MathCoin: A Blockchain Proposal that Helps Verify Mathematical Theorems In Public

Borching Su

***Abstract*—A public blockchain is proposed in an attempt to enable the coin holders to participate in verifying mathematical theorems for public access. Incentives are designed to encourage any party to contribute their knowledge by buying tokens of mathematical propositions that they believe are true. The proposed blockchain is a platform for people to exchange their belief in mathematical propositions. An implementation of this blockchain proposal, once established, will provide the general public with an easy and instant access to reliable knowledge without having to read difficult proofs or having to blindly trust a small number of experts. Conversely, experts from various fields may find it much easier for making their work appreciated by more people, leading to a better impact. According to the incentive inherently provided by the blockchain, they can even earn significantly if they do prove some theorems that were not previously known by the blockchain. Foundations who are interested in the validity of a particular proposition not yet explicitly recorded on the blockchain can donate a fund, which will distribute to experts who contribute positive efforts toward solving the specified problems. Only the people who erroneously create or buy tokens of a proposition that is eventually proven false will lose money. A reference design of the proposed blockchain that attempts to achieve the above-mentioned goal is described and reasoned.**

***Index Terms*—blockchain, cryptocurrencies, MathCoin, mathematical theorems, mathematical logic, Bitcoin, Ethereum, smart contracts, formal verification, Zermelo-Fraenkel set theory, functional programming.**

I. Introduction

Mathematical proofs are essential to any research work and serve as foundations of science. Once proven, any mathematical proposition can be said to be "known" to the human kind, since any qualified professional can read the proof and check the validity of the proposition. However, there are usually cases that the validity of a given proposition is not so "known" to human due to lack of qualified professionals and their time. For example, Andrew Wiles had a proof of Fermat's Last Theorem in 1993, but an error was found in the proof. So, he corrected the error and published the proof again in 1995 [1]. However, as the proof is extraordinarily long, except for the real experts in the field, a relatively small number of people actually have the time to go through the whole proof and check every detail. Therefore, the general public's belief that Fermat's Last Theorem is true is not directly based on the proof that Wiles published; rather, it is based on the trust of the experts in the field, or, based on the belief that "some experts have shown that, according to my math teacher, or according to the news reports, etc." One may argue that this is no big deal, since usual people do not really care, and the validity of Fermat's Last Theorem does not affect their lives anyway. However, many new theorems with similar difficulties in proof continue to arise, and some of them will be critical in determining the outcomes of some applications. For a business manager who needs to make a timely important decision that relies on the validity of a theorem, she may choose to hire an expert she trusts. Then, the quality of the decision will pretty much rely on whether the expert is able to give a reliable answer within a time frame given by the manager according to the requirements of her business. If the manager happens not to be able to find competent experts within a short time, or if the experts do not have sufficient time to give a confident answer, the manager will be forced to make a pre-mature decision using the partial knowledge she can get, even though the fact is already said to be "known" by some academic papers.

If the problem of interest can be found as a theorem with proofs in a given research article, things could be easier for the manager. If the manager is careful, she may choose to spend some experts' efforts in double-checking the result (with some more cost in time and budget). An even easier choice may be to trust whatever has been proven in the paper without reading the proofs. Ideally, mathematical proofs are reviewed during the peer review process so that after a paper is published, the proofs written on the paper are supposed to be correct. However, there is essentially no *incentive* for reviewers to find that a proof is wrong, especially when the proof is complicated. If a "proof outline" or a "proof sketch" look reasonable, most reviewers could just accept it because they found nothing wrong in the manuscript. The fact that they found nothing wrong (before the review deadline) does not mean that they endorse the proof. They could find something wrong sometime later (given some incentives in any form). If not (due to lack of incentive), they may just leave the bug there for years.

Recent advances in automated theorem proving (ATP) may provide a way for people to check the validity of theorems without blindly trusting published research results. Many of existing tools, such as Coq [2], Isabelle [3], etc., however, require the users to select heuristics in order to complete the proof. Therefore, these tools are considered proof assistants and are not truly automated yet. It still requires experts to interact with the tools to obtain reliable results. Indeed, many recent research results of mathematical Theorems [4] have been published using these tools. Besides human interactions, machine learning methods are also studied to choose good heuristic [5]. While ATP tools are handy to researchers to find

B. Su is affiliated with the Department of Electrical and Engineering and the Graduate Institute of Communication, National Taiwan University, Taipei 10617, Taiwan (R.O.C.).

a proof of a theorem, it is still not easy to convince non-specialists to trust the correctness of a theorem until they are suffciently familiar with the ATP tool.

Formal verification using ATP gains more and more research attention recently due to security issues in applications regarding cryptocurrencies and smart contracts [6] as well as time-critical and life-threatening applications. Instead of using just a finite set of random test inputs to demonstrate that the code *usually* does what it is expected to do, people would like to see a proof that the code *always* does what it is expected to do, in a hope to avoid serious problems, such as the DAO attack [7], to happen again. However, as mentioned earlier, it is still not easy to convince non-specialists about the correctness of the results given by an ATP tool if they are not sufficiently familiar with the tool. Even worse is that, the general public may have to first trust the correctness of the automated theorem prover used in the results, which require another formal verification of the computer software itself. The demand of making formal verification (a form of mathematical theorems) acceptable to general public in a reliable and instant manner, therefore, will become more and more important.

In this article, we propose to use the blockchain technology to solve the problem of gaining the trust from the general public about the correctness of mathematical theorems proven by experts, advanced ATP software, artificial intelligence that controls the software, and even a combination thereof. The public blockchain establishes a game that is accessible to everyone that is connected to the Internet. The game's rules are designed to reward those who contribute correct knowledge to the public and penalize only those who make erroneous claims on-chain. More specifically, all unproven propositions recorded on the blockchain are given a price between 0 and 1, with which all participants buy tokens to show their confidence on the truthfulness of the target proposition and expect to earn if the proposition is eventually proven true. This idea has been used in many applications in the prediction market [8] recently (e.g., [9] etc.) A major difference is that events to be predicted in a prediction market application usually have a definite date on which the outcome will be clear to everyone, whereas an unproven proposition on a MathCoin blockchain usually does not expect a finalized day (especially for difficult open questions). In addition, token prices of an unproven proposition are governed by a deterministic predictably determined by the amount of purchased tokens. Further, the outcome of a prediction market application may require witnesses to report when a dispute occurs. A proven proposition on the proposed blockchain is shown by mathematical logic and does not need further human intervention.

Since the blockchain is public information and essentially immutable by any party, as time grows, the entire human society will gradually obtain an encyclopedia of reliable mathematical theorems that are important in diverse fields selected through accumulation of wisdom and efforts from experts in various fields.

In this article, we will describe a reference design of a blockchain that attempts to achieve the aforementioned goal. Before the actual launch of any MathCoin blockchain, implementation details shall be carefully chosen, and the software should go through rigorous formal verification. We also elaborate several issues that should be carefully considered in a real implementation for the reference of any team who is interested in realizing the idea proposed here.

The rest of the article is organized as follows. Section II describes basic elements of the proposed MathCoin blockchain, including the block structure and basic functions. Section III describes the pricing rules for unproven propositions that are central to the proposal, followed by some illustrative examples. Section IV gives a list of practical implementation issues that have not been fully analyzed in this article but shall be seriously considered in a real implementataion. Section V gives the conclusion and future aspects.

II. Elements of a MathCoin Blockchain

A. Overview

A MathCoin network is designed to serve as a state-of-the-art knowledge base capable of verifying, upon request, any mathematical propositions that have been proven by human instantly. Anyone who believe she knows something more than what the network is covering can contribute that piece of knowledge to the network and get rewarded appropriately.

In the following, we describe the basic components and structures of a MathCoin blockchain. Figure 1 depicts an overview of the proposed blockchain. Each block starts with the hash of the previous block, followed by a number of transactions, and a nonce which makes the hash of the current block to satisfy a difficulty constraint, just like what Bitcoin [10] and many proof-of-work (POW) based blockchains do. The main difference of a MathCoin blockchain is that besides normal transactions, there are three other different types of transactions in each block, namely, the *token transactions*, the *new definition transactions*, and the *new proposition transactions*. Furthermore, similar to Bitcoin, in each block there is one special transaction, called the *coinbase transaction*, which generates new MathCoins to reward the miner who finds the block. The main difference in a MathCoin blockchain is that only a portion of the newly generated MathCoins goes to the miner; the remaining part will flow to the publicly-held reservoir. The ratio of the part flowing to the public fund to the total amount of newly generated MathCoins in creation of a block is called the *miner's tax rate*. The main purpose of establishing such a public fund is to reward MathCoin users who would contribute knowledge to the chain later. It is important to note

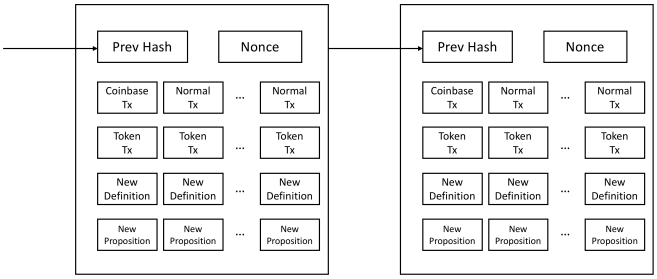

Fig. 1. Illustration of two blocks in a typical MathCoin blockchain.

that, unlike a human-based government which may be subject to corruption, a public fund held by the blockchain obeys a deterministic rule upon which all participating nodes agree when it is using the public MathCoins to reward knowledge contributors. The rewarding process, along with any other usages of the public fund, is therefore completely predictable and transparent.

Just like many other cryptocurrencies, there are multiple user-held addresses that can store MathCoins, and users can transfer their MathCoins to another user address by signing normal transactions using their private keys. Any of the three other types of MathCoin introduced above, however, is designed to be a transaction between a user and the network.

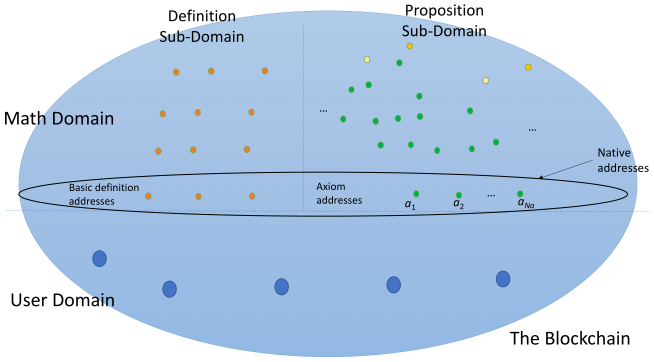

Fig. 2. Address domains and sub-domains of a MathCoin blockchain

Figure 2 illustrates the major types of addresses in a MathCoin blockchain. All MathCoin addresses are categorized into the *user domain* and the *math domain*. A user-domain address is held by a user with a private key. A math-domain address is conceptually similar to a contract address in a blockchain that supports smart contracts (e.g., Ethereum [11], Cardano, etc.). It can be either a *definition address* or a *proposition address*. So, the math domain is further divided into the *definition sub-domain* and the *proposition sub-domain*. In the math domain, there are *native addresses* that are present in the very beginning of a MathCoin blockchain, namely, the *basic definition addresses* and the *axiom addresses*, residing in the definition sub-domain and the proposition sub-domain, respectively. The rest of the math-domain addresses are created later by MathCoin users, using the earlier introduced new definition transactions and the new proposition transactions, respectively. Some of the user-created math-domain addresses issue tokens that can be held by users, in return for the MathCoins transferred to the math domain during the address creation. Finally, users can purchase additional tokens of math-domain addresses by signing a token transaction introduced earlier. A token transaction can also be used to sell some tokens back to the network and get MathCoins back to the user. [1]

[1]Whether tokens of math-domain addresses can be transferred to other users is left as a design issue. The author advocates that such a transfer is not necessary, making the network be the only possible party to trade math-domain tokens with.

Notation	Number of variables	Output	Remarks
$\varnothing$	0	set	The empty set
$\in \_\ \_$	2	proposition	Membership
$= \_\ \_$	2	proposition	Equality
$\forall \_\ \_$	2	proposition	Universal quantifier
$\exists \_\ \_$	2	proposition	Existential quantifier
$\neg \_$	1	proposition	Logical negation
$\rightarrow \_\ \_$	2	proposition	Logical implication
$\{\} \_$	1	set	set operation
$\cup \_$	1	set	Union set
$\{\} \_\ \_$	2	set	set builder

TABLE I
SUGGESTED LIST OF BASIC DEFINITIONS FOR A MATHCOIN BLOCKCHAIN

B. Native Addresses and Choices of Axiomatic Systems

In the beginning of a MathCoin blockchain, there is a number of pre-defined math-domain addresses, collectively called the native addresses. Native addresses include the basic definition addresses in the definition sub-domain, and the axiom addresses in the proposition sub-domain.

The choice of the set of native addresses is an implementation issue, and is usually based on an established foundation of mathematics. For the sake of simplicity, the following presentations adopt the commonly accepted Zermelo-Fraenkel set theory with axiom of choice (ZFC) [12] in our explanation. The choice in this article for the explanation purpose shall not exclude any team from choosing a different foundation, or a different set of native addresses. In fact, it is possible that two or more MathCoin blockchains with different choices of native addresses could exist simultaneously. In this case, the value of a particular MathCoin blockchain may be highly associated with the axiomatic system based on which it was built.

Based on the ZFC system, the following usual notations shall be included in the basic definition addresses:

1) Membership "$\in$"
2) Equality "$=$"
3) The empty set "$\varnothing$"
4) Quantifiers $\forall, \exists, \exists!$.
5) Logical connectives $\wedge, \vee, \neg, \rightarrow$.
6) The set operation "$\{\cdot\}$."
7) The union set operation "$\cup$."

Detailed interactions between these definitions will be elaborated in the subsequent subsections.

C. Definition Addresses and New Definition Transactions

Every definition address, including all the basic definition addresses, is either a constant set or a function that takes one or more variables. If it is a function that takes one variable, then the output shall be either a constant set or a proposition. If it is a function that takes n variables where $n \geq 2$, then by applying the function on one variable will produce a function that takes $n-1$ variables.

Table I presents a list of possibly the minimal set of basic definitions that a ZFC-based MathCoin blockchain needs before launching. We adopt the common notations used in functional programming languages by putting the operands of every binary operation after the operation even if is not

normally written this way. For example, the common membership notation $a \in A$ is expressed as "$\in A\ a$". Applying a binary function on a single operand will result in another unary function. So, "$\in A$ _" can be defined by combining the binary function $\in$ and a set A to form a unary function. The set builder "$\{\}$ _ _" may take two operands, the first being a set (say, X), and second being a propositional function (say, $P(\cdot)$), to form the set $\{x \in X \mid P(x)\}$. The "x" commonly written in a mathematical expression is not required. Similar things happen in, for example, the universal quantifier: if "$\forall$ _ _" takes a set X and a propositional function $P(\cdot)$ as operands, it results in the proposition normally expressed as "$\forall x \in X, P(x)$."

Notice that some more common notations are not listed in Table I, for example the logical connectives $\wedge$ and $\vee$, and the set intersection $\cap$. This is because these common operations can be defined as new definition addresses after a MathCoin blockchain is launched by combining basic definitions listed in Table I. For example, all the logical connectives, including $\wedge$ and $\vee$, can be defined by a number of combinations of $\neg$ and $\rightarrow$. The set intersection can be defined by combining the membership operation and the set builder[2][3]. It is up to a development team's choice to include these common notations in the basic definitions or not.

A larger set of basic definition addresses may include the set of natural numbers $\mathbb{N}$ along with the succeeding $(+1)$ and additive $(+)$ operations. More complicated, but still very common number sets, including $\mathbb{Z}, \mathbb{Q}$, or even $\mathbb{R}$ and $\mathbb{C}$, can also be considered in the basic definition addresses. But one can imagine that if a complicated set like $\mathbb{R}$ is included, the initial implementation can be very large. Nevertheless, existing libraries of sophisticated automated proof assistants can be a good source of reference if a team chooses to include these in the initial setup.

After a MathCoin blockchain is launched, users can create new definition addresses by combining existing definitions appropriately. To network charges the creator of any new definition address a fee that is proportional to the length of the description of the new definition. It may seem to a user that creating a definition is not beneficial at first. But if a user finds that she will use a definition multiple times in some propositions she is going to create later, such kind of fee would be worthy. From the network's perspective, the charging for a newly created definition is to prevent users from creating numerous lengthy definitions that may not be very useful. Users will then be guided to create definitions that are truly useful. Other users can then re-use existing definitions with no cost.

A valid new definition transaction should include the following information:

1) The description of the definition (building from existing definitions).
2) (Optional) The name of the definition.
3) The fee (calculated from the length of the description plus the name).
4) (Optional) Additional fee to the miner.

A human-readable text may be given as the name of the newly created definition. For example, one may define a set of prime numbers with a label "Prime" or "The set of prime numbers." However, such a text is not necessarily required to be recorded on-chain. Indeed, since the name of a definition is irrelevant to all subsequent logical operations it is going to involve with. Since a fee can occur in proportion to the length of the name, users are therefore discouraged to include the name on-chain in order to save their MathCoins. However, users are encouraged to give the name of a definition off-chain. In fact, people around the world shall be able to use their native languages to describe a mathematical object defined on a MathCoin blockchain, for example, when teaching children elementary mathematics in a non-English speaking society. There is absolutely no need to learn English before one can learn math well. The definition sub-domain of a MathCoin blockchain can become a universal, permanent, and language-independent reference for math teachers around the world.

D. Proposition Addresses and New Proposition Transactions

In the proposition sub-domain, there are axiom addresses and user-created proposition addresses. The axiom addresses are included before the blockchain is launched. For example, if the ZFC system is selected as the foundation of a MathCoin blockchain, then there can be nine axiom addresses. Every proposition address at any moment has a price that is publicly calculable and known. While user-created proposition addresses have varying prices, the axiom addresses always have a fixed price, (i.e., 1).

Users can create two types of new proposition addresses. The first type is to create the new proposition with a proof that it is a logical consequence of existing proven addresses. It is therefore a proven proposition address by creation. In the very beginning, a new proposition created this way can only be direct consequences of the few axiom addresses. But as time goes by, when the number of proven addresses increases, many more new proven addresses can be created this way (i.e., as a logical consequence of other proven propositions).

A new proposition address of the second type is created only with the description and without a proof. The truthfulness of such a newly created proposition is therefore uncertain to the public at the time of creation. Besides, if the newly created proposition is the combination of a finite number of logical connectives and existing proposition addresses, it is required that at least one of existing proposition addresses is unproven, such that the expression remains unproven in public. Otherwise, it would have reduced to a first-type transaction. The creator of a second-type proposition should deposit an amount of MathCoins, in additional to all fees, as a sign to the public about her confidence level on the proposition being true. She will not lose the deposit as long as the proposition is not eventually proven false. A price of a value between 0 and 1 is assigned to such an unproven proposition, and the proposition creator will get a certain amount of tokens of the proposition. Other users may purchase additional tokens from an existing unproven proposition. The price varies according

[2] $A \cap B = \{x \in A | x \in B\}$.

[3] One of the two quantifiers may also be removed from the suggested list of basic definitions

to token sale activities and is calculated according to a set of deterministic formulas that all parties in the network know. In general, the price shall increase when new tokens are issued and sold to users; and decrease when existing tokens are sold back to the network and vanished. In addition, users can also choose to purchase the tokens of the negation of any unproven proposition at a price that is complementary to that of the original proposition. When tokens of the negation proposition are issued and sold to users, the price of the original proposition decreases, and vice versa. The prices of the original proposition and the negation proposition always add up to 1. More details on the pricing of unproven proposition addresses will be described in the next section.

A valid new proposition transaction of the first type should include the following information:

1) The description of the proposition (building from existing definition addresses and proposition addresses).
2) A proof that the proposition is a logical consequence of existing proven propositions (building from existing definition address and proven propositions).
3) (Optional) The name of the theorem.
4) The fee (calculated from the sum of lengths of the proposition description, the proof, and the name).
5) (Optional) Additional fee to the miner.

A valid new proposition transaction of the second type should include the following information:

1) The description of the proposition (building from existing definition addresses and proposition addresses).
2) (Optional) The name of the conjecture.
3) The fee (calculated from the length of the proposition description plus that of the name).
4) (Optional) Initial deposits for getting the tokens.
5) (Optional) Additional fee to the miner.
6) (Optional) Additional fee to the network (public fund).

It is observed that the first-type new proposition transactions are very similar to new definition transactions in the way that users will only be charged fee and get nothing else. The name of the theorem, again, is not important and can be optionally left blank to save the fee. In addition, the longer the proof, the more fee is charged by the network. So, users are usually not encouraged to sign the first-type transaction (unless under some special circumstances, to be explained later). Instead, in the second-type transaction, not only is that a proof is not required, but the creator can acquire an amount of initial tokens issued by the newly created proposition, with a price usually lower than other users who purchase tokens of the same proposition later.

A proposition that was created via the second type is by design unproven to the public. It may later turn into a proven one if the following event occurs. Suppose some user creates a first-type proposition p in the following special form:

$$p_1 \to p_2$$

where p_1 is an existing proven proposition and p_2 is an existing unproven proposition. Since a first-type proposition is created along with its proof, the proposition $p = p_1 \to p_2$ is by creation a proven proposition. Now that both $p = p_1 \to p_2$ and p_1 are proven propositions, this implies p_2 is also proven. All owners of the tokens issued earlier from the proposition p_2 are rewarded with MathCoins in the same amount (i.e., the final token price = 1). In the contrary, all owners of the tokens issued from the negation proposition $\neg p_2$ are destroyed without any refund. Conversely and similarly, if a first-type proposition in the following special form

$$p_1 \to \neg p_2$$

where p_1 is an existing proven proposition, is created, then the proposition p_2 is proven false. All owners of the negation proposition tokens will be credited and the original proposition token owners will lose in this case.

It is possible (although not usual) that p_2, the proposition that just got proven, is also in the special form

$$p_2 := p_3 \to p_4$$

where p_3 is an existing proven proposition and p_4 is an existing unproven proposition. Then, p_4 is declared proven. Recursively, the form of p_4 will be checked again until the next proven proposition is no longer in this special form. All the associated activities described above shall be processed in the same block that includes the first-type proposition p in order to let all miners in the network agree that the block is valid.

E. Token Transactions

A valid token purchase transaction associated with an unproven proposition should include the following information:

1) The address of the unproven proposition.
2) Indicator of the intention of buying either the original proposition or the negation proposition.
3) (Optional) MathCoins for getting the tokens.
4) (Optional) Additional fee to the miner.
5) (Optional) Additional fee to the network (public fund). This becomes required if item 3 is zero.

In item 3, an amount of MathCoins is specified to obtain proposition tokens starting at the current price. The exact units of tokens obtained is calculated through a deterministic function to be described in Section III-A. The additional fee that goes to the public fund (i.e, item 5) is be preserved to reward those who help to prove or disprove the proposition eventually.

A valid token sale transaction associated with an unproven proposition should include the following information:

1) The address of the unproven proposition.
2) Indicator of the original proposition or the negation proposition.
3) The units of the tokens for sale.
4) (Optional) Additional fee to the miner.

III. Pricing Rules of Unproven Propositions

A. The Token Pricing Function

Figure 3 illstrates the idea of the token pricing function (TPF) and how the price of a proposition address varies according to activities of token buyers.

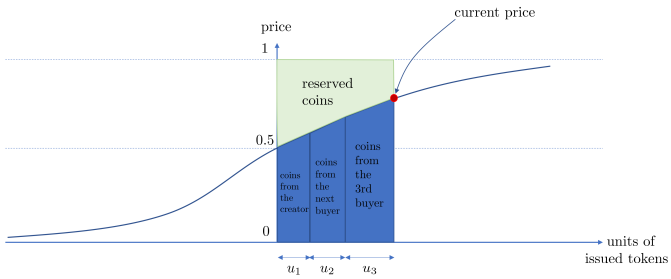

Fig. 3. Illustration of the token pricing function

The token pricing function $\phi : \mathbb{R} \to [0, 1]$ adopted by a MathCoin blockchain must satisify the following properites:

1) ϕ is monotonically non-decreasing: that is,

$$\forall x, y \in \mathbb{R}, x \geq y \Rightarrow \phi(x) \geq \phi(y).$$

2) $\int_0^\infty (1 - \phi(x))dx < +\infty$.
3) $\phi(x) + \phi(-x) = 1, \forall x \in \mathbb{R}$.

These properties deduce that $\phi(0) = 0.5$, $\lim_{x\to\infty} \phi(x) = 1$, and $\lim_{x\to-\infty} \phi(x) = 0$. Once a token pricing function ϕ is selected, the following definitions associated with ϕ will be found useful in the upcoming developments:

1) A finite positive number r_ϕ defined as

$$r_\phi := \int_{-\infty}^{0} \phi(x)dx \tag{1}$$

2) A function $\Gamma_\phi : \mathbb{R} \to \mathbb{R}$ defined as the integral of ϕ:

$$\Gamma_\phi(y) := \int_0^y \phi(x)dx. \tag{2}$$

3) A function $t_\phi : \mathbb{R} \to \mathbb{R}$ defined as the inverse function of Γ_ϕ:

$$t_\phi(x) := \Gamma_\phi^{-1}(x) \in \{u \in \mathbb{R} \mid \Gamma_\phi(u) = x\}. \tag{3}$$

The token pricing function ϕ can be understood as: the token price is $\phi(x)$ when the equivalent total amount of issued tokens is x units. As illustrated in Figure 3, the creator of the proposition address acquires u_1 units of tokens after he pays the initial "investment" to the network in the amount of

$$v_1 = \Gamma_\phi(u_1) = \int_0^{u_1} \phi(x)dx.$$

The units of tokens to be given to the creator is calculate through the t_ϕ function: $u_1 = t_\phi(v_1)$. After the purchase, the price rises from 0.5 to $\phi(u_1)$. Suppose that subsequently, some more token buyers made purchases of u_2 and u_3 units of the token, making the price rises to $\phi(u)$ where $u = u_1 + u_2 + u_3$. The red spot $(u, \phi(u))$ indicate the equivalent total amount of issued tokens (u) and the price $(\phi(u))$. It is known publicly to the network and informs the next token buyer that if you would like to acquire Δu more units of token, you shall pay MathCoin in the amount of

$$v = \int_u^{u+\Delta u} \phi(x)dx, \tag{4}$$

and after your purchase, the price will be updated to $\phi(u + \Delta u)$. It also tells all the token holders that if you would like to sell Δu units of the tokens at your hand, you will be given MathCoin in the amount of

$$v = \int_{u-\Delta u}^{u} \phi(x)dx,$$

and the token price will decrease to $\phi(u - \Delta u)$ after the sale. As the final value of any unit of token is either 1 or 0, all the token buyers are supposed to believe that the proposition is true (or at least a good chance to be true) and expect the final value is 1. Once the proposition is proven true (i.e., it is connected to the axiom nodes and becomes a logical consequence thereof), the token holders will be paid what they deserve.

Since the final payment back to token holders is larger than the sum of what all users transferred to the math-domain, the public fund will be responsible for paying the difference, using the public fund accumulated from miners' taxes of previous blocks. Fortunately, this extra payment can be found to be finite and bounded by the value r_ϕ. Therefore, the public fund can just reserve r_ϕ units of MathCoins for this conjecture. The actually amount that is finally paid by the public fund is

$$r = \int_0^u (1 - \phi(x))dx,$$

as shown in the green area in Figure 3. The remaining unused MathCoins in the amount $\int_u^\infty (1 - \phi(x))dx$ is returned back to public fund.

If the miner's tax rate is set as 0.5 (i.e., for every block found, half of the newly created coins goes to the miner and the other half goes to the public fund, then we have a good reason that there should be sufficient amount of math-domain coins to be paid to the winners, at least for the first proven proposition. However, after the reserved coins are awarded to the token buyers and flow to the user domain, there may be more user-domain coins than math-domain coins. Such an issue will be addressed in a later section.

B. Tokens of the Negation Proposition

Once a proposition address is created, users can also purchase tokens for the negation of the proposition. The price of the negation proposition is $1 - \phi(u)$. So if some user believes that this proposition is false, she can buy tokens of the negation of the proposition. As illustrated in Figure 4, the buyer of the negation proposition token can choose to buy Δu units and pay the amount of

$$v = \int_{u-\Delta u}^{u} (1 - \phi(x))dx.$$

Then, the price of the original token will reduce to $\phi(u - \Delta u)$. While the total amount of issued proposition tokens is still u units, the *equivalent* total amount of proposition tokens is updated to be $u - \Delta u$, which decides the updated token price that is publicly known.

More generally, let us denote u_p as the total amount of user-held tokens of the original proposition and u_n as the total

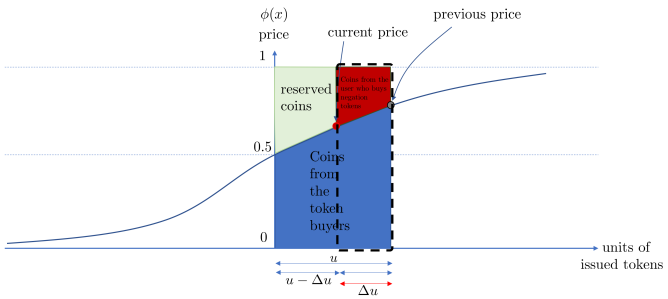

Fig. 4. Token price function with token buyers of the negation proposition.

amount of user-held tokens of the negation proposition. Then the *equivalent total amount of proposition tokens* is defined as

$$u = u_p - u_n$$

which can be any real number, positive or negative (or zero).

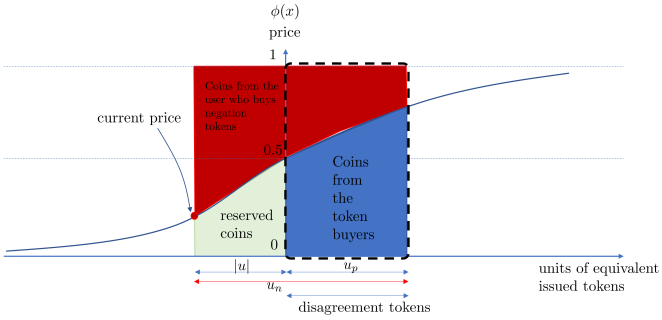

Fig. 5. Token price function with token buyers of the negation proposition.

Figure 5 shows the case when $u_n > u_p$. In this case, the price of the original proposition token, $\phi(u)$, becomes less than 0.5 because $u < 0$. The coins needed to be reserved by the network change to the bottom left part of the figure. If the negation proposition is eventually proven true (i.e., the original proposition proven false), these coins are used to pay the negation token buyers (together with the coins from the positive token buyers, who lose the game).

The smaller value between u_p and u_n, i.e., $\min\{u_p, u_n\}$, is called the *total amount of disagreement tokens*. For these tokens, the network do not need to preserve any coins to pay the winner. The coins collected from the users of both parties already cover it (see the dashed rectangles in Figures 4 and 5).

C. The Public Fund's Participation In Token Purchasing

The previous subsections describe the rules of price changes of any created proposition address involving only the token purchases and sales of the token holders of the proposition and its negation. It is interesting to note that the public fund can also involve in the game, as will be elaborated here. In this subsection, we propose ways to change prices of proposition addresses by the network itself and without user activities. The rule of thumbs is to change prices so that the whole network may converge to a point that is closer to the "truth." Of course, since the MathCoin blockchain may not be equipped with a sophisticated theorem prover (it should not; as this will increase unnecessary complexity for miners), the blockchain can use simple logic that it knows to do this task (and earn some coins, though in small amount, for the public fund).

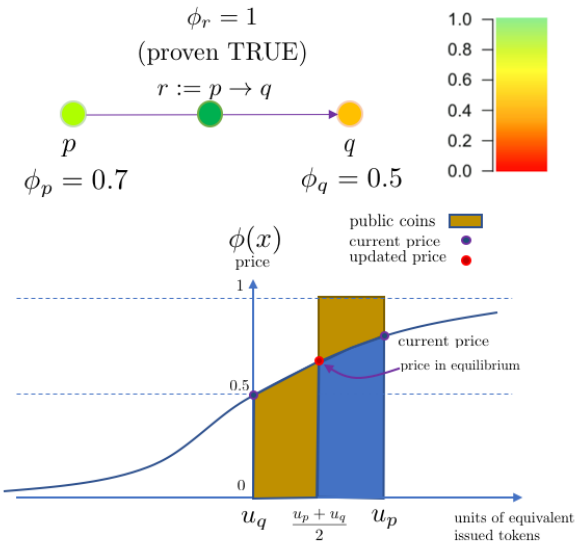

Fig. 6. Illustration of the activities of the public fund when $p \to q$ is a proven proposition on chain but p has a higher price than q.

Consider the case depicted in Figure 6, where we have two proposition addresses p and q, with their current price being $\phi_p := \phi(u_p) = 0.7$ and $\phi_q := \phi(u_q) = 0.5$ which reflect the current human confidence levels of these two propositions. Here, u_p represents the units of equivalent issued tokens for p, and u_q that for q. Now, suppose there is another proposition $r := p \to q$, whichends up being proven by the network. Then, a reasonable price of q should be greater than that of p. In presence of the event of $u_p > u_q$, the public fund will do the following things:

1) Let $\Delta u = (u_p - u_q)/2$.
2) Purchase Δu units of $\neg p$ and Δu units of q.

After the action, both the prices of p and q will reach to an equilibrium: $\phi((u_p + u_q)/2))$. It is not difficult to see that the public fund will not lose any MathCoins in the long term by the above actions. Instead, it may even earn some coins roughly in proportional to the price gap $\phi_p - \phi_q$, since at least one of $\neg p$ and q will eventually be proven true.

Now, consider the case where the proposition $r := p \to q$ has not been proven, but has a rather high price close to 1. Then, as long as the price gap between p and q, $\phi_p(u_p) - \phi_q(u_p)$, is greater than $1 - \phi_r(u_r)$, it is still possible for the public fund to do something to reduce the price gap:

1) Find $\Delta u > 0$ such that the updated price gap, written as $\phi_p(u_p - \Delta u) - \phi_q(u_q + \Delta u)$, is equal to $1 - \phi_r(u_r - \Delta u)$.
2) Purchase Δu units of $\neg p$, Δu units of q, and Δu units of $\neg r$.

After the operation, the prices of these three propositions will satisfy

$$\phi_p(u'_p) - \phi_q(u'_q) = 1 - \phi_r(u'_r)$$

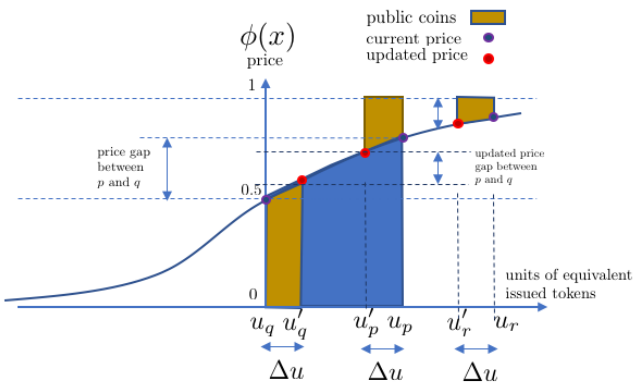

Fig. 7. Illustration of the activities of the public fund when $r := p \to q$ is an unproven yet high-priced proposition address and p has a higher price than q.

where $u'_p := u_p - \Delta u$, $u'_q := u_q + \Delta u$, and $u'_r := u_r - \Delta u$ are the updated units of equivalent issued tokens of p, q, and r, respectively.

A few remarks are made below.

1) The public fund is used to purchase Δu units of three proposition tokens ($\neg p, q$, and $\neg r = \neg q \to \neg p$). According to the basic logic, at least one of these three proposition is true. Therefore, the public fund will never lose for such kind of actions. It is somehow like an "arbitrage."
2) In practical situations, the occurrence that the price of p is greater than q with a large gap like this may be an unusual event, since the users and mining nodes on the network are smart enough to know a chance of "arbitrage." But there are a number of reasons to set up this public rule of a MathCoin blockchain.

We describe it from the perspective of the public fund. First, recall that the very reason for establishing the public fund is to encourage submissions of true knowledge that will end up helping "revealing" the true knowledge. Although the blockchain protocol itself is not equipped with a sophisticated theorem proving capability, it can still use the basic logic to help reveal the knowledge in case users of the network accidentally forget this opportunity of arbitrage. Secondly, for many users, the above action may not really seem to be a worthwhile investment. Although the action is guaranteed not to lose money, it does not earn much as well. For Δu units of the three kinds of tokens that guarantee a return no less than Δu MathCoin, one needs an amount very close to Δu MathCoins, not including the miner's fees (for the three separate token transactions). So, unless the price gap is very large, it is reasonable to say that users may not be very interested in it. The public fund, on the other hand, does not need to pay the miner's fee. In addition, note that the amount of MathCoins held by the public fund, until rewarded to users for proven propositions, is just a number recorded by the blockchain protocol and has never been actually coined yet. As long as it can be sure it will not lose the MathCoins (with the tokens), it does not care how long it will take for this "investment" to return, since the public fund is not for profit anyway.

Knowing that the blockchain protocol will have some, although not frequent, "interventions" of proposition prices, the users are encouraged to create new proposition addresses as logical implications of others (e.g., $r := p \to q$) as an act to support the proposition q they may have real interest in. In the next subsection, we elaborate on how different roles involved in the game can gain, leading to an all-win situation.

D. The Game of Creating New Proposition Addresses

The rules of creating new math-domain addresses shall be designed to encourage contributing privately-held knowledge to the public. Predictable incentive should be awarded to the contributor. The following reference design of the rules describe how this can be done.

1) Contributors: For a potential contributor who knows that a particular well-formed formula $s := s_1, ..., s_n$ is true, where $s_1, ..., s_n$ are existing math-domain addresses already defined on the chain. She can then claim the truthfulness of s by signing a proposition-creation transaction. As introduced in Section II-D, the information of the transaction should include 1) the statement $s_1, ..., s_n$ itself, 2) an optional name for the conjecture, 3) the required network fee v_{net}, 4) an initial deposit v_{init} used to acquire tokens, 5) an optional additional fee v_{donate} to the network, and 6) an additional fee v_{miner} to the miner who include this transaction in the next block. Accordingly, the total cost of the transaction in MathCoins would be:

$$v = v_{\text{init}} + v_{\text{miner}} + v_{\text{net}} + v_{\text{donate}}. \tag{5}$$

The number of tokens that the contributor will acquire is $u_{\text{init}} := t_\phi(v_{\text{init}})$, where t_ϕ was defined in (3), and the price will rise from $\phi(0) = 0.5$ to $\phi(u_{\text{init}})$. If the proposition is indeed true, the price of the proposition address will eventually rise and approach 1. So the gain that the contributor can expect is

$$v_{\text{gain}} = \int_0^{u_{\text{init}}} (1 - \phi(x))dx - v_{\text{miner}} - v_{\text{net}} - v_{\text{donate}}. \tag{6}$$

As long as the gain v_{gain} is positive, a contributor will eventually earn what she deserves.

The required network fee v_{net} is designed to be proportional to the length of the proposition. As the blockchain space is very precious, it is important to set up a criterion to prevent users from creating propositions that are long in description. If the contributor can reuse as many definitions already defined on-chain as possible, she will be able to save a great amount of network fee.

The miner's fee v_{miner} is optional. But in the case of a network congestion, a certain amount of miner's fee may help to expedite the process for the contribution being recorded on the chain.

The donation to the network v_{donate} is also optional. It is an amount provided to the network to reward those who help prove (or disprove) the proposition. Note that for every created

proposition, the network should reserve an amount to reward those who help prove or disprove the proposition:

$$v_{\text{prize}} = \int_0^\infty (1 - \phi(x))dx = \int_{-\infty}^0 \phi(x)dx = r_\phi$$

where r_ϕ was defined in (1). The source of this reservation of prize primarily comes from accumulation of miners' tax of previously blocks. But the donation v_{donate} by the contributor can be used to cover the cost from the network.

Normally, a contributor in the mindset of earning MathCoins will not donate. But another type of contributors, namely those who are eager to know the answer and willing to pay for the answer, may set the donation. The donation can be even larger than the maximum prize reservation, i.e., $v_{\text{donate}} \geq v_{\text{prize}}$, making it possible for the token holders of true propositions to receive more than 1 MathCoin per unit token. Smart contracts can be used here to let a contributor specify a deadline by which she hopes to see the answer: if the network is not able to answer her by the deadline, the donation is returned to the contributor. Details on how such smart contract templates can be designed are left to a development team and will not be elaborated in this article.

According to the rules described above, there will be many users who are trying to post propositions that are very easy to prove. They can expect to earn up to v_{prize} whenever such a transaction is successfully recorded by a miner. But such kinds of proposition requests will eventually drain out the public fund. In a viable design, some mechanisms should be employed to prevent users from creating propositions that are too easy to prove. We will elaborate on this in the next subsection.

2) Block Miners: For the block miners, the direct incentive to include a proposition-creation transaction is the fee to the miner (v_{miner}). So, in general, as long as the fee value v_{miner} is sufficiently attractive to a miner, it would choose to include it in a subsequent block it is finding. However, it is also clear to the miner about the potential gain that the contributor would get (see Eq. (6)). Suppose a miner possesses some capability of automated theorem proving. If the proposition is somewhat trivial and can be verified by a miner before it even finds the next block, then the miner may choose not to include the transaction. Instead, it may immediately broadcast another proposition-creation transaction of the same proposition, but with a higher miner's fee v_{miner}. Since only one of the identical propositions will be finally written on-chain, the other miners will naturally drop the one with a lower fee, making the original contributor gain nothing. This may discourage contributors from creating propositions that are too obvious.

The protocol can even be designed to allow a miner to take all the MathCoins the contributor provided (Eq. (5)) if the miner managed to write down the full proof in the next block it finds (using the first-type proposition transaction, with a network fee proportional to the length of the proof). With such a potential risk, a contributor will only post propositions that she believes are hard to prove by others. This also helps the blockchain to use the limited space and processing capability to reveal only the most difficult theorems to the world.

3) Endorsers: For any on-chain proposition whose price is still not converging (i.e., not close to 0 or 1), this is an opportunity to anyone who knows more about the proposition than the entire community. If one believes a proposition is true but is under-valued, she can purchase some units of tokens of the proposition and expects a price increase in the future. Token holders who hopes to expedite the price increase can create one or more propositions serving as the "lemmas" of the propositions. If the lemmas are easier to verify by the rest of the network, the prices of the lemmas are expected to rise much faster, which will eventually help the price increase of the original proposition.

The contributors or the endorsers of a proposition really do not have the obligation to post the proof to the blockchain. It is up to their choice whether to publish a proof in a research article (off-chain). In fact, a contributor does not even need to have a formal proof. As long as she is confident that the proposition can one day be proven true, she can make the contribution or endorsement, which serves as the price to be paid once she is proven wrong. On the other hand, if a token holder finds that the price of the proposition severely under-valued, she can reverse the trend by posting lemmas, which is an action of showing she is right (well, before doing so, she may have also purchased some more tokens at a low price).

4) Research community: For authors of research articles that contain some mathematical theorems, besides writing a proof or a proof sketch on the paper, they can also act as a contributor to the MathCoin blockchain. For reviewers of the research articles who are responsible to check the validity of the theorems, as an alternative to reading the proof meticulously, they can also simply check the MathCoin network to know the level of correctness of the theorem. Moreover, when the development of MathCoin blockchains becomes mature, some journal editors can even choose to require contributing authors to pre-verify their theorems on the MathCoin blockchain before the review process begins. This will substantially save the reviewers' resources, while maintaining (or even improving) the quality of the published results of a publication.

5) Investors / Funding Agencies: For investors whose business of interest involves some knowledge that can be formulated as some mathematical propositions, traditionally they may choose to hire experts of the interested fields as consultants and ask the consultants to advise. However, it may be difficult and inefficient to find a competent expert who really knows the problem with limited social connections, and within a short time. In the presence of a well-functioning MathCoin network, the investors can simply create an award (using a large v_{donote} described earlier in the subsection, before Eq. (5)) to the network to encourage real experts of this problem to show up, solve the problem, and get the award (without even having to know each other or meet each other). Given the above protocol, an investor who wants to know whether a proposition $s = s_1, ..., s_n$ is true or false can do the following:

1) Post a proposition-creation transaction for $s = s_1, ..., s_n$ with no initial deposit (because it does not know if it is

true of false), some fee to miners, and fee to the network, and a large additional fee v_{donote}.

The investor can also set a deadline using a smart contract, making the excess fund to be returned to the investor if the proposition addresses are still not proven (i.e., isolated to other proven proposition addresses on chain) by the deadline.

From the above descriptions, it is imaginable that in the long term, the eco-system of investors and experts can be changed in the following way:

1) The investors can be exposed to less risk related to technical development. They only need to focus on the real market value and pay only affordable (and reasonable) fees to the network and wait for the network to return a reliable answer.
2) Technical experts may not have to rely on a fixed employment. They can just scan on the MathCoin network to check whether there are open questions on chain that may be rewarding. As long as they are capable enough, they can choose not to be employed by anyone, and will be able to earn their own living by solving questions that the rest of the world do not know well. For the rest of the time, they can continue develop to skills and knowledge of their own interest, and live a happy life.

E. Illustrative Examples and Discussions

In this subsection, we use Figure 8 to illustrate a possible example sequence of proposition creations. The blockchain was born natively with N_a axiom addresses $a_1, a_2, ..., a_{N_a}$, whose prices are fixed to 1 forever (and no tokens will be issued from these addresses). Then, proposition p_1, a second-type proposition address, was created by some user (through exisiting definition addresses: $p_1 := s_3 s_6$), with an initial price of something slightly greater than 0.5, say, 0.55. Since the proposition p_1 are believed to be true by many users (some users even have a proof of the proposition themselves), they use their MathCoins to buy the p_1 tokens, driving the price of p_1 to 0.99. Later on, another proposition address p_2 was created by another user. The price does not increase as fast as p_1, probably because it is more difficult to prove. The token owners of p_2, in order to boost the price of p_2, are trying to persuade the community that p_2 is true. They might do all sorts of off-chain activities like publishing papers, giving seminar talks, making videos, etc., to "promote" the validity of p_2, in order to encourage other MathCoin owners to invest. On the other hand, they may also do some on-chain activities to promote p_2. A straightforward way might have been just posting the proof of p_2 on-chain. But the full proof they have at hand is too lengthy and would have cost a high network fee, so instead, they try to relate their proposition to some high-priced propositions. For example, they make a new proposition address p_3 defined as $p_1 \to p_2$. Since p_3 is easier to prove than p_2, the price p_3 goes up very quickly. The presence of p_3 creates a chance to help increase the price of p_2 from p_1. Since p_1 has a higher price than p_2, according to the mechanism described in Section III-C, the participation of the public fund will increase the price of p_2. Later on, someone even provides a direct proof of p_3 by expressing p_3 as a finite combination of axioms and definitions. Now, p_3's price becomes 1 and all token holders of p_3 are rewarded by the full price. p_3 becomes an address "connected" to the axioms and is label "proven" by the network. Now that p_3 is proven, value keeps flowing from p_1 to p_2 whenever p_1 has a higher price than p_2. The price of p_1 may be decreasing for a while since the public fund purchased the tokens of $\neg p_1$, but for the users who believe p_1 is true, it is not a big deal, but a good chance to invest.

It is not necessary for all the propositions to be connected to the axioms eventually (i.e., labeled "proven"). Knowledgeable persons (i.e., those who have checked the proof in private, or those with capable ATPs) may view it as an advantage against those who are not sure whether the proposition is correct. Until the proposition becomes "proven" and has a price fixed to 1 forever, they always have a chance to earn MathCoins by buying the proposition's tokens. Such an advantage disappears when someone else finally posts a valid proof on-chain.

Now, let us turn to propositions p_4, p_7 and p_8 on the upper-right corner of Figure 8. The token owners of p_7 find that the price is unfavored to them because many people on the network believe it is wrong and purchase many tokens of $\neg p_7$. In order to save the price, one of them creates $p_8 := p_4 \to p_7$, in an attempt to attract value to flow from p_4 to p_7. Unfortunately, p_8 is also not experiencing a good evaluation. However, as long as the price of p_8 is still large enough ($\phi_{p_8} \geq 1 - \phi_{p_4} + \phi_{p_7}$), there is always a chance for value to flow from p_4 to p_7. In order to remove this possibility, token owners of p_4 who do not believe p_7 is correct may try to invalidate p_8. One direct method is to post a proof that shows that p_8 implies the negation of any axiom or of any proven proposition. Once this is done, all values of p_8 go back to the public fund. It is important to note that even if p_8 is invalidated, it does not automatically invalidate p_7. The failure of p_8 makes proposition p_7 lose a value support from p_4, but believers of p_7 can still try to promote p_7 by posting other "proofs", either by directly showing that p_7 is a a logical consequence of some other proven propositions, or by showing that p_7 is a logical implication of another properly-priced proposition. The opponents of p_7 (i.e., token holders of $\neg p_7$), on the other hand, can try to prove that p_7 is wrong by showing that p_7 implies the negation of one axiom or of any proven proposition.

This is actually a race between advocates and opponents of p_7. They indeed disagree with each other. But they do not have to fight with each other in the real world. They do not have to meet to "discuss." In fact, they don't even need to know who the other party is. They just keep working on the proof of the proposition they believe is true. Posting a final proof on-chain is the end of the race. Whoever does it first wins the game. (Well, actually the winner was already known by God; the human are just doing the race to complete the game and reveal the truth). The loser of the game will not complain (and can not complain) since they see the proof of the negation of the proposition they once believed to be true. Although they lose some MathCoins, they finally learned the truth. At the end of the race, the rest of the people in the world is benefited by the augmentation of the knowledge on-chain, for free.

Now, let us go to p_2, p_{10}, and p_{11}. Proposition p_{10} was

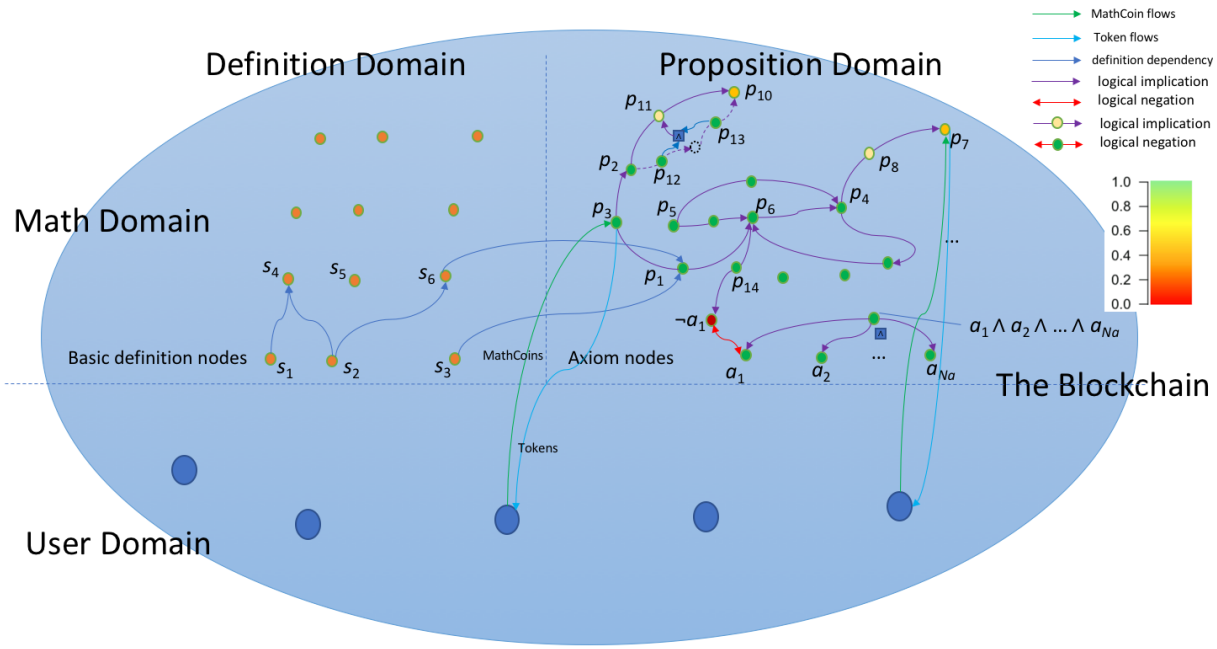

Fig. 8. Exemplary illustration of relationships of math-domain addresses

created by someone but undergoes unfavorable evaluation. The token owners try to claim that the properly-priced p_2 implies p_{10} by creating the proposition $p_{11} = p_2 \to p_{10}$. Unfortunately, p_{11} is still not well priced probably because it is intuitively wrong to many. Nevertheless, the believer of p_{11} and p_{10} managed to divide p_{11} into two parts that are easier to prove by the general public: p_{12} and p_{13}. They created two propositions p_{12} and p_{13} and show that p_{11} is a consequence of $p_{12} \wedge p_{13}$. After p_{12} and p_{13} receive good pricing (or even be proven "true" by the network), the prices of p_{11} and p_{10} will eventually goes up.

Note that any proposition can either be proven true or proven false. If any of them is proven both true and false, that means the underlying axioms are inconsistent, and therefore the entire blockchain will collapse. Before launching a MathCoin blockchain, it is important to make sure that the axioms are consistent (many experts in the field, other than the author of this paper, know how to choose a good set of axioms). It is also important to make the code go through formal verification before launching.

F. Accounting of the Public Fund

The public fund shall obey the following rules.

1) The usage of the public fund shall be completely predictable and transparent.
2) The public fund shall never go in deficit.

The first rule is usually feasible as long as all the activities that the public fund will involve (rewarding to the contributors and public token purchasing) are governed by formulas predefined in the protocol. At the end of a block, the balance of the public fund is known by all mining nodes. It is the balance of the previous block, plus the tax collected from the miner, plus fees from creators of new math-domain addresses, minus the distribution of rewards to token owners of all proven propositions, minus those used in public token purchases. The tokens that the public fund purchases should also be listed as assets. According to the analysis given earlier, these assets are always have a value greater than or equal to the total amount spent in public token purchases. So, the only possibility that the public fund may lose money is the distribution of rewards to token owners of proven propositions. For every created unproven proposition, the public fund will reserve MathCoins in the amount of $r_\phi = \int_0^\infty (1 - \phi(x))dx$ in preparation of paying to the token holders (no matter whether the proposition is eventually proven true or false). In order to keep the public fund to always remain nonnegative, a policy could be set up to limit the number of newly created proposition addresses within a period of time. When the public fund is low, the requests of proposition creation can be suspended until there is sufficient fund. Other approaches may include increasing the miner's tax rate in the event of a low public fund, or to change the token pricing function ϕ with a smaller reservation amount r_ϕ.

IV. Implementation Issues

A MathCoin blockchain that follows the above descriptions and principles can be implemented with various paramaters and assumptions. A successful implementation may be due to careful choices of these design parameters. The choice of the parameters are left to the design teams under their discretion in the market acceptance. We make a list of implementation issues here for a developer's reference:

A. Choice of the consensus algorithm and coin distribution

Just like Bitcoin and other blockchains, some design parameters need to be considered and shall include at least the

concensus algorithm, the block reward, the expected block interval. These parameters govern the mining and distribution of the coins supplied by the chain. The total supply of MathCoins should be set to be a fixed number. This means the miner's reward per block can not always be constant. It must decrease and converge to zero as time goes by. When the block reward approaches to zero, the speed of accumulation of public fund will also slow down and approach zero. What will the network use to attract new contributions then? If the price of MathCoin keeps going up (if the market keeps giving it an increasing value for its increasing contents), then the incentive will still be good.

B. Choice of the miner's tax rate

The miner's tax rate dictates the speed of accumulation of the public fund. And the size of the public fund affects the capability of the network to attract knowledge contributors. It is therefore preferable to have a higher tax rate. However, when the tax rate is too high, the regular miners may be discouraged and leave, leading to an immediate risk of the blockchain operation. The choice of the miner's tax rate is therefore a critical problem in establishing a successful MathCoin blockchain. A tax rate that varies according to runtime situations may also be considered (e.g., increase the tax rate when the fund size is low and the number of miners is sufficiently high) as long as a deterministic formula of tax rate is pre-defined and known by all miners.

C. Choice of the token pricing function

The token pricing function as introduced in Section III-A may have a wide range of choices to satisify the constraints described therein. Many logistic functions ranging from -1 to 1 can be chosen. For example, the hyperbolic tangent functions

$$\phi(x) = \tanh(\lambda x) = \frac{e^{\lambda x} - e^{-\lambda x}}{e^{\lambda x} + e^{-\lambda x}}$$

where $\lambda > 0$ are a perfect class of functions satisfying the TPF constraints introduced in Section III-A. When the parameter λ is chosen to be a large number, a steep slope in ϕ will result in a sharp price change. Also, the maximum amount that the public fund needs to reserve for a single unproven proposition $r_\phi = \int_0^\infty (1 - \phi(x))dx$ is smaller. However, the problem with this kind of function is that it may be difficult to calculate the number of tokens (i.e., the inverse operation of the integral equation (3)). It should be noted that the TPF function is not necessarily differentiable nor continuous. Furthermore, since the total supply of MathCoins is a finite number, the domain of the TPF that is relevant is actually just a subset of $\mathbb{R}$ of a finite length (i.e., $(-a, a)$ where a is the maximum number of tokens issued to a single proposition when all MathCoins are used to buy its tokens). A non-decreasing piecewise constant function or a non-decreasing piecewise linear function are also good choices, despite that they are not "smooth," since the token number is much easier to calculate. Since every full node that participates in mining MathCoins would calculate this (in order to verify every block), it is of paramount importance to choose one that has a low computational burden, and also, a low chance of disagreement among participating mining nodes.

D. Encoding schemes of definitions and propositions

Any definition and proposition addresses is a sequence of finite existing addresses, including the basic definition symbols $(\forall, \neg, \rightarrow, \varnothing)$. It is reasonable to assume that basic definition symbols are more frequently used than definitions created afterwards. While user-domain addresses can be fixed-length, it is suggested to let math-domain addresses have variable lengths. The native math-domain addresses (i.e., basic definitions and axioms) will have the shortest length since they will be likely to be used more frequently than many others.

E. Choice of the set of axioms

The axioms nodes are the foundation of the whole blockchain. They are the only nodes that are set to be true since the inception of the chain until forever. Any other proposition shall not violate any axioms, or their prices will fall down as time goes by. For example, one may choose to use the nine axioms in the Zermelo-Fraenkel with choice (ZFC) theory as the foundation of the blockchain and it can be marketed as the ZFC chain.

F. Choice of Basic definitions

It is up to an implementation team to decide how many basic math definitions are to be included in the initial chain. It can be chosen as the minimal suggestions listed in Table I. It can also include much more commonly used definition such as set of numbers, etc. The former will have a simple initial implementation while the latter makes the platform useful for much more people in the very beginning of the launch.

G. Choice of Allowing Transfers of Proposition Tokens

In the preceding presentations, the proposition tokens are designed in a way that one can only purchase tokens from the network and can only sell tokens back to the network. There is no token transfer transaction defined in this article. Although the author holds the stance that such a type of transactions is not necessary, it is up to a development team's consideration on whether to allow token transfers. Allowing token transfers between user-domain addresses may create opportunities to buy and sell proposition tokens using other currencies via some exchanges.

H. Standalone or non-standalone implementations

A new MathCoin cryptocurrency may be implemented as a standalone blockchain. It can also be deployed as a set of smart contracts on top of Ethereum or any other blockchain that supports smart contracts. Although the majority of the article is written assuming the standalone implementation, it should still be possible to change it as a smart contracts deploying on another chain. In fact, it may be easier and quicker for a team to realize the idea by deploying a smart contract on an existing blockchain (e.g., as an ERC20 token

on ETH) than starting everything from scratch. However, some constraints inherently from the underlying blockchain may arise (e.g., smallest decimal, block interval, congestions caused by other services on the underlying chain) and they may affect the long term developments of the new coin. On the other hand, if a team chooses to implement it as a standalone blockchain, it would be responsible for making sure the incentives given to the miners are sufficiently strong to avoid the events where miners may collectively refuse to include some legal transactions in a block. In either case, it is especially important to note that the foundation of the blockchain must be impeccably sound such that no malicious attacks can destroy the consistency of the network. A formal verification process that verifies the soundness of the implementation, not only to the team members themselves, but also to the general public who may participate as investors or contributors, is strongly recommended before the official launching.

I. Choice of external storages

As new definitions and propositions are created, the size of the blockchain may grow larger and larger. New materials may also have longer descriptions than existing ones that may require larger fees. In order to the blockchain continues to grow without bound, it should be in the consideration of a development team to survey the possibility of incorporating storage services (such as IPFS, Storj, etc.).

J. Price Stability of MathCoin

The incentives, rewards, penalties discussed in this article so far are all in MathCoins. However, the exchange rate of a MathCoin to other major currencies, crypto- or fiat, is an important factor. But the developers may not have too many means to control it except for marketing strategies. It is preferable to make the MathCoin price stable in the short term and make the price steadily and gradually grow in the long term. Some recent works on this issue [13] may be a good reference for this purpose.

K. Relationship with Existing ATP Tools

Although there are already many automated theorem proving (ATP) tools that can verify proofs interactively (e.g., Coq, Isabelle) or automatically, the contribution of the current proposal is still significant since the common public users will be given the ability to instantly and reliably learn the truthfulness of any given proposition recorded on-chain without having to get familiar with the ATP tools.

It should be emphasized that the proposal MathCoin blockchain is not intended as a replacement of existing ATP tools. Quite the contrary, it is intended to become a complement of these many excellent ATP tools. The blockchain structure described in this article does not actually contain powerful capabilities in automated theorem proving. Its (expected) strengths will come from the many contributors around the world, and the strengths of these people may be from these powerful ATP tools.

Therefore, an implementation of the MathCoin protocol does not have to specify any existing ATP tools. However, it should be expected that the participating users and miners may have their own ATP capability in order to gain themselves some advantages in the game. Conversely, a MathCoin protocol that can attract sufficient users and miners with good ATP capabilities has a better chance to maintain the long-term value of the whole blockchain.

V. Conclusions

In this article, we have presented MathCoin, a blockchain proposal aiming to deliver all proven mathematical theorems to every person in the world in a manner that is reliable, instant, and with low cost. Mechanisms are carefully designed to garner and accumulate wisdom of the whole human society (plus their machine assistants) by fairly paying the contributors of true knowledge. Specifically, experts who have correct proofs at hand of theorems that the rest of the world does not know can eventually earn MathCoins from the network. Funding agencies can pour their precious fund into specific problems of interest and encourage research activities around the problems, and expect to get a definite answer within a preferred time frame. Speculators who only guess the prices of open questions without really work on the proofs themselves will usually lose the game. Common people who are not familiar with technical details of a proof will be able to access to proven mathematical theorems with a high level of confidence without having to trust a small number of experts. They are required to pay a fee only when the problem of interest is still not well known by the chain. Years after the launch of any successful MathCoin blockchain described here, we may expect the world to be much more reasonable than today, yet without requiring many people to become geeks.

Although many design details are still needed for interested development teams to decide, materials provided in this article show that it is promising for such a plan to succeed. With a huge total gain to the entire human society, it is reasonable that some entrepreneurs can find profitable ways to develop.

Up to now in the article, we have only discussed the issue of verifying mathematical theorems. However, it is possible in the near future to consider an extension of the current blockchain design to include physically observable events into the system. The truthfulness of anything ranging from physical laws to historical events can be first added by anyone as a conjecture in the blockchain (with some bets or donations). And later on, using observable facts contributed by users (given some incentives, of course) and the blockchain's existing reliable mathematical and logical theorems, the conjectures can be logically connected to other observable facts. Since any fact is a logical consequence of other facts, any fact that was not clear or widely accepted can be gradually proven true on-chain as a logical consequence of confirmed observable physical events. How the extended blockchain may confirm observable events is left as a future work (existing cryptocurrencies involving the prediction market, e.g., the augur project [9], may be a good source of design reference). Markets may decide the values of finding a particular fact, but it is imaginable that the total cost for finding the truth will be much cheaper than today, and every person can contribute freely within their expertise and earn fairly from and for the rest of the world.

With contributors around the world and from various disciplines, the extended blockchain will be able to reveal virtually every fact and their logical consequences to everyone connected to the Internet. When this eventually happens, it will be effortless to combat rumors, as rumor spreaders will find it hard to let their statements achieve a good price in the extended blockchain.

ACKNOWLEDGMENTS

The author would like to thank Byung-Jun Yoon, Chia-Chieh Chu, and Era Yu, respectively, for their valuable discussions and encouragements during the inception of this work.

REFERENCES

[1] A. Wiles, "Modular elliptic curves and fermat's last theorem," *Annals of Mathematics*, vol. 141, no. 3, pp. 443–551, 1995. [Online]. Available: http://www.jstor.org/stable/2118559

[2] G. Gonthier and A. Mahboubi, "An introduction to small scale reflection in coq," *Journal of Formalized Reasoning*, vol. 3, no. 2, pp. 95–152, 2010. [Online]. Available: https://jfr.unibo.it/article/view/1979

[3] N. T., P. L., and W. M., *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*. Springer, 2002.

[4] "Archive of formal proofs." [Online]. Available: https://www.isa-afp.org/

[5] J. P. Bridge, S. B. Holden, and L. C. Paulson, "Machine learning for first-order theorem proving," *Journal of Automated Reasoning*, vol. 53, no. 2, pp. 141–172, Aug 2014. [Online]. Available: https://doi.org/10.1007/s10817-014-9301-5

[6] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy, and S. Zanella-Béguelin, "Formal verification of smart contracts: Short paper," in *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, ser. PLAS '16. New York, NY, USA: ACM, 2016, pp. 91–96. [Online]. Available: http://doi.acm.org/10.1145/2993600.2993611

[7] V. Buterin, "Critical update re: Dao vulnerability," 2016. [Online]. Available: https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability

[8] J. Wolfers and E. Zitzewitz, "Prediction markets," *Journal of Economic Perspectives*, vol. 18, no. 2, pp. 107–126, June 2004. [Online]. Available: http://www.aeaweb.org/articles?id=10.1257/0895330041371321

[9] J. Peterson and J. Krug, "Augur: a decentralized, open-source platform for prediction markets," *CoRR*, vol. abs/1501.01042, 2015. [Online]. Available: http://arxiv.org/abs/1501.01042

[10] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[11] V. Buterin, "A next generation smart con-tract and decentralized application platform," 2013. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[12] K. Ciesielski, *Set Theory for the Working Mathematician.* Cambridge University Press, 1997.

[13] K. Saito and M. Iwamura, "How to make a digital currency on a blockchain stable," 2018. [Online]. Available: https://arxiv.org/abs/1801.06771