

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

Marek Pycia and Peter Troyan*

March 2026

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 (2023b) (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, Parag Pathak, Ran Shorrer, Erling Skancke, Utku Ünver, Martin Valdevenito, 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, the 2024 Conference on Mechanism and Institution Design, and the 2025 Workshop on Normative Aspects of Economic Policy. 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.¹ 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 in which the random allocation is created by drawing an ordering of agents uniformly at random and calling the agents, one-by-one, to select their favorite object from those that were not selected by earlier agents.²

The popularity of Random Priority largely derives from its desirable efficiency, fairness, and incentive (or simplicity) properties. A long-standing open question has been whether any *other* mechanism also satisfies such properties, or whether Random Priority is the unique mechanism to do so. On the positive side, when $n = 3$, Bogomolnaia and Moulin (2001) characterize Random Priority as the unique mechanism satisfying Pareto efficiency, strategy-proofness, and equal treatment of equals; Brandt, Greger, and Romen (2023) extended this result to $n = 4$.³ However, Basteck and Ehlers (2024) show it does not extend further: for $n \geq 5$, they provide an example of a mechanism that is strategy-proof, Pareto efficient, satisfies equal treatment of equals, and not only is different than Random Priority but also leads to distributions of individual agents' outcomes that are different than in Random Priority.⁴

A characterization of Random Priority thus requires strengthening at least one of strategy-proofness, Pareto efficiency, or equal treatment of equals.⁵ Random Priority already fails natural strengthenings of Pareto efficiency: beyond very small markets it is neither ordinally efficient in the sense of Bogomolnaia and Moulin (2001) nor ex ante efficient in the sense of

¹In the main body of the paper we focus on the canonical environment with equal numbers of agents and objects. In Appendix D, we show that our results remain valid with no restrictions on the relative cardinalities of these sets.

²Random Priority is also called Random Serial Dictatorship, see e.g., Abdulkadiroğlu and Sönmez (1998).

³The result for $n = 4$ was also proven by Ehlers and Unver (private communication, 2010) and Echenique, Pomatto, Root, Sandomirskiy, and Tamuz (private communication, 2020). Pycia and Ünver (2015) discuss methodological tools developed in a failed attempt to prove the conjecture.

⁴Prior to Basteck and Ehlers (2024), Erdil (2014) shows that the classic axioms of Pareto efficiency, equal treatment, and strategy-proofness do not characterize Random Priority when the number of agents and objects are different, and Pycia and Troyan (2023a) construct mechanisms different from Random Priority that are strongly obviously strategy-proof (and hence obviously strategy-proof and strategy-proof), Pareto efficient, and satisfy equal treatments of equals.

⁵We focus on finite markets of any fixed size. In large markets with many copies of each object, Liu and Pycia (2011) show that asymptotically all ordinally efficient, equal treatment, strategy-proof mechanisms with small agents have the same marginal distributions as Random Priority (Random Priority satisfies ordinal efficiency asymptotically (Che and Kojima, 2010) and, in large markets, ex post and ordinal efficiency coincide (Liu and Pycia, 2011)).

Zhou (1990).⁶ Random Priority does, however, satisfy natural strengthenings of the equal-treatment property and of strategy-proofness: it is *symmetric* and its extensive-form implementation is *obviously strategy-proof (OSP)*.⁷ Symmetry, which is also called anonymity, is a natural fairness requirement that says that the outcome of a mechanism should not depend on the identities of the agents. Obvious strategy-proofness is a strengthening of strategy-proofness introduced by Li (2017), who establishes its desirability as a simplicity criterion for designers who want to ensure that a mechanism can be recognized as as strategy-proof by agents unable to engage in contingent reasoning. Moreso, the possibility of obviously strategy-proof extensive-form implementation matters even for designers restricted to static mechanisms, as it allows such designers to explain the mechanism in a simple way.⁸

Can these properties be used to obtain a characterization of Random Priority? That is, is Random Priority the only mechanism that is Pareto efficient, symmetric, and obviously strategy-proof? The main result of this paper is to show that the answer is yes. Further, counterexamples of Pycia and Troyan (2023a) and Basteck and Ehlers (2024) 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.

Our work is related to the literature that shows the equivalence between Random Priority and certain trading-style mechanisms based on Gale’s top trading cycles (TTC) algorithm. Abdulkadiroğlu and Sönmez (1998) and Knuth (1996) show that Random Priority is equivalent to the core from random endowments, which first randomly allocates objects to agents and then allows them to trade via TTC. Pathak and Sethuraman (2011) show further that Random Priority is also equivalent to Partitioned Random Priority and Partitioned Random Endowment. They also extend this analysis to school choice problems, in which schools can have multiple seats, and show that it is equivalent for a school district to use a common lottery to determine the priority ordering at all schools or multiple independent lotteries across schools.⁹ Carroll (2014) provides a more general equivalence of symmetrizations of Pápai’s

⁶Bogomolnaia and Moulin (2001) show that Pareto efficiency is equivalent to ordinal efficiency for $n = 3$ but no ordinally efficient and strategy-proof mechanism satisfies equal treatment for $n \geq 4$. For $n \geq 3$, Zhou (1990) shows that there is no strategy-proof and ex-ante efficient mechanism that satisfies equal treatments of equals. Random Priority fails ex-ante efficiency even asymptotically, as can be inferred from Abdulkadiroğlu et al. (2011), Featherstone and Niederle (2016), and Miralles (2008).

⁷Random Priority fails envy-freeness (another natural strengthening of equal treatment) as shown by Bogomolnaia and Moulin (2001).

⁸For the importance of simple descriptions, see e.g. Bó and Hakimov (2023), Breitmoser and Schweighofer-Kodritsch (2019), and Gonczarowski, Heffetz, and Thomas (2023). An experimental literature—starting with Li (2017)—also shows that correct play is indeed more prevalent in obvious strategy-proof mechanisms than merely strategy-proof ones.

⁹Pycia (2019) shows that symmetrizations of all Abdulkadiroğlu and Sönmez (2003) top trading cycle mechanisms are equivalent.

(2000) hierarchical exchange mechanisms.¹⁰ While related, our results are distinct because we consider a different class of mechanisms, all obviously strategy-proof and efficient ones: not all efficient obviously strategy-proof mechanisms are trading mechanisms and trading mechanisms, while efficient, are in general not obviously strategy-proof.¹¹

Our analysis also contains methodological innovations that might be more generally useful. For instance, we show how to reduce the problem of characterizing symmetric mechanisms to the simpler problem of characterizing mechanisms that are obtained by symmetric randomizations over agents’ roles in a base mechanism, so called symmetrizations of the base mechanism. The remainder of our proof strategy builds at a high level on the bijective approach pioneered by Abdulkadiroğlu and Sönmez (1998) and Knuth (1996) and extended by Carroll (2014) to some mechanisms where a priority list for each object is given in terms of n abstract *roles*. The allocation is then determined by (i) randomly selecting a *role assignment function* that determines which agent is in which role and (ii) running the TTC algorithm under the selected role assignment. This can then be used to show there is a bijection between the role assignments and Random Priority orderings, which implies the mechanisms are equivalent. Our proof extends this role assignment approach to the class of obviously strategy-proof and Pareto efficient mechanisms, where the roles correspond to players in an extensive-form game. We use Pycia and Troyan’s (2023b) characterization of obviously strategy-proof mechanisms and results from Bade and Gonczarowski (2017) on Pareto efficient and obviously strategy-proof mechanisms to obtain a representation of this class. Given the richness of the class of games we allow, our construction of the bijection from role assignments to orderings is more complex than the linear approach employed by the previous work. We first construct a (one-to-one) injection from role assignments to merely partial orderings, which we call “codings”. We show that for any coding, any way of breaking ties delivers a serial dictatorship ordering that results in the same outcome as the original mechanism. The bulk of the proof is then devoted to showing that there exists a way to break the ties and create complete orderings that preserves the one-to-one property, thereby constructing a bijection between role assignments and serial dictatorship orderings.

Following on our work, Basteck (2024) provided a second axiomatic characterization of

¹⁰Bade (2020) further extends it to Pycia and Ünver (2017)’s trading-cycle mechanisms.

¹¹Li (2017) shows that the classic TTC mechanism in which each agent begins by owning a unique object is not obviously strategy-proof. More general formulations of TTC where each object has a priority relation of the agents describing the “property rights” over the objects might be obviously strategy-proof in some cases. When each object has the exact same priority relation, TTC is trivially equivalent to RP (and thus is obviously strategy-proof), but more generally, Troyan (2019) and Mandal and Roy (2022) show that this cannot be extended very far: the class of priority structures for which TTC is obviously strategy-proof is restrictive, essentially requiring that at any point, “ownership” of all of the objects is divided among at most two agents.

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 The Allocation Problem

2.1 Environment

Let \mathcal{N} be a set of **agents** and \mathcal{X} a set of **objects**, where $|\mathcal{N}| = |\mathcal{X}|$.¹² Each agent $i \in \mathcal{N}$ has a **strict preference relation**, $>_i$, over the set of objects \mathcal{X} , where for any $x, y \in \mathcal{X}$, $x >_i y$ denotes that object x is strictly preferred to object y . We also refer to $>_i$ as agent i 's **type**, and write $x \succeq_i y$ to denote that either $x >_i y$ or $x = y$. Let \mathcal{P} denote the set of possible types, which consists of all possible strict rankings of the objects. We write $>_{\mathcal{N}} = (>_i)_{i \in \mathcal{N}}$ to denote a profile of types, one for each agent. A (deterministic) **allocation** $\mu : \mathcal{N} \rightarrow \mathcal{X}$ is a bijective function that assigns exactly one of the objects to each agent $i \in \mathcal{N}$. Let \mathcal{M} be the set of deterministic allocations.

2.2 Extensive Game Forms

To determine the final allocation that will be implemented, the planner designs a **game form**, Γ . We consider imperfect-information, extensive game forms 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 $h \in \mathcal{H}$. We use h_{\emptyset} to denote the initial history, and terminal histories are denoted with bars, \bar{h} . 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

¹²We relax this assumption in Appendix D.

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} , $\mu(i) = x$.

Given the agents' preferences over outcomes, the game form Γ determines an incomplete information extensive-form game, and, with slight abuse of terminology, in the sequel, we often refer to Γ simply as a “game” or “extensive-form game.”

2.3 Strategies, Mechanisms, and Equivalence

A **strategy** $S_i(>_i)$ for type $>_i$ of agent i specifies an action for each information set, $S_i(>_i)(I_i) \in A(I_i)$.¹³ We use $S = ((S_i(>_i))_{>_i \in \mathcal{P}})_{i \in \mathcal{N}}$ 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 . Given a profile of strategies and type profile $>_{\mathcal{N}}$, we say that an information set I is **on the path of play**, or **on path**, if there exists a strategy for Nature such that the path from the root of the game to the terminal history that is reached when agents follow $S(>_{\mathcal{N}})$ and Nature follows its strategy passes through I . For a type $>_j$ of agent i , we say that information set I is **on-path for $>_i$** if there exists a type profile of the remaining agents $>_{\mathcal{N} \setminus \{i\}}$ and Nature such that I is on-path for $(>_i, >_{\mathcal{N} \setminus \{i\}})$.

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 $>_{\mathcal{N}}$, 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.

¹³We restrict attention to pure strategies. Allowing for mixed strategies would not change any of our results.

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.

The Random Priority mechanism has desirable simplicity, efficiency, and fairness properties. Li (2017) shows that Random Priority is obviously strategy-proof.¹⁴ A strategy $S_i(>_i)$ is **obviously dominant** for player i (of type $>_i$) in game Γ if for each on-path information set $I^* \in \mathcal{I}_i$, the worst possible outcome for i according to $>_i$ 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 $a' \neq S_i(I^*)$.¹⁵ If there exists a profile of strategies S such that $S_i(>_i)$ is obviously dominant in Γ for all i and all $>_i$, 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 obviously 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 $>_{\mathcal{N}}$, 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; we then also say that the induced direct mechanism is Pareto efficient. 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.

¹⁴Pycia and Troyan (2023b) show that it satisfies the even stronger simplicity standards of one-step simplicity (OSS) and strong obvious strategy-proofness (SOSP).

¹⁵Li (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.

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.¹⁶ 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 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 strategy-proof, 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. *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 outline the main arguments.¹⁷ Given any arbitrary obviously strategy-proof mechanism, we transform it into an equivalent *millipede mechanism*. Millipede mechanisms are a class of mechanisms introduced by Pycia and Troyan (2023b).¹⁸ They are perfect information mechanisms in which Nature

¹⁶In 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 $\sigma : \mathcal{N} \rightarrow \mathcal{N}$.

¹⁷For $|\mathcal{N}| = 1$, 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. Bogomolnaia and Moulin (2001) analyze the three-agent case, but their approach is different and, because of its reliance on ordinal efficiency, not applicable beyond three agents.

¹⁸Pycia and Troyan (2023b) define millipede mechanisms for a broad class of economic environments that includes the allocation problem we study. We adapt their definitions to our environment.

either moves once, at h_\emptyset , or Nature has no moves. Each time an agent $i \in \mathcal{N}$ is called to move, she is presented with some subset of objects she can take immediately, and leave the game; we refer to this as *clinging* an object, and the corresponding action as a *clinging action*.¹⁹ The agent may also be offered the opportunity to take a *passing action*. If an agent passes, she remains in the game, with the possibility of being offered additional objects for clinging in the future. A key feature of millipede mechanisms is that at each history, there can be at most one passing action. To satisfy obvious strategy-proofness, following a passing action, the mechanism must make certain guarantees to the agent in the continuation game that ensure she will never “regret” her decision to pass. Pycia and Troyan (2023b) show that millipede mechanisms are obviously strategy-proof and any obviously strategy-proof mechanism is equivalent to a millipede mechanism in which all agents follow greedy strategies, where a strategy is called *greedy* if, at any history h at which the agent moves, if the top object that is still possible for her at h is clinchable at h , the agent clinches the object; otherwise, the agent passes.²⁰

The extensive-form game presented in Figure 1 (taken from Pycia and Troyan (2023b)) is an example of a millipede that allocates three objects x, y , and z to three agents, i_1, i_2 , and i_3 . Agent i_1 moves first and can clinch either object x or y (and leave the game), or can pass (and remain in the game). If i_1 passes, agent i_2 can either clinch 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 clinch any object. If i_3 clinches x or y , then the allocation is determined, and if agent i_3 clinches z then i_1 can choose between x and y .

For an illustration of how the resulting millipede mechanism—the game and players’ greedy strategies—guarantee obvious strategy-proofness, consider agent i_1 . For all types of i_1 that rank x or y first, greedy strategies direct i_1 to clinch her top object at the first move, which is clearly obviously strategy-proof. Consider now any type of i_1 that ranks z first. At her first move, z is possible only following a pass, and therefore, any greedy strategy directs the agent to pass.²¹ Notice that at all terminal histories following a pass, agent i_1 receives z (her top object), except if i_3 takes z at her move. In this latter case, following i_3 ’s clinching of z , the mechanism returns to agent i_1 , and offers her the opportunity to clinch either x or y . Thus, for i_1 , the worst-case outcome from passing is at least as good as anything she could have clinched at the initial move, and obvious strategy-proofness is satisfied.

¹⁹At any history, there may be multiple clinging actions that clinch the same object for the agent who moves, but may affect the continuation game. An agent might also clinch an object—in the sense that irrespective of any further actions by any agent or nature the agent receives this object—at a history at which the agent does not move.

²⁰We provide further details on millipede mechanisms in Step 2 of the proof in Appendix B.

²¹An object x is said to be *possible* for agent i at a history h if there is some terminal history $\bar{h} \supseteq h$ at

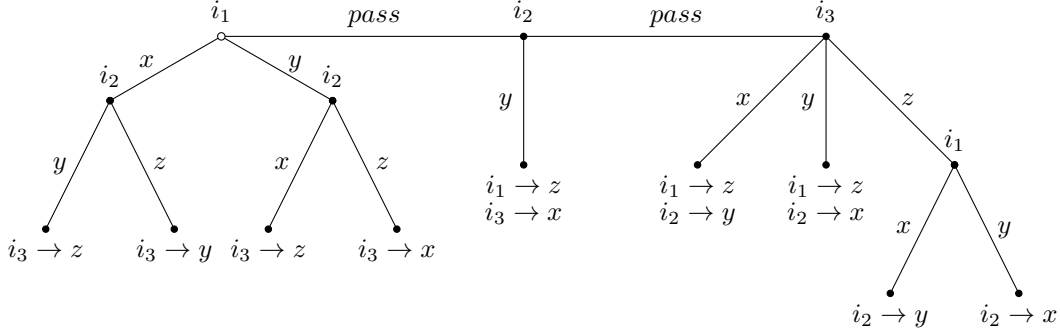


Figure 1: An obviously strategy-proof and Pareto efficient game Γ with three agents and three objects.

The next step of the proof combines the structure of millipede mechanisms with properties of general Pareto efficient and obviously strategy-proof mechanisms identified by Bade and Gonczarowski (2017). Pareto efficiency mean that every object must be possible for the first agent. If this agent clinches, then they are no longer active, and we enter a smaller subgame on the remaining agents and objects. When the first agent, say i , passes, the mechanism is able to infer that none of the objects offered for clinching is her top possible object, but otherwise, the possible rankings of the objects for the agent remain arbitrary. If two or more objects might be the agent's favorite, then again by Pareto efficiency, all unclinched objects are possible for the next agent, say j (else, if we were to exclude an object x from j , it might be that j ranked x first while all other agents, including i , ranked it last, which would violate Pareto efficiency).

With two active agents for whom all objects are possible and whose top objects we do not know, all obvious dominance constraints can be simultaneously honored; with three or more active agents, it is generally no longer possible to do so. The exception to this is when we know which among possible objects is the most preferred for one of the currently active agents. The only way this may happen is if (cumulatively, at previous histories) the mechanism offered this agent for clinching everything possible for her except for this object; if the agent has passed on all objects possible for her except for one, then we can infer that the single object not yet offered is her favorite. Following Bade and Gonczarowski (2017), we refer to agents who have been offered every object possible for them except for one as *lurkers* and the unique object they have not been offered to clinch as a *lurked object*.²² When one of the currently active agents becomes a lurker, this allows the mechanism to

which i receives x .

²²Our formal definition of a lurker given in the appendix is slightly more restrictive than theirs. See Appendices B and C for further details on lurkers and the role they play in obviously strategy-proof and Pareto efficient mechanisms.

introduce an additional active agent. Because for a lurked object we know that it is some currently active agent's favorite, the mechanism can now exclude the newly active agent from ever receiving any lurked object without violating Pareto efficiency while at the same time respecting obvious dominance constraints for all of the active agents.

For example, consider again Figure 1. At the first move, agent i_1 is offered x and y . By greedy strategies, if she passes, the mechanism is able to infer that i_1 's top choice is z . Upon passing, i_1 becomes a lurker for the object z , and the mechanism is able to exclude i_2 from receiving z , in effect making z possible for only i_1 and i_3 . Notice that this can be done while preserving Pareto efficiency: even though the mechanism has no information on i_2 or i_3 's preferences yet, if it turns out that i_3 's top choice is not z , the mechanism can safely give z to agent i_1 at that point without violating Pareto efficiency.

Pareto efficient millipede mechanisms need not be symmetric in general. Returning again to the mechanism whose game form is illustrated in Figure 1, consider the preference profile $\succ_{\mathcal{N}} = (\succ_{i_1}, \succ_{i_2}, \succ_{i_3})$ defined by

$$\begin{aligned} \succ_{i_1}: x, y, z, \\ \succ_{i_2}: x, y, z, \\ \succ_{i_3}: z, y, x. \end{aligned}$$

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 (we give a more formal definition of role assignments in Appendix A). The continuation game Γ_σ is isomorphic to Γ from Figure 1 with the agents permuted. 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^*) .

In the proof, we show that it is sufficient to prove Theorem 1 for symmetrizations. We state this formally as the following:

Proposition 1. *Suppose that, for every deterministic obviously strategy-proof and Pareto*

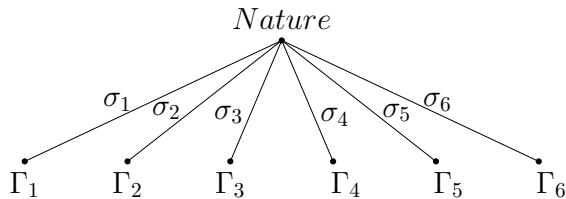


Figure 2: The symmetrization of the mechanism from Figure 1. The mechanism begins with Nature drawing a role assignment function σ_k uniformly at random (there are $3! = 6$ possible permutations of the agents, and thus 6 role assignments). Then, the agents proceed to play the continuation game Γ_k ,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

efficient perfect-information mechanism, its symmetrization is equivalent to Random Priority. Then, every symmetric, obviously strategy-proof and Pareto efficient mechanism is equivalent to Random Priority.

Proposition 1 illustrates a general insight: because symmetrizations do not change symmetric mechanisms, to establish a property for all symmetric mechanisms from a class of mechanisms, it is sufficient to establish it for symmetrizations of all mechanisms from this class.

By Proposition 1, it is sufficient to show that the symmetrization of every obviously strategy-proof and Pareto efficient mechanism is equivalent to Random Priority. In our example, we take the obviously strategy-proof 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).

Letting Σ be the set of all possible role assignment functions, we construct a mapping $f : \Sigma \rightarrow Ord$ 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 $f_\sigma(1), f_\sigma(2), \dots, f_\sigma(N)$ and (ii) f is a bijection.²³ 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

²³ $f_\sigma \in Ord$ is an ordering of all of the agents in \mathcal{N} such that $f_\sigma(j)$ is the j^{th} agent in this ordering.

divided by $N!$. If there exists a bijection as 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.²⁴

At the core of the proof of Theorem 1 is our general construction of the bijection f and the proof that it is indeed a bijection. For sake of illustration, consider again the example with the preference profile $\succ_{\mathcal{N}}$ given above.²⁵ 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 if all moves are clinching moves, $(\Gamma_{\sigma}, S_{\sigma})$ always result in the same allocation as the corresponding serial dictatorship.

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 mapping σ'' into a serial dictatorship with agents ordered i_3, i_1, i_2 , and mapping σ''' to a dictatorship with agents ordered i_3, i_2, i_1 achieves the desired bijection. 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 move in Γ_{σ} , nor do they order agents in the order in which they take their objects.

²⁴The 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 it relies on the properties of Pareto efficient millipede games.

²⁵The 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.

We proceed by dividing any run of the Pareto efficient millipede mechanism (under a specific role assignment and preference profile) into “rounds”; a round consists of a sequence of consecutive moves that determine the assignment of some (endogenously determined) number of lurked objects and one or two unlurked objects. The agents assigned these objects never move after the end of the round, and the continuation mechanism (the continuation game and strategies) after the round ends is again a Pareto efficient millipede for fewer agents and objects. The division into rounds hence has a recursive structure. We order the agents assigned in earlier rounds above those assigned in later rounds, and within a round, agents who receive lurked objects are ordered according to the order in which the object they receive became lurked. This ordering is not necessarily consistent with the order in which agents move or become lurkers, as a lurker might end up with an object different than the one she lurks. A key obstacle in the proof is that the information from the round is sometimes not sufficient to order agents who receive unlurked objects. We deal with this issue by initially having such agents “tie”. We then show that the mapping from role assignments to such partial orderings (with ties allowed) is injective, and that no matter how the ties are broken, the resulting serial dictatorship ends up with all agents receiving the same allocation as in the arbitrary Pareto efficient and obviously strategy-proof mechanism. The bulk of the proof is devoted to showing that it is possible to break these ties—and thus, create a mapping to complete serial dictatorship orderings—in a way that preserves the injectivity property, which establishes that the final mapping from role assignments to orderings of agents is a bijection. The full details of this construction can be found in the appendix.

5 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 of Pycia and Troyan (2023b)—impose no further limitations on Pareto efficient and symmetric implementation in the house allocation environment.

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.
- (2003): “School Choice: A Mechanism Design Approach,” *American Economic Review*, 93, 729–747.
- ASHLAGI, I. AND Y. A. GONCZAROWSKI (2018): “Stable matching mechanisms are not obviously strategy-proof,” *Journal of Economic Theory*, 177, 405–425.
- BADE, S. (2020): “Random Serial Dictatorship: the One and Only,” *Mathematics of Operations Research*, 45, 353–368.
- 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, 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,” .
- MACKENZIE, A. (2020): “A Revelation Principle for Obviously Strategy-proof Implementation,” *Games and Economic Behavior*, 124, 512–533.
- MANDAL, P. AND S. ROY (2022): “Obviously Strategy-Proof Implementation Of Assignment Rules: A New Characterization,” *International Economic Review*, 63, 261–290.
- 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 (2023a): “Strategy-proof, Efficient, and Fair Allocation: Beyond Random Priority,” .
- (2023b): “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.
- (2017): “Incentive Compatible Allocation and Exchange of Discrete Resources,” *Theoretical Economics*, 12, 287–329.
- TROYAN, P. (2019): “Obviously Strategy-Proof Implementation of Top Trading Cycles,” *International Economic Review*, 60.
- 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 $\rho : \mathcal{H} \rightarrow \mathcal{R}$ 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 $\sigma : \mathcal{R} \rightarrow \mathcal{N}$ 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 .²⁶

²⁶While 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 $S_{\sigma, i}(>_i)$ 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 $S_{A, i}(>_i)(h_A) = S_{B, i}(>_i)(h_B)$ for any equivalent histories h_A and h_B in these two games. Note that this element of the construction relies on the fact that the strategies are dominant; that is, they remain optimal regardless of strategies played by other agents.

B Proof of Theorem 1

We break the proof down into 6 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 (2023b). Step 4 constructs a coding algorithm that maps each of the permuted mechanisms $(\Gamma_\sigma, S_\sigma)$ that make up the symmetrization into a partial ordering of the agents that might include ties. Step 5 shows that for any tie-breaking of the partial ordering produced by the coding algorithm, a serial dictatorship run under this agent ordering produces the same allocation as the $(\Gamma_\sigma, S_\sigma)$. Step 6 then shows that it is possible to break ties in such a way that the final mapping is in fact a bijection between permuted mechanisms and serial dictatorship orderings.

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 symmetrization over Pareto efficient deterministic mechanisms. This is so because every symmetric mechanism can be expressed as a lottery over symmetrizations. If each of these symmetrizations is equivalent to Random Priority, then so is the lottery over them. We stated this insight as Proposition 1 above and we prove it now.

Proof of Proposition 1. Take a symmetric, obviously strategy-proof, and Pareto efficient mechanism (Γ, S) . Lemma A.4 of Pycia and Troyan (2023b) shows that for every obviously strategy-proof mechanism, there is an equivalent obviously strategy-proof mechanism with perfect information in which Nature moves at most once, as the first mover.²⁷ 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' after Nature's move in (Γ, S) . The mechanism given by game Γ' , together with the strategy profile induced from Γ , is obviously strategy-proof and Pareto efficient, and hence by the assumption of the proposition it is equivalent to Random Priority. Because every lottery over Random Priority lotteries is still equivalent to Random Priority, the proposition 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.

²⁷Ashlagi and Gonczarowski (2018) briefly mention this result in a footnote while Mackenzie (2020) extends Lemma A.4 to extensive-form games without perfect recall.

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 strategy-proof 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 (2023b), who show that, in a broad class of preference environments that include our setting, any obviously strategy-proof 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 presented with a menu of objects. If the agent i who moves at some h chooses to take object x from the menu, then they receive x , and leave the game (never move again after h); in this case, we say that i has *clinched* x , and call the action taken by i at h a *clinching action*. At h , the agent may also be offered an opportunity to pass. If the agent passes, the object they receive is not yet determined and they might move again, 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 (2023b).

- Possible objects: Recall that 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 . Object x **becomes impossible** for i at h if $i_h = i$ or h is terminal, $i_{h'} = i$ for some $h' \not\supseteq h$, and $x \in P_i(h')$ for all $h' \not\supseteq h$ such that $i_{h'} = i$, but $x \notin P_i(h)$.
- 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 **clinchant** for agent i at history h if i moves at h and 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 clinchant for agent i at h .
- Clinching actions: An action $a \in A(h)$ is called a **clinching action** if agent i_h (the agent 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**.

Remark 1. In the above definitions, note the distinction between a “clinching action” and an object being “clinched”. In particular, it is possible for an agent i to have clinched an object without being called to actively take a clinching action.²⁸ Note also that at any history h at which i moves, for any object $x \in C_i(h)$, there might be multiple actions $a \in A(h)$ that clinch x .

At a terminal history \bar{h} , no agent is called to move and there are no actions. However, at terminal histories \bar{h} , it is notationally convenient to use the notation $i_{\bar{h}}$ to refer to any agent $i \in \mathcal{N}$ and 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, $C_i^{\subseteq}(h) = \{x : x \in C_i(h') \text{ for some } h' \subseteq h \text{ s.t. } i_{h'} = i\}$.
- $C_i^{\not\subseteq}(h)$ is the set of objects that have been previously clinchable for i at some *strict* subhistory of h ; formally, $C_i^{\not\subseteq}(h) = \{x : x \in C_i(h') \text{ for some } h' \subsetneq h \text{ s.t. } i_{h'} = i\}$. If $x \notin C_i^{\not\subseteq}(h)$, then we say x is **previously unclinched** at h .

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

- (M1) Nature either moves once, at the empty history h_{\emptyset} , or Nature has no moves.
- (M2) 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.
- (M3) At all h , terminal and non-terminal, if there exists a previously unclinched object x that becomes impossible for agent i_h at h , then $C_{i_h}^{\not\subseteq}(h) \subseteq C_{i_h}(h)$.

Given a mechanism (Γ, S) and a type \succ_i , a strategy $S_i(\succ_i)$ is 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 object; otherwise, $S_i(\succ_i)(h)$ is the passing action. 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

²⁸This distinction matters in our analysis of efficient and obviously strategy-proof games (efficient millipedes). In these games, clinching without being called to move can happen when an agent i has been offered to clinch and has passed on all possible objects for them except for one, x . This implies that x is i 's favorite object that was ever possible and might allow the mechanism to later assign x to i without calling i to move (depending on the actions of other agents). A specific example of this arises at histories satisfying property 5 of efficient millipedes that we describe in the sequel.

object at h whenever it is clinchable. Condition (M3) of the millipede definition says that when some previously unclinched object becomes impossible for an agent, the next time she moves, she is offered at least a clinching action for each object 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, 2023b). *Every obviously strategy-proof mechanism is equivalent to a millipede mechanism.*

Using the above result from Pycia and Troyan (2023b), it is without loss of generality to assume in Proposition 2 that (Γ, S) is a millipede mechanism. Further, a mechanism (Γ, S) is said to be **pruned** if every history is on the path of play for some type profile (when information set I is on path for type profile $\succ_{\mathcal{N}}$ or type \succ_i , we also say that $\succ_{\mathcal{N}}$ or \succ_i **reaches** I). By Li (2017)’s Pruning Principle, it is without loss of generality to assume that all millipedes (and obviously strategy-proof mechanisms in general) discussed below are pruned. Thus, to prove Proposition 2 we must show that the symmetrization of any Pareto efficient and pruned millipede mechanism is equivalent to Random Priority.

Step 3: Pareto Efficient Millipedes

Obvious strategy-proofness allows us to assume that (Γ, S) is a pruned 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 Pareto efficient obviously strategy-proof mechanisms. Informally, a lurker is an agent who has previously been offered a clinching action for each object that is possible for her except for one, which she is said to “lurk”.

Let (Γ, S) be a pruned 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 .²⁹ 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

²⁹An agent might still be called to play at a history h at which she has clinched an object. At this point, the agent may still influence the continuation game, but can no longer actively influence the object they themselves receive, and so formally, we do not classify them as active. See footnote 38 for the context in which this may occur.

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 .³⁰

We define lurkers recursively as follows. At the empty history no agent is a lurker. Consider a history h and an agent i . If i has moved at a strict subhistory of h then let $h' \not\subseteq h$ 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) $C_i^c(h') = P_i(h') \setminus \{x\}$, and (iv) $x \notin C_j^c(h')$ for any other $j \neq i$ that is active at h' and is not a lurker at h' . Otherwise, i is not a lurker at h . Thus, i is not a lurker for any object x at h if i has not moved at any strict subhistory of h or if i has moved at some strict subhistory but at least one of the conditions (i)-(iv) fails.³¹ 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).³² By conditions (ii) and (iii), each lurker or BG lurker lurks only one object, and Bade and Gonczarowski (2017) show that each BG-lurked object has only one BG lurker. The analogous claim holds for lurked objects; see Lemma 6. BG also show that at any history at most two active agents are not BG lurkers. Our Lemma 12 shows that this also continues to hold for our definition of lurkers.³³

³⁰Note the distinction between guaranteeable 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 (2023b).

³¹An agent might be a lurker at a history h but no longer be a lurker at a continuation history $h'' \not\supseteq h$ at which condition (i) no longer holds because the agent has clinched an object by h'' . When the context is clear, for simplicity we might refer to such agents as lurkers, with the understanding that we are referring to the agents who were lurkers at h . See footnote 38 for the context in which we do so.

³²BG 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: all lurkers in our sense are BG lurkers, but the converse need not hold. In consequence, our definition of a non-lurker is more permissive: a non-lurker in our usage may fail to be a BG non-lurker. We include (iv) in the definition of a lurker to facilitate the construction of our coding algorithm in Step 4 that maps role assignment functions to agent 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.

³³While the formal proofs (provided in Appendix C) require more care, here is a rough intuition why at any history there might be at most two active non-lurkers and why no two lurkers may lurk the same object. Pareto efficiency requires that everything is possible for the first agent to move. If this first mover clinches an object they stop being active. The structure of the millipede games imply that a mover continues to be active only when they pass and a greedy strategy means that we can only establish that one of the objects not yet offered for clinching is the top possible object; the ranking of other objects can then remain arbitrary. If on path following the pass, two or more objects might be the top choice for the passing agent, say i , then Pareto efficiency implies that all not yet clinched objects are possible for the next agent, say j , that moves for the first time (else, if we exclude object a from j we have no guarantee that j wanted a as their top choice while all other agents, including i rank a as their worst object, in which case the exclusion would violate Pareto efficiency). With two active agents for whom all objects are possible and whose top objects we do

At any h , we partition the set of active agents as $\mathcal{A}(h) = \mathcal{L}(h) \cup \bar{\mathcal{L}}(h)$. The set $\mathcal{L}(h) = \{\ell_1^h, \dots, \ell_{\lambda(h)}^h\}$ 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 $\mathcal{X}^{\mathcal{L}}(h) = \{x_1^h, \dots, x_{\lambda(h)}^h\}$ is the set of lurked objects and $\bar{\mathcal{X}}^{\mathcal{L}}(h) = \mathcal{X}(h) \setminus \mathcal{X}^{\mathcal{L}}(h)$ 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 $\ell_{m'}^h$. For BG lurkers, we use the same older and younger comparisons and analogous notation ($\mathcal{L}^{BG}(h)$, $\bar{\mathcal{L}}^{BG}(h)$, $\mathcal{X}^{\mathcal{L},BG}(h)$, $\bar{\mathcal{X}}^{\mathcal{L},BG}(h)$, and $\lambda^{BG}(h)$).³⁴

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 .³⁵ Further, by Lemma 7 in Appendix C, any agent i who moves at a history h immediately following a passing action (that is, $h = (h', a^*)$, where h' is the immediate predecessor of h) is not a lurker at h . When an agent i clinches an object x at some such h then we can then determine the assignments of all lurkers at h (if there are any) 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 set $\mathcal{X}(h) \setminus \{x_1^h, \dots, x_{m_1}^h\}$.
 - 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 $\mathcal{X}(h) \setminus \{x_1^h, \dots, x_{m_2}^h\}$.

not know, we can simultaneously honor all of their obvious dominance constraints. However, with three or more such active agents, it is no longer possible to simultaneously honor all of these constraints. Becoming a lurker roughly means that the mechanism has “learned” that i ’s favorite object that was ever possible for her is the one that she lurks, x . This allows the mechanism to exclude other agents from ever receiving x without violating Pareto efficiency, and is what allows the introduction of a third active agent while still being able to satisfy obvious dominance. If one of these agents becomes a lurker, then a fourth agent can become active, etc. In effect, we might have an arbitrary number of lurkers, but at most two active agents who are not lurkers.

³⁴The ordering of the lurkers by age is strict because an agent becomes a lurker immediately after taking a passing action at a history by which she has (cumulatively) been offered a clinching action for each object possible for her except for one.

³⁵Lemma 8 in Appendix C shows that once an agent becomes a lurker, she remains a lurker at any continuation history following a sequence of passing actions: that is, if $h' = (h, a^*, \dots, a^*)$, then $\mathcal{L}(h) \subseteq \mathcal{L}(h')$.

- This process is repeated until some lurker ℓ_m^h receives an unlurked object, at which point all remaining unassigned lurkers are assigned their lurked objects.

These assignments follow from Lemma 16 in Appendix C (which in turn is derived from Lemma E.17 in BG, which shows these assignments hold for BG lurkers). Notice that in the above 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 at h , or one of the lurkers later in the chain. Because there can be at most two non-lurkers at any h , one of whom was the agent i who moves at h , there might be at most one additional active agent, j' who is not part of the above assignments. If such a j' exists and $y \in C_{j'}^{\subseteq}(h)$ then j' receives her favorite object among objects that are possible for her after the above assignments are carried out.³⁶ (If $y \notin C_{j'}^{\subseteq}(h)$, then the assignment of j' is not yet settled, and will be determined later in the mechanism.)

As we formalize in Lemma 2 below, Step 2 and the above structure of assignments allow us to assume that our base game Γ is a millipede game that has the following properties:³⁷

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 $h' = \overbrace{(h, a^*, \dots, a^*)}^{n \text{ times}}$ 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. At any non-terminal history h , if there exists one or more agents such that $P_i(h) = G_i(h)$, then one of these agents moves at h .
4. If i moves at h and $P_i(h) = G_i(h)$, then $C_i(h) = P_i(h)$, there is no passing action at h , and i is not called to move at any $h' \not\preceq h$.

³⁶This follows by property (M3) of millipede games. Intuitively, $C_{j'}^{\subseteq}(h) = C_{j'}^{\subseteq}(h)$ because j' (being different from i) does not move at h . Hence, $y \in C_{j'}^{\subseteq}(h)$ means that j' passed on y at some $h' \not\preceq h$. By greedy strategies, passing implies that y is not the agent's top possible choice, but y might still be the second-best object for j' . Once y is taken by agent j , the mechanism must guarantee j' gets her top choice, else it would not be obviously dominant for her to pass on y at h' . For a formal proof, see Lemma 17 in Appendix C.

³⁷Property 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 (2023b). 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 obviously strategy-proof games.

5. At any history (h, a^*) that ends with a passing action, the agent who moves is not a lurker. At any history (h, a_x) that ends with a clinching action (where x is the object clinched by action a_x):

(a) If x is a lurked object at h , all lurkers whose lurked object is strictly older than x have clinched their lurked object at (h, a_x) . Such older lurkers may or may not be called to move following (h, a_x) .³⁸ If any such lurkers are called to move, they do so in order from oldest to youngest lurker, and we let h' denote the history that ends with the clinching action of the youngest of these lurkers; otherwise, we let $h' = (h, a_x)$. None of the lurkers who have clinched their object at h' is called to move at or following h' . At h' , the agent who lurked x at h is called to move. This agent is offered for clinching each object that has not been clinched by h' and there is no passing action.

(b) If x is not a lurked object at h , then all lurkers at h have clinched their lurked object at (h, a_x) . The lurkers at h may or may not be called to move following (h, a_x) .³⁹ If any such lurkers are called to move, they do so in order from oldest to youngest lurker, and we let h' denote the history that ends with the clinching action of the youngest of these lurkers; otherwise, we let $h' = (h, a_x)$. None of the lurkers at h is called to move at or following h' .

If there exists an agent j' who was active at h and has not yet clinched an object at h' , then j' moves at h' and:⁴⁰

(i) If $x \in C_{j'}^{\text{cl}}(h)$, then j' is offered a clinching action for each object that remains possible for her.⁴¹ There is no passing action.

(ii) If $x \notin C_{j'}^{\text{cl}}(h)$, then j' is offered at least a clinching action for each object in $C_{j'}^{\text{cl}}(h)$; she may also have other clinching actions and/or a passing action.⁴²

³⁸The moment x is clinched at (h, a_x) all of these older lurkers have clinched their lurked objects without being called to move (see Remark 1 on the distinction between clinching an object and a clinching action). These agents are thus no longer “active” (in the sense that they can no longer actively influence their own object; cf. footnote 29), but might still be called to move once more and can influence the continuation game. These agents are also no longer lurkers, but with some abuse of terminology, we continue to refer to them as lurkers when the context is clear that we are referring to agents who were lurkers at h (cf. footnote 31).

³⁹See footnote 38.

⁴⁰This agent j' is not a lurker because of the first sentence of 5(b). Furthermore, there can be at most one such agent j' . This follows as Lemma 12 tells us that there are at most two active non-lurkers at any history following a sequence of passes, and the above structure implies that once a clinching action is taken by some non-lurker $i \neq j'$ at such a history, only lurkers are called to move (and thus no new agents become active) until the point at which all active lurkers have clinched an object.

⁴¹See footnote 36.

⁴²By definition, j' is not a lurker, and as we show in Appendix C, her set of previously clinchable objects

If no such j' exists, then all agents who were active at h have clinched an object by h' . If there remain agents who have not yet clinched an object, then one of these agents moves at h' , and a continuation game begins among the remaining unmatched agents and objects. 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 obviously strategy-proof and Pareto efficient mechanism (Γ, S) is equivalent to a millipede mechanism satisfying properties 1-5.*

Remark 2 (Recursive structure). Any Pareto efficient millipede game is equivalent to one with a recursive structure that proceeds in $k = 1, \dots, K$ rounds, with each round divided into two phases. We call these phases a *passing phase* and a *clinging phase*.

Let h_{\emptyset}^k denote the **initiating history** of round k . For $k = 1$, $h_{\emptyset}^k = h_{\emptyset}$, the root of the game; for $k > 1$, h_{\emptyset}^k will be defined recursively, as the conclusion of the previous round.

Passing phase:

- Starting from h_{\emptyset}^k , agents take some number (possibly zero) of passing actions. At any $h' = (h_{\emptyset}^k, a^*, \dots, a^*)$ in the passing phase:
 - The agent called to move is not a lurker at h' . (Lemma 7)
 - There is at most one BG lurker who is not a lurker at h' , and if such an agent exists, there is no passing action at h' .⁴³ (Lemma 11)
 - There are at most 2 active non-lurkers at h' . (Lemma 12)
 - If a lurked object is clinchable at h' , then there is no passing action at h' (that is, h' is a terminating history; see footnote 43). (Lemma 14)

$C_{j'}^{\bar{c}}(h)$ cannot contain any lurked objects (note that $C_{j'}^{\bar{c}}(h) = C_{j'}^c(h)$ as j' does not move at h). Since in this case we assume that $x \notin C_{j'}^{\bar{c}}(h)$, all of the objects in $C_{j'}^{\bar{c}}(h)$ remain unassigned, and thus may be offered to j' . See Lemma 18 in Appendix C for details.

⁴³Following BG, we call such h' the *terminating history* of the round; formally, the terminating history of round k is the maximal superhistory of the form $h' = (h_{\emptyset}^k, a^*, \dots, a^*)$. Note that the terminating history is not a terminal history, nor is it necessarily the conclusion of the current round (which will be defined recursively below); rather it is the “termination” of the passing path. Depending on the preferences, the path through the mechanism may or may not reach the terminating history of a round. See Appendix C for details.

- At some history $h^k = (h_{\emptyset}^k, a^*, \dots, a^*)$, an agent $i \notin \mathcal{L}(h^k)$ takes a clinching action $a_x \in A(h^k)$ (it is possible that $h^k = h_{\emptyset}^k$, in which case the first agent called to play in the round clinches an object, and there are no passing actions). The history (h^k, a_x) begins the clinching phase.

Clinching phase:

- In the clinching phase, following the clinching of a lurked object x , all lurkers at h^k who lurk objects strictly older than x clinch their lurked object (see footnote 38). These agents may or may not be called to move, and if any of them are called to move, they move in order from oldest to youngest lurker, have only clinching actions, and never move again. Following the last clinching action of these agents, the next agent to move is the agent who lurked x at h^k (notice that some other agent is the one who clinches x), and she is given at least a clinching action for each lurked object strictly younger than x and each unlurked object. There is no passing action. (Lemma 16)
- The game continues in this manner: each time a lurked object x' is clinched, all lurkers who lurk objects strictly older than x' clinch their lurked object with or without being called to move. After these agents (if any) are called to move, the next agent called to move is the agent who lurked x' , and she is given at least a clinching action for each lurked object strictly younger than x' and each unlurked object.
- At some point in this sequence of moves, some agent—either an agent who was a lurker at h^k , or possibly i , the non-lurker who moved at h^k to initiate the clinching phase—clinches an unlurked object $\hat{x} \in \bar{\mathcal{X}}^{\mathcal{L}}(h^k)$. All remaining lurkers clinch their lurked object with or without being called to move. If any of these lurkers are called to move, let $\overset{\circ}{h}$ be the history that ends with the action of the last of these agents; otherwise, let $\overset{\circ}{h}$ be the history that ends with the clinching of \hat{x} . (Lemma 16)
- At $\overset{\circ}{h}$, there might be at most one remaining active agent j who has not yet clinched an object; if such a j exists, they were active at h^k , but did not move at h^k nor were they a lurker h^k .⁴⁴ If such a j exists and $\hat{x} \in C_j^{\subseteq}(h^k)$, then agent j is called to move at $\overset{\circ}{h}$ and is offered at least a clinching action for each object that is still possible for her. There is no passing action. (Lemma 17)

The history $\overset{\circ}{h}$ is the beginning of round $k + 1$ (and thus the end round k), except if there is an agent j as described in the last bullet point, and $\hat{x} \in C_j^{\subseteq}(h^k)$; in this latter case, round

⁴⁴Such agent j also does not move at any h' such that $h^k \subseteq h' \not\subseteq \overset{\circ}{h}$.

$k + 1$ begins following the clinching action of this j . Thus, in the former case the initiating history for round $k + 1$ is $h_{\emptyset}^{k+1} = \overset{\circ}{h}$, and in the latter case it is $h_{\emptyset}^{k+1} = (\overset{\circ}{h}, a_j)$, where a_j is the clinching action of agent j at $\overset{\circ}{h}$.

If all agents active at h^k have clinched an object by h_{\emptyset}^{k+1} , then some previously inactive agent i' moves at h_{\emptyset}^{k+1} (unless all agents have clinched an object, in which case the game ends). If there remains an agent j who was active at h^k and has not clinched an object at h_{\emptyset}^{k+1} —which can only occur when there exists an active non-lurker j at h^k who does not move at h^k and $\hat{x} \notin C_j^{\subseteq}(h^k)$ —then Lemma 18 implies that j moves at h_{\emptyset}^{k+1} to begin the next round, and is offered for clinching some set of objects $C_j(h_{\emptyset}^{k+1}) \supseteq C_j^{\subseteq}(h_{\emptyset}^{k+1})$.⁴⁵ At h_{\emptyset}^{k+1} , agent j may be offered to clinch additional objects beyond those in $C_j^{\subseteq}(h_{\emptyset}^{k+1})$, and may or may not have a passing action.

In either case, the continuation game starting at h_{\emptyset}^{k+1} is another Pareto efficient and pruned millipede game on the remaining unmatched agents and objects.⁴⁶ This continuation game starting from h_{\emptyset}^{k+1} also satisfies properties 1-5, and so continues to have the same passing-and-clinching phase structure.

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 (for a fixed preference profile) 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 divided by $N!$. Thus, if we can find a bijection $f : \Sigma \rightarrow Ord$ 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 $f_{\sigma}(j)$

⁴⁵Notice that j does not move between h^k and h_{\emptyset}^{k+1} , so $C_j^{\subseteq}(h_{\emptyset}^{k+1}) = C_j^{\subseteq}(h^k)$.

⁴⁶In particular, when restricted to the objects that are not yet clinched at h_{\emptyset}^{k+1} , all possible ranking types are present at h_{\emptyset}^{k+1} for all remaining agents. For those agents who have never been called to move, this is obvious, as they have not yet revealed any preference information. For the agent j who might be carried over from the previous round, it follows because each type that ranks \hat{x} first (over the set of all objects \mathcal{X}) reaches h_{\emptyset}^{k+1} , and \hat{x} is no longer available at h_{\emptyset}^{k+1} ; cf. Lemma 18.

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 $f : \Sigma \rightarrow Ord$.

The intuitive idea behind constructing \succ is as follows. Fixing a preference profile and beginning with round 1 of the mechanism, we find the first agent to clinch some object x after a (possibly empty) series of passes at some history h^1 . This initiates the clinching phase of round 1, in which all but at most one of the agents who are active at h^1 clinch their objects, as described in Step 3 of the proof. 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 \hat{x} ; if \hat{x} was previously offered to the remaining active agent (if such an agent exists), then we add both remaining agents to the order without distinguishing between them, i.e., these two agents tie; if \hat{x} was not previously offered to the other remaining active agent (or no such agent exists), then we only add to the ordering the agent who clinched \hat{x} . 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 round, this agent is still active with the carried over “endowment” of previously clinchable objects, $C_j^\sigma(h)$ (see Remark 2). After clearing this first segment of agents, we repeat this procedure starting from the initiating history of the next round.

To illustrate, consider Figure 3, which reproduces 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 and the given preference profile (which is restated in the caption of the figure) is shown by the dashed lines. 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),

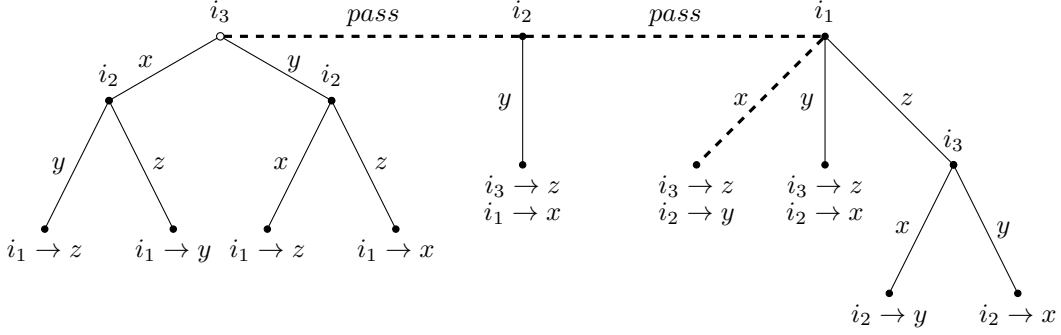


Figure 3: The example from Section 4 under the role assignment $\sigma''(r_1) = i_3$, $\sigma''(r_2) = i_2$, and $\sigma''(r_3) = i_1$ and preferences $\succ_{i_1}: x, y, z$, $\succ_{i_2}: x, y, z$, $\succ_{i_3}: z, y, x$. The dashed lines show the path of play.

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. In this case, both z and x are lurked (in this order), and so the corresponding serial dictatorship is $f_{\sigma''}(1) = i_3$, $f_{\sigma''}(2) = i_1$, and $f_{\sigma''}(3) = i_2$.

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

Coding Algorithm. Consider a permuted mechanism $(\Gamma_\sigma, S_\sigma)$, and fix a preference profile $\succ_{\mathcal{N}}$. Consider the game path from the root node h_\emptyset to a terminal node \bar{h} when agents follow the strategy profile $S_\sigma(\succ_{\mathcal{N}})$. 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 Let $h_\emptyset^1 = h_\emptyset$ and $h^1 = (h_\emptyset^1, a^*, \dots, a^*)$ be the history that initiates the first clinching phase of the game (it is possible that $h^1 = h_\emptyset^1$, when the first agent to move takes a clinching action). Let x_1 be the object clinched by agent i_1 at h^1 .⁴⁷ Let $\mathcal{L}(h^1) = \{\ell_1, \dots, \ell_{\lambda(h^1)}\}$ be the set of lurkers, and $\mathcal{X}^{\mathcal{L}}(h^1) = \{x_1, \dots, x_{\lambda(h^1)}\}$ 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} .⁴⁸

⁴⁷That 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 7, $i^1 \notin \mathcal{L}(h^1)$. 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 become lurked. In the notation for lurkers $\ell_k^{h^1}$ and lurked objects $x_k^{h^1}$ we suppress the history superscript.

⁴⁸Note that i_{x_k} is not necessarily the agent who lurks x_k at h^1 .

2. Let $j \in \mathcal{L}(h^1) \cup \{i^1\}$ be the unique agent that is not one of the agents $i_{x_1}, \dots, i_{x_{\lambda(h^1)}}$ 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 $y \in C_{j'}^{\subseteq}(h^1)$, then define $\tilde{\succ}^1$ as:

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

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

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

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

Step k Let h_{\emptyset}^k be the initiating history that begins round k of the mechanism (as described in Remark 2), and let $h^k = (h_{\emptyset}^k, a^*, \dots, a^*)$ be the history that initiates the round k clinching phase (as in step 1, it is possible that $h^k = h_{\emptyset}^k$). Let x^k be the object clinched by i^k at history h^k . Let $\mathcal{L}(h^k) = \{\ell_1, \dots, \ell_{\lambda(h^k)}\}$ be the set of lurkers, and $\mathcal{X}^{\mathcal{L}}(h^k) = \{x_1, \dots, x_{\lambda(h^k)}\}$ 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 $(\tilde{\succ}^1, \dots, \tilde{\succ}^K)$, 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 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 $i \not\succeq j$ and $j \not\succeq i$, 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 3. Different preference profiles $\succ_{\mathcal{N}}$ may lead to different codings. For a given $\succ_{\mathcal{N}}$, the coding algorithm divides the game path from the root to the terminal node into a series of K steps. Each step of the coding algorithm corresponds precisely to a round of the mechanism (as described in Remark 2), and the agents coded in step k are precisely those agents who clinched an object during the round k clinching phase of the mechanism.⁴⁹ In

⁴⁹At the end of coding step k , there may be one agent, say j' , who was active during round k of the mechanism, and was not coded in step k . When this occurs, at the initiating history for round $k+1$, h_{\emptyset}^{k+1} ,

step $k + 1$ of the coding algorithm, we start at the initiating history h_{\emptyset}^{k+1} , and find the history $h^{k+1} = (h_{\emptyset}^{k+1}, a^*, \dots, a^*)$ that corresponds to the first clinching action taken by an agent in round $k + 1$ of the mechanism. The step $k + 1$ coding is the equivalent to the initial coding on the smaller Pareto efficient millipede continuation game that begins at history h_{\emptyset}^{k+1} . This recursive structure is further discussed in Remark 2.

Each role assignment function σ induces a permuted mechanism $(\Gamma_{\sigma}, S_{\sigma})$, and each permuted mechanism has an associated coding \succ_{σ} 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_{σ} . We say that a total ordering of the agents f_{σ} is **consistent** with \succ_{σ} 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_{σ} , total order f_{σ} is consistent if there is some possible way to break the ties in \succ_{σ} that delivers f_{σ} . We further say that f_{σ} is **consistent with \succ_{σ} 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_{σ} 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 $\succ_{\sigma_A} = \succ_{\sigma_B}$, 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_{σ} , any way of breaking the ties (if any ties exist) between agents produces a total order f_{σ} that is consistent with \succ_{σ} . 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} .

agent j' is called to move, and is offered the to clinch at least each object that she has been offered to clinch previously in the game (and might have other moves), as explained in Remark 2.

Step 6: Bijectivity

Finally, we show that it is possible to break the ties in the codings produced by the coding algorithm, $(\succ_\sigma)_{\sigma \in \Sigma}$, in such a way as to produce a mapping $f : \Sigma \rightarrow \text{Ord}$ 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, $h_A^1 = (h_\emptyset, a^*, \dots, a^*)$ 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 $k' = 1, \dots, k$ 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 is injective. Each coding is a partial ordering and what we are ultimately after is a mapping from role assignments to strict total orderings of the agents. We generate these strict orderings by breaking ties in partial orderings. In effect, what remains to be shown is that it is possible to break the ties in all codings in a way that maps each partial order to a unique strict order. Lemma 5 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, \dots, 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, \dots, 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).

Before stating the next lemma, to understand the issue better, we discuss a small example.

Say that there exists some role assignment function such that the coding to partial orders is $\{i, k_1\} \succ_1 \dots$. To create a strict ordering of the agents, we must (somehow) break the tie between i and k_1 . By Lemma 3, either way of breaking this tie will produce a serial dictatorship that gives the same allocations as Γ_1 , and so from this perspective, it does not matter how the tie is broken. While Lemma 4 shows that the mapping of role assignments to partial orders is injective, we also need to ensure that after any tie-breaking, the mapping to strict total orders continues to be injective, which means that no other coding is mapped to the same strict total order after the tiebreak. In particular, there might be another coding $k_1 \succ_2 i \succ_2 \dots$. To preserve injectivity, we then break the tie in \succ_1 so that $i \succ_1 k_1 \succ_1 \dots$ (because if we broke the tie such that $k_1 \succ_1 i \succ_1 \dots$, both \succ_1 and \succ_2 could map to the same strict total order, violating injectivity). Can we do so without a conflict with some other coding \succ' that also begins with $i \succ' k_1 \succ' \dots$? Lemma 5 below is at the core of our argument that we can. Indeed, if there is some \succ_3 such that $i \succ_3 k_1 \dots$, then, Lemma 5 implies that k_1 must tie with some other agent k'_2 . Therefore, in the coding \succ_3 , we can break the tie so that $i \succ_3 k'_2 \succ_3 k_1 \succ_3 \dots$, and in so doing, ensure that there are no injectivity conflicts with the tiebreak of \succ_1 . We must also make sure that after breaking the tie in \succ_3 , there is also no conflict with any other \succ_4 such that $i \succ_4 k'_2 \succ_4 k_1 \succ_4 \dots$, and so forth. Lemma 5 allows us to do so.

To formulate the lemma we introduce the notion an arm. Given a coding \succ_1 in which two agents i and k_1 tie, an **arm** of length $n + 1 \geq 1$ is a sequence of $n + 1$ role assignments $\alpha = (\sigma_1, \sigma_2, \dots, \sigma_n, \sigma_{n+1})$ such that the resulting codings have the following structure:

$$\begin{aligned}
\text{Arm } \alpha: \quad & j_1 \dots j_P \succ_1 \{i, k_1\} \succ_1 \dots \\
& j_1 \dots j_P \succ_2 k_1 \succ_2 \{i, k_2\} \succ_2 \dots \\
& j_1 \dots j_P \succ_3 k_1 \succ_3 k_2 \succ_3 \{i, k_3\} \succ_3 \dots \\
& \vdots \\
& j_1 \dots j_P \succ_n k_1 \succ_n k_2 \succ_n k_3 \succ_n \dots \succ_n k_{n-1} \succ_n \{i, k_n\} \succ_n \dots \\
& j_1 \dots j_P \succ_{n+1} k_1 \succ_{n+1} k_2 \succ_{n+1} k_3 \succ_{n+1} \dots \succ_{n+1} k_{n-1} \succ_{n+1} k_n \succ_{n+1} i \dots
\end{aligned}$$

If $n = 0$ then the arm α consists of just one coding $j_1 \dots j_P \succ_1 \{i, k_1\} \succ_1 \dots$. Note that an arm α satisfies the following properties: (i) the partial orderings of all codings coincide on all agents ranked above k_1 (represented by $j_1 \dots j_P$, and we allow that there may be ties among some of the j_p 's); (ii) the first coding in the arm ranks the tie $\{i, k_1\}$ strictly below $j_1 \dots j_P$; (iii) there is a set of agents k_2, \dots, k_n such that the n^{th} element of the arm ranks as shown above. In particular, in coding \succ_n , agent i ties with k_n , while in \succ_{n+1} , agent i may or may

not tie with some other agent.

If there exists a coding \succ_1 with a tie $\{i, k_1\}$ and an arm α of the above form in which i does not tie with another agent under \succ_{n+1} , then, in order to preserve injectivity, we break the tie in \succ_1 such that $i \succ_1 k_1$; in this case, we say that α **breaks the tie $\{i, k_1\}$ in \succ_1 in favor of i** .

Lemma 5. *Suppose that for positive integers $n, m \geq 0$ there are two arms $\alpha = \{\sigma_1, \sigma_2, \dots, \sigma_n, \sigma_{n+1}\}$ and $\alpha' = \{\sigma'_1, \sigma'_2, \sigma'_3, \dots, \sigma'_m, \sigma'_{m+1}\}$ such that $\sigma_1 = \sigma'_1$ and:*

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

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

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} .

If either n or m is 0, the lemma is a tautology. The proof shows that it continues to hold for arbitrary $n, m \geq 1$.

We call such α and α' as defined in the lemma **dual 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 which i and k_1 tie.
- In arm α , every subsequent coding ranks k_1 strictly ahead of all other agents (besides

the j_p 's), while in α' , every subsequent coding ranks i ahead of all other agents (besides the j_p 's).

- Within arm α , the only difference from ℓ to $\ell + 1$ is that the agent k_ℓ who tied with i in \succ_ℓ 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 α' .
- Across the two arms, it is possible that some or all of the agents k_2, \dots, k_n are the same as the agents k'_2, \dots, k'_m , though it is not necessarily assumed.
- We do not require $m = n$.
- In arm α , any agent $\ell \in \{j_1, \dots, j_P, i, k_1\}$ receives the same object for all role assignments $\sigma_1, \dots, \sigma_{n+1}$ of the arm. Indeed, there is an ordering of agents j_1, \dots, j_P, k_1, i that is consistent with both codings \succ_1 and \succ_2 and hence Lemma 3 implies that each $\ell \in \{j_1, \dots, j_P, i, k_1\}$ receives the same objects under σ_1 and σ_2 . Similarly, there is an ordering $j_1, \dots, j_P, k_1, k_2, i$ that is consistent with both \succ_2 and \succ_3 , and so by Lemma 3 again, each $\ell \in \{j_1, \dots, j_P, i, k_1\}$ receives the same object under σ_2 and σ_3 (and hence, this is also the same as the object received under σ_1). Continuing in this way, we conclude that agents all j_1, \dots, j_P, i, k_1 receive the same object under all role assignments in the arm. A similar argument shows that for $n' \in \{2, \dots, n\}$, agent $k_{n'}$ receives the same object for all role assignments $\sigma_{n'}, \sigma_{n'+1}, \dots, \sigma_{n+1}$.⁵⁰ An analogous observation holds for arm α' . Notice also that both arm α and α' begin with the same role assignment, and so this further implies that the agents j_1, \dots, j_P, i, k_1 receive the same objects for all role assignments in both arms.

The proof of Lemma 5 can be found in Appendix C.2. The upshot of the lemma is that, given some coding \succ_1 with a tie $\{i, k_1\}$ if there exists an arm α that breaks the tie in favor of i , then, in the final coding of any dual arm α' , agent k_1 must tie, and thus breaking the tie in \succ_1 in favor of i will not lead to a conflict with another coding that will break injectivity.

More formally, let Ψ be the initial mapping from role assignments σ to partial orderings \succ_σ generated by the coding algorithm from Step 4 of the proof. By Lemma 4, Ψ is injective. We use Lemma 5 to break ties to create from each \succ_σ a consistent total order f_σ in a way that preserves the injectivity by iteratively executing the following tie-breaking step. We break ties one at a time, and after breaking a tie, we update the mapping Ψ so that $\Psi(\sigma)$ is the previous partial ordering except for the now-broken former tie. Importantly, the above

⁵⁰Notice that this does not apply for role assignments $\sigma_{n''}$ where $n'' < n'$, because in such role assignments $k_{n''}$ is ordered below i .

concept of an arm is applicable to any collection of partial orderings, and we show below that injectivity and the claim of Lemma 5 continue to hold after each tie is broken.

Tie-Breaking Step. Let $P \in \{1, \dots\}$ be the smallest integer such that there is some role assignment σ and agents $j_1, \dots, j_{P-1}, i, k_1$ such that the initial segment of the partial ordering $\Psi(\sigma)$ is $j_1 \succ_\sigma \dots \succ_\sigma j_{P-1} \succ_\sigma \{i, k_1\}$.⁵¹ If such an integer P does not exist, then all ties are already broken and the tie-breaking procedure terminates. There are two cases:

(a) If there exists such σ for which there is an arm α breaking the tie $\{i, k_1\}$ in position P in favor of one of the agents, say i , then we choose one such σ and break this tie accordingly. By Lemma 5, this is well-defined, in the sense that if there is an arm that breaks the tie in favor of i , then there is no arm α' that breaks the tie in favor of k_1 .

(b) Else, we choose an arbitrary σ with a tie in position P and we break it arbitrarily.

After executing a single tie-break by either step (a) or (b) above, we update the mapping Ψ so that it is the same as before, except with the now-broken tie in $\Psi(\sigma)$; without loss of generality, we denote this broken tie in favor of i , i.e., $i \succ_\sigma k_1$.

For the updated Ψ and the resulting collection of partial orderings, we show the following:

(i) Ψ continues to be injective and (ii) the conclusion of Lemma 5 continues to hold.

For (i), towards a contradiction, assume that injectivity fails after we broke the tie at role assignment σ . Since injectivity held before the tie-break, and the only tie broken was in σ , we must have that the updated $\Psi(\sigma)$ is the same partial ordering as $\Psi(\sigma')$ for some $\sigma' \neq \sigma$. Thus before the tie-break, $\Psi(\sigma')$ must have been a partial ordering that starts with $j_1 \succ_{\sigma'} \dots \succ_{\sigma'} j_{P-1} \succ_{\sigma'} i \succ_{\sigma'} k_1 \succ_{\sigma'} \dots$. This implies that before the tie-break, $\alpha' = (\sigma, \sigma')$ is an arm that breaks the tie in σ in favor of k_1 , which contradicts Lemma 5.⁵²

For (ii), suppose that after the tie-break in σ , there are now two dual arms α and α' that each begin with $j'_1, \dots, j'_{P'-1} \succ_{\sigma_1} \{i', k'_1\}$ for some role assignment σ_1 and that have the structure as in the statement of Lemma 5. As in the lemma, let the length of arm α be $n + 1$, and the length of arm α' be $m + 1$. Further suppose that, in violation of the claim of Lemma 5, partial ordering \succ_{n+1} in arm α ranks i' strictly, and partial ordering \succ'_{m+1} in arm α' ranks k_1 strictly; note in particular that this implies that $n, m \geq 1$.

By the order in which we break the ties, all ties before P have been already broken, and hence $P' \geq P$. Because the claim of the lemma was true before we broke the tie $\{i, k_1\}$ at role assignment σ , it must then be that after the tie-break, $j_1 \succ_\sigma \dots \succ_\sigma j_{P-1} \succ_\sigma i \succ_\sigma k_1 \succ_\sigma$

⁵¹Note that here we index the j agents from $1, \dots, P-1$ (rather than P for most of the rest of the proof) so that we can refer to a tie in the P^{th} (rather than $(P+1)^{\text{th}}$) position. Of course, this indexing is arbitrary.

⁵²For further detail, notice that if the tie-break in σ was broken via (a) above, then we have two dual arms, α and α' , one of which breaks the tie in favor of i and the other that breaks the tie in favor of k_1 , which is a contradiction to Lemma 5. If the tie was broken via (b) above, then, there must not have been any tie for which there was an arm that broke the tie in favor of one of the agents, which again is a contradiction.

is part of at least one of these arms; without loss of generality, assume that this is arm α , and so $\succ_{\ell} = \succ_{\sigma}$ for some $\ell \leq n + 1$ in arm α (an analogous argument applies to the case that the relevant arm is α'). This implies that $j'_1 = j_1, \dots, j'_{P-1} = j_{P-1}$. Further, we know that i and k_1 are ranked strictly in positions P and $P + 1$, which implies $j'_P = i$, $j'_{P+1} = k_1$, and $P' \geq P + 2$. Next, notice that before the tie-break, there cannot be any other role assignment $\sigma'' \neq \sigma$ whose corresponding partial ordering begins with $j_1 \succ_{\sigma''} \dots \succ_{\sigma''} j_{P-1} \succ_{\sigma''} i \succ_{\sigma''} k_1 \succ_{\sigma''}$.⁵³ Because the only tie that was broken was in the partial ordering for role assignment σ , the same holds true after the tie-break as well. This implies that the arm α must be a trivial arm ($n = 0$), because if $n \geq 1$, then, by the structure of an arm and the fact that $P' \geq P + 2$, there are at least two partial orderings that begin by ranking $j_1, \dots, j_{P-1}, i, k_1$ strictly and in this order, in contradiction with the previous sentence. However, $n = 0$ contradicts the previously reached conclusion that $n \geq 1$. This contradiction shows that Lemma 5 continues to hold for the updated mapping Ψ .

We continue executing the above tie-breaking step until we reach a mapping Ψ from role assignments to complete orderings (with no ties remaining). Following each individual tie-break, we can repeat the argument just made to conclude that injectivity is preserved, and thus the final Ψ is injective. 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.

Recap

To recap, we have shown the following:

1. Every Pareto efficient, obviously strategy-proof 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).

⁵³This follows via a similar argument in the proof of claim (i) in the paragraph above: if such a σ'' existed, then, before the tie-break, there was an arm $\alpha'' = (\sigma, \sigma'')$ that breaks the tie in σ in favor of k_1 , and we reach a contradiction via the same argument as in footnote 52.

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 obviously strategy-proof, Pareto efficient and deterministic perfect-information mechanism (Γ, S) is equivalent to Random Priority, by Lemma 1 every symmetric, obviously strategy-proof, and Pareto efficient mechanism is equivalent to Random Priority.

C Proofs of Key Lemmas 2-5

C.1 Proof of Lemma 2

Properties 1-4 follow from the millipede theorem of Pycia and Troyan (2023b), as explained in footnote 37. Thus, we focus on establishing property 5.

Given a subset of objects $X' \subseteq \mathcal{X}$ and a preference ranking for agent i , $>_i$, let $Top(>_i, X')$ be the highest $>_i$ -ranked object in the set X' . Following Bade and Gonczarowski (2017) (hereafter BG), given some history h , we call the maximal superhistory of the form $h' = (h, a^*, \dots, a^*)$ a **terminating history** and the agent who moves at h' a **terminator**; when it is clear what the terminating history is, we denote this agent by t . The terminating history provides an upper bound on the number of passes that can be taken in a row: at the terminating history, the terminator has only clinching actions. Note that the definition of the terminating history is only a function of the game form Γ , and is independent of the players' strategies; it also does not invoke the concept of a lurker.

Our proof of Lemma 2 builds on Lemmas 6-18 below. These lemmas are stated relative to the initial passing path of the game to the terminating history of this path; that is, for all h in these lemmas, we assume that $h = (h_\emptyset, a^*, \dots, a^*)$ for some (possibly zero) number of passing actions a^* .⁵⁴ This will be sufficient because Pareto efficient millipede games will be equivalent to a game that has a recursive structure that proceeds in “rounds”.

The results below combine the structure of millipede games with the properties of general Pareto efficient obviously strategy-proof games identified by BG. For completeness, we provide proofs of all lemmas. Some of these lemmas follow immediately from BG, in which case the proof is just a reference to the corresponding BG lemma. Some others have a BG “analogue” or can be inferred from BG, but require combining different parts of their analysis

⁵⁴Recall that there is at most one passing action at any history, so this is uniquely defined. If the first agent i takes a clinching action at h_\emptyset , and clinches object x , then at the history $h = (h_\emptyset, a_x)$, no other agent has yet moved, and so h is the beginning of a smaller Pareto efficient millipede game on the agents $\mathcal{N} \setminus \{i\}$ and objects $\mathcal{X} \setminus \{x\}$, and we just make the same arguments starting from h . Notice also that if there are lurkers at a history h , the path from h_\emptyset to h must contain a passing action.

with additional arguments to apply to our millipede structure and lurker definition. When there is a BG analogue, we note it in the statement of the lemma. Any proofs not found here can be found in the Supplementary Appendix.

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. By definition, at any h , any lurker is also a BG lurker: $\mathcal{L}(h) \subseteq \mathcal{L}^{BG}(h)$.

Lemma 6. (BG Lemma E.16) *Let $\mathcal{L}(h) = \{\ell_1^h, \dots, \ell_{\lambda(h)}^h\}$ be the set of lurkers at $h = (h_\emptyset, a^*, \dots, a^*)$ and $\mathcal{X}^{\mathcal{L}}(h) = \{x_1^h, \dots, x_{\lambda(h)}^h\}$, 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 $\ell_{m'}$ became a lurker. Then,*

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

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

The next two lemmas shows that once an agent becomes a lurker, she is never called to move again along the passing path (she might be called to move again, but only after some other agent has clinched an object), and that once they are a lurker, they do not “stop” being a lurker along the passing path.

Lemma 7. *Let $h^0 = h_\emptyset$ and $h^n = (h_\emptyset, a^*, \dots, a^*)$ be the history reached by taking n passing actions from h_\emptyset . Let h^N be the terminating history. Take some h^n and $\ell \in \mathcal{L}^{BG}(h^n)$. Then, ℓ does not move at $h^{n'}$ for any $n' \in \{n, n+1, \dots, N\}$. In particular, the same is true for lurkers: if $\ell \in \mathcal{L}(h^n)$, then ℓ does not move at $h^{n'}$ for any $n' \in \{n, n+1, \dots, N\}$.*

Lemma 8. *Let $h^0 = h_\emptyset$ and $h^n = (h_\emptyset, a^*, \dots, a^*)$ be the history reached by taking n passing actions from h_\emptyset . Let h^N be the terminating history. For all $n = 1, \dots, N$, $\mathcal{L}^{BG}(h^{n-1}) \subseteq \mathcal{L}^{BG}(h^n)$. Further, the same is true for lurkers: $\mathcal{L}(h^{n-1}) \subseteq \mathcal{L}(h^n)$.*

The next lemma states that if a BG-lurked object is previously clinchable for an active BG non-lurker, then h is a terminating history, and this BG non-lurker is the terminator.

Lemma 9. (BG Lemma E.14) *Let $h = (h_\emptyset, a^*, \dots, a^*)$ 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.*

The next 3 lemmas relate lurkers to BG lurkers. They show that there can be at most 2 non-lurkers at any history and at most one BG lurker that is not a lurker; further, if such an agent exists, it is the youngest BG lurker, and we have reached the terminating history.

Lemma 10. Let $h^0 = h_\emptyset$ and $h^n = (h_\emptyset, a^*, \dots, a^*)$ be the history reached by taking n passing actions from h_\emptyset . Let h^N be the terminating history. For all $n < N$, $\mathcal{L}(h^n) = \mathcal{L}^{BG}(h^n)$.

Lemma 11. At any $h = (h_\emptyset, a^*, \dots, a^*)$, there is at most one BG lurker that is not a lurker. Further, if such an agent i exists, then: (i) i is the youngest BG lurker, (ii) i moves at the immediate predecessor of h , and first becomes a BG lurker at h (iii) h is a terminating history, (iv) the set of active agents at h is $\mathcal{A}(h) = \mathcal{L}(h) \cup \{i, t\} = \mathcal{L}^{BG}(h) \cup \{t\}$, where the agent t is the terminator and t moves at h and (v) t is not a lurker at h .

Lemma 12. (BG Lemma E.19) For all $h = (h_\emptyset, a^*, \dots, a^*)$, $|\mathcal{A}(h) \setminus \mathcal{L}(h)| \leq 2$.

Proof. If $\mathcal{L}(h) = \mathcal{L}^{BG}(h)$, then this follows from BG Lemma E.19, which shows there can be at most two active BG non-lurkers at any h . If $\mathcal{L}(h) \neq \mathcal{L}^{BG}(h)$, by Lemma 11, h is a terminating history and the active agents are $\mathcal{A}(h) = \mathcal{L}(h) \cup \{i, t\}$, where t is the terminator, and neither i nor t are lurkers. Thus, there are at most two non-lurkers at h . ■

The next lemma shows that at a history h , all unlurked objects have been previously offered to every lurker, while for any non-lurker, such objects have been previously offered or are still possible.

Lemma 13. (BG Lemma E.11) If agent i is active at a history h , then $\bar{\mathcal{X}}^{\mathcal{L}}(h) \subseteq P_i(h) \cup C_i^{\bar{c}}(h)$. If $i \in \mathcal{L}(h)$, then $\bar{\mathcal{X}}^{\mathcal{L}}(h) \subseteq C_i^{\bar{c}}(h)$.

Lemma 13 has the following corollary, which will be useful in the proofs of Lemmas 3, 4, and 5 later.

Corollary 1. If, at history h , agent i of type $>_i$ clinches $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ that is unlurked at h , then $x = \text{Top}(>_i, \bar{\mathcal{X}}^{\mathcal{L}}(h))$.

Proof. By Lemma 13, all unlurked objects have either been clinchable at some subhistory of h , or are still possible. Thus, if $x \neq \text{Top}(>_i, \bar{\mathcal{X}}^{\mathcal{L}}(h))$, it would not be obviously dominant for agent i to clinch $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ at h , a contradiction. ■

The next lemma is an analogue of Lemma 9 for non-lurkers (who may not be BG non-lurkers).

Lemma 14. Let $h = (h_\emptyset, a^*, \dots, a^*)$. If i is an active non-lurker at h and $x_\ell \in C_i^{\bar{c}}(h)$ for some lurked object $x_\ell \in \mathcal{X}^{\mathcal{L}}(h)$, 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).

The next lemma collects some results about the terminator that will be useful in the proofs of Lemmas 3, 4, and 5 later (parts (i)-(iii) of this lemma follow readily from BG; we show part (iv) in the Supplementary Appendix).

Lemma 15. (BG Lemmas E.18 and E.20) Let $h = (h_\emptyset, a^*, \dots, a^*)$ be a history with lurked objects. Let $i_{h'} = t$ be the agent who moves at the terminating history $h' = (h, a^*, \dots, a^*)$. Then:

- (i) Agent t is not a lurker at h .
- (ii) $C_t^c(h') = \mathcal{X}$.
- (iii) If $i_h \neq t$, then $C_{i_h}(h) \cap C_t^c(h) = \emptyset$.
- (iv) If $x_\ell \in P_j(h)$ for some non-lurker j and lurked object $x_\ell \in \mathcal{X}^L(h)$, then $j = t$.

The next lemma shows how, the first time an object is clinched after a series of passes, to determine the assignments of the active lurkers (if any).

Lemma 16. (BG Lemma E.17) Let $h = (h_\emptyset, a^*, \dots, a^*)$, and let i be the agent who moves at h . Let $a_x \in A(h)$ be a clinching action that clinches x for i . Let $\mathcal{L}(h) = \{\ell_1^h, \dots, \ell_{\lambda(h)}^h\}$ be the set of lurkers at h and $\mathcal{X}^L(h) = \{x_1^h, \dots, x_{\lambda(h)}^h\}$ the set of lurked objects, 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 is an older lurker than $\ell_{m'}^h$ (these sets may be empty). Consider a type profile \succ_N that reaches (h, a_x) . Let $\bar{h} \supseteq (h, a_x)$ be the unique terminal history that is reached by following strategy profile $S(\succ_N)$ starting from (h, a_x) . The object assigned to each agent in $\mathcal{L}(h)$ at \bar{h} can be determined using the following procedure:

- If $x \in \bar{\mathcal{X}}^L(h)$, each lurker ℓ_m^h receives her lurked object, x_m^h .
- If $x = x_{m_1}^h$ for some lurked $x_{m_1}^h$, then all older lurkers $\ell_{m'}^h$ for $m' < m_1$ receive their lurked object. Lurker $\ell_{m_1}^h$ receives $x' = \text{Top}(\succ_{\ell_{m_1}^h}, \mathcal{X} \setminus \{x_1^h, \dots, x_{m_1}^h\})$.
 - If $x' \in \bar{\mathcal{X}}^L(h)$, all remaining lurkers $\{\ell_{m_1+1}^h, \dots, \ell_{\lambda(h)}^h\}$ receive their lurked objects.
 - If $x' = x_{m_2}^h$ for some lurked object $m_2 > m_1$, all older unmatched lurkers (ℓ_m^h for $m_1 < m' < m_2$) receive their lurked objects. Lurker $\ell_{m_2}^h$ receives $x'' = \text{Top}(\succ_{\ell_{m_2}^h}, \mathcal{X} \setminus \{x_1^h, \dots, x_{m_2}^h\})$.

This procedure continues with each lurker whose lurked object is taken by an older lurker receiving her top remaining object among all that have not been assigned, until some agent (either i , or some lurker in $\mathcal{L}(h)$) receives an object that was unlurked at h , denoted \hat{x} . All remaining unmatched lurkers receive their lurked objects.

Proof. By Lemma 7, the agent i who moves at h is not a BG lurker, and so we can apply BG Lemma E.17. As every lurker is a BG lurker, the result follows by BG Lemma E.17. ■

Lemma 16 allows us to calculate the objects received by all active lurkers after the first clinching action following a sequence of passes (as well as the agent i who takes this first clinching action at h and who, as noticed in the proof, is not a lurker at h). Exactly one

of these agents receives an object that is unlurked at h , and we let \hat{x} denote this unlurked object. If the object x clinched by the agent i who moves at h is unlurked, then $\hat{x} = x$; otherwise, \hat{x} is assigned to one of the lurkers in $\mathcal{L}(h)$.

The assignments from Lemma 16 can be implemented as follows. After i clinches x at h , then all lurkers at h (if any exist) who lurked an object older than x have immediately clinched the object they lurked at (h, a_x) (if i clinches an unlurked object, i.e., $x = \hat{x}$, then this is all lurkers at h). These lurkers may but do not need to be asked to move following h . If they are called to move, by properties 3 and 4 of Pareto efficient millipedes, they have only clinching actions for their lurked object (their action does not affect their own object, but may influence the continuation game), and they are never called to move again. If x is a lurked object, then the agent who lurked x at h , say ℓ , is the oldest lurker who does not receive their lurked object. Following the move by i at h and possible moves by the older lurkers who clinched their lurked objects at (h, a_x) , by properties 3 and 4 of Pareto efficient millipedes, ℓ is called to move and is offered for clinching all objects that are possible for them, which consists of all lurked objects younger than the one they lurked at h , plus all unlurked objects at h ; they do not have passing actions.⁵⁵ If ℓ takes a lurked object, say x' , then we repeat the above process whereby lurkers who lurked objects older than x' but younger than x may or may not be called to move (and if they do move, they have only clinching actions for their lurked object), then the lurker who lurked x' is called to move, etc. until someone (either i , the agent who moved at h , or one of the lurkers at h) clinches an unlurked object, \hat{x} . After this clinching of \hat{x} , all remaining lurkers have clinched their lurked object. As above, these agents may or may not be called to move, and if they are called to move, they have only clinching actions, and are never called to move again.

Remark 4. The above paragraph describes how lurkers (if any exist) are called to move after an initial clinching action is taken at a history h (though as described, not all need to be called to move). We use the notation $\overset{\circ}{h}$ to denote the history that is reached after the completion of all such moves by lurkers. That is, $\overset{\circ}{h}$ is the smallest history such that $\overset{\circ}{h} \supseteq (h, a_x)$ and no lurker at h is called to play at any continuation history $h' \supseteq \overset{\circ}{h}$ (if no lurkers at h are called to play following h —including the case that none exist—then $\overset{\circ}{h} = (h, a_x)$). How the game proceeds starting from $\overset{\circ}{h}$ is the subject of the next lemmas.

As there can be at most 2 non-lurkers at h (Lemma 12), and the agent who moves (and clinches) at h is not a lurker at h (Lemma 7), there might be at most one active agent at h whose assignment is not determined by Lemma 16. As we show next, if such an agent j

⁵⁵It is possible that some lurker ℓ , for whom there is only one object in $P_\ell(h)$ minus the set of objects taken by older lurkers, is not asked to move (this case can only happen for the youngest lurker at the end of the game).

exists, then it will be this agent who moves at $\overset{\circ}{h}$. The assignment of this agent (that is j) is the subject of the next two lemmas. Lemma 17 considers the case where $\hat{x} \in C_j^c(h)$, while Lemma 18 considers the case where $\hat{x} \notin C_j^c(h)$. Note that in these lemmas, j is a non-lurker, but may or may not be a BG lurker.

Lemma 17. *Consider the environment described in Lemma 16 and assume there exists an active non-lurker $j \in \mathcal{A}(h) \setminus \mathcal{L}(h)$ who does not move at h (thus $j \neq i$) and is such that $\hat{x} \in C_j^c(h)$, where $\hat{x} \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ is the unlurked object from Lemma 16. Then, at \bar{h} , j receives $Top(>_j, \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\})$.*

If an agent j described as in Lemma 17 exists, then by properties 3 and 4, agent j is called to move at the history $\overset{\circ}{h}$ defined in Remark 4. At this move, agent j is offered at least one clinching action for each object that remains possible for her. There is no passing action.⁵⁶

The final case to consider is when there is an active non-lurker j' at h who does not move at h , and $\hat{x} \notin C_{j'}^c(h)$.

Lemma 18. *Consider the environment described in Lemma 16 and assume there exists an active non-lurker $j \in \mathcal{A}(h) \setminus \mathcal{L}(h)$ who does not move at h (thus $j \neq i$) and is such that $\hat{x} \notin C_j^c(h)$, where $\hat{x} \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ is the unlurked object from Lemma 16. Let $\overset{\circ}{h} \supseteq (h, a_x)$ be the history defined in Remark 4. Then, there exists an equivalent millepede mechanism that satisfies properties 1-5 and in which j moves at $\overset{\circ}{h}$ and $C_j^c(h) \subseteq C_j(\overset{\circ}{h})$ (there may be additional clinching options and/or a passing action as well).*

The above lemmas establish property 5 of Pareto efficient millipede games for the first round described in Remark 2. To extend these properties to the subsequent rounds, and hence the entire game tree, we define the initiating history of the next round, $h_{\mathcal{O}}^2$, as follows. Let $h_{\mathcal{O}}^2 = \overset{\circ}{h}$, except if there exists a j as in Lemma 17 (in particular, $\hat{x} \in C_j^c(h)$); in the latter case, $h_{\mathcal{O}}^2 = (\overset{\circ}{h}, a_j)$, where a_j is the clinching action taken by agent j who moves at $\overset{\circ}{h}$. In the former case (i.e., $h_{\mathcal{O}}^2 = \overset{\circ}{h}$), if there is a j as in Lemma 18 (in particular, $\hat{x} \notin C_j^c(h)$), agent j moves at $h_{\mathcal{O}}^2$, by Lemma 18. If no such j exists, then some previously inactive agent moves at $h_{\mathcal{O}}^2$ (unless all agents have clinched an object, in which case the game ends).

The continuation game starting at the new initiating history $h_{\mathcal{O}}^2$ is another Pareto efficient and pruned millipede game on the remaining unmatched agents and objects.⁵⁷ The

⁵⁶Also, note while the lemma says that j receives her top object from $\bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\}$, pruning implies that when j moves, she need not be actually offered all of these objects at her move; in particular, the mechanism only offers her those objects which were possible for her when she entered the game and have not yet been clinchable for her.

⁵⁷In particular, when restricted to the objects that are not yet clinched at $h_{\mathcal{O}}^2$, all possible ranking types

characterization implied by the above lemmas continues to apply (where the new initiating history h_\emptyset^2 plays the role of h_\emptyset), and so the continuation game starting from h_\emptyset^2 has the same passing-and-clinching structure. We repeat these arguments throughout the game tree by applying the above lemmas starting from h_\emptyset^k , the initiating history of round k . This completes the proof of Lemma 2.

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 .⁵⁸ In the proofs, we sometimes 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 the game form, and are independent how the specific agents are assigned to the roles in the game form.

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 direct consequence of part (iv) of the definition of a lurker, but deserves an emphasis, as it arises frequently in the arguments below.

Remark 5. If, at a history $h = (h_\emptyset^k, a^*, \dots, a^*)$ in a passing phase of the game (where h_\emptyset^k is the initiating history of the round k that contains h), $x \in C_j(h)$ for an active non-lurker j at h , then x cannot become the next lurked object along the passing path (h, a^*, \dots, a^*) .⁵⁹

are present at h_\emptyset^2 for all agents. For those agents who have never been called to move, this is obvious, as they have not yet revealed any preference information. For the agent j who might be carried over from the previous round, it follows because each type that ranks \hat{x} first (over the set of all objects \mathcal{X}) reaches h_\emptyset^2 , and \hat{x} is no longer available at h_\emptyset^2 .

⁵⁸To avoid layered subscripts, we write Γ_A instead of Γ_{σ_A} .

⁵⁹If x is already lurked at h , it cannot become lurked (and further, by Lemma 14, h is a terminating

We first state a preliminary lemma that will be useful in the main proofs of Lemmas 3, 4, and 5.

Lemma 19. *Consider a round of the mechanism that begins with an initiating history h_{\emptyset}^k , and let i and j be active non-lurkers at a history $h = (h_{\emptyset}^k, a^*, \dots, a^*)$. Let $y \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ be an unlurked object at h . Further, assume that $i_h = i$ and $y \in C_i(h) \cap C_j^{\bar{c}}(h)$. Consider a type \succ_j that reaches h , and define $\bar{x} = \text{Top}(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h))$. Then, $\bar{x} \succ_j y$.*

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 $\mathcal{X}^{\mathcal{L}}(h^*) = \{x_1, \dots, x_n\}$ 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 $f_{\sigma} : \ell_1, \ell_2, \dots, \ell_n, i^*$. By Lemma 6, for each lurker ℓ_m , we have $P_{\ell_m}^h(h) = \mathcal{X}(h) \setminus \{x_1^h, \dots, x_{m-1}^h\}$ and hence $x_m = \text{Top}(\succ_{\ell_m}, \mathcal{X} \setminus \{x_1, \dots, x_{m-1}\})$. 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 1, $y = \text{Top}(\succ_{i^*}, \bar{\mathcal{X}}^{\mathcal{L}}(h^*))$. 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 x_m at h^* in Γ_{σ} , we

history so there is no further passing path). Thus, suppose x is unlurked and it first becomes lurked at a history $h' = (h, a^*, \dots, a^*)$, and no other object (besides those already lurked at h , if any) becomes newly lurked between h and h' . Then, some agent i moves at the immediate predecessor of h' , say h'' , and becomes a lurker for x at $h' = (h'', a^*)$. Clearly, $i \neq j$ (as j has been offered x previously, she cannot lurk x), and as no other object has become lurked (or clinched) at h'' , j is an active non-lurker at h'' and $x \in C_j^{\bar{c}}(h'')$. Therefore, condition (iv) in the definition of a lurker is violated. An equivalent statement can be obtained by replacing “ $C_j(h)$ ” with “ $C_j^{\bar{c}}(h)$ ”, using the same argument. Of course, x may eventually become lurked, but not before some other object becomes lurked (or is clinched). This is also an important distinction between lurkers/lurked objects and BG lurkers/lurked objects: it is possible that x is the next BG-lurked object, but if this occurs, we immediately reach the terminating history (by Lemma 14), and we do not classify the agent who BG-lurks x as a lurker according to our definition.

have $x_m = \text{Top}(>_i, \mathcal{X})$.⁶⁰ Thus, at her turn in the SD, she 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.⁶¹ 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 $\ell_1 \tilde{>}^1 \dots \tilde{>}^1 \ell_n \tilde{>}^1 \{i^*, j\}$. 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 $h' \not\subseteq h^*$. Let $z = \text{Top}(>_j, \bar{\mathcal{X}}^\mathcal{L}(h^*))$, and note that by Lemma 19, $z >_j y$. Since i clinched y at h^* , we have $y >_i z$. 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 coding algorithm. Notice that step 1 of the coding algorithm corresponds to round 1 of the Pareto efficient millipede game described in the proof of Lemma 2; that is, the agents coded in step 1 are precisely the agents whose assignments are determined in Lemmas 16 and 17, and thus those whose assignments are determined in the ‘‘clinching phase’’ of round 1, as described in the proof of Lemma 2. If at the end of this clinching phase, all agents have been coded, the algorithm ends and the proof is complete. If there exist remaining unmatched agents, we consider the initiating history h_\emptyset^2 that starts the continuation subgame that begins round 2. When there exists an agent j' who was active prior to h_\emptyset^2 and whose assignment was not determined by Lemmas 16 and 17, this agent is not coded in step 1 of the coding

⁶⁰By Lemma 14, h^* is the terminating history, and i^* is the terminator. By Lemma 15 $C_{i^*}^\subseteq(h^*) = \mathcal{X}$, i.e., i^* has been offered all objects in \mathcal{X} and chooses x_m , and so $x_m = \text{Top}(>_i, \mathcal{X})$.

⁶¹When 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.

algorithm and the next round begins with j' moving at h_{\emptyset}^2 and being offered to clinch at least each object in $C_{j'}^{\neq}(h_{\emptyset}^2)$ (and given possibly more actions as well), as described in the proof of Lemma 2. When no such j' exists, some other previously inactive agent k moves at h_{\emptyset}^2 . In either case, Step 2 of coding of the original game is the same as Step 1 of coding of the continuation subgame starting from h_{\emptyset}^2 ; note that, like the original game, this continuation game is a Pareto efficient millipede in which the unmatched agents are assigned the unmatched objects. Coding step 2 of the overall game is thus equivalent to the initial coding step on this smaller subgame, and the same arguments show that agents in this coding step receive the same objects under the corresponding serial dictatorship. We continue to proceed recursively in this way through steps $3, \dots, K$ to finish the proof. ■

Proof of Lemma 4

First consider $k = 1$ and suppose \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 , plus 1. We add the “+1” so that this number corresponds to i 's picking order in the resulting serial dictatorship (i.e., if $\succ_A: i, j, k \dots$ then k will pick third in the resulting serial dictatorship, and $g_A(k) = 2 + 1 = 3$). This will hold for all agents except for those who tie under \succ_A ; if i and i' tie, then $g_A(i) = g_A(i')$. Define g_B similarly.

Claim 1. If \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 $h_A^1 \not\subseteq h_B^1$. 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 (because $h_A^1 \not\subseteq h_B^1$, which implies $h_B^1 = (h_A^1, a^*, \dots, a^*)$), object x_A is unlurked at h_A^1 , by Lemma 14. Since i_A clinches an unlurked object at h_A^1 , we have $x_A = \text{Top}_{>_{i_A}}(\bar{\mathcal{X}}^{\mathcal{L}}(h_A^1))$ by Corollary 1. 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 \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.⁶²

We claim that $\mathcal{X}^{\mathcal{L}}(h_A^1) = \mathcal{X}^{\mathcal{L}}(h_B^1)$. First, notice that $h_A^1 \not\subseteq h_B^1$ 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

⁶²This is a key point, and its analogue remains true in the alternate case $h_B^1 \not\subseteq h_A^1$. 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 $h_B^1 \not\subseteq h_A^1$ 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 \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 .

object becomes lurked, and once an object is lurked, it remains lurked along the passing path (by Lemmas 6 and 8). If $\mathcal{X}^{\mathcal{L}}(h_B^1) \not\supseteq \mathcal{X}^{\mathcal{L}}(h_A^1)$, 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 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 remark 5, 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 $h_A^1 \not\supseteq h_B^1$. 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 thus is not a lurker at h_A^1), and there is a passing action, so another active non-lurker moves at the history (h_A^1, a^*) . Since $\mathcal{L}_R(h_A^1) = \mathcal{L}_R(h_B^1)$, there are no new lurkers at h_B^1 , and there can be at most two active non-lurkers at any history, it must be role s' that moves at (h_A^1, a^*) , and both roles s and s' remain active non-lurkers at h_B^1 . Further, as only non-lurkers are called to move along the passing path, one of these roles must move 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 $x_A \in C_s^{\bar{c}}(h_B^1)$ 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, by Lemma 14. 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 15. 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 \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 $x_A \in C_{s'}^{\bar{c}}(h_A^1)$ 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 $h' \not\supseteq h_A^1$ at which she was offered to clinch x_A in Γ_B . By Lemma 19, $Top(\succ_{i_A}, \bar{\mathcal{X}}^{\mathcal{L}}(h_A^1)) \succ_{i_A} x_A$,

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 14, 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 $x_A \in C_{s'}^{\neq}(h_A^1)$, which contradicts Lemma 15, part (iii).

The case $h_B^1 \not\subseteq h_A^1$ follows an analogous argument; cf. footnote 62 for the needed adjustments. ■

Thus far, we have shown that if \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

$$\sigma_A(r_0) \rightarrow x_{a_1} \rightarrow \sigma_A(r_{a_1}) \rightarrow x_{a_2} \rightarrow \cdots \rightarrow \sigma_A(r_{a_M}) \rightarrow x_{a_{M+1}} \quad (\text{A})$$

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

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

Note that the agents who begin the chains, $\sigma_A(r_0)$ and $\sigma_B(r_0)$ are not lurkers in their respective games,⁶³ while all of the remaining agents are lurkers. 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 $m = 0, \dots, M$, 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, $\sigma_A(r) = \sigma_B(r)$.

⁶³If there are no lurkers at h^1 , this is obvious; if there are lurkers, it follows from Lemma 7.

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 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 \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 $x_{M+1} \notin C_s^{\neq}(h^1)$, then this role is not coded in step 1, and we are done. Thus, assume that s exists, and that $x_{M+1} \in C_s^{\neq}(h^1)$. 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, \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 $x_{a_1} \neq x_{b_1}$ by Lemma 3. If $M = M' = 0$, then each of the two chains has only one agent, $\sigma_A(r_0)$ or $\sigma_B(r_0)$, and these agents 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 \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 $x_{a_1}, x_{b_1} \in C_s^{\neq}(h^1)$. Since i clinches an unlurked object x_{a_1} at h^1 in Γ_A , we have $x_{a_1} = \text{Top}(\succ_i, \bar{\mathcal{X}}^{\mathcal{L}}(h^1))$, by Corollary 1. Now, consider game Γ_B . Since $\sigma_B(s) = i$ and $x_{a_1} \in C_s^{\neq}(h^1)$, in game Γ_B , there is some history $h' \not\subseteq h^1$ such that $x_{a_1} \in C_i(h')$. By Lemma 19, we have $\text{Top}(\succ_i, \bar{\mathcal{X}}^{\mathcal{L}}(h^1)) \succ_i x_{a_1}$, 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 14. It also implies 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 .⁶⁴ Similarly, 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} .⁶⁵ As we proceed in this way, agent $\sigma_A(r_{a_2})$ must lurk x_{a_3} in Γ_B , etc., until agent $\sigma_A(r_{a_{M-1}})$, who must lurk object x_{a_M} in Γ_B .

Finally, consider agent $\sigma_A(r_{a_M})$, and, for shorthand, define $k := \sigma_A(r_{a_M})$. We claim that k must be in role s —the active non-lurker role at h^1 that does not move at h^1 —in Γ_B , i.e., $\sigma_B^{-1}(k) = s$ (and in particular, such a role s must exist at h^1). To see this, 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 (because $h_A^1 = h_B^1$, as shown above), this is only possible if agent k is also coded in step 1 of Γ_B , and thus k must be active at h^1 . Further, because k lurks x_{a_M} in Γ_A , she strictly prefers x_{a_M} to any younger lurked object and also to any unlurked object. Thus, at h^1 in Γ_B , k cannot be a lurker for any lurked object older than x_{a_M} , because otherwise k would have been offered to clinch x_{a_M} , and by obvious dominance, cannot end up with something she strictly disprefers. Next, k cannot lurk x_{a_M} itself in Γ_B , because as we just argued, agent $\sigma_A(r_{a_{M-1}})$ is in this role in Γ_B . Nor can k be a lurker for any lurked object younger than x_{a_M} . This is because k chose the (unlurked) object $x_{a_{M+1}}$ in Γ_A , which implies $x_{a_{M+1}}$ is her top choice among the set of objects that consist of all unlurked objects plus all lurked objects younger than x_{a_M} (as all of these objects were available to her in Γ_A when she was called to move after the object she lurked, x_{a_M} , was taken by another agent). Finally, k cannot be in role r_0 in Γ_B (the terminator role), because at h^1 , all objects have been offered to the terminator, and thus k would choose to clinch her top object among all of \mathcal{X} (which we have shown is an object different from $x_{a_{M+1}}$), which contradicts Lemma 3. As we know that k must be active at h^1 , the only remaining possibility is $\sigma_B^{-1}(k) = s$.

Because k is coded in step 1 of Γ_B and is in role s , it must be the case that k ties in \succ_B ,

⁶⁴Since agent $\sigma_A(r_0)$ clinches x_{a_1} at the terminating history in Γ_A , we know by Lemma 15 that $x_{a_1} = \text{Top}(\succ_{\sigma_A(r_0)}, \mathcal{X})$. By Lemma 3, agent $\sigma_A(r_0)$ continues to receive x_{a_1} in game Γ_B . Since x_{a_1} is lurked, this object might be received in Γ_B only by an agent in role r_{a_1} , an older lurker, or 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} prior to becoming a lurker, and since it is her top possible object, by greedy strategies, she would have clinched it prior to h^1 . Nor can she be the terminator, because, by assumption, $\sigma_B(r_0) \neq \sigma_A(r_0)$. Therefore, she must be in role r_{a_1} in Γ_B .

⁶⁵This is because by definition of a lurker, agent $\sigma_A(r_{a_1})$ strictly prefers x_{a_1} to all younger lurked objects 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 by obvious dominance 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 $h_A^1 = h_B^1$ is the terminating history, all objects would have been offered to the terminator, and so the terminator would receive her top choice among all possible objects which is different than x_{a_2} (the object she received in Γ_A), which is again a contradiction to Lemma 3.

and so 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 $x_{a_{M+1}} \in C_s^c(h^1)$. Finally, since $\sigma_B^{-1}(k) = s$, we have $x_{a_{M+1}} \in C_k^c(h^1)$ in Γ_B , and by Lemma 19, $Top(>_k, \bar{\mathcal{X}}^{\mathcal{L}}(h^1)) >_k x_{a_{M+1}}$. However, since k chose to clinch $x_{a_{M+1}}$ in Γ_A and $x_{a_{M+1}}$ was unlurked, we have $Top(>_k, \bar{\mathcal{X}}^{\mathcal{L}}(h^1)) = x_{a_{M+1}}$, 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. \blacksquare

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 $1, \dots, k$ of the coding algorithm. At the conclusion of the round k clinching phase initiated at h_A^k (which is the same as h_B^k), we enter a continuation game on the remaining unmatched agents and objects (round $k + 1$) that begins with the initiating history h_{\emptyset}^{k+1} . By the inductive assumption, this initiating history h_{\emptyset}^{k+1} is the same under both σ_A and σ_B . As argued in Remark 2, this continuation game continues to have the structure of a millipede mechanism satisfying properties 1-5. Let $h_A^{k+1} \supseteq h_{\emptyset}^{k+1}$ 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 h_{\emptyset}^{k+1} ; define $h_B^{k+1} \supseteq h_{\emptyset}^{k+1}$ analogously. If now \triangleright_A^{k+1} equals to an initial segment of \triangleright_B restricted to the remaining agents still unmatched at the beginning of step $k + 1$, 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. \blacksquare

Proof of Lemma 5

As noticed in the main text, for $n = 0$ or $m = 0$ the lemma is immediate. Hence in the rest of the proof we assume that $n, m \geq 1$. 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,

$h_\tau^* = (h_\emptyset, a^*, \dots, a^*)$, where a^*, \dots, a^* are the 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 .

It is without loss of generality to assume that for all games Γ_τ in arm α and its dual arm α' , at least $P+1$ agents are coded in the first step of the coding algorithm that is initiated at h_τ^* . To see this, note that if not, then the agents coded in the first step are $j_1, \dots, j_{p'}$ for some $p' \leq P$. By Lemma 4 we have that in all of the games under consideration in the two arms, 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. The game that starts at this history is also an efficient millipede and if the lemma holds true for this smaller game then it holds for the arms α and α' in the original game. Repeating this argument, we conclude that to prove the lemma it is sufficient to prove it for dual arms in which all agents up to at least k_1 (for arm α) or i (for arm α') are coded in the first round of the coding of the game.

Thus, for the entirety of this proof (including all sublemmas stated therein), we assume that the agents j_1, \dots, j_P are all coded in the first round of the games we consider. Since ties can only occur at the very end of a coding step, this implies that each coding in an arm begins $j_1 \succ_\tau j_2 \succ_\tau \dots \succ_\tau j_P \succ_\tau \dots$ and that all x_{j_p} for $p = 1, \dots, P$ are lurked objects at h_τ^* . Letting $\check{h} = (h_\emptyset, a^*, \dots, a^*)$ be the earliest history at which the youngest of these objects becomes lurked (hence, all x_{j_p} are lurked at \check{h}), we also have $h_\tau^* = (\check{h}, h, a^*, \dots, a^*)$ for some number (possibly zero) of passing actions a^* . It is possible that $P = 0$, 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 $x_i = \text{Top}(\succ_i, \bar{\mathcal{X}}^\mathcal{L}(h_1^*))$. 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 $x_i = \text{Top}(\succ_i, \bar{\mathcal{X}}^\mathcal{L}(h_{n'}^*))$ because each of the agents i, j_1, \dots, 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 possibly the objects assigned to j_1, \dots, j_P , and in game $\Gamma_{n'}$ none of these objects belongs to $\bar{\mathcal{X}}^\mathcal{L}(h_{n'}^*)$.

We begin with the following Lemmas 20-24, which show that, under certain conditions, either statement (I) or (II) from Lemma 5 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.

Lemma 20. *Assume that there exists an arm α 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) $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n-1}}$.⁶⁶ Then, at h_{n+1}^* in Γ_{n+1} , there is*

⁶⁶We 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, and is the $(P+1)^{\text{th}}$ lurked

an agent $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n, i$ such that ℓ is an active non-lurker at h_{n+1}^* that does not move at h_{n+1}^* and $x_i \in C_\ell^{\neq}(h_{n+1}^*)$. Further, i must tie with some other agent in \succ_{n+1} , and we label this agent k_{n+1} .

Remark 6. A supposition in Lemma 20 (and in Lemma 25, below) is that the first lurked objects of the game form are $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n-1}}$, in this order, where $n \geq 2$. Given our assumption that objects x_{j_1}, \dots, x_{j_P} are all lurked, 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 $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n'-1}}$ 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 . ■

Lemma 21. *Assume that there exists an arm α as defined in the statement of Lemma 5, and such that $n \geq 2$. Consider game Γ_n , and let Q be the step of the coding algorithm in which i is coded in game Γ_n . Let h_n^{q*} be the history at which the first clinching action is taken in step $q \in \{1, \dots, Q\}$ of game Γ_n .⁶⁷ 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 $x_i \in C_\ell^{\neq}(h_n^{1*})$, then, in \succ_{n+1} , agent i must tie with some agent k_{n+1} .*

Lemma 22. *Assume that there exists an arm α as defined in the statement of Lemma 5, and such that $n \geq 2$. Further, assume that the $(P+1)^{\text{th}}$ lurked object along the initial passing path of the game form is x_{k_1} . Then, in \succ_{n+1} , agent i must tie with some agent k_{n+1} .*

Lemma 23. *Assume that there exists an arm α 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 arm α , i must tie with some agent k_{n+1} .

object. Note that this only says that there is some history $h = (h_\emptyset, a^*, \dots, a^*)$ of the game form at which x_{k_1} is lurked. Whether the history h is reached in any particular game Γ_τ depends on the role assignment and the preference profile of the agents.

⁶⁷That is, $h_n^{q*} = (h_\emptyset^q, a^*, \dots, a^*)$ and the agent who moves at h_n^{q*} takes a clinching action. Notice that for $q = 1$, we have $h_n^{1*} = h_n^*$ in our earlier notation.

Remark 7 (Symmetry). Lemmas 20–23 were stated for arm α , and concluded that i must tie in \succ_{n+1} . There are also symmetric versions of these lemmas that apply to arm α' and conclude that k_1 must tie in \succ'_{m+1} that have the exact same proof.

Lemma 24. *Assume that there exist dual arms α and α' 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}, \dots, 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} of arm α , agent i must tie with some agent k_{n+1} .*
2. *In \succ'_{m+1} of arm α' , agent k_1 must tie with some agent k'_{m+1} .*

With Lemmas 20–24 in hand, we 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 23 to arm α 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 23 (see Remark 7) to arm α' to conclude that (II) holds.

If there do not exist such σ_0 nor σ'_0 as in the above bullet points:

- If $n = 1$, then arm α has length $n + 1 = 2$ and the final partial ordering of the arm begins $j_1 \cdots j_P \succ_2 k_1 \succ_2 i \cdots$. Since in this case we assume there is no σ'_0 of the form above (in which i is ranked strictly), it must be that i ties in \succ_2 , and thus (I) holds.
- If $m = 1$, a similar symmetric argument to the previous bullet point can be made for arm α' to show that (II) holds.
- If $n, m \geq 2$:
 - If x_{k_1} is the $(P+1)^{th}$ lurked object along the initial passing path of the game form, then we apply Lemma 22 to arm α conclude that (I) holds.
 - If x_i is the $(P+1)^{th}$ lurked object along the initial passing path of the game form, then we apply the symmetric version of Lemma 22 (see Remark 7) to arm α' 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 of the game form, we apply Lemma 24 to conclude that either (I) or (II) must hold.

The above covers all possibilities,⁶⁸ which completes the proof of Lemma 5. ■

⁶⁸Recall that the lemma is immediate if either m or n are equal to 0, and thus we only need consider $n, m \geq 1$.

D Extensions

So far we have focused on the canonical case in which the number of objects and the number of agents are equal: $|\mathcal{X}| = |\mathcal{N}|$. In this section, we argue that our characterization of Random Priority continues to hold for arbitrary cardinalities of these two sets. Allowing $|\mathcal{X}| < |\mathcal{N}|$ requires us to amend the model by allowing agents to not receive an object. We model this by assigning an “outside option” to agents who do not receive an object

Formally, let \mathcal{N} be the set of agents and \mathcal{X} be the set of objects; we allow arbitrary non-empty finite sets \mathcal{N} and \mathcal{X} , imposing no assumptions on their relative cardinalities. We use the notation o to denote the outside option. A deterministic allocation is now a function $\mu : \mathcal{N} \rightarrow \mathcal{X} \cup \{o\}$ such that, for each $x \in \mathcal{X}$, $|\mu^{-1}(x)| \leq 1$; that is, for each $x \in \mathcal{X}$, there is at most one agent $i \in \mathcal{N}$ such that $\mu(i) = x$, while the outside option o can be assigned to an arbitrary number of agents. Each agent $i \in \mathcal{N}$ has a strict preference relation, \succ_i , over $\mathcal{X} \cup \{o\}$ and prefers any object (that is any $x \in \mathcal{X}$) to the outside option o . Note that if $|\mathcal{X}| \geq |\mathcal{N}|$ then any Pareto efficient mechanism assigns an object to each agent and none of the agents receives the outside option; if $|\mathcal{X}| < |\mathcal{N}|$ then any Pareto efficient mechanism assigns the outside option to exactly $|\mathcal{N}| - |\mathcal{X}|$ agents.

Our main result (Theorem 1) extends to this general environment: *An obviously strategy-proof mechanism is symmetric and Pareto efficient if and only if it is equivalent to Random Priority.* In remainder of this section, we show how to leverage our base proof to derive this more general result.

If $|\mathcal{X}| \geq |\mathcal{N}|$, the presence of outside options does not affect efficient allocations and our base proof applies with no changes needed.⁶⁹ Suppose hence that $|\mathcal{X}| < |\mathcal{N}|$. The proof for this case has the same structure as the base proof. Steps 1 and 2 of the base proof are unchanged; in particular, Pycia and Troyan (2023b) established all results on which we rely in Step 2 for a general class of environments that include the environment with outside options we study now. We can hence again restrict attention to perfect information extensive-form millipede mechanisms with greedy strategies. Let (Γ, S) be such a millipede mechanism with game form Γ and a profile S of greedy strategies. The key difference with the base case is that at any non-terminal history h , the agent who moves at h might be offered to clinch o . We call any such clinching action an ***o-action*** and any history when an *o*-action is offered an

⁶⁹In particular, the results of Pycia and Troyan (2023b) and Bade and Gonczarowski (2017) on which we build were established for $|\mathcal{X}| \geq |\mathcal{N}|$. We do not rely on $|\mathcal{X}| = |\mathcal{N}|$ directly in our proof either; e.g., at no point in our base proof do we infer the existence of agents from the base model assumption that there at least as many agents as objects. The only equality of cardinalities that matter in the argument is the equality between the number of possible role assignments and the number of possible orderings of agents; this equality—a prerequisite for the existence of a bijection between allocations in a symmetrized millipede and Random Priority—obtains whether $|\mathcal{X}| = |\mathcal{N}|$ holds or not.

o -history.⁷⁰ We also refer to any agent receiving o at all terminal histories $\bar{h} \supseteq h$ as an **o -agent** at h . Denote the set of o -histories as \mathcal{H}_o .

Steps 3-6 of the base proof are not directly valid; however, the following proposition allows us to rely on results from these steps without the need to re-prove them.⁷¹

Proposition 3. *Suppose (Γ, S) is a pruned obviously strategy-proof mechanism. If agent i has an o -action at some history h then this agent is never offered to clinch any object $x \in \mathcal{X}$ at any subhistory of h nor at any continuation history of h .*

If (Γ, S) is a millipede mechanism, then given our definition of greedy strategies, we can say more: if a path of successive histories from the root of the game to a terminal history includes an o -history at which agent i moves then this agent moves exactly once through this path of histories.

Proof. Consider an agent i that has an o -action at some history h . Suppose i was offered to clinch an object x at h or some subhistory $h' \subseteq h$. By obvious strategy-proofness, at any terminal history that follows h' , including all terminal histories that follow h , this agent must receive either x or an object strictly preferred to x . As the agent strictly prefers x over o , in a greedy strategy this agent would never chose an o -action at any continuation history of h' , including at h . In effect, in a pruned mechanism o -actions are not offered at such h .

Similarly, if there is some continuation history $h'' \supsetneq h$ at which agent i is offered to clinch an object x , then there must be some action $a \in A(h)$ for which x is possible at (h, a) . Thus, any greedy strategy directs all types of agent i to choose an action other than o -action. In effect, again, in a pruned mechanism o -actions are not offered at such h . ■

The above implies that if i is an o -agent at h , then she receives o at all terminal histories \bar{h} such that $\bar{h} \supseteq h'$ where h' is either a subhistory or a continuation history of h .⁷² As we assume each history in a game form has at least 2 actions, at any o -history there are at least two o -actions. Let us denote by c^o a collection of possible choices o -agents take at o -histories, i.e., for any $h \in \mathcal{H}_o$, $c^o(h) \in A(h)$ and $c^o = (c^o(h))_{h \in \mathcal{H}_o}$. Let \mathcal{C}^o be the set of all possible c^o . Fixing $c^o \in \mathcal{C}^o$, at each o -history h we can remove all actions different from $c^o(h)$ and all of their continuation histories, thus turning each such o -history h into a history at which only one action is offered. If h was the root history, we can hence start the game at the next history. Similarly, if some agents who are not o -agents moved before h , we can let

⁷⁰There might be multiple o -actions at an o -history which can affect the continuation game.

⁷¹One reason we need to make these adjustments is that the results of Bade and Goncezarowski (2017) we draw on in the base analysis were stated for $|\mathcal{X}| \geq |\mathcal{N}|$. See Appendix E.1 of their paper for a discussion of complications that arise in their results when we allow outside options.

⁷²Note that this does not imply i receives o at all terminal histories; indeed, there may be other \bar{h}' that do not contain h or any subhistory of h where i receives an object from \mathcal{X} .

the last move of such agent to lead directly to the first continuation history that is not an o -history; note that such a history exists (as terminal histories are not o -histories) and is uniquely determined (as there is a single action left at any o -history). Proceeding in this way, we create a uniquely-determined game form $\Gamma(c^o)$ that includes no o -histories. Let $S(c^o)$ be the projection of S onto histories in $\Gamma(c^o)$ that correspond to histories we did not remove. The pair $\Gamma(c^o)$ and $S(c^o)$ create a c^o -reduced mechanism $(\Gamma(c^o), S(c^o))$. Note at any terminal history of $\Gamma(c^o)$ all agents (including o -agents) receive the same outcomes as in the corresponding terminal history of Γ .

So far we have assumed that (Γ, S) is a millipede mechanism. As in the base proof, in Steps 3-6, we assume without loss of generality that (Γ, S) is a Pareto efficient millipede mechanism.

Because (Γ, S) is Pareto efficient so is the mechanism $(\Gamma(c^o), S(c^o))$ for any fixed c^o (because the corresponding outcomes are the same). Because (Γ, S) is obviously strategy-proof, so is the mechanism $(\Gamma(c^o), S(c^o))$ for any fixed c^o (because the set of outcomes possible at each history in $(\Gamma(c^o), S(c^o))$ is a weak subset of the set of outcomes possible at the corresponding history in (Γ, S) , and so any time an agent is called to move in the reduced mechanism, the worst case outcome from her called-for action only improves while the best case from any alternative only disimproves, and thus if the obvious dominance condition held previously, it will continue to hold in the reduced game). While we still have fewer objects than agents, given the recursive structure of our analysis, all of our results—from Steps 3-6—are valid for the mechanisms $(\Gamma(c^o), S(c^o))$.⁷³ We use these results to extend our bijection construction from our base case to the present general case.

In order to simplify Steps 4-6, we assume that in Γ all o -agents are given at least one move at some $h \not\subseteq \bar{h}$ for any terminal history \bar{h} . This assumption is without the loss of generality because if some o -agents do not move by some terminal history \bar{h} , we can enlarge the game tree by adding o -histories for all such agents right after the immediate predecessor $h' \not\subseteq \bar{h}$ of any terminal history \bar{h} .⁷⁴ All newly created terminal histories correspond to the same allocation μ as the original \bar{h} . Such enlargement of the game has no impact on its outcome, efficiency, or obvious strategy-proofness properties.

We extend Step 4—the construction of the coding algorithm—as follows. Given the mech-

⁷³Our analysis (as well as analysis of BG on which we build) is carried out recursively, within endogenous rounds, starting at the root of the game. The analysis of each round does not rely on any aspects of the continuation game beyond the current round other than its efficiency and obvious strategy-proofness. Having fewer objects than agents overall makes no difference for the analysis of any of these rounds.

⁷⁴At these o -histories, we can, e.g., give each of these agents two actions that lead to identical continuation games, at the end of which all o -agents receive the outside option and all other agents receive the object they clinched before the first added o -action. (This is always possible if $|\mathcal{X}| \geq 2$; if $|\mathcal{X}| \in \{0, 1\}$ the result we are proving becomes straightforward).

anism (Γ, S) and the fixed preference profile of the agents, the role assignment determines the profile c^o of all possible o -actions. We construct our new coding in two steps:

- First we apply our base coding algorithm to $(\Gamma(c^o), S(c^o))$ and obtain a coding that partially orders all agents except for o -agents.
- Second, we extend this partial order to an order on all agents by ranking the o -agents strictly below the other agents. We rank the o -agents strictly in the order in which they first moved.⁷⁵

Given this coding construction, the analogue of Lemma 3 (and hence our Step 5) is an immediate corollary of our base Lemma 3.⁷⁶

It remains to extend Step 6, that is to break the ties in the coding while preserving the injectivity of the mapping from role assignment to resulting complete ordering of agents. As a preparation to extend Step 6, we define the following partition of the set of role assignments Σ : two role assignments σ and σ' are in the same element of the partition if and only if the resulting class of o -agents is the same under σ and σ' and the order in which o agents are called to move (for their first move) is also the same under σ and σ' . This is well defined, as for this part of the base proof we work with a fixed preference profile, and thus for a given role assignment, the game path is uniquely determined.

Fix an element Π of the above partition of role assignments. As before, each role assignment $\sigma \in \Pi$ determines the profile c_σ^o and hence the game $\Gamma(c_\sigma^o)$ and the profile $S(c_\sigma^o)$ of greedy strategies of the players in this game. Importantly, c_σ^o is the same for all $\sigma \in \Pi$. Hence, if we restrict Lemmas 4 and 5 to role assignments in Π , then the claims of these Lemmas remain valid.⁷⁷

The extension of Lemma 4 tells us that the mapping from role assignments in Π to codings is injective. The extension of Lemma 5 allows us to break the ties in the coding of $\sigma \in \Pi$ by using the same procedure as in the base proof provided we only check for arms composed of role assignments from Π . In particular, as before we start with ties closest to the top of the partial ordering and proceed by breaking one tie at a time. As our construction recognizes

⁷⁵In a millipede (Γ, S) , in which greedy strategy directs an agent to clinch their best possible payoff the first time a clinching action is offered, each o -agent actually moves once. Our argument however does not rely on this feature.

⁷⁶Indeed, Lemma 3 and its proof are restricted to a single role assignment σ and hence to obtain its analogue it is enough to analyze $(\Gamma(c^o), S(c^o))$ for c^o generated by σ .

⁷⁷As with Lemma 3, the proofs of Lemmas 4 and 5 analyze only games created by the role assignments listed in the (respective) statements of these lemmas. Thus, the proofs of these lemmas only require us to work with $(\Gamma(c^o), S(c^o))$ for c^o equal to the c_σ^o common to all $\sigma \in \Pi$. A final additional step needed in the proof of the analogue of Lemma 4 is to notice that the path through the game generated by any role assignment, including all $\sigma \in \Pi$, always has exactly $|\mathcal{N}| - |\mathcal{X}|$ o -agents.

fewer arms, the arms we recognize allows us to break fewer ties than we might have been able to break if we could rely on the full set of arms; as in the base construction, we break the remaining ties arbitrarily.⁷⁸ We first break all the ties in codings of role assignments from Π , before proceeding with other elements of the partition.⁷⁹

For each element Π of the partition, our base argument shows that the mapping from Π to orderings we constructed above is an injection. We claim that the union of these injections over all elements of the partition is an injection from the entire set of role assignments to orderings. Indeed, take any two role assignments, σ and σ' . If they belong to the same element of the partition, then we have just argued that they are mapped to different orderings. If they belong to different elements of the partition, then, by construction of the partition, the \mathcal{o} -agents for σ and σ' are either different or move in a different order; as these \mathcal{o} -agents are listed in order of their moves at the end of the orderings, the two orderings are hence also different.⁸⁰ As the set of all role assignments and the set of all orderings have the same cardinality ($|\mathcal{N}|!$) the injection from the former to the latter is indeed a bijection, and thus Theorem 1 continues to hold in this more general environment.

⁷⁸While one might be concerned this might break injectivity, we are about to show that it does not.

⁷⁹As there are no interactions between tie breaking across different elements of the partition, the order in which we handle partition elements, or whether we break only some ties in one partition element, before moving to the next one (to eventually come back to the first) does not matter.

⁸⁰This inference relies on the fact that the path through the game generated by any role assignment, including σ and σ' , always has exactly $|\mathcal{N}| - |\mathcal{X}|$ \mathcal{o} -agents.

Supplementary Appendix: Omitted Proofs (For Online Publication)

Proof of Lemma 7

We first show the result for BG lurkers.⁸¹ BG Lemma E.18 shows that the agent who moves at h^N is not a BG lurker at $h^{n'}$ for $n' \leq N$.⁸² Thus, we must show that ℓ does not move at $h^{n'}$ for any $n' = n, n+1, \dots, N-1$. It is sufficient to show the result for the earliest history h^n such that $\ell \in \mathcal{L}^{BG}(h^n)$. Since ℓ first becomes a BG lurker at h^n , we have that ℓ moves at h^{n-1} and: $P_\ell(h^n) \neq G_\ell(h^n)$, $x_\ell \in P_\ell(h^{n-1})$, and $C_\ell^\subseteq(h^{n-1}) = P_\ell(h^{n-1}) \setminus \{x_\ell\}$. Greedy strategies imply that all types \succ_ℓ that reach h^n , are such that $Top(\succ_\ell, P_\ell(h^n)) = x_\ell$. Since no agent moves twice in a row, ℓ does not move at h^n . Assume that ℓ moves at $h^{n'}$ for some $n' \in \{n+1, \dots, N-1\}$, and let $h^{n'}$ be the earliest such history where ℓ moves. If $x_\ell \in C_\ell(h^{n'})$, then by greedy strategies, all types of agent ℓ that reach $h^{n'}$ clinch x_ℓ , and thus there is no passing action (it can be pruned) which is a contradiction. If $x_\ell \notin P_\ell(h^{n'})$, we have:⁸³

$$P_\ell(h^{n'}) = P_\ell(h^{n'}) \setminus \{x_\ell\} \subseteq P_\ell(h^{n-1}) \setminus \{x_\ell\} = C_\ell^\subseteq(h^{n-1}) \subseteq C_\ell^\subseteq(h^{n'}) \subseteq C_\ell(h^{n'}).$$

In words, since x_ℓ is previously unclinchable and becomes impossible at $h^{n'}$, property (M3) of millipede games implies that the mechanism must offer ℓ at least a clinching action for each object that was previously clinchable, and since ℓ was already a lurker, this includes everything that was previously possible (except for x_ℓ , which is no longer possible). The above implies $P_\ell(h^{n'}) \subseteq C_\ell(h^{n'})$, i.e., everything possible at $h^{n'}$ is also clinchable at $h^{n'}$. By property 4, there is no passing action, again a contradiction. The only remaining possibility is that $x_\ell \in P_\ell(h^{n'}) \setminus C_\ell(h^{n'})$. By the structure of millipede games, x_ℓ can only be possible following the unique passing action at $h^{n'}$. Since $Top(\succ_\ell, P_\ell(h^{n'})) = x_\ell$ for all types of agent ℓ that reach $h^{n'}$, all types of agent ℓ who reach $h^{n'}$ must pass at $h^{n'}$, and since the game is pruned, there are no clinching actions at $h^{n'}$. But then, $h^{n'}$ is a trivial move with $|A(h)| = 1$, which is a contradiction. Thus, for any $\ell \in \mathcal{L}^{BG}(h^n)$, ℓ does not move at $h^{n'}$ for any $n' \in \{n, n+1, \dots, N\}$.

Because $\mathcal{L}(h) \subseteq \mathcal{L}^{BG}(h)$ at any h , that the same result holds for lurkers is immediate. ■

⁸¹This part of the lemma can be inferred from the analysis in BG; for completeness, we provide a proof.

⁸²To be precise, BG Lemma E.18 says if $\mathcal{L}^{BG}(h) \neq \emptyset$, then the agent t who moves at the maximal superhistory of the form $h' = (h, a^*, \dots, a^*)$ is not a BG lurker at h . It is obvious that t is also not a BG lurker at any \tilde{h} such that $\mathcal{L}^{BG}(\tilde{h}) = \emptyset$ (because there are no BG lurkers at \tilde{h}).

⁸³The first equality follows because $x_\ell \notin P_\ell(h^{n'})$, the first inclusion because the set of possible objects only shrinks as we move along the game path, the second equality because ℓ is a BG lurker for x_ℓ at h^n , the second inclusion because the set of previously clinchable objects only grows as we move along the game path, and the last inclusion by property (M3) of millipede games.

Proof of Lemma 8

We first show the result for BG lurkers.⁸⁴ For $n = 1$, $\mathcal{L}^{BG}(h^{n-1}) = \mathcal{L}^{BG}(h^0) = \emptyset$, and so the statement is obvious. So, assume it holds for all $n' = 1, \dots, n-1$. Take some $\ell \in \mathcal{L}^{BG}(h^{n-1})$ (if there is no such ℓ and so $\mathcal{L}^{BG}(h^{n-1}) = \emptyset$, then the result is obvious). With slight abuse of notation, we interpret ℓ to mean that agent ℓ is the ℓ^{th} oldest BG lurker, and let x_ℓ be the object that ℓ BG lurks. We will show that conditions (i)-(iii) in the definition of a BG-lurker hold at h^n . Notice that for conditions (ii) and (iii), they refer to the most recent move of ℓ strictly prior to h^n . Since $\ell \in \mathcal{L}^{BG}(h^{n-1})$, conditions (ii) and (iii) held at ℓ 's most recent move strictly prior to h^{n-1} , and by Lemma 7, ℓ does not move again at any $h^{n'}$ after becoming a BG-lurker, so conditions (ii) and (iii) continue to hold at h^n . Thus, we must show that condition (i) continues to hold at h^n .

Consider $n < N$. If $P_\ell(h^n) = G_\ell(h^n)$ (i.e., condition (i) fails), then properties 3 and 4 imply there is no passing action at h^n (i.e, h^n is a terminating history), which is a contradiction. Therefore, $P_\ell(h^n) \neq G_\ell(h^n)$, and condition (i) holds.

Now consider $n = N$. Since $\ell \in \mathcal{L}^{BG}(h^{N-1})$, we have $P_\ell(h^{N-1}) \neq G_\ell(h^{N-1})$. We want to show that $P_\ell(h^N) \neq G_\ell(h^N)$. So, assume not, i.e., $P_\ell(h^N) = G_\ell(h^N)$. We start with the following claim.

Claim. Let $n < N$ and consider a clinching action $a_x \in A(h^n)$ that clinches object x for the agent i that moves at h^n . Then, (i) x is not BG-lurked at h^n and (ii) at (h^n, a_x) , each BG lurker $\ell' \in \mathcal{L}^{BG}(h^n)$ has clinched their lurked object.

Proof. If $\mathcal{L}^{BG}(h^n) = \emptyset$, then part (i) is obvious and (ii) is vacuous. So, assume $\mathcal{L}^{BG}(h^n) \neq \emptyset$. By Lemma 7, $i \notin \mathcal{L}^{BG}(h^n)$. If $x \in \mathcal{X}^{\mathcal{L}, BG}(h^n)$, by BG Lemma E.14, $P_i(h^n) = G_i(h^n)$, and by properties 3 and 4, there is no passing action at h^n , a contradiction. This establishes part (i). Given part (i), BG Lemma E.17 shows that each lurker ℓ' at h^n receives their lurked object at all terminal $\bar{h} \supseteq (h^n, a_x)$, which establishes part (ii). ■

Since $P_\ell(h^{N-1}) \neq G_\ell(h^{N-1})$, it must be that $x_\ell \notin G_\ell(h^{N-1})$.⁸⁵ Since we've assumed $P_\ell(h^N) = G_\ell(h^N)$, this implies that $x_\ell \notin P_\ell(h^N)$, and so by obvious dominance, $G_\ell(h^N) = \mathcal{X} \setminus \{x_1, \dots, x_\ell\}$, where $x_1, \dots, x_{\ell-1}$ are the BG-lurked objects older than x_ℓ .⁸⁶

⁸⁴This part of the lemma can be inferred from the analysis in BG; for completeness, we provide a proof.

⁸⁵This follows because if x_ℓ is guaranteeable at h^N , then ℓ has effectively already clinched it prior to h^N , which means she was not a BG lurker. More formally, assume $x_\ell \in G_\ell(h^N)$, and let h^* be the earliest history at which ℓ is a BG-lurker for x_ℓ ; since $\ell \in \mathcal{L}^{BG}(h^{N-1})$, we have $h^* \subsetneq h^N$. By definition of a BG lurker and greedy strategies, $Top(>_\ell, P_\ell(h^*)) = x_\ell$ for any type that reaches h^* . By the claim just proven, ℓ has clinched x_ℓ following any clinching action at any (h^n, a_x) , where $h^* \subseteq h^n \subseteq h^{N-1}$. This means that if $x_\ell \in G_\ell(h^N)$, then x_ℓ is already guaranteeable at h^* : $x_\ell \in G_\ell(h^*)$. Further, no type $>_\ell$ that reaches h^* will follow any strategy for which anything but x_ℓ is a possible outcome, and so by pruning, $P_\ell(h^*) = G_\ell(h^*)$. But this contradicts that ℓ is a lurker at h^* .

⁸⁶BG Lemma E.16 implies that $P_\ell(h^{N-1}) = \mathcal{X} \setminus \{x_1, \dots, x_{\ell-1}\}$ (note that $x_\ell \in P_\ell(h^{N-1})$), and also

Now, at h^N , the agent who moves, t (the terminator, who is not a BG lurker by BG Lemma E.18) has only clinching actions. Since $G_\ell(h^N) = \mathcal{X} \setminus \{x_1, \dots, x_\ell\}$, none of these objects can be clinchable for t at h^N . In particular, $C_t(h^N) \subseteq \{x_1, \dots, x_\ell\}$. We further claim that $|C_t(h^N)| > 1$. To see this, presume that $|C_t(h^N)| = 1$, and let x' be the unique object that is clinchable at h^N for t . Let history $h^{n'}$ be the most recent history at which t moves strictly prior to h^N . Such a history exists because by BG Lemma E.18, $C_t^\subseteq(h^N) = \mathcal{X}$. Additionally, the fact that no agent moves twice in a row implies that $n' \leq N - 2$. Note also that $C_t^\subseteq(h^{n'}) \neq \mathcal{X}$ (as otherwise, by greedy strategies, all types of agent t will have clinched an object by $h^{n'}$, and thus by pruning, the passing path will not continue to h^N , which is a contradiction). Therefore, it must be $C_t^\subseteq(h^{n'}) = \mathcal{X} \setminus \{x'\}$. Greedy strategies imply that all types of agent t that reach $h^{n'+1}$ rank object x' first (over all objects). Consider the histories $h^{n'+1}, \dots, h^{N-1}$, and notice that at each such history, it is an agent $j \neq t$ that moves.

There are two cases: (i) x' is not clinchable at any $\tilde{h} \in \{h^{n'+1}, \dots, h^{N-1}\}$ or (ii) x' is clinchable at at least one such $\tilde{h} \in \{h^{n'+1}, \dots, h^{N-1}\}$. In case (i), notice that for all $\tilde{h} \in \{h^{n'+1}, \dots, h^{N-1}\}$, following any clinching action (\tilde{h}, a_z) , it is some $z \neq x'$ that is clinched. Therefore, obvious dominance requires that agent t receive x' at any terminal history $\bar{h} \supseteq (\tilde{h}, a_z)$. Further, t receives x' at any terminal $\bar{h} \supseteq h^N$ as well, which implies that t receives x' at all terminal $\bar{h} \supseteq h^{n'+1} = (h^{n'}, a^*)$. This means that t has clinched x' at $h^{n'+1}$ already, which contradicts that there is a passing action at $h^{n'}$. In case (ii), since x' is clinchable at \tilde{h} for an agent other than t , and all other objects $\mathcal{X} \setminus \{x'\}$ were previously clinchable for t , obvious dominance requires that following this clinching action, agent t must receive $Top(>_t, \mathcal{X} \setminus \{x'\})$; in particular, every object in $\mathcal{X} \setminus \{x'\}$ must remain possible at $h^{n'+1}$ for t . Object x' itself is also still possible, so we have $P_t(h^{n'+1}) = \mathcal{X}$. This implies that conditions (ii) and (iii) in the definition of a BG lurker for x' hold at $h^{n'+1}$.⁸⁷ If $P_t(h^{n'+1}) = G_t(h^{n'+1})$, then by properties 3 and 4 of millipede games, there is no passing action at $h^{n'+1}$, and as $n' \leq N - 2$, this a contradiction. Thus, $P_t(h^{n'+1}) \neq G_t(h^{n'+1})$, and condition (i) of a BG lurker holds as well. Thus, t is a BG lurker at $h^{n'+1}$. However, this is a contradiction to BG Lemma E.18 (cf. footnote 82).

Thus, we may assume that $|C_t(h^N)| > 1$. Since we have also shown that $C_t(h^N) \subseteq$

$C_\ell^\subseteq(h^{N-1}) = P_\ell(h_\ell^{N-1}) \setminus \{x_\ell\}$. If there is some $x \in \mathcal{X} \setminus \{x_1, \dots, x_\ell\}$ such that $x \notin G_\ell(h^N)$, then the type that ranks $>_\ell: x_\ell, x, \dots$ has no obviously dominant strategy: because x_ℓ is possible at h^{N-1} , but has never been clinchable, any obviously dominant strategy requires ℓ to pass at all $h' \subseteq h^{N-1}$ where she is called to move. Since $x_\ell \notin P_\ell(h^N)$, and $x \notin G_\ell(h^N)$, any such strategy has a worst-case outcome that is strictly worse than x . As x was clinchable for ℓ at some $\tilde{h} \not\subseteq h^{N-1}$, no such strategy can be obviously dominant.

⁸⁷Notice that the conditions refer to t 's most recent move strictly prior to $h^{n'+1}$, which, by definition is $h^{n'}$. Since $P_t(h^{n'+1}) = \mathcal{X}$, we have $P_t(h^{n'}) = \mathcal{X}$ as well, and it is obvious that $x' \in P_t(h^{n'})$, which is condition (ii). Condition (iii) says $C_t^\subseteq(h^{n'}) = P_t(h^{n'}) \setminus \{x'\}$. We showed earlier in the paragraph that $C_t^\subseteq(h^{n'}) = \mathcal{X} \setminus \{x'\}$ and so condition (iii) holds as well.

$\{x_1, \dots, x_\ell\}$, at least two objects in $\{x_1, \dots, x_\ell\}$ must be clinchable at h^N (this also implies that $\ell \geq 2$ in the present context). Choose any two of these objects, and denote them $x_{\ell'}$ and $x_{\ell''}$, where without loss of generality, $1 \leq \ell' < \ell'' \leq \ell$. These are both BG-lurked objects, and their BG lurkers are agents ℓ' , ℓ'' . If t clinches $x_{\ell'}$, at $(h^N, a_{x_{\ell'}})$, agent ℓ' must be able to guarantee all objects in $\mathcal{X} \setminus \{x_1, \dots, x_{\ell'}\}$, and agent ℓ must be able to guarantee all objects in $\mathcal{X} \setminus \{x_1, \dots, x_\ell\}$, which is a contradiction.⁸⁸

Therefore, $P_\ell(h^N) \neq G_\ell(h^N)$, condition (i) holds, and $\ell \in \mathcal{L}^{BG}(h^N)$.

Thus, we have shown that $\mathcal{L}^{BG}(h^{n-1}) \subseteq \mathcal{L}^{BG}(h^n)$ for all $n = 1, \dots, N$. To see that the same is true for lurkers, take some $\ell \in \mathcal{L}(h^{n-1})$. Notice that $\mathcal{L}(h^{n-1}) \subseteq \mathcal{L}^{BG}(h^{n-1})$, and so $\ell \in \mathcal{L}^{BG}(h^n)$, as we just showed above. That is, ℓ satisfies conditions (i)-(iii) in the definition. So all that is left to show is that condition (iv) holds at h^n . But, notice that similar to conditions (ii) and (iii), condition (iv) refers only to ℓ 's most recent move strictly prior to h^n . As $\ell \in \mathcal{L}(h^{n-1})$, condition (iv) holds at h^{n-1} , and by Lemma 7, ℓ does not move at any $h^{n'}$ after becoming a lurker for the first time. Thus, condition (iv) continues to hold at h^n , and $\ell \in \mathcal{L}(h^n)$. ■

Proof of Lemma 9

By BG Lemma E.14, we have $P_j(h) = G_j(h)$. By properties 3 and 4, there is no passing action at h , and thus, h is a terminating history, which establishes the first statement.

We must also show that j moves at h . Assume some $k \neq j$ moved at h , and thus k is the terminator. Let $\tilde{h} \subseteq h$ be the smallest subhistory of h such that x is BG-lurked at \tilde{h} (i.e., it is not BG-lurked at any predecessor of \tilde{h}). We claim that $x \in P_j(\tilde{h})$. To see this, assume $x \notin P_j(\tilde{h})$. Since $x \in C_j^\subseteq(h)$, we must have $x \in C_j(\hat{h})$ for some $\hat{h} \not\subseteq \tilde{h} \subseteq h$. This implies that both j was active strictly prior to \tilde{h} , and also $\tilde{h} = h$.⁸⁹ This further implies that k is not active strictly before h .⁹⁰ By BG Lemma E.18, $C_k^\subseteq(h) = \mathcal{X}$, and since h is the first time k is active, $C_k(h) = \mathcal{X}$, i.e., k can clinch every object at h . But, this means that at h , no object can be guaranteeable for j , which contradicts $P_j(h) = G_j(h)$. Therefore, $x \in P_j(\tilde{h})$.

⁸⁸This follows an argument similar to footnote 86: when $x_{\ell'}$ becomes impossible for ℓ' at $(h^N, a_{x_{\ell'}})$, obvious dominance requires that ℓ' be able to guarantee everything that was previously clinchable. By BG Lemma E.16 and the fact that ℓ' is a BG lurker for $x_{\ell'}$, this is all objects in $\mathcal{X} \setminus \{x_1, \dots, x_{\ell'}\}$. (Note that it is impossible for all objects to be BG-lurked, because as BG Lemma E.18 shows, the terminator is not a BG lurker. Therefore, $\{x_1, \dots, x_{\ell'}\} \not\subseteq \{x_1, \dots, x_\ell\} \not\subseteq \mathcal{X}$, and so $(\mathcal{X} \setminus \{x_1, \dots, x_{\ell'}\}) \cap (\mathcal{X} \setminus \{x_1, \dots, x_\ell\}) \neq \emptyset$.)

⁸⁹As j is a BG non-lurker at h , she is also a BG non-lurker at \hat{h} and \tilde{h} , by Lemma 8. Similar to the paragraph above, by BG Lemma E.14, $P_j(\tilde{h}) = G_j(\tilde{h})$, and so \tilde{h} is the terminating history, i.e., $\tilde{h} = h$.

⁹⁰Let h' be the immediate predecessor of h , so $h = (h', a^*)$, and let i be the agent who BG-lurks x . Since x became BG-lurked only at the terminating history h , agents i and j are active BG non-lurkers at h' , and by BG Lemma E.19, there can be at most two active BG non-lurkers at any history. Therefore, at h' , k is either inactive, or a BG lurker. BG Lemma E.18 shows that the terminator is not a BG lurker at any $\tilde{h} \subseteq h$ (see footnote 82), and thus k must become active only at h .

BG Lemma E.18 then implies that j is the terminator, and thus j moves at h . ■

Proof of Lemma 10

The statement is obvious for $n = 0$. So, presume it holds for $1, \dots, n-1$, where $n < N$. We show that $\mathcal{L}(h^n) = \mathcal{L}^{BG}(h^n)$.

Notice that we can write $\mathcal{L}^{BG}(h^{n-1}) = \mathcal{L}(h^{n-1}) \subseteq \mathcal{L}(h^n) \subseteq \mathcal{L}^{BG}(h^n)$, where the first equality is the inductive hypothesis, the first inclusion is Lemma 8, and the second inclusion is because at any history h , any lurker is also a BG lurker. Thus, if $\mathcal{L}^{BG}(h^{n-1}) = \mathcal{L}^{BG}(h^n)$, then the inclusions must be equalities, and we are done.

So, consider the case that $\mathcal{L}^{BG}(h^{n-1}) \neq \mathcal{L}^{BG}(h^n)$, and assume that $\mathcal{L}(h^n) \neq \mathcal{L}^{BG}(h^n)$. As $\mathcal{L}^{BG}(h^{n-1}) \subseteq \mathcal{L}^{BG}(h^n)$ and $h^n = (h^{n-1}, a^*)$, it must be that there is some agent i who first becomes a BG lurker at h^n , and i must move at h^{n-1} .⁹¹ In particular, $\mathcal{L}^{BG}(h^n) = \mathcal{L}^{BG}(h^{n-1}) \cup \{i\}$. As $\mathcal{L}^{BG}(h^{n-1}) = \mathcal{L}(h^{n-1}) \subseteq \mathcal{L}(h^n) \subseteq \mathcal{L}^{BG}(h^n) = \mathcal{L}^{BG}(h^{n-1}) \cup \{i\}$, if $\mathcal{L}(h^n) \neq \mathcal{L}^{BG}(h^n)$, then it must be that $i \notin \mathcal{L}(h^n)$. Since $i \in \mathcal{L}^{BG}(h^n)$, conditions (i)-(iii) hold at h^n for i , and thus, it must be condition (iv) that is violated, i.e., at h^{n-1} , there is some active non-lurker j at h^{n-1} such that $x \in C_j^c(h^{n-1})$ (where x is the object that i BG-lurks). Since $\mathcal{L}^{BG}(h^{n-1}) = \mathcal{L}(h^{n-1})$, agent j is also a BG nonlurker at h^{n-1} , and thus, by Lemma 9, there is no passing action at h^n , a contradiction. ■

Proof of Lemma 11

As above, let $h^n = (h_\emptyset, a^*, \dots, a^*)$, where h^N is the terminating history. Lemma 10 shows that at h^n for $n < N$, we have $\mathcal{L}^{BG}(h^n) = \mathcal{L}(h^n)$. The only remaining history to consider is the terminating history h^N . Note that $\mathcal{L}^{BG}(h^{N-1}) = \mathcal{L}(h^{N-1}) \subseteq \mathcal{L}(h^N) \subseteq \mathcal{L}^{BG}(h^N)$, where the first equality follows from Lemma 10, the first inclusion by Lemma 8, and the second inclusion by the fact that any lurker is a BG lurker. If $\mathcal{L}^{BG}(h^{N-1}) = \mathcal{L}^{BG}(h^N)$, then the inclusions must be equalities, and so all BG lurkers are lurkers. If $\mathcal{L}^{BG}(h^{N-1}) \neq \mathcal{L}^{BG}(h^N)$, it must be that $\mathcal{L}^{BG}(h^N) = \mathcal{L}^{BG}(h^{N-1}) \cup \{i\}$ for some i who first becomes a BG lurker at h^N and also i moves at h^{N-1} (see footnote 91). So, we have $\mathcal{L}^{BG}(h^{N-1}) = \mathcal{L}(h^{N-1}) \subseteq \mathcal{L}(h^N) \subseteq \mathcal{L}^{BG}(h^N) = \mathcal{L}^{BG}(h^{N-1}) \cup \{i\}$. If $i \in \mathcal{L}(h^N)$, then $\mathcal{L}(h^N) = \mathcal{L}^{BG}(h^N)$. If $i \notin \mathcal{L}(h^N)$, then $\mathcal{L}(h^N) = \mathcal{L}^{BG}(h^N) \setminus \{i\}$. In either case, there is at most one BG lurker that is not a lurker. Notice also that we have shown that $\mathcal{L}(h^n) \neq \mathcal{L}^{BG}(h^n)$ only if $n = N$, i.e., this occurs only

⁹¹For any j that does not move at h^{n-1} , then her most recent move was at some strict subhistory $h' \subsetneq h^{n-1}$. Conditions (ii)-(iii) are thus the same when checking whether j is a BG lurker at h^{n-1} and h^n , so if j is not a BG lurker at h^{n-1} , it must be condition (i) that is violated, i.e., $P_j(h^{n-1}) = G_j(h^{n-1})$. But, by properties 3 and 4, this means there is no passing action at h^{n-1} , a contradiction.

at the terminating history. Further, the above argument establishes that when this occurs, it is the youngest BG lurker (agent i) who is not a lurker, and i moves at h^{N-1} .

For statement (iv) and (v), since h is the terminating history, the agent who moves at h is the terminator t , who is not a BG lurker (BG Lemma E.18), and thus is also not a lurker. To see that the active agents are $\mathcal{L}^{BG}(h) \cup \{t\}$, note that when i became a BG lurker but not a lurker at $h^N (= h)$, it must be because there was some other j who was an active non-lurker at h^{N-1} and $x \in C_j^{\subseteq}(h^{N-1})$, where x is the object that i BG-lurks. At h^{N-1} , $\mathcal{L}^{BG}(h^{N-1}) = \mathcal{L}(h^{N-1})$, which also means that all non-lurkers are also BG non-lurkers at h^{N-1} , and so j is a BG non-lurker at h^{N-1} . Since there can be at most two active BG non-lurkers at any history (BG Lemma E.19), at h^{N-1} , the active agents are $\mathcal{L}^{BG}(h^{N-1}) \cup \{i, j\}$. Agent i moves at h^{N-1} , and by Lemma 9, j is the terminator, i.e., $j = t$ and so j moves at h^N . Therefore, at h^N , i becomes a BG lurker and no new agent enters the game, so the active agents at h^N are $\mathcal{L}^{BG}(h^N) \cup \{t\} = \mathcal{L}(h^N) \cup \{i, t\}$. ■

Proof of Lemma 13

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 $x \in \mathcal{X}^{\mathcal{L}, BG}(h)$. By Lemma 11, there is only one such object, and it is $x = x_{\lambda^{BG}(h)}$, the youngest BG-lurked object at h . Further, this condition only obtains when h is a terminating history, and the active agents at h are $\ell_1, \dots, \ell_{\lambda^{BG}(h)}, t$ where: $\ell_1, \dots, \ell_{\lambda^{BG}(h)-1}$ are both lurkers and BG lurkers, $\ell_{\lambda^{BG}(h)}$ is a BG lurker but not a lurker, and t is the terminator (and neither a lurker nor a BG lurker). By BG Lemma E.16, $x_{\lambda^{BG}(h)} \in P_{\ell'}(h)$ for all $\ell' \in \{\ell_1, \dots, \ell_{\lambda^{BG}(h)}\}$, while by BG Lemma E.18, $x_{\lambda^{BG}(h)} \in C_t^{\subseteq}(h)$.

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

Proof of Lemma 14

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 9. 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 $x_\ell \notin C_i^{\subseteq}(h)$, 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 11, h must be a terminating history, agent i moves at h' and passes, and becomes a BG lurker at h . Note that since x_ℓ is lurked at h , it is also BG-lurked at h . If $x_\ell \in C_i^{\subseteq}(h)$, then, since i does not move at h , we have $x_\ell \in C_i^{\subseteq}(h')$ as well. Further, x_ℓ cannot be the object i BG lurks at h (because it has been previously clinchable for her), and so x_ℓ must be BG-lurked at h' by some other agent. But then, at h' , i is not a BG lurker, and

$x_\ell \in C_i^c(h')$ for some BG-lurked object x_ℓ . Thus, by Lemma 9, h' is a terminating history, which is a contradiction. ■

Proof of Lemma 15

Since there is a lurked object at h , there is also a BG-lurked object at h . Parts (ii) and (iii) then follow immediately from the corresponding statements in BG Lemmas E.18 and E.20. BG Lemma E.18 part (i) says that agent t is not a BG lurker at h , and thus, agent t is not a lurker either, which shows part (i). What remains is to show part (iv). For all $h \not\subseteq h'$, any non-lurker is also a BG non-lurker by Lemma 11, 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 11, 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 11, 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. ■

Proof of Lemma 17

At h , j is a non-lurker, but may or may not be a BG lurker. First consider the case that j is a BG lurker at h , and let y be the object that j BG-lurks. By Lemma 11, j is the youngest BG-lurker, and the set of BG-lurked objects is $\mathcal{X}^{\mathcal{L},BG}(h) = \mathcal{X}^{\mathcal{L}}(h) \cup \{y\}$, where $y \notin \mathcal{X}^{\mathcal{L}}(h)$ and y is the youngest BG-lurked object at h . In addition, the set of unlurked objects is $\bar{\mathcal{X}}^{\mathcal{L},BG}(h) = \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{y\}$. Because j BG-lurks y , we have $y = \text{Top}(>_j, \bar{\mathcal{X}}^{\mathcal{L}}(h))$. If $y = \hat{x}$, then BG Lemma E.17 implies that j receives $\text{Top}(>_j, \bar{\mathcal{X}}^{\mathcal{L},BG}(h))$, and further, $y = \hat{x}$ implies $\bar{\mathcal{X}}^{\mathcal{L},BG}(h) = \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\}$, and so the statement holds. If $y \neq \hat{x}$, then BG Lemma E.17 implies that j receives y . Since $y \neq \hat{x}$ and $\hat{x} \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$, we have $y = \text{Top}(>_j, \bar{\mathcal{X}}^{\mathcal{L}}(h)) = \text{Top}(>_j, \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\})$, and again, the statement holds.

Second, consider the case that j is not a BG lurker at h . In this case, we claim that $\mathcal{L}(h) = \mathcal{L}^{BG}(h)$. Indeed, the agent who moves at h , agent i , is not a lurker nor a BG lurker (Lemma 7), and so both i and j are both non-lurkers and BG non-lurkers at h . As there can be at most 2 non-lurkers at any h (Lemma 12) and the same is true for BG non-lurkers (BG Lemma E.19), all other active agents at h are both lurkers and BG lurkers. Similar statements apply to non-lurkers ($\bar{\mathcal{L}}(h) = \bar{\mathcal{L}}^{BG}(h)$), lurked objects ($\mathcal{X}^{\mathcal{L}}(h) = \mathcal{X}^{\mathcal{L},BG}(h)$), and unlurked objects ($\bar{\mathcal{X}}^{\mathcal{L}}(h) = \bar{\mathcal{X}}^{\mathcal{L},BG}(h)$).

We next claim that $P_j(h) \subseteq \bar{\mathcal{X}}^{\mathcal{L}}(h)$. Indeed, if there was some lurked object $x_\ell \in \mathcal{X}^{\mathcal{L}}(h)$ such that $x_\ell \in P_j(h)$, then by Lemma 15, part (iv), j is the terminator. This further implies that the agent i who moves at h is not the terminator, and again by Lemma 15, part (iv), object x (the object i clinches at h) is unlurked at h . This means that $\hat{x} = x$, by definition of \hat{x} . However, because j is the terminator, Lemma 15, part (iii) implies $C_i(h) \cap C_j^c(h) = \emptyset$, and in particular, $\hat{x} \notin C_i(h)$, which is a contradiction. Therefore, $P_j(h) \subseteq \bar{\mathcal{X}}^{\mathcal{L}}(h)$.

Let $y = \text{Top}(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h))$. Consider first the case $y \notin C_j^c(h)$, and note that this implies that $y \neq \hat{x}$. Because j does not move at h , $C_j^c(h) = C_j^{\bar{c}}(h)$ and hence $y \notin C_j^{\bar{c}}(h)$. As $\bar{\mathcal{X}}^{\mathcal{L}}(h) = \bar{\mathcal{X}}^{\mathcal{L},BG}(h)$ and BG Lemma E.11 tells us that $\bar{\mathcal{X}}^{\mathcal{L},BG}(h) \subseteq P_j(h) \cup C_j^{\bar{c}}(h)$, we conclude that $y \in P_j(h)$. Property (M3) of millipede games then implies that j receives y .⁹² Since $y \neq \hat{x}$ and $\hat{x} \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$, we have $y = \text{Top}(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h)) = \text{Top}(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\})$, and the claim of the lemma obtains.

It remains to consider the case $y \in C_j^c(h)$. Let z be the object j receives at \bar{h} , and note that $z \in \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\}$ ($z \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$ follows from $P_j(h) \subseteq \bar{\mathcal{X}}^{\mathcal{L}}(h)$, and $z \neq \hat{x}$ follows because \hat{x} is assigned to another agent at \bar{h} , by Lemma 16). Obvious dominance implies that $z \succeq_j y$; otherwise, type \succ_j 's strategy would not obviously dominate the alternative strategy that clinches y at the $h' \not\subseteq h$ where it was offered to her for clinching. Since $z \in \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\}$ and $z \succeq_j y = \text{Top}(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h))$, we must have $z = y$ and $y = \text{Top}(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\})$, which completes the proof.⁹³ ■

Proof of Lemma 18

We first claim that there exists some $y \in C_j^c(h)$ such that $y \in \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\}$. This follows because by supposition j moves prior to h , and so must be offered some object y for clinching when she moved. Lemma 14 implies $y \notin \mathcal{X}^{\mathcal{L}}(h)$, and by supposition of the current lemma, $\hat{x} \notin C_j^c(h)$. Therefore, we must have $y \in \bar{\mathcal{X}}^{\mathcal{L}}(h) \setminus \{\hat{x}\}$.

Next, we claim that $C_j^c(h) \subseteq G_j(\hat{h})$.⁹⁴ To see this, take an arbitrary $y \in C_j^c(h)$; as we just showed, such y exists. As $\hat{x} \notin C_j^c(h)$, we know that $y \neq \hat{x}$ and hence a type $\succ_j: \hat{x}, y, \dots$ is

⁹²If j does not receive y , then at some point on the path from h to \bar{h} , y became impossible for j before being clinchable (or else by greedy strategies, j would have clinched y , which follows from the definition of y as $\text{Top}(\succ_j, \bar{\mathcal{X}}^{\mathcal{L}}(h))$ and the fact that $y \in P_j(h) \subseteq \bar{\mathcal{X}}^{\mathcal{L}}(h)$). Property (M3) requires that agent j then either receives or is reoffered an opportunity to clinch each object that was previously clinchable, which includes \hat{x} . However, \hat{x} is clinched by another agent before j might be called to move following h (by Lemma 16), and so this is a contradiction. Therefore, y cannot become impossible for j before being clinchable, and as it is her top still-possible choice, when it is offered to her, she will clinch it.

⁹³Notice that the above proof continues to apply when there are no lurked objects at h . In this case, the only active agents at h are i and j , and $\hat{x} = x$. When i clinches \hat{x} such that $\hat{x} \in C_j^c(h)$, then j still receives her top choice among all remaining unlurked objects, which in this case is all objects except for \hat{x} .

⁹⁴Note that $C_j^{\bar{c}}(\hat{h}) = C_j^c(h)$ because j does not move at any h' such that $h \subseteq h' \not\subseteq \hat{h}$.

well defined. As $\hat{x} \notin C_j^c(h)$, Lemma 13 implies that $\hat{x} \in P_j(h)$. Because $\hat{x} \in P_j(h) \setminus C_j^c(h)$, greedy strategies imply that type \succ_j must pass at all $h' \not\subseteq h$ where j moves, and as j does not move between h and \hat{h} , this type reaches \hat{h} . If $y \notin G_j(\hat{h})$, then, for any strategy type \succ_j follows starting from \hat{h} , type \succ_j might end up with an object strictly worse than y (because \hat{x} has been clinched by another agent, and y is not guaranteeable at \hat{h}). However, j could have clinched y at some history $h' \not\subseteq h$, which contradicts that her strategy is obviously dominant. Thus, $y \in G_j(\hat{h})$ and, as y was an arbitrary element of $C_j^c(h)$, we conclude that $G_j(\hat{h}) \supseteq C_j^c(h)$. The procedure from the proof of the millipede theorem of Pycia and Troyan (2023b) allows us to transform the millipede, while maintaining properties 1-5, so that j moves at \hat{h} and any object that is guaranteeable for agent j at history \hat{h} is clinchable at \hat{h} , so that $G_j(\hat{h}) = C_j(\hat{h})$. As we just showed that $G_j(\hat{h}) \supseteq C_j^c(h)$, we conclude that $C_j^c(h) \subseteq C_j(\hat{h})$. At \hat{h} , agent j may also be offered to clinch additional objects, and may or may not have a passing action. ■

Proof of Lemma 19

First, consider the case that there are no lurked objects at h , and so there are also no lurked objects at any h' such that $h_\emptyset^k \subseteq h' \subseteq h$, by Lemma 8. There is then a history h' such that $h_\emptyset^k \subseteq h' \not\subseteq h$ and $y \in C_j(h')$. Indeed, if j was not active until the current round, then the existence of h' follows from the assumption that $y \in C_j^c(h)$. If j was active in a previous round then there are two subcases: if j was offered y in a previous round, then j moves at h_\emptyset^k and $y \in C_j(h_\emptyset^k)$, and we can set $h' = h_\emptyset^k$; if j wasn't offered y in a previous round, then there must be some h' in the current round where she was offered it. Hence, history h' is well defined. Since type \succ_j reaches h , this type must pass at h' . By greedy strategies, there must be some $x \in P_j(h')$ such that $x \succ_j y$. As there are no lurkers (or lurked objects) at h' or h , $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h') = \bar{\mathcal{X}}^{\mathcal{L}}(h)$. Thus, $\bar{x} \succ_j y$.

Now, consider the case where there are lurked objects (and lurkers) at h . By Lemma 15, part (iii), agent j cannot be the terminator. Since j is not the terminator, Lemma 15, part (iv) implies that $P_j(h) \subseteq \bar{\mathcal{X}}^{\mathcal{L}}(h)$. Since i can clinch y at h , there must be some $x \in P_j(h)$ such that $x \succ_j y$, by obvious strategy-proofness; indeed, otherwise, when i clinches y , type \succ_j will end up with an object strictly worse than y , and so the worst-case outcome is strictly worse than the outcome of the alternative strategy of j clinching y at the $h' \not\subseteq h$ where it was clinchable for j . Since $P_j(h) \subseteq \bar{\mathcal{X}}^{\mathcal{L}}(h)$, we have $x \in \bar{\mathcal{X}}^{\mathcal{L}}(h)$, i.e., $\bar{x} \succ_j y$. ■

Proof of Lemma 20

We start with the following lemma.

Lemma 25. *Assume that there exists an arm α 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) $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n-1}}$.⁹⁵ 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) $h_1^* \not\subseteq h_2^* \not\subseteq \dots \not\subseteq h_{n-1}^* \not\subseteq h_n^*$.

(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) $h_{n-1}^* \not\subseteq h_{n+1}^*$ and the number of lurked objects at h_{n+1}^* is at least $P + n - 1$.

Proof of Lemma 25. Part (a). Let $\lambda_{n'}^*$ be the number of lurked objects at history $h_{n'}^*$. Notice that since $\triangleright_{n'}$ has a tie in the $(P + n')^{\text{th}}$ 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 $\triangleright_{n'}$ implies that the lurked objects are $\{x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{\lambda_{n'}^* - P}}\}$,⁹⁶ 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 14). 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 $\triangleright_{n'}$, 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 passing path (Remark 5), 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 $n' = 1, \dots, n-1$, there are $\lambda_{n'}^* = P + n' - 1$ lurked objects at $h_{n'}^*$, which immediately implies that $h_1^* \not\subseteq h_2^* \not\subseteq \dots \not\subseteq h_{n-2}^* \not\subseteq h_{n-1}^*$ (because the number of lurked objects only grows as we go down the initial passing path).

It remains to show that $h_{n-1}^* \not\subseteq h_n^*$. 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 $\{x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{\lambda_n^*}}\}$. If a lurked

⁹⁵Cf. footnote 66.

⁹⁶This 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.

object is clinched at h_n^* , then h_n^* is the terminating history, and there is no passing action at h_n^* (Lemma 14). 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 5), which is a contradiction. Therefore, $h_{n-1}^* \not\subseteq h_n^*$.

Part (c). Part (a) shows this for $n' \leq n-1$. So, we must show $\lambda_n^* = P+n-1$. Notice that $h_{n-1}^* \not\subseteq h_n^*$ 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 $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n-2}}$, and the agents coded in step 1 are $j_1, \dots, j_P, k_1, \dots, k_{n-2}, k_{n-1}$. 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 14). 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 5), 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 $h_{n-1}^* \not\subseteq h_{n+1}^*$, 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 $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{\bar{n}-1}}$, and the agents coded in step 1 are $j_1, \dots, j_P, k_1, \dots, k_{\bar{n}}$. 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 $h_{n-1}^* \not\subseteq h_{n+1}^*$ 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 $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n-2}}$, and the agents coded in step 1 are $j_1, \dots, j_P, k_1, \dots, k_{n-2}, k_{n-1}$. 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 14). 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 5), a contradiction. Therefore, $\lambda_{n+1}^* \geq P+n-1$.

This completes the proof of Lemma 25. ■

Continuing with the proof of Lemma 20, we first show the first statement that at h_{n+1}^* in Γ_{n+1} , there is an agent $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n, i$ such that ℓ is an active non-lurker at h_{n+1}^* that does not move at h_{n+1}^* , and $x_i \in C_\ell^{\mathbb{F}}(h_{n+1}^*)$. By Lemma 25, we have (i) $h_{n-1}^* \not\subseteq h_n^*, h_{n+1}^*$

(ii) $\lambda_n^* = P + n - 1$ and (iii) $\lambda_{n+1}^* \geq P + n - 1$. In particular, the lurked objects at h_n^* are $\{x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n-1}}\}$. 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., $\sigma_n(s) = i$.⁹⁷ Further, we have $h_n^* \not\subseteq h_{n+1}^*$. 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 14), 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 $\{x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{\bar{n}}}\}$, 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 5). Therefore, $h_n^* \not\subseteq h_{n+1}^*$.

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 $h_n^* \not\subseteq h' \subseteq h_{n+1}^*$. But, notice that there is still some role r such that, at h' , r is an active non-lurker, and $x_i \in C_r^{\not\subseteq}(h')$. 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 immediately following clinching phase of the game.

We next claim that in Γ_{n+1} , $\sigma_{n+1}^{-1}(i) \neq s, s'$. To see this, first note that if $\sigma_{n+1}^{-1}(i) = s$, 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 $h_n^* \not\subseteq h_{n+1}^*$. Therefore, $\sigma_{n+1}^{-1}(i) \neq s$. Next, assume that $\sigma_{n+1}(s') = i$. Notice that role s' cannot be the terminator role, by Lemma 15(iii) and the fact that $x_i \in C_s(h_n^*)$ and $x_i \in C_{s'}^{\not\subseteq}(h_n^*)$. 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 $h' \not\subseteq h_n^* \subseteq h_{n+1}^*$, 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, by Lemma 15(ii). 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

⁹⁷Agent 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 $j_1, \dots, j_P, k_1, \dots, k_{n-1}$, 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.

that $\sigma_{n+1}^{-1}(\ell) = s$ or s' . But then, since i is unlurked at h_{n+1}^* , we have that $x_i \in C_\ell^{\bar{\varphi}}(h_{n+1}^*)$, 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 $\sigma_{n+1}^{-1}(\ell) = s$ or s' , and that is still active when j clinches at h_{n+1}^* , which implies that $x_i \in C_\ell^{\bar{\varphi}}(h_{n+1}^*)$, as desired.

Case 2: x_{k_n} is not the next lurked object along the initial passing path. By Lemma 25, 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} .⁹⁸ 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 $\tilde{h} \not\varphi h_n^*$ such that $x_{k_n} \in C_i(\tilde{h})$.

Claim 5. The following are true: (a) $h_{n-1}^* \not\varphi h_{n+1}^* \not\varphi h_n^*$ 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 $h_{n-1}^* \not\varphi h_{n+1}^*$ follows from Lemma 25. So, we must show that $h_{n+1}^* \not\varphi h_n^*$. Towards a contradiction assume that $h_n^* \subseteq h_{n+1}^*$. Since $h_{n-1}^* \not\varphi h_n^*$, we have $h_{n-1}^* \not\varphi h_n^* \subseteq h_{n+1}^*$. Lemma 25 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 impossible for k_n to be ranked $(P + n)^{th}$ strictly, without ties, in \succ_{n+1} ,⁹⁹ which is a contradiction. Thus, we have shown that $h_{n+1}^* \not\varphi h_n^*$, which is part (a).

⁹⁸Note that this does not necessarily mean that the object clinched at h_n^* is x_i or x_{k_n} .

⁹⁹Note that x_{k_n} cannot be the next lurked object, so, there must be no newly lurked objects at h_{n+1}^* (Remark 5). 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.

Part (b). Part (a) plus Lemma 25 implies that $\lambda_{n+1}^* = P + n - 1$. Additionally, $h_{n+1}^* \not\subseteq h_n^*$ 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 25, 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 $h_{n-1}^* \not\subseteq h_{n+1}^*$ 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 $\sigma_{n+1}^{-1}(i) = r'$. By construction, $x_i \in C_r(h_{n-1}^*)$, where $x_i \in C_{r'}(\tilde{h})$ for some $\tilde{h} \not\subseteq h_{n-1}^*$. This implies that role r' cannot be the terminator role, by Lemma 15(iii), and the fact that $x_i \in C_r(h_{n-1}^*)$. Since role r' is not the terminator role, only unlurked objects are possible for role r' , by Lemma 15(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 $h_{n-1}^* \not\subseteq h_{n+1}^* \not\subseteq h_n^*$ 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 $h_{n-1}^* \not\subseteq h' \not\subseteq \hat{h}$, where \hat{h} is the history at which 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 $n' < n$ or $\sigma_n(s) = j_p$ for some $p = 1, \dots, P$, and k_n is in the lurker role for some object $x' \in \{x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n-1}}\}$. The former fact implies that h_n^* is the terminating history, while the latter fact implies that k_n strictly prefers x' 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' 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 $x_i \in C_\ell^{\not\subseteq}(h_{n+1}^*)$, 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 $\tilde{h} \not\subseteq h_n^*$. 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 $h_{n-1}^* \not\subseteq h_{n+1}^* \subseteq h_n^*$. This combined with Lemma 25 implies that there are $P + n - 1$ lurkers at h_n^* , and the structure of \triangleright_{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 15 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 $x_i \in C_{s'}^{\subseteq}(h_{n-1}^*)$.

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 15(iv)), and, since we know that x_i is i 's favorite unlurked object, if she were in role s' , she would clinch at $\tilde{h} \not\subseteq h_{n+1}^*$, 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 $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n$, 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 $\sigma_{n+1}^{-1}(i) \neq s, s'$. 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 15 and the fact that $x_i \in C_s(h_n^*)$ and $x_i \in C_{s'}^{\neq}(h_n^*)$. 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 $h' \not\subseteq h_n^* \subseteq h_{n+1}^*$, which is a contradiction. Therefore, $\sigma_{n+1}^{-1}(i) \neq s, s'$, and so there must be some $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n$ such that $x_i \in C_\ell(h_{n+1}^*)$, as desired.

It remains to consider $h_{n+1}^* \not\subseteq h_n^*$. 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 active at history h_{n-1}^* . Indeed, notice that because i clinches at h_{n-1}^* in Γ_{n-1} , we have that $x_i \in C_s^{\neq}(h_{n-1}^*)$. This implies that role s is not the terminator role, which follows by Lemma 15 and the fact that $x_i \in C_{s'}(h')$ for some $h' \not\subseteq h_{n-1}^*$. 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 $h' \not\subseteq h_n^*$ 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 $j_1, \dots, j_P, k_1, \dots, k_n, i$. Thus, there must be some $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n, i$ such that ℓ is an active non-lurker that does not move at h_{n+1}^* and $x_i \in C_\ell^{\neq}(h_{n+1}^*)$, 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 $\ell \neq j_1, \dots, j_P, k_1, \dots, k_n, i$ such that ℓ is an active non-lurker that does not move at h_{n+1}^* and $x_i \in C_\ell^{\neq}(h_{n+1}^*)$ in game Γ_{n+1} . Recall that, by Lemma 25, $\lambda_{n+1}^* \geq P+n-1$. 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 $x_i \in C_\ell^{\neq}(h_{n+1}^*)$, 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 at h_{\emptyset}^2 (the beginning of step 2 of the coding algorithm) agent ℓ moves and is offered to clinch 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 at $h_{\mathcal{D}}^2$ agent ℓ was offered to clinch 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 20. ■

Proof of Lemma 21

We start with the following lemma. Let h_A^* denote the history at which the first clinching action is taken in Γ_A , h_B^* denotes the history at which the first clinching action is taken in Γ_B .

Lemma 26. *Consider two games Γ_A and Γ_B , with corresponding role assignment functions σ_A and σ_B , and resulting codings \succ_A and \succ_B . Assume that \succ_A begins as $\{i, j\} \succ_A \dots$, and \succ_B begins as: $j \succ_B i \dots$. 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 $h_B^* \not\subseteq h_A^*$, and there is some agent $k \neq i, j$ 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 26. 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) $x_i \in C_s(h)$ (iii) $x_j \notin C_{s'}^{\subseteq}(h)$ and (iv) $x_i, x_j \notin C_s^{\subseteq}(h)$. 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 $h' \not\subseteq h_A^*$. 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 $h_B^* \not\subseteq h_A^*$.¹⁰⁰ Let $k := \sigma_B(s')$, and notice that, by the assumptions of the lemma, $x_j \notin C_s^{\subseteq}(h)$, and so $h \not\subseteq h_B^*$, and therefore $x_i \in C_{s'}^{\subseteq}(h_B^*)$. 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 $h \not\subseteq h_B^*$ 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), in Γ_B , there is an agent k such that $x_i \in C_k^{\subseteq}(h_B^*)$ and k is not coded in the coding step initiated at h_A^* . Let $h_B^{**} \not\subseteq h_B^*$ be the history at which the next clinching occurs in Γ_B . Since k was offered x_i in the previous coding step,

¹⁰⁰The 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.

but is still active, at h_{\emptyset}^2 , agent k is offered to clinch x_i again (see Remark 3, in particular footnote 49). Thus, x_i cannot be the first lurked object on the initial passing path of the continuation game form (Remark 5), and so since i is ranked immediately following j in \succ_B , 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 $x_i \in C_k^c(h_B^{**})$.

The other possibility is that i clinches at h_A^* in Γ_A , which implies that $x_i \in C_{s'}(h')$ for some $h' \not\subseteq h_A^*$. For j to be ranked first without ties in \succ_B , at h_B^* in Γ_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.¹⁰¹

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.¹⁰² This implies that $h_B^* \not\supseteq h_A^*$, 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.¹⁰³ Recall that i is ranked immediately after j in \succ_B , and because $x_i \in C_\ell^c(h_B^*)$, this is only possible if i ties with ℓ . To see this, assume that i was ranked strictly, i.e., $j \succ_B i \succ_B \dots$. This can only happen if either: (i) x_j is the oldest lurked object and x_i is second-oldest lurked object at h_B^* (ii) the only lurked object at h_B^* is x_j , while x_i is unlurked at h_B^* , i clinches x_i at h_B^* , and x_i has not been previously offered to another active non-lurker or (iii) agent j is the only agent coded in step 1 of Γ_B , and agent i is the first agent coded in step 2, and she is ranked strictly. But, as x_i has been offered to both j and ℓ weakly prior to h_A^* and $h_A^* \not\subseteq h_B^*$, none of these three possibilities can hold, and therefore, i must tie in \succ_B .¹⁰⁴

Case: $\sigma_B(s') = \ell'$ for some $\ell' \neq i, j$. First, notice that $\sigma_B(s) = \ell$ for some $\ell \neq i$. To

¹⁰¹Note that x_j has not been offered to any agent at h , by the presumptions of the lemma.

¹⁰²Notice that in Γ_A , at h_A^* , only two agents are active, i and j . If these agents are in the same roles in Γ_B , then \succ_B would also begin $\{i, j\} \succ_B \dots$.

¹⁰³The 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' .

¹⁰⁴Because x_i has been offered to both j and ℓ weakly prior to h_A^* , it cannot be the first or second lurked object (see Remark 5), which rules out (i). Since $x_i \in C_\ell^c(h_B^*)$, (ii) cannot hold. For (iii), since j is the only agent coded in step 1, but $x_i \in C_\ell^c(h_B^*)$, we have that ℓ begins the round 2 continuation game by being offered x_i (plus possibly other objects) at h_{\emptyset}^2 . Since $x_i \in C_\ell(h_{\emptyset}^2)$, x_i cannot be the first lurked object in round 2 (again by Remark 5), nor can i be ranked first in this round and strictly so (for i to be ranked strictly first in the continuation game, either x_i —which Lemma 3 tells us i gets—has to be the first lurked object, or there are no lurked objects and i clinches x_i while it is unlurked and it has not been offered to the other active non-lurker; neither of these can happen because the continuation game begins with x_i being offered to ℓ).

see this, assume that $\ell = i$. Then, i is in the same role in Γ_A and Γ_B . This implies that $h_B^* \not\preceq h_A^*$, because if h_A^* is reached in Γ_B , i would clinch there, and be ranked above j . But, $h_B^* \not\preceq h_A^*$ 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 $h_B^* \not\preceq h$,¹⁰⁵ which implies that $x_i \in C_{\bar{\ell}}^c(h_B^*)$. But, since x_i has been offered to agent ℓ' , it is impossible for i to be ranked immediately after j in \succ_B without ties. Therefore, i must tie in \succ_B .

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 $h_B^* \not\preceq h_A^*$. For i to be ranked second without ties in this case, there must be two lurked objects at h_B^* ,¹⁰⁶ and x_i must be the second lurked object (after x_j). But, at the history $h'' \not\preceq h_A^*$ 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. This completes the proof of Lemma 26. ■

Continuing with the proof of Lemma 21, first, consider $Q = 1$. Then, all agents $j_1, \dots, j_P, k_1, \dots, k_n, i$ are coded in step 1 of game Γ_n . By Remark 6, $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n-1}}$ all become lurked on the initial passing path of the game form, and further, since $n \geq 2$, we can apply Lemma 20, which completes the proof for this case.

It remains to consider $Q \geq 2$. Since at least $P + 1$ agents are coded in step 1 (see the discussion at the beginning of the proof of Lemma 5), 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 step Q is:

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

Consider the sequence of games $\Gamma_{\bar{n}}, \Gamma_{\bar{n}+1}, \dots, \Gamma_n, \Gamma_{n+1}$. 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 $j_1, \dots, j_P, k_1, \dots, k_{\bar{n}-1}$ 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 initiating history of the continuation game that begins step Q is the also the same in all of these games; label this

¹⁰⁵In case (a), this follows because there are no lurkers at h ; in case (b), it follows from the assumption of the lemma that $x_j \notin C_s^c(h)$.

¹⁰⁶Since 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.

history h_\emptyset^Q . Now, applying the coding algorithm to the sequence of continuation games of $\Gamma_{\bar{n}}, \dots, \Gamma_n, \Gamma_{n+1}$ starting from history h_\emptyset^Q , we get the sub-codings:

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

There are two cases.

Case 1: $\bar{n} < n$. In this case, we can apply Lemma 20 to the game form starting from h_\emptyset^Q to conclude that i must tie in \succ_{n+1} . To see this, simply note that upon reindexing to start from h_\emptyset^Q rather than h_\emptyset , the condition “ $n \geq 2$ ” becomes “ $n \geq \bar{n} + 1$ ”. Then, we have that $x_{k_{\bar{n}}}, \dots, x_{k_{n-1}}$ all become lurked on the initial passing path of the game form starting from h_\emptyset^Q , which follows from Remark 6, $n \geq \bar{n} + 1$, and the fact that i is coded in the initial step of the continuation game of Γ_n starting from h_\emptyset^Q . Thus, all of the conditions of Lemma 20 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:

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

Notice that here, we can no longer apply Lemma 20, since we do not have at least two games in which i ties in the sequence. Our goal is to apply Lemma 26 instead, but to do so, we must show that the conditions (i)-(iv) of Lemma 26 are satisfied at h_\emptyset^Q .

For each coding step $q = 1, \dots, Q$ 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 $h_n^{\emptyset^q}$ 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, $h_n^{\emptyset^1} = h_\emptyset$, 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:

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

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. Assume that $Q > 1$, and that at h_n^{1*} , there is an agent ℓ that is an active non-lurker at h_n^{1*} that does not move at h_n^{1*} and $x_i \in C_\ell^{\neq}(h_n^{1*})$. For all $q < Q$, there is an agent $\ell \neq k_1, \dots, k_{n^q}, i$ such that ℓ is an active nonlurker at h_n^{q*} that does not move at h_n^{q*} , and $x_i \in C_\ell^{\neq}(h_n^{q*})$.

Claim 6 (whose proof can be found starting with the next paragraph, after the conclusion of the proof of this lemma) implies that when we reach step Q in Γ_n , at the initiating history of the continuation game $h_n^{\emptyset Q}$ that begins this step, there is some agent $\ell \neq k_1, \dots, k_{n-1}, i$ such that $x_i \in C_\ell(h_n^{\emptyset Q})$. 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 26 by setting $A = n$, $B = n + 1$, $h = h_n^{\emptyset Q}$, $j = k_n$, and $i = i$ to conclude that i must tie in \succ_{n+1} .¹⁰⁷ This completes the proof of Lemma 21 (modulo the proof of Claim 6 to which we turn next). ■

Proof of Claim 6 (stated above, in the proof of Lemma 21). By supposition, there is an agent ℓ that is an active nonlurker at h_n^{1*} that does not move at h_n^{1*} , and $x_i \in C_\ell^{\neq}(h_n^{1*})$. It is clear that ℓ is not coded in step 1 (since there is no tie in step 1), and so $\ell \neq k_1, \dots, k_{n-1}$. 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 $h_n^{\emptyset 2}$ and produces the subcoding:

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

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 $h_n^{\emptyset 2}$. The result from step 1 implies that at $h_n^{\emptyset 2}$, some agent $\ell \neq k_1, \dots, k_{n^1}, i$ moves and $x_i \in C_\ell(h_n^{\emptyset 2})$.

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 $h_n^{\emptyset 2} = h_{n^1+1}^{\emptyset 2}$;

¹⁰⁷Condition (i) of Lemma 26 is immediate. Condition (ii) was just shown. Condition (iii) holds because, if $x_{k_n} \in C_{k_n}(h_n^{\emptyset Q})$, then k_n would immediately clinch it at $h_n^{\emptyset n}$, and would not tie with i in \succ_n . Condition (iv) is also immediate, as i has not yet been called to move at $h_n^{\emptyset Q}$.

for shorthand, define $h^{\emptyset_2} := h_n^{\emptyset_2} = h_{n^1+1}^{\emptyset_2}$. The second step continuation games of Γ_{n^1+1} and Γ_n both start from h^{\emptyset_2} , and lead to the initial subcodings:

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

Let role s be the role that moves at h^{\emptyset_2} , and role s' be the second role that becomes active on the initial passing path of the game form starting from h^{\emptyset_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^{\emptyset_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 $x_{k_{n^1+1}} \notin C_{k_{n^1+1}}(h_n^{\emptyset_2})$; indeed, if this were true, then k_{n^1+1} would clinch it at $h_{n^1+1}^{\emptyset_2}$ 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 $x_i \in C_{s'}(h_{n^1+1}^{2*})$.¹⁰⁸ It also means that $h_n^{2*} \not\preceq h_{n^1+1}^{2*}$, 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 $x_i \in C_s(h_n^{\emptyset_2})$. Thus, when k_{n^1+1} clinches at h_n^{2*} in Γ_n , we have $x_i \in C_{\ell}^{\neq}(h_n^{2*})$, as desired.

Case 2: $n^2 > n^1 + 1$. Consider games $\Gamma_{n^1+1}, \Gamma_{n^1+2}, \dots, \Gamma_n$ 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, \dots, 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^{\emptyset_2} .

Consider the continuation game starting at h^{\emptyset_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^{\emptyset_2} . Notice that by the structure of \succ_n , the objects $x_{k_{n^1+1}}, \dots, x_{k_{n^2-1}}$ are lurked at h_n^{2*} in Γ_n , while $x_{k_{n^2}}$ is not, i.e., objects $x_{k_{n^1+1}}, \dots, x_{k_{n^2-1}}$ are the first lurked objects (in order) along the initial passing path of the game form, beginning at h^{\emptyset_2} .

¹⁰⁸If 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.

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

$$\begin{aligned} & \{i, k_{n^1+1}\} \succ_{n^1+1} \dots \\ & \vdots \\ & k_{n^1+1} \succ_{n^2} k_{n^1+2} \succ_{n^2} \dots \succ_{n^2} k_{n^2-1} \succ_{n^2} \{i, k_{n^2}\} \succ_{n^2} \dots \\ & k_{n^1+1} \succ_{n^2+1} k_{n^1+2} \succ_{n^2+1} \dots \succ_{n^2+1} k_{n^2} \succ_{n^2+1} \{i, k_{n^2+1}\} \dots \end{aligned}$$

By Lemma 20 applied to the continuation game and subcodings beginning at h^{\emptyset^2} , in Γ_{n^2+1} , at $h_{n^2+1}^{2*}$, there is an agent ℓ such that ℓ is an active non-lurker at $h_{n^2+1}^{2*}$ that does not move at $h_{n^2+1}^{2*}$ and $x_i \in C_\ell^{\neq}(h_{n^2+1}^{2*})$. 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 $h_n^{2*} = h_{n^2+1}^{2*}$, by Lemma 4. This implies that at h_n^{2*} , there is an agent ℓ that is an active non-lurker at h_n^{2*} that does not move at h_n^{2*} and $x_i \in C_\ell^{\neq}(h_n^{2*})$ (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 $x_i \in C_i^{\neq}(h_n^{(Q-1)*})$, 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. 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 Claim 6. ■

Proof of Lemma 22

The proof strategy is to apply Lemma 21. Thus, consider game Γ_n , and, as in the statement of Lemma 21, let Q be the step of the coding algorithm in which i is coded in game Γ_n and h_n^{1*} be the history at which the first clinching action is taken in step $q \in \{1, \dots, Q\}$ of game Γ_n . If $Q = 1$, we can apply Lemma 21 immediately to conclude i ties with some agent k_{n+1} in \succ_{n+1} . So, consider $Q > 1$. To apply Lemma 21 in this case, we must show that at h_n^{1*} , there is an agent ℓ that is an active non-lurker at h_n^{1*} that does not move at h_n^{1*} and $x_i \in C_\ell^{\neq}(h_n^{1*})$.

We first claim that at h_n^{1*} , there are at least $P + 1$ lurked objects. As discussed at the beginning of the proof of Lemma 5, we can assume that $j_1 \succ_n \dots \succ_n j_P \succ_n \dots$ and that the

objects x_{j_1}, \dots, x_{j_P} are all lurked at $h_n^{1*} = (\check{h}, a^*, \dots, a^*)$, where $\check{h} = (h_\emptyset, a^*, \dots, a^*)$ is the earliest history at which the youngest lurked object among the x_{j_p} 's, x_{j_P} , becomes lurked. So, there are at least P lurked objects at h_n^{1*} . If there are exactly P lurked objects at h_n^{1*} , then no new object has become lurked on the passing path from \check{h} to h_n^{1*} , which implies that h_n^{1*} is not a terminating history. This follows because a presumption in the statement of the lemma is that x_{k_1} is the $(P+1)^{th}$ lurked object along the initial passing path, so there must be some continuation history $(h_n^{1*}, a^*, \dots, a^*)$ at which x_{k_1} becomes lurked. We also conclude that the agents coded in step 1 of the coding algorithm applied to Γ_n are j_1, \dots, j_P, k_1 ,¹⁰⁹ and so one of these agents moves at h_n^{1*} . Recall that by Lemma 3, all of these agents receive the same object in all games under consideration, namely x_{j_p} for agent j_p and x_{k_1} for agent k_1 (cf. the last bullet point in the discussion of arms on page 36). Assume first that the agent who moves at h_n^{1*} is some j_p . This means that j_p must be clinching x_{j_p} at h_n^{1*} . As x_{j_p} is a lurked object at h_n^{1*} , Lemma 14 implies that h_n^{1*} is the terminating history, which is a contradiction. So, the agent who moves at h_n^{1*} must be agent k_1 , who clinches x_{k_1} . But then, at h_n^{1*} , x_{k_1} has been offered to an active non-lurker, which implies that x_{k_1} cannot be the next lurked object along the initial passing path (Remark 5), which is again a contradiction.

Therefore, there are at least $P+1$ lurked object at h_n^{1*} . In particular, the objects $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n'-1}}$ are all lurked at h_n^{1*} , while $x_{k_{n'}}$ is not, for some $n > n' > 1$.¹¹⁰ Consider game $\Gamma_{n'+1}$ and notice that \succ_n is equivalent to $\succ_{n'+1}$ up to agent $k_{n'}$. By Lemma 4, all agents $j_1, \dots, j_P, k_1, \dots, k_{n'}$ are in the same roles in all of these games, and $h_n^{1*} = h_{n'+1}^{1*}$. By Lemma 20,¹¹¹ in $\Gamma_{n'+1}$, at $h_{n'+1}^{1*}$, there is an agent ℓ such that ℓ is an active non-lurker at $h_{n'+1}^{1*}$ that does not move at $h_{n'+1}^{1*}$, and $x_i \in C_\ell^\times(h_{n'+1}^{1*})$. Since $h_n^{1*} = h_{n'+1}^{1*}$, we conclude the same is true at h_n^{1*} . Thus, we can apply Lemma 21 to conclude that agent i must tie with some agent k_{n+1} in \succ_{n+1} . ■

Proof of Lemma 23

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

¹⁰⁹This is because the agents coded in a step that is initiated at some h_n^{1*} are: (i) the lurkers at h_n^{1*} , (ii) the non-lurker who moves at h_n^{1*} , and (iii) possibly one more active non-lurker, and if this agent is coded, there must be a tie at the end of the step. As there is no tie in $(P+1)^{th}$ place under \succ_n , there is no agent of type (iii), and thus at most $P+1$ agents are coded, namely the agents j_1, \dots, j_P, k_1 .

¹¹⁰Because $Q \geq 2$, 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 at this point. Thus, the last lurked object is at most $x_{k_{n-2}}$, which implies $n' < n$. That $n' > 1$ follows from the supposition that x_{k_1} is the $(P+1)^{th}$ lurked object on the initial passing path of the game form, and, as we just showed, at least $P+1$ objects are lurked at h_n^{1*} .

¹¹¹In this case, we apply Lemma 20 to the arm $(\sigma_1, \dots, \sigma_{n'+1})$.

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

$$\begin{aligned} j_1 \cdots j_P \succ_0 i \succ_0 k_1 \succ_0 \cdots \\ j_1 \cdots j_P \succ_1 \{i, k_1\} \succ_1 \cdots \end{aligned}$$

Then:

(a) We have $h_0^* \not\subseteq h_1^*$, 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 $h_0^* \not\subseteq h_2^*$. Further, in \succ_2 , agent i must tie with some other agent k_2 .

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

The proof of this claim can be found immediately after the proof of this lemma. Now, consider a sequence α 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 $j_1, \dots, j_P, k_1, \dots, k_{n-1}$. Let Q be the step number in which i is coded in game Γ_n . The goal is to apply Lemma 21, 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 $x_i \in C_\ell^{\subseteq}(h_n^*)$.

The proof of this claim is found below, immediately after the proof of Claim 7. Given Claim 8, we can apply Lemma 21 to conclude that i must tie in \succ_{n+1} , which completes the proof of Lemma 23. ■

Proof of Claim 7 (stated above, in the proof of Lemma 23). 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}, \dots, 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 $x_i \in C_{s'}^{\subseteq}(h_1^*)$ and $x_i \in C_s(h_1^*)$. Now, for i to be ranked first without ties in \succ_0 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 $x_{k_1} \in C_s^{\neq}(h_1^*)$ and $x_{k_1} \in C_{s'}^{\subseteq}(h_1^*)$. Now, $h_1^* \subseteq h_0^*$ implies that in Γ_0 , $\sigma_0^{-1}(k_1) \neq s, s'$.¹¹² 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 5), and so for k_1 to be ranked second, there can be at most 3 active agents 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^* .¹¹³ 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 initiating 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}, \dots, 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 $x_i \in C_{s'}^{\neq}(h_1^*)$. 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}, \dots, 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 $h' \not\subseteq h_1^*$. 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.¹¹⁴ 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.

¹¹²If $\sigma_0^{-1}(k_1) = s$, then k_1 would clinch at some $h' \not\subseteq h_1^*$; 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 .

¹¹³If 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 .

¹¹⁴This follows from Lemma 15. If this role were the terminator, then role s could not be offered x_i at $h_1^* \not\supseteq h'$.

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 $x_{k_1} \in C_s(h_1^*)$. 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 114). 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 114) 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 $\ell \neq j_1, \dots, j_P, i, k_1$ 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 $p = 1, \dots, P$. 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 $x_i \notin C_{s'}^\neq(h_1^*)$.

So, because there is a tie in Γ_1 , it must be that $x_{k_1} \in C_{s'}^{\neq}(h_1^*)$.

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 $h' \not\subseteq h_1^*$ at which it was offered to her, a contradiction.

Therefore, there is some $\ell \neq j_1, \dots, j_P, i, k_1$ 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 $h_0^* \not\subseteq h_1^*$. To finish the proof of part (a), we must show that agent i moves at h_0^* in Γ_0 . Notice that $h_0^* \not\subseteq h_1^*$ 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 $h_0^* \not\subseteq h_1^*$. Therefore, no agent j_1, \dots, 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 $h_0^* \not\subseteq h_2^*$. By part (a), $h_0^* \not\subseteq h_1^*$. 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 $h_0^* \not\subseteq h_1^*$. Therefore, it must be k_1 that moves at h_0^* in Γ_1 .

By way of contradiction suppose that $h_0^* \not\subseteq h_2^*$ 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 $h_2^* \not\subseteq h_0^*$.¹¹⁵ So, in Γ_0 , it must be some agent $\ell \neq j_1, \dots, j_P, i, k_1$ 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, $h_0^* \not\subseteq h_2^*$.

Thus, we have $h_0^* \not\subseteq h_1^*, h_2^*$, 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 $h_1^* \not\subseteq h_2^*$. 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

¹¹⁵If 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.

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 15(ii), which is a contradiction. So, it must be some $\ell \neq j_1, \dots, j_P, i, k_1$ that moves at h_1^* in Γ_2 , and so $x_i \in C_\ell^{\neq}(h_2^*)$ 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, \dots, 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 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 $h' \not\subseteq h_1^*$ 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.¹¹⁶ We also cannot have $\sigma_2(r) = i$, because i would clinch x_i at the history $h' \not\subseteq h_1^*$ 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 initiating 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 $h_0^* \not\subseteq h_2^*$ 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 $h_1^* \not\subseteq h_2^*$. 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 $h' \not\subseteq h_2^*$). 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 $x_i \in C_\ell^{\neq}(h_2^*)$ in Γ_2 . ■

Proof of Claim 8 (stated above, in the proof of Lemma 23). Since it is without loss of

¹¹⁶Note 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.

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,¹¹⁷ and so, by Claim 7 part (c), there is an agent ℓ that is an active non-lurker at h_2^* that does not move at h_2^* and $x_i \in C_\ell^{\neq}(h_2^*)$. 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 $x_{j_1}, \dots, x_{j_P}, x_{k_1}, \dots, x_{k_{n'-1}}$ are lurked at h_n^* , while $x_{k_{n'}}$ is not, where $n > n' > 1$.¹¹⁸ 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 $k_1, \dots, k_{n'}$ are in the same roles in all of these games, and $h_n^* = h_{n'+1}^*$. By Lemma 20, 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 $x_i \in C_\ell^{\neq}(h_{n'+1}^*)$. Since $h_n^* = h_{n'+1}^*$, the result holds. ■

Proof of Lemma 24

By the assumption that $n, m \geq 2$ in α and α' , we have that there exist (at least) the following codings:

$$\begin{aligned} j_1 \cdots j_P \succ_1 \{i, k_1\} \succ_1 \cdots \\ j_1 \cdots j_P \succ_2 k_1 \succ_2 \{i, k_2\} \cdots \\ j_1 \cdots j_P \succ_2' i \succ_2' \{k_1, k_2\} \cdots \end{aligned}$$

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 $x_i \in C_\ell^{\neq}(h_2^*)$.
- Condition 2': In $\Gamma_{2'}^*$, at $h_{2'}^*$, there is an active non-lurker ℓ such that ℓ does not move

¹¹⁷If 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 path (because no such object exists).

¹¹⁸Because $Q \geq 2$, 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 $n' < n$.

at $h_{2'}^*$ and $x_{k_1} \in C_\ell^{\text{ff}}(h_{2'}^*)$.¹¹⁹

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 24 holds. The two statements are symmetric, so this will also show that Condition 2' implies Statement 2 of Lemma 24.

To show Condition 2 implies Statement 1, we use Lemma 21. So, consider the sequence 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 $(p + 1)^{\text{th}}$ lurked object would have to be x_{k_1} , which contradicts the supposition of the lemma. Therefore, for all $n' = 1, \dots, n + 1$, at $h_{n'}^*$ in game $\Gamma_{n'}$, there are exactly P lurked objects, and by definition, these must be x_{j_1}, \dots, 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, \dots, 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 $h_n^{1*} = h_2^*$. By Condition 2, there is some agent ℓ such that ℓ is an active non-lurker that does not move at h_2^* and $x_i \in C_\ell^{\text{ff}}(h_2^*)$. Since $h_n^{1*} = h_2^*$, we have that in Γ_n , there is some agent ℓ' that is an active non-lurker at h_n^{1*} and that does not move at h_n^{1*} and $x_i \in C_{\ell'}^{\text{ff}}(h_n^{1*})$. Further, $Q \geq 2$. Thus, all of the conditions of Lemma 21 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 24 by showing that at least one of Condition 2 or Condition 2' must hold. This assertion is proven as Lemma 27 below.

¹¹⁹We 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).

Lemma 27. *Assume that there are three codings:*

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

such that:

- At each of h_A^*, h_B^*, h_C^* , the objects x_{j_1}, \dots, x_{j_P} are all lurked, in this order, and
- Neither x_i nor x_k are the $(P+1)^{\text{th}}$ 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 $x_k \in C_\ell^{\bar{c}}(h_B^*)$.

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 $x_i \in C_\ell^{\bar{c}}(h_C^*)$.

Proof of Lemma 27. 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}, \dots, 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 $x_i \in C_{s'}^{\bar{c}}(h_A^*)$, 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 $x_k \notin C_{s'}^{\bar{c}}(h_A^*)$; indeed, if this were true, then agent 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 15(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, \dots, 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 $\sigma_C(s) = j_{p'}$ 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 $\ell \neq j_1, \dots, j_P, k$ such that $\sigma_C(s') = \ell$. We claim that $\ell \neq i$. Indeed, if $\ell = i$, then there is some history $h' \not\subseteq h_C^*$ 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, $x_k \notin C_{s'}^{\subseteq}(h_A^*)$ and h_A^* is the terminating history, so $h_C^* \subseteq h_A^*$. Let $h' \not\subseteq h_A^*$ be the history at which role s' is offered to clinch x_i .

If $h_C^* \not\supseteq h'$, then, by similar logic to subcase 1.1, $\sigma_C(s') = \ell$ for some $\ell \neq j_1, \dots, j_P, k, i$, and Condition (C) holds.

Finally, consider $h_C^* \not\subseteq h'$.¹²⁰ 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 $h_B^* \subseteq h' \not\subseteq h_A^*$. 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 $h_B^* \not\subseteq h_C^*$, then by similar logic to the above, Condition (C) holds. If $h_C^* \not\subseteq h_B^*$, then $x_k \in C_s^{\subseteq}(h_B^*)$ 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 $h_C^* \not\subseteq h_B^*$ in Γ_B , a contradiction. It is also immediate that $\sigma_B(s) \neq j_1, \dots, j_P$, since they must be in the lurker roles for their respective objects. Thus, $\sigma_B(s) = \ell$ for some $\ell \neq j_1, \dots, j_P, i, k$, and Condition (B) holds.

Case 2: $\sigma_A(s) = i$. We once again have that role s' is not the terminator role,¹²¹ 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 $x_i \in C_{s'}^{\subseteq}(h_C^*)$. Since this agent is not coded in step 1, we know that $\sigma_C(s') \neq j_1, \dots, j_P, k$. If $\sigma_C(s') = i$, then i is offered to clinch x_i at some $h' \not\subseteq h_C^*$, and since s' is not the terminator role, only unlurked objects are possible for her, and therefore,

¹²⁰Note that $h_C^* = h'$ is ruled out because role s' moves at h' , while role s moves at h_C^* .

¹²¹This follows from Lemma 15.

since x_i is i 's top object, she would clinch at h' , a contradiction. Thus, $\sigma_C(s') = \ell$ for some $\ell \neq j_1, \dots, j_P, i, k$, and $x_i \in C_\ell^{\neq}(h_C^*)$, 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 $p = 1, \dots, P$.

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 $\tilde{h} \not\subseteq h_A^* \subseteq h_C^*$, and since x_i is i 's favorite unlurked object, she will clinch it at \tilde{h} , a contradiction. Therefore, in either case, there is some agent $\ell \neq j_1, \dots, j_P, i, k$ such that $x_i \in C_\ell^{\neq}(h_C^*)$, and Condition (C) holds.

It remains to consider $h_C^* \not\subseteq h_A^*$. Here, we must have $\sigma_C(s) = k$, because if $\sigma_C(s') = k$, then as we showed above, $x_k \notin C_{s'}^{\neq}(h_A^*)$, 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, \dots, j_P, i .

If $h_B^* \subseteq h_C^*$, then, $h_B^* \not\subseteq h_A^*$, 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 $h_B^* \not\subseteq h_A^*$. 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 $x_i \in C_{s'}^{\neq}(h_C^*)$. 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 $\ell \neq j_1, \dots, j_P, i, k$ and $x_i \in C_\ell^{\neq}(h_C^*)$, and thus, Condition (C) holds.

If $h_C^* \not\subseteq h_B^*$, 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 \triangleright_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 \triangleright_B , a contradiction. Thus, $\sigma_B(s') = i$. This means that h_C^* and h_B^* are controlled by different roles, and $x_k \in C_s^{\neq}(h_B^*)$. Finally, we cannot have $\sigma_B(s) = k$, because then k is in the same role as Γ_C , and would clinch at $h_C^* \not\subseteq h_B^*$ in Γ_B . So, we must have $\sigma_B(s) = \ell$ for some $\ell \neq j_1, \dots, j_P, i, k$, and in Γ_B , $x_k \in C_\ell^{\neq}(h_B^*)$. 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. ■