Journal of Artificial Intelligence Research 81 (2024) 71-116 Submitted 10/2023; published 09/2024

Separating and Collapsing Electoral Control Types

Benjamin Carleton CARLETON@CS.CORNELL.EDU
Department of Computer Science, Cornell University, Ithaca, NY, USA

Michael C. Chavrimootoo MCHAVRIM@U.ROCHESTER.EDU
Department of Computer Science, Denison University, Granville, OH, USA

Lane A. Hemaspaandra LANE@CS.ROCHESTER.EDU
Department of Computer Science, University of Rochester, Rochester, NY, USA

David E. Narváez DAVID.NARVAEZ@COMPUTER.ORG
Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA

Conor Taliancich CTALIANC@U.ROCHESTER.EDU
Property Matrix, Culver City, CA, USA

Henry B. Welles HWELLES@U.ROCHESTER.EDU
Department of Computer Science, University of Rochester, Rochester, NY, USA

Abstract

Electoral control refers to attacking elections by adding, deleting, or partitioning voters or candidates. Hemaspaandra, Hemaspaandra, and Menton recently discovered, for seven pairs $(\mathcal{T}, \mathcal{T}')$ of seemingly distinct standard electoral control types, that $\mathcal{T}$ and $\mathcal{T}'$ are in practice identical: For each input I and each election system $\mathcal{E}$, I is a "yes" instance of both $\mathcal{T}$ and $\mathcal{T}'$ under $\mathcal{E}$, or of neither. Surprisingly, this had previously gone undetected even as the field was score-carding how many standard control types various election systems were resistant to; various "different" cells on such score cards were, unknowingly, duplicate effort on the same issue. This naturally raises the worry that perhaps other pairs of control types are identical, and so work still is being needlessly duplicated.

We completely determine, for all standard control types, which pairs are, for elections whose votes are linear orderings of the candidates, always identical. In particular, we prove that no identical control pairs exist beyond the known seven. We also for three central election systems completely determine which control pairs are identical ("collapse") with respect to those particular election systems, and we also explore containment and incomparability relationships between control pairs. For approval voting, which has a different "type" for its votes, Hemaspaandra, Hemaspaandra, and Menton's seven collapses still hold (since we observe that their argument applies to all election systems). However, we find 14 additional collapses that hold for approval voting but do not hold for some election systems whose votes are linear orderings of the candidates. We find one new collapse for veto elections and none for plurality. We prove that each of the three election systems mentioned have no collapses other than those inherited from Hemaspaandra, Hemaspaandra, and Menton or added in the present paper. We establish many new containment relationships between separating control pairs, and for each separating pair of standard control types

classify its separation in terms of either containment (always, and strict on some inputs) or incomparability.

Our work, for the general case and these three important election systems, clarifies the landscape of the 44 standard control types, for each pair collapsing or separating them, and also providing finer-grained information on the separations.

1. Introduction

Preference aggregation is a central focus of social choice theory, while a central focus of AI is the ways in which agents interact, often in tasks or decision-making settings. It thus is natural that ever since computational social choice theory emerged as a field, it has been prominent at AI venues. The rise of online decision-making and the increasing use of agents in our increasingly online world make computational social choice, and in particular its approach to the study of preference aggregation—including its focus on the study of the complexity of attacks on elections—ever more important. This paper explores, and for some central cases definitively determines, the extent of a type of co-incidence (namely, the equality of seemingly different notions) that affects one of the most important models—control attacks—in the richly studied and important area of the (non)resistance to attacks of preference aggregation by voting. This is important in computational social choice and to AI, since finding collapses of two control types for a given election system means the two cases are effectively identical as sets, and so saves researchers from doing the duplicate work of separately studying the control types as to any set-based property, such as complexity, etc.

Let us start with a brief parallel. Suppose Professor Fou specializes in complexity classification, and for each problem that comes through the door, Professor Fou tries to prove it complete for Fou's pet list of classes: NP, coNP, and cocoNP. So on Monday, a problem P_1 from scheduling theory would come in the door and Fou might prove it complete for coNP and would give arguments suggesting NP-completeness and cocoNP-completeness are each unlikely. On Tuesday, a problem P_2 from computational social choice might come in the door and Fou might prove it complete for NP and cocoNP, and would give evidence suggesting coNP-completeness is wildly unlikely. And so on for many years.

Most CS academics would be horrified at the situation and would say: How can Fou not have stepped back and tried to see the big picture—and realized that NP and cocoNP are exactly the same class, and thus that Fou was doing duplicate work!

This seems at first a comic situation, yet a more subtle cousin of it has been in play in the computational social choice world for many years.

Let us explain what we mean by that. Control and manipulation are the two families of attack types that were the focus of the seminal papers of Bartholdi, Tovey, and Trick (1989a, 1989b, 1992) on computational social choice. The various control attacks model different attempts to affect the outcome of elections by changes to their structure, namely, via adding, deleting, or partitioning voters or candidates. The control attack types also vary as to whether the goal is to make the focus candidate a winner (perhaps tied; this is known as the nonunique-winner (NUW) or cowinner setting), or to be a winner who is not tied with anyone else (this is known as the unique-winner (UW) setting), or to *not* (again regarding one of those two variants regarding ties) win. Over time, there has been something of a

race or contest in computational social choice to find election systems for which a very large number of control types have the property that it is NP-hard to determine on a given input whether that type of control can succeed (see the discussion in our Related Work section).

But what if two (compatible, i.e., having the same input type) control types were in fact the same? That is, what if for every election system (whose votes are linear orders over the candidates) and on every input, despite the control types seeming to model different actions, in fact either for both control types the answer is yes (i.e., for each of the two control types, there is a way to achieve the goal of making the focus candidate win (or lose)), or for both the answer is no (i.e., for each of the two control types, there is no way to achieve the goal of making the focused candidate win (or lose)). We will say that the types "collapse" in this case. Then the two types *for all practical purposes* are the same. In such a case, all research that separately studied the two types—for example to determine their computational complexity for some election system (over linear orders)—would be doing the same work twice. As our abstract mentioned, surprisingly, it has recently been observed by Hemaspaandra, Hemaspaandra, and Menton (2020) that there are (at least) seven such collapsing pairs of control types. In fact, that paper shows that four control types pairwise collapse—yielding six collapsing pairs—and one other pair also collapses.

The natural question this raises is whether there are other undiscovered collapses—either ones that hold for all election systems or, failing that, ones that hold when our focus is limited to one of the most important election systems. The universe of control types we study is that of the 22 "standard" control types used by Hemaspaandra et al. (2020) and other papers (and these are exactly the same 22 control types that appear in the key table summarizing control research as found in the Handbook of Computational Social Choice, see Faliszewski & Rothe, 2016), each studied in both the NUW and the UW winner model; so 44 control types in total. For this control-type universe, we completely resolve whether additional collapses exist, both as to whether any additional collapses beyond the seven hold in general—we show that the remaining hundreds of pairs all separate—and as to the major election systems plurality, veto, and approval, for two of which we do discover additional collapses (one additional collapsing pair for veto and 14 additional collapsing pairs for approval) and for all three of which we then separate all remaining pairs.

Why is this important? As pointed out above, collapses reveal that seemingly different control types are for all practical purposes the same ones in disguise, and so general collapses give a clearer picture of the world of control types, and can help us avoid duplicate work, as has indeed already occurred.[1] Separations on the other hand assure us that the types differ, and so we are not in the above way doing duplicate work in studying separately the two control types. (Of course, even for differing control types there may be connections of various sorts between the two types, e.g., if both are NP-complete, they will many-one polynomial-time interreduce.)

1. Such duplicate work has already occurred extensively. Each time a paper built polynomial-time algorithms for both elements of a collapsing pair of control types, or proved NP-completeness for both elements, the paper did needless work on one element of the pair, since the sets involved in such a pair are the same set and so perforce they are of identical complexity. As just a few of the many papers that would have been saved a bit or a lot of work by either the seven general collapses of Hemaspaandra et al. (2020) or the additional concrete-system collapses established in the present paper, we mention Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009), Hemaspaandra, Hemaspaandra, and Rothe (2007), Menton (2013), and Neveling and Rothe (2017).

Since (general-case) collapse is defined as universal over election systems, its negation (separation) is existential over election systems, and as this paper shows, specific systems may have additional collapses. Thus it is important for the field, as to its understanding of control types, to find the collapse/separation behavior for important election systems, and we completely resolve this for the three arguably most important ones: plurality, veto, and approval. We hope that future papers by others will study additional systems or, even better, will (as we do for some cases in Section 3.3) define sufficient conditions that in one fell swoop provide collapses that apply to groups of election systems.

Separations themselves can be refined, for two (compatible) control types, if on each input (except with no focus candidate specified in the input) we consider for each of the two control types the set of all candidates c such that if c is the focal candidate, the given control action can succeed. It might be the case that for the first control type on all inputs that set is a subset of that set for the second control type; or it could always be a superset; or neither of these cases could hold. Of course, if the first two cases hold simultaneously, that is the same as the two types collapsing. For *all* our control pairs that separate, we refine our separations into the three cases just described: "always $\subseteq$, and on some instance $\subsetneq$"; "always $\supseteq$, and on some instance $\supsetneq$"; and neither of the above (i.e., "incomparable"). We find, for various separating pairs, such containment relationships, and not all are UW and NUW variants of each other (though UW and NUW always have an obvious $\subseteq$ or $\supseteq$ relation; which one of those holds depends on whether the type is "constructive" ($\subseteq$ holds) or "destructive" ($\supseteq$ holds); we will refer back to this fact later as $(**)$).

We draw on a wide variety of approaches to establish our separations and collapses. Many of our separations are established by human-constructed example; since the number of pairs we separate is large, when we can we build examples that simultaneously yield many separations. Some of our separation constructions are obtained by computer-aided search. Some of our separation results on the "general-case" tables are obtained by inheritance from a specific-case table. As to our new collapses, some are achieved by direct arguments that exploit some feature of the setting, and most are proven via an axiomatic-sufficient-condition approach. Also, some of the collapse (equality) entries in our tables for specific election systems are inherited from the general-case tables. Some of our collapses draw on multiple approaches. For example, our proof of Theorem 3.14 is a direct proof based on aspects of the setting but also uses an axiomatic property (immunity).

Big picture, this paper completely resolves the extent to which control-type pairs collapse or fail to collapse, both in general (i.e., universally quantified over election systems) and with regard to plurality, veto, and approval elections.

2. Preliminaries

For sets A and B, $A - B$ denotes $A \cap \overline{B}$, i.e., "$-$" denotes the set difference operator.

For consistency, some of the standard definitions that appear in this section are taken, at times verbatim, from Hemaspaandra et al. (2020).

An election comprises a finite set C of candidates (each identified uniquely by a name[2]) and a finite collection V of votes over C. Except in one section of the paper (Section 3.3, where we will study a system using a different vote type, known as approval vectors), we throughout this paper take the "type" of a vote as being a linear ordering over C (note: linear orderings—complete, transitive, antisymmetric binary relations—are inherently tie-free). A simple example of a 3-candidate, 4-voter election thus is $C = \{a, b, c\}$ and the votes being $a > b > c$, $a > b > c$, $a > c > b$, and $b > c > a$.[3]

As is standard in computational social choice, an election system $\mathcal{E}$ maps an election (a pair (C, V)) to a (possibly nonstrict) subset of C (the set of winners). As is often done in computational social choice papers, we do not forbid the case of having no winners; see Hemaspaandra et al. (2020, Footnote 3) for a discussion of why allowing that is natural. The specific election systems plurality, veto, and approval are widely known and studied; we will explicitly define each in the results section devoted to it.

The study of electoral control from a computational perspective was initiated by Bartholdi et al. (1992). Their notion, known as constructive control, focuses on making a particular candidate a winner by some "control" action. Hemaspaandra et al. (2007) define the natural variants of those where instead the goal is to *prevent* a particular candidate from winning. This is known as destructive control.

The control actions are: partition of voters, partition of candidates, run-off partition of candidates, deleting voters, deleting candidates, adding voters, and adding candidates (both limited and unlimited). For each partition-based control type, we have two subvariants to handle the outcome of the subelections: in the TE (ties eliminate) subvariant, a candidate proceeds to the final round exactly if that candidate is the unique winner of that subelection, and in the TP (ties promote) subvariant, every winner of a subelection (including nonunique winners) proceeds to the final round. Hemaspaandra et al. (2020) suggest that it is more natural to study the TE subvariant when dealing with the unique-winner model and the TP subvariant when dealing with the nonunique-winner model. In this paper, since our goal is to uncover all possible collapses and separations (within the standard control types and their variants), we consider both TE and TP subvariants, regardless of the winner model. Since the nonpartition types among the standard control types do not involve two-stage elections, for those we of course will not split them into TE and TP versions, since that would not make sense.

2. By allowing names we potentially allow the names to be nefariously exploited by election systems. However, that model as to the use of names in fact makes our collapses and containments *stronger* than if those results were in a model where (candidate-)neutrality is assumed/required (the same applies to all collapses from Hemaspaandra et al. (2020), as those results are in this same with-names model). Although use of names would make our separations weaker, we address that by having ensured that every separation we establish in this paper is achieved via a (candidate-)neutral election system. In contrast, the one separation proven in Hemaspaandra et al. (2020) uses a system that is not (candidate-)neutral, see the discussion in our Related Work section.
3. The treatment of V as a multiset—which we use to match Hemaspaandra et al.'s (2020) models—in effect limits the election systems the model covers to so-called (voter-)anonymous election systems. In fact, all general-case collapses of Hemaspaandra et al. (2020) and all new general-case containments of our paper also hold (in the natural nonanonymous model) for all nonanonymous election systems, as one can easily see from those results' proofs. Also, we mention that all three concrete election systems that we study in fact *are* anonymous.

We take the following definition from Hemaspaandra et al. (2020) (which itself is taking parts from earlier papers) to help us define the (constructive) control problems (in the nonunique-winner model) succinctly.[4]

Definition 2.1 (see Hemaspaandra et al., 2020)**.** *Let $\mathcal{E}$ be an election system.*

1. *In the **constructive control by adding candidates** problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-AC-NUW), we are given two disjoint sets of candidates C and A, V a collection of votes over $C \cup A$, a candidate $p \in C$, and a nonnegative integer k. We ask if there is a set $A' \subseteq A$ such that (i) $\|A'\| \leq k$ and (ii) p is a winner of $\mathcal{E}$ election $(C \cup A', V)$.*

2. *In the **constructive control by unlimited adding candidates** problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-UAC-NUW), we are given two disjoint sets of candidates C and A, V a collection of votes over $C \cup A$, and a candidate $p \in C$. We ask if there is a set $A' \subseteq A$ such that p is a winner of $\mathcal{E}$ election $(C \cup A', V)$.*[5]

3. *In the **constructive control by deleting candidates** problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-DC-NUW), we are given an election (C, V), a candidate $p \in C$, and a nonnegative integer k. We ask if there is a set $C' \subseteq C$ such that (i) $\|C'\| \leq k$, (ii) $p \notin C'$, and (iii) p is a winner of $\mathcal{E}$ election $(C - C', V)$.*

4. *In the **constructive control by adding voters** problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-AV-NUW), we are given a set of candidates C, two collections of votes, V and W, over C, a candidate $p \in C$, and a nonnegative integer k. We ask if there is a collection $W' \subseteq W$ such that (i) $\|W'\| \leq k$ and (ii) p is a winner of $\mathcal{E}$ election $(C, V \cup W')$.*

5. *In the **constructive control by deleting voters** problem for $\mathcal{E}$ (denoted by $\mathcal{E}$-CC-DV-NUW), we are given an election (C, V), a candidate $p \in C$, and a nonnegative integer k. We ask if there is a collection $V' \subseteq V$ such that (i) $\|V'\| \leq k$ and (ii) p is a winner of $\mathcal{E}$ election $(C, V - V')$.*

6. *In the **constructive control by partition of voters** problem for $\mathcal{E}$, in the TP or TE tie-handling rule model (denoted by $\mathcal{E}$-CC-PV-TP-NUW or $\mathcal{E}$-CC-PV-TE-NUW,*

4. Regarding the partition-based control types, in Definition 2.1 and elsewhere in the paper we sometimes will speak of elections whose candidate set is C' but whose votes, due to candidate partitioning and/or first-round candidate eliminations, are over a set $C \supseteq C'$. As is standard in the literature, in such cases we always take this to mean that the votes are each restricted to just the candidates in C'.

5. In Definition 2.1, parts 1 and 3–5 each have among their inputs a parameter k that limits the number of additions or deletions. Among those four parts, why does only part 1 have an "unlimited" variant (part 2 of the definition) included in the standard set of control types? The answer comes from the history of control. The seminal paper on control (Bartholdi et al., 1992) defined addition/deletion of voters and deletion of candidates with a limiting parameter k, but in a curious asymmetry—perhaps due to which version the authors had obtained results for—defined and studied adding candidates in the unlimited variant. Later papers (the first of them being Hemaspaandra et al., 2007) ended the omission by studying the version of adding candidates that had a limiting parameter k. However, most papers—including the papers this paper is most closely related to—broadly investigating or "score-carding" properties of a "standard" set of control types have, in addition to studying that with-limiting-parameter-k version of adding candidates, also retained and studied the unlimited version. They likely did that in part since the unlimited version had been introduced in the seminal paper on control, and in part to support comparisons with other papers that included the unlimited version.

respectively), we are given an election (C, V), and a candidate $p \in C$. We ask if there is a partition[6] *of V into V_1 and V_2 such that p is a winner of the two-stage election where the winners of subelection (C, V_1) that survive the tie-handling rule compete (with respect to vote collection V) along with the winners of subelection (C, V_2) that survive the tie-handling rule. Each election (in both stages) is conducted using election system $\mathcal{E}$.*

7. *In the* ***constructive control by run-off partition of candidates*** *problem for $\mathcal{E}$, in the TP or TE tie-handling rule model (denoted by $\mathcal{E}$-CC-RPC-TP-NUW or $\mathcal{E}$-CC-RPC-TE-NUW, respectively), we are given an election (C, V), and a candidate $p \in C$. We ask if there is a partition of C into C_1 and C_2 such that p is a winner of the two-stage election where the winners of subelection (C_1, V) that survive the tie-handling rule compete (with respect to vote collection V) against the winners of subelection (C_2, V) that survive the tie-handling rule. Each election (in both stages) is conducted using election system $\mathcal{E}$.*

8. *In the* ***constructive control by partition of candidates*** *problem for $\mathcal{E}$, in the TP or TE tie-handling rule model (denoted by $\mathcal{E}$-CC-PC-TP-NUW or $\mathcal{E}$-CC-PC-TE-NUW, respectively), we are given an election (C, V), and a candidate $p \in C$. We ask if there is a partition of C into C_1 and C_2 such that p is a winner of the two-stage election where the winners of subelection (C_1, V) that survive the tie-handling rule compete (with respect to vote collection V) against all candidates in C_2. Each election (in both stages) is conducted using election system $\mathcal{E}$.*

There are 11 control "types" listed above (applied regarding generic election system $\mathcal{E}$). For each, we can change "is a winner" to "is an untied (i.e., unique) winner"; for those 11 variants, we change the "-NUW" into "-UW." Thus we have 22 control types. Those 22 are all trying to make the focus candidate win. And so they are all spoken of as "constructive" control types (thus the "CC" in their naming strings). Finally, for each of those now 22 control types, we can ask whether one can ensure that the focus candidate is *not* a winner or *not* a unique winner; those are known as the "destructive" control variants, and in the names of those, the CC is replaced by a DC, e.g., $\mathcal{E}$-DC-AC-UW.

We thus have 44 total types of control, which we will view as the "standard" control types. We thus have $\binom{44}{2} = 946$ pairs of control types. Fortunately, 624 of those pairs are incompatible—the two control types have different input fields from each other[7] and so comparing them would not even make sense—and so we will exclude them from our study. So we have "only" 322 pairs to focus on in our study. Table 2 shows the five compatibility equivalence classes that the 44 types partition into.

When speaking of control types we often will be speaking of the control model itself, and when doing so, we generally do not include the "$\mathcal{E}$-" prefix. In some sense, we view,

6. A partition of a collection V is a pair of collections V_1 and V_2 such that $V_1 \cup V_2 = V$, where $\cup$ denotes multiset union. A partition of a set C is a pair of sets C_1 and C_2 such that $C_1 \cup C_2 = C$ and $C_1 \cap C_2 = \emptyset$, where $\cup$ and $\cap$ are standard set union and intersection.

7. So for example CC-PC-TP-NUW and CC-RPC-TE-UW are compatible, but CC-PC-TP-NUW and CC-AV-NUW are not, since CC-AV-NUW has a nonnegative integer field k. Throughout this paper, whenever we speak of pairs of control types, we refer only to compatible pairs (even if that is not explicitly stated, although for emphasis we often do state it).

for example, CC-AC-NUW as a control type (model)—one among the 44 standard such models—and *we view* $\mathcal{E}$-CC-AC-NUW *as the set of input strings that are "yes" instances, for election system* $\mathcal{E}$*, under that model of control.*

Keeping that in mind, we now define precisely what we mean by two (compatible) control types collapsing or separating, and then introduce a function approach that will allow us to explore in a more refined way the nature of the separations. Let $\mathcal{E}$ (e.g., Plurality) be an election system and let $\mathcal{T}$ and $\mathcal{T}'$ be two (compatible) control types from our 44 standard ones (e.g., CC-AC-NUW and CC-AC-UW). Then if $\mathcal{E}$-$\mathcal{T} = \mathcal{E}$-$\mathcal{T}'$ we say that the control types $\mathcal{T}$ and $\mathcal{T}'$ collapse (for election system $\mathcal{E}$), and if $\mathcal{E}$-$\mathcal{T} \neq \mathcal{E}$-$\mathcal{T}'$ we say that the control types $\mathcal{T}$ and $\mathcal{T}'$ separate (for election system $\mathcal{E}$). We will also use the terms collapse and separate for a more general case, namely, the one of quantifying universally over all election systems whose votes are linear orders. When speaking in that case, we will say that two (compatible) control types from among our 44 collapse if the two control types collapse for every election system $\mathcal{E}$ whose vote type is linear orders, and otherwise we will say that the control types separate.

As an example, if we fix the election system $\mathcal{E}$ to be plurality, we note that in the 3-candidate, 4-voter election example given at the beginning of this section a is a unique winner. Yet, we could make b a unique winner by deleting candidate a, hence we have $(C, V, b, 1) \in$ Plurality-CC-DC-UW. Also, since a unique winner is a winner in the nonunique-winner model as well, we have $(C, V, b, 1) \in$ Plurality-CC-DC-NUW. (In this paper we will not focus on encoding details, since they are not important to our study.)

Note that, in this paper, when we prove that two compatible control types separate we are unconditionally proving that they are not the same set. That contrasts with the weaker—namely, conditional—separation one might claim if one of the types was known to be NP-hard and the other was known to be a set (other than the empty set or Σ^*) in P, since arguing separation just from those facts would require one to have proven that P $\neq$ NP. Also, one needs to be wary about casually arguing that certain (compatible) control types "inherently," universally differ. For example, it might be tempting to say just based on intuition that CC-DV-UW and CC-DC-NUW inherently differ for all election systems. However, for the election system where all candidates always lose, these two do collapse, since both are the empty set. Also, the already-known universally collapsing control pairs found by Hemaspaandra et al. (2020) were long assumed to be obviously different, but that turned out to not be the case. (However, though in this paper we do not look at showing when pairs of compatible types will differ for every election system, we mention that in some cases one can indeed see that that holds. For example, for each election system $\mathcal{E}$, $\mathcal{E}$-CC-DV-NUW and $\mathcal{E}$-DC-DV-NUW differ since any given legal, syntactically well-formed input with one candidate and zero votes, that input will be in exactly one of the two sets.)

Compatible control types that separate—i.e., are different sets—can do so in different ways, some of which reflect containment relationships. In order to be able to seek such relationships, we define the three different ways that two (compatible) separating control types, $\mathcal{T}$ and $\mathcal{T}'$, for $\mathcal{E}$ can separate. To support this refinement, we introduce a function model. In particular, we will define functions that, for a given control type, map from inputs (that differ from those used so far for that control type only in *not* having a focus candidate specified) to the set of all candidates c such that if c is made the focus candidate for that

input, successful control is possible. In a bit more detail: Each of our control types has certain inputs, and all include a focus candidate, c. Let us for any of our 44 control types, $\mathcal{T}$, refer to an input to it, except with the field containing the focus candidate removed, as the reduced form of that input. If we want for a reduced input to add back the name of a particular candidate to be the focus candidate, we will say that is inflating the reduced input by that candidate. For any of our 44 control types, $\mathcal{T}$, and any election system $\mathcal{E}$, we define the function $f_{\mathcal{E}\text{-}\mathcal{T}}$ to be the function that, for a given reduced input $\tilde{I}$, outputs the set of all candidates c such that $\tilde{I}$ inflated by c belongs to the set $\mathcal{E}$-$\mathcal{T}$. For example, given as input $(C, V, 3)$, the output of $f_{\mathcal{E}\text{-DC-DV-NUW}}$ is the set of all candidates c that can by deleting less than or equal to three votes be prevented from being a winner.

Clearly, two (compatible) control types collapse exactly if their thus-defined functions are equal. But for two separating (compatible) control types, there are three different ways they can be separated. One way is if, for each reduced input $\tilde{I}$: (a) $f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) \subseteq f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I})$ and (b) for some reduced input $\tilde{I}'$, $f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}') - f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}') \neq \emptyset$. We will (in a slight abuse of notation) refer to the case where (a) holds as the "$\subseteq$" case, and the case where both (a) and (b) hold as the "$\subsetneq$" case. The "$\supseteq$" and "$\supsetneq$" cases are analogously defined. If the two (compatible) types separate but neither the "$\subsetneq$" case nor the "$\supsetneq$" case holds, we will say the two types are incomparable: each will sometimes have successful focus candidates that the other does not. If both directions of noncontainment can be witnessed by the same reduced input, we will say the two types are strongly incomparable, i.e., there is a reduced input $\tilde{I}$ such that $f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) - f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}) \neq \emptyset$ and $f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}) - f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) \neq \emptyset$. Arguably, strong incomparability is a more satisfying strength of incomparability, since it makes clear that one does not have to cobble the two directions of the incomparability together from different supporting instances. These notions are relative to each specific election system, $\mathcal{E}$.

For the general case—where our collapses are universally quantified over all election systems whose vote type is linear orders—we define three increasingly strong types of incomparability. The weakest incomparability notion is simply that for at least one such election system $\mathcal{E}$, for some reduced input $\tilde{I}$, $f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) - f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}) \neq \emptyset$ holds, and for at least one such election system $\mathcal{E}'$, for some reduced input $\tilde{I}'$, $f_{\mathcal{E}'\text{-}\mathcal{T}'}(\tilde{I}') - f_{\mathcal{E}'\text{-}\mathcal{T}}(\tilde{I}') \neq \emptyset$ holds. We will call this being weakly incomparable.[8] We will say the pair is incomparable if there is some election system, over linear orders, in which the pair is incomparable in the sense of the previous paragraph. And we will say the pair is strongly incomparable if there is some election system, over linear orders, in which the pair is strongly incomparable in the sense of the previous paragraph. For the general case, we will say "$\subseteq$" (resp., "$\supseteq$") holds if for every election system $\mathcal{E}$ whose vote type is linear orders, the "$\subseteq$" (resp., "$\supseteq$") case for $\mathcal{E}$, as defined above, holds.

We now cover, and give brief reference labels to (though we at times may use these inheritances tacitly when the use is clear), some collapse, containment, and separation inheritances that hold. We will start these labels with the letter "I" as a mnemonic for that fact that these concepts are about inheritance. General-case collapse, $\subseteq$, and $\supseteq$ results

8. We will state no weak incomparability results in this paper, since whenever we obtained a weak incomparability result we in fact were able to even establish incomparability, which we will define in a moment in the main text. Though in this paper we thus never need to prove weak incomparability results, we state the notion so that future research on other control types will have a three-level menu of incomparability strengths to draw on if needed.

imply, for each election system over linear orders, resp., collapse, $\subseteq$, and $\supseteq$ results (let us shorthand that fact as I1). If a given general case result of this type *happens* to in addition hold for every election system (not merely those over linear orders), we will refer to that as additionally being an I1$^+$ case. Simply because their particular proofs do not ever draw on the vote types, the seven general-case collapses of Hemaspaandra et al. (2020), and also our general containment results of Theorem 3.1, hold even as I1$^+$ cases. As to separation inheritances, if for some election system $\mathcal{E}$ over linear orders and some reduced input $\tilde{I}$ we have that $f_{\mathcal{E}\text{-}\mathcal{T}'}(\tilde{I}) - f_{\mathcal{E}\text{-}\mathcal{T}}(\tilde{I}) \neq \emptyset$ (as, crucially, is always the case if we have that for $\mathcal{E}$ the $\subsetneq$ relation holds between $\mathcal{T}$ and $\mathcal{T}'$) and we in addition happen to have that in the general case the $\subseteq$ relation is known to hold between $\mathcal{T}$ and $\mathcal{T}'$, then we may conclude that $\subsetneq$ holds in our general case; the analogous claim holds for the $\supsetneq$ case, and we will refer to either of these, when they hold, as an I2 inheritance case. Incomparability in an election system (over linear orders) implies general-case incomparability (we will refer to this as I3), and strong incomparability in an election system (over linear orders) implies general-case strong incomparability (I3*).

3. Results

In the following three subsections, for the general case and for the cases of plurality, veto, and approval voting, we completely determine which pairs of (compatible) standard control types collapse, and which separate. Beyond that, we refine every separation into one of the three disjoint cases: $\subsetneq$, $\supsetneq$, and incomparability.

We discover a number of previously unknown collapses (for two of the three specific systems) and also for many noncollapsing control-type pairs establish new containments in one direction. Regarding the former, for veto we discover a new collapsing pair (Veto-DC-PV-TE-NUW = Veto-DC-PV-TE-UW) and for approval we extend a four-type collapse by Hemaspaandra et al. (2020) to a six-type collapse, and we find five additional collapsing pairs, for a total of $\binom{6}{2} + 5 - \binom{4}{2}$, i.e., 14, new collapsing control-type pairs for approval.

Overall, we establish that, for the $4 \times 322 = 1{,}288$ cases we study (i.e., all compatible pairs for each of the four cases), no containments or collapses exist other than those provided by either Hemaspaandra et al. (2020) or this paper.

3.1 The General Case and Plurality

We show that the only (compatible) pairs that collapse in general (i.e., that collapse for every election system) are the seven found by Hemaspaandra et al. (2020), namely, for every election system $\mathcal{E}$, $\mathcal{E}$-DC-RPC-TE-NUW = $\mathcal{E}$-DC-RPC-TE-UW = $\mathcal{E}$-DC-PC-TE-NUW = $\mathcal{E}$-DC-PC-TE-UW (six pairs) and $\mathcal{E}$-DC-RPC-TP-NUW = $\mathcal{E}$-DC-PC-TP-NUW (one pair). Surprisingly, we will be able to do so using just constructions about plurality elections, combined with uses of inheritance.

In a plurality election, for each vote where a particular candidate is ranked first, that candidate receives a point, and a winner is a candidate with the highest number of points among all the candidates (naturally, there can be multiple winners). The votes in plurality elections are linear orders. Recall that by the "general case," we mean the general case with respect to elections where the votes are linear orders. Thus to separate two control types in the general case, it certainly suffices to show that they separate under plurality, as such

separations inherit back to the general case via the I3 and I2 types of inheritance discussed in our preliminaries.[9] The reason we know that no separation for the general case is missed is that our results show that plurality has no $\subseteq$, $\supseteq$, or collapse results not also possessed by the general case; so our I2 cases are valid, and every general-case $\subsetneq$, $\supsetneq$, or incomparability (or even strong incomparability) that holds is yielded by our inheritances.

Let us now turn our attention to the two new general-case containments shown in this paper (these containments are in addition to the obvious ones, noted at location (∗∗) of Section 1, about two control types that only differ in their winner model) and also argue that both are strict in the general case (by which, recall, we mean that there is at least one election system under which the containment is not an equality). The following two containments apply to all vote types (not just linear orders; and we argue in Section 3.3 that the collapses by Hemaspaandra et al., 2020 also apply to all vote types).

Theorem 3.1. *Let $\mathcal{E}$ be an election system. For each $\mathcal{T} \in \{$DC-RPC-TP-UW, DC-PC-TP-UW$\}$, $\mathcal{E}$-$\mathcal{T} \subseteq \mathcal{E}$-DC-RPC-TE-NUW.*

Proof. Let $\mathcal{T} \in \{$DC-RPC-TP-UW, DC-PC-TP-UW$\}$. We will show $\mathcal{E}$-$\mathcal{T} \subseteq \mathcal{E}$-DC-RPC-TE-NUW. Suppose $(C, V, p) \in \mathcal{E}$-$\mathcal{T}$. Let (C_1, C_2) be a partition that witnesses (C, V, p)'s membership in $\mathcal{E}$-$\mathcal{T}$. It holds that either p participates in and loses in a subelection, or p participates in and does not win uniquely in the final round. If the former holds, then the partition (C_1, C_2) suffices as p will be eliminated in a subelection and thus will not proceed to the final round. If the latter holds, then let D denote the set of candidates present during the final round. Clearly $p \in D$. Thus the partition $(D, C - D)$ witnesses (C, V, p)'s membership in $\mathcal{E}$-DC-RPC-TE-NUW since p either loses in the first subelection (D, V) or ties in that subelection and is eliminated by the tie-handling rule. ❑

For both containments above (i.e., for each of the two values of $\mathcal{T}$ stated in the theorem), Table 5 contains a pointer to a separation witness in Table 4 that shows that the containment is not an equality in the general case (i.e., that for some election system—in fact, plurality—the containment is strict). More generally, Table 5, for each compatible pair of control types $\mathcal{T}$ and $\mathcal{T}'$, gives us an election witnessing Plurality-$\mathcal{T}$ − Plurality-$\mathcal{T}' \neq \emptyset$ if that holds, and gives an election witnessing Plurality-$\mathcal{T}'$ − Plurality-$\mathcal{T} \neq \emptyset$ if that holds (and if both hold Table 5 gives witnesses for each). Of course, we are not claiming that the containments in that theorem are strict under all election systems; for example, Corollary 3.11 and Theorem 3.12 show that all three types mentioned in the above theorem pairwise collapse under approval voting.

Some of the constructions in Table 4 are quite simple. For example, with $C = \{a, b\}$ and $V = \{a > b\}$ (denoted by "Plur.3") we clearly get incomparability between all 144 pairs of partition types where one type is constructive and the other is destructive. (This holds since in that election, under every constructive or destructive partition-control action, a is the unique winner and b is not a winner.) However, showing separations for every pair that can be separated is no trivial matter. Some of our separation examples for plurality are quite large, with up to 18 votes and up to seven candidates. The search for those examples was computer-aided and extensive.

9. We note that for some cases we achieve strong incomparability in our general-case results, always via I3*. In such cases, incomparability still holds, since strong incomparability is simply a specific subcase of incomparability.

3.2 Veto

In a veto election, for each vote where a particular candidate is not ranked last, that candidate receives a point, and a winner is a candidate with the highest number of points among all the candidates (naturally, there can be multiple winners). For example, if $C = \{a, b, c\}$ and $V = \{a > b > c, c > a > b\}$, then b and c each receive one point, and a receives two points and wins.

We now establish every equality and containment that holds for veto elections but was not established by one of the results of Hemaspaandra et al. (2020). For all other veto cases, we have constructed counterexamples. Some of those counterexamples were obtained through computer search.

Theorem 3.2. Veto-DC-PV-TE-UW = Veto-DC-PV-TE-NUW.

Proof. The $\supseteq$ relationship is immediate. The approach that follows resembles closely that of Maushagen and Rothe (2018). Let $(C, V, p) \in$ Veto-DC-PV-TE-UW. If there is a partition such that p is not a winner of the two-stage election, then $(C, V, p) \in$ Veto-DC-PV-TE-NUW. Otherwise, there is a partition and $c \in C$, $c \neq p$, such that p and c are both winners of the two-stage election. Suppose $\|C\| = 2$. Then p and c are both winners of the election (C, V). In this case, consider the partition $(V, \emptyset)$. Both candidates will tie and be eliminated, in both subelections. Suppose $\|C\| \geq 3$. Then there are two distinct candidates $d, e \in C - \{p\}$. Let V_1 denote the set of votes in which e is vetoed and let V_2 be the remaining votes. Now, consider the two-stage election with partition (V_1, V_2). Since e is never vetoed in V_2, p can at best tie with e in the subelection (C, V_2). Finally, d is never vetoed in V_1 (since all votes there veto e), so p cannot be a unique winner of (C, V_1). Thus p is eliminated in both first-round elections. ❑

Theorem 3.3. *For each* $\mathcal{T} \in$ {DC-PV-TP-UW, DC-PV-TP-NUW}, Veto-$\mathcal{T} \subsetneq$ Veto-DC-PV-TE-NUW.

Proof. Let $\mathcal{T} \in$ {DC-PV-TP-UW, DC-PV-TP-NUW}. Suppose $(C, V, p) \in$ Veto-$\mathcal{T}$, and let (V_1, V_2) be a partition that witnesses this membership. Using the same technique as in the proof of Theorem 3.2, we can handle the case where $\|C\| \geq 3$ and thus we only need to consider the case where $\|C\| = 2$ in this proof. Let $C = \{c, p\}$. First, suppose that p is present in the final round. Then, since p is not a unique winner of the final round, all candidates in C are present in this round. Thus p is not a unique winner of the election (C, V), and so the partition $(V, \emptyset)$ witnesses (C, V, p)'s membership in Veto-DC-PV-TE-NUW. Now, suppose that p is absent from the final round. Then p is not a winner of (C, V_1) or (C, V_2), and so p receives strictly more vetoes than c in both V_1 and V_2. Thus p receives more vetoes than c in V, so again, p is not a unique winner of (C, V). A separation witness for the strict containment can be found in Table 7. ❑

Theorem 3.4. *For each* $\mathcal{T} \in$ {DC-RPC-TE-NUW, DC-RPC-TP-UW, DC-RPC-TP-NUW, DC-PC-TP-UW}, Veto-$\mathcal{T} \subsetneq$ Veto-DC-PV-TE-NUW.

Proof. Let $\mathcal{T} \in$ {DC-RPC-TE-NUW, DC-RPC-TP-UW, DC-RPC-TP-NUW, DC-PC-TP-UW}. Suppose $(C, V, p) \in$ Veto-$\mathcal{T}$, and let (C_1, C_2) be a partition that witnesses this membership. As per the two preceding proofs (namely, of Theorems 3.2 and 3.3), we need

only consider the case where $\|C\| = 2$. The case where p is present in the final round is handled as in Theorem 3.3, so suppose that p is absent from the final round. Without loss of generality, we may assume that $p \in C_1$. (This is certainly true when $\mathcal{T}$ is about one of the theorem's three RPC cases, due to the symmetric nature of that partitioning action. It also holds when $\mathcal{T}$ is about the theorem's one PC case, since p being in C_2 would imply that p is in the final round, which contradicts the case we are in.) Then p is not a unique winner of (C_1, V), so $C_1 = C$. Thus p is not a unique winner of (C, V), and the partition $(V, \emptyset)$ witnesses (C, V, p)'s membership in Veto-DC-PV-TE-NUW. A separation witness for the strict containment can be found in Table 7. ❑

3.3 Approval Voting

Approval voting differs from the other election systems discussed so far in this paper as to its vote type. In an approval election (C, V), each vote is a bit-vector of length $\|C\|$, with each bit being associated with a candidate. If a candidate's bit is 1, then that candidate is approved by that vote. In approval voting, the winner set is composed of each candidate c such that no other candidate is approved by strictly more votes than c is. We will sometimes speak of the "score" of a candidate in the rest of this section. In the context of approval voting, the score of a candidate in an election is the number of votes that approve that candidate (and as noted above, the set of winners are those candidates with maximal score).

Although the collapses shown by Hemaspaandra et al. (2020) were stated for election systems where votes are linear orders, we note that in their proof they do not use the vote type and thus those collapses hold regardless of what type of votes the election system is over. Thus we have the following corollary.

Corollary 3.5 (see Hemaspaandra et al., 2020)**.** *1.* Approval-DC-RPC-TE-UW = Approval-DC-RPC-TE-NUW = Approval-DC-PC-TE-NUW = Approval-DC-PC-TE-UW.

2. Approval-DC-RPC-TP-NUW = Approval-DC-PC-TP-NUW.

In this section, we prove a number of collapses and inclusions that hold for approval voting. Some of these results draw on previously established immunity[10] arguments that draw on properties satisfied by approval voting, thereby allowing us to generalize our results. Other results are provided by direct arguments. Our direct arguments often rely on the fact that, since the votes are bit-vectors, a candidate's score in an election is independent of the other candidates present in the election. We will start by discussing the results that draw on the aforementioned immunity arguments.

10. In the unique-winner model, we say an election system is immune to a particular type of control if the given type of control can never change a candidate from not uniquely winning to uniquely winning (if the control type is constructive) or change a candidate from uniquely winning to not uniquely winning (if the control type is destructive) (Hemaspaandra et al., 2007, which clarified a slightly flawed immunity definition of Bartholdi et al., 1992). In the nonunique-winner model, we say an election system is immune to a particular type of control if the given type of control can never change a nonwinner to a winner (if the control type is constructive) or change a winner to a nonwinner (if the control type is destructive).

Let us first consider what is known in the social choice literature as Property α (Chernoff, 1954, see also Sen, 1971).[11] Property α is defined as the property that p winning an election (C, V) implies that p remains a winner of every election (C', V) for which $p \in C' \subseteq C$. (Clearly, some election systems will satisfy this property and some will not.) The "unique" version of this property (which is called Property Unique-α) only differs in that it requires p to be a unique winner, i.e., it requires that p uniquely winning in an election (C, V) implies that p uniquely wins in each election (C', V) for which $p \in C' \subseteq C$ (Hemaspaandra et al., 2007). Hemaspaandra et al. (2007) note that any election system that satisfies Property Unique-α is immune to several types of control, namely to (i) destructive control by partition of candidates and run-off partition of candidates (under both TP and TE tie-handling rules) in the UW model, and (ii) destructive control by deleting candidates in the UW model. From this, we obtain the following results.[12]

Theorem 3.6. *Let $\mathcal{E}$ be an election system that satisfies Property Unique-α. Then $\mathcal{E}$-DC-PC-TP-UW = $\mathcal{E}$-DC-PC-TE-UW.*

Proof. Fix any $\mathcal{E}$ that satisfies Property Unique-α. Let $\mathcal{E}\text{-}\mathcal{T}_1 = \mathcal{E}$-DC-PC-TP-UW and let $\mathcal{E}\text{-}\mathcal{T}_2 = \mathcal{E}$-DC-PC-TE-UW. Consider the two sets $A_{\mathcal{E}} = \{(C, V, p) \mid p \in C$ and p is a unique winner of $\mathcal{E}$ election $(C, V)\}$ and $B_{\mathcal{E}} = \{(C, V, p) \mid p \in C$ and p is not a unique winner of $\mathcal{E}$ election $(C, V)\}$. These sets form a partition of $Y = \{(C, V, p) \mid p \in C$ and (C, V) is an $\mathcal{E}$ election$\}$, so of course $Y = A_{\mathcal{E}} \cup B_{\mathcal{E}}$. Clearly we also have that $\mathcal{E}\text{-}\mathcal{T}_1 \subseteq Y$ and that $\mathcal{E}\text{-}\mathcal{T}_2 \subseteq Y$. We will argue that $\mathcal{E}\text{-}\mathcal{T}_1 = B_{\mathcal{E}} = \mathcal{E}\text{-}\mathcal{T}_2$. Since $\mathcal{E}$, like all systems satisfying Property Unique-α, is immune to both control types in the theorem statement, we have (recall that both of these types are destructive types) that $\mathcal{E}\text{-}\mathcal{T}_1 \cap A_{\mathcal{E}} = \emptyset$ and $\mathcal{E}\text{-}\mathcal{T}_2 \cap A_{\mathcal{E}} = \emptyset$, and thus it holds that $\mathcal{E}\text{-}\mathcal{T}_1 \subseteq B_{\mathcal{E}}$ and $\mathcal{E}\text{-}\mathcal{T}_2 \subseteq B_{\mathcal{E}}$. Fix $(C, V, p) \in B_{\mathcal{E}}$. Then the partition $(\emptyset, C)$ witnesses both $(C, V, p) \in \mathcal{E}\text{-}\mathcal{T}_1$ and $(C, V, p) \in \mathcal{E}\text{-}\mathcal{T}_2$, since in both cases the final round will simply be (C, V) and we know that since $(C, V, p) \in B_{\mathcal{E}}$, p will not be a unique winner of the final round. ❑

Theorem 3.7. *Let $\mathcal{E}$ be an election system that satisfies Property Unique-α. Then the following hold.*

11. In a number of papers in the computational social choice literature, this property has been spoken of as if it were the Weak Axiom of Revealed Preference. It in fact is not, though they are related; this issue however did not make any of the earlier papers wrong, as they merely were unwisely but internally consistently using a notation in a way that did not match the way it was used elsewhere. This unfortunate terminological confusion came in due to some ambiguous wording in one of the seminal papers on computational social choice theory. In the present paper, we follow the lead of Carleton, Chavrimootoo, Hemaspaandra, Narváez, Taliancich, and Welles (2022, 2024) and use the (more consistent with the social choice literature) terminology "Property α" and, for Property α's "unique-winner" analogue, "Property Unique-α." See Carleton et al. (2022) for a more detailed discussion of this issue, including its history and relation to the Weak Axiom of Revealed Preference.

12. The theorem certainly could also include $\mathcal{E}$-DC-RPC-TE-UW in the equality. However, including that in the theorem would make little sense, since $\mathcal{E}$-DC-RPC-TE-UW = $\mathcal{E}$-DC-PC-TE-UW does not rely on Property Unique-α, but in fact holds for *all* election systems in light of Hemaspaandra et al. (2020) and our comment in the paragraph before Corollary 3.5.

    Also, note that in our sequence of result statements from Theorem 3.6 through Corollary 3.10, each is (intentionally) stated as applying to all election systems satisfying the relevant Property α or Unique-α, regardless of what vote "type"—such as approval vectors or linear orders—the election system's votes are over.

1. $\mathcal{E}$-DC-DC-UW $\subseteq$ $\mathcal{E}$-DC-DV-UW.

2. $\mathcal{E}$-DC-DC-NUW $\subseteq$ $\mathcal{E}$-DC-DV-UW.

Proof. Fix any $\mathcal{E}$ that satisfies Property Unique-α.

1. The proof is analogous to that of Theorem 3.6. Since $\mathcal{E}$ is immune to destructive control by deleting candidates in the unique-winner model (as per the discussion immediately before Theorem 3.6), the inclusion is immediate as $\mathcal{E}$-DC-DC-UW is comprised of those inputs where the focus candidate is already not a unique winner.

2. It trivially holds that $\mathcal{E}$-DC-DC-NUW $\subseteq$ $\mathcal{E}$-DC-DC-UW. Thus the proof follows from the previous part of this theorem. ❑

Theorem 3.8. *Let $\mathcal{E}$ be an election system that satisfies Property α. Then* $\mathcal{E}$-DC-DC-NUW $\subseteq$ $\mathcal{E}$-DC-DV-NUW.

Proof. Fix any $\mathcal{E}$ that satisfies Property α. The proof is analogous to that of Theorem 3.7. Given an election (C, V) and $p \in C$, if p is a winner of $\mathcal{E}$ election (C, V), by Property α it follows that no deletion of candidates (other than p) can prevent p from being a winner. Thus $\mathcal{E}$-DC-DC-NUW is comprised of those inputs where the focus candidate is already not a winner. ❑

The next two results will use the notion of being strongly voiced. An election system $\mathcal{E}$ is said to be strongly voiced exactly if for every election (C, V) with $C \neq \emptyset$, there is at least one winner of (C, V) under $\mathcal{E}$ (Hemaspaandra et al., 2007). (Most earlier versions of this paper had different hypotheses in what here are Theorem 3.9 and Corollary 3.10 and flawed proofs of those; the current version corrects that.)

Theorem 3.9. *Let $\mathcal{E}$ be a strongly voiced election system that satisfies Property α. Then* $\mathcal{E}$-CC-PC-TP-UW $= \mathcal{E}$-CC-RPC-TP-UW.

Proof. Fix any $\mathcal{E}$ that is strongly voiced and satisfies Property α. Let $\mathcal{E}$-$\mathcal{T}_1 = \mathcal{E}$-CC-PC-TP-UW and let $\mathcal{E}$-$\mathcal{T}_2 = \mathcal{E}$-CC-RPC-TP-UW. Consider the two sets $A_{\mathcal{E}} = \{(C, V, p) \mid p \in C$ and p is a unique winner of $\mathcal{E}$ election $(C, V)\}$ and $B_{\mathcal{E}} = \{(C, V, p) \mid p \in C$ and p is not a unique winner of $\mathcal{E}$ election $(C, V)\}$. These sets form a partition of $Y = \{(C, V, p) \mid p \in C$ and (C, V) is an $\mathcal{E}$ election$\}$, so of course $Y = A_{\mathcal{E}} \cup B_{\mathcal{E}}$. Clearly we also have that $\mathcal{E}$-$\mathcal{T}_1 \subseteq Y$ and that $\mathcal{E}$-$\mathcal{T}_2 \subseteq Y$. We will show that $\mathcal{E}$-$\mathcal{T}_1 = A_{\mathcal{E}} = \mathcal{E}$-$\mathcal{T}_2$. First we show that $\mathcal{E}$-$\mathcal{T}_1 \subseteq A_{\mathcal{E}}$ and $\mathcal{E}$-$\mathcal{T}_2 \subseteq A_{\mathcal{E}}$.

Consider an element $(C, V, p) \in \mathcal{E}$-$\mathcal{T}_1$ and assume for the sake of contradiction that p is not a unique winner of $\mathcal{E}$ election (C, V), i.e., $(C, V, p) \in \mathcal{E}$-$\mathcal{T}_1 \cap B_{\mathcal{E}}$. Since $(C, V, p) \in B_{\mathcal{E}}$ and since $\mathcal{E}$ is strongly voiced, there is a candidate $d \in C - \{p\}$ that is a winner of (C, V) under $\mathcal{E}$. By Property α, that candidate is a winner of any subelection they participate in (regardless of how the candidates are partitioned), and since the tie-handling rule never eliminates candidates, d will always proceed to the final round and be a winner (again by Property α). This contradicts the fact that $(C, V, p) \in \mathcal{E}$-$\mathcal{T}_1$, so $\mathcal{E}$-$\mathcal{T}_1 \cap B_{\mathcal{E}} = \emptyset$. Since $\mathcal{E}$-$\mathcal{T}_1 \subseteq Y = A_{\mathcal{E}} \cup B_{\mathcal{E}}$ and since $\mathcal{E}$-$\mathcal{T}_1 \cap B_{\mathcal{E}} = \emptyset$, it follows that $\mathcal{E}$-$\mathcal{T}_1 \subseteq A_{\mathcal{E}}$. The same

argument establishes that $\mathcal{E}\text{-}\mathcal{T}_2 \subseteq A_{\mathcal{E}}$. We will now complete this proof by arguing that $A_{\mathcal{E}} \subseteq \mathcal{E}\text{-}\mathcal{T}_1$ and $A_{\mathcal{E}} \subseteq \mathcal{E}\text{-}\mathcal{T}_2$.

Fix $(C, V, p) \in A_{\mathcal{E}}$. Then the partition $(\emptyset, C)$ witnesses that $(C, V, p) \in \mathcal{E}\text{-}\mathcal{T}_1$ as the final round will simply be (C, V) and we know that since $(C, V, p) \in A_{\mathcal{E}}$, p will be a unique winner of the final round. Additionally, the partition $(\emptyset, C)$ also witnesses that $(C, V, p) \in \mathcal{E}\text{-}\mathcal{T}_2$ as no one will proceed from subelection $(\emptyset, V)$, and only p will proceed from subelection (C, V) (since $(C, V, p) \in A_{\mathcal{E}}$), and p must be the unique winner of the final round $(\{p\}, V)$ (this follows from both strong voicedness and Property α). ❑

We can build on Theorem 3.9 to show additional containments.

Corollary 3.10. *Let $\mathcal{E}$ be a strongly voiced election system that satisfies Property α and Property Unique-α. Then, for each $\mathcal{T} \in$ {CC-PC-TE-UW, CC-PC-TE-NUW, CC-RPC-TE-UW, CC-RPC-TE-NUW, CC-PV-TE-UW, CC-PV-TE-NUW, CC-PV-TP-UW, CC-PV-TP-NUW}, it holds that $\mathcal{E}$-CC-PC-TP-UW $\subseteq \mathcal{E}\text{-}\mathcal{T}$ (equivalently, $\mathcal{E}$-CC-RPC-TP-UW $\subseteq \mathcal{E}\text{-}\mathcal{T}$).*

Proof. Fix any $\mathcal{E}$ that is strongly voiced and that satisfies Property α and Property Unique-α, fix a $\mathcal{T} \in$ {CC-PC-TE-UW, CC-PC-TE-NUW, CC-RPC-TE-UW, CC-RPC-TE-NUW, CC-PV-TE-UW, CC-PV-TE-NUW, CC-PV-TP-UW, CC-PV-TP-NUW}, and fix $(C, V, p) \in \mathcal{E}$-CC-PC-TP-UW. By the proof of Theorem 3.9, $\mathcal{E}$-CC-PC-TP-UW is comprised of only those inputs where p is already a unique winner of the $\mathcal{E}$ election (C, V), so the trivial partition (i.e., $(\emptyset, C)$ if $\mathcal{T}$ is about partitioning candidates and $(\emptyset, V)$ if $\mathcal{T}$ is about partitioning votes) is a witness that $(C, V, p) \in \mathcal{E}\text{-}\mathcal{T}$ as p will uniquely win at least one subelection in which it participates (if $\mathcal{T}$ is about partitioning votes, then p participates in two subelections) as one of those two subelections is (C, V). Moreover, p uniquely wins the final round, which it participates in, because $\mathcal{E}$ satisfies Property Unique-α. The "equivalently" follows from Theorem 3.9. ❑

Since approval voting is clearly strongly voiced (Hemaspaandra et al., 2007) and satisfies Property Unique-α (Hemaspaandra et al., 2007), and (clearly) Property α, the above theorems and corollaries apply to approval voting. For each case where only the containment (and not the collapse) is shown, the containment can be made strict under approval voting. See Table 9 for the separation witnesses.

Corollary 3.11. *1.* Approval-DC-PC-TP-UW = Approval-DC-PC-TE-UW = Approval-DC-RPC-TE-UW = Approval-DC-RPC-TE-NUW = Approval-DC-PC-TE-NUW.

2. Approval-DC-RPC-TP-NUW = Approval-DC-PC-TP-NUW.

3. Approval-DC-DC-UW $\subsetneq$ Approval-DC-DV-UW.

4. Approval-DC-DC-NUW $\subsetneq$ Approval-DC-DV-NUW.

5. Approval-DC-DC-NUW $\subsetneq$ Approval-DC-DV-UW.

6. Approval-CC-PC-TP-UW = Approval-CC-RPC-TP-UW.

7. *For each* $\mathcal{T} \in \{$CC-PC-TE-UW, CC-PC-TE-NUW, CC-RPC-TE-UW, CC-RPC-TE-NUW, CC-PV-TE-UW, CC-PV-TE-NUW, CC-PV-TP-UW, CC-PV-TP-NUW$\}$, *it holds that* Approval-CC-PC-TP-UW $\subsetneq$ Approval-$\mathcal{T}$.

We can extend the five-type collapse in Part 1 of Corollary 3.11 to a six-type collapse.

Theorem 3.12. Approval-DC-RPC-TP-UW = Approval-DC-PC-TP-UW.

Proof. We will show that both sets coincide with the set $B = \{(C, V, p) \mid p \in C$ and p is not a unique winner of approval election $(C, V)\}$.

It follows from the fact that approval is immune with respect to DC-RPC-TP-UW and DC-PC-TP-UW (Hemaspaandra et al., 2007) that Approval-DC-RPC-TP-UW and Approval-DC-PC-TP-UW are both subsets of B. Fix $(C, V, p) \in B$. Then the partition $(\emptyset, C)$ witnesses that $(C, V, p) \in$ Approval-DC-PC-TP-UW as the final stage will be approval election (C, V). Similarly, the partition $(\emptyset, C)$ also witnesses that $(C, V, p) \in$ Approval-DC-RPC-TP-UW. This follows from the fact that p is not a unique winner of the approval election (C, V) and so there must exist a different candidate d who is a winner of election (C, V) and so whose score is at least as large as p's score. No one will proceed from subelection $(\emptyset, V)$ and d will proceed from subelection (C, V) (along with those candidates, if any, who tie with d) and be a winner of the final round. Thus even if p proceeds to the final round, it will not be a unique winner. ❑

Additionally, we prove the following immunity result about approval voting in the nonunique-winner model; the analogous result for the unique-winner model was obtained by Hemaspaandra et al. (2007).

Theorem 3.13. *Approval is immune to constructive control by partition of candidates and run-off partition of candidates under the* TP *tie-handling rule in the nonunique-winner model.*

Proof. We first show that a winner (possibly tied) of the two-stage approval election induced by partition of candidates under the TP tie-handling rule must be a winner without any control action (i.e., in the input's election). Fix an election (C, V) and let $p \in C$ be a candidate such that $(C, V, p) \in$ Approval-CC-PC-TP-NUW. Thus it holds that p's score is at least as high as the score of the other candidates present in the final round. Since no candidate is eliminated by the tie-handling rule, each candidate that is eliminated in the first round has score strictly less than some candidate that is present in the final round. It follows that p's score is at least as high as the score of every other candidate in C. Thus p is a winner of the approval election (C, V).

Essentially the same argument line can be used to show that for each $(C, V, p) \in$ Approval-CC-RPC-TP-NUW, it holds that p is a winner of approval election (C, V). ❑

Theorem 3.14. Approval-CC-PC-TP-NUW = Approval-CC-RPC-TP-NUW.

Proof. We will show that Approval-CC-PC-TP-NUW and Approval-CC-RPC-TP-NUW are comprised of exactly those inputs where the focus candidate is already a nonunique winner, following the approach in the proofs of Theorems 3.6 and 3.9.

Consider the following two disjoint sets: $A = \{(C, V, p) \mid p \in C$ and p is a winner of approval election $(C, V)\}$ and $B = \{(C, V, p) \mid p \in C$ and p is not a winner of approval election $(C, V)\}$. It's clear that Approval-CC-PC-TP-NUW $\subseteq (A \cup B)$ and Approval-CC-RPC-TP-NUW $\subseteq (A \cup B)$. By Theorem 3.13, it holds that approval is immune to constructive control by partition of candidates and run-off partition of candidates under the TP tie-handling rule in the nonunique-winner model. Thus Approval-CC-PC-TP-NUW$\cap B = \emptyset$ and Approval-CC-RPC-TP-NUW $\cap B = \emptyset$, and it holds that Approval-CC-PC-TP-NUW $\subseteq A$ and Approval-CC-RPC-TP-NUW $\subseteq A$. Fix $(C, V, p) \in A$. Then the partition $(\emptyset, C)$ witnesses that both $(C, V, p) \in$ Approval-CC-PC-TP-NUW and $(C, V, p) \in$ Approval-CC-RPC-TP-NUW as no candidate will have score higher than p's score (because p is a winner of approval election (C, V)), and since the tie-handling rule does not eliminate candidates, p will always proceed from the subelection it participates in to the final round. Thus no candidate can defeat p in the final round, making p a winner of the two-stage election. ❑

Corollary 3.15. *For each* $\mathcal{T} \in \{$CC-PC-TE-NUW*,* CC-RPC-TE-NUW*,* CC-PV-TP-NUW$\}$*, it holds that* Approval-CC-PC-TP-NUW $\subsetneq$ Approval-$\mathcal{T}$.

Proof. Fix $\mathcal{T} \in \{$CC-PC-TE-NUW, CC-RPC-TE-NUW, CC-PV-TP-NUW$\}$ and fix $(C, V, p) \in$ Approval-CC-PC-TP-NUW. By the proof of Theorem 3.14, we know that Approval-CC-PC-TP-NUW is comprised of only those inputs where p is already a nonunique winner of the approval election (C, V). In the first case, where $\mathcal{T}$ is about partitioning candidates, then the partition $(\{p\}, C - \{p\})$ is a witness that $(C, V, p) \in$ Approval-$\mathcal{T}$ as p will uniquely win the subelection it participates in and proceed to the final round. Regardless of which candidates proceed to the final round from the other subelection, they will not have score greater than p's (or they would have defeated p in the (C, V) election) and so p is a winner of the two-stage election. In the second case, where $\mathcal{T}$ is about partitioning votes (i.e., $\mathcal{T} =$ CC-PV-TP-NUW), then the partition $(\emptyset, V)$ suffices as everyone proceeds to the final round (because all the candidates tie in the subelection $(C, \emptyset)$ and the tie-handling rule does not eliminate candidates) and since p is already a winner of the approval election (C, V), they are a winner of the final round. The corresponding separation witness can be found in Table 9. ❑

Finally, we provide direct arguments that yield additional collapses and containments.

Theorem 3.16. Approval-DC-PV-TE-UW = Approval-DC-PV-TE-NUW.

Proof. The $\supseteq$ relationship is immediate.

$\subseteq$: Let $(C, V, p) \in$ Approval-DC-PV-TE-UW. Note that this implies $\|C\| \geq 2$, since in one-candidate approval elections the one candidate always wins. Let (V_1, V_2) be a vote partition that witnesses $(C, V, p) \in$ Approval-DC-PV-TE-UW. As the tie-handling rule is the TE rule, it must hold (since if p uniquely won both subelections it necessarily would uniquely win the second-round election) that either p is eliminated in both subelections (either by tieing or by losing) or p uniquely wins one subelection, some other candidate d uniquely wins the other subelection, and p does not uniquely win the final round (thus d's score must be at least as large as p's score when using vote set V). In the first case, the partition (V_1, V_2) witnesses $(C, V, p) \in$ Approval-DC-PV-TE-NUW since p will not proceed to the final round. In the second case, using the partition $(V, \emptyset)$ suffices for the following

reasons. In the subelection (C, V), since d's score is at least as large as that of p, either d uniquely wins the subelection or both p and d tie and are eliminated by the tie-handling rule. In the subelection $(C, \emptyset)$ every candidate ties and so all candidates are eliminated by the tie-handling rule (since $\|C\| \geq 2$). Thus p is eliminated in both subelections, does not proceed to the final round, and so is not a final-round winner. ❑

Theorem 3.17. Approval-CC-PC-TE-NUW = Approval-CC-RPC-TE-NUW.

Proof. We structure this proof so as to yield not just this theorem but also the related result that we state as Corollary 3.18.

$\subseteq$: Let $(C, V, p) \in$ Approval-CC-PC-TE-NUW and let (C_1, C_2) be a candidate partition that witnesses this membership. Consider the case where p uniquely wins the final round. If $p \in C_1$, then p uniquely wins (C_1, V) and in the final round defeats all candidates in C_2. If $p \in C_2$, then in the final round p defeats the candidate (if any) that survives the TE tie-handling rule regarding the subelection (C_1, V) as well as all the candidates in $C_2 - \{p\}$. Regardless of which case holds, the partition (C_1, C_2) will witness $(C, V, p) \in$ Approval-CC-RPC-TE-NUW since p will uniquely win its subelection and then will defeat any candidate that moves forward from the other subelection. If p does not uniquely win the final round, then there is at least one other candidate that ties with p in the final round. If $p \in C_1$, then p must uniquely win in (C_1, V) and as no candidate in C_2 can have a score greater than p's (since (C_1, C_2) witnesses $(C, V, p) \in$ Approval-CC-PC-TE-NUW), the partition (C_1, C_2) suffices to witness $(C, V, p) \in$ Approval-CC-RPC-TE-NUW. If $p \in C_2$, then under partition (C_1, C_2) p could first-round tie with a candidate and be eliminated (under run-off partition of candidates due to the TE rule). However, let T denote the (possibly empty) set of candidates (other than p) that tie with p in (C_2, V). Then the partition $(C_1 \cup T, C_2 - T)$ witnesses $(C, V, p) \in$ Approval-CC-RPC-TE-NUW, since p will uniquely win $(C_2 - T, V)$ and will either tie or defeat the winner (if any) of $(C_1 \cup T, V)$.

$\supseteq$: Let $(C, V, p) \in$ Approval-CC-RPC-TE-NUW and let (C_1, C_2) be a candidate partition that witnesses this membership. Without loss of generality, assume that $p \in C_1$. Thus it holds that p uniquely wins (C_1, V). If p uniquely wins the final round, then p also defeats the candidate (if any) that moves forward from (C_2, V). Thus the partition (C_2, C_1) will also witness $(C, V, p) \in$ Approval-CC-PC-TE-NUW (since p has strictly more approvals than any candidate other than itself). If p does not uniquely win the final round, then there is another candidate d, who is the unique winner of (C_2, V) and ties with p in the final round. Again the partition (C_2, C_1) suffices to witness $(C, V, p) \in$ Approval-CC-PC-TE-NUW since d proceeds to the final round and both p and d win there due to their numbers of approvals. ❑

Corollary (to the Proof) 3.18. Approval-CC-PC-TE-UW = Approval-CC-RPC-TE-UW.

Proof. This is an immediate corollary to the above proof of Theorem 3.17, as the proof was intentionally structured to ensure that if the witness of one type made p a unique winner of the final round, then the constructed-above (sometimes different) partition for the other type also made p a unique winner of the final round under that other type of control. ❑

We can additionally show the following three containments. Theorem 3.20 and Corollary 3.21 are surprising, since they are about partitioning different components of elections.

Theorem 3.19. Approval-DC-PV-TP-UW $\subsetneq$ Approval-DC-PV-TE-NUW.

Proof. The proof is similar in flavor to that of Theorem 3.16. Let $(C, V, p) \in$ Approval-DC-PV-TP-UW and let (V_1, V_2) be a vote partition that witnesses this membership. There are two cases to consider. Either p uniquely wins in exactly one subelection (it certainly cannot uniquely win in both, else p would be the unique final-round winner), or p is not a unique winner in either subelection. In the latter case, the partition (V_1, V_2) witnesses that $(C, V, p) \in$ Approval-DC-PV-TE-NUW as p never survives the TE tie-handling rule (because it is not a unique winner in either subelection). In the former case, the partition $(V, \emptyset)$ witnesses that $(C, V, p) \in$ Approval-DC-PV-TE-NUW. This is so because, in that case, there must exist a candidate $d \neq p$ that in the Approval-DC-PV-TP-UW setting under partition (V_1, V_2) proceeds to the final stage and in the second round has a score at least as high as that of p. So in the Approval-DC-PV-TE-NUW setting's first round under the partition $(V, \emptyset)$, in the subelection (C, V) candidate p can at best tie d, and thus certainly cannot move forward under the TE tie-handling rule. Additionally, no one moves forward from subelection $(C, \emptyset)$ (everyone ties and, since in the current case we know there are at least two candidates, everyone is eliminated from that subelection). This shows the containment. The separation witness for the strict containment can be found in Table 9. ❑

Theorem 3.20. Approval-DC-RPC-TE-NUW $\subsetneq$ Approval-DC-PV-TP-UW.

Proof. Let $(C, V, p) \in$ Approval-DC-RPC-TE-NUW and let (C_1, C_2) be a candidate partition that witnesses this membership. Then, with respect to that partition, p is eliminated either in its subelection (either by tieing or by losing) or in the final round. In both cases, this must happen because there is another candidate d such that p's score is at most d's score (using vote set V). By using the partition $(\emptyset, V)$, we know that every candidate will proceed to the final round (everyone ties in $(C, \emptyset)$ and proceeds to the final round) and in that final round, either d will tie p or d will defeat p. In either case, p is not a unique winner. This shows the containment. The separation witness for the strict containment can be found in Table 9. ❑

As an immediate corollary to Theorems 3.19 and 3.20, we have the following result.

Corollary 3.21. Approval-DC-RPC-TE-NUW $\subsetneq$ Approval-DC-PV-TE-NUW.

4. Related Work

Electoral control attacks were introduced and studied by Bartholdi, Tovey, and Trick in a tremendously influential 1992 paper (Bartholdi et al., 1992). They defined a set of attacks based on adding/deleting/partitioning candidates and voters, and those attacks have been studied ever since.

Bartholdi et al. (1992) studied only the goal of making a particular focus candidate be an untied winner of the election (i.e., constructive control in the unique-winner model).

Hemaspaandra et al. (2007) introduced "destructive control" versions of each constructive control type: The goal is to prevent (via the given control action) the focus candidate from becoming a unique winner. Later papers often additionally, or only, used the nonunique-winner model, in which the goal is to make the focus candidate be (or, for the destructive case, not be) a winner. As far as we know, the first example of this was a study of Llull and Copeland elections by Faliszewski et al. (2007, 2009) that studied both winner models, and Hemaspaandra, Lavaee, and Menton (2016) (see also Hemaspaandra et al., 2020, Footnote 5) argue that the nonunique-winner model is a better model to study than the unique-winner model.

Hemaspaandra et al. (2007) addressed the ways of handling ties in the first-round elections of the (two-round) "partition" control types of Bartholdi et al. (1992), and framed the two now-standard approaches called "ties promote" and "ties eliminate"; they did this because although tie-handling had previously been suggested as being unimportant, their paper establishes that, for example, for plurality the choice between those two rules spells the difference between being NP-complete and belonging to P. The "adding candidates" control attack of Bartholdi et al. (1992) was anomalously defined, and Hemaspaandra, Hemaspaandra, and Rothe (2009) kept the original notion but renamed it "unlimited adding of candidates," and introduced, under the thus-available name "adding candidates," the version that is analogous to the other Bartholdi et al. (1992) add/delete types, and the subsequent papers have followed that notational shift.

Altogether, the Bartholdi et al. (1992) control attack set, under the above clarifications and enrichments, yields a set of eleven constructive control attacks and eleven destructive control attacks. As mentioned earlier, this in some sense forms a "standard" benchmark set of attacks (though some papers use subsets of that collection, and other papers have taken control in additional directions, e.g., resolute control Yang & Wang, 2017; Gupta, Roy, Saurabh, & Zehavi, 2022). For example, the excellent survey chapter on control and bribery by Faliszewski and Rothe (2016) uses precisely those 22 control attacks, as does the recent paper on search versus decision of Hemaspaandra et al. (2020). Since the field has not yet resolved whether nonunique-winner or unique-winner is the right standard—indeed, the two just-cited sources make different choices—and because discovering cases when a control type in one of those models turns out to be identical to a control type in the other is itself interesting, in this paper we covered both models, and thus $2 \times (11 + 11) = 44$ control types.

Of the $\binom{44}{2} = 946$ pairs (322 of them compatible) of those control types, seven were proven to collapse in Hemaspaandra et al. (2020). That is the paper that most strongly inspired the present paper. It also separates one pair: DC-RPC-TP-UW and DC-PC-TP-UW. However, it does so using an election system that is not (candidate-)neutral. In our present paper, regarding that separation and all separations, we obtain them directly in systems that are (candidate-)neutral or by inheritance from systems that are (candidate-)neutral.

The additional, different control types known as resolute control (Yang & Wang, 2017; Gupta et al., 2022) are quite interesting. Resolute control asks whether there is some action (from a certain range of actions) that keeps every one of a certain collection of candidates from being a winner. This might seem to be the same as our function model for the case of (nonunique-winner) destructive control, but it is not. In our function model for nonunique-winner destructive control, we are speaking of the collection of candidates who

can individually be prevented from winning by some control action. But even if two or more candidates belong to our function's output, they could be put into that output by different control actions, and there might be no single action that blocks both simultaneously. In brief, resolute control is focused on blocking whole groups, and our function model in contrast is a refinement of set separations and focuses on individual candidates to let us identify new containment patterns between control types.

Collapsing or separating control types is not *directly* about complexity. However, doing so is highly relevant to complexity, as the types were defined as natural benchmarks whose complexity could be studied. In fact, there has been something of a race to find natural systems in which very many control attacks are NP-hard to carry out.[13] Among the many systems that have done well or very well in that race are, along with some of the key papers that analyzed the complexity of each, Schulze elections (Parkes & Xia, 2012; Menton & Singh, 2013; Hemaspaandra et al., 2016), ranked-pair elections (Parkes & Xia, 2012; Hemaspaandra et al., 2016), SP-AV elections (Erdélyi, Nowak, & Rothe, 2009), normalized range voting (Menton, 2013), fallback elections (Erdélyi & Rothe, 2010; Erdélyi, Piras, & Rothe, 2011, see also Erdélyi, Fellows, Rothe, & Schend, 2015), and Bucklin elections (Erdélyi et al., 2011, see also Erdélyi et al., 2015). Faliszewski and Rothe (2016, Table 7.3) provide a lovely table, for the 22 unique-winner control cases, of what is known for twelve voting systems. Regarding the three important systems we spotlight in the present paper, plurality's control was explored by Bartholdi et al. (1992) and Hemaspaandra et al. (2007), veto has been studied by Lin (2012) and Maushagen and Rothe (2018) (see also Erdélyi, Reger, & Yang, 2019, Table 1), and approval has been studied by Hemaspaandra et al. (2007) (see also Baumeister, Erdélyi, Hemaspaandra, Hemaspaandra, & Rothe, 2010).

5. Comments on Checking and Reproducibility

As mentioned earlier, a number of our election examples showing separations were found with the aid of computer searches. Other examples were human generated.

In addition to the verification code within the computer-search programs, we coded—with a different person doing the coding—stand-alone verification routines, as a double-

13. Regarding the "race" to find systems that are resistant to a large number of control types, one might ask whether it still is fair to count it as multiple strengths if the elements of collapsing pairs (or larger equivalence classes) are resistant (and so NP-hard), or to count it as multiple weaknesses if the elements of collapsing pairs (or larger equivalence classes) are vulnerable (and thus in P). In some sense, one could argue that this is unfair, as it is putting extra weight on the "same" type.

However, one could also argue that having a collapsing control type foursome count as four is fair and natural, since the actual mechanisms of the four are different, and so one either is fighting off attacks through NP-completeness, or is vulnerable to multiple attack lines.

In any case, if one focuses solely on number of resistances, one is implicitly saying that all control types are equally important, and we do not think that saying that is correct. Rather, what is most important is that the field knows, for the full palette of control types, which ones are resistant and which ones are vulnerable with respect to a given election system. That will allow the people choosing which election system to use in a given setting to choose an election system that is resistant to as many as possible of the attacks that they expect are the most likely within the setting. And for that, there is no doubt that avoiding duplicate work through exploiting collapses is helpful.

Finally, we mention that merging collapsed types on each system's "score card" would be problematic, since different elections have different collapses, and so different election systems' score cards would not even have the same number of entries.

Table 1: Summary of separations and collapses. Blue indicates results due to or inherited from (Hemaspaandra et al., 2020). Red indicates results due to the present paper. The general-case line shows when collapses occur for all election systems over linear orders.

	Set Classification			Subclassification of Separations		
Election System	Separations	Collapses	Open	"$\subsetneq$"/"$\supsetneq$"	Incomparable	Open
General Case	$1 + 314 = 315^{\dagger}$	7	0	38	277	0
Plurality	315	7	0	38	277	0
Veto	314	$7 + 1 = 8$	0	58	256	0
Approval Voting	301	$7 + 14 = 21$	0	88	213	0

†Or $0 + 315$ for the pure social choice approach to candidate names (see Footnote 2 and the discussion in the Related Work section).

check of the correctness of all separation examples used in this paper. Whether computer-generated or human-generated, every example was checked via those separate verification programs. Also, every computer-generated example was verified by a human.

To support reproducibility, and to aid researchers who might wish to carry our study to other cases, we have made the computer-search programs, and their inputs and outputs, publicly available, in effect as an online appendix of this paper, in an online repository. That repository is available at `https://github.com/MikeChav/SCT_Code`.

Our search programs use randomization, and the repository captures and documents the randomization used in each run generating our examples. Thus skeptical researchers could, if they wanted, simulate our code, using the same randomization, to assure themselves that our codes indeed produced the examples we claim they did. The repository also includes some information about what systems the programs were run on and how long certain runs took. (In fact, each program was run once, and the run itself randomly tried many examples until a counterexample was found; even the longest-running program ran for at most 21 CPU seconds.)

The repository additionally includes the separate verification programs mentioned above.

6. Conclusions and Open Problems

Table 1 summarizes our results. We established that in the general (universally quantified) case there are no collapsing pairs (by which we always mean among the standard 44 control types) other than the seven collapsing pairs identified by Hemaspaandra et al. (2020), and that plurality has no collapsing pairs beyond those seven. For veto and approval voting we discovered additional collapsing pairs beyond those inherited seven, but we also established that veto and approval voting, after our work, have no remaining undiscovered collapsing pairs.

Our work helps clarify the landscape of which control pairs do or do not collapse, both in the general (universally quantified) case, and for plurality, veto, and approval voting.

We also refined all our separations, and in doing so uncovered containment relationships—including many that do not follow from the relationship between the nonunique-winner model and the unique-winner model.

A number of interesting open directions are suggested by our work. One is, for important concrete election systems other than plurality, veto, and approval voting, to com-

pletely classify the collapses and separations that hold for those specific systems. Another direction—building on the results of Section 3.3—is to find sufficient conditions (or, better still, necessary-and-sufficient conditions) for many control-pair collapses in terms of axiomatic properties of election systems. Though our goal for separations was to subclassify each separation into exactly one of the three cases "$\subsetneq$", "$\supsetneq$", or incomparability, in our tables we have also noted those cases where our constructions establish strong incomparability; one could for the cases where we list incomparability try to establish strong incomparability. An additional open direction is to see whether already-studied control types beyond the 44 investigated here collapse with each other or with some of the 44, either in general or for important concrete election systems.

Finally, control types are defined as sets. When a pair of control types collapses, those sets are equal and thus certainly are of the same complexity. However, it would be interesting to see whether for collapsing control types, their complexities as *search* problems are or are not polynomially related. That issue, inspired by the work of Hemaspaandra et al. (2020) and an earlier version of the present paper, has recently been studied by Carleton, Chavrimootoo, Hemaspaandra, Narváez, Taliancich, and Welles (2023a, 2022, 2024).

Acknowledgments

We thank the NSF and the CRA for support under grants CCF-2006496, DUE-2135431, and CIF2020-UR-36. This work was done in part while authors Carleton, Chavrimootoo, Narváez, and Taliancich were at the University of Rochester's Department of Computer Science. A preliminary version of this paper appeared in the *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems* (Carleton, Chavrimootoo, Hemaspaandra, Narváez, Taliancich, & Welles, 2023b). We thank the anonymous referees and Ulle Endriss for their valuable comments, suggestions, and guidance.

Appendix A. Compatible Control Types

Table 2 shows how the 44 standard control types partition into five compatibility equivalence classes. Every other (distinct) pair among the 44 is not a compatible pair.

Table 2: The 44 types of control and a description of which components are part of the input for each one. The input type thus partitions the control types into five equivalence classes as to compatibility of inputs.

Control Type	Candidates	Votes	Focus Candidate	Spoiler Candidates	Spoiler Votes	Limit (Natural Number)
CC-PV-TE-UW, CC-PV-TE-NUW, CC-PV-TP-UW, CC-PV-TP-NUW, CC-PC-TE-UW, CC-PC-TE-NUW, CC-PC-TP-UW, CC-PC-TP-NUW, CC-RPC-TE-UW, CC-RPC-TE-NUW, CC-RPC-TP-UW, CC-RPC-TP-NUW, DC-PV-TE-UW, DC-PV-TE-NUW, DC-PV-TP-UW, DC-PV-TP-NUW, DC-PC-TE-UW, DC-PC-TE-NUW, DC-PC-TP-UW, DC-PC-TP-NUW,	Yes	Yes	Yes	No	No	No

Control Type	Candidates	Votes	Focus Candidate	Spoiler Candidates	Spoiler Votes	Limit (Natural Number)
DC-RPC-TE-UW, DC-RPC-TE-NUW, DC-RPC-TP-UW, DC-RPC-TP-NUW						
CC-AC-UW, CC-AC-NUW, DC-AC-UW, DC-AC-NUW	Yes	Yes	Yes	Yes	No	Yes
CC-DC-UW, CC-DC-NUW, CC-DV-UW, CC-DV-NUW, DC-DC-UW, DC-DC-NUW, DC-DV-UW, DC-DV-NUW	Yes	Yes	Yes	No	No	Yes
CC-AV-UW, CC-AV-NUW, DC-AV-UW, DC-AV-NUW	Yes	Yes	Yes	No	Yes	Yes
CC-UAC-UW, CC-UAC-NUW, DC-UAC-UW, DC-UAC-NUW	Yes	Yes	Yes	Yes	No	No

Appendix B. Tables

Tables 4, 6, and 8 list each of the elections (identified by a candidate set C and a vote set V) used to show separations and incomparability. Each line of these tables also has some columns that may have no entries (denoted by "-"). These columns are S (to denote additional candidates when the control type is about adding candidates), U (to denote additional votes when the control type is about adding votes), and k (to denote the limit imposed on a particular control, e.g., limited adding of candidates, or deleting candidates/voters).

We assign an ID to each election and use those IDs from Tables 4, 6, and 8 in Tables 5, 7, and 9 to identify which tool is a separation witness. The obvious containments (that are about UW and NUW variants of the same control type, see location (**) in Section 1) are implicit in this table and thus we do not explicitly mark when they are used.

Additionally, we color code Tables 5, 7, and 9 in a specific manner: The entries that have the same (nonwhite) background color form an equivalence class. Each entry that is not colored (i.e., that has a white background) is an equivalence class of size one, and we will not color such entries. Table 3 provides a summary of the equivalence classes and their corresponding colors.

Each boldfaced entry indicates the canonical element of its equivalence class. Those are also indicated in Tables 5, 7, and 9. Having canonical elements helped make the proof process more economical. If we separate $\mathcal{T}$ from $\mathcal{T}'$, we implicitly have separated $\mathcal{T}$ from all members of the equivalence class of $\mathcal{T}'$. In the proof process we focused on separations regarding only two canonical elements (where we view each singleton type as a stand-alone canonical element of its size-one equivalence class, though we don't boldface those).

Table 3: Equivalence classes and their respective colors. Each boldfaced entry indicates the canonical element of its equivalence class.

Class	Color
Plurality-DC-RPC-TE-NUW, Plurality-DC-RPC-TE-UW, Plurality-DC-PC-TE-UW, Plurality-DC-PC-TE-NUW	
Plurality-DC-RPC-TP-NUW, Plurality-DC-PC-TP-NUW	
Veto-DC-RPC-TE-NUW, Veto-DC-RPC-TE-UW, Veto-DC-PC-TE-UW, Veto-DC-PC-TE-NUW	

Class	Color
Veto-DC-RPC-TP-NUW, Veto-DC-PC-TP-NUW	
Veto-DC-PV-TE-NUW, Veto-DC-PV-TE-UW	
Approval-DC-PV-TE-NUW, Approval-DC-PV-TE-UW	
Approval-DC-RPC-TE-NUW, Approval-DC-RPC-TE-UW, Approval-DC-PC-TE-UW, Approval-DC-PC-TE-NUW, Approval-DC-RPC-TP-UW, Approval-DC-PC-TP-UW	
Approval-DC-RPC-TP-NUW, Approval-DC-PC-TP-NUW	
Approval-CC-RPC-TP-UW, Approval-CC-PC-TP-UW	
Approval-CC-RPC-TP-NUW, Approval-CC-PC-TP-NUW	
Approval-CC-RPC-TE-UW, Approval-CC-PC-TE-UW	
Approval-CC-RPC-TE-NUW, Approval-CC-PC-TE-NUW	

Table 4: List of separation witnesses in plurality. We note the computer-generated entries with a "†" superscript.

ID	C	S	V	U	k
Plur.1	$\{a,b,c\}$	-	$\{a>b>c, b>a>c, c>a>b\}$	-	-
Plur.2	$\{a,b\}$	-	$\{a>b, b>a\}$	-	-
Plur.3	$\{a,b\}$	-	$\{a>b\}$	-	-
Plur.4†	$\{a,b,c,d\}$	-	$\{b>c>d>a, d>a>c>b, b>c>d>a, a>c>b>d, a>b>d>c, d>a>b>c, c>d>b>a, d>a>c>b, a>c>b>d, d>c>b>a, b>c>d>a, a>b>d>c, d>b>c>a, a>d>c>b, b>c>d>a, c>a>b>d, b>a>d>c, a>c>d>b\}$	-	-
Plur.5	$\{a,b,c\}$	-	$\{a>b>c, a>b>c, a>c>b, a>c>b, b>a>c, b>a>c, b>a>c, c>a>b, c>a>b, c>a>b\}$	-	-
Plur.6	$\{a,b,c\}$	-	$\{b>c>a, c>b>a\}$	-	-
Plur.7	$\{a,b,c,d\}$	-	$\{a>b>c>d, c>d>a>b, d>b>a>c\}$	-	-
Plur.8	$\{a,b,c\}$	-	$\{a>b>c, a>b>c, b>a>c, b>a>c, c>b>a\}$	-	-
Plur.9	$\{a,b,c,d\}$	-	$\{a>b>c>d, a>b>c>d, a>b>c>d, b>a>c>d, c>b>a>d, d>b>a>c\}$	-	-
Plur.10	$\{a,b,c\}$	-	$\{a>b>c, a>b>c, b>a>c, c>a>b\}$	-	-
Plur.11	$\{a,b,c,d,e\}$	-	$\{c>b>a>d>e, c>d>e>a>b, a>d>b>c>e, c>d>b>e>a, c>b>e>d>a, d>e>b>c>a, d>b>e>c>a, a>b>d>e>c, e>c>b>d>a, c>a>b>d>e, b>e>a>c>d, a>d>b>e>c, d>a>c>e>b, a>b>c>e>d, c>d>e>b>a, e>d>c>a>b, e>d>a>b>c\}$	-	-
Plur.12	$\{a,b,c\}$	-	$\{a>b>c, a>c>b, b>a>c, c>a>b\}$	-	-
Plur.13	$\{a,b,c\}$	-	$\{a>b>c, a>c>b, b>c>a, c>b>a\}$	-	-
Plur.14	$\{a,b,c\}$	-	$\{a>b>c, a>c>b, b>c>a, b>c>a, b>c>a, c>b>a, c>b>a, c>b>a\}$	-	-
Plur.15	$\{a,b,c\}$	-	$\{a>b>c, a>c>b, b>a>c, b>c>a, c>b>a\}$	-	-
Plur.16	$\{a,b,c\}$	-	$\{a>b>c, a>c>b, b>a>c, b>c>a, c>a>b\}$	-	-
Plur.17	$\{a,b,c\}$	-	$\{a>b>c, a>c>b, b>a>c, b>c>a, c>a>b, c>b>a\}$	-	-
Plur.18	$\{a,b,c\}$	-	$\{a>b>c, b>c>a, b>c>a, c>b>a, c>b>a, c>b>a\}$	-	-
Plur.19	$\{a,b,c,d\}$	-	$\{a>b>c>d, a>b>c>d, a>b>c>d, b>a>c>d, c>b>a>d, d>b>a>c\}$	-	-
Plur.20	$\{a,b,c,d\}$	-	$\{a>c>b>d, b>a>c>d, b>a>c>d, c>b>a>d, d>c>b>a\}$	-	-
Plur.21	$\{a,b,c,d\}$	-	$\{a>b>c>d, a>b>c>d, b>c>a>d, b>c>a>d, c>d>b>a\}$	-	-
Plur.22	$\{a,b,c,d\}$	-	$\{a>b>c>d, a>b>c>d, a>b>c>d, b>c>d>a, c>b>d>a, d>a>c>b, d>b>c>a, d>b>c>a\}$	-	-
Plur.23	$\{a,b,c\}$	-	$\{b>a>c, b>a>c, b>a>c, b>a>c, b>a>c, a>b>c, a>b>c, a>b>c, a>b>c, a>b>c, c>b>a, c>b>a, c>b>a\}$	-	-
Plur.24	$\{a,b,c,d\}$	-	$\{b>a>d>c, b>a>d>c, b>a>d>c, b>a>d>c, b>a>d>c, c>a>b>d, c>a>b>d, c>a>b>d, c>a>b>d, d>a>b>c, d>a>b>c\}$	-	-
Plur.25†	$\{a,b,c,d,e,f\}$	-	$\{d>e>b>f>c>a, b>f>c>a>e>d, b>e>c>a>d>f, f>e>a>b>d>c, b>a>e>d>f>c, a>c>d>e>b>f, c>e>f>b>a>d\}$	-	-
Plur.26†	$\{a,b,c,d,e,f,g\}$	-	$\{c>d>g>f>b>e>a, a>f>b>c>d>g>e, g>c>a>d>e>b>f, a>g>f>d>e>b>c, e>g>a>d>b>c>f, d>f>e>a>g>c>b, f>a>d>g>e>c>b, b>g>a>c>f>d>e, a>c>g>b>f>d>e\}$	-	-

ID	C	S	V	U	k
Plur.27	$\{a,b\}$	$\{c\}$	$\{c>b>a\}$	-	1
Plur.28	$\{a,b\}$	-	$\{a>b\}$	$\{a>b\}$	1
Plur.29	$\{a,b\}$	$\{c\}$	$\{a>c>b\}$	-	1
Plur.30	$\{a,b\}$	-	$\{a>b, a>b\}$	-	0
Plur.31	$\{a,b,c,d\}$	-	$\{b>c>d>a, b>c>d>a, a>b>c>d\}$	-	2
Plur.32	$\{a,b,c,d\}$	-	$\{a>b>c>d, a>b>c>d, a>b>c>d, a>b>c>d, a>b>c>d, a>b>c>d, b>c>d>a, b>c>d>a, b>c>d>a, c>b>d>a, c>b>d>a, c>b>d>a, d>b>c>a, d>b>c>a, d>b>c>a\}$	-	2
Plur.33	$\{a\}$	$\emptyset$	$\{a\}$	-	0
Plur.34	$\{a\}$	-	$\{a\}$	-	0
Plur.35	$\{a\}$	-	$\{a\}$	$\emptyset$	0
Plur.36	$\{a\}$	$\emptyset$	$\{a\}$	-	-
Plur.37	$\{a,b\}$	$\emptyset$	$\{a>b, b>a\}$	-	0
Plur.38	$\{a,b\}$	-	$\{a>b, b>a\}$	-	0
Plur.39	$\{a,b\}$	-	$\{a>b, b>a\}$	$\emptyset$	0
Plur.40	$\{a,b\}$	$\emptyset$	$\{a>b, b>a\}$	-	-
Plur.41	$\{a,b\}$	-	$\{b>a, b>a\}$	-	1
Plur.42	$\{a,b\}$	-	$\{a>b, a>b, b>a\}$	-	1
Plur.43	$\{a,b\}$	-	$\{a>b, b>a\}$	-	1
Plur.44	$\{a,b,c\}$	-	$\{a>b>c, b>c>a, b>c>a, c>b>a, c>b>a\}$	-	-
Plur.45	$\{a,b,c\}$	-	$\{a>b>c, b>c>a, c>b>a\}$	-	-
Plur.46	$\{a,b,c\}$	-	$\{a>b>c, b>c>a, b>c>a, b>c>a, c>b>a, c>b>a, c>b>a\}$	-	-
Plur.47†	$\{a,b,c,d\}$	-	$\{c>b>a>d, d>c>a>b, b>a>d>c, c>b>d>a, a>b>c>d, d>b>c>a, a>d>b>c\}$	-	-
Plur.48†	$\{a,b,c,d,e\}$	-	$\{a>c>b>d>e, d>c>b>a>e, c>d>b>e>a, e>d>b>a>c, a>d>b>e>c, b>e>d>a>c, a>d>e>b>c, e>d>a>b>c, c>a>e>d>b, b>e>d>a>c, d>c>e>b>a\}$	-	-
Plur.49†	$\{a,b,c,d\}$	-	$\{c>a>b>d, b>a>c>d, c>b>a>d, b>d>c>a, d>a>b>c, c>b>d>a, a>d>b>c, a>b>d>c, c>d>a>b\}$	-	-
Plur.50†	$\{a,b,c,d,e\}$	-	$\{a>d>e>b>c, e>c>b>a>d, c>b>a>e>d, e>a>d>b>c, b>d>a>e>c, e>a>b>d>c, b>c>e>a>d, d>c>b>a>e, d>c>b>a>e\}$	-	-

Table 5: Table of separations and collapses (here denoted by EQ) in plurality voting. The Classification column partitions each separation into one of the three cases, $\subsetneq$, $\supsetneq$, and INCOMP. (For cases of INCOMP for which we happen to have established that the separation holds with strong incomparability, we have noted that with a "$*$" superscript.)

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-PV-TE-UW	CC-PV-TE-NUW	$\subsetneq$	Plur.2
CC-PV-TE-UW	CC-PV-TP-UW	INCOMP	Plur.1 + Plur.50
CC-PV-TE-UW	CC-PV-TP-NUW	INCOMP	Plur.2 + Plur.46
CC-PV-TE-UW	CC-RPC-TE-UW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-UW	CC-RPC-TE-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-UW	CC-RPC-TP-UW	INCOMP	Plur.1 + Plur.24
CC-PV-TE-UW	CC-RPC-TP-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-UW	CC-PC-TE-UW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-UW	CC-PC-TE-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-UW	CC-PC-TP-UW	INCOMP	Plur.1 + Plur.24
CC-PV-TE-UW	CC-PC-TP-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-UW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-PV-TE-UW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-PV-TE-UW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-PV-TE-UW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-PV-TE-UW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-PV-TE-UW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-PV-TE-UW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-PV-TE-UW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-PV-TE-UW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-PV-TE-UW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-PV-TE-UW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-PV-TE-UW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-PV-TE-NUW	CC-PV-TP-UW	INCOMP	Plur.2 + Plur.50

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-PV-TE-NUW	CC-PV-TP-NUW	INCOMP	Plur.14 + Plur.50
CC-PV-TE-NUW	CC-RPC-TE-UW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-NUW	CC-RPC-TE-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-NUW	CC-RPC-TP-UW	INCOMP	Plur.2 + Plur.24
CC-PV-TE-NUW	CC-RPC-TP-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-NUW	CC-PC-TE-UW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-NUW	CC-PC-TE-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-NUW	CC-PC-TP-UW	INCOMP	Plur.2 + Plur.24
CC-PV-TE-NUW	CC-PC-TP-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TE-NUW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-PV-TE-NUW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-PV-TE-NUW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-PV-TE-NUW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-PV-TE-NUW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-PV-TE-NUW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-PV-TE-NUW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-PV-TE-NUW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-PV-TE-NUW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-PV-TE-NUW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-PV-TE-NUW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-PV-TE-NUW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-PV-TP-UW	CC-PV-TP-NUW	$\subsetneq$	Plur.21
CC-PV-TP-UW	CC-RPC-TE-UW	INCOMP	Plur.20 + Plur.23
CC-PV-TP-UW	CC-RPC-TE-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TP-UW	CC-RPC-TP-UW	INCOMP	Plur.1 + Plur.23
CC-PV-TP-UW	CC-RPC-TP-NUW	INCOMP	Plur.2 + Plur.23
CC-PV-TP-UW	CC-PC-TE-UW	INCOMP	Plur.18 + Plur.23
CC-PV-TP-UW	CC-PC-TE-NUW	INCOMP	Plur.2 + Plur.23
CC-PV-TP-UW	CC-PC-TP-UW	INCOMP	Plur.1 + Plur.23
CC-PV-TP-UW	CC-PC-TP-NUW	INCOMP	Plur.2 + Plur.23
CC-PV-TP-UW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-PV-TP-UW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-PV-TP-UW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-PV-TP-UW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-PV-TP-UW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-PV-TP-UW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-PV-TP-UW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-PV-TP-UW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-PV-TP-UW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-PV-TP-UW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-PV-TP-UW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-PV-TP-UW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-PV-TP-NUW	CC-RPC-TE-UW	INCOMP	Plur.2 + Plur.24
CC-PV-TP-NUW	CC-RPC-TE-NUW	INCOMP	Plur.18 + Plur.23
CC-PV-TP-NUW	CC-RPC-TP-UW	INCOMP	Plur.2 + Plur.24
CC-PV-TP-NUW	CC-RPC-TP-NUW	INCOMP	Plur.15 + Plur.24
CC-PV-TP-NUW	CC-PC-TE-UW	INCOMP	Plur.2 + Plur.24
CC-PV-TP-NUW	CC-PC-TE-NUW	INCOMP	Plur.18 + Plur.23
CC-PV-TP-NUW	CC-PC-TP-UW	INCOMP	Plur.2 + Plur.24
CC-PV-TP-NUW	CC-PC-TP-NUW	INCOMP	Plur.20 + Plur.23
CC-PV-TP-NUW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-PV-TP-NUW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-PV-TP-NUW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-PV-TP-NUW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-PV-TP-NUW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-PV-TP-NUW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-PV-TP-NUW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-PV-TP-NUW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-PV-TP-NUW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-PV-TP-NUW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-PV-TP-NUW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-PV-TP-NUW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-RPC-TE-UW	CC-RPC-TE-NUW	$\subsetneq$	Plur.2
CC-RPC-TE-UW	CC-RPC-TP-UW	INCOMP*	Plur.22
CC-RPC-TE-UW	CC-RPC-TP-NUW	INCOMP	Plur.2 + Plur.44
CC-RPC-TE-UW	CC-PC-TE-UW	INCOMP	Plur.22 + Plur.49
CC-RPC-TE-UW	CC-PC-TE-NUW	INCOMP	Plur.2 + Plur.49

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-RPC-TE-UW	CC-PC-TP-UW	INCOMP	Plur.18 + Plur.47
CC-RPC-TE-UW	CC-PC-TP-NUW	INCOMP	Plur.2 + Plur.44
CC-RPC-TE-UW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-RPC-TE-UW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-RPC-TE-UW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-RPC-TE-UW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-RPC-TE-UW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-RPC-TE-UW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-RPC-TE-UW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-RPC-TE-UW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-RPC-TE-UW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-RPC-TE-UW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-RPC-TE-UW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-RPC-TE-UW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-RPC-TE-NUW	CC-RPC-TP-UW	INCOMP	Plur.2 + Plur.4
CC-RPC-TE-NUW	CC-RPC-TP-NUW	INCOMP	Plur.18 + Plur.45
CC-RPC-TE-NUW	CC-PC-TE-UW	INCOMP	Plur.2 + Plur.47
CC-RPC-TE-NUW	CC-PC-TE-NUW	INCOMP	Plur.15 + Plur.49
CC-RPC-TE-NUW	CC-PC-TP-UW	INCOMP	Plur.2 + Plur.47
CC-RPC-TE-NUW	CC-PC-TP-NUW	INCOMP	Plur.6 + Plur.45
CC-RPC-TE-NUW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-RPC-TE-NUW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-RPC-TE-NUW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-RPC-TE-NUW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-RPC-TE-NUW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-RPC-TE-NUW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-RPC-TE-NUW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-RPC-TE-NUW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-RPC-TE-NUW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-RPC-TE-NUW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-RPC-TE-NUW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-RPC-TE-NUW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-RPC-TP-UW	CC-RPC-TP-NUW	$\subsetneq$	Plur.2
CC-RPC-TP-UW	CC-PC-TE-UW	INCOMP	Plur.18 + Plur.47
CC-RPC-TP-UW	CC-PC-TE-NUW	INCOMP	Plur.2 + Plur.49
CC-RPC-TP-UW	CC-PC-TP-UW	INCOMP	Plur.7 + Plur.47
CC-RPC-TP-UW	CC-PC-TP-NUW	INCOMP	Plur.2 + Plur.49
CC-RPC-TP-UW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-RPC-TP-UW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-RPC-TP-UW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-RPC-TP-UW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-RPC-TP-UW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-RPC-TP-UW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-RPC-TP-UW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-RPC-TP-UW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-RPC-TP-UW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-RPC-TP-UW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-RPC-TP-UW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-RPC-TP-UW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-RPC-TP-NUW	CC-PC-TE-UW	INCOMP	Plur.2 + Plur.9
CC-RPC-TP-NUW	CC-PC-TE-NUW	INCOMP	Plur.6 + Plur.49
CC-RPC-TP-NUW	CC-PC-TP-UW	INCOMP	Plur.2 + Plur.47
CC-RPC-TP-NUW	CC-PC-TP-NUW	INCOMP	Plur.8 + Plur.49
CC-RPC-TP-NUW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-RPC-TP-NUW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-RPC-TP-NUW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-RPC-TP-NUW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-RPC-TP-NUW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-RPC-TP-NUW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-RPC-TP-NUW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-RPC-TP-NUW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-RPC-TP-NUW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-RPC-TP-NUW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-RPC-TP-NUW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-RPC-TP-NUW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-PC-TE-UW	CC-PC-TE-NUW	$\subsetneq$	Plur.2
CC-PC-TE-UW	CC-PC-TP-UW	INCOMP	Plur.17 + Plur.48

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-PC-TE-UW	CC-PC-TP-NUW	INCOMP	Plur.2 + Plur.44
CC-PC-TE-UW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-PC-TE-UW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-PC-TE-UW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-PC-TE-UW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-PC-TE-UW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-PC-TE-UW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-PC-TE-UW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-PC-TE-UW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-PC-TE-UW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-PC-TE-UW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-PC-TE-UW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-PC-TE-UW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-PC-TE-NUW	CC-PC-TP-UW	INCOMP	Plur.2 + Plur.48
CC-PC-TE-NUW	CC-PC-TP-NUW	INCOMP	Plur.6 + Plur.48
CC-PC-TE-NUW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-PC-TE-NUW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-PC-TE-NUW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-PC-TE-NUW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-PC-TE-NUW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-PC-TE-NUW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-PC-TE-NUW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-PC-TE-NUW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-PC-TE-NUW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-PC-TE-NUW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-PC-TE-NUW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-PC-TE-NUW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-PC-TP-UW	CC-PC-TP-NUW	$\subsetneq$	Plur.2
CC-PC-TP-UW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-PC-TP-UW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-PC-TP-UW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-PC-TP-UW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-PC-TP-UW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-PC-TP-UW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-PC-TP-UW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-PC-TP-UW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-PC-TP-UW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-PC-TP-UW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-PC-TP-UW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-PC-TP-UW	DC-PC-TP-NUW	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-PV-TE-UW	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-PV-TE-NUW	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-PV-TP-UW	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-PV-TP-NUW	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-RPC-TE-UW	INCOMP*	Plur.3
CC-PC-TP-NUW	**DC-RPC-TE-NUW**	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-RPC-TP-UW	INCOMP*	Plur.3
CC-PC-TP-NUW	**DC-RPC-TP-NUW**	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-PC-TE-UW	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-PC-TE-NUW	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-PC-TP-UW	INCOMP*	Plur.3
CC-PC-TP-NUW	DC-PC-TP-NUW	INCOMP*	Plur.3
DC-PV-TE-UW	DC-PV-TE-NUW	$\supsetneq$	Plur.9
DC-PV-TE-UW	DC-PV-TP-UW	INCOMP	Plur.12 + Plur.26
DC-PV-TE-UW	DC-PV-TP-NUW	INCOMP	Plur.2 + Plur.26
DC-PV-TE-UW	DC-RPC-TE-UW	INCOMP	Plur.10 + Plur.25
DC-PV-TE-UW	**DC-RPC-TE-NUW**	INCOMP	Plur.10 + Plur.25
DC-PV-TE-UW	DC-RPC-TP-UW	INCOMP	Plur.16 + Plur.25
DC-PV-TE-UW	**DC-RPC-TP-NUW**	INCOMP	Plur.2 + Plur.25
DC-PV-TE-UW	DC-PC-TE-UW	INCOMP	Plur.10 + Plur.25
DC-PV-TE-UW	DC-PC-TE-NUW	INCOMP	Plur.10 + Plur.25
DC-PV-TE-UW	DC-PC-TP-UW	INCOMP	Plur.13 + Plur.25
DC-PV-TE-UW	DC-PC-TP-NUW	INCOMP	Plur.2 + Plur.25
DC-PV-TE-NUW	DC-PV-TP-UW	INCOMP	Plur.19 + Plur.10
DC-PV-TE-NUW	DC-PV-TP-NUW	INCOMP	Plur.2 + Plur.26
DC-PV-TE-NUW	DC-RPC-TE-UW	INCOMP	Plur.19 + Plur.10
DC-PV-TE-NUW	**DC-RPC-TE-NUW**	INCOMP	Plur.19 + Plur.10

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
DC-PV-TE-NUW	DC-RPC-TP-UW	INCOMP	Plur.16 + Plur.25
DC-PV-TE-NUW	**DC-RPC-TP-NUW**	INCOMP	Plur.2 + Plur.25
DC-PV-TE-NUW	DC-PC-TE-UW	INCOMP	Plur.19 + Plur.10
DC-PV-TE-NUW	DC-PC-TE-NUW	INCOMP	Plur.19 + Plur.10
DC-PV-TE-NUW	DC-PC-TP-UW	INCOMP	Plur.19 + Plur.10
DC-PV-TE-NUW	DC-PC-TP-NUW	INCOMP	Plur.2 + Plur.25
DC-PV-TP-UW	DC-PV-TP-NUW	$\subsetneq$	Plur.2
DC-PV-TP-UW	DC-RPC-TE-UW	INCOMP	Plur.5 + Plur.25
DC-PV-TP-UW	**DC-RPC-TE-NUW**	INCOMP	Plur.5 + Plur.25
DC-PV-TP-UW	DC-RPC-TP-UW	INCOMP	Plur.13 + Plur.25
DC-PV-TP-UW	**DC-RPC-TP-NUW**	INCOMP	Plur.2 + Plur.25
DC-PV-TP-UW	DC-PC-TE-UW	INCOMP	Plur.5 + Plur.25
DC-PV-TP-UW	DC-PC-TE-NUW	INCOMP	Plur.5 + Plur.25
DC-PV-TP-UW	DC-PC-TP-UW	INCOMP	Plur.13 + Plur.25
DC-PV-TP-UW	DC-PC-TP-NUW	INCOMP	Plur.2 + Plur.25
DC-PV-TP-NUW	DC-RPC-TE-UW	INCOMP	Plur.2 + Plur.5
DC-PV-TP-NUW	**DC-RPC-TE-NUW**	INCOMP	Plur.2 + Plur.5
DC-PV-TP-NUW	DC-RPC-TP-UW	INCOMP	Plur.2 + Plur.5
DC-PV-TP-NUW	**DC-RPC-TP-NUW**	INCOMP	Plur.1 + Plur.5
DC-PV-TP-NUW	DC-PC-TE-UW	INCOMP	Plur.2 + Plur.5
DC-PV-TP-NUW	DC-PC-TE-NUW	INCOMP	Plur.2 + Plur.5
DC-PV-TP-NUW	DC-PC-TP-UW	INCOMP	Plur.2 + Plur.5
DC-PV-TP-NUW	DC-PC-TP-NUW	INCOMP	Plur.1 + Plur.5
DC-RPC-TE-UW	**DC-RPC-TE-NUW**	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-UW	DC-RPC-TP-UW	$\subsetneq$	Plur.13 + Theorem 3.1
DC-RPC-TE-UW	**DC-RPC-TP-NUW**	$\subsetneq$	Plur.2 + Theorem 3.1
DC-RPC-TE-UW	DC-PC-TE-UW	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-UW	DC-PC-TE-NUW	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-UW	DC-PC-TP-UW	$\subsetneq$	Plur.13 + Theorem 3.1
DC-RPC-TE-UW	DC-PC-TP-NUW	$\subsetneq$	Plur.2 + Theorem 3.1
DC-RPC-TE-NUW	DC-RPC-TP-UW	$\subsetneq$	Plur.13 + Theorem 3.1
DC-RPC-TE-NUW	**DC-RPC-TP-NUW**	$\subsetneq$	Plur.2 + Theorem 3.1
DC-RPC-TE-NUW	DC-PC-TE-UW	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-NUW	DC-PC-TE-NUW	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-NUW	DC-PC-TP-UW	$\subsetneq$	Plur.13 + Theorem 3.1
DC-RPC-TE-NUW	DC-PC-TP-NUW	$\subsetneq$	Plur.2 + Theorem 3.1
DC-RPC-TP-UW	**DC-RPC-TP-NUW**	$\subsetneq$	Plur.2 + Theorem 3.1
DC-RPC-TP-UW	DC-PC-TE-UW	$\subsetneq$	Plur.13 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-UW	DC-PC-TE-NUW	$\subsetneq$	Plur.13 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-UW	DC-PC-TP-UW	INCOMP	Plur.16 + Plur.11
DC-RPC-TP-UW	DC-PC-TP-NUW	$\subsetneq$	Plur.2 + Hemaspaandra et al. (2020)
DC-RPC-TP-NUW	DC-PC-TE-UW	$\subsetneq$	Plur.2 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-NUW	DC-PC-TE-NUW	$\subsetneq$	Plur.2 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-NUW	DC-PC-TP-UW	$\subsetneq$	Plur.2 + Hemaspaandra et al. (2020)
DC-RPC-TP-NUW	DC-PC-TP-NUW	EQ	Hemaspaandra et al. (2020)
DC-PC-TE-UW	DC-PC-TE-NUW	EQ	Hemaspaandra et al. (2020)
DC-PC-TE-UW	DC-PC-TP-UW	$\subsetneq$	Plur.13 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-PC-TE-UW	DC-PC-TP-NUW	$\subsetneq$	Plur.2 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-PC-TE-NUW	DC-PC-TP-UW	$\subsetneq$	Plur.13 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-PC-TE-NUW	DC-PC-TP-NUW	$\subsetneq$	Plur.2 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-PC-TP-UW	DC-PC-TP-NUW	$\subsetneq$	Plur.2
CC-AC-UW	CC-AC-NUW	$\subsetneq$	Plur.37
CC-AC-UW	DC-AC-UW	INCOMP	Plur.33 + Plur.27
CC-AC-UW	DC-AC-NUW	INCOMP	Plur.33 + Plur.27
CC-AC-NUW	DC-AC-UW	INCOMP	Plur.33 + Plur.27
CC-AC-NUW	DC-AC-NUW	INCOMP	Plur.33 + Plur.27
DC-AC-UW	DC-AC-NUW	$\subsetneq$	Plur.37
CC-DC-UW	CC-DC-NUW	$\subsetneq$	Plur.38
CC-DC-UW	CC-DV-UW	INCOMP	Plur.41 + Plur.31
CC-DC-UW	CC-DV-NUW	INCOMP	Plur.38 + Plur.41
CC-DC-UW	DC-DC-UW	INCOMP	Plur.38 + Plur.30
CC-DC-UW	DC-DC-NUW	INCOMP	Plur.34 + Plur.30
CC-DC-UW	DC-DV-UW	INCOMP	Plur.38 + Plur.30
CC-DC-UW	DC-DV-NUW	INCOMP	Plur.34 + Plur.30
CC-DC-NUW	CC-DV-UW	INCOMP	Plur.38 + Plur.31
CC-DC-NUW	CC-DV-NUW	INCOMP	Plur.41 + Plur.31
CC-DC-NUW	DC-DC-UW	INCOMP	Plur.34 + Plur.30

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-DC-NUW	DC-DC-NUW	INCOMP	Plur.34 + Plur.30
CC-DC-NUW	DC-DV-UW	INCOMP	Plur.34 + Plur.30
CC-DC-NUW	DC-DV-NUW	INCOMP	Plur.34 + Plur.30
CC-DV-UW	CC-DV-NUW	$\subsetneq$	Plur.38
CC-DV-UW	DC-DC-UW	INCOMP	Plur.38 + Plur.30
CC-DV-UW	DC-DC-NUW	INCOMP	Plur.34 + Plur.30
CC-DV-UW	DC-DV-UW	INCOMP	Plur.38 + Plur.30
CC-DV-UW	DC-DV-NUW	INCOMP	Plur.34 + Plur.30
CC-DV-NUW	DC-DC-UW	INCOMP	Plur.34 + Plur.30
CC-DV-NUW	DC-DC-NUW	INCOMP	Plur.34 + Plur.30
CC-DV-NUW	DC-DV-UW	INCOMP	Plur.34 + Plur.30
CC-DV-NUW	DC-DV-NUW	INCOMP	Plur.34 + Plur.30
DC-DC-UW	DC-DC-NUW	$\supsetneq$	Plur.38
DC-DC-UW	DC-DV-UW	INCOMP	Plur.42 + Plur.32
DC-DC-UW	DC-DV-NUW	INCOMP	Plur.38 + Plur.31
DC-DC-NUW	DC-DV-UW	INCOMP	Plur.38 + Plur.32
DC-DC-NUW	DC-DV-NUW	INCOMP	Plur.43 + Plur.32
DC-DV-UW	DC-DV-NUW	$\supsetneq$	Plur.38
CC-AV-UW	CC-AV-NUW	$\subsetneq$	Plur.39
CC-AV-UW	DC-AV-UW	INCOMP	Plur.39 + Plur.28
CC-AV-UW	DC-AV-NUW	INCOMP	Plur.35 + Plur.28
CC-AV-NUW	DC-AV-UW	INCOMP	Plur.35 + Plur.28
CC-AV-NUW	DC-AV-NUW	INCOMP	Plur.35 + Plur.28
DC-AV-UW	DC-AV-NUW	$\supsetneq$	Plur.39
CC-UAC-UW	CC-UAC-NUW	$\subsetneq$	Plur.40
CC-UAC-UW	DC-UAC-UW	INCOMP	Plur.40 + Plur.29
CC-UAC-UW	DC-UAC-NUW	INCOMP	Plur.36 + Plur.29
CC-UAC-NUW	DC-UAC-UW	INCOMP	Plur.36 + Plur.29
CC-UAC-NUW	DC-UAC-NUW	INCOMP	Plur.36 + Plur.29
DC-UAC-UW	DC-UAC-NUW	$\supsetneq$	Plur.40

Table 6: List of separation witnesses in veto. We note the computer-generated entries with a "†" superscript.

ID	C	S	V	U	k
Veto.1	$\{a,b\}$	-	$\{a>b, b>a\}$	-	-
Veto.2	$\{a,b,c\}$	-	$\{a>b>c, a>b>c\}$	-	-
Veto.3	$\{a,b,c\}$	-	$\{a>b>c, c>a>b, c>b>a, c>b>a\}$	-	-
Veto.4	$\{a,b,c\}$	-	$\{a>b>c\}$	-	-
Veto.5	$\{a,b\}$	-	$\{a>b\}$	-	-
Veto.6	$\{a,b\}$	-	$\{b>a\}$	-	-
Veto.7	$\{a,b,c\}$	-	$\{a>b>c, a>c>b, b>c>a, b>c>a\}$	-	-
Veto.8	$\{a,b,c\}$	-	$\{a>b>c, a>b>c, c>a>b, c>b>a, c>b>a\}$	-	-
Veto.9	$\{a,b,c,d\}$	-	$\{a>b>c>d, a>b>c>d, b>d>c>a\}$	-	-
Veto.10	$\{a,b,c\}$	-	$\{b>c>a, c>b>a\}$	-	-
Veto.11	$\{a,b,c\}$	-	$\{b>a>c\}$	-	-
Veto.12	$\{a,b,c,d\}$	-	$\{a>b>c>d, b>c>d>a, c>a>d>b\}$	-	-
Veto.13	$\{a,b,c\}$	-	$\{a>b>c, a>b>c, a>b>c, c>a>b, c>a>b, c>b>a, c>b>a\}$	-	-
Veto.14	$\{a,b,c,d\}$	-	$\{c>d>a>b, c>d>a>b, d>b>a>c\}$	-	-
Veto.15	$\{a,b,c,d\}$	-	$\{a>b>c>d, a>b>c>d, b>c>a>d, c>a>b>d, c>d>b>a\}$	-	-
Veto.16†	$\{a,b,c,d\}$	-	$\{a>b>c>d, b>a>c>d, b>a>c>d, b>a>c>d, c>b>a>d, d>c>a>b, d>c>a>b, d>c>a>b, d>c>b>a, d>c>b>a\}$	-	-
Veto.17	$\{a,b,c\}$	-	$\{b>a>c, c>a>b\}$	-	-
Veto.18	$\{a,b,c\}$	-	$\{a>b>c, a>b>c, a>b>c, a>c>b, a>c>b, a>c>b, b>c>a, b>c>a\}$	-	-
Veto.19	$\{a,b,c,d\}$	-	$\{a>b>c>d, b>d>a>c, c>d>a>b\}$	-	-
Veto.20	$\{a,b,c\}$	-	$\{a>b>c, c>a>b\}$	-	-
Veto.21	$\{a,b\}$	$\{c\}$	$\{b>a>c\}$	-	1
Veto.22	$\{a,b,c\}$	$\{d\}$	$\{b>a>c>d\}$	-	1
Veto.23	$\{a,b\}$	$\{c\}$	$\{a>b>c\}$	-	1
Veto.24	$\{a,b\}$	$\{c\}$	$\{b>c>a, c>b>a\}$	-	1
Veto.25	$\{a,b,c,d\}$	-	$\{d>c>a>b\}$	-	1
Veto.26	$\{a,b,c\}$	-	$\{a>b>c, a>b>c, c>a>b, c>b>a, c>b>a\}$	-	0
Veto.27	$\{a,b,c\}$	-	$\{a>c>b, a>c>b\}$	-	1

ID	C	S	V	U	k
Veto.28	$\{a,b,c\}$	-	$\{c>a>b, c>a>b\}$	-	1
Veto.29	$\{a,b\}$	-	$\{a>b\}$	-	1
Veto.30	$\{a,b,c\}$	-	$\{c>b>a, c>a>b, b>a>c\}$	-	1
Veto.31	$\{a,b,c\}$	-	$\{a>c>b, a>c>b, a>c>b, c>b>a, c>b>a\}$	-	1
Veto.32	$\{a,b,c,d\}$	-	$\{a>b>c>d, a>b>c>d, b>d>c>a\}$	-	0
Veto.33	$\{a,b,c\}$	-	$\{b>c>a, b>c>a, b>a>c, b>a>c, a>c>b\}$	-	2
Veto.34	$\{a,b,c\}$	-	$\{c>b>a, c>b>a, b>c>a\}$	-	1
Veto.35	$\{a,b,c\}$	-	$\{c>a>b\}$	-	1
Veto.36	$\{a,b,c\}$	-	$\{c>a>b\}$	$\{c>a>b\}$	1
Veto.37	$\{a,b\}$	-	$\{a>b\}$	$\{a>b\}$	1
Veto.38	$\{a,b\}$	-	$\{b>a\}$	$\{b>a\}$	1
Veto.39	$\{a,b,c\}$	$\{d\}$	$\{d>c>a>b\}$	-	-
Veto.40	$\{a,b\}$	$\{c\}$	$\{a>c>b, a>b>c\}$	-	-
Veto.41	$\{a,b\}$	$\{c\}$	$\{c>b>a\}$	-	-
Veto.42	$\{a,b,c\}$	-	$\{a>b>c, a>c>b\}$	-	-
Veto.43†	$\{a,b,c,d\}$	-	$\{c>d>b>a, a>b>c>d, b>d>a>c, b>d>c>a, a>b>d>c, a>b>d>c\}$	-	-
Veto.44	$\{a,b,c\}$	-	$\{a>b>c\}$	$\emptyset$	0
Veto.45	$\{a,b\}$	-	$\{a>b\}$	-	0
Veto.46	$\{a,b,c,d,e\}$	-	$\{b>c>d>e>a, b>c>d>e>a, d>b>c>a>e, e>b>c>a>d, e>c>d>a>b, e>b>d>a>c\}$	-	2

Table 7: Table of separations and collapses (here denoted by EQ) in veto voting. The Classification column partitions each separation into one of the three cases, $\subsetneq$, $\supsetneq$, and INCOMP. (For cases of INCOMP for which we happen to have established that the separation holds with strong incomparability, we have noted that with a "$*$" superscript.)

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-PV-TE-UW	CC-PV-TE-NUW	$\subsetneq$	Veto.1
CC-PV-TE-UW	CC-PV-TP-UW	INCOMP	Veto.2 + Veto.3
CC-PV-TE-UW	CC-PV-TP-NUW	INCOMP	Veto.1 + Veto.3
CC-PV-TE-UW	CC-RPC-TE-UW	INCOMP	Veto.4 + Veto.8
CC-PV-TE-UW	CC-RPC-TE-NUW	INCOMP	Veto.1 + Veto.8
CC-PV-TE-UW	CC-RPC-TP-UW	INCOMP	Veto.4 + Veto.8
CC-PV-TE-UW	CC-RPC-TP-NUW	INCOMP	Veto.1 + Veto.8
CC-PV-TE-UW	CC-PC-TE-UW	INCOMP	Veto.4 + Veto.8
CC-PV-TE-UW	CC-PC-TE-NUW	INCOMP	Veto.1 + Veto.8
CC-PV-TE-UW	CC-PC-TP-UW	INCOMP	Veto.4 + Veto.8
CC-PV-TE-UW	CC-PC-TP-NUW	INCOMP	Veto.1 + Veto.8
CC-PV-TE-UW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-UW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	CC-PV-TP-UW	INCOMP	Veto.1 + Veto.2
CC-PV-TE-NUW	CC-PV-TP-NUW	INCOMP	Veto.4 + Veto.7
CC-PV-TE-NUW	CC-RPC-TE-UW	INCOMP	Veto.1 + Veto.2
CC-PV-TE-NUW	CC-RPC-TE-NUW	INCOMP	Veto.4 + Veto.8
CC-PV-TE-NUW	CC-RPC-TP-UW	INCOMP	Veto.1 + Veto.2
CC-PV-TE-NUW	CC-RPC-TP-NUW	INCOMP	Veto.4 + Veto.8
CC-PV-TE-NUW	CC-PC-TE-UW	INCOMP	Veto.1 + Veto.2
CC-PV-TE-NUW	CC-PC-TE-NUW	INCOMP	Veto.4 + Veto.8
CC-PV-TE-NUW	CC-PC-TP-UW	INCOMP	Veto.1 + Veto.2
CC-PV-TE-NUW	CC-PC-TP-NUW	INCOMP	Veto.4 + Veto.8
CC-PV-TE-NUW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-PV-TE-NUW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TE-NUW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	CC-PV-TP-NUW	$\subsetneq$	Veto.1
CC-PV-TP-UW	CC-RPC-TE-UW	INCOMP	Veto.4 + Veto.8
CC-PV-TP-UW	CC-RPC-TE-NUW	INCOMP	Veto.1 + Veto.8
CC-PV-TP-UW	CC-RPC-TP-UW	INCOMP	Veto.4 + Veto.8
CC-PV-TP-UW	CC-RPC-TP-NUW	INCOMP	Veto.1 + Veto.8
CC-PV-TP-UW	CC-PC-TE-UW	INCOMP	Veto.4 + Veto.8
CC-PV-TP-UW	CC-PC-TE-NUW	INCOMP	Veto.1 + Veto.8
CC-PV-TP-UW	CC-PC-TP-UW	INCOMP	Veto.4 + Veto.8
CC-PV-TP-UW	CC-PC-TP-NUW	INCOMP	Veto.1 + Veto.8
CC-PV-TP-UW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-UW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	CC-RPC-TE-UW	INCOMP	Veto.1 + Veto.9
CC-PV-TP-NUW	CC-RPC-TE-NUW	INCOMP	Veto.10 + Veto.8
CC-PV-TP-NUW	CC-RPC-TP-UW	INCOMP	Veto.1 + Veto.9
CC-PV-TP-NUW	CC-RPC-TP-NUW	INCOMP	Veto.11 + Veto.9
CC-PV-TP-NUW	CC-PC-TE-UW	INCOMP	Veto.1 + Veto.9
CC-PV-TP-NUW	CC-PC-TE-NUW	INCOMP	Veto.10 + Veto.8
CC-PV-TP-NUW	CC-PC-TP-UW	INCOMP	Veto.1 + Veto.9
CC-PV-TP-NUW	CC-PC-TP-NUW	INCOMP	Veto.12 + Veto.8
CC-PV-TP-NUW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PV-TP-NUW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	CC-RPC-TE-NUW	$\subsetneq$	Veto.1
CC-RPC-TE-UW	CC-RPC-TP-UW	INCOMP	Veto.10 + Veto.13
CC-RPC-TE-UW	CC-RPC-TP-NUW	INCOMP	Veto.1 + Veto.10
CC-RPC-TE-UW	CC-PC-TE-UW	INCOMP	Veto.12 + Veto.14
CC-RPC-TE-UW	CC-PC-TE-NUW	INCOMP	Veto.1 + Veto.15
CC-RPC-TE-UW	CC-PC-TP-UW	INCOMP	Veto.10 + Veto.13
CC-RPC-TE-UW	CC-PC-TP-NUW	INCOMP	Veto.1 + Veto.10
CC-RPC-TE-UW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-RPC-TE-UW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-UW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	CC-RPC-TP-UW	INCOMP	Veto.1 + Veto.13
CC-RPC-TE-NUW	CC-RPC-TP-NUW	INCOMP	Veto.10 + Veto.13
CC-RPC-TE-NUW	CC-PC-TE-UW	INCOMP	Veto.1 + Veto.14
CC-RPC-TE-NUW	CC-PC-TE-NUW	INCOMP	Veto.11 + Veto.15
CC-RPC-TE-NUW	CC-PC-TP-UW	INCOMP	Veto.1 + Veto.13
CC-RPC-TE-NUW	CC-PC-TP-NUW	INCOMP	Veto.10 + Veto.11
CC-RPC-TE-NUW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TE-NUW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	CC-RPC-TP-NUW	$\subsetneq$	Veto.1
CC-RPC-TP-UW	CC-PC-TE-UW	INCOMP	Veto.10 + Veto.12
CC-RPC-TP-UW	CC-PC-TE-NUW	INCOMP	Veto.1 + Veto.15
CC-RPC-TP-UW	CC-PC-TP-UW	INCOMP	Veto.12 + Veto.14
CC-RPC-TP-UW	CC-PC-TP-NUW	INCOMP	Veto.1 + Veto.15
CC-RPC-TP-UW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-UW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	CC-PC-TE-UW	INCOMP	Veto.1 + Veto.10
CC-RPC-TP-NUW	CC-PC-TE-NUW	INCOMP	Veto.10 + Veto.15
CC-RPC-TP-NUW	CC-PC-TP-UW	INCOMP	Veto.1 + Veto.14
CC-RPC-TP-NUW	CC-PC-TP-NUW	INCOMP	Veto.11 + Veto.15
CC-RPC-TP-NUW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-RPC-TP-NUW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	CC-PC-TE-NUW	$\subsetneq$	Veto.1
CC-PC-TE-UW	CC-PC-TP-UW	INCOMP	Veto.10 + Veto.13
CC-PC-TE-UW	CC-PC-TP-NUW	INCOMP	Veto.1 + Veto.10
CC-PC-TE-UW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-PC-TE-UW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-UW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	CC-PC-TP-UW	INCOMP	Veto.1 + Veto.16
CC-PC-TE-NUW	CC-PC-TP-NUW	INCOMP	Veto.10 + Veto.16
CC-PC-TE-NUW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TE-NUW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	CC-PC-TP-NUW	$\subsetneq$	Veto.1
CC-PC-TP-UW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-UW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	DC-PV-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	**DC-PV-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	DC-PV-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	DC-PV-TP-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	DC-RPC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	**DC-RPC-TE-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	DC-RPC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	**DC-RPC-TP-NUW**	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	DC-PC-TE-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	DC-PC-TE-NUW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	DC-PC-TP-UW	INCOMP	Veto.5 + Veto.6
CC-PC-TP-NUW	DC-PC-TP-NUW	INCOMP	Veto.5 + Veto.6
DC-PV-TE-UW	**DC-PV-TE-NUW**	EQ	Theorem 3.2
DC-PV-TE-UW	DC-PV-TP-UW	$\not\subseteq$	Veto.42 + Theorem 3.3
DC-PV-TE-UW	DC-PV-TP-NUW	$\not\subseteq$	Veto.1 + Theorem 3.3
DC-PV-TE-UW	DC-RPC-TE-UW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-UW	**DC-RPC-TE-NUW**	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-UW	DC-RPC-TP-UW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-UW	**DC-RPC-TP-NUW**	$\not\subseteq$	Veto.1 + Theorem 3.4
DC-PV-TE-UW	DC-PC-TE-UW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-UW	DC-PC-TE-NUW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-UW	DC-PC-TP-UW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-UW	DC-PC-TP-NUW	$\not\subseteq$	Veto.1 + Theorem 3.4 + Hemaspaandra et al. (2020)
DC-PV-TE-NUW	DC-PV-TP-UW	$\not\subseteq$	Veto.42 + Theorem 3.3
DC-PV-TE-NUW	DC-PV-TP-NUW	$\not\subseteq$	Veto.1 + Theorem 3.3
DC-PV-TE-NUW	DC-RPC-TE-UW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-NUW	**DC-RPC-TE-NUW**	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-NUW	DC-RPC-TP-UW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-NUW	**DC-RPC-TP-NUW**	$\not\subseteq$	Veto.1 + Theorem 3.4
DC-PV-TE-NUW	DC-PC-TE-UW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-NUW	DC-PC-TE-NUW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-NUW	DC-PC-TP-UW	$\not\subseteq$	Veto.42 + Theorem 3.4
DC-PV-TE-NUW	DC-PC-TP-NUW	$\not\subseteq$	Veto.1 + Theorem 3.4 + Hemaspaandra et al. (2020)
DC-PV-TP-UW	DC-PV-TP-NUW	$\not\subseteq$	Veto.1
DC-PV-TP-UW	DC-RPC-TE-UW	INCOMP	Veto.17 + Veto.18
DC-PV-TP-UW	**DC-RPC-TE-NUW**	INCOMP	Veto.17 + Veto.18
DC-PV-TP-UW	DC-RPC-TP-UW	INCOMP	Veto.4 + Veto.20
DC-PV-TP-UW	**DC-RPC-TP-NUW**	INCOMP	Veto.1 + Veto.19

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
DC-PV-TP-UW	DC-PC-TE-UW	INCOMP	Veto.17 + Veto.18
DC-PV-TP-UW	DC-PC-TE-NUW	INCOMP	Veto.17 + Veto.18
DC-PV-TP-UW	DC-PC-TP-UW	INCOMP	Veto.20 + Veto.18
DC-PV-TP-UW	DC-PC-TP-NUW	INCOMP	Veto.1 + Veto.19
DC-PV-TP-NUW	DC-RPC-TE-UW	INCOMP	Veto.1 + Veto.18
DC-PV-TP-NUW	**DC-RPC-TE-NUW**	INCOMP	Veto.1 + Veto.18
DC-PV-TP-NUW	DC-RPC-TP-UW	INCOMP	Veto.1 + Veto.18
DC-PV-TP-NUW	**DC-RPC-TP-NUW**	INCOMP	Veto.11 + Veto.18
DC-PV-TP-NUW	DC-PC-TE-UW	INCOMP	Veto.1 + Veto.18
DC-PV-TP-NUW	DC-PC-TE-NUW	INCOMP	Veto.1 + Veto.18
DC-PV-TP-NUW	DC-PC-TP-UW	INCOMP	Veto.1 + Veto.18
DC-PV-TP-NUW	DC-PC-TP-NUW	INCOMP	Veto.11 + Veto.18
DC-RPC-TE-UW	**DC-RPC-TE-NUW**	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-UW	DC-RPC-TP-UW	$\subsetneq$	Veto.4 + Theorem 3.1
DC-RPC-TE-UW	**DC-RPC-TP-NUW**	$\subsetneq$	Veto.1 + Theorem 3.1
DC-RPC-TE-UW	DC-PC-TE-UW	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-UW	DC-PC-TE-NUW	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-UW	DC-PC-TP-UW	$\subsetneq$	Veto.17 + Theorem 3.1
DC-RPC-TE-UW	DC-PC-TP-NUW	$\subsetneq$	Veto.1 + Theorem 3.1
DC-RPC-TE-NUW	DC-RPC-TP-UW	$\subsetneq$	Veto.4 + Theorem 3.1
DC-RPC-TE-NUW	**DC-RPC-TP-NUW**	$\subsetneq$	Veto.1 + Theorem 3.1
DC-RPC-TE-NUW	DC-PC-TE-UW	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-NUW	DC-PC-TE-NUW	EQ	Hemaspaandra et al. (2020)
DC-RPC-TE-NUW	DC-PC-TP-UW	$\subsetneq$	Veto.17 + Theorem 3.1
DC-RPC-TE-NUW	DC-PC-TP-NUW	$\subsetneq$	Veto.1 + Theorem 3.1
DC-RPC-TP-UW	**DC-RPC-TP-NUW**	$\subsetneq$	Veto.1
DC-RPC-TP-UW	DC-PC-TE-UW	$\subsetneq$	Veto.4 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-UW	DC-PC-TE-NUW	$\subsetneq$	Veto.4 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-UW	DC-PC-TP-UW	INCOMP	Veto.4 + Veto.43
DC-RPC-TP-UW	DC-PC-TP-NUW	$\subsetneq$	Veto.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-NUW	DC-PC-TE-UW	$\subsetneq$	Veto.1 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-NUW	DC-PC-TE-NUW	$\subsetneq$	Veto.1 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-NUW	DC-PC-TP-UW	$\subsetneq$	Veto.1 + Hemaspaandra et al. (2020)
DC-RPC-TP-NUW	DC-PC-TP-NUW	EQ	Hemaspaandra et al. (2020)
DC-PC-TE-UW	DC-PC-TE-NUW	EQ	Hemaspaandra et al. (2020)
DC-PC-TE-UW	DC-PC-TP-UW	$\subsetneq$	Veto.17 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-PC-TE-UW	DC-PC-TP-NUW	$\subsetneq$	Veto.1 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-PC-TE-NUW	DC-PC-TP-UW	$\subsetneq$	Veto.17 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-PC-TE-NUW	DC-PC-TP-NUW	$\subsetneq$	Veto.1 + Theorem 3.1 + Hemaspaandra et al. (2020)
DC-PC-TP-UW	DC-PC-TP-NUW	$\subsetneq$	Veto.1
CC-AC-UW	CC-AC-NUW	$\subsetneq$	Veto.22
CC-AC-UW	DC-AC-UW	INCOMP	Veto.22 + Veto.24
CC-AC-UW	DC-AC-NUW	INCOMP	Veto.21 + Veto.23
CC-AC-NUW	DC-AC-UW	INCOMP*	Veto.26
CC-AC-NUW	DC-AC-NUW	INCOMP*	Veto.32
DC-AC-UW	DC-AC-NUW	$\subsetneq$	Veto.22
CC-DC-UW	CC-DC-NUW	$\subsetneq$	Veto.25
CC-DC-UW	CC-DV-UW	INCOMP	Veto.27 + Veto.30
CC-DC-UW	CC-DV-NUW	INCOMP	Veto.25 + Veto.31
CC-DC-UW	DC-DC-UW	INCOMP*	Veto.26
CC-DC-UW	DC-DC-NUW	INCOMP	Veto.25 + Veto.34
CC-DC-UW	DC-DV-UW	INCOMP*	Veto.45
CC-DC-UW	DC-DV-NUW	INCOMP	Veto.25 + Veto.34
CC-DC-NUW	CC-DV-UW	INCOMP	Veto.27 + Veto.46
CC-DC-NUW	CC-DV-NUW	INCOMP	Veto.35 + Veto.31
CC-DC-NUW	DC-DC-UW	INCOMP	Veto.25 + Veto.34
CC-DC-NUW	DC-DC-NUW	INCOMP	Veto.25 + Veto.34
CC-DC-NUW	DC-DV-UW	INCOMP	Veto.25 + Veto.34
CC-DC-NUW	DC-DV-NUW	INCOMP	Veto.25 + Veto.34
CC-DV-UW	CC-DV-NUW	$\subsetneq$	Veto.27
CC-DV-UW	DC-DC-UW	INCOMP*	Veto.45
CC-DV-UW	DC-DC-NUW	INCOMP	Veto.25 + Veto.28
CC-DV-UW	DC-DV-UW	INCOMP*	Veto.45
CC-DV-UW	DC-DV-NUW	INCOMP	Veto.25 + Veto.34
CC-DV-NUW	DC-DC-UW	INCOMP*	Veto.26
CC-DV-NUW	DC-DC-NUW	INCOMP	Veto.25 + Veto.34
CC-DV-NUW	DC-DV-UW	INCOMP*	Veto.26

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-DV-NUW	DC-DV-NUW	INCOMP	Veto.25 + Veto.34
DC-DC-UW	DC-DC-NUW	$\subsetneq$	Veto.27
DC-DC-UW	DC-DV-UW	INCOMP	Veto.29 + Veto.31
DC-DC-UW	DC-DV-NUW	INCOMP	Veto.27 + Veto.33
DC-DC-NUW	DC-DV-UW	INCOMP	Veto.27 + Veto.31
DC-DC-NUW	DC-DV-NUW	INCOMP	Veto.28 + Veto.33
DC-DV-UW	DC-DV-NUW	$\supsetneq$	Veto.27
CC-AV-UW	CC-AV-NUW	$\subsetneq$	Veto.44
CC-AV-UW	DC-AV-UW	INCOMP	Veto.37 + Veto.38
CC-AV-UW	DC-AV-NUW	INCOMP	Veto.37 + Veto.38
CC-AV-NUW	DC-AV-UW	INCOMP	Veto.37 + Veto.38
CC-AV-NUW	DC-AV-NUW	INCOMP	Veto.37 + Veto.38
DC-AV-UW	DC-AV-NUW	$\supsetneq$	Veto.36
CC-UAC-UW	CC-UAC-NUW	$\subsetneq$	Veto.39
CC-UAC-UW	DC-UAC-UW	INCOMP	Veto.39 + Veto.40
CC-UAC-UW	DC-UAC-NUW	INCOMP	Veto.40 + Veto.41
CC-UAC-NUW	DC-UAC-UW	INCOMP	Veto.40 + Veto.41
CC-UAC-NUW	DC-UAC-NUW	INCOMP	Veto.39 + Veto.41
DC-UAC-UW	DC-UAC-NUW	$\subsetneq$	Veto.39

Table 8: List of separation witnesses in approval voting. (Example Appr.7 can also be found on page 220 of the Handbook of Approval Voting, Baumeister et al., 2010.)

ID	C	S	V	U	k
Appr.1	$\{a,b\}$	-	$\{10\}$	-	-
Appr.2	$\{a,b\}$	-	$\{10,01\}$	-	-
Appr.3	$\{a,b,c\}$	-	$\{101,110\}$	-	-
Appr.4	$\{a,b,c\}$	-	$\{110,110,010,101,101,001\}$	-	-
Appr.5	$\{a,b,c\}$	-	$\{100,011,011\}$	-	-
Appr.6	$\{a,b,c\}$	-	$\{100,110,011,011\}$	-	-
Appr.7	$\{a,b,c\}$	-	{100, 100, 100, 100, 100, 110, 010, 010, 010, 010, 001, 001, 001, 001, 001, 001, 001}	-	-
Appr.8	$\{a,b,c\}$	-	{100, 100, 100, 100, 010, 010, 010, 010, 010, 001, 001, 001}	-	-
Appr.9	$\{a,b,c,d\}$	-	{1001, 1001, 1001, 1000, 0100, 0100, 0100, 0100, 0100, 0010, 0010, 0010, 0010, 0010}	-	-
Appr.10	$\{a,b\}$	-	$\{10\}$	-	0
Appr.11	$\{a,b\}$	$\{c\}$	$\{111\}$	-	1
Appr.12	$\{a,b\}$	$\{c\}$	$\{010\}$	-	1
Appr.13	$\{a,b\}$	$\{c\}$	$\{100\}$	-	1
Appr.14	$\{a,b,c\}$	-	$\{111\}$	-	1
Appr.15	$\{a,b\}$	-	$\{11\}$	-	1
Appr.16	$\{a,b,c\}$	-	$\{011\}$	-	1
Appr.17	$\{a,b,c\}$	-	$\{011,011\}$	-	1
Appr.18	$\{a,b,c\}$	-	$\{100,111\}$	-	1
Appr.19	$\{a,b,c\}$	-	$\{100,011\}$	-	1
Appr.20	$\{a,b\}$	-	$\{10,01,01,01\}$	-	1
Appr.21	$\{a,b,c,d\}$	-	$\{1000,0111,0111\}$	-	2
Appr.22	$\{a,b\}$	-	$\{10,10,01\}$	-	2
Appr.23	$\{a\}$	-	$\{1\}$	-	1
Appr.24	$\{a,b\}$	-	$\{01\}$	$\{10\}$	1
Appr.25	$\{a,b\}$	-	$\{10\}$	$\{10\}$	1
Appr.26	$\{a,b\}$	-	$\{01\}$	$\{01\}$	1
Appr.27	$\{a,b\}$	-	$\{10,01\}$	$\emptyset$	0
Appr.28	$\{a,b\}$	$\{c\}$	$\{100\}$	-	-
Appr.29	$\{a,b,c\}$	-	$\{101,110\}$	-	0
Appr.30	$\{a,b\}$	$\emptyset$	$\{10,01\}$	-	-
Appr.31	$\{a,b,c,d,e,f,g,h\}$	-	$\{10111100,10111100,11100000,01000001,01000001,00010001,00001011,00000111,10111110,11011110\}$	-	-
Appr.32	$\{a,b,c,d,e,f,g,h,i,j\}$	-	$\{1110111100,1110111100,1110111100,1111000001,0001000001,0001000001,0001000001,0000100001,0000010011,0000001011,0000000111,1011111110,1101111110,1110111110\}$	-	-

Table 9: Table of separations and collapses (here denoted by EQ) in approval voting. The Classification column partitions each separation into one of the three cases, $\subsetneq$, $\supsetneq$, and INCOMP. (For cases of INCOMP for which we happen to have established that the separation holds with strong incomparability, we have noted that with a "$*$" superscript.)

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-PV-TE-UW	CC-PV-TE-NUW	$\subsetneq$	Appr.2
CC-PV-TE-UW	CC-PV-TP-UW	INCOMP	Appr.5 + Appr.31
CC-PV-TE-UW	CC-PV-TP-NUW	INCOMP	Appr.2 + Appr.5
CC-PV-TE-UW	**CC-RPC-TE-UW**	INCOMP	Appr.6 + Appr.9
CC-PV-TE-UW	**CC-RPC-TE-NUW**	INCOMP	Appr.2 + Appr.6
CC-PV-TE-UW	**CC-RPC-TP-UW**	$\supsetneq$	Appr.5 + Corollary 3.10
CC-PV-TE-UW	**CC-RPC-TP-NUW**	INCOMP	Appr.2 + Appr.5
CC-PV-TE-UW	CC-PC-TE-UW	INCOMP	Appr.6 + Appr.9
CC-PV-TE-UW	CC-PC-TE-NUW	INCOMP	Appr.2 + Appr.6
CC-PV-TE-UW	CC-PC-TP-UW	$\supsetneq$	Appr.5 + Corollary 3.10
CC-PV-TE-UW	CC-PC-TP-NUW	INCOMP	Appr.2 + Appr.5
CC-PV-TE-UW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-PV-TE-UW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-PV-TE-UW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-PV-TE-UW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-PV-TE-UW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-PV-TE-UW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-PV-TE-UW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-PV-TE-UW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-PV-TE-UW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-PV-TE-UW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-PV-TE-UW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-PV-TE-UW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-PV-TE-NUW	CC-PV-TP-UW	INCOMP	Appr.2 + Appr.32
CC-PV-TE-NUW	CC-PV-TP-NUW	INCOMP*	Appr.5
CC-PV-TE-NUW	**CC-RPC-TE-UW**	INCOMP	Appr.2 + Appr.9
CC-PV-TE-NUW	**CC-RPC-TE-NUW**	INCOMP	Appr.6 + Appr.5
CC-PV-TE-NUW	**CC-RPC-TP-UW**	$\supsetneq$	Appr.2 + Corollary 3.10
CC-PV-TE-NUW	**CC-RPC-TP-NUW**	INCOMP*	Appr.5
CC-PV-TE-NUW	CC-PC-TE-UW	INCOMP	Appr.2 + Appr.9
CC-PV-TE-NUW	CC-PC-TE-NUW	INCOMP	Appr.6 + Appr.5
CC-PV-TE-NUW	CC-PC-TP-UW	$\supsetneq$	Appr.2 + Corollary 3.10
CC-PV-TE-NUW	CC-PC-TP-NUW	INCOMP*	Appr.5
CC-PV-TE-NUW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-PV-TE-NUW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-PV-TE-NUW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-PV-TE-NUW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-PV-TE-NUW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-PV-TE-NUW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-PV-TE-NUW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-PV-TE-NUW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-PV-TE-NUW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-PV-TE-NUW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-PV-TE-NUW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-PV-TE-NUW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-PV-TP-UW	CC-PV-TP-NUW	$\subsetneq$	Appr.2
CC-PV-TP-UW	**CC-RPC-TE-UW**	INCOMP	Appr.5 + Appr.8
CC-PV-TP-UW	**CC-RPC-TE-NUW**	INCOMP	Appr.2 + Appr.8
CC-PV-TP-UW	**CC-RPC-TP-UW**	$\supsetneq$	Appr.7 + Corollary 3.10
CC-PV-TP-UW	**CC-RPC-TP-NUW**	INCOMP	Appr.2 + Appr.8
CC-PV-TP-UW	CC-PC-TE-UW	INCOMP	Appr.5 + Appr.8
CC-PV-TP-UW	CC-PC-TE-NUW	INCOMP	Appr.2 + Appr.8
CC-PV-TP-UW	CC-PC-TP-UW	$\supsetneq$	Appr.7 + Corollary 3.10
CC-PV-TP-UW	CC-PC-TP-NUW	INCOMP	Appr.2 + Appr.8
CC-PV-TP-UW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-PV-TP-UW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-PV-TP-UW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-PV-TP-UW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-PV-TP-UW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-PV-TP-UW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-PV-TP-UW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-PV-TP-UW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-PV-TP-UW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-PV-TP-UW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-PV-TP-UW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-PV-TP-UW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-PV-TP-NUW	**CC-RPC-TE-UW**	INCOMP	Appr.2 + Appr.5
CC-PV-TP-NUW	**CC-RPC-TE-NUW**	INCOMP	Appr.5 + Appr.8
CC-PV-TP-NUW	**CC-RPC-TP-UW**	$\subsetneq$	Appr.2 + Corollary 3.10
CC-PV-TP-NUW	**CC-RPC-TP-NUW**	$\subsetneq$	Appr.7 + Corollary 3.15
CC-PV-TP-NUW	CC-PC-TE-UW	INCOMP	Appr.2 + Appr.5
CC-PV-TP-NUW	CC-PC-TE-NUW	INCOMP	Appr.5 + Appr.8
CC-PV-TP-NUW	CC-PC-TP-UW	$\subsetneq$	Appr.2 + Corollary 3.10
CC-PV-TP-NUW	CC-PC-TP-NUW	$\subsetneq$	Appr.7 + Corollary 3.15
CC-PV-TP-NUW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-PV-TP-NUW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-PV-TP-NUW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-PV-TP-NUW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-PV-TP-NUW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-PV-TP-NUW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-PV-TP-NUW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-PV-TP-NUW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-PV-TP-NUW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-PV-TP-NUW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-PV-TP-NUW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-PV-TP-NUW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-RPC-TE-UW	**CC-RPC-TE-NUW**	$\subsetneq$	Appr.2
CC-RPC-TE-UW	**CC-RPC-TP-UW**	$\subsetneq$	Appr.5 + Corollary 3.10
CC-RPC-TE-UW	**CC-RPC-TP-NUW**	INCOMP	Appr.2 + Appr.5
CC-RPC-TE-UW	CC-PC-TE-UW	EQ	Corollary 3.18
CC-RPC-TE-UW	CC-PC-TE-NUW	$\subsetneq$	Appr.2
CC-RPC-TE-UW	CC-PC-TP-UW	$\subsetneq$	Appr.5 + Corollary 3.10
CC-RPC-TE-UW	CC-PC-TP-NUW	INCOMP	Appr.2 + Appr.5
CC-RPC-TE-UW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-RPC-TE-UW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-RPC-TE-UW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-RPC-TE-UW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-RPC-TE-UW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-RPC-TE-UW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-RPC-TE-UW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-RPC-TE-UW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-RPC-TE-UW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-RPC-TE-UW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-RPC-TE-UW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-RPC-TE-UW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-RPC-TE-NUW	**CC-RPC-TP-UW**	$\subsetneq$	Appr.2 + Corollary 3.10
CC-RPC-TE-NUW	**CC-RPC-TP-NUW**	$\subsetneq$	Appr.5 + Corollary 3.15
CC-RPC-TE-NUW	CC-PC-TE-UW	$\subsetneq$	Appr.2
CC-RPC-TE-NUW	CC-PC-TE-NUW	EQ	Theorem 3.17
CC-RPC-TE-NUW	CC-PC-TP-UW	$\subsetneq$	Appr.2 + Corollary 3.10
CC-RPC-TE-NUW	CC-PC-TP-NUW	$\subsetneq$	Appr.5 + Corollary 3.15
CC-RPC-TE-NUW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-RPC-TE-NUW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-RPC-TE-NUW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-RPC-TE-NUW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-RPC-TE-NUW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-RPC-TE-NUW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-RPC-TE-NUW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-RPC-TE-NUW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-RPC-TE-NUW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-RPC-TE-NUW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-RPC-TE-NUW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-RPC-TE-NUW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-RPC-TP-UW	**CC-RPC-TP-NUW**	$\subsetneq$	Appr.2
CC-RPC-TP-UW	CC-PC-TE-UW	$\subsetneq$	Appr.5 + Corollary 3.10
CC-RPC-TP-UW	CC-PC-TE-NUW	$\subsetneq$	Appr.2 + Corollary 3.10
CC-RPC-TP-UW	CC-PC-TP-UW	EQ	Corollary 3.11

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-RPC-TP-UW	CC-PC-TP-NUW	$\subsetneq$	Appr.2
CC-RPC-TP-UW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-RPC-TP-UW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-RPC-TP-UW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-RPC-TP-UW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-RPC-TP-UW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-RPC-TP-UW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-RPC-TP-UW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-RPC-TP-UW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-RPC-TP-UW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-RPC-TP-UW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-RPC-TP-UW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-RPC-TP-UW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-RPC-TP-NUW	CC-PC-TE-UW	INCOMP	Appr.2 + Appr.5
CC-RPC-TP-NUW	CC-PC-TE-NUW	$\subsetneq$	Appr.5 + Corollary 3.15
CC-RPC-TP-NUW	CC-PC-TP-UW	$\subsetneq$	Appr.2 + Corollary 3.15
CC-RPC-TP-NUW	CC-PC-TP-NUW	EQ	Corollary 3.11
CC-RPC-TP-NUW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-RPC-TP-NUW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-RPC-TP-NUW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-RPC-TP-NUW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-RPC-TP-NUW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-RPC-TP-NUW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-RPC-TP-NUW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-RPC-TP-NUW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-RPC-TP-NUW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-RPC-TP-NUW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-RPC-TP-NUW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-RPC-TP-NUW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-PC-TE-UW	CC-PC-TE-NUW	$\subsetneq$	Appr.2
CC-PC-TE-UW	CC-PC-TP-UW	$\subsetneq$	Appr.5 + Corollary 3.10
CC-PC-TE-UW	CC-PC-TP-NUW	INCOMP	Appr.2 + Appr.5
CC-PC-TE-UW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-PC-TE-UW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-PC-TE-UW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-PC-TE-UW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-PC-TE-UW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-PC-TE-UW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-PC-TE-UW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-PC-TE-UW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-PC-TE-UW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-PC-TE-UW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-PC-TE-UW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-PC-TE-UW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-PC-TE-NUW	CC-PC-TP-UW	$\subsetneq$	Appr.2 + Corollary 3.10
CC-PC-TE-NUW	CC-PC-TP-NUW	$\subsetneq$	Appr.5 + Corollary 3.15
CC-PC-TE-NUW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-PC-TE-NUW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-PC-TE-NUW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-PC-TE-NUW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-PC-TE-NUW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-PC-TE-NUW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-PC-TE-NUW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-PC-TE-NUW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-PC-TE-NUW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-PC-TE-NUW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-PC-TE-NUW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-PC-TE-NUW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-PC-TP-UW	CC-PC-TP-NUW	$\subsetneq$	Appr.2
CC-PC-TP-UW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-PC-TP-UW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-PC-TP-UW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-PC-TP-UW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-PC-TP-UW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-PC-TP-UW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-PC-TP-UW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-PC-TP-UW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
CC-PC-TP-UW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-PC-TP-UW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-PC-TP-UW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-PC-TP-UW	DC-PC-TP-NUW	INCOMP*	Appr.1
CC-PC-TP-NUW	DC-PV-TE-UW	INCOMP*	Appr.1
CC-PC-TP-NUW	**DC-PV-TE-NUW**	INCOMP*	Appr.1
CC-PC-TP-NUW	DC-PV-TP-UW	INCOMP*	Appr.1
CC-PC-TP-NUW	DC-PV-TP-NUW	INCOMP*	Appr.1
CC-PC-TP-NUW	DC-RPC-TE-UW	INCOMP*	Appr.1
CC-PC-TP-NUW	**DC-RPC-TE-NUW**	INCOMP*	Appr.1
CC-PC-TP-NUW	DC-RPC-TP-UW	INCOMP*	Appr.1
CC-PC-TP-NUW	**DC-RPC-TP-NUW**	INCOMP*	Appr.1
CC-PC-TP-NUW	DC-PC-TE-UW	INCOMP*	Appr.1
CC-PC-TP-NUW	DC-PC-TE-NUW	INCOMP*	Appr.1
CC-PC-TP-NUW	DC-PC-TP-UW	INCOMP*	Appr.1
CC-PC-TP-NUW	DC-PC-TP-NUW	INCOMP*	Appr.1
DC-PV-TE-UW	**DC-PV-TE-NUW**	EQ	Theorem 3.16
DC-PV-TE-UW	DC-PV-TP-UW	$\subsetneq$	Appr.3 + Theorem 3.19
DC-PV-TE-UW	DC-PV-TP-NUW	$\subsetneq$	Appr.2 + Theorem 3.19
DC-PV-TE-UW	DC-RPC-TE-UW	$\subsetneq$	Appr.3 + Corollary 3.11 + Corollary 3.21
DC-PV-TE-UW	**DC-RPC-TE-NUW**	$\subsetneq$	Appr.3 + Corollary 3.21
DC-PV-TE-UW	DC-RPC-TP-UW	$\subsetneq$	Appr.3 + Theorem 3.12 + Corollaries 3.11 and 3.21
DC-PV-TE-UW	**DC-RPC-TP-NUW**	$\subsetneq$	Appr.2 + Corollary 3.21 + Corollary 3.11
DC-PV-TE-UW	DC-PC-TE-UW	$\subsetneq$	Appr.3 + Corollary 3.11 + Corollary 3.21
DC-PV-TE-UW	DC-PC-TE-NUW	$\subsetneq$	Appr.3 + Corollary 3.11 + Corollary 3.21
DC-PV-TE-UW	DC-PC-TP-UW	$\subsetneq$	Appr.3 + Corollary 3.11 + Corollary 3.21
DC-PV-TE-UW	DC-PC-TP-NUW	$\subsetneq$	Appr.2 + Corollary 3.21 + Corollary 3.11
DC-PV-TE-NUW	DC-PV-TP-UW	$\subsetneq$	Appr.3 + Theorem 3.19
DC-PV-TE-NUW	DC-PV-TP-NUW	$\subsetneq$	Appr.2 + Theorem 3.19
DC-PV-TE-NUW	DC-RPC-TE-UW	$\subsetneq$	Appr.3 + Corollary 3.21 + Corollary 3.11
DC-PV-TE-NUW	**DC-RPC-TE-NUW**	$\subsetneq$	Appr.3 + Corollary 3.21
DC-PV-TE-NUW	DC-RPC-TP-UW	$\subsetneq$	Appr.3 + Corollaries 3.11 and 3.21 + Theorem 3.12
DC-PV-TE-NUW	**DC-RPC-TP-NUW**	$\subsetneq$	Appr.2 + Corollary 3.21 + Corollary 3.11
DC-PV-TE-NUW	DC-PC-TE-UW	$\subsetneq$	Appr.3 + Corollary 3.21 + Corollary 3.11
DC-PV-TE-NUW	DC-PC-TE-NUW	$\subsetneq$	Appr.3 + Corollary 3.21 + Corollary 3.11
DC-PV-TE-NUW	DC-PC-TP-UW	$\subsetneq$	Appr.3 + Corollary 3.21 + Corollary 3.11
DC-PV-TE-NUW	DC-PC-TP-NUW	$\subsetneq$	Appr.2 + Corollary 3.21 + Corollary 3.11
DC-PV-TP-UW	DC-PV-TP-NUW	$\subsetneq$	Appr.2
DC-PV-TP-UW	DC-RPC-TE-UW	$\subsetneq$	Appr.4 + Theorem 3.20 + Corollary 3.11
DC-PV-TP-UW	**DC-RPC-TE-NUW**	$\subsetneq$	Appr.4 + Theorem 3.20
DC-PV-TP-UW	DC-RPC-TP-UW	$\subsetneq$	Appr.4 + Theorems 3.12 and 3.20 + Corollary 3.11
DC-PV-TP-UW	**DC-RPC-TP-NUW**	$\subsetneq$	Appr.2 + Corollary 3.15
DC-PV-TP-UW	DC-PC-TE-UW	$\subsetneq$	Appr.4 + Theorem 3.20 + Corollary 3.11
DC-PV-TP-UW	DC-PC-TE-NUW	$\subsetneq$	Appr.4 + Theorem 3.20 + Corollary 3.11
DC-PV-TP-UW	DC-PC-TP-UW	$\subsetneq$	Appr.4 + Theorem 3.20 + Corollary 3.11
DC-PV-TP-UW	DC-PC-TP-NUW	$\subsetneq$	Appr.2 + Corollary 3.15
DC-PV-TP-NUW	DC-RPC-TE-UW	INCOMP	Appr.2 + Appr.8
DC-PV-TP-NUW	**DC-RPC-TE-NUW**	INCOMP	Appr.2 + Appr.8
DC-PV-TP-NUW	DC-RPC-TP-UW	INCOMP	Appr.2 + Appr.8
DC-PV-TP-NUW	**DC-RPC-TP-NUW**	$\subsetneq$	Appr.4 + Theorem 3.1 + Theorem 3.20
DC-PV-TP-NUW	DC-PC-TE-UW	INCOMP	Appr.2 + Appr.8
DC-PV-TP-NUW	DC-PC-TE-NUW	INCOMP	Appr.2 + Appr.8
DC-PV-TP-NUW	DC-PC-TP-UW	INCOMP	Appr.2 + Appr.8
DC-PV-TP-NUW	DC-PC-TP-NUW	$\subsetneq$	Appr.4 + Theorem 3.1 + Theorem 3.20
DC-RPC-TE-UW	**DC-RPC-TE-NUW**	EQ	Corollary 3.11
DC-RPC-TE-UW	DC-RPC-TP-UW	EQ	Corollary 3.11 + Corollary 3.11
DC-RPC-TE-UW	**DC-RPC-TP-NUW**	$\subsetneq$	Appr.2 + Theorem 3.1
DC-RPC-TE-UW	DC-PC-TE-UW	EQ	Corollary 3.11
DC-RPC-TE-UW	DC-PC-TE-NUW	EQ	Corollary 3.11
DC-RPC-TE-UW	DC-PC-TP-UW	EQ	Corollary 3.11 + Corollary 3.11
DC-RPC-TE-UW	DC-PC-TP-NUW	$\subsetneq$	Appr.2 + Theorem 3.1
DC-RPC-TE-NUW	DC-RPC-TP-UW	EQ	Corollary 3.11
DC-RPC-TE-NUW	**DC-RPC-TP-NUW**	$\subsetneq$	Appr.2 + Theorem 3.1
DC-RPC-TE-NUW	DC-PC-TE-UW	EQ	Corollary 3.11
DC-RPC-TE-NUW	DC-PC-TE-NUW	EQ	Corollary 3.11
DC-RPC-TE-NUW	DC-PC-TP-UW	EQ	Corollary 3.11
DC-RPC-TE-NUW	DC-PC-TP-NUW	$\subsetneq$	Appr.2 + Theorem 3.1

$\mathcal{T}$	$\mathcal{T}'$	Classification	Justification(s)
DC-RPC-TP-UW	**DC-RPC-TP-NUW**	$\subsetneq$	Appr.2
DC-RPC-TP-UW	DC-PC-TE-UW	EQ	Corollary 3.11 + Theorem 3.12
DC-RPC-TP-UW	DC-PC-TE-NUW	EQ	Corollary 3.11 + Theorem 3.12
DC-RPC-TP-UW	DC-PC-TP-UW	EQ	Theorem 3.12
DC-RPC-TP-UW	DC-PC-TP-NUW	$\subsetneq \cap \supsetneq$	Appr.2 + Theorems 3.1 and 3.12 + Corollary 3.11
DC-RPC-TP-NUW	DC-PC-TE-UW	$\subsetneq \cap \supsetneq$	Appr.2 + Theorem 3.1 + Corollary 3.11
DC-RPC-TP-NUW	DC-PC-TE-NUW	$\subsetneq \cap \supsetneq$	Appr.2 + Theorem 3.1 + Corollary 3.11
DC-RPC-TP-NUW	DC-PC-TP-UW	$\subsetneq \cap \supsetneq$	Appr.2 + Theorem 3.1 + Corollary 3.11
DC-RPC-TP-NUW	DC-PC-TP-NUW	EQ	Corollary 3.11
DC-PC-TE-UW	DC-PC-TE-NUW	EQ	Corollary 3.11
DC-PC-TE-UW	DC-PC-TP-UW	EQ	Corollary 3.11 + Corollary 3.11
DC-PC-TE-UW	DC-PC-TP-NUW	$\subsetneq$	Appr.2 + Theorem 3.1 + Corollary 3.11
DC-PC-TE-NUW	DC-PC-TP-UW	EQ	Corollary 3.11 + Corollary 3.11
DC-PC-TE-NUW	DC-PC-TP-NUW	$\supsetneq$	Appr.2 + Theorem 3.1 + Corollary 3.11
DC-PC-TP-UW	DC-PC-TP-NUW	$\subsetneq \cap \supsetneq$	Appr.2
CC-AC-UW	CC-AC-NUW	$\subsetneq$	Appr.11
CC-AC-UW	DC-AC-UW	INCOMP	Appr.13 + Appr.11
CC-AC-UW	DC-AC-NUW	INCOMP	Appr.13 + Appr.12
CC-AC-NUW	DC-AC-UW	INCOMP	Appr.13 + Appr.12
CC-AC-NUW	DC-AC-NUW	INCOMP	Appr.13 + Appr.12
DC-AC-UW	DC-AC-NUW	$\subsetneq$	Appr.11
CC-DC-UW	CC-DC-NUW	$\subsetneq$	Appr.14
CC-DC-UW	CC-DV-UW	INCOMP	Appr.15 + Appr.19
CC-DC-UW	CC-DV-NUW	INCOMP	Appr.20 + Appr.14
CC-DC-UW	DC-DC-UW	INCOMP	Appr.23 + Appr.14
CC-DC-UW	DC-DC-NUW	INCOMP*	Appr.10
CC-DC-UW	DC-DV-UW	INCOMP	Appr.23 + Appr.14
CC-DC-UW	DC-DV-NUW	INCOMP*	Appr.10
CC-DC-NUW	CC-DV-UW	INCOMP	Appr.14 + Appr.21
CC-DC-NUW	CC-DV-NUW	INCOMP	Appr.16 + Appr.20
CC-DC-NUW	DC-DC-UW	INCOMP	Appr.23 + Appr.16
CC-DC-NUW	DC-DC-NUW	INCOMP*	Appr.29
CC-DC-NUW	DC-DV-UW	INCOMP*	Appr.29
CC-DC-NUW	DC-DV-NUW	INCOMP*	Appr.29
CC-DV-UW	CC-DV-NUW	$\subsetneq$	Appr.14
CC-DV-UW	DC-DC-UW	INCOMP	Appr.23 + Appr.14
CC-DV-UW	DC-DC-NUW	INCOMP	Appr.23 + Appr.16
CC-DV-UW	DC-DV-UW	INCOMP	Appr.23 + Appr.14
CC-DV-UW	DC-DV-NUW	INCOMP	Appr.23 + Appr.16
CC-DV-NUW	DC-DC-UW	INCOMP	Appr.23 + Appr.17
CC-DV-NUW	DC-DC-NUW	INCOMP*	Appr.29
CC-DV-NUW	DC-DV-UW	INCOMP	Appr.23 + Appr.17
CC-DV-NUW	DC-DV-NUW	INCOMP*	Appr.29
DC-DC-UW	DC-DC-NUW	$\subsetneq$	Appr.14
DC-DC-UW	DC-DV-UW	$\subsetneq \cap \supsetneq$	Theorem 3.7 + Appr.18
DC-DC-UW	DC-DV-NUW	INCOMP	Appr.14 + Appr.22
DC-DC-NUW	DC-DV-UW	$\subsetneq \cap \supsetneq$	Theorem 3.7 + Appr.18
DC-DC-NUW	DC-DV-NUW	$\subsetneq \cap \supsetneq$	Theorem 3.8 + Appr.19
DC-DV-UW	DC-DV-NUW	$\subsetneq$	Appr.14
CC-AV-UW	CC-AV-NUW	$\subsetneq$	Appr.24
CC-AV-UW	DC-AV-UW	INCOMP	Appr.25 + Appr.24
CC-AV-UW	DC-AV-NUW	INCOMP	Appr.25 + Appr.24
CC-AV-NUW	DC-AV-UW	INCOMP	Appr.25 + Appr.26
CC-AV-NUW	DC-AV-NUW	INCOMP	Appr.25 + Appr.26
DC-AV-UW	DC-AV-NUW	$\subsetneq$	Appr.27
CC-UAC-UW	CC-UAC-NUW	$\subsetneq$	Appr.30
CC-UAC-UW	DC-UAC-UW	INCOMP	Appr.28 + Appr.24
CC-UAC-UW	DC-UAC-NUW	INCOMP	Appr.28 + Appr.24
CC-UAC-NUW	DC-UAC-UW	INCOMP	Appr.28 + Appr.24
CC-UAC-NUW	DC-UAC-NUW	INCOMP	Appr.28 + Appr.24
DC-UAC-UW	DC-UAC-NUW	$\subsetneq$	Appr.30

References

Bartholdi, III, J., Tovey, C., & Trick, M. (1989a). The computational difficulty of manipu-

lating an election. *Social Choice and Welfare*, *6*(3), 227–241.

Bartholdi, III, J., Tovey, C., & Trick, M. (1989b). Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, *6*(2), 157–165.

Bartholdi, III, J., Tovey, C., & Trick, M. (1992). How hard is it to control an election?. *Mathematical and Computer Modeling*, *16*(8–9), 27–40.

Baumeister, D., Erdélyi, G., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2010). Computational aspects of approval voting. In Laslier, J., & Sanver, M. (Eds.), *Handbook on Approval Voting*, pp. 199–251. Springer.

Carleton, B., Chavrimootoo, M., Hemaspaandra, L., Narváez, D., Taliancich, C., & Welles, H. (2022). Search versus search for collapsing electoral control types. Tech. rep. arXiv:2207.03049 [cs.GT], Computing Research Repository, arXiv.org/corr/. Revised, February 2024.

Carleton, B., Chavrimootoo, M., Hemaspaandra, L., Narváez, D., Taliancich, C., & Welles, H. (2023a). Search versus search for collapsing electoral control types (extended abstract). In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, pp. 2682–2684. International Foundation for Autonomous Agents and Multiagent Systems.

Carleton, B., Chavrimootoo, M., Hemaspaandra, L., Narváez, D., Taliancich, C., & Welles, H. (2023b). Separating and collapsing electoral control types. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, pp. 1743–1751. International Foundation for Autonomous Agents and Multiagent Systems.

Carleton, B., Chavrimootoo, M., Hemaspaandra, L., Narváez, D., Taliancich, C., & Welles, H. (2024). Search versus search for collapsing electoral control types. In *Proceedings of the 21st European Conference on Multi-Agent Systems*. To appear.

Chernoff, H. (1954). Rational selection of decision functions. *Econometrica*, *22*(4), 422–443.

Erdélyi, G., Fellows, M., Rothe, J., & Schend, L. (2015). Control complexity in Bucklin and fallback voting: A theoretical analysis. *Journal of Computer and System Sciences*, *81*(4), 632–660.

Erdélyi, G., Nowak, M., & Rothe, J. (2009). Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Mathematical Logic Quarterly*, *55*(4), 425–443.

Erdélyi, G., Piras, L., & Rothe, J. (2011). The complexity of voter partition in Bucklin and fallback voting: Solving three open problems. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pp. 837–844. International Foundation for Autonomous Agents and Multiagent Systems.

Erdélyi, G., Reger, C., & Yang, Y. (2019). Towards completing the puzzle: Solving open problems for control in elections. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pp. 846–854. International Foundation for Autonomous Agents and Multiagent Systems.

Erdélyi, G., & Rothe, J. (2010). Control complexity in fallback voting. In *Proceedings the 16th Australasian Theory Symposium*, pp. 39–48. Australian Computer Society.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2007). Llull and Copeland voting broadly resist bribery and control. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 724–730. AAAI Press.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2009). Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, *35*, 275–341.

Faliszewski, P., & Rothe, J. (2016). Control and bribery in voting. In Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. (Eds.), *Handbook of Computational Social Choice*, pp. 146–168. Cambridge University Press.

Gupta, S., Roy, S., Saurabh, S., & Zehavi, M. (2022). Resolute control: Forbidding candidates from winning an election is hard. *Theoretical Computer Science*, *915*, 74–89.

Hemaspaandra, E., Hemaspaandra, L., & Menton, C. (2020). Search versus decision for election manipulation problems. *ACM Transactions on Computation Theory*, *12*(#1, Article 3), 1–42.

Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2007). Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, *171*(5–6), 255–285.

Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2009). Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, *55*(4), 397–424.

Hemaspaandra, L., Lavaee, R., & Menton, C. (2016). Schulze and ranked-pairs voting are fixed-parameter tractable to bribe, manipulate, and control. *Annals of Mathematics and Artificial Intelligence*, *77*(3–4), 191–223.

Lin, A. (2012). *Solving Hard Problems in Election Systems*. Ph.D. thesis, Rochester Institute of Technology, Rochester, NY.

Maushagen, C., & Rothe, J. (2018). Complexity of control by partitioning veto elections and of control by adding candidates to plurality elections. *Annals of Mathematics and Artificial Intelligence*, *82*(4), 219–244.

Menton, C. (2013). Normalized range voting broadly resists control. *Theory of Computing Systems*, *53*(4), 507–531.

Menton, C., & Singh, P. (2013). Control complexity of Schulze voting. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 286–292. AAAI Press.

Neveling, M., & Rothe, J. (2017). Solving seven open problems of offline and online control in Borda elections. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 3029–3035. AAAI Press.

Parkes, D., & Xia, L. (2012). A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp. 1429–1435. AAAI Press.

Sen, A. (1971). Choice functions and revealed preference. *The Review of Economic Studies*, *38*(3), 307–317.

Yang, Y., & Wang, J. (2017). Anyone but them: The complexity challenge for a resolute election controller. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1133–1141. International Foundation for Autonomous Agents and Multiagent Systems.