Blockchain : Computation, Shutdown problem, and Governability

Stéphane Caporali,
Caporali Conseil

June 2021

Abstract

This document describes the computational models associated with blockchain and, in a second step, what is for blockchain the shutdown problem. He then asks questions about the link between shutdown problem, computation, and governability in a blockchain project. This document aims to contribute to the implementation of industrial good practices. At the same time, it opens questions to possible areas of research.

1 Introduction : definition of the 51% attack

The process of confirmation of a transaction with blockchain is associated with a proof mechanism, such as bitcoin Proof of Work (PoW). It involves the generation of a hard and useless work to solve a cryptographic proof. This work requires a certain amount of hash rate. According to the site bitcoin.org [1] it is called 51% Attack or Majority Hash Rate Attack the ability of someone controlling a majority of network hash rate to revise transaction history and prevent new transactions from confirming. The blockchain is considered very secure, but the 51% attack is a fundamental and perhaps structural vulnerability of the blockchain.

2 Blockchain and computation

2.1 Existing models of computation

A model of computation describes how an output of a mathematical function is computed given an input. There are many variants of calculation models; three main models are:

- State machine. Lewis, Harry, Papadimitriou & Christos (1998) [2] say that what makes the finite automaton such a restricted model of real computers is the complete absence of memory outside its fixed memory processor.
- Turing machine. Current computers are not Turing machines, in the strict sense, but they use the computational model of the Turing machine. One innovation of the Turing machine lies in the use of memory. A machine is called Turing complete if it can compute all operations computable by a Turing machine.

- Quantum Turing machine.

2.2 Blockchain as combination of two models of computation

What is the model of computation associated with the blockchain ? The diagram below describes the sequence of blocks. To add a new block in the chain, it is necessary to confirm the transaction. Wang et al.(2019) [3] explain that the sequence of blocks represents the blockchain state, and the confirmation of a transaction results in a blockchain state transition. The associated model of computation is that of the replicated infinite state machine. However, the confirmation of the transaction is executed by a computer, it includes the consensus algorithm (to validate the order of events in the chain), the confirmation of the coherence of the transaction, and may confirms execution of one or more smart contracts. The process is usually executed by a language considered to be Turing complete. However, some blockchains are not Turing complete. Bitcoin is not Turing complete, whereas Ethereum claims to be Turing complete. The model of computation associated with the confirmation of the transaction is that of the Turing machine. Malkhi (2018) [4] explains that blockchain is a Byzantine Fault Tolerant (BFT) replicated state machine, in which each state-update is itself a Turing machine with bound resources. The blockchain could be considered to be a combination of two computational models: a replicated infinite state machine, and a Turing machine.

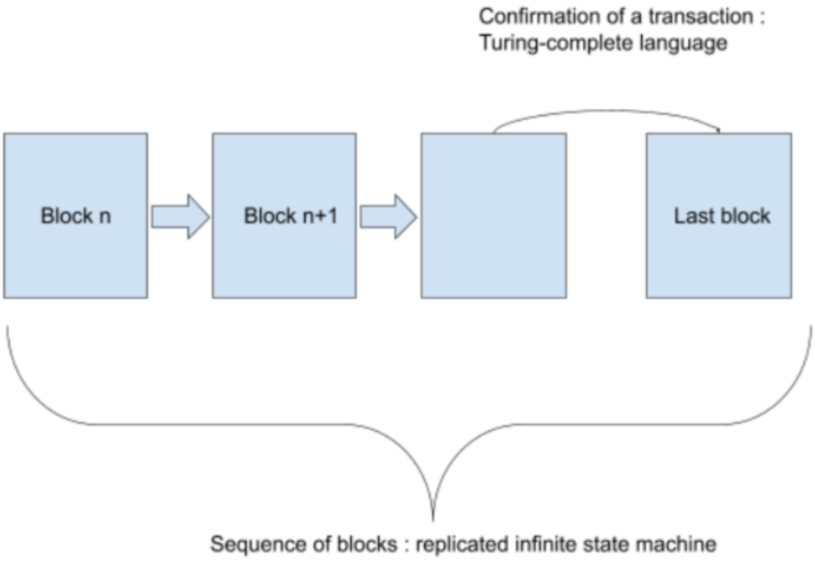
Confirmation of a transaction :
Turing-complete language
Block n
Block n+1
Last block
Sequence of blocks : replicated infinite state machine

Figure 1: adding a new block in the chain

3 Relevance of a computational approach

The possibility exists to open research in the direction of the computation field, and especially the computation complexity, in order to consider blockchain as a model of computation in itself. If a model of computation describes how an output of a mathematical function is computed given an input, the output could be considered as the ending of the chain, rather than just the result of a specific transaction.

4 Description of the shutdown problem

4.1 Definition

Day (2019) [5] writes that we call the shutdown problem how to gracefully end the system's operation at the end of its useful life. Later in the paper, the author writes that in most cases, for most blockchain systems, the shutdown problem is hard or impossible. There is no straightforward way to declare that a history is finished and archived.

4.2 Why does the shutdown problem exist ?

Confirmation of the transaction may involve a smart contract, or algorithms responsible for verifying the consistency of the transaction, depending on the social consensus and the consensus algorithm itself. In computability theory, the halting problem is the problem of determining, from a description of an arbitrary computer program and an input, whether the program will finish running (i.e., halt) or will continue to run forever. Turing machines are well known to be concerned by the halting problem. However, it is a theoretical problem. In practice, computers have a finite amount of memory and will only run for a finite amount of time before being turned off. Similarly, to solve this issue, Ethereum relies on gas. Once the gas is consumed, the execution is halted. *The consensus algorithm* itself that can determine whether the blockchain never ends. As long as the blockchain is used, with miners validating the transaction and a user initiating a transaction with enough energy, the blockchain will continue to confirm transactions without ever ending. There is no central administration or prevention mechanism that can stop the blockchain.

4.3 Possibility of what means a good end

In its paper, Day (2019) [5] describes what means that a blockchain end: the stored information should be in a stable form. Since the system is not continuing to operate, there should not be additional changes to the information. There should not be an ongoing commitment to operating or supporting the old system. However if there is, there should be a sharp reduction in the cost of maintaining the stored information.

4.4 Mitigation strategies

The author goes on to describe practical but limited mitigation strategies:

- Hard fork solution to establish a stable consensus.
- Increase the number of nodes.

He considers the Nakamoto Consensus family of consensus algorithms to be more vulnerable to the shutdown problem.

5 Is shutdown problem a theoretical problem ?

Can we compare the:

- Halting problem for the Turing machine.
- May be Hilbert's tenth problem for the quantum Turing machine, as described by Kieu. (2003) [6]
- and the shutdown problem for blockchain ?

The problems are very different in their origin, but are the similar consequences of an endless process. What importance must we give to the shutdown problem in the theoretical description of what a blockchain is ?

6 Dependence between shutdown problem and 51% attack

Addressing the shutdown problem could help to prevent a 51% attack : it could make it possible to stop an attacked blockchain. The condition is that this is only possible when the attack is detected. The questions opened by this article are:

- What is the cause-and-effect relationship between the shutdown problem and the 51% problem ?
- What are the scenarios where limiting the shutdown problem could limit the effect of the 51% attack ?
- What are the practical difficulties in such a defense strategy ?

7 Classification of consensus algorithms

Consensus algorithms could be classified in classes, for example as we already classify mathematical algorithms in terms of complexity in time or complexity in space. It could be classify consensus algorithms by classes:

- Sensibility to the shutdown problem.
- and by extension by degree of vulnerability about the 51% problem.

8 Governance considerations

When a blockchain project is started, one of the top risk issues is the requirement, or not, to be able to shutdown the blockchain and to terminate the project. There is no question to creating a new mode of governance but of understanding the limits of governance itself. Technical scenarios for terminating a blockchain should be described, before and after a 51% attack, based on experience gained through projects and experiments.

9 Conclusion

The originality of this article is to raise questions and discussions concerning the possible link between 51% attack, blockchain shutdown, and the models of computation. This work is relevant in a normative approach because the purpose of this article is not to provide a comparison between one or another algorithm but to question the technical limits in the use that can be made of an algorithm. In others words, its governability, as the quality of being governable. It can lead to a methodological support for the implementation of blockchain projects.

References

[1] Glossary — bitcoin. https://developer.bitcoin.org/glossary.html. (Accessed on 06/18/2021).

[2] Harry R. Lewis and Christos H. Papadimitriou. *Elements of the Theory of Computation*, volume 29. Association for Computing Machinery, New York, NY, USA, September 1998.

[3] Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen, and Dong In Kim. A survey on consensus mechanisms and mining strategy management in blockchain networks. https://ieeexplore.ieee.org/document/8629877, 2019.

[4] Dahlia Malkhi. Blockchain in the lens of bft. https://www.usenix.org/conference/atc18/presentation/malkhi, 2018.

[5] Mark Stuart Day. The shutdown problem: how does a blockchain system end? https://arxiv.org/abs/1902.07254, 2019.

[6] Tien D Kieu. Computing the non-computable. https://www.tandfonline.com/doi/abs/10.1080/00107510302712, 2003.