

# A theory of Automated Market Makers in DeFi

Massimo Bartoletti<sup>1</sup>[0000-0003-3796-9774],  
James Hsin-yu Chiang<sup>2</sup>[0000-0002-5126-9494], and  
Alberto Lluch-Lafuente<sup>2</sup>[0000-0001-7405-0818]

<sup>1</sup> Università degli Studi di Cagliari, Cagliari, Italy [bart@unica.it](mailto:bart@unica.it)

<sup>2</sup> Technical University of Denmark, DTU Compute, Copenhagen, Denmark  
{[jchi](mailto:jchi@dtu.dk),[albl](mailto:albl@dtu.dk)}@dtu.dk

**Abstract.** Automated market makers (AMMs) are one of the most prominent decentralized finance (DeFi) applications. They allow users to exchange units of different types of crypto-assets, without the need to find a counter-party. There are several implementations and models for AMMs, featuring a variety of sophisticated economic mechanisms. We present a theory of AMMs. The core of our theory is an abstract operational model of the interactions between users and AMMs, which can be concretised by instantiating the economic mechanisms. We exploit our theory to formally prove a set of fundamental properties of AMMs, characterizing both structural and economic aspects. We do this by abstracting from the actual economic mechanisms used in implementations and identifying sufficient conditions which ensure the relevant properties. Notably, we devise a general solution to the *arbitrage problem*, the main game-theoretic foundation behind the economic mechanisms of AMMs.

## 1 Introduction

Decentralized finance (DeFi) is emerging as an alternative to the traditional finance, boosted by blockchain-based crypto-tokens and smart contracts. One of the main DeFi applications are *Automated Market Makers (AMMs)*, which allow users to exchange crypto-tokens of different types without the intermediation of third parties. As of April 2021, the two AMM platforms leading by user activity, Uniswap [14] and Curve Finance [5], alone hold \$8.1B and \$6.2B worth of tokens, and process \$1.5B and \$210M worth of transactions daily [4, 12].

AMMs are inherently hard to design, implement and understand, since they involve sophisticated economic incentive mechanisms. Although they generally only expose a handful of callable functions, interactions with AMMs are sensitive to transaction ordering [21, 23, 27, 30]: thus, actors with the power to influence the order of transactions in the blockchain may be incentivized to do so for profit or to harm specific users. Thus, there exists a need for foundational work to devise formal models of AMMs which allow the study of their fundamental properties, transaction concurrency and the effect of economic incentives.

Current descriptions of AMMs are either economic models [17–19, 24], which focus on the efficacy of incentive design, or the actual implementations. While

economic models are useful to understand the macroscopic financial aspects of AMMs, they do not precisely describe the interactions between AMMs and their users. Still, understanding these interactions is crucial to determine possible deviations from the expected behaviour. Implementations, instead, reflect the exact behaviour of AMMs, but at a level of detail that hampers high-level understanding and reasoning. Moreover, the rich variety of implementations, proposals and models for AMMs, each featuring different sophisticated economic mechanisms, makes it difficult to establish comparisons between AMM designs or to provide a clear contour for the space of possible “well behaving” designs.

**Contributions** In this paper we address these challenges by developing a theory of AMMs. The core of our theory is a formal model of AMMs (§2), based on a thorough inspection of leading AMM implementations like Uniswap [13], Curve [16], and Balancer [3], as well as existing models from the literature [1, 30]. Our model precisely describes the interactions between users and AMMs, and their main economic features. An original aspect of our model is that it is parametric with respect to the key economic mechanism — the *swap invariant* — that algorithmically determines exchange rates between tokens. This makes our model general enough to encompass the mainstream implementations and models of AMMs. With respect to economic models, our theory considers implementation details that are crucial to guarantee (efficient) computability in practice. Our model features an *executable semantics*, which can support implementations and analysis tools. As a matter of fact, an open-source Ocaml implementation of our executable semantics is provided as a companion of this paper.<sup>3</sup>

Building upon our model, we prove a set of properties characterizing both structural (§3) and economic (§4) aspects of AMMs. With respect to previous works, which focus on specific economic mechanisms, all our results are parametric with respect to swap invariants. We identify indeed, for each property, a set of conditions on swap invariants that are sufficient for the property to hold. Our results include fundamental structural properties such as *net worth preservation* (“value cannot be created/destroyed”), *liquidity* (“assets cannot be frozen within an AMM”), and *transaction concurrency* (“two transactions can be executed in any order”), as well as fundamental economic properties such as *incentive-consistency*, which ensures an incentive feedback loop between deposits and swaps of tokens. Most notably, we generalize the formulation and the solution to the so-called *arbitrage problem*, the main game-theoretic foundation behind the economic aspects of AMMs. We show that users are incentivized to perform actions that keep the swap rates aligned with the exchange rates given by price oracles. Namely, if an AMM offers a better swap rate than the oracles’ exchange rate, rational users will perform swaps to narrow the gap. Further, we show that, under certain conditions, deposits and swaps incentivize each other.

Overall, our theory encompasses and generalizes the main functional and economic aspects of the mainstream AMM implementations, providing solid grounds for the design of future AMMs.

---

<sup>3</sup> <https://github.com/blockchain-unica/defi-workbench>

## 2 A formal model of Automated Market Makers

We introduce a formal, operational model of AMMs, focussing on the common features implemented by the main AMM platforms. We discuss in §6 the differences between these platforms and our model.

### 2.1 AMM states

**Basics** We assume a set of *users*  $\mathbb{A}$ , ranged over by  $A, A', \dots$ , and a set of *token types*  $\mathbb{T}$ , ranged over by  $\tau, \tau', \dots$ . We denote with  $\mathbb{T}_0 \subseteq \mathbb{T}$  a specific subset of token types that we call *initial* (they include, e.g., native blockchain tokens). The rest of the token types in  $\mathbb{T}$  represent *minted* tokens, denoted as pairs  $(\tau, \tau')$  of distinct token types, and which represent shares in an AMM. We use  $v, v', r, r'$  to range over nonnegative real numbers ( $\mathbb{R}_0^+$ ), and we write  $r : \tau$  to denote  $r$  units of token type  $\tau$ . We denote with  $\text{dom } f$  the domain of a partial map  $f$ . We model the *wallet* of a user  $A$  as a term  $A[\sigma]$ , where the partial map  $\sigma \in \mathbb{T} \rightarrow \mathbb{R}_0^+$  represents  $A$ 's token holdings. We model an *AMM* as a pair of the form  $(r_0 : \tau_0, r_1 : \tau_1)$ , representing the fact that the AMM is holding, respectively,  $r_0$  and  $r_1$  units of token types  $\tau_0$  and  $\tau_1$ .

**States** We formalise the interaction between users and AMMs as a labelled transition system (LTS). Its labels  $\mathbb{T}, \mathbb{T}', \dots$  represent blockchain *transactions*, while the *states*  $\Gamma, \Gamma', \dots$  are compositions of wallets and AMMs:

$$A_1[\sigma_1] \mid \dots \mid A_n[\sigma_n] \mid (r_1 : \tau_1, r'_1 : \tau'_1) \mid \dots \mid (r_k : \tau_k, r'_k : \tau'_k)$$

where all  $A_i$  are distinct, and for all  $i \neq j$ :  $\tau_i \neq \tau'_i$  (i.e., the token types in an AMM are *distinct*), and  $(\tau_i = \tau_j \Rightarrow \tau'_i \neq \tau'_j) \wedge (\tau_i = \tau'_j \Rightarrow \tau'_i \neq \tau_j)$  (i.e., distinct AMMs cannot hold exactly the same token types). Two AMMs can indeed have a common token type  $\tau$ , as in  $(r_1 : \tau_1, r : \tau)$ ,  $(r' : \tau, r'_2 : \tau'_2)$ , thus enabling indirect trades between token pairs not directly provided by any AMM. A state  $\Gamma$  is *initial* when it only contains wallets with initial tokens. We treat states as sets of terms (wallets/AMMs): hence,  $\Gamma$  and  $\Gamma'$  are equivalent when they contain the same terms; for a term  $Q$ , we write  $Q \in \Gamma$  when  $\Gamma = Q \mid \Gamma'$ , for some  $\Gamma'$ .

*Example 1.* Figure 1 shows an execution trace in our model, that we will explain in detail later in Example 5. We write  $\Gamma \xrightarrow{\mathbb{T}} \Gamma'$  for a state transition from  $\Gamma$  to  $\Gamma'$ , triggered by a transaction  $\mathbb{T}$ . The first two states are initial, while the others contain an AMM for a token pair  $(\tau_0, \tau_1)$ .  $\square$

**Token supply** We define the *supply* of a token type  $\tau$  in a state  $\Gamma$  as the sum of the balances of  $\tau$  in all the wallets and the AMMs occurring in  $\Gamma$ . Formally:

$$\text{sply}_\tau(A[\sigma]) = \begin{cases} \sigma(\tau) & \text{if } \tau \in \text{dom } \sigma \\ 0 & \text{otherwise} \end{cases} \quad \text{sply}_\tau(r_0 : \tau_0, r_1 : \tau_1) = \begin{cases} r_i & \text{if } \tau = \tau_i \\ 0 & \text{otherwise} \end{cases}$$

$$\text{sply}_\tau(\Gamma \mid \Gamma') = \text{sply}_\tau(\Gamma) + \text{sply}_\tau(\Gamma')$$

$$\begin{aligned}
& \mathbf{A}[70 : \tau_0, 80 : \tau_1] \mid \mathbf{B}[30 : \tau_0] \\
& \xrightarrow{\mathbf{A}:\text{xfer}(\mathbf{B}, 10 : \tau_1)} \mathbf{A}[70 : \tau_0, 70 : \tau_1] \mid \mathbf{B}[30 : \tau_0, 10 : \tau_1] \quad (1) \\
& \xrightarrow{\mathbf{A}:\text{dep}(70 : \tau_0, 70 : \tau_1)} \mathbf{A}[70 : (\tau_0, \tau_1)] \mid \mathbf{B}[\dots] \mid (70 : \tau_0, 70 : \tau_1) \quad (2) \\
& \xrightarrow{\mathbf{B}:\text{swapL}(30 : \tau_0, 20 : \tau_1)} \mathbf{A}[\dots] \mid \mathbf{B}[0 : \tau_0, 31 : \tau_1] \mid (100 : \tau_0, 49 : \tau_1) \quad (3) \\
& \xrightarrow{\mathbf{B}:\text{swapR}(29 : \tau_0, 21 : \tau_1)} \mathbf{A}[\dots] \mid \mathbf{B}[30 : \tau_0, 10 : \tau_1] \mid (70 : \tau_0, 70 : \tau_1) \quad (4) \\
& \xrightarrow{\mathbf{B}:\text{rdm}(30 : (\tau_0, \tau_1))} \mathbf{A}[30 : \tau_0, 30 : \tau_1, 40 : (\tau_0, \tau_1)] \mid \mathbf{B}[\dots] \mid (40 : \tau_0, 40 : \tau_1) \quad (5) \\
& \xrightarrow{\mathbf{B}:\text{swapL}(30 : \tau_0, 16 : \tau_1)} \mathbf{A}[\dots] \mid \mathbf{B}[0 : \tau_0, 27 : \tau_1] \mid (70 : \tau_0, 23 : \tau_1) \quad (6) \\
& \xrightarrow{\mathbf{A}:\text{rdm}(30 : (\tau_0, \tau_1))} \mathbf{A}[82 : \tau_0, 47 : \tau_1, 10 : (\tau_0, \tau_1)] \mid \mathbf{B}[\dots] \mid (18 : \tau_0, 6 : \tau_1) \quad (7)
\end{aligned}$$

Fig. 1: Interactions between two users and an AMM.

*Example 2.* Consider the first state in Figure 1,  $\Gamma_1 = \mathbf{A}[70 : \tau_0, 80 : \tau_1] \mid \mathbf{B}[30 : \tau_0]$ . We have that  $\text{sply}_{\tau_0}(\Gamma_1) = 70 + 30 = 100$ , while  $\text{sply}_{\tau_1}(\Gamma_1) = 80$ . Observe that the supply of both token types remains constant in Figure 1; we will show in Lemma 2 that the supply of initial token types is always preserved.  $\square$

**Token prices and net worth** Assume that initial tokens are priced by a global oracle  $P^0 \in \mathbb{T}_0 \rightarrow \mathbb{R}_0^+$ . We then define the **price**  $P_\tau(\Gamma)$  of a token  $\tau \in \mathbb{T}$  (either initial or minted) in a state  $\Gamma$  inductively as follows:

$$\begin{aligned}
P_\tau(\Gamma) &= P^0(\tau) && \text{if } \tau \in \mathbb{T}_0 \\
P_{(\tau_0, \tau_1)}(\Gamma) &= \frac{r_0 \cdot P_{\tau_0}(\Gamma) + r_1 \cdot P_{\tau_1}(\Gamma)}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)} && \text{if } (r_0 : \tau_0, r_1 : \tau_1) \in \Gamma
\end{aligned} \quad (8)$$

The main idea is that initial tokens are priced using directly the global oracle while minted tokens are priced under the assumption that they can be redeemed. Their price, hence, is obtained by (recursively) calculating the price of the tokens that can be obtained by redeeming them (i.e. the proportions of the reserves  $r_0$  and  $r_1$  given the current supply). This intuition will be further formalized later in Lemma 6.

*Example 3.* Let  $\Gamma_7 = \mathbf{A}[82 : \tau_0, 47 : \tau_1, 10 : (\tau_0, \tau_1)] \mid \dots$  be the final state in Figure 1. We have that  $\text{sply}_{(\tau_0, \tau_1)}(\Gamma_7) = 10$ . Assume that the prices of initial tokens are  $P^0(\tau_0) = 5$  and  $P^0(\tau_1) = 9$ . The price of the minted token  $(\tau_0, \tau_1)$  is hence:

$$P_{(\tau_0, \tau_1)}(\Gamma_7) = \frac{1}{10} \left( 18 \cdot P_{\tau_0}(\Gamma_7) + 6 \cdot P_{\tau_1}(\Gamma_7) \right) = \frac{18}{10} \cdot 5 + \frac{6}{10} \cdot 9 = 14.4 \quad \square$$

We now define a key concept to understand the incentives for users to participate in AMMs, namely the **net worth** of a user  $\mathbf{A}$  in a state  $\Gamma$ :

$$W_{\mathbf{A}}(\Gamma) = \begin{cases} \sum_{\tau \in \text{dom } \sigma} \sigma(\tau) \cdot P_\tau(\Gamma) & \text{if } \mathbf{A}[\sigma] \in \Gamma \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

The *global net worth*  $W(I)$  of a state  $I$  is the sum of the net worth in users' wallets. The token units held in AMMs are not accounted for by  $W(I)$ , because their value is already recorded by minted tokens held in users' wallets. Indeed, the equality  $sply_{(\tau_0, \tau_1)}(I) \cdot P_{(\tau_0, \tau_1)}(I) = r_0 \cdot P_{\tau_0}(I) + r_1 \cdot P_{\tau_1}(I)$  between the net value of a minted token and the value of the AMM is a direct consequence of the definition of price in (8).

As we shall see later, one of the main goals of users is to maximize their net worth. This can be achieved through different interactions with the AMM (e.g., by investing tokens or trading units of differently priced token types).

*Example 4.* Recall from Figure 1 the state  $I_1 = \mathbf{A}[70 : \tau_0, 80 : \tau_1] \mid \mathbf{B}[30 : \tau_0]$ , where  $\tau_0$  and  $\tau_1$  are initial tokens. Assume again that the prices are  $P^0(\tau_0) = 5$  and  $P^0(\tau_1) = 9$ . The users' net worth in  $I_1$  are then:

$$W_{\mathbf{A}}(I_1) = 70 \cdot P^0(\tau_0) + 80 \cdot P^0(\tau_1) = 1070 \quad W_{\mathbf{B}}(I_1) = 30 \cdot P^0(\tau_0) = 150$$

In  $I_7 = \mathbf{A}[82 : \tau_0, 47 : \tau_1, 10 : (\tau_0, \tau_1)] \mid \mathbf{B}[0 : \tau_0, 27 : \tau_1] \mid (18 : \tau_0, 6 : \tau_1)$  we have:

$$\begin{aligned} W_{\mathbf{A}}(I_7) &= 82 \cdot P_{\tau_0}(I_7) + 47 \cdot P_{\tau_1}(I_7) + 10 \cdot P_{(\tau_0, \tau_1)}(I_7) = 977 \\ W_{\mathbf{B}}(I_7) &= 27 \cdot P_{\tau_1}(I_7) = 243 \end{aligned}$$

Note that the net worth of  $\mathbf{A}$  has decreased w.r.t. the initial state, while the net worth of  $\mathbf{B}$  has increased. One may think that  $\mathbf{B}$  has been more successful than  $\mathbf{A}$ , but this depends on the users' goals. Note, e.g., that  $\mathbf{A}$  holds 10 units of the minted token  $(\tau_0, \tau_1)$ , whose price may increase in the future.  $\square$

## 2.2 AMM semantics

We now formally describe the interactions of the AMM that give rise to state transitions. State transitions are triggered by the *transactions* in Table 1. We formalise below their behaviour, but we give before an overview of our running example from Figure 1.

*Example 5.* Figure 1 actually displays a sequence of transitions in the LTS of our model. To keep the example simple, we have used there the *constant product* swap invariant, which requires swap transactions to preserve the product between the amounts of the two tokens in the AMM; further, we have assumed no fees. In step (1),  $\mathbf{A}$  transfers 10  $\tau_1$  from her wallet to  $\mathbf{B}$ 's. In step (2),  $\mathbf{A}$  creates a new AMM, depositing 70  $\tau_0$  and 70  $\tau_1$ ; in return, she receives 70 units of the minted token  $(\tau_0, \tau_1)$ . In step (3),  $\mathbf{B}$  swaps 30 of his units of  $\tau_0$  for *at least* 20 units of  $\tau_1$ . The actual amount of units of  $\tau_1$  received by  $\mathbf{B}$  is 21: indeed,  $(70 + 30) \cdot (70 - 21) = 70 \cdot 70$ , hence 21 satisfies the constant product swap invariant. In step (4),  $\mathbf{B}$  reverses his prior action by swapping 21 of his units of  $\tau_1$  for *at least* 29 units of  $\tau_0$ . Here, the actual amount of units of  $\tau_1$  received by  $\mathbf{B}$  is 30, which also satisfies the constant product swap invariant. In step (5),  $\mathbf{B}$  redeems 30 units of the minted token  $(\tau_0, \tau_1)$ , accordingly reducing the funds in the AMM. Note that the received tokens exhibit the same 1-to-1 ratio as in the

$A : \text{xfer}(B, v : \tau)$	$A$ transfers $v : \tau$ to $B$
$A : \text{dep}(v_0 : \tau_0, v_1 : \tau_1)$	$A$ deposits $v_0 : \tau_0$ and $v_1 : \tau_1$ to an AMM $(r_0 : \tau_0, r_1 : \tau_1)$ , receiving in return some units of the minted token $(\tau_0, \tau_1)$
$A : \text{swapL}(v_0 : \tau_0, v_1 : \tau_1)$	$A$ transfers $v_0 : \tau_0$ to an AMM $(r_0 : \tau_0, r_1 : \tau_1)$ , receiving in return <i>at least</i> $v_1$ units of $\tau_1$
$A : \text{swapR}(v_0 : \tau_0, v_1 : \tau_1)$	$A$ transfers $v_1 : \tau_1$ to an AMM $(r_0 : \tau_0, r_1 : \tau_1)$ , receiving in return <i>at least</i> $v_0$ units of $\tau_0$
$A : \text{rdm}(v : \tau)$	$A$ redeems $v$ units of minted token $\tau = (\tau_0, \tau_1)$ from an AMM $(r_0 : \tau_0, r_1 : \tau_1)$ , receiving in return some units of $\tau_0$ and $\tau_1$

Table 1: AMM transactions.

initial deposit at step (2). In step (6),  $B$  swaps 30 of his units of  $\tau_0$  for *at least* 16 units of  $\tau_1$ . Unlike in the previous swap at step (3), now the actual amount of  $\tau_1$  received by  $B$  is 17. Note that the implied *swap rate* between received  $\tau_1$  units and sent  $\tau_0$  units has deteriorated w.r.t. step (3), even if the pair  $(\tau_0, \tau_1)$  had the same 1-to-1 ratio of funds. This is caused by the reduction in funds resulting from  $A$ 's redeem action: thus, the swap rate is sensitive to both the ratio of funds in the pair as well as their absolute balances, a key property of the incentive mechanisms, as we shall see later in §4. Finally, in step (7)  $A$  performs another redeem of 30 units of the minted token  $(\tau_0, \tau_1)$ , thereby extracting 52 units of  $\tau_0$  and 17 units of  $\tau_1$  from the AMM. Note that the ratio of redeemed tokens is no longer 1-to-1 as in the previous redeem action (5), as the prior left swap has changed the ratio between the funds of  $\tau_0$  and  $\tau_1$  in the AMM.  $\square$

We now formalise the transition rules. We use the standard notation  $\sigma\{v/x\}$  to update a partial map  $\sigma$  at point  $x$ : namely,  $\sigma\{v/x\}(x) = v$ , while  $\sigma\{v/x\}(y) = \sigma(y)$  for  $y \neq x$ . Given a partial map  $\sigma \in \mathbb{T} \rightarrow \mathbb{R}_0^+$ , a token type  $\tau \in \mathbb{T}$  and a partial operation  $\circ \in \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ , we define the partial map  $\sigma \circ v : \tau$  as follows:

$$\sigma \circ v : \tau = \begin{cases} \sigma\{\sigma(\tau) \circ v/\tau\} & \text{if } \tau \in \text{dom } \sigma \text{ and } \sigma(\tau) \circ v \in \mathbb{R}_0^+ \\ \sigma\{v/\tau\} & \text{if } \tau \notin \text{dom } \sigma \end{cases}$$

**Token transfer** A user  $A$  can transfer some of her tokens to another user  $B$ , provided that there are enough units of the token in  $A$ 's wallet. Formally:

$$\frac{\sigma_A(\tau) \geq v}{A[\sigma_A] \mid B[\sigma_B] \mid \Gamma \xrightarrow{A:\text{xfer}(B,v:\tau)} A[\sigma_A - v : \tau] \mid B[\sigma_B + v : \tau] \mid \Gamma} \text{[XFER]}$$

A consequence of this rule is that tokens (both initial and minted) are *fungible*, i.e. individual units of the same token type are interchangeable. In particular, amounts of tokens of the same type can be split into smaller parts, and two amounts of tokens of the same type can be joined.

**Deposit** Any user can create an AMM for a token pair  $(\tau_0, \tau_1)$  provided that such an AMM is not already present in the state. This is achieved by the transaction  $A : \text{dep}(v_0 : \tau_0, v_1 : \tau_1)$ , through which  $A$  transfers  $v_0 : \tau_0$  and  $v_1 : \tau_1$  to

the new AMM. In return for the deposit, **A** receives a certain positive amount of units of a new token type  $(\tau_0, \tau_1)$ , which is minted by the AMM. The exact amount of units received is irrelevant. In our model we choose  $v_0$  but any other choice would be valid. We formalise this behaviour by the rule:

$$\frac{\sigma(\tau_i) \geq v_i > 0 \ (i \in \{0, 1\}) \quad \tau_0 \neq \tau_1 \quad (- : \tau_0, - : \tau_1), (- : \tau_1, - : \tau_0) \notin \Gamma}{\text{A}[\sigma] \mid \Gamma \xrightarrow{\text{A:dep}(v_0 : \tau_0, v_1 : \tau_1)} \text{A}[\sigma - v_0 : \tau_0 - v_1 : \tau_1 + v_0 : (\tau_0, \tau_1)] \mid (v_0 : \tau_0, v_1 : \tau_1) \mid \Gamma} \text{[DEP0]}$$

Once an AMM is created, any user can deposit tokens into it, as long as doing so preserves the ratio of the token holdings in the AMM. When a user deposits  $v_0 : \tau_0$  and  $v_1 : \tau_1$  to an existing AMM, it receives in return an amount of minted tokens of type  $(\tau_0, \tau_1)$ . This amount is the ratio between the deposited amount  $v_0$  and the *redeem rate* of  $(\tau_0, \tau_1)$  in the current state  $\Gamma$ , i.e. the ratio between the amount  $r_0$  of  $\tau_0$  stored in the AMM, and the total supply  $\text{sply}_{(\tau_0, \tau_1)}(\Gamma)$  of the minted token in the state.

$$\frac{\sigma(\tau_i) \geq v_i > 0 \ (i \in \{0, 1\}) \quad r_1 v_0 = r_0 v_1 \quad v = \frac{v_0}{r_0} \cdot \text{sply}_{(\tau_0, \tau_1)}(\Gamma)}{\Gamma = \text{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma' \xrightarrow{\text{A:dep}(v_0 : \tau_0, v_1 : \tau_1)} \text{A}[\sigma - v_0 : \tau_0 - v_1 : \tau_1 + v : (\tau_0, \tau_1)] \mid (r_0 + v_0 : \tau_0, r_1 + v_1 : \tau_1) \mid \Gamma'} \text{[DEP]}$$

Note that the premise  $r_1 v_0 = r_0 v_1$  ensures that the ratio between the holdings of  $\tau_0$  and  $\tau_1$  in the AMM is preserved by the **dep** transaction, i.e.:

$$\frac{r_1 + v_1}{r_0 + v_0} = \frac{r_1}{r_0}$$

As we shall see in §4, users are incentivized to invest tokens into AMMs by the fact that trading operations (i.e., swaps) are subject to a fee mechanism that makes the redeem rate increase over time.

**Swap** As shown in step (3) and on of Example 5, users can increase their net worth by swapping tokens. Any user **A** can swap units of  $\tau_0$  in her wallet for units of  $\tau_1$  in an AMM  $(r_0 : \tau_0, r_1 : \tau_1)$  by firing a transaction **A** : **swapL** $(v_0 : \tau_0, v_1 : \tau_1)$ . Here,  $v_0$  is the amount of  $\tau_0$  transferred from **A**'s wallet to the AMM, while  $v_1$  is a *lower bound* on the amount of  $\tau_1$  that **A** will receive in return. The actual amount  $v$  is determined by a *swap invariant*  $I \in \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ , that must hold between the amounts of  $\tau_0$  and  $\tau_1$  held in the AMM before and after the swap. To determine  $v$ , the AMM requires a fraction  $0 < \phi \leq 1$  of  $v_0$ ; the rest is considered as a fee (the parameter  $\phi$  is the *fee rate*). Formally:

$$\frac{\sigma(\tau_0) \geq v_0 > 0 \quad I(r_0 + \phi v_0, r_1 - v) = I(r_0, r_1) \quad 0 < v_1 \leq v \leq r_1}{\text{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma \xrightarrow{\text{A:swapL}(v_0 : \tau_0, v_1 : \tau_1)} \text{A}[\sigma - v_0 : \tau_0 + v : \tau_1] \mid (r_0 + v_0 : \tau_0, r_1 - v : \tau_1) \mid \Gamma} \text{[SWAPL]}$$

The effect of the fee is that the redeem rate of minted tokens increases; intuitively, the AMM retains a portion of the swapped amounts, but the overall

reserve is still distributed among all minted tokens, thereby ensuring liquidity (as we shall formally establish liquidity later on in Lemma 4).

Although actual AMM implementations use a variety of different swap invariants, with the common aim to incentivize users to perform swaps, all these invariants share a few common design choices. A crucial one is that there exists *exactly* one  $v$  which satisfies the equation in the premise of [SWAPL]; further, swapping 0 units of  $\tau_0$  results in 0 units of  $\tau_1$ . Formally, for all  $r_0, r_1 > 0$ :

$$\forall v \in \mathbb{R}_0^+ : \exists! v' \in \mathbb{R}_0^+ : I(r_0 + v, r_1 - v') = I(r_0, r_1) \quad (10)$$

Hereafter, we assume that  $I$  always respects this condition. A common swap invariant, implemented e.g. by Uniswap [13] and Mooniswap [7] (and also used in Example 5), is the *constant product invariant*, which requires that the product of the amounts of  $\tau_0$  and  $\tau_1$  in the AMM remains constant, i.e.  $I(r_0, r_1) = r_0 \cdot r_1$ .

The rule [SWAPR] allows for swaps in the other direction:

$$\frac{\sigma(\tau_1) \geq v_1 > 0 \quad I(r_0 - v, r_1 + \phi v_1) = I(r_0, r_1) \quad 0 < v_0 \leq v \leq r_0}{\text{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma \xrightarrow{\text{A:swapR}(v_0 : \tau_0, v_1 : \tau_1)} \text{A}[\sigma + v : \tau_0 - v_1 : \tau_1] \mid (r_0 - v : \tau_0, r_1 + v_1 : \tau_1) \mid \Gamma} \quad \text{[SWAPR]}$$

where we assume that  $I$  enjoys the “right” version of the condition (10).

It is worth explaining why the swap transactions specify lower bounds for the amount of return tokens, instead of an exact amount. In practice, when a user emits a transaction, she cannot predict the exact state in which the transaction will be actually committed. This makes it unfeasible to guess the exact amount that will preserve the swap invariant: hence, users can only specify a lower bound that they are willing to accept.

**Redeem** Any user can redeem units of a minted token  $(\tau_0, \tau_1)$ , obtaining in return units of the underlying tokens  $\tau_0$  and  $\tau_1$ . The redeemable amounts are determined by the redeem rate: each unit of  $(\tau_0, \tau_1)$  can be redeemed for equal fractions of  $\tau_0$  and  $\tau_1$  remaining in the AMM:

$$\frac{\sigma(\tau_0, \tau_1) \geq v > 0 \quad v_0 = v \frac{r_0}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)} \quad v_1 = v \frac{r_1}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)}}{\Gamma = \text{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma' \xrightarrow{\text{A:redm}(v : (\tau_0, \tau_1))} \text{A}[\sigma + v_0 : \tau_0 + v_1 : \tau_1 - v : (\tau_0, \tau_1)] \mid (r_0 - v_0 : \tau_0, r_1 - v_1 : \tau_1) \mid \Gamma'} \quad \text{[RDM]}$$

*Example 6.* Figure 2 shows the evolution of the AMM token holdings resulting from the trace in Figure 1, presented with Example 5. Recall that we have assumed a constant product swap invariant  $x \cdot y = k$ , and no swap fees ( $\phi = 1$ ). We refer to a state in Figure 1 by the action number preceding it: the AMM  $(70 : \tau_0, 70 : \tau_1)$  in state (2) is shown in Figure 2. Subsequent left (3) and right (4) swaps result in a traversal along  $k = 70 \cdot 70$  from  $(70 : \tau_0, 70 : \tau_1)$  to  $(100 : \tau_0, 49 : \tau_1)$ , and back as the swap invariant must hold for swap actions. The redeem action (5) reduces the holdings of both tokens by the same factor to reach  $(40 : \tau_0, 40 : \tau_1)$ . A left swap (6) traverses  $k' = 40 \cdot 40$  to reach  $(70 : \tau_0, 23 : \tau_1)$  in state (6), which is then followed by another redeem (7) action, reducing both token holdings proportionally to  $(18 : \tau_0, 6 : \tau_1)$ .



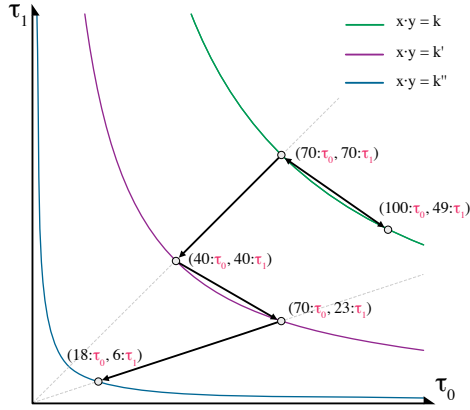


Fig. 2: Evolution of balances of AMM  $(\tau_0, \tau_1)$  along the trace in Figure 1.

### 3 Structural properties of AMMs

We now establish some structural properties of AMMs, which do not depend on the design of the economic mechanisms, i.e. on the choice of the swap invariant. We denote with  $\rightarrow^*$  the reflexive and transitive closure of  $\rightarrow$ . Given a finite sequence of transactions  $\lambda = \mathbb{T}_1 \cdots \mathbb{T}_k$ , we write  $\Gamma \xrightarrow{\lambda} \Gamma'$  when  $\Gamma \xrightarrow{\mathbb{T}_1} \cdots \xrightarrow{\mathbb{T}_k} \Gamma'$ . We say that a state  $\Gamma$  is *reachable* if  $\Gamma_0 \rightarrow^* \Gamma$  for some initial  $\Gamma_0$ . We denote with  $type(\mathbb{T})$  the type of  $\mathbb{T}$  (i.e., `xfer`, `dep`, ...), with  $wal(\mathbb{T})$  the set of wallets affected by  $\mathbb{T}$  (e.g.,  $wal(\mathbb{A} : \text{xfer}(\mathbb{B}, v : \tau)) = \{\mathbb{A}, \mathbb{B}\}$ ), and with  $tok(\mathbb{T})$  the set of token types affected by  $\mathbb{T}$  (e.g.,  $tok(\mathbb{A} : \text{swapl}(v_0 : \tau_0, v_1 : \tau_1)) = \{\tau_0, \tau_1\}$ ).

First, we establish that the AMMs' LTS is deterministic. Note that, in `swap` rules, an unconstrained swap invariant  $I$  could admit different solutions to the equation in the premise: determinism is ensured by condition (10), which we assume to be true for all swap invariants.

**Lemma 1 (Determinism).** *If  $\Gamma \xrightarrow{\mathbb{T}} \Gamma'$  and  $\Gamma \xrightarrow{\mathbb{T}} \Gamma''$ , then  $\Gamma' = \Gamma''$ .*

We can lift the statement to sequences of transactions by using a simple inductive argument. The same applies to other single-step results in this section.

Lemma 2 ensures that the supply of each *initial* token type  $\tau$  is preserved by transitions (of any type). Note that preservation does not hold for *minted* tokens, as they can be created (by rule [DEP]) and destroyed (by rule [RDM]).

**Lemma 2.** *For all  $\tau \in \mathbb{T}_0$ , if  $\Gamma \rightarrow \Gamma'$  then  $sply_\tau(\Gamma) = sply_\tau(\Gamma')$ .*

Lemma 3 ensures that the *global* net worth is preserved by transactions, whereas the user's net worth is preserved only by redeems/deposits.

**Lemma 3 (Preservation of net worth).** *Let  $\Gamma \xrightarrow{\mathbb{T}} \Gamma'$ . Then,  $W(\Gamma) = W(\Gamma')$ . Further, if  $type(\mathbb{T}) \in \{\text{dep}, \text{rdm}\}$  or  $\mathbb{A} \notin wal(\mathbb{T})$ , then  $W_{\mathbb{A}}(\Gamma) = W_{\mathbb{A}}(\Gamma')$ .*

Lemma 4 ensures that funds cannot be *frozen* in an AMM, i.e. that users can always redeem arbitrary amounts of the tokens deposited in an AMM.

**Lemma 4 (Liquidity).** *Let  $\Gamma$  be a reachable state such that  $(r_0 : \tau_0, r_1 : \tau_1) \in \Gamma$  with  $r_0 + r_1 > 0$ . Then: (a)  $\text{sply}_{(\tau_0, \tau_1)}(\Gamma) > 0$ ; (b) for all  $r'_0 \leq r_0$ , there exists  $r'_1 \leq r_1$  such that  $\Gamma \rightarrow^* (r'_0 : \tau_0, r'_1 : \tau_1) \mid \dots$ ; (c) for all  $r'_1 \leq r_1$ , there exists  $r'_0 \leq r_0$  such that  $\Gamma \rightarrow^* (r'_0 : \tau_0, r'_1 : \tau_1) \mid \dots$ .*

We now study the concurrency of transactions. Two finite sequences of transactions  $\lambda_0$  and  $\lambda_1$  are *observationally equivalent*, in denoted  $\lambda_0 \sim \lambda_1$ , when, for all states  $\Gamma$ , if  $\Gamma \xrightarrow{\lambda_0} \Gamma_0$  and  $\Gamma \xrightarrow{\lambda_1} \Gamma_1$  then  $\Gamma_0 = \Gamma_1$ . We say that two distinct transactions  $\mathbb{T}, \mathbb{T}'$  are *concurrent* (denoted,  $\mathbb{T} \# \mathbb{T}'$ ) if  $\mathbb{T}\mathbb{T}' \sim \mathbb{T}'\mathbb{T}$ . Note that this does not mean that  $\mathbb{T}$  and  $\mathbb{T}'$  cannot disable each other as demanded by stricter notions of concurrency. Lemma 5 provides sufficient conditions for two transactions to be concurrent: intuitively, two non-**swap** transactions are always concurrent, while **swap** transactions are concurrent with **xfer** transactions, and with any transactions which do not affect the same token types.

**Lemma 5.** *Two distinct transactions  $\mathbb{T}_0, \mathbb{T}_1$  are concurrent if, for  $i \in \{0, 1\}$ ,  $\text{type}(\mathbb{T}_i) \in \{\text{swapL}, \text{swapR}\}$  implies  $\text{tok}(\mathbb{T}_i) \cap \text{tok}(\mathbb{T}_{1-i}) = \emptyset$  or  $\text{type}(\mathbb{T}_{1-i}) = \text{xfer}$ .*

As we shall see later in §4, it is actually desirable, and crucial for the economic mechanism of AMMs, that **swap** transactions interfere with other transactions that trade the same token type.

The theory of Mazurkiewicz’s trace languages [26] allows us to lift Lemma 5 to sequences of transactions. Let  $R$  be a symmetric and irreflexive relation on the set  $\mathbb{X}$  of all transactions. The *Mazurkiewicz equivalence*  $\sim_R$  is the least congruence in the free monoid  $\mathbb{X}^*$  such that:  $\forall \mathbb{T}, \mathbb{T}' \in \mathbb{X}: \mathbb{T} R \mathbb{T}' \implies \mathbb{T}\mathbb{T}' \sim_R \mathbb{T}'\mathbb{T}$ . Theorem 1 states that the Mazurkiewicz equivalence constructed on the concurrency relation  $\#$  is an observational equivalence.

**Theorem 1 (Concurrent transactions can be reordered).**  $\sim_{\#} \subseteq \sim$ .

A direct consequence of Theorem 1 is that we can transform a finite sequence of transactions into an observationally equivalent one by repeatedly exchanging adjacent concurrent transactions — provided that both sequences are executable in the LTS. For example, sequences of  $\mathbb{A} : \text{rdm}(\_)$  transactions can be freely reordered, resulting in the same, unique state. This is exploited in the following lemma, which supports the inductive definition of the price of minted tokens in (8): indeed, computing the net worth of a user  $\mathbb{A}$  under that price definition corresponds to making  $\mathbb{A}$  first redeem all her minted tokens, and then summing the price of the resulting initial tokens.

**Lemma 6.** *For all states  $\Gamma$  and users  $\mathbb{A}$ , let  $\text{rdm}_{\mathbb{A}}(\Gamma)$  be the unique state reached from  $\Gamma$  by performing only  $\mathbb{A} : \text{rdm}(\_)$  actions, such that  $\mathbb{A}$ ’s wallet in  $\text{rdm}_{\mathbb{A}}(\Gamma)$ , only contains initial tokens. Then:*

$$W_{\mathbb{A}}(\Gamma) = \sum_{\tau \in \text{dom } \sigma} \sigma(\tau) \cdot P^0(\tau) \quad \text{if } \mathbb{A}[\sigma] \in \text{rdm}_{\mathbb{A}}(\Gamma)$$

*Example 7.* Recall from Example 4 that  $W_A(I_7) = 977$ . Assume that **A** performs a further transaction to redeem all 10 units of  $(\tau_0, \tau_1)$  from her wallet. The resulting state is  $I_8 = \mathbf{A}[100 : \tau_0, 53 : \tau_1] \mid \dots$ . We compute **A**'s net worth in that state, using the oracle token prices:  $W_A(I_8) = 100 \cdot P_{(\tau_0)}(I_7) + 53 \cdot P_{(\tau_1)}(I_7) = 100 \cdot 5 + 53 \cdot 9 = 977$ , as correctly predicted by Lemma 6.  $\square$

## 4 Properties of AMM incentives

We now study the incentive mechanisms of AMMs. We start in §4.1 by introducing a few notions of *exchange rate*, which are pivotal to understanding these mechanisms. In §4.2 we devise general conditions on swap invariants, overall named *incentive-consistency*, which guarantee that AMMs enjoy relevant economic properties. In §4.3 we study solutions to the *arbitrage problem*, which is the key to incentivize users to perform *swap* operations towards an ideal state where the AMM's exchange rates align with the exchange rates set by price oracles. Finally, in §4.4 we study the incentives to swap and deposit larger amounts.

### 4.1 Exchange rates

The *exchange rate* between two token types is the number of units of one token needed to buy one unit of the other token at the current price. We define *Left* and *Right* versions of this notion, that reflect the direction of the exchange:

$$XL_\Gamma(\tau_0, \tau_1) = P_{\tau_0}(\Gamma)/P_{\tau_1}(\Gamma) \quad XR_\Gamma(\tau_0, \tau_1) = P_{\tau_1}(\Gamma)/P_{\tau_0}(\Gamma) \quad (11)$$

The *swap rate* between  $\tau_0$  and  $\tau_1$  upon a payment of  $v_i : \tau_i$  (for  $i \in \{0, 1\}$ ) is the ratio between  $v$  and  $v_i$ , where  $v$  is the received amount of  $\tau_{1-i}$  resulting from a swap action on an AMM  $(r_0 : \tau_0, r_1 : \tau_1)$ . We first introduce an auxiliary notion, parameterized over the balances  $r_0$  and  $r_1$ , instead of the token types:

$$\begin{aligned} XL_\phi^{\text{swap}}(v_0, r_0, r_1) &= v/v_0 & \text{if } I(r_0, r_1) &= I(r_0 + \phi v_0, r_1 - v) \\ XR_\phi^{\text{swap}}(v_1, r_0, r_1) &= v/v_1 & \text{if } I(r_0, r_1) &= I(r_0 - v, r_1 + \phi v_1) \end{aligned} \quad (12)$$

The swap rate is parameterized over the fee rate  $\phi$ : the case where  $\phi = 1$  represents an ideal scenario with no fees: in this case, we write just  $XL^{\text{swap}}(v_0, r_0, r_1)$ . We define the swap rate in a state  $\Gamma$  such that  $(r_0 : \tau_0, r_1 : \tau_1) \in \Gamma$  as follows:

$$XL_{\Gamma, \phi}^{\text{swap}}(v_0, \tau_0, \tau_1) = XL_\phi^{\text{swap}}(v_0, r_0, r_1) \quad XR_{\Gamma, \phi}^{\text{swap}}(v_1, \tau_0, \tau_1) = XR_\phi^{\text{swap}}(v_1, r_0, r_1)$$

We also define the *redeem rate*. The left version is:

$$XL_\Gamma^{\text{rdm}}(\tau_0, \tau_1) = r_0 / \text{sply}_{(\tau_0, \tau_1)}(\Gamma) \quad \text{if } (r_0 : \tau_0, r_1 : \tau_1) \in \Gamma \quad (13)$$

### 4.2 General properties of swap invariants

We now introduce a set of properties of swap invariants, called cumulatively *incentive-consistency*, which overall incentivize users to interact with AMMs by performing swap and deposit actions.

**Swap-rate continuity** This property requires that, for all  $r_0, r_1 > 0$ :

$$\lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}(\varepsilon, r_0, r_1) = 1 / \lim_{\varepsilon \rightarrow 0} XR^{\text{swap}}(\varepsilon, r_0, r_1) \in \mathbb{R}^+ \quad (14)$$

Figure 3 (left) illustrates this property, displaying the points  $(x, y)$  which satisfy the constant product invariant  $x \cdot y = k$ . The left swap rate limit for the constant product invariant and  $\phi = 1$  is  $\lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}(\varepsilon, r_0, r_1) = r_1/r_0$ , while for the right swap we have  $\lim_{\varepsilon \rightarrow 0} XR^{\text{swap}}(\varepsilon, r_0, r_1) = r_0/r_1$ . Coinciding left swap limit and right swap limit inverse are illustrated as the slope of the product constant curve at a selected point in Figure 3 (left). The constant product invariant satisfies (14), i.e. it is swap-rate continuous.

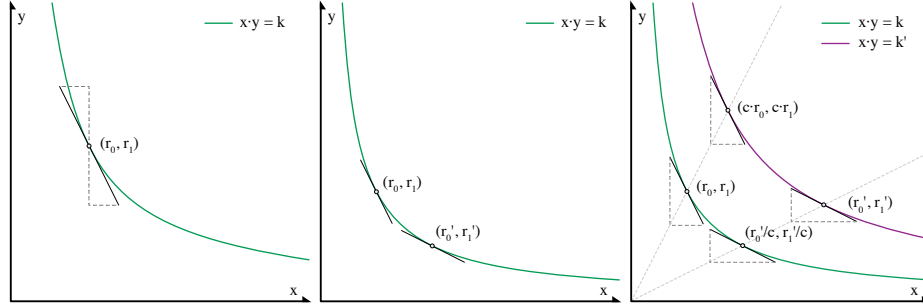


Fig. 3: The constant product invariant  $I(x, y) = x \cdot y$  is swap-rate-consistent (left), demand-sensitive (center), non-depletable, funds-consistent (right) and swap-rate consistent (right).

**Demand-sensitivity** A swap invariant is demand-sensitive if the swap rate strictly decreases with demand. Formally, for all  $r_0, r_1, r'_0, r'_1 > 0$ :

$$I(r_0, r_1) = I(r'_0, r'_1) \wedge r'_0 > r_0 \implies \lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}_{\phi}(\varepsilon, r_0, r_1) > \lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}_{\phi}(\varepsilon, r'_0, r'_1) \quad (15)$$

We implicitly require that (15) and the subsequent properties stated for the left version of an exchange rate also hold for the right version.

Figure 3 (center) depicts two points  $(r_0, r_1), (r'_0, r'_1)$  on the constant product curve, which satisfy  $x \cdot y = k$  for identical  $k$ . For the constant product invariant, the left swap limit can be expressed as  $\lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}(\varepsilon, r_0, r_1) = \phi \cdot r_1/r_0$ . For the given  $k$  and points in Figure 3 (center):

$$\lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}_{\phi}(\varepsilon, r_0, r_1) = \phi \cdot k/r_0^2 \quad \lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}_{\phi}(\varepsilon, r'_0, r'_1) = \phi \cdot k/r_0'^2$$

Thus for  $r'_0 > r_0$ ,  $\lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}_{\phi}(\varepsilon, r_0, r_1) > \lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}_{\phi}(\varepsilon, r'_0, r'_1)$ : the constant product invariant is demand-sensitive.

**Non-depletion** This property ensures that the balance of tokens within an AMM cannot be zeroed via swaps. Formally,  $I$  is non-depletable when, for all  $r_0, r_1 > 0$  and  $r'_0, r'_1 \geq 0$ :

$$I(r_0, r_1) = I(r'_0, r'_1) \implies r'_0, r'_1 \neq 0 \quad (16)$$

Note that the constant product invariant trivially satisfies this property.

**Funds-consistency** Deposits to an AMM ensure higher swap rates for a given input amount  $v$ , whereas redeems will reduce the swap rates for  $v$ . This behaviour is formalized later on in Theorem 4, but is a consequence of the funds-consistency property of the swap invariant. Formally, we require that for all  $r_0, r_1, r'_0, r'_1 > 0$ :

$$\begin{aligned} I(r_0, r_1) \neq I(r'_0, r'_1) &\iff \\ \exists! c \in \mathbb{R}^+ \setminus \{1\} : I(c \cdot r_0, c \cdot r_1) &= I(r_0, r_1) \wedge I\left(\frac{r_0}{c}, \frac{r_1}{c}\right) = I(r'_0, r'_1) \end{aligned} \quad (17)$$

Figure 3 (right) illustrates funds-consistency for the constant product invariant. Here,  $r_0 \cdot r_1 = k \neq k' = r'_0 \cdot r'_1$ . We observe that there exists a unique  $c > 0$  in  $(c \cdot r_0) \cdot (c \cdot r_1) = r'_0 \cdot r'_1 = k'$ : namely,  $c = \sqrt{(r'_0 \cdot r'_1)/(r_0 \cdot r_1)}$ . Conversely,  $(r'_0/c) \cdot (r'_1/c) = r_0 \cdot r_1$ , which holds for the same value of  $c$ .

**Swap-rate-consistency** The design of AMMs aims to ensure that redeems and deposits do not interfere with the alignment of the swap rate towards the exchange rate. Since both deposits and redeems preserve the balance ratio of a token pair, we require swap rate limits for all balances of a given ratio to be constant. For all  $r_0, r_1, c > 0$ :

$$\lim_{\varepsilon \rightarrow 0} XL_{\phi}^{\text{swap}}(\varepsilon, r_0, r_1) = \lim_{\varepsilon \rightarrow 0} XL_{\phi}^{\text{swap}}(\varepsilon, c \cdot r_0, c \cdot r_1) \quad (18)$$

Figure 3 (right) illustrates equal swap rate limits for given  $r_0, r_1$  and  $c \cdot r_0, c \cdot r_1$ . Here,  $\lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}(\varepsilon, r_0, r_1) = \phi \cdot r_1/r_0 = \phi \cdot (c \cdot r_1)/(c \cdot r_0)$  for  $c > 0$ .

Finally, the following lemma establishes that the constant product swap invariant (the one used e.g. by Uniswap and Mooniswap) is indeed incentive-consistent. We conjecture that the same is true for the swap invariants implemented by the other mainstream AMM platforms.

**Lemma 7.** *The constant product swap invariant is incentive-consistent.*

### 4.3 The arbitrage game

We now study the incentive mechanisms of AMMs from a game-theoretic perspective. Indeed, AMMs can be seen as multi-player games where users collaborate or compete to achieve possibly conflicting goals. In such games the allowed moves of users are the interactions with other users and with AMMs, while their goal is typically to increase their net worth.

The **arbitrage problem** is an interesting example of an AMM game since it is directly linked to the incentive of swaps in a way that makes AMMs track exchange rates. The arbitrage problem has been formalized for specific swap

invariants, namely the *weighted* and *constant product* swap invariant [17, 19]. We generalize here the arbitrage problem to *arbitrary* swap invariants. We provide sufficient conditions for the existence of solutions, and we link the solutions to the expected relation between AMMs and exchange rates.

We model the arbitrage problem as a single-player, single-round game. The *initial game state* is  $\Gamma_0 = \mathbf{A}[\sigma] \mid (r_1 : \tau_0, r_1 : \tau_1)$ , where  $\mathbf{A}$  is the only *player*. The *moves* of  $\mathbf{A}$  are all the possible transactions fired by  $\mathbf{A}$ ; we also consider doing nothing as a possible move. The *goal* of  $\mathbf{A}$  is to maximize her net worth, i.e. to maximize  $W_{\mathbf{A}}(\Gamma) - W_{\mathbf{A}}(\Gamma_0)$ , where  $\Gamma$  is the state resulting from executing the selected move. A *solution* to the game is a move that satisfies the goal, i.e. one of the optimal moves. We further assume that  $\mathbf{A}$  holds no minted tokens containing  $(\tau_0, \tau_1)$  as a subterm (i.e.,  $(\tau_0, \tau_1)$  itself,  $((\tau_0, \tau_1), \tau_2)$ , etc.). In this way, any change in  $\mathbf{A}$ 's net worth only depends on the exchange rate between  $\tau_0$  and  $\tau_1$ , and on the transfer of value resulting from  $\mathbf{A}$ 's move.

Before presenting the solution to the game we examine the potential candidates for the solution. First, note that transfers are not valid solutions, as they can only decrease  $\mathbf{A}$ 's net worth. A second observation is that doing nothing, depositing or redeeming do not alter  $\mathbf{A}$ 's net worth (cf. Lemma 3). Hence, if one of such moves is a solution, so are the other two. The only moves that may affect  $\mathbf{A}$ 's net worth are swaps. For a swap to be a solution to the game, it must, first of all, result in a positive change of  $\mathbf{A}$ 's net worth. This happens when the swap rate is greater than the exchange rate. Theorem 2 presents the solution to the game. Note that if  $\text{swapL}(v_0 : \tau_0, v_1 : \tau_1)$  is a solution, then for all  $v'_1 \leq v_1$ , also  $\text{swapL}(v_0 : \tau_0, v'_1 : \tau_1)$  is a solution. Without loss of generality, our statement singles out the solution with the greatest  $v_1$  (similarly for the right swap).

**Theorem 2.** *Let  $I$  be demand-sensitive and non-depletable, and let the initial state of the game be  $\Gamma_0 = \mathbf{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1)$ , with  $r_0, r_1 > 0$ . Let  $\sigma(\tau_0), \sigma(\tau_1)$  be large enough to enable any needed swap. Then, the solution to the game is:*

–  $\mathbf{A} : \text{swapL}(v_0 : \tau_0, v_1 : \tau_1)$  if  $\Gamma_0 \xrightarrow{\mathbf{A}:\text{swapL}(v_0:\tau_0,v_1:\tau_1)} \Gamma$ , and:

$$(1) \lim_{\varepsilon \rightarrow 0} XL_{\Gamma_0, \phi}^{\text{swap}}(\varepsilon, \tau_0, \tau_1) > XL_{\Gamma_0}(\tau_0, \tau_1)$$

$$(2) \lim_{\varepsilon \rightarrow 0} XL_{\phi}^{\text{swap}}(\varepsilon, r_0 + \phi \cdot v_0, r_1 - v_1) = XL_{\Gamma}(\tau_0, \tau_1) \quad \text{where } \exists! \delta :$$

$$I(r_0, r_1) = I(r_0 + \phi \cdot v_0, r_1 - v_1) = I(r_0 + \phi \cdot (v_0 + \varepsilon), r_1 - (v_1 + \delta))$$

–  $\mathbf{A} : \text{swapR}(v_0 : \tau_0, v_1 : \tau_1)$  if  $\Gamma_0 \xrightarrow{\mathbf{A}:\text{swapR}(v_0:\tau_0,v_1:\tau_1)} \Gamma$ , and:

$$(1) \lim_{\varepsilon \rightarrow 0} XR_{\Gamma_0, \phi}^{\text{swap}}(\varepsilon, \tau_0, \tau_1) > XR_{\Gamma_0}(\tau_0, \tau_1)$$

$$(2) \lim_{\varepsilon \rightarrow 0} XR_{\phi}^{\text{swap}}(\varepsilon, r_0 - v_0, r_1 + \phi \cdot v_1) = XR_{\Gamma}(\tau_0, \tau_1) \quad \text{where } \exists! \delta :$$

$$I(r_0, r_1) = I(r_0 - v_0, r_1 + \phi \cdot v_1) = I(r_0 - (v_0 + \delta), r_1 + \phi \cdot (v_1 + \varepsilon))$$

– do nothing (or do any deposit or redeem), otherwise.

Intuitively, condition (1) requires that the swap rate for infinitesimal amounts is greater than the exchange rate in the initial state; (2) requires that in the state  $\Gamma$  reached by performing the move of the solution, the swap rate for infinitesimal amounts tends to the exchange rate — thus achieving one of the main desiderata on AMMs. Note that  $\Gamma$  is an *equilibrium*: no move from there can improve **A**'s net worth, i.e. doing nothing is a solution for the arbitrage problem in  $\Gamma$ .

Note that for the `swapL`/`swapR` solutions, the swapped amounts are unique: this is a consequence on the assumption (10). An implicit desideratum on these solutions is that, given a specific instance of the swap invariant, they are efficiently computable: this is the case, e.g., for the constant product invariant [17].

For  $\phi = 1$  we can observe by inspection of (14) that the do-nothing solution for Theorem 2 only holds for:

$$\lim_{\varepsilon \rightarrow 0} XL_{\Gamma_0}^{\text{swap}}(\varepsilon, \tau_0, \tau_1) = 1 / \lim_{\varepsilon \rightarrow 0} XR_{\Gamma_0}^{\text{swap}}(\varepsilon, \tau_0, \tau_1) = XL_{\Gamma_0}(\tau_0, \tau_1) \quad (19)$$

Thus, the solution to the game results in the AMM tracking global exchange rates precisely: any infinitesimal deviation of the global exchange rate implies a swap action in the arbitrage game.

The assumption that the players' wallets are sufficiently large is common in formulations of the arbitrage problem. We note that any rational agent is incentivized to perform such a swap: the optimal solution to the arbitrage game can thus be approximated by multiple users exchanging smaller swap amounts. Furthermore, the availability of flash-loans [28, 29] can provide up-front funds, and thus significantly reduce the balance requirements for arbitrage swaps.

Finally, we prove that a AMM deposits and redeems do not affect the solution type of the arbitrage game. If the arbitrage solution prior to a deposit or redeem is `swapL`, `swapR` or nothing, the arbitrage solution in the subsequent state should remain of the same type.

**Theorem 3.** *Let  $I$  be incentive-consistent. Let  $(r_0 : \tau_0, r_1 : \tau_1) \in \Gamma$  with  $r_0, r_1 > 0$ . If  $\Gamma \xrightarrow{\mathbb{T}} \Gamma'$ ,  $\text{type}(\mathbb{T}) \in \{\text{dep}, \text{rdm}\}$ , then the arbitrage solutions in  $\Gamma$  and  $\Gamma'$  will have the same type or both be nothing.*

In other words, the design of AMMs aims to ensure that deposits and redeems do not interfere with the alignment of the swap rate towards the exchange rate.

*Example 8.* Consider the arbitrage game with player **B** and initial state  $\Gamma_7 = \mathbb{B}[0 : \tau_0, 27 : \tau_1] \mid (18 : \tau_0, 6 : \tau_1) \mid \dots$  resulting after the last step in Figure 1. Assuming the constant product invariant and no fees (i.e.,  $\phi = 1$ ), we have that:

$$\begin{aligned} \lim_{\varepsilon \rightarrow 0} XL_{\Gamma_7}^{\text{swap}}(\varepsilon, \tau_0, \tau_1) &= r_1/r_0 = 6/18 < 5/9 = XL_{\Gamma_7}(\tau_0, \tau_1) \\ \lim_{\varepsilon \rightarrow 0} XR_{\Gamma_7}^{\text{swap}}(\varepsilon, \tau_0, \tau_1) &= r_0/r_1 = 18/6 > 9/5 = XR_{\Gamma_7}(\tau_0, \tau_1) \end{aligned}$$

Hence, by Theorem 2 it follows that the optimal move is `swapR`( $v_0 : \tau_0, v_1 : \tau_1$ ), for suitable  $v_0$  and  $v_1$ . To find these values, we must solve for  $v_0$  and  $v_1$  the equations in item (2) of Theorem 2, i.e.:

$$\lim_{\varepsilon \rightarrow 0} XR_{\Gamma}^{\text{swap}}(\varepsilon, r_0 - v, r_1 + v_1) = XR_{\Gamma}(\tau_0, \tau_1) \quad I(r_0, r_1) = I(r_0 - v_0, r_1 + v_1)$$

Solving these equations gives:

$$v_1 = \sqrt{\frac{5}{9} \cdot r_0 r_1} - r_1 \approx 1.74 \quad v_0 = \frac{r_0 v_1}{r_1 + v_1} \approx 4$$

By performing  $\text{swapR}(v_0 : \tau_0, v_1 : \tau_1)$  with these values from  $\Gamma_7$ , we obtain:

$$\Gamma = \mathbf{B}[4 : \tau_0, 25.26 : \tau_1] \mid (14 : \tau_0, 7.74 : \tau_1) \mid \dots$$

This action maximizes  $\mathbf{B}$ 's net worth: indeed, we have  $W_{\mathbf{B}}(\Gamma_7) = 243$  and  $W_{\mathbf{B}}(\Gamma) = 247.6$ ; any other action will result in a lower net worth for  $\mathbf{B}$ .  $\square$

#### 4.4 Incentivizing deposits and swaps

Theorem 2 ensures that incentive-consistent AMMs incentivize swaps to align to exchange rates. We now show that, under certain conditions, deposits and swaps incentivize each other. The intuition is that larger amounts of tokens in an AMM provide better swap rates, therefore attracting users interested in swaps. These swaps, in turn, result in increased redeem rates, making the AMM attractive for further deposits. Note that this behaviour relies on an underlying assumption of our model, i.e. that exchange rates are stable: oracle prices are fixed. In the wild, exchange rates can vary over time, possibly making the net worth of users holding minted AMM tokens decrease: this phenomenon is commonly referred to as *impermanent loss* [9].

The following theorem shows that deposits increase swap rates, hence incentivizing swaps, whilst redeems have the opposite effect.

**Theorem 4.** *Let  $I$  be incentive-consistent. Let  $\Gamma = (r_0 : \tau_0, r_1 : \tau_1) \mid \dots$ , with  $r_0, r_1 > 0$ , and let  $\Gamma \xrightarrow{A:\ell} \Gamma'$ . Then, for all  $v \in \mathbb{R}^+$ :*

$$\begin{aligned} &XL_{\Gamma, \phi}^{\text{swap}}(v, \tau_0, \tau_1) \circ XL_{\Gamma', \phi}^{\text{swap}}(v, \tau_0, \tau_1) \\ &XR_{\Gamma, \phi}^{\text{swap}}(v, \tau_0, \tau_1) \circ XR_{\Gamma', \phi}^{\text{swap}}(v, \tau_0, \tau_1) \end{aligned} \quad \text{where } \circ = \begin{cases} < & \text{if } \ell = \text{dep}(- : \tau_0, - : \tau_1) \\ > & \text{if } \ell = \text{rdm}(- : (\tau_0, \tau_1)) \end{cases}$$

We now show that, under certain conditions, swaps incentivize deposits. Intuitively, swaps contribute to higher redeem rates, which increase the net wealth of the holders of minted AMM tokens:

**Theorem 5.** *Let  $I$  be incentive-consistent. Let  $\Gamma = (r_0 : \tau_0, r_1 : \tau_1) \mid \dots$  and  $\Gamma \rightarrow^* \Gamma'$ , where  $\Gamma' = (r'_0 : \tau_0, r'_1 : \tau_1) \mid \dots$ . If  $r_1/r_0 = r'_1/r'_0$  then:*

$$XL_{\Gamma}^{\text{rdm}}(\tau_0, \tau_1) \leq XL_{\Gamma'}^{\text{rdm}}(\tau_0, \tau_1) \quad XR_{\Gamma}^{\text{rdm}}(\tau_0, \tau_1) \leq XR_{\Gamma'}^{\text{rdm}}(\tau_0, \tau_1)$$

Recall that a user who deposits into an AMM  $(r_0 : \tau_0, r_1 : \tau_1)$  in state  $\Gamma$  receives in return an amount of minted tokens. A consequence of Theorem 5 is that these minted tokens can be redeemed with a higher redeem rate in any subsequent state  $\Gamma'$  which preserves the funds ratio  $r_1/r_0$ . Note that swaps are the only actions that may affect the redeem rate along the run  $\Gamma \rightarrow^* \Gamma'$ .



Therefore, performing swaps that eventually re-align the funds ratio to  $r_1/r_0$  incentivizes deposits.

The condition of constant funds ratio in Theorem 5 is practically relevant. For instance, for stable exchange rates, such as in the case of exchanges between stable coins [6], the arbitrage game ensures stable fund ratios: users are hence incentivized to provide funds, as the redeem rate is likely to increase over time.

## 5 Related Work

To the best of our knowledge, our work is the first to study AMMs abstracting from the swap invariant. All works in literature consider concrete swap invariants; most of them focus on the constant product, popularized by Uniswap [13]. The arbitrage problem for constant-product swap invariants has been formalized in [17, 19], which show that the solution can be efficiently computed, and suggest that constant product AMMs accurately tend towards exchange rates. Our work generalizes such results. Furthermore, as we have shown in §4.3 (19), the fee-rate  $\phi$  determines how much AMMs deviate from global exchange rates: higher fees, however, also result in reduced swap amounts in arbitrage actions, negatively affecting fee accrual. In [24], the optimal fee-rate that maximizes the fee accrual for the depositing user is analytically derived.

A executable model of Uniswap [13] has been specified in [1] to analyze integer rounding errors in the Uniswap implementation.

A few alternatives to the constant product invariant have been proposed. Curve features a peculiar invariant [22] optimized for large swap volumes between *stable coins*, where the swap rate can support large amounts with small sensitivity. To efficiently compute swap invariant, implementations perform numerical approximations [15]. Should these approximations fail to converge, these implementations still guarantee that the AMM remains liquid. We conjecture that the invariants in [3, 22] are incentive-consistent. The work [25] proposes a constant product invariant that is adjusted dynamically based on the oracle price feed, thus reducing the need for arbitrage transactions, but at the cost of lower fee accrual. AMMs with *virtual* balances have been proposed [2] and implemented [7, 8]. In these AMMs, the swap rate depends on past actions, besides the current funds balances in the AMM. This, similarly to [25], aims to minimize the need for arbitrage transactions to ensure the local AMM swap rate tends towards the exchange rates.

Some implementations [3] generalise AMM pairs to  $n$ -tokens, allowing users to swap any non-intersecting sets of token types. For example, the constant-product invariant becomes  $I(r_0, \dots, r_n) = r_0^{w_0} \cdot \dots \cdot r_n^{w_n}$  where  $\sum_{i=0}^n w_i = 1$ .

## 6 Conclusions

We have proposed a theory of AMMs, featuring a model of their behaviour and a formally proven set of fundamental properties, characterizing both structural and economic aspects. Our theory is parametric w.r.t. platform-specific features

(e.g., swap invariants), and it abstracts from implementation-specific features, and from the features that are orthogonal to the core functionality of AMMs (e.g., governance).

There are some differences between our model and the existing AMM platforms. Uniswap implements flash-loans as part of the swap actions: namely, the user can optionally borrow available pair funds [10] whilst returning these within the same *atomic group* of actions. Further, Uniswap implements an exchange rate oracle, allowing smart contracts to interpret (averages of) recent swap rates as exchange rates [11]. Balancer [3] extends token pairs to token *tuples*: a user can swap any two non-coinciding sets of supported tokens, such that the swap invariant is maintained. In all AMM implementations, token balances are represented as integers: consequently, they are subject to rounding errors [1]. AMM platforms frequently implement a governance logic, which allow “governance token” holders to coordinate changes to AMM fee-rates or swap invariant parameters.

AMM platforms like Uniswap [13] and Curve [22] have overtaken centralized cryptocurrency markets in size and usage. On the one hand, a better understanding of AMM design in cases where AMMs host the majority of the token’s global swap volume is critical [18]. It would be interesting to investigate how our theory can be used to formally explain such behaviours. On the other hand, the growth of AMMs is making them more attractive for malicious users. Current research efforts [21, 23, 27, 30] are devoted to understanding vulnerabilities and attacks, which we plan to investigate formally, exploiting our theory.

This paper, together with our work on formalizing another DeFi archetype called *lending pool* [20], is the first step towards a general theory of DeFi. We believe that a general theory encompassing interactions between different DeFi archetypes is crucial to be able to reason about their structural, economic and security aspects, as typical DeFi applications operate within a wider ecosystem, composed by a set of collaborating or competing agents, which interact through possibly separate execution environments.

**Acknowledgements** Massimo Bartoletti is partially supported by Conv. Fondazione di Sardegna & Atenei Sardi project F74I19000900007 *ADAM*. James Hsin-yu Chiang is supported by the PhD School of DTU Compute. Alberto Lluch Lafuente is partially supported by the EU H2020-SU-ICT-03-2018 Project No. 830929 CyberSec4Europe ([cybersec4europe.eu](http://cybersec4europe.eu)).

## References

1. Formal specification of constant product market maker model & implementation (2018), <https://github.com/runtimeverification/verified-smart-contracts/blob/uniswap/uniswap/x-y-k.pdf>
2. Improving frontrunning resistance of  $x*y=k$  market makers (2018), <https://ethresear.ch/t/improving-front-running-resistance-of-x-y-k-market-makers/1281>
3. Balancer whitepaper (2019), <https://balancer.finance/whitepaper/>
4. Curve statistics (2020), <https://www.curve.fi/dailystats>

5. Curve website (2020), <https://www.curve.fi>
6. Makerdao website (2020), <https://makerdao.com>
7. Mooniswap implementation (2020), <https://github.com/1inch-exchange/mooniswap/blob/02dccfab2ddb8a409400288cb13441763370350/contracts/Mooniswap.sol>
8. Mooniswap whitepaper (2020), <https://mooniswap.exchange/docs/MooniswapWhitePaper-v1.0.pdf>
9. Uniswap Documentation: Understanding Returns (2020), <https://uniswap.org/docs/v2/advanced-topics/understanding-returns/>
10. Uniswap flash loan implementation (2020), <https://github.com/Uniswap/uniswap-v2-core/blob/4dd59067c76dea4a0e8e4bfdda41877a6b16dedc/contracts/UniswapV2Pair.sol#L172>
11. Uniswap oracle template (2020), <https://github.com/Uniswap/uniswap-v2-periphery/blob/dda62473e2da448bc9cb8f4514dadda4aeede5f4/contracts/examples/ExampleOracleSimple.sol>
12. Uniswap statistics (2020), <https://info.uniswap.org>
13. Uniswap token pair implementation (2020), <https://github.com/Uniswap/uniswap-v2-core/blob/4dd59067c76dea4a0e8e4bfdda41877a6b16dedc/contracts/UniswapV2Pair.sol>
14. Uniswap website (2020), <https://www.uniswap.org>
15. Curve computation of invariant constant (2021), <https://github.com/curvefi/curve-contract/blob/a1b5a797790d3f5ef12b0e358892a0ce47c12f85/contracts/pool-templates/base/SwapTemplateBase.vy#L206>
16. Curve token pair implementation (2021), <https://github.com/curvefi/curve-contract/blob/a1b5a797790d3f5ef12b0e358892a0ce47c12f85/contracts/pool-templates/base/SwapTemplateBase.vy>
17. Angeris, G., Chitra, T.: Improved price oracles: Constant function market makers. In: ACM Conference on Advances in Financial Technologies (AFT). pp. 80–91. ACM (2020). <https://doi.org/10.1145/3419614.3423251>, <https://arxiv.org/abs/2003.10001>
18. Angeris, G., Evans, A., Chitra, T.: When does the tail wag the dog? curvature and market making. arXiv preprint arXiv:2012.08040 (2020), <https://arxiv.org/abs/2012.08040>
19. Angeris, G., Kao, H.T., Chiang, R., Noyes, C., Chitra, T.: An analysis of Uniswap markets. *Cryptoeconomic Systems Journal* (2019), <https://ssrn.com/abstract=3602203>
20. Bartoletti, M., Chiang, J.H., Lluch-Lafuente, A.: SoK: Lending pools in decentralized finance. In: Workshop on Trusted Smart Contracts. LNCS, Springer (2021), (To appear)
21. Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., Juels, A.: Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In: IEEE Symposium on Security and Privacy. pp. 910–927. IEEE (2020). <https://doi.org/10.1109/SP40000.2020.00040>
22. Egorov, M.: Stableswap - efficient mechanism for stablecoin (2019), <https://www.curve.fi/stableswap-paper.pdf>
23. Eskandari, S., Moosavi, S., Clark, J.: SoK: Transparent Dishonesty: Front-Running Attacks on Blockchain. In: *Financial Cryptography*. pp. 170–189. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-43725-1\\_13](https://doi.org/10.1007/978-3-030-43725-1_13)
24. Evans, A., Angeris, G., Chitra, T.: Optimal fees for geometric mean market makers (2021), <https://web.stanford.edu/~guillelan/papers/g3m-optimal-fee.pdf>

25. Krishnamachari, B., Feng, Q., Grippo, E.: Dynamic curves for decentralized autonomous cryptocurrency exchanges. arXiv preprint arXiv:2101.02778 (2021), <https://arxiv.org/abs/2101.02778>
26. Mazurkiewicz, A.W.: Basic notions of trace theory. In: Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency. LNCS, vol. 354, pp. 285–363. Springer (1988). <https://doi.org/10.1007/BFb0013025>
27. Qin, K., Zhou, L., Gervais, A.: Quantifying blockchain extractable value: How dark is the forest? (2021), <https://arxiv.org/abs/2101.05511>
28. Qin, K., Zhou, L., Livshits, B., Gervais: Attacking the DeFi Ecosystem with Flash Loans for Fun and Profit. In: Financial Cryptography (2021), (to appear) <https://arxiv.org/abs/2003.03810>
29. Wang, D., Wu, S., Lin, Z., Wu, L., Yuan, X., Zhou, Y., Wang, H., Ren, K.: Towards understanding flash loan and its applications in defi ecosystem. arXiv preprint arXiv:2010.12252 (2020), <https://arxiv.org/abs/2010.12252>
30. Zhou, L., Qin, K., Torres, C.F., Le, D.V., Gervais, A.: High-Frequency Trading on Decentralized On-Chain Exchanges. arXiv preprint arXiv:2009.14021 (2020), <https://arxiv.org/abs/2009.14021>

## A Proofs

**Proof of Lemma 1** Straightforward inspection of the rules in §2.  $\square$

**Proof of Lemma 2** We proceed by cases on the rule used in the last transition: it is straightforward to check that, in all the rules, changes applied to initial tokens cancel out.  $\square$

**Proof of Lemma 3** We first prove that the net worth of a user  $\mathbf{A}$ , i.e.  $W_{\mathbf{A}}(\Gamma)$ , is preserved by transactions of type **dep** and **rdm**, or by transactions which do not affect  $\mathbf{A}$ 's wallet. More precisely, we prove that:

$$\Gamma \xrightarrow{\mathbb{T}} \Gamma' \wedge (\text{type}(\mathbb{T}) \in \{\text{dep}, \text{rdm}\} \vee \mathbf{A} \notin \text{wal}(\mathbb{T})) \implies W_{\mathbf{A}}(\Gamma) = W_{\mathbf{A}}(\Gamma') \quad (20)$$

We have the following cases:

- [XFER]. Let  $\mathbf{C} : \text{xfer}(\mathbf{B}, v : \tau)$  be the fired transaction. By hypothesis, it must be  $\mathbf{A} \notin \text{wal}(\mathbb{T}) = \{\mathbf{B}, \mathbf{C}\}$ , hence the thesis follows by inspection of the rule.
- [DEP0]. Let  $\mathbf{B} : \text{dep}(v_0 : \tau_0, v_1 : \tau_1)$  be the fired transaction. We have that:

$$\begin{aligned} \Gamma &= \mathbf{B}[\sigma] \mid \Gamma_0 \\ \Gamma' &= \mathbf{B}[\sigma - v_0 : \tau_0 - v_1 : \tau_1 + v_0 : (\tau_0, \tau_1)] \mid (v_0 : \tau_0, v_1 : \tau_1) \mid \Gamma_0 \end{aligned}$$

If  $\mathbf{B} \neq \mathbf{A}$ , then  $\mathbf{A}$ 's net worth is unaffected. Otherwise, if  $\mathbf{B} = \mathbf{A}$ , then:

$$\begin{aligned} W_{\mathbf{A}}(\Gamma') &= W_{\mathbf{A}}(\Gamma) - v_0 P_{\tau_0}(\Gamma) - v_1 P_{\tau_1}(\Gamma) + v_0 P_{(\tau_0, \tau_1)}(\Gamma) \\ &= W_{\mathbf{A}}(\Gamma) - v_0 P_{\tau_0}(\Gamma) - v_1 P_{\tau_1}(\Gamma) + v_0 \frac{v_0 P_{\tau_0}(\Gamma) + v_1 P_{\tau_1}(\Gamma)}{v_0} \\ &= W_{\mathbf{A}}(\Gamma) \end{aligned}$$

- [DEP]. Let  $\mathbf{B} : \text{dep}(v_0 : \tau_0, v_1 : \tau_1)$  be the fired transaction. We have that:

$$\begin{aligned} \Gamma &= \mathbf{B}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma_0 \\ \Gamma' &= \mathbf{B}[\sigma - v_0 : \tau_0 - v_1 : \tau_1 + v : (\tau_0, \tau_1)] \mid (r_0 + v_0 : \tau_0, r_1 + v_1 : \tau_1) \mid \Gamma_0 \end{aligned}$$

where  $v = (v_0 \cdot s)/r_0$ , with  $s = \text{sply}_{(\tau_0, \tau_1)}(\Gamma)$ . If  $\mathbf{B} \neq \mathbf{A}$ , then  $\mathbf{A}$ 's net worth is unaffected. Otherwise, if  $\mathbf{B} = \mathbf{A}$ , then:

$$\begin{aligned} W_{\mathbf{A}}(\Gamma') &= W_{\mathbf{A}}(\Gamma) - v_0 P_{\tau_0}(\Gamma) - v_1 P_{\tau_1}(\Gamma) + v P_{(\tau_0, \tau_1)}(\Gamma) \\ &= W_{\mathbf{A}}(\Gamma) - v_0 P_{\tau_0}(\Gamma) - v_1 P_{\tau_1}(\Gamma) + v \frac{r_0 P_{\tau_0}(\Gamma) + r_1 P_{\tau_1}(\Gamma)}{s} \\ &= W_{\mathbf{A}}(\Gamma) - v_0 P_{\tau_0}(\Gamma) - v_1 P_{\tau_1}(\Gamma) + \frac{v_0}{r_0} \left( r_0 P_{\tau_0}(\Gamma) + r_1 P_{\tau_1}(\Gamma) \right) \\ &= W_{\mathbf{A}}(\Gamma) - v_1 P_{\tau_1}(\Gamma) + \frac{v_0}{r_0} r_1 P_{\tau_1}(\Gamma) \\ &= W_{\mathbf{A}}(\Gamma) \end{aligned}$$

where the last equality follows from the premise  $r_1 v_0 = r_0 v_1$  of rule [DEP].

- [SWAPL]. Let  $\mathbf{B} : \text{swapl}(v_0 : \tau_0, v_1 : \tau_1)$  be the fired transaction. By hypothesis, it must be  $\mathbf{A} \notin \text{wal}(\mathbf{T}) = \{\mathbf{B}\}$ . The thesis follows by inspection of the rule.
- [SWAPR]. Similar to the previous case.
- [RDM]. Let  $\mathbf{B} : \text{rdm}(v : (\tau_0, \tau_1))$  be the fired transaction. We have that:

$$\begin{aligned} \Gamma &= \mathbf{B}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma_0 \\ \Gamma' &= \mathbf{B}[\sigma + v_0 : \tau_0 + v_1 : \tau_1 - v : (\tau_0, \tau_1)] \mid (r_0 - v_0 : \tau_0, r_1 - v_1 : \tau_1) \mid \Gamma_0 \end{aligned}$$

where  $v_0 = (vr_0)/s$  and  $v_1 = (vr_1)/s$ , with  $s = \text{sply}_{(\tau_0, \tau_1)}(\Gamma)$ . If  $\mathbf{B} \neq \mathbf{A}$ , then  $\mathbf{A}$ 's net worth is unaffected. Otherwise, if  $\mathbf{B} = \mathbf{A}$ , then:

$$\begin{aligned} W_{\mathbf{A}}(\Gamma') &= W_{\mathbf{A}}(\Gamma) + v_0 P_{\tau_0}(\Gamma) + v_1 P_{\tau_1}(\Gamma) - v P_{(\tau_0, \tau_1)}(\Gamma) \\ &= W_{\mathbf{A}}(\Gamma) + v_0 P_{\tau_0}(\Gamma) + v_1 P_{\tau_1}(\Gamma) - \frac{vr_0}{s} P_{\tau_0}(\Gamma) - \frac{vr_1}{s} P_{\tau_1}(\Gamma) \\ &= W_{\mathbf{A}}(\Gamma) + \frac{vr_0}{s} P_{\tau_0}(\Gamma) + \frac{vr_1}{s} P_{\tau_1}(\Gamma) - \frac{vr_0}{s} P_{\tau_0}(\Gamma) - \frac{vr_1}{s} P_{\tau_1}(\Gamma) \\ &= W_{\mathbf{A}}(\Gamma) \quad \square \end{aligned}$$

We now prove that the global net worth is preserved by *any* transactions. First, we recall the definition of global net worth. Let:

$$\Gamma = \mathbf{A}_1[\sigma_1] \mid \cdots \mid \mathbf{A}_n[\sigma_n] \mid (r_1 : \tau_1, r'_1 : \tau'_1) \mid \cdots \mid (r_k : \tau_k, r'_k : \tau'_k)$$

The global net worth of  $\Gamma$  is defined as:

$$W(\Gamma) = \sum_{i \in 1..n} W_{\mathbf{A}_i}(\Gamma) \quad (21)$$

We prove that:

$$\Gamma \xrightarrow{\mathbf{T}} \Gamma' \implies W(\Gamma) = W(\Gamma') \quad (22)$$

We have the following cases:

- [XFER]. A *xfer* transaction subtracts an amount of tokens from the wallet of a user, and adds an equal amount of the same token type to the wallet of another user. Therefore, it preserves the global net worth.
- [DEP0], [DEP], [RDM]. These rules affect the number of tokens in AMMs, which does not contribute to the global net worth, and the balances of users, which are preserved (20). Therefore, the global net worth is preserved.
- [SWAPL]. We have that:

$$\begin{aligned} \Gamma &= \mathbf{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma_0 \\ \Gamma' &= \mathbf{A}[\sigma - v_0 : \tau_0 + v : \tau_1] \mid (r_0 + v_0 : \tau_0, r_1 - v : \tau_1) \mid \Gamma_0 \end{aligned}$$

Note that  $sply_{\tau_0}(\Gamma') = sply_{\tau_0}(\Gamma)$ , and  $sply_{\tau}(\Gamma') = sply_{\tau}(\Gamma)$ . By (8), we have that  $P_{\tau_0}(\Gamma') = P_{\tau_0}(\Gamma)$ . Therefore:

$$\begin{aligned}
W(\Gamma') &= W(\Gamma_0) - v_0 P_{\tau_0}(\Gamma') + v P_{\tau_1}(\Gamma') \\
&\quad - sply_{(\tau_0, \tau_1)}(\Gamma) P_{(\tau_0, \tau_1)}(\Gamma) + sply_{(\tau_0, \tau_1)}(\Gamma') P_{(\tau_0, \tau_1)}(\Gamma') \\
&= W(\Gamma_0) - v_0 P_{\tau_0}(\Gamma) + v P_{\tau_1}(\Gamma) \\
&\quad - sply_{(\tau_0, \tau_1)}(\Gamma) \cdot (P_{(\tau_0, \tau_1)}(\Gamma) + P_{(\tau_0, \tau_1)}(\Gamma')) \\
&= W(\Gamma_0) - v_0 P_{\tau_0}(\Gamma) + v P_{\tau_1}(\Gamma) \\
&\quad - r_0 P_{\tau_0}(\Gamma) + r_1 P_{\tau_1}(\Gamma) + (r_0 + v_0) P_{\tau_0}(\Gamma) + (r_1 - v) P_{\tau_1}(\Gamma) \\
&= W(\Gamma)
\end{aligned}$$

– [SWAPR]. Similar to the previous case.  $\square$

**Proof of Lemma 4** For item (a) we proceed by induction on the length of the computation  $\Gamma_0 \rightarrow^* \Gamma$ , with  $\Gamma_0$  initial. The base case is trivial, since  $\Gamma_0$  contains no AMMs. For the inductive step, we must show that (a) is preserved by single transitions. Assume that  $\Gamma$  satisfies (a), and that  $\Gamma \rightarrow \Gamma'$ . Assume that  $\Gamma'$  contains an AMM  $(r'_0 : \tau_0, r'_1 : \tau_1)$  with  $r'_0 + r'_1 > 0$ , i.e.  $r'_0 > 0$  or  $r'_1 > 0$ . There are the following cases, depending on the rule used to infer  $\Gamma \rightarrow \Gamma'$ :

- [XFER], [SWAPL], [SWAPR]. trivial, because these actions do not affect the supply of minted tokens.
- [DEP0], [DEP]. trivial, because deposits increase the supply of minted tokens.
- [RDM]. Assume that the state of the AMM in  $\Gamma$  is  $(r_0 : \tau_0, r_1 : \tau_1)$ . By contradiction, suppose that the [RDM] action burns all the  $v = sply_{(\tau_0, \tau_1)}(\Gamma)$  units of the minted token. The rule premise requires  $v > 0$ , and it implies:

$$r'_0 = r_0 - v \frac{r_0}{sply_{(\tau_0, \tau_1)}(\Gamma)} = 0 \quad r'_1 = r_1 - v \frac{r_1}{sply_{(\tau_0, \tau_1)}(\Gamma)} = 0$$

Therefore, we would have  $r'_0 = r'_1 = 0$  — contradiction.

For item (b), we proceed by induction on the number of AMMs in the state. The base case is trivial, since the premise of (b) is false. For the inductive step, let  $\Gamma = (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma''$  be a reachable state such that  $r_0 + r_1 > 0$ . Let  $r'_0 \leq r_0$ , and let  $v = sply_{(\tau_0, \tau_1)}(\Gamma)$ . By item (a) we have that  $v > 0$ . By the induction hypothesis, there exists a sequence of transactions  $\Gamma'' \rightarrow^* \Gamma_1$  where users can redeem  $v : (\tau_0, \tau_1)$  from other AMMs. From  $\Gamma_1$ , users can perform another sequence of transactions  $\Gamma_1 \rightarrow^* \Gamma_2$  to transfer these units to the wallet of a user, say **A**. Summing up, we have that  $\Gamma \rightarrow^* (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma_2$ . Now, let:

$$\tilde{v} = \frac{(r_0 - r'_0)}{r_0} sply_{(\tau_0, \tau_1)}(\Gamma)$$

Since  $\tilde{v} < v$ ,  $\mathbf{A}$  can fire  $\mathbf{A} : \text{rdm}(\tilde{v} : (\tau_0, \tau_1))$ , obtaining, for some  $r'_1 \leq r_1$ :

$$\begin{aligned} (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma_2 &\rightarrow \Gamma' = (r_0 - \tilde{v} \frac{r_0}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)} : \tau_0, r'_1 : \tau_1) \mid \dots \\ &= (r'_0 : \tau_0, r'_1 : \tau_1) \mid \dots \end{aligned}$$

The proof of item (c) is similar.  $\square$

**Proof of Lemma 5** Let  $\Gamma \xrightarrow{\mathbf{T}_0 \mathbf{T}_1} \Gamma_{01}$  and  $\Gamma \xrightarrow{\mathbf{T}_1 \mathbf{T}_0} \Gamma_{10}$  with the following intermediate states:

$$\Gamma \xrightarrow{\mathbf{T}_0} \Gamma_0 \xrightarrow{\mathbf{T}_1} \Gamma_{01} \quad \Gamma \xrightarrow{\mathbf{T}_1} \Gamma_1 \xrightarrow{\mathbf{T}_0} \Gamma_{10}$$

We have the following exhaustive cases on the type of the two transactions:

1.  $\mathbf{T}_0 = \mathbf{A}_0 : \text{xfer}(\mathbf{B}_0, v_0 : \tau_0)$ .
  - (a)  $\mathbf{T}_1 = \mathbf{A}_1 : \text{xfer}(\mathbf{B}_1, v_1 : \tau_1)$ . Straightforward from rule  $[\text{XFER}]$ .
  - (b)  $\mathbf{T}_1 = \mathbf{A}_1 : \text{dep}(v_1 : \tau_1, v'_1 : \tau'_1)$ . We have that:

$$\begin{aligned} \Gamma &= \mathbf{A}_0[\sigma_{\mathbf{A}_0}] \mid \mathbf{B}_0[\sigma_{\mathbf{B}_0}] \mid \Gamma' \\ \Gamma_0 &= \mathbf{A}_0[\sigma_{\mathbf{A}_0} - v_0 : \tau_0] \mid \mathbf{B}_0[\sigma_{\mathbf{B}_0} + v_0 : \tau_0] \mid \Gamma' \end{aligned}$$

If  $\mathbf{A}_1 \neq \mathbf{A}_0, \mathbf{B}_0$ , then the two transactions affect different part of the state, so the thesis follows trivially. Otherwise, we have two subcases, depending on whether  $[\text{DEP0}]$  or  $[\text{DEP}]$  was used.

If  $[\text{DEP0}]$  was used, then:

$$\Gamma_1 = \mathbf{A}_1[\sigma_{\mathbf{A}_1} - v_1 : \tau_1 - v'_1 : \tau'_1 + v_1 : (\tau_1, \tau'_1)] \mid (v_1 : \tau_1, v'_1 : \tau'_1) \mid \Gamma'$$

If  $\mathbf{A}_1 = \mathbf{A}_0$ , then  $\sigma_{\mathbf{A}_0} = \sigma_{\mathbf{A}_1}$ , and the thesis  $\Gamma_{01} = \Gamma_{10}$  follows by:

$$\begin{aligned} \Gamma_{01} &= \mathbf{A}_0[\sigma_{\mathbf{A}_0} - v_0 : \tau_0 - v_1 : \tau_1 - v'_1 : \tau'_1 + v_1 : (\tau_1, \tau'_1)] \mid \\ &\quad \mathbf{B}_0[\sigma_{\mathbf{B}_0} + v_0 : \tau_0] \mid (v_1 : \tau_1, v'_1 : \tau'_1) \mid \Gamma' \\ \Gamma_{10} &= \mathbf{A}_0[\sigma_{\mathbf{A}_0} - v_1 : \tau_1 - v'_1 : \tau'_1 + v_1 : (\tau_1, \tau'_1) - v_0 : \tau_0] \mid \\ &\quad \mathbf{B}_0[\sigma_{\mathbf{B}_0} + v_0 : \tau_0] \mid (v_1 : \tau_1, v'_1 : \tau'_1) \mid \Gamma' \end{aligned}$$

If  $\mathbf{A}_1 = \mathbf{B}_0$ , then  $\sigma_{\mathbf{B}_0} = \sigma_{\mathbf{A}_1}$ , and the thesis  $\Gamma_{01} = \Gamma_{10}$  follows by:

$$\begin{aligned} \Gamma_{01} &= \mathbf{A}_0[\sigma_{\mathbf{A}_0} - v_0 : \tau_0] \mid \\ &\quad \mathbf{B}_0[\sigma_{\mathbf{B}_0} + v_0 : \tau_0 - v_1 : \tau_1 - v'_1 : \tau'_1 + v_1 : (\tau_1, \tau'_1)] \mid (v_1 : \tau_1, v'_1 : \tau'_1) \mid \Gamma' \\ \Gamma_{10} &= \mathbf{A}_0[\sigma_{\mathbf{A}_0} - v_0 : \tau_0] \mid \\ &\quad \mathbf{B}_0[\sigma_{\mathbf{B}_0} - v_1 : \tau_1 - v'_1 : \tau'_1 + v_1 : (\tau_1, \tau'_1) + v_0 : \tau_0] \mid (v_1 : \tau_1, v'_1 : \tau'_1) \mid \Gamma' \end{aligned}$$

The case where  $[\text{DEP}]$  was used is similar. In the following cases we omit the detailed computation of the states as done here, and we appeal the intuition to argue for the equivalence of the reached states.



- (c)  $T_1 = A_1 : \text{swapL}(v_1 : \tau_1, v'_1 : \tau'_1)$ . The equality of  $\Gamma_{01}$  and  $\Gamma_{10}$  follows from the fact that neither the updated users' balances nor the amounts which update the AMM depend on the current state. In particular, the swap invariant only depends on the values  $r_1$  and  $r'_1$ .
  - (d)  $T_1 = A_1 : \text{swapR}(v_1 : \tau_1, v'_1 : \tau'_1)$ . Similar to the previous case.
  - (e)  $T_1 = A_1 : \text{rdm}(v_1 : \tau_1)$ . The thesis follows from the fact that the only part of the state which affects the update of the users' balances and the AMM is the supply of the minted token, which is not affected by  $T_0$ .
2.  $T_0 = A_0 : \text{dep}(v_0 : \tau_0, v'_0 : \tau'_0)$ .
    - (a)  $T_1 = A_1 : \text{xfer}(B_1, v_1 : \tau_1)$ . Symmetric to case (1b).
    - (b)  $T_1 = A_1 : \text{dep}(v_1 : \tau_1, v'_1 : \tau'_1)$ . If  $(\tau_0, \tau'_0) \neq (\tau_1, \tau'_1)$ , then the thesis is straightforward. Otherwise, note that the amount of tokens minted by the AMM only depends on the ratio between the amounts of  $\tau_0$  and  $\tau'_0$  initially held by the AMM, and the values subsequently transferred to the AMM by  $T_0$  and  $T_1$ .
    - (c)  $T_1 = A_1 : \text{swapL}(v_1 : \tau_1, v'_1 : \tau'_1)$ . By hypothesis,  $\{\tau_1, \tau'_1\}$  and  $\{\tau_0, \tau'_0\}$  are disjoint. The thesis is straightforward, because AMMs operating on distinct token pairs do not interfere.
    - (d)  $T_1 = A_1 : \text{swapR}(v_1 : \tau_1, v'_1 : \tau'_1)$ . Similar to the previous case.
    - (e)  $T_1 = A_1 : \text{rdm}(v_1 : \tau_1)$ . If  $(\tau_0, \tau'_0) \neq (\tau_1, \tau'_1)$ , then the thesis is straightforward. Otherwise, note that the amount of tokens minted by the AMM in response of  $T_0$  only depends on the ratio between the amounts of  $\tau_0$  and  $\tau'_0$  initially held by the AMM, and the values transferred to the AMM by  $T_0$ . Similarly, the tokens paid by the AMM in response of  $T_1$  only depend on the ratio between the amounts of  $\tau_0$  and  $\tau'_0$  initially held by the AMM, which are constrained to preserve the ratio, and the amount deposited by  $T_0$ . Then, the two transactions are concurrent.
  3.  $T_0 = A_0 : \text{rdm}(v_0 : \tau_0)$ .
    - (a)  $T_1 = A_1 : \text{xfer}(B_1, v_1 : \tau_1)$ . Symmetric to case (1e).
    - (b)  $T_1 = A_1 : \text{dep}(v_1 : \tau_1, v'_1 : \tau'_1)$ . Symmetric to case (2e).
    - (c)  $T_1 = A_1 : \text{swapL}(v_1 : \tau_1, v'_1 : \tau'_1)$ . By hypothesis,  $\{\tau_1, \tau'_1\}$  and  $\{\tau_0, \tau'_0\}$  are disjoint. The thesis is straightforward, because AMMs operating on distinct token pairs do not interfere.
    - (d)  $T_1 = A_1 : \text{swapR}(v_1 : \tau_1, v'_1 : \tau'_1)$ . Similar to the previous case.
    - (e)  $T_1 = A_1 : \text{rdm}(v_1 : \tau_1)$ . The tokens paid by the AMM in response of  $T_0$  and  $T_1$  only depend on the ratio between the amounts of  $\tau_0$  and  $\tau'_0$  initially held by the AMM, which are constrained to preserve the ratio.
  4.  $T_0 = A_0 : \text{swapL}(v_0 : \tau_0, v'_0 : \tau'_0)$ . The only case not covered by the previous one is when  $T_1 = A_1 : \text{swapL}(v_1 : \tau_1, v'_1 : \tau'_1)$ . By hypothesis,  $\{\tau_1, \tau'_1\}$  and  $\{\tau_0, \tau'_0\}$  are disjoint. The thesis is straightforward, because AMMs operating on distinct token pairs do not interfere.
  5.  $T_0 = A_0 : \text{swapR}(v_0 : \tau_0, v'_0 : \tau'_0)$ . Similar to the previous case. □

**Lemma A.1**  $\lambda \sim \lambda' \implies \forall \lambda_0, \lambda_1 : \lambda_0 \lambda \lambda_1 \sim \lambda_0 \lambda' \lambda_1$ .

*Proof.* Direct from the fact that semantics of transactions is a function (Lemma 1), and it only depends on the states after the execution of  $\lambda$  and  $\lambda'$ , which are equal starting from any state  $\Gamma$ , since  $\lambda \sim \lambda'$ .

**Proof of Theorem 1** By definition,  $\sim_{\#}$  is the least equivalence relation closed under the following rules (where  $\varepsilon$  denotes the empty sequence):

$$\frac{}{\varepsilon \sim_{\#} \varepsilon}^{[0]} \quad \frac{}{\overline{\mathbb{T}} \sim_{\#} \overline{\mathbb{T}}}^{[1]} \quad \frac{\mathbb{T} \# \mathbb{T}'}{\overline{\mathbb{T}\mathbb{T}'} \sim_{\#} \overline{\mathbb{T}'\mathbb{T}}}^{[2]} \quad \frac{\lambda_0 \sim_{\#} \lambda'_0 \quad \lambda_1 \sim_{\#} \lambda'_1}{\lambda_0 \lambda_1 \sim_{\#} \lambda'_0 \lambda'_1}^{[3]}$$

Let  $\lambda \sim_{\#} \lambda'$ . We have to show  $\lambda \sim \lambda'$ . We proceed by induction on the rules above. For rules [0] and [1], the thesis follows by reflexivity, since  $\sim$  is an equivalence relation. For rule [2], the thesis follows immediately by definition of the concurrency relation  $\#$ . For rule [3], first note that  $\lambda = \lambda_0 \lambda_1$  and  $\lambda' = \lambda'_0 \lambda'_1$ . By the induction hypothesis it follows that:

$$\lambda_0 \sim \lambda'_0 \quad \text{and} \quad \lambda_1 \sim \lambda'_1$$

Therefore, by two applications of Lemma A.1:

$$\lambda = \lambda_0 \lambda_1 \sim \lambda_0 \lambda'_1 \sim \lambda'_0 \lambda'_1 = \lambda' \quad \square$$

**Proof of Lemma 6** Define the *rank*  $\|\tau\|$  of a token  $\tau$  inductively as follows:

$$\|\tau\| = \begin{cases} 0 & \text{if } \tau \in \mathbb{T}_0 \\ \max\{\|\tau_0\|, \|\tau_1\|\} & \text{if } \tau = (\tau_0, \tau_1) \end{cases}$$

and accordingly define the rank of  $\mathbf{A}$  in  $\Gamma$  as:

$$\|\Gamma\|_{\mathbf{A}} = \begin{cases} \max\{\|\tau\| \mid \tau \in \text{dom } \sigma\} & \text{if } \mathbf{A}[\sigma] \in \Gamma \\ 0 & \text{otherwise} \end{cases}$$

We proceed by induction on  $n = \|\Gamma\|_{\mathbf{A}}$ . In the base case  $n = 0$ ,  $\mathbf{A}$ 's wallet in  $\Gamma$  contains only initial tokens: here,  $\text{rdm}_{\mathbf{A}}(\Gamma) = \Gamma$ , so the thesis is immediate. For the inductive case, let  $n > 0$ , and assume that the statement holds for all  $n' < n$ . Let  $\tau_1, \dots, \tau_k$  be the tokens of rank  $n$  in  $\mathbf{A}$ 's wallet  $\mathbf{A}[\sigma]$ . Let:

$$\Gamma = \Gamma_0 \xrightarrow{\mathbf{A}:\text{rdm}(\sigma_0(\tau_1):\tau_1)} \Gamma_1 \xrightarrow{\mathbf{A}:\text{rdm}(\sigma_1(\tau_2):\tau_2)} \dots \xrightarrow{\mathbf{A}:\text{rdm}(\sigma_{k-1}(\tau_k):\tau_k)} \Gamma_k$$

where  $\Gamma_i = \mathbf{A}[\sigma_i] \mid \Gamma'_i$  for all  $i < k$ . The rank of each token in  $\mathbf{A}$ 's wallet in  $\Gamma_k$  is strictly less than  $n$ . Therefore, by the induction hypothesis:

$$W_{\mathbf{A}}(\Gamma_k) = \sum_{\tau \in \text{dom } \sigma_k} \sigma_k(\tau) \cdot P^0(\tau)$$

By Lemma 3, we have that  $W_{\mathbf{A}}(\Gamma_0) = W_{\mathbf{A}}(\Gamma_k)$ , from which the thesis follows.  $\square$

**Lemma A.2** *Let  $I$  be swap-rate continuous. Then, for all  $v_0, v_1$  and  $0 < \phi \leq 1$ :*

$$\begin{aligned} XL^{\text{swap}}(v_0, r_0, r_1) &\geq XL_{\phi}^{\text{swap}}(v_0, r_0, r_1) \\ XR^{\text{swap}}(v_1, r_0, r_1) &\geq XR_{\phi}^{\text{swap}}(v_1, r_0, r_1) \end{aligned}$$

*Proof.* For  $\phi = 1$  the statement holds trivially, since  $XL^{\text{swap}}(\varepsilon, r_0, r_1)$  is a shorthand for  $XL_{\phi=1}^{\text{swap}}(\varepsilon, r_0, r_1)$ . So, let  $\phi < 1$ . By (12), for any  $v_0$  we have that:

$$XL_{\phi}^{\text{swap}}(v_0, r_0, r_1) = \frac{v_{\phi}}{v_0} \quad \text{where } \exists! v_{\phi} : I(r_0, r_1) = I(r_0 + \phi v_0, r_1 - v_{\phi}) \quad (23)$$

$$XL^{\text{swap}}(v_0, r_0, r_1) = \frac{v}{v_0} \quad \text{where } \exists! v : I(r_0, r_1) = I(r_0 + v_0, r_1 - v) \quad (24)$$

We show that  $v > v_{\phi}$ , from which the thesis follows. By (24), we have that the left swap rate for input  $(1 - \phi)v_0$  on the AMM  $(r_0 + \phi v_0 : \tau_0, r_1 - v_{\phi} : \tau_1)$  is:

$$XL^{\text{swap}}((1 - \phi)v_0, r_0 + \phi v_0, r_1 - v_{\phi}) = \frac{x}{(1 - \phi)v_0} \quad (25)$$

where  $x$  is the unique value satisfying:

$$\begin{aligned} I(r_0 + \phi v_0, r_1 - v_{\phi}) &= I(r_0 + \phi v_0 + (1 - \phi)v_0, r_1 - v_{\phi} - x) \\ &= I(r_0 + v_0, r_1 - (v_{\phi} + x)) \end{aligned}$$

Since, by (23),  $I(r_0 + \phi v_0, r_1 - v_{\phi}) = I(r_0, r_1)$ , then  $x$  must satisfy:

$$I(r_0 + v_0, r_1 - (v_{\phi} + x)) = I(r_0, r_1)$$

By (24), the unique solution  $y$  to the equation  $I(r_0, r_1) = I(r_0 + v_0, r_1 - y)$  is  $y = v$ . Therefore, it must be  $v = v_{\phi} + x$ , from which we obtain  $x = v - v_{\phi}$ . Since  $I$  is swap-rate continuous, by (14) it follows that the swap rate (25) must be positive, and so  $v - v_{\phi} > 0$ . The proof for the right swap rate is similar.  $\square$

**Proof of Theorem 2** Let  $\Gamma_0 = \mathbf{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1)$ . The statement of the theorem singles out three cases for the solutions of the arbitrage game:

- (a)  $\lim_{\varepsilon \rightarrow 0} XL_{\phi}^{\text{swap}}(\varepsilon, r_0, r_1) > XL_{\Gamma_0}(\tau_0, \tau_1)$
- (b)  $\lim_{\varepsilon \rightarrow 0} XR_{\phi}^{\text{swap}}(\varepsilon, r_0, r_1) > XR_{\Gamma_0}(\tau_0, \tau_1)$
- (c) otherwise

These cases are trivially exhaustive; we now show that they are also mutually exclusive. Assume that condition (a) holds. By Lemma A.2, we have that:

$$\forall \varepsilon > 0 : XL^{\text{swap}}(\varepsilon, r_0, r_1) \geq XL_{\phi}^{\text{swap}}(\varepsilon, r_0, r_1)$$

Therefore, by the limit inequality theorem and by (a):

$$\lim_{\varepsilon \rightarrow 0} XL^{\text{swap}}(\varepsilon, r_0, r_1) \geq \lim_{\varepsilon \rightarrow 0} XL_{\phi}^{\text{swap}}(\varepsilon, r_0, r_1) > XL_{\Gamma_0}(\tau_0, \tau_1)$$

Since  $I$  is swap-rate-continuous (14), this implies that:

$$1 / \lim_{\varepsilon \rightarrow 0} XR^{\text{swap}}(\varepsilon, r_0, r_1) > XR_{\Gamma_0}(\tau_0, \tau_1)$$

from which by (11) it follows that:

$$\lim_{\varepsilon \rightarrow 0} XR^{\text{swap}}(\varepsilon, r_0, r_1) < XR_{\Gamma_0}(\tau_0, \tau_1)$$

which contradicts condition (b). A similar argument can be used to show that if condition (b) is true, then condition (a) is false. Therefore, the two conditions are mutually exclusive.

Now, assume that case (a) applies. We prove that the solution prescribed by Theorem 2 for this case is optimal. In order for the net worth of user A to increase with a left swap, the following must hold:

$$\begin{aligned} W_A(\Gamma) - W_A(\Gamma_0) &= -v_0 \cdot P(\tau_0) + v_0 \cdot XL_{\Gamma_0, \phi}^{\text{swap}}(v_0, \tau_0, \tau_1) \cdot P(\tau_1) > 0 \\ XL_{\Gamma_0, \phi}^{\text{swap}}(v_0, \tau_0, \tau_1) &> P(\tau_0)/P(\tau_1) \end{aligned} \quad (26)$$

Given case (a), there must exist a finite  $v_0$  which satisfies (26). Demand-sensitivity (15) ensures that  $XL_{\Gamma_0, \phi}^{\text{swap}}(v_0, \tau_0, \tau_1)$  decreases to zero with increasing  $v_0$ . To find optimal  $v_0$  which maximizes the user net worth, we consider the behaviour of  $XL_{\Gamma_0, \phi}^{\text{swap}}(v_0, \tau_0, \tau_1) = XL^{\text{swap}}(v_0, r_0, r_1)$  as  $v_0$  is increased by  $\Delta$  for  $(r_0 : \tau_0, r_1 : \tau_1) \in \Gamma_0$ :

$$\begin{aligned} XL_{\phi}^{\text{swap}}(v_0, r_0, r_1) &= \frac{v}{\phi \cdot v_0} & XL_{\phi}^{\text{swap}}(v_0 + \Delta, r_0, r_1) &= \frac{v + \Delta'}{\phi \cdot (v_0 + \Delta)} \\ && \text{where } \exists! v, \Delta' : & \end{aligned} \quad (27)$$

$$\textcircled{1} I(r_0 + \phi \cdot v_0, r_1 - v) = \textcircled{2} I(r_0 + \phi \cdot (v_0 + \Delta), r_1 - v - \Delta') = I(r_0, r_1)$$

From  $\textcircled{1}$  and  $\textcircled{2}$  and swap rate definition (12) we can infer:

$$XL_{\phi}^{\text{swap}}(\Delta, r_0 + \phi \cdot v_0, r_1 - v) = \Delta'/\Delta \quad (28)$$

For an incremental increase in swap amount  $v_0$  by  $\Delta$ , the net wealth is only further increased if  $\Delta'/\Delta > P(\tau_0)/P(\tau_1)$  (26). Due to demand-sensitivity, this ratio is ever decreasing at higher values of  $v_0$ . Thus, for optimal  $v_0$  in  $XL_{\phi}^{\text{swap}}(v_0, r_0, r_1)$ :

$$\lim_{\varepsilon \rightarrow 0} XL_{\phi}^{\text{swap}}(\varepsilon, r_0 + \phi \cdot v_0, r_1 - v) = XL_{\Gamma}(\tau_0, \tau_1) \quad (29)$$

Any infinitesimal increase in  $v_0$  no longer increases net wealth. We omit the proof for case (b) as it follows that of (a).

To prove case (c), we observe that since neither (a) nor (b) hold, there exists no swap which results in a net positive increase in net wealth. Thus, the strategy of the user will be to perform no action at all.  $\square$

**Proof of Theorem 3** We demonstrate that the same cases in Theorem 2 apply to both  $\Gamma, \Gamma'$  where  $\Gamma \xrightarrow{\mathbb{T}} \Gamma'$  and  $\text{type}(\mathbb{T}) \in \{\text{dep}, \text{rdm}\}$ . For a  $\Gamma = (r_0, \tau_0, r_1, \tau_1) \mid \dots$

- [DEP] maintains the ratio of holdings  $(\tau_0, \tau_1)$ , such that  $\Gamma' = (c \cdot r_0, \tau_0, c \cdot r_1, \tau_1) \mid \dots$  where  $c > 1$

- [RDM] maintains the ratio of holdings  $(\tau_0, \tau_1)$ , such that  $\Gamma' = (c' \cdot r_0, \tau_0, c' \cdot r_1, \tau_1) \mid \dots$  where

$$c' = 1 - \frac{v}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)} \quad (30)$$

By SR-consistency (18), for  $c'' > 0$

$$\lim_{\varepsilon \rightarrow 0} XL_{\phi}^{\text{swap}}(\varepsilon, r_0, r_1) = \lim_{\varepsilon \rightarrow 0} XL_{\phi}^{\text{swap}}(\varepsilon, c'' \cdot r_0, c'' \cdot r_1) \quad (31)$$

and since we have shown that the ratio of funds is preserved for either deposit or redeem step,

$$\lim_{\varepsilon \rightarrow 0} XL_{\Gamma, \phi}^{\text{swap}}(\varepsilon, \tau_0, \tau_1) = \lim_{\varepsilon \rightarrow 0} XL_{\Gamma', \phi}^{\text{swap}}(\varepsilon, \tau_0, \tau_1) \quad (32)$$

And similarly for the right swap rate limit. Thus the arbitrage solution type will remain the same.  $\square$

**Proof of Theorem 4** By inspection of swap rate definition (17) and funds-consistency (17).

For  $\Gamma \xrightarrow{\mathbb{T}} \Gamma'$  where  $\text{type}(\mathbb{T}) \in \{\text{dep}, \text{rdm}\}$  and  $(r_0 : \tau_0, r_1 : \tau_1) \in \Gamma$ , we demonstrate that  $(c \cdot r_0 : \tau_0, c \cdot r_1 : \tau_1) \in \Gamma'$ , where  $c > 0$ .

- [DEP]. The balance of  $\tau_0, \tau_1$  are increased by  $c > 0$  and their ratio preserved.
- [RDM]. The balance of  $\tau_0, \tau_1$  after redeeming  $v$  units of  $(\tau_0, \tau_1)$  in  $(r_0 : \tau_0, r_1 : \tau_1) \mid \dots$  are

$$\begin{aligned} & r_0 - \frac{v \cdot \tau_0}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)}, r_1 - \frac{v \cdot \tau_1}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)} \\ &= r_0 \cdot \left(1 - \frac{v}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)}\right), r_1 \cdot \left(1 - \frac{v}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)}\right) \end{aligned} \quad (33)$$

and thus the holdings are reduced by a factor  $1 > c > 0$  and their ratio is preserved.

Next, consider a left swap performed in  $\Gamma = (r_0 : \tau_0, r_1 : \tau_1) \mid \dots$

$$XL_{\Gamma, \phi}^{\text{swap}}(v, \tau_0, \tau_1) = \frac{r_1' - r_1}{v} \quad \text{where } \exists! r_1' : I(r_0, r_1) = I(r_0 + \phi \cdot v, r_1') \quad (34)$$

Then, from funds-consistency (17), it follows that the swap invariant equality above is preserved if we multiply lhs and rhs parameters of  $I$  with  $c > 0$

$$I(c \cdot r_0, c \cdot r_1) = I(c \cdot (r_0 + \phi \cdot v), c \cdot r_1') \quad (35)$$

From this and the swap rate definition (12) we can infer the left swap rate for input amount  $c \cdot v$  performed in  $\Gamma' = (c \cdot r_0 : \tau_0, c \cdot r_1 : \tau_1) \mid \dots$

$$XL_{\Gamma', \phi}^{\text{swap}}(c \cdot v, \tau_0, \tau_1) = \frac{c \cdot r_1' - c \cdot r_1}{c \cdot v} \quad (36)$$

Thus,  $XL_{\Gamma,\phi}^{\text{swap}}(v, \tau_0, \tau_1) = XL_{\Gamma',\phi}^{\text{swap}}(c \cdot v, \tau_0, \tau_1)$ . Here, we consider a change of the swap amount in the rhs to  $v$ , so both input amounts are equal. Due to demand-sensitivity (15), for a  $\mathbb{T} = \text{dep}$  ( $c > 1$ ) in  $\Gamma \xrightarrow{\mathbb{T}} \Gamma'$  this implies

$$XL_{\Gamma,\phi}^{\text{swap}}(v, \tau_0, \tau_1) < XL_{\Gamma',\phi}^{\text{swap}}(v, \tau_0, \tau_1)$$

as the input swap amount is reduced from  $c \cdot v$  to  $v$  and conversely for  $\mathbb{T} = \text{rdm}$  and ( $1 > c > 0$ ), where it is increased.

$$XL_{\Gamma,\phi}^{\text{swap}}(v, \tau_0, \tau_1) > XL_{\Gamma',\phi}^{\text{swap}}(v, \tau_0, \tau_1) \quad \square$$

**Proof of Theorem 5** Let  $I$  be incentive-consistent, and let:

$$\Gamma = (r_0 : \tau_0, r_1 : \tau_1) \mid \cdots \rightarrow^* (r'_0 : \tau_0, r'_1 : \tau_1) \mid \cdots = \Gamma' \quad \text{where } \frac{r_1}{r_0} = \frac{r'_1}{r'_0} \quad (37)$$

We have to prove that:

$$XL_{\Gamma}^{\text{rdm}}(\tau_0, \tau_1) \leq XL_{\Gamma'}^{\text{rdm}}(\tau_0, \tau_1) \quad XR_{\Gamma}^{\text{rdm}}(\tau_0, \tau_1) \leq XR_{\Gamma'}^{\text{rdm}}(\tau_0, \tau_1) \quad (38)$$

We begin with some auxiliary definitions. Firstly, we define *contours*, i.e. sets of pairs which fulfill a specific incentive-consistent swap invariant constraint:

$$c_k = \{(x, y) \mid I(x, y) = k\} \quad (39)$$

Next, for all  $R > 0$  and for all states  $\Gamma = (x : \tau_0, y : \tau_1) \mid \cdots$  we define the *projected* left redeem rate in  $\Gamma$  as follows (the right version is similar):

$$XL_{\Gamma,R}^{\text{rdm}}(\tau_0, \tau_1) = \frac{x'}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)} \quad (40)$$

s.t.  $I(x', R \cdot x') = I(r_0, r_1)$  for  $(r_0 : \tau_0, r_1 : \tau_1) \in \Gamma$

Note that for a state  $\Gamma_i = (r_{i,0} : \tau_0, r_{i,1} : \tau_1) \mid \cdots$ , the left redeem rate in  $\Gamma_i$  equals to the projected left redeem rate when choosing  $R$  as the funds ratio, i.e.:

$$XL_{\Gamma_i, R_{\Gamma_i}}^{\text{rdm}}(\tau_0, \tau_1) = XL_{\Gamma_i}^{\text{rdm}}(\tau_0, \tau_1) \quad \text{if } R_{\Gamma_i} = r_{i,1}/r_{i,0} \quad (41)$$

Since for  $\Gamma \rightarrow^* \Gamma'$  in (37), the funds ratio in  $\Gamma, \Gamma'$  are equal (i.e.,  $R_{\text{init}} = r_1/r_0 = r'_1/r'_0 = R_{\text{final}}$ ), the redeem and projected redeem rates in  $\Gamma$  and  $\Gamma'$  coincide. So, to prove (38) it is sufficient to show that  $XL_{\Gamma_i, R_{\text{init}}}^{\text{rdm}}(\tau_0, \tau_1)$  increases at each step  $\Gamma_i = (r_{i,0} : \tau_0, r_{i,1} : \tau_1) \mid \cdots \rightarrow \Gamma_{i+1} = (r_{i+1,0} : \tau_0, r_{i+1,1} : \tau_1) \mid \cdots$  along the trace  $\Gamma \rightarrow^* \Gamma'$ . We proceed to observe the evolution of the projected redeem rate for each possible action.

– [DEP0]. This rule is not used in  $\Gamma \rightarrow^* \Gamma'$  since  $(- : \tau_0, - : \tau_1) \in \Gamma$ .

– [RDM]/[DEP]. The evolution of the projected redeem rate is:

$$\begin{aligned} XL_{\Gamma_i, R_{\text{init}}}^{\text{rdm}}(\tau_0, \tau_1) &= \frac{x'_i}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma_i)} & \textcircled{1} \quad I(r_{i,0}, r_{i,1}) &= I(x'_i, R_{\text{init}} \cdot x'_i) \\ XL_{\Gamma_{i+1}, R_{\text{init}}}^{\text{rdm}}(\tau_0, \tau_1) &= \frac{c \cdot x'_i}{c \cdot \text{sply}_{(\tau_0, \tau_1)}(\Gamma_{i+1})} & \textcircled{2} \quad I(c \cdot r_{i,0}, c \cdot r_{i,1}) &= I(c \cdot x'_i, R_{\text{init}} \cdot c \cdot x'_i) \end{aligned} \quad (42)$$

From [DEP]/[RDM],  $r_{i+1,0}, r_{i+1,1} = c \cdot r_{i,0}, c \cdot r_{i,1}$  where  $c > 0$ , resulting in the lhs of  $\textcircled{2}$ . This can be inferred as follows: in [DEP] the holdings of  $\tau_0, \tau_1$  are increased and their ratio preserved. In [RDM], the holdings of  $\tau_0, \tau_1$  after redeeming  $v$  units of  $(\tau_0, \tau_1)$  in  $(r_0 : \tau_0, r_1 : \tau_1) \mid \dots$  are

$$\begin{aligned} & r_0 - \frac{v \cdot r_0}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)}, r_1 - \frac{v \cdot r_1}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)} \\ &= r_0 \cdot \left(1 - \frac{v}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)}\right), r_1 \cdot \left(1 - \frac{v}{\text{sply}_{(\tau_0, \tau_1)}(\Gamma)}\right) \end{aligned} \quad (43)$$

and thus their ratio is also preserved. The rhs of  $\textcircled{2}$  in (42) is inferred from the funds-consistency property (17). If  $R_{\text{init}}$  is defined in  $\textcircled{1}$ , then  $\textcircled{1} \implies \textcircled{2}$ . The projected redeem rate must therefore be maintained for a redeem or deposit, because both numerator and denominator are scaled by the same factor  $c$ .

– [SWAPL]/[SWAPR]. We interpret a swap action  $\mathbf{A} : \text{swapL}(v_0 : \tau_0, v_1 : \tau_1)$  in  $\Gamma_i$  as (1) a traversal from  $(r_{i,0}, r_{i,1})$  to  $(r'_{i,0}, r'_{i,1}) = (r_{i,0} + \phi \cdot v_0, r_{i,1} - v)$  such that

$$I(r_{i,0}, r_{i,1}) = I(r_{i,0} + \phi \cdot v_0, r_{i,1} - v) \quad (44)$$

and (2) a subsequent traversal to  $(r_{i+1,0}, r_{i+1,1}) = (r'_{i,0} + (1 - \phi) \cdot v_0, r'_{i,1})$  in  $\Gamma_{i+1} = (r_{i+1,0} : \tau_0, r_{i+1,1} : \tau_1) \mid \dots$ . Traversal (1) is within a contour  $c_k$ , where as traversal (2) occurs from  $c_k$  to  $c'_k$  where  $k \neq k'$ : we show this by contradiction. If  $c_k = c'_k$ , then for traversal (2), the following would hold

$$I(r'_{i,0}, r'_{i,1}) = I(r'_{i,0} + (1 - \phi) \cdot v_0, r'_{i,1}) \quad (45)$$

and imply  $\lim_{\varepsilon \rightarrow \infty} XL^{\text{swap}}(\varepsilon, r'_{i,0}, r'_{i,1}) = 0$ , violating SR-continuity (14), which requires swap rate limits to be positive.

We now examine the pair  $(r'_{i,0} - \Delta, r'_{i,1} + \Delta') \in c_k$ , which has the same ratio as  $R_{\Gamma_{i+1}} = r_{i+1,1}/r_{i+1,0}$  following the swap action, where  $(r_{i+1,0}, r_{i+1,1}) \in c_{k'}$ .

$$\begin{aligned} R_{\Gamma_{i+1}} &= \textcircled{1} \frac{r_{i+1,1}}{r_{i+1,0}} = \textcircled{2} \frac{r'_{i,1} - \Delta'}{r'_{i,0} + \Delta} = \textcircled{3} \frac{r_{i+1,1} - \Delta'}{r'_{i,0} + \Delta} \\ &\text{s.t. } I(r'_{i,0} - \Delta) = I(r'_{i,1} + \Delta') \end{aligned} \quad (46)$$

For  $\textcircled{2}$  and  $\textcircled{3}$ , we can equate  $r'_{i,1}$  and  $r_{i+1,1}$  because the funds of  $\tau_1$  are maintained in traversal (b). Comparing numerators and denominators in  $\textcircled{1}$  and  $\textcircled{3}$ , we can infer

$$r_{i+1,1} - \Delta' < r_{i+1,1} \quad r'_{i,0} + \Delta < r_{i+1,0} \quad (47)$$

Thus,  $c > 1$  in  $(r_{i+1,0}, r_{i+1,1}) = c \cdot (r'_{i,0} + \Delta, r'_{i,1} - \Delta')$ , where  $(r_{i+1,0}, r_{i+1,1}) \in c'_k$  and  $(r'_{i,0} + \Delta, r'_{i,1} - \Delta') \in c_k$ . From funds-consistency (17), we can infer that for two pairs  $x, y \in c_k$  and  $x', y' \in c'_k$  where  $y'/x' = y/x$ , then  $x', y' > x, y$ .

Finally, we can compare projected redeem rates in  $\Gamma_i$  and  $\Gamma_{i+1}$  when performing a left swap step.

$$\begin{aligned} XL_{\Gamma_i, R_{\text{init}}}^{\text{rdm}}(\tau_0, \tau_1) &= \frac{x'_i}{\text{sply}_{\tau_0, \tau_1}(\Gamma_i)} \quad \textcircled{1} I(r_{i,0}, r_{i,1}) = I(x'_i, R_{\text{init}} \cdot x'_i) \\ XL_{\Gamma_{i+1}, R_{\text{init}}}^{\text{rdm}}(\tau_0, \tau_1) &= \frac{x'_{i+1}}{\text{sply}_{\tau_0, \tau_1}(\Gamma_{i+1})} \quad \textcircled{2} I(r_{i+1,0}, r_{i+1,1}) = I(x'_{i+1}, R_{\text{init}} \cdot x'_{i+1}) \end{aligned} \quad (48)$$

Since swap actions do not affect the supply of minted tokens, we only need to relate  $x'_i$  to  $x'_{i+1}$  to compare projected redeem rates in  $\Gamma_i$  and  $\Gamma_{i+1}$ . In  $\textcircled{1}$   $(x'_i, R_{\text{init}} \cdot x'_i) \in c_k$  and in  $\textcircled{2}$   $(x'_{i+1}, R_{\text{init}} \cdot x'_{i+1}) \in c'_k$ , so the former must be strictly smaller than the latter. Thus

$$XL_{\Gamma_i, R_{\text{init}}}^{\text{rdm}}(\tau_0, \tau_1) < XL_{\Gamma_{i+1}, R_{\text{init}}}^{\text{rdm}}(\tau_0, \tau_1) \quad (49)$$

Therefore, since the projected left redeem rate increases with each step and coincides with the left redeem rate in  $\Gamma$  and  $\Gamma'$  of Theorem 5

$$XL_{\Gamma}^{\text{rdm}}(\tau_0, \tau_1) \leq XL_{\Gamma'}^{\text{rdm}}(\tau_0, \tau_1) \quad XR_{\Gamma}^{\text{rdm}}(\tau_0, \tau_1) \leq XR_{\Gamma'}^{\text{rdm}}(\tau_0, \tau_1)$$

The proof for the right redeem rate follows similarly and is omitted.  $\square$

## B Constant product swap rate limit

For a left swap  $A : \text{swapL}(v_0 : \tau_0, v_1 : \tau_1)$  in  $\Gamma = (r_0 : \tau_0, r_1 : \tau_1) \mid \dots$ , the following must hold for the constant product funds invariant

$$(r_0 + \phi \cdot v_0)(r_1 - v) = r_0 \cdot r_1 \quad (50)$$

Thus for a given  $v_0$ , the received amount  $v : \tau_1$  must be

$$v = \frac{\phi \cdot r_1 \cdot v_0}{r_0 + \phi \cdot v_0} \quad (51)$$

By definition, the right swap rate is simply  $v/v_0$ , or

$$XL_{\Gamma}^{\text{swap}}(v_0, \tau_0, \tau_1) = \frac{\phi \cdot r_1}{r_0 + \phi \cdot v_0} \quad (52)$$

The right swap rate limit is thus

$$\lim_{\varepsilon \rightarrow 0} XL_{\Gamma}^{\text{swap}}(\varepsilon, \tau_0, \tau_1) = \frac{\phi \cdot r_1}{r_0} \quad (53)$$



$$\begin{array}{c}
\frac{\sigma_A \tau \geq v}{\mathbf{A}[\sigma_A] \mid \mathbf{B}[\sigma_B] \mid \Gamma \xrightarrow{\mathbf{A}:xfer(\mathbf{B},v:\tau)} \mathbf{A}[\sigma_A - v : \tau] \mid \mathbf{B}[\sigma_B + v : \tau] \mid \Gamma} \text{ [XFER]} \\
\\
\frac{\sigma \tau_i \geq v_i > 0 \ (i \in \{0,1\}) \quad \tau_0 \neq \tau_1 \quad (- : \tau_0, - : \tau_1), (- : \tau_1, - : \tau_0) \notin \Gamma}{\mathbf{A}[\sigma] \mid \Gamma \xrightarrow{\mathbf{A}:dep(v_0:\tau_0,v_1:\tau_1)} \mathbf{A}[\sigma - v_0 : \tau_0 - v_1 : \tau_1 + v_0 : (\tau_0, \tau_1)] \mid (v_0 : \tau_0, v_1 : \tau_1) \mid \Gamma} \text{ [DEP0]} \\
\\
\frac{\sigma \tau_i \geq v_i > 0 \ (i \in \{0,1\}) \quad r_1 v_0 = r_0 v_1 \quad v = \frac{v_0}{r_0} \cdot sply_{(\tau_0, \tau_1)}(\Gamma)}{\Gamma = \mathbf{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma' \xrightarrow{\mathbf{A}:dep(v_0:\tau_0,v_1:\tau_1)} \mathbf{A}[\sigma - v_0 : \tau_0 - v_1 : \tau_1 + v : (\tau_0, \tau_1)] \mid (r_0 + v_0 : \tau_0, r_1 + v_1 : \tau_1) \mid \Gamma'} \text{ [DEP]} \\
\\
\frac{\sigma \tau_0 \geq v_0 > 0 \quad I(r_0 + \phi v_0, r_1 - v) = I(r_0, r_1) \quad 0 < v_1 \leq v \leq r_1}{\mathbf{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma \xrightarrow{\mathbf{A}:swapL(v_0:\tau_0,v_1:\tau_1)} \mathbf{A}[\sigma - v_0 : \tau_0 + v : \tau_1] \mid (r_0 + v_0 : \tau_0, r_1 - v : \tau_1) \mid \Gamma} \text{ [SWAPL]} \\
\\
\frac{\sigma \tau_1 \geq v_1 > 0 \quad I(r_0 - v, r_1 + \phi v_1) = I(r_0, r_1) \quad 0 < v_0 \leq v \leq r_0}{\mathbf{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma \xrightarrow{\mathbf{A}:swapR(v_0:\tau_0,v_1:\tau_1)} \mathbf{A}[\sigma + v : \tau_0 - v_1 : \tau_1] \mid (r_0 - v : \tau_0, r_1 + v_1 : \tau_1) \mid \Gamma} \text{ [SWAPR]} \\
\\
\frac{\sigma(\tau_0, \tau_1) \geq v > 0 \quad v_0 = v \frac{r_0}{sply_{(\tau_0, \tau_1)}(\Gamma)} \quad v_1 = v \frac{r_1}{sply_{(\tau_0, \tau_1)}(\Gamma)}}{\Gamma = \mathbf{A}[\sigma] \mid (r_0 : \tau_0, r_1 : \tau_1) \mid \Gamma' \xrightarrow{\mathbf{A}:rdm(v:(\tau_0, \tau_1))} \mathbf{A}[\sigma + v_0 : \tau_0 + v_1 : \tau_1 - v : (\tau_0, \tau_1)] \mid (r_0 - v_0 : \tau_0, r_1 - v_1 : \tau_1) \mid \Gamma'} \text{ [RDM]}
\end{array}$$

Fig. 4: Operational semantics of AMMs.