Appendix

IX. Σ_2-Completeness

In this section, we give the complete, detailed proof described in VI. 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 $\exists\forall$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][6]. 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$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 $\exists\forall$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 4. *$\exists\forall\mathsf{DNF}_{1x}$ 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$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* SwapAtomic *be the decision problem of deciding whether a swap system has an atomic protocol.* SwapAtomic *is Σ_2^{P}-complete.*

[6]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 piecewise 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 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 $\boldsymbol{\phi}$ to the $\mathbf{x}$-variables will kill some terms, while other will survive. Thus α will be true for assigment $\boldsymbol{\phi}$ 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 assigment $\boldsymbol{\phi}$ 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 $\mathcal{G}$.

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:

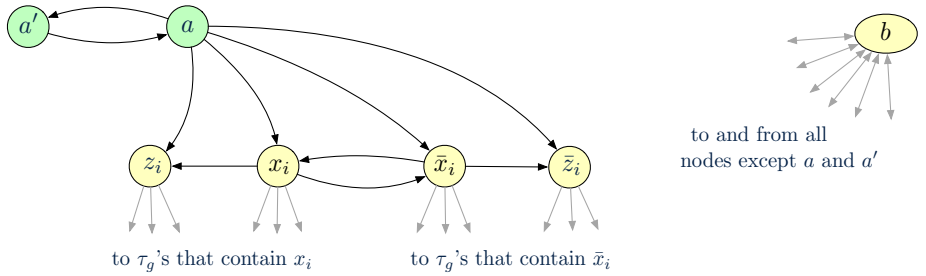

Fig. 9. The construction of digraph $\mathcal{D}$ in the proof of Σ_2^{P}-hardness. This figure shows vertices a, a', b, and an $\exists$-gadget for variable x_i. The arcs to and from b are shown as bi-directional arrows at b.

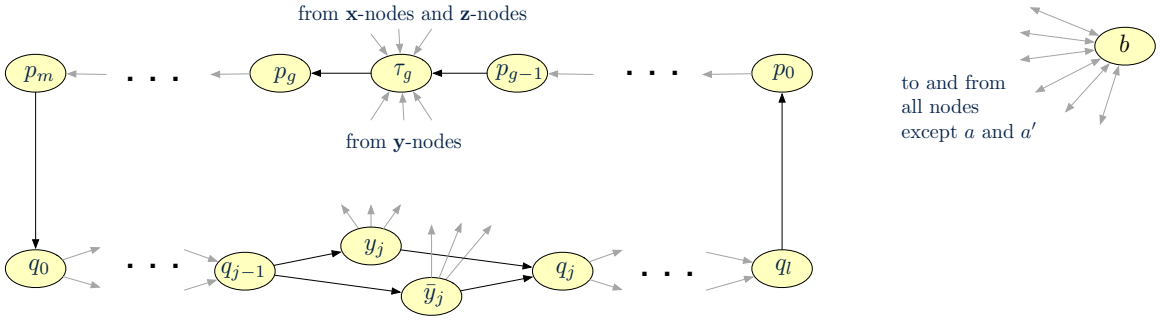

Fig. 10. The construction of digraph $\mathcal{D}$ in the proof of Σ_2^{P}-hardness. This figures shows the $\forall$-gadget, namely the part of $\mathcal{D}$ that contains the vertices that simulate setting the values of the y_j-variables and the terms τ_g. The arcs to and from b are shown as bi-directional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

— 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), 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 **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 II, 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 $\textsc{Deal}_{\tilde{x}_i} \prec \langle b \,|\, b, \bar{\tilde{x}}_i, T(\tilde{x}_i) \rangle$ and $\textsc{Deal}_{\bar{\tilde{x}}_i} \prec \langle b, \bar{\tilde{x}}_i \,|\, b, \tilde{z}_i \rangle$, 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 $\textsc{Deal}_{\tilde{z}_i} \prec \langle b \,|\, b \rangle$ and $\textsc{Deal}_{\tilde{z}} \prec \langle b, \tilde{x}_i \,|\, b, T(\tilde{x}_i) \rangle$.
— For each literal $\tilde{y}_j$, its generators are $\textsc{Deal}_{\tilde{y}_j} \prec \langle b \,|\, b \rangle$ 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 $\textsc{Deal}_{q_j} \prec \langle b \,|\, b \rangle$ 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 $\textsc{Deal}_{q_0} \prec \langle b \,|\, b \rangle$ 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 $\textsc{Deal}_{q_l} \prec \langle b \,|\, b \rangle$ 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 **x**-literal in τ_g, its generators are: $\textsc{Deal}_{\tau_g} \prec \langle b, \tilde{x}_i \,|\, b \rangle$, $\langle b, \tilde{x}_i \,|\, b \rangle \prec \langle p_{g-1}, L \,|\, p_g \rangle$ for any subset L of the **y**-literals in τ_g, $\textsc{Deal}_{\tau_g} \prec \langle b, \tilde{z}_i \,|\, b \rangle$, 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 **y**-literals in τ_g. For each p_g, where $g \notin \{0, m\}$, its generators are $\textsc{Deal}_{p_g} \prec \langle b \,|\, b \rangle$ and $\langle b \,|\, b \rangle \prec \langle \tau_g \,|\, \tau_{g+1} \rangle$.
— The generators of p_0 are $\textsc{Deal}_{p_0} \prec \langle b \,|\, b \rangle$ and $\langle b \,|\, b \rangle \prec \langle q_l \,|\, \tau_1 \rangle$. The generators of p_m are $\textsc{Deal}_{p_m} \prec \langle b \,|\, b \rangle$ 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

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

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 $\textsc{Deal}_v^{\mathcal{G}} \succeq \textsc{Deal}_v^{\mathcal{D}}$. For $v \in \{a', b\}$, we have $\textsc{Deal}_v^{\mathcal{G}} = \textsc{Deal}_v^{\mathcal{D}}$. For $v = a$, $\textsc{Deal}_a^{\mathcal{G}} = \langle a' \,|\, a' \rangle \succ \textsc{Deal}_v^{\mathcal{D}}$. For $v = \tilde{x}_i$ there are two cases: either $\textsc{Deal}_{\tilde{x}_i}^{\mathcal{G}} = \langle b, \bar{\tilde{x}}_i \,|\, b, \tilde{z}_i \rangle$ (if $\phi(\tilde{x}_i) = 1$) or $\textsc{Deal}_{\tilde{x}_i}^{\mathcal{G}} = \langle b \,|\, b, \bar{\tilde{x}}_i, T(\tilde{x}_i) \rangle$ (if $\phi(\tilde{x}_i) = 0$); in both cases $\textsc{Deal}_{\tilde{x}_i}^{\mathcal{G}} \succeq \textsc{Deal}_{\tilde{x}_i}^{\mathcal{D}}$. For $v = \tilde{z}_i$, similarly, either $\textsc{Deal}_{\tilde{z}_i}^{\mathcal{G}} = \langle b, \tilde{x}_i \,|\, b, T(\tilde{x}_i) \rangle$ (if $\phi(\tilde{x}_i) = 1$) or $\textsc{Deal}_{\tilde{z}_i}^{\mathcal{G}} = \langle b \,|\, b \rangle$ (if $\phi(\tilde{x}_i) = 0$); in both cases $\textsc{Deal}_{\tilde{z}_i}^{\mathcal{G}} \succeq \textsc{Deal}_{\tilde{z}_i}^{\mathcal{D}}$. 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 $\textsc{Deal}_v^{\mathcal{G}} = \langle b \,|\, b \rangle \succeq \textsc{Deal}_v^{\mathcal{D}}$. 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 $\textsc{Deal}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{z}_i \,|\, b \rangle$, and if $\phi(\tilde{x}_i) = 0$ then $\textsc{Deal}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{x}_i \,|\, b \rangle$. In both cases, $\textsc{Deal}_{\tau_j}^{\mathcal{G}} \succeq \textsc{Deal}_{\tau_j}^{\mathcal{D}}$.

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 $\textsc{Deal}_v^{\mathcal{H}} \succeq \textsc{Deal}_v^{\mathcal{G}}$ 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 $\textsc{Deal}_b^{\mathcal{H}} \succeq \textsc{Deal}_b^{\mathcal{G}}$, $\mathcal{H}$ 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 $\textsc{Deal}_v^{\mathcal{G}}$ 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 $\textsc{Deal}_v^{\mathcal{G}}$. (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 $\textsc{Deal}_u^{\mathcal{G}}$. These observations imply that $\textsc{Deal}_v^{\mathcal{H}} \succeq \textsc{Deal}_v^{\mathcal{G}}$ 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 X.)

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 $\textsc{Deal}_{\tilde{x}_i}^{\mathcal{G}} = \langle b, \bar{\tilde{x}}_i \,|\, b, \tilde{z}_i \rangle$ 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 $\bar{\tilde{x}}_i, \textsc{Deal}_{\bar{\tilde{x}}_i}^{\mathcal{G}} = \langle b \,|\, b, \tilde{x}_i, T(\bar{\tilde{x}}_i) \rangle$. There is no outcome of $\bar{\tilde{x}}_i$ better than $\textsc{Deal}_{\bar{\tilde{x}}_i}^{\mathcal{G}}$ that includes arc $(\bar{\tilde{x}}_i, \bar{\tilde{z}}_i)$. For a vertex $\tilde{z}_i$ (still assuming that $\phi(\tilde{x}_i) = 1$), $\textsc{Deal}_{\tilde{z}_i}^{\mathcal{G}} = \textsc{Deal}_{\tilde{z}_i}^{\mathcal{D}} = \langle b, \tilde{x}_i \,|\, b, T(\tilde{x}_i) \rangle$; 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 $\textsc{Deal}_{\bar{\tilde{z}}_i}^{\mathcal{G}} = \langle b \,|\, b \rangle$.

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 $\textsc{Deal}_v^{\mathcal{G}} = \langle b \,|\, b \rangle$ 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 $\textsc{Deal}_v^{\mathcal{G}}$. 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 $\textsc{Deal}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{z}_i \,|\, b \rangle$ (if $\phi(\tilde{x}_i) = 1$) or $\textsc{Deal}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{x}_i \,|\, b \rangle$ (if $\phi(\tilde{x}_i) = 0$). In either case, as before, there is no outcome better than $\textsc{Deal}_{\tau_g}^{\mathcal{G}}$ 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 $\textsc{Deal}_v^{\mathcal{G}}$ 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 $\textsc{Deal}_v^{\mathcal{G}}$ 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 $\textsc{Deal}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{x}_i \,|\, b\rangle$. 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 $\textsc{Deal}_{\tau_g}^{\mathcal{G}} = \langle b, \tilde{z}_i \,|\, b\rangle$. 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 $v = \tilde{y}_j$, $\textsc{Deal}_{\tilde{y}_j}^{\mathcal{G}} = \langle b \,|\, b\rangle$. 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 $v = q_j$, $\textsc{Deal}_{q_j}^{\mathcal{G}} = \langle b \,|\, b\rangle$. By the preference poset of q_j, where $j \notin \{0, l\}$, the following outcomes of q_j are better than $\textsc{Deal}_{q_j}^{\mathcal{G}}$: $\langle y_j \,|\, y_{j+1}\rangle$, $\langle \bar{y}_j \,|\, y_{j+1}\rangle$, $\langle y_j \,|\, \bar{y}_{j+1}\rangle$ 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 $v = p_g$, $\textsc{Deal}_{p_g}^{\mathcal{G}} = \langle b \,|\, b\rangle$. 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 $\textsc{Deal}_v^{\mathcal{G}}$. 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 $\textsc{Deal}_{\tau_g}^{\mathcal{G}}$; 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 $\textsc{Deal}_{\tau_g}^{\mathcal{G}}$ 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 $\textsc{Deal}_{\tau_g}^{\mathcal{G}}$ 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 $\textsc{Deal}_v^{\mathcal{D}}$ 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 $\langle b, \bar{\tilde{x}}_i \,|\, b, \tilde{z}_i\rangle$ 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 $\textsc{Deal}_{\tilde{x}_i}^{\mathcal{G}}$ 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 $\textsc{Deal}_{\tilde{x}_i}^{\mathcal{G}}$ 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 $\textsc{Deal}_{\tilde{x}_i} \prec \langle b \,|\, b, \bar{\tilde{x}}_i, T(\tilde{x}_i)\rangle$

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. □

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 $\textsc{Deal}^{\mathcal{G}}_{\tilde{x}_i}$ is a 0-outcome then $\boldsymbol{\xi}(\tilde{x}_i) = 0$, and if $\textsc{Deal}^{\mathcal{G}}_{\tilde{x}_i}$ 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 $v \in \{p_g\}_{g=0}^{m} \cup \{\tilde{y}_j\}_{j=1}^{l} \cup \{q_j\}_{j=0}^{l}$, $\textsc{Deal}^{\mathcal{G}}_{v} \prec \textsc{Deal}^{\mathcal{H}}_{v}$ 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 $\textsc{Deal}^{\mathcal{H}}_{\tau_g}$ is an improvement in comparison to $\textsc{Deal}^{\mathcal{G}}_{\tau_g}$ 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 $\textsc{Deal}^{\mathcal{G}}_{\tilde{x}_i}$ 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 $\text{DEAL}_{\tau_g}^{\mathcal{G}}$. With this, we have proven every vertex $v \in \mathcal{H}$ improves their outcome from $\text{DEAL}_v^{\mathcal{G}}$, 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.

□

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.

□

X. 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 V.

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 II, 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 $\text{DEAL}_{\bar{x}_i} \prec \langle b \,|\, \bar{\tilde{x}}_i \rangle$, 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 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 $\text{DEAL}_v^{\mathcal{D}} \preceq \text{DEAL}_v^{\mathcal{G}}$ 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 $\text{DEAL}_{x_i}^{\mathcal{G}} = \langle b, \bar{x}_i \,|\, C(x_i) \rangle \succ \text{DEAL}_{x_i}^{\mathcal{D}}$, and $\text{DEAL}_{\bar{x}_i}^{\mathcal{G}} = \langle b \,|\, x_i \rangle \succ \text{DEAL}_{\bar{x}_i}^{\mathcal{D}}$. 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 $\text{DEAL}_{c_j}^{\mathcal{G}} = \langle T(c_j) \,|\, b \rangle \succeq \langle \tilde{x}_i \,|\, b \rangle \succ \text{DEAL}_{c_j}^{\mathcal{D}}$. Checking

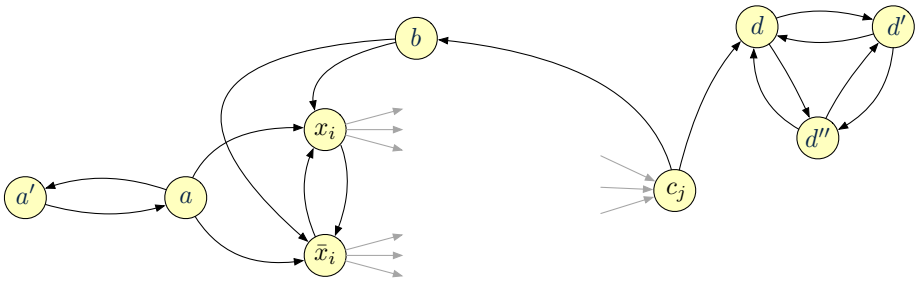

Fig. 11. The variable and clause gadgets in the proof of Theorem 7.

that $\text{DEAL}_v^{\mathcal{D}} \preceq \text{DEAL}_v^{\mathcal{G}}$ 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 $\text{DEAL}_v^{\mathcal{H}} \succeq \text{DEAL}_v^{\mathcal{G}}$ 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 $\text{DEAL}_v^{\mathcal{G}}$. (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 can be verified by inspection. Starting with a, for each literal $\tilde{x}_i$, there is no outcome of a that is better than $\text{DEAL}_a^{\mathcal{G}}$ that includes arc $(a, \tilde{x}_i)$ or does not include arc (a', a). For a', there is no outcome better than $\text{DEAL}_a^{\mathcal{G}} = \langle a \,|\, a \rangle$ 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 $\text{DEAL}_{x_i}^{\mathcal{G}} = \langle b, \bar{x}_i \,|\, C(x_i) \rangle$ 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 $\text{DEAL}_{\bar{x}_i}^{\mathcal{G}}$ 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 $\text{DEAL}_{c_j}^{\mathcal{G}} = \langle T(c_j) \,|\, b \rangle$. 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 $\text{DEAL}_b^{\mathcal{G}}$. 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 $\text{DEAL}_{\tilde{x}_i}^{\mathcal{G}} \succeq \text{DEAL}_{\tilde{x}_i}^{\mathcal{D}}$, 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 $\text{DEAL}_{x_i}^{\mathcal{G}} \succeq \text{DEAL}_{x_i}^{\mathcal{D}}$, 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 $\text{DEAL}_{\bar{x}_i}^{\mathcal{G}} \succeq \text{DEAL}_{\bar{x}_i}^{\mathcal{D}}$, $\bar{x}_i$ 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. □

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.

XI. Experiments

To further study the complexity of SwapAtomic (*i.e.*, given a swap system $\mathcal{S} = (\mathcal{D}, \mathcal{P})$, decide whether it has an atomic protocol), we programmed a simple implementation in C++. We note that this algorithm would be run by the party assembling the swap system, preceding any interaction with any blockchain. This would normally be a market clearing service.

The algorithm runs in three phases. Each phase is a filter for a condition in Theorem 3. We start with every possible graph $\mathcal{G}$, and pass each of them through the three filters. If there is a graph remaining, then we decide yes, otherwise we decide no. The first condition is that $\mathcal{G}$ is spanning, piece-wise strongly connected, and contains no isolated vertices. We first check that $\mathcal{G}$ contains every vertex, each with at least one incoming and outgoing arc. If so, we find the strongly connected components of $\mathcal{G}$ using Kosaraju's algorithm [31]. We then check for every arc (u, v) in $\mathcal{G}$ that u and v are in the same component. If so, then $\mathcal{G}$ is piece-wise strongly connected, and we pass this graph to the second phase.

The second condition is that $\mathcal{G}$ dominates $\mathcal{D}$, the original digraph. That is, for every vertex v, $\textsc{Deal}_v^{\mathcal{D}} \preceq \textsc{Deal}_v^{\mathcal{G}}$, where $\textsc{Deal}_v^{\mathcal{D}}$ is the outcome for v if every arc in $\mathcal{D}$ were triggered, and $\textsc{Deal}_v^{\mathcal{G}}$ is the outcome for v if every arc in $\mathcal{G}$ were triggered. This is simple. We say $\textsc{Deal}_v^{\mathcal{D}} \succeq \textsc{Deal}_v^{\mathcal{G}}$ if (1) they are the same outcome, (2) $\textsc{Deal}_v^{\mathcal{G}}$ is inclusively monotone of $\textsc{Deal}_v^{\mathcal{D}}$, or (3) $\textsc{Deal}_v^{\mathcal{D}} \succ \textsc{Deal}_v^{\mathcal{G}}$ by a non-generic generator (and transitivity). If this holds for every vertex, then $\mathcal{G}$ dominates $\mathcal{D}$ and we pass $\mathcal{G}$ to the third phase.

The last condition is that there is no subgraph $\mathcal{H}$ of $\mathcal{D}$ that strictly dominates $\mathcal{G}$. To verify this, we generate every possible subgraph $\mathcal{H}$. Then, for every vertex v in $\mathcal{H}$, we see if $\textsc{Deal}_v^{\mathcal{G}} \preceq \textsc{Deal}_v^{\mathcal{H}}$ and at least one vertex where $\textsc{Deal}_v^{\mathcal{G}} \prec \textsc{Deal}_v^{\mathcal{H}}$. If not, then $\mathcal{H}$ does not strictly dominate $\mathcal{G}$. If no $\mathcal{H}$ strictly dominates $\mathcal{G}$, then we decide yes. However, if after all three phases no graph remains, we decide no.

Results and Assessment. We ran this program on the example swap systems presented in this paper. The program was written in C++11 and compiled with g++ 12.2.0. It was ran on a Windows 10 machine with a Intel Core i5-11400F 6-Core 2.6GHz CPU and 16 GB RAM. We list the mean of ten runs of each swap system. We provide three additional datapoints: (1) number of arcs in the digraph, (2) number of non-generic preferences generators, and (3) whether or not the swap system ended up permitting an atomic protocol.

Results				
Swap System	Runtime	Arcs	Preferences	Protocol?
$\mathcal{S}_1$	0.0567s	6	5	Yes
$\mathcal{S}_2$	0.016s	6	2	No
$\mathcal{S}_3$	123.116s	14	14	Yes
$\mathcal{S}_4$	61.851s	14	12	No
$\mathcal{S}_5$	328.904s	17	14	No

Swap system $\mathcal{S}_1$ is the system defined in Example 1. Swap system $\mathcal{S}_2$ is the system defined in Figure 3. Swap system $\mathcal{S}_3$ is the system defined in Example 4. Swap system $\mathcal{S}_4$ is $\mathcal{S}_3$, except the two preference generators $\textsc{Deal}_{t_1} \prec \langle t_2 \,|\, t_2 \rangle$ and $\textsc{Deal}_{t_2} \prec \langle t_1 \,|\, t_1 \rangle$ are removed. Swap system $\mathcal{S}_5$ is $\mathcal{S}_3$, except we add a new party s_1 and arcs (u_1, s_1), (u_2, s_1), and (s_1, t_1). Non-generic preferences are not changed.

As we can see in $\mathcal{S}_1$ and $\mathcal{S}_2$, it is feasible to compute SwapAtomic for small swap systems, as expected. The runtimes are less than a second. We next look at larger graphs and highlight the difficulty of SwapAtomic. We observe that $\mathcal{S}_3$ and $\mathcal{S}_4$ have higher runtimes. Further, their runtimes are not in the same ballpark although they have the same number of arcs. Firstly, because piece-wise strong connectivity is a requirement, one might suspect that the cause is the number of arcs or the degree of the vertices. However, the digraphs for both swap systems are exactly the same. The natural reaction is to look at the preference posets. We removed two generators from $\mathcal{S}_3$ to $\mathcal{S}_4$. This made it so the swap system no longer had an atomic protocol, which reduced the runtime. This is because in phase three, the program halts as soon as it finds an $\mathcal{H}$ for every $\mathcal{G}$ (that passed phases one and two). On the other hand, when the system does permit a protocol, the entirety of phase three needs to finish. That is, it needs to check all possible $\mathcal{H}$ to verify $\mathcal{G}$ has no strictly dominating subgraphs. Lastly, from $\mathcal{S}_3$ to $\mathcal{S}_5$, we added one party and three arcs, but no non-generic preferences were changed. Although $\mathcal{S}_5$ ended up not permitting a protocol, it scaled poorly with respect to $\mathcal{S}_3$.

In practice, the runtimes may not be predictable, as is the case with NP-Hard problems. Needless to say, an increase in the number of arcs will generally increase the running time as there are more rounds of Kosaraju's algorithm in phase one. Additionally, if one is to believe the swap system does indeed permit a protocol, then one should expect a long runtime as well, as the program needs to verify every subgraph in phase three.