House Allocation with Existing Tenants: Two Equivalence Results

Ekici, Özgün

Ozyegin University

2011

House Allocation with Existing Tenants: Two Equivalence Results*

Özgün Ekici
Özyeğin University, Istanbul, Turkey
ozgun.ekici@ozyegin.edu.tr

April, 2019

Abstract

We study the house allocation problem with existing tenants: n houses (stand for "indivisible objects") are to be allocated to n agents; each agent needs exactly one house and has strict preferences; k houses are initially unowned; k agents initially do not own houses; the remaining $n-k$ agents (the so-called "existing tenants") initially own the remaining $n-k$ houses (each owns one). In this setting, we consider various randomized allocation rules under which voluntary participation of existing tenants is assured and the randomization procedure either treats agents equally or discriminates against some (or all) of the existing tenants. We obtain two equivalence results, which generalize the equivalence results in Abdulkadiroğlu and Sönmez (1998) and Sönmez and Ünver (2005).

Key Words: house allocation with existing tenants; house allocation; housing market; equivalence of mechanisms.

JEL Codes: C78; D71; D78

*The paper benefited greatly from various comments by Onur Kesten, Utku Ünver, Tayfun Sönmez, William Thomson, Isa Hafalir, Micheal Trick, Gabriel Carroll, Rick Green, and an anonymous referee.

1 Introduction

We consider the classical problem of allocating n indivisible objects to n agents. We assume that each agent needs exactly one object, agents' preferences over objects are strict, and monetary transfers are not allowed. As real-life applications, consider the problems of allocating posts at hospitals to medical interns, dormitory rooms to college students, or kidneys for transplant to patients with kidney disease. In the literature it is conventional to refer to objects as "houses" and in our paper we follow this convention.

This problem has three variants in the literature: In the so-called *house allocation problem* it is presumed that houses are initially unowned. In the so-called *housing market* it is presumed that each agent initially owns one of the houses.[1] The third variant is the hybrid case, introduced by Abdulkadiroğlu and Sönmez [2] and formulated in their paper as a *house allocation problem with existing tenants*: k houses are initially vacant; k agents are newcomers; and each of the remaining $n-k$ houses is occupied by one of the remaining $n-k$ agents (referred to as "existing tenants"). In our paper, we study this hybrid case and follow its formulation in their paper. Throughout, we assume that if they wish so, existing tenants can keep their occupied houses or even trade them with one another. Therefore, although doing so is an oxymoron, we will speak of existing tenants as the "owners" of their occupied houses.

In real-life applications in which there are no existing tenants, the allocation of houses is often carried out by a mechanism (allocation rule) called *random priority*: First, a priority order of agents is chosen uniformly at random. Then the first agent in the order receives her top choice, the next agent receives her top choice among the remaining houses, and so on. The random priority has some very appealing properties. Most notably, it is simple, strategy-proof (i.e., it is immune to misrepresentation of preferences) and efficient (i.e., it always induces Pareto-efficient allocations). Alas, the presence of existing tenants pose two challenges not addressed under random priority. First, the allocation it induces is not always "group-rational." This means that under this mechanism a subset of existing tenants is not assured that their coalitional allocation will always be (in the Pareto sense) at least as good as any coalitional allocation that they can attain by trading their occupied houses. Without this assurance, it is conceivable that these existing tenants opt out, which may lead to a loss in potential gains from trade. Therefore, to assure the voluntary participation of existing tenants we take it to be the case that the allocation rule used always induces a group-rational allocation. Another feature of random priority is that under its random component (the random choice of a priority order) agents are put on an equal footing: The agents are ranked high in the priority order with equal probabilities. While this may be a desirable feature in real-life applications in which there are no existing tenants, when they are present it is conceivable that the mechanism designer deem it justified to discriminate against all or a subset of existing tenants. For instance, in kidney exchange practices, there are some patients ("existing tenants") who already have compatible donors. The remaining patients however either have no donors ("newcomers") or have incompatible donors (also "existing tenants"). For this latter group of patients finding a kidney transplant from a compatible donor is a life-and-death matter. In the eyes of the medical authority, therefore, it may be justified to treat more favorably this latter group of patients even if it means discrimination against patients with compatible donors.[2]

In this paper, to tackle the above-mentioned two challenges we propose four random mechanisms. They are defined by means of two algorithms: The Y-I algorithm and the augmented

[1] The house allocation problem was introduced by Hylland and Zeckhauser [6]), and the housing market, by Shapley and Scarf [17].

[2] For studies of the kidney exchange problem, see [14, 15, 16, 20].

TTC algorithm. The Y-I algorithm allocates houses to agents by means of a priority order of agents: The first agent in the order receives her top choice, the next agent receives her top choice among the remaining houses, and so on. However, if an agent a requests the occupied house of an existing tenant e and e has not been assigned a house yet, the remainder of the order is updated by moving e to the top (right above a) and then we proceed.[3] As it turns out, thanks to this update protocol the Y-I algorithm always induces a group-rational allocation (see Ekici [5]). The augmented TTC algorithm allocates houses to agents by means of an "augmenting function" v: For each agent, v either assigns her the ownership of a vacant house or appoints her as the "inheritor" of an existing tenant. When our original problem is augmented by v, this defines a private-ownership economy in which vacant houses are now also owned. Then the houses are allocated to agents by identifying (top trading) "cycles": A cycle is a series $a^1, a^2, \cdots, a^s = a^1$ of agents such that the favorite house of a^1 is owned by a^2; the favorite house of a^2 is owned by a^3; and so on. In each cycle the corresponding trades are performed and then all the agents belonging to cycles are removed together with their assignments. If an existing tenant owns two houses, when she is removed one of her houses still remains. This house is then given to the inheritor of the existing tenant (the inheritor appointed by v). If the inheritor has been removed, too, the house that remains is given to the inheritor of the inheritor, and so on. Then in the reduced private-ownership economy we proceed similarly: by identifying new cycles and performing the corresponding trades and so on.[4] As it turns out, the augmented TTC algorithm also always induces a group-rational allocation.

The first two random mechanisms that we propose are for real-life scenarios in which discrimination against existing tenants is deemed unjustified (perhaps, when allocating dormitory rooms to college students). They are as follows:

(1) *The random Y-I mechanism*: A priority order f of agents is chosen uniformly at random. Then the Y-I algorithm is executed using the priority order f.

(2) *The random augmented TTC mechanism*: An augmenting function v is chosen uniformly at random. Then the augmented TTC algorithm is executed using the augmenting function v.

Note that the random components of these two mechanisms put agents on an equal footing: f and v are chosen uniformly at random.

The next two random mechanisms that we propose are for real-life scenarios in which discrimination against a subset E of existing tenants is justified. They are as follows:

(3) *The E-discriminating random Y-I mechanism*: A priority order f of agents is chosen as follows. The existing tenants in E are placed at the bottom in some fixed order. The remaining agents are ordered at the top uniformly at random. Then the Y-I algorithm is executed using the priority order f.

(4) *The E-discriminating random augmented TTC*: An augmenting function v is chosen as follows. The existing tenants in E are appointed to be the inheritors of one another in some fixed order. The remainder of v is chosen uniformly at random. Then the augmented TTC algorithm is executed using the augmenting function v.

[3] The Y-I algorithm was introduced by Abdulkadiroğlu and Sönmez [2]. In its execution if a "loop" arises, where a number of existing tenants request the occupied houses of one another in succession, they are assigned the houses they request and then we proceed. For other studies related to this algorithm, see [4, 5, 18].

[4] The augmented TTC algorithm is well-defined (see footnote 8 in Section 2). It is a generalization of the TTC algorithm, attributed to Gale in Shapley and Scarf [17]. For other studies related to TTC see [1, 9, 12, 13].

Notice how the random components of these two mechanisms discriminate against the existing tenants in E (or, equally, notice how they favor the remaining agents): When the priority order f is chosen, they are placed at the bottom of the order and hence put at a disadvantage. When the augmenting function v is chosen, they never get a chance to receive vacant houses and hence are put at a disadvantage in the subsequent trade protocol. As an illustration of this discrimination, imagine the case where every agent prefers any vacant house to any occupied house. Then, under these two mechanisms no existing tenant in E ever receives one of these coveted vacant houses.

Above, the four random mechanisms that we propose are defined by means of two seemingly different algorithms: (1) and (3) are defined by means of the Y-I algorithm, and (2) and (4) are defined by means of the augmented TTC algorithm. Even so, the main theoretical results of our paper show that they are closely related: In Theorem 1, we show that the random Y-I and the random augmented TTC mechanisms are equivalent. (That is, they induce any given allocation with exactly the same probability.) In Theorem 2, we show that the E-discriminating random Y-I and the E-discriminating random augmented TTC mechanisms are also equivalent. Indeed, it is Theorem 2 which is the main theoretical contribution of our paper. It helps unify and generalize the following equivalence results:

– Abdulkadiroğlu and Sönmez [1] study the special case of our problem in which there are no existing tenants. In this context, they show that the mechanisms random priority[5] and "core from random endowments" are equivalent. The E-discriminating random mechanisms in our paper reduce to these two mechanisms when there are no existing tenants. Therefore, their equivalence result follows from Theorem 2.

– Sönmez and Ünver [19] study this problem and they show the equivalence of two random mechanism. The E-discriminating random mechanisms in our paper reduce to these two mechanisms when the set E includes every existing tenant. Therefore, their equivalence result follows from Theorem 2.

– Theorem 1 in this paper indeed also follows from Theorem 2. We present Theorem 1 as a separate result, however, for expositional reasons and since we prove Theorem 2 using tools introduced to show Theorem 1.[6]

A mechanism is said to be deterministic if for each preference profile of agents its allocation choice is certain. The Y-I algorithm defines a class of deterministic mechanisms, each specified by the choice of the priority order f. Similarly, the augmented TTC algorithm defines a class of deterministic mechanisms, each specified by the choice of the augmenting function v. Since our random mechanisms randomize over these two classes of mechanisms, they inherit their nice theoretical properties: They are strategy-proof and efficient, and they always induce group-rational allocations. In showing Theorem 1, we indeed show that there is a one-to-one correspondence between these two classes of mechanisms: For each priority order f, there exists a distinct augmenting function v such that the Y-I mechanism specified by f is the same as the augmented TTC mechanism specified by v (due to Lemmas 1 and 4). In other words, we show that these two classes of mechanisms indeed coincide. These two classes of mechanisms are subsets of the more general class of "hierarchical exchange rules," introduced by Pápai [10] and which she characterized by a nice set of theoretical properties. We believe that the correspondence between the

[5]This mechanism is referred to as "random serial dictatorship" in Abdulkadiroğlu and Sönmez [1].

[6]Note that Theorem 1 and the result in Sönmez and Ünver [19] both imply the equivalence result in Abdulkadiroğlu and Sönmez [1] but otherwise these two results are independent. For two other studies with equivalence results in this literature, see [3, 11]. In a more recent paper, Lee and Sethuraman [8] develop a new technique based upon induction to prove equivalence results.

classes of Y-I and augmented TTC mechanisms them assigns a focal place in the larger class of Pápai's hierarchical exchange rules. Also, to our best knowledge, the E-discriminating random mechanisms in our paper are the first random mechanisms in the literature proposed for problems where it is deemed socially desirable to discriminate against only a subset of existing tenants in order to favor the remaining agents. The choice between them turns out to be inconsequential since we show that they are equivalent.

Much of the terminology that we use is taken from Abdulkadiroğlu and Sönmez [1]. The proof strategy that we use to show Theorem 1 also follows the outline of their proof. There are some additional complications in our context, however, due to the presence of existing tenants. To deal with them, we introduce the "clone problem" (a different formulation of the house allocation problem with existing tenants) as well as some additional tools, which may be of independent interest. The rest of the paper is organized as follows: Section 2 introduces the problem and the Y-I and the augmented TTC algorithms. Section 3 introduces the clone problem. Section 4 introduces the Y-I and the augmented TTC algorithms in the context of the clone problem. Section 5 introduces and explores the "chain order," a construct that helps us link these two algorithms. Section 6 presents our results. The proofs of two lemmas are given in the Appendix.

2 Preliminaries

A *house allocation problem with existing tenants* is a five-tuple $\Pi : \langle A_N, H_V, A_E, H_O, \succ\rangle$ s.t.:

– $A_N : \{a_1, a_2, ..., a_k\}$ is a finite set of "newcomers";

– $H_V : \{h_1, h_2, \ldots, h_k\}$ is a finite set of "vacant houses";

– $A_E : \{e_{k+1}, e_{k+2}, ..., e_n\}$ is a finite set of "existing tenants";

– $H_O : \{o_{k+1}, o_{k+2}, ..., o_n\}$ is a finite set of "occupied houses";

– $\succ$: $(\succ_a)_{a\in A_N\cup A_E}$ is the profile of agents' *strict* preference relations over $H_V \cup H_O$.

Let A and H be the sets of agents and houses (i.e., $A = A_N \cup A_E$ and $H = H_V \cup H_O$). We write $h \succ_a h'$ to indicate that agent a prefers house h to h'. We denote the set of preference relations by $\succ$ (so, $\succ\in\succ^n$). Let $\succsim_a$ be the "at least as good as" relation for agent a (i.e., $h \succsim_a h'$ means $h \succ_a h'$ or $h = h'$).

An *allocation* $\mu : A \to H$ is a one-to-one mapping from the set of agents to the set of houses. We denote the set of allocations by $\mathcal{M}$. Given $\succ\in\succ^n$, an allocation μ is:

– *Pareto-efficient* if there exists no $\mu' \in \mathcal{M}$ such that for each $a \in A$, $\mu'(a) \succsim_a \mu(a)$, and for some $a \in A$, $\mu'(a) \succ_a \mu(a)$.

– *group-rational* if there exists no triplet $\langle A'_E, H'_O, \theta\rangle$ where $A'_E \subseteq A_E$; $H'_O \subseteq H_O$ is the set of occupied houses of existing tenants in A'_E; and $\theta : A'_E \to H'_O$ is a bijection such that for each $a \in A'_E$, $\theta(a) \succsim_a \mu(a)$, and for some $a \in A'_E$, $\theta(a) \succ_a \mu(a)$.

A *mechanism* (or, an *allocation rule*) is a systematic way to choose an allocation for any given preference profile. In this paper, we study both deterministic and random mechanisms:

– A *deterministic mechanism* $\varphi :\succ^n \rightarrow \mathcal{M}$ is a mapping from the set of preference profiles to the set of allocations.

  That is, for any given preference profile, the allocation choice is certain.

– A *probability distribution* $\delta : \mathcal{M} \rightarrow [0, 1]$ is a mapping such that $\sum_{\mu \in \mathcal{M}} \delta(\mu) = 1$.

  For each allocation μ, δ specifies the probability $\delta(\mu)$ with which it is chosen. Let $\Delta(\mathcal{M})$ be the set of probability distributions over the set of allocations.

– A *random mechanism* $\varphi :\succ^n \rightarrow \Delta(\mathcal{M})$ is a mapping from the set of preference profiles to the set of probability distributions over the set of allocations.

  That is, given $\succ$, the allocation choice is randomly made as specified by the probability distribution $\varphi(\succ)$.

– Two (random) mechanisms φ_1 and φ_2 are *equivalent* if for each $\succ \in \succ^n$, $\varphi_1(\succ) = \varphi_2(\succ)$.

  That is, for each preference profile, they induce the same probability distribution over the set of allocations.

The random mechanisms that we consider derive from two algorithms: The first one is the "You-request-my-house–I-get-your-turn algorithm." It is due to Abdulkadiroğlu and Sönmez [2] and in short, we call it the Y-I algorithm. This algorithm allocates houses by means of a priority order of agents. Thus, each priority order defines a distinct "Y-I mechanism."

Let $f : A \rightarrow \{1, 2, \cdots, n\}$ be a one-to-one mapping that orders agents—called a *priority order*. Let $\mathcal{F}$ be the set of priority orders. The Y-I mechanism defined by f is denoted by Y-I(f) and it proceeds as follows:

Y-I(f). *The agent ordered first in f is assigned her favorite house; the next agent is assigned her favorite house among the remaining ones; and so on, until an agent $\boldsymbol{a}$ requests an occupied house $\boldsymbol{o}_s$. If at that point $\boldsymbol{e}_s$ has already been assigned a house, $\boldsymbol{a}$ is assigned $\boldsymbol{o}_s$ and we proceed. Otherwise, the remainder of the priority order is updated: $\boldsymbol{e}_s$ is inserted at the top, right above $\boldsymbol{a}$. Then we proceed. If at any point a "loop" forms, in which a subset of existing tenants request the occupied houses of one another in succession, they are assigned the houses they request and then we proceed.*

The second algorithm that we will need is the "augmented top trading cycle algorithm." In short, we call it the augmented TTC algorithm. This algorithm allocates houses by means of an "augmenting function." The augmenting function appoints an owner for each vacant house. Also, for each existing tenant, it appoints an "inheritor." Thereby, a private-ownership economy is induced, in which vacant houses are now also owned, and for each agent with multiple houses there is an appointed inheritor. The assignments are then made using Gale's TTC protocol and the inheritance relationships among agents.

In order to indicate the "inheritor" of an existing tenant, we introduce the mathematical construct called an "inheritance right": For each existing tenant e_s, let i_s be the associated "inheritance right". Let $I : \{i_s\}_{s=k+1}^{n}$ be the set of inheritance rights. Let $v : A \rightarrow H_V \cup I$ be a one-to-one mapping from the set of agents to the set vacant houses and inheritance rights—called an *augmenting function*: For agent a, if $v(a) \in H_V$, it means v assigns a the ownership of vacant house $v(a)$. Otherwise, $v(a)$ is an inheritance right, say i_s, which means v appoints a as the "inheritor" of existing tenant e_s. The augmented TTC mechanism defined by v is denoted by TTC(v) and it proceeds as follows:

TTC(v). *Consider the private-ownership economy induced when the problem is augmented by* v. *Let each agent point to her favorite house and each house point to its owner. There exists one or more (top trading) cycles: A "cycle" is a series* $a^1, a^2, \cdots, a^s = a^1$ *of agents such that* a^1 *points to a house owned by* a^2 *which points to* a^2*;* a^2 *points to a house owned by* a^3 *which points to* a^3*; and so on. In each cycle perform the corresponding trades and then remove all the agents belonging to cycles together with their assignments. For an existing tenant who owns two houses, when she is removed, one of her houses still remains. The ownership of this house is then given to her inheritor (the inheritor appointed by* v*). If the inheritor has been removed, too, then the house is given to the inheritor of inheritor, and so on.*[7] *Then in the reduced private-ownership economy the allocation process continues similarly by identifying new cycles and performing the corresponding trades and so on.*

3 The Clone Problem

We introduce next our "clone problem," derived from Π and denoted by $Cl\{\Pi\}$. The clone problem $Cl\{\Pi\}$ is just a different formulation of the problem Π. In Section 4, we introduce the Y-I and the augmented TTC mechanisms in the context of the clone problem. This becomes extremely useful in simplifying our exposition and showing our results. In the clone problem, associated with each existing tenant e_s, we introduce a "clone," c_s. It is presumed that the clone c_s has the same preferences as e_s and she owns o_s (i.e., the occupied houses are presumed to be the properties of the clones, not the existing tenants). Also, the "inheritance rights" that we defined in Section 2 are now embedded in the clone problem definition.

A *clone problem* is a seven-tuple $Cl\{\Pi\} : \langle A_N, H_V, A_E, H_O, \succ, C, I\rangle$ s.t.:

– $A_N, H_V, A_E, H_O, \succ$ are as defined in Π;

– $C : \{c_{k+1}, c_{k+2}, ..., c_n\}$ is a finite set of "clones";

– $I : \{i_{k+1}, i_{k+2}, ..., i_n\}$ is a finite set of "inheritance rights."

As in the preceding section, A and H denote the sets of agents and houses.[8] As need arises, we will introduce some additional terms associated with the clone problem. To ease reference we present them in bullet points. The first group of these terms are introduced below.

- **ec-pair, items, hi-items.**

  For ease of reference, we refer to:

  – an existing tenant e_s and her clone c_s jointly as an *ec-pair*;

  – houses and inheritance rights jointly as *items*;

[7] It turns out that there is always an "inheritor" among the remaining agents and hence the augmented TTC algorithm is well-defined. To see this, let e be an existing tenant who owns two houses, h and h'. Let e be removed and h be the house that still remains. Let agent a be the inheritor of e. If a is a newcomer, a is among the remaining agents because she had no houses to trade with earlier. If a is an existing tenant, she assumes the ownership of h if she is among the remaining agents. Otherwise, we turn to the inheritor of agent a, say a'. But then the previous arguments can be repeated for agent a' and so on, until among the remaining agents we identify an "inheritor" who assumes the ownership of h when e leaves.

[8] i.e., $A = A_N \cup A_E$ and $H = H_V \cup H_O$. Note that we do not refer to clones as "agents." Thus, in the context of the clone problem whenever we speak of agents, it is understood that we mean by it the newcomers and existing tenants.

– vacant houses and inheritance rights jointly as *hi-items* (since they are denoted by h_1, h_2, i_8, i_9 and so on).

We introduce next a "clone allocation," defined in the context of the clone problem but which can easily be converted into an (ordinary) allocation.

- **clone allocation.**

  A *clone allocation* $\overline{\mu} : A \cup C \rightarrow H \cup I$ is a one-to-one mapping from the set of agents and clones to the set of items such that for each ec-pair e_s and c_s, one item in the set $\{\overline{\mu}(c_s), \overline{\mu}(e_s)\}$ is a house and the other is i_s.

  That is, at a clone allocation $\overline{\mu}$, for an ec-pair, only one of them is assigned a house. Thus, the (ordinary) allocation corresponding to a clone allocation is obtained in a straightforward manner via the following mapping ρ:

  For $\overline{\mu}$, the corresponding (ordinary) allocation $\rho(\overline{\mu}) \in \mathcal{M}$ is such that for a newcomer a, $\rho(\overline{\mu})(a) = \overline{\mu}(a)$, and for an existing tenant e_s, $\rho(\overline{\mu})(e_s)$ is the house in the set $\{\overline{\mu}(c_s), \overline{\mu}(e_s)\}$.

In the rest of the paper, for illustrative purposes, we refer to the clone problem presented in Example 1.

EXAMPLE 1 **(a clone problem)** *Consider the clone problem such that $n = 12$, $k = 8$, and agents' preference rankings of houses (not fully specified) are as in the following table:*

a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8, c_8	e_9, c_9	e_{10}, c_{10}	e_{11}, c_{11}	e_{12}, c_{12}
h_3	h_1	h_3	h_6	h_4	o_{11}	o_8	o_9	h_2	h_1	o_{10}	h_5
$\vdots$	$\vdots$	o_{10}	$\vdots$	$\vdots$	o_8	h_1	$\vdots$	$\vdots$	o_{11}	$\vdots$	$\vdots$
		o_{12}			$\vdots$	o_{10}			$\vdots$		
		$\vdots$				h_7					
						$\vdots$					

◊

4 The Clone Specifications of the Mechanisms

In this section we introduce the clone specifications of the Y-I and the augmented TTC mechanisms. The clone specifications of these mechanisms operate in the clone problem $Cl\{\Pi\}$ and induce clone allocations. It turns out that they are equivalent to the ordinary specifications in the following sense: For a Y-I or an augmented TTC mechanism, if its clone specification induces in problem $Cl\{\Pi\}$ the clone allocation $\overline{\mu}$, its ordinary specification induces in problem Π the allocation $\rho(\overline{\mu})$.

In the clone specifications, in the context of a diagram we will speak of each agent and clone pointing to an item and each item pointing to an agent or a clone. This will give rise to what we call "blocks," "chains," and "cycles". We elaborate on these notions below but to ease reading for now note that a cycle will consist of blocks pointing to one another or of chains pointing to one another but the notions of a block and a chain are not the same.

Below we elaborate on the notions of a block and a cycle and we also introduce some related concepts.

- **block, source agent, sink hi-item.**

  A *block* is an ordered list $a \to x$ or $a \to o^1 \to c^1 \to \cdots o^q \to c^q \to x$ (pointers added) such that:

– a is an agent and x is an hi-item;

– for each s, o^s is an occupied house and c^s is the clone who owns it;

– the members of the list point to one another in succession (as indicated).

  We refer to a and x as the *source agent* and the *sink hi-item* of the block. When a block is "executed," it means each agent and clone in the block is assigned the item to which she points and then the agents and clones in the block are removed together with their assignments.

- **cycle**

  A *cycle* is an ordered list $x^1 \to a^1 \to \cdots \to x^q \to a^q \to x^1$ (pointers added) such that:

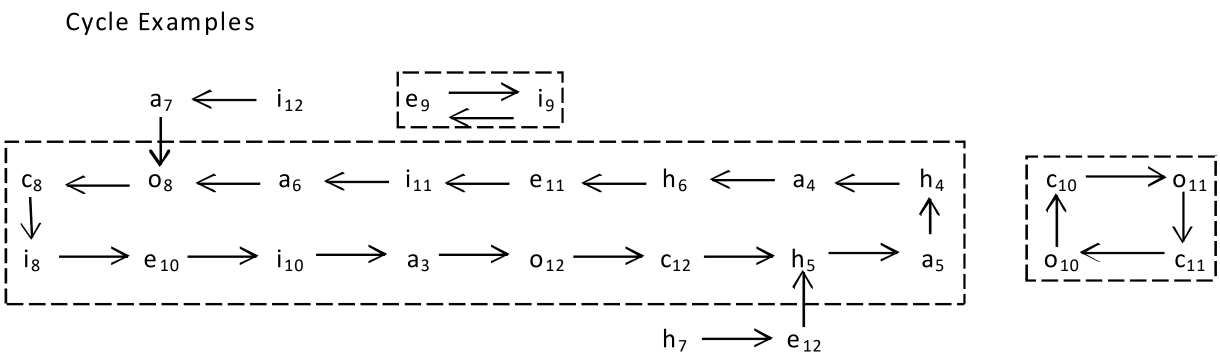
– $\{a^s\}_{s=1}^q \subseteq A \cup C$ and $\{x^s\}_{s=1}^q \subseteq H \cup I$;

– for each s, if x^s is an occupied house, then a^s is the clone who owns it;

– the members of the list point to one another in succession (as indicated).[9]

  When a cycle is "executed," it means each agent and clone in the cycle is assigned the item to which she points and then the agents and clones in the cycle are removed together with their assigned items.

As an illustration, the following diagram indicates three cycles in dashed boxes.



In a cycle there may not be any agent. In the above diagram, for instance, the cycle on the right involves only clones and occupied houses. If there are one or more agents in a cycle, however, this cycle consists of blocks: The sink hi-items of the blocks point to the source agents of the blocks in succession and hence the cycle forms. In the above diagram, for instance, the small cycle on the left consists of the block $e_9 \to i_9$ (i_9 points to e_9 and hence the cycle forms) and the large cycle consists of the blocks: $a_5 \to h_4$, $a_4 \to h_6 \to e_{11} \to i_{11}$, $a_6 \to o_8 \to c_8 \to i_8$,

[9]Note that the last agent or clone in the list, a^q, points to the first item in the list, x^1. Also, notice that while the "cycles" mentioned in the description of the TTC(v) mechanism in Section 2 involved existing tenants and occupied houses, here a cycle involves agents, clones and items (i.e., there are now also newcomers, clones, vacant houses and inheritance rights).

$e_{10} \to i_{10}$, $a_3 \to o_{12} \to c_{12} \to h_5$ (h_4 points to a_4, i_{11} points to a_6, and so on, and hence the cycle forms).[10]

In our clone specifications, in the context of a diagram we will speak of agents pointing to their "favorite remaining items," which we define next.

- **favorite remaining item.**

  The *favorite remaining item* of a newcomer is her favorite house among those remaining (i.e., among the houses that are still in the diagram).

  The *favorite remaining item* of an existing tenant e_s (of a clone c_s) is:

  - i_s, if c_s (if e_s) has been assigned a house earlier;
  - her favorite house among those remaining, if otherwise.

We are now ready to introduce the clone specifications. The clone specification of Y-I(f) proceeds as follows:

Y-I (f). *Consider a diagram consisting of agents, clones and items.*

> *Let a be the agent ordered first in f. Let a point to her favorite remaining item, say x. If x is an hi-item, this results in a block: $a \to x$. Otherwise, x is an occupied house owned by a clone, say c. In that case, let x point to c and c point to her favorite remaining item, say x'. If x' is an hi-item, this results in a block: $a \to x \to c \to x'$. Otherwise, x' is an occupied house owned by a clone, say c'. In that case, let x' point to c' and c' point to her favorite remaining item, say x''. Proceed similarly until either a block forms with source agent a, or a cycle forms consisting of a number of clones and occupied houses. Execute this block or cycle. Then reiterate the process starting with the agent ordered at the top of the remainder of f.*[11]

The algorithm terminates when every agent and clone has been assigned an item (or, equivalently, when the remainder of the priority order is empty).

Notice that the key distinction in the above clone specification of a Y-I mechanism is the following: For an ec-pair e_s and c_s, each "attempts" to be assigned the best house that she can until one succeeds. Then in succeeding steps the remaining one simply points to i_s. Since no one else ever points to i_s, she receives it eventually. That is why the clone specification always induces a clone allocation as we defined it. The clone and the ordinary specifications of a Y-I mechanism turn out to be equivalent: i.e., when the former induces a clone allocation $\overline{\mu}$, the latter induces the (ordinary) allocation $\rho(\overline{\mu})$. To see this, consider the following illustrative cases that may arise while running the clone specification:

- A block forms involving no clones and whose sink hi-item is a vacant house; e.g., $a_1 \to h_3$. Then a_1 is assigned h_3. For the ordinary specification this means the following: When it is her turn, a_1 requests h_3 and she is assigned h_3.

[10] For a cycle that involves one or more agents, note that executing this cycle is the same as executing the blocks in the cycle: Either way, the same assignments are made, and hence, the same agents, clones and items are removed. This property turns out to be useful in our arguments later on (in showing Claim 1 in the Appendix).

[11] Note that if a is not assigned a house, the agent ordered at the top is still a; otherwise, the agent ordered at the top is the one that comes after a in f.

– A block forms involving no clones and whose sink hi-item is an inheritance right; e.g., $e_{11} \to i_{11}$. Then e_{11} is assigned i_{11}. For the ordinary specification this means that e_{11} has already been assigned a house (so she is not in the remainder of the priority order).

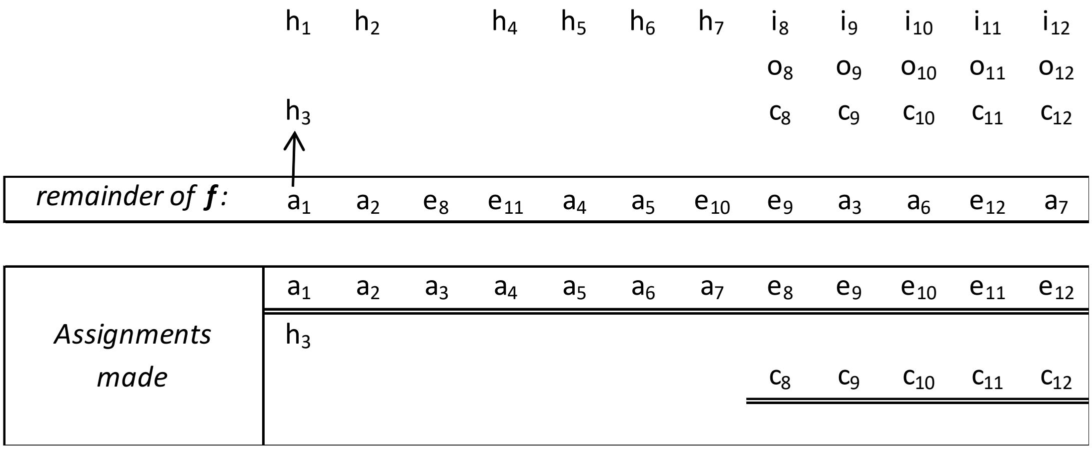
– A block forms that involves clones and whose sink hi-item is a vacant house; e.g., $a_3 \to o_{12} \to c_{12} \to h_5$. Then a_3 is assigned o_{12} and c_{12} is assigned h_5. For the ordinary specification this means the following: When it is her turn, a_3 requests o_{12}. Thus e_{12} moves to the top of the remainder of the priority order. Then e_{12} requests h_5 and she is assigned h_5. Then a_3 again moves to the top of the remainder of the priority order. Then a_3 again requests o_{12} but this time she receives it.

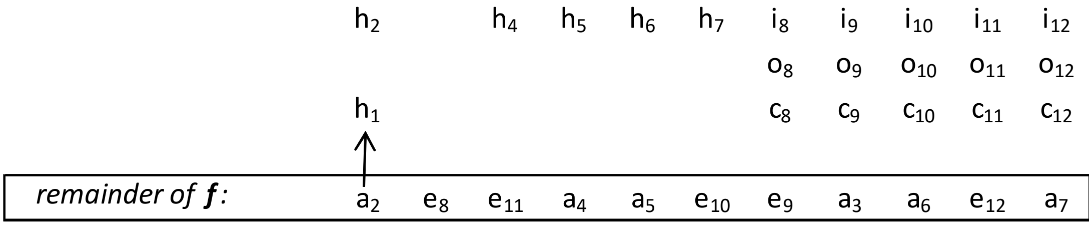
– A block forms that involves clones and whose sink hi-item is an inheritance right; e.g., $a_6 \to o_8 \to c_8 \to i_8$. Then a_6 is assigned o_8 and c_8 is assigned i_8. For the ordinary specification this means the following: When it is her turn, a_6 requests o_8 and she is assigned o_8 (because it turns out that e_8 has been assigned a house earlier).

– A cycle forms consisting of clones and occupied houses; e.g., $c_{10} \to o_{11} \to c_{11} \to o_{10} \to c_{10}$. Then c_{10} is assigned o_{11} and c_{11} is assigned o_{10}. For the ordinary specification this means the following: When it is the turn of e_{10} (or e_{11}), a loop forms in which e_{10} and e_{11} successively request the occupied houses of one another. Then e_{10} is assigned o_{11} and e_{11} is assigned o_{10}.

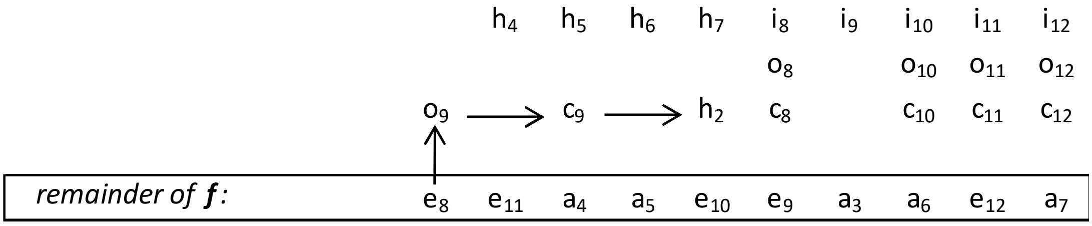
The following example illustrates the workings of the clone specification of a Y-I mechanism.

Example 2 **(workings of a Y-I mechanism)** *Consider the clone problem presented in Example 1. Let priority order f be such that agents are ordered as follows: $a_1, a_2, e_8, e_{11}, a_4, a_5, e_{10}, e_9, a_3, a_6, e_{12}, a_7$. The following series of figures illustrate the workings of the mechanism Y-I(f): Agents successively move to the top of the remainder of the priority order, blocks and cycles form, and then agents are assigned items by executing these blocks and cycles. The table at the very end presents the clone allocation induced.*

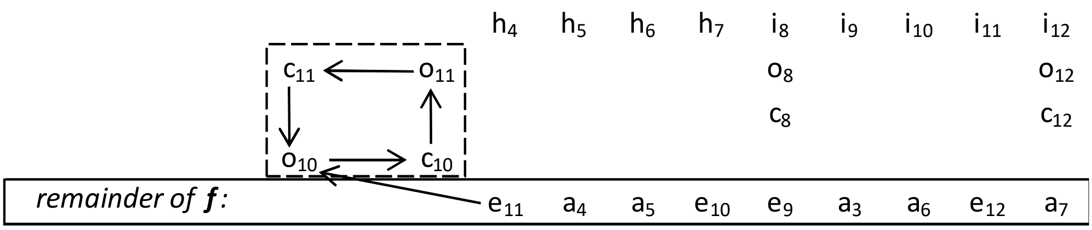




Assignments made	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
	h_3	h_1										
								c_8	c_9	c_{10}	c_{11}	c_{12}



Assignments made	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
	h_3	h_1						o_9				
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2			



Assignments made	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
	h_3	h_1						o_9				
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2	o_{11}	o_{10}	

	h_4	h_5	h_6	h_7	i_8	i_9	i_{10}		i_{12}
					o_8				o_{12}
	i_{11}				c_8				c_{12}
*remainder of **f**:*	e_{11}	a_4	a_5	e_{10}	e_9	a_3	a_6	e_{12}	a_7

Assignments made	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
	h_3	h_1						o_9			i_{11}	
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2	o_{11}	o_{10}	

	h_4	h_5	h_7	i_8	i_9	i_{10}		i_{12}
				o_8				o_{12}
		h_6		c_8				c_{12}
*remainder of **f**:*	a_4	a_5	e_{10}	e_9	a_3	a_6	e_{12}	a_7

Assignments made	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
	h_3	h_1		h_6				o_9			i_{11}	
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2	o_{11}	o_{10}	

	h_5	h_7	i_8	i_9	i_{10}		i_{12}
			o_8				o_{12}
	h_4		c_8				c_{12}
*remainder of **f**:*	a_5	e_{10}	e_9	a_3	a_6	e_{12}	a_7

Assignments made	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
	h_3	h_1		h_6	h_4			o_9			i_{11}	
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2	o_{11}	o_{10}	

					h_5		h_7	i_8	i_9			i_{12}
								o_8				o_{12}
							i_{10}	c_8				c_{12}
							↑					
remainder of ***f*** *:*							e_{10}	e_9	a_3	a_6	e_{12}	a_7

Assignments made	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
	h_3	h_1		h_6	h_4			o_9		i_{10}	i_{11}	
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2	o_{11}	o_{10}	

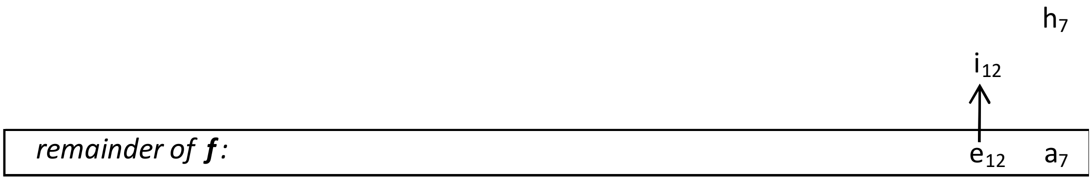
					h_5					h_7	i_8	i_{12}
											o_8	o_{12}
								i_9			c_8	c_{12}
								↑				
remainder of ***f*** *:*								e_9	a_3	a_6	e_{12}	a_7

Assignments made	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
	h_3	h_1		h_6	h_4			o_9	i_9	i_{10}	i_{11}	
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2	o_{11}	o_{10}	

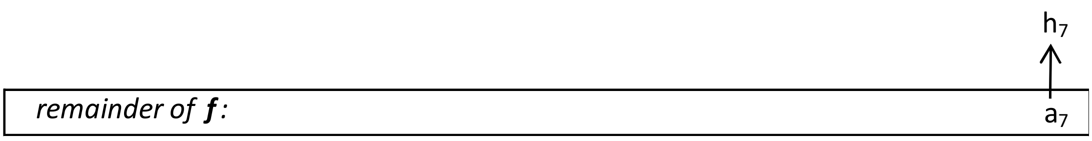
										h_7	i_8	i_{12}
					h_5	←	c_{12}	←	o_{12}		o_8	
											c_8	
									↑			
remainder of ***f*** *:*									a_3	a_6	e_{12}	a_7

Assignments made	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
	h_3	h_1	o_{12}	h_6	h_4			o_9	i_9	i_{10}	i_{11}	
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2	o_{11}	o_{10}	h_5

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
Assignments made	h_3	h_1	o_{12}	h_6	h_4	o_8		o_9	i_9	i_{10}	i_{11}	
								c_8	c_9	c_{10}	c_{11}	c_{12}
								i_8	h_2	o_{11}	o_{10}	h_5



	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
Assignments made	h_3	h_1	o_{12}	h_6	h_4	o_8		o_9	i_9	i_{10}	i_{11}	i_{12}
								c_8	c_9	c_{10}	c_{11}	c_{12}
								i_8	h_2	o_{11}	o_{10}	h_5



	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
Assignments made	h_3	h_1	o_{12}	h_6	h_4	o_8	h_7	o_9	i_9	i_{10}	i_{11}	i_{12}
								c_8	c_9	c_{10}	c_{11}	c_{12}
								i_8	h_2	o_{11}	o_{10}	h_5

$\Diamond$

The clone specification of TTC(v) proceeds as follows:

TTC (v). *Consider a diagram consisting of agents, clones and items.*

Step t ($t \geq 1$, t odd): Let each clone point to its favorite remaining item and let each occupied house point to its owner (a clone). If there are no cycles, proceed to Step $t+1$. If there are cycles, execute them and repeat Step t until no cycle arises. Then proceed to Step $t+1$.

Step t ($t \geq 2$, t even): Let each agent and clone point to its favorite remaining item and let each occupied house point to its owner (a clone). Also, let each hi-item point to the agent who owns it under v. Execute the resulting cycles and then proceed to Step $t+1$.

The algorithm terminates when every agent and clone is assigned an item. It turns out that this happens at an even step.[12]

Notice that the key distinction in the clone specification of an augmented TTC mechanism is the following: For an ec-pair e_s and c_s, each "attempts" to be assigned the best house that she can until one succeeds. Then in succeeding steps the remaining one simply points to i_s. Since no one else ever points to i_s she receives it eventually. That is why the clone specification always induces a clone allocation as we defined it. The clone and the ordinary specifications of an augmented TTC mechanism turn out to be equivalent: i.e., when the former induces a clone allocation $\overline{\mu}$, the latter induces the (ordinary) allocation $\rho(\overline{\mu})$. To see this, consider the following illustrative cases that may arise while running the clone specification:

- As part of a cycle, a block forms involving no clones and whose sink hi-item is a vacant house; e.g., $a_1 \to h_3$. Then a_1 is assigned h_3. In the ordinary specification, in the corresponding cycle a_1 points to h_3. Then a_1 is assigned h_3.

- As part of a cycle, a block forms involving no clones and whose sink hi-item is an inheritance right; e.g., $e_{11} \to i_{11}$. Then e_{11} is assigned i_{11}. In the ordinary specification, this corresponds to e_{11} having been assigned a house earlier.

- As part of a cycle, a block forms that involves clones and whose sink hi-item is a vacant house; e.g., $a_3 \to o_{12} \to c_{12} \to h_5$. Then a_3 is assigned o_{12} and c_{12} is assigned h_5. In the ordinary specification, in the corresponding cycle a_3 points to o_{12}, o_{12} points to e_{12}, and e_{12} points to h_5. Then a_3 is assigned o_{12} and e_{12} is assigned h_5.

- As part of a cycle, a block forms that involves clones and whose sink hi-item is an inheritance right; e.g., $a_6 \to o_8 \to c_8 \to i_8$. Then a_6 is assigned o_8 and c_8 is assigned i_8. In the ordinary specification, in the corresponding cycle a_6 points to o_8 and o_8 points to an inheritor of e_8 (i.e., the inheritor of e_8 or the inheritor of inheritor of e_8 and so on). (Because it turns that e_8 has been assigned a house earlier and hence o_8 is now owned by an inheritor.) Then a_6 is assigned o_8.

- A cycle forms consisting of clones and occupied houses; e.g., $c_{10} \to o_{11} \to c_{11} \to o_{10} \to c_{10}$. Then c_{10} is assigned o_{11} and c_{11} is assigned o_{10}. In the ordinary specification, in the corresponding cycle e_{10} and e_{11} point to one another's occupied houses. Then e_{10} is assigned o_{11} and e_{11} is assigned o_{10}.

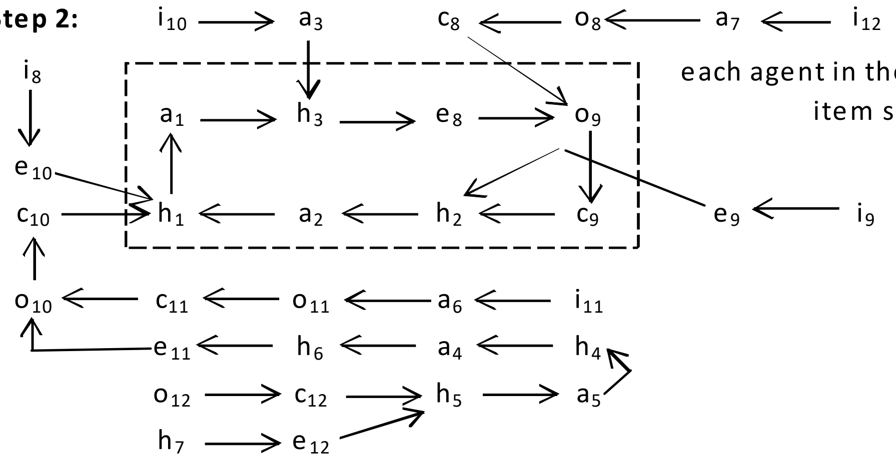
The following example illustrates the workings of the clone specification of an augmented TTC mechanism.

EXAMPLE 3 **(workings of an augmented TTC mechanism)** *Consider the clone problem presented in Example 1. Let augmenting function* v *be such that:* $v(a_1) = h_1$, $v(a_2) = h_2$, $v(a_3) = i_{10}$, $v(a_4) = h_4$, $v(a_5) = h_5$, $v(a_6) = i_{11}$, $v(a_7) = i_{12}$, $v(e_8) = h_3$, $v(e_9) = i_9$, $v(e_{10}) = i_8$, $v(e_{11}) = h_6$, $v(e_{12}) = h_7$. *The following series of figures illustrate the workings of the mechanism TTC(v) and the assignments made at successive steps. (For simplicity, at odd steps we present only the cycles formed, if any.)*

[12] The assignments cannot be finalized at an odd step for the following reason: At odd steps, the clones are are assigned occupied houses. Thus, if c_s is assigned an occupied house at Step t, it must be that e_s is assigned i_s at a subsequent even step.

Step 1: No cycles; no assignments made.

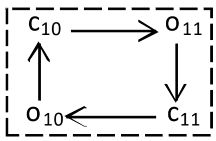
Step 2:



each agent in the cycle is assigned the item she points to

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
Assignments made	h_3	h_1						o_9				
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2			

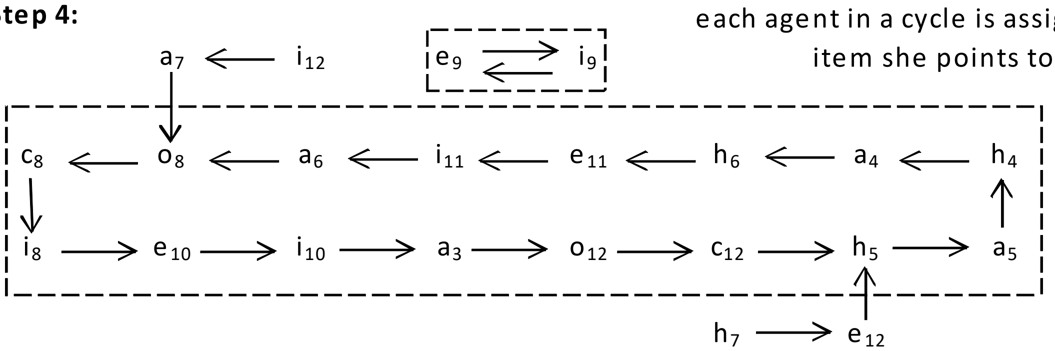
Step 3:



each agent in the cycle is assigned the item she points to

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
Assignments made	h_3	h_1						o_9				
								c_8	c_9	c_{10}	c_{11}	c_{12}
									h_2	o_{11}	o_{10}	

Step 4:

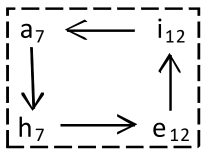


each agent in a cycle is assigned the item she points to

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
Assignments made	h_3	h_1	o_{12}	h_6	h_4	o_8		o_9	i_9	i_{10}	i_{11}	
								c_8	c_9	c_{10}	c_{11}	c_{12}
								i_8	h_2	o_{11}	o_{10}	h_5

Step 5: No cycles; no assignments made.

Step 6:



each agent in the cycle is assigned the item she points to

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	e_8	e_9	e_{10}	e_{11}	e_{12}
Assignments made	h_3	h_1	o_{12}	h_6	h_4	o_8	h_7	o_9	i_9	i_{10}	i_{11}	i_{12}
								c_8	c_9	c_{10}	c_{11}	c_{12}
								i_8	h_2	o_{11}	o_{10}	h_5

$\Diamond$

5 The Chain Order

In the remainder of the paper we study only the clone problem. Therefore, when we speak of the Y-I and the augmented TTC mechanisms, it is understood that we mean their clone specifications. In this section, on the basis of the workings of an augmented TTC mechanism, we define what we call a "chain order." We denote the chain order corresponding to the mechanism TTC(v) by $ch(v)$. A chain order is simply a priority order (i.e., $ch(v) \in \mathcal{F}$). It helps us link the mechanisms TTC(v) and Y-I($ch(v)$): As it turns out, they are equivalent (Lemma 1).

On the basis of the workings of TTC(v): Let $A^t(v), C^t(v), H^t(v), I^t(v)$ be, respectively, the sets of agents, clones, houses, and inheritance rights that are assigned at Step t. For t odd, note that $H^t(v) \subseteq H_O$ and $A^t(v) = I^t(v) = \emptyset$.

Under TTC(v), for $t \geq 2$ and t even, consider Steps t and $t+2$. The cycles that arise at Step $t+2$ consist of the members of the set $A^{t+2}(v) \cup C^{t+2}(v) \cup H^{t+2}(v) \cup I^{t+2}(v)$. As it turns out, at Step t the members of this set form what we call "chains". Below we elaborate on the notion of a chain and we also introduce some related concepts.

- **chain, the tail and the head of a chain, the agent-head and the agent-tail of a chain**

  A *chain* at Step t is an ordered list $x^1 \to a^1 \to \cdots \to x^q \to a^q$ (pointers added) such that:

  – $\{a^s\}_{s=1}^q \subseteq A^{t+2}(v) \cup C^{t+2}(v)$ and $\{x^s\}_{s=1}^q \subseteq H^{t+2}(v) \cup I^{t+2}(v)$;

  – for each s, if x^s is an occupied house then a^s is the clone who owns it;

  – for each s, if x^s is an hi-item then a^s is the agent who owns it under v;

  – the members of the list point to one another in succession (as indicated);

  – there does not exist $a \in A^{t+2}(v) \cup C^{t+2}(v)$ such that a points to x^1 at Step t;

  – and at Step t, a^q points to a house in $H^t(v) \cup H^{t+1}(v)$.

  We refer to x^1 as the *tail* and a^q as the *head* of the chain. Note that the reason the members of this chain are assigned at Step $t+2$, and not at Step t, is the following: At Step t, a^q points to a house which is assigned to someone else at Step t or $t+1$.[13] Note that except for the head of the chain, every other agent and clone in the chain is assigned at Step $t+2$ the item to which it points at Step t.

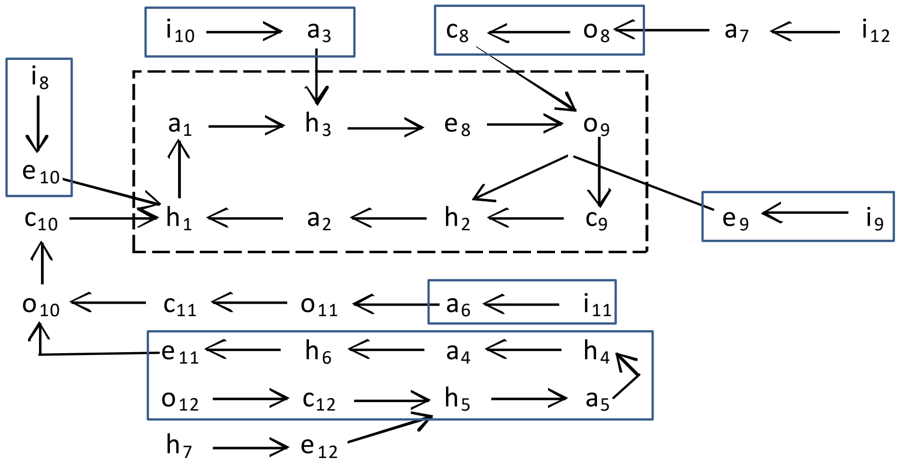
  Although "blocks" and "chains" are indicated using a similar representation (as an ordered list of agents, clones and items with pointers), note that they are different mathematical constructs: In a block there is exactly one agent. In a chain there may be one or more agents, or even none. If a chain involves one or more agents, we refer to the first agent in the ordered list above as the *agent-tail* of the chain and last the one as the *agent-head* of the chain. Note that if in a chain there is exactly one agent then the agent-tail and the agent-head of the chain are the same. Also, above if a^q is an agent (not a clone), then a^q is both the head and the agent-head of the chain. The agent-tail and the agent-head notions become useful below when we define the "chain order."

  The chains that arise at Step t are disjoint by definition. At Step $t+2$, the heads of these chains point to the tails of one another and thereby form the cycles. As an illustration, the figures below show how the chains that arise at Step 2 in Example 3, shown in solid boxes,

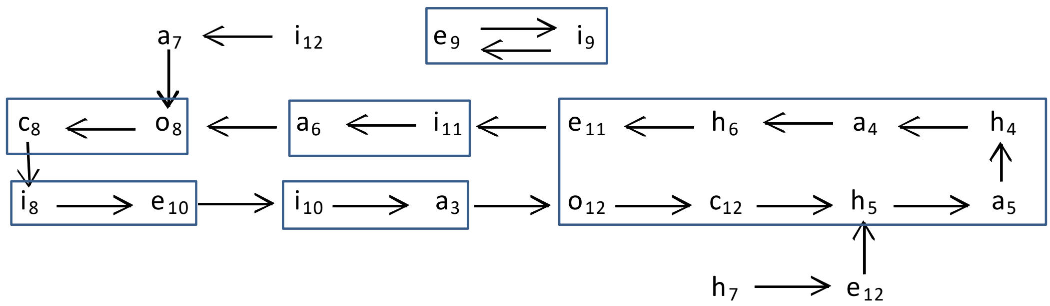
[13] Note that at Step t, a^q cannot be pointing to an inheritance right $i_s \in I^t(v)$: If it were so, it would mean that i_s is assigned to someone else at Step t. But under TTC (v), no agent or clone ever points to i_s except the one who is eventually assigned it (which is e_s or c_s).

form the cycles at Step 4.

Chains at Step 2



How chains at Step 2 form cycles at Step 4



We are now ready to introduce our "chain order."

- **Chain order** $ch(v)$

  For $v \in \mathcal{V}$, on the basis of the execution of the mechanism TTC(v), we construct the *chain order* $ch(v) \in \mathcal{F}$ according to the following three rules:

  - **chain order rule 1:** *Order agents in* $A^t(v)$ *before agents in* $A^{t+2}(v)$ for all $t \geq 2$, t even.
  - **chain order rule 2:** *Order agents in* $A^2(v)$ *in order of the indices of the hi-items that they own under* v.
  - **chain order rule 3:** *Order agents in* $A^{t+2}(v)$ $(t \geq 2$, t *even) by looking at the chains that are arise at Step* t*: Agents in a chain are ordered from its agent-head to its agent-tail, successively. Agents in different chains are ordered looking at the indices of the hi-items pointing to the agent-tails of the chains: The agents in the chain where this index takes the smallest value are ordered first; the agents in the chain where this index takes the second smallest value are ordered next, and so on.*

The following example illustrates the construction of the chain order.

EXAMPLE 4 **(construction of the chain order)** *Consider the execution of TTC*(v) *presented in Example 3. We find that:* $A^2(v) = \{a_1, a_2, e_8\}$, $A^4(v) = \{a_3, a_4, a_5, a_6, e_9, e_{10}, e_{11}\}$, *and* $A^6(v) = \{a_7, e_{12}\}$.

By chain order rule 1, we order the agents in $A^2(v)$ before those in $A^4(v)$ and the agents in $A^4(v)$ before those in $A^6(v)$.

By chain order rule 2, the agents in $A^2(v)$ are ordered as a_1, a_2, e_8. (Note that $v(a_1) = h_1$, $v(a_2) = h_2$, $v(e_8) = h_3$; and hence the indices of the hi-items under v of a_1, a_2, e_8 are 1, 2, 3.)

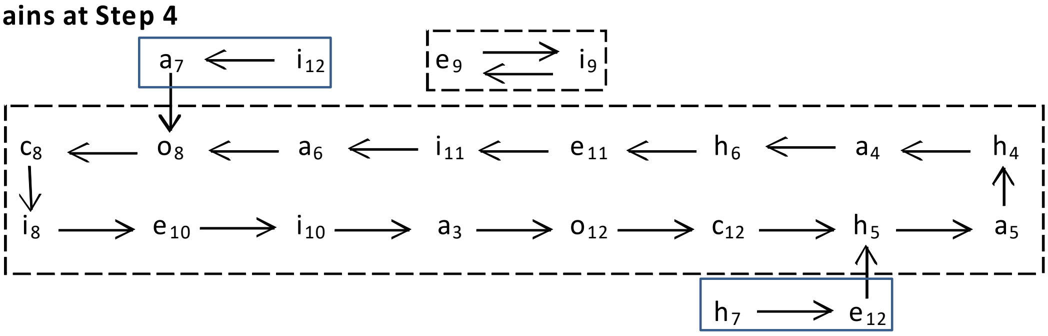
Consider now the agents in $A^4(v)$. In the above two figures we have shown how the chains that arise at Step 2 (shown in solid boxes) form the cycles at Step 4. As it can be seen there, the following chains arise at Step 2:[14]

- $o_{12} \to c_{12} \to h_5 \to a_5 \to h_4 \to a_4 \to h_6 \to e_{11}$
- $i_{11} \to a_6$
- $i_8 \to e_{10}$
- $i_{10} \to a_3$
- $i_9 \to e_9$

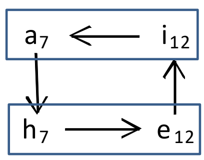
In the chains listed above, in order: The agent-tails are $a_5, a_6, e_{10}, a_3, e_9$; the hi-items that point to them are $h_5, i_{11}, i_8, i_{10}, i_9$; the indices of these hi-items are 5, 11, 8, 10, 9. Looking at the indices, by chain order rule 3 we order the agents in the above chains as: $e_{11}, a_4, a_5, e_{10}, e_9, a_3, a_6$.[15]

Consider now the agents in $A^6(v)$. The following figures show how the chains that arise at Step 4 (shown in solid boxes) form the only cycle at Step 6.

Chains at Step 4



How chains at Step 4 form cycles at Step 6



$\diamond$

As can be seen, the following chains arise at Step 4: $i_{12} \to a_7$ and $h_7 \to e_{12}$. In order: The agent-tails are a_7 and e_{12}; the hi-items that point to them are i_{12} and h_7; the indices of these

[14] The chain $o_8 \to c_8$ arises, too, but it involves no agent and hence is not needed in the construction of the chain order.

[15] Note that the agents in the chain $o_{12} \to c_{12} \to h_5 \to a_5 \to h_4 \to a_4 \to h_6 \to e_{11}$ are ordered as e_{11}, a_4, a_5, because by chain order rule 3 the agents in a chain are ordered from its agent-head to its agent-tail.

hi-items are 12 and 7. Looking at the indices, by chain order rule 3 we order the agents in these chains as: e_{12}, a_7.

Therefore, we obtain that the chain order $ch(v)$ *is as follows:*

$$ch(v): \quad \underbrace{a_1, a_2, e_8}_{A^2(v)} \quad \underbrace{e_{11}, a_4, a_5, e_{10}, e_9, a_3, a_6}_{A^4(v)} \quad \underbrace{e_{12}, a_7}_{A^6(v)}$$

Notice that the same clone allocation is induced by the Y-I mechanism in Example 2 and the augmented TTC mechanism in Example 3. This is no coincidence. The chain order induced by the execution of the augmented TTC mechanism in Example 3 (as described in Example 4) is precisely the same as the priority order of the Y-I mechanism in Example 2. And as it turns out, for any augmenting function v, the mechanisms TTC(v) and Y-I$(ch(v))$ induce the same clone allocation (i.e., the two mechanisms are equivalent). This is because by construction of the chain order $ch(v)$:

– The cycles that arise at odd steps when TTC(v) is executed also arise when Y-I$(ch(v))$ is executed.[16]

– The cycles that arise at even steps when TTC(v) is executed consist of the blocks that arise when Y-I(f) is executed.

We present this observation as a formal lemma.

LEMMA 1 *The mechanisms TTC*(v) *and Y-I*$(ch(v))$ *are equivalent.*

We conclude this section by presenting three more lemmas: Lemma 2 helps us show Lemma 3, and Lemma 3 helps us show Lemma 4. The proofs of Lemmas 2 and 3 are given in the Appendix for the interested reader. The theorems presented in Section 6 are proved using Lemmas 1 and 4.

LEMMA 2 *For two augmenting functions* v *and* v' *such that* $ch(v) = ch(v')$, $A^t(v) = A^t(v')$ *for all* $t \geq 2$, t *even.*

LEMMA 3 *Given a chain order* $ch(v)$, *we can uniquely identify* v.

LEMMA 4 *The chain order mapping* $ch : \mathcal{V} \to \mathcal{F}$ *is a bijection.*

Proof. By Lemma 3, the mapping ch is injective. Since $|\mathcal{V}| = |\mathcal{F}| = n!$, the mapping ch must also be surjective. Therefore, the mapping ch is a bijection. ■

[16] Note that these cycles involve only clones and occupied houses. There is, however, the following difference regarding the timing of when these cycles are executed: Under TTC(v), given the assignments made earlier, at an odd step all cycles that involve exclusively clones and occupied houses are preemptively identified and executed. Under Y-I$(ch(v))$ these cycles are executed on an "as they arise" basis, however. This difference is nonetheless inconsequential.

6 Results

In this section we study four random mechanisms. They are derived from the classes of the Y-I and the augmented TTC mechanisms. We present the main results of our paper which show how these random mechanisms are related.

First, we consider two random mechanisms whose random components treat existing tenants and newcomers alike.

random Y-I. Draw a priority order $f \in \mathcal{F}$ uniformly at random. Then execute Y-I(f).

random augmented TTC. Draw an augmenting function $v \in \mathcal{V}$ uniformly at random. Then execute TTC(v).

The random components in the above two mechanisms are the choices of the priority order f and the augmenting function v. Since f and v are chosen uniformly at random, the newcomers and existing tenants are treated alike under the random components of these two mechanisms. As it turns out, these two random mechanisms are equivalent.

THEOREM 1 *The mechanisms random Y-I and random augmented TTC are equivalent.*

Proof. This immediately follows from Lemma 1 and Lemma 4. ■

In their paper, Abdulkadiroğlu and Sönmez [1] study the "house allocation problem," which is the special case of our problem in which there are no existing tenants. In this context, they show that the mechanisms "random priority"[5] and "core from random endowments" are equivalent. The random Y-I and the random augmented TTC mechanisms reduce to these two mechanisms when there are no existing tenants. Therefore, their equivalence result follows as a corollary of our Theorem 1.

COROLLARY 1 **(Abdulkadiroğlu and Sönmez [1])** *In a house allocation problem, the mechanisms "random priority" and "core from random endowments" are equivalent.*

We consider next two random mechanisms whose random components discriminate against a subset $E \subseteq A_E$ of existing tenants. Note that this is the same as favoring the set of agents in $A \setminus E$. Without loss of generality, let $E = \{e_q, e_{q+1}, \cdots, e_n\}$. (To achieve this configuration we can relabel the existing tenants as necessary.)

Let $\mathcal{F}^E \subseteq \mathcal{F}$ be the subset of priority orders such that $e_q, e_{q+1}, \cdots, e_n$ are ordered at the bottom in a given fixed order. Let $\mathcal{V}^E \subseteq \mathcal{V}$ be the subset of augmenting functions under which $e_q, e_{q+1}, \cdots, e_n$ are assigned the inheritance rights $i_q, i_{q+1}, \cdots, i_n$ in a given fixed order. The random components of the following two random mechanisms discriminate against the existing tenants in E.[17]

$E-$**discriminating random Y-I.** Draw a priority order $f \in \mathcal{F}^E$ uniformly at random. Then execute Y-I(f).

[17]Note that the sets $\mathcal{F}^E$ and $\mathcal{V}^E$ are not fully specified: For $\mathcal{F}^E$, the fixed order in which $e_q, e_{q+1}, \cdots, e_n$ are placed at the bottom is not given. And for $\mathcal{V}^E$, the fixed order in which $e_q, e_{q+1}, \cdots, e_n$ are assigned $i_q, i_{q+1}, \cdots, i_n$ is not given. These fixed orders turn out to be inconsequential, however. To save space we leave the verification of this to the interested reader.

$E-$**discriminating random augmented TTC.** Draw an augmenting function $v \in \mathcal{V}^E$ uniformly at random. Then execute TTC(v).

The discrimination against the existing tenants in E in the above two random mechanisms is clear: Under $E-$discriminating random Y-I, when the priority order f is chosen they are always placed at the bottom and hence put at a disadvantage. Under $E-$discriminating random augmented TTC, when the augmenting function v is chosen they never get a chance to receive vacant houses and hence put at a disadvantage in the subsequent trade protocol. As an illustration, imagine the case where every agent prefers any vacant house to any occupied house. Then, under these two random mechanisms, the coveted vacant houses are never assigned to the existing tenants in E. As it turns out, these two random mechanisms are also equivalent.

THEOREM 2 *The mechanisms $E-$discriminating random Y-I and $E-$discriminating random augmented TTC are equivalent.*

In a study related to ours, Sönmez and Ünver [19] study our problem and show the equivalence of two random mechanisms. The random mechanisms that they study turn out to be the special cases of our $E-$discriminating random mechanisms where the discrimination is against all existing tenants (i.e., $E = A_E$). Therefore, their equivalence result follows as a corollary of our Theorem 2.

COROLLARY 2 **(Sönmez and Ünver [19])** *The mechanisms $E-$discriminating random Y-I and $E-$discriminating random augmented TTC are equivalent when $E = A_E$.*

Indeed, our Theorem 1 (and hence Corollary 1) also follows from Theorem 2 (when we set $E = \emptyset$). Thus, the main theoretical contribution of our paper is an equivalence result (Theorem 2) that subsumes and generalizes the above equivalence results. The remainder of this section presents the proof of Theorem 2.

Proof of Theorem 2.

We prove the theorem using a "trick" and then by applying Lemma 1. The trick is adding to our problem an existing tenant whose occupied house is everyone's last choice. We denote our "extended problem" by $\Pi^+ : \left\langle A_N, H_V, A_E^+, H_O^+, \succ^+ \right\rangle$, where:

- A_N and H_V are as in Π.
- $A_E^+ = A_E \cup \{e_{n+1}\}$ and $H_O^+ = H_O \cup \{o_{n+1}\}$.
- For each $a \in A_N \cup A_E$, $\succ_a^+$ is such that o_{n+1} is the least preferable house and the preferences over the houses in $H_V \cup H_O$ are the same as in $\succ_a$.
- For e_{n+1}, a house with a smaller index is more preferable (i.e., h_1 is the most and o_{n+1} is the least preferable house).

We will study the clone problems $Cl\{\Pi\}$ and $Cl\{\Pi^+\}$. Recall that in the clone problem $Cl\{\Pi\}$, we use A, C, H, I to denote the sets of agents, clones, houses and inheritance rights. In the clone problem $Cl\{\Pi^+\}$ we will use A^+, C^+, H^+, I^+ to denote the corresponding sets (i.e., $A^+ = A \cup \{e_{n+1}\}$, $C^+ = A \cup \{c_{n+1}\}$, $H^+ = H \cup \{o_{n+1}\}$, and $I^+ = I \cup \{i_{n+1}\}$).

Let φ and φ^+ be two random mechanisms, defined in the contexts of the clone problems $Cl\{\Pi\}$ and $Cl\{\Pi^+\}$, respectively. With some abuse of language, we say that the mechanisms φ and φ^+ are *equivalent* if under φ and φ^+ the probabilistic assignments of the agents and clones in $A \cup C$ are the same. Our proof follows the following outline:

– In the context of the clone problem $Cl\left\{\Pi^{+}\right\}$ we introduce two random mechanisms, φ_1^+ and φ_2^+.

– In the sense defined above, we show that φ_1^+ is equivalent to the $E-$discriminating random Y-I mechanism and φ_2^+ is equivalent to the $E-$discriminating random augmented TTC mechanism.

– In the context of the clone problem $Cl\left\{\Pi^{+}\right\}$ we show that the mechanisms φ_1^+ and φ_2^+ are equivalent.

  This will prove that our two $E-$discriminating random mechanisms are equivalent.

For each $f \in \mathcal{F}^E$, we define the priority order $f^+ : A^+ \to \{1, 2, \cdots, n+1\}$ as follows:

– $f^+(a) = f(a)$ for $a \in A \setminus E$.

– $f^+(e_{n+1}) = q$.

– $f^+(e_s) = f(e_s) + 1$ for $e_s \in E$.

  That is, f^+ is obtained by inserting e_{n+1} in f right above $e_q, e_{q+1}, \cdots, e_n$, however they are ordered in f.

When we execute Y-I(f^+) in the clone problem $Cl\left\{\Pi^{+}\right\}$, by the time e_{n+1} moves to the top of the remainder of the priority order all newcomers and vacant houses are assigned (not necessarily to one another). The only houses remaining are certain occupied houses owned by clones. Since o_{n+1} is everyone's last choice, under Y-I(f^+) the clone c_{n+1} is assigned o_{n+1}, and hence e_{n+1} is assigned i_{n+1}. Since f and f^+ are otherwise the same, the addition of e_{n+1} and o_{n+1} to the problem changes nothing regarding the execution of the Y-I algorithm: The assignments of agents and clones in $A \cup C$ are the same under the mechanisms Y-I(f) and Y-I(f^+). In other words, in the sense defined above the mechanisms Y-I(f) and Y-I(f^+) are equivalent. Therefore, in the sense defined above the $E-$discriminating random Y-I mechanism is equivalent to the following random mechanism:

Let $\mathcal{F}^{E+} : \{f^+ | f \in \mathcal{F}^E\}$.

φ_1^+ : Draw $f^+ \in \mathcal{F}^{E+}$ uniformly at random. Then execute Y-I(f^+) in the clone problem $Cl\left\{\Pi^{+}\right\}$.

For $v \in \mathcal{V}^E$, we define the augmenting function $v^+ : A^+ \to H^+ \cup I^+$ as follows:

– $v^+(a) = v(a)$ for $a \in A \setminus E$.

– $v^+(e_{n+1}) = i_q$.

– $v^+(e_s) = i_{s+1}$ for $e_s \in E$.

When we execute TTC(v^+) in the clone problem $Cl\left\{\Pi^{+}\right\}$, since o_{n+1} is everyone's last choice, c_{n+1} is assigned o_{n+1}, and hence, e_{n+1} is assigned i_{n+1}. Also, under both TTC(v) and TTC(v^+), by construction, the existing tenants $e_q, e_{q+1} \cdots, e_n$ are assigned the inheritance rights $i_q, i_{q+1} \cdots, i_n$, in order. Since v and v^+ are otherwise the same, the assignments of agents and clones in $A \setminus E$ are also the same. That is, under TTC(v^+), c_{n+1} is assigned o_{n+1}, e_{n+1} is assigned i_{n+1}, and the assignments of the agents and clones in $A \cup C$ are exactly the same as under TTC(v). In other words, in the sense defined above the mechanisms TTC(v) and TTC(v^+) are equivalent. Therefore, in the sense defined above the $E-$discriminating random augmented TTC mechanism is equivalent to the following random mechanism:

Let $\mathcal{V}^{E+} : \{v^+ | v \in \mathcal{V}^E\}$.

φ_2^+ : Draw $v^+ \in \mathcal{V}^{E+}$ uniformly at random. Then execute TTC(v^+) in the clone problem $Cl\{\Pi^+\}$.

We complete the proof by showing that in the context of the clone problem $Cl\{\Pi^+\}$ the random mechanisms φ_1^+ and φ_2^+ are equivalent.

Consider the execution of TTC(v^+) in the clone problem $Cl\{\Pi^+\}$. Since the inheritance rights $i_{q+1}, \cdots, i_{n+1}, i_q$ point to the existing tenants $e_q, \cdots, e_{n+1}$ (in order), at some even step they form the following cycle, to be called $Cycle(i_q, i_{n+1})$:

$$\begin{tikzcd}
e_{n+1} \arrow[d] & i_q \arrow[l] & e_q \arrow[l] & \dots \arrow[l] & e_{n-3} \arrow[l] & i_{n-2} \arrow[l] \\
i_{n+1} \arrow[r] & e_n \arrow[r] & i_n \arrow[r] & e_{n-1} \arrow[r] & i_{n-1} \arrow[r] & e_{n-2} \arrow[u]
\end{tikzcd}$$

Since c_{n+1} owns o_{n+1} and o_{n+1} is everyone's last choice, c_{n+1} is assigned o_{n+1} after all other houses are assigned. This happens then at the final odd step, say at Step T (T odd). The algorithm then terminates at Step $T+1$ (an even step), and at Step $T+1$ the cycles formed involve only existing tenants and inheritance rights.

Since e_{n+1} does not point to i_{n+1} until c_{n+1} is assigned o_{n+1}, one of the cycles that arise at Step $T+1$ is $Cycle(i_q, i_{n+1})$. Consider the chains that arise at Step $T-1$ and which form the cycles at Step $T+1$. Some of these chains form $Cycle(i_q, i_{n+1})$ at Step $T+1$. Since the items that point to the agents in $Cycle(i_q, i_{n+1})$ (all existing tenants) have the greatest indices, the chains that form $Cycle(i_q, i_{n+1})$ must be ordered at the very end of the associated chain order $ch(v^+)$ (by chain order rule 3). Also, note that irrespective of how $Cycle(i_q, i_{n+1})$ is broken into chains, these agents are ordered in $ch(v^+)$ at the end as $e_{n+1}, e_q, e_{q+1}, \cdots, e_n$.

Therefore, we obtain that for each $v^+ \in \mathcal{V}^{+E}$, $ch(v^+) \in \mathcal{F}^{+E}$. Note that $\left|\mathcal{V}^{+E}\right| = \left|\mathcal{F}^{+E}\right| = (q-1)!$. Therefore, the chain order mapping induces a bijection from $\mathcal{V}^{+E}$ to $\mathcal{F}^{+E}$. Then we apply by Lemma 1 and obtain that the random mechanisms φ_1^+ and φ_2^+ are equivalent.

Therefore, the mechanisms $E-$discriminating random Y-I and $E-$discriminating random augmented TTC are also equivalent. ■

References

[1] A. Abdulkadiroğlu, Atila and T. Sönmez (1998), "Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems," *Econometrica* 66:3 689-701.

[2] A. Abdulkadiroğlu, Atila and T. Sönmez (1999), "House Allocation with Existing Tenants," *Journal of Economic Theory* 88:2 233-260.

[3] G. Carroll (2014), "A General Equivalence Theorem for Allocation of Indivisible Objects," *Journal of Mathematical Economics* 51 163-177.

[4] Y. Chen and T. Sönmez (2002), "Improving efficiency of on-campus housing: An experimental study," *American Economic Review* 92:5 1669-1686.

[5] Ö. Ekici (2013), "Reclaim-proof Allocation of Indivisible Objects," *Games and Economic Behavior* 81 1-10.

[6] A. Hylland and R. Zeckhauser (1979), "The efficient allocations of individuals to positions," *Journal of Political Economy* 87:2 293-314.

[7] D. E. Knuth (1996), "An Exact Analysis of Stable Allocation," *Journal of Algorithms* 20:2 431-442.

[8] T. Lee and J. Sethuraman (2011), "Equivalence Results in the Allocation of Indivisible Objects: A Unified View," working paper, Columbia University.

[9] Ma J. (1994) "Strategy-proofness and strict core in a market with indivisibilities," *International Journal of Game Theory* 23:1 75-83.

[10] Pápai S. (2000) "Strategyproof assignment by hierarchical exchange," *Econometrica* 68:6 1403-1433.

[11] P. A. Pathak and J. Sethuraman (2011), "Loteries in Student Assignment," *Theoretical Economics* 6:1 1-17.

[12] A. E. Roth (1982), "Incentive Compatibility in a market with indivisible goods," *Economics Letters* 9:2 127-132.

[13] A. E. Roth and A. Postlewaite (1977), "Weak versus strong domination in a market with indivisible goods," *Journal of Mathematical Economics* 4:2 131-137.

[14] A. E. Roth, T. Sönmez, and M.U. Ünver (2004), "Kidney Exchange," *Quarterly Journal of Economics* 119:2 457-488.

[15] A. E. Roth, T. Sönmez, and M.U. Ünver (2005), "Pairwise Kidney Exchange," *Journal of Economic Theory* 125:2 151-188.

[16] A. E. Roth, T. Sönmez, and M.U. Ünver (2007), "Efficient Kidney Exchange: Coincidence of Wants in Markets with Compatibility-Based Preferences," *American Economic Review* 97:3 828-851.

[17] L. Shapley and H. Scarf (1974), "On cores and indivisibility," *Journal of Mathematical Economics* 1:1 23-28.

[18] T. Sönmez and M. U. Ünver (2010), "House allocation with existing tenants: A characterization," *Games and Economic Behavior*.69:2 425-445.

[19] T. Sönmez and M. U. Ünver (2005), "House allocation with existing tenants: An equivalence," *Games and Economic Behavior* 52:1 153-185.

[20] M. U. Ünver (2010), "Dynamic Kidney Exchange," *Review of Economic Studies* 77:1 371-414.

Appendix

In the proofs of Lemmas 2 and 3 we use two observations regarding the execution of an augmented TTC mechanism. We present them below as Claims 1 and 2 and then we proceed with the proofs of Lemmas 2 and 3.

CLAIM 1 *Under TTC(v) ($v \in \mathcal{V}$), suppose that we know $A^2(v), A^4(v), \cdots, A^t(v)$ (t even) but not v. Then we can identify the assignments made up to Step $t+1$ (inclusive).*

Proof.

Consider Step 1 under TTC(v): Note that Step 1 is independent of v. Thus we can identify all the assignments made at Step 1. Proceed to Step 2.

Consider Step 2 under TTC(v): Note that we cannot identify the cycles that arise at Step 2 because we do not know the owners of the hi-items (because v is unknown). But we can still identify the blocks that form these cycles: Let each agent and clone point to her favorite remaining item. Let each occupied house point to its owner (a clone). This gives rise to blocks. The blocks that form the cycles at Step 2 are those whose source agents are the agents in $A^2(v)$. (The cycles at Step 2 form when the sink hi-items of these blocks point to the source agents of the blocks.) Since executing the cycles at Step 2 is the same as executing the blocks forming these cycles (either way the same assignments are made), we can identify the assignments made at Step 2. Proceed to Step 3.

Consider Step 3 under TTC(v): Note that the proceedings of TTC(v) at Step 3 depends upon only the assignments made at Steps 1 and 2 (i.e., the knowledge of v is not necessary). Thus we can identify all the assignments made at Step 3. Proceed to Step 4.

By iterating the above arguments we can identify all the assignments made up to Step $t+1$.
■

CLAIM 2 *Under TTC(v) ($v \in \mathcal{V}$), consider a chain $x^1 \to a^1 \to \cdots \to x^q \to a^q$ at Step t ($t \geq 2$, t even) in which there are one or more agents. Let a^s be the agent-head of this chain so that the chain ends with the component $a^s \to x^{s+1} \to \cdots \to a^q$ (or simply a^q if $a^s = a^q$). Then the two blocks that arise at Steps t and $t+2$ and in which a^s is the source agent both begin with this component. (i.e., they are both of the form $a^s \to x^{s+1} \to \cdots \to a^q \to \cdots$ if $a^s \neq a^q$ and of the form $a^q \to \cdots$ if $a^s = a^q$.)*[18]

Proof. In the chain $x^1 \to a^1 \to \cdots \to x^q \to a^q$ since a^s is the agent-head, $a^{s+1}, a^{s+2}, \cdots, a^q$ are clones. (If $a^s = a^q$ then there are no clones that come after a^s.) Thus the block that arises at Step t and whose source agent is a^s begins with the component $a^s \to x^{s+1} \to \cdots \to a^q$ (or, it begins with a^q if $a^s = a^q$). Also, note that by definition of a chain, with the exception of a^q each member of this chain points to the very same thing at Steps t and $t+2$. Thus, the block that arises at Step $t+2$ and whose source agent is a^s also begins with the component $a^s \to x^{s+1} \to \cdots \to a^q$ (or, it begins with a^q if $a^s = a^q$). ■

We are now ready to proceed with the proofs of Lemmas 2 and 3.

[18]Note that the claim does not say that these two blocks are the same because the parts that come after a^q may be different. (And indeed, the parts that come after a^q *are* different.)

Proof of Lemma 2.

In $ch(v)$ $(= ch(v'))$ let agents be ordered as follows: $a^1, a^2, \cdots, a^n$. The proof is by induction. The arguments used to show the base case and the inductive step are similar. Thus, to avoid repetition we present below only the inductive step.[19]

<u>*Inductive step:*</u> Given that $A^s(v) = A^s(v')$ for $s = 2, 4, \cdots, t$, show that $A^{t+2}(v) = A^{t+2}(v')$.

By way of contradiction, suppose that $A^{t+2}(v) \neq A^{t+2}(v')$. W.l.o.g., let $A^{t+2}(v) = \{a^p, a^{p+1}, \cdots, a^r\}$ and $A^{t+2}(v') = \{a^p, a^{p+1}, \cdots, a^l\}$ where $r < l$. Thus, $a^{r+1} \notin A^{t+2}(v)$, $a^{r+1} \in A^{t+2}(v')$.

By Claim 1, the assignments made up to Step $t+1$ (inclusive) are the same under TTC(v) and TTC(v'). Thus, the agents, clones and items remaining are the same at Step $t+2$ under these two mechanisms. Thus, the same blocks arise at Step $t+2$ under these two mechanisms. Let $a^{r+1} \to \cdots \to x$ be the block that arises at Step $t+2$ and whose source agent is a^{r+1}.

Consider TTC(v'): Since $a^{r+1} \in A^{t+2}(v')$, at Step $t+2$ this block becomes part of a cycle and hence is executed. Hence, every agent and clone in this block is assigned her favorite remaining item at Step $t+2$ (i.e., the item that she points to in the block $a^{r+1} \to \cdots \to x$).

Consider TTC(v): Since $a^{r+1} \notin A^{t+2}(v)$ and a^{r+1} is ordered in $ch(v)$ right after the agents in $A^{t+2}(v)$, $a^{r+1} \in A^{t+4}(v)$ and a^{r+1} is the agent-head of a chain at Step $t+2$. (See chain order rules 1 and 3.) Let y be the head of this chain. By Claim 2, at Step $t+2$, y is in the block $a^{r+1} \to \cdots \to x$. Since y is the head of a chain at Step $t+2$, she is not assigned her favorite remaining item at Step $t+2$.

But then TTC(v) and TTC(v') do not induce the same clone allocation: y (an agent or a clone) is not assigned the same item under these two mechanisms. But this contradicts Lemma 1 according to which TTC(v) and TTC(v') are equivalent (because both are equivalent to Y-I($ch(v)$)). Therefore, we must have $A^{t+2}(v) = A^{t+2}(v')$. ∎

Proof of Lemma 3.

Let agents be ordered in $ch(v)$ as $a^1, a^2, \cdots, a^n$. By Lemma 2 we know that the chain order $ch(v)$ uniquely identifies the sets $A^t(v)$ for all $t \geq 2$, t even. Also, as argued in the Proof of Claim 1, for all $t \geq 2$, t even, we can identify the blocks that arise at Step t and whose source agents are those in $A^t(v)$. Using these blocks and applying the chain order rules, we explain below how we can uniquely identify v.

By chain order rule 1, in $ch(v)$: the agents in $A^2(v)$ are ordered before those in $A^4(v)$; the agents in $A^4(v)$ are ordered before those in $A^6(v)$; and so on.

Consider now Step 2 under TTC(v). Let the agents in $A^2(v)$ be $a^1, a^2, \cdots, a^s$. Let the blocks that form the cycles at Step 2 be as follows:

– $a^1 \to \cdots \to x^1$

– $a^2 \to \cdots \to x^2$

$\vdots$

[19] The base case is $A^2(v) = A^2(v')$. The proof of the base case is obtained if we substitute t with 0 in the proof of the inductive step.

– $a^s \to \cdots \to x^s$

Above, $x^1, \cdots, x^s$ are the hi-items that are assigned to agents and clones at Step 2. By chain order rule 2, we can uniquely identify $v(a^{\overline{s}})$ for $\overline{s} = 1, 2, \cdots, s$ as follows: In the set $\{x^1, x^2, \cdots, x^s\}$, $v(a^1)$ is the hi-item whose index is smallest; $v(a^2)$ is the hi-item whose index is the second smallest, and so on.

Consider now Step $t+2$ ($t \geq 2$, t even) under TTC(v). Let the agents in $A^{t+2}(v)$ be $a^{t_0+1}, a^{t_0+2}, \cdots, a^{t_1}; a^{t_1+1}, a^{t_1+2}, \cdots, a^{t_2}; \cdots; a^{t_{s-1}+1}, a^{t_{s-1}+2}, \cdots, a^{t_s}$. (The choice of $t_1, \cdots, t_s$ is explained below.)

At Step $t+2$ let the blocks whose source agents are those in $A^{t+2}(v)$ be as follows:

– $a^{t_0+1} \to \cdots \to x^{t_0+1}$, $a^{t_0+2} \to \cdots \to x^{t_0+2}$, $\cdots$, $a^{t_1} \to \cdots \to x^{t_1}$;

– $a^{t_1+1} \to \cdots \to x^{t_1+1}$, $a^{t_1+2} \to \cdots \to x^{t_1+2}$, $\cdots$, $a^{t_2} \to \cdots \to x^{t_2}$;

$\vdots$

– $a^{t_{s-1}+1} \to \cdots \to x^{t_{s-1}+1}$, $a^{t_{s-1}+2} \to \cdots \to x^{t_{s-1}+2}$, $\cdots$, $a^{t_s} \to \cdots \to x^{t_s}$.

Note that some of the agents and clones in these blocks are heads of chains at Step t. Note that we can identify them: If an agent or a clone does not point to the same item at Steps t and $t+2$ then she is the head of a chain at Step t. Thanks to Claim 2 we can also identify which agents in $A^{t+2}(v)$ are agent-heads of chains at Step t: An agent in $A^{t+2}(v)$ is the agent-head of a chain at Step t if in the pertaining block above there exists an agent or a clone who is the head of a chain at Step t. Without loss of generality, let $a^{t_0+1}, a^{t_1+1}, \cdots, a^{t_{s-1}+1}$ be the agent-heads. (And hence is our choice of the superscripts $t_1, \cdots, t_s$. Note that a^{t_0+1} is an agent-head because she is the agent in $A^{t+2}(v)$ who is ordered in $ch(v)$ before others.) Then, by chain order rule 3, at Step t:

– $a^{t_1}, a^{t_2}, \cdots, a^{t_s}$ are agent-tails;

– $a^{t_0+1}, a^{t_0+2}, \cdots, a^{t_1}$ are in the same chain and these agents are ordered in $ch(v)$ before others;

– $a^{t_1+1}, a^{t_1+2}, \cdots, a^{t_2}$ are in the same chain and these agents are ordered in $ch(v)$ next;

– and so on.

Then, by chain order rule 3 we can uniquely identify which hi-items are assigned to these agents under v:

– for $a^s \in A^{t+2}(v) \setminus \{a^{t_1}, a^{t_2}, \cdots, a^{t_s}\}$, $v(a^s) = x^{s+1}$;

– for agent-tails $a^{t_1}, a^{t_2}, \cdots, a^{t_s}$, in the set $\{x^{t_0+1}, x^{t_1+1}, \cdots, x^{t_{s-1}+1}\}$, $v(a^{t_1})$ is the hi-item whose index is smallest, $v(a^{t_2})$ is the hi-item whose index is second smallest, and so on.

Therefore, using chain order $ch(v)$ we can uniquely identify $v \in \mathcal{V}$. ■