Clone Structures in Voters' Preferences

Edith Elkind[1], Piotr Faliszewski[2], and Arkadii Slinko[3]

[1] School of Physical and Mathematical Sciences, Nanyang Tech. Univ., Singapore
[2] Department of Computer Science, AGH Univ. of Science and Technology, Poland
[3] Department of Mathematics, University of Auckland, New Zealand

Abstract. In elections, candidates that are ranked consecutively (though possibly in different order) by all voters are called *clones*. Clones can occur naturally, especially if the set of candidates has an internal structure, or they can be introduced by a malicious manipulator, who wants to change the election outcome. In this paper, we study the properties of clone structures, i.e., families of sets such that for some preference profile, each set in the family consists of candidates that appear consecutively in all votes in this profile. We provide an axiomatic characterization of clone structures: A set family satisfies our axioms if and only if there is a preference profile that has this set family as its clone structure. Further, we show that clone structures have a natural hierarchical structure that is very conveniently represented by PQ-trees. Using this PQ-tree representation we show, somewhat unexpectedly, that every clone structure can be obtained from a preference profile of at most three voters.

1 Introduction

Group decision making plays an important role in the proper functioning of human societies and multiagent systems. Collective decisions are often made by aggregating the preferences of individual agents by means of *voting*: each agent ranks the available alternatives, and a voting rule is used to select one or more winners (see [2] for a general overview of voting, and [7] for a more algorithmic perspective). In general, the structure of the set of alternatives may be quite complex. For instance, Ephrati and Rosenschein [6] explore the situation where multiple agents try to coordinate their actions in order to devise a global plan. Then the space of alternatives, i.e., of possible plans, may be huge, with some alternatives being very similar to each other. In such a case it may be reasonable to establish which plans under consideration differ fundamentally, and which are viewed as minor variations of each other.

Such structured decision-making environments have been studied in the social choice literature: for instance, Laffond et al. [9] describe the situation when a group of agents has to choose from a set that is partitioned into several "projects," where each project is defined as a set of possible variants. In this setting, all agents are likely to rank the variants of each projects contiguously. This model was further investigated by Laslier [10, 11]. Tideman and Zavist [14, 15] suggest a different explanation of why several alternatives in an election may be very similar to each other: a malicious party may try to "duplicate" an existing candidate in order to change the voting outcome. This procedure is known as *cloning* and the alternatives that appear together in all preference profiles (though not necessarily in the same order) are called *clones*. A more nuanced model of cloning was recently studied by Elkind et al. [5], who analyze the success probability of manipulation by cloning for different voting rules.

Both when clones arise naturally and when they are created by a manipulator, it may be useful to understand the internal structure of the resulting clone sets. Indeed, such an understanding could be instrumental in uncovering hidden properties of voters' preferences such as, for example, a hierarchical structure of the alternative set. Further, it may enable us to discover that, by eliminating a small number of clones, we obtain an election that has certain desirable propeties, such as, e.g., single-peakedness.[4] In either case, we could improve the election by using a better suited voting rule: in the former case, we could use hierarchic voting, and in the latter case, we could use the median voter rule—which is known to be strategy-proof for single-peaked profiles—to select a group of clones, and then pick the final winner from among them. Such an approach is likely to produce a better voting outcome as well as to reduce the voters' incentives for manipulation.

The goal of this paper is to provide a formal understanding of what families of clone sets (which we call *clone structures*) can arise in elections, and to provide convenient means of representing them. To achieve the former goal, we give an axiomatic characterization of possible clone structures. To achieve the latter goal, we show that PQ-trees of Booth and Lueker [4] very conveniently describe clone structures. In addition—and somewhat unexpectedly—we show that every clone structure, no matter how complex, can be implemented by a profile with three voters only; this profile can be constructed efficiently from the corresponding PQ-tree. We believe that our results are useful for understanding the impact of clones in decision-making scenarios, and will help in developing algorithms for settings where some of the candidates may be very similar to each other.

2 Preliminaries

Given a finite set C of *candidates* (or *alternatives*), a *preference order* (or *ranking*) over C is a total order over C, i.e., a complete, transitive and antisymmetric relation on C. Intuitively, a preference order is a ranking of the candidates from the most desirable one to the least desirable one. Given a preference order $\succ$ over C, we denote by $\overleftarrow{\succ}$ the linear order on C that is obtained by reversing $\succ$, that is, $j \overleftarrow{\succ} i$ if and only if $i \succ j$. For two disjoint sets $X, Y \subseteq C$ and an order $\succ$, we write $X \succ Y$ if $x \succ y$ for all $x \in X$ and all $y \in Y$. Given three pairwise disjoint subsets X, Y, Z of C, and an order $\succ$, we say that X *separates* Y and Z in $\succ$ if either $Y \succ X \succ Z$ or $Z \succ X \succ Y$. We say that an alternative $a \in C$ *splits*

[4] For the definition of single-peakedness, see, e.g., the classic work of Black [3] or the handbook [2]. Very briefly put, single-peakedness models societies that care about a single issue, such as, e.g., taxation level.

a subset $X \subseteq C$ with respect to an order $\succ$ if X can be partitioned into two nonempty sets X_1 and X_2 such that $\{a\}$ separates X_1 and X_2 in $\succ$; note that this implies $a \notin X$. Given two sets $X, Y \subseteq C$, we say that X is a *proper subset* of Y if $X \subseteq Y$ and $1 < |X| < |Y|$. We say that X and Y *intersect non-trivially* and write $X \bowtie Y$ if $X \cap Y \neq \emptyset$, $X \setminus Y \neq \emptyset$ and $Y \setminus X \neq \emptyset$.

A *preference profile* $\mathcal{R} = (R_1, \ldots, R_n)$ on C is a collection of n preference orders over C, where each order R_i, $1 \leq i \leq n$, represents the preference of the i-th voter; for readability, we sometimes write $\succ_i$ in place of R_i. An *election* over C is a pair $\mathcal{E} = (C, \mathcal{R})$, where $\mathcal{R}$ is a preference profile over C. A *voting rule* is a mapping $\mathcal{F}$ that, given an election $\mathcal{E}$ over C, outputs a set $\mathcal{F}(\mathcal{E}) \subseteq C$; the elements of $\mathcal{F}(\mathcal{E})$ are called the *election winners*.

Example 1. Consider $C = \{a, b, c, d\}$ and a 3-voter preference profile (R_1, R_2, R_3) such that $R_1 : a \succ_1 b \succ_1 c \succ_1 d$, $R_2 : b \succ_2 d \succ_2 c \succ_2 a$, and $R_3 : a \succ_3 b \succ_3 d \succ_3 c$. Under the Plurality rule, the candidate ranked first by most voters wins. If there are many such candidates then they all tie for victory; in practice some tie-breaking rule has to be applied. In our example a is the unique Plurality winner. On the other hand, under Borda's rule for m candidates, each candidate c receives $m - k$ points for each vote where c is ranked k-th. In our example, b is the unique Borda winner with 7 points.

There are many more voting rules, both used in practice and studied theoretically, than those presented in Example 1 (see [2]). However, in this paper we focus on the nature of preference profiles; thus, our results do not depend on the choice of a voting rule.

The following definition, inspired by [14], is fundamental for our work.

Definition 2. *Let $\mathcal{R} = (R_1, \ldots, R_n)$ be a preference profile over a candidate set C. We say that a non-empty subset $X \subseteq C$ is a* clone set *for $\mathcal{R}$ if $c \succ_i a \Longrightarrow c' \succ_i a$ and $a \succ_i c \Longrightarrow a \succ_i c'$ for every $c, c' \in X$, every $a \in C \setminus X$, and every $i = 1, 2, \ldots, n$.*

Unlike paper [14], we define singletons to be clone sets; in the election from Example 1 each of $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{d, c\}$, $\{b, c, d\}$, and $\{a, b, c, d\}$ is a clone set.

3 Axiomatic Characterization of Clone Structures

The goal of this section is to understand which set families can be represented as clone structures. That is, given a collection $\mathcal{C}$ of subsets of a candidate set C, we want to determine if there exists a preference profile $\mathcal{R}$ over C such that each clone in $\mathcal{R}$ appears in our collection and vice versa; we will say that such $\mathcal{R}$ *implements* $\mathcal{C}$. The main technical results of this section are (a) an axiomatic characterization of implementable collections of subsets, and (b) a polynomial-time algorithm for recognizing such families. While deriving these results, we lay out the groundwork for understanding clone sets in general.

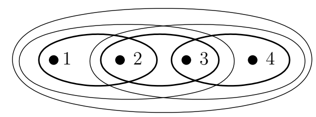
In this section, we will consider elections over the set $[m] = \{1, \ldots, m\}$. We will write $[j, k]$ to denote $\{j, j + 1, \ldots, k\}$ for $j, k \in [m]$.

Definition 3. *Given a profile $\mathcal{R} = (R_1, \ldots, R_n)$ over $[m]$, let $\mathcal{C}(\mathcal{R}) \subseteq 2^{[m]}$ be a collection of all clone sets for $\mathcal{R}$. We say that a family $\mathcal{C} \subseteq 2^{[m]}$ is a* clone structure *on $[m]$ if it is equal to $\mathcal{C}(\mathcal{R})$ for some profile $\mathcal{R}$ on $[m]$.*

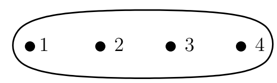
Let us now consider two examples of clone structures that will play an important role in our analysis.

Example 4. Let $\mathcal{R}$ consist of a single linear order $R : 1 \succ 2 \succ \cdots \succ m$. Then $\mathcal{C}(\mathcal{R}) = \{[i, j] \mid i \leq j\}$ (see Figure 1(a)). Let $\mathcal{R}'$ be a cyclic profile on $[m]$, i.e., $\mathcal{R}' = (R_1, \ldots, R_m)$, and the preferences of the i-th voter are given by $R_i : i \succ_i i+1 \succ_i \cdots \succ_i m \succ_i 1 \succ_i \cdots \succ_i i-1$. Then $\mathcal{C}(\mathcal{R}') = \{[m]\} \cup \{\{i\} \mid i \in [m]\}$ (see Figure 1(b)).

We call the first clone structure from Example 4 a *string of sausages* and that second one a *fat sausage*. Observe that any clone structure over $[m]$ consists of at most $\frac{m(m+1)}{2}$ sets, since each clone set can be described by its location (i.e., beginning and end) in the preference ordering of a fixed voter. Thus, a string of sausages and a fat sausage can be thought of as, respectively, the maximal and the minimal clone structure over $[m]$.



(a) A string of sausages.



(b) A fat sausage.

Fig. 1. Diagrams representing clone structures from Example 4 for $m = 4$.

The rest of this section is structured as follows. In Section 3.1 we derive a number of properties of clone structures. In Sections 3.2 and 3.3 we show that these properties indeed constitute an axiomatic characterization of clone structures. We conclude in Section 3.4 with several remarks on our proof approach.

3.1 Basic Properties of Clone Structures

Let us now establish some basic properties of clone structures. First, we observe that if a clone structure can be implemented by some profile, then it can also be implemented by the same profile with some of the preference orders reversed. This observation will prove very useful in some of the subsequent proofs.

Proposition 5. *Given a profile $\mathcal{R} = (R_1, \ldots, R_n)$, let $\mathcal{R}' = (R'_1, \ldots, R'_n)$ be a profile such that $R'_i \in \{R_i, \overleftarrow{R_i}\}$ for all $i = 1, \ldots, n$. Then $\mathcal{C}(\mathcal{R}) = \mathcal{C}(\mathcal{R}')$.*

The next proposition provides four necessary conditions for a family of subsets of $[m]$ to be a clone structure.

Proposition 6. *Let $\mathcal{R}$ be a profile on $[m]$. Then (1) $\{i\} \in \mathcal{C}(\mathcal{R})$ for any $i \in [m]$; (2) $\emptyset \notin \mathcal{C}$ and $[m] \in \mathcal{C}(\mathcal{R})$; (3) if C_1 and C_2 are in $\mathcal{C}(\mathcal{R})$ and $C_1 \cap C_2 \neq \emptyset$, then $C_1 \cup C_2$ and $C_1 \cap C_2$ are also in $\mathcal{C}(\mathcal{R})$; (4) if C_1 and C_2 are in $\mathcal{C}(\mathcal{R})$ and $C_1 \bowtie C_2$, then $C_1 \setminus C_2$ and $C_2 \setminus C_1$ are also in $\mathcal{C}$.*

Proposition 6 does not give sufficient conditions for a family of subsets of $[m]$ to be a clone structure. For example, $P = 2^{[m]} \setminus \{\emptyset\}$, where $m \geq 3$, satisfies all the conditions of Proposition 6. Yet, the cardinality of P is $2^m - 1$, whereas each clone structure over $[m]$ has at most $\frac{m(m+1)}{2}$ elements. The next proposition provides a further necessary condition for a family of subsets of $[m]$ to be a clone structure. It is strong enough to exclude the collection $2^{[m]} \setminus \{\emptyset\}$ for $m > 3$.

Given a profile $\mathcal{R}$ over $[m]$ and a set $X \in \mathcal{C}(\mathcal{R})$, we say that a set $Z \in \mathcal{C}(\mathcal{R})$ is a *proper minimal superset* of X if $X \subseteq Z$, $X \neq Z$, and there is no set $Y \in \mathcal{C}(\mathcal{R})$ such that $X \neq Y$, $Y \neq Z$ and $X \subseteq Y \subseteq Z$.

Proposition 7. *For any profile $\mathcal{R}$ on $[m]$, each $X \in \mathcal{C}(\mathcal{R})$ has at most two proper minimal supersets in $\mathcal{C}(\mathcal{R})$.*

Note, however, that for $m = 3$ the set family $2^{[m]} \setminus \{\emptyset\}$ satisfies the conclusion of Proposition 7. Yet, it is obviously not a clone structure, since it contains a "cycle" $\{1, 2\}, \{2, 3\}, \{3, 1\}$. More generally, consider a set family over $[m]$ that can be obtained from a string of sausages by adding the "missing link", i.e., the set $\{m, 1\}$ as well as all of its supersets that are necessary to satisfy the conclusions of Proposition 6; we will call this set family a *ring of sausages*. Clearly, a ring of sausages is not a clone structure, because it cannot be implemented by an acyclic preference relation; yet, the conclusion of Proposition 7 is satisfied. Thus, to obtain an axiomatic characterization of clone structures, we require an axiom that excludes such cyclic families. Simply prohibiting rings of sausages is not enough, and we need a somewhat more general condition.

Definition 8. *We say that a set family $\{A_0, \ldots, A_{k-1}\}$ is a* bicycle chain *if $k \geq 3$ and for all $i = 0, \ldots, k-1$ it holds that (1) $A_{i-1} \bowtie A_i$; (2) $A_{i-1} \cap A_i \cap A_{i+1} = \emptyset$; (3) $A_i \subseteq A_{i-1} \cup A_{i+1}$, where all indices are computed modulo k.*

Proposition 9. *If $\mathcal{C}$ is a clone structure, it does not contain a bicycle chain.*

Putting together the properties from from Propositions 6, 7 and 9 we obtain the following set of axioms:

A1. $\{f\} \in \mathcal{F}$ for any $f \in F$, $\emptyset \notin \mathcal{F}$, and $F \in \mathcal{F}$.
A2. if C_1 and C_2 are in $\mathcal{F}$ and $C_1 \cap C_2 \neq \emptyset$, then $C_1 \cup C_2$ and $C_1 \cap C_2$ are in $\mathcal{F}$.
A3. If C_1 and C_2 are in $\mathcal{F}$ and $C_1 \bowtie C_2$, then $C_1 \setminus C_2$ and $C_2 \setminus C_1$ are in $\mathcal{F}$.
A4. Each $C \in \mathcal{F}$ has at most two proper minimal supersets in $\mathcal{F}$.
A5. $\mathcal{F}$ does not contain a bicycle chain.

Our next goal is to show that these five axioms indeed characterize clone structures. In Section 3.2 we build up the necessary tools to be used in our inductive proof of this fact; the proof itself appears in Section 3.3.

3.2 Embedding and Collapsing Set Families

Let $\mathcal{E}$ and $\mathcal{F}$ be two families of subsets on two disjoint finite sets E and F, respectively. We can embed $\mathcal{F}$ into $\mathcal{E}$ as follows. Given $e \in E$, let $\mathcal{E}(e \to \mathcal{F})$ denote the family of subsets $\mathcal{E}' \cup \mathcal{F} \subseteq 2^{(E \setminus \{e\}) \cup F}$, where $\mathcal{E}'$ is obtained from $\mathcal{E}$ by replacing each set X containing e with $(X \setminus \{e\}) \cup F$.

Example 10. Consider set families $\mathcal{D} = \{\{x\}, \{y\}, \{x, y\}\}$ and $\mathcal{C} = \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}$ (both are strings of sausages and hence clone structures). Then, $\mathcal{C}(b \to \mathcal{D}) = \{\{a\}, \{x\}, \{y\}, \{x, y\}, \{c\}, \{a, x, y\}, \{x, y, c\}, \{a, x, y, c\}\}$. It is easy to check that this, again, is a clone structure.

If $\mathcal{E}$ and $\mathcal{F}$ satisfy axioms A1–A5 then so does $\mathcal{E}(e \to \mathcal{F})$. We prove it directly (it also follows from Theorem 17 combined with Proposition 16).

Proposition 11. *Let $\mathcal{E}$ and $\mathcal{F}$ be families of subsets on disjoint sets E and F, respectively, that satisfy A1–A5. Then for any $e \in E$ the set family $\mathcal{E}(e \to \mathcal{F})$ also satisfies A1–A5.*

Next, we would like to define an inverse operation to embedding, which we will call *collapsing*. Observe that when we embed $\mathcal{F} \subseteq 2^F$ into $\mathcal{E} \subseteq 2^E$, any $C \in \mathcal{E}(e \to \mathcal{F})$ is either a subset of F, a superset of F, or does not intersect F at all. Thus, for a set family $\mathcal{C}$ on A to be collapsible, it should contain a set A' that does not intersect non-trivially with any other set in $\mathcal{C}$.

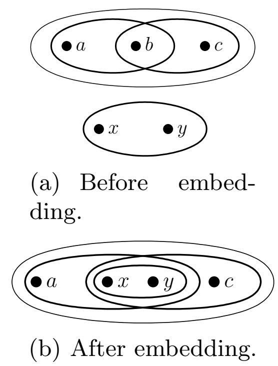


Fig. 2. Clone structures from Example 10.

Definition 12. *Let $\mathcal{F}$ be a family of subsets on a finite set F. A subset $\mathcal{E} \subseteq \mathcal{F}$ is a* subfamily *of $\mathcal{F}$ if there is a set $E \in \mathcal{F}$ such that (i) $\mathcal{E} = \{F \in \mathcal{F} \mid F \subseteq E\}$; (ii) for any $X \in \mathcal{F} \setminus \mathcal{E}$ we have either $E \subseteq X$ or $X \cap E = \emptyset$. The set E is called the* support *of $\mathcal{E}$. $\mathcal{E}$ is called a* proper subfamily *of $\mathcal{F}$ if E is a proper subset of F.*

It is not hard to check that if $\mathcal{F}$ satisfies axioms A1–A5 and $\mathcal{E}$ is a subfamily of $\mathcal{F}$, then $\mathcal{E}$ satisfies A1–A5 as well. In particular, note that we require $E \in \mathcal{F}$ (rather than just $E \subseteq F$), and hence $E \in \mathcal{E}$.

Let $\mathcal{F}$ be a family of subsets on F that satisfies A1–A5 and let $\mathcal{E}$ be a proper subfamily of $\mathcal{F}$ on $E \subset F$. Then no set $Y \in \mathcal{F}$ intersects E non-trivially, and hence $\mathcal{E}$ can be "collapsed". That is, we can obtain a new set family $\mathcal{B}$ from $\mathcal{F}$ by picking some alternative $b \notin F$, removing all sets $X \in \mathcal{E} \setminus \{E\}$ from $\mathcal{F}$, and replacing each set Y that contains E with $(Y \setminus E) \cup \{b\}$. It is not hard to check that $\mathcal{B}$ satisfies A1–A5; the proof is similar to that of Proposition 11. We will write $\mathcal{F}(\mathcal{E} \to b)$ to denote the set family obtained by collapsing a subfamily $\mathcal{E}$ of $\mathcal{F}$. That is, we have $\mathcal{B} = \mathcal{F}(\mathcal{E} \to b)$ if and only if $\mathcal{F} = \mathcal{B}(b \to \mathcal{E})$.

Now, suppose that $\mathcal{F}$ has no proper subfamilies; we will call such subset families *irreducible*. Observe that if $\mathcal{F}$ is irreducible, any proper subset $E \in \mathcal{F}$ violates the condition (ii) in Definition 12, i.e., for any proper subset $E \in \mathcal{F}$ there exists a subset $X \in \mathcal{F}$ such that $X \bowtie E$. We will use irreducible set families as a base for our inductive argument in the following section. The following preliminary observations will be useful.

Proposition 13. *Let $\mathcal{F}$ be an irreducible family of subsets of $[m]$ that satisfies A1–A5, and let D be a minimal proper subset of $\mathcal{F}$. Then $|D| = 2$.*

Proposition 14. *Let $\mathcal{F}$ be a family of subsets of $[m]$ that satisfies A1–A5. Then each candidate $i \in [m]$ belongs to at most two minimal proper subsets in $\mathcal{F}$.*

3.3 Proof of Correctness of the Axiomatic Characterization

We are ready to prove our axiomatic characterization.

Theorem 15. *Any irreducible family of subsets satisfying A1–A5 is either a string of sausages or a fat sausage.*

Proof. Let $\mathcal{F}$ be an irreducible family of subsets over $[m]$ that satisfies A1–A5. If $\mathcal{F}$ does not contain any proper subsets, then it is a fat sausage. Thus, for the remainder of the proof let us assume that $\mathcal{F}$ does contain at least one proper subset.

Let us consider a graph G whose vertices are elements of $[m]$ and there is an edge between i and j if and only if $\{i, j\}$ is a minimal proper subset of $\mathcal{F}$. By Proposition 14, the degree of each vertex in G is at most 2. Further, G cannot contain cycles, since each cycle in G would correspond to a bicycle chain in $\mathcal{F}$ formed by the two-element subsets $\{i, j\}$. Thus, G is a colleciton of paths. We will now prove that G has at most one connected component, and hence $\mathcal{F}$ is a string of sausages.

Let G' be a maximal connected component in G, and let F be the set of vertices of G'. Suppose that $F \neq [m]$. Note that by A3, F is a subset in $\mathcal{F}$. Since $\mathcal{F}$ is not a fat sausage, by Proposition 13 we have $|F| \geq 2$. Let us rename the alternatives so that $F = \{f_1, \ldots, f_k\}$ and each $\{f_i, f_{i+1}\}$, $1 \leq i < k$, is an edge of G'.

If $F \neq [m]$, there exists a proper subset $E \in \mathcal{F}$ such that $E \bowtie F$. Let us pick such a set E for which $|E \setminus F|$ is smallest. By A3, the set $E \setminus F$ belongs to $\mathcal{F}$. We consider two cases.

$|\mathbf{E} \setminus \mathbf{F}| = \mathbf{1}$. Observe that in this case $|E \cap F| \geq 2$: otherwise, E would be an edge of G'. Let e be a member of $E \setminus F$. Suppose first that $E \cap F$ is not a contiguous subset of F, that is, there are some $i, j, \ell \in [k]$ such that $i < \ell < j$, and (i) $f_i \in E$ and $f_s \notin E$ for $s < i$, (ii) $f_j \in E$ and $f_t \notin E$ for $t > j$, and (iii) $f_\ell \notin E$. Then either $E \cap F = \{f_i, f_j\}$, or $E \cap F$ intersects $\{f_{i+1}, \ldots, f_{j-1}\}$ and we have $\{f_i, f_j\} = (E \cap F) \setminus \{f_{i+1}, \ldots, f_{j-1}\}$. In both cases, we can use axiom A3 to conclude that $\{f_i, f_j\}$ belongs to $\mathcal{F}$, and hence G' contains a cycle, a contradiction. Thus, we have $E \cap F = \{f_i, \ldots, f_j\}$ for some $1 \leq i < j \leq k$.

Suppose that $j \neq k$. Then, since $i < j$, by A3 the set $E \setminus \{f_1, \ldots, f_{j-1}\} = \{e, f_j\}$ is in $\mathcal{F}$. However, this means that $e \in F$, which is a contradiction. Thus, $j = k$. Similarly, we can argue that $i = 1$. Hence, we have $F \subseteq E$, a contradiction.

$|\mathbf{E} \setminus \mathbf{F}| > \mathbf{1}$. By A3, $E \setminus F$ is a proper subset in $\mathcal{F}$. Thus, since $\mathcal{F}$ is irreducible, there is a proper subset H in $\mathcal{F}$ such that $H \bowtie (E \setminus F)$.

Suppose first that $F \subseteq H$, and consider the set $H' = H \cap E$. Since E intersects F, we have $H' \cap F \neq \emptyset$. Further, $H' \cap (E \setminus F) = H \cap (E \setminus F) \neq \emptyset$, so $H' \setminus F \neq \emptyset$. Finally, $F \setminus E \neq \emptyset$ and $H' \subseteq E$, so $F \setminus H' \neq \emptyset$. Thus, $F \bowtie H'$. However, $H' \setminus F = H \cap (E \setminus F)$ is a strict subset of $E \setminus F$, so $|H' \setminus F| < |E \setminus F|$, a contradiction with our choice of E.

Thus, we have $F \not\subseteq H$. If, nevertheless, $F \cap H \neq \emptyset$, we set $H'' = H \cap (E \cup F)$. Clearly, we have $F \cap H'' \neq \emptyset$. Since H'' is a subset of H, we also have $F \setminus H'' \neq \emptyset$.

Finally, since $H \cap (E \setminus F) \neq \emptyset$. we have $H'' \setminus F \neq \emptyset$. Thus, $H'' \bowtie F$, yet $H'' \setminus F = H \cap (E \setminus F)$ is a strict subset of $E \setminus F$, so $|H'' \setminus F| < |E \setminus F|$, a contradiction with our choice of E.

Hence, $H \cap F = \emptyset$. However, this means that $E \setminus H$ still intersects F non-trivially, and $|(E \setminus H) \setminus F| < |E \setminus F|$, a contradiction again.

We have shown that assuming that $F \neq [m]$ leads to a contradiction. Hence, $F = [m]$, which means that $\mathcal{F}$ is a string of sausages. $\square$

Thus, any irreducible set family that satisfies A1–A5 is a clone structure. This provides the basis for our inductive argument. For the inductive step, we need to show that if $\mathcal{C}$ and $\mathcal{D}$ are two clone structures over disjoint sets C and D, and c is some candidate in C, then $\mathcal{C}(c \to \mathcal{D})$ is a clone structure. However, the proof of this fact is somewhat more complicated than one might expect. Indeed, suppose that we have a pair of profiles $\mathcal{R} = (R_1, \ldots, R_n)$ and $\mathcal{Q} = (Q_1, \ldots, Q_n)$ over sets C and D, respectively, such that $\mathcal{C} = \mathcal{C}(\mathcal{R})$ and $\mathcal{D} = \mathcal{C}(\mathcal{Q})$. One might think that, given $c \in C$, we can obtain a preference profile $\mathcal{R}'$ such that $\mathcal{C}(c \to \mathcal{D}) = \mathcal{C}(\mathcal{R}')$ simply by substituting Q_i for c in R_i, for $i = 1, \ldots, n$. This intuition is not entirely correct: without additional precautions, we may introduce "parasite" clones, i.e., clones that cross the boundary between C and D. However, we can construct $\mathcal{R}'$, containing n preference orders, from $\mathcal{R}$ and $\mathcal{Q}$ by tweaking this construction slightly.

Proposition 16. *Let $\mathcal{C}$ and $\mathcal{D}$ be two clone structures over sets C and D, respectively, where $|C| = m$, $|D| = k$, and $C \cap D = \emptyset$. Then for each $c \in C$, the family of subsets $\mathcal{C}(c \to \mathcal{D})$ is a clone structure.*

Propostion 16, used inductively on top of Theorem 15, gives our main result.

Theorem 17. *A family $\mathcal{F}$ of subsets of $[m]$ is a clone structure if and only if it satisfies conditions A1–A5.*

Corollary 18. *Any irreducible clone structure is either a string of sausages or a fat sausage. Further, any subfamily of a clone structure is a clone structure.*

Based on Theorem 17, we derive a polynomial-time algorithm for testing if a given set family is a clone structure. We assume that the input set family is represented explicitly, by listing the members of each set in the family.

Theorem 19. *There exists a polynomial-time algorithm that, given a family of subsets $\mathcal{F}$ over a finite set F, checks if $\mathcal{F}$ is a clone structure over F.*

3.4 Remarks

Note that axioms A1–A3 and axioms A4–A5 play different roles in our characterization of clone structures. Indeed, A1–A3 require the set family to be "rich" enough, i.e., to be closed under various operations, and A4 and A5 require it to be "not too rich", i.e., they say that if certain sets belong to the family, then other sets should not. In fact, axioms A4 and A5 can be replaced by the requirement that there exists a linear order over the alternatives such that each set in the family occurs contiguously in this order, or, in other words, that our set family is contained in a clone structure. Checking the latter requirement reduces to the "consecutive 1s" problem, where we are given a 0-1 matrix, and are asked to reorder its columns so that 1s in each row appear consecutively; this problem is known to be polynomial-time solvable [4, 8] and is indeed closely related to our work. This provides an alternative proof of Theorem 19.

The advantage of using axioms A4 and A5 instead of the order axiom is that it is fairly easy to check whether a set family satisfies A4 and A5. In fact, we can strengthen the proof of Theorem 19 to obtain a logarithmic space algorithm. (The only hard part is to verify axiom A5 in logarithmic space; this can be done by reducing the verification problem to connectivity testing for undirected graphs and by applying the break-through result of Reingold [13]). It is unclear if we can directly verify the order axiom in logarithmic space.

Our characterization and the high-level proof strategy we used to obtain it are similar in spirit to the very general work of Möhring [12]. Since we need some of our partial results from Section 3 later on, we chose to present a self-contained argument instead of translating Möhring's results to our setting. We remark that the latter approach would be far from straightforward, since our axioms are somewhat different from those in paper [12],

4 Compact Representations of Clone Structures

In this section we consider the issue of representing clone structures. The most direct representation, suggested by Definition 3, is to list all the sets included in the clone structure. However, this representation is often wasteful and does not reveal any information about the internal workings of the clone structure. Thus, we now consider two compact representations: one using PQ-trees of Booth and Lueker [4], and one listing votes that implement a particular clone structure.

4.1 PQ-Tree Representation

In the previous section we have seen that, intuitively, clone structures are organized hierarchically. Thus, it is natural to represent them using trees. The specific type of trees that are most convenient for this task are PQ-trees introduced by Booth and Lueker [4] in the context of consecutive-1s property (and several other problems). In this section we will describe how one can derive PQ-trees for clone structures.

A PQ-tree T over a set $A = \{a_1, \dots, a_n\}$ is an ordered tree that represents a family of permutations over A as follows. The leaves of the tree correspond to the elements of A. Each internal node is either of type P or of type Q. A *frontier* of T is a permutation of A obtained by reading the leaves of T from left to right (recall that T is ordered). The following operations are allowed on the tree: If a node is of type P, then its children can be permuted arbitrarily. If a node is of type Q, then the order of its children can be reversed. A given permutation π of A is *consistent* with a PQ-tree T, if we can obtain π as the frontier of T by applying the above operations.

We now describe a natural way to represent clone structures as PQ-trees. Consider a clone structure $\mathcal{C}$ over a finite set C. Our characterization of irreducible clone structures implies that any two proper irreducible subfamilies of $\mathcal{C}$ have non-intersecting supports.

Proposition 20. *Let $\mathcal{C}$ be a clone structure over a finite set C, and let $\mathcal{B}$ and $\mathcal{D}$ be two proper irreducible subfamilies of $\mathcal{C}$ on sets $B \subseteq C$ and $D \subseteq C$, respectively. Then $B \cap D = \emptyset$.*

Proposition 20 implies that every element of C belongs to at most one proper irreducible subfamily of $\mathcal{C}$. Thus, given a clone structure $\mathcal{C} \subseteq 2^C$, there is a unique maximal collection of pairwise disjoint sets $Dec(C) = \{C_1, \dots, C_k\}$ such that $C_i \subseteq C$, $|C_i| \geq 2$, and for each $i = 1, \dots, k$ the set family $\mathcal{C}_i = \{C \in \mathcal{C} \mid C \subseteq C_i\}$ is an irreducible subfamily of $\mathcal{C}$ (if $\mathcal{C}$ is itself irreducible, then $k = 1$ and $C_1 = C$). This collection can be efficiently constructed by identifying the minimal (with respect to inclusion) non-singleton sets in $\mathcal{C}$: any such set of size $s \geq 3$ is itself an irreducible clone structure (a fat sausage), and for a set of size $s = 2$ we need to find the maximal string of sausages that contains it. Note that it need not be the case that $\cup_{i=1}^k C_i = C$: some elements may not belong to any proper irreducible clone structure (consider, for instance, the clone structure over $\{a, b, c, d\}$ given by $\{\{a\}, \{b\}, \{c\}, \{d\}, \{b, c\}, \{a, b, c, d\}\}$). We will refer to the collection $Dec(C)$ as the *decomposition* of $\mathcal{C}$.

We can now inductively define a PQ-tree $T(\mathcal{C})$ associated with a clone structure $\mathcal{C} \subseteq 2^C$ (for convenience, our PQ-tree will be labeled). Suppose first that $\mathcal{C}$ is an irreducible clone structure over the set $C = \{c_1, \dots, c_m\}$. Then by Theorem 15 it is either a string of sausages or a fat sausage. In the former case, assume without loss of generality that $\mathcal{C}$ is associated with the order $c_1 \succ c_2 \succ \dots \succ c_m$, i.e., it contains sets $\{c_i, c_{i+1}\}$ for $i = 1, \dots, m-1$. In both cases, we let $T(\mathcal{C})$ to be a tree of depth 1 that has m (ordered) leaves. The i-th leaf is labeled by c_i. If $\mathcal{C}$ is a string of sausages, the root of the tree is of type Q and is labeled by $c_1 \oplus \dots \oplus c_m$; if $\mathcal{C}$ is a fat sausage, the root is of type P and is labeled by $c_1 \odot \dots \odot c_m$. Note that when $m = 2$ the clone structure $\mathcal{C}$ is both a string of sausages and a fat sausage; to avoid ambiguity, we treat it as a fat sausage.

Now, if $\mathcal{C}$ is reducible, we compute its decomposition $Dec(C) = \{C_1, \dots, C_k\}$. For $i = 1, \dots, k$, we set $\mathcal{C}_i = \{X \in \mathcal{C} \mid X \subseteq C_i\}$, pick $c^1, \dots, c^k \notin C$, and let $\mathcal{C}'$ be the set family on the set $C' = (C \setminus \bigcup_{i=1}^k C_i) \cup \{c^1, \dots, c^k\}$ given by $\mathcal{C}' = \mathcal{C}(\mathcal{C}_1 \to c^1, \dots, \mathcal{C}_k \to c^k)$. We then construct the tree $T(\mathcal{C}')$. This tree has leaves labeled by $c^1, \dots, c^k$. We replace each such leaf c^i by the labeled tree $T(\mathcal{C}_i)$ for the irreducible set family $\mathcal{C}_i$.

We can verify by induction on m that each leaf of $T(\mathcal{C})$ is labeled with an element of C, each element of C appears as a label of some leaf, and the internal nodes of $T(\mathcal{C})$ are labeled with expressions of the form $c^1 \oplus \dots \oplus c^k$ (nodes of type Q) or $c^1 \odot \dots \odot c^k$ (nodes of type P); for the inductive proof, it suffices to observe that, whenever we construct $\mathcal{C}'$ from $\mathcal{C}$, we have $|C'| < |C|$. Further, the tree $T(\mathcal{C})$ is unique up to the standard transformations of PQ-trees. Given the tree $T(\mathcal{C})$, we can reconstruct the clone structure $\mathcal{C}$ in an obvious way. To illustrate this discussion, in Figure 3 we give a PQ-tree for the clone structure from Example 10.

We remark that the descendants of any internal node of $T(\mathcal{C})$ form a clone set. However, the converse is not necessarily true, i.e., there are clone sets that cannot be obtained in this way: if an internal node v is labeled with a string of sausages and has k children, $k \geq 3$, the descendants of any ℓ consecutive children of v, $\ell < k$, form a clone set. Indeed, it is not hard to see that any clone set corresponds either to a subtree of $T(\mathcal{C})$ or to a collection of subtrees of $T(\mathcal{C})$ whose roots are consecutive children of the same Q-node.

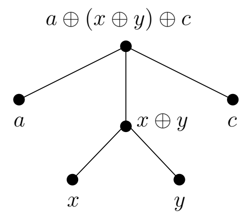


Fig. 3. Tree representation of the embedded clone structure from Example 10.

Note that the PQ-tree representation provides yet another proof of Theorem 19. Indeed, Propositions 13 and 14 hold for any subset family that satisfies A1–A5. Using this observation, we can modify the proof of Theorem 15 to show that any irreducible family of subsets satisfying A1–A5 is either a string of sausages, a ring of sausages or a fat sausage. Further, the procedure for constructing the tree $T(\mathcal{C})$ works for any set family that satisfies A1–A5; the only difference is that some nodes may have to be labeled by rings of sausages, i.e., expressions of the form $c^1 \oplus \ldots \oplus c^k \oplus c^1$. Thus, to determine whether a given subset family $\mathcal{C}$ is a clone structure, we can check whether it satisfies A1–A4, construct the tree $T(\mathcal{C})$, and verify that none of its nodes is labeled with a ring of sausages.

4.2 Minimal-Cardinality Profiles for Clone Structures

One might expect that to obtain a complex clone structure we need an election with many voters. However, it turns out that this is not true: any clone structure can be implemented by a profile with at most three voters. In what follows, we say that a clone structure $\mathcal{C}$ is *k-implementable* if there is a k-voter profile $\mathcal{R}$ such that $\mathcal{C} = \mathcal{C}(\mathcal{R})$. Thus, our goal is to show that any clone structure is 3-implementable. We deal with irreducible clone structures first.

Proposition 21. *Let $\mathcal{C}$ be an irreducible clone structure over $[m]$. If $\mathcal{C}$ is a string of sausages, it is* 1*-implementable. If $\mathcal{C}$ is a fat sausage and $m > 3$, then $\mathcal{C}$ is* 2*-implementable, but not* 1*-implementable. If $\mathcal{C}$ is a fat sausage and $m = 3$, then $\mathcal{C}$ is* 3*-implementable, but not* 2*-implementable.*

Now, the general case follows from Proposition 16.

Theorem 22. *Any clone structure $\mathcal{C}$ is* 3*-implementable. Moreover, if the tree $T(\mathcal{C})$ does not have nodes that carry labels of the form $x \odot y \odot z$, then $\mathcal{C}$ is* 2*-implementable. If $\mathcal{C}$ is a string of sausages then it is* 1*-implementable.*

5 Conclusions and Future Work

We have characterized the set families that can be obtained as clone structures in elections. We have demonstrated that every clone structure is organized hierarchically and can be conveniently represented using PQ-trees. We have also discussed implementing clone structures using a small number of voters.

We are currently working on applying the techniques and results of this paper to compute "the distance from single-peakedness" of a given profile, measured as the number of clones that need to be eliminated to obtain a single-peaked election. Our preliminary results indicate that this quantity can be efficiently computed. We believe that our analysis of clone structures may also prove useful in the context of decloning problems considered in [5], i.e., determining whether a given candidate can be made a winner with respect to a given voting rule by removing at most k clones; using our techniques for this type of problems is a fruitful direction for future research.

References

1. K. Arrow, A. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare*. Elsevier, 2002.
2. D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.
3. K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976.
4. E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 768–773. AAAI Press, July 2010.
5. E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.
6. P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Commun. ACM*, 53(11):74–82, 2010.
7. D. Fulkerson and G. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15(5):835–855, 1965.
8. G. Laffond, J. Laine, and J. Laslier. Composition consistent tournament solutions and social choice functions. *Social Choice and Welfare*, 13(1):75–93, 1996.
9. J. Laslier. Rank-based choice correspondencies. *Economics Letters*, 52(3):279–286, 1996.
10. J. Laslier. Aggregation of preferences with a variable set of alternatives. *Social Choice and Welfare*, 17(2):269–282, 2000.
11. R. Möhring. Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and boolean functions. *Annals of Operations Research*, 4:195–225, 1985.
12. O. Reingold. Undirected ST-connectivity in log-space. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 376–385. ACM Press, May 2005.
13. T. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.
14. T. Zavist and T. Tideman. Complete independence of clones in the ranked pairs rule. *Social Choice and Welfare*, 64(2):167–173, 1989.

A Missing Proofs

Proposition 6. *Let $\mathcal{R}$ be a profile on $[m]$. Then (1) $\{i\} \in \mathcal{C}(\mathcal{R})$ for any $i \in [m]$; (2) $\emptyset \notin \mathcal{C}$ and $[m] \in \mathcal{C}(\mathcal{R})$; (3) if C_1 and C_2 are in $\mathcal{C}(\mathcal{R})$ and $C_1 \cap C_2 \neq \emptyset$, then $C_1 \cup C_2$ and $C_1 \cap C_2$ are also in $\mathcal{C}(\mathcal{R})$; (4) if C_1 and C_2 are in $\mathcal{C}(\mathcal{R})$ and $C_1 \bowtie C_2$, then $C_1 \setminus C_2$ and $C_2 \setminus C_1$ are also in $\mathcal{C}$.*

Proof. Properties (1) and (2) are immediate. Let us prove (3). Let C_1 and C_2 be two sets in $\mathcal{C}(\mathcal{R})$ with $I = C_1 \cap C_2 \neq \emptyset$, and let R_i be an arbitrary preference order from $\mathcal{R}$. Consider some $a \in [m]$. Since $C_1 \in \mathcal{C}(\mathcal{R})$, if $a \in [m] \setminus C_1$, then a does not split C_1 and, as a result, a does not split I. Similarly, no alternative in $[m] \setminus C_2$ can split I. Thus, no element outside of I can split I, and hence members of I are ranked contiguously in R_i. Since this holds for any R_i in $\mathcal{R}$, we have $I \in \mathcal{C}(\mathcal{R})$.

Now, suppose that there is an alternative $a \in [m] \setminus (C_1 \cup C_2)$ that splits $C_1 \cup C_2$ in some order R_i. We know that a splits neither C_1 nor C_2, hence either $C_1 \succ_i a \succ_i C_2$ or $C_2 \succ_i a \succ_i C_1$, which is impossible since the intersection of C_1 and C_2 is nonempty. Thus, $C_1 \cup C_2 \in \mathcal{C}(\mathcal{R})$. This proves that (3) holds.

Let us now consider property (4). Suppose $C_1, C_2 \in \mathcal{C}(\mathcal{R})$ and $C_1 \bowtie C_2$. Consider the set $C_1 \setminus C_2$; for $C_2 \setminus C_1$ the argument is similar. First, no element outside of C_1 can split $C_1 \setminus C_2$, because otherwise it will split C_1 too. Further, in each R_i, the intersection $C_1 \cap C_2$ separates $C_1 \setminus C_2$ and $C_2 \setminus C_1$. Hence elements of $C_1 \cap C_2$ cannot split $C_1 \setminus C_2$ either, and the property follows. $\square$

Proposition 7. *For any profile $\mathcal{R}$ on $[m]$, each $X \in \mathcal{C}(\mathcal{R})$ has at most two proper minimal supersets in $\mathcal{C}(\mathcal{R})$.*

Proof. For the sake of contradiction, assume that there are three distinct sets Y, Z, W in $\mathcal{C}(\mathcal{R})$ such that each of them is a proper minimal superset of X. It is easy to see that $Y \cap Z = X$: by Proposition 6, $Y \cap Z \in \mathcal{C}(\mathcal{R})$, so if $(Y \cap Z) \setminus X \neq \emptyset$, neither Y nor Z would be a proper minimal superset of X. Similarly, $Y \cap W = X$ and $Z \cap W = X$. Pick two alternatives y, z so that $y \in Y \setminus X$ and $z \in Z \setminus X$. Let R_i be a preference order from $\mathcal{R}$. The set X separates $Y \setminus X$ and $Z \setminus X$, and so either $y \succ_i X \succ_i z$ or $z \succ_i X \succ_i y$; by Proposition 5 we may assume the former. Now, pick $w \in W \setminus X$. A similar argument shows that we have $y \succ_i X \succ_i w$ (as $w \succ_i X \succ_i y$ leads to a contradiction). But now we must have $z \succ_i X \succ_i w$ or $w \succ_i X \succ_i z$, none of which is possible. $\square$

Proposition 9. *If $\mathcal{C}$ is a clone structure, it does not contain a bicycle chain.*

Proof. Suppose that a clone structure $\mathcal{C}$ contains a bicycle chain $\{A_0, \ldots, A_{k-1}\}$, and let $\mathcal{R} = (R_1, \ldots, R_n)$ be a preference profile such that $\mathcal{C} = \mathcal{C}(\mathcal{R})$.

As argued in the proof of Proposition 6, the set $A_0 \cap A_1$ separates $A_0 \setminus A_1$ and $A_1 \setminus A_0$ in R_1. Thus, by Proposition 5, we can assume that we have

$$A_0 \setminus A_1 \succ_1 A_0 \cap A_1 \succ_1 A_1 \setminus A_0.$$

Further, by the definition of the bicycle chain we have $A_1 \setminus A_0 = A_1 \cap A_2$, $A_1 \setminus A_2 = A_0 \cap A_1$. Again, we have

$$A_1 \cap A_0 = A_1 \setminus A_2 \succ_1 A_1 \cap A_2 \succ_1 A_2 \setminus A_1.$$

Now, if $k = 3$, we have a contradiction already: since $A_0 \setminus A_1 \neq \emptyset$ and $A_0 \subseteq A_1 \cup A_2$, it has to be the case that A_0 intersects A_2, yet all elements of A_2 are ranked strictly below A_0. If $k > 3$, continuing inductively, we obtain that for each $i = 1, \ldots, k-1$ the set A_i is ranked below $A_{i-1} \setminus A_i$ in R_1. Hence, all elements of A_{k-1} are ranked below A_0 in R_1. However, we have $A_0 \cap A_{k-1} \neq \emptyset$, a contradiction. □

Proposition 11. *Let $\mathcal{E}$ and $\mathcal{F}$ be families of subsets on disjoint sets E and F, respectively, that satisfy A1–A5. Then for any $e \in E$ the set family $\mathcal{E}(e \to \mathcal{F})$ also satisfies A1–A5.*

Proof. We have $\{e'\} \in \mathcal{E}$ for all $e' \in E$, $\{f\} \in \mathcal{F}$ for all $f \in F$, so $\{g\} \in \mathcal{E}(e \to \mathcal{F})$ for all $g \in (E \setminus \{e\}) \cup F$. Clearly, $\emptyset \notin \mathcal{E}(e \to \mathcal{F})$. Further, $E \in \mathcal{E}$ and $e \in E$, so $(E \setminus \{e\}) \cup F \in \mathcal{E}(e \to \mathcal{F})$. Thus, A1 is satisfied.

Throughout the rest of the proof, we will use the observation that no set $D \in \mathcal{E}(e \to \mathcal{F})$ can intersect F non-trivially, i.e., we have that for each $D \in \mathcal{E}(e \to \mathcal{F})$ it holds that either $D \cap F = \emptyset$ or $D \setminus F = \emptyset$ or $F \setminus D = \emptyset$.

We will now show that $\mathcal{E}(e \to \mathcal{F})$ satisfies A3 and A4. Consider two sets C_1, C_2 in $\mathcal{E}(e \to \mathcal{F})$ such that $C_1 \cap C_2 \neq \emptyset$. If $C_1 \subseteq C_2$ or $C_2 \subseteq C_1$, then A2 and A3 trivially hold, so we can assume that $C_1 \bowtie C_2$.

Suppose first that $C_1 \subseteq F$. Then $C_2 \cap F \neq \emptyset$ and it cannot be the case that $F \subseteq C_2$, since we assume $C_1 \setminus C_2 \neq \emptyset$. Hence, $C_2 \subseteq F$, so $C_1, C_2 \in \mathcal{F}$, and the sets $C_1 \cap C_2, C_1 \cup C_2, C_1 \setminus C_2, C_2 \setminus C_1$ belong to $\mathcal{F}$ and hence to $\mathcal{E}(e \to \mathcal{F})$.

Next, suppose that $F \subseteq C_1$. Set $C_1' = (C_1 \setminus F) \cup \{e\}$. Since $C_2 \not\subseteq C_1$, we have $C_2 \not\subseteq F$, and hence either $F \subseteq C_2$ or $F \cap C_2 = \emptyset$. In the former case, set $C_2' = (C_2 \setminus F) \cup \{e\}$; in the latter case, set $C_2' = C_2$. In both cases, we have $C_1', C_2' \in \mathcal{E}$, and $C_1' \bowtie C_2'$. Therefore, the sets $C_1' \cap C_2', C_1' \cup C_2', C_1' \setminus C_2', C_2' \setminus C_1'$ belong to $\mathcal{E}$, and hence the sets $C_1 \cap C_2, C_1 \cup C_2, C_1 \setminus C_2, C_2 \setminus C_1$ belong to $\mathcal{E}(e \to \mathcal{F})$.

If $F \subseteq C_2$, the argument is similar. Thus, it remains to consider the case $C_1 \cap F = \emptyset$, $C_2 \cap F = \emptyset$. Then $C_1, C_2 \in \mathcal{E}$. Thus, the sets $C_1 \cap C_2, C_1 \cup C_2, C_1 \setminus C_2, C_2 \setminus C_1$ belong to $\mathcal{E}$ and do not contain e, and hence they belong to $\mathcal{E}(e \to \mathcal{F})$ as well. Thus, axioms A2 and A3 are satisfied.

To show that A4 holds, assume for the sake of contradiction that some set $C \in \mathcal{E}(e \to \mathcal{F})$ has three proper minimal supersets X, Y, and Z in $\mathcal{E}(e \to \mathcal{F})$. If we have $C \subseteq F$, then the sets X, Y and Z cannot strictly contain F (or they would not be proper minimal supersets), but have to intersect F, so it has to be the case that $X, Y, Z \subseteq F$. Thus, C has three proper minimal supersets in $\mathcal{F}$, a contradiction. Next, suppose that $F \subseteq C$. Then all three sets X, Y and Z are supersets of F, too. Consider the sets $C' = (C \setminus F) \cup \{e\}$, $X' = (X \setminus F) \cup \{e\}$, $Y' = (Y \setminus F) \cup \{e\}$, $Z' = (Z \setminus F) \cup \{e\}$. All these sets are in $\mathcal{E}$. Moreover, X', Y' and Z' are distinct, and each of them is a superset of C'. To see that all of them are proper supersets of C', observe that if $C' \subset T' \subset X'$, then $C \subset (T' \setminus \{e\}) \cup F \subset X$, a contradiction with X being a minimal proper superset of C. Thus, C' has three minimal proper supersets in $\mathcal{E}$, a contradiction. Finally, if $C \cap F = \emptyset$, we have $C \in \mathcal{E}$. For $T = X, Y, Z$, let $T' = T$ if $F \cap T = \emptyset$ and $T' = (T \setminus F) \cup \{e\}$ otherwise. Clearly, the sets X', Y' and Z' are in $\mathcal{E}$. By the same argument as above, we can show that C has three minimal proper supersets in $\mathcal{E}$, a contradiction. Thus, $\mathcal{E}(e \to \mathcal{F})$ satisfies A4.

Finally, let $\mathcal{E}(e \to \mathcal{F})$ contain a bicycle chain $\{A_0, \ldots, A_{k-1}\}$; in what follows, all indices are computed modulo k. Suppose first that we have $A_i \subseteq F$ for some $i = 0, \ldots, k-1$. Then $A_{i-1} \cap F \neq \emptyset$, $A_{i+1} \cap F \neq \emptyset$. Since both A_{i-1} and A_{i+1} intersect A_i non-trivially, neither of them can contain F, and therefore both of them are subsets of F. Applying this argument inductively, we conclude that all sets A_i, $i = 0, \ldots, k-1$, are subsets of F, i.e., $\mathcal{F}$ contains a bicycle chain, a contradiction. Thus, we can assume that for each $i = 0, \ldots, k-1$ either $F \subseteq A_i$ or $F \cap A_i = \emptyset$. For each $i = 0, \ldots, k-1$, set $A'_i = (A_i \setminus F) \cup \{e\}$ if $F \subseteq A_i$ and set $A'_i = A_i$ otherwise. It is straightforward to check that the set family $\{A'_0, \ldots, A'_{k-1}\}$ is a bicycle chain in $\mathcal{E}$, a contradiction. Thus, $\mathcal{E}(e \to \mathcal{F})$ satisfies A5. $\square$

Proposition 13. *Let $\mathcal{F}$ be an irreducible family of subsets of $[m]$ that satisfies A1–A5, and let D be a minimal proper subset of $\mathcal{F}$. Then $|D| = 2$.*

Proof. Suppose for the sake of contradiction that $|D| \geq 3$. The set family $\mathcal{D} = \{F \in \mathcal{F} \mid F \subseteq D\}$ is not a subfamily of $\mathcal{F}$, which means that $\mathcal{F}$ contains a proper subset E such that $D \bowtie E$. However, by A2 and A3, both $D \cap E$ and $D \setminus E$ must belong to $\mathcal{F}$, both are strict subsets of D, and at least one of them has at least two elements. Thus, D is not a minimal proper subset, a contradiction. $\square$

Proposition 14. *Let $\mathcal{F}$ be a family of subsets of $[m]$ that satisfies A1–A5. Then each candidate $i \in [m]$ belongs to at most two minimal proper subsets in $\mathcal{F}$.*

Proof. Suppose for the sake of contradiction, that i belongs to three minimal proper subsets in $\mathcal{F}$. Since these subsets are minimal proper subsets, they are also minimal proper supersets of $\{i\}$. However, by A4, no subset of $\mathcal{F}$ has more than two minimal proper supersets, a contradiction. $\square$

Proposition 16. *Let $\mathcal{C}$ and $\mathcal{D}$ be two clone structures over sets C and D, respectively, where $|C| = m$, $|D| = k$, and $C \cap D = \emptyset$. Then for each $c \in C$, the family of subsets $\mathcal{C}(c \to \mathcal{D})$ is a clone structure.*

Proof. Fix a $c \in C$, and let $\mathcal{R} = (R_1, \ldots, R_n)$ and $\mathcal{Q} = (Q_1, \ldots, Q_{n'})$ be two profiles of voters such that $\mathcal{C} = \mathcal{C}(\mathcal{R})$ and $\mathcal{D} = \mathcal{C}(\mathcal{Q})$. Since duplicating linear orders in $\mathcal{R}$ and $\mathcal{Q}$ does not change $\mathcal{C}$ and $\mathcal{D}$, we can assume without loss of generality that $n = n' \geq 2$. Our goal is to construct a profile $\mathcal{R}'$ such that $\mathcal{C}(c \to \mathcal{D}) = \mathcal{C}(\mathcal{R}')$. This profile will have n voters and $m + k - 1$ alternatives, that is, $\mathcal{R}' = (R'_1, \ldots, R'_n)$. We will construct $\mathcal{R}'$ in two steps. First, for each $i = 1, \ldots, n$, we set R^0_i to be identical to R_i except that the occurrence of c

is replaced by Q_i; denote the resulting profile by $\mathcal{R}^0 = (R_1^0, \dots, R_n^0)$ and let $\mathcal{C}^0 = \mathcal{C}(\mathcal{R}^0)$. It is easy to see that all elements of $\mathcal{C}(c \to \mathcal{D})$ are clones in $\mathcal{R}^0$, so $\mathcal{C}(c \to \mathcal{D}) \subseteq \mathcal{C}^0$. If also $\mathcal{C}^0 \subseteq \mathcal{C}(c \to \mathcal{D})$, we are done, since in this case we can set $\mathcal{R}' = \mathcal{R}^0$.

Otherwise, we flip Q_n. That is, assuming without loss of generality that Q_n ranks the elements of D as

$$Q_n : d_1 \succ d_2 \succ \dots \succ d_k$$

and R_n is given by $C_1 \succ c \succ C_2$, we define

$$R'_n : C_1 \succ d_k \succ \dots \succ d_1 \succ C_2,$$

where we assume that R'_n orders the elements of C_1 and C_2 in the same way as R_n does; we also set $R'_i = R_i^0$ for $i = 1, \dots, n-1$. Consider the resulting profile $\mathcal{R}'$, and let $\mathcal{C}' = \mathcal{C}(\mathcal{R}')$. We claim that $\mathcal{C}' = \mathcal{C}(c \to \mathcal{D})$. As above, it is easy to see that $\mathcal{C}(c \to \mathcal{D}) \subseteq \mathcal{C}'$. It remains to show that $\mathcal{C}' \subseteq \mathcal{C}(c \to \mathcal{D})$.

Let X be the "parasite" clone in $\mathcal{C}^0 \setminus \mathcal{C}(c \to \mathcal{D})$. Clearly, it cannot be the case that $X \subseteq C$ or $X \subseteq D$. Further, if $D \subseteq X$, then $(X \setminus D) \cup \{c\}$ is a clone in $\mathcal{C}$, and hence $X \in \mathcal{C}(c \to \mathcal{D})$. Thus, the sets $C_X = X \cap C$ and $D_X = X \cap D$ are both non-empty, and $D_X \neq D$. By Proposition 5, we may assume that each order in $\mathcal{R}^0$ is of the form $\dots \succ C_X \succ D_X \succ D \setminus D_X \succ \dots$.

Now, suppose for the sake of contradiction that Y is a clone in $\mathcal{C}' \setminus \mathcal{C}(c \to \mathcal{D})$. By the same argument as in the previous paragraph, we conclude that $Y \cap D \neq \emptyset$, $Y \cap C \neq \emptyset$, and $D \not\subseteq Y$. Thus, we have two possibilities:

- $d_1 \in Y$, $d_k \notin Y$. Then, since Y is contiguous in R'_1 and $Y \cap C \neq \emptyset$, we have $Y \cap C_X \neq \emptyset$. However, in R'_n the element d_k splits Y and C_X, a contradiction.
- $d_1 \notin Y$, $d_k \in Y$. Then, since Y is contiguous in R'_n and $Y \cap C \neq \emptyset$, we have $Y \cap C_X \neq \emptyset$. However, in R'_1 the element d_1 splits Y and C_X, a contradiction.

Hence, we have $Y \in \mathcal{C}(c \to \mathcal{D})$. The proof is complete. □

The above proof could be simplified if we were willing to use more voters in the profile for $\mathcal{C}(c \to \mathcal{D})$. However, the current version of the proof is very useful when we consider the number of voters needed to implement a particular clone structure.

Theorem 17. *A family $\mathcal{F}$ of subsets of $[m]$ is a clone structure if and only if it satisfies conditions A1–A5.*

Proof. We have already argued that any clone structure satisfies A1–A5; it remains to prove that the converse is also true.

Our proof is by induction on m. Clearly the theorem holds for $m = 1$ and for $m = 2$. For the inductive step, assume it holds for each $m' < m$. Let $\mathcal{F}$ be a family of subsets of $[m]$ that satisfies A1–A5. If $\mathcal{F}$ is irreducible then, by Theorem 15, it is either a string of sausages or a fat sausage and thus a clone structure. Otherwise, $\mathcal{F}$ contains a proper subfamily $\mathcal{D}$. Let $\mathcal{F}' = \mathcal{F}(\mathcal{D} \to e)$ for some $e \notin [m]$. We have argued that $\mathcal{D}$ and $\mathcal{F}'$ satisfy axioms A1–A5. Hence, by our inductive hypothesis both $\mathcal{F}'$ and $\mathcal{D}$ are clone structures and so, by Proposition 16, $\mathcal{F} = \mathcal{F}'(e \to \mathcal{D})$ is a clone structure as well. This completes the proof. □

Theorem 19. *There exists a polynomial-time algorithm that, given a family of subsets $\mathcal{F}$ over a finite set F, checks if $\mathcal{F}$ is a clone structure over F.*

Proof. It is easy to see that we can check in polynomial time whether $\mathcal{F}$ satisfies A1–A5. Now, suppose that $\mathcal{F}$ has passed this check, and it remains to verify that it satisfies A5. We can directly check if $\mathcal{F}$ contains a bicycle chain of size 3, by considering all possible triples of the subsets in $\mathcal{F}$. To check for bicycle chains of size 4 or more, we will construct a directed graph G as follows. The vertices of G are ordered pairs (X, Y), where X and Y are two subsets in $\mathcal{F}$ such that $X \bowtie Y$. There is a directed edge from (X, Y) to (Y', Z) if $Y = Y'$, $X \cap Y \cap Z = \emptyset$. and $Y \subseteq X \cup Z$. Intuitively, G has an edge from (X, Y) to (Y, Z) if X, Y and Z can be three consecutive sets in a bicycle chain. It is not hard to verify that G contains a directed cycle if and only if $\mathcal{F}$ contains a bicycle chain of size 4 or more. Indeed, let $\{A_0, \dots, A_{k-1}\}$ be a bicycle chain of size $k \geq 4$ in $\mathcal{F}$. Then any pair (A_i, A_{i+1}) is a vertex of G. Moreover, there is an edge in G between (A_{i-1}, A_i) and (A_i, A_{i+1}), so $(A_0, A_1), (A_1, A_2), \dots, (A_{k-1}, A_0)$ is a directed cycle in G (as always in our discussion of bicycle chains, the indices are computed modulo k). Conversely, if G contains a directed cycle of the form $(X_0, X_1), (X_1, X_2), \dots, (X_{k-1}, X_0)$, then the sets $X_0, \dots, X_{k-1}$ form a bicycle chain. □

Propostion 21. *Let $\mathcal{C}$ be an irreducible clone structure over $[m]$. If $\mathcal{C}$ is a string of sausages, it is 1-implementable. If $\mathcal{C}$ is a fat sausage and $m > 3$, then $\mathcal{C}$ is 2-implementable, but not 1-implementable. If $\mathcal{C}$ is a fat sausage and $m = 3$, then $\mathcal{C}$ is 3-implementable, but not 2-implementable.*

Proof. If $\mathcal{C}$ is a string of sausages, it can be implemented using a single order, namely, $1 \succ \dots \succ m$.

Now, suppose that $\mathcal{C}$ is a fat sausage. Clearly, it cannot be implemented with a single order, as the clone structure that corresponds to the latter is a string of sausages.

Suppose first that $m = 2k$. For convenience, set $x_i = i$, $y_i = k + i$ for $i = 1, \dots, k$. We define $\mathcal{R} = (R_1, R_2)$ as follows.

$$R_1 : x_1 \succ \dots \succ x_k \succ y_1 \succ \dots \succ y_k,$$
$$R_2 : y_1 \succ x_1 \succ y_2 \succ x_2 \succ \dots \succ y_k \succ x_k.$$

We claim that $\mathcal{C} = \mathcal{C}(\mathcal{R})$. Clearly, we have $\mathcal{C} \subseteq \mathcal{C}(\mathcal{R})$. Now, suppose that $D \in \mathcal{C}(\mathcal{R}) \setminus \mathcal{C}$, i.e., $|D| \neq 1, m$. Since D has to be contiguous in R_1, we have one of the following three cases:

(a) $D = \{x_i, \dots, x_j\}$ for some $1 \leq i < j \leq k$;
(b) $D = \{y_i, \dots, y_j\}$ for some $1 \leq i < j \leq k$;

(c) $D = \{x_i, \ldots, y_j\}$ for some $1 \le i \le k$, $1 \le j \le k$.

Case (a) is impossible since in R_2 the element y_j appears between x_i and x_j. Similarly, case (b) is impossible since in R_2 the element x_i appears between y_i and y_j. In case (c) we have $x_k, y_1 \in D$. Since these elements appear at the opposite ends of R_2, we conclude that $D = [m]$, a contradiction.

Next, suppose that $m = 2k+1$, $k > 1$. Set $x_i = i$, $y_i = k+i$ for $i = 1, \ldots, k$, $z = 2k+1$. We define $\mathcal{R} = (R_1, R_2)$ as follows.

$$R_1 : x_1 \succ \ldots \succ x_k \succ y_1 \succ \ldots \succ y_{k-1} \succ z \succ y_k,$$
$$R_2 : y_1 \succ x_1 \succ y_2 \succ x_2 \succ \ldots \succ y_k \succ x_k \succ z.$$

Again, it is clear that $\mathcal{C} \subseteq \mathcal{C}(\mathcal{R})$. Now, suppose that $D \in \mathcal{C}(\mathcal{R}) \backslash \mathcal{C}$, i.e., $|D| \neq 1, m$. As in the case of even m, D cannot be of the form $\{x_i, \ldots, x_j\}$ for $1 \le i < j \le k$, or of the form $\{y_i, \ldots, y_j\}$ for $1 \le i < j \le k-1$. Further, if D is of the form $\{x_i, \ldots, y_j\}$ for some $i = 1, \ldots, k$ and some $j = 1, \ldots, k-1$, then D must contain all elements that appear between x_k and y_1 in R_2, i.e., either $D = [m]$ or $D = [m] \setminus \{z\}$, which is impossible. Now, if D contains z, it must also contain the only element that is adjacent to it in R_2, i.e., x_k. As y_1 appears between x_k and z in R_1, we have $y_1 \in D$. But then $D = [m]$, since y_1 and z are extreme elements of R_2.

Finally, if $m = 3$, we can set $\mathcal{R} = (R_1, R_2, R_3)$, where $R_1 : 1 \succ 2 \succ 3$, $R_2 : 2 \succ 1 \succ 3$, $R_3 : 2 \succ 3 \succ 1$. To see that $\mathcal{C}$ cannot be implemented by any 2-voter profile (R_1, R_2), observe that we can assume without loss of generality that R_1 is of the form $1 \succ 2 \succ 3$, and in R_2 element 2 is adjacent to at least one of the remaining elements (and hence forms a clone with that element). □

Proposition 20. *Let $\mathcal{C}$ be a clone structure over a finite set C, and let $\mathcal{B}$ and $\mathcal{D}$ be two proper irreducible subfamilies of $\mathcal{C}$ on sets $B \subseteq C$ and $D \subseteq C$, respectively. Then $B \cap D = \emptyset$.*

Proof. We have $B \in \mathcal{B}$, $D \in \mathcal{D}$, so, by definition of a subfamily, it cannot be the case that $B \bowtie D$. Further, if, say, $B \subseteq D$, each element of $\mathcal{B}$ would be a subset of D, so $\mathcal{B}$ would be a subfamily of $\mathcal{D}$, a contradiction with the irreducibility of $\mathcal{D}$. Similarly, $D \subseteq B$ leads to a contradiction as well. □

Theorem 22. *Any clone structure $\mathcal{C}$ is 3-implementable. Moreover, if the tree $T(\mathcal{C})$ does not have nodes that carry labels of the form $x \odot y \odot z$, then $\mathcal{C}$ is 2-implementable. If $\mathcal{C}$ is a string of sausages then it is 1-implementable.*

Proof. If $\mathcal{C}$ is a string of sausages, then it clearly is 1-implementable. Otherwise, the following argument proves the theorem.

Fix a clone structure $\mathcal{C}$ on a set C of size m. If $T(\mathcal{C})$ does not have nodes that carry labels of the form $x \odot y \odot z$, then set $k = 2$. Otherwise set $k = 3$. The proof is by induction on m. If $m = 1$ or $m = 2$, the theorem is obviously true. Further, if $\mathcal{C}$ is irreducible, the theorem follows from Proposition 21. Otherwise, $\mathcal{C}$ contains a proper subfamily $\mathcal{D}$. By the inductive assumption, the clone structures

$\mathcal{D}$ and $\mathcal{C}(\mathcal{D} \to d)$, where $d \notin C$, are k-implementable. Let $\mathcal{R} = (R_1, \dots, R_k)$ and $\mathcal{Q} = (Q_1, \dots, Q_k)$ be the respective preference profiles, i.e., $\mathcal{C}(\mathcal{D} \to d) = \mathcal{C}(\mathcal{R})$, $\mathcal{D} = \mathcal{C}(\mathcal{Q})$. Then the proof of Proposition 16 shows how to combine $\mathcal{R}$ and $\mathcal{Q}$ to obtain a preference profile $\mathcal{R}'$ with k voters such that $\mathcal{C} = \mathcal{C}(\mathcal{R}')$. $\square$