Proof-of-Stake Mining Games with Perfect Randomness

Matheus V. X. Ferreira[1] and S. Matthew Weinberg*[1]

[1]Computer Science, Princeton University

July 12, 2021

Abstract

Proof-of-Stake blockchains based on a longest-chain consensus protocol are an attractive energy-friendly alternative to the Proof-of-Work paradigm. However, formal barriers to "getting the incentives right" were recently discovered, driven by the desire to use the blockchain itself as a source of pseudorandomness [4].

We consider instead a longest-chain Proof-of-Stake protocol with perfect, trusted, external randomness (e.g. a randomness beacon). We produce two main results.

First, we show that a strategic miner can strictly outperform an honest miner with just 32.8% of the total stake. Note that a miner of this size *cannot* outperform an honest miner in the Proof-of-Work model [22]. This establishes that even with access to a perfect randomness beacon, incentives in Proof-of-Work and Proof-of-Stake longest-chain protocols are fundamentally different.

Second, we prove that a strategic miner cannot outperform an honest miner with 30.8% of the total stake. This means that, while not quite as secure as the Proof-of-Work regime, desirable incentive properties of Proof-of-Work longest-chain protocols can be approximately recovered via Proof-of-Stake with a perfect randomness beacon.

The space of possible strategies in a Proof-of-Stake mining game is *significantly* richer than in a Proof-of-Work game. Our main technical contribution is a characterization of potentially optimal strategies for a strategic miner, and in particular, a proof that the corresponding infinite-state MDP admits an optimal strategy that is positive recurrent.

*Supported by NSF CAREER Award CCF-1942497

Contents

1 Introduction

Blockchains have been a resounding success as a disruptive technology. However, the most successful implementations (including Bitcoin [19] and Ethereum [23]) are built on a concept called proof-of-work. That is, participants in the protocol are selected to update the blockchain proportionally to their computational power. The consensus protocols underlying Bitcoin and Ethereum (and many other proof-of-work cryptocurrencies) have been secure in practice, and robust against strategic manipulation. There is even a theoretical foundation supporting this latter property: honestly following the Bitcoin protocol is a Nash equilibrium in a stylized model when no miner controls more than $\alpha^{\text{PoW}} \approx 0.329$ of the total computational power in the network [22, 16] (we will use the notation α^{model} to denote the supremum α such that whenever no miner is selected to create the next block with probability bigger than α, it is a Nash equilibrium for all miners to follow the longest-chain protocol in the referenced model).[1]

However, one major drawback of proof-of-work blockchains is their massive energy consumption. For example, Bitcoin currently consumes more electricity than all but 26 countries annually.[2] The need for specialized hardware and low-cost electricity/cooling/etc. also leads to concentration of the mining process among the few entities who have access to the necessary technology [1]. One popular emerging alternative is a paradigm termed proof-of-stake, where participants are selected proportionally to their stake in the currency itself.

Proof-of-stake cryptocurrencies do not suffer from this drawback, but raise new technical challenges, especially from the incentives perspective. Indeed, Brown-Cohen et al. [4] identifies several formal barriers to designing incentive compatible longest-chain proof-of-stake cryptocurrencies (that is, proof-of-stake protocols "like Bitcoin"). Their work highlights one key barrier: in existing proof-of-stake protocols, *the blockchain itself serves as a source of pseudorandomness*, whereas in proof-of-work protocols *the pseudorandom selection of participants is completely independent of the blockchain.* Specifically, they pose a stylized model with *No External Randomness* and show that it is *never* a Nash equilibrium for all miners to honestly follow the longest-chain protocol no matter how small they are (that is, $\alpha^{\text{PoSNER}} = 0$).

In this work, we investigate the incentive compatibility of longest-chain proof-of-stake protocols with access to *perfect external randomness, completely independent of the blockchain*, often termed a randomness beacon [21] (for brevity of notation, we'll refer to this model simply as PoS). We provide two main results, which give a fairly complete picture:

- We establish that $\alpha^{\text{PoS}} \leq 0.327 < \alpha^{\text{PoW}}$. That is, *even with access to perfect external randomness, longest-chain proof-of-stake protocols admit richer strategic manipulation than their proof-of-work counterparts.* We do this by designing a new strategic deviation that we term nothing-at-stake selfish mining, and establish that it is strictly more profitable than honest behavior for any miner with $\gtrapprox 0.327$ of the total stake (Theorem 3.4).

- We prove that $\alpha^{\text{PoS}} \gtrapprox 0.308$ (Theorem 6.1). In particular, this means that access to a randomness beacon fundamentally changes longest-chain proof of stake protocols:

[1]Kiayias et al. [16] proves that $\alpha^{\text{PoW}} \gtrapprox 0.308$, and [22] estimates α^{PoW} to high precision as ≈ 0.329.
[2]Source: `https://cbeci.org/`. Accessed 3/25/2021.

without one $\alpha^{\text{PoSNER}} = 0$, and any miner can profit by deviating. With a randomness beacon, the incentives are (quantitatively) almost as good as proof-of-work.

We now provide a high-level overview of our model (a significantly more detailed description of the model appears in Section 2), a brief overview of the key technical highlights, and an overview of related work.

1.1 Brief Overview of Model

Seminal work of Eyal and Sirer poses an elegant abstraction of the Bitcoin protocol (that we call the PoW model) [10]. Specifically, the game proceeds in infinitely many discrete rounds. In each round, a single miner is chosen proportionally to their computational power, and creates a block. Immediately upon creating a block, the miner must choose its contents (including its predecessor in the blockchain). The strategic decisions a miner makes are: a) which predecessor to select when they create a block, and b) when to publish that block to the other miners. The fact that predecessors must be chosen *upon creation* of the block captures that the contents of a block created via proof-of-work are fixed upon creation.

A key concept in Bitcoin is the *longest-chain protocol*. Specifically, the longest chain is the published block with the most ancestors.[3] Each miner's reward is equal to the fraction of blocks they produce in the longest chain (taking a limit as rounds go to ∞). A miner honestly follows the longest-chain protocol if: a) they always select the (current) longest chain as the predecessor of any created node, and b) they publish all created blocks during the round in which it's created. Eyal and Sirer [10] establishes that $\alpha^{\text{PoW}} \leq 1/3$ (previously, it was believed that $\alpha^{\text{PoW}} = 1/2$ as proposed in [19]), and follow-up work further nailed down $\alpha^{\text{PoW}} \approx 0.329$ [22].

Brown-Cohen et al. [4] modify this model to capture proof-of-stake with no external randomness. In their model, a random coin is selected in each round *independently for each block*. That is, for each round, and each block B, an independent random miner is selected who is eligible to create a block with B as a predecessor with probability equals to their fraction of all the coins in the system. This captures that the protocol must use the chain itself as a source of pseudorandomness. Their work establishes that $\alpha^{\text{PoSNER}} = 0$: it is never a Nash equilibrium to honestly follow the longest-chain protocol in their model. This result is entirely driven by the fact that the protocol has no external randomness, and therefore, miners can make non-trivial predictions about future pseudorandomness.

Our model lies between these two, and captures proof-of-stake with perfect external randomness. Specifically, in each round a single coin is chosen to create a block. Thus a single miner is chose to create a block (just as in PoW) with probability proportional to their fraction of the coins in the system. The strategic decisions are now better phrased as: a) when to publish a created block, and b) which predecessor to select *when publishing*. Our model captures the following: perfect external randomness allows the protocol to select a random miner independently of all previous selected miners and all previously published blocks. The distinction to proof-of-work is that it is now computationally tractable to set the contents of the block, including its predecessor, at any point before it is published.

[3]An ancestor is any block that can be reached by following a path of predecessors. Observe that because each block has a single predecessor, there is a single path of predecessors out of any block.

Note that our model does stipulate that the winner of round t can publish a single block with timestamp t. In a proof-of-stake protocol, there is no technical barrier to creating and publishing any number of blocks using the same timestamp (indeed, this is precisely because it is computationally efficient to produce blocks in proof-of-stake). However, it will be immediately obvious to the rest of the network that a miner has deviated from the longest-chain protocol in this specific way, and it will be immediately obvious which miner cheated.[4] A common solution to strongly disincentivize such behavior is a *slashing protocol*: any miner can include pointers to two blocks created using the same timestamp and the cheating miner will be steeply fined. While we will not rigorously model the incentives induced by a slashing protocol, our model implicitly assumes a sufficiently strong disincentive for miners to publish multiple blocks (and capture this in our model by simply hard-coding that miners must publish at most a single block with each timestamp).

To get intuition for the types of protocols our stylized model aims to capture, below is a sample (simplified) protocol to have in mind:[5]

- In order to be eligible to mine, a coin must be frozen for (large) T rounds, along with a deposit equal to (large) L times its value (that is, the coin and its deposit must be owned by the same miner for T rounds in a row).
- After being used for mining, a coin and its deposit must be frozen for T rounds.
- During each round t, the randomness beacon outputs a random number. This is mapped to a random eligible coin, and its owner is the selected miner at round t. The selected miner can create blocks with timestamp t.
- If a miner ever publishes distinct blocks with the same timestamp, any other miner can include pointers to those two blocks in a block of their own. This will cause the deviant miner to lose their entire deposit (if desired, a $1-\varepsilon$ fraction of it can be destroyed, and an ε fraction can be awarded to the altruistic miner).

Importantly, we are claiming neither that randomness beacons exist (either in theory or in practice), nor that slashing protocols that perfectly disincentivize detectable cheating (without affecting any other incentives) exist. Theorem 3.4 shows that *even if* these primitives existed, a longest-chain proof-of-stake protocol assuming them would still be (slightly) more vulnerable to strategic manipulation than a proof-of-work protocol. On the other hand, Theorem 6.1 establishes in some sense a reduction from proof-of-stake protocols that nearly match the incentive guarantees of proof-of-work protocols to the design of randomness beacons and slashing protocols.

[4]Observe that deviations from the longest-chain protocol that select strategic predecessors or publish at strategic times cannot be definitively attributed to a cheating miner, as these deviations have an innocent explanation: latency. That is, perhaps the reason a miner chose the wrong predecessor is because news of the true longest chain had not yet reached them. Alternatively, perhaps the miner tried to publish their block during the correct round, but it only propagated through the network several rounds later due. Like all prior work, we do not rigorously model latency, and stick to the elegant model proposed in [10].

[5]We are not claiming that this protocol is secure in a rich model, nor will we reason formally about properties of the proposed slashing mechanism. We provide this just to give intuition for why our stylized model captures the salient features of a longest-chain protocol with trusted external randomness.

1.2 Brief Technical Overview

Theorem 3.4 ($\alpha^{\text{PoS}} \lessapprox 0.327$) follows by designing our nothing-at-stake selfish mining strategy, and analyzing its expected payoff. While the insights to design our strategy are novel, the analysis is similar to those used in prior work to analyze the payoff of the resulting Markov Decision Process (MDP). We defer to Section 3 a description of our strategy and intuition for why it succeeds.

The proof of Theorem 6.1 is the bulk of our technical work. To start, we observe that our model admits an infinite-state MDP (just as in [22]). However, the space of strategies available to a miner in our setting is *significantly* richer than in the PoW model. We provide several examples demonstrating why counterintuitive behavior (such as orphaning one's own blocks) could a priori be part of an optimal strategy. So our main technical results characterize possible optimal strategies for this infinite-state MDP, culminating in a strong enough characterization to lower bound the optimal payoff for Theorem 6.1 and concluding $\alpha^{\text{PoS}} \geq 0.308$.

1.3 Related Work

The most related work is already overviewed above: Eyal and Sirer [10] provide the PoW model, develop the selfish mining attack, and prove that $\alpha^{\text{PoW}} \leq 1/3$. Sapirshtein et al. [22] estimates $\alpha^{\text{PoW}} \approx 0.329$ by solving the associated MDP to high precision, and Kiayias et al. [16] prove that $\alpha^{\text{PoW}} \gtrapprox 0.308$. Brown-Cohen et al. [4] study a related proof-of-stake model with no external randomness, and show that $\alpha^{\text{PoSNER}} = 0$. Other works also study similar questions in variants of this model (e.g. [5, 20]).

There is a rapidly-growing body of work at the intersection of mechanism design and cryptocurrencies [18, 6, 1, 14, 11]. Some of these works further motivate the consideration of proof-of-stake cryptocurrencies [1], while others motivate the choice to restrict attention to Bitcoin's proportional reward scheme [6, 18], but there is otherwise little overlap between our works.

In practice, implementing a random beacon is a complex task [2, 3, 7] and is outside the scope of this paper. As previously noted, our results can be viewed either as a reduction to designing a randomness beacon (Theorem 6.1), or an impossibility result even under the assumption of a randomness beacon (Theorem 3.4).

Finally, it is worth noting that many existing proof-of-stake protocols fit the longest-chain paradigm [17, 13, 8], while others are fundamentally different [12]. Protocols based on Byzantine consensus are a growing alternative to the longest-chain paradigm, although both paradigms are well-represented in theory and in practice. Byzantine consensus protocols are outside the scope of our analysis.

1.4 Roadmap

Section 2 provides a very detailed description of our model, along with examples to help illustrate its distinction from proof-of-work. Section 3 provides our nothing-at-stake selfish mining, and Theorem 3.4. Sections 5 and 4 narrow the space of optimal strategies through a series of reductions. Section 6 overviews Theorem 6.1. Various helpful examples and all omitted proofs are in the appendix.

2 Model

A mining protocol is a Nash equilibrium if no miner wishes to unilaterally change their strategy provided all miners are following the intended protocol. Thus it suffices to consider a two-player game between Miner 1 and Miner 2. Think of Miner 2 as the "rest of the network", which is honestly executing the longest-chain protocol, and think of Miner 1 as the "potential attacker" that optimizes his strategy provided Miner 2 is honest. Following [10, 16, 22] and subsequent works, the game proceeds in discrete time steps (abstracting away the exponential rate at which blocks are found) which we call *rounds*, and the rounds are indexed by $\mathbb{N}_+$. The state B of the game is a tuple $(\textsc{Tree}(B), \mathcal{U}_1(B), \mathcal{U}_2(B), T_1(B), T_2(B))$ (each of these terms will be explained subsequently).

Rounds. During every round n, a single miner creates a new block, and we denote that miner by $\gamma_n \in \{1, 2\}$. We denote by $\gamma := \langle\gamma_n\rangle_{n\in\mathbb{N}_+}$ the full ordered list of miners for each round. In an execution of the game, each γ_n is drawn i.i.d., and equal to 1 with probability $\alpha < 1/2$. We let $T_i := \{n \mid \gamma_n = i\}$ as the rounds during which Miner i creates a new block. $T_i(B)$ denotes all blocks created by Miner i at state B. We abuse notation and might refer to n as the state at round n (after block n is created and all actions are taken, but before round $n + 1$ starts). That is, $(\textsc{Tree}(n), \mathcal{U}_1(n), \mathcal{U}_2(n), T_1(n), T_2(n))$ is the state at round n.

Blocks. The second basic element is a *block*. Each block has a label in $\mathbb{N}$. Blocks are totally ordered by their labels and we say block s was created before block v if $s < v$. We overload notation and also use n to refer to the block produced in round n. All blocks are initially *unpublished*, and can later become *published* due to actions of the miners. Once a block n is published, it has a pointer to exactly one predecessor n' created earlier (that is $n' < n$) and we write $n \to n'$.

Block Tree. Because all published blocks have a pointer to an earlier block, this induces, at all rounds, a block tree $\textsc{Tree}$. We will also refer to V, E as the nodes and edges in $\textsc{Tree} = (\mathrm{V}, \mathrm{E})$. Here, the nodes are all blocks which have been published. Every node has exactly one outgoing (directed) edge towards its predecessor. Before the game begins, the block tree contains only block 0, which we refer to as the *genesis block* and not created by Miner 1 nor Miner 2. We let $\mathcal{U}_i$ denote the set of blocks which have been created by Miner i, but are not yet published. We refer to

$$B_0 := ((\{0\}, \emptyset), \emptyset, \emptyset, \emptyset, \emptyset) \tag{1}$$

as the initial state before any blocks are created and the block tree contains only the genesis block.

Ancestor Blocks. We say that block a is an *ancestor* of block $b \in V$ if there is a directed path from b to a (so b is an ancestor of itself). We write $A(b)$ to denote the set of all ancestors of b (observe that a block can never gain new ancestors, so this is well-defined without referencing the particular state S, or the round, etc.). We use $h(b) := |A(b)| - 1$ as short-hand for the height of block b (the genesis block is the only block with height 0).

Longest Chain. The *longest chain* $\mathcal{C} := \arg\max_{b \in V}\{h(b)\}$ is the leaf in V with the longest path to the genesis block, breaking ties in favor of the first block published, and then in favor of the earliest-indexed block. We use H_i to refer to the block in $v \in A(\mathcal{C})$ with height i and we refer to $A(\mathcal{C})$ as the *longest path.* We say a block q is *forked* (or orphaned) when $q \in A(\mathcal{C})$, but a new block $\mathcal{C}'$ becomes the longest chain and $q \notin A(\mathcal{C}')$.

Successor blocks. We say block a is a *successor* of block $b \in V$ if $a \neq b$ is in the unique path from $\mathcal{C}$ to b. We write $\textsc{Succ}(b)$ to denote the set of successors of b.

Actions. During round n, Miner i knows γ_ℓ for all $\ell \leq n$, and can take the following actions:

1. Wait: wait for the next round, and do nothing.
2. PublishSet(V', E'): publish a set of blocks V' with pointers E'. This adds V' to V, E' to E, and changes all blocks in V' from unpublished to published. To be valid, it must be that:
   - $V' \subseteq \mathcal{U}_i$ (Miner i actually has blocks V' to publish).
   - For all $v \to v' \in E'$, $v \in V'$, $v' \in V \cup V'$ (syntax check for edges in E').
   - For all $v \to v' \in E'$, $v > v'$ (pointers are to earlier blocks).
   - For all $v \in V'$, there is exactly one outgoing edge in E' (every block has exactly one pointer).

Clarifying Order of Operations. At the beginning of round n, there is a block tree $\textsc{Tree} = \textsc{Tree}(n-1)$, and each miner i has a set of unpublished blocks $\mathcal{U}_i = \mathcal{U}_i(n-1)$. Then:

1. γ_n is drawn, and equal to 1 with probability α, and 2 with probability $1-\alpha$. This updates $\mathcal{U}_{\gamma_n} := \mathcal{U}_{\gamma_n}(n-1) \cup \{n\}$. For the other miner, $\mathcal{U}_{3-\gamma_n} := \mathcal{U}_{3-\gamma_n}(n-1)$.
2. Miner 2 takes an action. If that action is PublishSet(V', E'), add the nodes V' and edges E' to $\textsc{Tree}$, and update $\mathcal{U}_2 := \mathcal{U}_2 \setminus V'$.
3. Miner 1 takes an action. If that action is PublishSet(V', E'), add the nodes V' and edges E' to $\textsc{Tree}$, and update $\mathcal{U}_1 := \mathcal{U}_1 \setminus V'$.
4. At this point, round n is over, so $\textsc{Tree}(n) := \textsc{Tree}$, $\mathcal{U}_i(n) := \mathcal{U}_i$, etc.

Predecessor state. For state B, we define $B^{\textsc{Half}}$ as the state prior to B before Miner 1 took their most recent action and after Miner 2 took their most recent action. Similarly, we define $(\textsc{Tree}^{\textsc{Half}}(n), \mathcal{U}_1^{\textsc{Half}}(n), \mathcal{U}_2^{\textsc{Half}}(n), \mathcal{C}^{\textsc{Half}}(n))$ as the subsequent state to $(\textsc{Tree}(n-1), \mathcal{U}_1(n-1), \mathcal{U}_2(n-1), \mathcal{C}(n-1))$ after block n was created, Miner 2 takes their action and before Miner 1 takes their action.

Recall that Miner 2 acts first in every round, so the second tie-breaker in deciding what is the longest chain is only used to distinguish between two blocks of Miner 1 published during the same round, and will never be invoked in a "reasonable" strategy for Miner 1 (see Observation 4.3).[6] Like Kiayias et al. [16], we use FRONTIER to refer to the honest strategy, which never forks and it will be Miner 2's strategy.

Definition 2.1 (Frontier Strategy). *During all rounds n, the* FRONTIER *strategy for miner i does the following:*

- *If $\gamma_n \neq i$, Wait.*
- *If $\gamma_n = i$, PublishSet($\{n\}, \{n \to \mathcal{C}\}$) (publishes the new block pointing to the longest chain).*

Definition 2.2 (Rewards). *For any two states B and B', define* Miner k's reward *as the integer-valued function r^k from state B to B' as the difference between the number of blocks created by Miner k in the longest path at state B' and B. That is,*

$$r^k(B, B') := |A(\mathcal{C}(B')) \cap T_k(B')| - |A(\mathcal{C}(B)) \cap T_k(B)|. \tag{2}$$

Payoffs. As in [10] and follow-up work, miners receive steady-state revenue *proportional to their fraction of blocks in the longest chain.* Recall that in our notation, $\mathcal{C}(n)$ denotes the longest chain after the conclusion of round n, $A(\mathcal{C}(n))$ denotes the ancestors of $\mathcal{C}(n)$, and T_i denotes all blocks created by Miner i. Therefore, we define the payoff to Miner 1, when Miner i uses strategy π_i as:

$$\textsc{Rev}_\gamma^{(n)}(\pi_1, \pi_2) := \frac{|A(\mathcal{C}(n)) \cap T_1|}{h(\mathcal{C}(n))}, \qquad \text{and} \qquad \textsc{Rev}(\pi_1, \pi_2) := \mathbb{E}_\gamma\left[\liminf_{n\to\infty} \textsc{Rev}_\gamma^{(n)}(\pi_1, \pi_2)\right],$$

where the expectation is taken over γ, recalling that each γ_n is i.i.d. and equals to 1 with probability α, and 2 with probability $1-\alpha$.

We will study in particular the payoff of strategies π_1 against FRONTIER. For simplicity of notation later, we will refer to $\mathcal{U} := \mathcal{U}_1$ (because FRONTIER has no unpublished blocks, so $\mathcal{U}_2$ is unnecessary), $\pi := \pi_1$ (because $\pi_2 :=$ FRONTIER). We will also list all states only as ($\textsc{Tree}(n)$,$\mathcal{U}(n)$, $T_1(n)$) (because the other variables can be inferred from these, conditioned on $\pi_2 :=$ FRONTIER). We further define:

$$\textsc{Rev}_\gamma^{(n)}(\pi_1) := \textsc{Rev}_\gamma^{(n)}(\pi_1, \text{FRONTIER}), \quad \textsc{Rev}(\pi_1) := \textsc{Rev}(\pi_1, \text{FRONTIER}). \tag{3}$$

A strategy π^* is optimal if $\textsc{Rev}(\pi^*) = \max_\pi \textsc{Rev}(\pi)$. Thus FRONTIER is a *Nash equilibrium* if FRONTIER is an optimal strategy for Miner 1.

Proof-of-Work vs. Proof-of-Stake. Our model, as stated, captures Proof-of-Stake protocols with perfect/trusted external randomness. Importantly, this means that once a miner knows they created a block during round n, they do not need to decide precisely the contents of that block until they publish it (because there is no computational difficulty to produce a block). In Proof-of-Work and the model of Eyal and Sirer [10], the miner of block n must decide *in round n* the contents of that block (because the contents are locked in as soon

[6]Eyal and Sirer [10] also considers the case where Miner 1 wins a tie-breaking with probability β. In principle, our model can easily accommodate any $\beta \in [0, 1]$, but we focus on the case $\beta = 0$ since it is the most pessimistic for the attacker.

as the miner succeeds in proving work). Crucially, the miner *must* decides the ancestor of block n during time $t = n$ while in our model, the miner decides the ancestor of block n any time $t \geq n$ before block n is published. This is the only difference between the two models. Observe that any Proof-of-Work strategy is also valid in our model: this would just be a strategy which chooses the pointer for block n in round n, and does not change it when publishing later. Example 2.3 helps illustrate this distinction.

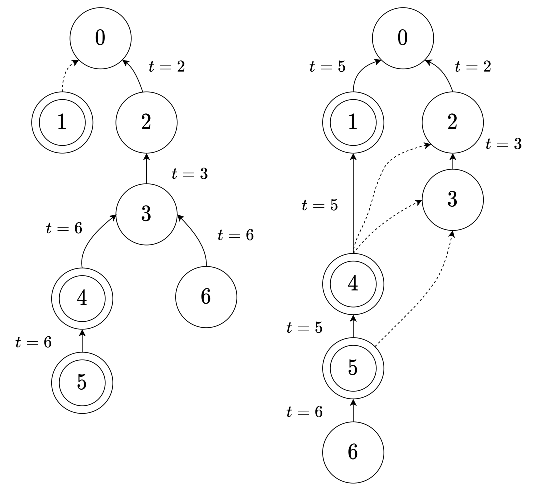

Figure 1: Diagram representing the mining game in Example 2.3. We use double circles for blocks owned by Miner 1 and single circles for blocks owned by Miner 2. The genesis block – i.e., block 0 – is not owned by neither Miner 1 nor Miner 2. The time stamps near the solid edges represent the round where the edge was published, and the number inside the circle represents the round when the block was created. The dashed edges represent some edges that could have been created (edges from any nodes in $\{4, 5\}$ to any nodes in $\{0, 1, 2, 3\}$ are also feasible — i.e., any edge from a later node to an earlier node is feasible).

Example 2.3 (Proof-of-Work vs Proof-of-Stake)**.** *Consider the following 6-round example where Miner 1 creates blocks 1, 4 and 5, and Miner 2 create blocks 2, 3 and 6 as depicted in Figure 1. If Miner 1 was following the Selfish Mining strategy [10] (Definition 3.1), they would decide to create and withhold* $1 \to 0$ *(at time* $t = 1$*). Miner 2 would then publish* $3 \to 2 \to 0$*. At this point, Miner 1 would never attempt to publish* $1 \to 0$ *and we say block 1 becomes permanently orphan. Next, Miner 1 creates and withhold* $4 \to 3$ *(at* $t = 4$*) and* $5 \to 4$ *(at* $t = 5$*). When Miner 2 publishes* $6 \to 3$*, Miner 1 publishes* $5 \to 4 \to 3$ *(at* $t = 6$*),* forking *block* 6 *from the longest path. This nets 2 blocks in the longest path for Miner 1, and 2 for Miner 2.*

*Here is a viable strategy in the Proof-of-Stake mining game: Miner 1 still withholds block 1 (*but does not yet decide where it will point*), and it is still initially orphaned when Miner 2 publishes blocks 2 and 3. When Miner 1 creates block 4, they withhold it (*but does not yet decide where it will point*). When they create block 5, they decide to publish block 4 (*deciding only now to point to block 1*) and block 5 (*deciding only now to point to block 4*). This creates a new longest path. Miner 2 then publishes* $6 \to 5$*. This nets 3 blocks in the longest path for Miner 1, and 1 for Miner 2.*

Importantly, observe that in the proof-of-work model, it would be exceptionally risky for Miner 1 to pre-emptively decide to point block 4 to block 1 at time $t = 4$ *without knowing that they will create block 5 (because maybe Miner 2 creates block 5, and then they would be an additional block behind). But in the proof-of-stake model, Miner 1 can wait to gather* *more information before deciding where to point. In particular, if they happened to instead create block 6 but not 5, they could have published* $6 \to 4 \to 3$. *In proof-of-stake, Miner 1 has the flexibility to make this decision later. In proof-of-work, they have to decide immediately whether to have block 4 pointing to 1 or 3.*

Reminder of Notation. Table 1 in Appendix H is a reminder of our notation.

2.1 Payoff as Fractional of Blocks in the Longest Path

Eyal and Sirer [10] motivates the use of the fraction of blocks as a miner's utility due to the difficulty adjustment in Bitcoin's PoW protocol: Bitcoin adjusts PoW difficult so that, on expectation, miners create one block every 10 minutes and the creator of each block receives new Bitcoins as block reward. Thus a miner maximizes their expected number of blocks in the longest path up to time T by maximizing their expected fraction of blocks in the longest path up to time T.

For PoS, a random beacon outputs a random string at a fixed rate, independent of the blockchain state.[7] Although difficult adjustment is absent in proof-of-stake, the probability of a miner creating the next block is proportional to α, their fraction of coins in the system. Although α is approximately constant over short time horizons, over long time horizons, α will depend on the fraction of block rewards Miner 1 collects. Thus, in the *long-term*, Miner 1 will maximize block rewards by maximizing their fraction of blocks in the longest path.

In Figure 2, we simulate the fraction of coins owned by Miner 1 overtime when Miner 1 follows FRONTIER or the Nothing-at-Stake Selfish Mining (NSM in Definition 3.3) and Miner 2 follows FRONTIER. From the simulation, we observe NSM allows Miner 1 to add a higher fraction of blocks in the longest path when compared with FRONTIER as long as Miner 1 owns more than 32.8% of the coins. We confirm this empirical result in Theorem 3.4. This allows *Miner 1 to eventually own an arbitrarily large fraction of the coins.* Thus the security of a longest chain proof-of-stake protocol depend on a formal guarantees that *no strategy is more profitable than* FRONTIER when a profit maximizing miner is maximizing their fraction of blocks in the longest path. We accomplish this task in Theorem 6.1.

2.2 Capitulating a State

For some states of the game, Miner 1 might follow a strategy that will never fork some blocks from the block tree. Then, it is safe to say that Miner 1 deletes those blocks from the state (or treats the one with highest height as the new genesis block) and consider a trimmed version of the state variable. As example, define $B_{0,1}$ where Miner 2 creates and publishes block 1. Thus

$$B_{0,1} := ((\{0,1\}, \{1 \to 0\}), \emptyset, \emptyset). \tag{4}$$

If Miner 1 never forks block 1, then it is safe to say that in the view of Miner 1 state $B_{0,1}$ is equivalent to state B_0 (after treating block 1 as the new genesis block). Then, we say Miner 1 capitulates from state $B_{0,1}$ to B_0. Since Miner 1 can induce the mining game to return to

[7]The National Institute of Standards and Technology (NIST) random beacon outputs 512 bits every 60 seconds [15].

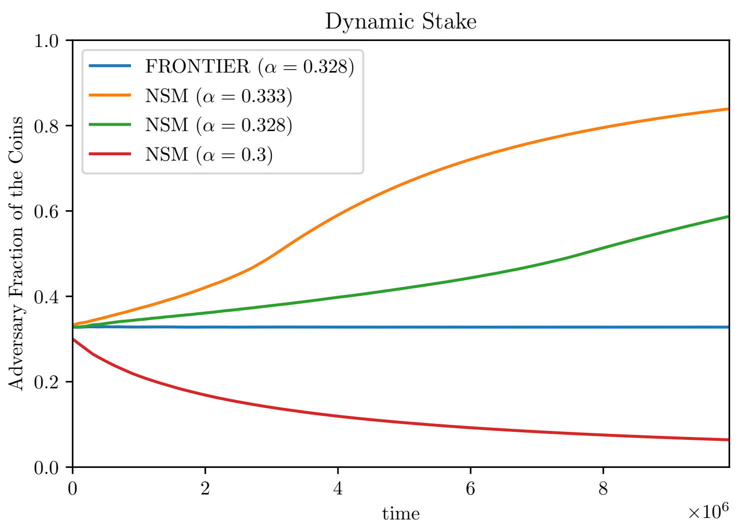

Figure 2: Simulation of Miner 1's dynamic stake over one million rounds when Miner 1 either follow FRONTIER or the Nothing-at-Stake Selfish Mining (NSM) strategy for distinct initial values of α. As initial condition, the system has 100 thousand coins with Miner 1 initialing owning α fraction of the coins. We observe when following FRONTIER, Miner 1 cannot increase their fraction of the stake. We also observe NSM allows Miner 1 to increase their fraction of the stake when they initially own 32.8% of the coins, but NSM is not profitable when Miner 1 owns less than 32.77%, as we find in Theorem 3.4.

prior states, it is convenient to think of Miner 1 optimizing an underlying Markov Decision Process. Next, we provide a definition and formalize the payoff of the MDP in Appendix C.

Markov Decision Process. A *Markov Decision Process* (MDP) for the mining game where Miner 1 follows strategy π and Miner 2 follows FRONTIER is a sequence $(X_t)_{t\geq 0}$ where X_t is a random variable representing the state by the end of round t and before any actions have been take in round $t+1$. Unless otherwise stated, we initialize $X_0 = B_0$ (Equation 1). The game transitions from state X_t to X_{t+1} once the next block is created followed by Miner 2 taking their action followed by Miner 1 taking their action.

For a mining game $(X_t)_{t\geq 0}$ that starts at state $X_0 = B_0$, let

$$\tau := \min\{t \geq 1 : \text{State } X_t \text{ is equivalent to state } B_0 \text{ in the view of Miner 1}\} \tag{5}$$

be the first time step Miner 1 capitulates to state B_0. Similary, let τ' be the second time step Miner 1 capitulates to state B_0. Then, the sequences of rewards

$$r^k(X_0, X_1), r^k(X_1, X_2), \dots, r^k(X_{\tau-1}, X_\tau) \qquad r^k(X_\tau, X_{\tau+1}), r^k(X_{\tau+1}, X_{\tau+2}), \dots, X_{\tau'-1}, X_{\tau'})$$

are independent and identically distributed for $k = 1, 2$. One fundamental question is to understand if $\mathbb{E}[\tau] < \infty$ when Miner 1 is following an optimal strategy (that is, does Miner 1 capitulate to state B_0 at some point with probability 1?). In the proof-of-stake setting, this is not obviously true, and Example 2.3 gives some intuition why: while a proof-of-work Selfish Miner capitulates to state B_0 at round 3 (allowing Miner 2 to keep block 2 in the longest path), a Proof-of-Stake miner might prefer to wait for an opportunity to use block

1, and it is not a priori clear at what point it is safe to conclude that any optimal strategy would have given up on block 1 by now.

2.3 Recurrence

Definition 2.4 (Recurrence)**.** *Consider a mining game starting at state $X_0 = B_0$ where Miner 1 follows strategy π. Let E be the event Miner 1 capitulates to state B_0 at some time $\tau < \infty$. We say π is*

- *Transient if $Pr[E] < 1$.*
- *Recurrent if $Pr[E] = 1$.*
- *Null recurrent if it is recurrent and $\mathbb{E}[\tau] = \infty$.*
- *Positive recurrent if it is recurrent and $\mathbb{E}[\tau] < \infty$.*

Observe Miner 1 never forks the longest chain when using FRONTIER. Thus Miner 1 capitulates to state B_0 at every time step and $\tau = 1$.

Observation 2.5. FRONTIER *is positive recurrent.*

For Proof-of-Work mining games, Kiayias et al. [16] and Sapirshtein et al. [22] assumes Miner 1 will always follow a positive recurrent strategy. To motivate this technical assumption, they assume Miner 1 will never fork a block published by himself, which is sensible because Miner 1 can only mine on a single branch of the blockchain. This is not the case for Proof-of-Stake blockchains, and it is not a priori clear that Miner 1 will never fork a block that they created themselves. To see why this may occur, consider the following example.

First, define $B_{k,0}$, for $k \geq 0$, as the state where Miner 1 creates and withhold blocks $\{1, 2, \ldots, k\} = [k]$. Thus

$$B_{k,0} := ((\{0\}, \emptyset), [k], [k]). \tag{6}$$

Then define $B_{1,1}$ as the state after $B_{1,0}$ where Miner 2 creates block 2 and publishes $2 \to 0$. Thus

$$B_{1,1} := ((\{0, 2\}, \{2 \to 0\}), \{1\}, \{1\}). \tag{7}$$

Example 2.6. *Consider a game at state $B_{1,1}$. After round 2, Miner 2 creates blocks $3, 4, \ldots, 9, 10, 12$ and publishes $12 \to 10 \to 9 \ldots \to 4 \to 3 \to 2$ and Miner 1 creates and withholds blocks $11, 13, 14, \ldots, 24$. At time step 13, Miner 1 publishes $13 \to 11 \to 10$ (this follows the classical selfish mining strategy: it gives up on block 1, but publishes $11 \to 10$ and $13 \to 11$ to fork block 12). This is reasonable, because it is unlikely that Miner 1 can add block 1 to longest path and Miner 1 risks losing blocks 11 and 13 if Miner 2 creates and publishes block 14. However, in the event Miner 1 is lucky and creates blocks $14, 15, \ldots, 24$, Miner 1 can fork all blocks from the current longest path (including his own blocks 11 and 13), resulting in a new longest path with blocks $1, 14, 15, \ldots, 24$ (consisting entirely of Miner 1's blocks). Indeed, upon creating block 14, Miner 1 need not immediately decide whether to make its predecessor 13, or whether to wait and see if they get an extremely lucky run to override the entire chain.*

Note that we are not claiming that this is the optimal decision for Miner 1 from this state, or even that an optimal strategy may ever find itself in this state.[8] *However, this example helps demonstrate that a significantly richer space of strategies are potentially optimal in our model, as compared to proof-of-work.*

This example shows that we must be careful to not exclude optimal strategies when claiming any restrictions on strategies considered by Miner 1. We will eventually address this by introducing the notion of a *checkpoint*, Section 5.1, and prove that there are *some* conditions that allow us to claim that any optimal strategy for Miner 1 will not fork a checkpoint (however, we do not prove that Miner 1 will never consider forking their own blocks).

3 Enhancing Selfish Mining with Nothing-at-Stake

In this section, we show explicitly an strategy that outperforms FRONTIER even when $\alpha = 0.3277$. From Sapirshtein et al. [22], FRONTIER is optimal for Proof-of-Work mining games when Miner 1 has mining power $\alpha \leq 0.329$ (i.e., $\alpha^{\text{PoW}} \approx 0.329$). Thus our strategy witnesses that Proof-of-Stake mining games admits strategies that are more profitable than any strategy in a Proof-of-Work mining game – that is, will establish $\alpha^{\text{PoS}} < \alpha^{\text{PoW}}$.

Our strategy will be a subtle modification from the Selfish Mining strategy of Eyal and Sirer [10] that leverages the Nothing-at-Stake vulnerability in Proof-of-Stake blockchains.[9]

Let's first define the states of interest for our strategy. Recall B_0 is the state where the block tree contains only the genesis block; $B_{1,0}$ is the state after Miner 1 creates and withholds block 1 (Equation 6); $B_{2,0}$ is the state after Miner 1 creates and withholds blocks 1 and 2 (Equation 6); $B_{0,1}$ is the state after Miner 2 publishes $1 \to 0$ (Equation 4); $B_{1,1}$ is the state after $B_{1,0}$ if Miner 2 publishes $2 \to 0$ (Equation 7). Additionally, define the following states.

- $B_{1,2}$ is the state after $B_{1,1}$ if Miner 2 creates block 3 and publishes $3 \to 2$:

$$B_{1,2} := ((\{0,2,3\}, \{3 \to 2 \to 0\}), \{1\}, \{1\}). \tag{8}$$

- $B_{2,2}$ is the state after $B_{1,2}$ if Miner 1 creates and withhold block 4:

$$B_{2,2} := ((\{0,2,3\}, \{3 \to 2 \to 0\}), \{1,4\}, \{1,4\}). \tag{9}$$

These and other relevant states are depicted in Figure 3.

[8] For example, if this were the optimal decision from this state, it would likely be because α is small, and Miner 1 should just take whatever opportunities they have to publish blocks. However, if α is small, that may mean it is better for Miner 1 to just be honest, and they would never find themselves in this situation. The point is that some quantitative comparison is necessary in order to determine whether the optimal strategy for Miner 1 would ever take this action.

[9] The nothing-at-stake vulnerability refers to the fact the algorithm for which a miner can verify a block validity is computationally efficient. Thus miners have no cost to choosing the block content (including its ancestor) at the moment the block is about to be published .

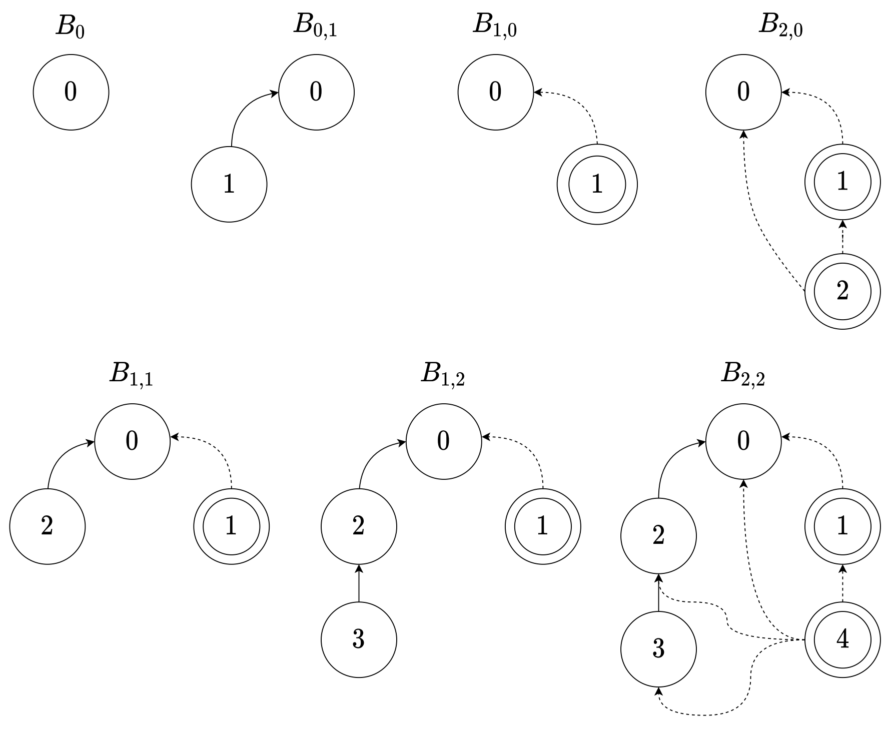

Figure 3: Diagram representing states B_0, $B_{0,1}$, $B_{0,2}$, $B_{1,0}$, $B_{1,1}$, $B_{1,2}$ and $B_{2,2}$. Block 0 has no owner. All double circles are Miner 1's hidden blocks. All circles are Miner 2's published blocks. Dashed lines are edges Miner 1 can publish.

Definition 3.1 (Selfish Mining [10]). *Let $(X_t)_{t\geq 0}$ be a mining game starting at state $X_0 = B_0$. Miner 1 uses the* Selfish Mining (SM) *strategy, Figure 4, which takes the following actions:*

- *Wait at states B_0 and $B_{1,0}$.*
- *At state $B_{0,1}$, capitulate to state B_0.*
- *If $X_2 = B_{1,1}$ and Miner 1 creates block 3, publishes $3 \to 1 \to 0$, then capitulates to state B_0.*
- *If $X_2 = B_{1,1}$ and Miner 2 creates block 3 and publishes $3 \to 2$, then Miner 1 capitulates to state B_0.*
- *If $X_2 = B_{2,0}$, Miner 1 plays Wait until the first time step $\tau \geq 3$ where $|T_2(X_\tau)| = |T_1(X_\tau)| - 1$. At time step τ, Miner 1 publishes all of $\mathcal{U}(X_\tau) = T_1(X_\tau)$ pointing to 0 and forking $T_2(X_\tau)$, then capitulates to state B_0.*

Theorem 3.2 (Equation 8 in [10]). *For the selfish mining strategy*

$$\textsc{Rev}(\text{SM}) = \frac{\alpha^2(4\alpha^2 - 9\alpha + 4)}{\alpha^3 - 2\alpha^2 - \alpha + 1}.$$

Moreover, $\textsc{Rev}(\text{SM}) > \alpha = \textsc{Rev}(\text{FRONTIER})$ *for $\alpha > 1/3$.*

Our Nothing-at-Stake Selfish Mining is similar, with one key difference: In Selfish Mining, Miner 1 capitulates immediately after a loss (specifically, if Miner 2 creates block 3 and publishes $3 \to 2$ from state $B_{1,1}$, Miner 1 immediately accepts that block 1 is now permanently

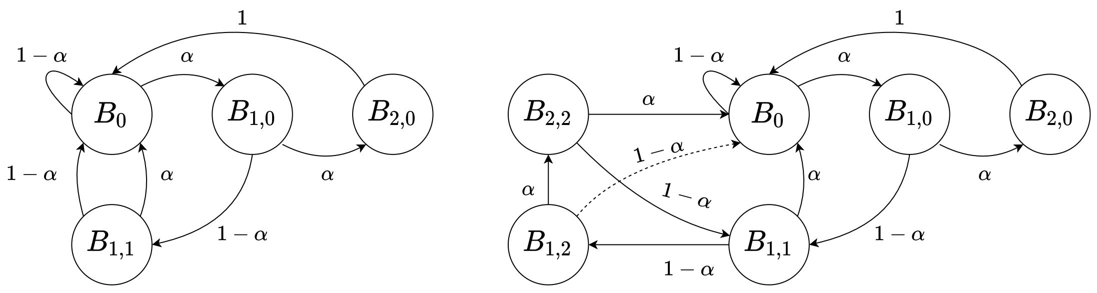

Figure 4: Markov chain representing the Selfish Mining strategy (left) and the Nothing-at-Stake Selfish Mining strategy (right).

orphaned and capitulates to B_0). Nothing-at-Stake Selfish Mining instead remembers this orphaned block, and considers bringing it back later. Importantly, Nothing-at-Stake Selfish Mining can wait to see whether it finds many blocks (in which case it will try to publish the block 1) or not (in which case it will let block 1 remain orphaned) before deciding what to do. Below is a formal description.

Definition 3.3 (Nothing-at-Stake Selfish Mining). *Let $(X_t)_{t\geq 0}$ be a mining game starting at state $X_0 = B_0$. Miner 1 uses the* Nothing-at-Stake Selfish Mining (NSM) *strategy , right of Figure 4, which takes the following actions:*

- *Wait at states B_0, $B_{1,0}$ and $B_{1,2}$.*
- *At state $B_{0,1}$, capitulate to state B_0.*
- *If $X_t = B_{1,1}$ and Miner 1 creates block 3, publishes $3 \to 1 \to 0$, then capitulates to state B_0.*
- *If $X_t = B_{1,2}$ and Miner 2 creates block 4 and publishes $4 \to 3$, then Miner 1 capitulates to state B_0.*
- *If $X_t = B_{2,2}$ and Miner 1 creates block 5, publishes $5 \to 4 \to 1 \to 0$, then Miner 1 capitulates to state B_0.*
- *If $X_t = B_{2,2}$ and Miner 2 creates block 5 and publishes $5 \to 3$, Miner 1 capitulates to state $B_{1,1}$. That is, Miner 1 allows Miner 2 to walk away with blocks 2 and 3 and forgets about unpublished block 1, but remembers unpublished block 4 in the hope of forking block 5 in the future. The resulting state is equivalent to $B_{1,1}$ since we can relabel block 3 as 0, 4 as 1 and 5 as 2.*
- *If $X_2 = B_{2,0}$, Miner 1 plays Wait until the first time step $\tau \geq 3$ where $|T_2(X_\tau)| = |T_1(X_\tau)| - 1$. At time step τ, Miner 1 publishes all of $\mathcal{U}(X_\tau) = T_1(X_\tau)$ pointing to 0 forking blocks $T_2(X_\tau)$, then Miner 1 capitulates to state B_0.*

Let's quickly understand why this strategy is not possible in the proof-of-work model. Zero in on Miner 1's behavior at $B_{2,2}$. If Miner 1 creates block 5, Miner 1 publishes blocks $1, 4, 5$, and in particular has their block 4 point to block 1. However, if Miner 2 creates block

5, Miner 1 capitulates to state $B_{1,1}$. From here, if Miner 1 creates block 6, they immediately publish 4 and 6, *having block* 4 *point to block* 3.

That is, while using this strategy, Miner 1 does not decide where block 4 will point upon mining it, but only upon publishing it. In a Proof-of-Work blockchain, Miner 1 must commit to the ancestor of block 4 at time step 4, so this strategy cannot be used. Intuitively, a nothing-at-stake selfish miner remembers an orphaned block to see if they might get lucky in the future. Importantly, in the PoS model they can *still* wait to decide whether to try and bring this block into the longest chain *even after finding their next block* (but before deciding where to publish it). This extra power enables not only a slight improvement over standard selfish mining but also a strategy the is strictly better than any other valid strategies for the Proof-of-Work mining game.

Theorem 3.4. *For the nothing-at-stake selfish mining strategy,*

$$\textsc{Rev}(\text{NSM}) = \frac{\alpha^2(3\alpha^7 - 13\alpha^6 + 18\alpha^5 - 4\alpha^4 - 12\alpha^3 + 15\alpha^2 - 12\alpha + 4)}{3\alpha^9 - 17\alpha^8 + 40\alpha^7 - 50\alpha^6 + 36\alpha^5 - 14\alpha^4 + \alpha^3 + \alpha^2 - 2\alpha + 1}.$$

Moreover $\textsc{Rev}(\text{NSM}) > \alpha$ *for* $\alpha > 0.3277$.

Recall Sapirshtein et al. [22] estimates $\alpha^{PoW} \approx 0.329$. Thus Theorem 3.4 implies

$$\alpha^{PoS} < 0.3277 < 0.329 \approx \alpha^{PoW}.$$

Interestingly, Nothing-at-Stake Selfish Mining is not better than Selfish Mining for all α. In Figure 6, by plotting the difference $\textsc{Rev}(\text{NSM}) - \textsc{Rev}(\text{SM})$ as a function of α, we observe Selfish Mining is better than Nothing-at-Stake Selfish Mining for $\alpha > 0.44$.

To get intuition why this happens, consider how SM and NSM differs in the event Miner 1 creates blocks 1, 4, 5 and Miner 2 creates blocks 2 and 3. By the end of the 5-th round, SM is at state $B_{2,0}$ while NSM just moved from state $B_{2,2}$ to B_0. The main intuition is that being at state $B_{2,0}$ is a highly profitable for Miner 1 when α is large. In the proof of Theorem 3.2, Appendix D, we show Miner 1 creates, on expectation, $\frac{\alpha}{1-2\alpha}$ blocks from the moment the game reaches state $B_{2,0}$ until the moment the game first returns to state B_0. Moreover, SM has a bigger probability of being at state $B_{2,0}$ than NSM because SM capitulates to state B_0 once it reaches state $B_{1,2}$ but NSM does not.

4 Trimming the Strategy Space

Analyzing the revenue of *all* possible strategies for Miner 1 is quite unwieldy. Therefore, our first goal is to reduce the space of possible strategies to ones which are simpler to analyze *while guaranteeing that this simpler space still contains an optimal strategy.* We accomplish this through a series of reductions. This section provides a series of three "elementary" reductions which build upon each other. That is, the conclusions in each section should not be surprising, although it is challenging to rigorously prove this (examples throughout Appendix E are used to highlight the challenges). Sections 4.1 through 4.3 provide our three reductions. Section 4.4 provides the main theorem statement of this section: there is an optimal *trimmed* strategy. Many proofs are omitted, and can be found in Appendix E.

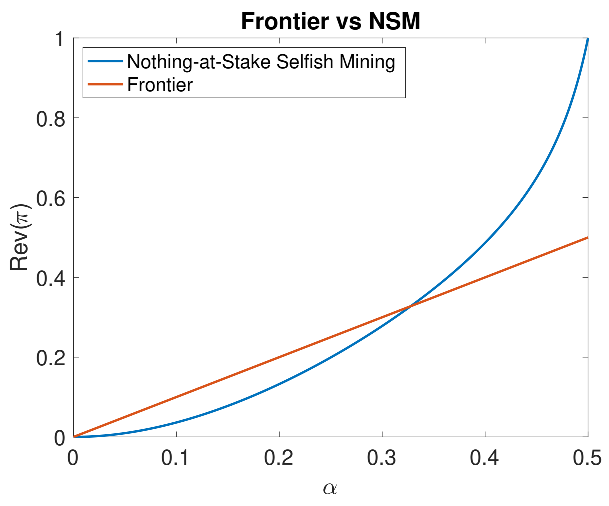

Figure 5: Payoff comparison between FRONTIER and Nothing-at-Stake Selfish Mining.

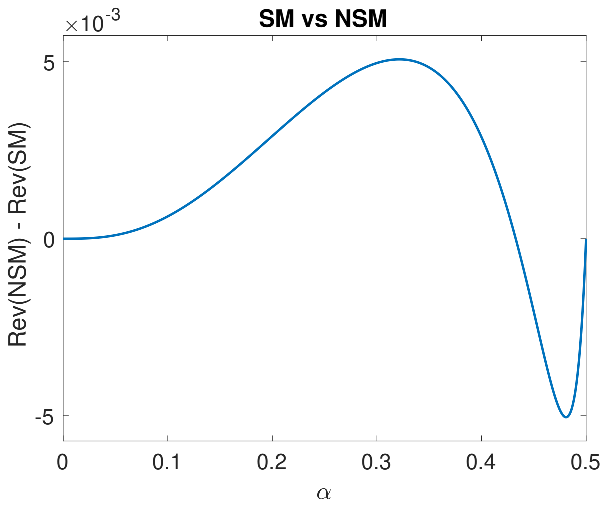

Figure 6: Payoff comparison between Selfish Mining and Nothing-at-Stake Selfish Mining. Observe NSM is slightly better than SM for α close to 1/3, but SM outperforms NSM for $\alpha > 0.44$.

4.1 Step 1: Timeserving

We first show that, w.l.o.g., every strategy only publishes blocks which will be ancestors of the longest chain at the end of that round.

Definition 4.1 (Timeserving)**.** *The action PublishSet(V', E') is* Timeserving *if all blocks in V' immediately enter the longest chain (formally: if the action is taken during round n, then $V' \subseteq A(\mathcal{C}(n))$). A strategy is Timeserving if when played against* FRONTIER*, with probability 1, all PublishSet(V', E') actions it takes are Timeserving.*

It is easy to see that FRONTIER is itself Timeserving: it publishes at most a single block at a time, and that block is the new unique longest chain. We first argue that there exists an optimal strategy against FRONTIER which is also Timeserving.

Theorem 4.2 (Timeserving)**.** *For any strategy π, there is a strategy $\tilde{\pi}$ that is Timeserving, takes a valid action at every step, and satisfies $\text{REV}_\gamma^{(n)}(\tilde{\pi}) = \text{REV}_\gamma^{(n)}(\pi)$ for all γ and $n \in \mathbb{N}$.*

We now state three basic properties of Timeserving strategies.

Observation 4.3. *If π is Timeserving, then*

(i) Whenever π publishes blocks, it publishes a single path. Formally, whenever π takes action PublishSet(V', E') in round n, with $V' = \{b_1, \dots, b_k\}$ ($b_i < b_{i+1}$ for all i), then E' contains an edge $b_{i+1} \to b_i$ for all $i \in [k-1]$, and an edge $b_1 \to b$ for some $b \in V$.

(ii) There are never two leaves of the same height. Formally, for all leaves $q \neq \tilde{q} \in V$, $h(q) \neq h(\tilde{q})$.

(iii) *Whenever π forks, it publishes at least two blocks. Formally, whenever π takes the action PublishSet(V', E') which removes the old longest chain from the longest path, then $|V'| \geq 2$.*

Observation 4.3 gives us some nice structure about Timeserving strategies (and Theorem 4.2 asserts that it is w.l.o.g. to study such strategies). In particular, we only need to consider strategies which publish a single path at a time. Formally, we may w.l.o.g. replace the action PublishSet(V', E') with the action:

Definition 4.4 (PublishPath)**.** *Taking action PublishPath(V', u) with $u \in V$ and $V' \subseteq \mathcal{U}$ is equivalent to taking action PublishSet(V', E'), where E' contains an edge from the minimum element of V' to u, and an edge from v to the largest element of V' strictly less than v, for all other $v \in V'$.*

4.2 Step 2: Orderly

Section 4.1 provides structure on *when* we may assume blocks are announced, but it does not yet provide structure on *which blocks* are announced. Specifically, for all we know right now it could still be that when a strategy chooses to take action PublishPath(V', u), and $|V'| = k$, the precise k blocks it chooses to publish matter (e.g. in state $B_{2,0}$ it could choose to publish $2 \to 0$ versus $1 \to 0$). Our next reduction shows that it is without loss to consider only strategies which are *Orderly*, and always publish the earliest legal blocks. Intuitively, this gives the strategy more flexibility later on. For simplicity of notation, we introduce the terms $\min^{(k)}\{S\} \subseteq S$ to refer to the $\min\{k, |S|\}$ smallest elements in S and $\max^{(k)}\{S\} \subseteq S$ to refer to the $\min\{k, |S|\}$ largest elements in S.

Definition 4.5 (Orderly)**.** *The action PublishPath(V', u) is* Orderly *if $V' = \min^{(|V'|)}(\mathcal{U} \cap (u, \infty))$. That is, an action is Orderly if it publishes the smallest $|V'|$ blocks it could have possibly published on top of u. A strategy is Orderly if when played against* FRONTIER*, with probability* 1*, all PublishPath$(\cdot, \cdot)$ actions it takes are Orderly.*

Theorem 4.6 (Orderly)**.** *Let π be any Timeserving strategy. Then there is a valid, Timeserving, Orderly strategy $\tilde{\pi}$ that satisfies $\text{Rev}_\gamma^{(n)}(\tilde{\pi}) = \text{Rev}_\gamma^{(n)}(\pi)$ for all γ and $n \in \mathbb{N}$.*

We conclude this section by noting that, after restricting attention to Orderly strategies, we can further replace the action PublishPath(V', u) with the action:

Definition 4.7 (Publish)**.** *Taking action Publish(k, u) with $k \in \mathbb{N}_+$ and $u \in V$ is equivalent to taking the action PublishPath$(\min^{(k)}(\mathcal{U} \cap (u, \infty)), u)$.*

4.3 Step 3: Longest Chain Mining

We now have structure on *when* blocks are published, and *which* blocks are published, but not yet on *where* those blocks are published. Specifically, an *orphaned chain* is a path in Tree that used to be part of the longest path $A(\mathcal{C})$ but was overtaken by another path. Intuitively, a chain can only be orphaned by Miner 1 and if Miner 1 is playing according to an optimal strategy, publishing blocks which build on top of orphaned chains should be sub-optimal. We define a strategy as *Longest Chain Mining* if it never publishes on top of a block in an orphaned chain.

Definition 4.8 (Longest Chain Mining). *Action Publish(k,u) is* Longest Chain Mining *(LCM) if $u \in A(\mathcal{C})$ is a block in the longest path. That is, an action is LCM if it builds on top of some block within the longest path (not necessarily the leaf). A strategy is LCM if, with probability* 1*, every Publish action it takes against* FRONTIER *is LCM.*

Previous work on Proof-of-Work mining games [16, 22] assume all strategies are LCM. For Proof-of-Stake mining games, Theorem 4.9 proves that it is w.l.o.g. to assume an LCM strategy.

Theorem 4.9 (LCM). *Let π be any Timeserving, Orderly strategy. Then there is a $\tilde{\pi}$ that is Timeserving, Orderly, LCM, takes a valid action at every step, and satisfies $\textsc{Rev}_\gamma^{(n)}(\tilde{\pi}) \geq \textsc{Rev}_\gamma^{(n)}(\pi)$ for all γ and $n \in \mathbb{N}$.*

4.4 Step 4: Trimmed

With Theorems 4.2, 4.6 and 4.9, we can immediately conclude that there exists an optimal strategy satisfying several structural properties. We wrap up by showing one final property, and will show that there exists an optimal strategy which is *Trimmed.*

Definition 4.10 (Trimmed Action). *Action Publish(k,v) is* Trimmed *if $v \in A(\mathcal{C})$ and whenever v is not* the *longest chain (that is, $v \neq \mathcal{C}$), and u is the unique node in $A(\mathcal{C})$ with an edge to v, then u was created by Miner* 2 *(that is, $u \in T_2$).*

Put another way, every Publish(k,v) either builds on top of the longest chain (in which case it is vacuously Trimmed), or kicks out the successors of v. In the latter case, an action is Trimmed if and only if the *minimum* successor of v was created by Miner 2.

Definition 4.11 (Trimmed Strategy). *A strategy is* Trimmed *if every action it takes is either Wait or Trimmed.*

We now conclude our main theorem of this section.

Theorem 4.12 (Trimming). *For all strategies π, there is a Trimmed strategy $\tilde{\pi}$ that take valid actions in every step, and $\textsc{Rev}_\gamma^{(n)}(\tilde{\pi}) \geq \textsc{Rev}_\gamma^{(n)}(\pi)$ for all γ and $n \in \mathbb{N}$.*

5 Trimming the State Space

So far, we have greatly simplified strategies which we need to consider. However, we still have not even established that there exists an optimal strategy which is *recurrent.* That is, for all we know so far, the optimal strategy might need to store not only the entire longest chain, but also all blocks which have ever been published, and all unpublished blocks which they ever created. The goal in this section is to establish that an optimal strategy exists which is recurrent: it will eventually (with probability 1) reach a "checkpoint" which the strategy treats as a new genesis block that will never be overridden.

5.1 Checkpoints and Weak Recurrence

We iteratively define a sequence of blocks $P_0, P_1, \ldots$ in the longest path $A(\mathcal{C})$ to be checkpoints as follows.

Definition 5.1 (Checkpoints). *Based on the current state, checkpoints are iteratively defined as follows.*

- *The first checkpoint, P_0, is the genesis block.*
- *If P_{i-1} is undefined, then P_i is undefined as well.*
- *If P_{i-1} is defined, then v is a* potential i^{th} checkpoint *if:*
  - *$v > P_{i-1}$.*
  - *$v \in A(\mathcal{C})$.*
  - *Among blocks that Miner 1 created between P_{i-1} and v (including v, not including P_{i-1}), more are in the longest chain than unpublished. That is, $|A(\mathcal{C}) \cap (P_{i-1}, v] \cap T_1| \geq |\mathcal{U}_1 \cap (P_{i-1}, v]|$.*
- *If there are no potential i^{th} checkpoints, then P_i is undefined.*
- *Else, then P_i is defined to be the minimum potential i^{th} checkpoint.*

Note that each P_i is again a random variable, meaning that a priori P_i might change over time, including from undefined to defined. For example, $P_i(n)$ would denote the i^{th} checkpoint, as defined by the state after the conclusion of the n^{th} round (we will later prove that there exists an optimal strategy which never changes or undefines P_i once it is defined. But this will be a result, and not a definition).

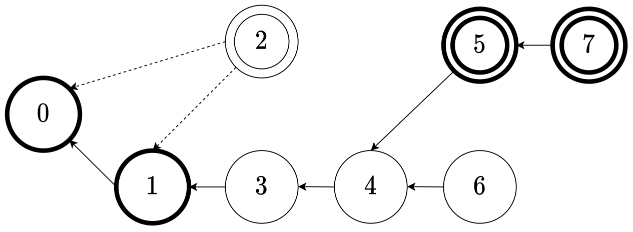

Figure 7: Example of a state and its checkpoints. Blocks with thicker lines denote blocks that are checkpoints and blocks with thinner lines denote blocks that are not checkpoints. From Definition 5.1, blocks 0, 1, 5, and 7 are checkpoints.

Example 5.2. *Consider the state in Figure 7 where blocks 0, 1, 5, and 7 are the checkpoints. By definition, block 0 is the base-case and is always a checkpoint. Block 1 is a checkpoint because Miner 1 has no unpublished blocks in the interval* $(0, 1]$. *Block 2 is unpublished and thus not a checkpoint. Block 3 is not a checkpoint because Miner 1 has one unpublished block in the interval* $(1, 3]$ *and zero published block in the path from* $3 \to 1$ *(not counting block 1). From a similar reasoning, blocks 4 and 6 are not checkpoints. Block 5 is a checkpoint because Miner 1 has one unpublished block in the interval* $(1, 5]$ *and one block in the path* $5 \to 4 \to 3 \to 1$ *(not counting block 1). Block 7 is a checkpoint because Miner 1 has no unpublished blocks in the interval* $(5, 7]$.

The main result of this section is stated below, and claims that there exists an optimal strategy which treats checkpoints like the genesis block.

Definition 5.3 (Checkpoint Recurrent)**.** *A strategy π is* Checkpoint Recurrent *if when π is played against* FRONTIER*:*

- *For all $i \in \mathbb{N}$, if P_i changes from undefined to defined, P_i never changes again (in particular, this implies that once P_i is defined, it remains in $A(\mathcal{C})$ forever).*
- *Immediately when P_i becomes defined, neither player has any unpublished blocks $> P_i$.*

When bullet one is satisfied, checkpoints are never overridden. Given that bullet one holds, bullet two implies that immediately when P_i is defined, it is essentially a genesis block (because bullet one holds, no unpublished blocks $< P_i$ can ever enter the longest chain. If bullet two also holds, then there are no unpublished blocks $> P_i$, so there are no relevant unpublished blocks, and P_i is in the longest chain forever, just like the genesis block in round 0). This implies that when optimizing over Checkpoint Recurrent strategies, it suffices to consider only strategies that reset its state space whenever a new checkpoint is defined. That is, whenever a new checkpoint is defined Miner 1 capitulates to state B_0.

Theorem 5.4 (Weak-Recurrence)**.** *There exists an optimal strategy which is checkpoint recurrent.*

The weak-recurrence theorem provides a useful tool to reduce the state space of optimal strategies; however, it does not say how often (if ever) the block tree reaches a new checkpoint. Fortunately, each new checkpoint give us important information about the payoff of a strategic miner: *if the block tree never reaches a checkpoint, then at all times Miner 1 has at least half of their blocks unpublished.* Next, we check such strategies are not better than FRONTIER before diving into the proof of the weak-recurrence theorem.

Proposition 5.5. *If π is checkpoint recurrent and P_1 is never defined, then* $\textsc{Rev}(\pi) \leq \textsc{Rev}(\text{FRONTIER})$.

As a first step toward proving upper bounds in the revenue, we will require a simply but useful fact about rate of growth of the block tree.

Lemma 5.6 (Minimum Growth Rate)**.** *For any mining game starting at state $X_0 = B_0$,*

$$\liminf_{n\to\infty} \frac{h(\mathcal{C}(X_n))}{n} \geq 1 - \alpha, \quad \textit{with probability 1.} \tag{10}$$

Corollary 5.7. *For any optimal strategy π,* $\textsc{Rev}(\text{FRONTIER}) = \alpha \leq \textsc{Rev}(\pi) \leq \frac{\alpha}{1-\alpha}$.

Both Lemma 5.6 and Corollary 5.7 will be useful to prove Proposition 5.5. We need to understand one property of checkpoints, and then we can complete the proof. Intuitively, Proposition 5.8 and Corollary 5.9 just apply the definition of checkpoints to relate the number of blocks that Miner 1 has unpublished vs. published in the longest path.

Proposition 5.8. *For all $v \in A(\mathcal{C})$,* *(i) If v is a checkpoint, then for all checkpoints $P_i > v$, $|A(\mathcal{C}) \cap (v, P_i] \cap T_1| \geq |\mathcal{U} \cap (v, P_i)|$.*

(ii) If v is not a checkpoint and $P_i = \max\{P_j : P_j < v\}$, then $|A(\mathcal{C}) \cap (P_i, v] \cap T_1| < |\mathcal{U} \cap (P_i, v]|$.

(iii) If v is not a checkpoint, then for all checkpoints $P_i > v$, $|A(\mathcal{C}) \cap (v, P_i] \cap T_1| > |\mathcal{U} \cap (v, P_i]|$.

Corollary 5.9. *Suppose $v \in A(\mathcal{C})$ is not a checkpoint and let P_i be the highest checkpoint below v. Then Miner 1 publishes in the longest path less than half of all blocks they created from time $P_i + 1$ to v. That is, $|A(\mathcal{C}) \cap (P_i, v] \cap T_1| < \frac{|T_1 \cap (P_i, v]|}{2}$.*

Proof. Bullet (ii), Proposition 5.8, implies

$$|A(\mathcal{C}) \cap (P_i, v] \cap T_1| + |\mathcal{U} \cap (P_i, v]| > 2|A(\mathcal{C}) \cap (P_i, v] \cap T_1|.$$

Suppose for contradiction $|A(\mathcal{C}) \cap (P_i, v] \cap T_1| \geq \frac{|T_1 \cap (P_i, v)|}{2}$. Then, the number of unpublished blocks plus blocks in the longest path would be strictly bigger than the number of blocks Miner 1 created, a contradiction. □

Proof of Proposition 5.5. From the strong law of large numbers, Corollary 5.9 and Lemma 5.6,

$$\limsup_{n\to\infty} \frac{|A(\mathcal{C}(X_n)) \cap T_1|}{|A(\mathcal{C}(X_n))|} \leq \limsup_{n\to\infty} \frac{\frac{|T_1 \cap (0,n]|}{2n}}{\frac{|A(\mathcal{C}(X_n))|}{n}} \leq \frac{\limsup_{n\to\infty} \frac{|T_1 \cap (0,n]|}{2n}}{\liminf_{n\to\infty} \frac{|A(\mathcal{C}(X_n))|}{n}} \leq \frac{\alpha/2}{(1-\alpha)} \leq \alpha.$$

where the last inequality uses the fact $\alpha \leq 1/2$. Intuitively, if Miner 1 never reaches a checkpoint, then they are publishing at most $\alpha n/2$ blocks in expectation by round n (and clearly at most $\alpha n/2$ of these can be in the longest path). But Lemma 5.6 asserts that there are at least $(1-\alpha)n$ blocks in expectation in the longest path by round n. Therefore, the fraction produced by Miner 1 cannot be too high (and in particular, honesty would have been better in expectation). □

We prove Theorem 5.4 in two steps, Section 5.2 and Section 5.3.

5.2 Step 1: Checkpoint Preserving

The first step is to show the existence of an optimal strategy that never forks a checkpoint. For that, we will give an explicitly procedure f to transform any strategy π that could fork checkpoints into another strategy $f(\pi)$ that does not fork checkpoints satisfying $\text{Rev}(f(\pi)) \geq \text{Rev}(\pi)$.

Definition 5.10 (Finality)**.** *A block $q \in A(\mathcal{C})$ reaches* finality *with respect to strategy π if, with probability 1, π takes no action that removes q from longest path.*

Definition 5.11 (Checkpoint Preserving)**.** *A strategy π is* checkpoint preserving *if whenever a new checkpoint P_i is defined, P_i reaches finality with respect to π.*

Theorem 5.12. *For every strategy π, there is a trimmed, checkpoint preserving strategy $f(\pi)$ with $\text{Rev}(f(\pi)) \geq \text{Rev}(\pi)$.*

5.3 Step 2: Opportunistic

We have shown the existence of an optimal strategy that would never fork the longest chain $\mathcal{C}$ when it becomes a checkpoint. However, to be checkpoint recurrent, we must also show Miner 1 has no unpublished blocks bigger than $\mathcal{C}$ when the longest chain is a checkpoint. The converse can only happen when Miner 1 is about to take action PublishPath(Q, v) and $\max\ Q$ will reach finality with respect to Miner 1's strategy, but Miner 1 would leave an unpublished block $q > \max\ Q$. The intuition is that Miner 1 can wait instead of publishing Q pointing to v in current round. If Miner 1 creates the next block, Miner 1 can publish Q pointing to v as before. If Miner 2 creates the next block, Miner 1 can still take action PublishPath$(Q \cup \{q\}, v)$ adding $Q \cup \{q\}$ to the longest path.

Definition 5.13 (Opportunistic)**.** *Let π be a strategy and let B be a state. Action PublishPath(Q, v) is* opportunistic *with respect to B and π if*

- *PublishPath(Q, v) is a valid action at state B.*
- *If π takes action PublishPath(Q, v) where* $\max\ Q$ *reaches finality with respect to π (Definition 5.10), then $Q = \mathcal{U}(B) \cap (v, \infty)$.*

Strategy π is opportunistic *if at all states B, π waits or takes an opportunistic action with respect to B and π.*

Theorem 5.14. *For any strategy π, there is a valid, trimmed, checkpoint preserving and opportunistic strategy $f(\pi)$ with* $\textsc{Rev}(f(\pi)) \geq \textsc{Rev}(\pi)$.

Proof of Theorem 5.4. Theorem 5.14 directly implies the weak-recurrence theorem since a checkpoint preserving and opportunistic strategy is also checkpoint recurrent. □

5.4 Step 3: Strong Recurrence

So far, we have shown that there exist an optimal strategy that is checkpoint recurrent. That is, once we reach a state X_t where $\mathcal{C}(X_t)$ is a checkpoint, Miner 1 capitulates to state B_0. Next, we will aim for a stronger result.

Theorem 5.15 (Strong recurrence)**.** *There exists an optimal checkpoint recurrent and positive recurrent strategy.*

For a proof sketch, observe the Weak Recurrence Theorem implies there exists an optimal strategy π that is checkpoint recurrent. We will assume π is not positive-recurrent (i.e., the expected time $\mathbb{E}[\tau]$ to define a new checkpoints is infinite) and derive that $\textsc{Rev}(\pi) \leq \alpha = \textsc{Rev}(\text{FRONTIER})$. The case where Miner 1 never defines checkpoint P_1 – i.e., $\tau = \infty$ with probability 1 in Proposition 5.5 – give us intuition why the claim should hold to the more general case where $\mathbb{E}[\tau] = \infty$. Once we proof $\textsc{Rev}(\pi) \leq \textsc{Rev}(\text{FRONTIER})$, we just observe FRONTIER is checkpoint and positive recurrent. Thus there exists an optimal checkpoint and positive recurrent strategy.

6 Nash Equilibrium

We briefly give intuition behind our second main result, which leverages Theorem 5.15 to lower bound α^{PoS}.

Theorem 6.1. *For $\alpha \leq 0.308$,* FRONTIER *is an optimal strategy for Miner 1 when Miner 2 follows* FRONTIER*.*

We defer the proof to Appendix G. The main idea behind the proof is to show that Nothing-at-Stake Selfish Mining is almost optimal when $\alpha < 1/3$. The following proof-sketch highlights the main insights of the proof.

Selfish Mining is optimal when Miner 2 creates the first block (when $\alpha < 1/3$). We know that there is an optimal checkpoint recurrent strategy, Theorem 5.15. Therefore, it is optimal for Miner 1 to capitulate to state B_0 if Miner 2 creates and publishes $1 \to 0$ (which is exactly what selfish mining does).

Selfish Mining is optimal after Miner 1 creates and withholds blocks 1 and 2. Starting from state $B_{2,0}$, selfish mining will wait until the first time step τ when the lead decrease to a single block to fork all of Miner 2 blocks. We show that waiting until time τ is indeed optimal for any value of α (which is not surprising). Less obvious is why Miner 1 must publishes all his blocks at time τ when they still have a lead of a single block. Indeed we should not expect this to be optimal for all values of α. If Miner 1 waits at time τ and creates the next block, they will again have a lead of two blocks and can resume to "selfish mine". Here, we shown that Miner 1 creates $\frac{\alpha}{1-2\alpha}$ blocks on expectation from the time they have a lead of two blocks to the moment the lead decreases to a single block. This quantity can be arbitrarily large but it is at most 1 when $\alpha < 1/3$ so it is a risky action for Miner 1. That is because if (instead) Miner 2 creates next block, there is a tie (Miner 1 does not have enough blocks to fork the longest chain) and we can show that the probability Miner 1 will ever publish any blocks created before time τ is at most $\frac{\alpha}{1-\alpha}$. Formally, we will prove Miner 1 maximizes rewards by waiting until time τ and immediately publishing all unpublished blocks (which is what selfish mining does).

There is little window to improve Selfish Mining when Miner 1 creates and withhold block 1 and Miner 2 publishes $2 \to 0$. Winning the tie-breaking at state $B_{1,1}$ is another source of revenue for Miner 1. In fact, the only improvement that Nothing-at-Stake Selfish Mining provides over standard Selfish Mining is increasing the probability that Miner 1 wins the tie-breaking between blocks 1 and 2. From a similar argument from previous bullet, we can show the probability of adding block 1 to the longest path is at most $\frac{\alpha}{1-\alpha}$. Next, we observe Miner 1 has no more advantage of being the creator of the block at height $\ell \geq 2$ at state $B_{1,1}$ than being the creator of the block at height $\ell - 1$ at state B_0. We formalize this intuition by showing that, by ignoring blocks 1 and 2, any action taken on a state reachable from $B_{1,1}$ can be converted into an action for an state reachable from B_0. The only advantage state $B_{1,1}$ provides over state B_0 is that Miner 1 has a probability (of at most $\frac{\alpha}{1-\alpha}$) of adding block 1 to the longest path.

Wrapping up. From the discussion above, state $B_{1,1}$ is the only state where we could possible search for a better strategy than Nothing-at-Stake Selfish Mining when $\alpha < 1/3$, but there is little window to improve Miner 1's action at state $B_{1,1}$. As a result, we will obtain FRONTIER is optimal when $\alpha \leq 0.308$ as desired.

7 Conclusion

We study miner incentives in longest-chain proof-of-stake protocols with perfect external randomness. We show both that such protocols are strictly more vulnerable to manipulation than those based on proof-of-work (Theorem 3.4), but also that it is a Nash equilibrium for all miners to follow the longest-chain protocol as long as no miner has more than ≈ 0.308 of the total stake (Theorem 6.1). Our main technical results characterize potentially optimal strategies in a complex, infinite-state MDP (Theorem 5.15). Our work motivates several natural open problems:

- Theorem 5.15, combined with the analysis in Theorem 6.1, provides strong structure on optimal strategies. It is therefore conceivable that a simulation-based approach with MDP solvers (as in [22]) could estimate α^{PoS} to high precision.
- Our Theorem 6.1 provides a reduction from incentive-compatible longest-chain proof-of-stake protocols to designing a randomness beacon and a slashing protocol. Clearly, it is important for future work to construct these primitives, although these are well-known and ambitious open problems. In our setting, it is further important to understand what are the *minimal* assumptions on a randomness beacon or slashing protocol necessary to leverage Theorem 6.1.

References

[1] Nick Arnosti and S. Matthew Weinberg. 2019. Bitcoin: A Natural Oligopoly. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA (LIPIcs, Vol. 124)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 5:1–5:1.

[2] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. 2018. Verifiable delay functions. In *Annual international cryptology conference*. Springer, 757–788.

[3] Joseph Bonneau, Jeremy Clark, and Steven Goldfeder. 2015. On Bitcoin as a public randomness source. *IACR Cryptol. ePrint Arch.* 2015 (2015), 1015.

[4] Jonah Brown-Cohen, Arvind Narayanan, Alexandros Psomas, and S Matthew Weinberg. 2019. Formal barriers to longest-chain proof-of-stake protocols. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 459–473.

[5] Miles Carlsten, Harry A. Kalodner, S. Matthew Weinberg, and Arvind Narayanan. 2016. On the Instability of Bitcoin Without the Block Reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. ACM, 154–167.

[6] Xi Chen, Christos H. Papadimitriou, and Tim Roughgarden. 2019. An Axiomatic Approach to Block Rewards. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies, AFT 2019, Zurich, Switzerland, October 21-23, 2019*. ACM, 124–131.

[7] Jeremy Clark and Urs Hengartner. 2010. On the Use of Financial Data as a Random Beacon. *EVT/WOTE* 89 (2010).

[8] Phil Daian, Rafael Pass, and Elaine Shi. 2019. Snow White: Robustly Reconfigurable Consensus and Applications to Provably Secure Proof of Stake. In *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 11598)*. Springer, 23–41.

[9] Rick Durrett. 2019. *Probability: theory and examples*. Vol. 49. Cambridge university press.

[10] Ittay Eyal and Emin Gün Sirer. 2014. Majority is not enough: Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security*. Springer, 436–454.

[11] Matheus V. X. Ferreira, Daniel J. Moroz, David C. Parkes, and Mitchell Stern. 2021. Dynamic Posted-Price Mechanisms for the Blockchain Transaction-Fee Market. arXiv:2103.14144 [cs.GT]

[12] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*. 51–68.

[13] LM Goodman. 2014. Tezos: A self-amending crypto-ledger position paper. *Aug* 3 (2014), 2014.

[14] Gur Huberman, Jacob Leshno, and Ciamac Moallemi. 2020. Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System. *Review of Economic Studies* (2020).

[15] John Kelsey, Luís TAN Brandão, Rene Peralta, and Harold Booth. 2019. *A reference for randomness beacons: Format and protocol version 2*. Technical Report. National Institute of Standards and Technology.

[16] Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. 2016. Blockchain mining games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. 365–382.

[17] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. 2017. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*. Springer, 357–388.

[18] Jacob Leshno and Philipp Strack. 2020. Bitcoin: An Impossibility Theorem for Proof-of-Work based Protocols. *American Economics Review: Insights* (2020).

[19] Satoshi Nakamoto. 2007. *Bitcoin: A peer-to-peer electronic cash system.* Technical Report.

[20] Michael Neuder, Daniel J. Moroz, Rithvik Rao, and David C. Parkes. 2021. Selfish Behavior in the Tezos Proof-of-Stake Protocol. *Cryptoeconomic Systems* 0, 1 (5 4 2021). `https://doi.org/10.21428/58320208.27350920` https://cryptoeconomicsystems.pubpub.org/pub/neuder-selfish-behavior-tezos.

[21] Michael O Rabin. 1983. Transaction protection by beacons. *J. Comput. System Sci.* 27, 2 (1983), 256–267.

[22] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2016. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 515–532.

[23] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151, 2014 (2014), 1–32.

In Appendix A, we give relevant real analysis background. In Appendix B, we give relevant probability theory background. In Appendix C, we introduce the Markov Decision Process formulation. In Appendix H, we provide a table of notation. In the remaining sections, we provide omitted proves.

A Real Analysis Background

Let $a_1, a_2, \dots$ be a sequence of real numbers. The limit of a sequence $a_1, a_2, \dots$ exists if the sequence converges to $a \in \mathbb{R}$. We write

$$a_n \to a, \quad \text{or} \quad \lim_{n\to\infty} a_n = a.$$

The limit inferior and limit superior of a sequence $a_1, a_2, \dots$ is defined as

$$\liminf_{n\to\infty} a_n := \lim_{n\to\infty} \inf_{k\geq n} a_k := \sup_{n\geq 1} \inf_{k\geq n} a_k,$$

$$\limsup_{n\to\infty} a_n := \lim_{n\to\infty} \sup_{k\geq n} a_k := \inf_{n\geq 1} \sup_{k\geq n} a_k,$$

respectively. The limit of $a_1, a_2, \dots$ exists and is equals to a if and only if

$$\liminf_{n\to\infty} a_n = \limsup_{n\to\infty} a_n = a.$$

Lemma A.1 (Properties of lim inf and lim sup)**.** *Whenever the right hand side is well-defined (not of the form $\pm\infty + \mp\infty$ or $0 \cdot \pm\infty$).*

(i) Supperadditivity. $\liminf_{n\to\infty}(a_n + b_n) \geq \liminf_{n\to\infty} a_n + \liminf_{n\to\infty} b_n$.

(ii) Subadditivity. $\limsup_{n\to\infty}(a_n + b_n) \leq \limsup_{n\to\infty} a_n + \limsup_{n\to\infty} b_n$.

(iii) Supermultiplicativity. $\liminf_{n\to\infty}(a_n \cdot b_n) \geq \liminf_{n\to\infty} a_n \cdot \liminf_{n\to\infty} b_n$.

(iv) Submultiplicativity. $\limsup_{n\to\infty}(a_n \cdot b_n) \leq \limsup_{n\to\infty} a_n \cdot \limsup_{n\to\infty} b_n$.

(v) If $a_n \to a$, $\liminf_{n\to\infty} a_n + b_n = a + \liminf_{n\to\infty} b_n$.

(vi) If $a_n \to a$, $\limsup_{n\to\infty} a_n + b_n = a + \limsup_{n\to\infty} b_n$.

(vii) If $a_n \to a$ and b_n is bounded, $\liminf_{n\to\infty} a_n b_n = a \liminf_{n\to\infty} b_n$.

(viii) If $a_n \to a$ and b_n is bounded, $\limsup_{n\to\infty} a_n b_n = a \limsup_{n\to\infty} b_n$.

B Probability Theory Background

B.1 Convergence of Random Variables

Definition B.1 (Almost sure convergence)**.** *We say a sequence of random variables $X_1, X_2, \ldots$ converges almost surely to random variable X (and we write $X_n \overset{a.s.}{\to} X$) if*

$$Pr\left[\lim_{n\to\infty} X_n = X\right] = 1.$$

Proposition B.2. *The following are equivalent:*

- $X_n \overset{a.s.}{\to} X$*.*
- $\lim_{n\to\infty} Pr\left[\cup_{i=n}^{\infty}\{|X_i - X| \geq 1/k\}\right] = 0$ *for any* $k \geq 1$*.*
- $\lim_{n\to\infty} Pr\left[\cap_{i=n}^{\infty}\{|X_i - X| < 1/k\}\right] = 1$ *for any* $k \geq 1$*.*

B.2 Laws of Large Numbers

Definition B.3 (Absolutely Integrable)**.** *A random variable X is* absolutely integrable *if* $\mathbb{E}\left[|X|\right] < \infty$*.*

Lemma B.4 (Strong Law of Large Numbers (SSLN), Theorem 2.4.1 in [9])**.** *Let $X_1, X_2, \ldots$ be an i.i.d. sequence of copies of absolutely integrable random variable X. Then*

$$\frac{1}{n}\sum_{i=1}^{n} X_i \overset{a.s.}{\to} \mathbb{E}\left[X\right].$$

Remark: The strong law can be generalized for the case where X is non-negative and not *absolutely integrable. That is, if $\mathbb{E}\left[X\right] = \infty$ and $Pr\left[X \geq 0\right] = 1$, then $\frac{1}{n}\sum_{i=1}^{n} X_i \overset{a.s.}{\to} \infty$.*

Definition B.5 (Counting Process)**.** *The* interarrival times *$\tau_1, \tau_2, \ldots$ is an i.i.d. sequence of positive and absolutely integrable random variables. Let $X_n = \sum_{i=1}^{n} \tau_i$ with $X_0 = 0$. The random variable $N_n = \sum_{i=1}^{\infty} \mathbb{1}_{X_i \leq n}$ denotes the* counting proces *associated with interarrival times $\tau_1, \tau_2, \ldots$.*

Observation B.6. *If $N_n = \sum_{i=1}^{\infty} \mathbb{1}_{X_i \leq n}$ is a counting process, for all $n \in \mathbb{N}$, $X_{N_n} \leq n < X_{N_n+1}$.*

Lemma B.7 (SLLN for Counting Processes, Theorem 2.4.7 in [9])**.** *If $N_n = \sum_{i=1}^{\infty} \mathbb{1}_{\tau_i \leq n}$ is a counting process, then*

$$\frac{N_n}{n} \overset{a.s.}{\to} \frac{1}{\mathbb{E}[\tau_i]}.$$

C Markov Decision Process

In this section, we show we can obtain a closed form for the revenue of positive recurrent strategies by using the strong law of large numbers. See Appendix B for basic background on probability theory. A sequence of random variables $X_1, X_2, \ldots$ *converges almost surely* (or with probability 1) to random variable X (and we write $X_n \overset{a.s.}{\to} X$) if $Pr[\lim_{n\to\infty} X_n = X] = 1$. Our first result, is a generalization of Corollary 9 in [22] to Proof-of-Stake mining games with positive recurrent strategies.

Theorem C.1. *Let $X_0, X_1, \ldots$ be a mining game starting at state $X_0 = B_0$ where Miner 1 follows a positive recurrent strategy. Let $R^k = \sum_{t=1}^{\tau} r^k(X_{t-1}, X_t)$ be Miner k's total reward before the first time step $\tau \geq 1$ where Miner 1 capitulates to state B_0. Then*

(i) For $k = 1, 2$, $\frac{|A(\mathcal{C}(X_n))\cap T_k|}{n} \overset{a.s.}{\to} \frac{\mathbb{E}[R^k]}{\mathbb{E}[\tau]}$.

(ii) $\textsc{Rev}(\pi) = \frac{\mathbb{E}[R^1]}{\mathbb{E}[R^1+R^2]}$.

Proof. Without loss of generality assume $X_0 = B_0$ and let $n_0 = 0$. Let $X_{n_1}, X_{n_2}, \ldots$ be the sequence of states where Miner 1 capitulates to state B_0. Let R_i^k be Miner k's reward from time step n_{i-1} to time n_i. That is,

$$R_i^k = |A(\mathcal{C}(X_{n_i})) \cap T_k| - |A(\mathcal{C}(X_{n_{i-1}})) \cap T_k|.$$

Observe $R_1^k, R_2^k, \ldots$ and $n_1 - n_0, n_2 - n_1, \ldots$ are i.i.d. sequences of random variables (because Miner 1 capitulates to state B_0 at time step n_i). Additionally $\mathbb{E}\left[n_{i+1} - n_i\right] < \infty$ because Miner 1's strategy is positive recurrent. Thus define the counting process $N_n = \sum_{i=1}^{\infty} \mathbb{1}_{n_i \leq n}$. From Observation B.6, $n_{N_n} \leq n < n_{N_n+1}$, and since Miner 2 never forks Miner 1's blocks, the number of Miner 1's blocks in the longest path is an increasing function of time. Thus

$$\frac{|A(\mathcal{C}(X_{n_{N_n}})) \cap T_1|}{n} \leq \frac{|A(\mathcal{C}(X_n)) \cap T_1|}{n} \leq \frac{|A(\mathcal{C}(X_{n_{N_n+1}})) \cap T_1|}{n}.$$

Observe $\sum_{i=1}^{N_n} R_i^k = |A(\mathcal{C}(X_{N_n})) \cap T_k|$ is a telescoping sum. Thus

$$\frac{N_n}{n}\frac{\sum_{i=1}^{N_n} R_i^1}{N_n} \leq \frac{|A(\mathcal{C}(X_n)) \cap T_1|}{n} \leq \frac{N_n+1}{n}\frac{\sum_{i=1}^{N_n+1} R_i^1}{N_n+1}.$$

Observe $N_n \to \infty$ as $n \to \infty$ (because Miner 1's strategy is recurrent) and $\mathbb{E}[R^k] \leq \mathbb{E}[\tau] < \infty$ (because Miner 1's strategy is positive recurrent). From the SLLN for Counting Processes (Lemma B.7) and SLLN (Lemma B.4),

$$\frac{N_n}{n}\frac{1}{N_n}\sum_{i=1}^{N_n} R_i^k \overset{a.s.}{\to} \frac{\mathbb{E}[R^k]}{\mathbb{E}[\tau]} \quad \text{and} \quad \frac{N_n}{n}\frac{1}{N_n+1}\sum_{i=1}^{N_n+1} R_i^k \overset{a.s.}{\to} \frac{\mathbb{E}[R^k]}{\mathbb{E}[\tau]}.$$

From the Sandwich Theorem,

$$\frac{|A(\mathcal{C}(X_n)) \cap T_1|}{n} \overset{a.s.}{\to} \frac{\mathbb{E}[R^1]}{\mathbb{E}[\tau]}.$$

Observe the height of the block tree is also a increasing function of time. Thus with an similar argument from above, we sandwich $h(\mathcal{C}(X_n))$ between $h(\mathcal{C}(X_{n_{N_n}}))$ and $h(\mathcal{C}(X_{n_{N_n+1}}))$ to get

$$\frac{N_n}{n}\frac{h(\mathcal{C}(X_n))}{n} \overset{a.s.}{\to} \frac{\mathbb{E}[R^1+R^2]}{\mathbb{E}[\tau]}.$$

From linearity of expectation, we get $\frac{|A(\mathcal{C}(X_n))\cap T_2|}{n} \overset{a.s.}{\to} \frac{\mathbb{E}[R^2]}{\mathbb{E}[\tau]}$. This proves the first bullet. From the first bullet,

$$\textsc{Rev}(\pi) = \mathbb{E}\left[\liminf_{n\to\infty} \frac{\frac{1}{n}|A(\mathcal{C}(X_n)\cap T_1|}{\frac{1}{n}|A(\mathcal{C}(X_n)\cap T_1| + \frac{1}{n}|A(\mathcal{C}(X_n))\cap T_2|}\right] = \frac{\mathbb{E}[R^1]}{\mathbb{E}[R^1]+\mathbb{E}[R^2]}.$$

This proves the second bullet. □

Theorem C.1 will be useful to study the revenue of positive recurrent strategies such as the nothing-at-stake selfish mining strategy—which will be positive recurrent by design. It also come in hand when studying optimal strategies; however, it require us to assume the existence an optimal positive recurrent strategy. In fact, Sapirshtein et al. [22] and Kiayias et al. [16] relies on the structure of proof-of-work mining games to conjecture the existence an optimal positive recurrent strategy. In proof-of-stake, there is no obvious incentives for a miner to forget unpublished blocks. Thus it is not clear if an optimal strategy would be (positive) recurrent (recall Example 2.6 where Miner 1 might be motivated to fork their own blocks). As one of our technical contributions, Theorem 5.15 proves proof-of-stake mining games admits optimal strategies that are positive recurrent.

Additionally, Sapirshtein et al. [22] uses Theorem C.1 to compute an optimal positive recurrent strategy by optimizing a Markov Decision Process (MDP). Instead of explicitly computing optimal strategies, we will use their MDP to give sufficient conditions on α for which FRONTIER is an optimal strategy (Theorem 6.1). Given a parameter λ, their MDP defines the real-valued function r_λ as the reward from transition from state B to B':

Definition C.2 (Mining Game Reward). *For $\lambda \in \mathbb{R}$, the* mining game reward *is the real-valued function r_λ from states B and B' to*

$$\mathrm{r}_\lambda(B,B') := (1-\lambda)r^1(B,B') - \lambda r^2(B,B'). \tag{11}$$

Remark: For any states X_t, X_{t+1}, and X_{t+2}, the mining game reward function satisfy the identity

$$\mathrm{r}_\lambda(X_t,X_{t+1}) + \mathrm{r}_\lambda(X_{t+1},X_{t+2}) = \mathrm{r}_\lambda(X_t,X'_{t+2}). \tag{12}$$

Definition C.3 (Value Function). *The* objective function *for mining game $(X_t)_{t\geq 0}$ is a real-value function $\mathcal{V}_\pi^\lambda$ from state B to*

$$\mathcal{V}_\pi^\lambda(B) := \mathbb{E}\left[r_\lambda(X_0,X_\tau)\middle|X_0=B\right] = \mathbb{E}\left[r_\lambda(X_0,X_1) + \mathcal{V}_\pi^\lambda(X_1)\cdot \mathbb{1}_{\tau\neq 1}|X_0=B\right], \tag{13}$$

the expected game reward from a mining game starting at state $X_0 = B$ and stopping at state X_τ where $\tau \geq 1$ is the first time step Miner 1 capitulates to state B_0. Define the value function *$\mathcal{V}$ as the real-valued function from state B to*

$$\mathcal{V}(B) := \mathcal{V}_{\pi^*}^{\lambda^*}(B) \tag{14}$$

where $\lambda^* = \max_\pi \textsc{Rev}(\pi)$ and π^* is an optimal positive recurrent strategy.

First, they observed that by setting $\lambda^* = \max_\pi \textsc{Rev}(\pi)$, any positive recurrent strategy π that maximizes $\mathcal{V}_\pi^{\lambda^*}(B_0)$ is an optimal positive recurrent strategy.

Lemma C.4. *Let π^* be an optimal positive recurrent strategy and let $\lambda^* = \textsc{Rev}(\pi^*)$. Then $\pi^* \in \arg\max_\pi \mathcal{V}_\pi^{\lambda^*}(B_0)$ and $\mathcal{V}(B_0) = 0$.*

Proof. Given two positive recurrent strategies π and π', the following claim allow us to compare their revenue.

Claim C.5. *For positive recurrent strategies π and $\tilde{\pi}$,*

(i) $\mathcal{V}_\pi^\lambda(B_0) = 0$ *if and only if* $\lambda = \textsc{Rev}(\pi)$.

(ii) $\textsc{Rev}(\pi) > \textsc{Rev}(\tilde{\pi})$ *if and only if* $\mathcal{V}_\pi^{\textsc{Rev}(\tilde{\pi})}(B_0) > 0$.

(iii) $\textsc{Rev}(\pi) < \textsc{Rev}(\tilde{\pi})$ *if and only if* $\mathcal{V}_\pi^{\textsc{Rev}(\tilde{\pi})}\pi(B_0) < 0$.

Proof. Recall τ denotes the first time step Miner 1 capitulates to state B_0, Equation 5, and let $R^k = \sum_{t=1}^{\tau} r^k(X_{t-1}, X_t)$. Observe

$$\mathcal{V}_\pi^\lambda(B_0) = \mathbb{E}\left[(1-\lambda)R^1 - \lambda R^2 | X_0 = B_0\right].$$

From Theorem C.1,

$$\frac{\mathbb{E}\left[R^1\right]\mathbb{E}\left[R^2\right]}{\mathbb{E}\left[R^1\right]+\mathbb{E}\left[R^2\right]} = \textsc{Rev}(\pi)\mathbb{E}\left[R^2\right] \quad \text{and} \quad \frac{\mathbb{E}\left[R^1\right]\mathbb{E}\left[R^2\right]}{\mathbb{E}\left[R^1\right]+\mathbb{E}\left[R^2\right]} = (1-\textsc{Rev}(\pi))\mathbb{E}\left[R^1\right].$$

Taking the difference and setting $\lambda = \textsc{Rev}(\pi)$,

$$\mathcal{V}_\pi^{\textsc{Rev}(\pi)}(B_0) = \mathbb{E}\left[(1-\textsc{Rev}(\pi))R^1 - (1-\textsc{Rev}(\pi))R^2\right] = \frac{\mathbb{E}\left[R^1\right]\mathbb{E}\left[R^2\right]}{\mathbb{E}\left[R^1\right]+\mathbb{E}\left[R^2\right]} - \frac{\mathbb{E}\left[R^1\right]\mathbb{E}\left[R^2\right]}{\mathbb{E}\left[R^1\right]+\mathbb{E}\left[R^2\right]} = 0.$$

This proves (i). Rewrite $\mathcal{V}_\pi^\lambda(B_0)$ as

$$\mathcal{V}_\pi^\lambda(B_0) = \mathbb{E}\big[R^1\big] - \lambda\mathbb{E}[R^1 + R^2].$$

Note that $\mathbb{E}[R^1 + R^2]$ is the expected height of the block tree at time τ. Since $\tau \geq 1$ and Miner 2 mines a block with probability $1-\alpha > 0$, the expected height of the block tree at time τ is at least $1-\alpha > 1/2$. Thus $\mathcal{V}_\pi^\lambda(B_0)$ is strictly decreasing function of λ. This proves (ii), and (iii). □

Thus if π^* is an optimal positive recurrent strategy with $\lambda^* = \textsc{Rev}(\pi^*)$, we directly obtain $\mathcal{V}(B_0) = \mathcal{V}_{\pi^*}^{\lambda^*}(B_0) = 0$. For any positive recurrent strategy π, $\mathcal{V}_\pi^{\lambda^*}(B_0) \leq \mathcal{V}(B_0) = 0$. Therefore, $\pi^* \in \arg\max_\pi \mathcal{V}_\pi^{\lambda^*}(B_0)$. □

Thus if π maximizes $\mathcal{V}_\pi^{\lambda^*}(B_0)$ (Equation 13), then π must maximize $\mathcal{V}_\pi^{\lambda^*}(X_t)$ for all subsequent states X_t. As a corollary, we recover the well known Bellman's principle of optimality.

Lemma C.6 (Bellman's Principle of Optimality)**.** *For all states B, for all positive recurrent strategies π, $\mathcal{V}(B) \geq \mathcal{V}_\pi^{\max_\pi \textsc{Rev}(\pi)}(B)$.*

Proof. Let π^* be an optimal positive recurrent strategy with $\lambda^* = \textsc{Rev}(\pi^*)$. Let $(X_t^\pi)_{t\geq 0}$ be a mining game starting from state $X_0 = B_0$ when Miner 1 follows strategy π. From Definition C.3,

$$\mathcal{V}(B_0) = \mathcal{V}_{\pi^*}^{\lambda^*}(B_0) = \mathbb{E}\left[r_{\lambda^*}(B_0, X_1^{\pi^*}) + V(X_1^{\pi^*}) | X_0^{\pi^*} = B_0\right],$$

and from Lemma C.4, $\mathcal{V}(B_0) = \max_\pi \mathcal{V}_\pi^{\lambda^*}(B_0) = 0$. Let $\pi(X_t)$ be the action π takes at state $X_t^{\textsc{Half}}$ and let $\pi : \pi(X_t)$ be all strategies conditioned on π taking action $\pi(X_t)$ at state $X_t^{\textsc{Half}}$. Note X_t^π depends only on actions taken up to time t. Thus

$$\begin{aligned}
\mathcal{V}(B_0) &= \max_\pi \mathbb{E}\left[r_{\lambda^*}(B_0, X_1^\pi) + \mathcal{V}_\pi^{\lambda^*}(X_1^\pi) | X_0^\pi = B_0\right] \\
&= \max_{\pi(X_0)} \mathbb{E}\left[r_{\lambda^*}(B_0, X_1^{\pi(X_0)}) + \max_{\pi:\pi(X_0)} \mathcal{V}_\pi^{\lambda^*}(X_1^{\pi(X_0)}) | X_0^{\pi(X_0)} = B_0\right] \\
&= \max_{\pi(X_0)} \mathbb{E}\left[r_{\lambda^*}(B_0, X_1^{\pi(X_0)}) + \mathcal{V}(X_1^{\pi^*}) | X_0^{\pi(X_0)} = B_0\right].
\end{aligned}$$

The last line observes the expected value takes its maximum by taking action $\pi(X_0) = \pi^*(X_0)$. Thus $\mathcal{V}(B) = \max_\pi \mathcal{V}_\pi^{\lambda^*}(B) \geq \mathcal{V}_\pi^{\lambda^*}(B)$ for all positive recurrent π and states B. ☐

We will use Lemma C.6 to witnesses when a particular action is optimal for a particular state B. That is, we can guess a particular strategy π takes an optimal action at state B and assume the optimal strategy π^* takes a distinct action. By deriving $\mathcal{V}_\pi^{\lambda^*}(B) \geq \mathcal{V}(B)$, we conclude that π's action at state B is indeed optimal.

D Omitted Proofs from Section 3

First, we compute the expected number of blocks Miner 1 creates from the moment it reaches state $X_2 = B_{2,0}$ until they capitulate to state B_0. For that, we will define a coupling between the mining game $X_2, X_3, \ldots, X_\tau$ with a biased one-dimensional random walk.

Lemma D.1. *Let $(N_t)_{t\geq 0}$ be a biased one-dimensional random walk with initial state $N_0 \in \mathbb{Z}$ and for $t \geq 1$, $N_t = N_{t-1} + 1$ with probability α; otherwise, $N_t = N_{t-1} - 1$. Let $\tau = \min\{t \geq 1 : N_t = N_0 - 1\}$. We say the state N_t increments if $N_{t+1} = N_t + 1$ and decrements if $N_{t+1} = N_t - 1$. Let $X = \sum_{t=1}^{\tau} \mathbb{1}_{N_t = N_{t-1}+1}$ be the number of state increments up to time τ. Similarly let $Y = \sum_{t=1}^{\tau} \mathbb{1}_{N_t = N_{t-1}-1}$ be the number of state decrements up to time τ. Then*

$$\mathbb{E}[X] = \frac{\alpha}{1-2\alpha} \qquad \mathbb{E}[Y] = \frac{1-\alpha}{1-2\alpha} \qquad \mathbb{E}[\tau] = \frac{1}{1-2\alpha}.$$

Proof. For the case $N_1 = N_0 - 1$ and $X = 0$, $Y = 1$ (with probability $1 - \alpha$). For the case $N_1 = N_0 + 1$ (with probability α), the expected number of state increments until the Markov process first returns to state N_0 is equals to $\mathbb{E}[X]$. Once the Markov process first returns to state N_0, the expected number of time steps until the Markov process first reaches state $N_0 - 1$ is also $\mathbb{E}[X]$. Thus

$$\begin{aligned}
\mathbb{E}[X] &= \alpha(1 + \mathbb{E}[X] + \mathbb{E}[X]) \\
&= \alpha + 2\alpha\mathbb{E}[X].
\end{aligned}$$

Solving for $\mathbb{E}[X]$ proves $\mathbb{E}[X] = \frac{\alpha}{1-2\alpha}$. A similar argument proves $\mathbb{E}[Y] = \frac{1-\alpha}{1-2\alpha}$. Since $\tau = X + Y$, linearity of expectation proves $\mathbb{E}[\tau] = \frac{1}{1-2\alpha}$ as desired. $\square$

Lemma D.2. *Let $X_0 = B_{2,0}$ and let $\tau = \min\{t \geq 1 : |T_1(X_\tau)| = |T_2(X_\tau)| + 1\}$. Then the expected number of blocks Miner 1 creates from time 1 to τ is $\mathbb{E}[|T_1(X_\tau)| - 1] = \frac{\alpha}{1-2\alpha}$. Moreover, the expected number of blocks Miner 2 creates from time 1 to τ is $\frac{1-\alpha}{1-2\alpha}$.*

Proof. Define the biased one-dimensional random walk $Y_t = |T_1(X_t)| - |T_2(X_t)| - 1$ for $t \geq 0$. Then τ is the first time step where $Y_\tau = 0$ (observe $Y_0 = 1$). The random variable $|T_1(X_t)| - 2$ counts the number of time steps where $Y_t = Y_{t-1} + 1$ for $t \leq \tau$. Thus from Lemma D.1, the expected number of blocks Miner 1 creates from time 1 to τ is $\mathbb{E}[|T_1(X_t)| - 2] = \frac{\alpha}{1-2\alpha}$ and the expected number of blocks Miner 2 creates from time 1 to τ is $\frac{1-\alpha}{1-2\alpha}$ as desired. $\square$

Proof of Theorem 3.2. Let $(X_t)_{t\geq 0}$ be a mining game starting at state $X_0 = B_0$ where Miner 1 uses the SM strategy (Definition 3.1). Let $\tau \geq 1$ be the first time step where Miner 1 capitulates to state B_0. If Miner 1 did not capitulate by time step $t = 3$, it is because $X_2 = B_{2,0}$. In this case, $\tau = |T_1(X_\tau)| + |T_2(X_\tau)| = 2|T_1(X_\tau)| - 1$. From Lemma D.2, $\mathbb{E}[\tau] < 3 + \frac{2\alpha}{1-2\alpha}$. Thus SM is positive recurrent and from Lemma C.4,

$$0 = \mathcal{V}^{\lambda}_{\mathrm{SM}}(B_0) = \mathbb{E}[\mathrm{r}_\lambda(B_0, X_\tau)|X_0 = B_0]$$

where $\lambda = \textsc{Rev}(\mathrm{SM})$. To compute this quantity, we consider the following events:

- When the game reaches state $B_{0,1}$, Miner 1 capitulates to state B_0. Miner 2 owns one block in the longest path. Thus the reward is $\mathrm{r}_\lambda(B_0, X_\tau) = -\lambda$.
- When the game reaches state $B_{2,0}$, Miner 1 wait until the first time step τ where $|T_2(X_\tau)| = |T_1(X_\tau)| - 1$, then Miner 1 publishes blocks $T_1(X_\tau)$ and capitulates to state B_0. Miner 1 owns $|T_1(X_\tau)|$ blocks in the longest path. From Lemma D.2, the expected reward is

$$\mathbb{E}[\mathrm{r}_\lambda(B_0, X_\tau)|X_2 = B_{2,0}] = \mathbb{E}[T_1(X_\tau)|X_2 = B_{2,0}](1-\lambda) = \left(2 + \frac{\alpha}{1-2\alpha}\right)(1-\lambda).$$

- When the game reaches state $B_{1,1}$, we consider two cases:
  - Miner 1 creates block 3, publishes $3 \to 1 \to 0$ and capitulates to state B_0. Miner 1 owns two blocks in the longest path. Thus the reward is $2(1-\lambda)$.
  - Miner 2 publishes $3 \to 2$, then Miner 1 capitulates to state B_0. Miner 2 owns two blocks in the longest path. Thus the reward is -2λ.

Substituting in the equation above,

$$0 = \mathcal{V}^{\lambda}_{\mathrm{SM}}(B_0) = \alpha^2\left(2 + \frac{\alpha}{1-2\alpha}\right)(1-\lambda) + 2\alpha^2(1-\alpha)(1-\lambda) - 2\alpha(1-\alpha)^2\lambda - (1-\alpha)\lambda.$$

Solving for λ and observing $\lambda = \textsc{Rev}(\mathrm{SM})$ proves the theorem. $\square$

Proof of Theorem 3.4. Consider the mining game starting at state $X_0 = B_0$. Let $\tau \geq 1$ be the first time step Miner 1 capitulates to state B_0. Let's first check the Markov chain induced by the NSM strategy (Definition 3.3) is positive recurrent. If Miner 1 did not capitulate by time step 2, $X_2 \in \{B_{2,0}, B_{1,1}\}$. For the case $X_2 = B_{2,2}$, $\tau = |T_1(X_\tau)| + |T_2(X_\tau)| = 2|T_1(X_\tau)| - 1$. From Lemma D.2, $\mathbb{E}[\tau | X_2 = B_{2,2}] = 3 + \frac{2\alpha}{1-2\alpha} < \infty$. For the case $X_2 = B_{1,1}$, the probability that the Markov process returns to state B_0 before returning to state $B_{1,1}$ is $\alpha + (1-\alpha)^2 + \alpha^2(1-\alpha)$. Thus $\mathbb{E}[\tau | X_2 = B_{1,1}] = 2 + \frac{1}{\alpha + (1-\alpha)^2 + \alpha^2(1-\alpha)} < \infty$. Thus the Markov chain is positive recurrent and from Lemma C.4,

$$0 = \mathcal{V}^{\lambda}_{\text{NSM}}(B_0) = \mathbb{E}[\mathrm{r}_\lambda(B_0, X_\tau) | X_0 = B_0]$$

where $\lambda = \textsc{Rev}(\text{NSM})$. To compute the expected value above, we consider the following events:

- When $X_1 = B_{0,1}$, Miner 1 capitulates to state B_0. Miner 2 owns one block in the longest path. Thus the reward is $-\lambda$.
- When $X_2 = B_{2,0}$, Miner 1 waits until the first time step $\tau \geq 3$ where $|T_2(X_\tau)| = |T_1(X_\tau)| - 1$, then Miner 1 publishes blocks $T_1(X_\tau)$ and capitulates to state B_0. Miner 1 owns $|T_1(X_\tau)|$ blocks in the longest path at time τ. From Lemma D.2, the expected reward is

$$\mathbb{E}[\mathrm{r}_\lambda(B_0, X_\tau) | X_2 = B_{2,0}] = \mathbb{E}[T_1(X_\tau) | X_2 = B_{2,0}](1-\lambda) = \left(2 + \frac{\alpha}{1-2\alpha}\right)(1-\lambda).$$

- When the game reaches state X_t equivalent to state $B_{1,1}$, there is four scenarios:
  - Miner 1 creates block 3, then publishes $3 \to 1 \to 0$ and capitulates to state B_0. Miner 1 owns two blocks in the longest path. Thus the reward is $2(1-\lambda)$.
  - Miner 2 publishes $4 \to 3 \to 2$, then Miner 1 capitulates to state B_0. Miner 2 owns three blocks in the longest path. Thus the reward is -3λ.
  - Miner 2 publishes $3 \to 2$ while Miner 1 creates block 4 and 5. Then Miner 1 publishes $5 \to 4 \to 1 \to 0$ and capitulates to state B_0. Miner 1 owns three blocks in the longest path. Thus the reward is $3(1-\lambda)$.
  - Miner 2 publishes $5 \to 3 \to 2$ while Miner 1 creates and withholds block 4. Then Miner 1 capitulates to state $B_{1,1}$ allowing Miner 2 to stay with blocks 2 and 3. Thus the reward is -2λ (repeat until Miner 1 capitulates to state B_0).

From the work above, the reward when $X_2 = B_{1,1}$ is

$$\mathbb{E}[\mathrm{r}_\lambda(X_0, X_\tau) | X_2 = B_{1,1}] = \sum_{i=0}^{\infty} (\alpha(1-\alpha)^2)^i (2\alpha(1-\lambda) - 3(1-\alpha)^2\lambda + 3\alpha^2(1-\alpha)(1-\lambda) - 2i\lambda).$$

To get a closed form, recall the geometric series (for $x < 1$)

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x} \quad \text{and} \quad \sum_{i=1}^{\infty} i x^i = \frac{x}{(1-x)^2}.$$

Substituting in the equation above, we get

$$0 = \mathcal{V}^{\lambda}_{\text{NSM}}(B_0) = \alpha^2\left(2 + \frac{\alpha}{1-2\alpha}\right)(1-\lambda) - (1-\alpha)\lambda$$
$$+ \frac{\alpha(1-\alpha)}{1-\alpha(1-\alpha)^2}\left(2\alpha(1-\lambda) - 3(1-\alpha)^2\lambda + 3\alpha^2(1-\alpha)(1-\lambda) - \frac{2\alpha(1-\alpha)^2}{1-\alpha(1-\alpha)^2}\lambda\right).$$

Solving for λ and observing $\lambda = \textsc{Rev}(\text{NSM})$ proves Theorem 3.4. □

E Omitted Proofs from Section 4

E.1 Timeserving Reduction

We will now define a reduction that takes as input a strategy π, and produces a new strategy $\tilde{\pi}$. The strategy $\tilde{\pi}$ will *simulate* π, which we clarify below.

Definition E.1 (Simulating a strategy)**.** *A strategy $\tilde{\pi}$ may wish to* simulate *π, and take actions based on this simulation. Specifically, we will add a subscript of $_\pi$ to random variables like $\textsc{Tree}, V, \mathcal{C}$, etc. to denote the state of this simulation. Formally, X_π denotes "what the state of variable X would have been, had Miner 1 been using strategy π all along (with the same γ)".*

If there is no subscript of π, then these variables retain their original meaning (and refer to the state when Miner 1 uses their actual strategy $\tilde{\pi}$).

Definition E.2 (Timeserving Reduction)**.** *For any strategy π, define the* Timeserving Reduction *of π to take the following action during round n:*

- *If $A(\mathcal{C}_\pi(n)) \cap \mathcal{U}(n) = \emptyset$, then Wait. That is, if the longest chain from simulating π is already published, don't publish further.*
- *Else, PublishSet$(A(\mathcal{C}_\pi(n)) \cap \mathcal{U}(n), E')$, where E' contains all pointers leaving $A(\mathcal{C}_\pi(n)) \cap \mathcal{U}(n)$ in $E_\pi(n)$. That is, if the longest chain from simulating π is not yet published, publish whatever part of the chain is not yet published.*

Essentially, the Timeserving Reduction of π simply holds off on publishing blocks which are not yet in the longest chain, and only publishes them when necessary to support a longest chain. Below, note that the conclusion of Theorem 4.2 does not necessarily hold against *arbitrary* strategies for Miner 2, but it does hold against FRONTIER (so we must necessarily use properties of FRONTIER in the proof).

Proof of Theorem 4.2. Let $\tilde{\pi}$ denote the Timeserving Reduction of π. Here is the main idea: all actions taken by the FRONTIER strategy *only depend on the longest chain, and not on what other nodes are published.* This is formally stated in the following observation.

Observation E.3. *Let $\pi, \tilde{\pi}$ be two strategies such that when used against* FRONTIER*, with probability* 1*, for all n, $\mathcal{C}_\pi(n) = \mathcal{C}_{\tilde{\pi}}(n)$. Then in every round* FRONTIER *takes identical actions against π and $\tilde{\pi}$.*

It is clear that π and $\tilde{\pi}$ have the same longest chain at the end of each round, by definition of $\tilde{\pi}$. Specifically, $\tilde{\pi}$ will never publish a block that has not yet been published by π, and it makes sure the longest chain under π is always published. Therefore, FRONTIER will take the same actions against both. Because FRONTIER is taking the same actions, and the longest chain is the same at the end of each round, this guarantees that $\textsc{Rev}(\pi) = \textsc{Rev}(\tilde{\pi})$.

Finally, it is clear that $\tilde{\pi}$ is Timeserving. The longest chain at the end of round n is indeed $\mathcal{C}_\pi(n)$, and $\tilde{\pi}$ only publishes ancestors of $\mathcal{C}_\pi(n)$ (if it publishes at all). □

Proof of Observation 4.3. Proof of (i). Ancestors of the longest chain form a single path. If π does not publish a single path, then it is not possible for all of these nodes to immediately be ancestors of the longest chain.

Proof of (ii). Suppose for contradiction there is some state where $\textsc{Tree}$ has two distinct leaves $q \neq \tilde{q}$ with the same height, and consider when this first happens. First, q and $\tilde{q}$ can't have been published by the same action by Property (i). If they are published during distinct actions, then w.l.o.g. let q be published before $\tilde{q}$ (and therefore the most recent action to update $\textsc{Tree}$ must have published $\tilde{q}$).

Because q has the same height as $\tilde{q}$ and was published first, $\tilde{q}$ is not in $\mathcal{C}$. Also, because $\tilde{q}$ is a leaf, it cannot be in $A(\mathcal{C})$ unless it is $\mathcal{C}$ itself. Therefore, no Timeserving strategy could have published $\tilde{q}$ (because Miner 1 is Timeserving, and FRONTIER only publishes a longest chain no matter what), a contradiction.

Proof of (iii). Suppose for contradiction Miner 1 forks by publishing a single block q on top of r. If Miner 1 can fork the longest chain with a single block, then there must have previously been two leaves of the same height, a contradiction to (ii). □

E.2 Orderly Reduction

Example E.4. *Consider the game in Figure 8 where Miner 1 creates blocks 1, 2, 3, 4 and 5. Looking at strategy π on the left, observe that the action PublishPath$(\{4, 5\}, 0)$ taken in round* 5 *is not Orderly. Instead, if an Orderly action is to publish two blocks pointing to* 0 *in round* 5*, it must publish* $2 \to 1 \to 0$. *This initially suggests a local fix: simply swap the roles of* 1 *and* 4*, and also* 2 *and* 5*. This switch is shown in the center.*

Unfortunately, there is a problem: while it is certainly safe to publish $1 \to 0$ *instead of* $4 \to 0$ *(this makes it only easier to build later blocks, as* $1 < 4$*), it is* not safe *to publish* 4 *where* 1 *used to be. Indeed, π previously used the action PublishPath$(\{1\}, 0)$ in round* 6*, and then the action PublishPath$(\{3, 6\}, 1)$ in round* 7*. Simply swapping all the* 1*s and* 4*s results in an infeasible action in round* 7 *because the edge* $3 \to 4$ *is invalid.*

Instead, what we want is to apply this reduction ad infinitum. Indeed, the key observation is that the action PublishPath$(\{4\}, 0)$ is itself not Orderly, and should instead be changed to PublishPath$(\{3\}, 0)$, as is done in $\tilde{\pi}$ on the right of Figure 8. The same reduction should again be applied in round 7*.*

This example illustrates our reduction (defined formally in Definition E.5), but also establishes the importance of doing a reduction "all at once" rather than a sequence of local changes.

We now provide our Orderly Reduction, which generalizes the ideas in Example E.4.

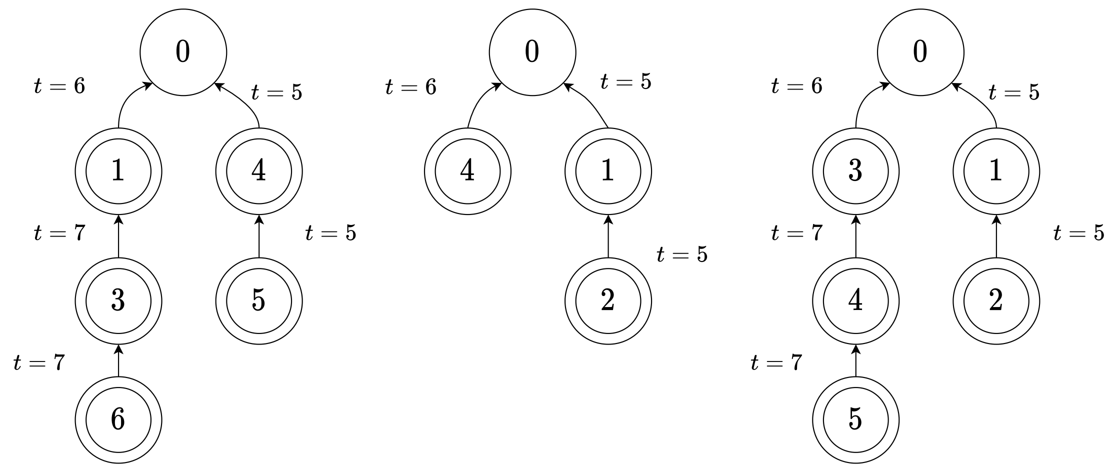

Figure 8: Mining game with a strategy π (left) where the action in round 5 is not Orderly. An intermediate "reduction" which is infeasible (center), and a transformed strategy $\tilde{\pi}$ (right) where all actions are feasible and Orderly.

Definition E.5 (Orderly Reduction)**.** *Let π be any timeserving strategy. The* Orderly Reduction *of π is the strategy $\tilde{\pi}$ which does the following:*

- *Maintain a mapping $\sigma : \mathbb{N}_+ \to \mathbb{N}_+$. Initialize $\sigma(b) = b$ for all $b \in \mathbb{N}_+$.*
- *During round $n+1$, if π takes action Wait from state $\text{TREE}_\pi^{\text{HALF}}(n)$, then Wait. That is, if π waits in the simulation, then also wait.*
- *Else, if π takes the action PublishPath(V', u) from state $\text{TREE}_\pi^{\text{HALF}}(n)$, let $V'' := \min^{|V'|}(\mathcal{U} \cap (\sigma(u), \infty))$. Take the action PublishPath($V'', \sigma(u)$), and make the following updates to σ:*
  - *For all $v \in V'$, if v is the ℓ^{th} smallest element of V', update $\sigma(v)$ to be the ℓ^{th} smallest element of V''. That is, update σ so that it maps newly-published blocks under π to newly-published blocks under $\tilde{\pi}$, and is monotone increasing within this domain.*
  - *For all $v \in (\mathcal{U})_\pi$, if v is the ℓ^{th} smallest element of $(\mathcal{U})_\pi$, update $\sigma(v)$ to be the ℓ^{th} smallest element of $\mathcal{U}$. That is, update σ so that it maps still-unpublished blocks under π to still-unpublished blocks under $\tilde{\pi}$, and is monotone increasing within this domain.*

Intuitively, this reduction hard-codes that every PublishPath($\cdot, \cdot$) action is Orderly, and uses $\sigma(\cdot)$ to remember which "new blocks" after the reduction correspond to the "old blocks" before the reduction. Specifically, the intent of σ is to be an isomorphism between TREE_π and TREE, and to also map the ℓ^{th} smallest element of $(\mathcal{U})_\pi$ to the ℓ^{th} smallest element of $\mathcal{U}$ at all times. Note that the second bullet updating $\sigma(\cdot)$ is simply to maintain $\sigma(\cdot)$ as a useful helper for proofs, and not necessary to actually execute the reduction.

Observe that, by definition, whenever $\tilde{\pi}$ takes a PublishPath($\cdot, \cdot$) action, that action is Orderly (as long as it is valid). So our only task is to confirm that every action is valid, and also that the payoffs are identical against FRONTIER. Observe further that the only way the PublishPath$(\cdot,\cdot)$ action could be invalid is if there simply aren't $|V'|$ elements of $\mathcal{U}_i \cap (\sigma(u), \infty)$: all elements of $(\sigma(u), \infty)$ clearly come after $\sigma(u)$, so the action would otherwise be valid.

We first prove a helper lemma, which describes how σ evolves over time. In this lemma (and subsequent proofs), it will be helpful to introduce the following notation. Observe that every time a PublishPath action is taken by Miner 1, that the function $\sigma(\cdot)$ changes. In order to cleanly reference "the state of $\sigma(\cdot)$, before the action was taken", we use the notation $\sigma_{\text{old}}(\cdot)$. In order to cleanly reference "the state of $\sigma(\cdot)$, after the action was taken", we use the notation $\sigma_{\text{new}}(\cdot)$. More generally, we will add a subscript of $_{\text{old}}$ to reference the state of a variable before a referenced action is taken, and $_{\text{new}}$ to reference the state of a variable after a referenced action is taken.

Lemma E.6. *Consider any action PublishPath(V', u) action taken by Miner 1 using π. If it further holds that for all $v \in V'$, $\sigma_{old}(v) > \sigma_{old}(u)$, then for all v, the following hold:*

- *If v is already published, then $\sigma_{new}(v) = \sigma_{old}(v)$.*
- *If $v \in V'$, then $\sigma_{new}(v) \leq \sigma_{old}(v)$.*
- *If $v \notin V'$, then $\sigma_{new}(v) \geq \sigma_{old}(v)$.*

Proof. Observe that bullet one clearly holds, as we don't change $\sigma(v)$ for any v which is already published. The two interesting bullets will follow from the following observation: *If for all $v \in V'$, $\sigma_{old}(v) > \sigma_{old}(u)$, then the set $\sigma_{old}(V')$ is valid to publish on top of $\sigma_{old}(u)$.* Because V'' is the smallest $|V'|$ elements which are valid to publish on top of $\sigma_{\text{old}}(u)$, this immediately implies: For all $v \in V'$, if v is the ℓ^{th} smallest element of V', then the ℓ^{th} smallest element of V'' is at most $\sigma_{\text{old}}(v)$. As $\sigma_{\text{new}}(v)$ is exactly the ℓ^{th} smallest element of V'', we conclude bullet two.

Now that we have bullet two, bullet three readily follows. Consider when v was previously the ℓ^{th} smallest element of $((\mathcal{U})_\pi)_{\text{old}}$. If there are k elements smaller than v in V', then v is now the $(\ell - k)^{th}$ smallest element of $((\mathcal{U})_\pi)_{\text{new}}$. If there are $\geq k$ elements smaller than $\sigma_{\text{old}}(v)$ in V'', then the $(\ell - k)^{th}$ smallest element of $(\mathcal{U})_{\text{new}}$ is *at least as large* as the ℓ^{th} smallest element of $(\mathcal{U})_{\text{old}}$ (which is exactly $\sigma_{\text{old}}(v)$). Observe, however, that there are exactly k elements smaller than $\sigma_{\text{old}}(v)$ in $\sigma_{\text{old}}(V')$. By bullet two, this means that there are indeed at least k elements smaller than $\sigma_{\text{old}}(v)$ in V'', and we conclude bullet three.

□

We use this technical lemma to conclude the following, which describes how TREE_π and TREE evolve isomorphically over time.

Corollary E.7. *At all points in time, the following hold:*

- *The graphs TREE_π and TREE are isomorphic. Specifically:*
  - *For all $v \in V_\pi$, $\sigma(v)$ is published by $\tilde{\pi}$ during the same round that v is publishedby π, and these are the only nodes in V.*
  - *For all $(u, v) \in E_\pi$, $(\sigma(u), \sigma(v)) \in E$, and these are the only edges in E.*

- *$\sigma(\cdot)$ is injective and surjective (from $\mathbb{N}_+$ onto $\mathbb{N}_+$).*
- *For all v created by Miner 2, $\sigma(v) = v$.*
- *For any $v \notin V_\pi$, and any $u \in \mathbb{N}_+$, $v > u \Rightarrow \sigma(v) > \sigma(u)$.*

Proof. We'll prove the claim by induction. Clearly at initialization, all claims hold because both $\textsc{Tree}_\pi$ and $\textsc{Tree}$ contain just the genesis block and $\sigma(b) = b$ for all b.

Now assume that all four claims hold prior to an action being taken (in the execution with π) for inductive hypothesis, and we will establish that all four claims continue to hold after the action is taken. Clearly, if that action is Wait, then the execution with $\tilde{\pi}$ is also wait, and neither the graphs nor σ change, so the claim still holds.

If that action is a PublishPath action taken by Miner 2, it will publish a single block r on top of the longest chain $\mathcal{C}_\pi$. Observe that by inductive hypothesis, specifically bullet one, that $\mathcal{C} = \sigma(\mathcal{C}_\pi)$. Therefore, Miner 2 will take the action $\text{PublishPath}(\{r\}, \mathcal{C}) := \text{PublishPath}(\{r\}, \sigma(\mathcal{C}_\pi))$. This maintains that σ maps $\textsc{Tree}_\pi$ to $\textsc{Tree}$. Because no changes are made to σ, bullets two, three, and four continue to hold.

The interesting case to consider is if the action taken is PublishPath $V'u$ by Miner 1 using π. Then the action taken by $\tilde{\pi}$ is $\text{PublishPath}(V'', \sigma(u))$. Because we update $\sigma(V') := V''$ upon publishing (while being monotone increasing), this would guarantee that the first bullet continues to hold, *if we can guarantee that the action PublishPath$(V'', \sigma(u))$ is valid.* However, bullet four readily lets us claim this. Indeed, because $\text{PublishPath}(V', u)$ is valid for π, we have both: (a) all $v \in V'$ are not in $(V_\pi)_{\text{old}}$ and (b) $v > u$ for all $v \in V'$. Bullet four immediately yields that $\sigma_{\text{old}}(v) > \sigma_{\text{old}}(u)$ for all $v \in V'$, meaning that there are at least $|V'|$ nodes which can be published on top of $\sigma_{\text{old}}(u)$, and therefore $\text{PublishPath}(V'', \sigma_{\text{old}}(u))$ is valid. This establishes that bullet one continues to hold.

Now we just need to confirm that the three bullets regarding $\sigma(\cdot)$ continue to hold. It is easy to see that σ remains injective and surjective. It is also easy to see that σ remains unchanged for any blocks created by Miner 2, so bullet three continues to hold. It is also easy to verify that bullet four continues to hold *when both v and u are unpublished.* This is simply because $\sigma(\cdot)$ maps the ℓ^{th} smallest unpublished element under π to the ℓ^{th} smallest unpublished element under $\tilde{\pi}$. If u is published, then we will use Lemma E.6.

Indeed, the fact that bullet four previously held lets us conclude that $v > u \Rightarrow \sigma_{\text{old}}(v) > \sigma_{\text{old}}(u)$ prior to the action. By Lemma E.6, $\sigma_{\text{new}}(v) \geq \sigma_{\text{old}}(v)$ (if v was not published, and therefore $\notin (V_\pi)_{\text{new}}$). By the same Lemma E.6, $\sigma_{\text{new}}(u) \leq \sigma_{\text{old}}(u)$ if u is published by the action (or was already published). Therefore, we get the following chain of inequalities (the outer two are by Lemma E.6, the middle inequality is by inductive hypothesis).

$$v \notin (V_\pi)_{\text{new}} \text{ AND } u \in (V_\pi)_{\text{new}} \text{ AND } v > u \Rightarrow \sigma_{\text{new}}(v) \geq \sigma_{\text{old}}(v) > \sigma_{\text{old}}(u) \geq \sigma_{\text{new}}(u),$$

as desired. We have now completed the inductive step, and shown that if all four properties hold prior to an action, they all hold after that action as well. Because they all hold at initialization, they hold at all times. □

Proof of Orderly Theorem 4.6. The proof readily follows from bullet one of Corollary E.7. Indeed, because $\textsc{Tree}_\pi$ and $\textsc{Tree}$ are isomorphic, and because $\sigma(b)$ is published by $\tilde{\pi}$ during

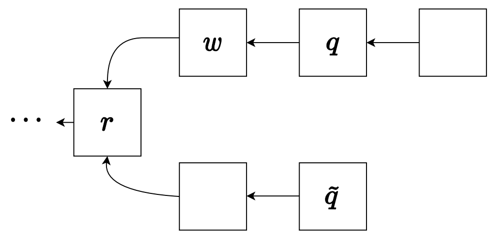

Figure 9: Instance example for the Fork Ownership Lemma. Empty squares represent blocks that are irrelevant to the claim. Lemma E.10 states that if Miner 1 plays a strategy which is Timeserving and LCM, and q is a block in the longest chain, and there is a block $\tilde{q}$ with the same height as q, then Miner 1 must have created q and w (but perhaps not r, and perhaps not $\tilde{q}$).

the same action that π publishes b for all b, it's always the case that $\mathcal{C} = \sigma(\mathcal{C}_\pi)$. The fact that they are isomorphic also implies that every action taken by $\tilde{\pi}$ is valid), and also that for all n, γ: $\text{Rev}_\gamma^{(n)}(\pi) = \text{Rev}_\gamma^{(n)}(\tilde{\pi})$. □

E.3 Longest Chain Mining Reduction

As in the previous sections, we devise a reduction from any strategy which is Timeserving, and Orderly, but not LCM, to one which is Timeserving, Orderly, and LCM.

Definition E.8 (LCM Reduction). *Let π be any Timeserving, Orderly strategy. The LCM Reduction of π for Player 1 is the strategy $\tilde{\pi}$ which does the following:*

- *During round $n+1$, if π takes action Wait from state $\text{Tree}_\pi^{\text{Half}}(n)$, then Wait. That is, if π waits in the simulation, then also wait.*
- *Else, if π takes the action Publish(k, u) from state $\text{Tree}_\pi^{\text{Half}}(n)$, let v be the unique block in $A(\mathcal{C})$ with $h_\pi(u) = h(v)$. Take the action Publish(k, v).*

In bullet two above, it should be clear that the desired block v is unique *if it exists.* It is not immediately obvious that v exists (but we will prove that it does).

The following observation will be useful, both in proving Theorem 4.9, and in drawing conclusions from it.

Observation E.9. *Let π be any LCM strategy. Then if v is ever published, but not in the longest chain, v will never again enter the longest chain (formally, if $v \in V$ has $v \notin A(\mathcal{C})$, then $v \notin A(\mathcal{C})$ for the rest of the game).*

Proof. If $v \notin A(\mathcal{C})$, then all of v's descendants are also not in $A(\mathcal{C})$. Therefore, no LCM action can point to v or any of its descendants, and the longest chain will never contain v. □

Next, we prove a helper lemma stating that Miner 1 must have created all blocks in the longest chain *since the most recent fork.* Figure 9 provides an example illustrating the claims of the lemma. The intuition is as follows: if Miner 1 is using a Timeserving, LCM strategy, and $\tilde{q}$ was published, then $\tilde{q}$ was at some point in the longest chain. Therefore, Miner 2 (using FRONTIER) would not publish elsewhere (so if a new longest chain is created, it must have come entirely from Miner 1).

Lemma E.10 (Fork Ownership Lemma)**.** *Let π be any Timeserving, LCM strategy. Let $q \in A(\mathcal{C})$ be a block in the longest chain, and let $\tilde{q} \in V$ be another block of the same height (formally: $\tilde{q} \neq q, h(\tilde{q}) = h(q)$). If $r \in A(\mathcal{C})$ is the least common ancestor of $\tilde{q}$ and q, then Miner 1 created all blocks between r and q (including q, not necessarily including r). Formally, $A(\mathcal{C}) \cap (r, q] \subseteq T_1$.*

Proof. Observe that if we can prove the lemma when $\tilde{q}$ is a leaf, then we will have in fact proved the lemma for all $\tilde{q}$ (for arbitrary $\tilde{q}$, simply pick any leaf $\tilde{q}_2$ which is a descendent of $\tilde{q}$, and apply the lemma statement with q_2, the node in $A(\mathcal{C})$ with $h(q_2) = h(\tilde{q}_2)$. The lemma statement for $q_2, \tilde{q}_2$ implies the lemma statement for $q, \tilde{q}$). So we proceed assuming that $\tilde{q}$ is a leaf.

Let now $\tilde{Q} := A(\tilde{q}) \cap (r, \tilde{q}]$ denote the ancestors of $\tilde{q}$ (including $\tilde{q}$, up to but not including r). Let $Q := A(q) \cap (r, q]$ denote the same for q. Recall that as r is the least common ancestor of $q, \tilde{q}$, these two sets are disjoint.

Observe before continuing that because Miner 1 is Timeserving (and Miner 2 is using FRONTIER, which is also Timeserving), and that $\tilde{q}$ is a leaf, that $\tilde{q}$ is the longest chain immediately following the action in which it is published. In particular, this means that $\tilde{q}$ is published before q.

Our goal is to prove that Miner 1 created every block in Q, so assume for contradiction that there exists a $w \in Q$ which was created by Miner 2. First, observe that w clearly was not published during the same action as $\tilde{q}$, as FRONTIER publishes only a single block at a time (and $w \in Q$, while $\tilde{q} \notin Q$, so $w \neq \tilde{q}$).

Observe also that w cannot have been published *before* $\tilde{q}$. Indeed, if w were published before $\tilde{q}$, then immediately after $\tilde{q}$ is published, w is both published and not in the longest chain. By Observation E.9, w would never again be in the longest chain, contradicting that q becomes the longest chain.

Finally, we show that w cannot be published by an action *after* $\tilde{q}$. Indeed, once $\tilde{q}$ is published, no element of $Q \cup \{r\}$ can possibly be the longest chain (because $\tilde{q}$ is longer), and therefore FRONTIER would publish on top of $\tilde{q}$ rather than any block in $Q \cup \{r\}$. Therefore, Miner 2 could not possibly have published a block w in Q (which necessarily would point to a block in $Q \cup \{r\} \setminus \{q\}$) after $\tilde{q}$ was already published.

In summary, we have shown that no block $w \in Q$ could have been created by Miner 2 during the same round that $\tilde{q}$ was published (because Miner 2 is using FRONTIER), after $\tilde{q}$ was published (also because Miner 2 is using FRONTIER), or before $\tilde{q}$ was published (because Miner 1 and Miner 2 are both LCM). Therefore, Miner 2 cannot have created any blocks $w \in Q$, and Miner 1 must have created them all, completing the proof. $\square$

From here, a complete proof of Theorem 4.9 becomes unwieldy to do entirely in one shot (like Theorem 4.6). Instead, we will complete the proof by changing π one action at a time, and interleaving these changes with the Orderly Reduction. Specifically, we will need the following definitions:

Definition E.11 (N-LCM and N-Orderly)**.** *A strategy is N-LCM if, when played against* FRONTIER*, every action it takes up to and including round N is LCM (with probability* 1*). A strategy is N-Orderly if, when played against* FRONTIER*, every action it takes up to and including round N is Orderly.*

Observe that a strategy is LCM/Orderly if and only if it is N-LCM/N-Orderly for all N.

Towards Theorem 4.9, we will now define a simpler one-step reduction. Importantly, note that we return to the language PublishPath instead of simply Publish.

Definition E.12 (Step-N LCM Reduction)**.** *Let π be any Timeserving, N-Orderly, N-LCM strategy. The Step-N LCM Reduction of π for Player* 1 *is the strategy $\tilde{\pi}$ which does the following:*

- *During round any round $\neq N+1$, if π takes the action Wait, then also Wait. If π takes the action PublishPath(V', u), take action PublishPath(V', u).*
- *During round $N+1$, if π takes action Wait from state $\text{Tree}_\pi^{\text{Half}}(N)$, then Wait. That is, if π would have waited, then also wait.*
- *Else, if π takes the action PublishPath(V', u) from state $\text{Tree}_\pi^{\text{Half}}(n)$, let v be the unique block in $A(\mathcal{C})$ with $h_\pi(u) = h(v)$. That is, let v be the block of height $h_\pi(u)$ within the longest chain of $\text{Tree}^{\text{Half}}(n)$. Take the action PublishPath(V', u).*

Importantly, observe that the Step-N Reduction only changes the action taken in step $N+1$, and in a very simple way: it tries to publish exactly the same blocks, just on top of a different node. This makes it significantly simpler to prove claims about validity. See Figure 10 for an example.

Lemma E.13. *Let π be any Timeserving, N-Orderly, N-LCM strategy, and let $\tilde{\pi}$ be its Step-N LCM Reduction. Then $\tilde{\pi}$ is Timeserving, N-Orderly, $(N+1)$-LCM, and takes a valid action during every step. Moreover, for all γ, n: $\text{Rev}_\gamma^{(n)}(\tilde{\pi}) \geq \text{Rev}_\gamma^{(n)}(\pi)$.*

Proof. Observe, importantly, that $\tilde{\pi}$ takes identical actions to π *except during round $N+1$*, and moreover that during round $N+1$ π *attempts to publish the exact same path at the exact same height* (just perhaps at a different location). This means that all the desired properties immediately follow if we can prove that the action taken in round $N+1$ is valid. Indeed, as long as $\tilde{\pi}$ can publish the same path (just perhaps in a different location), this implies that the longest chain will be the same (just perhaps with different ancestors), and therefore that Miner 2 using FRONTIER will use the same action in every future round, and therefore that all actions taken by Miner 1 will be Timeserving and valid in future rounds. It is also clear that if the action taken by $\tilde{\pi}$ is valid, then it is LCM. Also note that we have *not* claimed that $\tilde{\pi}$ is $(N+1)$-Orderly.

So, to confirm that the action taken during round $N+1$ is valid, we just need to confirm that whenever π tries to publish a set of nodes V' on top of u (and this is valid), that it is also valid to publish the same V' on top of v (where $v \in A(\mathcal{C})$ has the same height as u). Note that if we knew that $v \leq u$, the claim would be trivial, but we might have $v > u$.

To this end, we use Lemma E.10 and the fact that π is N-Orderly. Let r denote the least common ancestor of v and u. Then whenever $v \neq u$, Lemma E.10 asserts that Miner 1

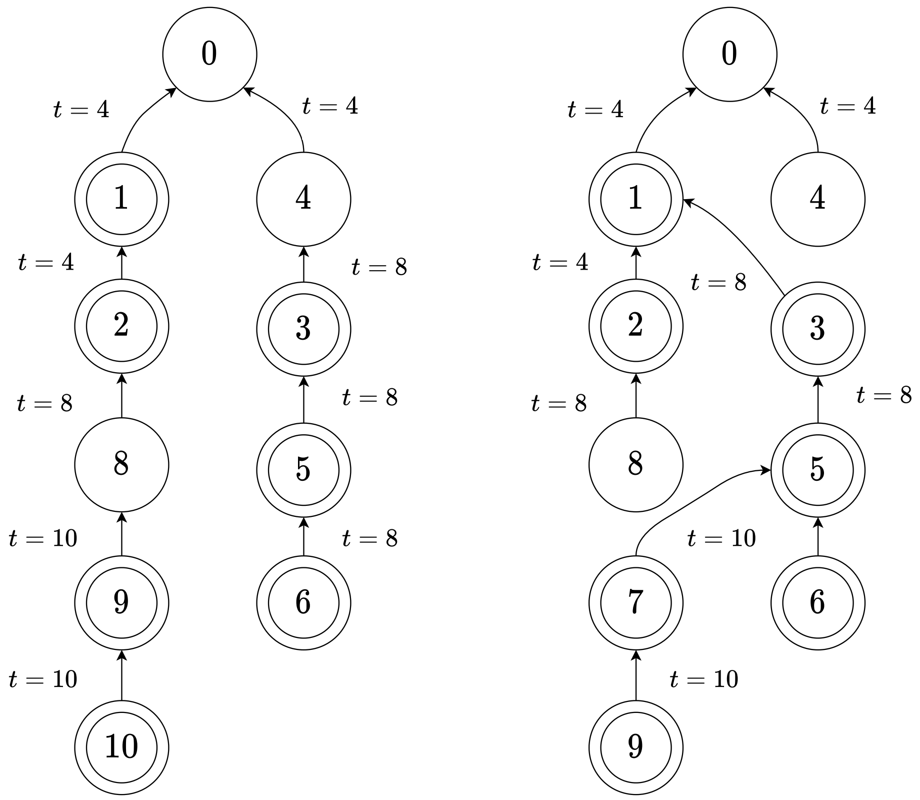

Figure 10: On the left is the history for a Timeserving and Orderly strategy π. In the middle is the Step-9 LCM Reduction of π, which is not 10-Orderly. The history on the right further applies the Orderly Reduction.

published not only v, but the entire path from v to r (not necessarily including r). Moreover, because π is N-Orderly, there must not be any blocks in $\mathcal{U}(N) \cap (r, v]$ (otherwise, π must have previously taken a non-Orderly action). But now, as $u > r$, this immediately implies that any $V' \subseteq \mathcal{U}(N)$ which is valid to publish on top of u is also valid to publish on top of v. Specifically, because $\mathcal{U}(N) \cap (r, v] = \emptyset$, and $V' \subseteq \mathcal{U}(N)$, and $w > u > r$ for all $w \in V'$, $w > v$ for all $w \in V'$ as well.

This concludes the first half of Lemma E.13: we have just shown in the previous path that the action taken in round $N + 1$ is valid. The previous paragraphs establish that this implies that all actions taken in all rounds are valid, Timeserving, and LCM. Because no actions are changed during the first N rounds, $\tilde{\pi}$ is still N-Orderly. The remaining step is to confirm that for all γ, n: $\textsc{Rev}_{\gamma}^{(n)}(\tilde{\pi}) \geq \textsc{Rev}_{\gamma}^{(n)}(\pi)$.

To see this, observe we have just argued that $\mathcal{C}_{\pi}(n) = \mathcal{C}(n)$ for all n. The only difference is that perhaps $A(\mathcal{C}_{\pi}(n)) \neq A(\mathcal{C}(n))$. In fact, the only potential difference is that perhaps $A(\mathcal{C}_{\pi}(n))$ contains the path $(r, u]$, while $A(\mathcal{C}(n))$ contains instead the path $(r, v]$, but they must otherwise be the same. Fortunately, Lemma E.13 also asserts that Miner 1 created *every node* in $(r, u]$. Therefore, we have both that $|A(\mathcal{C}(n))| = |A(\mathcal{C}_{\pi}(n))|$ for all n, and also that $|A(\mathcal{C}(n)) \cap T_1| \geq |A(\mathcal{C}_{\pi}(n)) \cap T_1|$ for all n. This directly implies that $\textsc{Rev}_{\gamma}^{(n)}(\tilde{\pi}) \geq \textsc{Rev}_{\gamma}^{(n)}(\pi)$ for all γ, n. □

Now, we can complete the proof of Theorem 4.9.

Proof of Longest Chain Mining Theorem 4.9. The proof will boil down to showing that we can alternate between the Step-N LCM Reduction and the Orderly Reduction to implement the LCM reduction. Specifically, recursively define $\pi_0 := \pi$, and π_{N+1} to be the Orderly Reduction applied to the Step-N LCM Reduction applied to π_N. We make a series of observations.

Observation E.14. *For all N, π_N is Orderly, N-LCM, and Timeserving.*

Proof. We prove by induction. π_0 is clearly Orderly, 0-LCM, and Timeserving. Assume then that π_N is Orderly, N-LCM, and Timeserving. Then the Step-N LCM Reduction applied to π_N results in a strategy which is $(N + 1)$-LCM, and Timeserving. The Orderly Reduction applied to this strategy does not change where blocks are published (up to the isomorphism), but only which blocks are published. Therefore, the Orderly reduction preserves Timeserving and $(N + 1)$-LCM, but makes the strategy Orderly, as desired. □

Observation E.15. *For all N, and all $n \leq N + 1$, $\tilde{\pi}$ and π_N take the same action during round n.*

Proof. We also prove this by induction. The claim clearly holds for $N = 0$. Assume now that it holds for some N and we will prove it for $N+1$. π_N and π_{N+1} take the same actions for the first $N + 1$ rounds, so the inductive hypothesis immediately implies the desired conclusion for all $n \leq N + 1$. We just need to prove that the action taken in round $N + 2$ by $\tilde{\pi}$ and π_{N+1} are the same. Indeed, they will publish the same number of blocks, and at the same location. Because both are Orderly, they will publish the same set of blocks as well. □

By the two observations above, we immediately conclude that for all N, $\tilde{\pi}$ takes a valid action during round N, and is also N-LCM. Therefore, $\tilde{\pi}$ takes a valid action during every step, and is LCM. Also, it is clear that as long as $\tilde{\pi}$ takes a valid action during each timestep that $\tilde{\pi}$ is Timeserving and Orderly. □

E.4 Trimmed Strategies

Proof of Theorem 4.12. We can immediately apply Theorems 4.2, 4.6 and 4.9 to conclude that there exists a strategy $\hat{\pi}$ such that every non-Wait action it takes satisfies bullet (i) of being Trimmed. We just need to show that $\hat{\pi}$ can be modified to also satisfy bullet (ii).

So consider any action $\hat{\pi}$ takes which does not satisfy bullet (ii). This means that the action is of the form Publish(k, v), u is the unique node in $A(\mathcal{C})$ with an edge to v, but u was created by Miner 1. Let us further consider the entire set of descendents of v in $A(\mathcal{C})$ created by Miner 1. That is, let U denote the maximal path in $A(\mathcal{C})$, which terminates at v, where every node is created by Miner 1. By hypothesis that the taken action violates bullet (ii), we have that $|U| \geq 1$. Let w denote the maximal node in U (that is, the other end of the path).

We know that, because $\hat{\pi}$ is Timeserving, that $k > |U|$. Consider instead taking the action Publish$(k - |U|, w)$. Observe first that this action is valid, because $\hat{\pi}$ is Orderly. Indeed, not only can $k - |U|$ of Miner 1's unpublished nodes be placed on top of w, but the exact same set of k nodes which could be placed on top of u can be placed on w (if not, this would violate Orderly). Moreover, now that this action is valid, it is also Trimmed (because w does not have an immediate descendant in $A(\mathcal{C})$ created by Miner 1, by definition).

We just need to confirm that all future actions can be modified to be valid after this change, and that the reward to Miner 1 is the same. Intuitively, this occurs because the new longest chain has *exactly* the same height as the original, the owners of each block along the path are *exactly* the same as the original (i.e .we replaced one sub-chain of length $|U|$, all created by Miner 1 with U, another sub-chain of length $|U|$, all created by Miner 1), and those new blocks were only created *earlier* than the original blocks, making them easier to build upon.

Indeed, we claim that taking actions in all future rounds according to the LCM reduction of $\hat{\pi}$ results in feasible actions and identical payoffs. Indeed, the reward immediately after this action is swapped does not change. In all future rounds, because every block in U was created before every block originally published, every future action is still valid. And moreover, the rewards of all future rounds are also identical. Because this holds pointwise, for all rounds, we can apply the same reduction ad infinitum to swap all non-Trimmed actions for Trimmed actions without changing the payoff at all. □

F Omitted Proofs from Section 5

F.1 Checkpoints

Proof of Lemma 5.6. Observe a miner following the FRONTIER strategy never publishes two blocks at the same height. This implies that $h(\mathcal{C}(X_n))$ is at least the number of blocks mined by Miner 2 (note that these blocks are not necessarily themselves *in* the longest path, but *some* block of the same height must be). Thus $h(\mathcal{C}(X_n)) \geq \sum_{i=1}^{n} \mathbb{1}_{i \in T_2}$. From the Strong Law of Large Numbers, $\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{i \in T_2} \overset{a.s.}{\to} 1-\alpha$. This proves Lemma 5.6. $\square$

Proof of Corollary 5.7. For the lower bound, we use FRONTIER to witness that $\textsc{Rev}(\pi) \geq \alpha$ because $\textsc{Rev}(\text{FRONTIER}) = \alpha$. If both miners follow the FRONTIER strategy, all blocks Miner 1 and Miner 2 creates are part of the longest path. Thus $\textsc{Rev}_{\gamma}^{(n)}(\text{FRONTIER}) = \frac{\sum_{i=1}^{n} \mathbb{1}_{i \in T_1}}{n}$. From the Strong Law of Large Numbers, $\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{i \in T_1} \overset{a.s.}{\to} \alpha$.

For the upper bound, we claim $\limsup_{n\to\infty} \textsc{Rev}_{\gamma}^{(n)}(\pi) \leq \frac{\alpha}{1-\alpha}$, for any strategy π, with probability 1. Observe the number of blocks Miner 1 owns in the longest path is at most the number of blocks Miner 1 creates. Moreover, Miner 2 never publishes two blocks at the same height implying the height of the longest chain is at least the number of blocks Miner 2 creates. Then

$$\textsc{Rev}_{\gamma}^{(n)}(\pi) \leq \frac{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{i \in T_1}}{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{i \in T_2}}.$$

From the Strong Law of Large Numbers, $\frac{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{i \in T_1}}{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{i \in T_2}} \overset{a.s.}{\to} \frac{\alpha}{1-\alpha}$ as desired. $\square$

Proof of Proposition 5.8. Proof of (i). The proof is clear from recursively applying the checkpoint definition.

Proof of (ii). Assume the converse. If P_i is the highest checkpoint bellow v, v is a checkpoint, a contradiction.

Proof of (iii). Let $P_k = \min\{P_j : P_j > v\}$. Then

$$\begin{aligned}
|A(\mathcal{C}) \cap (v, P_k]| &= |A(\mathcal{C}) \cap (P_{k-1}, P_k]| - |A(\mathcal{C}) \cap (P_{k-1}, v]| \\
&> |\mathcal{U} \cap (P_{k-1}, P_k]| - |\mathcal{U} \cap (P_{k-1}, v]| \\
&= |\mathcal{U} \cap (v, P_k].
\end{aligned}$$

The second line follows from (i) and (ii). The inequality above and (i) implies

$$\begin{aligned}
|A(\mathcal{C}) \cap (v, P_i] \cap T_1| &= |A(\mathcal{C}) \cap (v, P_k] \cap T_1| + |A(\mathcal{C}) \cap (P_k, P_i] \cap T_1| \\
&> |\mathcal{U} \cap (v, P_k]| + |\mathcal{U} \cap (P_k, P_i]| \\
&= |\mathcal{U} \cap (v, P_i]|
\end{aligned}$$

as desired. $\square$

F.2 Checkpoint Preserving Reduction

The next example highlights the main ideas behind the proof.

Example F.1. *In Figure 11, consider a mining game where Miner 1 creates and withhold blocks 2 and 3, and Miner 2 creates block 1 and publishes* $1 \to 0$. *Then Miner 1 publishes* $3 \to 2 \to 0$ *during the third round forking the checkpoint* $P_1 = 1$. *To transform the original strategy into a checkpoint preserving strategy, observe that a new strategy could publishes* $3 \to 2 \to 1$ *instead of* $3 \to 2 \to 0$. *Note Miner 1 has an advantage of two blocks over Miner*

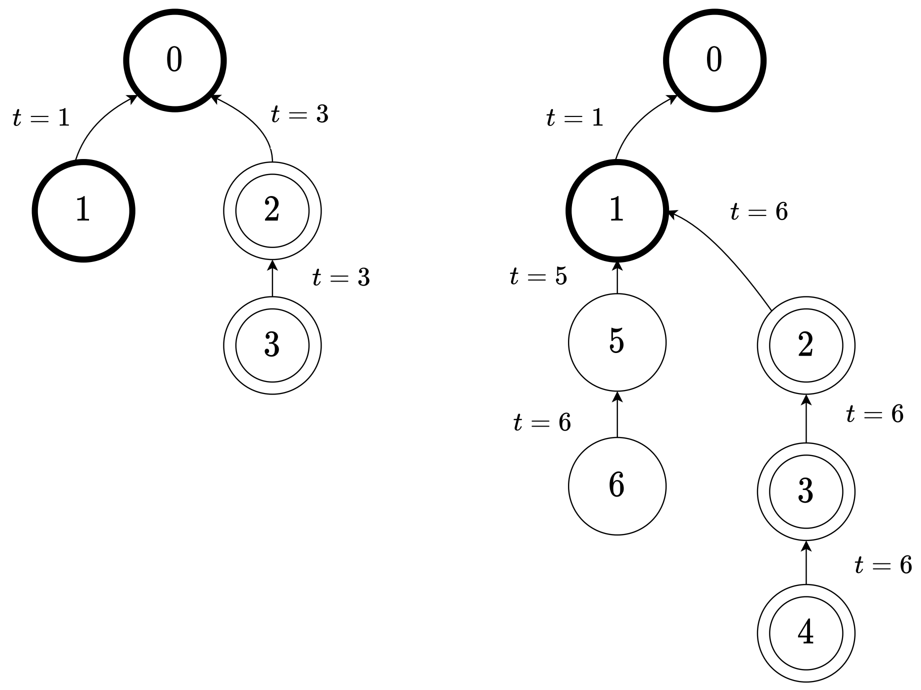

Figure 11: The mining game from non-checkpoint preserving (left) and transformation to a checkpoint preserving strategy (right). Double circles denote blocks created by Miner 1 and single circles denote blocks created by Miner 2. Thicker blocks denote blocks that were checkpoints at some point in time.

2 so there is no risk into waiting to publishing $3 \to 2 \to 1$*. Therefore, to further improve the new strategy, let the new strategy wait until the first time step where Miner 1's advantage reduces to a single block. For example, if Miner 1 creates and withhold block* 4 *and Miner 2 creates blocks 5 and 6 and publishes* $6 \to 5 \to 1$*, Miner 1 can safely publish* $4 \to 3 \to 2 \to 1$ *forking blocks 5 and 6 during the sixth round.*

Remark. Note it is not at all obvious the transformation in Example F.1 guarantees the new strategy is at least as good as the original one. It is true the number of blocks Miner 1 owns in the longest path increases; however, the number of blocks Miner 2 owns in the longest path also increases. Thus the main challenge in proving Theorem 5.12 is showing that on expectation, the payoff of the new strategy is at least as good as the payoff of the original one.

To show the transformed strategy in Example F.1 is at least as good as the initial strategy, we will show that whenever Miner 1 takes PublishPath(Q, v) (at time t) and forks a checkpoint $c \in A(\mathcal{C})$,

- PublishPath$(Q \cap (c, \infty), c)$ is a valid trimmed action Miner 1 can take instead.
- Regardless if Miner 1 takes action PublishPath(Q, v) or PublishPath$(Q \cap (c, \infty), c)$, max Q will becomes a checkpoint in the subsequent state.

By taking PublishPath$(Q \cap (c, \infty), c)$ instead of PublishPath(Q, v), we show

- The number of blocks Miner 1 does not publish $|Q \cap (0, c)|$ is at most the number of blocks created by Miner 1 that Miner 1 does not fork $|A(\mathcal{C}) \cap (v, c] \cap T_1|$. Thus Miner 1 receives at least the same reward by taking PublishPath$(Q \cap (c, \infty), c)$ instead of PublishPath(Q, v).

- If Miner 2's reward increases by $k = |A(\mathcal{C}) \cap (v, c] \cap T_2|$ when Miner 1 does not fork $\textsc{Succ}(v) \cap (v, c)$, then Miner 1 has a lead of $k+1$ blocks. That is

  $$|Q \cap (c, \infty)| \geq |\textsc{Succ}(c)| + k + 1$$

  Thus instead of taking action PublishPath$(Q \cap (c, \infty), c)$ at time t, Miner 1 can safely wait until the first time step

  $$\tau = \min\{t' \geq t : |T_2 \cap (t, t']| = |T_1 \cap (t, t']| + k\}$$

  where Miner 2 creates k more blocks than Miner 1 (since time $t+1$). At time τ, we let Miner 1 take PublishPath$((Q \cap (c, \infty)) \cup (T_1 \cap (t, t']), c)$ forking $\textsc{Succ}(c) \cup (T_2 \cap (t, t'])$. We show the expected number of blocks Miner 1 creates from time $t+1$ to τ, outweighs the cost of allowing Miner 2 to walk away with k additional blocks at time t.

F.2.1 Preliminaries

Next, we define a transformation of trimmed action PublishPath(Q, v) that forks a checkpoint $c \in A(\mathcal{C})$ into another trimmed action PublishPath(Q', v') that does not forks a checkpoint (with $Q' \subseteq Q$ and $v' > v$). Recall PublishPath(Q, v) is trimmed if it is timeserving – i.e., Q becomes part of the longest path which requires $|Q| \geq |\textsc{Succ}(v)| + 1$.

Definition F.2 (Lift). *For valid, timeserving PublishPath(Q, v) action, let $Q' \subseteq Q$ and $v' \in \textsc{Succ}(v)$. If PublishPath(Q', v') is timeserving, then PublishPath(Q', v') is a* lift *of PublishPath(Q, v).*

Definition F.3 (Safe Lift). *PublishPath(Q', v') is a* safe lift *of PublishPath(Q, v) if PublishPath(Q', v') is a* lift *of PublishPath(Q, v) and taking action PublishPath(Q', v') gives at least the same reward as taking action PublishPath(Q, v). That is, $|(\textsc{Succ}(v) \setminus \textsc{Succ}(v')) \cap T_1| \geq |Q \setminus Q'|$.*

Definition F.4 (Checkpoint Lift). *Let PublishPath(Q, v) be a trimmed action and assume $\textsc{Succ}(v)$ contains a checkpoint. Let $c_v :=$ max $\cup\{P_i > v : P_i \in A(\mathcal{C})$ is a checkpoint$\}$ be the most recent checkpoint. Define PublishPath$(Q \cap (c_v, \infty), c_v)$ as the* checkpoint lift *of PublishPath(Q, v).*

Lemma F.5 (Checkpoint Lift Lemma). *Suppose Miner 1's strategy is trimmed. If PublishPath(Q', v') is the* checkpoint lift *of trimmed action PublishPath(Q, v), then PublishPath(Q', v') is trimmed and a* safe lift *of PublishPath(Q, v).*

Proof. By assumption, $\textsc{Succ}(v)$ contains a checkpoint and v' be the most recent checkpoint in $\textsc{Succ}(v)$. To show that PublishPath(Q', v') is trimmed we need to show that

1. If v' has any successors, min $\textsc{Succ}(v')$ was created by Miner 2.

2. PublishPath(Q', v') is timeserving.

For (1), suppose Miner 1 created min $\textsc{Succ}(v')$. Then min $\textsc{Succ}(v')$ is a checkpoint because Miner 1's strategy is orderly, a contradiction to v' being the most recent checkpoint. This proves (1). For (2), suppose for contradiction PublishPath(Q', v') is not timeserving. Then

$$\begin{aligned}
|Q| &= |Q \cap (v, v']| + |Q \cap (v', \infty)| \\
&\le |\mathcal{U} \cap (v, v']| + |Q \cap (v', \infty)| && \text{Because PublishPath}(Q, v) \text{ is orderly.} \\
&\le |\mathcal{U} \cap (v, v']| + |\textsc{Succ}(v')| && \text{From the assumption PublishPath}(Q', v') \text{ is not timeserving.} \\
&\le |A(\mathcal{C}) \cap (v, v']| + |\textsc{Succ}(v')| && \text{From Proposition 5.8 and the fact } v' \text{ is a checkpoint.} \\
&= |\textsc{Succ}(v)| \Rightarrow\Leftarrow .
\end{aligned}$$

The chain of inequalities contradicts the assumption that PublishPath(Q, v) is timeserving. This proves (2) and proves PublishPath(Q', v') is trimmed. Next, we show PublishPath(Q', v') is a safe lift. Because v' is a checkpoint,

$$\begin{aligned}
|(\textsc{Succ}(v) \setminus \textsc{Succ}(v')) \cap T_1| &= |A(\mathcal{C}) \cap (v, v'] \cap T_1| \\
&\ge |\mathcal{U} \cap (v, v']| && \text{From Proposition 5.8.} \\
&\ge |Q \cap (v, v']| && \text{Because PublishPath}(Q, v) \text{ is orderly.} \\
&= |Q \setminus Q'| && \text{Because } Q' = Q \cap (v', \infty).
\end{aligned}$$

The chain of inequalities witnesses PublishPath(Q', v') is a safe lift of PublishPath(Q, v). $\square$

Lemma F.6 (Checkpoint Override Lemma)**.** *Let PublishPath(Q, v) be a trimmed action. If $\{v\} \cup \textsc{Succ}(v)$ contains a checkpoint, the new longest chain becomes a checkpoint after Miner 1 plays PublishPath(Q, v).*

Proof. We divide the proof into two cases:

Case 1. If v is a checkpoint, all blocks Miner 1 publishes become checkpoints because PublishPath(Q, v) is orderly.

Case 2. If v is not a checkpoint, let $s = \max\{c \in A(v) : c \text{ is a checkpoint}\}$ be the most recent checkpoint smaller than v and let $c = \min\{c \in \textsc{Succ}(v) : c \text{ is a checkpoint}\}$ be the oldest checkpoint bigger than v. Let $t \in Q$ be the block that would take the height of c in the longest path once $Q \to v$ is published. It suffices to show that t becomes a checkpoint because if t becomes a checkpoint, max Q becomes a checkpoint because PublishPath(Q, v) is orderly (i.e., $(\mathcal{U} \setminus Q) \cap (t, \max\ Q) = \emptyset$).

Because $s < v$, s will continue to be a checkpoint and to show that t becomes a checkpoint, it suffices to show that $|(A(\mathcal{C}) \cap (s, v] \cap T_1) \cup (Q \cap (v, t])| \ge |(\mathcal{U} \setminus Q) \cap (s, t]|$. Indeed,

$$\begin{aligned}
|(A(\mathcal{C}) \cap (s, v] \cap T_1) \cup (Q \cap (v, t])| &\ge |A(\mathcal{C}) \cap (s, c] \cap T_1| \\
&\ge |\mathcal{U} \cap (s, c]| && \text{Because } s \text{ and } c \text{ are checkpoints.} \\
&\ge |(\mathcal{U} \setminus Q) \cap (s, c]| \\
&\ge |(\mathcal{U} \setminus Q) \cap (s, t]|
\end{aligned}$$

The first line observes t will have the same height as c and Miner 1 *will be the creator of all blocks in the path $v \to \ldots \to t$*. For the fourth line, the case where $t < c$ is clear. For the case where $t > c$, we claim $(\mathcal{U} \setminus Q) \cap (c,t) = \emptyset$. If there exists $q \in (\mathcal{U} \setminus Q) \cap (c,t)$, $q > v$ because v is ancestor of c. Thus $q < t$ is a block Miner 1 could have published pointing to v instead of t, a contradiction to PublishPath(Q,v) being orderly. This proves t (and $\max\ Q$) becomes a checkpoint as desired. □

F.2.2 The Reduction

In this section, we proof Theorem 5.12.

Definition F.7 (Potential reward)**.** *For a state B, let B' be a* successor *of B if there is a valid action Miner 1 can take at state B such that the subsequent state is B'. Let $\textsc{Succ}_s(B)$ be the set of all successors of B. The* potential reward *δ^k of Miner k maps a state B to the maximum absolute reward Miner k can obtain from state B to a successor of B. That is*

$$\delta^k(B) = \max_{B' \in \textsc{Succ}_s(B)} |\,\mathrm{r}^k(B,B')|.$$

Proposition F.8. *Let π be a trimmed strategy. Let B be a state satisfying the bullets:*

- *B is reachable from a mining game starting at state B_0 with Miner 1 following strategy π.*
- *At B, π takes trimmed action PublishPath(Q,v) forking the most recent checkpoint P_i.*
- *Block v reached finality with respect to π.*

Then there exists a valid, trimmed strategy $f(\pi)$ where $f(\pi)$ take the same actions as π up to state B and forks no checkpoint at state B. To define $f(\pi)$ at state B onward, let $(X_t^{f(\pi)})_{t\geq 0}$ be a mining game starting at $X_0^{f(\pi)} = B$ and where Miner 1 follows strategy $f(\pi)$. Let $N = \max\ \mathrm{V}(B)$ and

$$Q_t = |Q \cap (P_i, \infty)| + |T_1(X_t^{f(\pi)}) \cap (N, N+t]|,$$

$$Z_t = |\,\textsc{Succ}^B(P_i)| + |T_2(X_t^{f(\pi)}) \cap (N, N+t]| \qquad \textit{and} \qquad W_t = Q_t - Z_t - 1.$$

Let $\tau = \min\{t \geq 0 : W_t = 0\}$. At state $(X_t^{\textsc{Half}})^{f(\pi)}$, for $t < \tau$, let $f(\pi)$ wait. At state $(X_\tau^{\textsc{Half}})^{f(\pi)}$, let $f(\pi)$ take action

$$\textit{PublishPath}(Q \cap (P_i, \infty) \cup T_1(X_\tau^{f(\pi)}) \cap (N, N+\tau], P_i)$$

forking blocks $\textsc{Succ}^B(P_i) \cup T_2(X_t^{f(\pi)}) \cap (N, N+\tau]$. To define $f(\pi)$'s actions in future states, let B' be the subsequent state to B after π publishes blocks Q. Let $(X_t)_{t\geq\tau}$ be a mining game starting at $X_\tau = B'$ and where Miner 1 follows strategy π. Complete the sequence $(X_t)_{t\geq 0}$ with

$$X_0 = B \qquad \textit{and} \qquad X_1 = X_2 = \ldots = X_{\tau-1} = B'.$$

Let $(X_t, X_t^{f(\pi)})_{t\geq\tau}$ be a coupling where Miner k creates block $t \geq \tau + 1$ in both games. For all $t \geq \tau + 1$, at state $(X_t^{\textsc{Half}})^{f(\pi)}$, let $f(\pi)$ take the same actions π takes at state $X_t^{\textsc{Half}}$.

Given the coupling $(X_t, X_t^{f(\pi)})_{t\geq 0}$*, for all* $t \geq 0$*,*

$$\textsc{Rev}_\gamma^{(t)}(f(\pi)) \geq \frac{|A(\mathcal{C}(X_t)) \cap T_1| + \mathbb{1}_{t\geq\tau} \cdot \sum_{i=1}^{W_0} Z_i - \delta^1(X_t) \cdot \mathbb{1}_{1\leq t<\tau}}{h(\mathcal{C}(X_t)) + \mathbb{1}_{t\geq\tau} \cdot \left(W_0 + \sum_{i=1}^{W_0} Z_i\right) + \delta^1(X_t) \cdot \mathbb{1}_{1\leq t<\tau}}.$$

where $W_0 \geq 1$ *and* $(Z_i)_{i=1}^{W_0}$ *are i.i.d. random variables with expected value* $\frac{\alpha}{1-2\alpha}$*.*

Proof. First, we show $W_0 \geq 1$.

Claim F.9. $W_0 \geq |\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)| - |Q \cap (v, P_i]| \geq 1$.

Proof. Because PublishPath(Q, v) is timeserving, Q successfully becomes part of the longest path once π publishes Q which implies $|Q| \geq |\textsc{Succ}^B(v)| + 1$. Then

$$\begin{aligned} W_0 &= |Q \cap (P_i, \infty)| - |\textsc{Succ}^B(P_i)| - 1 \\ &= |Q| - |Q \cap (v, P_i]| - |\textsc{Succ}^B(P_i)| - 1 \\ &\geq |\textsc{Succ}^B(v)| - |\textsc{Succ}^B(P_i)| - |Q \cap (v, P_i]| \quad \text{Because PublishPath}(Q, v) \text{ is timeserving.} \\ &= |\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)| - |Q \cap (v, P_i]|. \end{aligned}$$

This proves the first inequality. To show the second inequality, recall PublishPath$(Q \cap (P_i, \infty), P_i)$ is the checkpoint lift of PublishPath(Q, v) (and thus a safe lift according to Lemma F.5), then, at state B, the number of blocks Miner 1 would add to the longest path by taking action PublishPath$(Q \cap (P_i, \infty), P_i)$ is at least the number of blocks Miner 1 would add to the longest path by taking action PublishPath(Q, v). That is

$$(\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)) \cap T_1| \geq |Q \cap (v, P_i]|. \qquad (15)$$

Note $\textsc{Succ}^B(v)$ is non-empty (since it contains a checkpoint). The fact π is trimmed implies the immediate successor of block v was created by Miner 2. Therefore,

$$|(\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)) \cap T_2| \geq 1.$$

Combining the work above

$$\begin{aligned} |\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)| - |Q \cap (v, P_i]| &\geq |\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)| \\ &\qquad - |(\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)) \cap T_1| \\ &= |(\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)) \cap T_2| \\ &\geq 1 \end{aligned}$$

The chain of inequalities witnesses $W_0 \geq 1$ as desired. □

Observe π never forks $\mathcal{C}(X_\tau)$. Suppose the converse. By assumption, v reached finality with respect to π. Thus if π forks $\mathcal{C}(X_\tau)$, π takes some action PublishPath(Q', v') where v' is in the path from v to $\mathcal{C}(X_\tau)$; however, the path from v to $\mathcal{C}(X_\tau)$ only contain blocks created by Miner 1 which implies the immediate successor of block v' was created by Miner

1, a contradiction to π being trimmed. This proves π never forks $\mathcal{C}(X_\tau)$ as desired. Next, observe

$$|\mathcal{U}(X_\tau) \cap (\mathcal{C}(X_\tau), \infty)| = |\mathcal{U}(X_\tau^{f(\pi)}) \cap (\mathcal{C}(X_\tau^{f(\pi)}), \infty)|.$$

Therefore, up to relabeling of blocks, states X_τ and $X_\tau^{f(\pi)}$ are equivalent with respect to trimmed strategy π. Thus it is valid for $f(\pi)$ to take the same actions as π after time τ.

Let's check $f(\pi)$ is valid and trimmed. By inspection, $f(\pi)$ is valid because π is valid. Up to state B, $f(\pi)$ is trimmed since $f(\pi)$ is equals to π up to state B. From first to $(\tau-1)$-th step, $f(\pi)$ publishes no blocks and at step τ, $f(\pi)$ publishes blocks pointing to P_i. If $\textsc{Succ}(P_i)$ is the empty-set, $f(\pi)$'s action is clearly trimmed. If $\textsc{Succ}(P_i)$ is non-empty, $\min\ \textsc{Succ}(P_i)$ is a block created by Miner 2. Suppose the contrary, then $\min\ \textsc{Succ}(P_i)$ existed at state B (since this is the first time $f(\pi)$ publishes blocks since state B). But B was reachable from a trimmed (and orderly) strategy which implies $\min\ \textsc{Succ}(P_i)$ is also a checkpoint, a contradiction to P_i being the most recent checkpoint at state B. Thus $f(\pi)$ is trimmed up to time τ. After time τ, $f(\pi)$ takes the same actions as π which, by assumption, is a trimmed strategy. This proves $f(\pi)$ is a trimmed strategy as desired.

Let's now compare the revenue of $f(\pi)$ with π. From the first to $(\tau-1)$-th step, π publishes Q pointing to v, but $f(\pi)$ publishes no blocks. During the same time interval, in $(X_t^{f(\pi)})_{t\geq 0}$, Miner 2 publishes all blocks created from time 1 to time $\tau-1$ while in $(X_t)_{t\geq 0}$, Miner 2 creates no blocks from time 1 to τ. Since $\delta^1(X_t^{f(\pi)})$ denotes the maximum reward Miner 1 can obtain from state X_t,

$$|A(\mathcal{C}(X_t^{f(\pi)})) \cap T_1| + \delta^1(X_t^{f(\pi)}) \geq |A(\mathcal{C}(X_t)) \cap T_1|.$$

$$h(\mathcal{C}(X_t^{f(\pi)})) - \delta^1(X_t^{f(\pi)}) \leq h(\mathcal{C}(X_t)).$$

At the τ-th step, $f(\pi)$ publishes $Q \cap (P_i, \infty) \cup T_1(X_\tau^{f(\pi)}) \cap (N, N+\tau]$ pointing to P_i and forking $\textsc{Succ}^B(P_i) \cup T_2(X_\tau^{f(\pi)}) \cap (N, N+\tau]$. At the same time, π publishes no blocks. Thus

$$\begin{aligned}
|A(\mathcal{C}(X_t^{f(\pi)})) \cap T_1| &= |A(\mathcal{C}(X_t)) \cap T_1| - |Q \cap (v, P_i]| + |\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i) \cap T_1| \\
&\quad + |T_1(X_t^{f(\pi)}) \cap (N, N+\tau]| \\
&\geq |A(\mathcal{C}(X_t)) \cap T_1| + |T_1(X_t^{f(\pi)}) \cap (N, N+\tau]| \qquad \text{From Equation 15.}
\end{aligned}$$

$$|A(\mathcal{C}(X_t^{f(\pi)})) \cap T_2| = |A(\mathcal{C}(X_t)) \cap T_2| + |(\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)) \cap T_2|.$$

Adding $|A(\mathcal{C}(X_t^{f(\pi)})) \cap T_1|$ and $|A(\mathcal{C}(X_t^{f(\pi)})) \cap T_2|$, the height of the longest path is

$$\begin{aligned}
h(\mathcal{C}(X_t^{f(\pi)})) &= |A(\mathcal{C}(X_t^{f(\pi)})) \cap T_1| + |A(\mathcal{C}(X_t^{f(\pi)})) \cap T_2| \\
&= h(\mathcal{C}(X_t)) + |\textsc{Succ}^B(v) \setminus \textsc{Succ}^B(P_i)| + |Q \cap (v, P_i]| + |T_1(X_t^{f(\pi)}) \cap (N, N+\tau]| \\
&\leq h(\mathcal{C}(X_t)) + W_0 + |T_1(X_t^{f(\pi)}) \cap (N, N+\tau]| \qquad \text{From Claim F.9.}
\end{aligned}$$

We will now derive a closed form for $|T_1(X_t^{f(\pi)}) \cap (N, N+\tau]|$.

Claim F.10. $|T_1(X_t^{f(\pi)}) \cap (N, N+\tau]| = \sum_{i=1}^{W_0} Z_i$ *where* $(Z_i)_{i=1}^{W_0}$ *are i.i.d. random variables with expected* $\frac{\alpha}{1-2\alpha}$.

Proof. Observe $(W_t)_{t\geq 0}$ is a one-dimensional biased random walk since W_t increments whenever Miner 1 creates a block (with probability α) and decrements whenever Miner 2 creates a block (with probability $1-\alpha$). That is,

$$W_t = \begin{cases} W_{t-1}+1 & \text{with probability } \alpha, \\ W_{t-1}-1 & \text{with probability } 1-\alpha. \end{cases}$$

For $i \geq 0$, let $\tau_i = \min\{t \geq 0 : W_t = W_0 - i\}$, then $\tau_{i+1} - \tau_i$ denotes the number of time steps the random walk takes to first reach state $W_0 - i$ since the first time step it reached stated $W_0 - i + 1$. Let $Z_{i+1} = |T_1(X_t^{f(\pi)}) \cap (N + \tau_i, N + \tau_{i+1}]|$ and note $Z_1, Z_2, \ldots$ are i.i.d. with

$$|T_1(X_t^{f(\pi)}) \cap (N, \tau_{W_0}]| = \sum_{i=1}^{W_0} |T_1(X_t^{f(\pi)}) \cap (N + \tau_{i-1}, N + \tau_i]| = \sum_{i=1}^{W_0} Z_i.$$

From Lemma D.1, $\mathbb{E}\left[|T_1(X_t^{f(\pi)}) \cap (N + \tau_i, N + \tau_{i+1}]\right] = \frac{\alpha}{1-2\alpha}$ as desired. □

Combining the inequalities above, we obtain that for all $t \geq 0$,

$$\begin{aligned} \textsc{Rev}_\gamma^{(t)}(f(\pi)) &= \frac{|A(\mathcal{C}(X_t^{f(\pi)})) \cap T_1|}{h(\mathcal{C}(X_t^{f(\pi)}))} \\ &\geq \frac{|A(\mathcal{C}(X_t)) \cap T_1| - \mathbb{1}_{t<\tau} \cdot \delta^1(X_t^{f(\pi)}) + \mathbb{1}_{t\geq\tau} \cdot \sum_{i=1}^{W_0} Z_i}{h(\mathcal{C}(X_t)) + \mathbb{1}_{t<\tau} \cdot \delta^1(X_t^{f(\pi)}) + \mathbb{1}_{t\geq\tau} \cdot \left(W_0 + \sum_{i=1}^{W_0} Z_i\right)} \end{aligned}$$

as desired. □

To simplify the expression for the revenue of $f(\pi)$, we first show $\delta^1(X_t)$ is a negligible term by deriving that $\frac{\delta^1(X_t)}{t} \overset{a.s.}{\to} 0$.

Lemma F.11 (Asymptotic Reward Lemma). *Let $(X_t)_{t\geq 0}$ be a mining game starting at state $X_0 = B_0$. For any $\epsilon > 0$, $Pr\left[\cup_{i=n}^{\infty}\{\delta^1(X_i) > i\epsilon\}\right] \leq e^{-\Omega(n)}$. Moreover,*

- $\frac{\delta^1(X_n)}{n} \overset{a.s.}{\to} 0$.
- $\frac{r^1(X_{n-1}, X_n)}{n} \overset{a.s.}{\to} 0$.

Proof. Event $\delta^1(X_i) > i\epsilon$ implies there is a timeserving action PublishPath(Q, v) Miner 1 can take at time step i with $|Q| > i\epsilon$. The fact PublishPath(Q, v) is timeserving implies Miner 1 creates more blocks than Miner 2 from time v to i. To derive an upper bound on v, observe at most i blocks were created by time i. Therefore, $|Q| + v \leq i$ which implies $v < (1-\epsilon)i$ since $|Q| > i\epsilon$. Thus

$$\begin{aligned} Pr\left[\cup_{i=n}^{\infty}\left\{\delta^1(X_i) > i\epsilon\right\}\right] &\leq \sum_{i=n}^{\infty} Pr\left[\delta^1(X_i) > i\epsilon\right] && \text{From union bound.} \\ &\leq \sum_{i=n}^{\infty} Pr\left[\exists v < (1-\epsilon)i, |T_1 \cap (v, i]| > |T_2 \cap (v, i]|\right] \\ &\leq \sum_{i=n}^{\infty} \sum_{v=0}^{(1-\epsilon)i} Pr\left[|T_1 \cap (v, i]| > |T_2 \cap (v, i]|\right] && \text{From union bound.} \end{aligned}$$

Observe $|T_k \cap (v, i]| = \sum_{j=v+1}^{i} \mathbb{1}_{j \in T_k}$ is the sum of Bernoulli random variables where $Pr[j \in T_1] = 1 - Pr[j \in T_2] = \alpha$. Thus event $|T_1 \cap (v, i]| > |T_2 \cap (v, i]|$ is equivalent to $\sum_{j=v+1}^{i} \mathbb{1}_{j \in T_1} > \frac{i-v}{2}$. To bound $Pr\left[\sum_{j=v+1}^{i} \mathbb{1}_{j \in T_1} > \frac{i-v}{2}\right]$, we will use the Chernoff bound and the fact $\alpha < 1/2$.

Theorem F.12 (Chernoff Bound)**.** *If $X_1, X_2, \ldots, X_n$ are independent indicator random variables where $\mathbb{E}[X_i] = \mu$, then for any $\delta > 0$,*

$$Pr\left[\sum_{i=1}^{n} X_i \geq (1+\delta)\mu\right] \leq e^{-\frac{\delta^2 \mu}{2+\delta}}.$$

Claim F.13. *Let $X_1, X_2, \ldots, X_n$ be i.i.d. copies of Bernoulli random variable X where $\mathbb{E}[X] = \alpha < 1/2$. Then $Pr[\sum_{i=1}^{n} X_i > n/2] \leq e^{-\frac{(1-2\alpha)n}{4}}$*

Proof. Let $\mu = E[\sum_{i=1}^{n} X_i] = \alpha n$ and $\delta = (1-2\alpha)/(2\alpha)$. From Chernoff Bound and the fact $\alpha < 1/2$,

$$Pr\left[\sum_{i=1}^{n} X_i > n/2\right] = Pr\left[\sum_{i=1}^{n} X_i > \mu(1+\delta)\right] \leq e^{-\frac{\delta^2 \mu}{2+\delta}} < e^{-\frac{(1-2\alpha)n}{4}}.$$

□

Therefore, $Pr\left[\sum_{j=v+1}^{i} \mathbb{1}_{j \in T_1} > \frac{i-v}{2}\right] \leq e^{-(1-2\alpha)(i-v)/2} < e^{-(1-2\alpha)\epsilon i}$ where the last inequality is the fact $v < (1-\epsilon)i$. We conclude

$$Pr\left[\cup_{i=n}^{\infty} \left\{\delta^1(X_i) > i\epsilon\right\}\right] \leq \sum_{i=n}^{\infty} \sum_{v=0}^{(1-\epsilon)i} e^{-(1-2\alpha)\epsilon i} \leq \sum_{i=n}^{\infty} e^{-\Omega(i)} = e^{-\Omega(n)}$$

as desired. For the "Moreover" part, we just observe $\frac{\delta^1(X_n)}{n} \overset{a.s.}{\to} 0$ is equivalent to

$$\lim_{n\to\infty} Pr\left[\cup_{i=n}^{\infty} \left\{\delta^1(X_i) > i\epsilon\right\}\right] = 0 \qquad \text{for all } \epsilon > 0.$$

This is clear since $e^{-\Omega(n)} \to 0$. For the second part, note from state X_{n-1} to X_n, a single block is created; therefore, $r^1(X_{n-1}, X_n) \leq \delta^1(X_n) + 1$. Thus

$$\frac{r^1(X_{n-1}, X_n)}{n} \leq \frac{\delta^1(X_n) + 1}{n} \overset{a.s.}{\to} 0$$

as desired. □

Proof of Theorem 5.12. Let π be a trimmed strategy. We claim there exists a strategy $f(\pi)$ that is checkpoint preserving. We will interactively transform π into $f(\pi)$. Initialize $f(\pi)$ to be equal to π.

Step 1. Whenever $f(\pi)$ reaches a state B where $f(\pi)$ is about to take action PublishPath(Q, v) where $\textsc{Succ}(v)$ contains a checkpoint, we will transform $f(\pi)$ so that it is does not fork a checkpoint. We consider two cases:

Case 1. Consider the case block v reached finality with respect to $f(\pi)$. Then $f(\pi)$ and B satisfy the conditions for Proposition F.8. Thus there is a trimmed strategy π' where π' is equals to $f(\pi)$ up to state B and at state B, π' follows the algorithm in Proposition F.8. Redefine $f(\pi)$ to be equal to π' and repeat until $f(\pi)$ reaches another state where $f(\pi)$ is about to fork a checkpoint. If no such state exists, $f(\pi)$ is checkpoint preserving.

Case 2. Consider the case block v does not reach finality with respect to $f(\pi)$. From Lemma F.6, max Q becomes a checkpoint after $f(\pi)$ publishes Q pointing to v. Observe that in the future, $f(\pi)$ will take action PublishPath(Q', v') with $h(v') < h(v)$ at most $h(v)$ times; otherwise, there is a state where $f(\pi)$ takes action PublishPath(Q', v') and the immediate successor of block v' was created by Miner 1, a contradiction to $f(\pi)$ being trimmed. Let B' be the last state reachable from B where $f(\pi)$ takes action PublishPath(Q', v') with $h(v') < h(v)$. Clearly v' reaches finality with respect to $f(\pi)$. Note $f(\pi)$ forks a checkpoint by publishing Q' pointing to v' since max Q became a checkpoint and any action thereafter that forks max Q induces the longest chain to become a checkpoint. Thus state B' and strategy $f(\pi)$ satisfy the conditions for Proposition F.8. Thus there is a trimmed strategy π' that is equals to $f(\pi)$ up to state B' and at state B', π' follows the algorithm in Proposition F.8. Redefine $f(\pi)$ to be equal to π' and return to Step 1 with strategy $f(\pi)$ and state B.

Since we visit Case 2 with state B at most $h(v)$ times, eventually, block v reaches finality with respect to $f(\pi)$ and we execute Case 1. Once Case 1 is executed with strategy $f(\pi)$ and state B, $f(\pi)$ becomes checkpoint preserving on a larger set of states. Ad infinitum $f(\pi)$ becomes checkpoint preserving as desired.

This proves there is a strategy $f(\pi)$ that is valid, trimmed and checkpoint preserving as desired. Next, we check $\textsc{Rev}(f(\pi)) \geq \textsc{Rev}(\pi)$. Let $(X_t^{f(\pi)})_{t\geq 0}$ be a mining game starting at $X_0^{f(\pi)} = B_0$ and where Miner 1 follows $f(\pi)$. Let t_i be the i-th time step where $f(\pi)$ would take action PublishPath(Q, v) and fork a checkpoint $P_i \in \textsc{Succ}(v)$, but, instead, we execute the algorithm in Proposition F.8. That is, $f(\pi)$ waits until time $\tau_i \geq t_i$ where $f(\pi)$ publishes $(Q \cap (P_i, \infty]) \cup (T_1(X_{\tau_i}^{f(\pi)}) \cap (t_i, \tau_i])$ pointing to P_i. Let $\tau_0 = 0$, $N_t = \sum_{i=1}^{\infty} \mathbb{1}_{\tau_i \leq t}$ and observe $\tau_{N_t} \leq t < \tau_{N_t+1}$.

Corollary F.14. *For any strategy π, there is a trimmed, checkpoint preserving strategy $f(\pi)$ where*

$$\textsc{Rev}_\gamma^{(t)}(f(\pi)) \geq \frac{|A(\mathcal{C}(X_t)) \cap T_1| + \sum_{j=1}^{N_t} \sum_{i=1}^{W_0^j} Z_i^j - \delta^1(X_t)}{h(\mathcal{C}(X_t)) + \sum_{j=1}^{N_t} W_0^j + \sum_{j=1}^{N_t} \sum_{i=1}^{W_0^j} Z_i^j + \delta^1(X_t)}.$$

where $(X_t)_{t\geq 0}$ is a mining game with $X_0 = B_0$ where Miner 1 follows π. For all j, $W_0^j \geq 1$ and $(Z_i^j)_{i,j}$ are i.i.d random variables with expected value $\frac{\alpha}{1-2\alpha}$.

Proof. The work above transform a strategy π into a trimmed, checkpoint preserving strategy $f(\pi)$. The revenue of $f(\pi)$ with respect to the initial strategy π follows from applying Proposition F.8 each time Case 1 is executed. □

From supperaddivity of lim inf,

$$\liminf_{t\to\infty} \textsc{Rev}_\gamma^{(t)}(f(\pi)) \geq \liminf_{t\to\infty} \textsc{Rev}_\gamma^{(t)}(\pi) + \liminf_{t\to\infty} \Big(\textsc{Rev}_\gamma^{(t)}(f(\pi)) - \textsc{Rev}_\gamma^{(t)}(\pi)\Big)$$

It suffices to show $\liminf_{t\to\infty}\left(\textsc{Rev}_\gamma^{(t)}(f(\pi)) - \textsc{Rev}_\gamma^{(t)}(\pi)\right) \geq 0$. Note Corollary F.14 provides a lower bound on $\textsc{Rev}_\gamma^{(t)}(f(\pi))$.

Proposition F.15. $\liminf_{t\to\infty}\left(\frac{|A(\mathcal{C}(X_t))\cap T_1|+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j-\delta^1(X_t)}{h(\mathcal{C}(X_t))+\sum_{j=1}^{N_t} W_0^j+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j+\delta^1(X_t)} - \textsc{Rev}_\gamma^{(t)}(\pi)\right) \geq 0.$

Proof. Recall Miner 2 never publishes two block at the same height; therefore, $h(\mathcal{C}(X_t)) \geq |T_2(X_t)|$. Since $|T_2(X_t)|$ is the sum of t i.i.d. random variables with expected value $1-\alpha$, from the strong law of large numbers

$$\frac{|T_2(X_t)|}{t} \overset{a.s.}{\to} 1-\alpha \qquad \text{and} \qquad \frac{|T_1(X_t)|}{t} = \frac{t-|T_2(X_t)|}{t} \overset{a.s.}{\to} \alpha$$

From Lemma F.11, $\frac{\delta^1(X_t)}{h(\mathcal{C}(X_t))} \overset{a.s.}{\to} 0$ and

$$\begin{aligned}
&\liminf_{t\to\infty}\left(\frac{|A(\mathcal{C}(X_t))\cap T_1|+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j-\delta^1(X_t)}{h(\mathcal{C}(X_t))+\sum_{j=1}^{N_t} W_0^j+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j+\delta^1(X_t)} - \textsc{Rev}_\gamma^{(t)}(\pi)\right)\\
&\quad= \liminf_{t\to\infty}\left(\frac{|A(\mathcal{C}(X_t))\cap T_1|+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j}{h(\mathcal{C}(X_t))+\sum_{j=1}^{N_t} W_0^j+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j} - \frac{|A(\mathcal{C}(X_t))\cap T_1|}{h(\mathcal{C}(X_t))}\right)\\
&\quad= \liminf_{t\to\infty}\frac{t\cdot\sum_{j=1}^{N_t} W_0^j}{t\cdot\sum_{j=1}^{N_t} W_0^j}\left(\frac{h(\mathcal{C}(X_t))\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j - |A(\mathcal{C}(X_t))\cap T_1|\left(\sum_{j=1}^{N_t} W_0^j+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j\right)}{\left(h(\mathcal{C}(X_t))+\sum_{j=1}^{N_t} W_0^j+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j\right)h(\mathcal{C}(X_t))}\right)\\
&\quad= \liminf_{t\to\infty} a_t.
\end{aligned}$$

First, consider the case $\lim_{t\to\infty} N_t < \infty$ which implies

$$\limsup_{t\to\infty}\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j \quad \text{and} \quad \limsup_{t\to\infty}\sum_{j=1}^{N_t} W_0^j < \infty.$$

Thus

$$\frac{\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j}{h(\mathcal{C}(X_t))} \overset{a.s.}{\to} 0 \qquad \frac{\sum_{j=1}^{N_t} W_0^j}{h(\mathcal{C}(X_t))} \overset{a.s.}{\to} 0.$$

Therefore, $\liminf_{t\to\infty} a_t = 0$ as desired. Next, we consider the case $\lim_{t\to\infty} N_t = \infty$. From supermultiplicativity of $\liminf$,

$$\liminf_{t\to\infty} a_t \geq \frac{\liminf_{t\to\infty}\frac{h(\mathcal{C}(X_t))\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j-|A(\mathcal{C}(X_t))\cap T_1|\left(\sum_{j=1}^{N_t} W_0^j+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j\right)}{t\cdot\sum_{j=1}^{N_t} W_0^j}}{\limsup_{t\to\infty}\frac{\left(h(\mathcal{C}(X_t))+\sum_{j=1}^{N_t} W_0^j+\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j\right)h(\mathcal{C}(X_t))}{t\cdot\sum_{j=1}^{N_t} W_0^j}} = \frac{\liminf_{t\to\infty} b_t}{\limsup_{t\to\infty} c_t}.$$

Next, we check $\frac{\liminf_{t\to\infty} b_t}{\limsup_{t\to\infty} c_t}$ is well-defined. For that, we will show $\liminf_{t\to\infty} b_t$ is lower bounded by 0 and $\limsup_{t\to\infty} c_t$ is bounded away from 0. Recall $\liminf_{t\to\infty} \frac{h(\mathcal{C}(X_t))}{t} \geq 1-\alpha$ almost surely. Thus

$$\limsup_{t\to\infty} c_t = \limsup_{n\to\infty} \frac{h(\mathcal{C}(X_t) + \sum_{j=1}^{N_t} W_0^j + \sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j}{\sum_{j=1}^{N_t} W_0^j} \frac{h(\mathcal{C}(X_t))}{t} \geq 1-\alpha > 0.$$

From Proposition F.8, $\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j$ is the sum of $\sum_{j=1}^{N_t} W_0^j$ i.i.d. random variables with expected value $\frac{\alpha}{1-2\alpha}$. From the strong law of large numbers,

$$\frac{h(\mathcal{C}(X_t))}{t} \frac{\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j}{\sum_{j=1}^{N_t} W_0^j} \geq \frac{|T_2(X_t)|}{\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j} \overset{a.s.}{\to} (1-\alpha)\frac{\alpha}{1-2\alpha}.$$

Observe $A(\mathcal{C}(X_t) \cap T_1 \subseteq T_1(X_t)$. Then

$$\frac{|A(\mathcal{C}(X_t)) \cap T_1|}{t} \frac{\sum_{j=1}^{N_t} W_0^j + \sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j}{\sum_{j=1}^{N_t} W_0^j} \geq \frac{|T_1(X_t)|}{t}\left(1 + \frac{\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j}{\sum_{j=1}^{N_t} W_0^j}\right) \overset{a.s.}{\to} \alpha\left(1 + \frac{\alpha}{1-2\alpha}\right)$$

The work above proves

$$\begin{aligned}\liminf_{t\to\infty} b_t &= \liminf_{t\to\infty} \frac{h(\mathcal{C}(X_t))\sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j - |A(\mathcal{C}(X_t)) \cap T_1|\left(\sum_{j=1}^{N_t} W_0^j + \sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j\right)}{t \cdot \sum_{j=1}^{N_t} W_0^j} \\ &\geq \frac{\alpha(1-\alpha)}{1-2\alpha} - \frac{\alpha(1-\alpha)}{1-2\alpha} = 0\end{aligned}$$

Thus $\liminf_{t\to\infty} a_t \geq 0$ as desired. □

From Proposition F.15, $\liminf_{t\to\infty} \textsc{Rev}_\gamma^{(t)}(f(\pi)) \geq \liminf_{t\to\infty} \textsc{Rev}_\gamma^{(t)}(\pi)$. Taking the expected value proves $\textsc{Rev}(f(\pi)) \geq \textsc{Rev}(\pi)$ as desired. □

F.3 Opportunistic Reduction

First, we give an example that highlights the main ideas of the proof.

Example F.16. *In the left of Figure 12, consider a state where Miner 1 withholds blocks 1 and 2 and publishes only* $1 \to 0$ *with block 1 reaching finality – that is, Miner 1 is following a strategy that will never fork block 1. Thus Miner 1's strategy is not opportunistic because block 2 remains unpublished. To transform Miner 1's strategy into an opportunistic one, Miner 1 will wait during the second round. For future rounds, we consider two cases depending on whom creates the third block.*
Case 1. *Consider the case Miner 1 creates block 3 (center of Figure 12). Let Miner 1 publish* $1 \to 0$ *and copy all the future actions of the original strategy as if nothing happened.*
Case 2. *Consider the case Miner 2 creates block 3 (right of Figure 12). Let the new strategy*

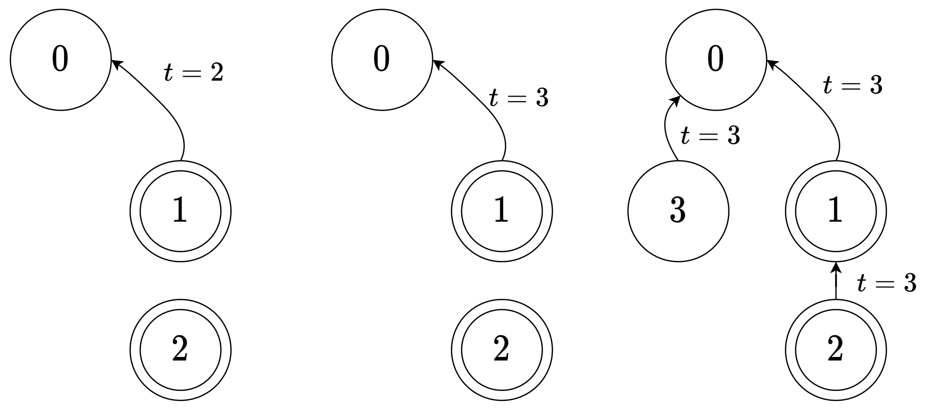

Figure 12: In the left, we have a checkpoint preserving strategy π that is not 1-Opportunistic. In the center and right, we have transform π into a 1-Opportunistic strategy $\hat{\pi}$. In the center, we consider the case where Miner 1 creates block 3 and in the right, we consider the case where Miner 2 creates block 3.

publish $2 \to 1 \to 0$. In future rounds, the new strategy will publishes all edges $u \to s$ the original strategy publishes except when $u = 2$ or $s = 3$. If $u = 2$, the new strategy ignores edge $u \to s$ because $u \to s$ was already published. That is, the original strategy must be publishing edge $2 \to 1$ (because block 1 reached finality), but the new strategy already published $2 \to 1$ during the second round. If $s = 3$, the new strategy publishes $u \to 2$ instead. The only difference between the game trees is that block 3 is replaced by block 2 (which was created by Miner 1).

Proposition F.17. *Let π be a trimmed strategy. Let B be a state where:*

- *State B is reachable from a mining game starting at state B_0 with Miner 1 following strategy π.*
- *π takes action PublishPath(Q, v).*
- *v reached finality with respect to π.*
- *$Q \subset \mathcal{U}(B) \cap (v, \infty)$.*

Then there is a trimmed strategy $f(\pi)$ that is equals to π up to state B. At state B, $f(\pi)$ waits. Let $N = \max\ V(B)+1$. To define $f(\pi)$ at future states, let $(X_t^{\pi})_{t\geq 0}$ and $(X_t^{f(\pi)})_{t\geq 0}$ be mining games where Miner 1 follows strategy π and $f(\pi)$ respectively with $X_0^{\pi} = X_0^{f(\pi)} = B$. Define the coupling $(X_t^{\pi}, X_t^{f(\pi)})_{t\geq 0}$ where Miner k creates block $t \geq N$ in both games. Let $q = \min\ (\mathcal{U}(B) \cap (v, \infty)) \setminus Q$. If Miner 1 creates block N, at state $(X_1^{\text{HALF}})^{f(\pi)}$, let $f(\pi)$ publish Q pointing to v. Else, let $f(\pi)$ publish $Q \cup \{q\}$ pointing to v. Moreover, at all time steps $t \geq 1$, whenever π publishes $u \to s$, we consider the cases:

- *If $u = q$, $f(\pi)$ does not publish.*
- *If $s = N$, $f(\pi)$ publishes $u \to q$ instead.*

- *If $u \neq q$ and $s \neq N$, $f(\pi)$ also publishes $u \to s$.*

Then, for all time steps $t \geq 0$,

$$\textsc{Rev}_{\gamma}^{(t)}(f(\pi)) \geq \frac{|A(\mathcal{C}(X_t^{\pi})) \cap T_1| - \mathbb{1}_{t=N-1} \cdot \delta^1(B)}{h(\mathcal{C}(X_t^{\pi}))}.$$

Proof. Let's check $f(\pi)$ is a trimmed and valid strategy. Up to state B, $f(\pi)$ is equals to π which, by assumption, is trimmed and valid. At state B, we consider the case where Miner 1 or Miner 2 creates block N separately. For the case Miner 1 creates block N, $f(\pi)$ is equals to π except $f(\pi)$ delays one round to publish Q pointing to v. Then $f(\pi)$ copy all future actions of π. Thus for the case Miner 1 creates block N, $f(\pi)$ is trimmed and valid since π is trimmed and valid.

For the case Miner 2 creates block N, $f(\pi)$ first publishes $Q \cup \{q\}$ pointing to v at step N. Let's check this action is valid and trimmed. Note $q = (\mathcal{U}(B) \cap (v, \infty)) \setminus Q$ is well-defined since, by assumption, Q is a strict subset of $\mathcal{U}(B) \cap (v, \infty)$. Moreover, the immediate successor of block v must be a block created by Miner 2; otherwise, PublishPath(Q, v) would not be a trimmed action at state B. Thus, at step N, publishing $Q \cup \{q\}$ pointing to v is valid and trimmed action.

Next, still considering the case Miner 2 creates block N, we proof the simulations of π and $f(\pi)$ satisfy the invariant that the longest paths $A(\mathcal{C}(X_t^{\pi}))$ and $A(\mathcal{C}(X_t^{f(\pi)}))$ are identical except for the edge $z \to \max\ Q$. In π's longest path, z is either equals to q or N, while in $f(\pi)$'s longest path, z is always equals to q. The invariant is clearly satisfied after π publishes $Q \cup \{q\}$ pointing to v so we need to check that, in the future, the invariant is preserved whenever π or $f(\pi)$ publishes blocks.

First, observe all blocks Q reached finality with respect to π. To check, by assumption, v reached finality with respect to π and the fact π is trimmed implies π would never take an action PublishPath(Q', v') where the immediate successor of block v' is in Q. Since $v' \geq v$ (since v reached finality), we must have $v' \geq \max\ Q$. This proves $\max\ Q$ (in fact all blocks in Q) reached finality with respect to π. Next, we prove the invariant and $f(\pi)$'s actions are valid and trimmed. Consider the event π publishes $u \to s$. We divide the proof into three cases:

- $u = q$. For this case, we claim $s = \max\ Q$. The fact $\max\ Q$ reached finality with respect to π implies $s \geq \max\ Q$. Because q was an unpublished block at state B an π is about to publish $q \to s$, we have $\max\ Q \leq s < q < N$. Thus s is block created by Miner 1 since any blocks Miner 2 publishes pointing to $\max\ Q$ are bigger or equal than N. Note PublishPath$(Q \cup \{q\}, v)$ was an orderly action at state B since PublishPath(Q, v) was an orderly action at state B and $q = \min\ \mathcal{U}(B) \cap (v, \infty) \setminus Q$. Thus π has no unpublished blocks between $\max\ Q$ and q which implies $s = \max\ Q$. This proves π is publishing edge $q \to \max\ Q$ as desired. Observe $f(\pi)$ already published this edge at step N. Because π is timeserving, $q \to \max\ Q$ is about to become an edge in the longest path (and $N \to \max\ Q$ is about to become an orphaned edge) as desired. Once q becomes part of the longest path, q reaches finality with respect to π since $\max\ Q$ reached finality with respect to π. To check, observe the fact π is trimmed implies π would never take an action PublishPath$(Q', \max\ Q)$ since the immediate successor of $\max\ Q$ would be block q (a block created by Miner 1).

- $s = N$. Recall $\max\ Q$ must still be a block in the longest path since $\max\ Q$ reached finality with respect to π. Thus $N \to \max\ Q$ is still an edge in the longest path while, by the inductive hypothesis, $q \to \max\ Q$ is the equivalent edge in the longest path of $f(\pi)$'s simulation. By the inductive hypothesis, the immediate successor of block N and q are identical. Moreover, the immediate successor of block N is a block created by Miner 2 (since π is trimmed and π is about to publish edge $u \to N$). Thus publishing $u \to q$ is a trimmed action for $f(\pi)$. Moreover, publishing edge $u \to q$ instead of $u \to N$ preserves the invariant as desired.

- $u \neq q$ and $s \neq N$. First, we claim $s > \max\ Q$. Recall block $\max\ Q$ reached finality with respect to π. Then $s \geq \max\ Q$. If $s = \max\ Q$, then $u = q$ because π is trimmed (and orderly) and q was one of π's unpublished blocks at state B. This contradicts $u \neq q$ and proves $s > \max\ Q$ as desired. From the inductive hypothesis, s is also a block in the longest path of $f(\pi)$'s simulation. Therefore, publishing edge $u \to s$ is a valid and trimmed action for $f(\pi)$ that preserves the invariant.

This proves $f(\pi)$ is valid and trimmed and proves the invariant that the longest path in π's and $f(\pi)$'s simulation are identical except when edge $N \to \max\ Q$, in π's simulation, is replaced by edge $q \to \max\ Q$, in $f(\pi)$'s simulation. Let's now compare the revenue of $f(\pi)$ and π. Clearly, the height of the longest path in $f(\pi)$ is at most the height of the longest path in π. In fact, the heights are identical until the $(N-2)$-th step. During the $(N-1)$-th step, the height can be strictly lower for $f(\pi)$ when $f(\pi)$ waits but π publishes Q pointing to v. After the N-th step, the longest paths have the same height.

Let's now compare how many blocks π and $f(\pi)$ have in the longest path. Until the $(N-2)$-th step, both π and $f(\pi)$ have the same number of blocks in the longest path. During the $(N-1)$-th step, $f(\pi)$ can have less blocks in the longest path when $f(\pi)$ waits, but π publishes Q pointing to v; however, the difference is bounded by $\delta^1(B)$, Definition F.7, the maximum reward $f(\pi)$ can obtain from state B. After the N-th step, $f(\pi)$ have at least the same number of blocks in the longest path as π. In fact, $f(\pi)$ can have one more block when Miner 2 creates block N and edge $N \to \max\ Q$ is replaced by edge $q \to \max\ Q$. Thus for all t, $\text{Rev}_\gamma^{(t)}(f(\pi)) \geq \frac{|A(\mathcal{C}(X_t^\pi)) \cap T_1| - \mathbb{1}_{t=N-1} \cdot \delta^1(B)}{h(\mathcal{C}(X_t^\pi))}$ as desired. □

Proof of Theorem 5.14. Without loss of generality, let π be a trimmed, checkpoint preserving strategy. Otherwise, from Theorem 5.12, there is a valid, trimmed strategy with the same revenue as π. Initialize $f(\pi)$ to be equal to π. To transform $f(\pi)$ into a trimmed, checkpoint preserving and opportunistic strategy execute the following procedure.

Step 1. While $f(\pi)$ is not opportunistic, there is a state B where $f(\pi)$ takes action PublishPath(Q, v) where v reaches finality with respect to π, but $Q \subset \mathcal{U}(B) \cap (v, \infty)$. Let B be the first of such states that $f(\pi)$ encounters during a mining game starting at state B_0. From Proposition F.17, there exists a strategy that is equals to $f(\pi)$ up to state B and at state B onward, such strategy executes the algorithm described in Proposition F.17. Update $f(\pi)$ to be such strategy. Note $f(\pi)$ is trimmed and both opportunistic and checkpoint preserving up to state B, but might not be opportunistic nor checkpoint preserving on subsequent states. From Proposition F.8, there is a strategy that is equals to $f(\pi)$ up to state B and is checkpoint preserving on all subsequent states. Let $f(\pi)$ be such strategy.

At this point $f(\pi)$ is trimmed, checkpoint preserving, and opportunistic up to state B, then return to Step 1.

Each time we execute Step 1, $f(\pi)$ is trimmed, checkpoint preserving, and opportunistic on a larger set of states. Therefore, ad infinitum $f(\pi)$ becomes an opportunistic strategy. Observe the end result of executing Step 1 ad infinitum is that whenever Miner 1 was about to take a non-opportunistic action with a lead of k blocks, Miner 1 waits until the lead reduces to a single block, then Miner 1 publishes. This sequence of actions is equivalent to the selfish mining strategy at state $B_{2,0}$.

Next, we compare the revenue of $f(\pi)$ and π. Let $(X_t)_{t\geq 0}$ be a mining game with initial state $X_0 = B_0$ where Miner 1 follows strategy π. From Proposition F.17 and Corollary F.14, the revenue of $f(\pi)$ with respect to π is given by

$$\textsc{Rev}_\gamma^{(t)}(f(\pi)) \geq \frac{|A(\mathcal{C}(X_t)) \cap T_1| + \sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j - \delta^1(X_t)}{h(\mathcal{C}(X_t)) + \sum_{j=1}^{N_t} W_0^j + \sum_{j=1}^{N_t}\sum_{i=1}^{W_0^j} Z_i^j + \delta^1(X_t)}.$$

where N_t denotes the number of times $f(\pi)$ was transformed by Proposition F.8, $W_0^j \geq 1$ and $(Z_i^j)_{i,j}$ are i.i.d random variables with expected value $\frac{\alpha}{1-2\alpha}$. From superadditivity of $\liminf$

$$\liminf_{t\to\infty} \textsc{Rev}_\gamma^{(t)}(f(\pi)) \geq \liminf_{t\to\infty} \textsc{Rev}_\gamma^{(t)}(\pi) + \liminf_{t\to\infty}\left(\textsc{Rev}_\gamma^{(t)}(f(\pi)) - \textsc{Rev}_\gamma^{(t)}(\pi)\right).$$

From Proposition F.15 and our lower bound on $\textsc{Rev}_\gamma^{(t)}(f(\pi))$,

$$\liminf_{t\to\infty}\left(\textsc{Rev}_\gamma^{(t)}(f(\pi)) - \textsc{Rev}_\gamma^{(t)}(\pi)\right) \geq 0.$$

Taking the expected value proves $\textsc{Rev}(f(\pi)) \geq \textsc{Rev}(\pi)$ as desired. $\square$

F.4 The Strong Recurrence Theorem

For a mining game $(X_t)_{t\geq 0}$ starting at state $X_0 = B_0$ where Miner 1 follows optimal checkpoint recurrent strategy π, define $t_0 = 0$ and let $t_1, t_2, \ldots$ be the sequence of time steps where Miner 1 capitulates to B_0. Let $\tau_i = \tau_i - t_{i-1}$ and recall $(\tau_i)_{i\geq 1}$ is an i.i.d. sequence of positive random variables. Thus $N_n = \sum_{i=1}^{\infty} \mathbb{1}_{t_i \leq n}$ (with $t_0 = 0$) is a counting process (Definition B.5) and satisfy the following zero-one law.

Claim F.18 (Zero-One Law). *One of the following holds,*

- *If Miner 1's strategy is transient,* $Pr\left[\lim_{n\to\infty} N_n < \infty\right] = 1$.
- *If Miner 1's strategy is recurrent,* $Pr\left[\lim_{n\to\infty} N_n = \infty\right] = 1$.

Proof. Without loss of generality, let $X_0 = B_0$. Let p be the probability that Miner 1 capitulates to state B_0 at time $\tau \geq 1$. If Miner 1's strategy is transient, $p < 1$ and we get

$$Pr\left[\lim_{n\to\infty} N_n < \infty\right] = \sum_{i=0}^{\infty} Pr\left[\lim_{n\to\infty} N_n = i\right] = \sum_{i=1}^{\infty} p^i(1-p) = (1-p)\frac{1}{1-p} = 1.$$

If $p = 1$, $Pr\left[\lim_{n\to\infty} N_n < \infty\right] = 0$ which implies $\lim_{n\to\infty} N_n = \infty$ almost surely. $\square$

For the case where Miner 1's strategy is transient, Claim F.18 implies that with probability 1, the game will reach a time step t_i where Miner 1 never defines another checkpoint. From the definition of checkpoints, this implies Miner 1 never publishes more than half of all the blocks they creates after time step t_i. This simple observation suffices to witness that FRONTIER has at least the same payoff as π.

Claim F.19. *If π is checkpoint recurrent and transient,* $\textsc{Rev}(\pi) \leq \alpha$.

Proof. Recall $t_{N_n} \leq n < t_{N_n+1}$ (because N_n is a counting process). From Claim F.18, $t_{N_n} < \infty$ almost surely as $n \to \infty$ (because $N_n < \infty$ almost surely as $n \to \infty$). Thus

$$\textsc{Rev}_{\gamma}^{(n)}(\pi) = \frac{|A(\mathcal{C}(X_{t_{N_n}})) \cap T_1| + \sum_{t=t_{N_n}+1}^{n} r^1(X_{t-1}, X_t)}{h(\mathcal{C}(X_n))} \leq \frac{t_{N_n} + \sum_{t=t_{N_n}+1}^{n} r^1(X_{t-1}, X_t)}{h(\mathcal{C}(X_n))}.$$

During time step $t_{N_n} \leq t < t_{N_n+1}$, Miner 1 has not yet reached a new checkpoint. Thus Miner 1 publishes at most half of all blocks created from time $t_{N_n} + 1$ to t:

$$\sum_{t=t_{N_n}+1}^{n} r^1(X_{t-1}, X_t) \leq \frac{1}{2} \sum_{i=1}^{n} \mathbb{1}_{i \in T_1}$$

where the inequality is Corollary 5.9. Recall the height of the longest path $h(\mathcal{C}(X_t))$ at time t is at least $\sum_{i=1}^{n} \mathbb{1}_{i \in T_2}$ – the number of blocks Miner 2 creates up to time t. Thus

$$\begin{aligned}
\frac{t_{N_n} + \sum_{t=t_{N_n+1}}^{n} r^1(X_{t-1}, X_t)}{h(\mathcal{C}(X_n))} &\leq \frac{t_{N_n} + \frac{1}{2}\sum_{i=1}^{n} \mathbb{1}_{i \in T_1}}{h(\mathcal{C}(X_n))} \\
&\leq \frac{t_{N_n} + \frac{1}{2}\sum_{i=1}^{n} \mathbb{1}_{i \in T_1}}{\sum_{i=1}^{n} \mathbb{1}_{i \in T_2}} \\
&= \frac{\frac{1}{n} t_{N_n} + \frac{1}{2n}\sum_{i=1}^{n} \mathbb{1}_{i \in T_1}}{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{i \in T_2}} \overset{a.s.}{\to} \frac{\alpha/2}{1-\alpha}
\end{aligned}$$

The last step observes $\frac{t_{N_n}}{n} \overset{a.s.}{\to} 0$, and uses the strong law of large numbers and the fact $Pr\left[i \in T_1\right] = \alpha$, and $Pr\left[i \in T_2\right] = 1 - \alpha$. Since $\alpha < 1/2$, we get $\textsc{Rev}(\pi) \leq \alpha$. $\square$

Next, we consider the case where π is checkpoint recurrent and null recurrent which implies $\mathbb{E}\left[\tau_i\right] = \infty$ for $i \geq 1$. To see that that $\textsc{Rev}(\pi) \leq \textsc{Rev}(\text{FRONTIER})$, let

$$A_S = \frac{\epsilon \sum_{i \in S} \tau_i}{|S|}$$

for any $\epsilon > 0$. Then partition $\{\tau_i\}_{1 \leq i \leq N}$ into $D_N = \{1 \leq i \leq N : \tau_i > A_{[N]}\}$ and $F_N = [N] \setminus D_N$. From the strong law of large numbers, we obtain $A_{[N]} \overset{a.s.}{\to} \infty$ because π is null recurrent. Interestingly, this implies $\min_{i \in D_N} \tau_i \overset{a.s.}{\to} \infty$ since $\min_{i \in D_N} \tau_i \geq A_{[N]}$.

Recall Miner 1 capitulates to state B_0 at time steps t_i and t_{i+1} (whenever Miner 1 defines a new checkpoint). Let k_i be the number of blocks Miner 1 creates from time $t_i + 1$ to t_{i+1}. By definition, Miner 1 does not define a new checkpoint from time $t_i + 1$ to $t_{i+1} - 1$; otherwise, Miner 1 would have capitulated to state B_0 between time $t_i + 1$ and $t_{i+1} - 1$. Thus up to time step $t_{i+1}-1$, Miner 1 publishes less than $k_i/2$ blocks (Corollary 5.9). If we hope for Miner 1's strategy to be better than FRONTIER, Miner 1's strategy must publish a significant fraction ϵk_i, for some $\epsilon > 0$, of their blocks at time step t_{i+1}. We expect the probability of Miner 1 publishing ϵk_i blocks diminishes exponentially in k_i because Miner 1 must create more blocks than Miner 2 (but Miner 2 has probability $1-\alpha > 1/2$ of creating each block). Lemma F.11 formalizes this intuition.

Claim F.20. *If π is checkpoint and null recurrent,* $\textsc{Rev}(\pi) \leq \alpha$.

Proof. Recall $t_{N_n} \leq n < t_{N_n+1}$ (because N_n is a counting process). Thus

$$|A(\mathcal{C}(X_n)) \cap T_1| = \sum_{t=1}^{n} r^1(X_{t-1}, X_t) \tag{16}$$

$$= \sum_{j=0}^{N_n-1} \left(r^1(X_{t_{j+1}-1}, X_{t_{j+1}}) + \sum_{t=t_j+1}^{t_{j+1}-1} r^1(X_{t-1}, X_t) \right) + \sum_{t=t_{N_n}}^{n} r^1(X_{t-1}, X_t). \tag{17}$$

From time step t_j+1 to t_{j+1}, for $j \geq 0$, Miner 1 publishes only blocks created from time step t_j+1 to t_{j+1} (because Miner 1 capitulates to state B_0 at time steps t_j and t_{j+1}). Observe Miner 1 did not reached a new checkpoint by time step $t_{j+1}-1$ (because the strategy is checkpoint recurrent). This implies Miner 1 publishes less than half of all the blocks created from time step t_j+1 to $t_{j+1}-1$ by time step $t_{j+1}-1$. From Corollary 5.9, we get

$$\sum_{t=t_j+1}^{t_{j+1}-1} r^1(X_{t-1}, X_t) \leq \frac{1}{2} \sum_{i=t_j+1}^{t_{j+1}} \mathbb{1}_{i \in T_1}. \tag{18}$$

From (16) and (18),

$$|A(\mathcal{C}(X_n)) \cap T_1| \leq \frac{1}{2} \sum_{i=1}^{n} \mathbb{1}_{i \in T_1} + \sum_{i=1}^{N_n} r^1(X_{t_i-1}, X_{t_i}).$$

Recall the height of the longest chain is at least the number of blocks Miner 2 creates. That is $h(\mathcal{C}(X_n)) \geq \sum_{t=1}^{n} \mathbb{1}_{i \in T_2}$. Thus

$$\begin{aligned}
\textsc{Rev}(\pi) = \mathbb{E}\left[\liminf_{n\to\infty} \textsc{Rev}_\gamma^{(n)}(\pi)\right] &\leq \mathbb{E}\left[\liminf_{n\to\infty} \frac{\frac{1}{2n}\sum_{i=1}^{n} \mathbb{1}_{i\in T_1} + \frac{1}{n}\sum_{j=1}^{N_n} r^1(X_{t_i-1}, X_{t_i})}{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{i \in T_2}}\right] \\
&= \mathbb{E}\left[\liminf_{n\to\infty} \frac{\frac{\alpha}{2} + \frac{1}{n}\sum_{j=1}^{N_n} r^1(X_{t_i-1}, X_{t_i})}{1-\alpha}\right] \quad \text{From the strong law of large numbers.} \\
&\leq \alpha + \mathbb{E}\left[\liminf_{n\to\infty} \frac{1}{n}\sum_{i=1}^{N_n} r^1(X_{t_i-1}, X_{t_i})\right] \quad \text{Because } \alpha < 1/2.
\end{aligned}$$

From the Zero-One Law there is there is a sequence of time steps $t_1, t_2, \ldots$ where Miner 1 capitulates to state B_0 (and $t_0 = 0$). Fix any $\epsilon > 0$. Let $\tau_i = t_i - t_{i-1}$ for $i \geq 1$. For a subset of $t_1, t_2, \ldots$, let $S \subset \mathbb{N}$, $C_S = \sum_{i \in S} \tau_i$ and $A_S = \frac{\epsilon C_S}{|S|}$ and observe $C_{[N]} = \tau_N$ for all $N \in \mathbb{N}$. Note $n \geq t_{N_n}$ (because N_n is a counting process). Then

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{N_n} r^1(X_{t_i-1}, X_{t_i}) \leq \frac{1}{t_{N_n}}\sum_{i=1}^{N_n} r^1(X_{t_i-1}, X_{t_i}) &= \frac{1}{C_{[N_n]}}\sum_{i=1}^{N_n} r^1(X_{t_i-1}, X_{t_i}) \\ &= \frac{1}{C_{[N_n]}}\sum_{i=1}^{N_n}\left(r^1(X_{t_i-1}, X_{t_i}) \cdot \mathbb{1}_{\tau_i < A_{[N_n]}} + r^1(X_{t_i-1}, X_{t_i}) \cdot \mathbb{1}_{\tau_i \geq A_{[N_n]}}\right).\end{aligned}$$

To bound the first term, note from time $t_{i-1}+1$ to t_i, Miner 1 only publishes blocks created from time $t_{i-1}+1$ to t_i (and at most τ_i blocks were created in this time interval). Thus

$$\frac{1}{C_{[N]}}\sum_{i=1}^{N} r^1(X_{t_i-1}, X_{t_i}) \cdot \mathbb{1}_{\tau_i < A_{[N]}} \leq \frac{1}{C_{[N]}}\sum_{i=1}^{N} \tau_i \cdot \mathbb{1}_{\tau_i < A_{[N]}} < \frac{NA_{[N]}}{C_{[N]}} = \epsilon.$$

To bound the second term,

$$\begin{aligned}\frac{1}{C_{[N]}}\sum_{i=1}^{N} r^1(X_{t_i-1}, X_{t_i}) \cdot \mathbb{1}_{\tau_i \geq A_{[N]}} &= \frac{1}{C_{[N]}}\sum_{i=1}^{N} \tau_i \frac{r^1(X_{t_i-1}, X_{t_i}) \cdot \mathbb{1}_{\tau_i \geq A_{[N]}}}{\tau_i} \\ &\leq \frac{\sum_{i=1}^{N}\tau_i}{C_{[N]}} \max_{i \in [N]} \frac{r^1(X_{t_i-1}, X_{t_i})}{\tau_i} \cdot \mathbb{1}_{\tau_i \geq A_{[N]}} \\ &= \max_{i \in [N]} \frac{r^1(X_{t_i-1}, X_{t_i})}{\tau_i} \cdot \mathbb{1}_{\tau_i \geq A_{[N]}} \quad \text{By definition } C_{[N]} = \sum_{i=1}^{N}\tau_i\end{aligned}$$

Next, we claim $\max_{i \in [N]} \frac{r^1(X_{t_i-1}, X_{t_i})}{\tau_i} \cdot \mathbb{1}_{\tau_i \geq A_{[N]}} \overset{a.s.}{\to} 0$. Recall $A_{[N]}$ is the average of i.i.d. random variables τ_i and the fact Miner 1's strategy is null recurrent implies $\mathbb{E}[\tau_i] = \infty$. Thus from the strong law of large numbers, $A_{[N]} \overset{a.s.}{\to} \infty$. Therefore, the event

$$\limsup_{N \to \infty}\left\{\max_{i \in [N]} \frac{r^1(X_{t_i-1}, X_{t_i})}{\tau_i} \cdot \mathbb{1}_{\tau_i \geq A_{[N]}} > 0\right\}$$

implies there is a mining game $(Y_t)_{t \geq 0}$ where $\limsup_{t \to \infty} \frac{r^1(Y_{t-1}, Y_t)}{t} > 0$; however, from Lemma F.11, $\frac{r^1(Y_{t-1}, Y_t)}{t} \overset{a.s.}{\to} 0$. Summing up,

$$\textsc{Rev}(\pi) \leq \alpha + \epsilon + \mathbb{E}\left[\liminf_{N \to \infty} \max_{i \in [N]} \frac{r^1(X_{t_i-1}, X_{t_i})}{\tau_i} \cdot \mathbb{1}_{\tau_i \geq A_{[N]}}\right] = \alpha + \epsilon$$

as desired. □

Proof of Strong Recurrence Theorem 5.15. From the Weak Recurrence Theorem 5.4 there exists an optimal strategy π that is checkpoint recurrent. Recall FRONTIER is a potential candidate since FRONTIER is checkpoint and positive recurrent (Observation 2.5), and $\textsc{Rev}(\text{FRONTIER}) = \alpha$ (Corollary 5.7). From Claim F.19 and Claim F.20, $\textsc{Rev}(\pi) \leq \alpha$ unless π is positive recurrent. Thus there exists an optimal strategy π that is checkpoint and positive recurrent. Suppose not, then $\textsc{Rev}(\pi) \leq \alpha = \textsc{Rev}(\text{FRONTIER})$ which witnesses FRONTIER is optimal. □

G Omitted Proofs From Section 6

The main observation to show FRONTIER is optimal (for sufficiently small α) is to note that any mining game starting at state B_0 will transition to either state $B_{0,1}$ or state $B_{1,0}$ (see Figure 3). Recall the value faction $\mathcal{V}$ (Definition C.3) evaluated at B_0 is $\mathcal{V}(B_0) = 0$ (Lemma C.4). From the recursive definition of the value function,

$$0 = \mathcal{V}(B_0) = \mathbb{E}\left[\mathrm{r}_{\lambda^*}(X_0, X_\tau)\right] = \alpha\,\mathcal{V}(B_{1,0}) + (1-\alpha)(\mathcal{V}(B_{0,1}) - \lambda^*)$$

where $\lambda^* = \max_\pi \textsc{Rev}(\pi)$ and $\tau \geq 1$ is the first time step where Miner 1 capitulates to state B_0. Note block 1 is a checkpoint in state $B_{0,1}$. From Theorem 5.15, Miner 1 has an optimal strategy that never forks a checkpoint (or block 1 in state $B_{0,1}$). Therefore, there is an optimal strategy that capitulates to state B_0 from state $B_{0,1}$. The following results will give us a closed form for the value function $\mathcal{V}$ evaluated at state $B_{0,1}$.

Proposition G.1. *For any state B, $\mathcal{V}(B) \geq 0$.*

Proof. From Bellman's principle of optimality (Lemma C.6),

$$\mathcal{V}(B) \geq \mathcal{V}_\pi^{\lambda^*}(B)$$

for any positive recurrent strategy π. Let π be a strategy that capitulates to state B_0 from state B. From state B, Miner 1 can follow any optimal strategy π^* as if it was at state B_0 by treating the longest chain $\mathcal{C}(B)$ as block 0. Thus $\mathcal{V}(B) \geq \mathcal{V}(B_0) = 0$. □

Proposition G.2. *If there is an optimal positive recurrent strategy that capitulates to state B_0 from state B, $\mathcal{V}(B) = 0$.*

Proof. Let π^* be an optimal strategy that capitulates to state B_0 from state B. From state B_0, Miner 1 can follow strategy π^* as if it was at state B by treating block 0 as the longest chain $\mathcal{C}(B)$. Then $0 = \mathcal{V}(B_0) \geq \mathcal{V}(B)$. From Proposition G.1, $\mathcal{V}(B) \geq 0$. Thus $\mathcal{V}(B) = 0$ as desired. □

Corollary G.3. *Once at state $B_{0,1}$, the optimal action for Miner 1 is to capitulate to state B_0. Thus $\mathcal{V}(B_{0,1}) = 0$.*

Proof. From the strong recurrence theorem, it is optimal for Miner 1 to capitulate to state B_0 from state $B_{0,1}$. Thus state $B_{0,1}$ satisfy the conditions for Proposition G.2 and $\mathcal{V}(B_{0,1}) = 0$. □

To show FRONTIER is optimal (for sufficiently small α), it suffices to show that $\mathcal{V}(B_{1,0})$ is maximized when Miner 1 publishes $1 \to 0$ and capitulates to state B_0. Then there is an optimal strategy taking the same actions as FRONTIER and capitulates to state B_0 every round as desired.

Formally, consider a strategy that, at state $B_{1,0}$, publishes $1 \to 0$ and capitulates to state B_0. The game reward (Definition C.2) is $1 - \lambda^*$ since Miner 1 adds one block to the longest path. From Bellman's principle of optimality,

$$\mathcal{V}(B_{1,0}) \geq 1 - \lambda^*.$$

Next, suppose strategy π does not publish $1 \to 0$ at state $B_{1,0}$. By showing $\mathcal{V}_{\pi}^{\lambda^*}(B_{1,0}) \leq 1 - \lambda^* \leq \mathcal{V}(B_{1,0})$ (when α is sufficiently small), we will derive a certificate that there is an optimal strategy that publishes $1 \to 0$ at state $B_{1,0}$. For that, we observe that starting from state $B_{1,0}$, and under the assumption Miner 1 does not publishes any blocks until at least the next round, the subsequent state is either $B_{2,0} = ((\{0\}, \emptyset), \{1,2\}, \{1,2\})$ when Miner 1 creates and withhold block 2, or $B_{1,1} = ((\{0,2\}, \{2 \to 0\}), \{1\}, \{1\})$ when Miner 2 creates block 2 and publishes $2 \to 0$. Then

$$\mathcal{V}_{\pi}^{\lambda^*}(B_{1,0}) = \alpha\, \mathcal{V}_{\pi}^{\lambda^*}(B_{2,0}) + (1-\alpha)(\mathcal{V}_{\pi}^{\lambda^*}(B_{1,1}) - \lambda^*) \leq \alpha\, \mathcal{V}(B_{2,0}) + (1-\alpha)(\mathcal{V}(B_{1,1}) - \lambda^*)$$

where the inequality is Bellman's principle of optimality.

As a warmup, we will show how we can derive the upper bound of $\mathcal{V}(B_{1,1}) \leq \frac{\alpha}{1-\alpha}$. As a thought experiment, image Miner 1 never forks block 2. Then block 1 is "useless" since it cannot point to any block ≥ 2. We would like to formalize what it means for Miner 1 to "forget" blocks 1 and 2 from state $B_{1,1}$. Once we forget blocks 1 and 2 from state $B_{1,1}$, state $B_{1,1}$ is equivalent to B_0 (or Miner 1 capitulates to state B_0).

At state B, we say a block $q \in \mathcal{V}(B) \cup \mathcal{U}(B)$ can *reach height* ℓ (from state B) if one of the two holds:

- if $q \in \mathcal{V}(B)$ was already published, then $h(q) \geq \ell$.
- if $q \in \mathcal{U}(B)$ is unpublished, then there is an action that Miner 1 can take from state B such that $h(q) \geq \ell$ in the subsequent state.

Recall any blocks created in the future can point to any block $\leq q$ but any block $\leq q$ cannot point to any block $> q$. Thus if block q cannot reach height ℓ from state B, then q cannot reach height ℓ from any state reachable from B.

Example G.4. *At state $B_{2,0} = ((\{0\}, \emptyset), \{1,2\}, \{1,2\})$, block 2 can reach heights 1 and 2 because Miner 1 can publish $2 \to 1 \to 0$, but block 2 cannot reach height ≥ 3.*

Definition G.5 (Induced Subgraph)**.** *Let $G = (V, E)$ be a graph and let $S \subseteq V$ be any subset of the vertices of G. The* induced subgraph *$G[S]$ is the graph whose vertex set is S and whose edge set consists of all edges in E with both end points in S.*

Definition G.6 (State Capitulation)**.** *Let $B = (\textsc{Tree}, \mathcal{U}, T_1)$ be a state and let $c \leq h(\mathcal{C}(B))$. Define $D \subseteq A(\mathcal{C}) \cup \mathcal{U} \setminus \{0\}$ as the set of blocks that cannot reach height $\geq c+1$ from state B. Define the c-Capitulation of B as the state*

$$B[V \setminus D] := (\textsc{Tree}[V \setminus D], \mathcal{U} \setminus D, T_1 \setminus D)$$

where $\textsc{Tree}[V \setminus D]$ is an induced subgraph of $\textsc{Tree}$ obtained by deleting blocks D.

Example G.7. *For state $B_{2,2} = ((\{0,2,3\}, \{3 \to 2 \to 0\}), \{1,4\}, \{1,4\})$, its 1-Capitulation deletes blocks 1 and 2, but not blocks 3 (since it is already at height 2) and 4 (since it can reach height 3 if Miner 1 publishes $4 \to 3$ or height 2 if Miner 1 publishes $4 \to 2$). Thus the 1-Capitulation of state $B_{2,2}$ is the state*

$$B_{2,2}[\{3,4\}] = ((\{0,3\}, \{3 \to 0\}), \{4\}, \{4\}).$$

Additionally, we could capitulate at height 2 to get state

$$B_{2,2}[\{4\}] = ((\{0\}, \emptyset), \{4\}, \{4\})$$

since block 4 can reach height 3 if Miner 1 publishes $4 \to 3$*, but all other blocks can only reach up to height 3.*

Recall $H_i(B)$ denotes the block at height i in state B – i.e., $h(H_i(B)) = i$. Next, we show how to upper bound $\mathcal{V}(B)$ in terms of $\mathcal{V}(B')$ where B' is the c-Capitulation of B.

Lemma G.8. *Let B be a state, $c \le h(\mathcal{C}(B))$, and let B' be its c-Capitulation. Then*

$$\mathcal{V}(B) \le \mathcal{V}(B') + \mathrm{r}_{\lambda^*}(B_0, B') - \mathrm{r}_{\lambda^*}(B_0, B) + \sum_{i=1}^{c} \left(Pr\left[H_i(X_\tau) \in T_1 | X_0 = B\right] - \lambda^* \right) \qquad (19)$$

where $\lambda^* = \max_\pi \textsc{Rev}(\pi)$ is the optimal revenue, and τ is the first time step Miner 1 capitulates to state B_0 in mining game $(X_t)_{t\ge 0}$ starting from state $X_0 = B$ where Miner 1 follows an optimal strategy.

For intuition behind Lemma G.8, observe a possible strategy at state B' is to copy the optimal strategy at state B by ignoring everything that happens at height $\le c$. From Bellman's principle of optimality, $\mathcal{V}(B')$ is lower bounded by the reward obtained by such strategy. Second, observe $\mathcal{V}(B) + r_{\lambda^*}(B_0, B)$ is equals to the expected number of blocks Miner 1 has in the longest path at X_τ minus the length of the longest path times λ^*. Finally, Lemma G.8 breaks down $\mathcal{V}(B)$ into four pieces: the first and second terms count the contributions from height $> c$ which is at most $\mathcal{V}(B') + r_{\lambda^*}(B_0, B')$; the fourth term, counts the contributions from heights $\le c$ that our strategy for B' ignores.

Example G.9. *Consider B where Miner 1 has on unpublished block 1, and Miner 2 has published* $4 \to 3 \to 2 \to 0$*. Let $c = 3$, then the c-Capitulation B' of B is B_0. Note $\mathcal{V}(B')$ is* 0*, $\mathrm{r}_{\lambda^*}(B_0, B')$ is* 0 *and $\mathrm{r}_{\lambda^*}(B_0, B)$ is $-3\lambda^*$. Thus Lemma G.8 implies*

$$\mathcal{V}(B) \le 0 + 0 + 3\lambda^* + \sum_{i=1}^{3} (Pr\left[H_i(X_\tau) \in T_1 | X_0 = B\right] - \lambda^*) \le 3.$$

Proof of Lemma G.8. Let $T = \max\{T_1(B), T_2(B)\}$ be the last block created at B. Define the mining game $(X_t)_{t\ge 0}$ starting at state $X_0 = B$ with Miner 1 following any optimal strategy π. Without loss of generality, π is a trimmed, positive recurrent strategy π (Theorem 5.15). Define the mining game $(X'_t)_{t\ge 0}$ starting at state $X'_0 = B'$ with Miner 1 following a strategy π' (that will depend on π). Define a coupling between each game where at time $t \ge 1$, Miner 1 creates block $t + T$ with probability α in both games; otherwise, Miner 2 creates block $t + T$ in both games. Define π' as follows:

- If π plays Wait in state $X_t^{\textsc{Half}}$, π' plays Wait in state $X_t'^{\textsc{Half}}$.
- If π capitulates to state B_0 from state X_t, π' capitulates to state B_0 from state X'_t.

- If π plays PublishPath(Q, v) at state $X_t^{\textsc{Half}}$, we will consider two distinct cases. Let

  $$Q' = \{q \in Q : h(q) \geq c+1 \text{ after } \pi \text{ plays PublishPath}(Q, v)\}$$

  Then, at state $X_t'^{\textsc{Half}}$,

  – If $h(v) > c$, π' plays PublishPath(Q', v).

  – If $h(v) \leq c$, π' plays PublishPath$(Q', 0)$.

For the graph $\textsc{Tree}(B) = (\mathrm{V}(B), \mathrm{E}(B))$, $\mathrm{V}(B)$ denote the set of blocks that were already published at state B. Let $S(X_t)$ be the blocks $q \in \mathrm{V}(X_t)$ with height $h(q) \geq c+1$ and let $S(X_t') = \mathrm{V}(X_t') \setminus \{0\}$. Then for all t, the coupling between $(X_t)_{t\geq 0}$ and $(X_t')_{t\geq 0}$ induces

$$\textsc{Tree}(X_t')[S(X_t')] = \textsc{Tree}(X_t)[S(X_t)]. \tag{20}$$

Let $\tau \geq 1$ be the time step Miner 1 capitulates to state B_0. Recall that for a state B'' reachable from state B, $r_\lambda(B, B'')$ denotes the game reward obtained from state B to B''. If B' is an intermediate state between B and B'', we naturally have $r_\lambda(B, B') + r_\lambda(B', B'') = r_\lambda(B, B'')$.

Observation G.10. *For any $\lambda \in \mathbb{R}$ and states B, B', and B'',*

$$r_\lambda(B, B') + r_\lambda(B', B'') = r_\lambda(B, B'').$$

From definition of $\mathcal{V}(B)$ and Observation G.10,

$$\mathrm{r}_{\lambda^*}(B_0, B) + \mathcal{V}(B) = \mathbb{E}\left[\mathrm{r}_{\lambda^*}(B_0, B) + \mathrm{r}_{\lambda^*}(X_0, X_\tau)\right] = \mathbb{E}\left[\mathrm{r}_{\lambda^*}(B_0, X_\tau)\right]. \tag{21}$$

Observation G.11. *For any states B and B',*

$$r_\lambda(B, B') = |A(\mathcal{C}(B')) \cap T_1| - |A(\mathcal{C}(B)) \cap T_1| - \lambda\left(h(\mathcal{C}(B')) - h(\mathcal{C}(B))\right)$$

From Observation G.11,

$$\mathbb{E}\left[\mathrm{r}_{\lambda^*}(B_0, X_\tau)\right] = \mathbb{E}\left[\sum_{i=1}^{h(\mathcal{C}(X_\tau))} \mathbb{1}_{H_i(X_\tau)\in T_1} - \lambda^* h(\mathcal{C}(X_\tau))\right]$$

$$= \mathbb{E}\left[\sum_{i=1}^{c} \mathbb{1}_{H_i(X_\tau)\in T_1} - \lambda^* c + \sum_{i=c+1}^{h(\mathcal{C}(X_\tau))} \mathbb{1}_{H_i(X_\tau)\in T_1} - \lambda^*(h(\mathcal{C}(X_\tau)) - c)\right]$$

$$= \sum_{i=1}^{c}(Pr\left[H_i(X_\tau) \in T_1\right] - \lambda^*) + \mathbb{E}\left[\sum_{i=1}^{h(\mathcal{C}(X_\tau'))} \mathbb{1}_{H_i(X_\tau')\in T_1} - \lambda^* h(\mathcal{C}(X_\tau'))\right] \quad \text{From (20).}$$

From Observation G.11,

$$\mathbb{E}\left[\sum_{i=1}^{h(\mathcal{C}(X'_\tau))} \mathbb{1}_{H_i(X'_\tau)\in T_1} - \lambda^* h(\mathcal{C}(X'_\tau))\right] = \mathbb{E}\left[\mathrm{r}_{\lambda^*}(B_0, X'_\tau)\right]$$

$$= \mathbb{E}\left[\mathrm{r}_{\lambda^*}(B_0, X'_0) + \mathrm{r}_{\lambda^*}(X'_0, X'_\tau)\right] \qquad \text{From Observation G.10,}$$
$$= \mathrm{r}_{\lambda^*}(B_0, B') + \mathcal{V}^{\lambda^*}_{\pi'}(B')$$
$$\leq \mathrm{r}_{\lambda^*}(B_0, B') + \mathcal{V}(B') \qquad \text{From Lemma C.6.}$$

This proves

$$\mathcal{V}(B) \leq \mathcal{V}(B') + \mathrm{r}_{\lambda^*}(B_0, B') - \mathrm{r}_{\lambda^*}(B_0, B) + \sum_{i=1}^{c}\left(Pr\left[H_i(X_\tau) \in T_1\right] - \lambda^*\right)$$

as desired. □

We are ready to show $\mathcal{V}(B_{1,1}) \leq \frac{\alpha}{1-\alpha}$. First, we will observe that the probability Miner 1 adds block 1 to the longest path once the game reaches state $B_{1,1}$ is at most $\frac{\alpha}{1-\alpha}$.

Lemma G.12. *Suppose at state B Miner 1 needs at least $0 \leq \ell \leq 2$ blocks to fork $\mathcal{C}(B)$—i.e., for all $v \in A(\mathcal{C}(B))$, $h(\mathcal{C}(B)) \geq h(v) + |\mathcal{U}(B) \cap (v, \infty)| + \ell - 1$. Then for a mining game starting a state $X_0 = B$, the probability Miner 1 removes $\mathcal{C}(B)$ from the longest path is at most $\left(\frac{\alpha}{1-\alpha}\right)^\ell$.*

For the proof, we will use a well known fact about one-dimensional random walks.

Lemma G.13. *Let $(M_t)_{t\geq 0}$ be a biased one-dimensional random walk with initial state $M_0 = i$ and for $t \geq 1$, $M_t = M_{t-1} - 1$ with probability $\alpha < \frac{1}{2}$ and $M_t = M_{t-1} + 1$ otherwise. Then the probability $M_\tau = 0$ for some $\tau \geq 0$ is $\left(\frac{\alpha}{1-\alpha}\right)^i$.*

Proof. Let $E_n = \cup_{t=0}^{n}\{M_t = 0\}$ be the event $M_\tau = 0$ for some $0 \leq \tau \leq n$. Note $\lim_{n\to\infty} E_n$ denotes the event where $M_\tau = 0$ for some $\tau \geq 0$. Since $E_n \subseteq E_{n+1}$ for all $n \geq 0$, the continuity theorem for probabilities implies

$$Pr\left[\lim_{n\to\infty} E_n | M_0 = i\right] = \lim_{n\to\infty} Pr\left[E_n | M_0 = i\right].$$

Let $p_i = Pr\left[E_n | M_0 = i\right]$. Thus it suffices to compute p_i and take the limit of $n \to \infty$.

Clearly $p_0 = 1$ and $p_i = 0$ for $i \geq n+1$ (since $M_n \geq i - n \geq 1$). For $1 \leq i \leq n$,

$$p_i = \alpha p_{i-1} + (1-\alpha)p_{i+1}.$$

Writing $p_i = \alpha p_i + (1-\alpha)p_i$, we get

$$p_i - p_{i+1} = \frac{\alpha}{1-\alpha}(p_{i-1} - p_i).$$

Claim G.14. *For $1 \leq i \leq n$, $p_i - p_{i+1} = \left(\frac{\alpha}{1-\alpha}\right)^i (1 - p_1)$.*

Proof. The proof is by induction. For the base, $i = 1$, we have $p_1 - p_2 = (1 - p_1)\alpha/(1 - \alpha)$. For $i \geq 2$, the inductive hypothesis gives $p_{i-1} - p_i = \left(\frac{\alpha}{1-\alpha}\right)^{i-1}(1 - p_1)$. Then

$$\begin{aligned} p_i - p_{i+1} &= \left(\frac{\alpha}{1-\alpha}\right)(p_{i-1} - p_i) \\ &= \left(\frac{\alpha}{1-\alpha}\right)^i (1 - p_1). \end{aligned}$$

□

Note $\sum_{i=1}^n (p_i - p_{i+1}) = p_1 - p_{n+1} = p_1$. Thus

$$\begin{aligned} p_1 = \sum_{i=1}^n (p_i - p_{i+1}) &= (1 - p_1) \sum_{i=1}^n \left(\frac{\alpha}{1-\alpha}\right)^i \\ &= (1 - p_1) \frac{\alpha - (1-\alpha)\left(\frac{\alpha}{1-\alpha}\right)^{n+1}}{1 - 2\alpha}. \end{aligned}$$

Because $\alpha < \frac{1}{2}$, $\left(\frac{\alpha}{1-\alpha}\right)^n \to 0$. Rearranging and taking the limit of $n \to \infty$, proves $p_1 = \frac{\alpha}{1-\alpha}$. Next, we claim that $p_{i-1} = \left(\frac{\alpha}{1-\alpha}\right)^{i-1}$ for $i \geq 2$ as $n \to \infty$. As inductive hypothesis, assume the statement holds for $i - 1$ (where the base case is implied from $p_1 = \frac{\alpha}{1-\alpha}$). From Claim G.14,

$$p_i = p_{i-1} - \left(\frac{\alpha}{1-\alpha}\right)^{i-1}(1 - p_1) = \left(\frac{\alpha}{1-\alpha}\right)^{i-1} - \left(\frac{\alpha}{1-\alpha}\right)^{i-1}\left(1 - \left(\frac{\alpha}{1-\alpha}\right)\right) = \left(\frac{\alpha}{1-\alpha}\right)^i.$$

This proves $p_i = \left(\frac{\alpha}{1-\alpha}\right)^i$ as desired. □

Proof of Lemma G.12. Consider a mining game starting at state $X_0 = B$. Let $(Y_t)_{t \geq 0}$ be biased one-dimensional random walk with $Y_0 = \ell$. For $t \geq 1$, let $Y_t = Y_{t-1} - 1$ if Miner 1 creates block t and $Y_t = Y_{t-1} + 1$ if Miner 2 creates block t. Miner 1 can only remove block $\mathcal{C}(B)$ from the longest path if Miner 1 creates ℓ more blocks than Miner 2 since the beginning of the game. In other words, we must have $Y_t = 0$ for some $t \geq 0$. Thus, the probability Miner 1 forks block $\mathcal{C}(B)$ is at most the probability $Y_t = 0$ for some $t \geq 0$. From Lemma G.13,

$$Pr\left[\cup_{t=1}^{\infty}\{Y_t = 0\}\right] = \left(\frac{\alpha}{1-\alpha}\right)^\ell.$$

□

Proposition G.15. $\mathcal{V}(B_{1,1}) \leq \frac{\alpha}{1-\alpha}$.

Proof. The 1-Capitulation of state $B_{1,1}$ is the state B_0 because blocks 1 and 2 can only be at height 1. From the Lemma G.8,

$$\begin{aligned} \mathcal{V}(B_{1,1}) &\leq \mathrm{r}_{\lambda^*}(B_0, B_0) + \mathcal{V}(B_0) + \lim_{t\to\infty} Pr\left[H_1(X_t) \in T_1 | X_0 = B_{1,1}\right] - \lambda^* - \mathrm{r}_{\lambda^*}(B_0, B_{1,1}) \\ &= \lim_{t\to\infty} Pr\left[H_1(X_t) \in T_1 | X_0 = B_{1,1}\right] \end{aligned}$$

The equality observes $\mathrm{r}_{\lambda^*}(B_0, B_0) = 0$, $\mathcal{V}(B_0) = 0$ (Lemma C.4) and $\mathrm{r}_{\lambda^*}(B_0, B_{1,1}) = -\lambda^*$. At state $B_{1,1}$, Miner 2 owns block 2 at height 1 and Miner 1 can only own the block at height 1 if Miner 1 removes block 2 from the longest path. For that, we must have a time $t \geq 1$ where Miner 1 creates one more block than Miner 2 from time 1 to t. From Lemma G.12, the probability of this event is at most $\frac{\alpha}{1-\alpha}$ which proves $\mathcal{V}(B_{1,1}) \leq \frac{\alpha}{1-\alpha}$ as desired. □

Next, we will show that it is optimal for Miner 1 to wait at state $B_{2,0}$. More generally, consider $B_{k,0} := \{(\{0\}, \emptyset), [k], [k]\}$ (Equation 6) as the state where Miner 1 creates and withholds blocks $1, 2, \ldots, k$. Intuitively, there is no advantage for Miner 1 to publish at $B_{k,0}$, for $k \geq 2$, because Miner 1 can safely wait to publish in the subsequent state. That is, even if Miner 2 publishes the subsequent block, Miner 1 would still have sufficient blocks to fork the block Miner 2 just published. We can further extend this intuition for a larger class of states where Miner 2 has published blocks in the longest path. Formally, we define $\mathrm{Ca}(B)$ as the collection of states B' where

- Miner 1 has no blocks in the longest path.
- Miner 1 has $h(\mathcal{C}(B'))$ hidden blocks that cannot reach height $> h(\mathcal{C}(B'))$.
- Capitulating B' at the longest chain result at state B. That is, if Miner 1 forgets all blocks that cannot reach height $> h(\mathcal{C}(B'))$, we obtain state B.

In other words,

$$\begin{aligned}\mathrm{Ca}(B) := \{B' \text{ is a state} : \quad & A(\mathcal{C}(B')) \cap T_1 = \emptyset, \\ & |T_1(B')| - |T_1(B)| = h(\mathcal{C}(B')), \\ & h(\mathcal{C}(B'))\text{-Capitulation of } B' \text{ is state } B\}.\end{aligned}$$

As an example, consider a state $B \in \mathrm{Ca}(B_{2,0})$ where Miner 1 creates blocks 1, 2, 3 and Miner 2 publishes block $4 \to 0$. Observe if $B \in \mathrm{Ca}(B_{k,0})$, then the subsequent state is $B' \in \mathrm{Ca}(B_{k+1,0})$ when Miner 1 creates and withholds the next block or $B' \in \mathrm{Ca}(B_{k-1,0})$ when Miner 2 creates and publishes the next blocks. When $k \geq 2$, we will show there is an optimal strategy that waits at state $B \in \mathrm{Ca}(B_{k,0})$ until the game reaches a state $B' \in \mathrm{Ca}(B_{1,0})$. Once at state B', the optimal strategy is to publish all blocks and fork the whole longest path when α is sufficiently small.

Let's check why the three conditions for a state B to belong to $\mathrm{Ca}(B_{k,0})$ might be necessary for this result. The condition that Miner 1 has no blocks in the longest path hints that forking all the blocks in the longest path is optimal since Miner 1 has no blocks in there. The condition that Miner 1 could fork all blocks in the longest path at B will imply it is optimal for Miner 1 to wait at state B. For example, consider the state where Miner 1 withholds blocks 1, 5, 6 and Miner 2 published $3 \to 2 \to 0$. This state satisfy the first and third conditions, but not the second. As a result, it is not clear how to argue that publishing $6 \to 5 \to 2$ is not optimal. The condition $B_{k,0}$ is the capitulation of B gives the property that the subsequent state is either in $\mathrm{Ca}(B_{k-1,0})$ or $\mathrm{Ca}(B_{k+1,0})$.

Proposition G.16. *Let $\alpha \leq \frac{1}{2}(3 - \sqrt{5})$. At state $B \in \mathrm{Ca}(B_{k,0})$, for $k \geq 2$, it is optimal to play Wait.*

Proof. Define a mining game $(X'_t)_{t\geq 0}$ starting at state $X'_0 = B$ where Miner 1 follows an optimal positive recurrent strategy π^*. Define a mining game $(X_t)_{t\geq 0}$ starting at state $X_0 = B$ where Miner 1 follows a positive recurrent strategy π (to be defined later). Couple the mining games $(X_t)_{t\geq 0}$ and $(X'_t)_{t\geq 0}$ so that Miner k creates block $n \geq 1 + \max \mathrm{V}(B)$ in the first game if and only if Miner k creates block n in the second game. Let τ (respectively τ') be the first time step Miner 1 capitulates to state B_0 in game $(X_t)_{t\geq 0}$ (respectively $(X'_t)_{t\geq 0}$). Without loss of generality, assume $\tau' \geq \tau$. Let T be the first time step $t \geq 1$ where $X_t^{\text{HALF}} \in \mathrm{Ca}(B_{1,0})$. Define π as follows:

- For all $t \leq T-1$, note $X_t^{\text{HALF}} \in \mathrm{Ca}(B_{k,0})$ with $k \geq 2$. Then π plays Wait at state X_t^{HALF}.
- If $X_T^{\text{HALF}} \neq X_T'^{\text{HALF}}$, π plays PublishPath$(T_1(X_T), 0)$ at state X_T^{HALF}. Then π capitulates to state B_0.
- If $X_T^{\text{HALF}} = X_T'^{\text{HALF}}$, π plays the same action as π^* at state X_t^{HALF} for all $t \geq T$.

To show it is optimal for Miner 1 to play Wait at state $B \in \mathrm{Ca}(B_{k,0})$, it suffices to show π is an optimal strategy. It is without loss of generality to assume $B \in \mathrm{Ca}(B_{2,0})$ because if π is optimal for a game starting at $B \in \mathrm{Ca}(B_{2,0})$, π is also optimal for a game starting at state $B \in \mathrm{Ca}(B_{k,0})$, for $k \geq 2$.

Let E be the event $X_T'^{\text{HALF}} = X_T^{\text{HALF}}$ and let E^c be the complement of E. If $X_T'^{\text{HALF}} = X_T^{\text{HALF}}$, then $X'_t = X_t$ for all $t \geq 0$ since π^* played the same actions as π up to time $T-1$ and π plays the same actions as π^* after time T (inclusive). Then

$$\mathbb{E}\left[\mathrm{r}_{\lambda^*}(X'_0, X'_{\tau'}) | E\right] = \mathbb{E}\left[\mathrm{r}_{\lambda^*}(X_0, X_\tau) | E\right].$$

If $X_T'^{\text{HALF}} \neq X_T^{\text{HALF}}$, Miner 1 owns at least one block in $A(\mathcal{C}(X_T'^{\text{HALF}}))$ (because they published at least one block up to time $T-1$). Next, we consider separately the case where Miner 1 has at most one block in the longest path $A(\mathcal{C}(X_T'^{\text{HALF}}))$ and the case where Miner 1 has at least two blocks in the longest path $A(\mathcal{C}(X_T'^{\text{HALF}}))$.

Case 1: Consider the case Miner 1 has at most one block in the longest path $A(\mathcal{C}(X_T'^{\text{HALF}}))$. For this case, $h(\mathcal{C}(X_T'^{\text{HALF}})) = |T_1(X_T)|$. Using the fact B_0 is the $h(\mathcal{C}(X_T'^{\text{HALF}}))$-Capitulation of state $X_T'^{\text{HALF}}$ in Lemma G.8 gives

$$\begin{aligned}
\mathrm{r}_{\lambda^*}(X'_0, X_T'^{\text{HALF}}) + \mathcal{V}(X_T'^{\text{HALF}}) &\leq \mathrm{r}_{\lambda^*}(B_0, B_0) + \mathcal{V}(B_0) \\
&\quad + \sum_{i=1}^{h(\mathcal{C}(X_T'^{\text{HALF}}))} (Pr[H_i(X'_{\tau'}) \in T_1] - \lambda^*) + \lambda^* h(\mathcal{C}(B)) \\
&\leq |T_1(X_T)|(1-\lambda^*) + \lambda^* h(\mathcal{C}(B)).
\end{aligned}$$

The second line observes $\mathcal{V}(B_0) = 0$ (Lemma C.4) and $\mathrm{r}_{\lambda^*}(B_0, B_0) = 0$.

Case 2: Consider the case Miner 1 has at least two blocks in the longest path $A(\mathcal{C}(X_T'^{\text{HALF}}))$. For $k = 1, 2$, let $M_k = \{i \leq h(\mathcal{C}(X_T'^{\text{HALF}})) : H_i(X_T'^{\text{HALF}}) \in T_k\}$ be the heights Miner k owns blocks in the longest path $A(\mathcal{C}(X_T'^{\text{HALF}}))$. Using the fact B_0 is the $h(\mathcal{C}(X_T'^{\text{HALF}}))$-Capitulation of state $X_T'^{\text{HALF}}$ in Lemma G.8 gives

$$\begin{aligned}
\mathrm{r}_{\lambda^*}(X_0', X_T'^{\text{HALF}}) + \mathcal{V}(X_T'^{\text{HALF}}) &\leq \mathrm{r}_{\lambda^*}(B_0, B_0) + \mathcal{V}(B_0) \\
&\quad + \sum_{i \in M_1 \cup M_2} (Pr\left[H_i(X_{\tau'}') \in T_1\right] - \lambda^*) + \lambda^* h(\mathcal{C}(B)) \\
&= |M_1|(1-\lambda^*) + \sum_{i \in M_2} (Pr\left[H_i(X_{\tau'}') \in T_1\right] - \lambda^*) + \lambda^* h(\mathcal{C}(B)) \\
&\leq |T_1(X_T)|(1-\lambda^*) + \lambda^* h(\mathcal{C}(B)) + \sum_{i \in M_2} (Pr\left[H_i(X_{\tau'}') \in T_1\right] - \alpha) \\
&\leq |T_1(X_T)|(1-\lambda^*) + \lambda^* h(\mathcal{C}(B)) + \sum_{i \in M_2} \left(\left(\frac{\alpha}{1-\alpha} \right)^2 - \alpha \right) \\
&\leq |T_1(X_T)|(1-\lambda^*) + \lambda^* h(\mathcal{C}(B))
\end{aligned}$$

The second line observes $|M_1|$ is at most the number of blocks Miner 1 creates up to time τ and $\lambda^* = \max_\pi \text{REV}(\pi) \geq \alpha$ because $\text{REV}(\text{FRONTIER}) = \alpha$. For the third line, note the event $H_i(X_{\tau'}') \in T_1$ for $i \in M_2$ implies Miner 1 forks the longest chain $\mathcal{C}(X_T'^{\text{HALF}})$ at some point in the future. Because Miner 1 publishes at least two blocks in the longest path $A(\mathcal{C}(X_T'^{\text{HALF}}))$ and $X_T^{\text{HALF}} \in \text{Ca}(B_{1,0})$, Miner 1 needs at least two blocks to fork the longest chain $\mathcal{C}(X_T'^{\text{HALF}})$. Thus from Lemma G.12, the probability of $H_i(X_{\tau'}') \in T_1$ for $i \in M_2$ is at most $\left(\frac{\alpha}{1-\alpha}\right)^2$. The last line observes $\left(\frac{\alpha}{1-\alpha}\right)^2 - \alpha \leq 0$ for $\alpha \leq \frac{1}{2}(3-\sqrt{5})$. This concludes the proof of the second case.

Case 1 and 2 proves

$$\begin{aligned}
\mathbb{E}\left[\mathrm{r}_{\lambda^*}(X_0', X_{\tau'}')|E^c\right] &= \mathbb{E}\left[\mathrm{r}_{\lambda^*}(X_0', X_\tau'^{\text{HALF}}) + \mathrm{r}_{\lambda^*}(X_\tau'^{\text{HALF}}, X_{\tau'}')|E^c\right] \\
&= \mathbb{E}\left[\mathrm{r}_{\lambda^*}(X_0', X_\tau'^{\text{HALF}}) + \mathcal{V}(X_\tau'^{\text{HALF}})|E^c\right] \\
&\leq \mathbb{E}\left[|T_1(X_T)|(1-\lambda^*) + \lambda^* h(\mathcal{C}(B))|E^c\right]
\end{aligned}$$

as desired. We claim

$$\mathbb{E}\left[\mathrm{r}_{\lambda^*}(X_0, X_\tau)|E^c\right] = \mathbb{E}\left[|T_1(X_T)|(1-\lambda^*) + \lambda^* h(\mathcal{C}(B))|E^c\right].$$

Recall, at state X_T^{HALF}, π takes action PublishPath$(T_1(X_T), 0)$ and capitulates to state B_0 whenever $X_T^{\text{HALF}} \neq X_T'^{\text{HALF}}$. The game reward from state X_0 to X_τ is $|T_1(X_T)|(1-\lambda^*)$, because Miner 1 adds $|T_1(X_T)|$ blocks in the longest path, plus $\lambda^* h(\mathcal{C}(B))$, because Miner 2 has $h(\mathcal{C}(B))$ blocks removed from the longest path. Thus

$$\mathbb{E}\left[\mathrm{r}_{\lambda^*}(X_0, X_\tau)|E^c\right] = \mathbb{E}\left[|T_1(X_T)|(1-\lambda^*) + \lambda^* h(\mathcal{C}(B))|E^c\right] \geq \mathbb{E}\left[\mathrm{r}_{\lambda^*}(X_0', X_{\tau'}')|E^c\right].$$

We conclude $\mathcal{V}_\pi^{\lambda^*}(B) \geq \mathcal{V}(B)$. From Bellman's principle of optimality, π is optimal. □

Because $B_{2,0} \in \text{Ca}(B_{2,0})$, we proved that once the mining game reaches state $B_{2,0}$, Miner 1 will prefer to wait until it reaches a state in $B \in \text{Ca}(B_{1,0})$. Next, we prove that at state $B \in \text{Ca}(B_{1,0})$, it is optimal for Miner 1 to publish all their hidden blocks (when α is sufficiently small).

Proposition G.17. *Let $\alpha \leq \frac{1}{2}(3-\sqrt{5})$. At state $B \in \mathrm{Ca}(B_{1,0})$, it is either optimal to take action Wait or PublishPath$(T_1(B), 0)$.*

Proof. If waiting at state B is not optimal, then one of the following is optimal:

1. Miner 1 takes action PublishPath$(T_1(B), 0)$.

2. Miner 1 takes action PublishPath(Q, v) for some $Q \subset T_1(B)$.

We will show the action in the first case is strictly better than the action in the second case.

In the first case, let B' be the subsequent state after Miner 1 plays PublishPath$(T_1(B), 0)$. From Proposition G.2, $\mathcal{V}(B') \geq 0$ since Miner 1 can capitulate to state B_0 from state B'. The game reward from state B to B' is $\lambda^* h(\mathcal{C}(B))$, because Miner 1 removes $h(\mathcal{C}(B))$ of Miner 2's blocks in the longest path, plus $(1-\lambda^*)(h(\mathcal{C}(B))+1)$, because Miner 1 adds $h(\mathcal{C}(B))+1$ blocks in the longest path. Thus the game reward from state B until Miner 1 capitulates to state B_0 is

$$\mathcal{V}(B) \geq h(\mathcal{C}(B)) + (1-\lambda^*) + \mathcal{V}(B') \geq h(\mathcal{C}(B)) + (1-\lambda^*).$$

In the second case, let B' be the subsequent state after Miner 1 takes action PublishPath(Q, v). Let $M_k = \{i \leq h(\mathcal{C}(B')) : H_i(B') \in T_k\}$ be the heights of the blocks Miner k owns in the longest path $A(\mathcal{C}(B'))$. The $h(\mathcal{C}(B'))$-Capitulation of B' is the state B_0. Thus from Lemma G.8,

$$\begin{aligned}
\mathcal{V}(B) &= \mathrm{r}_{\lambda^*}(B, B') + \mathcal{V}(B') = \mathrm{r}_{\lambda^*}(B_0, B') + \mathcal{V}(B') - \mathrm{r}_{\lambda^*}(B_0, B) \\
&\leq \mathrm{r}_{\lambda^*}(B_0, B_0) + \mathcal{V}(B_0) + \sum_{i \in M_1 \cup M_2} \left(Pr\left[H_i(X_\tau) \in T_1 | X_0 = B'\right] - \lambda^*\right) - \mathrm{r}_{\lambda^*}(B_0, B) \\
&= |M_1|(1-\lambda^*) + \sum_{i \in M_2} \left(Pr\left[H_i(X_\tau) \in T_1 | X_0 = B'\right] - \lambda^*\right) + \lambda^* h(\mathcal{C}(B)) \\
&\leq |M_1|(1-\lambda^*) + \sum_{i \in M_2} \left(\left(\frac{\alpha}{1-\alpha}\right)^2 - \alpha\right) + \lambda^* h(\mathcal{C}(B)) \\
&\leq |M_1|(1-\lambda^*) + \lambda^* h(\mathcal{C}(B)) \\
&= h(\mathcal{C}(B)) + (1-\lambda^*)
\end{aligned}$$

The third line observes $\mathcal{V}(B_0) = 0$ (Lemma C.4) and $\mathrm{r}_{\lambda^*}(B_0, B_0) = 0$. The fourth line observes $\lambda^* = \max_\pi \textsc{Rev}(\pi) \geq \alpha$ and Miner 1 needs at least 2 blocks to fork $\mathcal{C}(B')$ from the longest path. Thus from Lemma G.8, the probability Miner 1 will own the block at height $i \in M_2$ is at most $\left(\frac{\alpha}{1-\alpha}\right)^2$. The fifth line observes $\left(\frac{\alpha}{1-\alpha}\right)^2 \leq \alpha$ for $\alpha \leq \frac{1}{2}(3-\sqrt{5})$. The sixth line observes Miner 1 owns at most $|T_1(B)| = h(\mathcal{C}(B)) + 1$ blocks in the longest path. The chain of inequalities witnesses taking action PublishPath$(T_1(B), 0)$ is strictly better than taking action PublishPath(Q, v) for some $Q \subset T_1(B)$. □

We are ready to show that for any state $B \in \mathrm{Ca}(B_{1,0})$, it is optimal for Miner 1 to play PublishPath$(T_1(B), 0)$ (when α is sufficiently small).

Proposition G.18. *Let $\frac{\alpha(1-\alpha)^2}{(1-2\alpha)^2} \leq c+1$ where $c = h(\mathcal{C}(B))$. At state $B \in \mathrm{Ca}(B_{1,0})$, it is optimal for Miner 1 to take action PublishPath$(T_1(B), 0)$.*

Proof. Let $\lambda^* = \max_\pi \textsc{Rev}(\pi)$. The proof will be by induction on c. For the base case, we will proof the statement holds for $c \geq C$ where C is an absolute constant. Then we will prove the statement when $c = C-1, C-2, \ldots, 0$. If Miner 1 takes action PublishPath$(T_1(B), 0)$, the game reward is $(1-\lambda^*)(c+1)$, because Miner 1 adds $|T_1(B)| = c+1$ blocks in the longest path, plus $\lambda^* c$, because Miner 2 has c blocks removed from the longest path. Therefore,

$$\mathcal{V}(B) \geq c + (1-\lambda^*) + \mathcal{V}(B') \geq c + (1-\lambda^*)$$

where B' is the subsequent state. The inequality observes $\mathcal{V}(B') \geq 0$ because Miner 1 can capitulate to B_0 from state B' (Proposition G.2). By assumption $\alpha(1-\alpha)^2 \leq (1-2\alpha)^2$ since $c \geq 0$. Solving the inequality gives $\alpha \leq \frac{1}{2}(3-\sqrt{5})$. From Proposition G.17, the only alternative to playing PublishPath$(T_1(B), 0)$ is to play Wait. Let π be any positive recurrent strategy that plays Wait at state B. Let $Z_1 \in \mathrm{Ca}(B_{2,0})$ be the subsequent state if Miner 1 creates and withholds the next block and let $Z_2 \in \mathrm{Ca}(B_0)$ be the subsequent state when Miner 1 creates and publishes the next block. Then

$$\mathcal{V}_\pi^{\lambda^*}(B) = \alpha\,\mathcal{V}_\pi^{\lambda^*}(Z_1) + (1-\alpha)(\mathcal{V}(Z_2) - \lambda^*) \leq \alpha\,\mathcal{V}(Z_1) + (1-\alpha)(\mathcal{V}(Z_2) - \lambda^*)$$

where the inequality observes $\mathcal{V}_\pi^{\lambda^*}(B') \leq \mathcal{V}(B')$ for any states B' and strategies π (Lemma C.6). Observe the $(c+1)$-Capitulation of state Z_2 is state B_0. Then from Lemma G.8,

$$\begin{aligned}
\mathcal{V}(Z_2) &\leq \mathrm{r}_{\lambda^*}(B_0, B_0) + \mathcal{V}(B_0) + \sum_{i=1}^{c+1} (Pr\left[H_i(X_\tau) \in T_1 | X_0 = Z_2\right] - \lambda^*) - \mathrm{r}_{\lambda^*}(B_0, Z_2) \\
&= \sum_{i=1}^{c+1} (Pr\left[H_i(X_\tau) \in T_1 | X_0 = Z_2\right] - \lambda^*) + \lambda^*(c+1) \\
&\leq (c+1)\frac{\alpha}{1-\alpha}.
\end{aligned}$$

The first line is Lemma G.8 applied to states Z_2 and B_0. The second line observes Miner 2 owns $c+1$ blocks in the longest path. Thus $\mathrm{r}_{\lambda^*}(B_0, Z_2) = -\lambda^*(c+1)$. The third line observes Miner 1 will only own the block at height $i \leq c+1$ if Miner 1 can fork the longest chain $\mathcal{C}(Z_2)$. But Miner 1 needs at least one block to fork the longest chain $\mathcal{C}(Z_2)$. From Lemma G.12, the probability Miner 1 forks $\mathcal{C}(Z_2)$ is at most $\frac{\alpha}{1-\alpha}$.

Next, we upper bound $\mathcal{V}(Z_1)$. Observe the c-Capitulation of state Z_1 is state $B_{2,0}$. From Lemma G.8,

$$\mathcal{V}(Z_1) \leq \mathrm{r}_{\lambda^*}(B_0, B_{2,0}) + \mathcal{V}(B_{2,0}) - \mathrm{r}_{\lambda^*}(B_0, Z_1) = \mathcal{V}(B_{2,0}) + c(1-\lambda^*) + \lambda^* c = \mathcal{V}(B_{2,0}) + c$$

Recall the revenue $\lambda^* = \max_\pi \textsc{Rev}(\pi)$ of the optimal strategy is at least α and at most $\frac{\alpha}{1-\alpha}$ (Corollary 5.7). Then

$$\alpha \leq \lambda^* \leq \frac{\alpha}{1-\alpha} \tag{22}$$

The bounds on λ^*, allow us to derive an upper bound on $\mathcal{V}(B_{2,0})$.

Claim G.19. $\mathcal{V}(B_{1,0}) \leq 1$.

Proof. At state B_0, the subsequent state is either $B_{1,0}$ or $B_{0,1}$. Then

$$0 = \mathcal{V}(B_0) = \alpha\,\mathcal{V}(B_{1,0}) + (1-\alpha)(\mathcal{V}(B_{0,1}) - \lambda^*).$$

From Proposition G.1, $\mathcal{V}(B_{0,1}) \geq 0$. Thus

$$\mathcal{V}(B_{1,0}) = \lambda^* \frac{1-\alpha}{\alpha} \leq 1.$$

□

Claim G.20. $\mathcal{V}(B_{2,0}) \leq \frac{1}{\alpha} + 1$.

Proof. At state $B_{1,0}$, if Miner 1 takes action Wait, the subsequent state is either $B_{2,0}$ or $B_{1,1}$. Then

$$1 \geq \mathcal{V}(B_{1,0}) \geq \alpha\,\mathcal{V}(B_{2,0}) + (1-\alpha)(\mathcal{V}(B_{1,1}) - \lambda^*) \geq \alpha\,\mathcal{V}(B_{2,0}) - (1-\alpha)\lambda^*$$

The first inequality is Claim G.19. The second inequality is Bellman's principle of optimality (Lemma C.6). The third inequality uses the fact $\mathcal{V}(B_{1,1}) \geq 0$ (Proposition G.1). Solving for $\mathcal{V}(B_{2,0})$ gives

$$\mathcal{V}(B_{2,0}) \leq \frac{1}{\alpha} + \frac{1-\alpha}{\alpha}\lambda^* \leq \frac{1}{\alpha} + 1$$

□

From Claim G.20, we obtain

$$\mathcal{V}(Z_1) \leq \frac{1}{\alpha} + 1 + c$$

Combining the upper bounds for $\mathcal{V}(Z_1)$ and $\mathcal{V}(Z_2)$, we obtain

$$\mathcal{V}_\pi^{\lambda^*}(B) \leq 1 + \alpha + \alpha c + \alpha(c+1) - \lambda^*(1-\alpha) \leq c + (1-\lambda^*) + 2\alpha + \frac{\alpha^2}{1-\alpha} - c(1-2\alpha).$$

We would like to conclude by saying $V_\pi^{\lambda^*}(B) < \mathcal{V}(B)$. Unfortunately, the above upper bound on $\mathcal{V}_\pi^{\lambda^*}(B)$ is not strong enough for all c. Fortunately, for sufficiently large c, we obtain

$$\mathcal{V}_\pi^{\lambda^*}(B) < \mathcal{V}(B).$$

as desired. By Bellman's principle of optimality, we have a certificate that it is sub-optimal to wait at state B. The intuition is that Miner 1 *risks* not publishing blocks $T_1(B)$ by waiting at state B (because if Miner 2 creates and publish the next block, the probability Miner 1 can publish $T_1(B)$ is at most $\frac{\alpha}{1-\alpha}$). Thus when $c = |T_1(B)| - 1$ is large, the cost of not publishing $T_1(B)$ outweighs the expected gain from not publishing $T_1(B)$ at state B.

To extend the proof to smaller c, we will do induction on c. The work above proves the base case for any $c > \frac{\alpha(2-\alpha)}{(1-\alpha)(1-2\alpha)}$. As inductive hypothesis, assume for all states $B' \in \mathrm{Ca}(B_{1,0})$ with $h(\mathcal{C}(B')) \geq c+1$, it is optimal for Miner 1 to take action $\mathrm{PublishPath}(T_1(B'), 0)$. Note $\max T_1(B')$ becomes a checkpoint. From Theorem 5.15, there is an optimal checkpoint recurrent strategy which implies Miner 1 capitulates to state B_0 once a new checkpoint is defined. Thus

$$\mathcal{V}(B') = h(\mathcal{C}(B')) + (1-\lambda^*).$$

As before, let $B \in \text{Ca}(B_{1,0})$ and $c = h(\mathcal{C}(B))$. To improve our upper bound on $\mathcal{V}_\pi^{\lambda^*}(B)$, we will improve our upper bound on $\mathcal{V}(Z_1)$. In fact, our inductive hypothesis *will allow us to derive a closed form on* $\mathcal{V}(Z_1)$. Recall $Z_1 \in \text{Ca}(B_{2,0})$. Let $(X_t)_{t\geq 0}$ be a mining game starting at state $X_0 = Z_1$. From Proposition G.16, Miner 1 will wait until the first time step τ where $X_\tau^{\text{Half}} \in \text{Ca}(B_{1,0})$. Note $h(\mathcal{C}(X_\tau^{\text{Half}})) \geq c+1$ because Miner 2 published at least one block in the longest path since state X_0. From the inductive hypothesis, Miner 1 will take action PublishPath$(T_1(X_\tau), 0)$ at state X_τ^{Half} and

$$\mathcal{V}(X_\tau^{\text{Half}}) = h(\mathcal{C}(X_\tau^{\text{Half}})) + (1-\lambda^*).$$

From state Z_1 to X_τ^{Half}, Miner 2 publishes $h(\mathcal{C}(X_\tau)) - h(\mathcal{C}(Z_1))$ blocks. Thus the game reward from state Z_1 to X_τ^{Half} is

$$\mathrm{r}_{\lambda^*}(Z_1, X_\tau^{\text{Half}}) = -\lambda^*(h(\mathcal{C}(X_\tau^{\text{Half}})) - h(\mathcal{C}(Z_1))).$$

The random variable $h(\mathcal{C}(X_\tau^{\text{Half}})) - h(\mathcal{C}(Z_1))$ denotes the number of blocks Miner 2 creates from state $Z_1 \in \text{Ca}(B_{2,0})$ until state $X_\tau^{\text{Half}} \in \text{Ca}(B_{1,0})$. From Lemma D.2, we can construct a coupling $h(\mathcal{C}(X_\tau^{\text{Half}})) - h(\mathcal{C}(Z_1))$ with a random walk starting at state $Y_0 = 1$ and for $t \geq 1$, $Y_t = Y_{t-1}+1$ whenever Miner 1 creates the block at time t (with probability α); otherwise, $Y_t = Y_{t-1} - 1$. Then $h(\mathcal{C}(X_\tau^{\text{Half}})) - h(\mathcal{C}(Z_1))$ counts the number of time steps $t \leq \tau$ where $Y_t = Y_{t-1} - 1$ (i.e., the number of blocks Miner 2 creates). Thus

$$\mathbb{E}\left[h(\mathcal{C}(X_\tau^{\text{Half}})) - h(\mathcal{C}(Z_1))\right] = \frac{1-\alpha}{1-2\alpha}.$$

Note $h(\mathcal{C}(Z_1)) = h(\mathcal{C}(B)) = c$ since no blocks are published from state B to state Z_1. Combining the work above, we derive

$$\begin{aligned}\mathcal{V}(Z_1) &= \mathbb{E}\left[\mathrm{r}_{\lambda^*}(Z_1, X_\tau^{\text{Half}}) + \mathcal{V}(X_\tau^{\text{Half}})\right]\\ &= \mathbb{E}\left[h(\mathcal{C}(Z_1)) + (1-\lambda^*)(h(\mathcal{C}(X_\tau^{\text{Half}})) - h(\mathcal{C}(Z_1))) + (1-\lambda^*)\right]\\ &= c + (1-\lambda^*)\frac{1-\alpha}{1-2\alpha} + (1-\lambda^*)\end{aligned}$$

Substituting $\mathcal{V}(Z_1)$ and our upper bound for $\mathcal{V}(Z_2)$ in our upper bound for $\mathcal{V}_\pi^{\lambda^*}(B)$,

$$\begin{aligned}\mathcal{V}_\pi^{\lambda^*}(B) &\leq \alpha\,\mathcal{V}(Z_1) + (1-\alpha)(\mathcal{V}(Z_2) - \lambda^*)\\ &\leq \alpha\left(c + (1-\lambda^*)\frac{1-\alpha}{1-2\alpha} + 1 - \lambda^*\right) + (1-\alpha)\left((c+1)\frac{\alpha}{1-\alpha} + 1 - \lambda^* - 1\right)\\ &= \alpha c + (1-\alpha)c - (1-\alpha)c + (1-\lambda^*) + \alpha(1-\lambda^*)\frac{1-\alpha}{1-2\alpha} + \alpha(c+1) - (1-\alpha)\\ &\leq c + (1-\lambda^*) + \alpha\frac{(1-\alpha)^2}{1-2\alpha} - c(1-2\alpha) - (1-2\alpha) \quad \text{Because } \lambda^* \geq \alpha\\ &\leq \mathcal{V}(B) + \alpha\frac{(1-\alpha)^2}{1-2\alpha} - (c+1)(1-2\alpha) \quad \text{Because } \mathcal{V}(B) \geq c + (1-\lambda^*).\end{aligned}$$

By assumption, $\frac{\alpha(1-\alpha)^2}{(1-2\alpha)^2} \leq c+1$. Then

$$\mathcal{V}_\pi^{\lambda^*}(B) \leq \mathcal{V}(B)$$

as desired. The inequality witnesses that it is optimal for Miner 1 to take action $\text{PublishPath}(T_1(B), 0)$ at state $B \in \text{Ca}(B_{1,0})$. $\square$

We can now conclude that for $\alpha \leq 0.307979$, FRONTIER is an optimal strategy.

Proof of Theorem 6.1. Assume Miner 1 follows an optimal strategy. From Theorem 5.15, it is without loss of generality to assume such strategy is checkpoint recurrent and positive recurrent. Starting from state B_0, the subsequent state is either $B_{0,1}$ when Miner 2 creates block 1 and publishes $1 \to 0$, or $B_{1,0}$ when Miner 1 creates block 1, but yet did not decide between publishing $1 \to 0$ or waiting for the next round. From Corollary G.3, at state $B_{0,1}$, it is optimal for Miner 1 to capitulate to state B_0 (for any α). At Proposition G.18, let $B = B_{1,0}$. Then $c = 0$ and as long as $\alpha \leq 0.307979$, we satisfy the condition $\frac{\alpha(1-\alpha)^2}{(1-2\alpha)^2} \leq 1$. Thus it is optimal for Miner 1 to publish $1 \to 0$ at $B_{1,0}$. Once Miner 1 take this action, from Definition 5.1, block 1 becomes a checkpoint. Since Miner 1's strategy is checkpoint recurrent, Miner 1 capitulates to state B_0 once block 1 becomes a checkpoint. Thus Miner 1 capitulates to state B_0 by the end of round 1 with probability 1. Additionally, up to round 1, Miner 1 is taking the same actions as FRONTIER. This witnesses FRONTIER is an optimal strategy as desired. $\square$

H Table of Notation

Symbol	Domain	Usage
n	$\mathbb{N}_+$	Round, or block created during a round
γ_n	$\{1,2\}$	The miner during round n
T_i	$2^{\mathbb{N}_+}$	The set of rounds where Miner i mines
$\textsc{Tree}$	Directed trees with a single sink	Set of all published blocks, evolves over time
V	$\mathbb{N}_+$	Nodes in $\textsc{Tree}$
E	$\mathbb{N}_+$	Edges in $\textsc{Tree}$
$\mathcal{U}_i$	$\mathbb{N}_+$	Blocks created by Miner i, but not yet published, evolves over time
$A(b)$	$2^{\mathbb{N}}$	Ancestors of block $b \in V$
$\textsc{Succ}(b)$	$2^{\mathbb{N}_+}$	Successors of block $b \in V$
$h(b)$	$\mathbb{N}_+$	$:= \lvert A(b)\rvert - 1$, height of block $b \in V$
$\mathcal{C}$	$\mathbb{N}_+$	The longest chain and the block $v \in V$ with highest height
H_i	$\mathbb{N}_+$	Block $v \in A(\mathcal{C})$ with height $i = h(v)$.
(B)	N/A	Modifies $\{\textsc{Tree}, V, E, \mathcal{U}_i, \mathcal{C}, H\}$ to denote its state at B.
$B^{\textsc{Half}}$	N/A	Denote prior state prior to state B after the most recent block was created, Miner 2 has acted, but Miner 1 has not.
$r^k(B, B')$	$\mathbb{N}$	$:= \lvert A(\mathcal{C}(B')) \cap T_k\rvert - \lvert A(\mathcal{C}(B)) \cap T_k\rvert$, Miner k reward from state B to B'
$\mathrm{r}_\lambda(B, B')$	$\mathbb{R}$	$:= (1-\lambda) r^1(B, B') - \lambda r^2(B, B')$, game reward from state B to B'
$\textsc{Rev}(\pi)$	$\mathbb{R}_+$	$:= \mathbb{E}\left[\liminf_{n\to\infty} \frac{r^1(X_0,X_n)}{r^1(X_0,X_n)+r^2(X_0,X_n)} \mid X_0 = B_0\right]$, revenue of strategy π
$\textsc{Succ}_{\mathrm{s}}(B)$	Set of states	$:= \{B' : (B')^{\textsc{Half}} = B\}$, all states B' reachable from state B when Miner 1 takes a single action at B.
$\delta^k(B)$	$\mathbb{N}$	$:= \max_{B' \in \textsc{Succ}_{\mathrm{s}}(B)} \lvert r^k(B, B')\rvert$, Miner k potential reward at state B.
$\mathcal{V}_\pi^\lambda(B)$	$\mathbb{R}$	$:= \mathbb{E}\left[r_\lambda(X_0, X_\tau)\bar{X}_0 = B\right]$ where τ is the first time step Miner 1 capitulates to state B_0
$\mathcal{V}(B)$	$\mathbb{R}$	$:= V_{\pi^*}^{\textsc{Rev}(\pi^*)}(B)$ where $\pi^* = \arg\max_\pi \textsc{Rev}(\pi)$

Table 1: Notation.