arXiv:2210.11791v1 [cs.DC] 21 Oct 2022

Cross-chain Swaps with Preferences

Eric Chan
University of California at Riverside

Marek Chrobak
University of California at Riverside

Mohsen Lesani
University of California at Riverside

***Abstract*—Extreme valuation and volatility of cryptocurrencies require investors to diversify often which demands secure exchange protocols. A cross-chain swap protocol allows distrusting parties to securely exchange their assets. However, the current models and protocols assume predefined user preferences for acceptable outcomes. This paper presents a generalized model of swaps that allows each party to specify its preferences on the subsets of its incoming and outgoing assets. It shows that the existing swap protocols are not necessarily a strong Nash equilibrium in this model. It characterizes the class of swap graphs that have protocols that are safe, live and a strong Nash equilibrium, and presents such a protocol for this class. Further, it shows that deciding whether a swap is in this class is NP-hard through a reduction from 3SAT, and further is Σ_2^{P}-complete through a reduction from $\exists\forall\mathsf{DNF}$.**

1. Introduction

With the rise of a staggering number of cryptocurrencies, a challenging technical problem is secure exchange between them. In particular, the values of these currencies are extremely volatile; therefore, investors tend to diversify often and need to exchange their currencies. However, these currencies are hosted by distinct distributed blockchains and trading across blockchains is not atomic by default. This has led to the development of *cross-chain swap protocols* [19], [8], [10], [18], [13], [33] to allow parties with mutual distrust to securely exchange their assets.

In a pioneering work, Herlihy [19] formalizes a cross-chain swap as a directed graph where vertices represent parties, and arcs represent assets to be exchanged. An execution of a *swap graph* is represented as the subset of arcs that are triggered in that execution. The *outcome* for each party is captured in five *predefined classes*: DEAL, NODEAL, DISCOUNT, FREERIDE, and UNDERWATER. The classes DEAL and NODEAL represent outcomes for a party where respectively, all and none of the arcs of that party are triggered. The class DISCOUNT represents outcomes where some of the outgoing arcs are not triggered, and FREERIDE represents outcomes where at least one incoming but no outgoing arc is triggered. All the above classes are considered acceptable outcomes for every party. The class UNDERWATER captures all unacceptable outcomes. It represents the set of outcomes where at least one outgoing arc is triggered but not all incoming arcs are. Given this model of outcomes, he presented a *protocol based on hashed time-locks* and proved it to be *safe, live and a strong Nash equilibrium*.

However, as noted by the original proposal [19], there are outcomes in the class UNDERWATER that a party may be interested in accepting. In other words, a party may find profitable to exchange *some* of their outgoing assets for *some* of their incoming assets, depending on the assets.

As an example, suppose Alina and Bohdan are in the market for new outfits. Alina has a white shirt and white pants that she would like to trade for a black shirt and black pants. Coincidentally, Bohdan has exactly these items and is interested in the reverse trade. However, both of them would actually prefer to have one white article of clothing and one black article of clothing. Thus, it would be preferable for both parties to only swap the shirts or only swap the pants, although it is acceptable to swap both.

As another example, suppose Alina has ingredients to make a carrot cake. However, Alina is in the mood for vegetable soup, so she finds a transaction to exchange her carrot cake ingredients for vegetable soup ingredients. Although Alina finds this entire swap acceptable, she would actually prefer to forego the offered corn if it means she can retain her carrots to add to her soup.

The questions are how the *preference model* can be generalized so that each party can specify its personal preference over its outcomes, how should the safety, liveness and Nash equilibria definitions be adapted, and whether existing protocols preserve the properties. questions that were left open [19]. We formally define a *general swap model* that allows each party to specify its preference on its set of possible outcomes, and formally define *generalized properties* of swap protocols. We prove that in this model, Herlihy's protocol remains safe, live, and a Nash equilibrium but not necessarily a Strong Nash equilibrium. We present a *characterization* of the class of swap graphs that admit protocols that are safe, live and a Strong Nash equilibrium. The proof of correctness of the characterization embodies such a protocol. We further show that it is *NP-hard* to decide whether a swap graph is in this class through a reduction from 3SAT, and then further tighten the complexity classification and and show that it is Σ_2^{P}-complete through a reduction from $\exists\forall\mathsf{DNF}$. Although the second result subsumes the first, the first proof is simpler and we present it for exposition. The practical implications of these results are that not all swap graphs are acceptable, and deciding whether a proposed swap graph is acceptable is of high complexity.

In summary, this paper makes the following contributions.

- A general swap model with user-defined preferences of outcomes. (section 2)
- Generalized safety, liveness and Nash equilibria properties for swap protocols (section 2), and a study of whether existing protocols satisfy these properties (section 3).
- A characterization of swap graphs that admit protocols with these properties, and an instance of such protocols (section 4).
- NP-hardness and Σ_2^{P}-completeness of deciding whether a swap graph falls into this class (section 5 and section 6).

2. Swap Systems

As discussed in the introduction, Herlihy's model [19] for cross-chain swaps assumed that the rational behavior of participating parties is determined by preferences between five types of outcomes: DEAL, NODEAL, DISCOUNT, FREERIDE, and UNDERWATER. These preferences were assumed to be shared by all parties, and can be interpreted as a simple partial order on all possible outcomes. Some of these preferences are natural; for example, in DISCOUNT a party receives all incoming assets without trading all outgoing assets, making it preferable to DEAL. It is less clear why the outcomes in FREERIDE should be incomparable to DEAL. And even some outcomes designated as UNDERWATER in [19] may well be preferable to DEAL. As an example, suppose that Alina possesses items `A` and `B` that she values at \$10 and \$12, and Bohdan possesses items `X` and `Y` that Alina values at \$11 and \$14. Alina would then accept to join the swap that allows her to swap both `A` and `B` for Bohdan's `X` and `Y`, but she would be even happier if she could swap only `A` for `Y` instead.

To represent such individual preferences, we now refine Herlihy's model by allowing each party to specify a partial order on all her possible outcomes of a protocol. Our model is very general in that (unlike in the example above) a party's preferences are not determined by numerical values of individual assets, but rather involve comparing directly whole sets of assets. Such an approach can capture dependencies between assets, say when a party may value a set of assets higher or lower than the sum of their individual values. Say that Alina owns a power drill and a shovel, while Bohdan is in possession of a pair of skis. Alina would not swap any of her items for any single ski, but she may be happy to swap both of her items for the pair.

Swap Systems. A *swap system* is specified by a pair $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ consisting of a digraph $\mathcal{D}$ that represents the prearranged asset transfers and a collection $\mathcal{P}$ of posets that specifies the preferences of each involved party among all of its potential outcomes. Next, we give a formal definition of these two components of $\mathcal{S}$.

Digraph $\mathcal{D} = (V, A)$ is called a *swap digraph*. Each vertex $v \in V$ represents a party that participates in the swap, and each arc $(u, v) \in A$ represents an asset that is to be transferred from party u to party v. By A_v^{in} and A_v^{out} we will denote the sets of vertex v's incoming and outgoing arcs, respectively. If $(x, v) \in A_v^{in}$ then x is called an *in-neighbor* of v, and if $(v, x) \in A_v^{out}$ then x is called an *out-neighbor* of v. Throughout the paper we assume that $\mathcal{D}$ does not have multiple arcs[1]. We also assume that $\mathcal{D}$ is weakly connected (otherwise a swap can be arranged for each connected component separately). To exclude some degenerate scenarios, we also assume that $|V| \geq 2$ and that $A_v^{in} \neq \emptyset$ and $A_v^{out} \neq \emptyset$ for each $v \in V$.

An *outcome* of a party $v \in V$ is a pair $\omega = \langle \omega^{in} \,|\, \omega^{out} \rangle$, where $\omega^{in} \subseteq A_v^{in}$ and $\omega^{out} \subseteq A_v^{out}$. An outcome represents the sets of acquired and traded assets, ω^{in} and ω^{out} respectively. The set of all possible outcomes of v will be denoted Ω_v. To reduce clutter, instead of arcs, in $\langle \omega^{in} \,|\, \omega^{out} \rangle$ we will often list only the corresponding in-neighbors and out-neighbors of v; for example, instead of $\langle \{(x, v), (y, v)\} \,|\, \{v, z\} \rangle$ we will write $\langle x, y \,|\, z \rangle$.

The collection $\mathcal{P} = \{\mathcal{P}_v\}_{v \in V}$ consists of *preference posets*. The preference poset of a party $v \in V$ is $\mathcal{P}_v = (\Omega_v, \preceq_v)$, where $\preceq_v$ is a partial order on Ω_v. We will write $\omega \prec_v \omega'$ if $\omega \preceq_v \omega'$ and $\omega \neq \omega'$. This poset naturally represents v's evaluation of its potential outcomes; that is, relation $\omega \preceq_v \omega'$ holds if v views outcome ω' to be better than outcome ω. The outcome where v does not participate in any transfer is $\text{NODEAL}_v = \langle \varnothing \,|\, \varnothing \rangle$ and the outcome where all of v's transfers are realized is $\text{DEAL}_v = \langle A_v^{in} \,|\, A_v^{out} \rangle$. Each preference poset $\mathcal{P}_v$ is assumed to have the following properties:

(p.1) DEAL is better than NODEAL: $\text{NODEAL}_v \prec_v \text{DEAL}_v$. Naturally, each party prefers swapping all assets over being completely excluded, as otherwise it would not even join the swap system.

(p.2) Inclusive Monotonicity: $(\omega_1^{in} \subseteq \omega_2^{in} \wedge \omega_2^{out} \subseteq \omega_1^{out}) \Rightarrow \omega_1 \preceq_v \omega_2$, for every two outcomes $\omega_1, \omega_2 \in \Omega_v$. That is, it's better to receive more assets and to trade fewer assets[2].

The preference pairs $\omega_1 \prec_v \omega_2$ that are determined by rules (p.1) and (p.2) above will be called *generic*. The size of the preference poset may be exponentially large with respect to the size of the swap digraph $\mathcal{D}$, but it is not necessary for a party to specify generic preferences as they are implied from the above rules. Therefore, throughout the paper, we assume that $\mathcal{P}_v$ is specified by its *generator set*, which is a subset of its non-generic preference pairs that, together with the generic pairs and transitivity, generate the whole poset. A generator set of a poset may not be unique. We use this convention in our examples and running time bounds. (This does not affect our hardness results — they hold even if the preference poset of each party is specified by listing *all* preference pairs.)

An outcome $\omega \in \Omega_v$ is called *acceptable* if $\omega \succeq$

1. This assumption is only for convenience – our model and results trivially extend to multi-digraphs, although this requires more cumbersome notation and terminology.

2. Duuh.

$\textsc{NoDeal}_v$[3]. The set of acceptable outcomes of a node v will be denoted $\mathcal{A}_v$.

Throughout the paper, we will often omit subscript v in the notation for outcomes $\textsc{Deal}_v$ and $\textsc{NoDeal}_v$ (and others as well) and in relation $\prec_v$, if v is implicit in the context or irrelevant. On the other hand, if any ambiguity may arise, we will sometimes add a superscript to some notations specifying the digraph under consideration; for example we will write $\textsc{Deal}_v^{\mathcal{D}}$ to specify that outcome $\textsc{Deal}_v^{\mathcal{D}}$ is with respect to digraph $\mathcal{D}$.

Protocols. Given a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$, a *swap protocol* $\mathbb{P}$ for $\mathcal{S}$ specifies actions of each party over time, in particular it determines how assets change hands. Initially, an asset represented by an arc $(u, v) \in A$ is in the possession of u, and, when $\mathbb{P}$ completes, this asset must be in possession of either u or v. If (u, v) ends up in the possession of v, we will say that the arc (u, v) has been *triggered*. The outcome of v after executing $\mathbb{P}$ is $\langle \omega^{in} \,|\, \omega^{out} \rangle$, where ω^{in} and ω^{out} are the sets of incoming and outgoing arcs of v that are triggered in this execution. In particular, we write $\mathbb{P}(v)$ for the outcome of v in an execution of protocol $\mathbb{P}$ in which all parties follow $\mathbb{P}$. If some party (possibly v itself) deviates from $\mathbb{P}$, we assume that v's outcome is also finalized when $\mathbb{P}$ completes, although this outcome may be different from $\mathbb{P}(v)$.

A protocol may use appropriate cryptographic primitives. In particular, following [19], we assume the availability of *smart contracts*. A smart contract for an arc $a = (u, v)$ allows u to put asset a in an escrow secured with a suitable collection of hashed time-locks: each such time-lock is specified by a pair (h, τ), where $h = H(s)$ is a hashed value of a secret s and τ is a time-out value. In order to unlock this time-lock, v (and only v) must provide the value of s before time τ. If all time-locks of (u, v) are unlocked, v can claim a. This automatically triggers arc (u, v). If any time-lock times out, a is automatically returned to u.

Properties. For a swap protocol to be useful, it must guarantee that if all parties follow it then every party ends in an outcome at least as favorable as trading all their outgoing for all their incoming assets. Further, every conforming party should end up with an acceptable outcome, no matter whether other parties follow the protocol or not. Lastly, rational parties should have no incentive to deviate from the protocol. These properties are captured by the concepts of uniformity and Nash equilibrium, that we define next.

Uniformity. A swap protocol $\mathbb{P}$ is called *uniform* if it satisfies the following two conditions:

Liveness: If all parties follow $\mathbb{P}$, they all end in outcome $\textsc{Deal}$ *or better*, that is $\mathbb{P}(v) \succeq \textsc{Deal}_v$ for all $v \in V$.

Safety: If a party conforms to $\mathbb{P}$, then its outcome will be acceptable, independently of the behavior of other parties.

A less restrictive concept of uniformity may also be of interest: We say that a protocol $\mathbb{P}$ is *weakly uniform* if it satisfies the safety condition above, but the liveness condition is replaced by the following *weak liveness* requirement: if all parties follow $\mathbb{P}$, then at least one party ends in an outcome strictly better than $\textsc{NoDeal}$. The assumptions on preference posets imply directly that a protocol that is uniform is also weakly uniform.

Nash equilibria and atomicity. We extend the concept of outcomes to sets of parties, where an outcome of a set is just a vector of individual outcomes. On this set we can then define a preference relation in a standard way, via a coordinate-wise ordering of outcomes. Formally, for any set of parties $C \subseteq V$, an *outcome vector of* C is a $\bar{\omega} = (\omega_v)_{v \in C}$, where $\omega_v \in \Omega_v$ for all $v \in C$. Denote by $\bar{\Omega}_C$ the set of all outcome vectors of C. Given two outcome vectors $\bar{\omega}, \bar{\omega}' \in \bar{\Omega}_C$, we write $\bar{\omega} \preceq_C \bar{\omega}'$ if $\omega_v \preceq_v \omega'_v$ for all $v \in C$. If also $\bar{\omega} \neq \bar{\omega}'$ then we write $\bar{\omega} \prec_C \bar{\omega}'$. (In other words, $\bar{\omega} \prec_C \bar{\omega}'$ means that at least one party in C does strictly better in $\bar{\omega}'$ than in $\bar{\omega}$, and every party in C does at least as good. In this notation, if all parties follow a protocol $\mathbb{P}$, then the outcome vector $\mathbb{P}(C)$ of a protocol $\mathbb{P}$ for a set of parties C is $(\mathbb{P}(v))_{v \in C}$.

We will say that a protocol $\mathbb{P}$ is a *strong Nash equilibrium* if no coalition of participating parties can improve its vector outcome by deviating from $\mathbb{P}$; more precisely, for every set C of parties, if $\bar{\omega}$ denotes the outcome vector of C in some execution of $\mathbb{P}$ where all parties in $V \setminus C$ follow $\mathbb{P}$, then we cannot have $\bar{\omega} \succ_C \mathbb{P}(C)$. We will call $\mathbb{P}$ *atomic* if it is both uniform and a strong Nash equilibrium.

Example 1. Consider a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ whose digraph $\mathcal{D}$ is shown in Figure 1. The preference poset $\mathcal{P}_u$ of u is generated by two preference pairs $\textsc{Deal}_u \prec \langle v \,|\, v \rangle \prec \langle v \,|\, w \rangle$, the preference poset $\mathcal{P}_v$ of v is generated by two preference pairs $\textsc{Deal}_v \prec \langle u \,|\, u \rangle \prec \langle w \,|\, u \rangle$, and the preference poset $\mathcal{P}_w$ of w is generated by one preference pair $\textsc{Deal}_w \prec \langle u \,|\, v \rangle$.

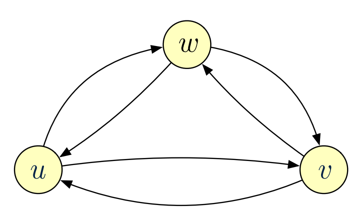

Figure 1. The digraph $\mathcal{D}$ in the example.

Consider also a swap protocol $\mathbb{P}$ for $\mathcal{S}$ such that if all parties follow $\mathbb{P}$ then all end up with outcome $\textsc{Deal}$. Then $\mathbb{P}$ is not a strong Nash equilibrium, because for $C = \{u, v\}$, the parties in C can ignore $\mathbb{P}$ altogether and simply swap their assets between themselves, improving their outcomes. Nevertheless, as we show later in Section 4, $\mathcal{S}$ does have an atomic protocol. Roughly, instead of using the whole digraph $\mathcal{D}$, in this protocol only assets represented by arcs (u, w), (w, v) and (v, u) will be swapped. Then the outcome

3. This definition can be relaxed to allow some outcomes incomparable to $\textsc{NoDeal}$ be acceptable. In this extended model, the set $\mathcal{A}_v$ of acceptable outcomes would be part of a swap system specification, and would have to satisfy three conditions: (i) $\{\omega : \omega \succeq \textsc{NoDeal}_v\} \subseteq \mathcal{A}_v$, (ii) $\{\omega : \omega \prec \textsc{NoDeal}_v\} \cap \mathcal{A}_v = \emptyset$, and (iii) $\omega \in \mathcal{A}_v \wedge \omega \preceq \omega' \Rightarrow \omega' \in \mathcal{A}_v$. Our results can be extended naturally to this model. We adopted the simpler definition to streamline the presentation.

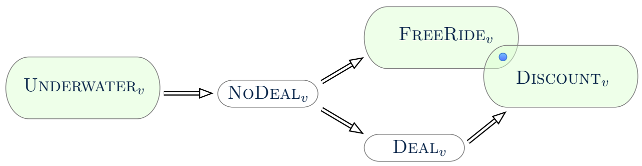

Figure 2. The structure of a preference poset of a party v in an h-swap system. The arrows symbolize the preference relation. The one outcome in $\text{Discount}_v \cap \text{FreeRide}_v$ is $\langle A_v^{in} \mid \varnothing \rangle$.

of each party will be better than DEAL, and u and v will have no incentive to deviate from this protocol.

3. Herlihy's Swap Model

In this section, we show that the concept of swap systems is a generalization of Herlihy's model [19]. To this end, we define a simple type of swap system under our model that we call h-swap systems, and we show that it captures the model in [19]. In particular we prove that in h-swap systems, our definition of atomicity is equivalent to the definition in [19].

h-Swap Systems. Given a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ and a party $v \in V$, define three sets of outcomes of v:

$$\begin{aligned} \text{Discount}_v &= \{\omega \mid \omega^{in} = A_v^{in} \wedge \omega^{out} \neq A_v^{out}\} \\ \text{FreeRide}_v &= \{\omega \mid \omega^{in} \neq \varnothing \wedge \omega^{out} = \varnothing\} \\ \text{Underwater}_v &= \{\omega \mid \omega^{in} \neq A_v^{in} \wedge \omega^{out} \neq \varnothing\} \end{aligned}$$

Since $A_v^{in} \neq \emptyset$ and $A_v^{out} \neq \emptyset$, all sets Discount_v, FreeRide_v and Underwater_v are well-defined, none of them contains NoDeal_v nor Deal_v, $\text{Underwater}_v \cap (\text{Discount}_v \cup \text{FreeRide}_v) = \emptyset$, $\text{Discount}_v \cap \text{FreeRide}_v = \{\langle A_v^{in} \mid \varnothing \rangle\}$, and

$$\Omega_v = \{\text{NoDeal}_v\} \cup \{\text{Deal}_v\} \cup \text{Discount}_v \cup \text{FreeRide}_v \cup \text{Underwater}_v.$$

The inclusive monotonicity property (p.2) implies that all outcomes in FreeRide_v are better than NoDeal_v, and all outcomes in Discount_v are better than Deal_v.

We will call $\mathcal{S}$ an *h-swap system* if it satisfies the following conditions for all $v \in V$:

(h.1) If $\omega \in \text{Underwater}_v$ then $\omega \prec_v \text{NoDeal}_v$,
(h.2) Party v has no other non-generic preferences besides these in (h.1).

In other words, in an h-swap system all preference posets are generated by relations $\omega \prec \text{NoDeal}$ for outcomes ω in UNDERWATER. Figure 2 illustrates the structure of a preference poset of an h-swap system[4]. Note that in an h-swap system, the set of acceptable outcomes of a node v is $\mathcal{A}_v = \Omega_v \setminus \text{Underwater}_v = \{\text{NoDeal}_v\} \cup \{\text{Deal}_v\} \cup \text{Discount}_v \cup \text{FreeRide}_v$. The preferences of an h-swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ are uniquely determined by its digraph $\mathcal{D}$, so it is not even necessary to specify $\mathcal{P}$.

4. This figure differs slightly from Figure 3 in [19], which mistakenly showed the sets Discount_v and FreeRide_v as disjoint.

The preference poset structure of h-swap systems, as defined above, captures the concept of a party's preferences assumed in the model from [19], except for the addition of preferences determined by inclusive monotonicity.

Comment. The model in [19] was not formulated in terms of posets, raising a question of how to formally capture a relation for pairs of outcomes between which preferences were not specified in [19]. In our model, such outcomes are considered incomparable in the poset (unless they are related by the inclusive monotonicity). One may try to consider another option: to allow arbitrary relations between such pairs, providing that the poset axioms are satisfied and the condition (h.1) holds. However, with this approach there is no meaningful way to extend such individual preferences to collective preferences of sets of parties (see the discussion later in this section).

h-Uniformity. To distinguish between our and Herlihy's definition of uniformity, we will refer to his concept as h-uniformity. A swap protocol $\mathbb{P}$ is called *h-uniform* if it satisfies the safety property and the following *h-liveness* condition: If all parties follow $\mathbb{P}$, they all end in outcome DEAL. This condition seems stricter than our definition of uniformity, but we show that in h-swap systems these two definitions are in fact equivalent. In fact, they are also equivalent to weak uniformity, as defined earlier in Section 2.

Lemma 1. *Let $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ be an h-swap system in which some subset of arcs in $\mathcal{D}$ are triggered, and let Q be a path in $\mathcal{D}$ whose all internal nodes are in acceptable outcomes. Then, along Q, all triggered arcs of Q are before all non-triggered arcs of Q.*

Proof. If all arcs on Q are triggered, except possibly for the last one, we are done. Otherwise, let (x, y) be the first non-triggered arc on Q and let z be the successor of y. Since y's outcome is acceptable and (x, y) is not triggered, this outcome must be either NoDeal_y or in FreeRide_y. Therefore (y, z) is also not triggered. Repeating this argument, we obtain that all arcs on Q after (x, y) are not triggered. □

Theorem 1. *Let $\mathbb{P}$ be a swap protocol for an h-swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$, where $\mathcal{D}$ is strongly connected. Then the following three conditions are equivalent: (i) $\mathbb{P}$ is uniform, (ii) $\mathbb{P}$ is weakly uniform, (iii) $\mathbb{P}$ is h-uniform.*

Proof. Trivially, h-uniformity implies uniformity, which in turn implies weak uniformity. Thus it is sufficient to show that weak uniformity implies h-uniformity.

So assume that $\mathbb{P}$ is weakly uniform. As the safety condition is the same, it is sufficient to show that $\mathbb{P}$ satisfies the h-liveness property. Assume that all parties follow $\mathbb{P}$. Then, from the assumptions about safety and weak liveness, all parties will end up in acceptable outcomes, with at least one party ending in an outcome strictly better than NODEAL.

Suppose, towards contradiction, that there is a party with outcome other than DEAL. This gives us that some arc (x, y) is not triggered. Further, since some party has an outcome other than NODEAL, there must be a triggered arc (x', y').

By strong connectivity, there is a path P from x to y' whose first arc is (x, y) and the last arc is (x', y'). Then the existence of this path contradicts Lemma 1. $\square$

h-Atomicity. The approach in [19] differs from ours in the way it formalizes the gain of a coalition (subset) of parties when they deviate from the protocol. Roughly, the definition in [19] captures a collective gain, while our definition views it as a vector of individual outcomes. In spite of this apparent difference, we show that in h-swap systems our concept of atomicity is in fact equivalent to the one in [19].

In the discussion below, let $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ be a fixed h-swap system. Following [19], we will define the h-outcome of a coalition $\mathcal{C}$ of parties by, in essence, contracting $\mathcal{C}$ into a single vertex. (The term "h-outcome" is ours, to better distinguish this concept from our concept of outcome vectors.) More formally, define $\mathcal{C}$'s incoming and outgoing arcs in a natural way: $A^{in}_{\mathcal{C}} = \bigcup_{v\in\mathcal{C}} A^{in}_v \setminus \bigcup_{v\in\mathcal{C}} A^{out}_v$ and similarly, $A^{out}_{\mathcal{C}} = \bigcup_{v\in\mathcal{C}} A^{out}_v \setminus \bigcup_{v\in\mathcal{C}} A^{in}_v$. The *h-outcomes* for $\mathcal{C}$ are pairs $\hat{\omega} = \langle \hat{\omega}^{in} \,|\, \hat{\omega}^{out} \rangle$ where $\hat{\omega}^{in} \subseteq A^{in}_{\mathcal{C}}$ and $\hat{\omega}^{out} \subseteq A^{out}_{\mathcal{C}}$. $\Omega_{\mathcal{C}}$ is the set of all h-outcomes of C. The preference poset and acceptable set of $\mathcal{C}$ are defined analogously to that of a single party in an h-swap system. That is, we define NoDeal_C, Deal_C, Discount_C, FreeRide_C, and Underwater_C in the natural way, and we assume the analogues of conditions (p.1) and (p.2) for swap systems (in Section 2) and conditions (h.1) and (h.2) for h-swap systems. The set of acceptable h-outcomes $\mathcal{A}_{\mathcal{C}}$ consists of all h-outcomes of $\mathcal{C}$ that are not in Underwater_C. (Note that if $\mathcal{C}$ consists of a single party then its h-outcome is identical to its outcome.)

Define a protocol $\mathbb{P}$ to be a *strong Nash h-equilibrium* if it satisfies the following condition for every set $\mathcal{C}$ of parties: providing that the parties outside $\mathcal{C}$ follow $\mathbb{P}$, the parties in $\mathcal{C}$ cannot end up in an h-outcome better than their outcome resulting from following $\mathbb{P}$. $\mathbb{P}$ is called *h-atomic* if it is h-uniform and a strong Nash h-equilibrium.

Culminating the earlier discussion, the following theorem establishes that our model indeed captures the model introduced in [19].

Theorem 2. *Let $\mathbb{P}$ be a protocol for an h-swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$. $\mathbb{P}$ is atomic if and only if it is h-atomic.*

Proof. ($\Rightarrow$) Suppose that $\mathbb{P}$ is atomic. Theorem 1 implies that $\mathbb{P}$ is h-uniform. Thus, from the definition of h-uniformity, if all parties follow $\mathbb{P}$ then each party's outcome will be Deal.

It remains to show that $\mathbb{P}$ is a strong Nash h-equilibrium. Let $C \subseteq V$, and consider an execution of $\mathbb{P}$ in which all parties outside C follow $\mathbb{P}$. Since $\mathbb{P}$ is a strong Nash equilibrium, the outcome vector of C is not better than $(\text{Deal}_v)_{v\in C}$. Denote by $\hat{\omega}$ the h-outcome of C. We need to show that $\hat{\omega}$ is not better than Deal_C.

Towards contradiction, suppose that $\hat{\omega} \succ \text{Deal}_C$. The definition of preference posets for h-outcomes gives us that $\hat{\omega} \in \text{Discount}_C$. Now consider another execution of $\mathbb{P}$ where the parties in C behave just like before, but they also trigger all arcs connecting two members of C. This will not affect the execution of $\mathbb{P}$ for parties outside C. Then the outcome vector $\bar{\omega}$ of C consists of all arcs between C and $V \setminus C$ (in both directions) that are triggered in $\hat{\omega}$, as well as all arcs with both endpoints inside C. Since $\hat{\omega} \in \text{Discount}_C$, each $v \in C$ has all its incoming arcs in $\bar{\omega}$, and there is at least one $u \in C$ that has one arc to $V \setminus C$ that is not in $\bar{\omega}$. So the outcome of each $v \in C$ is either Deal_v or Discount_v, and this u's outcome is Discount_u. But then $\bar{\omega}$ is better than $(\text{Deal}_v)_{v\in C}$, contradicting the assumption that $\mathbb{P}$ is a strong Nash equilibrium.

($\Leftarrow$) Now suppose that $\mathbb{P}$ is h-atomic; that is, $\mathbb{P}$ is h-uniform and is a strong Nash h-equilibrium. From Theorem 1 we obtain that $\mathbb{P}$ is uniform.

It remains to prove that $\mathbb{P}$ is a strong Nash equilibrium. Let $C \subseteq V$, and consider some execution of $\mathbb{P}$ in which all parties outside C follow $\mathbb{P}$. Since $\mathbb{P}$ is a strong Nash h-equilibrium, the h-outcome of C is not better than Deal_C. We need to show that C's outcome vector is not better than $(\text{Deal}_v)_{v\in C}$.

We again argue by contradiction. Suppose that C's outcome vector is $\bar{\omega} \succ (\text{Deal}_v)_{v\in C}$. Then each $v \in C$ has outcome in $\{\text{Deal}_v\} \cup \text{Discount}_v$ and there is some $u \in C$ with outcome in Discount_v. This implies that all parties in C have their incoming arcs in $\bar{\omega}$. Further, some outgoing arc of u is not in $\bar{\omega}$, and this arc must go to $V \setminus C$. We consider the h-outcome of C in the same run of $\mathbb{P}$, without changing the behavior of any members of C. (In the h-outcome of C the status of arcs internal to C is not relevant.) Denote this h-outcome by $\hat{\omega}$. Then $\hat{\omega}$ will include the same arcs between C and $V \setminus C$ (in both directions) as in $\bar{\omega}$. The properties of $\bar{\omega}$ established earlier imply that $\hat{\omega} \in \text{Discount}_C$, and thus $\hat{\omega} \succ \text{Deal}_C$, which contradicts our earlier assumption that $\mathbb{P}$ is a strong Nash h-equilibrium. $\square$

4. A Characterization of Swap Systems with Atomic Protocols

In this section, we characterize swap systems that have atomic protocols. Interestingly enough, we show that if a swap system admits an atomic protocol, then it also admits an atomic protocol that is essentially equivalent to running Herlihy's protocol on a suitable subgraph.

We denote Herlihy's protocol [19] as $\mathbb{H}$. Herlihy proved that $\mathbb{H}$ is h-atomic for h-swap systems (in our terminology).

Uniformity and Nash equilibrium of Herlihy's protocol. Let $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ be any swap system with strongly connected digraph $\mathcal{D}$. If all parties follow $\mathbb{H}$ then they all will end up in outcome Deal. If v follows $\mathbb{H}$ then either it does not trigger any outgoing arcs, and thus its outcome is in $\{\text{NoDeal}_v\} \cup \text{FreeRide}_v$, or it triggers some, but then also all its incoming arcs are triggered, so its outcome is in $\{\text{Deal}_v\} \cup \text{Discount}_v$. In each case, regardless of the behavior of other parties, this outcome is at least as good as NoDeal_v, and thus acceptable. This means that $\mathbb{H}$ is uniform.

We now claim that $\mathbb{H}$ is a *Nash equilibrium* in $\mathcal{S}$, in the sense that no single party can improve its outcome by deviating from $\mathbb{H}$, if all other parties follow $\mathbb{H}$. If all parties follow the protocol, all outcomes are $\textsc{Deal}$. If any party v has outcome $\omega \succ \textsc{Deal}_v$, then in ω there needs to be an incoming arc (u, v) but some outgoing arc (v, w) must be missing. (This holds for any preference poset, by properties (p.1) and (p.2).) In Herlihy's protocol a vertex triggers an outgoing arc only if all its incoming arcs are triggered. So if all parties other than v follow the protocol, then we get a contradiction by considering a path from w to u (following an argument similar to the proof of Theorem 1).

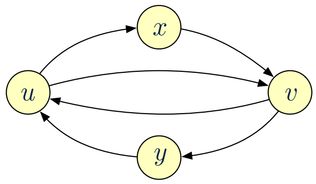

Figure 3. The example of a swap system in which $\mathbb{H}$ is not a strong Nash equilibrium.

Note that this argument does not work for larger coalitions. The reason is that a larger coalition can improve its outcome even without any arcs from outside being triggered. As an example, consider a swap system whose digraph is shown in Figure 3. The only non-generic preferences are: $\textsc{Deal}_v \prec \langle u \,|\, u\rangle$ for v, and $\textsc{Deal}_u \prec \langle v \,|\, v\rangle$ for u. Then u and v can cooperatively deviate from $\mathbb{H}$ by triggering only arcs (u, v) and (v, u), each obtaining a better outcome than if they followed $\mathbb{H}$. This shows that $\mathbb{H}$ is not a strong Nash equilibrium in arbitrary swap systems.

Characterization. Let $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ be a swap system for a set of parties V, and let $\mathcal{G}$ and $\mathcal{H}$ be two subgraphs of $\mathcal{D}$. $\mathcal{G}$ will be called *piece-wise strongly connected* if every connected component of $\mathcal{G}$ is strongly connected. $\mathcal{G}$ is called *spanning* if its vertex set is V. If $\mathcal{G}$ is spanning, we say that $\mathcal{H}$ *dominates* $\mathcal{G}$ if $\textsc{Deal}^{\mathcal{H}}_v \succeq \textsc{Deal}^{\mathcal{G}}_v$ for all vertices v in $\mathcal{H}$. In other words, if only the arcs in $\mathcal{H}$ are triggered, then all parties in $\mathcal{H}$ end in outcomes at least as good as if all their arcs in $\mathcal{G}$ were triggered. Also, $\mathcal{H}$ *strictly dominates* $\mathcal{G}$ if, in addition, there exists a party u of $\mathcal{H}$ such that $\textsc{Deal}^{\mathcal{H}}_u \succ \textsc{Deal}^{\mathcal{G}}_u$. That is, every party in $\mathcal{H}$ ends in an outcome at least as good and at least one party strictly improves their outcome when triggering the arcs of $\mathcal{H}$ instead of $\mathcal{G}$.

For example, consider the swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ in Example 1. The subgraph $\mathcal{G}_1 = \mathcal{D}$ is spanning, and is strictly dominated by the subgraph $\mathcal{H}$ consisting of vertices u, v and arcs (u, v) and (v, u). On the other hand, the subgraph $\mathcal{G}_2$ that has arcs (u, w), (w, v) and (v, u) is spanning, and there is no subgraph of $\mathcal{D}$ that strictly dominates it.

Theorem 3. *A swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ has an atomic swap protocol if and only if there exists a spanning subgraph $\mathcal{G}$ of $\mathcal{D}$ with the following properties: (c.1) $\mathcal{G}$ is piece-wise strongly connected and has no isolated vertices, (c.2) $\mathcal{G}$ dominates $\mathcal{D}$, and (c.3) no subgraph $\mathcal{H}$ of $\mathcal{D}$ strictly dominates $\mathcal{G}$.*

Proof. ($\Rightarrow$) Let $\mathbb{P}$ be an atomic swap protocol for $\mathcal{S}$. Define $\mathcal{G}$ to be the subgraph whose vertex set is V and whose arcs are the arcs triggered in an execution of $\mathbb{P}$ where all parties follow the protocol. By definition of $\mathbb{P}$'s atomicity, $\mathcal{G}$ is spanning.

We first show property (c.1). First, $\mathcal{G}$ cannot have any isolated vertices, since any isolated vertex v of $\mathcal{G}$ would have outcome $\textsc{NoDeal}^{\mathcal{D}}_v$ when all parties follow $\mathbb{P}$. This would contradict the uniformity (the liveness condition) of $\mathbb{P}$. Second, if $\mathcal{G}$ had a connected component B that is not strongly connected, then B would contain a strongly connected component C of $\mathcal{G}$ that has no arcs of $\mathcal{G}$ coming from $V \setminus C$ but has at least one arc of $\mathcal{G}$ going to $V \setminus C$. We could then consider another run of $\mathbb{P}$ in which the parties in C ignore $\mathbb{P}$ entirely and simply trigger the arcs of $\mathcal{G}$ that are within C. By the inclusive monotonicity property (p.2) of swap systems, this would strictly improve the outcome vector of C, contradicting $\mathbb{P}$ being a strong Nash equilibrium. We can thus conclude that such B cannot exist, completing the proof that $\mathcal{G}$ is piece-wise strongly connected.

Next, we consider property (c.2). By the uniformity (liveness) of $\mathbb{P}$, every party v must end in outcome $\textsc{Deal}^{\mathcal{D}}_v$ or better when all parties follow $\mathbb{P}$. The arcs that are triggered at the conclusion of $\mathbb{P}$ are exactly the arcs in $\mathcal{G}$. Therefore $\textsc{Deal}^{\mathcal{G}}_v \succeq \textsc{Deal}^{\mathcal{D}}_v$, for all parties v.

Finally, we show property (c.3). Suppose there is a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$, towards contradiction. Let C be the set of vertices of $\mathcal{H}$. Modify the behavior of the parties in C to ignore $\mathbb{P}$ and instead trigger exactly the arcs of $\mathcal{H}$, giving C the outcome vector $(\textsc{Deal}^{\mathcal{H}}_v)_{v \in C}$. Then, $\mathbb{P}(C) = (\textsc{Deal}^{\mathcal{G}}_v)_{v \in C} \prec_C (\textsc{Deal}^{\mathcal{H}}_v)_{v \in C}$, as $\mathcal{H}$ strictly dominates $\mathcal{G}$. This contradicts the assumption that $\mathbb{P}$ is a strong Nash equilibrium, proving that $\mathcal{H}$ does not exists.

($\Leftarrow$) Suppose that $\mathcal{G}$ is a spanning subgraph that satisfies properties (c.1), (c.2) and (c.3). We show that then there is an atomic protocol for $\mathcal{S}$.

Let $\mathcal{S}^{\mathcal{G}}$ be the h-swap system with digraph $\mathcal{G}$. Our protocol, denoted $\mathbb{H}_{\mathcal{G}}$, simply executes Herlihy's protocol $\mathbb{H}$ on $\mathcal{S}^{\mathcal{G}}$. For simplicity, assume that $\mathcal{G}$ is strongly connected; otherwise we can apply our reasoning below to each strongly connected component of $\mathcal{G}$ separately. By the h-liveness condition of $\mathbb{H}_{\mathcal{G}}$, if all parties follow $\mathbb{H}_{\mathcal{G}}$ then each will end up in outcome $\textsc{Deal}^{\mathcal{G}}$. Also, any party v that follows $\mathbb{H}_{\mathcal{G}}$ will not have any of its arcs outside $\mathcal{G}$ triggered and, by the safety property of $\mathbb{H}_{\mathcal{G}}$, will end up in an outcome that is acceptable in $\mathcal{S}^{\mathcal{G}}$, that is in $\{\textsc{NoDeal}^{\mathcal{G}}_v\} \cup \textsc{FreeRide}^{\mathcal{G}}_v \cup \{\textsc{Deal}^{\mathcal{G}}_v\} \cup \textsc{Discount}^{\mathcal{G}}_v$.

When comparing outcomes in the argument that follows, we will use notation "$\prec$" for the preference relation in the original swap system $\mathcal{S}$ (that is, *not* in the auxiliary system $\mathcal{S}^{\mathcal{G}}$). Similarly, unless stated otherwise, the term "acceptable" also refers to the acceptability of an outcome in $\mathcal{S}$.

We first show that $\mathbb{H}_{\mathcal{G}}$ is uniform. Suppose that every party follows $\mathbb{H}_{\mathcal{G}}$. Then, by the h-uniformity of $\mathbb{H}_{\mathcal{G}}$, the outcome of each party v will be $\textsc{Deal}^{\mathcal{G}}_v$. Using the assumptions that $\mathcal{G}$ is spanning and that it dominates $\mathcal{D}$, we obtain that $\textsc{Deal}^{\mathcal{G}}_v \succeq \textsc{Deal}^{\mathcal{D}}_v$ for all parties v, so $\mathbb{H}_{\mathcal{G}}$ indeed satisfies the liveness condition.

Next, we deal with the safety condition. Using the properties of $\mathbb{H}_\mathcal{G}$ established above, if a party v conforms to $\mathbb{H}_\mathcal{G}$ then we have two cases. Either the outcome ω of v satisfies $\omega \in \{\text{NoDeal}_v^\mathcal{G}\} \cup \text{FreeRide}_v^\mathcal{G}$, in which case $\omega \in \{\text{NoDeal}_v^\mathcal{D}\} \cup \text{FreeRide}_v^\mathcal{D}$ as well (because no edges of v outside $\mathcal{G}$ are triggered), so $\omega \succeq \text{NoDeal}_v^\mathcal{D}$, that is ω is acceptable. Or $\omega \in \{\text{Deal}_v^\mathcal{G}\} \cup \text{Discount}_v^\mathcal{G}$, in which case, using the monotonicity property (p.2) for $\mathcal{S}$ and assumption (c.2), we obtain $\omega \succeq \text{Deal}_v^\mathcal{G} \succeq \text{Deal}_v^\mathcal{D} \succ \text{NoDeal}_v^\mathcal{D}$; that is ω is acceptable in this case as well. We conclude that $\mathbb{H}_\mathcal{G}$ satisfies the safety property, completing the proof that $\mathbb{H}_\mathcal{G}$ is uniform.

It remains to show that $\mathbb{H}_\mathcal{G}$ is a strong Nash equilibrium for $\mathcal{S}$. Assume that it is not, towards contradiction. Then there exists a coalition $C \subseteq V$ that, by deviating from $\mathbb{H}_\mathcal{G}$, can end in an outcome vector $\bar{\omega} \succ (\text{Deal}_v^\mathcal{G})_{v\in C}$, even though all parties outside C follow $\mathbb{H}_\mathcal{G}$. We can assume C is maximal, in the sense that each party outside of C ends in an outcome that is not $\text{Deal}^\mathcal{G}$ nor in $\text{Discount}^\mathcal{G}$. Otherwise, we can add those parties to C and the relation $\bar{\omega} \succ (\text{Deal}_v^\mathcal{G})_{v\in C}$ will be preserved.

We first show that no arc $(u, v) \in A$ entering C from outside (that is $u \in V \setminus C$ and $v \in C$) is triggered. Assume such an arc is triggered, towards contradiction. Firstly, (u, v) must be in $\mathcal{G}$, otherwise u would not be following $\mathbb{H}_\mathcal{G}$ by creating/triggering this arc. By the h-safety property of $\mathbb{H}$ in $\mathcal{S}^\mathcal{G}$, $\mathbb{H}_\mathcal{G}$ guarantees that u must end up in an outcome acceptable in $\mathcal{S}^\mathcal{G}$. This means that u's outcome is in $\{\text{Deal}_u^\mathcal{G}\} \cup \text{Discount}_u^\mathcal{G}$, contradicting the assumption that C is maximal. So, indeed, (u, v) cannot be triggered.

Further, without loss of generality we can assume that no arc from C to $V \setminus C$ is triggered. This is because, as we just showed, for each $v \in C$, v only receives arcs from other parties in C. Then, no member of C can have its outcome worsened if v changes its behavior and does not trigger any arc to $V \setminus C$.

Thus all arcs that appear in $\bar{\omega}$ are between members of C. Let $\mathcal{H}$ be the subgraph with vertex set C and the arcs that are in $\bar{\omega}$, that is $\bar{\omega} = (\text{Deal}_v^\mathcal{H})_{v\in C}$. Since $\bar{\omega} \succ (\text{Deal}_v^\mathcal{G})_{v\in C}$, then $\text{Deal}_v^\mathcal{H} \succeq \text{Deal}_v^\mathcal{G}$ for all $v \in C$ and $\text{Deal}_w^\mathcal{H} \succ \text{Deal}_w^\mathcal{G}$ for some $w \in C$. This means that $\mathcal{H}$ strictly dominates $\mathcal{G}$, contradicting (c.3). We conclude that no such C exists, and thus $\mathbb{H}_\mathcal{G}$ is a strong Nash equilibrium protocol. □

Comment: As some readers may have noticed, the proof of the ($\Rightarrow$) implication in Theorem 3 does not use the safety property of protocol $\mathbb{P}$. What this shows, in essence, is that in our setting of swap systems, a swap protocol that has the liveness and strong Nash equilibrium properties can be modified to also satisfy the safety property.

Example 2. To illustrate Theorem 3, consider again the swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ in Example 1. Let $\mathcal{G}_1 = \mathcal{D}$. Then $\mathcal{G}_1$ is spanning, satisfies conditions (c.1) and (c.2), but it does not satisfy condition (c.3) because it is strictly dominated by subgraph $\mathcal{H}$ consisting of vertices u, v and arcs (u, v) and (v, u). We can take instead subgraph $\mathcal{G}_2$ that has arcs (u, w), (w, v) and (v, u). Then $\mathcal{G}_2$ is spanning, satisfies (c.1) and (c.2), and there is no subgraph of $\mathcal{D}$ that strictly dominates $\mathcal{G}_2$, so (c.3) holds as well. Therefore $\mathcal{S}$ has an atomic swap protocol.

Example 3. We now give a larger example. Consider the swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ with digraph $\mathcal{D}$ in Figure 4 and with preference posets defined as follows. For $i = 1, 2$:

- The preference poset of u_i is generated by $\text{Deal}_{u_i} \prec \langle v_i \,|\, v_i \rangle$ and $\text{Deal}_{u_i} \prec \langle u_{3-i} \,|\, u_{3-i} \rangle$.
- The preference poset of v_i is generated by $\text{Deal}_{v_i} \prec \langle u_i \,|\, u_i \rangle$ and $\text{Deal}_{v_i} \prec \langle t_{3-i} \,|\, t_i \rangle \prec \langle v_{3-i} \,|\, v_{3-i} \rangle$.
- The preference poset of t_i is generated by $\text{Deal}_{t_i} \prec \langle v_i \,|\, v_{3-i} \rangle$ and $\text{Deal}_{t_i} \prec \langle t_{3-i} \,|\, t_{3-i} \rangle$.

We consider three candidates for the spanning sugraph $\mathcal{G}$. One candidate is $\mathcal{G}_1 = \mathcal{D}$. It's obviously spanning and it satisfies conditions (c.1) and (c.2) from Theorem 3. However, it is strictly dominated by several subgraphs including the following two: subgraph $\mathcal{H}_1$ consisting of v_1 and v_2 with arcs (v_1, v_2) and (v_2, v_1), and subgraph $\mathcal{H}_2$ consisting of t_1 and t_2 with arcs (t_1, t_2) and (t_2, t_1).

Another candidate $\mathcal{G}_2$ consists of arcs (u_i, u_{3-i}), (v_i, t_i) and (t_i, v_{3-i}). for $i = 1, 2$. $\mathcal{G}_2$ is spanning and it satisfies condition (c.1). By inspecting the preferences of each vertex, it also satisfies (c.2). But it is strictly dominated by $\mathcal{H}_1$.

The third candidate $\mathcal{G}_3$ consists of arcs (u_i, v_i), (v_i, u_i), and (t_i, t_{3-i}), for $i = 1, 2$. It is also spanning and satisfies conditions (c.1) and (c.2). Also, the outcome of each vertex in $\mathcal{G}_3$ is maximal in its preference poset, so there is no subgraph of $\mathcal{D}$ that strictly dominates $\mathcal{G}_3$. Thus, by Theorem 3, $\mathcal{S}$ has an atomic swap protocol. This protocol is obtained by running $\mathbb{H}$ in $\mathcal{G}_3$.

5. NP-Hardness

Now that we have characterized the swap systems that permit an atomic protocol, a natural next question is the complexity of the corresponding decision problem: given a swap system, does it permits an atomic protocol? In the next two sections, we consider this. In this section, we define the corresponding decision problem and show it is in NP-hard. In the following section, we tighten this classification to Σ_2^P-complete. Although showing the problem is Σ_2^P-complete would imply it is NP-hard, we first present the NP-hardness proof since it is more digestible, and then present the more involved Σ_2^P-completeness proof.

Let SwapAtomic be the following decision problem: The input is a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$, where $\mathcal{D}$ is a (weakly) connected digraph with no vertices of in-degree or out-degree 0. The objective is to decide whether $\mathcal{S}$ has an atomic swap protocol.

Theorem 4. SwapAtomic *is* NP*-hard, even for swap systems $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ in which digraph $\mathcal{D}$ is strongly connected.*

Proof. The proof is by showing a polynomial-time reduction from CNF. Recall that in CNF we are given a boolean expression α in conjunctive normal form, and the objective is to determine whether there is a truth assignment that

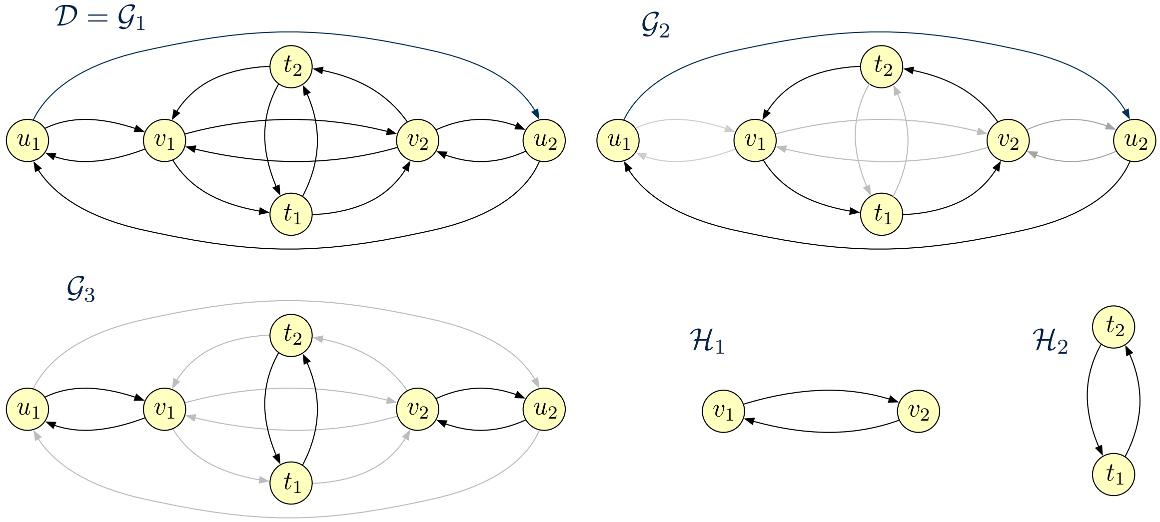

Figure 4. Digraph $\mathcal{D}$ in the example illustrating Theorem 3.

satisfies α. In our reduction we convert α into a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ such that α is satisfiable if and only if $\mathcal{S}$ has an atomic swap protocol.

Let $x_1, x_2, ..., x_n$ be the variables in α. The negation of x_i is denoted $\bar{x}_i$. We will use notation $\tilde{x}_i$ for an unspecified literal of variable x_i, that is $\tilde{x}_i \in \{x_i, \bar{x}_i\}$. Let $\alpha = c_1 \vee c_2 \vee ... \vee c_m$, where each c_j is a clause. Without loss of generality we assume that in each clause all literals are different.

We first describe the reduction. Digraph $\mathcal{D}$ consists of n gadgets corresponding to variables, m gadgets corresponding to clauses, and one more vertex called the *core vertex*. The x_i-*gadget* has vertices x_i and $\bar{x}_i$, that represent the literals of variable x_i, plus two additional vertices s_i and t_i. Its internal arcs are (s_i, x_i), $(s_i, \bar{x}_i)$, (s_i, t_i), and (t_i, s_i). The c_j-*gadget* has two vertices c_j and a_j, with internal arcs (c_j, a_j), (a_j, c_j). (See Figure 5.) We then add the following arcs: For each clause c_j and each literal $\tilde{x}_i$ in c_j, we add arc $(\tilde{x}_i, c_j)$. The core vertex, denoted b, is connected by arcs to and from each vertex in the above gadgets.

Next, we describe the preference posets $\mathcal{P}_v$, for each vertex v in $\mathcal{D}$. (See Figure 6.) As explained in Section 2, an outcome $\langle \omega^{in} \,|\, \omega^{out} \rangle$ of a vertex v is specified by lists ω^{in} and ω^{out} of its in-neighbors and out-neighbors, respectively, and any preference poset can be uniquely defined by an appropriate set of generators.

The center vertex b's preference poset is generated by all relations $\omega \prec \textsc{NoDeal}_b$, for $\omega \in \textsc{Underwater}_b$. (This is the same poset as in h-swap systems.) For each vertex s_i, its preference poset is generated by relations $\langle b, t_i \,|\, t_i, \bar{x}_i \rangle \prec \langle t_i \,|\, t_i \rangle$, $\langle t_i \,|\, t_i \rangle \prec \langle b \,|\, b, x_i \rangle$, and $\langle t_i \,|\, t_i \rangle \prec \langle b \,|\, b, \bar{x}_i \rangle$. For each vertex t_i, its generators are $\textsc{Deal}_{t_i} \prec \langle b \,|\, b \rangle$ and $\langle b, s_i \,|\, b \rangle \prec \langle s_i \,|\, s_i \rangle$. Each vertex $\tilde{x}_i \in \{x_i, \bar{x}_i\}$ has one generator $\textsc{Deal}_{\tilde{x}_i} \prec \langle b \,|\, b \rangle$. Each vertex c_j has generators $\textsc{Deal}_{c_j} \prec \langle b, \tilde{x}_i \,|\, b \rangle$, for each literal $\tilde{x}_i$ in c_j. The only generator of each vertex a_j is $\textsc{Deal}_{a_j} \prec \langle b \,|\, b \rangle$.

With this, the description of $\mathcal{S}$ is complete. The construction of $\mathcal{S}$ clearly takes time that is polynomial in the size of α.

Applying Theorem 3, it remains to show that α is satisfiable if and only if $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ with the following properties: (c.1) $\mathcal{G}$ is piece-wise strongly connected and has no isolated vertices, (c.2) $\mathcal{G}$ dominates $\mathcal{D}$, and (c.3) no subgraph $\mathcal{H}$ of $\mathcal{D}$ strictly dominates $\mathcal{G}$.

($\Rightarrow$) Suppose that α is satisfiable, and consider some satisfying assignment for α. Using this assignment, we construct a spanning subgraph $\mathcal{G}$ of $\mathcal{D}$ that satisfies the three conditions (c.1)-(c.3) above.

Graph $\mathcal{G}$ will contain all vertices from the above construction and all arcs that connect b to all other vertices, in both directions. This makes $\mathcal{G}$ spanning and strongly connected, so (c.1) holds. Other arcs of $\mathcal{G}$ are defined as follows. For each true literal $\tilde{x}_i$, add to $\mathcal{G}$ arc $(s_i, \tilde{x}_i)$. For each clause c_j and each true literal $\tilde{x}_i$ in c_j, add to $\mathcal{G}$ arc $(\tilde{x}_i, c_j)$.

Condition (c.2) can be verified through routine inspection, by observing that $\textsc{Deal}^{\mathcal{D}}_v \preceq \textsc{Deal}^{\mathcal{G}}_v$ holds for each vertex v, directly from the above specification of the preference posets. For example, for a vertex s_i, if $\tilde{x}_i$ is the true literal of x_i then we have $\textsc{Deal}^{\mathcal{G}}_{s_i} = \langle b \,|\, b, \tilde{x}_i \rangle \succeq \textsc{Deal}^{\mathcal{D}}_{s_i}$. For any $\tilde{x}_i$, let $C(\tilde{x}_i)$ be the set of clauses that contain $\tilde{x}_i$. If $\tilde{x}_i$ is true then $\textsc{Deal}^{\mathcal{G}}_{\tilde{x}_i} = \langle b, s_i \,|\, b, C(\tilde{x}_i) \rangle = \textsc{Deal}^{\mathcal{D}}_{\tilde{x}_i}$, and if $\tilde{x}_i$ is false then $\textsc{Deal}^{\mathcal{G}}_{\tilde{x}_i} = \langle b \,|\, b \rangle \succeq \textsc{Deal}^{\mathcal{D}}_{\tilde{x}_i}$. For each clause c_j, denote by $T(c_j)$ the set of true literals in c_j. Since we use a satisfying assignment, each $T(c_j)$ is non-empty. For a clause c_j, for any true literal $\tilde{x}_i \in T(c_j)$, applying the inclusive monotonicity property (p.2) we have $\textsc{Deal}^{\mathcal{G}}_{c_j} = \langle b, T(c_j) \,|\, b \rangle \succeq \langle b, \tilde{x}_i \,|\, b \rangle \succeq \textsc{Deal}^{\mathcal{D}}_{c_j}$.

It remains to verify condition (c.3). Let $\mathcal{H}$ be a subgraph of $\mathcal{D}$, and suppose that $\mathcal{H}$ dominates $\mathcal{G}$, that is $\textsc{Deal}^{\mathcal{H}}_v \succeq \textsc{Deal}^{\mathcal{G}}_v$ for all vertices v in $\mathcal{H}$.

We claim first that $\mathcal{H}$ must contain b. Indeed, otherwise for any vertex v of $\mathcal{H}$ we would have $\textsc{Deal}^{\mathcal{H}}_v = \omega = \langle \omega^{in} \,|\, \omega^{out} \rangle$ with $b \notin \omega^{in}$ and $\omega \succeq \textsc{Deal}^{\mathcal{G}}_v$. The only vertices that have such outcomes ω are t_i's. For any t_i, the only

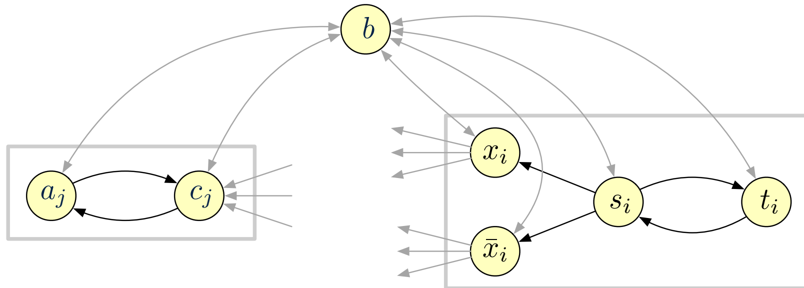

Figure 5. The variable and clause gadgets in the proof of Theorem 4. The arcs to and from the core vertex b are shown as bi-directional arcs.

$\mathcal{P}_{s_i}$ $\quad \text{DEAL}_{s_i} \prec \langle\, b, t_i \,|\, x_i, \bar{x}_i \,\rangle \prec \langle\, t_i \,|\, t_i \,\rangle$, with $\langle\, t_i \,|\, t_i \,\rangle \prec \langle\, b \,|\, b, x_i \,\rangle$ and $\langle\, t_i \,|\, t_i \,\rangle \prec \langle\, b \,|\, b, \bar{x}_i \,\rangle$

$\mathcal{P}_{a_j}$ $\quad \text{DEAL}_{a_j} \prec \langle\, b \,|\, b \,\rangle$

$\mathcal{P}_{\tilde{x}_i}$ $\quad \text{DEAL}_{\tilde{x}_i} \prec \langle\, b \,|\, b \,\rangle$

$\mathcal{P}_{t_i}$ $\quad \text{DEAL}_{t_i} \prec \langle\, b \,|\, b \,\rangle \prec \langle\, b, s_i \,|\, b \,\rangle \prec \langle\, s_i \,|\, s_i \,\rangle$

$\mathcal{P}_{c_j}$ $\quad \text{DEAL}_{c_j} \prec \langle\, b, \tilde{x}_i \,|\, b \,\rangle \quad \forall \tilde{x}_i \in c_j$

Figure 6. The specifications of preference posets in the reduction. We include in the figure some generic preferences for s_i and t_i, to illustrate the relationship of some outcomes to the corresponding DEAL outcome.

outcome ω that has these properties is $\langle s_i \,|\, s_i\rangle$. But then $\mathcal{H}$ would also have to contain s_i contradicting the earlier statement. (We note that $\text{DEAL}^{\mathcal{G}}_{s_i} = \langle b \,|\, b, \tilde{x}_i\rangle$ where $\tilde{x}_i$ is the true literal of x_i, and there is no strictly better outcome for s_i.)

So we can assume from now on that $\mathcal{H}$ contains b. The idea of the remaining argument is to show that the assumption that $\mathcal{H}$ dominates $\mathcal{G}$ implies that in fact $\mathcal{H} = \mathcal{G}$ — so $\mathcal{H}$ cannot strictly dominate $\mathcal{G}$. To this end, we examine the arcs of $\mathcal{D}$ one by one. For each arc (u, v), we use the relation $\text{DEAL}^{\mathcal{H}}_z \succeq \text{DEAL}^{\mathcal{G}}_z$ for some $z \in \{u, v\}$, to show that (u, v) belongs to $\mathcal{H}$ if and only if it belongs to $\mathcal{G}$. We will divide this argument into a sequence of claims.

Claim 1: *$\mathcal{H}$ contains all arcs (v, b) and (b, v), for $v \neq b$.* From the definition of the preference poset of b, $\mathcal{H}$ must contain all incoming arcs of b. This gives us that $\mathcal{H}$ is spanning. For each $v \neq b$, any outcome $\omega \succeq \text{DEAL}^{\mathcal{G}}_v$ that has outgoing arc (v, b) must also have incoming arc (b, v). This proves the claim.

Claim 2: *$\mathcal{H}$ does not contain any arc (a_j, c_j) or (c_j, a_j).* Indeed, no outcome of a_j that has arc (a_j, c_j) is better than $\langle b \,|\, b\rangle = \text{DEAL}^{\mathcal{G}}_{a_j}$, so $\mathcal{H}$ cannot contain (a_j, c_j). Similarly, no outcome of c_j that has arc (c_j, a_j) is better than $\langle b, T(c_j) \,|\, b\rangle = \text{DEAL}^{\mathcal{G}}_{c_j}$, so $\mathcal{H}$ cannot contain (c_j, a_j).

Claim 3: *For each literal $\tilde{x}_i$ in clause c_j, $\mathcal{H}$ contains $(\tilde{x}_i, c_j)$ iff $\tilde{x}_i$ is true.* Suppose first that $\tilde{x}_i$ is a literal in c_j that is true. There is no outcome of c_j that does not contain $(\tilde{x}_i, c_j)$ and is better than $\langle b, T(c_j) \,|\, b\rangle = \text{DEAL}^{\mathcal{G}}_{c_j}$, so $\mathcal{H}$ must contain $(\tilde{x}_i, c_j)$. Next, suppose that $\tilde{x}_i$ is false. Then no outcome of $\tilde{x}_i$ that contains $(\tilde{x}_i, c_j)$ is better than $\langle b \,|\, b\rangle = \text{DEAL}^{\mathcal{G}}_{\tilde{x}_i}$. Thus $\mathcal{H}$ cannot contain $(\tilde{x}_i, c_j)$.

Claim 4: *For each literal $\tilde{x}_i$, $\mathcal{H}$ contains arc $(s_i, \tilde{x}_i)$ iff $\tilde{x}_i$ is true.* Suppose first that literal $\tilde{x}_i$ is true. From the previous claim, we have that the outgoing arcs to $C(\tilde{x}_i)$ are in $\mathcal{H}$. There is no outcome of $\tilde{x}_i$ that contains the arcs to $C(\tilde{x}_i)$, does not contain arc $(s_i, \tilde{x}_i)$, and is better than $\langle b, s_i \,|\, b, C(\tilde{x}_i)\rangle = \text{DEAL}^{\mathcal{G}}_{\tilde{x}_i}$. Thus $\mathcal{H}$ must contain $(s_i, \tilde{x}_i)$. Next, suppose that $\tilde{x}_i$ is false, and let $\bar{\tilde{x}}_i$ be the negation of $\tilde{x}_i$ (that is, the true literal of x_i). Then there is no outcome of s_i that contains arc $(s_i, \tilde{x}_i)$ and is better than $\langle b \,|\, b, \bar{\tilde{x}}_i\rangle = \text{DEAL}^{\mathcal{G}}_{s_i}$. Thus $\mathcal{H}$ cannot contain $(s_i, \tilde{x}_i)$.

Claim 5: *$\mathcal{H}$ does not contain any arc (s_i, t_i) or (t_i, s_i).* There is no outcome of s_i that has an arc (s_i, t_i) and is better than $\langle b \,|\, b, \tilde{x}_i\rangle = \text{DEAL}^{\mathcal{G}}_{s_i}$, where $\tilde{x}_i$ is the true literal of x_i. So $\mathcal{H}$ cannot contain (s_i, t_i). Also, there is no outcome of t_i that has arc (t_i, s_i), does not have arc (s_i, t_i), and is better than $\langle b \,|\, b\rangle = \text{DEAL}^{\mathcal{G}}_{t_i}$. So $\mathcal{H}$ cannot contain (t_i, s_i).

($\Leftarrow$) Assume now that $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ that satisfies properties (c.1)-(c.3). From $\mathcal{G}$ we will construct a satisfying assignment for α.

Since $\text{DEAL}^{\mathcal{G}}_b \succeq \text{DEAL}^{\mathcal{D}}_b$, $\mathcal{G}$ must contain incoming arcs of b from all other vertices. For each $v \neq b$, any outcome of v that is at least as good as $\text{DEAL}^{\mathcal{D}}_v$ and contains arc (v, b) must also contain arc (b, v). So $\mathcal{G}$ contains all outgoing arcs of b.

Next, we claim that, for each variable x_i, $\mathcal{G}$ contains at most one of arcs (s_i, x_i) and $(s_i, \bar{x}_i)$. Indeed, towards contradiction, suppose that $\mathcal{G}$ contains both arcs (s_i, x_i) and $(s_i, \bar{x}_i)$. The best possible outcome of s_i with both arcs (s_i, x_i) and $(s_i, \bar{x}_i)$ is $\langle b, t_i \,|\, x_i, \bar{x}_i\rangle$, and using the preferences of s_i, we obtain $\langle t_i \,|\, t_i\rangle \succ \langle b, t_i \,|\, x_i, \bar{x}_i\rangle \succeq \text{DEAL}^{\mathcal{G}}_{s_i}$. Regarding t_i, we have already established that t_i has arcs to and from b, and the best such outcome for t_i is $\langle b, s_i \,|\, b\rangle$. Thus, using the preferences of t_i we obtain $\langle s_i \,|\, s_i\rangle \succ \langle b, s_i \,|\, b\rangle \succeq \text{DEAL}^{\mathcal{G}}_{t_i}$.

So we could take $\mathcal{H}$ to consist of s_i, t_i, and arcs (s_i, t_i) and (t_i, s_i), and this $\mathcal{H}$ would strictly dominate $\mathcal{G}$, contradicting our assumption that $\mathcal{G}$ satisfies condition (c.3).

Using the claim in the previous paragraph, we construct a satisfying assignment for α as follows: For each variable x_i, set it to true if $\mathcal{G}$ contains (s_i, x_i); otherwise set it to false. (Note that $\mathcal{G}$ may not contain any arc (s_i, x_i), $(s_i, \bar{x}_i)$, in which case we could set the value of x_i arbitrarily.) This truth assignment is well defined.

We now argue that this truth assignment satisfies α. Consider any clause c_j. Vertex c_j must have at least one incoming arc $(\tilde{x}_i, c_j)$ in $\mathcal{G}$, because otherwise we couldn't have $\text{DEAL}^{\mathcal{G}}_{c_j} \succeq \text{DEAL}^{\mathcal{D}}_{c_j}$. Similarly, if $\mathcal{G}$ contains this arc $(\tilde{x}_i, c_j)$ then it must also contain arc $(s_i, \tilde{x}_i)$, because otherwise we couldn't have $\text{DEAL}^{\mathcal{G}}_{\tilde{x}_i} \succeq \text{DEAL}^{\mathcal{D}}_{\tilde{x}_i}$. This implies that literal $\tilde{x}_i$ is true in our truth assignment, so clause c_j is true as well. As this holds for each clause, we can conclude that α is satisfied. □

6. Σ_2-Completeness

In the previous section, we showed that SwapAtomic is NP-hard. In this section, we tighten the complexity classification of SwapAtomic, and show that it is in fact Σ_2^{P}-complete. Recall that $\Sigma_2^{\mathsf{P}} = \mathsf{NP}^{\mathsf{NP}}$ is the class of problems at the 2nd level of the polynomial hierarchy that consists of problems solvable non-deterministically in polynomial time with an NP oracle.

Our proof is based on a reduction from a restricted variant of the $\exists\forall$DNF problem. An instance of $\exists\forall$DNF is a boolean expression $\alpha = \exists\mathbf{x}\forall\mathbf{y}\beta(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = (x_1, ..., x_k)$ and $\mathbf{y} = (y_1, ..., y_l)$ are vectors of boolean variables and $\beta(\mathbf{x}, \mathbf{y})$ is a quantifier-free boolean expression in disjunctive normal form, that is $\beta(\mathbf{x}, \mathbf{y}) = \tau_1 \vee \tau_2 \vee ... \vee \tau_m$, and each term τ_g is a conjunction of literals involving different variables. The goal is to determine whether α is true. $\exists\forall$DNF is a canonical Σ_2^{P}-complete problem [29], [26]. The restriction of $\exists\forall$DNF that we use in our proof, denoted $\exists\forall\mathsf{DNF}_{1x}$, consists of instances $\alpha = \exists\mathbf{x}\forall\mathbf{y}\beta(\mathbf{x}, \mathbf{y})$ where each term of β includes exactly one $\mathbf{x}$-literal and one or more $\mathbf{y}$-literals that involve different variables.

Lemma 2. $\exists\forall\mathsf{DNF}_{1x}$ *is* Σ_2^{P}*-complete.*

The proof can be found in the appendix.

We remember that SwapAtomic is the decision problem of deciding whether a swap system has an atomic protocol.

Theorem 5. SwapAtomic *is* Σ_2^{P}*-complete.*

The complete proof can be found in the appendix. In the remainder of this section, we briefly present a high-level description of our reduction and the accompanying proof.

According to Theorem 3, a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ has an atomic swap protocol if and only if $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ with the following properties: (c.1) $\mathcal{G}$ is piece-wise strongly connected and has no isolated vertices, (c.2) $\mathcal{G}$ dominates $\mathcal{D}$, and (c.3) no subgraph $\mathcal{H}$ of $\mathcal{D}$ strictly dominates $\mathcal{G}$. This characterization is of the form $\exists\mathcal{G}.\neg\exists\mathcal{H}.\,\pi(\mathcal{G}, \mathcal{H})$, where $\pi(\mathcal{G}, \mathcal{H})$ is a polynomial-time decidable predicate, so it immediately implies that SwapAtomic is in Σ_2^{P}. Thus it remains to show that SwapAtomic is Σ_2^{P}-hard.

To prove Σ_2^{P}-hardness, we present a polynomial-time reduction from the above-defined decision problem $\exists\forall\mathsf{DNF}_{1x}$. Let the given instance of $\exists\forall\mathsf{DNF}_{1x}$ be $\alpha = \exists\mathbf{x}\forall\mathbf{y}\beta(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = (x_1, ..., x_k)$ and $\mathbf{y} = (y_1, ..., y_l)$ are vectors of boolean variables and $\beta(\mathbf{x}, \mathbf{y}) = \tau_1 \vee \tau_2 \vee ... \vee \tau_m$, where each τ_g is a conjunction of one $\mathbf{x}$-literal and one or more $\mathbf{y}$-literals. Our reduction converts α into a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ such that α is true if and only if $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ that satisfies conditions (c.1)-(c.3) from Theorem 3.

The following informal interpretation of $\exists\forall\mathsf{DNF}_{1x}$ will be helpful in understanding our reduction. Say that a truth assignment to some variables "kills" a term τ_g if it sets one of its literals to false. A truth assignment $\boldsymbol{\phi}$ to the $\mathbf{x}$-variables will kill some terms, while others will survive. Thus α will be true for assignment $\boldsymbol{\phi}$ iff there is no assignment $\boldsymbol{\psi}$ for the $\mathbf{y}$-variables that kills all terms that survived $\boldsymbol{\phi}$. In our reduction, the existence of this assignment $\boldsymbol{\phi}$ will be represented by the existence of subgraph $\mathcal{G}$. The non-existence of $\boldsymbol{\psi}$ that kills all terms that survived $\boldsymbol{\phi}$ will be represented by the non-existence of a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$.

Throughout this section, the negation of a boolean variable x_i will be denoted $\bar{x}_i$. We will also use notation $\tilde{x}_i$ for an unspecified literal of x_i, that is $\tilde{x}_i \in \{x_i, \bar{x}_i\}$. The same conventions apply to the variables y_j.

We now give an overview of our reduction. The digraph $\mathcal{D}$ consists of several "gadgets". There will be $\exists$-gadgets, which correspond to the variables x_i and will be used to set their values, through the choice of subgraphs that $\mathcal{G}$ includes. Then there is the $\forall$-gadget, that contains "sub-gadgets" representing the literals $\tilde{y}_j$ and the terms τ_g. These gadgets will allow for the values of the variables y_j to admit all possible assignments. If any setting of these values kills all terms not yet killed by the variables x_i, this gadget will contain a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$. Figure 7 shows a single $\exists$-gadget and Figure 8 shows the $\forall$-gadget. As we explain the high-level intuition, we gradually visit vertices and explain their purpose.

The argument is based on several ideas. One, we design the preference posets of $\tilde{x}_i$'s so that $\mathcal{G}$ is forced to choose between two possible subsets of arcs within the $\exists$-gadget. The choice between these two subsets of arcs corresponds to choosing a truth assignment for variable x_i. We focus on the literals $\tilde{x}_i$ that are set to false, since these kill the terms where they appear. If $\tilde{x}_i$ is set to false, its arcs to the terms τ_g's in which the literal appears will be included in $\mathcal{G}$ (the first subset), otherwise its arc to $\tilde{z}_i$ will be included in $\mathcal{G}$ (the second subset).

Another idea is that vertices outside of the $\forall$-gadget have their preference posets defined in such a way that their arcs in $\mathcal{G}$ define an outcome that is already the best for them. Therefore, if a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$ does indeed exist, we know it must appear in the $\forall$-gadget. This leads into the key idea of the $\forall$-gadget. The vertices in this gadget can have outcomes that are better than their outcomes

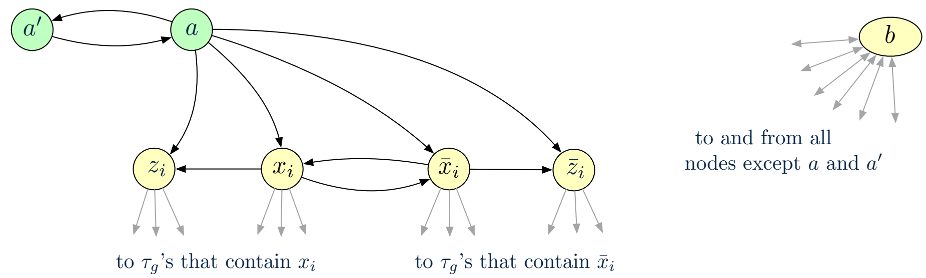

Figure 7. The construction of digraph $\mathcal{D}$ in the proof of Σ_2^{P}-hardness. This figure shows vertices a, a', b, and an $\exists$-gadget for variable x_i. The arcs to and from b are shown as bi-directional arrows at b.

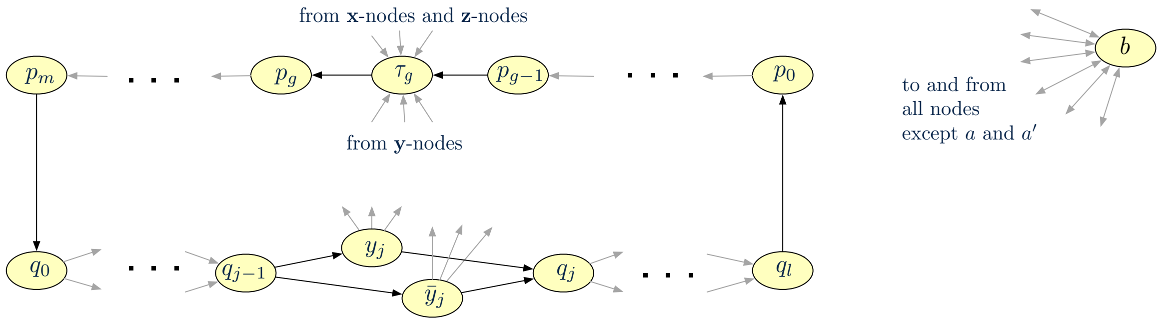

Figure 8. The construction of digraph $\mathcal{D}$ in the proof of Σ_2^{P}-hardness. This figures shows the $\forall$-gadget, namely the part of $\mathcal{D}$ that contains the vertices that simulate setting the values of the y_j-variables and the terms τ_g. The arcs to and from b are shown as bi-directional arrows at b.

in $\mathcal{G}$. All the arcs in these better outcomes together form the cycle

$$\begin{aligned}\mathcal{C} \quad = \quad & q_0 \to \tilde{y}_1 \to \ldots \to \tilde{y}_l \to q_l \to \\ & p_0 \to \tau_1 \to \ldots \tau_m \to p_m \to \\ & q_0\end{aligned} \tag{1}$$

for some choice of the literals $\tilde{y}_1, \ldots, \tilde{y}_l$. We design the preference posets of each τ_g so that its outcome in $\mathcal{G}$ can only be improved (specifically, towards $\mathcal{C}$) only if it receives an arc from one of its literals — in other words, if it is killed by that literal. This way, $\mathcal{G}$ will have a strictly dominating subgraph $\mathcal{H}$ (namely cycle $\mathcal{C}$) only if all terms are killed, i.e. when α is false.

Next, we provide brief insight to the important vertices and how they help capture the ideas above. Firstly, we want to simulate a truth assignment for variable x_i, which we represent by having $\mathcal{G}$ choose between two subsets of arcs in the corresponding $\exists$-gadget for x_i. Intuitively, one subset corresponds to assigning x_i to true while the other subset corresponds to assigning x_i to false. These two subsets of arcs are established by how we define the preference posets of x_i and $\bar{x}_i$. In order to force $\mathcal{G}$ to make a choice (instead of taking all the arcs), we introduce the auxiliary vertex a, which has arcs to every literal $\tilde{x}_i$. The graph $\mathcal{D}$ has two strongly connected components: (1) a and a', and (2) all the other vertices. We claim graph $\mathcal{G}$ cannot include any arcs from a to the literal vertices. Otherwise, since there is no edge from the second component to the first, dropping those arcs always results in a better outcome for the first component. However, that contradicts condition (c.3). Then, for $\mathcal{G}$ to satisfy condition (c.2), $\mathcal{G}$ is forced to make a choice between the two subsets of arcs. Specifically, $\mathcal{G}$ must choose to include all arcs from x_i to its terms (corresponds to setting x_i to be false) or all arcs from $\bar{x}_i$ to its terms (corresponds to setting x_i to be true).

Next, we want to simulate a term τ_g being killed. We achieve this by designing the preference poset of each τ_g so that if it receives its arc from its $\mathbf{x}$-literal, it would prefer its outcome in the cycle $\mathcal{C}$ over its outcome in $\mathcal{G}$. A term τ_g can also be killed by one of its $\mathbf{y}$-literals, which we describe later.

Now, we want to simulate checking whether there is a truth assignment for the $\mathbf{y}$-variables that make $\forall \mathbf{y} \beta(\boldsymbol{\phi}, \mathbf{y})$ false, where $\boldsymbol{\phi}$ is a truth assignment over the $\mathbf{x}$ variables. In other words, $\mathcal{G}$'s assignment of the $\mathbf{x}$ variables have killed some terms and now we want to see if $\mathcal{H}$ can give an assignment of the $\mathbf{y}$ variables that kill the surviving terms.

First, we need to simulate a truth assignment for each variable y_j. This is simple: we flank y_j and $\bar{y}_j$ by vertices q_{j-1} and q_j as seen in Figure 8. We define the preference posets of q_{j-1} and q_j in a way that only one of y_j or $\bar{y}_j$ can be included in the cycle $\mathcal{C}$, thus forcing $\mathcal{H}$ to choose between them. If $\mathcal{H}$ selects a vertex $\tilde{y}_j$, then $\tilde{y}_j$ will send an arc to every term it appears in. This corresponds to assigning $\tilde{y}_j$ to false.

We additionally define τ_g's poset so that if it receives an arc from any of its $\mathbf{y}$-literals, it wants to join $\mathcal{C}$. At this point, we have represented τ_g being killed if it receives either its $\mathbf{x}$-literal arc or any of its $\mathbf{y}$-literal arcs. (The $\tilde{z}_i$ vertices are actually used for this purpose. They help distinguish between when a term is killed by their $\mathbf{x}$-literal and when a term survived in which it needs to be killed by a $\mathbf{y}$-literal.)

The $\forall$-gadget is designed in the following manner: the preference posets of the q_j, $\tilde{y}_j$, and p_g vertices are such that if every τ_g prefers $\mathcal{C}$, so will they; otherwise, if any τ_g does not prefer $\mathcal{C}$, then none of the vertices can cooperatively deviate to improve their outcomes. In other words, each τ_g acts as a bottleneck for the cycle $\mathcal{C}$, thus we focus only on the τ_g's.

We give an analogy to better understand the remainder of the reduction. Each τ_g is given a vote to whether or not they want to participate in cycle $\mathcal{C}$. In order for $\mathcal{C}$ to pass, it must receive a *unanimous* vote from every τ_g. Vertex τ_g only casts its vote to join $\mathcal{C}$ if it receives an arc from either its $\mathbf{x}$-literal arc or any of its $\mathbf{y}$-literal arcs (corresponds to being killed). Then, $\mathcal{G}$'s selection in each $\exists$-gadget (truth assignment over $\mathbf{x}$ variables) caused *some* τ_g's to vote for $\mathcal{C}$. Now, $\mathcal{H}$ is tasked with selecting vertices $\tilde{y}_1, ..., \tilde{y}_l$ (truth assignment over $\mathbf{y}$ variables) so that the *remaining* τ_g's also vote for $\mathcal{C}$. At the end of this, if the τ_g's unanimously voted for $\mathcal{C}$, then there is an $\mathcal{H}$ that strictly dominates $\mathcal{G}$, namely the subgraph induced by $\mathcal{C}$. (This corresponds to giving an assignment $\mathbf{y} \mapsto \boldsymbol{\psi}$ such that $\beta(\boldsymbol{\phi}, \boldsymbol{\psi})$ is false.) Otherwise, if $\mathcal{H}$ cannot give such a selection over $\tilde{y}_1, ..., \tilde{y}_l$, then there is no $\mathcal{H}$ that strictly dominates $\mathcal{G}$. (This corresponds to $\forall \mathbf{y} \beta(\boldsymbol{\phi}, \mathbf{y})$ being true, i.e. α is true.)

The remaining vertices are primarily used for convenience and to influence the behavior/preference posets of their neighbors. In other words, they make the topology of $\mathcal{G}$ and $\mathcal{H}$ predictable, holding them to a particular form. For example, vertex b is used to guarantee condition (c.1), the piece-wise strong connectivity condition. Also, it is used where vertices would otherwise have no incoming or outgoing arcs.

To conclude the section, we provide some brief insight to both directions of the proof. In the ($\Rightarrow$) implication, we show that if α is true, then the swap graph $\mathcal{D}$ has a subgraph $\mathcal{G}$ that satisfies the properties of Theorem 3. We begin by fixing some truth assignment $\mathbf{x} \mapsto \boldsymbol{\phi}$ that makes α true. We convert $\boldsymbol{\phi}$ into a graph $\mathcal{G}$ that satisfies the properties of Theorem 3 using the ideas described above. Conditions (c.1) and (c.2) can be verified by routine inspection, leaving condition (c.3) that $\mathcal{G}$ does not have a strictly dominating subgraph $\mathcal{H}$. The idea is, towards contradiction, if such an $\mathcal{H}$ existed, we could convert it into an assignment $\boldsymbol{\psi}$ of the $\mathbf{y}$ variables so that $\beta(\boldsymbol{\phi}, \boldsymbol{\psi})$ is false. This contradicts the fact that α is true.

In the ($\Leftarrow$) implication, we show that if $\mathcal{D}$ has a subgraph $\mathcal{G}$ that satisfies the properties of theorem 3, then α is true. We begin by showing that the topology of $\mathcal{G}$ must have a certain form; specifically, it is representative of the graph $\mathcal{G}$ we constructed in the proof of the ($\Rightarrow$) implication. This allows us to reconstruct an assignment of the $\mathbf{x}$ variables. We then show, again by contradiction, that $\forall \mathbf{y} \beta(\boldsymbol{\phi}, \mathbf{y})$ must be true. If it were not, we can take a falsifying assignment $\mathbf{y} \mapsto \boldsymbol{\psi}$ and convert it into a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$. However, this contradicts condition (c.3) of $\mathcal{G}$.

7. Related Works

The fair exchange problem [23], [16], [7], [4], [5] was of interest even before the blockchain technology. It arises when two parties want to exchange their assets, and the outcome must be either that the two parties end up trading their assets, or that they both keep their assets. However, in contrast to the swap problem, some trust in a third party is often assumed. The optimistic fair exchange protocol [23] relies on invisible trusted parties: parties that work as a background service and intervene only in case of a misbehaviour. Similarly, the secure group barter protocol [16] studies multi-party barter with semi-trusted agents.

To the best of our knowledge, it was back in 2013 when the notion of cross-chain swaps first emerged in an online forum [34]. Atomic cross-chain swap is since an active problem for the blockchain community [8], [34], [9], [10]. The two wiki pages [8] and [34] and later platforms such as deCRED [13] proposed protocols for bilateral swaps. However, these projects offer only two-party transactions. Later, protocols for cross-chain swaps and transactions [19], [20], [18], [33] emerged that can work for an arbitrary number of parties; however, they assumed the predefined preference relation that we saw earlier for all the parties.

These protocols motivated a host of follow-up research. The time and space complexity [21] and privacy guarantees [14] of the protocol were improved. The former [21] uses a model in which each arc has explicit numeric values perceived by both sender and recipient. This implicitly induces a preference structure over each user's possible outcomes. Furthermore, it is used to restrict the type of swap graphs that permit atomic protocols. Further, extensions to support off-chain steps [30] and reduce the asset lock-up time [36] appeared. Others presented hardness and impossibility results [38], [12] formal verification [25], and protocols with all-or-nothing guarantees [37] and success guarantees under synchrony assumptions [35]. Others proposed moving assets [31] and smart contracts [17] across blockchains, and executing code that spans multiple blockchains [28], and presented implementations for industrial blockchains [3], [2], [11], [32].

Payment channel networks process multi-hop payments in the same blockchain through a sequence of channels using Hash Timelock Contracts [27], [1] or adaptor signatures [22]. Recent protocols such as AMCU [15], Sprites [24] and Thora [6] support more general topologies for transactions.

In contrast to previous work, this paper presented a generalized model of swaps where each party can specify a personalized preference on their set of incoming and outgoing assets in a finer manner, e.g. dependencies between subsets of acquired and traded assets.

8. Conclusion

We presented a general swap model that allows each party to specify their preference on their possible outcomes. We saw that Herlihy's pioneering protocol is a uniform and Nash strategy in this model; however, it is not a strong

Nash strategy. We presented a characterization of the class of swap graphs that have uniform and Strong Nash protocols. Interestingly, Herlihy's protocol is such a strategy when executed on a particular subgraph of the swap graphs in this class. We further presented reductions that shows the NP-harness and Σ_2^{P}-completeness of the decision problem for this class.

Acknowledgements. I would like to thank Annie Semb for her unconditional love and support over the years. I will remember you always. I miss you dearly. *Eric*.

References

[1] Raiden network. https://raiden.network/.

[2] Submarine swap in lightning network. https://wiki.ion.radar.tech/tech/research/submarine-swap.

[3] What is atomic swap and how to implement it. https://www.axiomadev.com/blog/what-is-atomic-swap-and-how-to-implement-it/.

[4] N. Asokan, Matthias Schunter, and Michael Waidner. Optimistic protocols for fair exchange. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, pages 7–17, New York, NY, USA, 1997. ACM. URL: http://doi.acm.org/10.1145/266420.266426, doi:10.1145/266420.266426.

[5] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18:593–610, 1997.

[6] Lukas Aumayr, Kasra Abbaszadeh, and Matteo Maffei. Thora: Atomic and privacy-preserving multi-channel updates. *Cryptology ePrint Archive*, 2022.

[7] Michael Ben-Or, Oded Goldreich, Silvio Micali, and Ronald L. Rivest. A fair protocol for signing contracts. In Wilfried Brauer, editor, *Automata, Languages and Programming*, pages 43–52, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[8] bitcoinwiki. Atomic swap. https://en.bitcoin.it/wiki/Atomic_swap. [Online; accessed 23-January-2021].

[9] bitcoinwiki. Hashed timelock contracts. https://en.bitcoin.it/wiki/Hashed_Timelock_Contracts. [Online; accessed 23-January-2021].

[10] Sean Bowe and Daira Hopwood. Hashed time-locked contract transactions. https://github.com/bitcoin/bips/blob/master/bip-0199.mediawiki. [Online; accessed 23-January-2021].

[11] Gewu Bu, Riane Haouara, Thanh-Son-Lam Nguyen, and Maria Potop-Butucaru. Cross hyperledger fabric transactions. In *Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pages 35–40, 2020.

[12] Eric Chan and Mohsen Lesani. Brief announcement: Brokering with hashed timelock contracts is np-hard. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, pages 199–202, 2021.

[13] deCRED. On-chain atomic swaps for decred and other cryptocurrencies. https://github.com/decred/atomicswap. [Online; accessed 27-January-2019].

[14] Apoorvaa Deshpande and Maurice Herlihy. Privacy-preserving cross-chain atomic swaps. In *International Conference on Financial Cryptography and Data Security*, pages 540–549. Springer, 2020.

[15] Christoph Egger, Pedro Moreno-Sanchez, and Matteo Maffei. Atomic multi-channel updates with constant collateral in bitcoin-compatible payment-channel networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 801–815, 2019.

[16] Matt Franklin and Gene Tsudik. Secure group barter: Multi-party fair exchange with semi-trusted neutral parties. In Rafael Hirchfeld, editor, *Financial Cryptography*, pages 90–102, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[17] Enrique Fynn, Alysson Bessani, and Fernando Pedone. Smart contracts on the move. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 233–244. IEEE, 2020.

[18] Ethan Heilman, Sebastien Lipmann, and Sharon Goldberg. The arwen trading protocols. In *International Conference on Financial Cryptography and Data Security*, pages 156–173. Springer, 2020.

[19] Maurice Herlihy. Atomic cross-chain swaps. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, PODC '18, pages 245–254, New York, NY, USA, 2018. ACM. URL: http://doi.acm.org/10.1145/3212734.3212736, doi:10.1145/3212734.3212736.

[20] Maurice Herlihy, Barbara Liskov, and Liuba Shrira. Cross-chain deals and adversarial commerce. *Proc. VLDB Endow.*, 13(2):100–113, October 2019.

[21] Soichiro Imoto, Yuichi Sudo, Hirotsugu Kakugawa, and Toshimitsu Masuzawa. Atomic cross-chain swaps with improved space and local time complexity. In *International Symposium on Stabilizing, Safety, and Security of Distributed Systems*, pages 194–208. Springer, 2019.

[22] Giulio Malavolta, Pedro Moreno-Sanchez, Clara Schneidewind, Aniket Kate, and Matteo Maffei. Anonymous multi-hop locks for blockchain scalability and interoperability. *Cryptology ePrint Archive*, 2018.

[23] Silvio Micali. Simple and fast optimistic protocols for fair electronic exchange. In *Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing*, PODC '03, pages 12–19, New York, NY, USA, 2003. ACM. URL: http://doi.acm.org/10.1145/872035.872038, doi:10.1145/872035.872038.

[24] Andrew Miller, Iddo Bentov, Surya Bakshi, Ranjit Kumaresan, and Patrick McCorry. Sprites and state channels: Payment networks that go faster than lightning. In *International Conference on Financial Cryptography and Data Security*, pages 508–526. Springer, 2019.

[25] Zeinab Nehaï, François Bobot, Sara Tucci-Piergiovanni, Carole Delporte-Gallet, and Hugues Fauconnier. A tla+ formal proof of a cross-chain swap. In *23rd International Conference on Distributed Computing and Networking*, pages 148–159, 2022.

[26] Christos Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, 1994.

[27] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[28] Peter Robinson and Raghavendra Ramesh. General purpose atomic crosschain transactions. In *IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2021, Sydney, Australia, May 3-6, 2021*, pages 1–3. IEEE, 2021. doi:10.1109/ICBC51069.2021.9461132.

[29] Marcus Schaefer and Christopher Umans. Completeness in the polynomial-time hierarchy: a compendium. *Sigact News*, 33(3):32–49, 2002.

[30] Narges Shadab, Farzin Houshmand, and Mohsen Lesani. Cross-chain transactions. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9. IEEE, 2020.

[31] Marten Sigwart, Philipp Frauenthaler, Christof Spanring, and Stefan Schulte. Decentralized cross-blockchain asset transfers. *arXiv preprint arXiv:2004.10488*, 2020.

[32] Sri AravindaKrishnan Thyagarajan, Giulio Malavolta, and Pedro Moreno-Sánchez. Universal atomic swaps: Secure exchange of coins across all blockchains. *Cryptology ePrint Archive*, 2021.

[33] Sri AravindaKrishnan Thyagarajan, Giulio Malavolta, and Pedro Moreno-Sanchez. Universal atomic swaps: Secure exchange of coins across all blockchains. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1299–1316. IEEE, 2022.

[34] Tier Nolan. Alt chains and atomic transfers. https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949, 2013. [Online; accessed 23-January-2021].

[35] Rob van Glabbeek, Vincent Gramoli, and Pierre Tholoniat. Feasibility of cross-chain payment with success guarantees. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 579–581, 2020.

[36] Yingjie Xue and Maurice Herlihy. Hedging against sore loser attacks in cross-chain transactions. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, PODC'21, page 155–164, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3465084.3467904.

[37] Victor Zakhary, Divyakant Agrawal, and Amr El Abbadi. Atomic commitment across blockchains. *Proceedings of the VLDB Endowment*, 13(9).

[38] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J Knottenbelt. Sok: Communication across distributed ledgers. In *International Conference on Financial Cryptography and Data Security*, pages 3–36. Springer, 2021.

Appendix

9. Σ_2-Completeness

In this section, we give the complete, detailed proof described in 6. That is, we consider the complexity of determining whether a swap system has an atomic swap protocol, showing that this problem is Σ_2^{P}-complete. Recall that $\Sigma_2^{\mathsf{P}} = \mathsf{NP}^{\mathsf{NP}}$ is the class of problems at the 2nd level of the polynomial hierarchy that consists of problems solvable non-deterministically in polynomial time with an NP oracle.

Our proof is based on a reduction from a restricted variant of the $\exists\forall\mathsf{DNF}$ problem. An instance of $\exists\forall\mathsf{DNF}$ is a boolean expression $\alpha = \exists \mathbf{x} \forall \mathbf{y} \beta(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = (x_1, ..., x_k)$ and $\mathbf{y} = (y_1, ..., y_l)$ are vectors of boolean variables and $\beta(\mathbf{x}, \mathbf{y})$ is a quantifier-free boolean expression in disjunctive normal form, that is $\beta(\mathbf{x}, \mathbf{y}) = \tau_1 \vee \tau_2 \vee ... \vee \tau_m$, and each term τ_g is a conjunction of literals involving different variables. The goal is to determine whether α is true. $\exists\forall\mathsf{DNF}$ is a canonical Σ_2^{P}-complete problem [29], [26][5]. The problem remains Σ_2^{P}-complete even restricted to instances where each term in β has only three literals. We denote this variant by $\exists\forall\mathsf{3DNF}$.

Throughout this section, the negation of a boolean variable x_i will be denoted $\bar{x}_i$. We will also use notation $\tilde{x}_i$ for an unspecified literal of x_i, that is $\tilde{x}_i \in \{x_i, \bar{x}_i\}$. The same conventions apply to the variables y_j.

The restriction of $\exists\forall\mathsf{DNF}$ that we use in our proof, denoted $\exists\forall\mathsf{DNF}_{1x}$, consists of instances $\alpha = \exists \mathbf{x} \forall \mathbf{y} \beta(\mathbf{x}, \mathbf{y})$ where each term of β includes exactly one $\mathbf{x}$-literal and one or more $\mathbf{y}$-literals that ifnvolve different variables.

We first prove the following lemma:

Lemma 3. *$\exists\forall\mathsf{DNF}_{1x}$ is Σ_2^{P}-complete.*

Proof. We show how to convert a given instance $\alpha = \exists \mathbf{x} \forall \mathbf{y} \beta(\mathbf{x}, \mathbf{y})$ of $\exists\forall\mathsf{3DNF}$ into an instance α' of $\exists\forall\mathsf{DNF}_{1x}$ such that α is true iff α' is true.

First, we can assume that β does not have terms with only $\mathbf{x}$-literals, since such formulas α are trivially true. All terms that have exactly one $\mathbf{x}$-literal will remain unchanged.

Consider a term with two $\mathbf{x}$-literals, say $\tau_g = \tilde{x}_p \wedge \tilde{x}_q \wedge \tilde{y}_r$. Add another variable y' and replace τ_g by $(\tilde{x}_p \wedge \tilde{y}_r \wedge y') \vee (\bar{y}' \wedge \tilde{x}_q \wedge \tilde{y}_r)$. Let β' be the boolean expression obtained from β by this replacement, and $\alpha' = \exists \mathbf{x} \forall \mathbf{y} \forall y' \beta'(\mathbf{x}, \mathbf{y}, y')$. Then, by straightforward verification, α is true for a given truth assignment for $\mathbf{x}$ if and only if α' is true for the same assignment for $\mathbf{x}$.

By applying these replacements, we will eventually eliminate all terms that have two or zero $\mathbf{x}$-literals, thus converting α into the $\exists\forall\mathsf{DNF}_{1x}$ form. □

Theorem 6. *Let* $\mathsf{SwapAtomic}$ *be the decision problem of deciding whether a swap system has an atomic protocol.* $\mathsf{SwapAtomic}$ *is Σ_2^{P}-complete.*

5. Notations for this problem and its variants vary across the literature. Our notations use the convention in [29].

Proof. According to Theorem 3, a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ has an atomic swap protocol if and only if $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ with the following properties: (c.1) $\mathcal{G}$ is piece-wise strongly connected and has no isolated vertices, (c.2) $\mathcal{G}$ dominates $\mathcal{D}$, and (c.3) no subgraph $\mathcal{H}$ of $\mathcal{D}$ strictly dominates $\mathcal{G}$. This characterization is of the form $\exists \mathcal{G}\,(\,\neg\exists\mathcal{H} : \pi(\mathcal{G}, \mathcal{H})\,)$, where $\pi(\mathcal{G}, \mathcal{H})$ is a polynomial-time decidable predicate, so it immediately implies that $\mathsf{SwapAtomic}$ is in Σ_2^{P}. Thus it remains to show that $\mathsf{SwapAtomic}$ is Σ_2^{P}-hard.

To prove Σ_2^{P}-hardness, we give a polynomial-time reduction from the above-defined decision problem $\exists\forall\mathsf{DNF}_{1x}$. Let the given instance of $\exists\forall\mathsf{DNF}_{1x}$ be $\alpha = \exists \mathbf{x} \forall \mathbf{y} \beta(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = (x_1, ..., x_k)$ and $\mathbf{y} = (y_1, ..., y_l)$ are vectors of boolean variables and $\beta(\mathbf{x}, \mathbf{y}) = \tau_1 \vee \tau_2 \vee ... \vee \tau_m$, with each τ_g being a conjunction of one $\mathbf{x}$-literal and one or more $\mathbf{y}$-literals. Our reduction converts α into a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ such that α is true if and only if $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ that satisfies conditions (c.1)-(c.3) from Theorem 3.

The following informal interpretation of $\exists\forall\mathsf{DNF}_{1x}$ will be helpful in understanding our reduction. Say that a truth assignment to some variables *kills* a term τ_g if it sets one of its literals to false. A truth assignment ϕ to the $\mathbf{x}$-variables will kill some terms, while other will survive. Thus α will be true for assigment ϕ iff there is no assignment ψ for the $\mathbf{y}$-variables that kills all terms that survived ϕ. In our reduction, the existence of this assigment ϕ will be represented by the existence of subgraph $\mathcal{G}$. The non-existence of ψ that kills all terms that survived ϕ will be represented by the non-existence of a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$.

We now describe our reduction. The digraph $\mathcal{D}$ will consists of several "gadgets". There will be $\exists$-gadgets, which correspond to the variables x_i and will be used to set their values, through an appropriate choices of subgraph $\mathcal{G}$. Then there is the $\forall$-gadget, that contains "sub-gadgets" representing the literals $\tilde{y}_j$ and the terms τ_g. These gadgets will allow for the values of the variables y_j to be set in all possible ways. If any setting of these values kills all terms not yet killed by the variables x_i, this gadget will contain a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$.

In addition to these gadgets, digraph $\mathcal{D}$ has three auxiliary vertices a, a' and b. Vertices a and a' are connected by arcs (a, a') and (a', a). Vertex a also has some outgoing arcs that will be described later. Vertex b is connected by arcs to and from all other vertices of $\mathcal{D}$ except a and a'.

Next, we describe the gadgets (for now, we specify only their vertices and arcs — the preference posets will be defined later). The $\exists$-gadget corresponding to x_i is shown in Figure 9. It's constructed as follows:

— For $i = 1, ..., k$, create vertices x_i, $\bar{x}_i$, z_i and $\bar{z}_i$, with arcs (a, x_i), $(a, \bar{x}_i)$, (a, z_i), $(a, \bar{z}_i)$, $(x_i, \bar{x}_i)$, $(\bar{x}_i, x_i)$, (x_i, z_i), and $(\bar{x}_i, \bar{z}_i)$. Throughout the proof we will use notation $\tilde{z}_i$ for the vertex corresponding to $\tilde{x}_i$, that is $\tilde{z}_i = z_i$ if $\tilde{x}_i = x_i$, and $\tilde{z}_i = \bar{z}_i$ if $\tilde{x}_i = \bar{x}_i$.

The $\forall$-gadget is shown in Figure 10. It's constructed as follows:

— For $j = 0, ..., l$, create vertices q_j. For $j = 1, ..., l$, create vertices y_j and $\bar{y}_j$ and arcs (q_{j-1}, y_j), $(q_{j-1}, \bar{y}_j)$, (y_j, q_j),

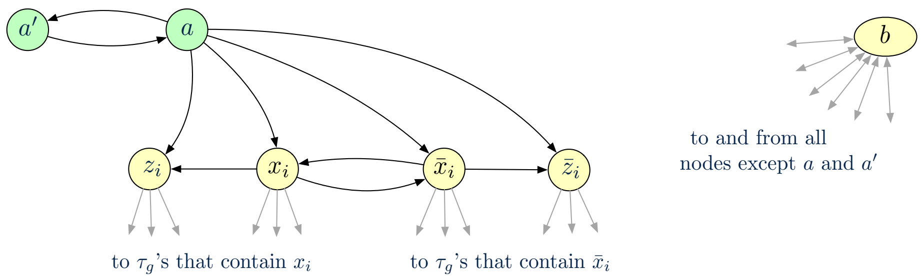

Figure 9. The construction of digraph $\mathcal{D}$ in the proof of Σ_2^{P}-hardness. This figure shows vertices a, a', b, and an $\exists$-gadget for variable x_i. The arcs to and from b are shown as bi-directional arrows at b.

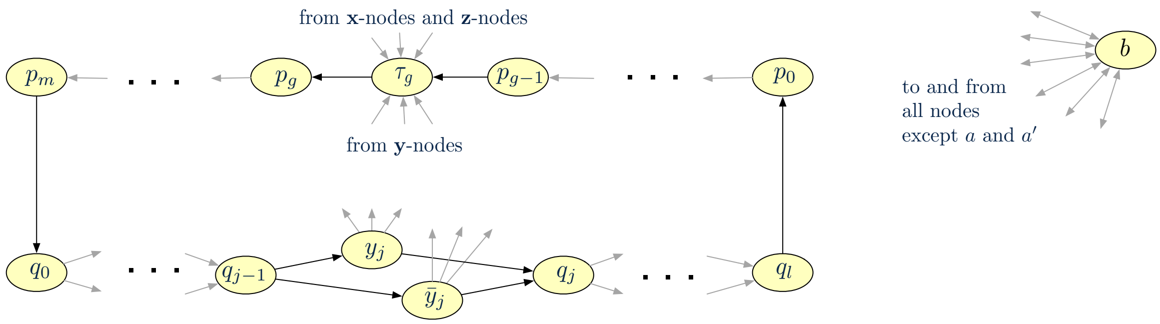

Figure 10. The construction of digraph $\mathcal{D}$ in the proof of Σ_2^{P}-hardness. This figures shows the $\forall$-gadget, namely the part of $\mathcal{D}$ that contains the vertices that simulate setting the values of the y_j-variables and the terms τ_g. The arcs to and from b are shown as bi-directional arrows at b.

and $(\bar{y}_j, q_j)$.

— For $g = 0, ..., m$, create vertices p_g. For $g = 1, ..., m$, create vertices τ_g and arcs (p_{g-1}, τ_g) and (τ_g, p_g).
— Create arcs (q_l, p_0) and (p_m, q_0).
— For each $g = 1, ..., m$, and for each literal $\tilde{y}_j$ in τ_g, create arc $(\tilde{y}_j, \tau_g)$.

To complete the construction of $\mathcal{D}$, we add arcs between $\exists$-gadgets and the $\forall$-gadget:

— For each $g = 1, ..., m$, if $\tilde{x}_i$ is the $\mathbf{x}$-literal in τ_g (there is exactly one, by the definition of $\exists\forall\mathsf{DNF}_{1x}$), create arcs $(\tilde{x}_i, \tau_g)$ and $(\tilde{z}_i, \tau_g)$.

Next, we need to define preference posets for all vertices. As explained in Section 2, all preference posets are specified by their list of generators. An outcome $\langle \omega^{in} \,|\, \omega^{out} \rangle$ of each vertex v is specified by lists ω^{in} and ω^{out} of its in-neighbors and out-neighbors, respectively. With this convention, the generators of all preference posets are:

— Vertices a, a', and b do not have any generators.
— The generators for the $\exists$-gadget corresponding to variable x_i are as follows. For each literal $\tilde{x}_i$, its generators are $\text{DEAL}_{\tilde{x}_i} \prec \langle b \,|\, b, \bar{\tilde{x}}_i, T(\tilde{x}_i) \rangle$ and $\text{DEAL}_{\tilde{x}_i} \prec \langle b, \bar{\tilde{x}}_i \,|\, b, \tilde{z}_i \rangle$, where $\bar{\tilde{x}}_i$ is the negation of $\tilde{x}_i$ and $T(\tilde{x}_i)$ is the set of terms that contain literal $\tilde{x}_i$. The generators of $\tilde{z}_i$ are $\text{DEAL}_{\tilde{z}_i} \prec \langle b \,|\, b \rangle$ and $\text{DEAL}_{\tilde{z}} \prec \langle b, \tilde{x}_i \,|\, b, T(\tilde{x}_i) \rangle$.
— For each literal $\tilde{y}_j$, its generators are $\text{DEAL}_{\tilde{y}_j} \prec \langle b \,|\, b \rangle$ and $\langle b \,|\, b \rangle \prec \langle q_{j-1} \,|\, q_j, T(\tilde{y}_j) \rangle$. The generators of q_j, where $j \notin \{0, l\}$, are $\text{DEAL}_{q_j} \prec \langle b \,|\, b \rangle$ and $\langle b \,|\, b \rangle \prec \langle \tilde{y}_j \,|\, \tilde{y}_{j+1} \rangle$, for all literals $\tilde{y}_j \in \{y_j, \bar{y}_j\}$ and $\tilde{y}_{j+1} \in \{y_{j+1}, \bar{y}_{j+1}\}$.
— The generators of q_0 are $\text{DEAL}_{q_0} \prec \langle b \,|\, b \rangle$ and $\langle b \,|\, b \rangle \prec \langle p_m \,|\, \tilde{y}_1 \rangle$, for all $\tilde{y}_1 \in \{y_1, \bar{y}_1\}$. The generators of q_l are $\text{DEAL}_{q_l} \prec \langle b \,|\, b \rangle$ and $\langle b \,|\, b \rangle \prec \langle \tilde{y}_l \,|\, p_0 \rangle$, for all $\tilde{y}_l \in \{y_l, \bar{y}_l\}$.
— For each term τ_g, letting $\tilde{x}_i$ be the unique $\mathbf{x}$-literal in τ_g, its generators are: $\text{DEAL}_{\tau_g} \prec \langle b, \tilde{x}_i \,|\, b \rangle$, $\langle b, \tilde{x}_i \,|\, b \rangle \prec \langle p_{g-1}, L \,|\, p_g \rangle$ for any subset L of the $\mathbf{y}$-literals in τ_g, $\text{DEAL}_{\tau_g} \prec \langle b, \tilde{z}_i \,|\, b \rangle$, and $\langle b, \tilde{z}_i \,|\, b \rangle \prec \langle p_{g-1}, L' \,|\, p_g \rangle$ for any *non-empty* subset L' of the $\mathbf{y}$-literals in τ_g. For each p_g, where $g \notin \{0, m\}$, its generators are $\text{DEAL}_{p_g} \prec \langle b \,|\, b \rangle$ and $\langle b \,|\, b \rangle \prec \langle \tau_g \,|\, \tau_{g+1} \rangle$.
— The generators of p_0 are $\text{DEAL}_{p_0} \prec \langle b \,|\, b \rangle$ and $\langle b \,|\, b \rangle \prec \langle q_l \,|\, \tau_1 \rangle$. The generators of p_m are $\text{DEAL}_{p_m} \prec \langle b \,|\, b \rangle$ and $\langle b \,|\, b \rangle \prec \langle \tau_m \,|\, q_0 \rangle$.

With this, the description of $\mathcal{S}$ is complete. The construction of $\mathcal{S}$ clearly takes time that is polynomial in the size of α. Applying Theorem 3, it remains to show that α is true if and only if $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ with properties (c.1)-(c.3).

The argument is based on several ideas. One, We design the preference posets of $\tilde{x}_i$'s so that $\mathcal{G}$ is forced to choose between two possible subsets of arcs within the $\exists$-gadget. The choice between these two subsets of arcs corresponds to choosing a truth assignment for variable x_i. We focus on the literals $\tilde{x}_i$ that are set to false, since these kill the terms where they appear. If $\tilde{x}_i$ is set to false, its arcs to the terms τ_g's in which the literal appears will be included in $\mathcal{G}$ (the first subset), otherwise its arc to $\tilde{z}_i$ will be included in $\mathcal{G}$ (the second subset).

Another idea is that vertices outside of the $\forall$-gadget have their preference posets defined in such a way that their arcs in $\mathcal{G}$ define an outcome that is already the best for them. Therefore, if a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$ does indeed exist, we know it must appear in the $\forall$-gadget. This leads into the key idea of the $\forall$-gadget. The vertices in this gadget can have outcomes that are better than their outcomes in $\mathcal{G}$. All the arcs in these better outcomes together form the cycle

$$\begin{aligned}\mathcal{C} \quad = \quad & q_0 \rightarrow \tilde{y}_1 \rightarrow ... \rightarrow \tilde{y}_l \rightarrow q_l \rightarrow \\ & p_0 \rightarrow \tau_1 \rightarrow ...\tau_m \rightarrow p_m \rightarrow q_0\end{aligned} \tag{2}$$

for some choice of the literals $\tilde{y}_1, ..., \tilde{y}_l$. We design the preference posets of each τ_g so that its outcome in $\mathcal{G}$ can only be improved (specifically, towards $\mathcal{C}$) only if it receives an arc from one of its literals — in other words, if it is killed by that literal. This way, $\mathcal{G}$ will have a strictly dominating subgraph $\mathcal{H}$ (namely cycle $\mathcal{C}$) only if all terms are killed, i.e. when α is false. The formal proof follows.

($\Rightarrow$) Suppose α is true. Fix some truth assignments $\mathbf{x} \mapsto \boldsymbol{\phi}$ for which $\forall \mathbf{y} \beta(\boldsymbol{\phi}, \mathbf{y})$ is true. This means that for each truth assignment $\mathbf{y} \mapsto \boldsymbol{\psi}$ the boolean expression $\beta(\boldsymbol{\phi}, \boldsymbol{\psi})$ is true. For each truth assignment $\mathbf{y} \mapsto \boldsymbol{\psi}$ we can thus choose an index $h(\boldsymbol{\psi})$ for which term $\tau_{h(\boldsymbol{\psi})}$ is true.

Using this assignment $\mathbf{x} \mapsto \boldsymbol{\phi}$, we construct a spanning subgraph $\mathcal{G}$ of $\mathcal{D}$ that satisfies the three conditions (c.1)-(c.3). $\mathcal{G}$ will contain all vertices from the above construction and all arcs that connect b to all other vertices except a and a', in both directions. Vertices a and a' will be connected by arcs (a, a') and (a', a). This makes $\mathcal{G}$ spanning and piece-wise strongly connected, with one strongly connected component consisting of vertices a and a' and the other consisting of all other vertices. So (c.1) holds.

Next, we define the arcs of $\mathcal{G}$ for the vertices in the $\exists$-gadgets. For any given i, if $\phi(x_i) = 1$, add to $\mathcal{G}$ the following arcs: (x_i, z_i), $(\bar{x}_i, x_i)$, all arcs (z_i, τ_j) for terms $\tau_j \in T(x_i)$, and all arcs $(\bar{x}_i, \tau_j)$ for terms $\tau_j \in T(\bar{x}_i)$. Symmetrically, if $\phi(x_i) = 0$, add to $\mathcal{G}$ the following arcs: $(\bar{x}_i, \bar{z}_i)$, $(x_i, \bar{x}_i)$, all arcs $(\bar{z}_i, \tau_j)$ for terms $\tau_j \in T(\bar{x}_i)$, and all arcs (x_i, τ_j) for terms $\tau_j \in T(x_i)$. (Note that we add the arcs from false literals to the terms that they kill, and from true literals to the corresponding nodes $\tilde{z}_i$.) We now need to verify conditions (c.2) and (c.3).

Condition (c.2) can be verified by routine inspection of all nodes. For each vertex v we need to check that $\text{DEAL}_v^{\mathcal{G}} \succeq \text{DEAL}_v^{\mathcal{D}}$. For $v \in \{a', b\}$, we have $\text{DEAL}_v^{\mathcal{G}} = \text{DEAL}_v^{\mathcal{D}}$. For $v = a$, $\text{DEAL}_a^{\mathcal{G}} = \langle a' \,|\, a' \rangle \succ \text{DEAL}_v^{\mathcal{D}}$. For $v = \tilde{x}_i$ there are two cases: either $\text{DEAL}_{\tilde{x}_i}^{\mathcal{G}} = \langle b, \bar{\tilde{x}}_i \,|\, b, \tilde{z}_i \rangle$ (if $\phi(\tilde{x}_i) = 1$) or $\text{DEAL}_{\tilde{x}_i}^{\mathcal{G}} = \langle b \,|\, b, \bar{\tilde{x}}_i, T(\tilde{x}_i) \rangle$ (if $\phi(\tilde{x}_i) = 0$); in both cases $\text{DEAL}_{\tilde{x}_i}^{\mathcal{G}} \succeq \text{DEAL}_{\tilde{x}_i}^{\mathcal{D}}$. For $v = \tilde{z}_i$, similarly, either $\text{DEAL}_{\tilde{z}_i}^{\mathcal{G}} = \langle b, \tilde{x}_i \,|\, b, T(\tilde{x}_i) \rangle$ (if $\phi(\tilde{x}_i) = 1$) or $\text{DEAL}_{\tilde{z}_i}^{\mathcal{G}} = \langle b \,|\, b \rangle$ (if $\phi(\tilde{x}_i) = 0$); in both cases $\text{DEAL}_{\tilde{z}_i}^{\mathcal{G}} \succeq \text{DEAL}_{\tilde{z}_i}^{\mathcal{D}}$. Finally, we examine the vertices in the $\forall$-gadget. If $v \in \{p_g\}_{g=0}^m \cup \{y_j, \bar{y}_j\}_{j=1}^l \cup \{q_j\}_{j=0}^l$ then $\text{DEAL}_v^{\mathcal{G}} = \langle b \,|\, b \rangle \succeq \text{DEAL}_v^{\mathcal{D}}$. Consider a vertex $v = \tau_g$, for some g, and let $\tilde{x}_i$ be the $\mathbf{x}$-literal in τ_g. If $\phi(\tilde{x}_i) = 1$ then $\text{DEAL}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{z}_i \,|\, b \rangle$, and if $\phi(\tilde{x}_i) = 0$ then $\text{DEAL}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{x}_i \,|\, b \rangle$. In both cases, $\text{DEAL}_{\tau_j}^{\mathcal{G}} \succeq \text{DEAL}_{\tau_j}^{\mathcal{D}}$.

It remains to verify condition (c.3). Let $\mathcal{H}$ be a subgraph of $\mathcal{D}$, and suppose that $\mathcal{H}$ dominates $\mathcal{G}$, that is $\text{DEAL}_v^{\mathcal{H}} \succeq \text{DEAL}_v^{\mathcal{G}}$ for all vertices v in $\mathcal{H}$. We will show that is possible only if $\mathcal{H}$ is either equal to $\mathcal{G}$ or to one of the two strongly connected components of $\mathcal{G}$.

$\mathcal{H}$ cannot contain any arcs from a to literals $\tilde{x}_i$, because then it would not dominate $\mathcal{G}$ at vertex a. There is also no subgraph consisting of a and a' that strictly dominates $\mathcal{G}$. We can thus assume that $\mathcal{H}$ is a subgraph of $\mathcal{D}' = \mathcal{D} \setminus \{a, a'\}$. Let also $\mathcal{G}' = \mathcal{G} \setminus \{a, a'\}$. The rest of the argument is divided into two cases, depending on whether $\mathcal{H}$ includes vertex b or not.

Suppose first that $\mathcal{H}$ includes vertex b. In this case, we claim that $\mathcal{H} = \mathcal{G}'$, and therefore $\mathcal{H}$ does not strictly dominate $\mathcal{G}$. To show this, observe first that since $\text{DEAL}_b^{\mathcal{H}} \succeq \text{DEAL}_b^{\mathcal{G}}$, $\mathcal{H}$ must contain all incoming arcs of b. So $\mathcal{H}$ must in fact contain all vertices of $\mathcal{D}'$. And each vertex $v \in \mathcal{D}' \setminus \{b\}$ does not have any outcome better than $\text{DEAL}_v^{\mathcal{G}}$ that does not have arc (b, v). Therefore $\mathcal{H}$ must also contain all outgoing arcs of b.

The idea now is to show that for each vertex $v \in \mathcal{D}' \setminus \{b\}$, the outcome of v in $\mathcal{G}'$ is already best possible among the outcomes that have incoming and outgoing arcs from b. A more formal argument actually focuses on arcs rather than vertices, and involves two observations: (i) For each arc $(u, v) \in \mathcal{G}'$, vertex v does not have any outcome that does not include incoming arc (u, v) and is better than $\text{DEAL}_v^{\mathcal{G}}$. (ii) For each arc $(u, v) \in \mathcal{D}' \setminus \mathcal{G}'$, vertex u does not have any outcome that includes outgoing arc (u, v) and is better than $\text{DEAL}_u^{\mathcal{G}}$. These observations imply that $\text{DEAL}_v^{\mathcal{H}} \succeq \text{DEAL}_v^{\mathcal{G}}$ for all $v \in \mathcal{D}'$, implying in turn that $\mathcal{H} = \mathcal{G}'$, as claimed.

Both observations (i) and (ii) can be established through routine although a bit tedious inspection of all arcs in $\mathcal{D}'$. (The process here is the same as in the NP-hardness proof in Section 10.)

We start with the vertices in the $\exists$-gadgets. Consider some $\tilde{x}_i$ and suppose $\phi(\tilde{x}_i) = 1$ (symmetric for when $\phi(\tilde{x}_i) = 0$). There is no outcome of $\tilde{x}_i$ better than $\text{DEAL}_{\tilde{x}_i}^{\mathcal{G}} = \langle b, \bar{\tilde{x}}_i \,|\, b, \tilde{z}_i \rangle$ that does not include the incoming arc $(\bar{\tilde{x}}_i, \tilde{x}_i)$. Also, there is no better outcome that includes arc $(\tilde{x}_i, \tau_j)$, for each term $\tau_j \in T(\tilde{x}_i)$. For $\bar{\tilde{x}}_i, \text{DEAL}_{\bar{\tilde{x}}_i}^{\mathcal{G}} = \langle b \,|\, b, \tilde{x}_i, T(\bar{\tilde{x}}_i) \rangle$. There is no outcome of $\bar{\tilde{x}}_i$ better than $\text{DEAL}_{\bar{\tilde{x}}_i}^{\mathcal{G}}$ that includes arc $(\bar{\tilde{x}}_i, \bar{\tilde{z}}_i)$. For a vertex $\tilde{z}_i$ (still assuming that $\phi(\tilde{x}_i) = 1$), $\text{DEAL}_{\tilde{z}_i}^{\mathcal{G}} = \text{DEAL}_{\tilde{z}_i}^{\mathcal{D}} = \langle b, \tilde{x}_i \,|\, b, T(\tilde{x}_i) \rangle$; there is no better outcome that does not include $(\tilde{x}_i, \tilde{z}_i)$. Lastly, there is no outcome of $\bar{\tilde{z}}_i$ better than $\text{DEAL}_{\bar{\tilde{z}}_i}^{\mathcal{G}} = \langle b \,|\, b \rangle$.

We move on to the vertices in the $\forall$-gadget. For any vertex $v \in \{p_g\}_{g=0}^m \cup \{\tilde{y}_j\}_{j=1}^l \cup \{q_j\}_{j=0}^l$ we have $\text{DEAL}_v^{\mathcal{G}} = \langle b \,|\, b \rangle$ and, by the earlier argument, $\mathcal{H}$ contains arc (v, b). But this v does not have any outcome with outgoing arc (v, b) that is better than $\text{DEAL}_v^{\mathcal{G}}$. The argument when $v = \tau_g$, for some g, is similar. If the unique $\mathbf{x}$-literal in τ_g is $\tilde{x}_i$, then $\text{DEAL}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{z}_i \,|\, b \rangle$ (if $\phi(\tilde{x}_i) = 1$) or $\text{DEAL}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{x}_i \,|\, b \rangle$ (if $\phi(\tilde{x}_i) = 0$). In either case, as before, there is no outcome better than $\textsc{Deal}^{\mathcal{G}}_{\tau_g}$ among the outcomes of τ_g that contain an outgoing arc to b.

Next, we consider the case when $\mathcal{H}$ does not include vertex b. First, we observe that $\mathcal{H}$ cannot contain any vertices in the $\exists$-gadgets (namely vertices $\tilde{x}_i$ and $\tilde{z}_i$). This is because for these vertices v there is no outcome that is better than $\textsc{Deal}^{\mathcal{G}}_{v}$ and does not include the incoming arc from b.

We can thus assume that $\mathcal{H}$ is a subgraph of the $\forall$-gadget. (This is actually the most crucial case.) Let $\mathcal{D}''$ be the subgraph of $\mathcal{D}$ induced by the vertices in the $\forall$-gadget. Observe that every vertex v in $\mathcal{D}''$ has at least one outcome better than $\textsc{Deal}^{\mathcal{G}}_{v}$ that does not include arcs to and from b, so now we need a more subtle argument than the one we used earlier. For $v = \tau_g$, there are two cases. The first is when τ_g has an incoming arc from its unique $\mathbf{x}$-literal $\tilde{x}_i$ (which means $\phi(\tilde{x}_i) = 0$), in which case $\textsc{Deal}^{\mathcal{G}}_{\tau_g} = \langle b, \tilde{x}_i \,|\, b\rangle$. By the preference poset of τ_g, τ_g can improve this outcome by switching to $\langle p_{g-1}, L \,|\, p_g\rangle$, for any set L of the $\mathbf{y}$-literals in τ_g. That is, this τ_g can improve its outcome regardless of whether it receives any arcs from its $\mathbf{y}$-literals. The second case is when τ_g does not have an incoming arc from its $\mathbf{x}$-literal $\tilde{x}_i$ (which means $\phi(\tilde{x}_i) = 1$), in which case $\textsc{Deal}^{\mathcal{G}}_{\tau_g} = \langle b, \tilde{z}_i \,|\, b\rangle$. By the preference poset of τ_g, τ_g can improve its outcome by switching to $\langle p_{g-1}, L' \,|\, p_g\rangle$ for any non-empty subset L' of the $\mathbf{y}$-literals in τ_g. That is, this τ_g can improve its outcome only if it receives an arc from at least one of its $\mathbf{y}$-literals. For $v = \tilde{y}_j$, $\textsc{Deal}^{\mathcal{G}}_{\tilde{y}_j} = \langle b \,|\, b\rangle$. By the preference poset of $\tilde{y}_j$, $\tilde{y}_j$ can improve its outcome by switching to $\langle q_{j-1} \,|\, q_j, T(\tilde{y}_j)\rangle$, which results in creating arcs to the terms in $T(\tilde{y}_j)$. For $v = q_j$, $\textsc{Deal}^{\mathcal{G}}_{q_j} = \langle b \,|\, b\rangle$. By the preference poset of q_j, where $j \notin \{0, l\}$, the following outcomes of q_j are better than $\textsc{Deal}^{\mathcal{G}}_{q_j}$: $\langle y_j \,|\, y_{j+1}\rangle$, $\langle \bar{y}_j \,|\, y_{j+1}\rangle$, $\langle y_j \,|\, \bar{y}_{j+1}\rangle$ or $\langle \bar{y}_j \,|\, \bar{y}_{j+1}\rangle$. This means the preference posets of q_{j-1} and q_j allow only one of y_j or $\bar{y}_j$ to make the switch described above. (This corresponds to choosing which of these two literals is false.) The same reasoning holds for q_0 and q_l, except their improved outcomes are $\langle p_m \,|\, \tilde{y}_1\rangle$ and $\langle \tilde{y}_l \,|\, p_0\rangle$ respectively. For $v = p_g$, $\textsc{Deal}^{\mathcal{G}}_{p_g} = \langle b \,|\, b\rangle$. By the preference poset of p_g, where $g \notin \{0, m\}$, p_g can improve its outcome by switching to $\langle \tau_g \,|\, \tau_{g+1}\rangle$. This means p_g can only switch given that τ_g makes one of switches described above (either from $\langle b, \tilde{x}_i \,|\, b\rangle$ to $\langle p_{g-1}, L \,|\, p_g\rangle$ or from $\langle b, \tilde{z}_i \,|\, b\rangle$ to $\langle p_{g-1}, L' \,|\, p_g\rangle$). The same reasoning holds for p_0 and p_m, except their improved outcomes are $\langle q_l \,|\, \tau_1\rangle$ and $\langle \tau_m \,|\, q_0\rangle$ respectively.

Importantly, the outcome improvements in the above paragraph are possible only if *all the vertices in $\mathcal{D}''$ together switch their outcomes as described in the above paragraph*. This would correspond to choosing a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$ (namely the cycle given in (2)). We now show this subgraph $\mathcal{H}$ cannot exist, by way of contradiction. Suppose such a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$ does exist. Since $\mathcal{H}$ strictly dominates $\mathcal{G}$, and all vertices must improve together, we know every vertex $v \in \mathcal{H}$ strictly improves their outcome from $\textsc{Deal}^{\mathcal{G}}_{v}$. We focus on the outcome improvements made by the term vertices $\tau_1 ... \tau_m$. Let us fix some term vertex τ_g and let $\tilde{x}_i$ be the unique $\mathbf{x}$-literal of τ_g.

As described above, τ_g can improves its outcome in one of two ways, depending on $\textsc{Deal}^{\mathcal{G}}_{\tau_g}$; specifically whether or not $(\tilde{x}_i, \tau_g) \in \mathcal{G}$. If $(\tilde{x}_i, \tau_g) \in \mathcal{G}$, then τ_g can improve its outcome from $\textsc{Deal}^{\mathcal{G}}_{\tau_g}$ by simply "switching". Otherwise, if $(\tilde{x}_i, \tau_g) \notin \mathcal{G}$, then τ_g can only switch to an improved outcome if it receives an arc from any of its $\mathbf{y}$-literals in $\mathcal{H}$. In other words, each τ_g must have either received its incoming arc from its $\mathbf{x}$-literal in $\mathcal{G}$ or received an incoming arc from any of its $\mathbf{y}$-literals in $\mathcal{H}$.

Recall though that τ_g receives an arc from one of its literals only if that literal is set to false. This implies that each term τ_g is killed, either by its $\mathbf{x}$-literal or one of its $\mathbf{y}$-literals, depending on how it improves its outcome. However, if each term is killed under the assignments $\mathbf{x} \mapsto \boldsymbol{\phi}$ and $\mathbf{y} \mapsto \boldsymbol{\psi}$, we know $\beta(\boldsymbol{\phi}, \boldsymbol{\psi})$ is false, contradicting our original assumption.

We show this more formally, starting with the terms being killed by the assignment of the $\mathbf{x}$ variables. In graph $\mathcal{G}$, for each variable x_i, if $\phi(x_i) = 1$, then for each term τ_g that contains $\bar{x}_i$, $(\bar{x}_i, \tau_g) \in \mathcal{G}$. On the other hand, if $\phi(x_i) = 0$, then for each term τ_g that contains x_i, $(x_i, \tau_g) \in \mathcal{G}$. In both cases, τ_g is killed. Within the swap system, this is signified by vertex τ_g's preference to switch from $\textsc{Deal}^{\mathcal{G}}_{\tau_g}$ to $\langle p_{g-1}, L \,|\, p_g\rangle$.

Now we address the terms survived by the assignment $\mathbf{x} \mapsto \boldsymbol{\phi}$. The surviving term vertices are those that did not receive their incoming arcs from their $\mathbf{x}$-literals in $\mathcal{G}$. Since we know each surviving term vertex τ_g strictly improves their outcome in $\mathcal{H}$, the only remaining option is that each τ_g has an incoming arc from one of their $\mathbf{y}$-literals in $\mathcal{H}$.

We use this to construct the assignment $\mathbf{y} \mapsto \psi$ so that $\beta(\boldsymbol{\phi}, \boldsymbol{\psi})$ is false. This is quite simple: for each $\mathbf{y}$-literal $\tilde{y}_j$ that has an outgoing arc to a surviving term vertex in $\mathcal{H}$, we assign $\psi(\tilde{y}_j) = 0$. We know that $\boldsymbol{\psi}$ must be a consistent assignment, i.e. it cannot be the case that $\tilde{y}_j$ and $\bar{\tilde{y}}_j$ are both assigned to true/false. This is because only either $\tilde{y}_j$ or $\bar{\tilde{y}}_j$ are in $\mathcal{H}$, by design of the preference posets of vertices q_{j-1} and q_j. Thus, since we can construct a consistent assignment $\mathbf{y} \mapsto \boldsymbol{\psi}$, given the assignment $\mathbf{x} \mapsto \boldsymbol{\phi}$, so that every term is killed, we know that $\beta(\boldsymbol{\phi}, \boldsymbol{\psi})$ is false, contradicting our original assumption.

($\Leftarrow$) Assume now that $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ that satisfies properties (c.1) and (c.2). From $\mathcal{G}$ we will construct an assignment $\boldsymbol{\phi}$ for the $\mathbf{x}$-variables that makes $\forall \mathbf{y} \beta(\boldsymbol{\phi}, \mathbf{y})$ true. Condition (c.1) implies that $\mathcal{G}$ cannot have any arcs $(a, \tilde{x}_i)$ nor $(a, \tilde{z}_i)$, so vertices $\{a, a'\}$ will form one strongly connected component of $\mathcal{G}$. As before, let $\mathcal{D}' = \mathcal{D} \setminus \{a, a'\}$ and $\mathcal{G}' = \mathcal{G} \setminus \{a, a'\}$. We focus on $\mathcal{G}'$.

We first argue that $\mathcal{G}'$ is in fact strongly connected and it contains b. This is quite simple. Condition (c.2) states that the outcome of b in $\mathcal{G}$ is at least as good as its outcome in $\mathcal{D}$, so $\mathcal{G}'$ must contain all incoming arcs of b. On the other hand, each vertex $v \in \mathcal{G}' \setminus \{b\}$ does not have an outcome better than $\textsc{Deal}^{\mathcal{D}}_{v}$ that includes outgoing arc (v, b) but does not include incoming arc (b, v). Thus, $\mathcal{G}'$ must also contain all outgoing arcs of b, which is already sufficient to make $\mathcal{G}'$ strongly connected.

For each literal vertex $\tilde{x}_i$, we will refer to any outcome that contains $T(\tilde{x}_i)$ in its set of outgoing arcs as a 0-outcome of $\tilde{x}_i$, and to the exact outcome $\langle b, \bar{\tilde{x}}_i \mid b, \tilde{z}_i \rangle$ as the 1-outcome $\tilde{x}_i$. We start with the following claim:

Claim 1: *For each i and each literal $\tilde{x}_i \in \{x_i, \bar{x}_i\}$, outcome* $\text{DEAL}^{\mathcal{G}}_{\tilde{x}_i}$ *is either a 0-outcome or the 1-outcome of $\tilde{x}_i$. Further, for at least one of x_i and $\bar{x}_i$ this outcome is a 0-outcome.*

Proof. Let us fix a single $\exists$-gadget. We first show that for literal $\tilde{x}_i \in \{x_i, \bar{x}_i\}$, the outcome $\text{DEAL}^{\mathcal{G}}_{\tilde{x}_i}$ is either a 0-outcome or the 1-outcome of $\tilde{x}_i$. Firstly, we know the incoming and outgoing arcs between $\tilde{x}_i$ and vertex b are included in $\mathcal{G}'$. Next, consider any term vertex τ_g in which term τ_g contains literal $\tilde{x}_i$. If we examine the generators of vertex τ_g, limiting ourselves only to the outcomes that include the arcs to and from vertex b, we see that τ_g must receive either an arc from $\tilde{x}_i$ or $\tilde{z}_i$ in order to satisfy condition (c.2).

We now have two cases: when τ_g receives an arc from $\tilde{x}_i$ and when τ_g receives an arc from $\tilde{z}_i$. We start with the latter case. If τ_g receives arc $(\tilde{z}_i, \tau_g)$, then by $\tilde{z}_i$'s generators, we know that $\tilde{z}_i$ must have received arc $(\tilde{x}_i, \tilde{z}_i)$. This then implies that $\tilde{x}_i$ received arc $(\bar{\tilde{x}}_i, \tilde{x}_i)$. At this point, $\tilde{x}_i$ is exactly in the 1-outcome. We reason similarly about $\bar{\tilde{x}}_i$: starting from some vertex τ_g for which term τ_g contains $\bar{\tilde{x}}_i$, we know that τ_g must receive either an arc from $\bar{\tilde{x}}_i$ or $\bar{\tilde{z}}_i$. We know τ_g cannot receive an arc from $\bar{\tilde{z}}_i$ because for $\bar{\tilde{z}}_i$ to pay arc $(\bar{\tilde{z}}_i, \tau_g)$, it must receive arc $(\bar{\tilde{x}}_i, \bar{\tilde{z}}_i)$. However, there is no outcome for $\bar{\tilde{x}}_i$ that satisfies condition (c.2) in which $\bar{\tilde{x}}_i$ pays both arcs $(\bar{\tilde{x}}_i, \tilde{x}_i)$ and $(\bar{\tilde{x}}_i, \bar{\tilde{z}}_i)$. Thus, we can conclude that $\bar{\tilde{x}}_i$ is the one to pay τ_g. We can reason about each $\tau_g \in T(\bar{\tilde{x}}_i)$ in the same manner, implying that $\bar{\tilde{x}}_i$ in fact pays every $\tau_g \in T(\bar{\tilde{x}}_i)$. This allows us to conclude that $\bar{\tilde{x}}_i$ is in a 0-outcome.

We move on to the former case, when τ_g receives an arc from $\tilde{x}_i$. It is easy to see that if $\tilde{x}_i$ pays any term vertex $\tau_g \in T(\tilde{x}_i)$, it must pay all term vertices in $T(\tilde{x}_i)$. This is because each term vertex $\tau_g \in T(\tilde{x}_i)$ must receive an arc from either $\tilde{x}_i$ or $\tilde{z}_i$, as previously stated. However, there is no outcome for $\tilde{x}_i$ that satisfies condition (c.2) in which $\tilde{x}_i$ pays τ_g and $\tilde{z}_i$, thus $\tilde{x}_i$ is responsible for paying all term vertices $\tau_g \in T(\tilde{x}_i)$. This is sufficient to show that $\tilde{x}_i$ is in a 0-outcome. We move onto vertex $\bar{\tilde{x}}_i$. Unlike the previous case, the outcome of $\bar{\tilde{x}}_i$ is not directly influenced by the outcome of $\tilde{x}_i$. When we consider some term vertex $\tau_g \in T(\bar{\tilde{x}}_i)$, it is possible for τ_g to receive an arc from either $\bar{\tilde{x}}_i$ or $\bar{\tilde{z}}_i$. We show that $\bar{\tilde{x}}_i$ ends in a 0-outcome or the 1-outcome, respectively. The first possibility is that τ_g receives arc $(\bar{\tilde{x}}_i, \tau_g)$. We apply the same reasoning as we did for $\tilde{x}_i$: if any $\tau_g \in T(\bar{\tilde{x}}_i)$ receives its arc from $\bar{\tilde{x}}_i$, then every $\tau_g \in T(\bar{\tilde{x}}_i)$ also receives its arc from $\bar{\tilde{x}}_i$. This is again sufficient to show that $\bar{\tilde{x}}_i$ is in a 0-outcome. The second possibility is that τ_g receives arc $(\bar{\tilde{z}}_i, \tau_g)$. For $\bar{\tilde{z}}_i$ to pay this arc, it must receive arc $(\bar{\tilde{x}}_i, \bar{\tilde{z}}_i)$. For $\bar{\tilde{x}}_i$ to pay this arc, it must receive arc $(\tilde{x}_i, \bar{\tilde{x}}_i)$. However, this is exactly the 1-outcome for $\bar{\tilde{x}}_i$. We note that this requires $\tilde{x}_i$ to pay arc $(\tilde{x}_i, \bar{\tilde{x}}_i)$, changing the outcome of $\tilde{x}_i$. Importantly though, $\tilde{x}_i$ remains in a 0-outcome and still satisfies condition (c.2) as $\text{DEAL}_{\tilde{x}_i} \prec \langle b \mid b, \bar{\tilde{x}}_i, T(\tilde{x}_i) \rangle$

It is easy to see that these two cases are exhaustive by inspection of the preference posets of τ_g. With this, we have shown both parts of claim (1): firstly, for each i, $\tilde{x}_i$ and $\bar{\tilde{x}}_i$ are either in a 0-outcome or the 1-outcome, and secondly, at least one of $\tilde{x}_i$ or $\bar{\tilde{x}}_i$ are in a 0-outcome, regardless of which case. $\square$

For convenience, we now introduce the concept of a *pseudo-truth assignment*. A pseudo-truth assignment is an assignment $\boldsymbol{\xi}$ of boolean values to the $\mathbf{x}$-literals (not just variables) such that for each variable x_i at most one of $\boldsymbol{\xi}(x_i)$ and $\boldsymbol{\xi}(\bar{x}_i)$ is 1. The value of $\forall \mathbf{y} \beta(\boldsymbol{\xi}, \mathbf{y})$, for such a pseudo-truth assignment $\boldsymbol{\xi}$, can be computed just like for standard truth assignments. If α has a satisfying pseudo-truth assignment $\boldsymbol{\xi}$ then it also has a satisfying standard truth assignment $\boldsymbol{\phi}$: simply let $\boldsymbol{\phi}(x_i) = \boldsymbol{\xi}(x_i)$ for all i. This works because if a term τ_g of β is not killed by $\boldsymbol{\xi}$ then it is also not killed by $\boldsymbol{\phi}$.

Thus it suffices to show how we can convert $\mathcal{G}$ into a pseudo-truth assignment $\boldsymbol{\xi}$ for the $\mathbf{x}$-variables that satisfies α. We define $\boldsymbol{\xi}$ as follows: for each i, if $\text{DEAL}^{\mathcal{G}}_{\tilde{x}_i}$ is a 0-outcome then $\boldsymbol{\xi}(\tilde{x}_i) = 0$, and if $\text{DEAL}^{\mathcal{G}}_{\tilde{x}_i}$ is the 1-outcome then $\boldsymbol{\xi}(\tilde{x}_i) = 1$.

Claim 2: *$\boldsymbol{\xi}$ is a satisfying pseudo-truth assignment for the $\mathbf{x}$-variables that satisfies α.*

Proof. We begin by supposing the pseudo-truth assignment $\boldsymbol{\xi}$ is not a satisfying assignment for α, towards contradiction. This would mean that $\forall y \beta(\boldsymbol{\xi}, \mathbf{y})$ is false. We fix an assignment of the $\mathbf{y}$-variables $\boldsymbol{\psi}$ such that $\beta(\boldsymbol{\xi}, \boldsymbol{\psi})$ is false. The idea is to now take $\boldsymbol{\psi}$ and construct a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$, contradicting our original assumption. Actually, $\mathcal{H}$ will be a subgraph of the $\forall$-gadget of the form given in (2), as before.

We now construct $\mathcal{H}$ as follows: add all vertices $v \in \{p_g\}_{g=0}^m \cup \{q_j\}_{j=0}^l \cup \{\tau_g\}_{g=1}^m$ to $\mathcal{H}$. For each j, if $\boldsymbol{\psi}(y_j) = 1$, add $\bar{y}_j$, otherwise, if $\boldsymbol{\psi}(y_j) = 0$, add y_j (we include the literal that is false). Now that we have all the vertices, we must define the arcs. Again, $\mathcal{H}$ will have the form of the cycle given in (2). For each $\tilde{y}_j \in \mathcal{H}$, add arcs $(q_{j-1}, \tilde{y}_j)$ and $(\tilde{y}_j, q_j)$. Add arcs (q_l, p_0) and (p_m, q_0). For each $\tau_g \in \mathcal{H}$, add arcs (p_{g-1}, τ_g) and (τ_g, p_g). Lastly, for each $\tilde{y}_j \in \mathcal{H}$, add arcs $(\tilde{y}_j, \tau_g)$ for $\tau_g \in T(\tilde{y}_j)$.

The next step is to show that $\mathcal{H}$ indeed strictly dominates $\mathcal{G}$. It is easy to see that for vertices $v \in \{p_g\}_{g=0}^m \cup \{\tilde{y}_j\}_{j=1}^l \cup \{q_j\}_{j=0}^l$, $\text{DEAL}^{\mathcal{G}}_v \prec \text{DEAL}^{\mathcal{H}}_v$ holds by simple inspection of each vertex's preference poset. Thus, we focus on the term vertices $\tau_1, .., \tau_m$. For each term vertex τ_g, outcome $\text{DEAL}^{\mathcal{H}}_{\tau_g}$ is an improvement in comparison to $\text{DEAL}^{\mathcal{G}}_{\tau_g}$ only if (at least) one of the two following conditions are satisfied: (1) τ_g received its incoming arc from its $\mathbf{x}$-literal in $\mathcal{G}$, or (2) τ_g receives an incoming arc from any of its $\mathbf{y}$-literals in $\mathcal{H}$.

We claim that one of these two conditions holds for every term τ_g. Suppose this is not true, towards contradiction, and there is a term vertex τ_g that does not satisfy either condition. Specifically, τ_g does not receive its incoming arc from its $\mathbf{x}$-literal in $\mathcal{G}$, nor does τ_g receive any of its incoming arcs from any of its $\mathbf{y}$-literals in $\mathcal{H}$. If this were the case, then τ_g is actually true, contradicting the fact that $\beta(\boldsymbol{\xi}, \boldsymbol{\psi})$ is false. Let $\tilde{x}_i$ be the $\mathbf{x}$-literal of τ_g. If $(\tilde{x}_i, \tau_g) \notin \mathcal{G}$, then $\text{DEAL}^{\mathcal{G}}_{\tilde{x}_i}$ is actually the 1-outcome for $\tilde{x}_i$. This implies that $\boldsymbol{\xi}(\tilde{x}_i) = 1$. Since τ_g does not satisfy the second condition, we know it does not receive a single arc from any of its $\mathbf{y}$-literals. However, recall how we used $\boldsymbol{\psi}$ to construct $\mathcal{H}$; a $\mathbf{y}$-literal is added to $\mathcal{H}$ only if that literal is *false* in $\boldsymbol{\psi}$. This means each of these $\mathbf{y}$-literals of τ_g are actually *true* in the original assignment of $\boldsymbol{\psi}$. This implies that the term τ_g is actually true, contradicting $\beta(\boldsymbol{\xi}, \boldsymbol{\psi})$ being false.

This contradiction gives us the fact that every term vertex τ_g indeed improves their outcome from $\text{DEAL}^{\mathcal{G}}_{\tau_g}$. With this, we have proven every vertex $v \in \mathcal{H}$ improves their outcome from $\text{DEAL}^{\mathcal{G}}_{v}$, meaning $\mathcal{H}$ strictly dominates $\mathcal{G}$. However, the existence of such an $\mathcal{H}$ contradicts our condition (c.3), implying claim (2), that the pseudo-truth assignment $\boldsymbol{\xi}$ is indeed a satisfying assignment of the $\mathbf{x}$-variables for α.

$\square$

With the truth assignment ϕ defined, we need to show that the non-existence of an $\mathcal{H}$ that strictly dominates $\mathcal{G}$ implies that the expression $\forall \mathbf{y} \beta(\boldsymbol{\phi}, \mathbf{y})$ is true. For this, it's easier to show the contrapositive, namely if there existed some assignment $\boldsymbol{\psi}$ for the $\mathbf{y}$-variables for which $\forall \mathbf{y} \beta(\boldsymbol{\phi}, \boldsymbol{\psi})$ is false, we could convert $\boldsymbol{\psi}$ into a subgraph $\mathcal{H}$ that strictly dominates $\mathcal{G}$.

We simply employ the exact same argument we saw in the proof for claim (2). We convert the assignment $\boldsymbol{\psi}$ in the exact same manner: for each y_j, if $\boldsymbol{\psi}(y_j) = 1$, add $\bar{y}_j$ to $\mathcal{H}$, otherwise, if $\boldsymbol{\psi}(y_j) = 0$, add y_j. The remainder of $\mathcal{H}$ is constructed in the exact same way as previously described. Likewise, the proof that $\mathcal{H}$ indeed strictly dominates $\mathcal{G}$ is the same. Since this contradicts condition (c.3), we know that the expression $\forall y \beta(\phi, y)$ is in fact true.

$\square$

10. Another Proof of NP-Hardness

In this section we give a proof of NP-hardness of SwapAtomic that is simpler than the one in Section 5.

Theorem 7. SwapAtomic *is* NP-*hard. It remains* NP-*hard even for strongly connected digraphs.*

Proof. The proof is by showing a polynomial-time reduction from CNF. Recall that in CNF we are given a boolean expression α in conjunctive normal form, and the objective is to determine whether there is a truth assignment that satisfies α. In our reduction we convert α into a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$ such that α is satisfiable if and only if $\mathcal{S}$ has an atomic swap protocol.

Let $x_1, x_2, ..., x_n$ be the variables in α. The negation of x_i is denoted $\bar{x}_i$. We will use notation $\tilde{x}_i$ for an unspecified literal of variable x_i, that is $\tilde{x}_i \in \{x_i, \bar{x}_i\}$. Let $\alpha = c_1 \vee c_2 \vee ... \vee c_m$, where each c_j is a clause. Without loss of generality we assume that each literal appears in at least one clause and that in each clause no two literals are equal or are negations of each other.

We first describe a reduction that uses a digraph $\mathcal{D}$ that is not strongly connected. Later we will show how to modify our construction to make $\mathcal{D}$ strongly connected. Digraph $\mathcal{D}$ is constructed as follows (see Figure 11) :

- — For $i = 1, ..., n$, create vertices x_i and $\bar{x}_i$, connected by arcs $(x_i, \bar{x}_i)$ and $(\bar{x}_i, x_i)$.
- — Create two vertices a, a' with arcs (a, a'), (a', a), and (a, x_i), $(a, \bar{x}_i)$ for all $i = 1, ..., n$.
- — For $j = 1, ..., m$, create vertices c_j. For each clause c_j and each literal $\tilde{x}_i$ in c_j, create arc $(\tilde{x}_i, c_j)$.
- — Create three vertices d, d', d'' with arcs (d, d'), (d', d), (d, d''), (d'', d), (d', d'') and (d'', d'). Create also arcs (c_j, d) for all $j = 1, ..., m$.
- — Create vertex b, with arcs (c_j, b) for all $j = 1, ..., m$ and (b, x_i), $(b, \bar{x}_i)$ for all $i = 1, ..., n$.

Next, we describe the preference posets $\mathcal{P}_v$, for each vertex v in $\mathcal{D}$. As explained in Section 2, an outcome $\langle \omega^{in} \,|\, \omega^{out} \rangle$ of a vertex v is specified by lists ω^{in} and ω^{out} of its in-neighbors and out-neighbors. The preference posets of the vertices in $\mathcal{D}$ are specified by their generators:

- — Vertices a,a', and b do not have any generators.
- — For each literal $\tilde{x}_i$, its generators are $\text{DEAL}_{\tilde{x}_i} \prec \langle b, \bar{\tilde{x}}_i \,|\, C(\tilde{x}_i) \rangle$ and $\text{DEAL}_{\tilde{x}_i} \prec \langle b \,|\, \bar{\tilde{x}}_i \rangle$, where $\bar{\tilde{x}}_i$ is the negation of $\tilde{x}_i$ and $C(\tilde{x}_i)$ is the set of clauses that contain literal $\tilde{x}_i$.
- — For each j, the generators of c_j are $\text{DEAL}_{c_j} \prec \langle \tilde{x}_i \,|\, b \rangle$ for each literal $\tilde{x}_i$ in c_j.
- — Vertices d, d', d'' have one generator each: $\text{DEAL}_d \prec \langle d'' \,|\, d' \rangle$, $\text{DEAL}_{d'} \prec \langle d \,|\, d'' \rangle$, $\text{DEAL}_{d''} \prec \langle d' \,|\, d \rangle$.

The construction of $\mathcal{S}$ clearly takes time that is polynomial in the size of α.

Applying Theorem 3, it remains to show that α is satisfiable if and only if $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ with the following properties: (c.1) $\mathcal{G}$ is piece-wise strongly connected and has no isolated vertices, (c.2) $\mathcal{G}$ dominates $\mathcal{D}$, and (c.3) no subgraph $\mathcal{H}$ of $\mathcal{D}$ strictly dominates $\mathcal{G}$.

($\Rightarrow$) Suppose that α is satisfiable, and fix some satisfying assignment for α. Using this assignment, we construct a spanning subgraph $\mathcal{G}$ of $\mathcal{D}$ that satisfies conditions (c.1)-(c.3).

Digraph $\mathcal{G}$ will contain all vertices of $\mathcal{D}$. For vertices a and a' it will include arcs (a, a') and (a', a). For vertex b, it will include all arcs (b, x_i), $(b, \bar{x}_i)$ and all arcs (c_j, b). Vertices d, d', d'' are connected by arcs (d, d'), (d', d'') and (d'', d). The remaining arcs are determined based on the satisfying assignment. Suppose that literal $\tilde{x}_i$ is true. Then $\mathcal{G}$ includes the arcs: $(\bar{\tilde{x}}_i, \tilde{x}_i)$ and $(\tilde{x}_i, c_j)$ for all clauses c_j that contain literal $\tilde{x}_i$. (Intuitively, the truth assignment corresponds to the direction of the arc between x_i and $\bar{x}_i$ in $\mathcal{G}$.)

Digraph $\mathcal{G}$ is spanning and has three strongly connected components: one is the cycle $a \to a' \to a$, another one is the cycle $d \to d' \to d'' \to d$, and the third consists of

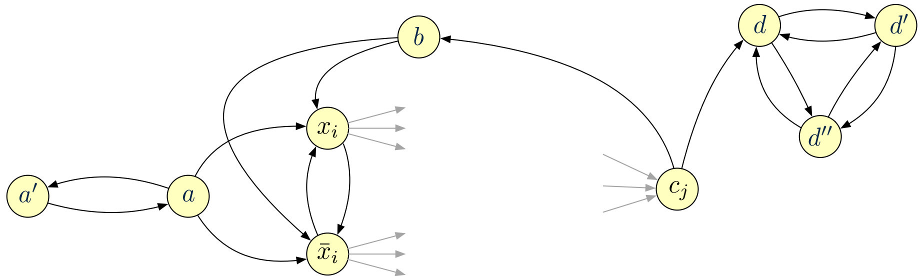

Figure 11. The variable and clause gadgets in the proof of Theorem 7.

all other vertices. This third component is indeed strongly connected because each clause c_j has a true literal, say $\tilde{x}_i$, so its corresponding vertex has incoming edge $(\tilde{x}_i, c_j)$. We then have arcs from all vertices c_j to b and from b to each pair x_i and $\bar{x}_i$. For each i, among x_i and $\bar{x}_i$, the true literal $\tilde{x}_i$ is connected to all clauses where it appears (and it must appear at least once, by our assumption), and its negation $\bar{\tilde{x}}_i$ is connected to $\tilde{x}_i$. So (c.1) holds.

Condition (c.2) can be verified by inspection, namely checking that $\textsc{Deal}^{\mathcal{D}}_v \preceq \textsc{Deal}^{\mathcal{G}}_v$ holds for each vertex v. For example, consider some variable x_i and assume that x_i is true (the case when x_i is false is symmetric). Then $\textsc{Deal}^{\mathcal{G}}_{x_i} = \langle b, \bar{x}_i \,|\, C(x_i)\rangle \succ \textsc{Deal}^{\mathcal{D}}_{x_i}$, and $\textsc{Deal}^{\mathcal{G}}_{\bar{x}_i} = \langle b \,|\, x_i\rangle \succ \textsc{Deal}^{\mathcal{D}}_{\bar{x}_i}$. Next, consider some clause c_j. Since our truth assignment satisfies c_j, c_j has some true literal $\tilde{x}_i$. Then $\mathcal{G}$ will have arc $(\tilde{x}_i, c_j)$. Denoting by $T(c_j)$ the set of true literals in c_j, we then have $\textsc{Deal}^{\mathcal{G}}_{c_j} = \langle T(c_j) \,|\, b\rangle \succeq \langle \tilde{x}_i \,|\, b\rangle \succ \textsc{Deal}^{\mathcal{D}}_{c_j}$. Checking that $\textsc{Deal}^{\mathcal{D}}_v \preceq \textsc{Deal}^{\mathcal{G}}_v$ holds for $v \in \{a, a', b, d, d', d''\}$ is straighforward. Thus, condition (c.2) is verified.

To establish condition (c.3), let $\mathcal{H}$ be a subgraph of $\mathcal{D}$ that dominates $\mathcal{G}$, that is $\textsc{Deal}^{\mathcal{H}}_v \succeq \textsc{Deal}^{\mathcal{G}}_v$ for each vertex $v \in \mathcal{H}$. We claim that then in fact we must have $\mathcal{H} = \mathcal{G}$, which will imply (c.3). This claim follows from the following two observations: (i) For each arc $(u, v) \in \mathcal{G}$, vertex v does not have any outcome that does not include incoming arc (u, v) and is better than $\textsc{Deal}^{\mathcal{G}}_v$. (ii) For each arc $(u, v) \in \mathcal{D} \setminus \mathcal{G}$, vertex u does not have any outcome that includes outgoing arc (u, v) and is better than $\textsc{Deal}^{\mathcal{G}}_u$.

These observations can be verified by inspection. Starting with a, for each literal $\tilde{x}_i$, there is no outcome of a that is better than $\textsc{Deal}^{\mathcal{G}}_a$ that includes arc $(a, \tilde{x}_i)$ or does not include arc (a', a). For a', there is no outcome better than $\textsc{Deal}^{\mathcal{G}}_a = \langle a \,|\, a\rangle$ that does not include arc (a, a'). Consider some x_i, and suppose that x_i is true in our truth assignment. There is no outcome of x_i better than $\textsc{Deal}^{\mathcal{G}}_{x_i} = \langle b, \bar{x}_i \,|\, C(x_i)\rangle$ that does not include arcs (b, x_i) and $(\bar{x}_i, x_i)$, or that includes arc $(x_i, \bar{x}_i)$. Regarding $\bar{x}_i$, there is no outcome of $\bar{x}_i$ better than $\textsc{Deal}^{\mathcal{G}}_{\bar{x}_i}$ that does not have arc $(b, \bar{x}_i)$ or that has any arc $(\bar{x}_i, c_j)$, for some clause c_j. Next, consider arcs between literals and clauses. For a clause c_j we have $\textsc{Deal}^{\mathcal{G}}_{c_j} = \langle T(c_j) \,|\, b\rangle$. There is no outcome of c_j that misses one of the arcs from $T(c_j)$ or includes arc (c_j, d) and is better than $\langle T(c_j) \,|\, b\rangle$. (And we have already showed that in $\mathcal{H}$, vertex c_j cannot have arcs from its false literals.) There is also no outcome of b without arc (c_j, b) better than $\textsc{Deal}^{\mathcal{G}}_b$. The verification of the two observations for the arcs between d, d' and d'' can be carried out in the same manner.

$(\Leftarrow)$ Assume now that $\mathcal{D}$ has a spanning subgraph $\mathcal{G}$ that satisfies properties (c.1) and (c.2). (We will not use (c.3) for now). From $\mathcal{G}$ we will construct a satisfying assignment for α. Condition (c.1) implies that $\mathcal{G}$ cannot have any arcs $(a, \tilde{x}_i)$, so vertices a, a' will form one strongly connected component of $\mathcal{G}$. Similarly, $\mathcal{G}$ cannot have any arcs (c_j, d), so vertices d, d', d'' will also form a strongly connected component. In the rest of the argument we focus on the remaining vertices.

For each literal $\tilde{x}_i$, since $\textsc{Deal}^{\mathcal{G}}_{\tilde{x}_i} \succeq \textsc{Deal}^{\mathcal{D}}_{\tilde{x}_i}$, and also using the preferences of $\tilde{x}_i$, we obtain that $\mathcal{G}$ must have arc $(b, \tilde{x}_i)$. Similarly, using the preferences of b, $\mathcal{G}$ must contain all arcs (c_j, b). (This also follows from the fact that c_j's cannot be singleton strongly connected components of $\mathcal{G}$.) This means that all vertices b, $\tilde{x}_i$ and c_j are in the same connected component of $\mathcal{G}$ which, by property (c.1), must be strongly connected.

From the above paragraph, by strong connectivity, for each i either x_i or $\bar{x}_i$ must have an arc to some clause vertex. Also, since $\textsc{Deal}^{\mathcal{G}}_{x_i} \succeq \textsc{Deal}^{\mathcal{D}}_{x_i}$, if x_i has an arc to a clause vertex then $\mathcal{G}$ must have arc $(\bar{x}_i, x_i)$ and $\mathcal{G}$ cannot have arc $(x_i, \bar{x}_i)$. In turn, since $\textsc{Deal}^{\mathcal{G}}_{\bar{x}_i} \succeq \textsc{Deal}^{\mathcal{D}}_{\bar{x}_i}$, $\bar{x}_i$ has no arcs in $\mathcal{G}$ to any clause vertices. Summarizing, we have this: exactly one of arcs $(x_i, \bar{x}_i)$ or $(\bar{x}_i, x_i)$ is in $\mathcal{G}$, and if $(\tilde{x}_i, \bar{\tilde{x}}_i)$ is in $\mathcal{G}$ then $\tilde{x}_i$ does not have any arcs to clause vertices. This allows us to define a satisfying assignment, as follows. If $\mathcal{G}$ has arc $(\bar{x}_i, x_i)$, set x_i to true, and if $\mathcal{G}$ has arc $(x_i, \bar{x}_i)$, then set x_i to false.

Using condition (c.1), in $\mathcal{G}$ each vertex c_j must have at least one incoming arc from some literal $\tilde{x}_i$ in c_j. By the previous paragraph, this literal is true in our truth assignment, so it satisfies c_j. This establishes that all clauses are satisfied.

To prove the second statement in the lemma, we modify our construction. Note that in the above proof we did not use property (c.3) in the $(\Leftarrow)$ implication. If $\mathcal{D}$ is strongly connected, then it's itself a candidate for $\mathcal{G}$, so the modified construction will need to rely on property (c.3) somehow.

This modification is in fact quite simple. Add arcs from all literal vertices $\tilde{x}_i$ to a, and set the preferences of a so that it prefers to drop the arcs to and from these literal vertices to form a coalition with a'. We apply the same trick to vertex d: it will have arcs going back to all c_j's, but it will be happy to drop these arcs, as well as the arc from d'', in exchange for dropping the arc to d'. Then in the proof for implication ($\Leftarrow$) we use condition (c.3) to argue that the arcs from a to all $\tilde{x}_i$'s will not be in $\mathcal{G}$, for otherwise a subgraph $\mathcal{D}$ consisting of a, a' and the arcs between them would strictly dominate $\mathcal{G}$. For the same reason, $\mathcal{G}$ will not have arcs from d to any c_j. $\square$

Comment: The NP-hardness result in Theorem 7 holds even if we require that preference posets are specified by listing all preference pairs (including the generic ones). This can be shown by modifying the construction so that all vertices in $\mathcal{D}$ have constant degree, and thus all preference posets will have constant size. To this end, we can use a variant of CNF where each clause has three literals and each variable appears at most three times. Then the only vertices of unbounded degree will be a, b, and d. For a, its set of outgoing arcs can be replaced by a chain of vertices each with one outgiong arc to one outneighbor of a. The same trick applies to the arcs of b and d.