

# Creation of a Self-Sustaining Token Economy

Oleksandr Letychevskyi

Department of Digital Automata Theory, Glushkov Institute of Cybernetics, Kyiv, Ukraine

**Correspondence:** [oleksandr.letychevskyi@litsoft.com.ua](mailto:oleksandr.letychevskyi@litsoft.com.ua)

**Received:** 29 December 2021 **Accepted:** 11 February 2022 **Published:** 20 February 2022

## Abstract

Creating a self-governing token economy is a challenge for any project seeking to create a product with a rational token run that provides motivation and profit. It is also important to maintain correct token liquidity after listing on an exchange. Developing an algorithm for tokenomics and manually planning initial coin-offering rounds is an extremely difficult task because the human mind cannot fully understand all the factors and probable scenarios of token behaviour during product operation and trading.

The safest and most effective way of building tokenomics is through formal methods. We offer algebraic modelling based on the theory of interactions between agents and environments. The behaviour of each agent, namely, the investor, product users, and traders on an exchange, and a smart contract, which controls the interaction, are defined by the set of actions presented in behaviour algebra. Symbolic modelling methods allow one to build the best scenarios for circulating tokens and maintain economic equilibrium over an extended period. Automatic proofs of statements in behaviour algebra make it possible to verify properties such as the impossibility of centralisation or the malicious actions of stakeholders who look to obtain illicit benefits.

We have developed an online automated web service called tokenomics constructor that makes it possible to build tokenomics with simulations of possible behavioural scenarios and provides proofs of properties of interest. The tokenomics constructor uses historical stock exchange data to model trade. The system is supported by an algebraic server that leverages algebraic behaviour methods and an automatic proof system.

**Keywords:** *token economy, behaviour algebra, formal methods, symbolic modelling*

**JEL Classifications:** *C51, C60*

## 1. Introduction

Today, the world is experiencing a boom in the adoption of blockchain. New projects and services evolve daily, creating a new ecosystem of crypto-related platforms and products that use cryptocurrencies and tokens.

Nevertheless, we are still in the very beginning of a blockchain-ruled world with only a few real benchmarks or examples of long-lasting, self-sustaining token economies. The main reason is that the creation of a self-governing token economy is a difficult task that requires considerable effort, knowledge, and provision.

It is easy to see that most creators, even if they have a profound white paper and seem to have a thoughtful token economy, intuitively build pyramid-like structures with a lack of real token utility and stakeholders' motivation to hold; most importantly, they use their tokens in daily activities within the platform. As a result, most projects never reach a system balance and become too dependent on the speculation of whale-investors.

Is there a simple and efficient way to help blockchain project creators avoid mistakes and create self-sustainable token economies right at the stage of white-paper and minimum viable product development?

The key to this is math and formal methods. Modelling is a mandatory activity that should be provided during tokenomics (token economy) creation. Algebraic modelling can help build and prove the economic equilibrium and analyse different undesirable properties, such as centralisation, or prevent malicious actions of unscrupulous stakeholders.

This study considers the technology and program systems that can be used in token economy creation.

## 2. What is token economy in the scope of formal methods?

At first glance, the scenarios that occur in tokenomics are like a game between participants who want to gain their own preferences. On the one hand, it is a sales game in which participants hold, sell, or buy tokens according to changes in the liquidity or marketing information. On the other hand, tokenomics is built according to some service, which is built based on blockchain platforms. In many projects, the token is seen as a means of motivation, deliberately created to increase its effectiveness. Thus, in education systems, the token is used to encourage students to study diligently to reduce the fee for educational services. In other services, tokens are paid for reputation, number of services consumed, or installation of equipment. In general, the tokenomics of such services is built

so that the more efficiently the consumers use the service and are rewarded with tokens, the more profit goes to the creators of the service.

For the most part, the ideas of such a game are offered by investors when compiling “white pages” and more often the rules of the game are more attractive at first glance and the business appears clear and transparent. However, even with very limited rules and factors that affect the token life cycle, the human brain does not always see all scenarios and always wants to believe that development will follow the best scenario. Even if calculations are made, they cannot cover the full range of possible events and only present some scenarios, generally the best ones.

Suppose we know the first steps in investing and the exact number of tokens in the exchange. The subsequent stages depend on many factors, and it is impossible to accurately compute the development of events with specific values.

Tokenomics, as a science, includes several fundamental studies on the structure, environment, stakeholder motivation, token life cycle, and more. This is described in the works by Au and Power and Khamisa [1,2]. Some mathematical aspects of tokenomics are presented in the work by Tan [3] to explain key statements.

Experimental data on the functioning of tokenomics in various projects led to the so-called laws of tokenomics [4] developed by Fred Krueger of Stanford University. The laws were based on a large amount of data presented in cryptocurrency market analysis.

The study of self-governing tokenomics has led to the emergence of justifications and research in several areas.

1. Game theory underlies parties’ motivations. If a blockchain-based platform embodies a service, the task of tokenomics encourages buyers and, which, profit sellers. With the help of tokenisation, such a task is considered successful if economic equilibrium is established over a period. This implies a stable change in the basic parameters of tokenomics, such as the token price.

2. Statistical methods underlying the prediction of token behaviour. Studying the probabilities of various events and historical data and using them for machine learning is useful for creating a model for predicting token behaviour.

3. Formal methods, mainly probabilistic methods, are used to analyse the resilience of malicious or dishonest actions, such as fake trading, centralisation, and collusion between stakeholders [5].

All these approaches can be useful, but the prediction accuracy is not always possible, provided that not all factors can be predicted in a token game.

From our viewpoint, it would be appropriate to use methods based on the algebraic approach, where instead of specific values of the parameters of tokenomics, we manipulate the formulas representing the set of possible values of the parameters.

This approach forms the basis of symbolic or algebraic modelling. When building a model, we assumed a set of actions that occur between participants in tokenomics and changed their parameters accordingly. Modelling makes it possible to observe the projected development of events over a certain period. But it must be understood that such a prediction is feasible if the maximum of the most influential factors is considered.

With a certain set of such data, it is possible to conduct a fairly accurate analysis of the properties of tokenomics. Such properties are primarily the tokenomics equilibrium or predicted fluctuations in the token price. In certain projects, stakeholders can influence development and invest heavily in stabilising or destabilising the tokenomics. Then, properties such as the possibility of centralisation or accumulation of manual control in the hands of two or three participants should also be analysed.

The factors that significantly affect the liquidity of a token cannot be determined unambiguously. An example is the price of Bitcoin, which significantly affects projects.

Here, it is possible to consider different scenarios with different possible behaviours of such factors.

Later in this study, we will consider the algebraic approach that we use to analyse and study tokenomics projects and will formalise and investigate one of the use cases. The compilation of tokenomics can be performed on a simplified basis to try understanding its problems. To do this, we developed a program tokenomics constructor, which after entering the parameters builds some analytics with the appropriate charts and will show certain scenarios for developing a tokenomics project. This program is available and can be used [6].

### 3. Theory of agents and environments and the algebra of behaviours in formalising tokenomics

Previous work in which we considered tokenomics [7,8] contains an approach based on the theory of interactions between agents and environments.

In 1997, English scientist David Gilbert and Ukrainian professor Alexander Letichevsky Senior created this theory, which also actively uses the concept of algebra of behaviours.

Environments and agents evolve and exhibit observable behaviour. An agent can be inserted into the environment, interact with other agents, and change the behaviour of the

environment and other agents. The change in the behaviour of an agent is determined by its atomic action.

The basics of this paradigm were set out in the literature [9,10].

Tokenomics determines a set of agents that interact with each other by performing atomic actions. Such agents include project stakeholders, investors, exchanges, and traders. An agent can bring together a group of individuals or represent a system that also changes the environment; for example, such an agent can be a smart contract.

Each agent can have a state determined by its attributes. As the attributes change, so does the agent's state. For example, attributes may be the number of tokens or fiat money from the agent. An agent can have a type defined by either a set of attributes or a common feature. For example, "Investor" may be defined as an agent with a specific number of tokens and fiat money. Agents of this type can be banks, people, or companies. In theory, agents have names.

At first glance, the concept of an agent is very close to the concept of a class in object-oriented languages. The difference is that a class is created to run a particular program and has many features to implement the execution – methods, interfaces, and other features. The formal definition of an agent is intended for the possibilities of algebraic modelling, which differs from the usual execution of a program by representing a set of scenarios of agent behaviour in an environment characterised by algebraic representation, such as formulas in the theories used here.

An example of a formal presentation of the agents' description is presented in the next section, where a specific case study is considered.

The interaction of agents in the environment is determined by some behaviour, which is also represented by formal specifications.

The algebra of behaviours was also developed by Gilbert and Letichevsky Sr. It is a two-sorted algebra over the actions of agents and behaviours. The main operations of algebra are prefixing and nondeterministic choices, which are extended by so-called compositions parallel and sequential.

Examples of behaviour equations include the following:

$$\begin{aligned} B0 &= a1.a2.B1 + a3.B2, \\ B1 &= a4.\Delta, \\ B2 &= \dots \end{aligned}$$

This implies that behaviour  $B0$  can be interpreted as a sequence of actions  $a1$  and  $a2$ , followed by behaviour  $B1$ , or as action  $a3$ , followed by behaviour  $B2$ . Behaviour  $B1$  finishes after action  $a4$ , where  $\Delta$  denotes the successful termination constant.

Tokenomics in behavioural equations is a tree of actions with possible cycles performed by the agents involved. They can act in parallel or sequentially, defining a set of tokenomics scenarios that consist of actions. Because an agent is an entity that has attributes, the set of attribute values determines the state of the agent, and the set of states of the agents determines the state of tokenomics.

The actions of agents themselves consist of a precondition that allows the agent to act in certain states and a postcondition that determines the change of its attributes and, accordingly, the change of state of the agent.

The peculiarity of the action, which is an algebraic entity, is that the state is determined by the formula over the attributes rather than specific values. The behaviour scenario of tokenomics agents may be a sequence of actions that changes the state of the agent represented in some theory, such as inequality in linear arithmetic. The predicates and operations in the formula depend on the type of agent attribute. Let us consider the formal representation of tokenomics in the algebra of behaviours using a specific example.

#### 4. Tokenisation of the Internet of Things as an example of tokenomics

We consider the formalisation of tokenomics as an example of an abstract Internet of Things (IoT) project.

The essence of the project is to create a network of base points that provides data exchange with IoT devices, such as animal trackers, human health devices, car gadgets, and other sensors and mobile gadgets. Such base points are antennas that provide a certain coverage for users.

The following agents function throughout the tokenomics system.

Base point owners earn revenue from the data traffic and the number of users who have joined it. Owners of IoT devices buy devices from their respective vendors and pay for their ability to use the network.

A token operates as a system for online payments and rewards. Accordingly, base point owners receive rewards in tokens, and users pay for traffic or bandwidth services in tokens. The token is used in the system to make a profit and appears in the investments.

To buy a token, it must be registered on the exchange. In modern exchanges, a certain amount of the presented token must be supported by a stable currency, meaning that, in addition to the tokens, a certain amount of fiat money is loaded on the exchange, which forms the price of the token.

In connection with the beginning of the sale and purchase of a token on the stock exchange, a new type of agent is created

that deals with trading, which depends on the increase or decrease in liquidity. These actions affect the price of the token and, accordingly, tokenomics as a whole. The exchange is also an agent in the tokenomics system.

The main agent is the one who represents the team of developers—entrepreneurs. An agent is also a product service or smart contract that defines the actions of a blockchain service. They collect tokens for transactions and direct some of the profits from the transactions to the team.

Finally, there is an agent that represents investors and can act as a trader, realising his or her tokens, which he or she collected in the first round of investment.

Figure 1 shows the relationship between the agents or the movement of the token between them.

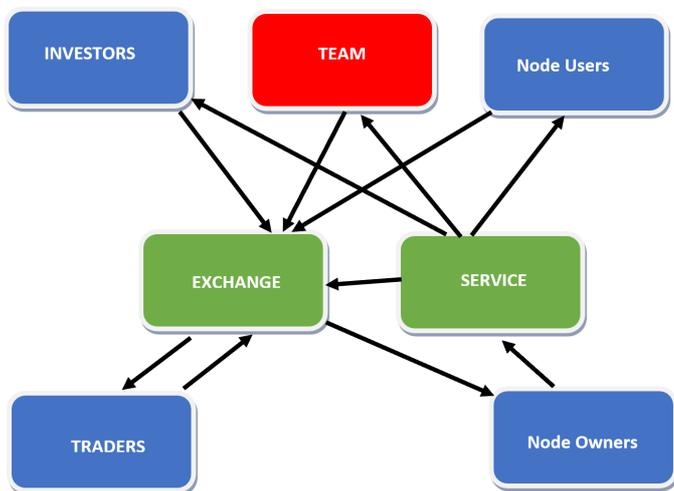


Figure 1. The token movement between agents

Next, we define a formal description of the tokenomics model in terms of the behavioural equations and actions of agents. Let us define the type of agent as the stakeholder of this token system, which has two attributes: *token*, the number of tokens, and *fiat*, the amount of fiat money. Let us mark the corresponding names of agents: *NodeOwners*, *NodeUsers*, *Team*, *Exchange*, *Traders*, *Investors*, and *Service Product*. We use the notation of the theory of agents and environments and determine access to their attributes by writing *AgentName.token*, such as *Investors.token* and *Exchange.fiat*.

The tokenomics of service comprises two parts. The first is investment. During the investment rounds, some of the tokens are sold privately, and afterwards, an open sale begins for everyone.

The second part concerns the service. Its launch starts with a certain token value on the stock exchange and certain results of the marketing group, which attracts network users to the service and owners of base points that expand the network.

Behavioural equation can be written as a sequential composition of two behaviours.

$$B_0 = INVEST; SERVICE$$

Behaviour *INVEST* is a cycle of several rounds when tokens are sold and funds are raised.

$$INVEST = round.INVEST + empty$$

The round action determines the flow of tokens to investors according to the period determined by the allowed time predicate *time (Round, X)*. The general type of action is as follows:

$$round = Time (round, X) > SmartContract.token = SmartContract.token + TokenBudget (X);$$

$$Investors.token = Investors.token - TokenBudget (X)$$

The semantics of the record are as follows: if the predicate *Time* is true, namely, the time of the round has come or the time is equal to the corresponding month; for example, *Round = X*, then the service product or smart contract must perform the planned budget actions, namely, to add tokens to investors from the smart contract.

If the service has already started during the investment, then we use a parallel composition of two behaviours, and the beginning of the service will be determined by the allowing predicate.

$$B_0 = INVEST || SERVICE$$

Here, if the token was registered during the investment, the behaviour of the exchange was also combined in parallel with it and the service.

$$B_0 = INVEST || SERVICE || EXCHANGE$$

*SERVICE* behaviour is the periodic execution of monthly purchases of new basic antennas and payment of rewards to their owners, the implementation of the plan for the use of traffic, and the work of the marketing team responsible for increasing the number of service users.

$$SERVICE =$$

$$nextMonth.addNewOwners.rewardToken.addNewUsers.deviceUsersToken.SERVICE$$

Here, it was convenient to consider a discrete model with respect to monthly intervals. We disregard the fact that the token price can change many times during the month. Therefore, a more accurate model was built when a one-day interval was used. The number of users involved per day also affects the course of events; however, this model is proposed as a general approach that can be transformed into different periods. This approach can also be used to predict the

behaviour of the token, allowing an even distribution of values throughout the location, which allows for an accurate investigation of the properties of tokenomics.

Let us now consider the actions separated by the prefixing operation involved in the *SERVICE* behaviour.

*addNewOwners* – Adds the number of new antenna owners in the postcondition based on various factors. This can be historical data from open blockchain resources, which can be combined with marketing effectiveness, depending on the amount of money invested.

*rewardToken* – Distributes tokens for antenna owners from the smart contract balance.

*addNewUsers* – Adds new users to IoT devices. This action uses a marketing effectiveness feature, combined with historical data.

*deviceUsersTokens* – This action deals with the purchase of tokens on the exchange by users, payment for the use of the network, and the corresponding change in the price of the token, as the tokens were purchased. Some of the tokens used for using the network are considered profit for the team.

We disregard the description of actions, as it may contain factors that we have not described; however, we have not considered critical actions or the inability to perform them. Such situations can arise when the balance of a smart contract does not have sufficient tokens for payment, when there are no funds for marketing, or when the exchange lacks tokens for sale. All such actions must also be present in the behavioural equations. Given our understanding of the model and method, we abstract from such things.

When formalising human behaviour, questions arise, such as in what sequence and at what time will this or that human action occur? Behavioural algebra refers to the ability to synchronise actions and perform them in parallel or in a certain sequence. In the process of modelling actions that correspond to a certain behaviour, all possible scenarios of human behaviour are considered up to the equivalence. An important factor in the accuracy of modelling is the choice of the minimum period that corresponds to the duration of action.

The actions themselves correspond to all possible operations in the tokenomics environment that are allowed by a smart contract or system. Some actions, such as user growth, depend on the external environment, namely, the success of marketing rather than tokenomics. When formalising such actions, historical data and different marketing scenarios should be considered. However, note that the algebraic method allows one to operate with a set of scenarios, including external factors.

*EXCHANGE* behaviour can significantly affect tokenomics scenarios. It includes actions that determine the behaviour of

traders, namely, holding, selling, or buying tokens. This behaviour can be predicted using different algorithms combined with historical data on the purchase and sale of tokens on different exchanges. Of course, various factors affect the increase or decrease in stock exchange activity. Therefore, to assess the forecast of tokenomics, it is necessary to take different options for the behaviour of the exchange based on the historical data of token sales. The more diverse the behaviours we consider, the more accurately we can determine the properties of tokenomics. Appropriate functions of the desire to hold, sell, or buy tokens were built for modelling.

One of the levers of influence on tokenomics is emissions, such as throwing fiat money or tokens on the exchange to stabilise the token price. This action can be considered and included in the formalisation of the *SERVICE* behaviour.

Thus, having a formal description of the parallel composition of behaviours together with actions, we can analyse the properties of tokenomics, in particular, constructing charts of different behaviours or algebraic charts for arithmetic data.

Property analysis was performed using modelling, both concrete (or simulation) and symbolic. If we determine all the initial conditions and criteria of behaviours on the stock exchange and the marketing plan, it is possible to build a chart of the token price as an indicator of equilibrium. However, note that most of the data are subjective, and there are significant doubts about their accuracy. An example token chart of the simulation is shown in Figure 2.



Figure 2. Token price diagram

The graph shows that the token fluctuations are stable; however, if adjustments are made and the function of the desire to sell tokens is increased, the price will fall quickly, and the equilibrium may not be compensated by increasing the number of users.

Here, to eliminate subjective data, we considered possible ranges of values instead of specific values. We used algebraic modelling, which works with intervals or more complex constructions. Here, instead of a chart, we obtain a set of formulas for each month, which represents the state of the tokenomics environment and depends on the number of tokens for each agent –  $x_1, x_2, \dots$  and the price of token  $t$ .

$$S_1(t, x_1, x_2, \dots), S_2(t, x_1, x_2, \dots), \dots, S_n(t, x_1, x_2, \dots), \dots$$

The equilibrium property of tokenomics can be expressed by an allowable interval for the token price, which is expressed by the arithmetic inequality. Note that the equilibrium formula can be more complex and contains other factors, such as the presence of sufficient tokens on the smart contract or the growing profit of the team.

In general, if we denote equilibrium formula as  $R(t, x_1, x_2, \dots)$ , then the conjunction of this formula and the state of tokenomics in month  $n$  would be

$$R(t, x_1, x_2, \dots) \wedge S_n(t, x_1, x_2, \dots)$$

We can prove the satisfiability of the formula. This proves that there are values at which the formula is true. Next, by performing inverse algebraic modelling, we can determine the intervals for the initial values that will lead to equilibrium. From these intervals, one can select the desired values to implement in tokenomics when starting the service or rounds.

Thus, the desired properties can be determined and the initial values of the attributes can be calculated. If most factors are determined by appropriate intervals or formulas, the prediction of tokenomics can be quite accurate.

### 5. Tokenomics constructor

A tokenomics constructor was developed for tokenomics creators who are beginners in this activity. This constructor assumes that the creator can enter specific data planned for the project – the total number of tokens, rewards for users, and marketing function – and define the functions of the exchange. Figure 4 illustrates a fragment of the tokenomics constructor site.

After entering the desired data, the algebraic server can be queried for the simulation. Because the data are specific, one can build a chart of the distribution of tokens by agents and calculate the price of the token depending on the time, such as a period of a month. Figure 3 provides an example of the distribution of tokens for different agents throughout the project and shows the relative balance. Other distribution charts and special properties can be specified during the query to the algebraic server.

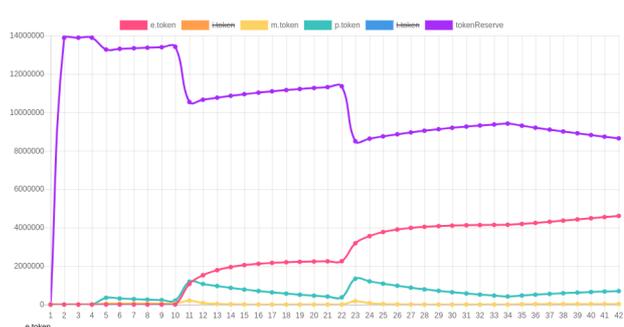


Figure 3. Token distribution between agents

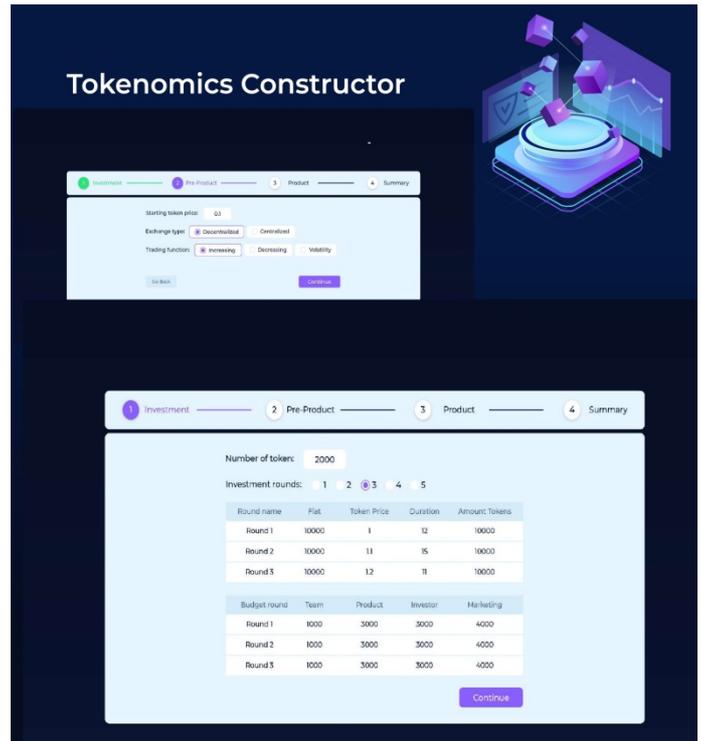


Figure 4. Tokenomics constructor site fragment

For more experienced users, it is possible to use the creator of insertion models in terms of the algebra of behaviours, agents, and environments. This system provides an interface for creating appropriate agents and environments and modelling the entire system with the output of graphs and corresponding scenarios in the form of message sequence chart traces. These traces contain a set of actions with the corresponding states of the agents, which, under certain initial conditions, lead to the violation of a certain property, for example, exceeding a certain threshold in the tokens for a particular agent. The model creator window is illustrated in Figure 5.

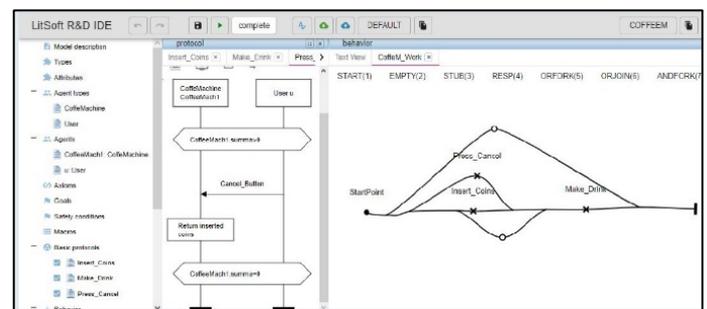


Figure 5. Model creator tools

Finally, for studies with more serious properties – equilibrium, which is determined by certain parameters; centralisation; resistance to possible attacks of fraud to gain illicit profits – the algebraic server or algebraic virtual machine is used. It contains a set of formal methods that work within the algebra of behaviours.

## 6. Conclusion

This technology provides an opportunity to create self-sustaining tokenomics for projects that use tokenisation and blockchain technology for the deployment of network services in various subject areas. At first glance, the formalisation of tokenomics is a time-consuming process that requires serious mathematical education. However, the use of a tokenomics constructor is the first step in assessing the problems that may arise in the project. Also the original project created in the tokenomics constructor can be downloaded and extended in the model creator. The specification of a behavioural algebraic language slightly differs from the specifications of common programming languages, but any mid-level programmer can program a system.

Thus, the creation of formal specifications of tokenomics and modelling provides an opportunity to observe the problems in the project and configure the actions of agents or smart contracts to maintain equilibrium or eliminate undesirable properties. On the other hand, the quality of system evaluation depends on the accuracy of formalisation and the completeness of the collection of all factors influencing tokenomics. This may depend on the interaction of experts, economists, and those experienced in blockchain projects with programmers who implement the tokenomics model. Additionally, there are unforeseen, unexpected, or force majeure circumstances that can significantly change the behaviour of the entire system, including natural disasters. This means that cryptocurrencies will experience significant fluctuations, which will be difficult to predict regardless of how perfect the service is.

Thus, it is inexpedient to completely ignore the modelling of tokenomics, and arguably doing so could be harmful. The human brain cannot assess all possible scenarios and abnormalities. Therefore, formal methods can be used to create more predictable and self-governing tokenomics.

## References

- [1] S. Au and T. Power, "Tokenomics. The Crypto Shift of Blockchains, ICOs, and tokens," *Paket Publishing*, 2018, 454 p.
- [2] A. Khamisa, "Token Economics." In *The Emerald Handbook of Blockchain for Business*. Emerald Publishing Limited. 2021
- [3] L. J. Tan, *Economics and Math of Token Engineering and DeFi: Fundamentals of Token Economics eBook: Kindle Edition*, Amazon.
- [4] F. Kruger, "How to Think about Tokenomics?," <https://medium.com/workcoin/how-to-think-about-tokenomics-b3da509444e5>
- [5] A. Bracciali, D. Grossi, and R. de Haan, "Decentralization in Open Quorum Systems: Limitative Results for Ripple and Stellar," *2nd International Conference on Blockchain Economics, Security and Protocols Tokenomics 2020, October 26–27, 2020, Toulouse, France*, pp.5:1–5:20
- [6] *Tokenomics Constructor site: <https://tokenomics.expert/>*
- [7] O. Letychevskiy, V. Peschanenko, M. Poltoratskiy, and Y. Tarasich, "Our Approach to Formal Verification of Token Economy Models," *Conference paper, ICTERI: International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications*, Springer, Cham, 2019, pp. 348–363.
- [8] O. Letychevskiy, V. Peschanenko, M. Poltoratskiy, P. Kovalenko, S. Mogyloko, and V. Radchenko, "Formal Verification of Token Economy Models," *IEEE Explore Digital Library, IEEE International Conference on Blockchain and Cryptocurrency (ICBC) 2019, Seoul, South Korea*.
- [9] D. Gilbert, A. Letychevskiy, "A model for interaction of agents and environments," in D. Bert, and C. Choppy, eds., *Recent Trends in Algebraic Development Techniques*, LNCS 1827, Springer-Verlag, Cham, Switzerland, 1999.
- [10] A. Letychevskiy, "Algebra of behavior transformations and its applications," in V. B. Kudryavtsev and I. G. Rosenberg, eds., *Structural Theory of Automata, Semigroups, and Universal Algebra*, NATO Science Series II. Mathematics, Physics and Chemistry – vol. 207, pp. 241–272, Springer, 2005

---

### Competing Interests

The author declares no conflict of interest.

### Ethical approval

Not applicable.

### Author's contribution

This manuscript was written by the main author responsible for writing the manuscript, collecting data, and proof reading.

### Funding

This study was not funded.

### Acknowledgements

Not applicable.

---