The Random Priority Mechanism is Uniquely Simple, Efficient, and Fair

Marek Pycia and Peter Troyan*

August 2024

Abstract

Random Priority is a popular mechanism used to allocate a set of objects to a set of agents without the use of monetary transfers. Random Priority is appealing because it satisfies desirable efficiency, fairness, and incentive properties. Is it the only mechanism with these properties? We answer this long-standing question in the positive: Random Priority is the unique mechanism that is Pareto efficient, symmetric, and obviously strategy-proof.

*Pycia: University of Zurich; Troyan: University of Virginia. This paper subsumes the analysis of Random Priority from 2016-2022 drafts of Pycia and Troyan (2023) (initially presented and distributed under the title "Obvious Dominance and Random Priority"). Due to length concerns, we were asked to shorten the paper after acceptance. As the characterization was essentially independent from the rest of the paper and took nearly half of its length, we proposed removing the characterization as a way to meet the length constraints, and the editor agreed to its removal. For their comments, we would like to thank Itai Ashlagi, Sarah Auster, Eduardo Azevedo, Roland Benabou, Dirk Bergemann, Tilman Börgers, Ernst Fehr, Dino Gerardi, Ben Golub, Yannai Gonczarowski, Ed Green, Samuel Haefner, Rustamdjan Hakimov, Stine Helmke, Shaowei Ke, Fuhito Kojima, Simon Lazarus, Jiangtao Li, Shengwu Li, Giorgio Martini, Nelson Mesker, Stephen Morris, Nick Netzer, Ryan Oprea, Ran Shorrer, Erling Skancke, Utku Ünver, Roberto Weber, anonymous referees, the Eco 514 students at Princeton, and the audiences at the 2016 NBER Market Design workshop, NEGT'16, NC State, ITAM, NSF/CEME Decentralization, the Econometric Society Meetings, UBC, the Workshop on Game Theory at NUS, UVa, ASSA, MATCH-UP, EC'19 (the Best Paper prize), ESSET, Wash U, Maryland, Warsaw Economic Seminars, ISI Delhi, Notre Dame, UCSD, Columbia, Rochester, Brown, Glasgow, Singapore Management University, Matching in Practice, Essex, European Meeting on Game Theory, GMU, Richmond Fed, Israel Theory Seminar, USC, Collegio Carlo Alberto, BC, Penn State, and the 2024 Conference on Mechanism and Institution Design. Pycia gratefully acknowledges the support of the UCLA Department of Economics and the William S. Dietrich II Economic Theory Center at Princeton. Troyan gratefully acknowledges support from the Bankard Fund for Political Economy and the Roger Sherman Fellowship at the University of Virginia.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement No 866376.

1 Introduction

Consider the problem of allocating n indivisible objects to n agents without the use of monetary transfers. Examples of such problems include assigning school seats to K12 students, dormitory rooms to college students, tasks to workers, offices to professors, or time slots on a common machine. A classic and oft-used solution to this problem is the *Random Priority* mechanism: an ordering of the agents is drawn uniformly at random, and agents are called, one-by-one, to select their favorite object from those that were not selected by earlier agents.[1]

The popularity of Random Priority largely derives from its desirable efficiency, fairness, and incentive (or simplicity) properties. A long-standing open question is whether any *other* mechanism also satisfies such properties, or whether Random Priority is the unique mechanism to do so. We provide a positive answer by proving that the extensive-form implementation of Random Priority is the only mechanism that is:

- *Pareto efficient:* for any preferences of the agents, the final allocation of Random Priority is Pareto efficient.
- *Symmetric:* if two agents swap their roles in the mechanism, its outcome is unaffected (this property is also known as anonymity).
- *Obviously strategy-proof* (in the sense of Li, 2017)*:* even agents unable to engage in contingent reasoning have dominant strategies.

The possibility of obviously strategy-proof extensive-form implementation matters even for designers restricted to static mechanisms. Indeed, obvious strategy-proofness allows such designers to explain the mechanism in a simple way.[2]

There is a long history of attempting to answer various conjectures about characterizations of Random Priority. On the positive side, Bogomolnaia and Moulin (2001) characterized Random Priority for $n = 3$, while Liu and Pycia (2011) showed that asymptotically, in large markets, all ordinally efficient, equal treatment, strategy-proof mechanisms with small agents have the same marginal distributions as Random Priority.[3] Also related are Abdulkadiroğlu and Sönmez (1998) and Knuth (1996) who showed that Random Priority is equivalent

[1] Random Priority also goes by the name Random Serial Dictatorship, see e.g., Abdulkadiroğlu and Sönmez (1998).

[2] For the importance of simple descriptions, see e.g. Bó and Hakimov (2023), Breitmoser and Schweighofer-Kodritsch (2019), and Gonczarowski, Heffetz, and Thomas (2023).

[3] The characterizations based on ordinal efficiency cannot be extended to finite markets because Bogomolnaia and Moulin (2001) showed that there is no ordinally efficient and strategy-proof mechanism satisfying equal treatment for $n \geq 4$. The asymptotic characterization was possible because Random Priority satisfies ordinal efficiency asymptotically (Che and Kojima, 2010) and, in large markets, ex post and ordinal efficiency coincide (Liu and Pycia, 2011). Relatedly, for $n \geq 3$, Zhou (1990) shows that there is no strategyproof and ex-ante efficient mechanism that satisfies equal treatments of equals; Random Priority fails ex-ante efficiency

to another mechanism called the core from random endowments, which works by first randomly assigning the objects to the agents and then allowing the agents to trade according to Gale's Top Trading Cycles algorithm; they pioneered a bijective approach to equivalence proofs, which we also partially rely on. This equivalence result has been extended, e.g., by Pathak and Sethuraman (2011), Carroll (2014), and Pycia (2019).

The above still left open the question of whether these (or closely related) characterizations hold for any finite market size greater than a few objects. The results here have been in the negative: Erdil (2014) shows that the classic axioms of Pareto efficiency, equal treatment, and strategyproofness do not characterize Random Priority when the number of agents and objects are different; Pycia and Troyan (2023) construct a class of counterexamples that are strongly-obviously strategy-proof (and hence obviously strategy-proof and strategy-proof), Pareto efficient, and satisfy equal treatments of equals; Basteck and Ehlers (2024) construct a counterexample mechanism that is strategy-proof, Pareto efficient, satisfies equal treatments of equals, and in which the distributions of individual agents' outcomes are different than in Random Priority.

Our result, on the other hand, is a positive characterization in terms of natural axioms that applies to any finite market size. The above-mentioned counterexamples show that our characterization is effectively tight: relaxing obvious strategy-proofness to strategy-proofness or relaxing symmetry to equal treatment of equals would break the characterization even if we only wanted to characterize the reduced form of the mechanism (that is the equality of the distributions of individual outcomes). As Random Priority also satisfies stronger incentive and simplicity properties such as one-step simplicity, and strong obvious strategy-proofness (Pycia and Troyan, 2023), our results imply that these stronger incentive and simplicity requirements impose no limitation on efficient, symmetric, and obviously strategy-proof mechanisms in the house allocation environment. Relying on Pycia (2019), we further show that these stronger requirements do not limit what anonymous statistics are achievable to designers of Pareto efficient and obviously strategy-proof mechanisms.

Our analysis contains methodological innovations that might be more generally useful. For instance, in the proof of our main result, we show how to reduce the problem of characterizing symmetric mechanisms to the simpler problem of characterizing mechanisms that are obtained by uniform randomizations over agents' roles in a base mechanism, so called symmetrizations of the base mechanism. In doing so, we generalize Carroll's (2014) terminology of priority roles in Pápai's (2000) Hierarchical Exchange mechanisms to general

even asymptotically (see Abdulkadiroğlu et al., 2011, Featherstone and Niederle, 2016, and Miralles, 2008). Ehlers and Unver (private communication, 2010), Echenique, Pomatto, Root, Sandomirskiy, and Tamuz (private communication, 2020), and Brandt et al. (2023) extended this result to $n = 4$. Pycia and Ünver (2015) discuss methodological tools developed in a failed attempt to prove the conjecture.

extensive-form games.

Following on our work, Basteck (2024) provided a second axiomatic characterization of Random Priority in finite markets relying on efficiency and symmetry, as well as a new axiom of probabilistic monotonicity that he introduced. Probabilistic monotonicity is an adapted version of Maskin monotonicity.

2 Model: The Allocation Problem and Extensive-Form Games

2.1 Environment

Let $\mathcal{N}$ be a set of **agents** and $\mathcal{X}$ a set of **objects**, where $|\mathcal{N}| = |\mathcal{X}|$. Each agent $i \in \mathcal{N}$ has a **strict preference relation**, $\succ_i$, over the set of objects $\mathcal{X}$, where for any $x, y \in \mathcal{X}$, $x \succ_i y$ denotes that object x is strictly preferred to object y. We also refer to $\succ_i$ as agent i's **type**, and write $x \succsim_i y$ to denote that either $x \succ_i y$ or $x = y$. Let $\mathcal{P}$ denote the set of possible types, which consists of all possible strict rankings of the objects. We write $\succ_{\mathcal{N}} = (\succ_i)_{i \in \mathcal{N}}$ to denote a profile of types, one for each agent. A (deterministic) **allocation** $\mu : \mathcal{N} \to \mathcal{X}$ is a bijective function that assigns each agent $i \in \mathcal{N}$ to exactly one of the objects. Let $\mathcal{M}$ be the set of deterministic allocations.

2.2 Extensive-form Games

To determine the final allocation that will be implemented, the planner designs an **extensive-form game**, Γ. We consider imperfect-information, extensive-form games with perfect recall, which are defined in the standard way: There is a finite collection of partially ordered **histories**, $\mathcal{H}$. The notation $h' \subseteq h$ denotes that h' is a subhistory of $h \in \mathcal{H}$. Terminal histories are denoted with bars, $\bar{h}$, and each terminal history $\bar{h} \in \mathcal{H}$ is associated with some allocation in $\mathcal{M}$. At every non-terminal history $h \in \mathcal{H}$, one agent, denoted i_h, is called to play and chooses an **action** from a finite set $A(h)$. We allow for random moves by a non-strategic agent, Nature, who is not one of the agents in $\mathcal{N}$; at any history h at which Nature moves, it selects an action from $A(h)$ according to some predetermined probability distribution. We write $h' = (h, a)$ to denote the history that is reached by starting at h, and following the action $a \in A(h)$. To avoid trivialities, we assume that no agent moves twice in a row, and that $|A(h)| > 1$ for all non-terminal $h \in \mathcal{H}$. The set of histories at which an agent i (either in $\mathcal{N}$ or Nature) moves is denoted $\mathcal{H}_i = \{h \in \mathcal{H} : i_h = i\}$.

To capture imperfect information, $\mathcal{H}_i$ is partitioned into **information sets**, denoted $\mathcal{I}_i$.

For any information set $I \in \mathcal{I}_i$ and $h, h' \in I$ and any subhistories $\hat{h} \subseteq h$ and $\hat{h}' \subseteq h'$ at which i moves, at least one of the following two symmetric conditions obtains: either (i) there is a history $\hat{h}^* \subseteq \hat{h}$ such that $\hat{h}^*$ and $\hat{h}'$ are in the same information set, $A(\hat{h}^*) = A(\hat{h}')$, and i chooses the same action at $\hat{h}^*$ and $\hat{h}'$, or (ii) there is a history $\hat{h}^* \subseteq \hat{h}'$ such that $\hat{h}^*$ and $\hat{h}$ are in the same information set, $A(\hat{h}^*) = A(\hat{h})$, and i chooses the same action at $\hat{h}^*$ and $\hat{h}$. We denote by $I(h) \in \mathcal{I}_i$ the information set containing history h. Given two information sets I_1 and I_2, if there exists $h_1 \in I_1$ and $h_2 \in I_2$ such that $h_1 \subseteq h_2$, then we write $I_1 \leq I_2$, and say that I_1 **precedes** I_2, and that I_2 is a **continuation** of I_1. With slight abuse of notation, we use $A(I)$ to denote the actions available at information set I. An object $x \in \mathcal{X}$ is **possible** for i at information set I if there is some $h \in I$ and some terminal history $\bar{h} \supseteq h$ such that at the allocation associated with $\bar{h}$, $\mu(i) = x$.

2.3 Strategies, Mechanisms, and Equivalence

A **strategy** $S_i(\succ_i)$ for type $\succ_i$ of agent i specifies an action for each information set, $S_i(\succ_i)(I_i) \in A(I_i)$.[4] We use $S = ((S_i(\succ_i))_{\succ_i \in \mathcal{P}})_{i \in \mathcal{N}}$ to denote a profile of strategies. To avoid notational clutter, when the context is clear, we suppress the type-dependence of a strategy, and write $S_i(I_i)$ for the action chosen by agent i at I_i. A **mechanism** (Γ, S) is an extensive-form game Γ together with a profile of strategies, S. Any mechanism induces a lottery over terminal histories, and thus, allocations. We say that two mechanisms (Γ, S) and (Γ', S') are **equivalent** if, for every profile of types $\succ_{\mathcal{N}}$, the distribution over allocations when agents follow S in Γ is the same as that when agents follow S' in Γ'. Every mechanism induces a mapping from type profiles to (random) allocations, which we call the **social choice rule** or the **direct mechanism**. If two mechanisms are equivalent, they implement the same social choice rule.

3 Random Priority and Its Properties

The Random Priority mechanism works as follows. Nature begins by first selecting an ordering of the agents uniformly at random from all possible agent orderings. Agents then move one at a time in this order, and each agent is given the opportunity to choose an object from a menu of all objects that are still available (i.e., that were not chosen by prior agents). At the end of the game, each agent is assigned to exactly one unique object, which determines the final allocation.

[4] We restrict attention to pure strategies. Allowing for mixed strategies would not change any of our results.

The Random Priority mechanism has desirable simplicity, efficiency, and fairness properties. Li (2017) shows that Random Priority is obviously strategy-proof.[5] A strategy $S_i(\succ_i)$ is **obviously dominant** for player i (of type $\succ_i$) in game Γ if for each on-path information set $I^* \in \mathcal{I}_i$, the worst possible outcome for i according to $\succ_i$ in the continuation game assuming i follows $S_i(I)$ at all $I \geq I^*$ is weakly preferred by i to the best possible outcome for i in the continuation game if i plays some other action $a' \neq S_i(I^*)$.[6] If there exists a profile of strategies S such that $S_i(\succ_i)$ is obviously dominant in Γ for all i and all $\succ_i$, then (Γ, S) is said to be **obviously strategy-proof (OSP)**. We then also say that the direct mechanism induced by (Γ, S) is obviously strategy-proof, thus recognizing that the existence of an obvious strategy-proof extensive-form allows designers to explain the direct mechanism in a simple way. Random Priority satisfies this criterion as at an agent's turn, she is able to select from all remaining possible objects. Thus, the worst-case (and in fact, only) outcome from selecting her most preferred remaining object is getting this object, which is clearly at least as good as (and in fact strictly better than) selecting anything else.

Pareto efficiency and fairness of Random Priority have been recognized at least since Abdulkadiroğlu and Sönmez (1998). We say that a deterministic allocation is **Pareto efficient** if, given a type profile $\succ_{\mathcal{N}}$, no other allocation is weakly preferred by all agents, and strictly preferred by at least one; similarly, a mechanism (Γ, S) is ex post Pareto efficient (**Pareto efficient** for brevity) if it leads to a Pareto-efficient allocation for all Nature's choices and agents' types. Random Priority clearly satisfies this property: since each agent selects her most preferred remaining object at her turn, the only way to make an agent strictly better off is to give her an object that was taken by an earlier agent. But then this agent must be given an object taken by an even earlier agent, and so on. Eventually, one of these agents will be unable to be made better off, and so Random Priority is Pareto efficient.

We use a standard fairness criterion of symmetry: the mechanism treats agents equally in the sense that it would not change if any two agents i and j were to switch roles.[7] More formally, a mechanism (Γ, S) is **symmetric** if, for any two agents $i, j \in \mathcal{N}$, the outcome distribution of the mechanism does not change when we transpose the types of agents i and j and at the same time transpose the objects the agents obtain. For instance, symmetry fails

[5] Pycia and Troyan (2023) show that it satisfies the even stronger simplicity standards of one-step simplicity (OSS) and strong obvious strategy-proofness (SOSP).

[6] Given a strategy S_i for i, an information set $I^* \in \mathcal{I}_i$ is on-path if there exist strategies for the other players $j \neq i$ and Nature such that I^* is on the path of play when i plays S_i and all other agents follow their respective strategies. Li (2017) presents the definition of obvious dominance in a slightly different way, using the notion of earliest points of departure. The two formulations are equivalent.

[7] In Appendix A, we define the concept of roles, which make this informal definition formal and equivalent to the in-text definition. Because any permutation can be decomposed into a composition of transpositions, we can equivalently state the symmetry property as $\sigma^{-1} \circ (\Gamma, S) \circ \sigma = (\Gamma, S)$ for all permutations $\sigma : \mathcal{N} \to \mathcal{N}$.

in a deterministic serial dictatorship in which i chooses first and j chooses second: if they have the same most preferred object $x \in \mathcal{X}$, then i obtains x in the original serial dictatorship; after transposing the types of i and j, i still obtains x, but after also transposing the received objects, i no longer obtains x, and so the mechanism is not symmetric. Random Priority, on the other hand, gives each ordering of the agents the same probability, and so in effect, the probability i obtains the preferred object is the same before and after the transposition.

4 The Main Result

Random Priority succeeds on three important design dimensions: it is obviously strategyproof, Pareto efficient, and symmetric. However, this is only a partial explanation of its success, as to now, it has remained unknown whether there exist other such mechanisms, and, if so, what explains the relative popularity of Random Priority over these alternatives. Our main result, Theorem 1 provides an answer to this question: not only does Random Priority have good efficiency, fairness, and simplicity properties, its social choice function is the *only* direct mechanism that does so, thus explaining the widespread popularity of Random Priority in practice.

Theorem 1. ***(Random Priority).*** *An obviously strategy-proof mechanism is symmetric and Pareto efficient if and only if it is equivalent to Random Priority.*

The proof of Theorem 1 can be found in the appendix. Here, we present a simple 3 agent, 3 object example that allows us to illustrate the methods used in the proof.[8] Consider the game presented in Figure 1.[9] The game allocates three objects x, y, and z to three agents. Agent i_1 moves first and can take one of the objects x or y (and leave the game), or can pass (and remain in the game). If i_1 passes, agent i_2 can either take y (in which case the allocation is fully determined: i_1 receives z and i_3 receives x) or pass. Agent i_3 only moves following two passes, and at this point, i_3 can take any object. If i_3 takes x or y, then the allocation is determined, and if agent i_3 takes z then i_1 can choose between x and y. It can be checked that this game is both OSP and Pareto efficient.

[8]For $|\mathcal{N}| = 1$, Theorem 1 follows from Pareto efficiency. For $|\mathcal{N}| = 2$, the equivalence is implied by Pareto efficiency when agents rank objects differently and it is implied by symmetry when they rank objects in the same way. Cf. Bogomolnaia and Moulin (2001) who also analyze the three-agent case; their approach is different and, because of its reliance on ordinal efficiency, not applicable beyond three agents.

[9]Figure 1 is taken from Pycia and Troyan (2023), who use it as an illustration of a "millipede mechanism", which is a class of mechanisms that have a clinch-or-pass structure as in this figure. They show that any OSP mechanism is equivalent to a millipede mechanism. We provide further details on millipede mechanisms in Step 2 of the proof in Appendix B.

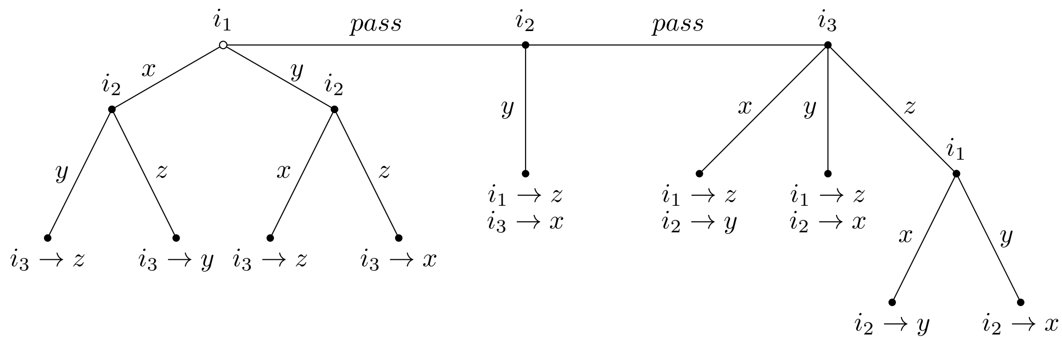

Figure 1: An OSP and Pareto-efficient game Γ with three agents and three objects.

The game in Figure 1 is not symmetric. For instance, consider the preference profile $\succ_{\mathcal{N}} = (\succ_{i_1}, \succ_{i_2}, \succ_{i_3})$ defined by

$$\succ_{i_1}: x, y, z$$

$$\succ_{i_2}: x, y, z$$

$$\succ_{i_3}: z, y, x.$$

Under this profile, the outcome of the mechanism is $\{(i_1, x), (i_2, y), (i_3, z)\}$. If we transpose the preferences of i_1 and i_2 while at the same time transposing the objects agents i_1 and i_2 receive, the outcome is $\{(i_1, y), (i_2, x), (i_3, z)\}$, and thus symmetry fails. However, the mechanism can be symmetrized as follows. Let (Γ^*, S^*) be the mechanism shown in Figure 2. Game Γ^* begins with a move by Nature, drawing a permutation of players σ uniformly at random; we refer to such permutations σ as role assignments. The continuation game Γ_σ is isomorphic to Γ (from Figure 1) with the agents permuted.[10] For instance, if in the first step Nature draws the role assignment $\sigma_1(i_1) = i_1$, $\sigma_1(i_2) = i_2$, and $\sigma_1(i_3) = i_3$, then the agents continue by playing precisely the game in Figure 1; if instead Nature draws the role assignment $\sigma_2(i_1) = i_2$, $\sigma_2(i_2) = i_1$, and $\sigma_2(i_3) = i_3$, then the agents continue by playing the game Figure 1 except with the roles of i_1 and i_2 swapped. Similarly to how randomizing over deterministic serial dictatorships (which are not symmetric) produces the symmetric Random Priority mechanism, randomizing over role assignments in Γ produces the symmetric mechanism (Γ^*, S^*).

The first step in the proof shows that it is sufficient to prove Theorem 1 for symmetrizations.

Proposition 1. *Suppose that, for every deterministic OSP and Pareto-efficient perfect-*

[10]We present the definition of role assignments more formally in Appendix A.

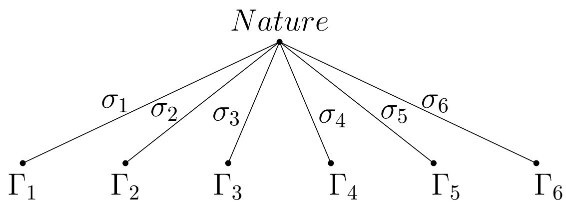

Figure 2: The symmetrization of the mechanism from Figure 1. The mechanism begins with Nature drawing a role assignment function σ_k uniformly at random (there are $3! = 6$ possible permutations of the agents, and thus 6 role assignments). Then, the agents proceed to play the continuation game Γ_k, where Γ_k is the game from Figure 1 with the roles of the agents permuted according to σ_k. Each continuation game $(\Gamma_1, \ldots, \Gamma_6)$ is OSP and Pareto efficient. The grand mechanism (including the draw of the role assignment function) is OSP, Pareto efficient, and symmetric.

information mechanism, its symmetrization is equivalent to Random Priority. Then, every symmetric, OSP and Pareto-efficient mechanism is equivalent to Random Priority.

Proposition 1 illustrates a general insight: establishing a property for symmetrizations of mechanisms from a class $\mathcal{C}$ is sufficient to infer that this property holds for all symmetric mechanisms whenever symmetric mechanisms are in $\mathcal{C}$.[11]

By Proposition 1, it is sufficient to show that the symmetrization of every OSP and Pareto efficient mechanism is equivalent to Random Priority. In our example, we take the OSP and Pareto efficient mechanism (Γ, S) in Figure 1 and construct its symmetrization by having Nature randomly draw a role assignment and then the agents play the game Γ_k, which is equivalent to the game Γ from Figure 1 except the roles of the agents are permuted according to σ_k (see Figure 2).

Next, we construct a mapping $f : \Sigma \to Ord$ between role assignment functions and serial dictatorship orderings such that (i) for each σ, the outcome of the continuation game $(\Gamma_\sigma, S_\sigma)$ is the same as a serial dictatorship in which agents choose in the order $f_\sigma(1), f_\sigma(2), f_\sigma(3)$ and (ii) f is a bijection.[12] Since the symmetrized mechanism uniformly randomizes over role assignment functions, the probability of achieving any particular allocation μ is just the number of role assignment functions such that $(\Gamma_\sigma, S_\sigma)$ results in allocation μ, divided by $N!$. Similarly, because Random Priority uniformly randomizes over serial dictatorship orderings, the probability of achieving any particular allocation μ is just the number of serial dictatorship orderings that result in μ divided by $N!$. If there exists a bijection as

[11] Proposition 1 is less general but stronger in that class $\mathcal{C}$ consists of deterministic OSP and Pareto-efficient perfect-information mechanisms and does not include symmetric mechanisms.

[12] $f_\sigma \in Ord$ is an ordering of all of the agents in $\mathcal{N}$ such that $f_\sigma(j)$ is the j^{th} agent in this ordering.

just described, these two numbers will be equal for any μ, and hence, the distribution over allocations in the symmetrized mechanism is the same as in Random Priority, i.e., the two mechanisms are equivalent.[13]

The bulk of the proof is devoted to constructing the bijection f and showing that it is indeed a bijection. For sake of illustration, consider the preference profile $\succ_{\mathcal{N}}$ given above.[14] Consider a role assignment function such that $\sigma(i_k) = i_k$ for $k = 1, 2, 3$. Under this role assignment, the game among the agents is that shown Figure 1, and the resulting play is as follows: agent i_1 moves first and clinches x, agent i_2 moves second and clinches y; agent i_3 receives z without being called to move. In this case, our bijection f maps σ to the following serial dictatorship ordering: $f_\sigma(1) = i_1, f_\sigma(2) = i_2, f_\sigma(3) = i_3$.

Both $(\Gamma_\sigma, S_\sigma)$ and a serial dictatorship with agent ordering f_σ result in the same outcome: $\{(i_1, x), (i_2, y), (i_3, z)\}$. If instead Nature draws the permutation $\sigma'(i_1) = i_2$, $\sigma'(i_2) = i_1$, $\sigma'(i_3) = i_3$, then the game path of $(\Gamma_{\sigma'}, S_{\sigma'})$ has agents i_2, i_1, and i_3 clinching x, y, and z (in this order). The associated serial dictatorship in this case is $f_{\sigma'}(1) = i_2$, $f_{\sigma'}(2) = i_1$, $f_{\sigma'}(3) = i_3$. Once again, it can be checked that both $(\Gamma_{\sigma'}, S_{\sigma'})$ and a serial dictatorship under agent ordering $f_{\sigma'}$ result in the same final allocation: $\{(i_1, y), (i_2, x), (i_3, z)\}$. Indeed, as we show below, any time the game Γ_σ starts with several agents choosing clinching moves, then we map it to a serial dictatorship that starts with the same agents moving in the same order, and it is easy to see that these two mechanisms always result in the same allocation.

The mapping of game paths that involve passing is more subtle. In the present example, passing is on the game path if the role of i_1 is assigned to agent i_3. There are two such permutations: if $\sigma''(i_2) = i_2$ then the resulting outcome is $\{(i_1, x), (i_2, y), (i_3, z)\}$, and if $\sigma'''(i_2) = i_1$, then the resulting outcome is $\{(i_1, y), (i_2, x), (i_3, z)\}$.

To what serial dictatorships should we map these two permutations? In this simple example, it can be checked by hand that the unique mapping achieving a bijection that results in the same allocations under all of the corresponding serial dictatorships maps σ'' into a serial dictatorship with agents ordered i_3, i_1, i_2, and maps σ''' to a dictatorship with agents ordered i_3, i_2, i_1. However, in general, there is no simple rule of thumb in mapping role assignment functions that entail passing on the path of play: notice that in the present example, the resulting serial dictatorships do not order agents in the order in which they

[13] The bijection idea was first employed by Abdulkadiroğlu and Sönmez (1998) and Knuth (1996), and has since been used by others (e.g., Pathak and Sethuraman (2011) and Carroll (2014)). Our construction is different from the bijections in the earlier literature, and relies on the properties of millipede games established in Pycia and Troyan (2023), and on the properties of Pareto-efficient OSP mechanisms subsequently obtained by Bade and Gonczarowski (2017).

[14] The bijection is constructed for a fixed preference profile. Different preference profiles will lead to different bijections, but we still have the outcome distributions for the two mechanisms the same profile-by-profile, and thus the mechanisms are equivalent.

move in Γ_σ, nor do they order agents in the order in which they take their objects. The bulk proof in the appendix is devoted to constructing the bijection for any game, and gives the details of how agents should be ordered when passing is on the path of play.

5 An Application to Simplicity Tradeoffs

In addition to providing an explanation for the popularity of Random Priority, our Theorem 1 has implications for how restrictive various simplicity standards are in the allocation environment we study. It shows that Random Priority, a very simple mechanism, is equivalent to all other obviously strategy-proof, efficient, and symmetric mechanisms. These mechanisms can vary in their simplicity and Pycia and Troyan (2023) introduced a gradated class of simplicity criteria (that includes one-step simplicity and strong obvious strategy-proofness) that differentiate among these various mechanisms. Theorem 1 tells us that imposing the more restrictive criteria does not restrict the designer's ability to implement efficient and symmetric objectives:

Corollary 1. *A Pareto efficient and symmetric social choice rule can be implemented via an obvious strategy-proof mechanism if and only if it can be implemented via a strongly obvious strategy-proof mechanism.*

Our Theorem 1 also allows us also to conclude that one-step simplicity and strong obvious strategy-proofness do not limit the means and medians of statistical outcomes that designers of obviously strategy-proof and efficient mechanisms can achieve, whether these mechanisms are symmetric or not. We formalize and derive this conclusion relying on the approach developed in Pycia (2019). Let us fix a set of classifications $K = \{1, ..., k\}$ and a mapping $f : \mathcal{P} \times \mathcal{X} \to K$ that allows us to classify agents' outcomes. A statistic $F : (\Theta \times A)_{i \in \mathcal{N}} \to [0,1]^K$ is an empirical distribution of the classifications of individual agents' outcomes.[15] Examples include: the ratio of applicants obtaining their top outcome; the ratio of applicants obtaining their two top outcomes; the ratio of applicants assigned objects from some fixed subset; or the ratio of applicants who prefer the object they are assigned to some reference object x. A distribution over $\mathcal{P}^{\mathcal{N}}$ is exchangeable if the probability of $\succ_{\mathcal{N}}$ is the same as the probability of the profile $\succ_{\sigma(\mathcal{N})}$ for any permutation of agents $\sigma : \mathcal{N} \to \mathcal{N}$. Our Theorem 1 and Lemma 1 from Pycia (2019) imply the following:

Corollary 2. *For any Pareto efficient and obviously strategy-proof mechanism, the mean (and median) of any statistic F with respect to any exchangeable distribution over $\mathcal{P}^{\mathcal{N}}$ is the same as the mean (and median) of F under Random Priority.*

[15] Pycia (2019) calls such statistics anonymous.

This result contributes to the burgeoning literature on the costs of strategic simplicity. For instance, Miralles (2008), Abdulkadiroğlu et al. (2011), and Featherstone and Niederle (2016) discuss the costs of strategy-proofness, and and Li (2017), Pycia and Troyan (2023), and Li and Dworczak (2020) discuss the costs of obvious strategy-proofness and other simplicity standards. While these papers illustrate the costs of strategic simplicity, our Corollaries 1 and 2 show that in the single-unit demand allocation problem we study farther simplifications beyond obvious strategy-proofness come at no cost.

6 Conclusion

We have resolved in the positive the long standing conjecture about Random Priority: it is the unique mechanism satisfying desirable incentive, efficiency, and fairness properties. This characterization provides an explanation for the popularity of Random Priority. This characterization also implies that imposing more restrictive simplicity standards than obvious strategy-proofness—e.g., one-step simplicity or strong obvious strategy-proofness—comes at no cost in the context of efficient and fair allocation. The duality lemma of Pycia (2019) allows us to conclude that from normative perspective, imposing these stronger simplicity standards is also without cost in the context of efficient allocation.

References

Abdulkadiroğlu, A., Y.-K. Che, and Y. Yasuda (2011): "Resolving conflicting preferences in school choice: The "Boston mechanism" reconsidered," *American Economic Review*, 101, 399–410.

Abdulkadiroğlu, A. and T. Sönmez (1998): "Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems," *Econometrica*, 66, 689–701.

Ashlagi, I. and Y. A. Gonczarowski (2018): "Stable matching mechanisms are not obviously strategy-proof," *Journal of Economic Theory*, 177, 405–425.

Bade, S. and Y. Gonczarowski (2017): "Gibbard-Satterthwaite Success Stories and Obvious Strategyproofness," .

Basteck, C. (2024): "An axiomatization of the random priority rule," WZB Discussion Paper SP II 2024-201.

Basteck, C. and L. Ehlers (2024): "On (constrained) efficiency of strategy-proof random assignment," Working Paper.

Bó, I. and R. Hakimov (2023): "Pick-an-object mechanisms," *Management Science*.

Bogomolnaia, A. and H. Moulin (2001): "A New Solution to the Random Assignment Problem," *Journal of Economic Theory*, 100, 295–328.

Brandt, F., M. Greger, and R. Romen (2023): "Characterizing Random Serial Dictatorship," Working Paper.

Breitmoser, Y. and S. Schweighofer-Kodritsch (2019): "Obviousness Around the Clock," WZB Discussion Paper SP II 2019–203.

Carroll, G. (2014): "A general equivalence theorem for allocation of indivisible objects," *Journal of Mathematical Economics*, 51, 163–177.

Che, Y.-K. and F. Kojima (2010): "Asymptotic Equivalence of Random Priority and Probabilistic Serial Mechanisms," *Econometrica*, 78, 1625–1672.

Erdil, A. (2014): "Strategy-Proof Stochastic Assignment," *Journal of Economic Theory*, 151, 146–162.

Featherstone, C. R. and M. Niederle (2016): "Boston versus deferred acceptance in an interim setting: An experimental investigation," *Games and Economic Behavior*, 100, 353–375.

Gonczarowski, Y. A., O. Heffetz, and C. Thomas (2023): "Strategyproofness-exposing mechanism descriptions," Tech. rep., National Bureau of Economic Research.

Knuth, D. E. (1996): "An exact analysis of stable allocation," *Journal of Algorithms*, 20, 431–442.

Li, J. and P. Dworczak (2020): "Are Simple Mechanisms Optimal when Agents are Unsophisticated?" Working Paper.

Li, S. (2017): "Obviously Strategy-Proof Mechanisms," *American Economic Review*, 107, 3257–87.

Liu, Q. and M. Pycia (2011): "Ordinal Efficiency, Fairness, and Incentives in Large Markets," .

Miralles, A. (2008): "School Choice: The Case for the Boston Mechanism," Boston University, unpublished mimeo.

Pápai, S. (2000): "Strategyproof Assignment by Hierarchical Exchange," *Econometrica*, 68, 1403–1433.

Pathak, P. A. and J. Sethuraman (2011): "Lotteries in student assignment: An equivalence result," *Theoretical Economics*, 6, 1–17.

Pycia, M. (2019): "Evaluating with Statistics: Which Outcome Measures Differentiate Among Matching Mechanisms?" Working Paper.

Pycia, M. and P. Troyan (2023): "A Theory of Simplicity in Games and Mechanism Design," *Econometrica*, 91(4), 1495–1526.

Pycia, M. and M. U. Ünver (2015): "Decomposing Random Mechanisms," *Journal of Mathematical Economics*, 61, 21–33.

Zhou, L. (1990): "On a Conjecture by Gale about One-Sided Matching Problems," *Journal of Economic Theory*, 52, 123–135.

A Roles and Role Assignment Functions

Our terminology of roles and role assignment generalizes Carroll's (2014) terminology of priority roles in Pápai (2000)'s hierarchical exchanges to general extensive-form games. In the definition of our fairness axiom and the proof of the main theorem below, we make use of the concepts of roles and role assignment functions, which we introduce here. Let $\mathcal{R}$ be a set of players such that $|\mathcal{R}| = |\mathcal{N}|$; we call each $r \in \mathcal{R}$ a **role**. Given any game Γ, we define a corresponding **proto-mechanism**, $(\tilde{\Gamma}, \tilde{S})$, which consists of a **proto-game**, $\tilde{\Gamma}$, and a profile of **proto-strategies,** $\tilde{S}$. The proto-game $\tilde{\Gamma}$ is equivalent to Γ, except that each history h is assigned to a particular role $r \in \mathcal{R}$ (rather than an agent in $\mathcal{N}$), with the restriction that if two histories are controlled by the same agent in Γ, then they are controlled by the same role in $\tilde{\Gamma}$. Formally, letting $\rho : \mathcal{H} \to \mathcal{R}$ be the function that maps each history h to the role that moves at h in $\tilde{\Gamma}$, we require that $\rho(h) = \rho(h')$ if and only if $i_h = i_{h'}$ in Γ. The proto-strategy profile $\tilde{S} = (\tilde{S}_r)_{r \in \mathcal{R}}$ is defined such that $\tilde{S}_r = S_i$, where r is the role that controls the same histories in $\tilde{\Gamma}$ that are controlled by agent i in Γ.

Let Σ be the set of bijections $\sigma : \mathcal{R} \to \mathcal{N}$ between the set of roles and the set of agents $\mathcal{N}$; we call these bijections **role assignment functions**. Given a proto-mechanism $(\tilde{\Gamma}, \tilde{S})$, each role assignment function $\sigma \in \Sigma$ determines a mechanism for the agents in $\mathcal{N}$, denoted $(\Gamma_\sigma, S_\sigma)$, as follows: Γ_σ is the extensive-form game with the same game tree as the proto-game $\tilde{\Gamma}$, and such that at each non-terminal history h, the agent called to move is $\sigma(\rho(h))$; at each terminal history in Γ_σ the object assigned to agent i is the same as the object assigned to role $\sigma^{-1}(i)$ in $\tilde{\Gamma}$; the strategy S_i of agent i in Γ_σ is the same as the strategy of role $\sigma^{-1}(i)$ in $(\tilde{\Gamma}, \tilde{S})$. There are $|\Sigma| = N!$ possible mechanisms $(\Gamma_\sigma, S_\sigma)$; we call them the permuted mechanisms. (See Section 4 for an example of how role assignments work.)

Given a mechanism (Γ, S), we further define the **symmetrization of** (Γ, S), denoted (Γ^*, S^*), to be the following random mechanism: first, Nature chooses a role assignment function σ uniformly at random from the set of all possible role assignment functions, and then, the agents play Γ_σ with strategies S_σ.[16]

B Proof of Theorem 1

We break the proof down into 7 steps. Step 1 shows that it is sufficient to consider symmetrized mechanisms. Steps 2 and 3 show that we can further restrict attention to a subset of the class of millipede mechanisms of Pycia and Troyan (2023). Step 4 constructs a coding algorithm that maps each of the permuted mechanisms $(\Gamma_\sigma, S_\sigma)$ that make up the symmetrization into a corresponding serial dictatorship. Step 5 shows that the resulting serial dictatorship produces the same allocation as the $(\Gamma_\sigma, S_\sigma)$. Step 6 shows that the mapping is in fact a bijection between permuted mechanisms and serial dictatorship orderings. Step 7 wraps up and concludes.

In the proof, we use the concept of roles and role assignments introduced in Appendix A. Proofs of some intermediate results not given here are found in Appendix C.

Step 1: Symmetrization Reduction

The first step in proving Theorem 1 is to recognize that it is sufficient to prove the theorem for any uniform randomization over Pareto-efficient deterministic mechanisms. It is sufficient to consider symmetric randomizations over Pareto-efficient deterministic OSP mechanisms because every symmetric mechanism can be expressed as a lottery over symmetric randomizations. If each of these randomizations is equivalent to Random Priority, then so is the the lottery over them. We stated this insight as Proposition 1 above and we prove it now.

Proof of Proposition 1. Take a symmetric, OSP, and Pareto-efficient mechanism (Γ, S). Lemma A.4 of Pycia and Troyan (2023) shows that for every OSP mechanism, there is an equivalent OSP mechanism with perfect information in which Nature moves at most once, as the first mover.[17] Thus, it is without loss of generality to assume that (Γ, S) has perfect information and that Nature moves only at the beginning of the game. Because (Γ, S) is symmetric, its symmetrization (Γ^*, S^*) is equivalent to (Γ, S). Furthermore, (Γ^*, S^*) is a lottery over symmetrizations of each deterministic perfect-information continuation game Γ'

[16] While this construction implies that different agents play the same strategies in the same role, our arguments only rely on the weaker assumption that an agent's strategy $S_{\sigma,i}(\succ_i)$ depends only on her own preferences and her role assignment, and not on the roles assigned to other agents. In other words, in any two subgames Γ_A and Γ_B following Nature's selection of role assignments σ_A and σ_B, if $\sigma_A^{-1}(i) = \sigma_B^{-1}(i) = r_n$, then $S_{A,i}(\succ_i)(h_A) = S_{B,i}(\succ_i)(h_B)$ for any equivalent histories h_A and h_B in these two games. As an aside note that this element of the construction relies on the fact that the strategies are dominant that is that they remain optimal regardless of strategies played by other agents.

[17] Ashlagi and Gonczarowski (2018) briefly mention this result in a footnote.

after Nature's move in (Γ, S). The mechanism given by game Γ', together with the strategy profile induced from Γ, is OSP and Pareto efficient, and hence by the assumption of the lemma it is equivalent to Random Priority. Because every lottery over Random Priority lotteries is still equivalent to Random Priority, the lemma obtains. ■

In light of the above proposition, it is sufficient to prove Theorem 1 for symmetrizations, i.e., it is sufficient to prove the following.

Proposition 2. *Let* (Γ, S) *be an obviously strategy-proof and Pareto-efficient deterministic perfect-information mechanism. Then, the symmetrization of* (Γ, S) *is equivalent to Random Priority.*

Steps 2-6 are devoted to showing Proposition 2, which, combined with Proposition 1, proves Theorem 1.

Step 2: Millipede Reduction

Let (Γ, S) be an obviously strategyproof and Pareto efficient deterministic perfect-information mechanism. The first step in the proof of Proposition 2 shows that it is without loss of generality to assume that (Γ, S) is a millipede mechanism. Millipede mechanisms are a class of mechanisms introduced in Pycia and Troyan (2023), who show that any, in a broad class of preference environments that include our setting, any OSP mechanism is equivalent to a millipede mechanism. Broadly speaking, in our environment, a millipede mechanism is a perfect-information, extensive-form game such that at each history, one agent is called to move and is offered the opportunity to clinch some subset of still-available objects (and leave the game); she may also be offered an opportunity to pass, and hence remain in the game, waiting for better clinching options in the future.

To formally define a millipede game, we need the following definitions, which are adapted from Pycia and Troyan (2023).

- Possible objects: Object x is **possible** for agent i at history h if there is a terminal history $\bar{h} \supseteq h$ at which i receives x. We let $P_i(h)$ denote the set of objects that are possible for i at h. If $x \in P_i(h')$ for all $h' \subsetneq h$ such that $i_{h'} = i$, but $x \notin P_i(h)$, then we say x **becomes impossible** for i at h.

- Clinchable objects: Object x has been **clinched** by agent i at history h if i receives x at all $\bar{h} \supseteq h$. Object x is **clinchable** for agent i at history h if i moves at h there is some action $a \in A(h)$ such that i has clinched x at $h' = (h, a)$. We let $C_i(h)$ denote the set of objects that are clinchable for agent i at h.

- Clinching actions: An action $a \in A(h)$ is called a **clinching action** if agent i (who moves at h) has clinched x at history $h' = (h, a)$.

- Passing actions: Any action $a \in A(h)$ that is not a clinching action is a **passing action**.

At a terminal history $\bar{h}$, no agent is called to move and there are no actions. However, it is notationally convenient to define $C_i(\bar{h}) = \{x\}$, where x is the object that i receives at $\bar{h}$. We further define the following pieces of notation:

- $C_i^{\subseteq}(h)$ is the set of objects that have been previously clinchable for i at some subhistory of h; formally, $C_i^{\subseteq}(h) = \{x : x \in C_i(h') \text{ for some } h' \subseteq h \text{ s.t. } i_{h'} = i\}$.

- $C_i^{\subset}(h)$ is the set of objects that have been previously clinchable for i at some *strict* subhistory of h; formally, $C_i^{\subset}(h) = \{x : x \in C_i(h') \text{ for some } h' \subsetneq h \text{ s.t. } i_{h'} = i\}$. If $x \notin C_i^{\subset}(h)$, then we say x is **previously unclinchable** at h.

Given a mechanism (Γ, S) and a type $\succ_i$, a strategy $S_i(\succ_i)$ a **greedy strategy** if at any history $h \in \mathcal{H}_i$ it satisfies the following: if the $\succ_i$-best still-possible object in $P_i(h)$ is clinchable at h, then $S_i(\succ_i)(h)$ clinches this payoff object; otherwise, $S_i(\succ_i)(h)$ is a passing action.

With these definitions, a **millipede game** is a finite extensive-form game of perfect information that satisfies the following properties:

1. Nature either moves once, at the empty history $h_\varnothing$, or Nature has no moves.

2. At any history at which an agent moves, all but at most one action are clinching actions, and following any clinching action, the agent does not move again.

3. At all h, if there exists a previously unclinchable payoff x that becomes impossible for agent i_h at h, then $C_{i_h}^{\subset}(h) \subseteq C_{i_h}(h)$.

A **millipede mechanism** is a millipede game with a profile of greedy strategies. In a millipede mechanism, it is obviously dominant for an agent to clinch the best possible object at h whenever it is clinchable. The last condition of the millipede definition says that when some previously unclinchable object becomes impossible for an agent, the next time she moves, she is offered the opportunity to clinch everything that was previously clinchable. This ensures that an agent never "regrets" her decision to pass on a previously offered object, and is formally what is needed to guarantee passing at h is obviously dominant when an agent's best possible object at h is not clinchable. An example of a millipede mechanism in a 3 agent, 3 object setting is given in Figure 1.

Lemma 1. ***(Pycia and Troyan, 2023).*** *Every OSP mechanism is equivalent to a millipede mechanism.*

Using the above result from Pycia and Troyan (2023), it is without loss of generality to assume in Proposition 2 that (Γ, S) is a millipede mechanism. Thus, to prove Proposition 2 we must show that the symmetrization of any Pareto-efficient millipede mechanism is equivalent to Random Priority.

Step 3: Efficient Millipedes

Obvious strategyproofness allows us to assume that (Γ, S) is a millipede mechanism, by Lemma 1. Adding Pareto efficiency allows us to further restrict attention to a subclass of millipede mechanisms that we describe in this step. To describe this class, we must first introduce the concept of a lurker, which is a modification of a similar concept in Bade and Gonczarowski's (2017, hereafter BG) analysis of efficient OSP mechanisms. Informally, a lurker is an agent who has been offered to clinch all objects that are possible for her except for one, which she is said to "lurk".

Let (Γ, S) be a Pareto-efficient millipede mechanism. Call an agent i **active** at h if she has been previously called to play at some $h' \subseteq h$, and has not yet clinched an object at h. Let $\mathcal{A}(h)$ denote the set of active agents at h. Recall that $C_i^{\subseteq}(h)$ is the objects agent i has been offered to clinch at some subhistory of h and $C_i^{\subset}(h)$ is the objects agent i has been offered to clinch at some strict subhistory of h. Further, define $G_i(h)$ as the set of objects that are **guaranteeable** for i at h; formally, $x \in G_i(h)$ if and only if there exists a continuation strategy S_i such that i receives object x at all terminal histories $\bar{h} \supseteq h$ that are consistent with i following strategy S_i starting from h.[18]

Consider a history h and an active agent i who has moved at a strict subhistory of h. Let $h' \subsetneq h$ be the maximal strict subhistory such that $i_{h'} = i$. Agent i is said to be a **lurker** for object x at h if (i) $P_i(h) \neq G_i(h)$, (ii) $x \in P_i(h')$, (iii) $C_i^{\subseteq}(h') = P_i(h') \setminus \{x\}$, and (iv) $x \notin C_j^{\subseteq}(h')$ for any other active $j \neq i$ that is not a lurker at h'. If some agent i is a lurker for an object x at a history h, then we say x is a **lurked object** at h. We use the term **BG lurker** to refer to any agent that satisfies (i), (ii), and (iii).[19] Bade and Gonczarowski

[18]Note the distinction between guaranteeble objects, $G_i(h)$, and clinchable objects, $C_i(h)$: informally, an object x is clinchable at h if there is action $a \in A(h)$ such that i receives x "immediately" (and so no other objects are possible for i following action a), whereas if x is guaranteeable at h, there may be other objects that are possible, but there is some continuation strategy such that if i sticks to this strategy in the continuation game, she can guarantee she will receive x, no matter what the other agents do. The concepts of active agents and guaranteeable objects were introduced in Pycia and Troyan (2023).

[19]BG lurkers were studied in Bade and Gonczarowski (2017), and we keep the term lurker for the redefined concept as an acknowledgment of their work. Because we impose condition (iv), our definition of a lurker is more restrictive than their Definition E.9 of Bade and Gonczarowski (2017): all lurkers in our sense are BG lurkers, but the converse need not hold. On the other hand, our definition of a non-lurker is more permissive: a non-lurker in our usage may not be a BG non-lurker. We include (iv) in the definition of a lurker because it is needed in the construction of our coding algorithm in Step 4 that maps role assignment functions to agent

(2017) show that each BG lurker lurks only one object, each BG-lurked object has only one BG lurker, and at any history, at most two active agents are not BG lurkers. Lemmas 9 and 10 in the Supplementary Appendix show that the same continues to hold for our definition of lurkers.

At any h, we partition the set of active agents as $\mathcal{A}(h) = \mathcal{L}(h) \cup \bar{\mathcal{L}}(h)$. The set $\mathcal{L}(h) = \{\ell_1^h, \ldots, \ell_{\lambda(h)}^h\}$ is the set of lurkers and $\bar{\mathcal{L}}(h)$ is the set of active non-lurkers, where $\lambda(h) = |\mathcal{L}(h)|$ denotes the number of lurkers at h. Let $\mathcal{X}(h)$ denote the set of still-available (unclinched) objects at h, and partition this set as $\mathcal{X}(h) = \mathcal{X}^{\mathcal{L}}(h) \cup \bar{\mathcal{X}}^{\mathcal{L}}(h)$, where $\mathcal{X}^{\mathcal{L}}(h) = \{x_1^h, \ldots, x_{\lambda(h)}^h\}$ is the set of lurked objects and $\bar{\mathcal{X}}^{\mathcal{L}}(h) = \mathcal{X}(h) \smallsetminus \mathcal{X}^{\mathcal{L}}(h)$ is the set of unlurked objects at h. We order the sets so that agent ℓ_m^h lurks objects x_m^h, and if $m' < m$, then lurker $\ell_{m'}^h$ is **older** than lurker ℓ_m^h, in the sense that $\ell_{m'}^h$ first became a lurker for $x_{m'}^h$ at a strict subhistory of the history at which ℓ_m^h became a lurker for x_m^h; we also say that lurker ℓ_m^h is **younger** than lurker $\ell_{m'}^h$. We use the same older and younger comparisons for BG lurkers.

As agents continue to take successive passing actions, the set of lurkers and the set of lurked objects continue to grow, until eventually, we reach a history h where some agent i clinches some object x.[20] By Lemma 13 in the Supplementary Appendix, any agent i who moves at a history h whose immediately preceding action is a passing action is not a lurker. When i clinches at h, this allows us to determine the assignments of all lurkers as follows:

- If $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$, each lurker $\ell_m^h \in \mathcal{L}(h)$ receives her lurked object, x_m^h.
- If $x = x_{m_1}^h$ for some lurked $x_{m_1}^h \in \mathcal{X}^{\mathcal{L}}(h)$, then all older lurkers $\ell_{m'}^h$ for $m' < m_1$ receive their lurked objects $x_{m'}^h$; lurker $\ell_{m_1}^h$, whose lurked object is assigned to i, receives her favorite object from the remaining set of unclinched objects, $\mathcal{X}(h) \smallsetminus \{x_1^h, \ldots, x_{m_1}^h\}$.
  - If $\ell_{m_1}^h$ is assigned an unlurked object, then all remaining lurkers get their lurked objects; if $\ell_{m_1}^h$ is assigned a lurked object $x_{m_2}^h$ for some $m_2 > m_1$, then all older unmatched lurkers ($\ell_{m'}^h$ for $m_1 < m' < m_2$) receive their lurked objects. Lurker $\ell_{m_2}^h$ gets his favorite object from $\mathcal{X}(h) \smallsetminus \{x_1^h, \ldots, x_{m_2}^h\}$.
  - This process is repeated until some lurker $\ell_{\bar{m}}^h$ receives an unlurked object, at which point all remaining unassigned lurkers are assigned their lurked objects.

These assignments are implied by Lemma E.17 in Bade and Gonczarowski (2017) (who show that it is valid under the definition of BG lurkers) and by our Lemma 7, which shows that,

orderings; our coding algorithm treats BG lurkers who do not satisfy (iv) the same as other non-lurkers and differently from how it treats lurkers.

[20] It is immediate that lurker conditions (i)-(iii) continue to hold at each history reached by passing from the history at which agent i became a lurker. That (iv) continues to hold follows from Lemma 12 in the appendix.

at any history, there is at most one BG lurker who is not a lurker and it is the youngest BG lurker. Notice that in the above structure of assignments, there is a unique active agent j who is assigned an unlurked object y; this agent might be the agent i who started the chain of assignments by clinching, or one of the lurkers. Lemmas 10 and 13 in the Supplementary Appendix imply that there might be at most one additional active agent, j', who is neither i nor one of the lurkers. If such a j' exists and $y \in C^{\subsetneq}_{j'}(h)$ then j' receives her favorite object that was neither assigned prior to h nor to other active agents at h.[21]

Now, the above structure of assignments and the millipede reduction theorem of Pycia and Troyan (2023) from Step 2 allows us to assume that our base game Γ is a millipede game that has the following properties:[22]

1. At each history h, there is at most one passing action in $A(h)$; this action, if it exists, is denoted $a^* \in A(h)$. With slight abuse of notation, when the context is clear, we use the symbol a^* to represent the unique passing action at any history h (if such an action exists), and write $h' = (h, \overbrace{a^*, \dots, a^*}^{n \text{ times}})$ to denote that history h' is the superhistory of h that is reached by starting at h and following n passing actions in a row; since there is at most one passing action at any given history, h' is uniquely defined.

2. If i moves at h and $x \in G_i(h)$, then there exists a clinching action $a_x \in A(h)$ that clinches x for i.

3. If i is the unique active agent for whom $P_i(h) = G_i(h)$, then i moves at h.

4. If i moves at h and $P_i(h) = G_i(h)$, then $C_i(h) = P_i(h)$, there there is no passing action at h, and i is not called to move at any $h' \supsetneq h$.

5. Following any clinching action $a' \in A(h)$ at a history h, any lurker at h who is assigned to their lurked object never moves after h, and hence become inactive. Further, at $h' = (h, a')$:

(a) If there are agents who were lurkers at h and are not assigned to their lurked objects, then the oldest such lurker moves at h'. This lurker is offered for clinching

[21] Let y' be the top choice for j' among objects that were neither assigned prior to h nor to other active agents at h. Then j' can at best receive y'. As there is a preference profile of other agents at which they rank y' lowest, making y' impossible for j' would violate Pareto efficiency. Thus y' is possible for j'. At the same time, the payoff guarantee properties of a millipede imply that j' is offered for clinching all objects that were possible but not clinchable for her when j' passed on y. Thus, the footnoted claim follows.

[22] Property 1 is the basic structure of millipedes presented in step 2; we restate it in order to introduce the a^* notation. That, without loss of generality, we can assume properties 2 and 3 is established in the proof of the millipede theorem of Pycia and Troyan (2023). We can assume property 4 because by property 2 and greedy strategies, any passing move at h can be pruned in the manner of Li (2017)'s Pruning Principle for OSP games.

all objects that have not been assigned prior to the move (there is no passing action).

(b) Otherwise, if there exists an agent j' who was active at h and has not yet been assigned an object at h', then j' moves at h' and:[23]

(i) If the object y that was clinched at h has been previously offered to j' (i.e., $y \in C^{\subseteq}_{j'}(h)$), then j' is offered to clinch all remaining unassigned objects.[24]

(ii) If the object y that was clinched at h has not been previously offered to j' (i.e., $y \notin C^{\subseteq}_{j'}(h)$), then j' is offered to clinch at least all objects in $C^{\subseteq}_{j'}(h)$; she may also have other clinching moves and/or a passing move.[25]

(c) If neither 5(a) nor 5(b) hold, then all agents who were active at h have been assigned. If there remain unassigned agents, then one of these agents moves at h' and a continuation game begins among the remaining unassigned agents and objets. Otherwise, the game ends.

We summarize the above discussion in the following lemma, the proof of which can be found in Appendix C.1.

Lemma 2. *Every OSP and Pareto efficient mechanism* (Γ, S) *is equivalent to a millipede mechanism satisfying properties 1-5.*

Remark 1 (*Recursive structure*). Property 5 guarantees that the games we study have a recursive structure: at the first clinching following a (possibly empty) sequence of passes, lurkers are assigned, in order of age, their best possible remaining object. When no further lurkers remain, there may be one remaining active agent, j'. The next move starts a continuation game that is just a smaller Pareto-efficient millipede game among j' and all of the remaining unmatched agents and objects. This continuation game has the same structure described above, and property 5(b) guarantees that j' moves first in this continuation game, and is able to clinch at least the set of objects she could have clinched up until this point in the game (and possibly more).

[23]Note that there can be at most one such agent j', and they are not a lurker.

[24]See footnote 21.

[25]By definition, j' is not a lurker, and so Lemma 12 in the Supplementary Appendix implies that her set of previously clinchable objects $C^{\subseteq}_{j'}(h)$ cannot contain any lurked objects. Since in this case we assume that $y \notin C^{\subseteq}_{j'}(h)$, all of the objects in $C^{\subseteq}_{j'}(h)$ remain unassigned, and thus may be offered to j'.

Step 4: Coding Algorithm

By Lemma 2, we may assume that the mechanism (Γ, S) in Proposition 2 is a millipede mechanism satisfying properties 1-5. At the core of the remainder of the proof of Proposition 2 is the construction of a bijection between role assignment functions for the permuted millipede mechanisms that make up the symmetrization of (Γ, S) and serial dictatorship orderings such that the outcomes of the permuted millipede and permuted serial dictatorship are exactly the same. More formally, let Ord denote the set of total linear orders over the set of agents $\mathcal{N}$. Random Priority draws an agent ordering uniformly at random from Ord, and thus the probability of any particular allocation μ is just the number of agent orderings such that a serial dictatorship under such an ordering results in μ, divided by $N!$, the total number of possible agent orderings. Similarly, in the symmetrization of (Γ, S), the probability of μ is the number of role assignment functions $\sigma \in \Sigma$ such that the permuted mechanism $(\Gamma_\sigma, S_\sigma)$ results in μ. Thus, if we can find a bijection $f : \Sigma \to Ord$ such that for every $\sigma \in \Sigma$, the permuted mechanism $(\Gamma_\sigma, S_\sigma)$ results in the same allocation as a serial dictatorship under agent ordering $f_\sigma(1), \dots, f_\sigma(N)$—where $f_\sigma(j)$ denote the j^{th} ranked agent under the agent ordering f_σ—the distribution over allocations in the symmetrized millipede will have been shown to be the same as the distribution over allocations in Random Priority, which will prove Proposition 2 (and hence, also Theorem 1).

The rest of the proof is devoted to constructing the necessary bijection f. In Step 4 here, we introduce a coding algorithm that takes a continuation game under a role assignment function Γ_σ and maps (or "codes") it to a partial ordering of the agents, denoted $\succ$. This partial ordering may include ties, and Steps 5 and 6 below show how to take these partial orderings and break ties to obtain the full bijection $f : \Sigma \to Ord$.

The intuitive idea behind constructing $\succ$ is as follows. We start by finding the first agent to clinch some object x after a (possibly empty) series of passes at some history h. This induces a chain of assignments of the active agents $\mathcal{A}(h)$ as in Step 3. We create $\succ$ by ordering agents who receive lurked objects in order of the "age" of the object they received, i.e., the first agent in the ordering is the agent who receives the object that became lurked first, the second is the agent who received the object that became lurked second, and so forth (note that this is different from ordering lurkers by their age, as a lurker may end up receiving a different object than the one she lurked).

After ordering the agents who receive lurked objects, there are at most 2 active agents who have yet to be coded, one of whom has clinched an unlurked object, say y;[26] if y was previously offered to the remaining active agent, then we add both remaining agents to the

[26] This is because, as shown in Step 3, there can be at most two active non-lurkers at any given point.

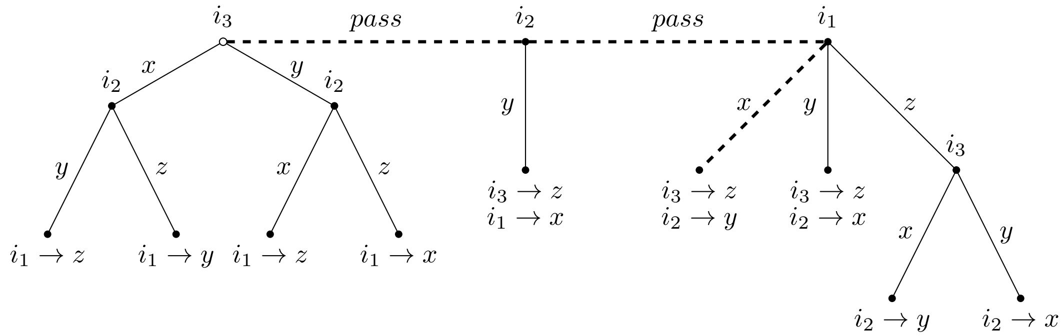

Figure 3: The example from Section 4 under the role assignment $\sigma''(r_1) = i_3$, $\sigma''(r_2) = i_2$, and $\sigma''(r_3) = i_1$ and preferences $\succ_{i_1}: x, y, z$, $\succ_{i_2}: x, y, z$, $\succ_{i_3}: z, y, x$. The dashed lines show the path of play.

order without distinguishing between them, i.e., these two agents tie; if y was not previously offered to the other remaining active agent, then we only add to the ordering the agent who clinched y. The other active agent (if such an agent exists) will be added in a later step triggered by a later clinching; at the beginning of the next segment this agent is still active with the carried over "endowment" of previously clinchable objects, $C_j^{\subseteq}(h)$ (cf. Remark 1). After clearing this first segment of agents, we continue along the game path and find the next agent to clinch an object, and repeat.

To illustrate, consider again the game from Section 4 under the role assignment function $\sigma''(r_1) = i_3$, $\sigma''(r_2) = i_2$, and $\sigma''(r_3) = i_1$. The path of play of the game under this role assignment is shown by the dashed lines in Figure 3. Agent i_3 moves first, and her top choice is z. She is offered to clinch only x and y at her first move, while z is possible later in the game if she passes, and so her obviously dominant strategy is to pass. Upon passing, i_3 has now been offered all objects that are possible for her in the game except for one (object z), and thus i_3 becomes a lurker, and z becomes a lurked object. We follow the dashed line until we find the first agent to clinch, which in this case is agent i_1, who clinches object x. This triggers the ordering of the currently active agents—which in this case is all of the agents—and orders them by first ordering agents who receive lurked objects according to the age of the lurked object they receive. Thus, agent i_3 is ordered first in the corresponding serial dictatorship, because when i_1 clinches the (unlurked) object x, i_3 receives the lurked object z, followed by i_1, and then i_2; in other words, $f_{\sigma''}(1) = i_3$, $f_{\sigma''}(2) = i_1$, and $f_{\sigma''}(3) = i_2$.[27]

[27]How to order the (up to two) active non-lurkers is a subtlety that we discuss when providing the full algorithm below.

We now present the formal definition of the coding algorithm just described.

Coding Algorithm. Consider a permuted mechanism $(\Gamma_\sigma, S_\sigma)$, and take the game path from the root node $h_\varnothing$ to a terminal node $\bar{h}$ when agents follow the strategy profile S_σ. Each step k of the algorithm below produces a partial ordering $\tilde{\succ}^k$ on the set of agents who are processed in step k. At the end of the final step K, we concatenate the K components to produce $\succ$, the final coding on the set of all agents $\mathcal{N}$.

Step 1 Find the first object to be clinched along the game path, say x^1 at history h^1 by agent i^1.[28] Let $\mathcal{L}(h^1) = \{\ell_1, \ldots, \ell_{\lambda(h^1)}\}$ be the set of lurkers, and $\mathcal{X}^{\mathcal{L}}(h^1) = \{x_1, \ldots, x_{\lambda(h^1)}\}$ be the set of lurked objects at h^1, where x_k is the k-th object to become lurked and ℓ_k the lurker of this object; if these sets are empty, skip directly to step 1.2 below.

1. For $x_k \in \mathcal{X}^{\mathcal{L}}(h^1)$, let i_{x_k} be the agent who receives x_k at $\bar{h}$.[29]

2. Let $j \in \mathcal{L}(h^1) \cup \{i^1\}$ be the unique agent that is not one of the agents $i_{x_1}, ..., i_{x_{\lambda(h^1)}}$ from step 1.1. Because we restricted attention to millipedes satisfying properties 1-5 above, j receives an unlurked object $y \in \bar{\mathcal{X}}^{\mathcal{L}}(h^1)$ and there may be at most one active agent $j' \in \mathcal{A}(h^1) \setminus (\mathcal{L}(h^1) \cup \{i^1\})$.

   (a) If such a j' exists and $y \in C^{\subseteq}_{j'}(h^1)$, then define $\tilde{\succ}^1$ as:

   $$i_{x_1} \tilde{\succ}^1 i_{x_2} \tilde{\succ}^1 \cdots \tilde{\succ}^1 i_{x_{\lambda(h^1)}} \tilde{\succ}^1 \{j, j'\}$$

   (b) Otherwise, define $\tilde{\succ}^1$ as

   $$i_{x_1} \tilde{\succ}^1 i_{x_2} \tilde{\succ}^1 \cdots \tilde{\succ}^1 i_{x_{\lambda(h^1)}} \tilde{\succ}^1 j$$

   In particular, if j' exists and $y \notin C^{\subseteq}_{j'}(h^1)$ then we do not yet order agent j'.

Step k Find the first object to be clinched along the game path by an agent that has not yet been ordered, say x^k at history h^k by agent i^k. Let $\mathcal{L}(h^k) = \{\ell_1, \ldots, \ell_{\lambda(h^k)}\}$ be the set of lurkers, and $\mathcal{X}^{\mathcal{L}}(h^k) = \{x_1, \ldots, x_{\lambda(h^k)}\}$ be the set of lurked objects, and carry out a procedure analogous to that from step 1 to produce the step k order $\tilde{\succ}^k$.

This produces a collection of codings $(\tilde{\succ}^1, \ldots, \tilde{\succ}^K)$, where each $\tilde{\succ}^k$ is a partial order on the agents processed in step k. We then create the final $\succ$ in the natural way: for any two agents

[28] That is, $i_{h^1} = i^1$, and i^1 selects a clinching action $a_{x^1} \in A(h^1)$ that clinches x^1. By Lemma 13, $i^1 \notin \mathcal{L}(h^1)$. Notice the difference between superscript in x^1, which refers to the step of the algorithm, and the subscripts in lurked objects, which refer to the order in which they were lurked. In the notation for lurkers $\ell_k^{h^1}$ and lurked objects $x_k^{h^1}$ we suppress the history superscript.

[29] Note that i_{x_k} is not necessarily the agent who lurks x_k at h^1.

i, j who were processed in the same step k, $i \succ j$ if and only if $i \tilde{\succ}^k j$. For any two agents i, j processed in different steps k and k' respectively, where $k < k'$, we order $i \succ j$.

The output of the coding algorithm is a partial order, $\succ$, on $\mathcal{N}$, the set of agents. If $i \succ j$, we say that i **precedes** j. If there are two agents i and j such that $i \nsucc j$ and $j \nsucc i$, then we say i and j **tie** under $\succ$. We also use the notation $i \succ \{j, k\} \succ \ell$ to denote that i precedes j and k, the latter two agents tie, and in turn these two agents precede ℓ. Note that by construction, all ties are of size at most 2, and agents can only tie if they are processed in the same step of the algorithm.

Remark 2. The coding algorithm divides the game path from the root to the terminal node into a series of K steps. At the end of each coding step, there may be one agent, say j', who was active during the step, and was not coded in the step. When this occurs, at the the initial history of the continuation game that begins after all agents from the previous step have been assigned their objects, agent j' is called to move, and is offered the to clinch everything that she has been offered to clinch previously in the game (and might have other moves). The next step of the coding algorithm is initiated the first time an agent clinches an object in this continuation game, and the process is repeated. This recursive structure is further discussed in Remark 1.

Each role assignment function σ induces a permuted mechanism $(\Gamma_\sigma, S_\sigma)$, and each permuted mechanism has an associated coding $\succ_\sigma$ obtained via the applying the coding algorithm to the mechanism $(\Gamma_\sigma, S_\sigma)$. This results in a collection of $N!$ codings $(\succ_\sigma)_{\sigma \in \Sigma}$. Codings do not map directly to serial dictatorship orderings, because some agents may tie. In the remainder of the proof, we show that (i) no matter how these ties are broken, the resulting serial dictatorship results in the same allocation as the original game $(\Gamma_\sigma, S_\sigma)$ (Step 5) and (ii) it is possible to break ties across all of the $N!$ codings in such a way that the resulting mapping from permuted games to serial dictatorship orderings is a bijection (Step 6).

Step 5: Same Allocations

Take a role assignment function σ and the resulting coding $\succ_\sigma$. We say that a total ordering of the agents f_σ is **consistent** with $\succ_\sigma$ if, for all j, j': $j \succ_\sigma j'$ implies $f_\sigma^{-1}(j) < f_\sigma^{-1}(j')$. In other words, given some coding $\succ_\sigma$, total order f_σ is consistent if there is some possible way to break the ties in $\succ_\sigma$ that delivers f_σ. We further say that f_σ is **consistent with $\succ_\sigma$ on an initial segment till an agent** i if, for all j, j' that either precede i or tie with i, if $j \succ_\sigma j'$ then $f_\sigma^{-1}(j) < f_\sigma^{-1}(j')$.

Lemma 3. *For any agent i and any total order f_σ consistent with $\succ_\sigma$ on an initial segment till i, the allocation of agents who precede or tie with i under the serial dictatorship with* *agent ordering f_σ is the same as their allocation in Γ_σ. In particular, given two games Γ_A and Γ_B played under role assignment functions σ_A and σ_B, respectively, if $\succ_A = \succ_B$, then Γ_A and Γ_B end with the same final allocations to all agents.*

We prove this lemma in Appendix C.2. Given $\succ_\sigma$, any way of breaking the ties (if any ties exist) between agents produces a total order f_σ that is consistent with $\succ_\sigma$. Thus, by Lemma 3, no matter how ties are broken, the mechanism $(\Gamma_\sigma, S_\sigma)$ ends with the same allocation as the serial dictatorship with agent ordering f_σ.

Step 6: Bijectivity

Finally, we show that it is possible to break the ties in $(\succ_\sigma)_{\sigma\in\Sigma}$ in such a way to produce a mapping $f : \Sigma \to Ord$ that is a bijection. We prove bijectivity using two lemmas—Lemmas 4 and 5—on the properties of the partial orders produced by the coding algorithm applied to games with different role assignments. The proofs of these lemmas can be found in Appendix C.2.

Let h_A^k be the history that initiates step k of the coding algorithm when it is applied to game Γ_A. For instance, $h_A^1 = (h_\varnothing, a^*, \ldots, a^*)$ is a history following a (possibly empty) sequence of passes such that agent $i_{h_A^1}$ moves at h_A^1 and is the first agent to clinch in the game. This induces a chain of assignments of the agents in $\mathcal{L}(h_A^1) \cup \{i_{h_A^1}\}$, plus possibly one other active non-lurker at h_A^1, as given in the description of millipede mechanisms with lurkers. History $h_A^2 \supsetneq h_A^1$ is then the next time along the game path that an agent who was not ordered in step 1 of the coding algorithm clinches an object, etc. Define h_B^k analogously, and let K_A and K_B be the total number of steps in the coding algorithm when applied to games Γ_A and Γ_B, respectively.

Lemma 4. *Let σ_A and σ_B be two role assignment functions, and Γ_A and Γ_B their associated games. Let $\succ_A^k$ be the initial segment of $\succ_A$ consisting of agents ordered up to and including step k of the coding algorithm in game Γ_A. If ordering $\succ_A^k$ equals to an initial segment of $\succ_B$, then $h_A^{k'} = h_B^{k'}$ for all $k' = 1, \ldots, k$ and $\sigma_A^{-1}(i) = \sigma_B^{-1}(i)$ for all agents i who are coded up to step k. In particular, if $\succ_A = \succ_B$, then $h_A^k = h_B^k$ for all k, $K_A = K_B$, and $\sigma_A^{-1}(i) = \sigma_B^{-1}(i)$ for all $i \in \mathcal{N}$.*

The proof of Lemma 4 can be found in Appendix C.2. The lemma shows that the mapping from role assignments to codings (partial orderings) is injective. As there may be ties in some codings, what remains to show is that it is possible to break the ties in all codings in such a way that preserves the injectivity. The next lemma provides the key tool needed to do this.

We write $j_1 \cdots j_P \succ i \succ j \succ \cdots$ when $\succ$ ranks $j_1, ..., j_P$ first, possibly with ties; ranks i immediately (and strictly) after, and then ranks j immediately (and strictly) after i. We write $j_1 \cdots j_P \succ i \succ \{j, k\} \cdots$ when $\succ$ ranks $j_1, ..., j_P$ first, possibly with ties, and then ranks the tie $\{j, k\}$ immediately after. We write $j_1 \cdots j_P \succ i \succ j \cdots$ to denote the case in which either of the two previously possibilities may hold (i.e., j may or may not tie with some other agent k).

Lemma 5. *Assume that there exist positive integers $n, m \geq 1$ and two sequences of role assignment functions, $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n, \sigma_{n+1}\}$ and $\Sigma' = \{\sigma'_1, \sigma'_2, \sigma'_3, \ldots, \sigma'_m, \sigma'_{m+1}\}$ such that $\sigma_1 = \sigma'_1$ and the resulting codings are:*

$$\begin{aligned}
\textit{Sequence } \Sigma\textit{:}\quad & j_1 \cdots j_P \succ_1 \{i, k_1\} \succ_1 \cdots \\
& j_1 \cdots j_P \succ_2 k_1 \succ_2 \{i, k_2\} \succ_2 \cdots \\
& j_1 \cdots j_P \succ_3 k_1 \succ_3 k_2 \succ_3 \{i, k_3\} \succ_3 \cdots \\
& \vdots \\
& j_1 \cdots j_P \succ_n k_1 \succ_n k_2 \succ_n k_3 \succ_n \cdots \succ_n k_{n-1} \succ_n \{i, k_n\} \succ_n \cdots \\
& j_1 \cdots j_P \succ_{n+1} k_1 \succ_{n+1} k_2 \succ_{n+1} k_3 \succ_{n+1} \cdots \succ_{n+1} k_{n-1} \succ_{n+1} k_n \succ_{n+1} i \cdots
\end{aligned}$$

$$\begin{aligned}
\textit{Sequence } \Sigma'\textit{:}\quad & j_1 \cdots j_P \succ'_1 \{i, k_1\} \succ'_1 \cdots \\
& j_1 \cdots j_P \succ'_2 i \succ'_2 \{k_1, k'_2\} \succ'_2 \cdots \\
& j_1 \cdots j_P \succ'_3 i \succ'_3 k'_2 \succ'_3 \{k_1, k'_3\} \succ'_3 \cdots \\
& \vdots \\
& j_1 \cdots j_P \succ'_m i \succ'_m k'_2 \succ'_m k'_3 \succ'_n \cdots \succ'_m k'_{m-1} \succ'_m \{k_1, k'_m\} \succ'_m \cdots \\
& j_1 \cdots j_P \succ'_{m+1} i \succ'_{m+1} k'_2 \succ'_{m+1} k'_3 \succ'_{m+1} \cdots \succ'_{m+1} k'_{m-1} \succ'_{m+1} k'_m \succ'_{m+1} k_1 \cdots
\end{aligned}$$

where the partial order on $j_1 \cdots j_P$ is the same in all above codings. Then, one of the following must hold:

(I) In $\succ_{n+1}$, agent i ties with some agent k_{n+1}; or

(II) In $\succ'_{m+1}$, agent k_1 ties with some agent k'_{m+1}.

Notice the symmetry between sequences Σ and Σ', to which we also refer as **arms**. They have the following properties:

- Each arm starts with the same role assignment and codings, i.e., $\sigma_1 = \sigma'_1$ and $\succ_1 = \succ'_1$.
- In arm Σ, every subsequent coding ranks k_1 strictly ahead of all other agents (besides the j_p's), while in Σ', every subsequent coding ranks i ahead of all other agents (besides the j_p's).

- Within arm Σ, the only difference from ℓ to $\ell+1$ is that the agent k_ℓ who tied with i in $\succ_\ell$ is now ranked strictly above i, with i now tied with a different agent, $k_{\ell+1}$ (except for $\succ_{n+1}$, in which case i is ranked next, but may or may not tie with another agent). A similar remark applies to Σ'.

- Across the two arms, it is possible that some or all of the agents $k_2, \ldots, k_n$ are the same as the agents $k'_2, \ldots, k'_m$, though it is not necessarily assumed. We also do not require $m = n$.

The proof of Lemma 5 can be found in Appendix C.2. By Lemma 4, the mapping from role assignments σ to codings $\succ_\sigma$ generated by the coding algorithm is injective. Using Lemma 5, we break the ties to create from each $\succ_\sigma$ a consistent total order f_σ in a way that preserves the injectivity. We proceed with the following two tie-breaking steps:

Tie-Breaking Step 1. For all role assignments σ, in coding $\succ_\sigma$ we break any tie $\{i, k_1\}$ so that $i \succ_\sigma k_1$ if and only if, in the original set of codings, there is an arm of the form Σ from Lemma 5 in which the second coding starts with $j_1 \cdots j_P \succ k_1 \succ$ for some $j_1, ..., j_P \neq i$ and in the last coding agent i does not tie; analogously, we break any tie $\{i, k_1\}$ so that $k_1 \succ_\sigma i$ if and only if there is an arm of the form Σ' from Lemma 5 in which the second coding starts with $j_1 \cdots j_P \succ i \succ$ for some $j_1, ..., j_P \neq k_1$ and in the last coding agent k_1 does not tie.

Lemma 5 guarantees that the tie-breaking procedure just described is well-defined, in the sense that it will produce no conflicts in how to break a given tie. In particular, if there is an arm that forces a tie-break such that, say, $i \succ_\sigma k_1$, then Lemma 5 implies that there cannot be an arm that forces a tie-break such that $k_1 \succ_\sigma i$.

Lemma 5 further implies that, if $\succ_\sigma$ starts with $j_1 \cdots j_P \succ_1 \{i, k_1\}$ and we broke the tie $i \succ_{\sigma'} k_1$ (the other fully case is symmetric) then (i) no other coding starts with $j_1 \cdots j_P \succ_{\sigma'} i \succ_{\sigma'} k_1 \succ_{\sigma'}$ and (ii) no other coding starts with $j_1 \cdots j_P \succ_{\sigma'} i \succ_{\sigma'} \{k_1, k_2\}$ for some k_2 and the above tie-breaking procedure breaks the tie so that $k_1 \succ_{\sigma'} k_2$. By applying observations (i) and (ii) to tie breaks, starting at the end of each coding, we infer that the resulting mapping from permutations to partially tie-broken codings remain injective.

Importantly, the above tie-breaking procedure did not create any new ties that could be broken as in Tie-Breaking Step 1. Indeed, if, say, a broken tie $\{i, k_\ell\}$ creates a new arm that would allow a tie break at $\{i, k_1\}$ then, the structure of the arms in the statement of Lemma 5 implies that before the former tie-break, the latter tie is broken by the union of the arm from $\{i, k_1\}$ till $\{i, k_\ell\}$ and the arm that allowed us to break the tie $\{i, k_\ell\}$.

Tie-Breaking Step 2. After the end of Tie-Breaking Step 1, there may still be ties remaining. If there are no ties remaining, then Step 1 has already produced an injective mapping from codings to consistent total orderings, and we skip to the last paragraph of the proof. If there are ties remaining, then it must be that all arms that begin with these ties end with the last agent being in a tie. We then proceed recursively. We look over all ties in the partial orders created in Tie-Breaking Step 1 across all permutations σ and find a tie—say $\{i, k_1\}$—that has the largest number of agents ranked above it. If such a tie $\{i, k_1\}$ exists then we break this tie arbitrarily. Because we broke only one such tie, the "at least one tie" structure of arms stated in Lemma 5 holds for the resulting set of partial orderings. We can thus perform the same tie breaking as was done in Tie-Breaking Step 1 and, as above, the resulting mapping from permutations to partially tie-broken codings remain injective and, in all remaining ties, all arms end with the last agent being in a tie.

We repeat the above tie-breaking procedure iteratively: we look over all ties in partial orders created so far in Tie-Breaking Step 2, across all permutations σ, and again find a tie that has the largest number of agents ranked above it and repeat the Step-2 tie break procedure above. We proceed in this way till all ties are broken and we have constructed an injective mapping from permutations to total orderings.

As the resulting total orderings are created by breaking ties in the original codings, the complete orderings are consistent with the original codings. Hence we created an injective mapping from permutations to total orderings that are consistent with codings. In this way we obtain an injection from role assignments σ to serial dictatorships with orders f_σ. Because in this injection the domain of role assignments σ and the range of serial dictatorship orderings f_σ are finite and have equal size, this injection is a bijection.

Step 7: Recap

To recap, we have shown the following:

1. Every Pareto-efficient, OSP mechanism (Γ, S) is equivalent to a (perfect-information) millipede mechanism satisfying properties 1-5 in which Nature moves once (if at all) as the first mover (Lemma 2).

2. For any millipede mechanism satisfying properties 1-5, there is a bijection f between role assignment functions and serial dictatorship orderings such that the final allocation of the permuted mechanism $(\Gamma_\sigma, S_\sigma)$ results in the same final allocation as a serial dictatorship using the agent ordering f_σ (Lemmas 3, 4 and 5).

3. Point (2) implies that the symmetrization of (Γ, S) is equivalent to Random Priority (see the argument in the first paragraph of Step 4).

4. Since the symmetrization of every OSP, Pareto-efficient and deterministic perfect-information mechanism (Γ, S) is equivalent to Random Priority, then every symmetric,

OSP, and Pareto-efficient mechanism is equivalent to Random Priority (Lemma 1).

This completes the proof of Theorem 1.

C Proofs of Auxiliary Results

C.1 Proof of Lemma 2

Properties 1-4 follow from the millipede theorem of Pycia and Troyan (2023), as explained in footnote 22. Thus, we focus on establishing property 5. We start with two results—Lemmas 6 and 7—on the connection between lurkers and BG lurkers.

Given a subset of objects $X' \subseteq \mathcal{X}$ and a preference ranking for agent i, $\succ_i$, let $Top(\succ_i, X')$ be the highest $\succ_i$ −ranked object in the set X'. Given some history h, let h' be the maximal superhistory of the form $h' = (h, a^*, \ldots, a^*)$. Following Bade and Gonczarowski (2017) (thereafter BG), we call h' a **terminating history**, and the agent who moves at h' a **terminator**. The terminating history provides an upper bound on the number of passes that can be taken in a row, i.e., at the terminating history, the agent that moves has only clinching actions. Note that there may be many terminating histories along the full game-path, and that the definition of the terminating history is only a function of the game form Γ, and is independent of the lurker definition that is considered.

Lemma 6. *Let h be a history such that there is an active BG non-lurker j such that $x \in C_j^{\subseteq}(h)$ for some object x that is BG-lurked at h. Then, h is a terminating history, and j is the terminator.*

Proof. Let $\bar{h}$ be the largest proper subhistory of h, $\bar{h} \subsetneq h$, such that the set of BG-lurked objects at $\bar{h}$ is empty. It is sufficient to show that for the smallest superhistory $h \supseteq \bar{h}$ that satisfies the statement of the lemma, h is a terminating history. Define h' such that $h = (h', a^*)$, i.e., h' is the immediate predecessor of h; such a predecessor exists because there are BG-lurked objects at h. By the supposition that h is the smallest superhistory of $\bar{h}$ that satisfies the statement of the lemma, we have that either (i) x is not BG-lurked at h' or (ii) x is BG-lurked at h', but $x \notin C_j^{\subseteq}(h')$.

For case (i), x first becomes BG-lurked at h. Let ℓ be the agent that BG-lurks x at h, and notice that it must be ℓ that moves at h'.[30] This implies that both j and ℓ are active at

[30] Assume not, i.e., assume some $k \neq \ell$ moved at h'. Then, the maximal strict subhistory of h where ℓ moves is some $h'' \subsetneq h'$, and by definition of a BG lurker (i) $P_\ell(h) \neq G_\ell(h)$, (ii) $x \in P_\ell(h'')$, and (iii) $C_\ell^{\subseteq}(h'') = P_\ell(h'') \setminus \{x\}$ hold.

This implies that ℓ is already a lurker for x at h': since $h'' \subsetneq h'$, (i) and (ii) continue to hold at h', while for for (iii), if $P_\ell(h') = G_\ell(h')$, then, since the game is a millipede game that satisfies properties 1-4, there is no passing action at h'. This contradicts that x is not lurked at h'.

h', and neither are BG lurkers. Because there can be at most two active BG non-lurkers at any history, all other active agents at h' are BG lurkers. Now, consider h. At h, $x \in C_j^{\subseteq}(h)$, and so Lemma E.14 of BG implies $P_j(h) = G_j(h)$. Further, j is the unique active agent such that $P_j(h) = G_j(h)$.[31] Thus, by properties 3 and 4, j moves at h and $P_i(h) = G_i(h) = C_i(h)$, and there is no passing action at h. Thus, h is the terminating history.

For case (ii), $x \notin C_j^{\subsetneq}(h')$ but $x \in C_j^{\subseteq}(h)$ implies that j must move at h, and $x \in C_j(h)$. By BG Lemma E.14, $P_j(h) = G_j(h)$. By property 4, $P_i(h) = G_i(h) = C_i(h)$, and there is no passing action at h. Thus, h is the terminating history. ∎

Lemma 7. *At any h, there is at most one BG lurker that is not a lurker. If such an agent i exists, then i is the youngest BG lurker at h, and h is a terminating history. Further, i does not move at h.*

Proof. Consider a history $\bar{h}$ at which there are no BG lurkers (and thus, also no lurkers). Because at each history, only one new BG lurker can be added, it is sufficient to show that if $h \supsetneq \bar{h}$ is the smallest superhistory of $\bar{h}$ such that there is a BG lurker that is not a lurker, then h is a terminating history. Thus, let $h = (h', a^*)$, where at h', all BG lurkers are lurkers, but at h, there is a BG lurker that is not a lurker; label this agent i. Then, it must be that i first becomes a BG lurker at h, and at h, point (iv) in the definition of a lurker fails, i.e., there is some active BG non-lurker $j \neq i$ that has been previously offered to clinch the object that i BG lurks. Lemma 6 implies that h is the terminating history, and agent j moves at h. Since no new agent has entered the game at h, and all agents other than j are BG lurkers at h, there is only one BG lurker that is not a lurker. The rest of the statements follow easily from the fact that h is a terminating history. ∎

The next four lemmas are analogues of statements derived for BG lurkers in BG; we give the analogous BG lemmas in parentheses. Recall that $\mathcal{L}(h)$ and $\mathcal{X}^{\mathcal{L}}(h)$ are the sets of lurkers and lurked objects, respectively, at history h. Let $\mathcal{L}^{BG}(h)$ and $\mathcal{X}^{\mathcal{L},BG}(h)$ denote the sets of BG lurkers and BG-lurked objects. Notice that $\mathcal{L}(h) \subseteq \mathcal{L}^{BG}(h)$ and $\mathcal{X}^{\mathcal{L}}(h) \subseteq \mathcal{X}^{\mathcal{L},BG}(h)$, by definition. Further, by Lemmas 6 and 7, if $\mathcal{L}(h) \subsetneq \mathcal{L}^{BG}(h) = \{\ell_1, \ldots, \ell_{\lambda^{BG}(h)}\}$, then $\mathcal{L}(h) = \mathcal{L}^{BG}(h) \smallsetminus \{\ell_{\lambda^{BG}(h')}\}$, where $\ell_{\lambda^{BG}(h')}$ is the youngest BG lurker. Similarly, if $\mathcal{X}^{\mathcal{L}}(h) \subsetneq \mathcal{X}^{\mathcal{L},BG}(h) = \{x_1, \ldots, x_{\lambda(h)}\}$ then $\mathcal{X}^{\mathcal{L}}(h) = \mathcal{X}^{\mathcal{L},BG}(h) \smallsetminus \{x_{\lambda^{BG}(h)}\}$, where $x_{\lambda^{BG}(h)}$ is the youngest BG-lurked object.

[31] For any active lurker ℓ at h, $P_\ell(h) \neq G_\ell(h)$ by definition. The only other possibility is that some k becomes active at h, and is such that $P_k(h) = G_k(h)$. If this is the case, by BG Lemma E.11, all BG-unlurked objects are possible for k at h. If $P_k(h) = G_k(h)$, then she can clinch any BG-unlurked object at h, by property 4. Consider k clinching some BG-unlurked object y. By BG Lemma E.17, all BG lurkers at h are assigned their BG lurked objects, and so no BG-lurked object is in $G_j(h)$. But, y was arbitrary, and so no BG-unlurked object is in $G_j(h)$ either, and so $G_j(h)$ is empty, which contradicts that $P_j(h) = G_j(h)$.

Lemma 8. *(BG Lemma E.11) If agent i is active at h, then $\bar{\mathcal{X}}^{\mathcal{L}}(h) \subseteq P_i(h) \cup C_i^{\subsetneq}(h)$. If $i \in \mathcal{L}(h)$, then $\bar{\mathcal{X}}^{\mathcal{L}}(h) \subseteq C_i^{\subsetneq}(h)$.*

Proof. For the first part, for any $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ that is also BG-unlurked, the statement follows from BG Lemma E.11. So, consider some $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ but $x \in \mathcal{X}^{\mathcal{L},BG}(h)$. As shown above, there is only one such object, and it is $x = x_{\lambda(h)}$, the youngest lurked object at h. Further, by Lemma 7, this condition only obtains when h is a terminating history, and the active agents at h are $\ell_1, \ldots, \ell_{\lambda(h)}, j$ where: $\ell_1, \ldots, \ell_{\lambda(h)-1}$ are both lurkers and BG lurkers, $\ell_{\lambda(h)}$ is a BG lurker but not a lurker, and j is the terminator (and neither a lurker nor a BG lurker). By BG Lemma E.16, $x_{\lambda(h)} \in P_{\ell'}(h)$ for all $\ell' \in \{\ell_1, \ldots, \ell_{\lambda(h)}\}$, while by BG Lemma E.18, $x_{\lambda(h)} \in C_j^{\subsetneq}(h)$.

The second part follows from the first part and the definition of a lurker. ■

Lemma 8 has the following corollary, which will be useful in the proof constructing the bijection between role assignments and SD orderings later.

Corollary 3. *If, at history h, agent i clinches $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ that is unlurked at h, then $x = Top(\succ_i, \bar{\mathcal{X}}^{\mathcal{L}}(h))$.*

Proof. By Lemma 8, all unlurked objects have either been clinchable at some subhistory of h, or are still possible. Thus, if $x \neq Top(\succ_i, \bar{\mathcal{X}}^{\mathcal{L}}(h))$, it would not be obviously dominant for agent i to clinch $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ at h, a contradiction. ■

Lemma 9. *(BG Lemma E.16) Let $\mathcal{L}(h) = \{\ell_1^h, \ldots, \ell_{\lambda(h)}^h\}$ be the set of lurkers at h and $\mathcal{X}^{\mathcal{L}}(h) = \{x_1^h, \ldots, x_{\lambda(h)}^h\}$, with ℓ_1^h lurking x_1^h, ℓ_2^h lurking x_2^h, etc., where $m < m'$ if and only if ℓ_m^h became a lurker at a strict subhistory of the history at which $\ell_{m'}^h$ became a lurker. Then,*

1. *$x_1^h, \ldots, x_{\lambda(h)}^h$ are all distinct objects.*
2. *For all $m = 1, \ldots \lambda(h)$, $P_{\ell_m^h}(h) = \mathcal{X}(h) \setminus \{x_1^h, \ldots, x_{m-1}^h\}$.*

Proof. Because any lurker is a BG lurker, and the same applies to lurked objects, this is immediate from BG Lemma E.16. ■

Lemma 10. *(BG Lemma E.19) For all h, $|\bar{\mathcal{L}}(h)| \leq 2$.*

Proof. By BG Lemma E.19, there can be at most two BG non-lurkers at h. If there exists a non-lurker that is not a BG non-lurker, by Lemmas 6 and 7, all active agents except for one are BG lurkers, and at most one BG lurker is a non-lurker. Thus, there are at most two non-lurkers at h. ■

Lemma 11. *(BG Lemma E.18, E.20) Let h be a history with lurked objects and let $i_{h'} = t$ be the agent who moves at the maximal superhistory of the form $h' = (h, a^*, \dots, a^*)$. Then:*

(i) Agent t is not a lurker at h.

(ii) $C_t^{\subsetneq}(h') = \mathcal{X}(h)$.

(iii) If $i_h \neq t$, then $C_{i_h}(h) \cap C_t^{\subsetneq}(h) = \varnothing$.

(iv) If $x_\ell \in P_j(h)$ for some non-lurker j and lurked object $x_\ell \in \mathcal{X}^{\mathcal{L}}(h)$, then $j = t$.

(v) $C_t^{\subsetneq}(h') = \mathcal{X}(h)$.

Proof. Notice first that parts (ii), (iii), and (v) do not make any reference to lurkers or lurked objects, and thus these parts follow immediately from the corresponding statements in BG Lemma E.18. BG Lemma E.18 part (i) says that agent t is not a BG lurker, and thus, agent t is not a lurker either, which shows part (i). What remains is to show part (iv). For all $h \subsetneq h'$, any non-lurker is also a BG non-lurker by Lemmas 6 and 7, and any lurked object is also a BG lurked object, and so the result follows from the corresponding lemma of BG. Thus, consider h'. By Lemma 6 and Lemma 7, at h', either $\mathcal{L}^{BG}(h') = \mathcal{L}(h')$ or $\mathcal{L}(h') = \mathcal{L}^{BG}(h') \setminus \{\ell_{\lambda^{BG}(h')}\}$. Similarly, either $\mathcal{X}^{\mathcal{L},BG}(h) = \mathcal{X}^{\mathcal{L}}(h)$ or $\mathcal{X}^{\mathcal{L}}(h') = \mathcal{X}^{\mathcal{L},BG}(h') \setminus \{x_{\lambda^{BG}(h)}\}$. If j is a BG non-lurker, then the result is immediate from the corresponding lemma of BG. It remains to consider j who is a non-lurker but a BG lurker. By Lemma 7, j is a BG lurker for $x_{\lambda^{BG}(h')}$. Notice that $x_{\lambda^{BG}(h')}$ is not lurked at h' (though it is BG-lurked). Thus, the lurked objects at h' are $\mathcal{X}^{\mathcal{L}}(h') = \{x_1, \dots, x_{\lambda^{BG}(h')-1}\}$. By Lemma E.16 from BG , $P_j(h') = \mathcal{X}(h') \setminus \{x_1, \dots, x_{\lambda^{BG}(h')-1}\}$; in other words, for any $x \in \mathcal{X}^{\mathcal{L}}(h')$, we have $x \notin P_j(h')$, and so the statement holds vacuously. ∎

We finish with three additional lemmas, Lemmas 12-14.

Lemma 12. *If $i \in \bar{\mathcal{L}}(h)$ and $x_\ell \in C_i^{\subsetneq}(h)$ for some $x_\ell \in \mathcal{X}^{\mathcal{L}}(h)$, then $i_h = i$, $P_i(h) = G_i(h) = C_i(h)$, and there is no passing action at h (that is, h is a terminating history).*

Proof. If x_ℓ is lurked at h then x_ℓ is BG-lurked at h; thus if i is a BG non-lurker at h, then the result follows from Lemma 6. So, assume that i is a non-lurker that is a BG lurker at h. We claim that for any lurked object $x_\ell \in \mathcal{X}^{\mathcal{L}}(h)$, we have $x_\ell \notin C_i^{\subsetneq}(h)$, and so the result holds vacuously. To show it, let h' be such that $h = (h', a^*)$, i.e., h' is the immediate predecessor of h. By Lemma 7, h must be a terminating history, agent i moves at h' and passes, and becomes a BG lurker at h. Note that x_ℓ is BG-lurked at h. If $x_\ell \in C_i^{\subsetneq}(h)$, then, since i does not move at h, we have $x_\ell \in C_i^{\subsetneq}(h')$ as well. Because x_ℓ cannot be the object i BG lurks at h, object x_ℓ must be BG-lurked at h' by some other agent. But then, at h', i is not a BG lurker, and has previously been offered to clinch a BG-lurked object. Thus, by Lemma 6, h' is a terminating history, which is a contradiction. ∎

Lemma 13. *For any history h and any superhistory $h' \supseteq h$ of the form $h' = (h, a^*, a^*, \ldots, a^*)$, we have $i_{h'} \notin \mathcal{L}(h)$ and $i_{h'} \notin \mathcal{L}(h')$.*

Proof. The claim is immediate if $\mathcal{L}(h) = \varnothing$. Suppose $\mathcal{L}(h) \neq \varnothing$. We only show $i_{h'} \notin \mathcal{L}(h)$ as $i_{h'} \notin \mathcal{L}(h')$ then follows by setting $h' = h$. Let $\mathcal{L}(h) = \{\ell_1^h, \ldots, \ell_{\lambda(h)}^h\}$ be the set of lurkers at h and $\mathcal{X}^{\mathcal{L}}(h) = \{x_1^h, \ldots, x_{\lambda(h)}^h\}$ the set of lurked objects.

First, assume $h \neq h'$. Assume that the statement was false, and let $h' = (h, a^*, a^*, \ldots, a^*)$ be the smallest superhistory of h such that $i_{h'} = \ell_m^h$ for a lurker ℓ_m^h (that is, $i_{h''} \notin \mathcal{L}(h)$ for all $h \subseteq h'' \subsetneq h'$). Note first that, for any h'' such that $h \subseteq h'' \subsetneq h'$, $i_{h'''} = j \in \bar{\mathcal{L}}(h)$, and if there exists some lurked $x_m^h \in C_j^{\subsetneq}(h'')$, by Lemma 12, there is no passing action at h'', which is a contradiction. Therefore, any clinching action $a_y \in A(h'')$ clinches some $y \in \mathcal{X}(h) \setminus \mathcal{X}^{\mathcal{L}}(h)$, and for all terminal histories $\bar{h} \supset (h'', a_y)$, each lurker $\ell_m^h \in \mathcal{L}(h)$ receives his lurked object x_m^h. Finally, consider history h'. By Lemma 9, for each $\ell_m^h \in \mathcal{L}(h)$, $P_{\ell_m^h}(h') = P_{\ell_m^h}(h) = \mathcal{X}(h) \setminus \{x_1^h, \ldots, x_{m-1}^h\}$ (note that h' is reached from h via a series of passes, and so $\mathcal{X}(h) = \mathcal{X}(h')$), and $Top(\succ_{\ell_m^h}, P_{\ell_m^h}(h')) = x_m^h$ for all types $\succ_{\ell_m^h}$ such that h' is on the path of play. Therefore, by property 4 and greedy strategies, at h', there is no clinching action a_x for any $x \in P_{\ell_m^h}(h') \setminus \{x_m^h\}$. Thus, the only possibility is that every action $a \in A(h')$ clinches x_m^h.[32] This then implies that ℓ_m^h gets x_m^h at all terminal $\bar{h} \supset h'$. Combining this with the previous statement that ℓ_m^h gets x_m^h for all terminal $\bar{h} \supset (h'', a_y)$ for any $h \subseteq h'' \subsetneq h'$ and clinching action $a_y \in A(h'')$, we conclude that ℓ_m^h gets x_m^h for all terminal $\bar{h} \supset h$, i.e., ℓ_m^h has already clinched his object x_m^h at h. Thus, by definition of a millipede game, $i_{h'} \neq \ell_m^h$, which is a contradiction proving the first claim for $h' \neq h$.

Second, if $h = h'$ then let $h^* \subsetneq h$ be the immediate predecessor history of h. By the just proven part of the lemma, i_h is not a lurker at h^*, and because i_h moves at h, she cannot move at h^*, and hence she is not a lurker at h. ∎

Lemma 14. *Let i and j be active non-lurkers at a history h, and let $y \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ be an unlurked object at h. Further, assume that $i_h = i$ and $y \in C_i(h) \cap C_j^{\subsetneq}(h)$. Consider a type $\succ_j$ that reaches h, and define $\bar{x} = Top(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h))$. Then, $\bar{x} \succ_j y$.*

Proof. By Lemma 11, part (iii), agent j cannot be the terminator. By Lemma 11, part (iv), $P_j(h) \subseteq \bar{\mathcal{X}}^{\mathcal{L}}(h)$. Since i can clinch y at h, there must be some $x \in P_j(h)$ such that $x \succ_j y$, by OSP. Since $P_j(h) \subseteq \bar{\mathcal{X}}^{\mathcal{L}}(h)$, we have $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$, i.e., $Top(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h)) \succ_j y$. ∎

[32]Note that there cannot be a passing action either: if there were, then, since every history is non-trivial, there must be another action. But, as just argued, there can be no clinching actions for any other $x \neq x_m^h$, and thus there must be a clinching action for x_m^h, and the passing action would be pruned.

C.2 Proofs of Lemmas 3, 4, and 5

In the proofs that follow, we refer to roles in a game form Γ to state properties of Γ that are independent of the specific agent that is assigned to that role. Analogously to the sets of clinchable and possible objects for agents in a game, we write $C_r(h)$ to refer to the set of outcomes that are clinchable for the role $r \in \mathcal{R}$ at h and $P_r(h)$ for the set of outcomes that are possible for role r. Note that these sets do not depend on the role assignment function σ, and if for a particular role assignment, $\sigma(r) = i$, then $C_i(h) = C_r(h)$, $P_i(h) = P_r(h)$, etc. Analogously to the sets $\mathcal{A}(h)$ and $\mathcal{L}(h)$ for active agents and lurkers at a history h, we write $\mathcal{A}_R(h)$ for the set of active roles at a history h, and $\mathcal{L}_R(h)$ for the set of roles that are lurkers at h. When we want to refer to the game form with agents assigned to roles via a specific role assignment function σ_A, we write Γ_A. In the proofs, we often move fluidly between agents and roles; to avoid confusion, we use the notation i, j, k to refer to specific agents, and the notation r, s, t to refer to generic roles. Finally, note that while the set of agents who are lurkers at any h may differ depending on the role assignment function, the set of lurked objects, the order in which they become lurked, and the set of lurker roles depend only on h, and are independent of the specific agent assigned to the role that moves at h.

Unless otherwise specified, when we write the phrase "i clinches x at h" (or similar variants), what is meant is that i moves at h, takes some clinching action $a_x \in A(h)$, and receives object x at all terminal histories $\bar{h} \supseteq (h, a_x)$.

The following is a restatement of part (iv) of the definition of a lurker, but deserves an emphasis, as it arises frequently in the arguments below.

Remark 3. If, at a history h, object x is such that $x \in C_j(h)$ for an active non-lurker j at h, then x cannot become the next lurked object along the passing path $(h, a^*, \cdots, a^*)$.

Proof of Lemma 3

We show the first statement; the second statement is then an immediate corollary. Suppose agent i is ordered in step k of the ordering algorithm. First consider the case $k = 1$ and let agent i^* be the first agent to clinch in game Γ_σ and let h^* be the history at which i^* clinches; this clinching induces the ordering of the first segment of agents in step 1 of the ordering algorithm. Let $\mathcal{X}^{\mathcal{L}}(h^*) = \{x_1, \ldots, x_n\}$ be the set of lurked objects at h^*; this set may be empty.

Case: $\mathcal{A}(h) = \mathcal{L}(h) \cup \{i^*\}$. If i^* clinches an unlurked object $y \in \bar{\mathcal{X}}^{\mathcal{L}}(h^*)$, then, in Γ_σ, all lurkers get their lurked objects (the oldest lurker ℓ_1 gets x_1, the second oldest lurker ℓ_2 gets x_2, etc.), and in the resulting SD f_σ, the agents are ordered $f_\sigma : \ell_1, \ell_2, \ldots, \ell_n, i^*$. By Lemma 9, for each lurker ℓ_m, we have $x_m = Top(\succ_{\ell_m}, \mathcal{X} \setminus \{x_1, \ldots, x_{m-1}\})$. When it is agent ℓ_m's turn in the SD, she is offered to choose from $\mathcal{X} \setminus \{x_1, \dots, x_{m-1}\}$, and thus selects x_m. Finally, consider agent i^*. In game Γ_σ, when she clinches y at h^*, it is unlurked. By Corollary 3, $y = Top(\succ_{i^*}, \bar{\mathcal{X}}^{\mathcal{L}}(h^*))$. At her turn in the SD, the set of objects remaining is precisely $\bar{\mathcal{X}}^{\mathcal{L}}(h^*)$, and so i^* selects y.

In the remaining possibility, i^* clinches some lurked object x_m. Then all older lurkers $\ell_1, \dots, \ell_{m-1}$ get their lurked objects in Γ_σ, and the resulting SD begins as $f_\sigma : \ell_1, \dots, \ell_{m-1}, i^*$. By an argument equivalent to the previous paragraph, each of the lurkers once again gets the same object under the SD. For agent i^*, since she took a lurked object at h^* in Γ_σ, we have $x_m = Top(\succ_i, \mathcal{X})$, and thus, at her turn in the SD, she once again selects x_m, since it is still available. Then, in Γ_σ, agent ℓ_m is offered to clinch anything from $\mathcal{X} \setminus \{x_1, \dots, x_m\}$. If ℓ_m takes another lurked object $x_{m'}$ for some $m' > m$, then each lurker $\ell_{m+1}, \dots, \ell_{m'-1}$ is assigned to their lurked object, and we add to the SD order as $f_\sigma : \ell_1, \dots, \ell_{m-1}, i^*, \ell_{m+1}, \dots, \ell_{m'-1}, \ell_m$. By the same argument as above, at their turn in the resulting SD, each agent $\ell_{m+1}, \dots, \ell_{m'-1}, \ell_m$ gets the same object in the SD.[33] This process continues until someone eventually takes an unlurked object, all remaining lurkers are ordered, and step 1 is completed.

Case: $\mathcal{A}(h) = \mathcal{L}(h) \cup \{i^*, j\}$ **for some** $j \in \mathcal{A}(h) \setminus (\mathcal{L}(h) \cup \{i^*\})$**.** First consider the case that i^* clinches an unlurked object $y \in \bar{\mathcal{X}}^{\mathcal{L}}(h^*)$. If $y \notin C_j^{\subseteq}(h^*)$, then the argument is exactly the same as in Case (1) (note that j is not ordered in step 1 in this case). If $y \in C_j^{\subseteq}(h^*)$, then the step 1 partial order is $\ell_1 \tilde{\triangleright}^1 \cdots \tilde{\triangleright}^1 \ell_n \tilde{\triangleright}^1 \{i^*, j\}$. We must show that any SD run under $f_\sigma : \ell_1, \dots, \ell_n, i^*, j, \dots$ and $f'_\sigma : \ell_1, \dots, \ell_n, j, i^*, \dots$ result in the same outcome as Γ_σ for these agents. For the lurkers, the argument is as above in either case. For i^* and j, in game Γ_σ, by construction, $y \in C_j(h')$ for some $h' \subsetneq h^*$. Let $z = Top(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h^*))$, and note that by Lemma 14, $z \succ_j y$. Since i clinched y at h^*, we have $y \succ_i z$. In the SD, after all lurkers have picked, the set of remaining objects is precisely $\bar{\mathcal{X}}^{\mathcal{L}}(h^*)$. Thus, it does not matter whether i^* or j is ordered next in the SD, as there is no conflict between them: in both cases, i^* takes y, and j takes z, and both f_σ and f'_σ give the same allocation as Γ_σ. For the case where i^* begins by clinching some lurked object $x_m \in \mathcal{X}^{\mathcal{L}}(h^*)$, we consider agent j and the lurker who, in the chain of assignments, eventually takes an unlurked object y; otherwise, the argument is analogous.

The proof so far has shown that we get the same allocation for all agents ordered in step 1 of the ordering algorithm. If $k > 1$ then we proceed recursively through steps $2, ..., k$, as follows: If all active agents at $\mathcal{A}(h^*)$ are processed in step 1 of the ordering algorithm, then we repeat the same argument for the continuation subgame following the clinching by i^* at

[33] When it is agent ℓ_m's turn in the SD, the set of available objects is a subset of the set of objects that were offered to her when she clinched in $\Gamma_\sigma : \mathcal{X} \setminus \{x_1, \dots, x_{m'-1}\} \subseteq \mathcal{X} \setminus \{x_1, \dots, x_m\}$. However, $x_{m'}$ belongs to both sets, and so since ℓ_m takes $x_{m'}$ in Γ_σ, she also takes it at her turn in the SD, when her offer set is smaller.

h^*; the second step of the coding algorithm for the original game is the same as the first step of the coding algorithm for this continuation subgame. If not all active agents at $\mathcal{A}(h^*)$ are processed in step 1, then there is at most one active agent $j \in \mathcal{A}(h^*)$ who is not processed in this step. Agent j has been previously offered some objects in the set $C_j^{\subsetneq}(h^*)$ where $C_j^{\subsetneq}(h^*) \subseteq \bar{\mathcal{X}}^{\mathcal{L}}(h)$. The coding in the continuation subgame following the clinching at h^* is the same as coding in the Pareto-efficient auxiliary millipede that begins with agent j being offered clinching from $C_j^{\subsetneq}(h^*)$ and passing, and that then moves into the above continuation subgame; the second step of the coding algorithm for the original game is the same as the first step of the coding algorithm for this auxiliary millipede. ■

Proof of Lemma 4

First consider $k = 1$ and suppose $\tilde{\succ}_A^1$ is equal to the initial part of the ordering $\succ_B$. Define the function $g_A(i) = |j \in \mathcal{N} : j \succ_A i| + 1$, which is the number of agents ranked strictly ahead of i under $\succ_A$. This function will almost correspond to i's picking order in the resulting serial dictatorship, except if i ties under $\succ_A$; if i and i' tie, then $g_A(i) = g_A(i')$. Define g_B similarly.

Claim 1. If $\tilde{\succ}_A^1$ is equal to an initial segment of $\succ_B$, then $h_A^1 = h_B^1$.

Proof of Claim 1. Note that both h_A^1 and h_B^1 consist of a, possibly empty, sequence of passing moves, and so one of these histories must be a subset of the other. Towards a contradiction, assume that $h_A^1 \neq h_B^1$.

First, consider the case $h_A^1 \subsetneq h_B^1$. Define i_A to be the agent that clinches at h_A^1, and x_A to be the object that is clinched. Since there is a passing action at h_A^1, object x_A is unlurked at h_A^1, by Lemma 12. Since i_A clinches an unlurked object at h_A^1, we have $x_A = Top(\succ_{i_A}, \bar{\mathcal{X}}^{\mathcal{L}}(h_A^1))$ by Corollary 3. By construction of the coding algorithm, $g_A(i_A) = \lambda(h_A^1) + 1$, where $\lambda(h_A^1) = |\mathcal{L}_R(h_A^1)|$ is the number of lurkers (and hence also the number of lurked objects) that are present at h_A^1. Since $\tilde{\succ}_A^1$ is equal to an initial segment of $\succ_B$ and i_A is ordered in step 1 of Γ_A, we have $g_B(i_A) = \lambda(h_A^1) + 1$ as well.[34]

We claim that $\mathcal{X}^{\mathcal{L}}(h_A^1) = \mathcal{X}^{\mathcal{L}}(h_B^1)$. First, notice that $h_A^1 \subsetneq h_B^1$ implies $\mathcal{L}_R(h_A^1) \subseteq \mathcal{L}_R(h_B^1)$ and $\mathcal{X}^{\mathcal{L}}(h_A^1) \subseteq \mathcal{X}^{\mathcal{L}}(h_B^1)$, which follows because at each history in the millipede at most one object becomes lurked, and once an object is lurked, it remains lurked until it is clinched. If $\mathcal{X}^{\mathcal{L}}(h_B^1) \supsetneq \mathcal{X}^{\mathcal{L}}(h_A^1)$, then the $(\lambda(h_A^1)+1)^{th}$ lurked object in Γ_B (denoted $x_{\lambda(h_A^1)+1}$) must be x_A because (i) the coding algorithm puts the agent who receives $x_{\lambda(h_A^1)+1}$ as the $(\lambda(h_A^1) + 1)^{th}$ agent, and hence this agent is i_A, and (ii) by Lemma 3, i_A receives the same object under

[34] This is a key point, and its analogue remains true in the alternate case $h_B^1 \subsetneq h_A^1$. There, $g_B(i_B) = \lambda(h_B^1) + 1$, and we infer that also $g_A(i_B) = \lambda(h_B^1) + 1$. This follows because $h_B^1 \subsetneq h_A^1$ implies $\lambda(h_A^1) \geq \lambda(h_B^1)$, and so at least $\lambda(h_B^1) + 1$ agents are coded in step 1 of $\tilde{\succ}_A^1$. Thus, at least the first $\lambda(h_B^1) + 1$ agents in $\succ_B$ are in the same position in $\succ_A$, which includes agent i_B.

both σ_A and σ_B. But, because $x_A \in C_r(h_A^1)$, where r is the role that moves at h_A^1 and is not a lurker, x_A cannot be the $(\lambda(h_A^1)+1)^{th}$ lurked object, by part (iv) of the definition of a lurker, which is a contradiction. Therefore, $\mathcal{X}^{\mathcal{L}}(h_A^1) = \mathcal{X}^{\mathcal{L}}(h_B^1)$. This also means that $\mathcal{L}_R(h_A^1) = \mathcal{L}_R(h_B^1)$ and $\lambda(h_A^1) = \lambda(h_B^1)$; for simplicity, define $\lambda^1 := \lambda(h_A^1) = \lambda(h_B^1)$. Since x_A is unlurked at h_A^1, it is also unlurked at h_B^1.

Next, notice that some $j \neq i_A$ moves at h_A^1 in Γ_B, because otherwise, i_A would take the same (clinching) action at h_A^1 in Γ_B, which contradicts $h_A^1 \subsetneq h_B^1$. Let $s = \rho(h_A^1)$ be the role that moves at h_A^1, and so by definition, $\sigma_A(s) = i_A$ and $\sigma_B(s) = j$. At h_B^1, there are two active non-lurker roles: role s and another role s'. This follows because role s moves at h_A^1, and there is a passing action, so the history $h' = (h_A^1, a^*)$ must be controlled by a different active non-lurker role. Since there are no new lurkers at h_B^1, and there can be at most two active non-lurkers at any history, both roles s and s' remain active non-lurkers at h_B^1.

We claim that i_A must tie with another agent in $\succ_B$. To see this, note that if role s' moves at h_B^1, then i_A will tie with agent j in $\succ_B$, since $x_A \in C_s^{\subsetneq}(h_B^1)$ and $\sigma_B(s) = j$. If role s moves at h_B^1, then it is j that clinches at h_B^1 in Γ_B. If j clinches an unlurked object at h_B^1, then $g_B(j) = \lambda^1 + 1$, and so i_A ties with j in $\succ_B$. If j clinches a lurked object, then role s is the terminator role. Therefore, agent i_A was in the terminator role in Γ_A, and, since she clinched x_A first, we have $x_A = Top(\succ_A, \mathcal{X})$, which follows because all available objects are possible for the agent in the terminator role, by Lemma 11. This implies that i_A cannot be a lurker at h_B^1 in Γ_B, because if she were, she would have been offered to clinch x_A, and since it is her top object, would have clinched it prior to h_B^1, by greedy strategies. Thus, the only way for agent i_A to be such that $g_B(i_A) = \lambda^1 + 1$ is if she is an active non-lurker that does not move at h_B^1, which means that she must tie in $\succ_B$ with some agent.

Thus, we have shown that i_A must tie with some agent k in $\succ_B$, i.e., $g_B(i_A) = g_B(k) = \lambda^1+1$ for some k. Since i_A is coded in step 1 of Γ_A, and $\tilde{\succ}_A^1$ is equal to an initial segment of $\succ_B$, we further have $g_A(i_A) = g_A(k) = g_B(i_A) = g_B(k) = \lambda^1 + 1$; in other words, agent i_A ties with agent k in both $\succ_A$ and $\succ_B$.

Since i_A ties with k in Γ_A, at h_A^1, we have $x_A \in C_{s'}^{\subsetneq}(h_A^1)$ for the other active non-lurker role s' at h_A^1. We have seen that $\sigma_B^{-1}(i_A) \neq s$. If $\sigma_B(s') = i_A$, then in Γ_B, i_A passed at some history $h' \subsetneq h_A^1$ at which she was offered to clinch x_A in Γ_B. By Lemma 14, $Top(\succ_{i_A}, \bar{\mathcal{X}}^{\mathcal{L}}(h_A^1)) \succ_{i_A} x_A$, which is a contradiction. Since we know that i_A is coded in step 1 of Γ_B, the only other possibility is that in Γ_B, i_A is a lurker for some object z at h_B^1, which implies that $z \succ_{i_A} x_A$. It also means that the agent that moves at h_B^1 in Γ_B is clinching a lurked object (because if an unlurked object were clinched, then i_A would be assigned to z, a contradiction). This implies that h_B^1 is the terminating history, by Lemma 12, and $\rho(h_B^1)$ is the terminator role. We cannot have $\rho(h_B^1) = s$, because then role s is the terminator role, and i_A is in the terminator role in Γ_A and would not clinch x_A first in Γ_A, a contradiction. Thus, $\rho(h_B^1) = s'$, and s' is the terminator role. Finally, notice that at h_A^1, role s is offered x_A and $x_A \in C_{s'}^{\subsetneq}(h_A^1)$, which contradicts Lemma 11, part (iii).

The case $h_B^1 \subsetneq h_A^1$ follows an analogous argument; cf. footnote 34 for the needed adjustments. ■

Thus far, we have shown that if $\tilde{\succ}_A^1$ is equal to the initial part of the ordering $\succ_B$, then $h_A^1 = h_B^1$. We next show that the same roles are coded in step 1 of Γ_A and Γ_B, and further that $\sigma_A(r) = \sigma_B(r)$ for all such roles r.

Define $h^1 := h_A^1 = h_B^1$. In both games, the first clinching is taken by the agent in role $\rho(h^1)$, and the set of lurked objects and active lurker-roles are equivalent at the first clinching in both Γ_A and Γ_B. Letting $r_0 = \rho(h^1)$, write

$$\sigma_A(r_0) \to x_{a_1} \to \sigma_A(r_{a_1}) \to x_{a_2} \to \cdots \to \sigma_A(r_{a_M}) \to x_{a_{M+1}} \tag{A}$$

to represent the chain of clinching that is initiated in Γ_A by agent $\sigma_A(r_0)$ at h^1: agent $\sigma_A(r_0)$ clinches some (possibly lurked) object x_{a_1}, the agent $\sigma_A(r_{a_1})$ who was lurking x_{a_1} clinches lurked object x_{a_2}, etc., until eventually agent $\sigma_A(r_{a_M})$ ends the chain by being the first agent to clinch an unlurked object $x_{a_{M+1}}$. Similarly, for Γ_B, write

$$\sigma_B(r_0) \to x_{b_1} \to \sigma_B(r_{b_1}) \to x_{b_2} \to \cdots \to \sigma_B(r_{b_{M'}}) \to x_{b_{M'+1}}. \tag{B}$$

Note that the agents who begin the chains, $\sigma_A(r_0)$ and $\sigma_B(r_0)$ are not lurkers in their respective games, while all of the remaining agents are lurkers.[35] Also, not all of the agents ordered in step 1 need to appear in the corresponding chain; in particular, any lurker who receives their lurked object does not appear, nor does the other active non-lurker, if such an agent exists. If $M = M'$ and $\sigma_A(r_{a_m}) = \sigma_B(r_{b_m})$ for all $m = 0, \ldots, M$, then we say (A) and (B) are **equivalent chains**.

Claim 2. Suppose that (A) and (B) are equivalent chains. Then, the same roles are coded in step 1 in Γ_A and Γ_B, and further, for all such roles, $\sigma_A(r) = \sigma_B(r)$.

Proof of Claim 2. By construction of the coding algorithm, the set of roles coded during the coding step initiated at h_A^1 consists of (i) all lurker-roles at h_A^1, (ii) the non-lurker-role that moves at h_A^1, and potentially (iii) the active non-lurker role that does not move at h_A^1; label this role s. Since $h_A^1 = h_B^1$, (i) and (ii) are the same in Γ_A and Γ_B. For (iii), role s is coded in Γ_A if and only if the first unlurked object in the chain, $x_{a_{M+1}}$, has been offered to role s to clinch prior to h_A^1. Since the chains are equivalent, this holds in Γ_A if and only if it

[35]If there are no lurkers at h^1, this is obvious; if there are lurkers, it follows from Lemma 13.

holds in Γ_B, which establishes the first statement.

To see that $\sigma_A(r) = \sigma_B(r)$ for all roles that are coded in step 1 of Γ_A (and hence also step 1 of Γ_B), note that because (A) and (B) are equivalent, the statement holds for any role that appears in the chain. For roles that do not appear in the chain, if r' is a lurker role that is active at h^1, the corresponding lurked object x' is assigned to its lurker in both Γ_A and Γ_B, and so $\tilde{\succ}_A^1$ equivalent to the initial part of the ordering $\succ_B$ implies that $\sigma_A(r') = \sigma_B(r')$ for all such roles, by Lemma 3.

It remains to consider the active non-lurker role s that does not move at h^1. Note that $M = M'$ and $\sigma_A(r_M) = \sigma_B(r_M)$ implies, by Lemma 3, that $x_{a_{M+1}} = x_{b_{M'+1}}$; let $x_{M+1} := x_{a_{M+1}} = x_{b_{M'+1}}$, and recall that x_{M+1} is unlurked. If there is no such active role s, or if $x_{M+1} \notin C_s^{\subsetneq}(h^1)$, then this role is not coded in step 1, and we are done. Thus, assume that s exists, and that $x_{M+1} \in C_s^{\subsetneq}(h^1)$. In this case, the agent assigned to role s is ordered in step 1 in both Γ_A and Γ_B, and by construction, ties with agent $\sigma(r_M) := \sigma_A(r_M) = \sigma_B(r_M)$ in both $\succ_A$ and $\succ_B$. Once again, $\tilde{\succ}_A^1$ equivalent to the initial part of the ordering $\succ_B$ implies that $\sigma_A(s) = \sigma_B(s)$. ∎

Claim 3. Chains (A) and (B) are equivalent.

Proof of Claim 3. We begin by showing that $\sigma_A(r_0) = \sigma_B(r_0)$. Towards a contradiction, assume that $\sigma_A(r_0) \neq \sigma_B(r_0)$, which implies also that that $x_{a_1} \neq x_{b_1}$ Lemma 3. If $M = M' = 0$, then both chains have only one agent, $\sigma_A(r_0)$ and $\sigma_B(r_0)$, who immediately clinch unlurked objects. Define $\sigma_A(r_0) = i$ and $\sigma_B(r_0) = j$, where $i \neq j$, since they are clinching different objects in their respective games. Since $\tilde{\succ}_A^1$ is equal to the initial part of $\succ_B$, and both i and j clinch unlurked objects, this implies that i and j must tie under $\succ_A$ and $\succ_B$. Thus, by construction of the coding algorithm, there must be another non-lurker role $s \neq r_0$ that is active at h^1, and $\sigma_A(s) = j$ and $\sigma_B(s) = i$, and $x_{a_1}, x_{b_1} \in C_s^{\subset}(h^1)$. Since i clinches an unlurked object x_{a_1} at h^1 in Γ_A, we have $x_{a_1} = Top(\succ_i, \bar{\mathcal{X}}^{\mathcal{L}}(h^1))$, by Corollary 3. Now, consider game Γ_B. Since $\sigma_B(s) = i$ and $x_{a_1} \in C_s^{\subsetneq}(h^1)$, in game Γ_B, there is some history $h' \subsetneq h^1$ such that $x_{a_1} \in C_i(h')$. By Lemma 14, we have $Top(\succ_i, \bar{\mathcal{X}}^{\mathcal{L}}(h^1)) \succ_i x_{a_1}$, which is a contradiction.

Now, consider the case that $M > 0$. This implies that a lurked object, x_{a_1}, is clinched at h^1 in Γ_A, which means that role r_0 is the terminator role by Lemma 12. It also implies that that x_{a_1} is agent $\sigma_A(r_0)$'s favorite object (among all objects $\mathcal{X}$). So, in game Γ_B, agent $\sigma_A(r_0)$ must be lurking object x_{a_1}, i.e., she is in role r_{a_1} in Γ_B.[36] Agent $\sigma_A(r_{a_1})$—the agent who lurks x_{a_1} in Γ_A—receives x_{a_2}, and so in Γ_B, must be the lurker for x_{a_2}.[37] Similarly,

[36] Since x_{a_1} is lurked, it is only possible for "older" lurkers and the terminator. Agent $\sigma_A(r_0)$ cannot be an older lurker in Γ_B, because then she would have been offered x_{a_1}, and, by greedy strategies, would have clinched it. Nor can she be the the terminator, because $\sigma_B(r_0) \neq \sigma_A(r_0)$. Therefore, she must be in role r_{a_1} in Γ_B.

[37] This is because by definition of a lurker, agent $\sigma_A(r_{a_1})$ strictly prefers x_{a_1} to all younger lurked objects

agent $\sigma_A(r_2)$ must lurk x_{a_3} in Γ_B, etc., until we reach agent $\sigma_A(r_M)$. By similar reasoning as footnote 37, we conclude that agent $\sigma_A(r_M)$ must be in role s in Γ_B. For shorthand, define $k := \sigma_A(r_M)$, and so $\sigma_B^{-1}(k) = s$.[38]

Finally, since $\sigma_B^{-1}(k) = s$ and k is ordered in step 1 of Γ_B (see footnote 38), there must be some other agent j such that $g_B(j) = \lambda(h^1) + 1$, and so $g_A(j) = g_A(k) = g_B(j) = g_B(k) = \lambda(h^1) + 1$. Since $g_A(j) = \lambda(h^1) + 1$, j must be clinching an unlurked object in Γ_A. Since the first person to clinch an unlurked object in Γ_A is k who clinches $x_{a_{M+1}}$, it must be that $\sigma_A^{-1}(j) = s$ and $x_{a_{M+1}} \in C_s^{\subseteq}(h^1)$. Finally, since $\sigma_B^{-1}(k) = s$, we have $x_{a_{M+1}} \in C_k^{\subseteq}(h^1)$ in Γ_B, and by Lemma 14, $Top(\succ_k, \bar{\mathcal{X}}^{\mathcal{L}}(h^1)) \succ_k x_{a_{M+1}}$. However, since k chose to clinch $x_{a_{M+1}}$ in Γ_A and $x_{a_{M+1}}$ was unlurked, we have $Top(\succ_k, \bar{\mathcal{X}}^{\mathcal{L}}(h^1)) = x_{a_{M+1}}$, which is a contradiction.

The case where x_{b_1} is lurked is analogous, and the argument is omitted. We have thus shown that $\sigma_A(r_0) = \sigma_B(r_0)$.

If agent $\sigma_A(r_0)$ clinches an unlurked object, then the proof is complete. If not, then the above arguments can be repeated to show that $\sigma_A(r_{a_1}) = \sigma_B(r_{b_1})$, etc., until an unlurked object is reached. This completes the proof of Claim 3. ∎

Claims 2 and 3 imply the following:

Claim 4. The same roles r' are coded in step 1 of the coding algorithm applied to games Γ_A and Γ_B, and for all these roles $\sigma_A(r') = \sigma_B(r')$.

To complete the proof we establish the claim of the lemma for steps $k > 1$ of the coding algorithm by an inductive argument. Suppose that the lemma obtains for steps $1, ..., k$ of the coding algorithm. After the chain of clinchings initiated at h_A^k (which is the same as h_B^k), we enter a subgame among agents and objects that were unmatched till step k. By the inductive assumption, these subgames begin at some history $\hat{h}^{k+1}$ that is the same under both σ_A and σ_B. As argued in Remark 1, these subgames continue to have the structure of a millipede mechanism satisfying properties 1-5. Let $h_A^{k+1} \supseteq \hat{h}^{k+1}$ be the first history at which a clinching action is taken following a (possibly empty) sequence of passes in the subgame of Γ_A starting at $\hat{h}^{k+1}$; define $h_B^{k+1} \supseteq h^{k+1}$ analogously. If now $\succ_A^{k+1}$ equals to an initial segment of $\succ_B$, then we can repeat the arguments developed for $k = 1$ above to show that $h_A^{k+1} = h_B^{k+1}$, the same roles are coded in step $k+1$ under σ_A and σ_B, and $\sigma_A(r') = \sigma_B(r')$ for all roles coded in step $k+1$. The inductive argument completes the proof. ∎

---

and all unlurked objects; thus, in Γ_B, she cannot be an older lurker, because she would have been offered x_{a_1}, and thus could not end up with something she strictly disprefers (recall that by Lemma 3, all agents receive the same objects in both games). She cannot be the terminator, because then, since $h_A^1 = h_B^1$, and all objects are possible for the terminator, she would clinch x_{a_1}, which is again a contradiction to Lemma 3.

[38]Note that k is coded in step 1 of the coding algorithm applied to Γ_A, and receives an unlurked object, so $g_A(k) = \lambda^1 + 1$, and therefore, $g_B(k) = \lambda^1 + 1$. Since at least $\lambda^1 + 1$ agents are coded in step 1 of Γ_B, this is only possible if agent k is also coded in step 1 of Γ_B, and thus she must be active at h^1, and so the only possibility is that $\sigma_B^{-1}(k) = s$.

Proof of Lemma 5

For a (fixed) game form Γ, we let Γ_τ denote the specific game under role assignment σ_τ. Note that the set of objects that are lurked at any given history depends only on the game form, and is independent of the specific role assignment. We use the notation h^*_τ for the first history at which an object is clinched in Γ_τ; that is, $h^*_\tau = (h_\varnothing, a^*, \ldots, a^*)$, where a^* is the number of passes taken by the agents until the agent who moves at h^*_τ chooses to clinch at this history. The number of passes will depend on τ. For any agent j, we write x_j to denote the object that is ultimately received by agent j.

Note that it is without loss of generality to assume that for all games Γ_τ that we consider, at h^*_τ, the objects $x_{j_1}, \ldots, x_{j_P}$ are all lurked, in this order. To see this, note that if not, then, there is some game Γ_τ and $p' < P$ such that the last lurked object is $x_{j_{p'}}$. Consider the smallest such p'. Since $p' < P$, this means that the agents coded in step 1 of the coding algorithm are $j_1, \ldots, j_{p'}, j_{p'+1}$, and possibly $j_{p'+2}$, which can only occur if there is a tie at the end of the step.[39] Now, since all codings under consideration are exactly the same on the agents $j_1, \ldots, j_{p'}, j_{p'+1}, j_{p'+2}$, by Lemma 4 we have that in all of the games we consider, all of these agents are in the same roles, and, at the end of the first coding step, we reach the same history in each game to begin the next coding step. Thus, we can disregard these agents, and begin the analysis for each game at this history. Repeating this argument, we continually eliminate all higher ranked agents until we reach a coding step at which all of the remaining agents ranked strictly head of k_1 are coded in the first step in of the relevant continuation game.

Thus, for the entirety of this proof (including all sublemmas stated therein), we assume that the objects $x_{j_1}, \ldots, x_{j_P}$ are all lurked at h^*_τ for all games we consider. Note that this also implies that all agents $j_1, \ldots, j_P$ are ranked strictly, without ties, in all codings, and that there are at least $P + 1$ agents coded in the first step of every game Γ_τ. We allow the case $P = 0$, in which case there are no agents j_p.

Since agent i ties in $\succ_1$, she receives an object that is unlurked at h^*_1, which means that $x_i = Top(\succ_i, \bar{\mathcal{X}}^{\mathcal{L}}(h^*_1))$. By the structure of the sequence, this also implies that for $n' \geq 2$, if $x_i \in \bar{\mathcal{X}}^{\mathcal{L}}(h^*_{n'})$, then $x_i = Top(\succ_i, \bar{\mathcal{X}}^{\mathcal{L}}(h^*_{n'}))$ because each of the agents $i, j_1, ..., j_P$ receives the same object under both σ_1 and $\sigma_{n'}$ (by Lemma 3), and from the game Γ_1 we infer that i prefers the object received (x_i) to all objects except the objects assigned to $j_1, ..., j_P$, and in game $\Gamma_{n'}$ no other object belongs to $\bar{\mathcal{X}}^{\mathcal{L}}(h^*_{n'})$.

[39] By construction of the coding algorithm, if there are p' lurked objects at the initiation of a coding step, then the number of agents coded in that step is either $p' + 1$ or $p' + 2$. Since all of the agents j_p are ranked strictly above the remaining agents, and $p' < P$, none of the agents i nor $k_{n'}$ can be coded in step 1 of the game.

We begin with the following Lemmas 15, 16, and 17, which show that, under certain conditions, either condition (I) or (II) in the statement of the lemma will hold. Then, we apply these lemmas to show that all cases are covered, which will prove the result. The proofs of these lemmas can be found in the Supplementary Appendix.

The first of these lemmas shows that if there is a sequence Σ such that $n \geq 2$ and such that the lurked objects on the initial passing path of the game form are (in order) $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{n-1}}$, then i must tie in $\succ_{n+1}$.

Lemma 15. *Assume that there exists a sequence of role assignment functions Σ as defined in the statement of Lemma 5, and such that $n \geq 2$. Further, assume that along the initial passing path of the game form, the first lurked objects are (in order) $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{n-1}}$.*[40] *Then, at h^*_{n+1} in Γ_{n+1}, there is an agent $\ell \neq j_1, \ldots, j_P, k_1, \ldots, k_n, i$ such that ℓ is an active non-lurker at h^*_{n+1} that does not move at h^*_{n+1} and $x_i \in C^{\subsetneq}_{\ell}(h^*_{n+1})$. Further, i must tie with some other agent in $\succ_{n+1}$, and we label this agent k_{n+1}.*

Remark 4. A supposition in Lemma 15 (and in Lemma 18, below) is that the first lurked objects of the game form are $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{n-1}}$, in this order, where $n \geq 2$. A sufficient condition for this to hold is the following: there is a game Γ_A such that $j_1 \cdots j_P \succ_A k_1 \succ_A \cdots \succ_A k_{n-1} \succ_A \{i, k_n\} \succ_A \cdots$ and i is coded in the initial step of the coding algorithm.

To see this, assume not, and let n' be the smallest n such that $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{n'-1}}$ become lurked, but $x_{k_{n'}}$ is not the next lurked object. This means that at h^*_A (the history of the first clinching in Γ_A), there are at most $\lambda^*_A = P + n' - 1$ lurked objects. Consider agent $k_{n'}$. By construction, $n' < n$, and so $k_{n'}$ does not tie in $\succ_n$. Thus, in the coding step in Γ_A that begins at h^*_A, agent $k_{n'}$ must be the first agent to clinch an unlurked object. This ends the coding step at $k_{n'}$, without a tie, which contradicts that i is coded in this step in game Γ_A. ■

The second of these lemmas shows that if there is a sequence Σ, plus an additional role assignment function σ_0 in which all $j_1, \ldots, j_P$ are ranked strictly above i, who is ranked strictly above k_1, who is ranked strictly above all other remaining agents, then i must tie in $\succ_{n+1}$.

Lemma 16. *Assume that there exists a sequence of role assignment functions Σ as defined in the statement of Lemma 5. If there exists another role assignment function σ_0 with a corresponding coding,*

$$j_1 \cdots j_P \succ_0 i \succ_0 k_1 \succ_0 \cdots,$$

then in $\succ_{n+1}$ of Σ, i must tie with some agent k_{n+1}.

[40] We allow for the possibility that $P = 0$, but whether $P = 0$ or $P > 0$, the assumption that $n \geq 2$ implies that along the initial passing path of the game form, at least x_{k_1} becomes lurked.

Remark 5 (Symmetry). Lemmas 15 and 16 were stated for sequence Σ, and concluded that i must tie in $\succ_{n+1}$. There are also symmetric versions of these lemmas that apply to sequence Σ' and conclude that k_1 must tie in $\succ_{m+1}$ that have the exact same proof.

The last of these lemmas deals with the case that neither x_i nor x_{k_1} are the $(P+1)^{th}$ lurked object on the initial passing path, nor does there exist a σ_0 as in Lemma 16.

Lemma 17. *Assume that there exist two sequences of role assignment functions Σ and Σ' as defined in the statement of Lemma 5 such that $n, m \geq 2$. Further, assume that along the initial passing path of the game form, the objects $x_{j_1}, \ldots, x_{j_P}$ all become lurked, in this order, but neither x_i nor x_{k_1} is the $(P+1)^{th}$ lurked object.Then, one of the following is true:*

1. *In $\succ_{n+1}$, agent i must tie with some agent k_{n+1}.*
2. *In $\succ'_{m+1}$, agent k_1 must tie with some agent k'_{m+1}.*

With these lemmas in hand, we can complete the proof of Lemma 5 as follows:

- If there exists σ_0 such that $j_1 \cdots j_P \succ_0 i \succ_0 k_1 \succ_0 \cdots$, then we apply Lemma 16 to Σ to conclude that (I) holds.
- If there exists σ'_0 such that $j_1 \cdots j_P \succ'_0 k_1 \succ'_0 i \succ'_0 \cdots$, then we apply the symmetric version of Lemma 16 with k_1 and i swapped to Σ' to conclude that (II) holds.
- If neither of the above two cases hold (i.e., there do not exist σ_0 nor σ'_0):[41]
  - If x_{k_1} is the $(P+1)^{th}$ lurked object along the initial passing path, then we apply Lemma 15 to Σ to conclude that (I) holds.
  - If x_i is the $(P+1)^{th}$ lurked object along the initial passing path, then we apply the symmetric version of Lemma 15 with k_1 and i swapped to Σ' to conclude that (II) holds.
  - If neither x_{k_1} nor x_i is the $(P+1)^{th}$ lurked object along the initial passing path, then we apply Lemma 17 to conclude that either (I) or (II) must hold. ■

[41]Notice that the assumption that there is no σ_0 or σ'_0 imply that $n, m \geq 2$, which is needed to apply Lemma 15.

Supplementary Appendix: Proofs of Lemmas 15, 16, and 17 (For Online Publication)

Proof of Lemma 15. We start with the following lemma.

Lemma 18. *Assume that there exists a sequence of role assignment functions* Σ *as defined in the statement of Lemma 5, and such that* $n \geq 2$*. Further, assume that along the initial passing path of the game form, the first lurked objects are (in order)* $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n-1}}$.[42] *Then:*

(a) For all $n' = 1, \dots, n-1$*, the agent that moves at* $h^*_{n'}$ *in* $\Gamma_{n'}$ *is agent* i*, and at* $h^*_{n'}$*, the number of lurked objects is* $P + n' - 1$.

(b) $h^*_1 \subsetneq h^*_2 \subsetneq \cdots \subsetneq h^*_{n-1} \subsetneq h^*_n$.

(c) For all $n' = 1, \dots, n$*, the number of lurked objects at* $h^*_{n'}$ *is* $P + n' - 1$.

(d) For all $n' = 1, \dots, n-1$, $p = 1, \dots, P$*, and* $n'' = 1, \dots, n'$*, in* $\Gamma_{n'}$*, agent* j_p *is in the role that lurks* x_{j_p} *and agent* $k_{n''}$ *is in the role that lurks* $x_{k_{n''}}$.

(e) $h^*_{n-1} \subsetneq h^*_{n+1}$ *and the number of lurked objects at* h^*_{n+1} *is at least* $P + n - 1$.

Proof of Lemma 18. **Part (a).** Let $\lambda^*_{n'}$ be the number of lurked objects at history $h^*_{n'}$. Notice that since $\succ_{n'}$ has a tie in the $(P+n')^{th}$ place, we have $\lambda^*_{n'} \leq P + n' - 1$ for all $n' = 1, \dots, n$. Towards a contradiction, assume there was a game $\Gamma_{n'}$ for which i does not move at $h^*_{n'}$.. Since $\lambda^*_{n'} \leq P + n' - 1$, the structure of $\succ_{n'}$ implies that the lurked objects are $\{x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{\lambda^*_{n'}-P}}\}$,[43] and the agents coded in step 1 of $\Gamma_{n'}$ are $\{j_1, \dots, j_P, k_1, \dots, k_{\lambda^*_{n'}-P+1}\}$ (if $\lambda^*_{n'} < P + n' - 1$) or $\{j_1, \dots, j_P, k_1, \dots, k_{\lambda^*_{n'}-P+1}, i\}$ (if $\lambda^*_{n'} = P + n' - 1$). and the set of lurked objects is $\{x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{\lambda^*_{n'}-P}}\}$. Now, notice that it cannot be a lurked object that is clinched at $h^*_{n'}$. Indeed, if this were true, then $h^*_{n'}$ is the terminating history, which implies that $x_{k_{\lambda^*_{n'}-P}}$ is the last lurked object on the initial passing path of the game (Lemma 12). But, this contradicts the assumption that $x_{k_{\lambda^*_{n'}-P+1}}$ is the next lurked object on the initial passing path, where notice that such an object exists because $\lambda^*_{n'} - P + 1 \leq n' \leq n-1$. Thus, it must be an unlurked object that is clinched at $h^*_{n'}$. In particular, by the structure of $\succ_{n'}$, the only possibilities are that agent $k_{\lambda^*_n - P+1}$ clinches object $x_{k_{\lambda^*_n-P+1}}$, or agent i clinches x_i, where the latter case is only possible if $\lambda^*_{n'} = P + n' + 1$. However, if agent $k_{\lambda^*_n-P+1}$ clinches object $x_{k_{\lambda^*_n-P+1}}$, then object $x_{k_{\lambda^*_n-P+1}}$ has been offered to an active non-lurker at $h^*_{n'}$, and so $x_{k_{\lambda^*_n-P+1}}$ cannot be the next lurked object along the initial

[42] We allow for the possibility that $P = 0$, but whether $P = 0$ or $P > 0$, the assumption that $n \geq 2$ implies that along the initial passing path of the game form, x_{k_1} is the $(P+1)^{th}$ lurked object. .

[43] This is implicitly assuming that $\lambda^*_{n'} > P$. An analogous argument works for the case that $\lambda^*_{n'} \leq P$, but, for brevity, this argument is omitted.

passing path (Remark 3), a contradiction. Therefore, it must be that $\lambda^*_{n'} = P + n' - 1$, and agent i is the agent that moves at $h^*_{n'}$.

Parts (b). As shown in part (a), for $n' = 1, \ldots, n-1$, there are $\lambda^*_{n'} = P + n' - 1$ lurked objects at $h^*_{n'}$, which immediately implies that $h^*_1 \subsetneq h^*_2 \subsetneq \cdots \subsetneq h^*_{n-2} \subsetneq h^*_{n-1}$ (because the number of lurked objects only grows as we go down the initial passing path).

It remains to show that $h^*_{n-1} \subsetneq h^*_n$. By way of contradiction, assume that $h^*_n \subseteq h^*_{n-1}$. Then, $\lambda^*_n \leq \lambda^*_{n-1} = P + n - 2$, and the lurked objects at h^*_n are $\{x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{\lambda^*_n}}\}$. If a lurked object is clinched at h^*_n, then h^*_n is the terminating history, and there is no passing action at h^*_n (Lemma 12). However, this contradicts that $x_{k_{\lambda^*_n+1}}$ is the next lurked object on the initial passing path. So, it must be an unlurked object that is clinched. By the structure of $\succ_n$, it must be $k_{\lambda^*_n+1}$ that clinches $x_{k_{\lambda^*_n+1}}$. But then, $x_{k_{\lambda^*_n+1}}$ has been offered to active nonlurker at h^*_n, and so $x_{k_{\lambda^*_n+1}}$ cannot be the next lurked object along the initial passing path (Remark 3), which is a contradiction. Therefore, $h^*_{n-1} \subsetneq h^*_n$.

Part (c). Part (a) shows this for $n' \leq n-1$. So, we must show $\lambda^*_n = P + n - 1$. Notice that $h^*_{n-1} \subsetneq h^*_n$ implies that $\lambda^*_n \geq \lambda^*_{n-1} = P + n - 2$, while the structure of $\succ_n$ (in particular, the tie between agent i and k_n), implies that $\lambda^*_n \leq P + n - 1$. Thus, we need to show $\lambda^*_n \neq P + n - 2$. Assume that $\lambda^*_n = P + n - 2$. Then, the lurked objects are $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{n-2}}$, and the agents coded in step 1 are $j_1, \ldots, j_P, k_1, \ldots, k_{n-2}, k_{n-1}$. If a lurked object is clinched at h^*_n, then this is the terminating history, which contradicts that $x_{k_{n-1}}$ is the next lurked object along the initial passing path (Lemma 12). If an unlurked object is clinched, then it must be k_{n-1} clinching $x_{k_{n-1}}$, but since this is offered to an active non-lurker, $x_{k_{n-1}}$ cannot be the next lurked object along the initial passing path (3), a contradiction. Therefore, $\lambda^*_n = P + n - 1$.

Part (d). By part (a), agent i moves at $h^*_{n'}$ in $\Gamma_{n'}$, and, since i ties in $\succ'_n$, object x_i is unlurked. Therefore, all lurked objects are immediately assigned to their lurkers, which delivers the result.

Part (e). To show $h^*_{n-1} \subsetneq h^*_{n+1}$, assume not. Then, $h^*_{n+1} \subseteq h^*_{n-1}$, and $\lambda^*_{n+1} = P + \bar{n} - 1$ for some $\bar{n} \leq n - 1$. So, the lurked objects at h^*_{n+1} are $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{\bar{n}-1}}$, and the agents coded in step 1 are $j_1, \ldots, j_P, k_1, \ldots, k_{\bar{n}}$. Since $\bar{n} \leq n - 1$, we know that $x_{k_{\bar{n}}}$ must be the next lurked object on the initial passing path. An argument analogous to those given above delivers a contradiction.

To show $\lambda^*_{n+1} \geq P + n - 1$, note that $h^*_{n-1} \subsetneq h^*_{n+1}$ implies $\lambda^*_{n+1} \geq \lambda^*_{n-1} = P + n - 2$. Thus, we must just show that $\lambda^*_{n+1} \neq P + n - 2$. So, assume this was the case. Then, the lurked objects are $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{n-2}}$, and the agents coded in step 1 are $j_1, \ldots, j_P, k_1, \ldots, k_{n-2}, k_{n-1}$. If a lurked object is clinched at h^*_{n+1}, then this is the terminating history, which contradicts that $x_{k_{n-1}}$ is the next lurked object along the initial passing path (Lemma 12). If an unlurked object is clinched, then it must be k_{n-1} clinching $x_{k_{n-1}}$, but since this is offered to an active non-lurker, $x_{k_{n-1}}$ cannot be the next lurked object along the initial passing path (Remark 3), a contradiction. Therefore, $\lambda^*_{n+1} \geq P + n - 1$.

This completes the proof of Lemma 18. ■

Continuing with the proof of Lemma 15, we first show the first statement, that at h^*_{n+1} there is an agent $\ell \neq j_1, \ldots, j_P, k_1, \ldots, k_n, i$ such that ℓ is an active non-lurker at h^*_{n+1} that does not move at h^*_{n+1}, and $x_i \in C^{\subsetneq}_{\ell}(h^*_{n+1})$. By Lemma 18, we have (i) $h^*_{n-1} \subsetneq h^*_n, h^*_{n+1}$ (ii) $\lambda^*_n = P + n - 1$ and (iii) $\lambda^*_{n+1} \geq P + n - 1$. In particular, the lurked objects at h^*_n are $\{x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{n-1}}\}$. Since there is a tie in $\succ_n$, there are two active non-lurker roles at h^*_n, and both of these roles have been offered to clinch x_i at h^*_n. Let s be the role that moves at h^*_n, and s' be the other active non-lurker that does not move at h^*_n.

Case 1: x_{k_n} is the next lurked object along the initial passing path of the game form. Since x_{k_n} is the next lurked object along the initial passing path, it must be i that moves at h^*_n and clinches x_i, i.e., $\sigma_n(s) = i$. [44] Further, we have $h^*_n \subsetneq h^*_{n+1}$. To see this, note that if not, then $h^*_{n+1} \subseteq h^*_n$, and x_{k_n} is not lurked at h^*_{n+1}. Thus, it cannot be a lurked object that is clinched at h^*_{n+1}, because this would imply that h^*_{n+1} is the terminating history (Lemma 12), which contradicts that x_{k_n} becomes lurked along the initial passing path. So, the object clinched at h^*_{n+1} must be unlurked, and so the set of lurked objects is $\{x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{\bar{n}-1}}\}$, where $x_{k_{\bar{n}}}$ is the unlurked object that is clinched, and $\bar{n} \leq n$, which follows because $h^*_{n+1} \subseteq h^*_n$. But then, $x_{k_{\bar{n}}}$ is offered to an active non-lurker at h^*_{n+1}, which contradicts that it is the next lurked object along the initial passing path (Remark 3). Therefore, $h^*_n \subsetneq h^*_{n+1}$.

Since x_{k_n} is the next lurked object along the initial passing path, we must have x_{k_n} becoming lurked at some h' such that $h^*_n \subsetneq h' \subseteq h^*_{n+1}$. But, notice that there is still some role r such that, at h', r is an active non-lurker, and $x_i \in C^{\subsetneq}_r(h')$. Thus, x_i cannot be the next lurked object along the initial passing path. Therefore, for i to be ranked immediately after k_n in $\succ_{n+1}$, she must clinch x_i while it is unlurked, either at h^*_{n+1}, or in the resulting step 1 assignment chain of the coding algorithm.

We next claim that in Γ_{n+1}, $\sigma^{-1}_{n+1}(i) \neq s, s'$. To see this, first note that if $\sigma^{-1}_{n+1}(i) = s$, then i has the same role in Γ_n and Γ_{n+1}, and thus would once again clinch at h^*_n in Γ_{n+1}, which contradicts $h^*_n \subsetneq h^*_{n+1}$. Therefore, $\sigma^{-1}_{n+1}(i) \neq s$. Next, assume that $\sigma_{n+1}(s') = i$. Notice that role s' cannot be the terminator role, by Lemma 11(iii) and the fact that $x_i \in C_s(h^*_n)$ and $x_i \in C^{\subsetneq}_{s'}(h^*_n)$. Thus, only objects that are unlurked at h^*_n are possible for role s', and so if $\sigma_{n+1}(s') = i$, since x_i is i's top unlurked object, she would clinch it at some history

[44] Agent k_n cannot move at h^*_n, because then x_{k_n} would have been offered to an active non-lurker at h^*_n, which contradicts that x_{k_n} is the next lurked object along the initial passing path. Nor can it be any $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, x_{k_{n-1}}$, because then they would be clinching a lurked object, and so h^*_n is the terminating history, which again contradicts that x_{k_n} is the next lurked object along the initial passing path.

$h' \subsetneq h_n^* \subseteq h_{n+1}^*$, which is a contradiction. Therefore, $\sigma_{n+1}^{-1}(i) \neq s, s'$.

We showed above that s' is not the terminator role. If s is the terminator role, then, when i clinches at h_n^*, we conclude that x_i is her top possible object among all of those that are available. This implies that i cannot be in a role that is a lurker at h_n^*. So, we have shown that in Γ_{n+1}, agent i is not a lurker at h_n^*, nor is she is role s or s'. Thus, i is not active at h_n^* in Γ_{n+1}, and so there must be some agent $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n$ such that $\sigma_{n+1}^{-1}(\ell) = s$ or s'. But then, since i is unlurked at h_{n+1}^*, we have that $x_i \in C_\ell^{\subsetneq}(h_{n+1}^*)$, as desired.

If s is not the terminator role, we once again claim that i cannot be in a role that is a lurker at h_n^*. Indeed, if this were true, then some agent j who is receiving a lurked object is not a lurker at h_n^*. Therefore, this agent must be in the terminator role, and clinch at h_{n+1}^*. Since the terminator role is not s or s', it is not yet active at h_n^*, and so j is not active at h_n^* in Γ_{n+1}. Therefore, there must be some $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n$ such that $\sigma^{-1}(\ell) = s$ or s', and that is still active when j clinches at h_{n+1}^*, which implies that $x_i \in C_\ell^{\subsetneq}(h_{n+1}^*)$, as desired.

Case 2: x_{k_n} is not the next lurked object along the initial passing path. By Lemma 18, at h_n^*, there are $P + n - 1$ lurked objects. This implies that both i and k_n are coded in step 1 of the coding algorithm for Γ_n, and thus that the first unlurked object that is clinched is either x_i or x_{k_n}.[45] This gives rise to two subcases.

Case 2.1: x_{k_n} is the first unlurked object that is clinched in the coding algorithm in Γ_n. In this case, $\sigma_n(s') = i$, and there is some history $\tilde{h} \subsetneq h_n^*$ such that $x_{k_n} \in C_i(h_n^*)$.

Claim 5. The following are true: (a) $h_{n-1}^* \subsetneq h_{n+1}^* \subsetneq h_n^*$ and (b) agent k_n clinches x_{k_n} at h_{n+1}^* in Γ_{n+1}, and x_{k_n} is unlurked at this history.

Proof of Claim 5. **Part (a).** First notice that $h_{n-1}^* \subsetneq h_{n+1}^*$ follows from Lemma 18. So, we must show that $h_{n+1}^* \subsetneq h_n^*$. Towards a contradiction assume that $h_n^* \subseteq h_{n+1}^*$. Since $h_{n-1}^* \subsetneq h_n^*$, we have $h_{n-1}^* \subsetneq h_n^* \subseteq h_{n+1}^*$. Lemma 18 also implies that $\lambda_n^* = P + n - 1$. Since i does not move at h_n^* in Γ_n, it must be some $j_1, \dots, j_P, k_1, \dots, k_n$ that does. If a lurked object is clinched at h_n^*, then h_n^* is the terminating history. It also implies that agent k_n is a lurker for some lurked object, and therefore in step 1 of the coding algorithm, some agent takes the object k_n lurks, and he ends the step by clinching x_{k_n}, which is unlurked. This means that x_{k_n} is his favorite object that is unlurked at h_n^*. Now, consider Γ_{n+1}, and note that $h_n^* \subseteq h_{n+1}^*$ and h_n^* being the terminating history implies that $h_n^* = h_{n+1}^*$. In Γ_{n+1}, the set of lurked objects is the same as in Γ_n, so x_{k_n} is again the first unlurked object that is clinched in step 1 of the coding algorithm. But, since $h_n^* = h_{n+1}^*$, there is again an agent in role s' who is an active non-lurker at h_{n+1}^*, and so this agent would once again tie with k_n in $\succ_{n+1}$, a contradiction. Therefore, it must be that k_n is the agent that moves at h_n^* in Γ_n, which means that x_{k_n} has been offered to both active non-lurker roles at h_n^*. Since we assumed that $h_n^* \subseteq h_{n+1}^*$, it is

[45] Note that this does not necessarily mean that the object clinched at h_n^* is x_i or x_{k_n}.

impossible for k_n to be ranked n^{th} strictly, without ties, in $\succ_{n+1}$,[46] which is a contradiction. Thus, we have shown that $h^*_{n+1} \subsetneq h^*_n$, which is part (a).

Part (b). Part (a) plus Lemma 18 implies that $\lambda^*_{n+1} = P + n - 1$. Additionally, $h^*_{n+1} \subsetneq h^*_n$ means that h^*_{n+1} is not the terminating history, so it must be an unlurked object that is clinched there. Thus, since k_n is ordered $(P+n)^{th}$ without ties, it must be that k_n clinches x_{k_n} at h^*_{n+1}, and x_{k_n} is unlurked. ■

By Lemma 18, the agent that moves at h^*_{n-1} must be agent i, and therefore, at h^*_{n-1}, there are two active non-lurker roles that both have been offered x_i. Let the role that moves at h^*_{n-1} be denoted r, and the other active non-lurker at h^*_{n-1} be denoted r'. Thus, by definition, $\sigma_{n-1}(r) = i$.

We claim that in Γ_{n+1}, i cannot be active at h^*_{n-1}. At h^*_{n-1}, there are $P + n - 2$ active lurker roles, and two active non-lurker roles, r and r'. First, it is clear that $\sigma_{n+1}(r) \neq i$, because otherwise i is in the same role in Γ_{n-1} and Γ_{n+1}, and so would clinch at h^*_{n-1} in Γ_{n+1}, which contradicts $h^*_{n-1} \subsetneq h^*_{n+1}$ from Claim 5. Second, assume that in Γ_{n+1}, agent i is in a lurker role for a lurked object at h^*_{n-1}, say y. By part (b) of Claim 5, agent k_n clinches an unlurked object at h^*_{n+1}, and so all lurkers are immediately assigned to their lurked objects, which means that i would receive y which is a contradiction.

It remains to rule out that $\sigma^{-1}_{n+1}(i) = r'$. By construction, $x_i \in C_r(h^*_{n-1})$, where $x_i \in C_{r'}(\tilde{h})$ for some $\tilde{h} \subsetneq h^*_{n-1}$. This implies that role r' cannot be the the terminator role, by Lemma 11(iii), and the fact that $x_i \in C_r(h^*_{n-1})$. Since role r' is not the terminator role, only unlukred objects are possible for role r', by Lemma 11(iv). As x_i is agent i's most preferred unlurked object, by greedy strategies, she would clinch at $\tilde{h}$, which is a contradiction. Therefore, i is not active at h^*_{n-1} in Γ_{n+1}.

We also claim that i is not active at h^*_{n-1} in Γ_n, either. The arguments are the same as above for Γ_{n+1}, except for the case in which i lurks some lurked object at h^*_{n-1}. This is ruled out by the fact that $\sigma_n(s') = i$, and s' is a non-lurker at h^*_{n-1}.

Next, we claim that $\sigma_{n+1}(s) \neq i$. To see this, recall that $\sigma_n(s') = i$, and, as we showed, i is not active at h^*_{n-1} in Γ_n or Γ_{n+1}. This means that $s' \neq r, r'$, or in other words, s' is a role that becomes active after h^*_{n-1}. Thus, we must have $s = r$ or r', and so role s is active at h^*_{n-1}, which implies that $\sigma_{n+1}(s) \neq i$.

Next, we claim that $\sigma_{n+1}(s') = k_n$. Indeed, since $h^*_{n-1} \subsetneq h^*_{n+1} \subsetneq h^*_n$ and k_n moves at h^*_{n+1}, k_n must be in role either s or s'. If $\sigma_{n+1}(s) = k_n$, then, since she does not tie in $\succ_{n+1}$, she must clinch x_{k_n} at some history h' such that $h^*_{n-1} \subsetneq h' \subsetneq \hat{h}$, where $\hat{h}$ is the history at which

[46] Note that x_{k_n} cannot be the next lurked object, so, there must be no newly lurked objects at h^*_{n+1} (Remark 3). If k_n clinches at h^*_{n+1}, she would tie with the other active non-lurker. If some other agent clinches at h^*_{n+1}, then either this agent is ranked strictly ahead of k_n, or she ties with k_n, which again is a contradiction.

role s' is offered to clinch x_{k_n}. This implies that $\sigma_n(s) \neq k_n$, or else in Γ_n, she would also clinch at h'. So, in Γ_n, $\sigma_n(s) = k_{n'}$ for some $n' < n$, and k_n is in the lurker role for some object $x_{k_{\bar{n}}}$. The former implies that h^*_n is the terminating history, while the latter implies that k_n strictly prefers $x_{k_{\bar{n}}}$ to x_{k_n}. But then, since $\sigma_{n+1}(s) = k_n$, agent k_n is in the terminator role in Γ_{n+1}, and thus $x_{k_{\bar{n}}}$ is a possible outcome for her, she would not choose to clinch x_{k_n} first at h^*_{n+1}, a contradiction. Therefore, $\sigma_{n+1}(s') = k_n$.

Concluding the argument for Case 2.1, because k_n clinches an unlurked object at h^*_{n+1} in Γ_{n+1}, all agents $j_1, \dots, j_P, k_1, \dots, k_{n-1}$ must be in the lurker role for their respective objects. Therefore, none of them are in role s. As just shown, $\sigma_{n+1}(s) \neq k_n$ or i, either. All of this means that $\sigma_{n+1}(s) = \ell$ for some $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n, i$, and in Γ_{n+1}, we have $x_i \in C^{\subsetneq}_\ell(h^*_{n+1})$, as desired.

Case 2.2: x_i is the first unlurked object that is clinched in step 1 of the coding algorithm in Γ_n. In this case, we have that $\sigma_n(s') = k_n$, and $x_i \in C_{s'}(\tilde{h})$ for some $\tilde{h} \subsetneq h^*_n$. There are two further subcases:

Case 2.2.1: $\sigma_n(s) \neq i$. In this subcase, $\sigma_n(s)$ is one of $j_1, \dots, j_P, k_1, \dots, k_{n-1}$, and is clinching a lurked object at h^*_n. This implies that h^*_n is the terminating history, and s is the terminator role, which also means that we have $h^*_{n-1} \subsetneq h^*_{n+1} \subseteq h^*_n$. This combined with Lemma 18 implies that there are $P + n - 1$ lurkers at h^*_n, and the structure of $\succ_{n+1}$ means that x_{k_n} is the first unlurked object clinched in step 1 of Γ_{n+1}, and, at h^*_{n+1}, x_{k_n} has not been offered to the active non-lurker who does not move at h^*_{n+1}.

We also claim that role s cannot be active at history h^*_{n-1}. Indeed, since i clinches at h^*_{n-1} in Γ_{n-1} and ties, we know that there are two active non-lurker roles, say r and r', and they both have been offered x_i. If role s were one of these roles, then, since s is the terminator role, Lemma 11 implies that $x_i \notin C_{s'}(\tilde{h})$, which is a contradiction. This implies that role s is a role that becomes active after h^*_{n-1}. Since there is only one new lurker between h^*_{n-1} and h^*_{n+1}, this further implies that role s' must have been active at h^*_{n-1}, and $x_i \in C^{\subsetneq}_{s'}(h^*_{n-1})$.

We next claim that $\sigma_{n+1}(s') \neq i$. To see why this is true, notice that s' is not the terminator role (because that is role s). Thus, only unlurked objects are possible for role s' (Lemma 11(iv)), and, since we know that x_i is i's favorite unlurked object, if she were in role s', she would clinch at $\tilde{h} \subsetneq h^*_{n+1}$, a contradiction. Therefore, $\sigma_{n+1}(s') \neq i$.

Now, if it is one of the $j_1, \dots, j_P, k_1, \dots, k_{n-1}$ that moves at h^*_{n+1}, then h^*_{n+1} is the terminating history, and so $h^*_{n+1} = h^*_n$. This implies that x_i has been offered to the agent in role $\sigma_{n+1}(s')$ (who is not coded in step 1). As we just showed that $\sigma_{n+1}(s') \neq i$, we have $\sigma_{n+1}(s') = \ell$ for some $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n$, and $x_i \in C_\ell(h^*_{n+1})$ in Γ_{n+1}, as desired.

Concluding subcase 2.2.1, assume that it is k_n that moves at h^*_{n+1} in Γ_{n+1}. This means that k_n is in role s or s' in Γ_{n+1}. Note that we cannot have $\sigma_{n+1}(s') = k_n$, because if this were true, then k_n has the same role in Γ_n as in Γ_{n+1}, and would pass at all histories in Γ_{n+1}, just as she did in Γ_n. Therefore, $\sigma_{n+1}(s) = k_n$. Again, as we know that $\sigma_{n+1}(s') \neq i$, we have that $\sigma_{n+1}(s') = \ell$ for some $\ell \neq j_1, \ldots, j_P, k_1, \ldots, k_n$, and $x_i \in C_\ell(h^*_{n+1})$ in Γ_{n+1}, as desired.

Case 2.2.2: $\sigma_n(s) = i$. In this subcase, i clinches x_i at h^*_n. If $h^*_n \subseteq h^*_{n+1}$, then notice that at h^*_n in Γ_{n+1}, there are two active non-lurker roles, s and s', that have been offered x_i. We claim that $\sigma^{-1}_{n+1}(i) \neq s, s'$. First, it is clear that $\sigma_{n+1}(s) \neq i$, as otherwise, i would clinch at h^*_n in Γ_{n+1}, just as she did in Γ_n. To see that $\sigma_{n+1}(s') \neq i$, notice that role s' cannot be the terminator role, by Lemma 11 and the fact that $x_i \in C_s(h^*_n)$ and $x_i \in C^{\subsetneq}_{s'}(h^*_n)$. Thus, only unlurked objects are possible for role s', and so if $\sigma_{n+1}(s') = i$, since x_i is i's top unlurked object, she would clinch it at some history $h' \subsetneq h^*_n \subseteq h^*_{n+1}$, which is a contradiction. Therefore, $\sigma^{-1}_{n+1}(i) \neq s, s'$, and so there must be some $\ell \neq j_1, \ldots, j_P, k_1, \ldots, k_n$ such that $x_i \in C_\ell(h^*_{n+1})$, as desired.

It remains to consider $h^*_{n+1} \subsetneq h^*_n$. Then, there are $P+n-1$ lurkers at h^*_{n+1}, and, since h^*_{n+1} is not the terminating history, it must be agent k_n that moves at h^*_{n+1}. This also implies that k_n is in role s or s'. If $\sigma_{n+1}(s') = k_n$, then k_n is in the same role in Γ_{n+1} as in Γ_n, and would pass at h^*_{n+1} in Γ_{n+1} as she did in Γ_n,which is a contradiction. Therefore, $\sigma_{n+1}(s) = k_n$.

We claim that role s is not an active at history h^*_{n-1}. Indeed, notice that because i clinches at h^*_{n-1} in Γ_{n-1}, we have that $x_i \in C^{\subsetneq}_s(h^*_{n-1})$. This implies that role s is not the terminator role, which follows by Lemma 11 and the fact that $x_i \in C_{s'}(h')$ for some $h' \supsetneq h^*_{n-1}$. This implies that only unlurked objects are possible for role s when she is called to play. Thus, if role s were an active non-lurker at history h^*_{n-1}, then, in Γ_n, when $\sigma_n(s) = i$, agent i is offered to clinch x_i at some $h' \subseteq h^*_{n-1}$. Since we know that only unlurked objects are possible, and x_i is i's top unlurked object, she would clinch at $h' \subsetneq h^*_n$ in Γ_n, which is a contradiction. Since role s is not active at h^*_{n-1}, there are two roles that are not s that are active non-lurkers at h^*_{n-1} and such that both have been offered to clinch x_i. At h^*_{n+1} in Γ_{n+1}, at least one of these roles must still be active and not assigned to any agent $j_1, \ldots, j_P, k_1, \ldots, k_n, i$. Thus, there must be some $\ell \neq j_1, \ldots, j_P, k_1, \ldots, k_n, i$ such that ℓ is an active non-lurker that does not move at h^*_{n+1} and $x_i \in C^{\subsetneq}_\ell(h^*_{n+1})$, as desired. This concludes the analysis of subcase 2.2.2, and hence of case 2.2.

The above shows that in all cases, there is some $\ell \neq j_1, \ldots, j_P, k_1, \ldots, k_n, i$ such that ℓ is an active non-lurker that does not move at h^*_{n+1} and $x_i \in C^{\subsetneq}_\ell(h^*_{n+1})$ in game Γ_{n+1}. Recall that, by Lemma 18, $\lambda^*_{n+1} \geq P+n-1$. If $\lambda^*_{n+1} > P+n-1$, then there are at least $P+n$ lurked objects at h^*_{n+1}, and the only way i can be ranked in the $(P+n+1)^{th}$ position in $\succ_{n+1}$ is if she is coded in the first step. Since there is some agent $\ell \neq i$ such that $x_i \in C^{\subsetneq}_\ell(h^*_{n+1})$, i can at best tie with this agent. If $\lambda^*_{n+1} = P+n-1$, then by the structure of $\succ_{n+1}$, it must be agent k_n that clinches at h^*_{n+1}, and there is no tie at the end of step 1. This means that ℓ is not coded in step 1, and so the continuation game that begins step 2 of the coding algorithm starts with agent ℓ being offered x_i. Now, for i to be ranked immediately after k_n, she must be ordered first in step 2 of the coding algorithm, and for i to be ordered first without ties, either she must lurk x_i and it is the first lurked object, or i must clinch x_i while there are no lurked objects and before x_i has not been offered to another active non-lurker. However, neither of these can occur because ℓ begins the step 2 continuation game being offered x_i. Therefore, in $\succ_{n+1}$, i must tie with some agent that we label k_{n+1}. This completes the proof of Lemma 15. ■

Before proving Lemmas 16 and 17, we first state and prove Lemma 19, on which both rely. To state the lemma, we introduce the following notation: define Q to be the step of the coding algorithm in which i is coded in game Γ_n. Also, define h_n^{q*} to be the history at which the first object is clinched in step q of the coding algorithm for game Γ_n.

Lemma 19. *Assume that there exists a sequence of role assignment functions Σ as defined in the statement of Lemma 5, and such that $n \geq 2$. If either (i) $Q = 1$, or (ii) $Q \geq 2$ and at h_n^{1*}, there is an agent ℓ that is an active nonlurker at h_n^{1*} that does not move at h_n^{1*}, and $x_i \in C_\ell^{\subsetneq}(h_n^{1*})$, then, in $\succ_{n+1}$, agent i must tie with some agent k_{n+1}.*

Proof of Lemma 19. We start with the following lemma.

Lemma 20. *Consider two games Γ_A and Γ_B, with corresponding role assignment functions σ_A and σ_B, and resulting agent orderings $\succ_A$ and $\succ_B$. Assume that $\succ_A$ begins as $\{i, j\} \succ_A \cdots$, and $\succ_B$ begins as: $j \succ_B i \cdots$. Further, assume that in game Γ_A, there is some history h where j moves such that: (i) $h \subseteq h_A^*$, (ii) $x_i \in C_j(h)$ (iii) $x_j \notin C_j^{\subseteq}(h)$ (iv) $x_i, x_j \notin C_i^{\subseteq}(h)$. Then:*

(a) If agent j clinches at h_A^* in Γ_A, then in Γ_B, agent j clinches at $h_B^* \subsetneq h_A^*$, and there is some agent $k \neq i$ that is an active non-lurker at h_B^* such that $x_i \in C_k(h_B^*)$.

(b) In $\succ_B$, agent i must tie with some other agent k.

Proof of Lemma 20. Let h_A^* and h_B^* be the first time an agent clinches in Γ_A and Γ_B. Notice that by the structure of $\succ_A$, at history h_A^*, there are two active roles, and both are nonlurkers at h_A^*; label the roles s and s', and, wlog, let $\sigma_A(s) = i$ and $\sigma_A(s') = j$. Using these definitions, we can write the presumptions of the lemma as (ii) $x_i \in C_j(h)$ (iii) $x_j \notin C_j^{\subseteq}(h)$ (iv)$x_i, x_j \notin C_i^{\subseteq}(h)$. Also, notice that $h \subseteq h_A^*$ implies that there are no lurkers at h, and so the only roles that may possibly be active at h are s and s'. Finally, since x_i and x_j tie for the top ranking in $\succ_A$, it must be that x_i is i's favorite object among all objects and x_j is j's favorite object among all objects. Therefore, by greedy strategies, if at any history i is able to clinch x_i, she will do so, and the same for j and x_j.

Part (a). The structure of $\succ_A$ implies that $x_j \in C_s(h')$ for some $h' \subsetneq h_A^*$. Now, consider Γ_B. The only way for j to be ranked first without ties is that $\sigma_B(s) = j$, and j clinches at

$h^*_B \subsetneq h^*_A$.[47] Let $k := \sigma_B(s')$, and notice that, by the assumptions of the lemma, $x_j \notin C^{\subseteq}_s(h)$, and so $h \subsetneq h^*_B$, and therefore $x_i \in C^{\subsetneq}_{s'}(h^*_B)$. It is clear that $k \neq j$. Further, $k \neq i$ because if $k = i$, then $x_i \in C_i(h)$ in Γ_B, and thus, i would clinch x_i at $h \subsetneq h^*_B$ in Γ_B, which contradicts that the first clinching in Γ_B is j clinching at h^*_B. Therefore, $\sigma_B(s') = k$ for some $k \neq i, j$, and k is an active non-lurker that does not move at h^*_B such that $x_i \in C_k(h^*_B)$ in Γ_B.

Part (b). If j clinches at h^*_A, ,then by part (a), there is an agent k such that $x_i \in C^{\subsetneq}_k(h^*_B)$ and k is not coded in the coding step initiated at h^*_B in Γ_B. Let $h^{**}_B \supsetneq h^*_B$ be the history at which the next clinching occurs in Γ_B. Since k was offered x_i in the previous coding step, but is still active, at the initial history of the continuation game that begins step 2, k is offered to clinch x_i again (see Remark 2). Thus, x_i cannot be the first lurked object on the initial passing path of the continuation game form (Remark 3), and so there must be no lurked objects at h^{**}_B. For i to be coded next, she must be active at h^{**}_B, and since there are no lurked objects, there are two active agents, i and k. If k clinches at h^{**}_B, it is obvious that i can at best tie; if i clinches at h^{**}_B, i once again ties with k, because $x_i \in C_k(h^{**}_B)$.

The other possibility is that i clinches at h^*_A, which implies that $x_i \in C_{s'}(h')$ for some $h' \subsetneq h^*_A$. For j to be ranked first without ties in $\succ_B$, at h^*_B, either (a) there are lurkers, and x_j is the first lurked object or (b) there are no lurkers, j clinches x_j, and x_j has not been offered to another non-lurker that is active at h^*_B. There are 3 cases:

Case: $\sigma_B(s') = i$. In this case, i would clinch x_i at h and would be ranked first in $\succ_B$, which is a contradiction.[48]

Case: $\sigma_B(s') = j$. Here, j is in the same role in both games, and therefore $\sigma_B(s) = \ell \neq i$, which follows because if $\ell = i$, then both j and i are in the same roles, and we would get the same initial orderings for $\succ_A$ and $\succ_B$, a contradiction. This implies that $h^*_B \supsetneq h^*_A$, because if $h^*_B \subseteq h^*_A$, then, since j is in the same role, she would clinch at h^*_B in Γ_A, a contradiction.[49] Now, notice that because x_i has been offered to both j and ℓ (weakly) prior to h^*_A, x_i cannot be the first or second lurked object of the game. This means that, for i to be ranked second, there can be at most one lurked object at h^*_B, and if it exists it must be x_j that is lurked.

If x_j is lurked at h^*_B, it must be by either j or ℓ. If it is lurked by ℓ, then x_j must clinch at h^*_B, but, since there is only one lurker, this implies that ℓ must clinch an unlurked object, and will be ranked second (possibly tied with i). If x_j is lurked by j, then ℓ is still an active non-lurker at h^*_B such that $x_i \in C_\ell(h^*_B)$. If i clinches x_i at h^*_B, she will tie with ℓ; if i does not clinch, she can at best tie with ℓ (and may be ranked strictly lower). In either case, the

[47] The only other way for j to be ranked first without ties is that x_j is the first lurked object; however, this cannot obtain, because $x_j \in C_s(h')$ at some history h' where there are no lurkers.

[48] Note that x_j has not been offered to any agent at h, by the presumptions of the lemma.

[49] The case $h^*_B = h^*_A$ is ruled out because i moves at h^*_A in Γ_A, and this history is controlled by role s, not s'.

result holds.

The final case is that nothing is lurked at h_B^*. This implies that x_j clinches at h_B^*, but again, $x_i \in C_\ell(h_B^*)$. Therefore, at the initial history of the continuation game that begins step 2 of the coding algorithm, x_i is offered to agent ℓ. Let h_B^{**} be the first time an object is clinched in this continuation game. Since x_i is offered to ℓ at the initial history, x_i cannot be the first lurked object, and so, for i to be ranked first in this continuation game without ties, she must clinch x_i while it is unlurked and has not been offered to another active non-lurker. But, we have just seen that x_i is offered to ℓ at the initial history, and so this cannot hold.

Case: $\sigma_B(s') = \ell'$ for some $\ell' \neq i, j$. First, notice that $\sigma_B(s) = \ell$ for some $\ell \neq i$. To see this, assume that $\ell = i$. Then, i is in the same role in Γ_A and Γ_B. This implies that $h_B^* \subsetneq h_A^*$, because if h_A^* is reached in Γ_B, i would clinch there, and be ranked above j. But, $h_B^* \subsetneq h_A^*$ implies that j is not ranked first in $\succ_B$ (since she is not yet active at h_B^*), which is a contradiction.

If $\sigma_B(s) = j$, then for j to be ranked first in $\succ_B$, either (a) x_j is the first lurked object on the path to h_B^* or (b) there are no lurked objects at h_B^*, j clinches x_j at h_B^*, and x_j has not been offered to another active non-lurker. Notice that $h_B^* \supsetneq h$,[50] which implies that agent $x_i \in C_{\ell'}(h_B^*)$. But, then it is impossible for i to be ranked immediately after $j \succ_B$ without ties, which is a contradiction.

If $\sigma_B(s) \neq j$, then roles s and s' are assigned to agents ℓ and ℓ' in Γ_B, neither of which are j or i. So, for j to be ranked first without ties, x_j must be the first lurked object (and be lurked by either ℓ or ℓ'), and j must clinch it at some $h_B^* \supsetneq h_A^*$. For i to be ranked second without ties in this case, there must be two lurked objects at h_B^*,[51] and x_i must be the second lurked object (after x_j). But, at the history $h'' \supsetneq h_A^*$ where x_j becomes lurked, one of agents ℓ or ℓ' is an active non-lurker who has been previously offered to clinch x_i, and so x_i cannot be the next lurked object, a contradiction. ■

Continuing with the proof of Lemma 19, first, consider $Q = 1$. Then, all agents $j_1, \ldots, j_P, k_1, \ldots, k_n, i$ are coded in step 1 of game Γ_n. By Remark 4, $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{n-1}}$ all become lurked on the initial passing path of the game form, and further, since $n \geq 2$, we can apply Lemma 15 to conclude that i ties in $\succ_{n+1}$.

It remains to consider $Q \geq 2$. Since we have assumed that $P + 1$ agents are coded in step 1, all agents j_p have been coded in the first step, and so the agent who is coded first in step Q of the coding algorithm of Γ_n is $k_{\bar{n}}$ for some $\bar{n} < n$. So, the subcoding of $\succ_n$ starting from

[50] In case (a), this follows because there are no lurkers at h; in case (b), it follows from the assumption of the lemma that $x_j \notin C_s^{\subseteq}(h)$.

[51] Since j clinches at h_B^*, if there is no other lurked object at h_B^*, the only active agents are ℓ, ℓ', and j, and so one of ℓ or ℓ' will be ranked above i in $\succ_B$, which is a contradiction.

step Q is:

$$k_{\bar{n}} \succ_n k_{\bar{n}+1} \succ_n \cdots \succ_n k_{n-1} \succ_n \{i, k_n\}.$$

Consider the sequence of games $\Gamma_{\bar{n}}, \Gamma_{\bar{n}+1} \ldots, \Gamma_n, \Gamma_{n+1}$. Notice that the codings for all of these games are exactly the same, up to agent $k_{\bar{n}-1}$. Therefore, by Lemma 4, all agents $j_1, \ldots, j_P, k_1, \ldots, k_{\bar{n}-1}$ are in the same roles in all of these games. In particular, agent $k_{\bar{n}-1}$ is the last agent coded in step $Q-1$ in all of these games, and the initial history of the continuation game that begins step Q is the also the same in all of these games; label this history $h_\varnothing^Q$. Now, applying the coding algorithm to the sequence of continuation games of $\Gamma_{\bar{n}}, \ldots, \Gamma_n, \Gamma_{n+1}$ starting from history $h_\varnothing^Q$, we get the sub-codings:

$$\begin{aligned}
&\{i, k_{\bar{n}}\} \succ_{\bar{n}} \cdots \\
&k_{\bar{n}} \succ_{\bar{n}+1} \{i, k_{\bar{n}+1}\} \succ_{\bar{n}+1} \cdots \\
&\vdots \\
&k_{\bar{n}} \succ_n k_{\bar{n}+1} \succ_n \cdots \succ_n k_{n-1} \succ_n \{i, k_n\} \succ_n \cdots. \\
&k_{\bar{n}} \succ_{n+1} k_{\bar{n}+1} \succ_{n+1} \cdots \succ_{n+1} k_n \succ_{n+1} i \cdots
\end{aligned}$$

There are two cases.

Case 1: $\bar{n} < n$. In this case, we can apply Lemma 15 to the game form starting from $h_\varnothing^Q$ to conclude that i must tie in $\succ_{n+1}$. To see this, simply note that upon reindexing to start from $h_\varnothing^Q$ rather than $h_\varnothing$, the condition "$n \geq 2$" becomes "$n \geq \bar{n}+1''$. Then, we have that $x_{k_{\bar{n}}}, \ldots, x_{k_{n-1}}$ all become lurked on the initial passing path of the game form starting from $h_\varnothing^Q$, which follows from Remark 4, $n \geq \bar{n}+1$, and the fact that i is coded in the initial step of the continuation game of Γ_n starting from $h_\varnothing^Q$. Thus, all of the conditions of Lemma 15 are satisfied.

Case 2: $\bar{n} = n$. In this case, the games we are concerned with are Γ_n and Γ_{n+1}, with subcodings:

$$\begin{aligned}
&\{i, k_n\} \succ_n \cdots \\
&k_n \succ_{n+1} i \cdots.
\end{aligned} \tag{C}$$

Notice that here, we can no longer apply Lemma 15, since we do not have at least two games in which i ties in the sequence. Our goal is to apply Lemma 20 instead, but to do so, we must show that the conditions (i)-(iv) of Lemma 20 are satisfied at $h_\varnothing^Q$.

For each each coding step $q = 1, \ldots, Q$ of game Γ_n, let h_n^{q*} denote history at which the first object is clinched in the q^{th} coding step, and let $h_n^{\varnothing_q}$ denote the initial history that begins the continuation game for the next step, after all of the agents in step $q-1$ are coded (in particular, $h_n^{\varnothing_1} = h_\varnothing$, and $h_n^{1*} = h_n^*$ in our earlier notation). In $\succ_n$, all agents who are coded in steps $q < Q$ are ranked strictly, without ties. Let k_{n^q} denote the agent who is coded ***last*** in the q^{th} step. With this notation, the subcoding from the q^{th} step is:

$$k_{n^{q-1}+1} \succ_n k_{n^{q-1}+1} \succ_n \cdots \succ_n k_{n^q},$$

where we define $n^0 = 0$. It is possible that $k_{n^{q-1}+1} = k_{n^q}$, in which case only one agent is coded in step q. Since there are no ties, agent k_{n^q} ends the coding step by clinching an unlurked object that has not been offered to another non-lurker who is active at h_n^{q*}.

Claim 6. For all $q < Q$, there is an agent $\ell \neq k_1, \ldots, k_{n^q}, i$ such that ℓ is an active nonlurker at h_n^{q*} that does not move at h_n^{q*}, and $x_i \in C_\ell^{\subsetneq}(h_n^{q*})$.

Claim 6 (whose proof can be found immediately after the proof of this lemma) implies that when we reach step Q in Γ_n, at the initial history of the continuation game $h_n^{\varnothing_Q}$ that begins this step, there is some agent $\ell \neq k_1, \ldots, k_{n-1}, i$ such that $x_i \in C_\ell(h_n^{\varnothing_Q})$. Since the subcodings for $\succ_n$ in this step begin with a tie between i and k_n (see Equation C), it must be that $\ell = k_n$. Finally, we apply Lemma 20 by setting $A = n$, $B = n+1$, $h = h_n^{\varnothing_Q}$, $j = k_n$, and $i = i$ to conclude that i must tie in $\succ_{n+1}$.[52] ■

Proof of Claim 6. By the supposition of the lemma, at h_n^{1*}, there is an agent ℓ that is an active nonlurker at h_n^{1*} that does not move at h_n^{1*}, and $x_i \in C_\ell^{\subsetneq}(h_n^{1*})$. It is clear that ℓ is not coded (since there is no tie in step 1), and so $\ell \neq k_1, \ldots, k_{n^1}$. To see that $\ell \neq i$, note that if $\ell = i$, then step 2 begins with agent i being offered to clinch x_i. If i is not coded in step 2, then step 3 begins with i being offered x_i, etc.. The same continues up to and including step Q, in which i is coded. Since i is coded first in step Q (tying with k_n) x_i is her top object among those that remain at the beginning of step Q. Since $x_i \in C_i(h_n^{(Q-1)*})$, agent i begins step Q by being offered to clinch x_i at the initial history of this step. Since x_i is her top remaining object, she would clinch it, and thus would not tie with k_n, which is a contradiction. Thus, the statement holds for $q = 1$.

Now, consider step $q = 2$ of game Γ_n, which begins at $h_n^{\varnothing_2}$ and produces the subcoding:

$$k_{n^1+1} \succ_n k_{n^1+2} \succ_n \cdots \succ_n k_{n^2} \succ_n .$$

Case 1: $n^2 = n^1 + 1$. Then only one agent, agent k_{n^1+1}, is coded in step 2 of game Γ_n, which begins with the continuation game that starts at history $h_n^{\varnothing_2}$. The result from step 1

[52] Condition (i) of Lemma 20 is immediate. For condition (ii) was just shown. Condition (iii) holds because, if $x_{k_n} \in C_{k_n}(h_n^{\varnothing_Q})$, then k_n would immediately clinch it at $h_n^{\varnothing_n}$, and would not tie with i in $\succ_n$. Condition (iv) is also immediate, as i has not yet been called to move at $h_n^{\varnothing_Q}$.

implies that at $h_n^{\varnothing_2}$, some agent $\ell \neq k_1, \ldots, k_{n^1}, i$ moves and $x_i \in C_\ell(h_n^{\varnothing_2})$.

Since k_{n^1+1} is the only agent coded in step 2 of Γ_n, and does not tie, she must clinch $x_{k_{n^1+1}}$ at h_n^{2*} in Γ_n while it is unlurked, and before it is offered to another active non-lurker. Now, since $\succ_n$ and $\succ_{n^1+1}$ are the same up til agent k_{n^1}, Lemma 4 implies that $h_n^{\varnothing_2} = h_{n^1+1}^{\varnothing_2}$; for shorthand, define $h^{\varnothing_2} := h_n^{\varnothing_2} = h_{n^1+1}^{\varnothing_2}$. The second step continuation games of Γ_{n^1+1} and Γ_n both start from $h^{\varnothing_2}$, and lead to the initial subcodings:

$$\{i, k_{n^1+1}\} \succ_{n^1+1} \cdots$$
$$k_{n^1+1} \succ_n \cdots$$

Let role s be the role that moves at $h^{\varnothing_2}$, and role s' be the second role that becomes active on the initial passing path of the game form starting from $h^{\varnothing_2}$. These two roles exist because there is an initial tie in $\succ_{n^1+1}$, and in Γ_{n^1+1}, s and s' are assigned to k_{n^1+1} and i, in some manner. If $\sigma_{n^1+1}(s) = i$, then i would clinch at $h^{\varnothing_2}$ in Γ_{n^1+1}, and would not tie, a contradiction. Therefore, $\sigma_{n^1+1}(s) = k_{n^1+1}$, which implies that $x_{k_{n^1+1}} \notin C_{k_{n^1+1}}(h_n^{\varnothing_2})$; indeed, if this were true, then k_{n^1+1} would clinch it at $h_{n^1+1}^{\varnothing_2}$ in Γ_{n^1+1}, which contradicts that k_{n^1+1} ties in $\succ_{n^1+1}$.

Now, if $\sigma_n(s) = k_{n^1+1}$, then k_{n^1+1} is in the same role in both games, and so it must be i that clinches at $h_{n^1+1}^{2*}$, which means that $x_i \in C_{s'}(h_{n^1+1}^{2*})$.[53] It also means that $h_n^{2*} \supsetneq h_{n^1+1}^{2*}$, and that $\sigma_n(s') \neq i$, and so, there exists some agent $\ell' \neq i$ such that in Γ_n, $x_i \in C_{\ell'}(h_n^{2*})$, which is what we wanted to show.

Last, if $\sigma_n(s) \neq k_{n^1+1}$, then $\sigma_n(s') = k_{n^1+1}$. Thus, in this case, there is some agent other agent ℓ such that $\sigma_n(s) = \ell$. Again, $\ell \neq i$, because $x_i \in C_s(h_n^{\varnothing_2})$. Thus, when k_{n^1+1} clinches at h_n^{2*} in Γ_n, we have $x_i \in C_\ell^{\subsetneq}(h_n^{2*})$, as desired.

Case 2: $n^2 > n^1 + 1$**.** Consider games $\Gamma_{n^1+1}, \Gamma_{n^1+2}, \ldots, \Gamma_n$ and notice that the codings for all of these games are equivalent up to agent k_{n^1}. Therefore, by Lemma 4, all agents $k_1, \ldots, k_{n^1}$ are in the same roles in all of these games, and so these agents will take the same actions, which implies that, for each of these games, step 2 of the coding algorithm begins at the same history of the game form, which we denote $h^{\varnothing_2}$.

Consider the continuation game form starting at $h^{\varnothing_2}$, and recall that $h_{n'}^{2*}$ is the first time an object is clinched in step 2 of game $\Gamma_{n'}$, which is also the first time an object is clinched in step 1 of the continuation game beginning at $h^{\varnothing_2}$. Notice that by the structure of $\succ_n$, the objects $x_{k_{n^1+1}}, \ldots, x_{k_{n^2-1}}$ are lurked at h_n^{2*} in Γ_n, while $x_{k_{n^2}}$ is not, i.e., objects $x_{k_{n^1+1}}, \ldots, x_{k_{n^2-1}}$ are the first lurked objects (in order) along the initial passing path of the

[53]If k_{n^1+1} clinched first in Γ_{n^1+1} and Γ_n, and is in the same role, then the subcodings $\succ_{n^1+1}$ and $\succ_n$ would be the same up to k_{n^1+1}, which is a contradiction.

game form, beginning at $h^{\varnothing_2}$.

The subcodings of games $\Gamma_{n^1+1}, \Gamma_{n^1+2}, \dots, \Gamma_{n^2+1}$ beginning at history $h^{\varnothing_2}$ are:

$$\{i, k_{n^1+1}\} \succ_{n^1+1} \cdots$$
$$\vdots$$
$$k_{n^1+1} \succ_{n^2} k_{n^1+2} \succ_{n^2} \cdots \succ_{n^2} k_{n^2-1} \succ_{n^2} \{i, k_{n^2}\} \succ_{n^2} \cdots.$$
$$k_{n^1+1} \succ_{n^2+1} k_{n^1+2} \succ_{n^2+1} \cdots \succ_{n^2+1} k_{n^2} \succ_{n^2+1} \{i, k_{n^2+1}\} \cdots$$

By Lemma 15 applied to the continuation game and subcodings beginning at $h^{\varnothing_2}$, in Γ_{n^2+1}, at $h^{2*}_{n^2+1}$, there is an agent ℓ such that ℓ is an active non-lurker at $h^{2*}_{n^2+1}$ that does not move at $h^{2*}_{n^2+1}$ and $x_i \in C^{\subsetneq}_\ell(h^{2*}_{n^2+1})$. Since $\succ_n$ is equivalent to $\succ_{n^2+1}$ up to agent k_{n_2}, and agent k_{n_2} is the last agent in a coding step of game Γ_n, we have that $h^{2*}_n = h^{2*}_{n^2+1}$, by Lemma 4. This implies that at h^{2*}_n, there is an agent ℓ that is an active non-lurker at h^{2*}_n that does not move at h^{2*}_n and $x_i \in C^{\subsetneq}_\ell(h^{2*}_n)$ (which may or may not be the same such agent in Γ_{n^2+1}, depending on the role assignment functions).

It remains to show that $\ell \neq k_1, \dots, k_{n^2}, i$. It is clear that $\ell \neq k_1, \dots, k_{n^2}$, since all of these agents are coded by the end of step 2 in Γ_n, while agent ℓ is not. If $\ell = i$, step 3 begins with agent i being offered to clinch x_i. If i is not coded in step 3, then i continues to be active in step 4, which begins with i being offered x_i, etc.. The same continues up to and including step Q, in which i is coded. Since i is coded first in step Q (tying with k_n) x_i is her top object among those that remain at the beginning of step Q. Since $x_i \in C_i(h^{(Q-1)*}_n)$, agent i begins step Q by being offered to clinch x_i at the initial history of this step. Since x_i is her top remaining object, she would clinch it, and thus would not tie with k_n, which is a contradiction. Therefore, $\ell \neq i$. This completes the result for $q = 2$.

We then just repeat the arguments for the $q = 2$ case for all $q = 3, 4, \dots, Q-1$, which completes the proof of Lemma 19. ■

Proof of Lemma 16. We begin by showing the result for $n = 1$, as part of the following claim.

Claim 7. Assume that there exist σ_0 and σ_1 such that:

$$j_1 \cdots j_P \succ_0 i \succ_0 k_1 \succ_0 \cdots$$
$$j_1 \cdots j_P \succ_1 \{i, k_1\} \succ_1 \cdots.$$

Then:

(a) We have $h^*_0 \subsetneq h^*_1$, and the agent that moves at h^*_0 in Γ_0 is agent i.

(b) If there exists a σ_2 such that $j_1 \cdots j_P \succ_2 k_1 \succ_2 i \cdots$, then $h_0^* \subsetneq h_2^*$. Further, in $\succ_2$, agent i must tie with some other agent k_2.

(c) If x_{k_1} is not the $(P+1)^{th}$ lurked object on the initial passing path, then in Γ_2, agent k_1 clinches at $h_2^* \subsetneq h_1^*$. Further, at h_2^*, there is an active non-lurker $\ell \neq j_1, \ldots, j_P, i, k_1$ such that $x_i \in C_{k_2}^{\subsetneq}(h_2^*)$.

The proof of this claim can be found at the end of the proof of the lemma. Now, consider a sequence Σ such that $n \geq 2$. We will show that i must tie in $\succ_{n+1}$.

In game Γ_n, i is coded in some step of the coding algorithm with some subset of the agents $j_1 \ldots, j_P, k_1, \ldots, k_{n-1}$. Let Q be the step number in which i is coded in game Γ_n. The goal is to apply Lemma 19, which the following claim allows us to do.

Claim 8. If $Q \geq 2$, then at h_n^*, there is an agent ℓ that is an active non-lurker at h_n^* that does not move at h_n^* and $x_i \in C_\ell^{\subsetneq}(h_n^*)$.

The proof of this claim is found below, immediately after the proof of Claim 7. Given Claim 8, we can apply Lemma 19 to conclude that i must tie in $\succ_{n+1}$, which completes the proof of Lemma 16. ■

Proof of Claim 7. Since we assume there are at least P lurkers at h_1^*, by the structure of $\succ_1$, there are exactly P lurkers at h_1^*. This implies that the first P lurked objects are $x_{j_1}, \ldots, x_{j_P}$. Additionally, objects x_i and x_{k_1} are unlurked at h_1^*, and so x_i and x_{k_1} are agent i and k_1's favorite objects among the set of those that are unlurked at h_1^*, respectively.

Part (a). Suppose not, then the passing structure of histories implies that $h_1^* \subseteq h_0^*$. Notice that at h_1^*, there must be two active non-lurker roles.

Case 1: $P = 0$**.** In this case, there are no agents j_p, so at h_1^*, there are exactly two active roles, label them s and s', and wlog, let $\sigma_1(s) = i$ and $\sigma_1(s') = k_1$. If i clinches at h_1^* in Γ_1, then $x_i \in C_{s'}^{\subsetneq}(h_1^*)$ and $x_i \in C_s(h_1^*)$. Now, for i to be ranked first without ties in $\succ_0$ is either (i) x_i is the first lurked object of the game or (ii) i clinches x_i first as an unlurked object, and it has not been offered to another active non-lurker. However, $h_1^* \subseteq h_0^*$ implies that neither (i) nor (ii) can obtain, as x_i has been offered to both active non-lurkers at h_1^*, which is a contradiction.

If k_1 clinches at h_1^* in Γ_1, then $x_{k_1} \in C_s^{\subsetneq}(h_1^*)$ and $x_{k_1} \in C_{s'}^{\subseteq}(h_1^*)$. Now, $h_1^* \subseteq h_0^*$ implies that in Γ_0, $\sigma_0^{-1}(k_1) \neq s, s'$.[54] Since k_1 is not in either of these roles, there is some $\ell \neq i, k_1$ that is active at h_1^* in Γ_0 and is such that $x_i \in C_\ell^{\subseteq}(h_1^*)$. Notice also that since x_{k_1} has been offered to both active agents at h_1^*, it cannot be the second lurked object along the initial passing path (Remark 3), and so for k_1 to be ranked second, there can be at most 3 active agents

[54] If $\sigma_0^{-1}(k_1) = s$, then k_1 would clinch at some $h' \subsetneq h_1^*$; if $\sigma_0^{-1}(k_1) = s'$, then k_1 is in the same role in Γ_0 and Γ_1, and thus would clinch at $h_0^* = h_1^*$, and would once again tie for first in $\succ_0$.

at h_0^*, in particular agents i, k_1, and ℓ. If k_1 moves at h_0^*, i must be lurking x_i, and k_1 will tie with agent ℓ. If ℓ moves at h_0^*, it is clear k_1 will not be ranked second without ties. If i moves at h_0^*, then there must be no lurked objects at h_0^*.[55] But, since $h_1^* \subseteq h_0^*$, we have $x_{k_1} \in C_\ell(h_0^*)$, and so, since ℓ was not coded in the first step, she begins the second step by being offered x_{k_1} at the initial history of the continuation game. Thus, it is impossible for k_1 to be ranked first without ties in this continuation game, a contradiction.

Case 2: $P \geq 1$**.** In this case, there is at least one lurker j_p at h_1^*. Further, at h_1^*, there are P active lurker roles for the objects $x_{j_1}, \ldots, x_{j_P}$, and 2 active non-lurkers roles; label the role that moves at h_1^* as s, and the other active nonlurker at h_1^* as s'. There are three subcases, depending on who is in role s.

Subcase 2.1. $\sigma_1(s) = i$**.** In this case, we have $\sigma_1(s') = k_1$ and $x_i \in C_{s'}^{\subsetneq}(h_1^*)$. We first claim that i cannot be active at h_1^* in Γ_0. First, notice that i cannot move at h_1^* in Γ_0, because if she did, she would choose the same action at h_1^* in both games, and would tie in $\succ_0$, just as she did in $\succ_1$. So, $\sigma_0(s) \neq i$. Next, assume i is a lurker at h_1^* in Γ_0, for some lurked object $x_{j_1}, \ldots, x_{j_P}$. Note that x_i cannot be the next object lurked along the initial passing path because it has been offered to (both) active non-lurkers at h_1^*, so at h_0^*, there must be no newly lurked objects, and roles s and s' are still active non-lurkers. The first coding of step Γ_0 thus ends when i clinches x_i, which is unlurked. But, because $h_1^* \subseteq h_0^*$, x_i has been offered to both role s and s' at h_0^*, and one of these is an active non-lurker who does not move at h_0^*, and so i would tie with this agent in $\succ_0$.

Second, assume that $\sigma_0(s') = i$. Then, notice that $x_i \in C_{s'}(h')$ for some $h' \subsetneq h_1^*$. We claim that i would clinch x_i at this history. Indeed, at h', role s' is an active non-lurker that is not the terminator.[56] This means that only unlurked objects are possible for the agent in this role, and since x_i is i's favorite unlurked object, she will clinch it at h', by greedy strategies. Therefore, i is not active at h_1^* in Γ_0.

Now, i is not active at h_1^* in Γ_0, but there are two active non-lurkers, those in roles s and s', and both of these have been offered x_i. Thus, x_i cannot be the next lurked object along the initial passing path of the game form, and so there can be no newly lurked objects at h_0^*. But then, i is not active at h_0^* (since no new agent can become active unless something else becomes lurked), and so i is not coded in this step, which contradicts that she is ranked $(P+1)^{th}$ in $\succ_0$.

Subcase 2.2: $\sigma_1(s) = k_1$**.** In this case, we have $x_{k_1} \in C_{s'}(h')$ for some $h' \subsetneq h_1^* \subseteq h_0^*$ and

[55] If there were, it must be x_i. It cannot be lurked by k_1, since this would mean x_i is her top object, which is a contradiction. So, it must be lurked by some $\ell \neq i, k_1$, and so ℓ will be ranked ahead of or tie with k_1 in $\succ_1$.

[56] This follows from Lemma 11. If this role were the terminator, then role s could not be offered x_i at $h_1^* \supsetneq h'$.

$x_{k_1} \in C_s(h_1^*)$. This implies that x_{k_1} cannot be either of the next two lurked objects on the initial passing path of the game form (if they exist). Since k_1 is ordered immediately after i in $\succ_0$ and k_1 does not tie, there can be at most one newly lurked object at h_0^*, and it must be x_i.

We next claim that k_1 cannot be active at h_1^* in Γ_0. It is clear that $\sigma_0(s) \neq k_1$, because otherwise k_1 would clinch at h_1^* in Γ_0, and once again tie in $\succ_0$. We also have that $\sigma_0(s') \neq k_1$. To see why, notice that s' is not the terminator role (see footnote 56). So, only unlurked objects are possible for the agent in this role, and thus, if k_1 was in this role, she would clinch x_{k_1} at $h' \subsetneq h_0^*$, since it is her favorite unlurked object. Last, if k_1 lurks some object x_{j_p} at h_1^*, then she strictly prefers x_{j_p} to x_{k_1}. It then must be some agent $j_{p'}$ that moves at h_0^* and clinches a lurked object $x_{j_{p'}}$. This means that $j_{p'}$ is in the terminator role. We claim that $\sigma_0^{-1}(j_{p'}) \neq s, s'$. We know (see footnote 56) that s' is not the terminator role, so $\sigma_0^{-1}(j_{p'}) \neq s'$. If $\sigma_0(s) = j_{p'}$, then s is the terminator role. But, this contradicts that k_1 clinched x_{k_1} first at h_1^* in Γ_1, since in that game she was in the terminator role and so x_{j_p} is possible for her, and she strictly prefers it. Therefore, in Γ_0, $j_{p'}$ is in some role s'' that was not active at h_1^*. This implies that one of s or s' is still active at h_0^* in Γ_0, and whoever it is, this agent has been offered x_{k_1} prior to h_0^*. So, k_1 would tie with this agent in $\succ_0$, a contradiction.

So, k_1 is not active at h_1^* in Γ_0. So, there is some agent $\ell \neq j_1, \ldots, j_P, i, k_1$ that is active at h_1^* in Γ_0. This agent cannot be a lurker at h_0^*, since if she were, she would necessarily be coded in step 1, and, as x_{k_1} is not lurked at h_0^*, k_1 could at best tie with her. Thus, $\sigma_0^{-1}(\ell) = s$ or s', and no matter which, we have $x_{k_1} \in C_\ell(h_1^*)$. If x_{k_1} is clinched in step 1, then k_1 can at best tie with ℓ. If k_1 is not coded in step 1, then in at the start of the continuation game for step 2, ℓ is offered x_{k_1}. But, if this is the case, then k_1 cannot be ordered first without ties in step 2, which contradicts the definition of $\succ_0$.

Subcase 2.3: $\sigma_1(s) = j_p$ for some $p = 1, \ldots, P$. In this case, agent j_p is clinching a lurked object at h_1^*, and so h_1^* is the terminating history. Then, $h_1^* \subseteq h_0^*$ implies that $h_1^* = h_0^*$. Thus, in Γ_0, x_i is the first (and only) unlurked object clinched in step 1, and so $x_i \notin C_{s'}^{\subsetneq}(h_1^*)$. So, because there is a tie in Γ_1, it must be that $x_{k_1} \in C_{s'}^{\subsetneq}(h_1^*)$.

Next, we claim that in Γ_0, k_1 is not active at h_1^*. Indeed, k_1 is not in role s (as that is occupied by j_p). She also cannot be a lurker, because she is not coded in step 1 (which ends with i). Finally, consider role s'. Notice that s' is not the terminator role (because that is role s), and so, if k_1 were in role s', she would clinch x_{k_1} at some history $h' \subsetneq h_1^*$ at which it was offered to her, a contradiction.

Therefore, there is some $\ell \neq j_1, \ldots, j_P, i, k_1$ that is such that $\sigma_0(s') = \ell$ and $x_{k_1} \in C_\ell(h_1^*)$. Since ℓ is not coded in step 1, she begins the continuation game for step 2 by being offered x_{k_1}. Thus, k_1 cannot be ordered first in step 2 without ties, which is a contradiction.

The above shows that $h_0^* \subsetneq h_1^*$. To finish the proof of part (a), we must show that agent i moves at h_0^* in Γ_0. Notice that $h_0^* \subsetneq h_1^*$ and the structure of $\succ_1$ implies there can be at most P lurkers at h_0^*. First, if there are no lurkers ($P = 0$) at h_0^*, then, it is clear that i must move at h_0^*, as that is the only way she can be ranked first without ties. Now, presume that $P > 0$. If it is some j_p that moves at h_0^*, then j_p clinches a lurked object x_{j_p}, which implies that h_0^* is the terminating history, which contradicts $h_0^* \subsetneq h_1^*$. Therefore, no agent $j_1, \ldots, j_P$ can move at h_0^*. Since there can be at most P lurkers at h_0^*, given that i is ranked $(P+1)^{th}$ without tying, the only other possibility is that it is agent i that moves at h_0^* and clinches x_i.

Part (b). We first show that $h_0^* \subsetneq h_2^*$. By part (a), $h_0^* \subsetneq h_1^*$. This means that agent i cannot move at h_0^* in Γ_1. Nor can any potential agent j_p, because if they did, they would be clinching a lurked object, which means h_0^* is the terminating history, which contradicts $h_0^* \subsetneq h_1^*$. Therefore, it must be k_1 that moves at h_0^* in Γ_1.

By way of contradiction suppose that $h_0^* \subsetneq h_2^*$ fails; because of the passing structure of this histories, it means that $h_2^* \subseteq h_0^*$. The structure of $\succ_2$ implies that k_1 clinches at h_2^* in Γ_2, which also means that h_2^* and h_0^* are controlled by different roles, and further $h_2^* \subsetneq h_0^*$.[57] So, in Γ_0, it must be some agent $\ell \neq j_1, \ldots, j_P, i, k_1$ that moves at h_2^*. But then, we have $x_{k_1} \in C_\ell(h_0^*)$, so at the initial history of the continuation game that begins step 2, agent ℓ is offered x_{k_1}, and so k_1 cannot be ordered first in step 2, which is a contradiction to the definition of Γ_0. Therefore, $h_0^* \subsetneq h_2^*$.

Thus, we have $h_0^* \subsetneq h_1^*, h_2^*$, and so agent i does not move at h_0^* in Γ_1 or Γ_2.

Case 1: Agent k_1 moves at h_0^* in Γ_2. Here, k_1 is in the same role as in Γ_1, and so $h_1^* \subsetneq h_2^*$. This implies that i must clinch at h_1^* in Γ_1, and so i does not move at h_1^* in Γ_2. If some j_p moves at h_1^* in Γ_2, then this agent must also clinch at h_2^*, and she must clinch a lurked object. This means that i must be a lurker for some $x_{j_{p'}}$, and so she strictly prefers $x_{j_{p'}}$ to x_i. But then, the agent that moves at h_1^* is in the terminator role, and so in Γ_1, i is in the terminator role, and since she clinches x_i at h_1^* , this implies that x_i is her top object (lurked or unlurked) by Lemma 11(v), which is a contradiction. So, it must be some $\ell \neq j_1, \ldots, j_P, i, k_1$ that moves at h_1^* in Γ_2, and so $x_i \in C_\ell^{\subsetneq}(h_2^*)$ in Γ_2. Since ℓ is not coded in step 1, she is offered x_i at the initial history of the continuation game that begins step 2. Therefore, i cannot be ranked first without ties in this continuation game.

Case 2: Some agent $j_1, \ldots, j_P$ moves at h_0^* in Γ_2. This agent, say j_p, must be the one clinching at h_2^* (since j_p is not a lurker at h_0^*, but ultimately receives a lurked object), and she must clinch a lurked object. This implies that the agent who moves at h_0^* is in the

[57] If they were the same role, then k_1 is in this role in Γ_1, and would clinch at h_2^* in Γ_1, which is a contradiction.

terminator role, and that h_2^* is the terminating history, so $h_1^* \subseteq h_2^*$. Let r be the other role that is active at h_1^*. Since there is a tie in $\succ_1$, this role must be such that either $x_i \in C_r^{\subseteq}(h_1^*)$ or $x_{k_1} \in C_r^{\subseteq}(h_1^*)$. In the latter subcase, x_{k_1} cannot be the next lurked object along the passing path (from h_1^*), and so there must be no newly lurked objects at h_2^*. Next, notice that $\sigma_2(r) \neq k_1$, because otherwise, k_1 would clinch x_{k_1} at the history $h' \subsetneq h_1^*$ where it was offered in Γ_2. Thus, k_1 can at best tie with the agent $\sigma_2(r)$, which is a contradiction.

For the subcase $x_i \in C_r^{\subseteq}(h_1^*)$, if $\sigma_2(r) = k_1$, then there is some agent $\ell \neq j_1, \dots, j_P, k_1$ who is a lurker for some $x_{j_1}, \dots, x_{j_P}$. We also have $\ell \neq i$. This is because the agent who moves at h_0^* is in the terminator role, and so in Γ_0, i is in this role, and since she clinches, x_i is her top available object (lurked or unlurked), and therefore i cannot lurk any of the x_{j_p}'s. Therefore, agent ℓ will be ranked ahead of i in $\succ_2$, a contradiction.[58] We also cannot have $\sigma_2(r) = i$, because i would clinch x_i at the history $h' \subsetneq h_1^*$ at which she was offered x_i. Thus, $\sigma_2(r) = \ell$ for some $\ell \neq j_1, \dots, j_P, i, k_1$. Agent ℓ is not coded in step 1, and thus, she is offered x_i at the initial history of the continuation game that begins step 2, and so i cannot be ranked first without tying in step 2.

Part (c). If x_{k_1} is not the $(P+1)^{th}$ lurked object, then, because k_1 is ordered without tying in $\succ_2$, at h_2^*, k_1 must clinch x_{k_1}, and it has not been offered to another active non-lurker. Notice also that $h_0^* \subsetneq h_2^*$ implies that i does not move at h_0^* in Γ_1 or Γ_2, and that k_1 moves at h_0^* in Γ_1. If k_1 moves at h_0^* in Γ_2, then she is in the same role in both games, and so $h_1^* \subsetneq h_2^*$. This also means that i moves at h_1^* in Γ_1 (because if it was k_1, then x_{k_1} is offered to both active roles at h_1^*, and so in Γ_2, k_1 would clinch at some $h' \subsetneq h_2^*$). Thus, x_i has been offered to both active non-lurker roles at h_1^*. This implies that i cannot be active at h_1^* in Γ_2, and so there is some $\ell \neq j_1, \dots, j_P, i, k_1$ such that $x_i \in C_\ell(h_2^*)$ in Γ_2. If k_1 does not move at h_0^* in Γ_2, then it is some $\ell \neq j_1, \dots, j_P, i, k_1$ that moves at h_0^*. In either case, we have $x_i \in C_\ell^{\subsetneq}(h_2^*)$ in Γ_2. ∎

Proof of Claim 8. (See above for the statement of the claim).Since it is without loss of generality to assume that there are at least P lurkers at h_n^*, there are two cases. Recall that k_1 is ranked strictly, without ties, in $\succ_n$.

Case 1: There are exactly P lurkers at h_n^*. In this case, k_1 is the last agent coded in step 1 of Γ_n. Consider game Γ_2, and notice that $\succ_n = \succ_2$ up to agent k_1. Since agent k_1 is the last agent in a coding step, by Lemma 4, all agents $j_1, \dots, j_P, k_1$ are in the same roles in Γ_2 and Γ_n, and $h_n^* = h_2^*$. Further, notice that x_{k_1} is not the $(P+1)^{th}$ lurked object along the initial passing path,[59] and so, by Claim 7 part (c), there is an agent ℓ that is an active

[58] Note that x_i cannot be lurked at h_2^*, since it has been offered to agent j_p at h_0^*, who is the terminator.

[59] If k_1 clinches at h_n^*, then x_{k_1} is offered to an active non-lurker, and so cannot be the next lurked object along the initial passing path; if some j_p clinches at h_n^*, then they are clinching a lurked object, and so h_n^* is the terminating history, which again implies that x_{k_1} is not $(P+1)^{th}$ lurked object along the initial passing

non-lurker at h_2^* that does not move at h_2^* and $x_i \in C_\ell^{\subsetneq}(h_2^*)$. Since $h_2^* = h_n^*$, the result holds.

Case 2: There are strictly greater than P lurkers at h_n^*. In this case, the objects $x_{j_1}, \ldots, x_{j_P}, x_{k_1}, \ldots, x_{k_{n'-1}}$ are lurked at h_n^*, while $x_{k_{n'}}$ is not, where $n > n' > 1$.[60] Consider game $\Gamma_{n'+1}$, and notice that $\succ_n$ is equivalent to $\succ_{n'+1}$ up to agent $k_{n'}$. Therefore, by Lemma 4, all agents $k_1, \ldots, k_{n'}$ are in the same roles in all of these games, and $h_n^* = h_{n'+1}^*$. By Lemma 15, in $\Gamma_{n'+1}$, at $h_{n'+1}^*$, there is an active agent ℓ such that ℓ is an active non-lurker at $h_{n'+1}^*$ that does not move at $h_{n'+1}^*$ and $x_i \in C_\ell^{\subsetneq}(h_{n'+1}^*)$. Since $h_n^* = h_{n'+1}^*$, the result holds. ■

Proof of Lemma 17. By the assumption that $n, m \geq 2$ in Σ and Σ', we have that there exist (at least) the following codings:

$$j_1 \cdots j_P \succ_1 \{i, k_1\} \succ_1 \cdots$$
$$j_1 \cdots j_P \succ_2 k_1 \succ_2 \{i, k_2\} \cdots$$
$$j_1 \cdots j_P \succ_2' i \succ_2' \{k_1, k_2'\} \cdots.$$

We start by presenting the following two conditions, one of which, when combined with prior lemmas, will imply that Statement 1 of the lemma holds, and the other of which will imply Statement 2 of the lemma holds.

- Condition 2: In Γ_2, at h_2^* there is an active non-lurker ℓ such that ℓ does not move at h_2^* and $x_i \in C_\ell^{\subsetneq}(h_2^*)$.
- Condition 2′: In Γ_2', at $h_{2'}^*$, there is an active non-lurker ℓ such that ℓ does not move at $h_{2'}^*$ and $x_{k_1} \in C_\ell^{\subsetneq}(h_{2'}^*)$.[61]

We first show that these conditions imply the lemma. Then, we show that one of these conditions must hold.

We will show that Condition 2 implies that Statement 1 of Lemma 17 holds. The two statements are symmetric, so this will also show that Condition 2′ implies Statement 2 of Lemma 17.

To show Condition 2 implies Statement 1, we use Lemma 19. So, consider the sequence

path (because no such object exists).

[60] Because $Q \geq 2$, the last agent coded in step 1 of Γ_n is at most k_{n-1}, which means that $x_{k_{n-1}}$ is not lurked, i.e., the last lurked object is at most $x_{k_{n-2}}$, which is why we have $n' < n$.

[61] We use $h_{2'}^*$ (instead of $h_2^{'*}$) to denote the first history at which an object is clinched in game Γ_2' (under role assignment σ_2').

of codings

$$
\begin{aligned}
&j_1 \cdots j_P \succ_1 \{i, k_1\} \succ_1 \cdots \\
&j_1 \cdots j_P \succ_2 k_1 \succ_2 \{i, k_2\} \cdots \\
&j_1 \cdots j_P \succ_3 k_1 \succ_3 k_2 \succ_3 \{i, k_3\} \succ_3 \cdots \\
&\vdots \\
&j_1 \cdots j_P \succ_n k_1 \succ_n k_2 \succ_n k_3 \succ_n \cdots \succ_n k_{n-1} \succ_n \{i, k_n\} \succ_n \cdots \\
&j_1 \cdots j_P \succ_{n+1} k_1 \succ_{n+1} k_2 \succ_{n+1} k_3 \succ_{n+1} \cdots \succ_{n+1} k_{n-1} \succ_{n+1} k_n \succ_{n+1} i \cdots
\end{aligned}
$$

Recall that it is wlog to assume that there are at least P lurked objects at $h^*_{n'}$ for each n'. We claim further that in this case, there are exactly P lurked objects at $h^*_{n'}$ for each n'. For $n' = 1$, this follows from the fact that i and k_1 tie. For $n' > 1$, the next ordered agent is k_1. So, if there were $p > P$ lurked objects at $h^*_{n'}$, the $(p+1)^{th}$ lurked object would have to x_{k_1}, which contradicts the supposition of the lemma. Therefore, for all $n' = 1, \ldots, n+1$, at $h^*_{n'}$ in game $\Gamma_{n'}$, there are exactly P lurked objects, and by definition, these must be $x_{j_1}, \ldots, x_{j_P}$, in this order.

Next, notice that for all $n' \geq 2$, since there are exactly P lurked objects at $h^*_{n'}$, the set of agents coded in step 1 of $\Gamma_{n'}$ must be $j_1, \ldots, j_P, k_1$. In particular, this is true for Γ_2 and Γ_n, and since $\succ_2$ is equivalent to $\succ_n$ up to agent k_1, by Lemma 4, all of these agents are in the same roles in both games, and $h^{1*}_n = h^*_2$. By Condition 2, there is some agent ℓ such that ℓ is an active non-lurker that does not move at h^*_2 and $x_i \in C^{\not\subseteq}_\ell(h^*_2)$. Since $h^{1*}_n = h^*_2$, we have that in Γ_n, there is some agent ℓ' that is an active non-lurker at h^{1*}_n and that does not move at h^{1*}_n and $x_i \in C^{\not\subseteq}_\ell(h^{1*}_n)$. Further, $Q \geq 2$. Thus, all of the conditions of Lemma 19 are satisfied, and we conclude that i must tie with some agent k_{n+1} in $\succ_{n+1}$.

We complete the proof of Lemma 17 by showing that at least one of Condition 2 or Condition 2′ must hold. This assertion is proven as Lemma 21 below. ■

Lemma 21. *Assume that there are three codings:*

$$
\begin{gathered}
j_1 \cdots j_P \succ_A \{i, k\} \succ_A \cdots \\
j_1 \cdots j_P \succ_B i \succ_B \cdots \\
j_1 \cdots j_P \succ_C k \succ_C \cdots
\end{gathered}
$$

such that:

- *At each of h^*_A, h^*_B, h^*_C, the objects $x_{j_1}, \ldots, x_{j_P}$ are all lurked, in this order, and*
- *Neither x_i nor x_k are the $(P+1)^{th}$ lurked object on the initial passing path of the game.*

Then, one of the following conditions must hold:

Condition (B): In Γ_B, at h_B^* there is an active non-lurker ℓ such that ℓ does not move at h_B^* and $x_k \in C_\ell^{\subsetneq}(h_B^*)$.

Condition (C): In Γ_C, at h_C^*, there is an active non-lurker ℓ such that ℓ does not move at h_C^* and $x_i \in C_\ell^{\subsetneq}(h_C^*)$.

Proof of Lemma 21. First, notice that in each of the games, there must be exactly P lurkers at h_γ^* for $\gamma = A, B, C$. It is a presumption of the lemma that there are at least P lurkers. To see that there are at most P lurkers, notice that, for Γ_A, this holds because i and k tie. In Γ_B, it holds because x_i is not the next lurked object along the initial passing path, and thus, x_i must be the first—and since there is no tie, only—unlurked object that is coded in step 1. The same applies to Γ_C. Therefore, in Γ_A, there are exactly $P + 2$ agents coded in step 1, while in Γ_B and Γ_C, there are exactly $P + 1$ agents coded in step 1.

In Γ_A, at h_A^*, there are P active lurker roles and two active non-lurker roles. The objects $x_{j_1}, \ldots, x_{j_P}$ are lurked, and x_i and x_k are unlurked. Let s be the active non-lurker role that moves at h_A^*, and s' the role of the other active non-lurker. One of x_i or x_k must be the first unlurked object that is clinched in step 1 of the coding algorithm, either at h_A^* itself, or in the chain of assignments that follows. Assume it is x_i (a symmetric argument works if it is x_k). This implies that $x_i \in C_{s'}^{\subsetneq}(h_A^*)$, and $\sigma_A(s') = k$. There are two cases, depending on who is in role s.

Case 1: $\sigma_A(s) = j_p$ for some p. Agent j_p must be clinching a lurked object at h_A^*, which implies that h_A^* is the terminating history, and s is the terminator role. This means that s' is *not* the terminator role, and so $x_k \notin C_{s'}^{\subsetneq}(h_A^*)$; indeed, if this were true, then x_k would have clinched it in Γ_A, because it is her favorite unlurked object and only unlurked objects are possible for a non-lurker who is not the terminator (Lemma 11(iv)). It also means that agent i must be a lurker for some object $x_{j_{\bar{p}}}$, and thus, agent i strictly prefers $x_{j_{\bar{p}}}$ to x_i.

Now, consider game Γ_C. The agents coded in step 1 of Γ_C are $j_1, \ldots, j_P, k$, and so it must be one of these agents that moves at h_C^*.

Subcase 1.1: The agent that clinches at h_C^* is some $j_{p'}$. Here, h_C^* must also be the terminating history, and so $\sigma_C(s) = x_{j_{p'}}$ and $h_A^* = h_C^*$. Since k is coded in step 1, she must then be a lurker, and so there is some other agent $\ell \neq j_1, \ldots, j_P, k$ such that $\sigma_C(s') = \ell$. We claim that $\ell \neq i$. Indeed, if $\ell = i$, then there is some history $h' \subsetneq h_C^*$ such that $x_i \in C_i(h')$. Since s' is not the terminator role, only unlurked objects are possible for i in Γ_C, and since x_i is her top unlurked object, she would clinch at h', a contradiction. Therefore, $\sigma_C(s') = \ell \neq i$, and Condition (C) holds.

Subcase 1.2: Agent k clinches at h_C^* in Γ_C. Here, we have $\sigma_C(s) = k$, because, as we saw above, $x_k \notin C_{s'}^{\subsetneq}(h_A^*)$ and h_A^* is the terminating history, so $h_C^* \subseteq h_A^*$. Let $h' \subsetneq h_A^*$ be the history at which role s' is offered to clinch x_i.

If $h_C^* \supsetneq h'$, then, by similar logic to subcase 1.1, $\sigma_C(s') = \ell$ for some $\ell \neq j_1, \ldots, j_P, k, i$, and Condition (C) holds.

Finally, consider $h_C^* \subsetneq h'$.[62] In Γ_B, since there are exactly $P+1$ agents coded in step 1, x_i is the first (and only) unlurked object that is clinched, and since there is no tie, it has not been offered to another active non-lurker. This implies that $h_B^* \subseteq \tilde{h} \subsetneq h_A^*$. Since h_B^* is not the terminating history, it must be an unlurked object that is clinched, and therefore, it must be i that clinches x_i. If $\sigma_B(s) = i$, then i is in the terminator role, and would not clinch x_i first at h_B^* (recall that she prefers $x_{j_{\bar{p}}}$ to x_i). Thus, it must be that $\sigma_B(s') = i$, and i clinches x_i at h_B^*. If $h_B^* \subsetneq h_C^*$, then by similar logic to the above, Condition (C) holds. If $h_C^* \subsetneq h_B^*$, then $x_k \in C_s^{\subsetneq}(h_B^*)$ for the agent in role s. Notice that $\sigma_B(s) \neq k$, because if so, then k has the same roles in Γ_B and Γ_C, and so would clinch at $h_C^* \subsetneq h_B^*$ in Γ_B, a contradiction. It is also immediate that $\sigma_B(s) \neq j_1, \ldots, j_P$, since they must be in the lurker roles for their respective objects. Thus, $\sigma_B(s) = \ell$ for some $\ell \neq j_1, \ldots, j_P, i, k$, and Condition (B) holds.

Case 2: $\sigma_A(s) = i$. We once again have that role s' is not the terminator role,[63] and so, as in Case 1, $x_k \notin C_{s'}^{\subseteq}(h_A^*)$. Once again, consider game Γ_C. As in Case 1, there are two subcases.

Subcase 2.1: The agent that clinches at h_C^* in Γ_C is some $j_{p'}$. Here, $j_{p'}$ clinches a lurked object at h_C^*, and so h_C^* is the terminating history. This implies that $h_A^* \subseteq h_C^*$, and $\sigma_C(s) = j_{p'}$. But then, notice that the agent in role s' is an active non-lurker at h_C^* that does not move at h_C^*, and $x_i \in C_{s'}^{\subsetneq}(h_C^*)$. Since this agent is not coded in step 1, we know that $\sigma_C(s') \neq j_1, \ldots, j_P, k$. If $\sigma_C(s') = i$, then i is offered to clinch x_i at some $h' \subsetneq h_C^*$, and since s' is not the terminator role, only unlurked objects are possible for her, and therefore, since x_i is i's top object, she would clinch at h', a contradiction. Thus, $\sigma_C(s') = \ell$ for some $\ell \neq j_1, \ldots, j_P, i, k$, and $x_i \in C_\ell^{\subsetneq}(h_C^*)$, i.e., Condition (C) holds.

Subcase 2.2: The agent that clinches at h_C^* in Γ_C is k. Since k clinches first, and x_k is unlurked, all lurked objects are immediately assigned to their lurkers, which implies that j_p is in the lurker role for x_{j_p} for all $p = 1, \ldots, P$.

If $h_A^* \subseteq h_C^*$, then, at h_C^*, there are two active non-lurkers, $\sigma_C(s)$ and $\sigma_C(s')$, and both have been offered x_i. One of these must be k. If $\sigma_C(s') = k$, then notice that $\sigma_C(s) \neq i$, because if $\sigma_C(s) = i$, then i is in the same role in Γ_A and Γ_C, and would clinch at h_A^* in Γ_C, which contradicts that k clinches first in Γ_C. Thus, $\sigma_C(s) = \ell \neq i$. If $\sigma_C(s) = k$, then if $\sigma_C(s') = i$, then i is in the non-terminator role, and $x_i \in C_i(\tilde{h})$ for some $\tilde{h} \subsetneq h_A^* \subseteq h_C^*$, and since x_i is i's favorite unlurked object, she will clinch it at $\tilde{h}$, a contradiction. Therefore, in

[62] Note that $h_C^* = h'$ is ruled out because role s' moves at h', while role s moves at h_C^*.

[63] This follows from Lemma 11.

either case, there is some agent $\ell \neq j_1, \ldots, j_P, i, k$ such that $x_i \in C_\ell^{\subsetneq}(h_C^*)$, and Condition (C) holds.

It remains to consider $h_C^* \subsetneq h_A^*$. Here, we must have $\sigma_C(s) = k$, because if $\sigma_C(s') = k$, then as we showed above, $x_k \notin C_{s'}^{\subsetneq}(h_A^*)$, which contradicts that k clinches at h_C^*. Now, consider Γ_B. In Γ_B, since there are exactly $P+1$ agents coded in step 1, x_i is the first (and only) unlurked object that is clinched, and the agents coded in step 1 are $j_1, \ldots, j_P, i$.

If $h_B^* \subseteq h_C^*$, then, $h_B^* \subsetneq h_A^*$, and h_B^* is not the terminating history. Thus, in Γ_B, agent i must move at h_B^* and clinch x_i. This implies that $\sigma_B(s') = i$, because if $\sigma_B(s) = i$, then i has the same role in Γ_A and Γ_B and clinches at both h_B^* and h_A^*, which contradicts that $h_B^* \subsetneq h_A^*$. Further, this means that $h_B^* \neq h_C^*$, because role s moves at h_C^* and role s' moves at h_B^*. Thus, at h_C^* in Γ_C, we have $x_i \in C_{s'}^{\subsetneq}(h_C^*)$. We cannot have $\sigma_C(s') = i$, because i would clinch at h_B^* in Γ_C, a contradiction. Therefore, $\sigma_C(s') = \ell$ for some $\ell \neq j_1, \ldots, j_P, i, k$ and $x_i \in C_\ell^{\subsetneq}(h_C^*)$, and thus, Condition (C) holds.

If $h_C^* \subsetneq h_B^*$, then if some $j_{p'}$ clinches at h_B^* in Γ_B, then h_B^* is the terminating history, and $h_A^* \subseteq h_B^*$. But then, there is an active non-lurker—the agent $\sigma_B(s')$—that has been offered to clinch x_i prior to h_B^*, and so i would at best tie with this agent in $\succ_B$, a contradiction. Thus, it must be i that clinches at h_B^* in Γ_B, which implies that $\sigma_B^{-1}(i) = s$ or s'. If $\sigma_B^{-1}(i) = s$, then i has the same roles in Γ_A and Γ_B, and so $h_A^* = h_B^*$, and i would tie with the agent in role s' in $\succ_B$, a contradiction. Thus, $\sigma_B(s') = i$. This means that h_C^* and h_B^* are controlled by different roles, and $x_k \in C_s^{\subsetneq}(h_B^*)$. Finally, we cannot have $\sigma_B(s) = k$, because then k is in the same role as Γ_C, and would clinch at $h_C^* \subsetneq h_B^*$ in Γ_B. So, we must have $\sigma_B(s) = \ell$ for some $\ell \neq j_1, \ldots, j_P, i, k$, and in Γ_B, $x_k \in C_\ell^{\subsetneq}(h_B^*)$. Therefore, Condition (B) holds.

Finally, notice that all of this was done under the assumption that x_i was the first unlurked object that was clinched in step 1 of the coding algorithm in Γ_A. The other possibility is that this object is x_k. However, everything is symmetric, and so the exact same argument, swapping the i and k, shows that either Condition (B) or Condition (C) must hold in this case as well. ■