Vinogradova, P., Wadler, P.: Native custom tokens in the extended UTXO model. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation: Applications - 9th International Symposium on Leveraging Applications of Formal Methods, ISoLA 2020, Rhodes, Greece, October 20-30, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12478, pp. 89–111. Springer (2020). https://doi.org/10.1007/978-3-030-61467-6_7, https://doi.org/10.1007/978-3-030-61467-6_7
3. Chakravarty, M.M.T., Chapman, J., MacKenzie, K., Melkonian, O., Müller, J., Jones, M.P., Vinogradova, P., Wadler, P., Zahnentferner, J.: UTXO$_{\mathsf{ma}}$: UTXO with multi-asset support. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation: Applications - 9th International Symposium on Leveraging Applications of Formal Methods, ISoLA 2020, Rhodes, Greece, October 20-30, 2020, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12478, pp. 112–130. Springer (2020). https://doi.org/10.1007/978-3-030-61467-6_8, https://doi.org/10.1007/978-3-030-61467-6_8
4. Chakravarty, M.M.T., Chapman, J., MacKenzie, K., Melkonian, O., Peyton Jones, M., Wadler, P.: The extended UTXO model. In: Bernhard, M., Bracciali, A., Camp, L.J., Matsuo, S., Maurushat, A., Rønne, P.B., Sala, M. (eds.) Financial Cryptography and Data Security - FC 2020 International Workshops, AsiaUSEC, CoDeFi, VOTING, and WTSC, Kota Kinabalu, Malaysia, February 14, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12063, pp. 525–539. Springer (2020). https://doi.org/10.1007/978-3-030-54455-3_37, https://doi.org/10.1007/978-3-030-54455-3_37
5. Chakravarty, Manuel M. T. and Karayannidis, Nikos and Kiayias, Aggelos and Peyton Jones, Michael and Vinogradova, Polina: Babel fees via limited liabilities. In: Applied Cryptography and Network Security: 20th International Conference, ACNS 2022, Rome, Italy, June 20–23, 2022, Proceedings. p. 707–726. Springer-Verlag, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-031-09234-3_35, https://doi.org/10.1007/978-3-031-09234-3_35
6. Christopher Goes, Awa Sun Yin, Adrian Brink: Anoma: a unified architecture for full-stack decentralised applications (2022), https://github.com/anoma/whitepaper/blob/main/whitepaper.pdf
7. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/en/bitcoin-paper (October 2008)
8. Polina Vinogradova and Orestis Melkonian and Philip Wadler and Manuel Chakravarty and Jacco Krijnen and Michael Peyton Jones and James Chapman and Tudor Ferariu : Structured contracts in the EUTxO ledger model (2024), https://fmbc.gitlab.io/2024/files/FMBC2024.pdf