Quantifying secure composability of smart contracts

A Thesis

submitted to
Indian Institute of Science Education and Research, Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Emily Priyadarshini

Indian Institute of Science Education and Research, Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

May, 2025

Supervisor: Prof. Massimo Bartoletti

Certificate

This is to certify that this dissertation entitled “Quantifying secure composability of smart contracts” towards the partial fulfillment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Emily Priyadarshini at The University of Cagliari under the supervision of Prof. Massimo Bartoletti, Associate Professor, Department of Mathematics and Computer Science, during the academic year 2024-2025.

Prof. Massimo Bartoletti

Committee:

Supervisor:	Prof. Massimo Bartoletti, UniCa
Primary Expert:	Prof. Vivek Mohan Mallick, IISER-P
Additional Expert:	Prof. Madhukar Kumar, IIT-D

This thesis is dedicated to my parents

Declaration

I hereby declare that the matter embodied in the report entitled "Quantifying secure composability of smart contracts" are the results of the work carried out by me at the Department of Mathematics and Computer Science, University of Cagliari under the supervision of Prof. Massimo Bartoletti, and the same has not been submitted elsewhere for any other degree.

Emily

Emily Priyadarshini

Acknowledgements

To begin with, I would like to thank my supervisor, Dr Bartoletti, for his exceptional mentorship. I have learned a lot from his approach to research—to build understanding by analyzing simple examples, make important design choices and, most importantly, to construct counter examples against any prospective conjecture (which are often trivial). I am ever thankful to him for having numerous open dialogues, his patience and continuous effort to bridge the communication gap when I was unable to convey my ideas effectively. My confidence in conducting academic work has grown immensely owing to his proactive suggestions on attending courses, workshops and writing a paper—most of which I would have never dreamed of undertaking in the past ten months. I have truly relished my work over the past year and I believe it constitutes a significant, formative phase of my life.

I would like to thank my parents for their constant support—emotional, financial and otherwise—and confidence in sending me to a distant place despite the ambiguity surrounding my work and its outcome. The virtues of hard work and perseverance, which they instilled in me, were instrumental in navigating my time in Cagliari. I extend my gratitude to my sister and my uncle for their sustained support during my thesis and university years. Their advice has significantly strengthened my emotional resilience and expanded my perspective.

Lastly, I would like to thank the people who have become very important in my life—Anish, for always cheering me on and for being understanding; Shashwati, for her life-saving advice and Ritesh, for being a reliable friend. I would also like to thank Bhagyalekshmy for our enjoyable discussions on math. Furthermore, I would like to acknowledge the wonderful people I've met in Cagliari—Angelica, for being the most supportive flatmate; Professoressa Monica, for being the best Italian teacher and Marta, for her kind conversations.

Abstract

"We relive the history of the design of the motor car. Gadgets and glitter prevail over fundamental concerns of safety and economy."—C.A.R. Hoare, The 1980 ACM Turing Award Lecture.

Since its inception in 2008 [22], distributed ledger technology (DLT), has enabled a suite of financial services to be offered without relying on trusted intermediaries. In contrast to traditional finance, decentralized finance (DeFi) built upon DLT empowers users to execute peer-to-peer electronic transactions in a trustless enivironment. Account-based blockchain models, such as Ethereum, implement DeFi using smart contracts—self-executing digital agreements—programmed to encode and execute intended financial mechanisms. These smart contracts function as "bricks of lego", enabling developers to construct complex DeFi services by composing individual components [25, 1]. These compositions introduce new, complex inter-dependencies and vulnerabilities which, owing to the transparent and public nature of the ledger, makes them susceptible to exploitation by malicious participants. The subject of economic exploitation is a recurring issue and has been studied in the literature as Maximal Extractable Value (MEV) [9, 4, 21, 24]. Surprisingly enough, existing studies on compositionality are few. The notion of secure composability in [4] suffers from usability and algorithmic issues, while failing to specify the contracts from which MEV is extracted and incorrectly classifying as not composable contracts that have intended MEV. While the notion of "MEV non-interference" introduced in [7] addresses these drawbacks and checks whether the contracts that will be deployed suffer a loss when adversaries manipulate their dependencies, it is limited. Since the notion is *qualitative*, it does not provide information about the *degree* of interference caused and possible upper bounds to the loss suffered by a compound contract. This thesis performs an exploratory study of *quantitative* notions for secure composability of smart contracts, eventually arriving at *MEV interference*. Our *MEV interference*, which we denote by $\mathcal{I}(S \rightsquigarrow \Delta)$, captures various security properties one would deem desirable. We study the theoretical properties of this notion and apply it to study paradigmatic contract compositions of Lending Pools, Automated Market Makers, and Betting contracts.

Contents

Chapter 1

Introduction

1.1 Motivation and Objectives

Developing decentralized applications nowadays involves suitably designing, assembling and customizing a multitude of smart contracts, resulting in complex interactions and dependencies. In particular, recent DeFi applications are highly interconnected compositions of smart contracts of various kinds, including tokens, derivatives, decentralized exchanges (DEX), and lending protocols [15, 16].

This complexity poses significant security risks, as adversaries targeting one of the components may compromise the security of the overall application. Note that, for this to happen, the attacked component does not even need to have a proper vulnerability to exploit. For example, in an application composed of a lending protocol and a DEX serving as a price oracle, adversaries could target the DEX in order to artificially inflate the price of an asset that they have previously deposited to the lending pool. This manipulation would allow adversaries to borrow other assets with an insufficient collateral, circumventing the intended economic mechanism of the lending protocol [13, 23, 6, 20, 2].

The first step to address these risks is to formally define when a system of smart contracts is secure. In recent years, a few security notions have emerged, starting from Babel, Daian, Kelkar and Juels' "Clockwork finance" paper [4]. Broadly, these definitions try to characterise the economic security of smart contract systems based on the extent of economic damage that adversaries can inflict on them. In this context, adversaries are typically assumed to have the powers of consensus nodes, namely they can reorder, drop or insert transactions in blocks. Accordingly, the economic damage on a system S can be quantified in terms of the Maximal Extractable Value (MEV) that adversaries can extract from S by leveraging these powers [11]. To provide a more concrete formulation of the existing notions, consider a set of contracts Δ to be deployed in a system S. We denote by $S \mid \Delta$ the system composed of S and Δ. The security criterion in [4] requires that $\mathrm{MEV}(S \mid \Delta) \leq (1+\varepsilon)\,\mathrm{MEV}(S)$: namely, the MEV extractable from $S \mid \Delta$ does not exceed the MEV extractable from S by more than a factor of ε. This notion does not capture our intuition of assessing the security of Δ in terms of the economic losses that Δ could incur due to adversaries interacting with the context S. For example, an airdrop contract Δ that gives away tokens would be deemed insecure, since interactions with S are immaterial.

In a different security setting, a similar intuition was the basis of Goguen and Meseguer' non-interference [12], which was originally formulated as follows:

> “One group of users, using a certain set of commands, is noninterfering with another group of users if what the first group does with those commands has no effect on what the second group of users can see”.

In the setting of smart contract compositions, this notion can be reinterpreted by requiring that adversaries interacting with S do not inflict economic damage to Δ. The notion of *MEV non-interference* introduced by [7] is based on this idea, using MEV as a measure of economic damage. The approaches in [14, 28] are also based on the idea of non-interference, but replacing MEV with an explicit tagging of contract variables into high-level or low-level variables.

A common aspect of these approaches to economic non-interference is their *qualitative* nature: namely, these definitions classify a composition as either secure or insecure, in a binary fashion. While such a qualitative evaluation is sufficient when a composition is deemed secure, in case it is not it fails to give any meaningful estimate of the *degree* of interference. For example, when in the above-mentioned (insecure) composition between a lending protocol and a DEX, a quantitative measure could provide insights into the extent to which the system state (e.g., the liquidity reserves in the DEX) and the contract parameters (e.g., the collateralization threshold) contribute to increasing the economic loss.

1.2 Contributions

This thesis studies various quantitative notions of economic security for smart contract compositions, ultimately arriving at one that best serves our needs. Our *MEV interference*, which we denote by $\mathcal{I}(S \rightsquigarrow \Delta)$, measures the increase of economic loss of contracts Δ that adversaries can achieve by manipulating the context S. We apply our notion to assess the security of some notable contract compositions, including a bet on a token price, and a lending protocol relying on a DEX as a price oracle. We prove some fundamental properties of our notion: more specifically, $\mathcal{I}(S \rightsquigarrow \Delta)$ increases when S is extended with contracts that are not in the dependencies of Δ (Theorem 5); $\mathcal{I}(S \rightsquigarrow \Delta)$ does not depend on the token balances of users except adversaries (Theorem 6); $\mathcal{I}(S \rightsquigarrow \Delta)$ is preserved when extending S with contracts Γ that enjoy some specific independency conditions with respect to Δ (Theorem 7).

1.3 Outline

This thesis is divided into 2 sections: Part I provides the background to this thesis— Chapter 1 describes the motivation to our research problem, objectives and the main contributions of this thesis; Chapter 2 introduces the blockchain model and definitions; Chapter 3 states a few basic results on MEV; Chapter 4 proposes an initial definition of quantitative MEV interference; Chapter 5 proposes an alternate definition of MEV interference which is independent of adversarial wealth. Part II provides a detailed account of the major contributions of this thesis— Chapter 6 provides the definition of *MEV Interference* and its theoretical properties; Chapter 7 illustrates our definition on a few paradigmatic smart contract compositions; Chapter 8 summarises the main contributions of this thesis, discusses its limitations and provides a sketch of avenues open for future work.

Part I

Background

Chapter 2

Blockchain model

This section presents a brief overview of our working blockchain model (refer to Section 2 in [7] for a more elaborate description and the underlying motivations).

2.1 Blockchain states

Our blockchain model defines a system comprising a set $\mathbb{T}$ of *token types* $(\mathtt{T}, \mathtt{T}', \ldots)$ and a countably infinite set $\mathbb{A}$ of *accounts*, which are further subdivided into *user accounts* $\mathtt{A}, \mathtt{B}, \ldots \in \mathbb{A}_u$ and *contract accounts* $\mathtt{C}, \mathtt{D}, \ldots \in \mathbb{A}_c$. We define a subset $\mathcal{M}$ of the user accounts to represent the adversarial entities within the system. The state of a user account, i.e. a ***wallet***, is denoted by $w \in \mathbb{T} \to \mathbb{N}$, which maps tokens to non-negative integers, representing the token balances in the account. The state of a contract account is a pair (w, σ), where w is a wallet and σ is a key-value store. ***Blockchain states*** $S, S', \ldots$ are finite maps from accounts to their respective states, where the user accounts include at least the adversary's wallets. Furthermore, we use the operator $\mid$ to deconstruct a blockchain state into its components.

2.2 Contracts

Contracts are defined as an associated set of methods, each capable of executing the following range of operations: (i) update the contract wallet and state, (ii) receive parameters and tokens from a caller, (iii) call other contracts (possibly transferring tokens along with the call), (iv) transfer tokens to user accounts, (v) return values and transfer tokens to a caller, (vi) abort. As usual, a method cannot drain tokens from other accounts: the only ways for a contract to receive tokens are (i) from a caller invoking one of its methods, or (ii) by calling a method of another contract that sends tokens to its caller. We assume that a contract $\mathtt{C}$ can only call methods of contracts deployed before it. Formally, defining "$\mathtt{C}$ *is called by* $\mathtt{D}$" when some method of $\mathtt{D}$ calls some method of $\mathtt{C}$, we are requiring that the transitive and reflexive closure $\sqsubseteq$ of this relation is a partial order. We also assume that blockchain states contain all the *dependencies* of their contracts: formally, if $\mathcal{C}$ are the contracts in S, we require that $\mathit{deps}(\mathcal{C}) = \{\mathtt{C}' \mid \exists \mathtt{C} \in \mathcal{C}.\ \mathtt{C}' \sqsubseteq \mathtt{C}\}$ are in S. States satisfying these assumptions are said *well-formed*: all states mentioned in our results (either in hypothesis or thesis) are always well-formed. We write $S = W \mid \Gamma$ for a blockchain state S composed of user wallets W and contract states Γ. We can deconstruct wallets, writing $S = W \mid W' \mid \Gamma$ when $\operatorname{dom} W$ and $\operatorname{dom} W'$ are disjoint, as well as contract states, writing $S = W \mid \Gamma \mid \Delta$. We denote by $\dagger\Gamma$ the set of contract accounts in Γ (i.e. $\dagger\Gamma = \operatorname{dom}\Gamma$), and let $\mathit{deps}(\Delta) = \mathit{deps}(\dagger\Delta)$. Finally, we assume that contracts cannot inspect the state of other accounts, including users' wallets and the state of other contracts. Formally, we are requiring that each transaction enabled in S produces the same effect in a "richer" state $S' \geq_{\$} S$ containing more tokens in users' wallets (Definition 5).

2.3 Transactions

We model contracts behaviour as a deterministic transition relation $\rightarrow$ between blockchain states, where state transitions are triggered by ***transactions*** $\mathsf{X}, \mathsf{X}', \ldots$. A transaction is a call to a contract method, written $\mathtt{A}\!:\!\mathtt{C.f}(\mathtt{args})$, where $\mathtt{A}$ is the user signing the transaction, $\mathtt{C}$ is the called contract, $\mathtt{f}$ is the called method, and $\mathtt{args}$ is the list of actual parameters, which can also include transfers of tokens from $\mathtt{A}$ to $\mathtt{C}$. Invalid transactions are rolled-back, i.e. $\rightarrow$ preserves the state. Given $\mathsf{X} = \mathtt{A}\!:\!\mathtt{C.f}(\mathtt{args})$, we write $\mathit{callee}(\mathsf{X})$ for the target contract $\mathtt{C}$. Methods can refer to $\mathtt{A}$ via the identifier `origin` and to the caller (contract or user) account via `sender`.

2.4 TxScript

We specify the contracts in our examples in a concrete contract language: TxScript (refer to [8] for the syntax of TxScript contracts and transactions): (i) the expression

$\#\mathtt{T}$ denotes the number of tokens $\mathtt{T}$ stored in the contract; (ii) the formal parameter $?\,x{:}\,\mathtt{T}$ requires the `sender` to transfer some tokens $\mathtt{T}$ to the contract along with the call (the unsigned integer variable x generalises Solidity's `msg.value` to multi-tokens); (iii) the command $\mathtt{a}!\,e{:}\,\mathtt{T}$ transfers e units of $\mathtt{T}$ from the contract to account $\mathtt{a}$, where e is an expression, and $\mathtt{a}$ could be either a user account or the method `sender`).

2.5 Wealth and gain

We denote by $\$_{\mathcal{A}}(S)$ the wealth of accounts $\mathcal{A}$ in S and by $\$\mathbf{1}_{\mathtt{T}}$ the price of token type $\mathtt{T}$.

Definition 1 (Wealth)**.** The wealth of $\mathcal{A} \subseteq \mathbb{A}$ in $S = W \mid \Gamma$ is given by:

$$\$_{\mathcal{A}}(S) \;=\; \sum_{\mathtt{A} \in \mathcal{A} \cap \operatorname{dom} W,\, \mathtt{T}} W(\mathtt{A})(\mathtt{T}) \cdot \$\mathbf{1}_{\mathtt{T}} \;+\; \sum_{\mathtt{C} \in \mathcal{A} \cap \operatorname{dom} \Gamma,\, \mathtt{T}} \mathit{fst}(\Gamma(\mathtt{C}))(\mathtt{T}) \cdot \$\mathbf{1}_{\mathtt{T}} \qquad (2.1)$$

To rule out ill-formed states with an infinite amount of tokens, we require blockchain states to enjoy the *finite tokens axiom*, i.e. $\sum_{\mathtt{A},\mathtt{T}} S(\mathtt{A})(\mathtt{T}) \in \mathbb{N}$. This makes the wealth always finite.

Definition 2 (Gain)**.** The gain of $\mathcal{A} \subseteq \mathbb{A}$ upon firing a transactions sequence $\vec{\mathfrak{X}}$ in S is given by $\gamma_{\mathcal{A}}(S, \vec{\mathfrak{X}}) = \$_{\mathcal{A}}(S') - \$_{\mathcal{A}}(S)$ if $S \xrightarrow{\vec{\mathfrak{X}}} S'$.

Chapter 3

Local MEV

We enlist below the results taken from Section 3 in [7], which are heavily used throughout this thesis, for the reader's convenience.

3.1 Results on Local MEV

We denote by $\kappa(\mathcal{M}, P)$ the set of transactions craftable by $\mathcal{M}$ using a mempool P, and by $\kappa(\mathcal{M}, P)^*$ their finite sequences. We assume that the mempool is empty, just writing $\kappa(\mathcal{M})$. This is because in defining secure composability we are concerned about the MEV extractable by exploiting new contracts, and not that extractable from the mempool.

Definition 3 (Local MEV)**.** Let $\kappa_{\mathcal{D}}(\mathcal{M}) = \{\mathsf{X} \in \kappa(\mathcal{M}) \mid \mathit{callee}(\mathsf{X}) \in \mathcal{D}\}$ be the set of transactions craftable by $\mathcal{M}$ and targeting contracts in $\mathcal{D}$. We define:

$$\mathrm{MEV}_{\mathcal{D}}(S, \mathcal{C}) = \max \left\{ -\gamma_{\mathcal{C}}(S, \vec{\mathfrak{X}}) \;\middle|\; \vec{\mathfrak{X}} \in \kappa_{\mathcal{D}}(\mathcal{M})^* \right\} \tag{3.1}$$

Hereafter, we abbreviate $\mathrm{MEV}_{\mathbb{A}_c}(S, \mathcal{C})$ as $\mathrm{MEV}(S, \mathcal{C})$.

In Item 3, we write $\Gamma \preceq \Delta$ whenever Δ is a widening of Γ. More precisely:

$$\Gamma \preceq \Delta \qquad \Longleftrightarrow \qquad \forall \mathtt{C} \in \mathcal{C}.\ \mathtt{C} \in \mathrm{dom}\,\Delta \;\wedge\; \Gamma(\mathtt{C}) = \Delta(\mathtt{C})$$

Therefore, the condition $\Gamma \preceq \Delta$ in Item 3 means that Δ is a widening of the state Γ with other arbitrary contract states.

Lemma 1 (Basic properties of MEV)**.** *For all* S*,* $\mathcal{C}, \mathcal{D} \subseteq \mathbb{A}_c$*:*

1. $\mathrm{MEV}_{\mathcal{D}}(S, \emptyset) = \mathrm{MEV}_{\emptyset}(S, \mathcal{C}) = 0$, $\mathrm{MEV}_{\mathbb{A}_c}(S, \mathbb{A}_c) \geq \mathrm{MEV}(S)$
2. *if* $\mathcal{D} \subseteq \mathcal{D}'$*, then* $\mathrm{MEV}_{\mathcal{D}}(S, \mathcal{C}) \leq \mathrm{MEV}_{\mathcal{D}'}(S, \mathcal{C})$
3. $\mathrm{MEV}_{\mathcal{D}}(W \mid \Gamma, \mathcal{C}) \leq \mathrm{MEV}_{\mathcal{D}}(W \mid \Delta, \mathcal{C})$ *if* $\Gamma \preceq \Delta$
4. $\mathrm{MEV}_{\mathcal{D}}(W \mid \Gamma, \mathcal{C}) = \mathrm{MEV}_{\mathcal{D}}(W \mid \Gamma, \mathcal{C} \cap \dagger\Gamma) = \mathrm{MEV}_{\mathcal{D} \cap \dagger\Gamma}(W \mid \Gamma, \mathcal{C})$
5. $0 \leq \mathrm{MEV}_{\mathcal{D}}(S, \mathcal{C}) \leq \$_{\mathcal{C}}(S)$

Definition 4 (Richer state)**.** We write $S \leq_{\$} S'$ when the state S' can be obtained from S by making the wallets larger, i.e. when $S = W \mid \Gamma$ and $S' = (W + W_\delta) \mid \Gamma$, for some W, W_δ, and Γ.

Definition 5 (Wallet-monotonicity)**.** A blockchain state $S = W \mid \Gamma$ is wallet-monotonic if, whenever $S \xrightarrow{\mathsf{X}} W' \mid \Gamma'$ for a valid transaction X, then $W + W_\delta \mid \Gamma \xrightarrow{\mathsf{X}} W' + W_\delta \mid \Gamma'$, for all W_δ.

Lemma 2 (MEV and adversaries' wallets)**.** *For all* S*,* S'*,* Δ*,* W*,* $W_{\mathcal{M}}$*:*

1. *if* $\mathrm{dom}\, W_{\mathcal{M}} = \mathcal{M}$*, then* $\mathrm{MEV}_{\mathcal{D}}(W_{\mathcal{M}} \mid W \mid \Gamma, \mathcal{C}) = \mathrm{MEV}_{\mathcal{D}}(W_{\mathcal{M}} \mid \Gamma, \mathcal{C})$

2. *if* $S \leq_{\$} S'$*, then* $\mathrm{MEV}_{\mathcal{D}}(S, \mathcal{C}) \leq \mathrm{MEV}_{\mathcal{D}}(S', \mathcal{C})$

Lemma 3 (Stability)**.** *For all* $\mathcal{C}$*,* $\mathcal{D}$*,* Γ*, there exists an adversary wallet* $W_{\mathcal{M}}$ *such that* $\mathrm{MEV}_{\mathcal{D}}(W_{\mathcal{M}} \mid \Gamma, \mathcal{C}) = \mathrm{MEV}_{\mathcal{D}}(W'_{\mathcal{M}} \mid \Gamma, \mathcal{C})$ *for all* $W'_{\mathcal{M}} \geq_{\$} W_{\mathcal{M}}$*.*

3.2 Results on Local MEV of wealthy adversaries

Definition 6 (Local MEV of wealthy adversaries)**.** For all $\mathcal{C}, \mathcal{D}, \Gamma$, let:

$$\mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C}) = \max_{W} \mathrm{MEV}_{\mathcal{D}}(W \mid \Gamma, \mathcal{C}) \tag{3.2}$$

Lemma 4 (Basic properties of MEV^{∞})**.** *For all* Γ*,* $\mathcal{C}, \mathcal{D} \subseteq \mathbb{A}_c$*:*

1. $\mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \emptyset) = \mathrm{MEV}^{\infty}_{\emptyset}(\Gamma, \mathcal{C}) = 0$
2. *if* $\mathcal{D} \subseteq \mathcal{D}'$*, then* $\mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C}) \leq \mathrm{MEV}^{\infty}_{\mathcal{D}'}(\Gamma, \mathcal{C})$
3. $\mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C}) \leq \mathrm{MEV}^{\infty}_{\mathcal{D}}(\Delta, \mathcal{C})$ *if* $\Gamma \preceq \Delta$
4. $\mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C}) = \mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C} \cap \dagger\Gamma) = \mathrm{MEV}^{\infty}_{\mathcal{D} \cap \dagger\Gamma}(\Gamma, \mathcal{C})$
5. $0 \leq \mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C}) \leq \$_{\mathcal{C}}(\Gamma)$

3.3 Results on MEV non-interference

Definition 7 (MEV non-interference)**.** A state S is MEV *non-interfering* with Δ, in symbols $S \not\rightsquigarrow \Delta$, when $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)$.

Definition 8 (MEV non-interference against wealthy adversaries)**.** A contract state Γ is MEV^{∞} *non-interfering* with Δ, in symbols $\Gamma \not\rightsquigarrow^{\infty} \Delta$, when $\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta)$.

Theorem 1 (Sufficient conditions for $\not\rightsquigarrow$)**.** *Let $S = W \mid \Gamma$. Each of the following conditions implies $S \not\rightsquigarrow \Delta$: (1) $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = 0$ (2) Γ and Δ are token independent in $S \mid \Delta$ and contract independent (3) Γ and Δ are token independent in $S \mid \Delta$ and Δ is stable w.r.t. moves of $\mathcal{M}$ on Γ.*

3.4 Supplementary results

This section consists of additional results on MEV, MEV^{∞} and non-interference that have been subsequently employed in this thesis. We would like to remark that this thesis contains alternately defined notions of *stripping* in Theorem 2 and *token independence* in Definition 9.

Lemma 5 states that widening the contract state Γ with new contracts Γ' preserves the MEV extractable from the target contracts. This is because the contracts allowed to be targeted by the adversary, i.e. $\mathcal{D}$, are not widened. It refines Item 3 of Lemma 1, giving an equality under the additional assumption $\mathcal{D} \subseteq \dagger\Gamma$.

Lemma 5. $\mathrm{MEV}_{\mathcal{D}}(W \mid \Gamma, \mathcal{C}) = \mathrm{MEV}_{\mathcal{D}}(W \mid \Delta, \mathcal{C})$ *when* $\mathcal{D} \subseteq \dagger\Gamma$ *and* $\Gamma \preceq \Delta$*.*

Proof. The inequality $\leq$ follows directly from Item 3 of Lemma 1. For the inequality $\geq$, assume that Δ is the composition of the contracts Γ with some other contracts $\bar{\Gamma}$, i.e. $\Gamma \preceq \Delta$, $\bar{\Gamma} \preceq \Delta$, and $\Delta \preceq \Gamma \mid \bar{\Gamma}$. Let $\vec{\mathcal{X}} \in \kappa_{\mathcal{D}}(\mathcal{M})^{*}$ be a valid sequence of transactions that maximizes the loss $-\gamma_{\mathcal{C}}(W \mid \Delta, \vec{\mathcal{X}})$. Since $\vec{\mathcal{X}}$ consists of transactions targeting contracts in $\mathcal{D} \subseteq \dagger\Gamma$ and since, by the well-formedness assumption, there are no internal calls from Γ to $\bar{\Gamma}$, the contracts in $\bar{\Gamma}$ are not affected by $\vec{\mathcal{X}}$. Hence, executing $\vec{\mathcal{X}}$ yields a transition of the form:

$$W \mid \Delta \xrightarrow{\vec{\mathcal{X}}} W' \mid \Delta' \qquad\qquad \text{where } \bar{\Gamma} \preceq \Delta'$$

As noted above, $\vec{\mathcal{X}}$ does not include any direct/indirect calls to $\dagger\bar{\Gamma}$, and so $\vec{\mathcal{X}}$ is also valid in $W \mid \Gamma$. Therefore, we also have some Γ' such that:

$$W \mid \Gamma \xrightarrow{\vec{\mathcal{X}}} W' \mid \Gamma'$$

To prove that the loss is constant, observe that:

$$\begin{aligned}
\gamma_{\mathcal{C}}(W \mid \Gamma, \vec{\mathcal{X}}) &= \$_{\mathcal{C}}(W' \mid \Gamma') - \$_{\mathcal{C}}(W \mid \Gamma) \\
&= \$_{\mathcal{C}}(\Gamma') - \$_{\mathcal{C}}(\Gamma) \\
&= \$_{\mathcal{C}}(\Delta') - \$_{\mathcal{C}}(\bar{\Gamma}) - \$_{\mathcal{C}}(\Delta) + \$_{\mathcal{C}}(\bar{\Gamma}) \\
&= \$_{\mathcal{C}}(\Delta') - \$_{\mathcal{C}}(\Delta) \\
&= \$_{\mathcal{C}}(W' \mid \Delta') - \$_{\mathcal{C}}(W \mid \Delta) \\
&= \gamma_{\mathcal{C}}(W \mid \Delta, \vec{\mathcal{X}})
\end{aligned}$$

This implies that:

$$\mathrm{MEV}_{\mathcal{D}}(W \mid \Delta, \mathcal{C}) \leq \mathrm{MEV}_{\mathcal{D}}(W \mid \Gamma, \mathcal{C})$$

which gives our thesis. □

Lemma 6. $\mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C}) = \mathrm{MEV}^{\infty}_{\mathcal{D}}(\Delta, \mathcal{C})$ *when* $\mathcal{D} \subseteq \dagger\Gamma$ *and* $\Gamma \preceq \Delta$.

Proof. This has an analogous statement in Lemma 5, which holds for any wallet state. Due to the stability lemma (Lemma 3) and the definition of MEV^{∞} (Definition 6), the "rich-adversary" version of the statement must also hold. □

Before stating Theorem 2, we formalize the notions of contract independence, token independence and sender-agnosticism.

Contract and Token Independence Intuitively, we say that S and Δ have *contract dependencies* when some contract in Δ calls a contract in S. Formally, contract states Γ and Δ are **contract independent** when their dependencies are disjoint, i.e. $\mathit{deps}(\Gamma) \cap \mathit{deps}(\Delta) = \emptyset$. For example, we will consider Section 7.2 in the upcoming chapter that shows contract dependencies between a bet contract and an Automated Market Maker (AMM) used as a price oracle.

We now formalize token independence. Intuitively, we say that S and Δ have *token dependencies* when some contract in Δ outputs tokens that can be used as input to contract in S, or vice-versa). Note that this is an alternately defined notion of that defined in [7]. Formalizing token independence requires two auxiliary notions: the token types that can be received by Γ in S, denoted $in_S(\Gamma)$, and those that can be sent, denoted by $out_S(\Gamma)$.

Definition 9 (Token independence)**.** Let $S = W \mid \Gamma$. We define:

- $in_S(\Gamma)$ as the set of token types $\mathtt{T}$ for which there exists a state S' reachable from S through a sequence of steps, containing a transaction that causes an inflow of tokens $\mathtt{T}$ to some contract in Γ.

- $out_S(\Gamma)$ as the set of token types $\mathtt{T}$ for which there exists a state S' reachable from S through a sequence of steps, containing a transaction that causes an outflow of tokens $\mathtt{T}$ from some contract in Γ.

We say that contracts in Γ and Δ are **token independent** in $S = W \mid \Gamma \mid \Delta$ when $in_S(\Gamma) \cap out_S(\Delta) = \emptyset = in_S(\Delta) \cap out_S(\Gamma)$.

Definition 10 (Sender-agnostic)**.** A contract is sender-agnostic if:

- the effect of calling each of its functions can be decomposed as: (i) updating the contracts' states (either directly or through internal calls); (ii) transferring tokens from and to users and contracts; (iii) transferring tokens to its `sender`.

- any call with the same arguments and `origin`, but distinct `sender`, has the same effect, except for the item *(iii)* where tokens are transferred to the new sender.

Theorem 2 gives sufficient conditions under which we can strip $\mathcal{D}$ of all the non-dependencies of $\mathcal{C}$ while preserving $\text{MEV}_{\mathcal{D}}(S, \mathcal{C})$. Condition (i) is that contract methods are sender-agnostic, i.e. they are not aware of the identity of the `sender`, being only able to use it as a recipient of token transfers. Condition (ii) ensures that $\mathcal{D}$ consists enough contracts to reproduce attacks in the stripped state. Condition (iii) requires that the dependencies and the non-dependencies of $\mathcal{C}$ in $\mathcal{D}$ are token independent in S. In other words, there are no token dependencies between $\mathcal{D} \cap deps(\mathcal{C})$ and $\mathcal{D} \setminus deps(\mathcal{C})$, which could have potentially be exploited by non-wealthy adversaries.

Theorem 2. $\mathrm{MEV}_{\mathcal{D}}(S, \mathcal{C}) = \mathrm{MEV}_{\mathcal{D} \cap \mathit{deps}(\mathcal{C})}(S, \mathcal{C})$ *holds if the contracts* $\mathcal{C}' = \mathit{deps}(\mathcal{C}) \cap \mathit{deps}(\mathcal{D} \setminus \mathit{deps}(\mathcal{C}))$ *satisfy: (i)* $\mathcal{C}'$ *are sender-agnostic, (ii)* $\mathcal{C}' \subseteq \mathcal{D}$*, and (iii)* $\mathit{deps}(\mathcal{D}) \cap \mathit{deps}(\mathcal{C})$ *and* $\mathit{deps}(\mathcal{D}) \setminus \mathit{deps}(\mathcal{C})$ *are token independent in* S*.*

Proof. First, note that the inequality $\mathrm{MEV}_{\mathcal{D}{\restriction_{\mathcal{C}}}}(S, \mathcal{C}) \leq \mathrm{MEV}_{\mathcal{D}}(S, \mathcal{C})$ follows from Item 2 of Lemma 1, so we just need to show that:

$$\mathrm{MEV}_{\mathcal{D}}(S, \mathcal{C}) \leq \mathrm{MEV}_{\mathcal{D}{\restriction_{\mathcal{C}}}}(S, \mathcal{C})$$

To do so, let $\vec{\mathcal{X}} \in \kappa_{\mathcal{D}}(\mathcal{M})^*$ be a sequence of transactions that maximizes the loss of $\mathcal{C}$ when executed in state S. We show that there exists $\vec{\mathcal{Y}} \in \kappa_{\mathcal{D}{\restriction_{\mathcal{C}}}}(\mathcal{M})^*$ that causes a loss to $\mathcal{C}$ equal to the one caused by $\vec{\mathcal{X}}$, i.e.:

$$\vec{\mathcal{Y}} \in \kappa_{\mathcal{D}{\restriction_{\mathcal{C}}}}(\mathcal{M})^* \qquad \gamma_{\mathcal{C}}(S, \vec{\mathcal{Y}}) = \gamma_{\mathcal{C}}(S, \vec{\mathcal{X}}) \tag{3.3}$$

W.l.o.g. we assume that all the transactions in $\vec{\mathcal{X}}$ are valid: indeed, invalid transactions in $\vec{\mathcal{X}}$ are reverted, so they can be removed without affecting the loss.

Note that each transaction $\mathsf{X}_i = \mathtt{M}[i]\colon \mathtt{C}_{\mathtt{i,1}}.\mathtt{f}_{\mathtt{i,1}}(\mathtt{args}_{\mathtt{i,1}})$ in $\vec{\mathcal{X}}$ can trigger a sequence of *internal* contract-to-contract function calls:

$$\mathtt{C}_{\mathtt{i,1}}\colon \mathtt{C}_{\mathtt{i,2}}.\mathtt{f}_{\mathtt{i,2}}(\mathtt{args}_{\mathtt{i,2}}) \;\; \mathtt{C}_{\mathtt{i,2}}\colon \mathtt{C}_{\mathtt{i,3}}.\mathtt{f}_{\mathtt{i,3}}(\mathtt{args}_{\mathtt{i,3}}) \;\; \cdots \;\; \mathtt{C}_{\mathtt{i,k-1}}\colon \mathtt{C}_{\mathtt{i,k}}.\mathtt{f}_{\mathtt{i,k}}(\mathtt{args}_{\mathtt{i,k}})$$

Let $\vec{x}$ be the sequence of all function calls (either external or internal) that are performed upon the execution of $\vec{\mathcal{X}}$ in state S. To construct $\vec{\mathcal{Y}}$, we start by considering the subsequence $\vec{y}$ of $\vec{x}$ containing all and only the calls of the form:

(a) $\mathtt{M}[i]\colon \mathtt{C}_{\mathtt{i,1}}.\mathtt{f}_{\mathtt{i,1}}(\mathtt{args}_{\mathtt{i,1}})$ where $\mathtt{C}_{\mathtt{i,1}} \in \mathit{deps}(\mathcal{C})$, or

(b) $\mathtt{C}_{\mathtt{i,j-1}}\colon \mathtt{C}_{\mathtt{i,j}}.\mathtt{f}_{\mathtt{i,j}}(\mathtt{args}_{\mathtt{i,j}})$, where $\mathtt{C}_{\mathtt{i,j-1}} \not\in \mathit{deps}(\mathcal{C})$ and $\mathtt{C}_{\mathtt{i,j}} \in \mathit{deps}(\mathcal{C})$.

Claim (1). If $\mathtt{C}_{\mathtt{i,j-1}}\colon \mathtt{C}_{\mathtt{i,j}}.\mathtt{f}_{\mathtt{i,j}}(\mathtt{args}_{\mathtt{i,j}}) \in \vec{y}$, then $\mathtt{C}_{\mathtt{i,j}} \in \mathcal{C}'$.

Proof of Claim (1). By hypothesis, $\mathtt{C}_{\mathtt{i,j}} \in \mathit{deps}(\mathcal{C})$. Let $\mathsf{X}_i \in \vec{\mathcal{X}}$ be the transaction that originated the call. Since $\mathsf{X}_i \in \kappa_{\mathcal{D}}(\mathcal{M})$, then $\mathtt{C}_{\mathtt{i,1}} \in \mathcal{D}$. Since $\mathit{deps}(\mathcal{C})$ is closed downward and $\mathtt{C}_{\mathtt{i,j-1}} \not\in \mathit{deps}(\mathcal{C})$, then $\mathtt{C}_{\mathtt{i,1}} \not\in \mathit{deps}(\mathcal{C})$. So, $\mathtt{C}_{\mathtt{i,1}} \in \mathcal{D} \setminus \mathit{deps}(\mathcal{C})$, and therefore $\mathtt{C}_{\mathtt{i,j}} \in \mathit{deps}(\mathcal{D} \setminus \mathit{deps}(\mathcal{C}))$. This completes the proof of Claim (1).

To describe the construction of $\vec{\mathcal{Y}}$, let the meta-variables $\mathtt{a}_i$ range over user and contract addresses, so to rewrite the sequence $\vec{y}$ as follows:

$$\mathtt{a}_1\colon \mathtt{C}_1.\mathtt{f}_1(\mathtt{args}_1) \;\; \mathtt{a}_2\colon \mathtt{C}_2.\mathtt{f}_2(\mathtt{args}_2) \;\; \cdots \;\; \mathtt{a}_n\colon \mathtt{C}_\mathtt{n}.\mathtt{f}_\mathtt{n}(\mathtt{args}_\mathtt{n}) \cdots$$

We translate $\vec{y}$ into the sequence of *transactions* $\vec{\mathcal{Y}}$ by preserving the senders $\mathtt{a}_i$ that are user accounts (i.e., $\mathtt{a}_i = \mathtt{M}[i]$), and by replacing the $\mathtt{a}_i$ that are contract accounts into the user account that originated the corresponding call. Namely, if $\mathtt{a}_i = \mathtt{C}_{\mathtt{i},\mathtt{j}-\mathtt{1}}$ is a contract account corresponding to the following call in $\vec{y}$:

$$\mathtt{C}_{\mathtt{i},\mathtt{j}-\mathtt{1}}\colon \mathtt{C}_{\mathtt{i},\mathtt{j}}.\mathtt{f}_{\mathtt{i},\mathtt{j}}(\mathtt{args}_{\mathtt{i},\mathtt{j}})$$

then the sender of the i-th transaction in $\vec{\mathcal{Y}}$ is $\mathtt{M}[i]$, i.e. the originator of the call. Note that each transaction Y_i in $\vec{\mathcal{Y}}$ can be funded by the adversary:

- if $\mathtt{a}_i = \mathtt{M}[i]$, then the fact that the corresponding transaction X_i in $\vec{\mathcal{X}}$ was valid implies that $\mathtt{M}[i]$ has the tokens needed to fund the call;
- if $\mathtt{a}_i = \mathtt{C}_{\mathtt{i},\mathtt{j}-\mathtt{1}}$, then there is no token transfer from $\mathtt{C}_{\mathtt{i},\mathtt{j}-\mathtt{1}}$ to $\mathtt{C}_{\mathtt{i},\mathtt{j}}$, and so Y_i does not need to be funded. This is because:
  - $\mathtt{C}_{\mathtt{i},\mathtt{j}-\mathtt{1}} \in \mathit{deps}(\mathcal{D}) \setminus \mathit{deps}(\mathcal{C})$: indeed, $\mathtt{C}_{\mathtt{i},\mathtt{j}-\mathtt{1}} \in \mathit{deps}(\mathcal{D})$ since $\mathsf{X}_i \in \kappa_{\mathcal{D}}(\mathcal{M})$, and $\mathtt{C}_{\mathtt{i},\mathtt{j}-\mathtt{1}} \notin \mathit{deps}(\mathcal{C})$ by definition of case (b);
  - $\mathtt{C}_{\mathtt{i},\mathtt{j}} \in \mathit{deps}(\mathcal{D}) \cap \mathit{deps}(\mathcal{C})$: indeed, $\mathtt{C}_{\mathtt{i},\mathtt{j}} \in \mathit{deps}(\mathcal{D})$ since $\mathsf{X}_i \in \kappa_{\mathcal{D}}(\mathcal{M})$, and $\mathtt{C}_\mathtt{i} \in \mathit{deps}(\mathcal{C})$ by definition of case (b);
  - $\mathit{deps}(\mathcal{D}) \restriction_{\mathcal{C}}$ and $\mathit{deps}(\mathcal{D}) \setminus \mathit{deps}(\mathcal{C})$ are token independent in S by assumption ((iii)).

Claim (2). $\vec{\mathcal{Y}} \in \kappa_{\mathcal{D}\restriction_{\mathcal{C}}}(\mathcal{M})^*$

Proof of Claim (2). Consider a transaction Y_i in $\vec{\mathcal{Y}}$. We have two cases, depending on whether Y_i is due to conditions (a) or (b):

(a) in this case, Y_i corresponds to some $\mathsf{X}_i = \mathtt{M}[i]\colon \mathtt{C}_{\mathtt{i},\mathtt{1}}.\mathtt{f}_{\mathtt{i},\mathtt{1}}(\mathtt{args}_{\mathtt{i},\mathtt{1}})$ in $\vec{\mathcal{X}}$ where $\mathtt{C}_{\mathtt{i},\mathtt{1}} \in \mathit{deps}(\mathcal{C})$. Since $\mathsf{X}_i \in \kappa_{\mathcal{D}}(\mathcal{M})$, then $\mathsf{Y}_i \in \kappa_{\mathcal{D}\restriction_{\mathcal{C}}}(\mathcal{M})$.

(b) by Claim (1), the callee of Y_i is in $\mathcal{C}' = \mathit{deps}(\mathcal{C}) \cap \mathit{deps}(\mathcal{D} \setminus \mathit{deps}(\mathcal{C}))$, which is included in $\mathcal{D}$ by assumption (ii). Note that $\mathcal{M}$ is able to craft the actual arguments of that call by simulating the execution of $\vec{\mathcal{X}}$. This implies that $\mathsf{Y}_i \in \kappa_{\mathcal{D}\restriction_{\mathcal{C}}}(\mathcal{M})$. This completes the proof of Claim (2).

We now show that $\vec{\mathcal{Y}}$ and $\vec{\mathcal{X}}$ modify the state of contracts in $\mathcal{C}$ in exactly the same way. Note that the transactions Y_i that are in $\vec{\mathcal{Y}}$ due to condition (b) have callee in $\mathcal{C}'$ by Claim (1), and so their functions are *sender-agnostic* by assumption (i). So, the fact that in the execution of Y_i they are called directly from a user address, while in the execution of X_i they are called from a contract address, does not affect the execution of these calls. Note that a call $\mathtt{C}_{\mathtt{i,j-1}}\mathtt{:}\,\mathtt{C}_{\mathtt{i,j}}.\mathtt{f}_{\mathtt{i,j}}(\mathtt{args}_{\mathtt{i,j}})$ in X_i could send tokens to the sender $\mathtt{C}_{\mathtt{i,j-1}}$, thus affecting its gain, while the corresponding call $\mathtt{M}[i]\mathtt{:}\,\mathtt{C}_{\mathtt{i,j}}.\mathtt{f}_{\mathtt{i,j}}(\mathtt{args}_{\mathtt{i,j}})$ would send these tokens to $\mathtt{M}[i]$. This difference however do not affect the gains and losses of $\mathcal{C}$, since $\mathtt{C}_{\mathtt{i,j-1}}$ is not in $\mathit{deps}(\mathcal{C})$ by condition (b).

Note that the sequence $\vec{h}$ of calls performed upon the execution of $\vec{\mathcal{Y}}$ is the subsequence of $\vec{x}$ that contains every call to functions of contracts in $\mathit{deps}(\mathcal{C})$. For this reason, both $\vec{x}$ and $\vec{h}$ modify the state of contracts $\mathit{deps}(\mathcal{C})$ in the same way — and, in particular, they cause exactly the same losses to the contracts in $\mathcal{C}$. This implies that $\vec{\mathcal{Y}}$ is valid in S and that $\gamma_{\mathcal{C}}(S, \vec{\mathcal{Y}}) = \gamma_{\mathcal{C}}(S, \vec{\mathcal{X}})$. Since we have proved (3.3) for all possible $\vec{\mathcal{X}}$, we obtain the thesis. □

Theorem 3 is an alternatively stated statement of Theorem 1 in [7].

Theorem 3. $\mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C}) = \mathrm{MEV}^{\infty}_{\mathcal{D}\cap \mathit{deps}(\mathcal{C})}(\Gamma, \mathcal{C})$ *holds if the contracts* $\mathcal{C}' = \mathit{deps}(\mathcal{C}) \cap \mathit{deps}(\mathcal{D} \setminus \mathit{deps}(\mathcal{C}))$ *satisfy: (i)* $\mathcal{C}'$ *are sender-agnostic, and (ii)* $\mathcal{C}' \subseteq \mathcal{D}$.

Proof. First note that $\mathrm{MEV}^{\infty}_{\mathcal{D}\cap \mathit{deps}(\mathcal{C})}(\Gamma, \mathcal{C}) \leq \mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C})$ holds by Item 2, so we just need to show that

$$\mathrm{MEV}^{\infty}_{\mathcal{D}}(\Gamma, \mathcal{C}) \leq \mathrm{MEV}^{\infty}_{\mathcal{D}\cap \mathit{deps}(\mathcal{C})}(\Gamma, \mathcal{C})$$

To do so, we consider a wallet $W_{\mathcal{M}}$ that maximizes $\mathrm{MEV}_{\mathcal{D}}(W_{\mathcal{M}} \mid \Gamma, \mathcal{C})$ (it must exist by Lemma 3), and show that, for any sequence of transactions $\vec{\mathcal{X}} \in \kappa_{\mathcal{D}}(\mathcal{M})^*$ that is valid in $S = W_{\mathcal{M}} \mid \Gamma$, we can find a $W'_{\mathcal{M}}$ and transaction sequence $\vec{\mathcal{Y}} \in \kappa_{\mathcal{D}\cap \mathit{deps}(\mathcal{C})}(\mathcal{M})^*$ valid in $S' = W'_{\mathcal{M}} \mid \Gamma$ such that $-\gamma_{\mathcal{C}}(S', \vec{\mathcal{Y}}) = -\gamma_{\mathcal{C}}(S, \vec{\mathcal{X}})$.

We will first construct $W'_{\mathcal{M}}$. Since the adversary may have different aliases, we rewrite $W_{\mathcal{M}}$ as the composition $w_1 \mid w_2 \mid \cdots$. Moreover, the transactions in $\vec{\mathcal{X}}$ are all valid in S, so their origin must be one of the aliases in the composition. The sequence $\vec{\mathcal{X}}$ is finite, so we can assume wlog that the aliases which are the origin of some transaction are the first n appearing in the composition. Finally, we let

$$W'_{\mathcal{M}} = w_1 + w' \mid \cdots \mid w_n + w' \mid w_{n+1} \mid \cdots$$

where w' consists of the sum of all tokens that have been transferred during the execution of $\vec{\mathcal{X}}$ in state S (both directly and due to internal method calls). Note that the tokens of w' are added only to the wallets that are origin of some transaction in $\vec{\mathcal{X}}$, so the finite tokens axiom is still satisfied.

We construct $\vec{\mathcal{Y}}$ exactly as in Theorem 2, and note that if any of the original calls transferred some tokens, then the corresponding transaction of $\vec{\mathcal{Y}}$ will provide the same amount of tokens, taking them from the wallet owned by the alias that originated the method call. By the construction of $W'_{\mathcal{M}}$, there are always enough tokens to do so. The rest of the proof remains the same as Theorem 2. □

Theorem 4 is an alternatively stated result of Theorem 4 in [7].

Theorem 4 (Contract Stripping). *If contracts in* $\mathit{deps}(\Delta) \cap \mathit{deps}(\dagger\Gamma \setminus \mathit{deps}(\Delta))$ *are sender-agnostic, then*

$$\Gamma \not\rightsquigarrow^{\infty} \Delta \iff \mathrm{MEV}^{\infty}_{\dagger(\Gamma \mid \Delta) \cap \mathit{deps}(\Delta)}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta)$$

Proof. We start by showing the following equality:

$$\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger(\Gamma \mid \Delta) \cap \mathit{deps}(\Delta)}(\Gamma \mid \Delta, \dagger\Delta) \tag{3.4}$$

By letting $\mathcal{D} = \dagger(\Gamma \mid \Delta)$, and

$$\mathcal{C}' = \mathit{deps}(\Delta) \cap \mathit{deps}(\dagger(\Gamma \mid \Delta) \setminus \mathit{deps}(\Delta)) = \mathit{deps}(\Delta) \cap \mathit{deps}(\dagger\Gamma \setminus \mathit{deps}(\Delta))$$

we can see that $\mathcal{C}' \subseteq \mathcal{D}$ and that contracts in $\mathcal{C}'$ are sender-agnostic (by assumption). This means that both conditions of Theorem 3 are satisfied, and we have proven (3.4). Now, recall that $\Gamma \not\rightsquigarrow^{\infty} \Delta$ implies:

$$\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta) \tag{3.5}$$

Now, combining (3.4) and (3.5) gives us our thesis. □

Chapter 4

Towards MEV Interference

In this chapter, we propose an initial definition of quantitative MEV interference: We quantify the interference caused by a blockchain state S on newly deployed contracts Δ as $p(S, \Delta)$. We study its theoretical properties and discuss why they are desirable. Simultaneously, we provide examples to demonstrate our results.

4.1 A preliminary definition

Definition 11. For a blockchain state $S = W \mid \Gamma$ and a contract state Δ, we quantify the MEV interference caused by S on Δ as:

$$p(S, \Delta) = \frac{\mathrm{MEV}(S \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} \qquad \textit{if } \$_{\dagger\Delta}(\Delta) \neq 0$$

When $\$_{\dagger\Delta}(\Delta) = 0$, we define $p(S, \Delta) = 0$.

Example 1. Consider the `Betp` contract in Figure 4.2, which allows a player to bet on the exchange rate between a token and `ETH`. It is parameterized over an `oracle` that is queried for token prices. `Betp` receives its initial pot from its owner upon deployment. To join, a player must pay an amount of `ETH` equal to the pot. The winner receives a proportion `potShare` (set by the owner on deployment) of the total pot, when the oracle exchange rate exceeds `potShare` times the bet rate. The remaining part is taken by the owner as a fee. Without loss of generality, we can assume that `potShare` $\geq 1/2$, since a smaller proportion would make the bet irrational for any player. Consider the following instance of the `Betp` contract using the `AMM`

```
contract AMM {
  addLiq(?x0:T0,?x1:T1) { // add liquidity to the AMM
    require #T0 * (#T1-x1) == (#T0-x0) * #T1 }
  getTokens() { return (T0,T1) }     // token pair
  getRate(t) {                       // exchange rate
    if (t==T0) return #T0/#T1        // r:T0 for 1:T1
    else if (t==T1) return #T1/#T0   // r:T1 for 1:T0
    else abort }
  swap(?x:t,ymin) {
    if (t==T0)
      { y=(x*#T1)/#T0; require ymin<=y<#T1; sender!y:T1 }
    else if (t==T1)
      { y=(x*#T0)/#T1; require ymin<=y<#T0; sender!y:T0 }
    else abort }
}
```

Figure 4.1: A constant-product AMM contract.

in Figure 4.1 as a price oracle, where tokens are assigned unitary prices:

$$\begin{aligned} S &= \mathtt{M}[320\colon \mathtt{ETH}] \mid \mathtt{AMM}[600\colon \mathtt{ETH}, 600\colon \mathtt{T}] \mid \mathtt{block.num} = d - k \mid \cdots \\ \Delta &= \mathtt{Betp}[10\colon \mathtt{ETH}, \mathtt{owner} = \mathtt{A}, \mathtt{tok} = \mathtt{T}, \mathtt{rate} = 2, \mathtt{deadline} = d, \mathtt{potShare} = 3/4] \end{aligned}$$

Consider the computation:

$$\begin{aligned} S \mid \Delta &\xrightarrow{\mathtt{M}:\mathtt{Betp.bet}(?\,10:\mathtt{ETH})} && \mathtt{M}[310\colon \mathtt{ETH}] \mid \mathtt{AMM}[600\colon \mathtt{ETH}, 600\colon \mathtt{T}] \mid \mathtt{Betp}[20\colon \mathtt{ETH}, \cdots] \\ &\xrightarrow{\mathtt{M}:\mathtt{AMM.swap}(?\,300:\mathtt{ETH},0)} && \mathtt{M}[10\colon \mathtt{ETH}, 200\colon \mathtt{T}] \mid \mathtt{AMM}[900\colon \mathtt{ETH}, 400\colon \mathtt{T}] \mid \mathtt{Betp}[20\colon \mathtt{ETH}, \cdots] \\ &\xrightarrow{\mathtt{M}:\mathtt{Betp.win}()} && \mathtt{M}[25\colon \mathtt{ETH}, 200\colon \mathtt{T}] \mid \mathtt{AMM}[900\colon \mathtt{ETH}, 400\colon \mathtt{T}] \mid \mathtt{Betp}[5\colon \mathtt{ETH}, \cdots] \\ &\xrightarrow{\mathtt{M}:\mathtt{AMM.swap}(?\,200:\mathtt{T},0)} && \mathtt{M}[325\colon \mathtt{ETH}] \mid \mathtt{AMM}[600\colon \mathtt{ETH}, 600\colon \mathtt{T}] \mid \mathtt{Betp}[5\colon \mathtt{ETH}, \cdots] \end{aligned}$$

By Definition 3, we have that $S \rightsquigarrow \Delta$, since:

$$\begin{aligned} \mathrm{MEV}(S \mid \Delta, \dagger\Delta) &= \mathrm{MEV}_{\{\mathtt{AMM},\mathtt{Betp}\}}(S \mid \Delta, \{\mathtt{Betp}\}) = ({}^{3}\!/_{4} \cdot 20) - 10 = 15 - 10 = 5 \\ \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) &= \mathrm{MEV}_{\{\mathtt{Betp}\}}(S \mid \Delta, \{\mathtt{Betp}\}) = 0 \end{aligned}$$

Quantitative MEV interference is estimated through Definition 11 as follows:

$$p(S, \Delta) = \frac{\mathrm{MEV}(S \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} = \frac{5 - 0}{\$_{\{\mathtt{Betp}\}}(\Delta)} = \frac{5}{10} = \frac{1}{2}$$

◇

```
contract Betp_oracle {
  constructor(?x:ETH,t,r,d,p) {
    require t!=ETH && oracle.getTokens()==(ETH,t);
    tok=t; rate=r; owner=origin; deadline=d; potShare=p
  }
  bet(?x:ETH) {
    require player==null && x==#ETH;
    player=origin
  }
  win() {
    require block.num<=deadline && origin==player;
    require oracle.getRate(ETH)>rate;
    player!(potShare*#ETH):ETH
  }
  close() {
    require block.num>deadline && origin==owner;
    owner!#ETH:ETH
  }
}
```

Figure 4.2: The Betp contract.

Proposition 1 relates the *qualitative* notion of interference introduced in [7] to the *quantitative* one: when S is non-interfering with Δ, our quantitative definition gives us an interference value of 0 and vice-versa.

Proposition 1 (p ***vs.*** $\not\rightsquigarrow$)**.** $p(S, \Delta) = 0 \iff S \not\rightsquigarrow \Delta$

Proof. If $p(S, \Delta) = 0$, then it must be $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)$ or $\$_{\dagger\Delta}(\Delta) = 0$. In the first case, by Definition 7 we directly have the thesis. In the second case, by Lemma 1(5) it must be $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) = 0$, from which the thesis follows. To prove the other implication, if $S \not\rightsquigarrow \Delta$ then by Definition 7 it must be $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)$. The thesis follows from Definition 11.

□

Lemma 7 provides some basic properties of p. Items 1 and 2 study border cases: Item 1 states that when $\Delta = \emptyset$, i.e. there are no newly deployed contracts to check interference on, the blockchain state S does not interfere with Δ, as one would naturally expect. Item 2 states that when the blockchain state S only consists of user wallets but no contracts, S non-interferes with any newly deployed contracts. Likewise, this is coherent with our intuition because there is no contract in S that an

```
contract Airdrop {
  constructor(?x:t) { tout=t }      // deposit any token t
  withdraw() { sender!#tout:tout } // any user withdraws
}
contract Exchange {
  constructor(?x:t1,t2,r) {
    require r>0;
    rate=r; tout=t1; tin=t2; owner=origin
  }
  getTokens() {
    return (tin,tout)
  }
  getRate() {
    return rate
  }
  setRate(newRate) {
    require origin==owner;
    rate=newRate
  }
  swap(?x:t) {             // receives x units of tin
    require t==tin && #tout>=x*rate;
    sender!x*rate:tout     // sends x*rate units of tout
  }
}
```

Figure 4.3: An airdrop and an exchange contract.

adversary could leverage to inflict a loss to Δ. Item 3 states that the interference is bounded between 0 and 1. An interference value of 0 corresponds to the case where we have no interference from S to Δ, while a value of 1 corresponds to the case where we have the maximum interference and Δ is drained of all funds. This is preferable since we would like to readily verify when contracts suffer *no* loss or a *complete* loss of funds upon deployment.

4.2 Theoretical Properties

Lemma 7 (Basic properties of p). *For all S, Δ:*

1. $p(S, \emptyset) = 0$
2. $p(W \mid \emptyset, \Delta) = 0$
3. $0 \leq p(S, \Delta) \leq 1$

Proof. For (1), note that $\$_{\dagger\emptyset}(\emptyset) = 0$, and so the thesis follows by Definition 11.

For (2), in the case where $\$_{\dagger\Delta}(\Delta) = 0$, we have $p(S, \Delta) = 0$ by definition. And in the case where $\$_{\dagger\Delta}(\Delta) \neq 0$, we have that:

$$p(W \mid \emptyset, \Delta) = \frac{\mathrm{MEV}(W \mid \emptyset \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(W \mid \emptyset \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)}$$

By Lemma 1(4), $\mathrm{MEV}(W \mid \emptyset \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(W \mid \emptyset \mid \Delta, \dagger\Delta)$, this gives $p(W \mid \emptyset, \Delta) = 0$.

For (3), in the case where $\$_{\dagger\Delta}(\Delta) = 0$, we have $p(S, \Delta) = 0$ by definition. And in the case where $\$_{\dagger\Delta}(\Delta) \neq 0$, we have that:

$$\begin{aligned} 0 &\leq \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) && \text{by Lemma 1(5)} \\ &\leq \mathrm{MEV}(S \mid \Delta, \dagger\Delta) && \text{by Lemma 1(2)} \\ &\leq \$_{\dagger\Delta}(\Delta) && \text{by Lemma 1(5)} \end{aligned}$$

which implies $0 \leq p(S, \Delta) \leq 1$.

□

Proposition 2 states that widening the blockchain state S potentially increases MEV interference. This is because, by appending S with contract states Γ, we are enabling the adversary to invoke a larger set of contracts, some of which may potentially be leveraged to cause a greater loss to newly deployed contracts Δ. Furthermore, we make note of a few important observations that substantiate the reasonableness of our thesis in Proposition 2. Let $S \mid \Gamma$ be a widening of S, i.e. $S \preceq S \mid \Gamma$. By hypothesis, S is well-formed, i.e. any valid transaction sent to S never calls methods of contracts outside of S. This requires that all the contract dependencies of $\dagger S$ are self-contained, i.e. $\mathit{deps}(S) \subseteq \dagger S$. (Notice that on the other hand, there is no such restriction on $\dagger\Gamma$, i.e. $\dagger\Gamma$ can have dependencies in Γ as well as in S. Formally, we have $\mathit{deps}(\Gamma) \subseteq \dagger(S \mid \Gamma)$.) Since we assume $S \mid \Delta$ to be well-formed and that the dependencies of contracts are fixed, we have that widening the state Γ does not affect the dependencies of $\dagger\Delta$, i.e. $\mathit{deps}(\Delta) \subseteq \dagger(S \mid \Delta)$. Moreover, we note that when we compose $S \mid \Gamma$ with Δ, we are implicitly assuming that the contracts names in Δ are disjoint from those in $S \mid \Gamma$, i.e. $\dagger(S \mid \Gamma) \cap \dagger\Delta = \emptyset$.

Proposition 2. *If $\mathit{deps}(\Delta) \cap \dagger\Gamma = \emptyset$, then:*

$$p(S, \Delta) \leq p(S \mid \Gamma, \Delta)$$

Proof. By Definition 11, we have two cases.

If $\$_{\dagger\Delta}(\Delta) = 0$, then $p(W \mid \Gamma, \Delta) = p(W \mid \Gamma', \Delta) = 0$, hence the thesis holds trivially.

Otherwise, assume that $\$_{\dagger\Delta}(\Delta) \neq 0$. Then, by Definition 11:

$$\begin{aligned} p(S, \Delta) &= \frac{\mathrm{MEV}(S \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} \\ p(S \mid \Gamma, \Delta) &= \frac{\mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} \end{aligned}$$

From Lemma 1, we have that:

$$\begin{aligned} \mathrm{MEV}_{\dagger(S \mid \Delta)}(S \mid \Delta, \dagger\Delta) &\leq \mathrm{MEV}_{\dagger(S \mid \Delta)}(S \mid \Gamma \mid \Delta, \dagger\Delta) && \text{by Item 3} \\ &\leq \mathrm{MEV}_{\dagger(S \mid \Gamma \mid \Delta)}(S \mid \Gamma \mid \Delta, \dagger\Delta) && \text{by Item 2} \end{aligned} \tag{4.1}$$

We know from (4.1),

$$\mathrm{MEV}(S \mid \Delta, \dagger\Delta) \leq \mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta)$$

By Lemma 5, we have

$$\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma \mid \Delta, \dagger\Delta) \tag{4.2}$$

Then:

$$\mathrm{MEV}(S \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) \leq \mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma \mid \Delta, \dagger\Delta)$$

which finally gives us:

$$\frac{\mathrm{MEV}(S \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} \leq \frac{\mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)}$$

which gives our thesis, i.e. $p(S, \Delta) \leq p(S \mid \Gamma, \Delta)$. □

Example 2. To show that a larger contract state might *strictly increase* MEV interference, consider a composition of the `Airdrop` and `Exchange` in Figure 7.1, where the state S only includes the adversary's wallet and no contract accounts. The intuition here is that the adversary `M` is able to leverage token dependencies between the newly deployed contracts $\dagger\Gamma_\varepsilon$ to extract more MEV from $\dagger\Delta$. More precisely, consider the following instance:

$$\begin{aligned} S &= W \mid \Gamma = \texttt{M}[0\colon \texttt{T}] \mid \emptyset \\ \Gamma_\varepsilon &= \texttt{Airdrop}[1\colon \texttt{T}, \texttt{tout} = \texttt{T}] \\ \Delta &= \texttt{Exchange}[100\colon \texttt{ETH}, \texttt{tin} = \texttt{T}, \texttt{tout} = \texttt{ETH}, \texttt{rate} = 10, \texttt{owner} = \texttt{B}] \end{aligned}$$

Observe that by Definition 3, assuming unitary prices, we have that:

$$\begin{aligned} &\$_{\dagger\Delta}(\Delta) = \$_{\{\texttt{Exchange}\}}(\Delta) = 100 \\ &\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\{\texttt{Exchange}\}}(S \mid \Delta, \{\texttt{Exchange}\}) = 0 \\ &\mathrm{MEV}(S \mid \Gamma_\varepsilon \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\{\texttt{Airdrop},\texttt{Exchange}\}}(S \mid \Gamma_\varepsilon \mid \Delta, \{\texttt{Exchange}\}) = 10 \\ &\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\{\texttt{Exchange}\}}(S \mid \Delta, \{\texttt{Exchange}\}) = 0 \\ &\mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma_\varepsilon \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\{\texttt{Exchange}\}}(S \mid \Gamma_\varepsilon \mid \Delta, \{\texttt{Exchange}\}) = 0 \end{aligned}$$

Therefore $S \not\rightsquigarrow \Delta$, while $S \mid \Gamma_\varepsilon \rightsquigarrow \Delta$. With Definition 11 we have:

$$\begin{aligned} &p(S, \Delta) = 0 \\ &p(S \mid \Gamma_\varepsilon, \Delta) = \frac{\mathrm{MEV}(S \mid \Gamma_\varepsilon \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma_\varepsilon \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} = \frac{10 - 0}{100} = 0.1 \end{aligned}$$

$\diamond$

In Remark 1, we show that we cannot infer in general that wealthier adversaries are always able to cause more interference. However, this does not imply that wealthier adversaries will never be able to cause more interference to a blockchain state. To demonstrate this, we show two examples after the proposition: in one case, the wealth of the adversary plays a role in determining the interference caused, while in the second case, the wealth of the adversary is not a determining factor.

Remark 1. *We observe that:*

$$S \leq_\$ S' \;\not\Rightarrow\; p(S, \Delta) \leq p(S', \Delta)$$

Example 3. To show that a wealthier adversary might potentially *increase* the MEV interference, consider the AMM-Bet composition with the entry fee set by the owner of the Bet contract is a large sum. For instance:

$$\begin{aligned}
S &= \texttt{M}[10\colon \texttt{ETH}] \mid \texttt{AMM}[600\colon \texttt{ETH}, 600\colon \texttt{T}] \mid \texttt{block.num} = d - k \mid \cdots \\
S' &= \texttt{M}[310\colon \texttt{ETH}] \mid \texttt{AMM}[600\colon \texttt{ETH}, 600\colon \texttt{T}] \mid \texttt{block.num} = d - k \mid \cdots \\
\Delta &= \texttt{Bet}[10\colon \texttt{ETH}, \texttt{owner} = \texttt{A}, \texttt{tok} = \texttt{T}, \texttt{rate} = 2, \texttt{deadline} = d, \texttt{potShare} = 3/4]
\end{aligned}$$

Observe here that when we are in state S, M will not be able to perform an attack on Bet using AMM because she does not possess the necessary wealth to both enter the bet and produce a volatility in AMM. Instead, when M possesses more wealth i.e. we are in state S', we have that $p(S', \Delta) = 1/2$ as shown in Example 1. Hence, this example shows an instance where a wealthier state implies an increased interference caused to Δ as compared to a poorer state. ◇

Example 4. To show that a wealthier adversary might potentially *decrease* the MEV interference caused to contracts $\dagger\Delta$, consider again the example of the Airdrop and Exchange in Figure 7.1. Let:

$$\begin{aligned}
S &= \texttt{M}[0\colon \texttt{T}] \mid \texttt{Airdrop}[2\colon \texttt{T}, \texttt{tout} = \texttt{T}] \\
S' &= \texttt{M}[9\colon \texttt{T}] \mid \texttt{Airdrop}[2\colon \texttt{T}, \texttt{tout} = \texttt{T}] \\
\Delta &= \texttt{Exchange}[100\colon \texttt{ETH}, \texttt{tin} = \texttt{T}, \texttt{tout} = \texttt{ETH}, \texttt{rate} = 10, \texttt{owner} = \texttt{B}]
\end{aligned}$$

Assuming unitary prices assigned to all tokens, we have

$$\$_{\dagger\Delta}(\Delta) = \$_{\{\texttt{Exchange}\}}(\Delta) = 100$$

In the case where our blockchain state is S, by Definition 3, we have that:

$$\begin{aligned}
\mathrm{MEV}(S \mid \Delta, \dagger\Delta) &= \mathrm{MEV}_{\{\texttt{Airdrop},\texttt{Exchange}\}}(S \mid \Delta, \{\texttt{Exchange}\}) = 20 \\
\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) &= \mathrm{MEV}_{\{\texttt{Exchange}\}}(S \mid \Delta, \{\texttt{Exchange}\}) = 0
\end{aligned}$$

Therefore $S \rightsquigarrow \Delta$, and with Definition 11 we have:

$$p(S, \Delta) = \frac{\mathrm{MEV}(S \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} = \frac{20 - 0}{100} = 0.2$$

Whereas, in the case where our blockchain state is S', by Definition 3, we have that:

$$\begin{aligned}
&\mathrm{MEV}(S' \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\{\texttt{Airdrop},\texttt{Exchange}\}}(S' \mid \Delta, \{\texttt{Exchange}\}) = 100 \\
&\mathrm{MEV}_{\dagger\Delta}(S' \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\{\texttt{Exchange}\}}(S' \mid \Delta, \{\texttt{Exchange}\}) = 90
\end{aligned}$$

Therefore $S' \rightsquigarrow \Delta$, and by Definition 11 we have:

$$p(S', \Delta) = \frac{\mathrm{MEV}(S' \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(S' \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} = \frac{100 - 90}{100} = 0.1$$

Hence, $p(S, \Delta) = 0.2$ while $p(S', \Delta) = 0.1$, i.e. the interference caused to `Exchange` is greater in a poorer state than in a wealthier state. This is because the richer adversary in S' gets less advantage from `Airdrop` to cause a loss to `Exchange`. $\diamond$

Proposition 3 states that to calculate MEV interference, we only need to focus on adversary wallets. Stated in another way, we have that the MEV interference caused to $\dagger\Delta$ by the state $W_{\mathcal{M}} \mid W \mid \Gamma$ is equal to that caused by $W_{\mathcal{M}} \mid \Gamma$, i.e with the non-adversarial wallets removed.

Proposition 3.

$$\operatorname{dom} W_{\mathcal{M}} = \mathcal{M} \implies p(W_{\mathcal{M}} \mid W \mid \Gamma, \Delta) = p(W_{\mathcal{M}} \mid \Gamma, \Delta)$$

Proof. By Definition 11, we have two cases. If $\$_{\dagger\Delta}(\Delta) = 0$, then $p(W_{\mathcal{M}} \mid W \mid \Gamma, \Delta) = p(W_{\mathcal{M}} \mid \Gamma, \Delta) = 0$, hence the thesis holds trivially. Otherwise, if $\$_{\dagger\Delta}(\Delta) \neq 0$:

$$p(W_{\mathcal{M}} \mid W \mid \Gamma, \Delta) = \frac{\mathrm{MEV}_{\dagger(\Gamma \mid \Delta)}(W_{\mathcal{M}} \mid W \mid \Gamma \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(W_{\mathcal{M}} \mid W \mid \Gamma \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} \quad \text{and}$$

$$p(W_{\mathcal{M}} \mid \Gamma, \Delta) = \frac{\mathrm{MEV}_{\dagger(\Gamma \mid \Delta)}(W_{\mathcal{M}} \mid \Gamma \mid \Delta, \dagger\Delta) - \mathrm{MEV}_{\dagger\Delta}(W_{\mathcal{M}} \mid \Gamma \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)}$$

Now, from Item 1 in Lemma 2, we have

$$\begin{aligned}
&\mathrm{MEV}_{\dagger(\Gamma \mid \Delta)}(W_{\mathcal{M}} \mid W \mid \Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger(\Gamma \mid \Delta)}(W_{\mathcal{M}} \mid \Gamma \mid \Delta, \dagger\Delta), \quad \text{and} \\
&\mathrm{MEV}_{\dagger\Delta}(W_{\mathcal{M}} \mid W \mid \Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(W_{\mathcal{M}} \mid \Gamma \mid \Delta, \dagger\Delta)
\end{aligned}$$

And so we have:

$$p(W_{\mathcal{M}} \mid W \mid \Gamma, \Delta) = p(W_{\mathcal{M}} \mid \Gamma, \Delta)$$

□

Remark 2. *Widening Δ does not necessarily increase the interference caused by S on Δ. In other words, given that $S \mid \Delta$ is a well-formed state, we have*

$$\Delta \preceq \Delta \mid \Delta_{\varepsilon} \not\Longrightarrow p(S, \Delta) \leq p(S, \Delta \mid \Delta_{\varepsilon})$$

Example 5. Consider the following instance to illustrate the non-implication above:

$$\begin{aligned} S = W \mid \Gamma &= \mathtt{M}[320\colon \mathtt{ETH}] \mid \mathtt{AMM}[600\colon \mathtt{ETH}, 600\colon \mathtt{T}] \mid \mathtt{block.num} = d - k \mid \cdots \\ \Delta &= \mathtt{Betp1}[10\colon \mathtt{ETH}, \mathtt{owner} = \mathtt{A}, \mathtt{tok} = \mathtt{T}, \mathtt{rate} = 2, \mathtt{deadline} = d, \mathtt{potShare} = 1] \\ \Delta \mid \Delta_{\varepsilon} &= \Delta \mid \mathtt{Betp2}[10\colon \mathtt{ETH}, \mathtt{owner} = \mathtt{A}, \mathtt{tok} = \mathtt{T}, \mathtt{rate} = 2, \mathtt{deadline} = d, \mathtt{potShare} = 3/4] \end{aligned}$$

Clearly, $p(S, \Delta) = 1$. While,

$$\begin{aligned} p(S, \Delta \mid \Delta_{\varepsilon}) &= \frac{\mathrm{MEV}(S \mid \Delta \mid \Delta_{\varepsilon}, \dagger(\Delta \mid \Delta_{\varepsilon})) - \mathrm{MEV}_{\dagger(\Delta \mid \Delta_{\varepsilon})}(S \mid \Delta \mid \Delta_{\varepsilon}, \dagger(\Delta \mid \Delta_{\varepsilon}))}{\$_{\dagger(\Delta \mid \Delta_{\varepsilon})}(\Delta \mid \Delta_{\varepsilon})} \\ &= \frac{10 + 5 - 0}{10 + 10} = \frac{15 - 0}{20} = \frac{3}{4} = 0.75 \end{aligned}$$

Hence $p(S, \Delta) \not\leq p(S, \Delta \mid \Delta_{\varepsilon})$. ◇

We note that the non-implication in Remark 2 arises primarily due to the following reason: when we widen the contract state Δ to $\Delta \mid \Delta_{\varepsilon}$, the total value locked of the *whole* contract state increases, i.e. $\$_{\dagger(\Delta \mid \Delta_{\varepsilon})}(\Delta \mid \Delta_{\varepsilon}) \geq \$_{\dagger\Delta}(\Delta)$. This results in a larger denominator in the computation of $S \rightsquigarrow \Delta \mid \Delta_{\varepsilon}$ (in comparison to $S \rightsquigarrow \Delta$) which reduces the interference caused by S to the *total* contract state $\Delta \mid \Delta_{\varepsilon}$.

Chapter 5

Revisiting MEV Interference

In this chapter, we study an alternate definition of MEV interference which we call p^{∞}: it represents a notion of interference independent of user wallets, and more specifically, of adversarial wealth. Throughout the chapter, we assume that adversaries possess unbounded wealth. By doing so, we are empowering the adversary to inflict more loss on contracts, since wealth is no longer a limitation on conducting attacks. In practice, this is possible when adversaries capitalize on *flash loans* which allow users to borrow assets without providing any upfront collateral with the condition that it must be returned within the same blockchain transaction. Otherwise, the entire transaction is reverted.

5.1 MEV Interference with wealthy adversaries

Definition 12. For contract states Γ and Δ, we quantify the amount of MEV$^{\infty}$ interference caused by Γ on Δ as:

$$p^{\infty}(\Gamma, \Delta) = \frac{\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) - \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} \qquad \textit{if } \$_{\dagger\Delta}(\Delta) \neq 0$$

When $\$_{\dagger\Delta}(\Delta) = 0$, we define $p^{\infty}(\Gamma, \Delta) = 0$.

Proposition 4, Lemma 8 and Proposition 5 state that p^{∞} adheres to the basic properties that we have seen before for p (Proposition 1, Lemma 7 and Proposition 2).

Proposition 4 (p^{∞} ***vs.*** $\not\rightsquigarrow^{\infty}$)**.** $p^{\infty}(\Gamma, \Delta) = 0 \iff \Gamma \not\rightsquigarrow^{\infty} \Delta$

Proof. If $p^{\infty}(\Gamma, \Delta) = 0$, then it must be $\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta)$ or $\$_{\dagger\Delta}(\Delta) = 0$. In the first case, by Definition 8 we directly have the thesis. In the second case, by Lemma 4(5) it must be $\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta) = 0$, from which the thesis follows. To prove the other implication, if $\Gamma \not\rightsquigarrow^{\infty} \Delta$ then by Definition 8 it must be $\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta)$. The thesis follows from Definition 12. □

5.2 Theoretical Properties

Lemma 8 (Basic properties of p^{∞}). *For all* Γ, Δ*:*

1. $p^{\infty}(\Gamma, \emptyset) = 0$
2. $p^{\infty}(\emptyset, \Delta) = 0$
3. $0 \leq p^{\infty}(\Gamma, \Delta) \leq 1$

Proof. Items (1), (2), (3) have analogous statements in Lemma 7 which hold for any wallet state. The corresponding results for MEV also hold for MEV^{∞} due to Lemma 3 and Definition 6. □

Proposition 5. *If* $\mathit{deps}(\Delta) \cap \dagger\Gamma_{\varepsilon} = \emptyset$*, then:*

$$p^{\infty}(\Gamma, \Delta) \leq p^{\infty}(\Gamma \mid \Gamma_{\varepsilon}, \Delta)$$

Proof. Because (4.1) and (4.2) hold for any wallet state, analogous "rich-adversary" versions of these equations can be proved using Item 2 and Item 2 of Lemma 4 and Lemma 6. We can then re-use the same reasoning to prove the proposition. □

Example 6. To demonstrate that a larger contract state might *strictly* increase the MEV^{∞} interference caused to Δ, consider the contracts in Figure 5.1 and let:

$$\Gamma = \texttt{C2}[0\colon \texttt{T}] \quad \Gamma_{\varepsilon} = \texttt{C1}[0\colon \texttt{T}] \quad \Delta = \texttt{C0}[5\colon \texttt{T}]$$

Observe that by Definition 6, assuming unitary prices, we have that $\$_{\dagger\Delta}(\Delta) = 5$ and

$$\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Gamma_{\varepsilon} \mid \Delta, \dagger\Delta) = 0$$

```
contract C0 { f() { require sender==C1; sender!5:T} }
contract C1 { g() { C0.f(); sender!#T:T} }
contract C2 { h() { C0.f(); sender!#T:T} }
```

Figure 5.1: Contracts for Example 6.

Instead, $\mathrm{MEV}^{\infty}(\Gamma \mid \Gamma_{\varepsilon} \mid \Delta, \dagger\Delta) = 5$. Therefore $\Gamma \not\rightsquigarrow^{\infty} \Delta$, while $\Gamma \mid \Gamma_{\varepsilon} \rightsquigarrow \Delta$. Now, with Definition 11, we have $p^{\infty}(\Gamma, \Delta) = 0$ while $p^{\infty}(\Gamma \mid \Gamma_{\varepsilon}, \Delta) = 1$. Consequently, we have $p^{\infty}(\Gamma, \Delta) < p^{\infty}(\Gamma \mid \Gamma_{\varepsilon}, \Delta)$. $\diamond$

Proposition 6 gives sufficient conditions under which it is possible to extend Γ with Γ_{ε} without altering the interference caused to contracts in Δ. When contracts $\mathcal{C}' = \mathit{deps}(\Delta) \cap \mathit{deps}(\dagger(\Gamma \mid \Gamma_{\varepsilon}) \setminus \mathit{deps}(\Delta))$ are sender-agnostic, i.e. they are not influenced by the caller identity, then we can strip away all the non-dependencies of Δ from contracts in $\Gamma \mid \Gamma_{\varepsilon}$. Since $\mathit{deps}(\Delta)$ is restricted to contracts $\subseteq \Gamma$, stripping away contracts in Γ_{ε} does not affect the interference that an adversary $\mathcal{M}$ can cause to $\dagger\Delta$. Furthermore, Proposition 7 provides a synonymous proposition where we prepend Γ with Γ_{ε} instead.

Proposition 6. *Given* $\mathit{deps}(\Delta) \cap \dagger\Gamma_{\varepsilon} = \emptyset$*,*

$$p^{\infty}(\Gamma, \Delta) = p^{\infty}(\Gamma \mid \Gamma_{\varepsilon}, \Delta)$$

when contracts in $\mathit{deps}(\Delta) \cap \mathit{deps}(\dagger(\Gamma \mid \Gamma_{\varepsilon}) \setminus \mathit{deps}(\Delta))$ *are sender-agnostic.*

Proof. For simplicity, we let $\Gamma' = \Gamma \mid \Gamma_{\varepsilon}$. By Definition 12, we have two cases. If $\$_{\dagger\Delta}(\Delta) = 0$, then $p^{\infty}(\Gamma, \Delta) = p^{\infty}(\Gamma', \Delta) = 0$, hence the thesis holds trivially. Otherwise, if $\$_{\dagger\Delta}(\Delta) \neq 0$, we can expand each of the interferences as follows:

$$p^{\infty}(\Gamma, \Delta) = \frac{\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) - \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)}$$
$$p^{\infty}(\Gamma', \Delta) = \frac{\mathrm{MEV}^{\infty}(\Gamma' \mid \Delta, \dagger\Delta) - \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma' \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)}$$

We will show the following two sets of equalities:

$$\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}(\Gamma' \mid \Delta, \dagger\Delta) \tag{5.1}$$
$$\mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger\Delta}(\Gamma' \mid \Delta, \dagger\Delta) \tag{5.2}$$

Observe that (5.2) follows directly from Lemma 6. Now, to prove (5.1), we start by proving the following equality using Theorem 3:

$$\mathrm{MEV}^{\infty}(\Gamma' \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger(\Gamma'|\Delta)\cap \mathit{deps}(\Delta)}(\Gamma' \mid \Delta, \dagger\Delta) \tag{5.3}$$

By letting $\mathcal{D} = \dagger(\Gamma' \mid \Delta)$, and

$$\mathcal{C}' = \mathit{deps}(\Delta) \cap \mathit{deps}(\dagger(\Gamma' \mid \Delta) \setminus \mathit{deps}(\Delta)) = \mathit{deps}(\Delta) \cap \mathit{deps}(\dagger\Gamma' \setminus \mathit{deps}(\Delta))$$

we can see that $\mathcal{C}' \subseteq \mathcal{D}$ and that contracts in $\mathcal{C}'$ are sender-agnostic (by assumption). This means that both conditions of Theorem 4 are satisfied and we have proven (5.3). Now, since $\mathit{deps}(\Delta) \cap \dagger\Gamma_{\varepsilon} = \emptyset$, we have that:

$$\dagger(\Gamma' \mid \Delta) \cap \mathit{deps}(\Delta) = \dagger(\Gamma \mid \Gamma_{\varepsilon} \mid \Delta) \cap \mathit{deps}(\Delta) = \dagger(\Gamma \mid \Delta) \cap \mathit{deps}(\Delta)$$

Hence, we can rewrite (5.3) as

$$\mathrm{MEV}^{\infty}(\Gamma' \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger(\Gamma|\Delta)\cap \mathit{deps}(\Delta)}(\Gamma' \mid \Delta, \dagger\Delta) \tag{5.4}$$

Next, using Theorem 3 again, we prove the equality

$$\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger(\Gamma|\Delta)\cap \mathit{deps}(\Delta)}(\Gamma \mid \Delta, \dagger\Delta) \tag{5.5}$$

First, observe

$$\begin{aligned}
&\mathit{deps}(\Delta) \cap \mathit{deps}(\dagger\Gamma' \setminus \mathit{deps}(\Delta)) \\
=&\mathit{deps}(\Delta) \cap \mathit{deps}((\dagger\Gamma \cup \dagger\Gamma_{\varepsilon}) \setminus \mathit{deps}(\Delta)) \\
=&\mathit{deps}(\Delta) \cap \mathit{deps}((\dagger\Gamma \setminus \mathit{deps}(\Delta)) \cup (\dagger\Gamma_{\varepsilon} \setminus \mathit{deps}(\Delta))) \\
=&\mathit{deps}(\Delta) \cap \{\mathit{deps}(\dagger\Gamma \setminus \mathit{deps}(\Delta)) \cup \mathit{deps}(\dagger\Gamma_{\varepsilon} \setminus \mathit{deps}(\Delta))\} \\
=&\{\mathit{deps}(\Delta) \cap \mathit{deps}(\dagger\Gamma \setminus \mathit{deps}(\Delta))\} \cup \{\mathit{deps}(\Delta) \cap \mathit{deps}(\dagger\Gamma_{\varepsilon} \setminus \mathit{deps}(\Delta))\}
\end{aligned}$$

Since $\mathcal{C}' = \mathit{deps}(\Delta) \cap \mathit{deps}(\dagger\Gamma' \setminus \mathit{deps}(\Delta))$ are sender-agnostic, we have that $\mathit{deps}(\Delta) \cap \mathit{deps}(\dagger\Gamma \setminus \mathit{deps}(\Delta))$ are also sender-agnostic. To prove (5.5), this time we let $\mathcal{D} = \dagger(\Gamma \mid \Delta)$ and

$$\mathcal{C}' = \mathit{deps}(\Delta) \cap \mathit{deps}(\dagger(\Gamma \mid \Delta) \setminus \mathit{deps}(\Delta)) = \mathit{deps}(\Delta) \cap \mathit{deps}(\dagger\Gamma \setminus \mathit{deps}(\Delta))$$

we can see that $\mathcal{C}' \subseteq \mathcal{D}$ and that contracts in $\mathcal{C}'$ are sender-agnostic (by assumption). This means that both conditions of Theorem 4 are satisfied. Now, using Lemma 6 we have:

$$\mathrm{MEV}^{\infty}_{\dagger(\Gamma \mid \Delta) \cap \mathit{deps}(\Delta)}(\Gamma' \mid \Delta, \dagger\Delta) = \mathrm{MEV}^{\infty}_{\dagger(\Gamma \mid \Delta) \cap \mathit{deps}(\Delta)}(\Gamma \mid \Delta, \dagger\Delta) \tag{5.6}$$

Therefore, we can prove (5.1) by using (5.4), (5.5) and (5.6). Observe:

$$\begin{array}{ccl}
\mathrm{MEV}^{\infty}_{\dagger(\Gamma \mid \Delta) \cap \mathit{deps}(\Delta)}(\Gamma' \mid \Delta, \dagger\Delta) = & \mathrm{MEV}^{\infty}_{\dagger(\Gamma \mid \Delta) \cap \mathit{deps}(\Delta)}(\Gamma \mid \Delta, \dagger\Delta) & \text{from } (5.6) \\
\| & \| & \text{from } (5.4) \text{ and } (5.5) \\
\mathrm{MEV}^{\infty}(\Gamma' \mid \Delta, \dagger\Delta) & \mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta) &
\end{array}$$

□

A direct consequence of Proposition 6, when the dependencies of Δ are sender-agnostic, is that the adversary attacking Δ gains no additional advantage by deploying contracts before Δ. This result is similar in essence to Corollary 2 in [7].

Corollary 1. *Given $\mathit{deps}(\Delta) \cap \dagger\Gamma_{\varepsilon} = \emptyset$, $p^{\infty}(\Gamma, \Delta) = p^{\infty}(\Gamma \mid \Gamma_{\mathcal{M}}, \Delta)$ when the contracts in $\mathit{deps}(\Delta)$ are sender-agnostic.*

Proof. The proof follows directly by replacing Γ_{ε} with $\Gamma_{\mathcal{M}}$. □

Proposition 7. *Given $\mathit{deps}(\Gamma \mid \Delta) \cap \dagger\Gamma_{\varepsilon} = \emptyset$,*

$$p^{\infty}(\Gamma, \Delta) = p^{\infty}(\Gamma_{\varepsilon} \mid \Gamma, \Delta)$$

when contracts in $\mathit{deps}(\Delta) \cap \mathit{deps}(\dagger(\Gamma_{\varepsilon} \mid \Gamma) \setminus \mathit{deps}(\Delta))$ are sender-agnostic.

Proof. Firstly, we note that without the condition $\mathit{deps}(\Gamma \mid \Delta) \cap \dagger\Gamma_{\varepsilon} = \emptyset$, the state $\Gamma \mid \Delta$ is not well-formed and the thesis does not make sense. The proof remains exactly the same as for Proposition 6. □

Remark 3. *Given that $\mathrm{MEV}^{\infty}(\Gamma \mid \Delta, \dagger\Delta)$ is determined solely by $\mathit{deps}(\Delta)$, it follows that $p^{\infty}(\Gamma, \Delta)$ is computable based exclusively on $\mathit{deps}(\Delta)$. Formally, if the conditions* *specified in Theorem 4 hold, we can rewrite* $p^\infty(\Gamma, \Delta)$ *as follows:*

$$p^\infty(\Gamma, \Delta) = \frac{\mathrm{MEV}^\infty_{\dagger(\Gamma|\Delta)\cap deps(\Delta)}(\Gamma \mid \Delta, \dagger\Delta) - \mathrm{MEV}^\infty_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta)}{\$_{\dagger\Delta}(\Delta)} \qquad \text{if } \$_{\dagger\Delta}(\Delta) \neq 0$$

and $p^\infty(\Gamma, \Delta) = 0$ *otherwise.*

5.3 Comparing existing perspectives on MEV Interference

There are a few differences between Definition 11 and Definition 12 that are to be noted. Firstly, in general, assuming particular fixed contract states Γ and Δ, taking the maximum of $p(\Gamma, \Delta)$ over all wallet states does not yield $p^\infty(\Gamma, \Delta)$, i.e.

Remark 4. *For given contract states* Γ *and* Δ*,*

$$p^\infty(\Gamma, \Delta) \neq \max_W \; p(W \mid \Gamma, \Delta)$$

This is because while Definition 11 captures both token and contract dependencies between Γ and Δ, Definition 12 only captures contract dependencies and is agnostic to token dependencies between S and Δ. That is, in the case that there are token-dependencies between S and Δ being leveraged by the adversary, Definition 12 would be unable to capture that. And hence, the value obtained in the RHS is larger than the one from the LHS in general.

Example 7. To demonstrate MEV interference which leverages token dependencies and hence is not captured by Definition 12, consider `Airdrop` and `Exchange` in Figure 7.1, and let:

$$S = W \mid \Gamma = \mathtt{M}[0{:}\,\mathtt{T}] \mid \mathtt{Airdrop}[1{:}\,\mathtt{T}, \mathtt{tout} = \mathtt{T}]$$
$$\Delta = \mathtt{Exchange}[100{:}\,\mathtt{ETH}, \mathtt{tin} = \mathtt{T}, \mathtt{tout} = \mathtt{ETH}, \mathtt{rate} = 10, \mathtt{owner} = \mathtt{B}]$$

Here, $\$_{\dagger\Delta}(\Delta) = \$_{\{\mathtt{Exchange}\}}(\Delta) = 100$. Let us calculate the interference values for this case using both definitions. To calculate $p(S, \Delta)$, observe that the unrestricted MEV of `Exchange` is $10 \cdot \$1_{\mathtt{ETH}}$, since `M` can first extract $1{:}\,\mathtt{T}$ from `Airdrop`, and then use Exchange, draining 10: ETH. Instead, its restricted MEV is zero, since M cannot obtain the needed 1: T.

By Definition 3, we have that:

$$\begin{aligned}
\mathrm{MEV}(S \mid \Delta, \dagger\Delta) &= \mathrm{MEV}_{\{\texttt{Airdrop},\texttt{Exchange}\}}(S \mid \Delta, \{\texttt{Exchange}\}) = 10 \\
\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) &= \mathrm{MEV}_{\{\texttt{Exchange}\}}(S \mid \Delta, \{\texttt{Exchange}\}) = 0
\end{aligned}$$

Therefore, by Definition 11 we have:

$$p(S, \Delta) = \frac{10 - 0}{100} = 0.1 \ \leq \ \max_{W} p(S, \Delta)$$

To calculate p^∞, by Definition 6, we have that:

$$\mathrm{MEV}^\infty(\Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}^\infty_{\dagger\Delta}(\Gamma \mid \Delta, \dagger\Delta) = 10$$

Therefore, by Definition 12 we have that:

$$p^\infty(\Gamma, \Delta) \ = \ 0 \ < \ \max_{W} p(S, \Delta) \qquad \diamond$$

Remark 5. *Let $W_{\mathcal{M}}$ be the threshold adversary wallet given by Lemma 3. Then, the interference values yielded by Definition 11 and Definition 12 match, i.e.:*

$$p^\infty(\Gamma, \Delta) = p(W_{\mathcal{M}} \mid \Gamma, \Delta)$$

Part II

Contributions

Chapter 6

Quantitative MEV Interference

In this chapter, we arrive on a final definition for quantitative MEV interference: $\mathcal{I}(S \rightsquigarrow \Delta)$. Our new notion satisfies properties that one would consider desirable for a quantitative notion of composability to provide: namely, (i) When Δ has zero wealth, then $\mathcal{I}(S \rightsquigarrow \Delta)$ is zero. (ii) $\mathcal{I}(S \rightsquigarrow \Delta)$ is zero when the contract dependencies and the token dependencies of Δ in S are irrelevant to the ability of inflicting a loss to Δ. (iii) $\mathcal{I}(S \rightsquigarrow \Delta)$ does not decrease when we extend S with contracts that are not dependencies of Δ. (iv) $\mathcal{I}(S \rightsquigarrow \Delta)$ is independent of the users' wallets in S, except for those belonging to adversaries. (v) $\mathcal{I}(S \rightsquigarrow \Delta)$ has a maximum, corresponding to the case where the economic loss that can be inflicted to Δ is purely due to the interactions of the adversary with S.

Definition 13 (Quantitative MEV interference)**.** For a blockchain state $S = W \mid \Gamma$ and a contract state Δ, we quantify the MEV interference caused by S on Δ as:

$$\mathcal{I}(S \rightsquigarrow \Delta) = 1 - \frac{\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\mathrm{MEV}(S \mid \Delta, \dagger\Delta)} \qquad \textit{if } \mathrm{MEV}(S \mid \Delta, \dagger\Delta) \neq 0$$

When $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = 0$, we define $\mathcal{I}(S \rightsquigarrow \Delta) = 0$.

Proposition 8 ($\mathcal{I}$ ***vs.*** $\not\rightsquigarrow$)**.** $\mathcal{I}(S \rightsquigarrow \Delta) = 0 \iff S \not\rightsquigarrow \Delta$

Proof. If $\mathcal{I}(S \rightsquigarrow \Delta) = 0$, then it must be $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)$ or $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = 0$. In the first case, by Definition 13 we directly have the thesis. In the second case, the thesis directly follows by Definition 7. To prove the other implication, if $S \not\rightsquigarrow \Delta$ then by Definition 7, it must be $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)$. The thesis directly follows from Definition 13. □

6.1 Theoretical Properties

We now study the theoretical properties of $\mathcal{I}(S \rightsquigarrow \Delta)$.

Lemma 9 (Basic properties of $\mathcal{I}$). *For all S, Δ:*

1. $\mathcal{I}(S \rightsquigarrow \emptyset) = 0$
2. $\mathcal{I}(W \mid \emptyset \rightsquigarrow \Delta) = 0$
3. $0 \leq \mathcal{I}(S \rightsquigarrow \Delta) \leq 1$

Proof. For Item 1, by Item 1 of Lemma 1 we have $\mathrm{MEV}(S \mid \emptyset, \dagger\emptyset) = 0$. The thesis follows by Definition 13.

For Item 2, by Item 4 of Lemma 1 we have:

$$\mathrm{MEV}(W \mid \emptyset \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(W \mid \emptyset \mid \Delta, \dagger\Delta)$$

which gives us $\mathcal{I}(W \mid \emptyset \rightsquigarrow \Delta) = 0$, and hence we have our thesis.

For Item 3, there are two cases. If $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = 0$, then $\mathcal{I}(S \rightsquigarrow \Delta) = 0$ holds by definition. Otherwise, by Items 2 and 5 of Lemma 1:

$$\begin{aligned}
& 0 \leq \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) \leq \mathrm{MEV}(S \mid \Delta, \dagger\Delta) \\
\Longrightarrow\ & 0 \leq \frac{\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\mathrm{MEV}(S \mid \Delta, \dagger\Delta)} \leq 1 \\
\Longrightarrow\ & 0 \leq 1 - \frac{\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\mathrm{MEV}(S \mid \Delta, \dagger\Delta)} \leq 1
\end{aligned}$$

which implies $0 \leq \mathcal{I}(S \rightsquigarrow \Delta) \leq 1$, giving us our thesis. □

Lemma 10. *If $\$_{\dagger\Delta}(\Delta) = 0$, then $\mathcal{I}(S \rightsquigarrow \Delta) = 0$.*

Proof. From Items 2 and 5 of Lemma 1, we have:

$$0 \leq \mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) \leq \mathrm{MEV}(S \mid \Delta, \dagger\Delta) \leq \$_{\dagger\Delta}(\Delta)$$

By hypothesis, $\$_{\dagger\Delta}(\Delta) = 0$. So, by the inequalities above, $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = 0$. Definition 13 gives the thesis. □

Theorem 5. *When deps$(\Delta) \cap \dagger\Gamma = \emptyset$: $\mathcal{I}(S \rightsquigarrow \Delta) \leq \mathcal{I}(S \mid \Gamma \rightsquigarrow \Delta)$*

Proof. By Definition 13, we have two cases.

If $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = 0$, then $\mathcal{I}(S \rightsquigarrow \Delta) = 0$. From Lemma 93, we have $\mathcal{I}(S \mid \Gamma \rightsquigarrow \Delta) \geq 0$. This implies the thesis, $\mathcal{I}(S \rightsquigarrow \Delta) \leq \mathcal{I}(S \mid \Gamma \rightsquigarrow \Delta)$.

Otherwise, assume that $\mathrm{MEV}(S \mid \Delta, \dagger\Delta) > 0$. Then, by Definition 13:

$$\mathcal{I}(S \rightsquigarrow \Delta) = 1 - \frac{\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\mathrm{MEV}(S \mid \Delta, \dagger\Delta)}$$

Now, by Item 3 of Lemma 1, we have that:

$$0 < \mathrm{MEV}(S \mid \Delta, \dagger\Delta) \leq \mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta)$$

Therefore, by Definition 13:

$$\mathcal{I}(S \mid \Gamma \rightsquigarrow \Delta) = 1 - \frac{\mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma \mid \Delta, \dagger\Delta)}{\mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta)}$$

From Lemma 1, we have that:

$$\begin{aligned} \mathrm{MEV}_{\dagger(S|\Delta)}(S \mid \Delta, \dagger\Delta) &\leq \mathrm{MEV}_{\dagger(S|\Delta)}(S \mid \Gamma \mid \Delta, \dagger\Delta) && \text{by Item 3} \\ &\leq \mathrm{MEV}_{\dagger(S|\Gamma|\Delta)}(S \mid \Gamma \mid \Delta, \dagger\Delta) && \text{by Item 2} \end{aligned} \tag{6.1}$$

We know from (6.1),

$$\mathrm{MEV}(S \mid \Delta, \dagger\Delta) \leq \mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta)$$

Taking the reciprocal on both sides gives us

$$\frac{1}{\mathrm{MEV}(S \mid \Delta, \dagger\Delta)} \geq \frac{1}{\mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta)}$$

By Lemma 5, we have $\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma \mid \Delta, \dagger\Delta)$. Then:

$$\frac{\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\mathrm{MEV}(S \mid \Delta, \dagger\Delta)} \geq \frac{\mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma \mid \Delta, \dagger\Delta)}{\mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta)}$$

which finally gives us

$$1 - \frac{\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta)}{\mathrm{MEV}(S \mid \Delta, \dagger\Delta)} \leq 1 - \frac{\mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma \mid \Delta, \dagger\Delta)}{\mathrm{MEV}(S \mid \Gamma \mid \Delta, \dagger\Delta)}$$

which gives our thesis, i.e. $\mathcal{I}(S \rightsquigarrow \Delta) \leq \mathcal{I}(S \mid \Gamma \rightsquigarrow \Delta)$. □

Theorem 6. *If* $\operatorname{dom} W_{\mathcal{M}} = \mathcal{M}$*, then* $\mathcal{I}(W_{\mathcal{M}} \mid W \mid \Gamma \rightsquigarrow \Delta) = \mathcal{I}(W_{\mathcal{M}} \mid \Gamma \rightsquigarrow \Delta)$*.*

Proof. From Item 1 of Lemma 2, we have:

$$\operatorname{dom} W_{\mathcal{M}} = \mathcal{M} \implies \mathrm{MEV}_{\mathcal{D}}(W_{\mathcal{M}} \mid W \mid \Gamma, \mathcal{C}) = \mathrm{MEV}_{\mathcal{D}}(W_{\mathcal{M}} \mid \Gamma, \mathcal{C})$$

This implies:

$$\begin{aligned} &\mathrm{MEV}_{\dagger(\Gamma \mid \Delta)}(W_{\mathcal{M}} \mid W \mid \Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger(\Gamma \mid \Delta)}(W_{\mathcal{M}} \mid \Gamma \mid \Delta, \dagger\Delta), \quad \text{and} \\ &\mathrm{MEV}_{\dagger\Delta}(W_{\mathcal{M}} \mid W \mid \Gamma \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(W_{\mathcal{M}} \mid \Gamma \mid \Delta, \dagger\Delta) \end{aligned}$$

which gives us our thesis, i.e. $\mathcal{I}(W_{\mathcal{M}} \mid W \mid \Gamma \rightsquigarrow \Delta) = \mathcal{I}(W_{\mathcal{M}} \mid \Gamma \rightsquigarrow \Delta)$ □

Theorem 7 provides sufficient conditions under which an adversary $\mathcal{M}$ attacking the newly deployed contracts in Δ gains no advantage by deploying malicious contracts $\dagger\Gamma_{\mathcal{M}}$ before the attack. Essentially, these conditions guarantee that the interference caused to Δ is preserved when the state S is extended with contracts $\Gamma_{\mathcal{M}}$ satisfying specific conditions. Condition (i) requires $\mathit{deps}(\Delta)$ to be sender-agnostic, i.e. its contract methods are unaware of the identity of the `sender`, only being able to use it as a recipient of token transfers. Condition (ii) requires token independence between the (contract) dependencies and the non-dependencies of Δ which could have possibly been exploited by $\mathcal{M}$. Since Definition 13 assumes that states are well-formed, Theorem 7 implicitly assumes that contracts in Δ do not have dependencies in $\Gamma_{\mathcal{M}}$.

Theorem 7. $\mathcal{I}(S \rightsquigarrow \Delta) = \mathcal{I}(S \mid \Gamma_{\mathcal{M}} \rightsquigarrow \Delta)$ *holds if (i)* $\mathit{deps}(\Delta)$ *are sender-agnostic, and (ii)* $\mathit{deps}(\Delta)$ *and* $\mathit{deps}(S \mid \Gamma_{\mathcal{M}}) \setminus \mathit{deps}(\Delta)$ *are token independent in* $S \mid \Gamma_{\mathcal{M}} \mid \Delta$*.*

Proof. We show the following two equalities, which imply the thesis:

$$\mathrm{MEV}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) \tag{6.2}$$

$$\mathrm{MEV}_{\dagger\Delta}(S \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger\Delta}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) \tag{6.3}$$

Observe that (6.3) follows directly from Lemma 5. To prove (6.2), we pass through two auxiliary results. We start by proving the following equality via Theorem 2:

$$\mathrm{MEV}_{\dagger(S \mid \Gamma_{\mathcal{M}} \mid \Delta)}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger S \cap \mathit{deps}(\dagger\Delta)}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) \tag{6.4}$$

In order to apply Theorem 2, let:

$$\mathcal{C} = \dagger\Delta \qquad \mathcal{D} = \dagger(S \mid \Gamma_{\mathcal{M}} \mid \Delta) \qquad \mathcal{C}' = \mathit{deps}(\Delta) \cap \mathit{deps}(\mathcal{D} \setminus \mathit{deps}(\mathcal{C}))$$

Note that the assumptions of Theorem 2 are satisfied:

- assumption (i): $\mathcal{C}'$ are sender-agnostic, by assumption (i);
- assumption (ii): $\mathcal{C}' \subseteq \mathcal{D}$ holds trivially;
- assumption (iii): since the state $S \mid \Delta$ is well-formed by assumption, then $\mathit{deps}(\Delta) \subseteq \dagger(S \mid \Delta)$, and so we have that:

$$\begin{aligned}\mathit{deps}(\mathcal{D}) \cap \mathit{deps}(\mathcal{C}) &= \mathit{deps}(S \mid \Gamma_{\mathcal{M}} \mid \Delta) \cap \mathit{deps}(\Delta) = \mathit{deps}(\Delta)\\ \mathit{deps}(\mathcal{D}) \setminus \mathit{deps}(\mathcal{C}) &= \mathit{deps}(S \mid \Gamma_{\mathcal{M}} \mid \Delta) \setminus \mathit{deps}(\Delta) = \mathit{deps}(S \mid \Gamma_{\mathcal{M}}) \setminus \mathit{deps}(\Delta)\\ &\subseteq \dagger(S \mid \Gamma_{\mathcal{M}}) \setminus \mathit{deps}(\Delta)\end{aligned}$$

  Since $S \mid \Gamma_{\mathcal{M}} \mid \Delta$ is well-formed and $\mathit{deps}(\Delta)$ and $\dagger(S \mid \Gamma_{\mathcal{M}}) \setminus \mathit{deps}(\Delta)$ are disjoint, then **??** (iii) ensures that these sets are token independent.

Therefore, by Theorem 2 it follows that (6.4) holds.

Next, using Theorem 2 again, we prove the equality:

$$\mathrm{MEV}_{\dagger(S \mid \Delta)}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) = \mathrm{MEV}_{\dagger S \cap \mathit{deps}(\dagger\Delta)}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) \tag{6.5}$$

This time, in order to apply Theorem 2 we let:

$$\mathcal{C} = \dagger\Delta \qquad \mathcal{D} = \dagger(S \mid \Delta) \qquad \mathcal{C}' = \mathit{deps}(\Delta) \cap \mathit{deps}(\mathcal{D} \setminus \mathit{deps}(\mathcal{C}))$$

Again, note that the assumptions of Theorem 2 are satisfied:

- assumption (i): $\mathcal{C}'$ are sender-agnostic, by assumption (i);
- assumption (ii): $\mathcal{C}' \subseteq \mathcal{D}$ holds trivially;
- assumption (iii): since the state $S \mid \Delta$ is well-formed by assumption, then $\mathit{deps}(\Delta) \subseteq \dagger(S \mid \Delta)$, and so we have that:

  $$\begin{aligned} \mathit{deps}(\mathcal{D}) \cap \mathit{deps}(\mathcal{C}) &= \mathit{deps}(S \mid \Delta) \cap \mathit{deps}(\Delta) = \mathit{deps}(\Delta) \\ \mathit{deps}(\mathcal{D}) \setminus \mathit{deps}(\mathcal{C}) &= \mathit{deps}(S \mid \Delta) \setminus \mathit{deps}(\Delta) = \mathit{deps}(S) \setminus \mathit{deps}(\Delta) \end{aligned}$$

  Condition (iii) ensures that these sets are token independent.

Therefore, by Theorem 2 it follows that (6.5) holds.

Now we can prove (6.2) by observing the following chain of equalities:

$$\begin{array}{lcll} \mathrm{MEV}_{\dagger(S\mid\Delta)}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) & = & \mathrm{MEV}_{\dagger(S\mid\Gamma_{\mathcal{M}}\mid\Delta)}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) & \text{from (6.4) and (6.5)} \\ \quad\quad \| & & \quad\quad \| & \text{from Lemma 5} \\ \mathrm{MEV}_{\dagger(S\mid\Delta)}(S \mid \Delta, \dagger\Delta) & & \mathrm{MEV}_{\dagger(S\mid\Gamma_{\mathcal{M}}\mid\Delta)}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) & \\ \quad\quad \| & & \quad\quad \| & \text{by Lemma 1(4)} \\ \mathrm{MEV}(S \mid \Delta, \dagger\Delta) & & \mathrm{MEV}(S \mid \Gamma_{\mathcal{M}} \mid \Delta, \dagger\Delta) & \end{array}$$

Now, the thesis directly follows from Equations (6.2) and (6.3). □

Chapter 7

Use Cases

We now apply our notion to analyze paradigmatic smart contract compositions.

7.1 Airdrop/Exchange

Consider an instance of the $\texttt{Exchange}$ contract in Figure 7.1, to be deployed in a blockchain state S containing an instance of the $\texttt{Airdrop}$ contract in Figure 7.1. More specifically, let:

$$\begin{aligned} S &= \texttt{M}[n_{\texttt{M}} \colon \texttt{T}] \mid \texttt{Airdrop}[n_{\texttt{A}} \colon \texttt{T}] \\ \Delta &= \texttt{Exchange}[n_{\texttt{E}} \colon \texttt{ETH}, \texttt{tin} = \texttt{T}, \texttt{tout} = \texttt{ETH}, \texttt{rate} = r, \texttt{owner} = \texttt{A}] \end{aligned}$$

The $\texttt{Exchange}$ contract allows any user to swap tokens of type $\texttt{tin}$ with tokens of type $\texttt{tout}$ (in the instance, $\texttt{T}$ and $\texttt{ETH}$, respectively), at an exchange rate of 1 unit of $\texttt{tin}$ for $\texttt{rate}$ units of $\texttt{tout}$. For simplicity, assume that $\texttt{rate}$ is a floating-point number, and arithmetic operations are floored, and that $\$\mathbf{1}_{\texttt{T}} = \$\mathbf{1}_{\texttt{ETH}} = 1$. We evaluate the MEV interference from S to Δ. When the exchange rate is favourable, i.e. the $\texttt{rate}$ is greater than 1, the adversary $\texttt{M}$ can extract MEV from Δ by exchanging $\texttt{T}$ for $\texttt{ETH}$. This is possible as far as $\texttt{Exchange}$ has enough $\texttt{ETH}$ balance. The MEV can be further increased by draining $n_{\texttt{A}} \colon \texttt{T}$ from $\texttt{Airdrop}$, and swapping these tokens through

```
contract Airdrop {
  constructor(?x:t) { tout=t }      // deposit any token t
  withdraw() { sender!#tout:tout } // any user withdraws
}
contract Exchange {
  constructor(?x:t1,t2,r) {
    require r>0;
    rate=r; tout=t1; tin=t2; owner=origin
  }
  getTokens() {
    return (tin,tout)
  }
  getRate() {
    return rate
  }
  setRate(newRate) {
    require origin==owner;
    rate=newRate
  }
  swap(?x:t) {               // receives x units of tin
    require t==tin && #tout>=x*rate;
    sender!x*rate:tout     // sends x*rate units of tout
  }
}
```

Figure 7.1: An airdrop and an exchange contract.

the Exchange. More precisely, we have:

$$\mathrm{MEV}_{\{\texttt{Exchange}\}}(S \mid \Delta, \{\texttt{Exchange}\}) = \begin{cases} \lfloor n_{\texttt{M}} \cdot r \rfloor & \text{if } \lfloor n_{\texttt{M}} \cdot r \rfloor < n_{\texttt{E}} \\ n_{\texttt{E}} & \text{otherwise} \end{cases}$$

$$\mathrm{MEV}(S \mid \Delta, \{\texttt{Exchange}\}) = \begin{cases} \lfloor (n_{\texttt{M}} + n_{\texttt{A}}) \cdot r \rfloor & \text{if } \lfloor (n_{\texttt{M}} + n_{\texttt{A}}) \cdot r \rfloor < n_{\texttt{E}} \\ n_{\texttt{E}} & \text{otherwise} \end{cases}$$

Therefore, the MEV interference from S on Δ is bounded by:

$$p(S, \Delta) \;\leq\; \begin{cases} {}^{n_{\texttt{A}}}\!/\!_{(n_{\texttt{M}}+n_{\texttt{A}})} & \text{if } \lfloor (n_{\texttt{M}} + n_{\texttt{A}}) \cdot r \rfloor < n_{\texttt{E}} \\ 1 - {}^{n_{\texttt{M}} \cdot r}\!/\!_{n_{\texttt{E}}} & \text{if } \lfloor (n_{\texttt{M}} + n_{\texttt{A}}) \cdot r \rfloor \geq n_{\texttt{E}} > \lfloor n_{\texttt{M}} \cdot r \rfloor \\ 0 & \text{otherwise} \end{cases}$$

When M is sufficiently rich, she can drain the Exchange without invoking the Airdrop. Instead, when M's wealth is limited, she is able to inflict a greater loss of Exchange

```
contract Betv_oracle {
  constructor(?x:ETH,t,d) {
    require t!=ETH && oracle.getTokens()==(ETH,t);
    tok=t; rate=r; owner=origin; deadline=d
  }
  bet(?x:ETH) {
    require player==null && x==#ETH;
    player=origin
  }
  win(v) {
    require block.num<=deadline && origin==player;
    require v>=0 && v<=1;
    if (oracle.getRate(ETH)>=v*rate) then
        player!(v*#ETH):ETH
    else abort
  }
  close() {
    require block.num>deadline && origin==owner;
    owner!#ETH:ETH
  }
}
```

Figure 7.2: Betv sends a variable proportion of the pot to the winner based on the oracle exchange rate.

by leveraging the Airdrop. So, the interference caused to Exchange in this case has a dual dependence on the adversary's and the Airdrop's wealth. Furthermore, the interference is inversely proportional to M's wealth, i.e. richer adversaries have less need to exploit the context, resulting in lower interference from S to Δ. This is coherent with our intuition, since we would expect a poorer adversary to benefit more from exploiting the Airdrop than a richer one. ◇

7.2 AMM/Betv

The Betv contract in Figure 7.2 allows a player to bet on the exchange rate between a token and ETH. It is parameterized over an oracle that is queried for the token price. Betv receives the initial pot from its owner upon deployment, and a player must match this amount to enter the bet. Before the deadline, the player can win a proportion potShare of the total pot if the oracle exchange rate exceeds or equals potShare times the bet rate. The remaining portion is taken by the owner as a fee. Consider an instance of Betv using the Automated Market Maker AMM in Figure 4.1

as a price oracle:

$$\begin{aligned} S &= \mathtt{M}[m\colon \mathtt{ETH}] \mid \mathtt{AMM}[r_0\colon \mathtt{ETH}, r_1\colon \mathtt{T}] \mid \mathtt{block.num} = d-k \mid \cdots \\ \Delta &= \mathtt{Betv}[b\colon \mathtt{ETH}, \mathtt{owner} = \mathtt{A}, \mathtt{tok} = \mathtt{T}, \mathtt{rate} = r, \mathtt{deadline} = d] \end{aligned}$$

When $\mathtt{M}$ is allowed to manipulate the $\mathtt{AMM}$, she can inflate the exchange rate of $\mathtt{ETH}$, provided that she possesses sufficient funds. Formally, if $\mathtt{M}$ swaps $x\colon\mathtt{ETH}$ for $y\colon\mathtt{T}$, then according to the criterion specified in $\mathtt{Bet}.\mathtt{win}()$, the winner receives an amount $\lfloor 2bp \rfloor$ only if $\mathtt{AMM}.\mathtt{getRate}(\mathtt{ETH}) = {}^{r_0+x}/_{r_1-y} \geq p \cdot r$. Assuming that $\mathtt{M}$ enters the bet only when she can choose x sufficiently high to satisfy this condition, and for $p \geq {}^1/_2$ (since a smaller proportion makes the bet irrational for her), she fires the following sequence of transactions: where, in the $\mathtt{swap}$ transaction, $x = m - b \geq 0$ is the number of $\mathtt{ETH}$ units sent to the $\mathtt{AMM}$, $y = \lfloor {}^{xr_1}/_{r_0+x} \rfloor$ is the number of $\mathtt{T}$ units received, and the value that $\mathtt{M}$ bets on is $p = {}^{r_0+x}/_{r(r_1-y)}$:

$$\begin{aligned} S \mid \Delta &\xrightarrow{\mathtt{M}:\mathtt{Bet}.\mathtt{bet}(?\,b:\mathtt{ETH},p)} && \mathtt{AMM}[r_0\colon \mathtt{ETH}, r_1\colon \mathtt{T}] \mid \mathtt{Bet}[2b\colon \mathtt{ETH}, \mathtt{potShare} = p, \cdots] \mid \cdots \\ &\xrightarrow{\mathtt{M}:\mathtt{AMM}.\mathtt{swap}(?\,x:\mathtt{ETH},0)} && \mathtt{AMM}[r_0 + x\colon \mathtt{ETH}, r_1 - y\colon \mathtt{T}] \mid \mathtt{Bet}[2b\colon \mathtt{ETH}, \cdots] \mid \cdots \\ &\xrightarrow{\mathtt{M}:\mathtt{Bet}.\mathtt{win}()} && \mathtt{AMM}[r_0 + x\colon \mathtt{ETH}, r_1 - y\colon \mathtt{T}] \mid \mathtt{Bet}[2b - \lfloor 2bp \rfloor\colon \mathtt{ETH}, \cdots] \mid \cdots \\ &\xrightarrow{\mathtt{M}:\mathtt{AMM}.\mathtt{swap}(?\,y:\mathtt{T},0)} && \mathtt{AMM}[r_0\colon \mathtt{ETH}, r_1\colon \mathtt{T}] \mid \mathtt{Bet}[2b - \lfloor 2bp \rfloor\colon \mathtt{ETH}, \cdots] \mid \cdots \end{aligned}$$

By Equation (3.2) we have:

$$\begin{aligned} \mathrm{MEV}(S \mid \Delta, \{\mathtt{Bet}\}) = b - (2b - \lfloor 2bp \rfloor) \; &= \lfloor 2bp \rfloor - b \; \leq 2bp - b \; = \frac{2b(r_0+x)}{r(r_1-y)} - b \\ &= \frac{2b(r_0+x)}{r\left(r_1 - \left\lfloor \frac{xr_1}{r_0+x} \right\rfloor\right)} - b \; \leq \; \frac{2b(r_0+x)}{r\left(r_1 - \frac{xr_1}{r_0+x}\right)} - b \\ &= \frac{2b(r_0+x)^2}{rr_0r_1} - b = \left(\frac{2(r_0+m-b)^2}{rr_0r_1} - 1\right) b \end{aligned}$$

Whereas, if $\mathtt{M}$ was restricted to interact with $\mathtt{Bet}$ only, there are two cases: if $\mathtt{AMM}.\mathtt{getRate} = {}^{r_0}/_{r_1} \geq p \cdot r$, then $\mathtt{M}$ wins the bet. Otherwise, she loses (and, therefore, $\mathtt{Bet}$ does not suffer an economic loss). Even in this case, $\mathtt{M}$ enters the bet only for

$p \geq 1/2$. Therefore Equation (3.1) gives us:

$$\begin{aligned}
\mathrm{MEV}_{\{\mathtt{Bet}\}}(S \mid \Delta, \{\mathtt{Bet}\}) &= \begin{cases} b - \left(2b - \left\lfloor \frac{2br_0}{rr_1} \right\rfloor\right) & \text{if } \frac{r_0}{rr_1} \geq 1/2 \\ 0 & \text{otherwise} \end{cases} \\
&= \begin{cases} \left\lfloor \frac{2br_0}{rr_1} \right\rfloor - b & \text{if } \frac{r_0}{rr_1} \geq 1/2 \\ 0 & \text{otherwise} \end{cases} \\
&> \begin{cases} \frac{2br_0}{rr_1} - b - 1 & \text{if } \frac{r_0}{rr_1} \geq 1/2 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

Hence MEV interference is estimated through Definition 11 as follows:

$$\begin{aligned}
p(S, \Delta) &< \begin{cases} 1 - \frac{\frac{2br_0}{rr_1} - b - 1}{\left(\frac{2(r_0+m-b)^2}{rr_0r_1} - 1\right)b} & \text{if } \frac{r_0}{rr_1} \geq 1/2 \\ 1 & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 - \frac{(2br_0 - brr_1 - rr_1)r_0}{b(2(r_0+m-b)^2 - rr_0r_1)} & \text{if } \frac{r_0}{rr_1} \geq 1/2 \\ 1 & \text{otherwise} \end{cases} \\
&= \begin{cases} 1 - \frac{2br_0^2 - rr_0r_1(b+1)}{2b(r_0+m-b)^2 - brr_0r_1} & \text{if } \frac{r_0}{rr_1} \geq 1/2 \\ 1 & \text{otherwise} \end{cases}
\end{aligned}$$

We observe maximum interference when M exploits the Betv by manipulating the AMM, which would be impossible by interacting exclusively with Betv. Furthermore, the interference value is proportional to the adversarial wealth, as one would anticipate. By contrast, even if M was fortunate to be draining a portion of the Betv by fair play, she can always increase this loss by manipulating the AMM (provided she owns adequate funds). Note that in the composition between Betv and Exchange, the MEV interference is zero, as the adversary cannot manipulate the exchange rate (unless she is the Exchange owner). ◇

7.3 AMM/LendingPool

The LP contract in Figure 7.3 implements a lending protocol, allowing users to deposit tokens and borrow them only if their *collateralization* is above a certain threshold. The collateralization is the ratio between the value of deposits and that of debits, and is a reflection of the borrowing capacity of a user. LP is parameterized over an oracle that is queried for the token prices. Below we analyse a well-known attack where the underlying oracle is an AMM, which an adversary manipulates to exceed her previously limited borrowing capacity [13, 23, 6, 20, 2]. Consider the following instance where we assume where we assume that the AMM is balanced and that M has not deposited or borrowed tokens from the LP yet:

$$S = \texttt{M}[n\colon \texttt{ETH}] \mid \texttt{AMM}[r\colon \texttt{ETH}, r\colon \texttt{T}] \qquad \Delta = \texttt{LP}[a\colon \texttt{ETH}, b\colon \texttt{T}, \texttt{Cmin} = C_{min}, \cdots]$$

To simplify the computations in this, we assume that LP permits users to trade real-valued amounts of tokens (i.e. $w \in \mathbb{T} \to \mathbb{R}$). Note that our simplified LP implementation only offers two functions, deposit and borrow. Calling deposit does not extract tokens from the LP, so the only action through which M could cause a loss to the LP is borrow.

We start by estimating the unrestricted local MEV, i.e. MEV($S \mid \Delta, \{\texttt{LP}\}$). When M can interact with the AMM, she can maximize the loss caused to LP by maximizing her loan amount, or in other words, by inflating her collateralization ratio. There is only one way to do so: by depositing a portion of her ETH to the LP and by inflating the exchange rate of ETH provided by the AMM. To this purpose, M partitions n into x and $n - x$, where she deposits $n - x\colon \texttt{ETH}$ in the LP and provides $x\colon \texttt{ETH}$ to the AMM in exchange for $y\colon \texttt{T}$. Let us suppose this allows M to borrow $t\colon \texttt{T}$ from the LP, which we assume to have sufficient reserves of T (i.e., $b \geq t$). We first note that according to LP.borrow, M can borrow $t\colon \texttt{T}$ whenever she is over-collateralized, i.e.:

$$\texttt{collateral}(\texttt{M}) = \frac{(n-x)(r+x)^2}{t(r-y)^2} \geq C_{min}$$

which gives us the maximum value of t that M can choose, which is:

$$t = \frac{(n-x)(r+x)^2}{C_{min}(r-y)^2}$$

```
contract LP {
  constructor(Cmin_) { Cmin = Cmin_; } // collateralization
     threshold
  collateral(a) { // return user a's collateralization
    val_minted = 0;
    for c in minted: val_minted += minted[t][a] * AMM.getRate(t);
    val_debts = 0;
    for c in debts:  val_debts  += debt[t][a] * AMM.getRate(t);
    return val_minted / val_debts;
  }
  deposit(?x:t) { // deposit x units of token t in the LP
    minted[t][sender] += x;  // record the deposited units in the
       minted map
  }
  borrow(x, t) { // borrow x units of token t in the LP
    require balance(t)>=x;
    debts[t][sender] += x;   // record the borrowed units in the
       debts map
    require collateral(sender)>=Cmin; // sender is over-
       collateralized
    sender!x:t;
  }
}
```

Figure 7.3: A Lending Pool contract.

To find the value of x which maximizes t, we maximize the function $t(x)$ that gives the loan amount as a function of the deposited amount x, subject to the constraint $0 \leq x \leq n$. Since we assume LP allows trading of real-valued amounts of tokens, we have that $t(x)$ is continuous. Thus, we compute its derivative w.r.t x and set it to 0:

$$\frac{dt(x)}{dx} = \frac{d}{dx}\left(\frac{(n-x)(r+x)^4}{C_{min}r^4}\right) = \frac{4(n-x)(r+x)^3-(r+x)^4}{C_{min}r^4} = 0$$

Since $r + x \neq 0$, we can simplify the above as:

$$4(n-x) = r + x \qquad \Longrightarrow x = \frac{4n-r}{5}$$

subject to the constraint $0 \leq x \leq n$. Therefore, $x = \frac{4n-r}{5}$ maximizes $t(x)$ when $4n \geq r$. Otherwise, the value of x that maximizes $t(x)$ is $x = 0$. In other words, when $4n < r$, M does not need to interact with the AMM to maximize her borrowing capacity.

We can check that $x = \frac{4n-r}{5}$ maximizes $t(x)$ by performing the double derivative test. We compute the double derivative of $t(x)$ w.r.t x, plugging in $x = \frac{4n-r}{5}$, and check if it is < 0.

Accordingly:

$$\begin{aligned}
\frac{d^2t(x)}{dx^2} &= \frac{d}{dx}\left(\frac{4(n-x)(r+x)^3 - (r+x)^4}{C_{min}r^4}\right) \\
&= \frac{12(n-x)(r+x)^2 - 4(r+x^3) - 4(r+x)^3}{C_{min}r^4} \\
&= \frac{12(n-x)(r+x)^2 - 8(r+x)^3}{C_{min}r^4}
\end{aligned}$$

Substituting $4(n-x) = r + x$ we get:

$$\begin{aligned}
\frac{d^2t(x)}{dx^2} &= \frac{3(r+x)^3 - 8(r+x)^3}{C_{min}r^4} \\
&= -\frac{5(r+x)^3}{C_{min}r^4} \\
&< 0
\end{aligned}$$

As a result, $\texttt{M}$ fires the following sequence of transactions with a loan amount $t = {}^{(n-x)(r+x)^2}/_{C_{min}(r-y)^2}$ and the amount received on swap $y = {}^{xr}/_{r+x}$:

$$\begin{aligned}
S \mid \Delta &\xrightarrow{\texttt{M:LP.deposit(M pays }(n-x)\texttt{:ETH)}} \texttt{AMM}[r\colon \texttt{ETH}, r\colon \texttt{T}] \mid \texttt{LP}[a+n-x\colon \texttt{ETH}, b\colon \texttt{T}, \cdots] \mid \cdots \\
&\xrightarrow{\texttt{M:AMM.swap(M pays }x\texttt{:ETH,0)}} \texttt{AMM}[r+x\colon \texttt{ETH}, r-y\colon \texttt{T}] \mid \texttt{LP}[a+n-x\colon \texttt{ETH}, b\colon \texttt{T}, \cdots] \mid \cdots \\
&\xrightarrow{\texttt{M:LP.borrow(}t\texttt{,T)}} \texttt{AMM}[r+x\colon \texttt{ETH}, r-y\colon \texttt{T}] \mid \texttt{LP}[a+n-x\colon \texttt{ETH}, b-t\colon \texttt{T}, \cdots] \mid \cdots \\
&\xrightarrow{\texttt{M:AMM.swap(M pays }y\texttt{:T,0)}} \texttt{AMM}[r\colon \texttt{ETH}, r\colon \texttt{T}] \mid \texttt{LP}[a+n-x\colon \texttt{ETH}, b-t\colon \texttt{T}, \cdots] \mid \cdots
\end{aligned}$$

Assuming $\$\mathbf{1}_{\texttt{ETH}} = 1 = \$\mathbf{1}_{\texttt{T}}$ for simplicity, by **??** we get:

$$\begin{aligned}
\mathrm{MEV}(S \mid \Delta, \{\texttt{LP}\}) = t + x - n \; &= \frac{(n-x)(r+x)^2}{C_{min}(r-y)^2} + x - n \\
&= (n-x)\left(\frac{(r+x)^2}{C_{min}(r - \frac{xr}{r+x})^2} - 1\right) \\
&= (n-x)\left(\frac{(r+x)^4}{r^4 C_{min}} - 1\right) \\
&= \begin{cases} \left(n - \frac{4n-r}{5}\right)\left(\frac{\left(r + \frac{4n-r}{5}\right)^4}{r^4 C_{min}} - 1\right) & \text{if } 4n \geq r \\ n\left(\frac{1}{C_{min}} - 1\right) & \text{otherwise} \end{cases} \\
&= \begin{cases} \left(\frac{n+r}{5}\right)\left(\frac{\left(\frac{4(n+r)}{5}\right)^4}{r^4 C_{min}} - 1\right) & \text{if } 4n \geq r \\ n\left(\frac{1}{C_{min}} - 1\right) & \text{otherwise} \end{cases} \\
&= \begin{cases} \left(\frac{n+r}{5}\right)\left(\frac{1}{C_{min}}\left(\frac{4(n+r)}{5r}\right)^4 - 1\right) & \text{if } 4n \geq r \\ n\left(\frac{1}{C_{min}} - 1\right) & \text{otherwise} \end{cases}
\end{aligned}$$

We note two key aspects of the transaction sequence fired by M: Firstly, the ordering of deposit and the (initial) swap transactions is irrelevant. Hence, they can be interchanged without causing a difference to the loss caused to LP. Secondly, firing the (final) swap, i.e. de-manipulating the AMM only affects the wealth of M and not the LP. Hence, it does not affect the MEV extractable from LP. Nevertheless, we include it in the transaction sequence to reflect the attack execution employed in practice.

We now calculate the restricted local MEV, i.e. $\mathrm{MEV}_{\{\texttt{LP}\}}(S \mid \Delta, \{\texttt{LP}\})$. In this case, the only way M can maximize her borrowing capacity is by depositing her total available capital to the LP. Hence, M deposits n: ETH, being able to borrow t': T if

$$\texttt{collateral}(\texttt{M}) = \frac{n}{t'} \geq C_{min}$$

Thus, the maximum amount that M can borrow equals $t' = {}^{n}/_{C_{min}}$. By Equation (3.1) we have:

$$\text{MEV}_{\{\texttt{LP}\}}(S \mid \Delta, \{\texttt{LP}\}) = \; t' - n \; = \; \frac{n}{C_{min}} - n \; = \; n\left(\frac{1}{C_{min}} - 1\right)$$

Accordingly, MEV interference is estimated through Definition 11 as follows:

$$p(S, \Delta) = \begin{cases} 1 \; - \; \frac{n\left(\frac{1}{C_{min}}-1\right)}{\left(\frac{n+r}{5}\right)\left(\frac{1}{C_{min}}\left(\frac{4(n+r)}{5r}\right)^4-1\right)} & \text{if } 4n \geq r \\ \\ 1 \; - \; \frac{n\left(\frac{1}{C_{min}}-1\right)}{n\left(\frac{1}{C_{min}}-1\right)} & \text{otherwise} \end{cases}$$

$$= \begin{cases} 1 \; - \; \frac{n(1-C_{min})}{C_{min}} \cdot \frac{5}{n+r} \cdot \frac{(5r)^4 C_{min}}{(4(n+r))^4-(5r)^4 C_{min}} & \text{if } 4n \geq r \\ 0 & \text{otherwise} \end{cases}$$

$$= \begin{cases} 1 \; - \; \frac{5^5 r^4 n(1-C_{min})}{(n+r)(4^4(n+r)^4-(5r)^4 C_{min})} & \text{if } 4n \geq r \\ 0 & \text{otherwise} \end{cases}$$

In accordance to our expectations, the interference is indeed proportional to the attack capital n of the adversary. Naturally, adversaries with a higher manipulation capital have an increased borrowing capacity. Moreover, the degree of interference is influenced by the AMM reserves since the profitability of the attack rests on the cost of manipulating and de-manipulating the AMM. $\diamond$

Chapter 8

Conclusions

In this thesis, we make an exploratory study of the aspects that a notion quantifying the security of smart contract compositions should possess. We extend the *qualitative* notion of MEV non-interference proposed in [7] to a *quantitative* one. This notion quantifies the economic loss that an adversary can inflict on a contract by targeting its dependencies. We study its theoretical properties, explaining why those are desirable, and finally apply it to assess the security of some common smart contract compositions. We discuss below a few limitations of our study and further directions of research.

8.1 Limitations

To keep our study simple, we make a few simplifications in our model. A first assumption is that token prices in our model are constant and do not depend on the blockchain state. This simplifying assumption allows local MEV to neglect the parts of the state that could affect token prices. Consequently, the amount of interference is not affected by fluctuations of these prices (while they could depend on the prices provided by DEXes, like in Sections 7.2 and 7.3). A more realistic handling of token prices would require to extend the model with a function that determines the token prices in a given state. Another assumption is that the notion of local MEV in Equation (3.1) assumes the mempool to be empty, i.e. $\kappa(\mathcal{M})$ instead of $\kappa(\mathcal{M}, P)$. This does not allow adversaries to exploit their knowledge of pending users' transactions (the public *mempool*). The rationale underlying this choice made in [7] was that MEV interference should be the basis for a static analysis of smart contracts, where dynamic data such as the mempool transactions are not known. Assuming an over-approximation of users' transactions, we could extend our MEV interference by making the mempool a parameter of local MEV, similarly to what done for MEV in [9].

8.2 Future work

Although a few tools exist for detecting price manipulation attacks in DeFi protocols [27, 18, 26], and others for estimating MEV opportunities [4, 5], none of the existing tools address general economic attacks to smart contract compositions. The technique underlying the detection of price manipulation attacks is *taint analysis*, which aims at identifying potential data flows from low-level to high-level data (in the DeFi setting, flows from to functions that manipulate token prices to functions that transfer tokens). While this technique could be generalised to analyse *qualitative* MEV non-interference, estimating our *quantitative* interference seems to require more advanced techniques. Some inspiration could be drawn from static analysis techniques for information-theoretic interference [10, 19, 17, 3]. We plan to explore this line of research in future work.

Bibliography

[1] Defi Pulse: What is DeFi? understanding Decentralized Finance, 2021. www.defipulse.com/blog/what-is-defi.

[2] Sanidhay Arora, Yingjiu Li, Yebo Feng, and Jiahua Xu. SecPLF: Secure protocols for loanable funds against oracle manipulation attacks. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS)*. ACM, 2024.

[3] Mounir Assaf, David A. Naumann, Julien Signoles, Eric Totel, and Frédéric Tronel. Hypercollecting semantics and its application to static analysis of information flow. In *ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)*, pages 874–887. ACM, 2017.

[4] K. Babel, P. Daian, M. Kelkar, and A. Juels. Clockwork finance: Automated analysis of economic security in smart contracts. In *IEEE Symposium on Security and Privacy*, pages 622–639. IEEE Computer Society, 2023.

[5] Kushal Babel, Mojan Javaheripi, Yan Ji, Mahimna Kelkar, Farinaz Koushanfar, and Ari Juels. Lanturn: Measuring economic security of smart contracts through adaptive learning. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1212–1226. ACM, 2023.

[6] Massimo Bartoletti, James Hsin-yu Chiang, and Alberto Lluch-Lafuente. SoK: Lending Pools in Decentralized Finance. In *Workshop on Trusted Smart Contracts*, volume 12676 of *LNCS*, pages 553–578. Springer, 2021.

[7] Massimo Bartoletti, Riccardo Marchesin, and Roberto Zunino. DeFi composability as MEV non-interference. In *Financial Cryptography*, LNCS. Springer, 2024. To appear. Extended version available as CoRR abs/2309.10781 (2023). `https://doi.org/10.48550/ARXIV.2309.10781`.

[8] Massimo Bartoletti and Roberto Zunino. A theoretical basis for blockchain extractable value. *CoRR*, abs/2302.02154, 2023.

[9] Massimo Bartoletti and Roberto Zunino. A theoretical basis for MEV. In *Financial Cryptography and Data Security*, LNCS. Springer, 2025. To appear.

[10] David Clark, Sebastian Hunt, and Pasquale Malacaria. A static analysis for quantifying information flow in a simple imperative language. *J. Comput. Secur.*, 15(3):321–371, 2007.

[11] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *IEEE Symp. on Security and Privacy*, pages 910–927. IEEE, 2020.

[12] Joseph A. Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society, 1982.

[13] Lewis Gudgeon, Daniel Pérez, Dominik Harz, Benjamin Livshits, and Arthur Gervais. The decentralized financial crisis. In *Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 1–15. IEEE, 2020.

[14] Samia Guesmi, Carla Piazza, and Sabina Rossi. Noninterference analysis for smart contracts: Would you bet on it? In *Distributed Ledger Technology Workshop (DLT)*, volume 3791 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2024.

[15] Stefan Kitzler, Friedhelm Victor, Pietro Saggese, and Bernhard Haslhofer. A systematic investigation of DeFi compositions in Ethereum. In *Financial Cryptography and Data Security Workshops*, volume 13412 of *LNCS*, pages 272–279. Springer, 2022.

[16] Stefan Kitzler, Friedhelm Victor, Pietro Saggese, and Bernhard Haslhofer. Disentangling Decentralized Finance (DeFi) compositions. *ACM Trans. Web*, 17(2):10:1–10:26, 2023.

[17] Vladimir Klebanov. Precise quantitative information flow analysis - a symbolic approach. *Theoretical Computer Science*, 538:124–139, 2014.

[18] Queping Kong, Jiachi Chen, Yanlin Wang, Zigui Jiang, and Zibin Zheng. DeFiTainter: Detecting price manipulation vulnerabilities in DeFi protocols. In *ACM SIGSOFT International Symposium on Software Testing and Analysis*, page 1144–1156, 2023.

[19] Boris Köpf and Andrey Rybalchenko. Automation of quantitative information-flow analysis. In *International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM)*, volume 7938 of *LNCS*, pages 1–28. Springer, 2013.

[20] Torgin Mackinga, Tejaswi Nadahalli, and Roger Wattenhofer. TWAP oracle attacks: Easier done than said? In *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–8. IEEE, 2022.

[21] Bruno Mazorra, Michael Reynolds, and Vanesa Daza. Price of MEV: towards a game theoretical approach to MEV. In *ACM CCS Workshop on Decentralized Finance and Security*, pages 15–22. ACM, 2022.

[22] Satoshi Nakamoto. Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2008.

[23] Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais. Attacking the DeFi ecosystem with Flash Loans for fun and profit. In *Financial Cryptography*, volume 12674 of *LNCS*, pages 3–32. Springer, 2021.

[24] Alejo Salles. On the formalization of MEV, 2021. https://writings.flashbots.net/research/formalization-mev.

[25] Sam M. Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William J. Knottenbelt. SoK: Decentralized finance (DeFi). *CoRR*, abs/2101.08778, 2021.

[26] Ka Wai Wu. Strengthening DeFi security: A static analysis approach to Flash Loan vulnerabilities. *CoRR*, abs/2411.01230, 2025.

[27] Siwei Wu, Dabao Wang, Jianting He, Yajin Zhou, Lei Wu, Xingliang Yuan, Qinming He, and Kui Ren. DeFiRanger: Detecting price manipulation attacks on defi applications. *CoRR*, abs/2104.15068, 2021.

[28] Siqiu Yao, Haobin Ni, Andrew C. Myers, and Ethan Cecchetti. SCIF: A language for compositional smart contract security, 2024.