Fate Protocol: Perpetual Prediction Pools

Zahnentferner
The Stable Order

Anjali Jha
The Stable Order

Raj Jitendra Shah
The Stable Order

Abstract

This paper introduces the Fate Protocol, an approach to prediction markets that is novel in two ways: firstly, it uses the new concept of *prediction pools* instead of traditional order books; and, secondly, it is *perpetual*, with no expiration dates.

2012 ACM Subject Classification Applied computing → Economics; Theory of computation → Algorithmic mechanism design

Keywords and phrases blockchain, prediction markets, liquidity pools, decentralized finance

Acknowledgements We thank the community of the Stability Nexus for their feedback and support.

1 Introduction

Traditional prediction markets have fixed expiration dates and an order book approach that may suffer from insufficient liquidity. The Fate Protocol addresses these issues by introducing *perpetual prediction pools* that enable investing in and hedging against the future values of any variable that is a function of time (e.g. asset prices, weather variables, ...).

The Fate Protocol allows users to buy and sell *bullCoins* and *bearCoins*, depending on whether they believe that the value of the tracked variable will continue to go up or down, respectively. These coins are backed by reserves of the asset deposited by users who buy the coins. When the tracked variable goes up, a portion of the reserve backing the bearCoins is redistributed to the reserve backing the bullCoins. Conversely, when the tracked variable goes down, a portion of the reserve backing the bullCoins is redistributed to the reserve backing the bearCoins. A core innovation of the protocol lies in its path-independent redistribution mechanism, which ensures that final reserves depend solely on the initial and final values of the variable, regardless of its intermediate values. This mathematical property, combined with a *zero-sum fee structure*, creates sustainable liquidity pools without traditional order book limitations.

The zero-sum fee structure refers to the property that fees collected from one operation benefit participants on the opposite side of the market: fees from bullCoin redemptions are added to the bearCoin reserve, and vice versa. This creates a closed economic system where no value leaves the protocol as external fees, and long-term participants benefit from the trading activity of short-term participants.

This paper presents the protocol's mathematical foundations and mechanics, and proves the above-mentioned path independence property.

2 Protocol Description

The Fate Protocol allows users to mint and burn two dual *prediction tokens*: *bullCoins* ($B^{\uparrow}$) that appreciate when the tracked variable increases, or *bearCoins* ($B^{\downarrow}$) that appreciate when

the variable decreases. Each token type is backed by dedicated reserves of a base fungible asset: $R^{\uparrow}$ and $R^{\downarrow}$, respectively. The protocol has three fundamental operations:

- **Token Purchase:** Users mint an amount of a prediction token of their choice by depositing an amount of the base asset into the corresponding reserve.
- **Token Redemption:** Users burn an amount of a prediction token to withdraw a proportional amount of the base asset from the respective reserve.
- **Variable Update:** Users may call an external oracle to update the variable value and redistribute a portion of the losing token's reserve to the winning token's reserve. Furthermore, a variable update is always called immediately before any token purchase or redemption, to ensure that purchase or redemption occurs with a token price that reflects that latest variable value.

3 Protocol Specification

The protocol's state is a tuple $(R^{\uparrow}, R^{\downarrow}, S^{\uparrow}, S^{\downarrow}, p)$, where:

- $R^{\uparrow}$ and $R^{\downarrow}$ are the bullCoin and bearCoin reserves.
- $S^{\uparrow}$ and $S^{\downarrow}$ are the bullCoin and bearCoin supplies.
- p is the variable value.

The protocol has a single fixed parameter: the fee ϕ, where $0 \leq \phi < 1$.

3.1 Protocol Initialization

Before any operations can occur, the protocol must be initialized with a valid starting state. The initialization requires:

- Initial reserves: $R_0^{\uparrow} > 0$ and $R_0^{\downarrow} > 0$
- Initial supplies: $S_0^{\uparrow} > 0$ and $S_0^{\downarrow} > 0$
- Initial variable value: $p_0 > 0$

These constraints ensure that initial token prices are well-defined and positive:

$$P_0^{\uparrow} = \frac{R_0^{\uparrow}}{S_0^{\uparrow}} > 0 \quad \text{and} \quad P_0^{\downarrow} = \frac{R_0^{\downarrow}}{S_0^{\downarrow}} > 0$$

A common initialization approach is to set equal reserves and supplies (e.g., $R_0^{\uparrow} = R_0^{\downarrow} = 1$ and $S_0^{\uparrow} = S_0^{\downarrow} = 1$), resulting in both token prices starting at 1. However, any positive values satisfying the above constraints are valid.

3.1.0.1 Invariant.

Throughout the protocol's operation, it is required that both reserves remain strictly positive: $R^{\uparrow} > 0$ and $R^{\downarrow} > 0$. Similarly, all variable values must be positive: $p > 0$. These invariants are preserved by the protocol operations under normal conditions (see Section 6 for edge case analysis). The prices of bullCoins and bearCoins are defined by $P^{\uparrow} = R^{\uparrow}/S^{\uparrow}$ and $P^{\downarrow} = R^{\downarrow}/S^{\downarrow}$, respectively. The *total reserve* is defined by $R = R^{\uparrow} + R^{\downarrow}$.

The following subsections mathematically define the behavior of each protocol operation.

3.2 Token Purchase

For a deposit of an amount n of the base asset, the protocol calculates the amount m of prediction tokens to be received and updates the state as follows.

For bullCoin purchases:

$$m^{\uparrow} = \frac{(1-\phi)n}{P^{\uparrow}} \tag{1}$$
$$R^{\uparrow}_{\text{new}} \leftarrow R^{\uparrow} + n \tag{2}$$
$$S^{\uparrow}_{\text{new}} \leftarrow S^{\uparrow} + m^{\uparrow} \tag{3}$$

For bearCoin purchases:

$$m^{\downarrow} = \frac{(1-\phi)n}{P^{\downarrow}} \tag{4}$$
$$R^{\downarrow}_{\text{new}} \leftarrow R^{\downarrow} + n \tag{5}$$
$$S^{\downarrow}_{\text{new}} \leftarrow S^{\downarrow} + m^{\downarrow} \tag{6}$$

3.3 Token Redemption

For a redemption of n prediction tokens, the protocol calculates the amount q of base asset to be received and updates the state as follows:

For bullCoin redemptions:

$$z = n \cdot P^{\uparrow} \tag{7}$$
$$m = (1-\phi)z \tag{8}$$
$$R^{\uparrow}_{\text{new}} \leftarrow R^{\uparrow} - z \tag{9}$$
$$R^{\downarrow}_{\text{new}} \leftarrow R^{\downarrow} + \phi z \tag{10}$$
$$S^{\uparrow}_{\text{new}} \leftarrow S^{\uparrow} - n \tag{11}$$

For bearCoin redemptions:

$$z = n \cdot P^{\downarrow} \tag{12}$$
$$m = (1-\phi)z \tag{13}$$
$$R^{\downarrow}_{\text{new}} \leftarrow R^{\downarrow} - z \tag{14}$$
$$R^{\uparrow}_{\text{new}} \leftarrow R^{\uparrow} + \phi z \tag{15}$$
$$S^{\downarrow}_{\text{new}} \leftarrow S^{\downarrow} - n \tag{16}$$

3.4 Variable Update

When time passes from t to $t+1$ and the tracked variable changes from p_t to p_{t+1}, the reserves are updated using the following two-step algorithm:

Step 1: Update

$$Q^{\uparrow}_{t+1} = R^{\uparrow}_{t} \cdot \frac{p_{t+1}}{p_t} \tag{17}$$
$$Q^{\downarrow}_{t+1} = R^{\downarrow}_{t} \cdot \frac{p_t}{p_{t+1}} \tag{18}$$

Step 2: Normalize

$$R^{\uparrow}_{t+1} = \frac{Q^{\uparrow}_{t+1}}{Q^{\uparrow}_{t+1} + Q^{\downarrow}_{t+1}} \cdot R_t \tag{19}$$

$$R^{\downarrow}_{t+1} = \frac{Q^{\downarrow}_{t+1}}{Q^{\uparrow}_{t+1} + Q^{\downarrow}_{t+1}} \cdot R_t \tag{20}$$

In the equations above, Step 1 computes intermediate values Q, which are temporary rescaled reserve values based on relative price changes. These serve as a transitional step used to capture how the reserves would shift if left unbounded. Step 2 then normalizes these intermediate values to ensure the updated reserves remain within the total constraint of the original reserve R_t. This prevents the system from deviating from its initial total reserve while still adjusting proportionally to price dynamics.

This two-step process can be simplified to a single step process by combining equation 19 with equation 17 and equation 20 with equation 18, resulting in the following equations:

$$R^{\uparrow}_{t+1} = R_t \cdot \frac{R^{\uparrow}_t \cdot (p_{t+1}/p_t)}{R^{\uparrow}_t \cdot (p_{t+1}/p_t) + R^{\downarrow}_t \cdot (p_t/p_{t+1})} \tag{21}$$

$$R^{\downarrow}_{t+1} = R_t \cdot \frac{R^{\downarrow}_t \cdot (p_t/p_{t+1})}{R^{\uparrow}_t \cdot (p_{t+1}/p_t) + R^{\downarrow}_t \cdot (p_t/p_{t+1})} \tag{22}$$

4 Example

To illustrate the Fate Protocol in action, consider a scenario where the tracked variable is the price of Bitcoin (BTC), and Ethereum (ETH) is the base reserve asset. Suppose at time $t = 0$, the BTC price is $p_0 = 100$ (in some unit), and the protocol is initialized with equal reserves and token supplies:

- Bull reserve: $R^{\uparrow} = 1$ ETH, $S^{\uparrow} = 1$ bullCoins
- Bear reserve: $R^{\downarrow} = 1$ ETH, $S^{\downarrow} = 1$ bearCoins

Each bullCoin and bearCoin is thus initially worth 1 ETH.

Alice buys bullCoins.

Alice deposits 50 ETH into the bull reserve. Assuming a protocol fee of $\phi = 1\%$, she receives:

$$m^{\uparrow} = \frac{(1-\phi)\cdot 50}{P^{\uparrow}} = \frac{0.99 \cdot 50}{1} = 49.5$$

bullCoins. The bull reserve and supply update to:

$$R^{\uparrow} = 1 + 50 = 51 \text{ ETH}, \quad S^{\uparrow} = 1 + 49.5 = 50.5 \text{ bullCoins}$$

Bob buys bearCoins.

Bob deposits 50 ETH into the bear reserve and receives 49.5 bearCoins under the same fee and price conditions. The bear reserve and supply update to:

$$R^{\downarrow} = 1 + 50 = 51 \text{ ETH}, \quad S^{\downarrow} = 1 + 49.5 = 50.5 \text{ bearCoins}$$

Price goes up.

The oracle updates the BTC price to $p_1 = 110$. According to the price update mechanism:

$$Q^{\uparrow} = 51 \cdot \frac{110}{100} = 56.1, \quad Q^{\downarrow} = 51 \cdot \frac{100}{110} \approx 46.36$$

These are normalized to maintain the total reserve of 102 ETH:

$$R^{\uparrow}_{\text{new}} = 102 \cdot \frac{56.1}{56.1 + 46.36} \approx 55.83$$

$$R^{\downarrow}_{\text{new}} = 102 - 55.83 = 46.17$$

New prices:

$$P^{\uparrow} = \frac{55.83}{50.5} \approx 1.105, \quad P^{\downarrow} = \frac{46.17}{50.5} \approx 0.914$$

Alice's potential profit (hypothetical).

Alice holds 49.5 bullCoins. If she were to redeem them at this point, she would receive:

$$(1 - \phi) \cdot 49.5 \cdot 1.105 \approx 0.99 \cdot 54.72 \approx 54.17 \text{ ETH}$$

This represents a profit of approximately 4.17 ETH over her original 50 ETH deposit, reflecting the upward price movement. However, let us assume Alice decides to hold her position.

Further price movement.

Suppose the BTC price later falls to $p_2 = 90$. Since neither Alice nor Bob have redeemed, the reserves are still $R^{\uparrow} = 55.83$ ETH, $R^{\downarrow} = 46.17$ ETH, with total reserve $R = 102$ ETH. The reserves update according to the new price:

$$Q^{\uparrow} = 55.83 \cdot \frac{90}{110} \approx 45.68, \quad Q^{\downarrow} = 46.17 \cdot \frac{110}{90} \approx 56.43$$

Normalize:

$$R^{\uparrow}_{\text{new}} = 102 \cdot \frac{45.68}{45.68 + 56.43} \approx 45.63, \quad R^{\downarrow}_{\text{new}} = 102 - 45.63 \approx 56.37$$

New token prices:

$$P^{\uparrow} \approx \frac{45.63}{50.5} \approx 0.904, \quad P^{\downarrow} \approx \frac{56.37}{50.5} \approx 1.116$$

Now bearCoin holders like Bob are in profit, while bullCoin holders like Alice would incur a loss if redeeming at this point.

Observation.

This example shows how users interact with the protocol by purchasing or redeeming bull/bearCoins, and how oracle-driven price updates affect token values and reserve balances. BullCoin holders benefit from upward price movements, while BearCoin holders profit when prices fall. By Theorem 4, the final state depends only on the start and end prices, not the intermediate path taken.

4.1 Numerical Analysis of Prediction Adequacy

To better understand how the protocol rewards correct predictions, consider the relationship between price movements and token returns. For a fee ϕ and a price movement of $m\%$ (where $m > 0$ indicates an increase), we analyze the value received upon redemption relative to the initial deposit.

4.1.0.1 Setup.

Assume balanced initial reserves ($R_0^{\uparrow} = R_0^{\downarrow} = R_0/2$) and a user who deposits 1 unit of the base asset to purchase prediction tokens. After the fee, the user receives $(1 - \phi)$ tokens (since initial price is 1).

4.1.0.2 After price movement.

Let $r = (1 + m/100)$ be the price ratio. From the path-independence formula, the new reserves are:

$$R_t^{\uparrow} = R_0 \cdot \frac{r}{r + 1/r} = R_0 \cdot \frac{r^2}{r^2 + 1} \tag{23}$$

For bullCoin holders, the new token price is:

$$P_t^{\uparrow} = \frac{R_t^{\uparrow}}{S^{\uparrow}} = \frac{R_0 \cdot r^2/(r^2 + 1)}{S^{\uparrow}} \tag{24}$$

4.1.0.3 Example returns.

The following table shows approximate net returns (after fees) for bullCoin purchases with fee $\phi = 1\%$ and various price movements:

Price Change	BullCoin Return	BearCoin Return
+50%	+29.6%	-31.0%
+20%	+15.4%	-17.3%
+10%	+8.5%	-10.3%
+5%	+4.6%	-6.4%
0%	-1.0%	-1.0%
-5%	-6.4%	+4.6%
-10%	-10.3%	+8.5%
-20%	-17.3%	+15.4%
-50%	-31.0%	+29.6%

4.1.0.4 Observations.

- Returns are symmetric: bullCoin returns for a $+m\%$ move equal bearCoin returns for a $-m\%$ move.
- The fee creates a small drag: even with no price movement, holders lose the fee upon redemption.
- Returns are sublinear: a 50% price increase yields only approximately 30% profit, due to the redistribution mechanism sharing gains with the opposing reserve.
- Correct predictions are always rewarded: any positive price movement benefits bullCoin holders, and any negative movement benefits bearCoin holders.

5 Theorems

The Fate protocol enjoys four important theorems. The first one shows that variable updates preserve the total reserve. This is important, since it shows that base assets will neither disappear (be burned) nor appear from nothing (be minted) as a result of variable updates.

▶ **Theorem 1** (Reserve Preservation). *If a variable update action occurs at time t, then $R_{t+1} = R_t$.*

Proof. The variable update at time t consists of two steps:

Step 1 (Update):

$$Q^{\uparrow}_{t+1} = R^{\uparrow}_t \cdot \frac{p_{t+1}}{p_t},$$
$$Q^{\downarrow}_{t+1} = R^{\downarrow}_t \cdot \frac{p_t}{p_{t+1}}.$$

Step 2 (Normalization):

$$R^{\uparrow}_{t+1} = R_t \cdot \frac{Q^{\uparrow}_{t+1}}{Q^{\uparrow}_{t+1} + Q^{\downarrow}_{t+1}},$$
$$R^{\downarrow}_{t+1} = R_t \cdot \frac{Q^{\downarrow}_{t+1}}{Q^{\uparrow}_{t+1} + Q^{\downarrow}_{t+1}}.$$

Summing the two normalized reserves yields:

$$\begin{aligned} R_{t+1} &= R^{\uparrow}_{t+1} + R^{\downarrow}_{t+1} \\ &= R_t \cdot \left(\frac{Q^{\uparrow}_{t+1}}{Q^{\uparrow}_{t+1} + Q^{\downarrow}_{t+1}} + \frac{Q^{\downarrow}_{t+1}}{Q^{\uparrow}_{t+1} + Q^{\downarrow}_{t+1}} \right) \\ &= R_t \cdot \frac{Q^{\uparrow}_{t+1} + Q^{\downarrow}_{t+1}}{Q^{\uparrow}_{t+1} + Q^{\downarrow}_{t+1}} \\ &= R_t. \end{aligned}$$

Thus, the total reserve remains unchanged by the variable update step:

$$R_{t+1} = R_t.$$

◀

The second and third theorems show that Fate behaves as a prediction market, rewarding bullCoin holders when the variable increases and rewarding bearCoin holders when the variable decreases. This happens indirectly, because the prices increase. To realize these gains, holders must redeem their prediction tokens.

▶ **Theorem 2** (Upward Prediction Adequacy). *If a variable update action occurs at time t and $p_{t+1} > p_t$, then $P^{\uparrow}_{t+1} > P^{\uparrow}_t$ and $P^{\downarrow}_{t+1} < P^{\downarrow}_t$.*

Proof. Assume $p_{t+1} > p_t$, which implies $\frac{p_{t+1}}{p_t} > 1$ and $\frac{p_t}{p_{t+1}} < 1$.

From the variable update mechanism, we have:

Step 1 (Update):

$$Q^{\uparrow}_{t+1} = R^{\uparrow}_t \cdot \frac{p_{t+1}}{p_t} \tag{25}$$
$$Q^{\downarrow}_{t+1} = R^{\downarrow}_t \cdot \frac{p_t}{p_{t+1}} \tag{26}$$

Step 2 (Normalization):

$$R^{\uparrow}_{t+1} = R_t \cdot \frac{Q^{\uparrow}_{t+1}}{Q^{\uparrow}_{t+1} + Q^{\downarrow}_{t+1}} \tag{27}$$

$$= R_t \cdot \frac{R^{\uparrow}_t \cdot \frac{p_{t+1}}{p_t}}{R^{\uparrow}_t \cdot \frac{p_{t+1}}{p_t} + R^{\downarrow}_t \cdot \frac{p_t}{p_{t+1}}} \tag{28}$$

$$= R_t \cdot \frac{R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t}}{R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} + R^{\downarrow}_t} \tag{29}$$

Similarly:

$$R^{\downarrow}_{t+1} = R_t \cdot \frac{R^{\downarrow}_t}{R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} + R^{\downarrow}_t} \tag{30}$$

To prove $P^{\uparrow}_{t+1} > P^{\uparrow}_t$, we need to show $R^{\uparrow}_{t+1} > R^{\uparrow}_t$ (since token supplies are unchanged during variable updates).

We need to verify:

$$R_t \cdot \frac{R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t}}{R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} + R^{\downarrow}_t} > R^{\uparrow}_t \tag{31}$$

Since $R_t = R^{\uparrow}_t + R^{\downarrow}_t$, this becomes:

$$(R^{\uparrow}_t + R^{\downarrow}_t) \cdot \frac{R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t}}{R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} + R^{\downarrow}_t} > R^{\uparrow}_t \tag{32}$$

Cross-multiplying (all terms are positive):

$$(R^{\uparrow}_t + R^{\downarrow}_t) \cdot R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} > R^{\uparrow}_t \cdot \left(R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} + R^{\downarrow}_t \right) \tag{33}$$

$$R^{\uparrow}_t \cdot R^{\downarrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} > R^{\uparrow}_t \cdot R^{\downarrow}_t \tag{34}$$

$$\frac{p^2_{t+1}}{p^2_t} > 1 \tag{35}$$

Since $p_{t+1} > p_t$, we have $\frac{p^2_{t+1}}{p^2_t} > 1$, so $P^{\uparrow}_{t+1} > P^{\uparrow}_t$.

For bearCoins, we need to show $R^{\downarrow}_{t+1} < R^{\downarrow}_t$:

$$(R^{\uparrow}_t + R^{\downarrow}_t) \cdot \frac{R^{\downarrow}_t}{R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} + R^{\downarrow}_t} < R^{\downarrow}_t \tag{36}$$

Cross-multiplying:

$$(R^{\uparrow}_t + R^{\downarrow}_t) \cdot R^{\downarrow}_t < R^{\downarrow}_t \cdot \left(R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} + R^{\downarrow}_t \right) \tag{37}$$

$$R^{\uparrow}_t \cdot R^{\downarrow}_t < R^{\downarrow}_t \cdot R^{\uparrow}_t \cdot \frac{p^2_{t+1}}{p^2_t} \tag{38}$$

$$1 < \frac{p^2_{t+1}}{p^2_t} \tag{39}$$

Since $p_{t+1} > p_t$, we have $\frac{p_{t+1}^2}{p_t^2} > 1$, so $P_{t+1}^{\downarrow} < P_t^{\downarrow}$.
Therefore, when $p_{t+1} > p_t$, bullCoin prices increase and bearCoin prices decrease. ◀

▶ **Theorem 3** (Downward Prediction Adequacy)**.** *If a variable update action occurs at time t and $p_{t+1} < p_t$, then $P_{t+1}^{\uparrow} < P_t^{\uparrow}$ and $P_{t+1}^{\downarrow} > P_t^{\downarrow}$.*

Proof. Analogous and dual to the proof of Theorem 2 ◀

Finally, the last theorem shows *path independence*: in a sequence of variable updates, the outcome depends exclusively on the initial protocol state, the initial variable value, and the final variable value, rendering intermediate variable movements irrelevant. This property ensures that the final outcome is independent of when variable updates and oracle consultations are triggered. It ensures that participants cannot try to earn more simply by calling variable update at specific intermediary times.

▶ **Theorem 4** (Path Independence)**.** *Consider a variable value path:*

$$p_0 \to p_1 \to \cdots \to p_t \tag{40}$$

where only variable updates occur (no token purchases or redemptions). Then the final reserves $R_t^{\uparrow}$ and $R_t^{\downarrow}$ depend only on:
- *Initial protocol state $(R_0^{\uparrow}, R_0^{\downarrow}, S_0^{\uparrow}, S_0^{\downarrow}, p_0)$,*
- *Final variable value p_t.*

Intermediate variable values $p_1, \ldots, p_{t-1}$ have no effect on the final reserves, and the token supplies remain unchanged: $S_t^{\uparrow} = S_0^{\uparrow}$ and $S_t^{\downarrow} = S_0^{\downarrow}$.

Proof. Since variable updates only affect reserves $(R^{\uparrow}, R^{\downarrow})$ and the variable value p, while preserving supplies $(S^{\uparrow}, S^{\downarrow})$, we focus on the reserve transformations.

By recursively applying equations (21) and (22) to unfold definitions of $R_k^{\uparrow}$ and $R_k^{\downarrow}$ for all k and simplifying the resulting expanded equations, we obtain:

$$R_t^{\uparrow} = R_0 \cdot \frac{R_0^{\uparrow} \cdot \prod_{i=1}^{t} \frac{p_i}{p_{i-1}}}{R_0^{\uparrow} \cdot \prod_{i=1}^{t} \frac{p_i}{p_{i-1}} + R_0^{\downarrow} \cdot \prod_{i=1}^{t} \frac{p_{i-1}}{p_i}} \tag{41}$$

$$= R_0 \cdot \frac{R_0^{\uparrow} \cdot (p_t/p_0)}{R_0^{\uparrow} \cdot (p_t/p_0) + R_0^{\downarrow} \cdot (p_0/p_t)} \tag{42}$$

Since $\prod_{i=1}^{t} \frac{p_i}{p_{i-1}} = \frac{p_t}{p_0}$, the final reserves depend only on the initial state components $R_0^{\uparrow}$, $R_0^{\downarrow}$, p_0, and the final variable value p_t, regardless of the intermediate values $p_1, \ldots, p_{t-1}$.

Therefore, any path from initial state $(R_0^{\uparrow}, R_0^{\downarrow}, S_0^{\uparrow}, S_0^{\downarrow}, p_0)$ to final variable value p_t yield the same final state:

$$R_t^{\uparrow} = R_0 \cdot \frac{R_0^{\uparrow} \cdot (p_t/p_0)}{R_0^{\uparrow} \cdot (p_t/p_0) + R_0^{\downarrow} \cdot (p_0/p_t)} \tag{43}$$

$$R_t^{\downarrow} = R_0 \cdot \frac{R_0^{\downarrow} \cdot (p_0/p_t)}{R_0^{\uparrow} \cdot (p_t/p_0) + R_0^{\downarrow} \cdot (p_0/p_t)} \tag{44}$$

$$S_t^{\uparrow} = S_0^{\uparrow} \tag{45}$$

$$S_t^{\downarrow} = S_0^{\downarrow} \tag{46}$$

$$p_t = p_t \tag{47}$$

◀

6 Edge Cases and Reserve Depletion

A natural concern arises regarding the behavior of the protocol under extreme price movements: could a sustained directional move cause one reserve to approach zero, leading to insolvency?

6.0.0.1 Reserve Asymptotic Behavior.

Consider a scenario where the tracked variable increases dramatically, i.e., $p_t \gg p_0$. From Theorem 4, we have:

$$R_t^{\downarrow} = R_0 \cdot \frac{R_0^{\downarrow} \cdot (p_0/p_t)}{R_0^{\uparrow} \cdot (p_t/p_0) + R_0^{\downarrow} \cdot (p_0/p_t)} \tag{48}$$

As $p_t \to \infty$, the term $(p_0/p_t) \to 0$ while $(p_t/p_0) \to \infty$, causing $R_t^{\downarrow} \to 0$. Similarly, if $p_t \to 0$, then $R_t^{\uparrow} \to 0$. However, neither reserve ever becomes exactly zero for any finite, positive price value. The reserves approach zero asymptotically but never reach it.

6.0.0.2 Practical Implications.

In practice, extreme price movements may cause one reserve to become very small, which has the following implications:

1. **Reduced liquidity:** As a reserve shrinks, the available liquidity for redemptions decreases.
2. **Price sensitivity:** Small reserves amplify the impact of new deposits or withdrawals on token prices.
3. **Rounding errors:** In implementations with finite precision, extremely small reserves may encounter numerical issues.

6.0.0.3 Mitigation Strategies.

Implementations may consider the following safeguards:

- **Minimum reserve thresholds:** Reject operations that would reduce a reserve below a minimum value.
- **Circuit breakers:** Temporarily halt trading if reserves become too imbalanced.
- **Reserve ratio limits:** Impose maximum imbalance ratios (e.g., $R^{\uparrow}/R^{\downarrow} < 100$).

For the formal theorems in Section 5, we assume that both reserves remain strictly positive ($R^{\uparrow} > 0$ and $R^{\downarrow} > 0$) and that all prices are positive ($p > 0$). Under these assumptions, the theorems hold.

7 Oracle Considerations

The Fate Protocol relies on an external oracle to provide the tracked variable's value. This dependency introduces potential vulnerabilities that implementations must address.

7.0.0.1 Oracle Failure Modes.

1. **Oracle unavailability:** If the oracle becomes temporarily unavailable, variable updates cannot occur. This freezes the protocol state at the last known variable value.

2. **Stale data:** Oracles may report outdated values due to network latency or infrequent updates. The protocol should include mechanisms to detect and handle stale data, such as maximum age thresholds for oracle values.
3. **Oracle manipulation:** Malicious actors may attempt to manipulate oracle values through flash loan attacks, sandwich attacks, or by exploiting thin markets.

7.0.0.2 Fallback Mechanisms.

Robust implementations should consider:

- **Multi-oracle aggregation:** Using multiple independent oracles and taking a median or weighted average to reduce manipulation risk.
- **Time-weighted average prices (TWAP):** Smoothing price feeds over time windows to reduce the impact of short-term manipulation.
- **Heartbeat checks:** Verifying that oracle updates occur within expected time intervals and pausing operations if updates are too infrequent.
- **Deviation bounds:** Rejecting oracle updates that deviate too dramatically from recent values (e.g., more than 50% change in a single update).

7.0.0.3 MEV and OEV Considerations.

The protocol's requirement that "a variable update is always called immediately before any token purchase or redemption" creates potential for Maximal Extractable Value (MEV) and Oracle Extractable Value (OEV) exploitation:

- **Front-running:** Miners or validators who observe pending transactions may insert their own transactions before user transactions to profit from anticipated price movements.
- **Sandwich attacks:** An attacker places transactions both before and after a victim's transaction to extract value from the price impact.

Mitigation approaches include:

- Using commit-reveal schemes for large transactions
- Implementing time-locks or batching mechanisms
- Employing private transaction pools or encrypted mempools
- Setting maximum slippage tolerances for users

8 Related Work

8.0.0.1 Prediction Market Mechanisms.

Traditional prediction markets have employed various mechanisms for price discovery and liquidity provision. The Logarithmic Market Scoring Rule (LMSR) [3, 4] is a prominent automated market maker for prediction markets that provides bounded loss guarantees for the market maker while ensuring liquidity at all price levels. In LMSR, prices are determined by a scoring function based on outstanding shares, and the market maker subsidizes liquidity.

The Fate Protocol differs from LMSR in several key aspects:

- **No market maker subsidy:** In Fate, all liquidity comes from participant deposits, whereas LMSR requires the market maker to provide initial capital and absorb potential losses.
- **Perpetual operation:** Fate has no expiration date, while traditional LMSR markets resolve at a predetermined time.
- **Price mechanism:** Fate derives prices from reserve ratios with oracle-driven redistribution, whereas LMSR uses a logarithmic cost function.

8.0.0.2 Constant Function Market Makers (CFMMs).

Decentralized exchanges such as Uniswap [6, 1] and Balancer [7] popularized Constant Function Market Makers (CFMMs), which derive prices implicitly from reserve ratios through invariant functions (e.g., $x \cdot y = k$ for constant product markets).

The Fate Protocol shares the pool-based architecture of CFMMs but differs fundamentally:

- **Oracle-driven vs. implicit pricing:** CFMMs derive prices from arbitrage with external markets; Fate explicitly uses an oracle to determine the tracked variable's value.
- **Reserve redistribution:** In CFMMs, reserves change only through swaps; in Fate, oracle updates trigger reserve redistribution between pools.
- **Purpose:** CFMMs facilitate token exchange; Fate provides exposure to price movements without requiring underlying asset ownership.

8.0.0.3 Crypto-Backed Stablecoins.

The Fate Prediction Pool Protocol is inspired by the hodlCoin Staking Protocol [5], which in turn evolved from the Djed Stablecoin Protocol [10, 9]. In hodlCoin, there is a single reserve for a single token, and there is only a fee for redemptions. Fate generalizes this to two reserves for two tokens, introduces a fee for purchases, and changes the beneficiary of the redemption fee (making it be the dual token's reserve instead of the reserve of the token being redeemed). Finally, most crucially, it introduces the variable update action, which makes the tokens, and their reserves, compete with each other, depending on the evolution of the value of an external oracle variable.

8.0.0.4 Liquidity Pool Innovation.

In the same way that liquidity pools [6, 2] introduced the concept of pools to overcome scalability, liquidity and cost limitations of order books in the context of on-chain decentralized exchanges, the Fate Protocol introduces the concept of prediction pools to overcome similar limitations in the context of on-chain prediction markets.

9 Discussion and Future Work

Due to its simplicity, adequacy and generality, the Fate Protocol has the potential to become a new composable primitive in decentralized finance. Furthermore, since any asset can be used as a base asset for prediction pools, Fate can bring more utility for such assets, increasing their total value locked (TVL) and reducing their circulating supply.

Fate's main intended use case is as a perpetual prediction market for any variable with values that change over time. As such, Fate can be used to manage and insure against risks related to weather data (e.g. temperature, precipitation, ...), asset prices, macro-economic indicators, ... Particularly, Fate can potentially capture one key use case for decentralized stablecoins such as Djed [10, 9] and Gluon [8]: holders of crypto-backed stablecoins often buy stablecoins with the underlying backing crypto asset to hedge against the risk that the price of this asset will fall; with Fate, they could achieve a similar effect by purchasing bearCoins associated with the price of this asset.

9.0.0.1 De-pegging Alert System.

An interesting application of Fate is as an early warning system for bridged or pegged assets. By tracking the price ratio between a bridged token and its canonical version, Fate pools

could provide:

- Real-time sentiment signals about bridge security
- Hedging mechanisms for bridge users
- Economic incentives for monitoring and reporting de-pegging events

9.0.0.2 Fate as an Oracle.

While Fate currently depends on external oracles, an intriguing possibility is for Fate to serve as an oracle itself. The rationale is as follows: the token prices $P^{\uparrow}$ and $P^{\downarrow}$ reflect market participants' aggregate beliefs about the tracked variable's future direction. The ratio $P^{\uparrow}/P^{\downarrow}$ could be interpreted as a sentiment indicator, where values greater than 1 suggest bullish sentiment and values less than 1 suggest bearish sentiment.

Furthermore, if multiple Fate pools track the same variable with different base assets or configurations, cross-referencing their states could provide robust, manipulation-resistant price signals. This approach would create a decentralized oracle mechanism where the oracle's accuracy is backed by the economic stakes of the participants. However, careful analysis of potential manipulation vectors and game-theoretic incentives is required before such an application could be deployed safely.

9.0.0.3 Future Work.

In our current and planned future work, we are investigating other variable update mechanisms, besides the one described in Subsection 3.4. This is likely going to lead to a family of Fate protocols. Additionally, we plan to:

- Analyze the game-theoretic properties of the fee structure and its impact on participant behavior
- Investigate optimal fee parameters for different market conditions
- Develop formal security models for oracle integration
- Explore integration with existing DeFi protocols for composability

References

1 Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core, 2021. `https://uniswap.org/whitepaper-v3.pdf`.

2 Vitalik Buterin. Path independent market makers, 2017. `https://vitalik.eth.limo/general/2017/06/22/marketmakers.html`.

3 Robin Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):107–119, 2003.

4 Robin Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *The Journal of Prediction Markets*, 1(1):3–15, 2007.

5 Djed Alliance Joachim Zahnentferner. hodlCoin: A financial game. 2023. URL: `https://eprint.iacr.org/2023/1029`.

6 Uniswap Labs. What is an automated market maker?, 2020. `https://blog.uniswap.org/what-is-an-automated-market-maker`.

7 Fernando Martinelli and Nikolai Mushegian. Balancer whitepaper, 2019. `https://balancer.fi/whitepaper.pdf`.

8 Stability Nexus. Gluon protocol overview, 2024. `https://docs.stability.nexus/gluon-protocols/gluon-overview`.

9 Joachim Zahnentferner and Dmytro Kaidalov. Formal verification of the djed stablecoin protocol. In *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2023.

10 Joachim Zahnentferner, Dmytro Kaidalov, Jean-Frédéric Etienne, and Javier Díaz. Djed: A formally verified crypto-backed pegged algorithmic stablecoin. 2021. URL: https://eprint.iacr.org/2021/1069, doi:10.1109/ICBC56567.2023.10174901.

A License

B Official Implementations and Deployments of Fate Protocol

https://github.com/StabilityNexus/

C Official Fate Protocol Channels

- https://x.com/StabilityNexus
- https://discord.gg/7jS9qJNjJv

[1] https://creativecommons.org/licenses/by-nc-nd/4.0/