Solving Tournaments by Binary Trees

Paul Harrenstein

Ludwig-Maximilians-Universität, München

Computational Social Choice, 12th December 2007

Introduction

- Determining a winner by an fixed schedule or *agenda*
- Example: Knock-out tournament
- Importance of a drawing
- Majority voting
- Strategic voting behaviour

Overview

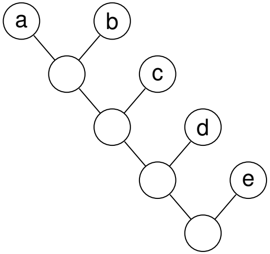
- Majority dominance graphs
- Social choice correspondences vs. social choice functions
- Restriction to tournaments
- Binary voting procedures
- Solving tournaments using binary voting trees

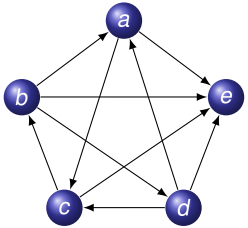
Binary Voting Procedures

- *Amendment procedure*: Two alternatives are paired for majority vote, the defeated proposal being eliminated as a possible decision, the surviving proposal is then paired with a third alternative at the second vote, and so forth. The proposal that survives the final vote is the decision.
- *Successive procedure*: The first proposal is voted up or down on a majority basis: if voted up, it is the decision, otherwise, the second proposal is voted up or voted down and so forth. If the first $m - 1$ proposals are voted down, the remaining proposal is the decision.

Sincere Voting under Amendment Procedure

Assumption: At each vote, each player votes for his most preferred alternative.





Sincere Voting under Successive Procedure

Assumption: At each vote, each player votes for a proposal if it is the proposal he most prefers among those that have not yet been voted down.

Fact: Under successive procedure, the sincere voting decision may not elect the Condorcet winner.

Proof: Voting order (a, b, c) and preference profile:

1	2	3
a	b	c
b	a	a
c	c	b

a is the Condorcet winner, but eliminated in the first round.

Sincere Voting under Successive Procedure

Assumption: At each vote, each player votes for a proposal if it is the proposal he most prefers among those that have not yet been voted down.

Fact: Under successive procedure, the sincere voting decision may not elect the Condorcet winner.

Proof: Voting order (a, b, c) and preference profile:

1	2	3
a	b	c
b	a	a
c	c	b

a is the Condorcet winner, but eliminated in the first round.

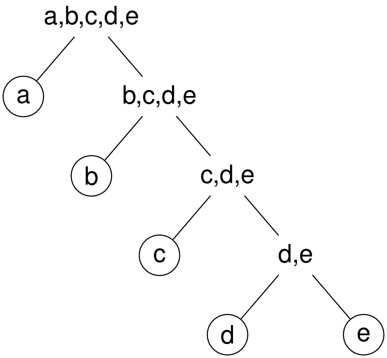
Sophisticated Voting under Successive Procedure

Example: Voting order (a, b, c) and preference profile:

1	2	3
a	b	c
b	a	a
c	c	b

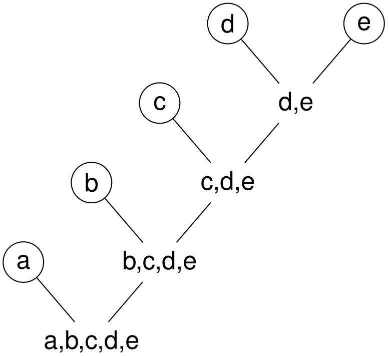
Sophisticated Voting under the Successive Procedure

Voting order (a, b, c, d, e)

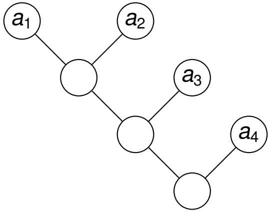


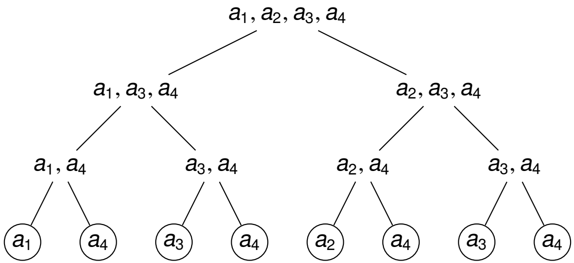
Sophisticated Voting under the Successive Procedure

Voting order (a, b, c, d, e)



Sophisticated Voting under the Amendment Procedure





Voting Trees

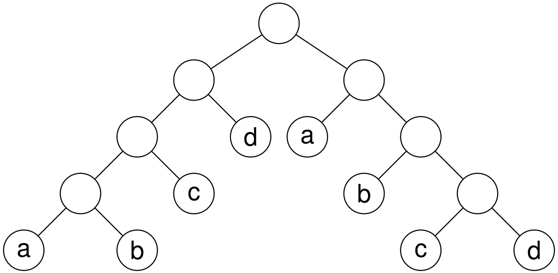
Definition: A *binary voting tree* Γ *on* A is a binary tree in which each terminal node is associated with an alternative from A, *each alternative in* A *occurring at least once.*

Voting Trees

Definition: A *binary voting tree* Γ *on* A is a triple (V, R, ϕ) in which:

- (V, R) is a binary tree on A,
- $\phi\colon Z \to A$ surjective, where Z is the set of terminal nodes.

Voting Trees



$$\Gamma = ((((a,b),c),d),(a,(b,(c,d))))$$

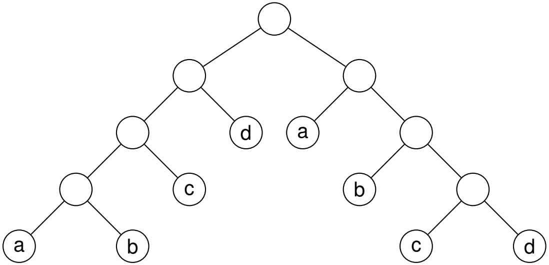
The Reduction Procedure

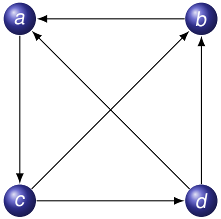
Definition: Let T be a tournament and $\Gamma = (V, R, \phi)$ a binary voting tree on A. Extend ϕ to $\phi^*\colon V \to A$, such that for all $v \in V$:

$$\phi^*(v) = \begin{cases} \phi(v) & \text{if } v \in Z \\ \phi^*(v') & \text{if } v', v'' \text{ are the children of } v \text{ and } \phi^*(v') \neq \phi^*(v'') \text{ implies } \phi^*(v') \succ \end{cases}$$

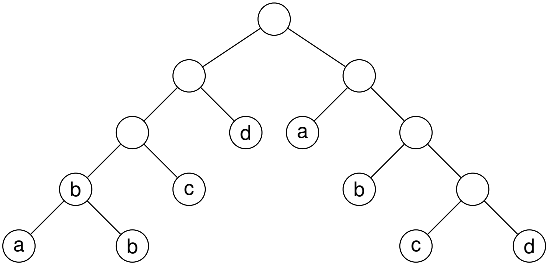
Then set $f^\Gamma(T) = \phi^*(v_0)$, where v_0 is the root of Γ.

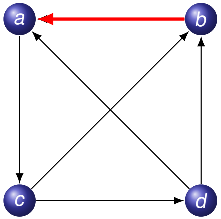
The Reduction Procedure



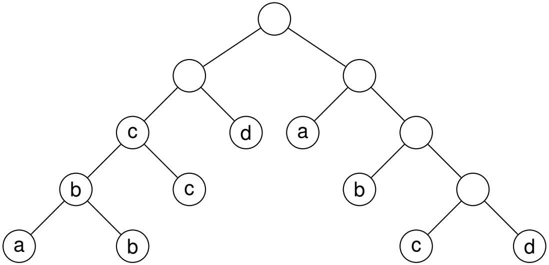


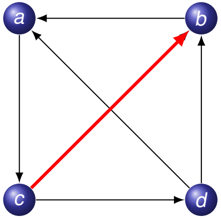
The Reduction Procedure



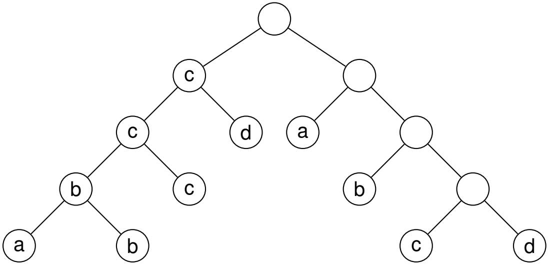


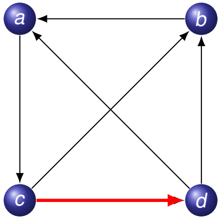
The Reduction Procedure



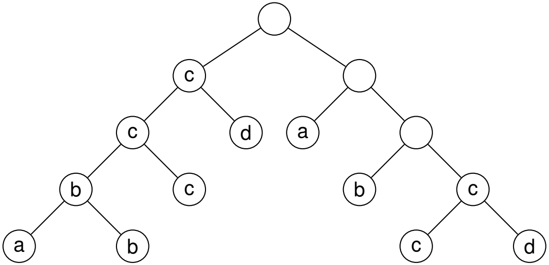


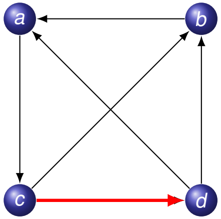
The Reduction Procedure



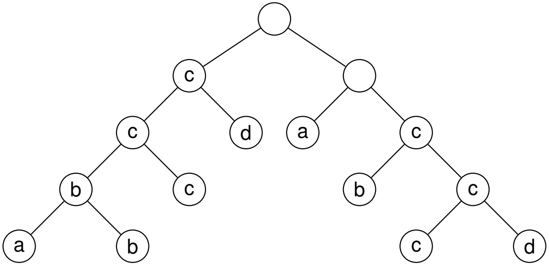


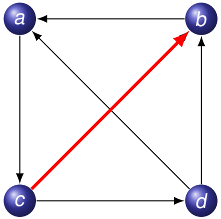
The Reduction Procedure



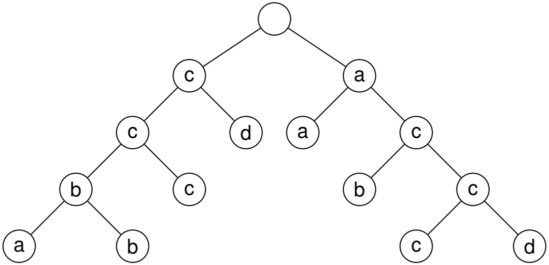


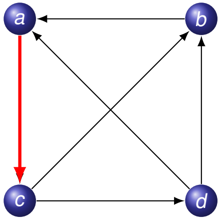
The Reduction Procedure



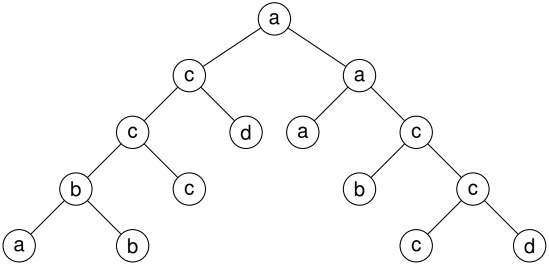


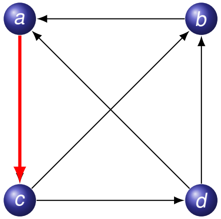
The Reduction Procedure



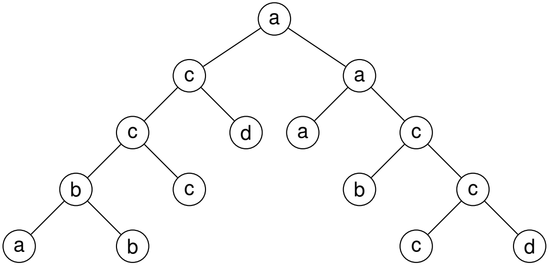


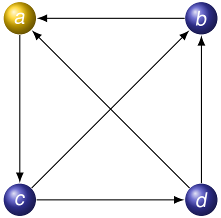
The Reduction Procedure





The Reduction Procedure





Characterization of the Good Set

Theorem: Let Γ be a binary voting tree and T a tournament on A. Then, $f^{\Gamma}(T) \in \mathrm{GO}(T)$.

Proof: Paint all outcomes of $\mathrm{GO}(T)$ red and all those in $A \setminus \mathrm{GO}(T)$ blue. By definition of the Good set $a > b$, for every $a \in \mathrm{GO}(T)$ and every $b \in A \setminus \mathrm{GO}(T)$, i.e., every red alternative beats any blue alternative. Since, in the binary voting tree there is at least one red terminal node—as every outcome appears at least once—the final winner must be red. $\square$

Characterization of the Good Set

Theorem: Let Γ be a binary voting tree and T a tournament on A. Then, $f^{\Gamma}(T) \in \mathrm{GO}(T)$.

Proof: Paint all outcomes of $\mathrm{GO}(T)$ red and all those in $A \setminus \mathrm{GO}(T)$ blue. By definition of the Good set $a \succ b$, for every $a \in \mathrm{GO}(T)$ and every $b \in A \setminus \mathrm{GO}(T)$, i.e., every red alternative beats any blue alternative. Since, in the binary voting tree there is at least one red terminal node—as every outcome appears at least once—the final winner must be red. $\square$

Agenda Paradox

See whiteboard.

Computability by Binary Trees

Notation: Be π be a permutation of A and $\Gamma = (V, R, \phi)$, then: $\pi(\Gamma) = (V, R, \pi \circ \phi)$.

Definition: For A a set of alternatives, T a tournament and Γ a binary trees on A define the social choice correspondence $S^\Gamma \colon \mathscr{T}(A) \to 2^A$ as:

$$S^\Gamma(T) = \{f^{\pi(\Gamma)}(T) \colon \pi \text{ a permutation of } A\}.$$

Definition: A social choice correspondence S is *computable by binary trees* if for every set of alternatives A there is a binary voting tree Γ such that for all tournaments $T \in \mathscr{T}(A)$:

$$S(T) = S^\Gamma(T).$$

Computability by Binary Trees

Notation: Be π be a permutation of A and $\Gamma = (V, R, \phi)$, then:
$\pi(\Gamma) = (V, R, \pi \circ \phi)$.

Definition: For A a set of alternatives, T a tournament and Γ a binary trees on A define the social choice correspondence $S^\Gamma \colon \mathscr{T}(A) \to 2^A$ as:

$$S^\Gamma(T) = \{f^{\pi(\Gamma)}(T) \colon \pi \text{ a permutation of } A\}.$$

Definition: A social choice correspondence S is *computable by binary trees* if for every set of alternatives A there is a binary voting tree Γ such that for all tournaments $T \in \mathscr{T}(A)$:

$$S(T) = S^\Gamma(T).$$

Computability by Binary Trees

Notation: Be π be a permutation of A and $\Gamma = (V, R, \phi)$, then: $\pi(\Gamma) = (V, R, \pi \circ \phi)$.

Definition: For A a set of alternatives, T a tournament and Γ a binary trees on A define the social choice correspondence $S^\Gamma \colon \mathscr{T}(A) \to 2^A$ as:

$$S^\Gamma(T) = \{f^{\pi(\Gamma)}(T) \colon \pi \text{ a permutation of } A\}.$$

Definition: A social choice correspondence S is *computable by binary trees* if for every set of alternatives A there is a binary voting tree Γ such that for all tournaments $T \in \mathscr{T}(A)$:

$$S(T) = S^\Gamma(T).$$

The Simple Agenda

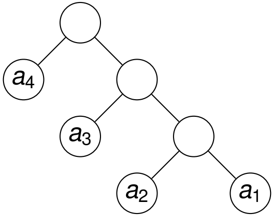
Definition: For each sequence of alternatives $a_1, \ldots, a_n$ define the *simple agenda* $\Gamma_{a_1,\ldots,a_n}$ as follows:

- $\Gamma_{a_1} = (a_1)$,
- $\Gamma_{a_1,\ldots,a_{n+1}} = (a_{n+1}, \Gamma_{a_1,\ldots,a_n})$.

The Simple Agenda

Definition: For each sequence of alternatives $a_1, \ldots, a_n$ define the *simple agenda* $\Gamma_{a_1,\ldots,a_n}$ as follows:

- $\Gamma_{a_1} = (a_1)$,
- $\Gamma_{a_1,\ldots,a_{n+1}} = (a_{n+1}, \Gamma_{a_1,\ldots,a_n})$.



Γ_{a_1,a_2,a_3,a_4}

The Simple Agenda

Definition: For each sequence of alternatives $a_1, \ldots, a_n$ define the *simple agenda* $\Gamma_{a_1,\ldots,a_n}$ as follows:

- $\Gamma_{a_1} = (a_1)$,
- $\Gamma_{a_1,\ldots,a_{n+1}} = (a_{n+1}, \Gamma_{a_1,\ldots,a_n})$.

Lemma: For any tournament, $x \in \mathrm{GO}(T)$ implies there is a complete path beginning with x

Theorem: Let $A = \{a_1, \ldots, a_n\}$ be a set of alternatives, $\Gamma_{a_1,\ldots,a_m}$ be a simple agenda on A. Then, for all tournaments $T \in \mathscr{T}(A)$:

$$\mathrm{GO}(T) = S^{\Gamma_{a_1,\ldots,a_m}}(T).$$

Pareto Efficiency

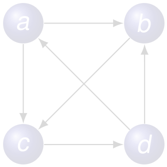
Definition: An alternative a *Pareto dominates* another alternative b whenever:

- $a \succsim_i b$, for all $i \in N$
- $a \succ_i b$, for some $i \in N$

An alternative is *Pareto efficient* if it is Pareto dominated by no alternative.

Fact: The Good set may contain Pareto dominated alternatives.

1	2	3
a	d	c
b	a	d
c	b	a
d	c	b


$$\begin{tikzcd} a \arrow[r] \arrow[d] & b \arrow[dl] \\ c \arrow[r] & d \arrow[u] \arrow[ul] \end{tikzcd}$$

Pareto Efficiency

Definition: An alternative *a* *Pareto dominates* another alternative *b* whenever:

- $a \succsim_i b$, for all $i \in N$
- $a \succ_i b$, for some $i \in N$

An alternative is *Pareto efficient* if it is Pareto dominated by no alternative.

Fact: The Good set may contain Pareto dominated alternatives.

1	2	3
a	d	c
b	a	d
c	b	a
d	c	b

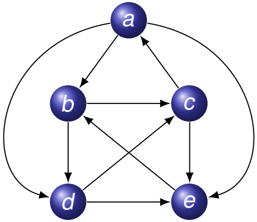
$$\begin{tikzcd}
a \arrow[r] \arrow[d] & b \arrow[dl] \\
c \arrow[r] & d \arrow[u] \arrow[ul]
\end{tikzcd}$$

Monotonicity

Definition: A social choice correspondence S is *monotonic* if for any $a \in A$ and all tournaments T and T' on A such that:

- $x \succ y$ iff $x \succ' y$, for all $x, y \in A$ with $x \neq a$ and $y \neq a$, and
- $a \succ x$ implies $a \succ' x$, for all $x \in A$,

we have $a \in S(T)$ implies $a \in S(T)$.

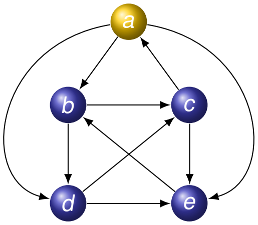


Monotonicity

Definition: A social choice correspondence S is *monotonic* if for any $a \in A$ and all tournaments T and T' on A such that:

- $x \succ y$ iff $x \succ' y$, for all $x, y \in A$ with $x \neq a$ and $y \neq a$, and
- $a \succ x$ implies $a \succ' x$, for all $x \in A$,

we have $a \in S(T)$ implies $a \in S(T)$.

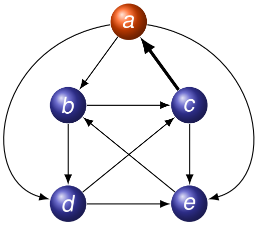


Monotonicity

Definition: A social choice correspondence S is *monotonic* if for any $a \in A$ and all tournaments T and T' on A such that:

- $x \succ y$ iff $x \succ' y$, for all $x, y \in A$ with $x \neq a$ and $y \neq a$, and
- $a \succ x$ implies $a \succ' x$, for all $x \in A$,

we have $a \in S(T)$ implies $a \in S(T)$.

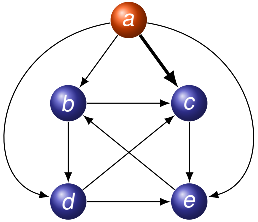


Monotonicity

Definition: A social choice correspondence S is *monotonic* if for any $a \in A$ and all tournaments T and T' on A such that:

- $x \succ y$ iff $x \succ' y$, for all $x, y \in A$ with $x \neq a$ and $y \neq a$, and
- $a \succ x$ implies $a \succ' x$, for all $x \in A$,

we have $a \in S(T)$ implies $a \in S(T)$.

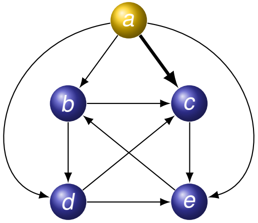


Monotonicity

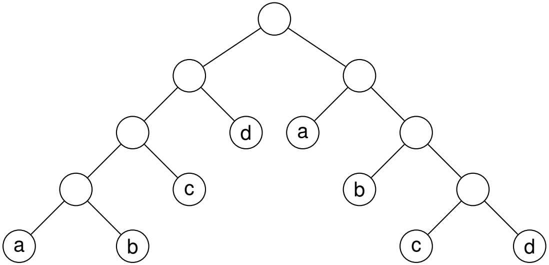
Definition: A social choice correspondence S is *monotonic* if for any $a \in A$ and all tournaments T and T' on A such that:

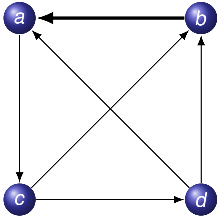
- $x \succ y$ iff $x \succ' y$, for all $x, y \in A$ with $x \neq a$ and $y \neq a$, and
- $a \succ x$ implies $a \succ' x$, for all $x \in A$,

we have $a \in S(T)$ implies $a \in S(T)$.

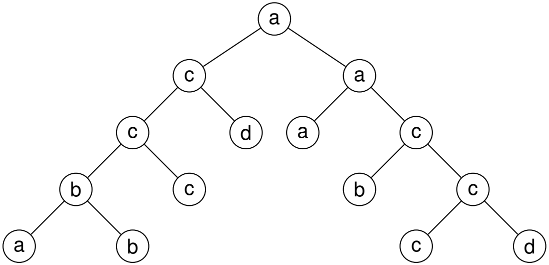


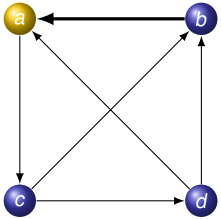
Failure of Monotonicity in Binary Voting Trees



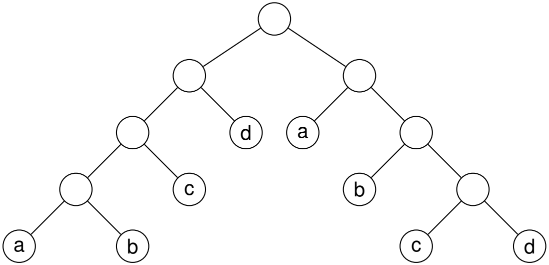


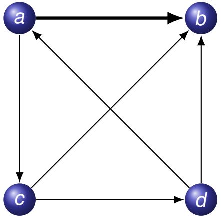
Failure of Monotonicity in Binary Voting Trees



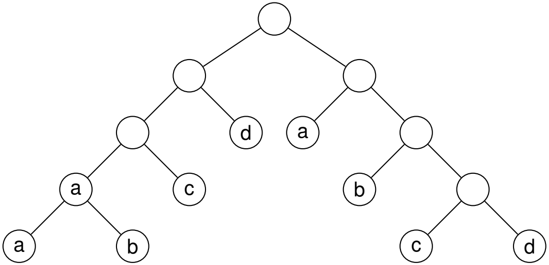


Failure of Monotonicity in Binary Voting Trees



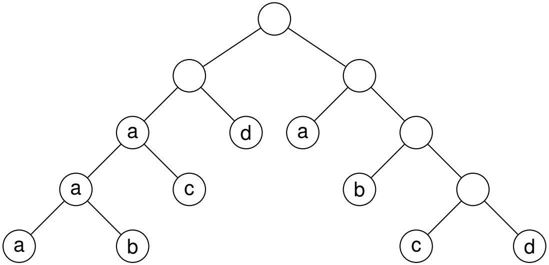


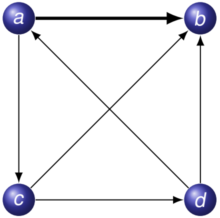
Failure of Monotonicity in Binary Voting Trees



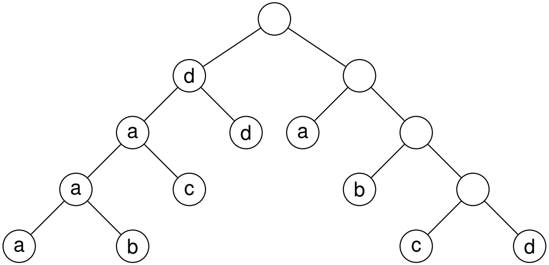
$$\begin{tikzcd}
a \arrow[r, line width=2pt] \arrow[d] & b \\
c \arrow[r] \arrow[ur] & d \arrow[u] \arrow[ul]
\end{tikzcd}$$

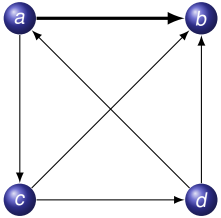
Failure of Monotonicity in Binary Voting Trees



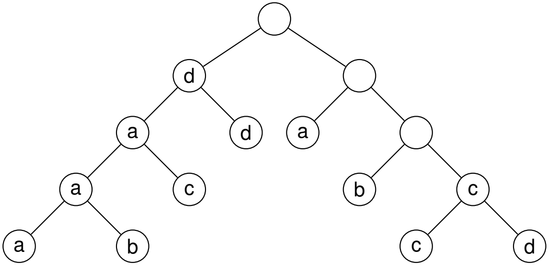


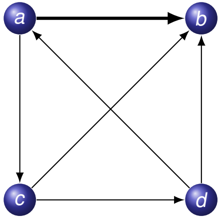
Failure of Monotonicity in Binary Voting Trees



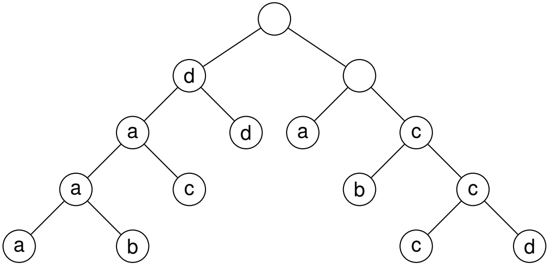


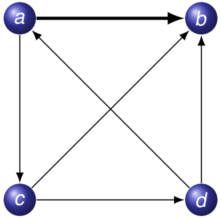
Failure of Monotonicity in Binary Voting Trees



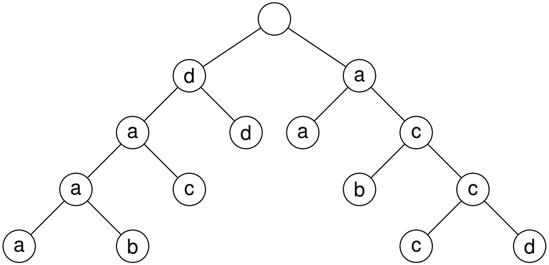


Failure of Monotonicity in Binary Voting Trees



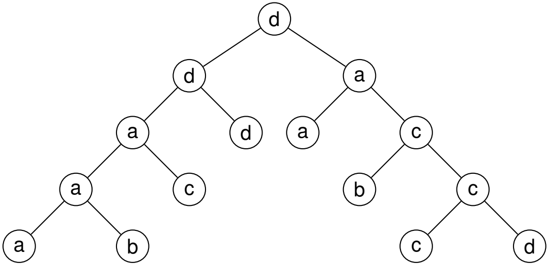


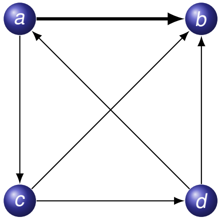
Failure of Monotonicity in Binary Voting Trees



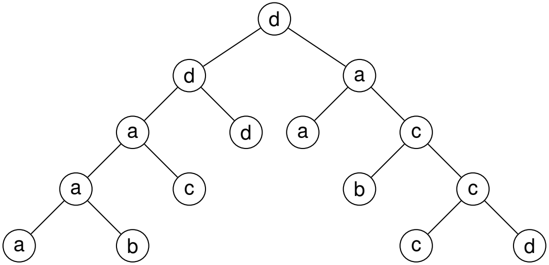
$$\begin{tikzcd}
a \arrow[r, thick] \arrow[d] & b \\
c \arrow[r] \arrow[ur] & d \arrow[u] \arrow[ul]
\end{tikzcd}$$

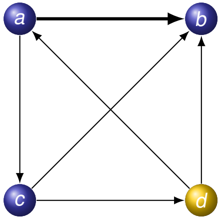
Failure of Monotonicity in Binary Voting Trees





Failure of Monotonicity in Binary Voting Trees



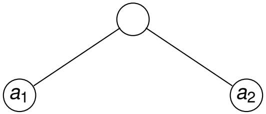


The Sophisticated Agenda

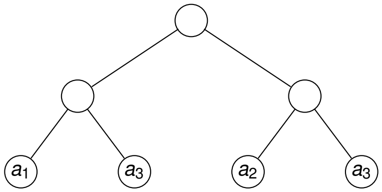
Definition: For each sequence of alternatives $a_1, \ldots, a_n$ define the *sophisticated agenda* $\Gamma^*_{a_1,\ldots,a_n}$ as follows:

- $\Gamma^*_{a_1} = (a_1)$,
- $\Gamma^*_{a_1,\ldots,a_{n+1}} = (\Gamma^*_{a_1,a_3,\ldots,a_{n+1}}, \Gamma^*_{a_2,a_3,\ldots,a_{n+1}})$.

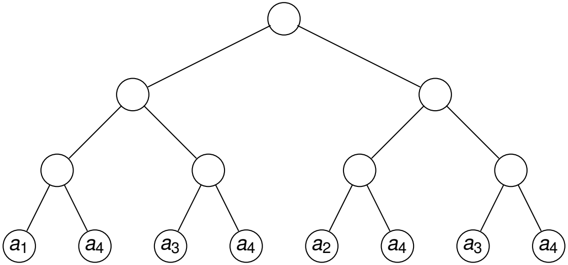
Sophisticated Agenda



Sophisticated Agenda



Sophisticated Agenda



The Banks Set and the Sophisticated Agenda

Definition: The *Banks set* $BA(T)$ of a tournament T on A consists of the maximal elements of the maximal transitive (cycle-free) subsets X of A.

Theorem: Let $A = \{a_1, \ldots, a_n\}$ be a set of alternatives, $\Gamma^*_{a_1,\ldots,a_m}$ be a sophisticated agenda on A. Then, for all tournaments $T \in \mathscr{T}(A)$:

$$\mathrm{BA}(T) = S^{\Gamma^*_{a_1,\ldots,a_m}}(T).$$

Proof: For T and each sequence $a_1, \ldots, a_m$ in A define the *sophisticated sequence* $\sigma_T = (\sigma_T(a_1), \ldots, \sigma_T(a_m))$ as:

$$\sigma(a_i) = \begin{cases} a_1 & \text{if } i = 1, \\ a_i & \text{if } a_i > \sigma_T(a_j) \text{ for all } 1 \leq j < i, \\ \sigma_T(a_{i-1}) & \text{otherwise.} \end{cases}$$

Then, by induction $f^{\Gamma^*_{a_m,\ldots,a_1}}(T) = \sigma_T(a_m)$. Moreover, $a \in BA(T)$ if and only if $a = \sigma_T(a_1, \ldots, a_m)$ for some sequence $a_1, \ldots, a_m$. □

The Banks Set and the Sophisticated Agenda

Definition: The *Banks set* $BA(T)$ of a tournament T on A consists of the maximal elements of the maximal transitive (cycle-free) subsets X of A.

Theorem: Let $A = \{a_1, \dots, a_n\}$ be a set of alternatives, $\Gamma^*_{a_1,\dots,a_m}$ be a sophisticated agenda on A. Then, for all tournaments $T \in \mathscr{T}(A)$:

$$\mathrm{BA}(T) = S^{\Gamma^*_{a_1,\dots,a_m}}(T).$$

Proof: For T and each sequence $a_1, \dots, a_m$ in A define the *sophisticated sequence* $\sigma_T = (\sigma_T(a_1), \dots, \sigma_T(a_m))$ as:

$$\sigma(a_i) = \begin{cases} a_1 & \text{if } i = 1, \\ a_i & \text{if } a_i > \sigma_T(a_j) \text{ for all } 1 \le j < i, \\ \sigma_T(a_{i-1}) & \text{otherwise.} \end{cases}$$

Then, by induction $f^{\Gamma^*_{a_m,\dots,a_1}}(T) = \sigma_T(a_m)$. Moreover, $a \in BA(T)$ if and only if $a = \sigma_T(a_1, \dots, a_m)$ for some sequence $a_1, \dots, a_m$. □

The Banks Set and the Sophisticated Agenda

Definition: The *Banks set* $BA(T)$ of a tournament T on A consists of the maximal elements of the maximal transitive (cycle-free) subsets X of A.

Theorem: Let $A = \{a_1, \ldots, a_n\}$ be a set of alternatives, $\Gamma^*_{a_1,\ldots,a_m}$ be a sophisticated agenda on A. Then, for all tournaments $T \in \mathscr{T}(A)$:

$$\mathrm{BA}(T) = S^{\Gamma^*_{a_1,\ldots,a_m}}(T).$$

Proof: For T and each sequence $a_1, \ldots, a_m$ in A define the *sophisticated sequence* $\sigma_T = (\sigma_T(a_1), \ldots, \sigma_T(a_m))$ as:

$$\sigma(a_i) = \begin{cases} a_1 & \text{if } i = 1, \\ a_i & \text{if } a_i \succ \sigma_T(a_j) \text{ for all } 1 \leq j < i, \\ \sigma_T(a_{i-1}) & \text{otherwise.} \end{cases}$$

Then, by induction $f^{\Gamma^*_{a_m,\ldots,a_1}}(T) = \sigma_T(a_m)$. Moreover, $a \in BA(T)$ if and only if $a = \sigma_T(a_1, \ldots, a_m)$ for some sequence $a_1, \ldots, a_m$. □

The Copeland Set

Definition: Let T be a tournament on A and $a \in A$

- *a's Copeland score*: $c(a) = |\{b \in A : a \succ b\}| - |\{b \in A : b \succ a\}|$
- $\mathrm{CO}(T) = \arg\max_{a \in A} c(a) = \{a \in A : c(a) \geq c(b) \text{ for all } b \in A\}$.

Components and Decompositions

Definition: A subset $X \subseteq A$ is a *component* of a tournament T if for all $x, x' \in X$ and all $y \notin X$:

$$y \succ x' \text{ iff } y \succ x'.$$

A *decomposition of a tournament T on A* is a partition of A into components.

Definition: For $\boldsymbol{X} = \{X_1, \ldots, X_k\}$ a decomposition of a tournament T, have $T/\boldsymbol{X}$ denote the tournament on $\boldsymbol{X}$ such that for all $X, X' \in \boldsymbol{X}$:

$$X \succ X' \text{ in } T/\boldsymbol{X} \text{ iff } X \neq X' \text{ and there are } x \in X \text{ and } x' \in X' \text{ with } x \succ x' \text{ in } T$$

For X a component of T we have $T_X = T/\boldsymbol{X}$ where $\boldsymbol{X} = \{\{a\} \colon a \notin X\} \cup \{X\}$.

Components and Decompositions

Definition: A subset $X \subseteq A$ is a *component* of a tournament T if for all $x, x' \in X$ and all $y \notin X$:

$$y \succ x' \text{ iff } y \succ x'.$$

A *decomposition of a tournament T on A* is a partition of A into components.

Definition: For $\boldsymbol{X} = \{X_1, \ldots, X_k\}$ a decomposition of a tournament T, have $T/\boldsymbol{X}$ denote the tournament on $\boldsymbol{X}$ such that for all $X, X' \in \boldsymbol{X}$:

$$X \succ X' \text{ in } T/\boldsymbol{X} \text{ iff } X \neq X' \text{ and there are } x \in X \text{ and } x' \in X' \text{ with } x \succ x' \text{ in } T$$

For X a component of T we have $T_X = T/\boldsymbol{X}$ where $\boldsymbol{X} = \{\{a\} \colon a \notin X\} \cup \{X\}$.

Components and Decompositions

Definition: A subset $X \subseteq A$ is a *component* of a tournament T if for all $x, x' \in X$ and all $y \notin X$:

$$y \succ x' \text{ iff } y \succ x'.$$

A *decomposition of a tournament T on A* is a partition of A into components.

Definition: For $\boldsymbol{X} = \{X_1, \dots, X_k\}$ a decomposition of a tournament T, have $T/\boldsymbol{X}$ denote the tournament on $\boldsymbol{X}$ such that for all $X, X' \in \boldsymbol{X}$:

$$X \succ X' \text{ in } T/\boldsymbol{X} \text{ iff } X \neq X' \text{ and there are } x \in X \text{ and } x' \in X' \text{ with } x \succ x' \text{ in } T$$

For X a component of T we have $T_X = T/\boldsymbol{X}$ where $\boldsymbol{X} = \{\{a\} \colon a \notin X\} \cup \{X\}$.

Components and Decompositions

Definition: A subset $X \subseteq A$ is a *component* of a tournament T if for all $x, x' \in X$ and all $y \notin X$:

$$y \succ x' \text{ iff } y \succ x'.$$

A *decomposition of a tournament T on A* is a partition of A into components.

Definition: For $\boldsymbol{X} = \{X_1, \dots, X_k\}$ a decomposition of a tournament T, have $T/\boldsymbol{X}$ denote the tournament on $\boldsymbol{X}$ such that for all $X, X' \in \boldsymbol{X}$:

$$X \succ X' \text{ in } T/\boldsymbol{X} \text{ iff } X \neq X' \text{ and there are } x \in X \text{ and } x' \in X' \text{ with } x \succ x' \text{ in } T$$

For X a component of T we have $T_X = T/\boldsymbol{X}$ where $\boldsymbol{X} = \{\{a\} \colon a \notin X\} \cup \{X\}$.

Composition Consistency

Definition: A social choice correspondence is *composition consistent* whenever for any tournament T and any decomposition $\boldsymbol{X}$ of T we have:

$$S(T) = \bigcup_{X \in S(T/\boldsymbol{X})} S(X).$$

Definition: A social choice correspondence is *weakly composition consistent* whenever for any tournaments T and T' on A with $X \subseteq A$ as a component such that $T_X = T'_X$ we have:

- $S(T) \backslash X = S(T') \backslash X$ and
- $S(T) \cap X \neq \emptyset$ implies $S(T') \cap X \neq \emptyset$

Binary Voting Trees and Weak Composition Consistency

Theorem: For every binary voting tree Γ, S^Γ is weakly composition consistent.

Proof: Let X be a component of T. Define for each binary voting tree $\Gamma = (V, R, \phi)$ the voting tree $\Gamma_X = (V, R, \phi_X)$ where:

$$\phi_X(z) = \begin{cases} \phi(z) & \text{if } \phi(z) \notin X, \\ X & \text{otherwise.} \end{cases}$$

Then, by induction of the reduction process both:

- $\Gamma(T) \in X$ iff $\Gamma_X(T_X) = X$,
- $\Gamma(T) = a$ iff $\Gamma_X(T_X) = a$, for all $a \notin X$.

The theorem then follows. □

Binary Voting Trees and Weak Composition Consistency

Theorem: For every binary voting tree Γ, S^Γ is weakly composition consistent.

Proof: Let X be a component of T. Define for each binary voting tree $\Gamma = (V, R, \phi)$ the voting tree $\Gamma_X = (V, R, \phi_X)$ where:

$$\phi_X(z) = \begin{cases} \phi(z) & \text{if } \phi(z) \notin X, \\ X & \text{otherwise.} \end{cases}$$

Then, by induction of the reduction process both:

- $\Gamma(T) \in X$ iff $\Gamma_X(T_X) = X$,
- $\Gamma(T) = a$ iff $\Gamma_X(T_X) = a$, for all $a \notin X$.

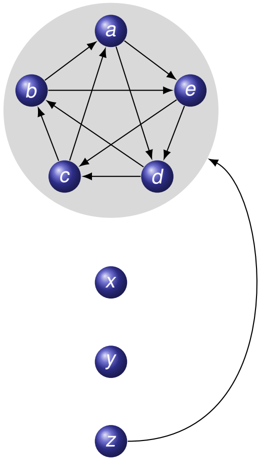
The theorem then follows. □

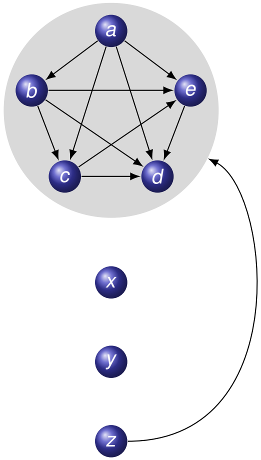
The Copeland Solution and Binary Trees

Theorem: The Copeland solution is not composition consistent.

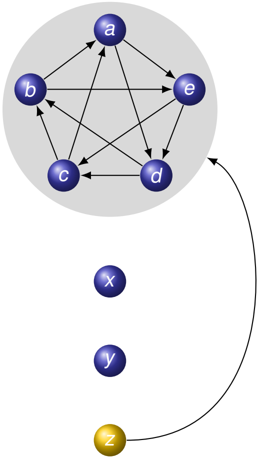
Corollary: There is no binary tree of which the choice function always selects a Copeland winner, provided there are at least 8 alternatives.

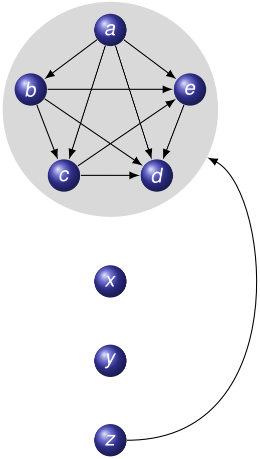
The Copeland Solution not Composition Consistent



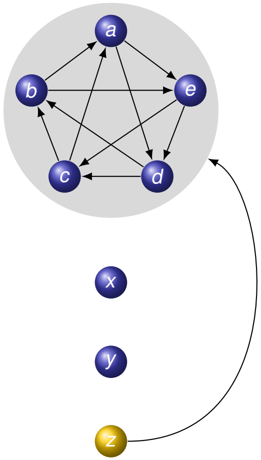


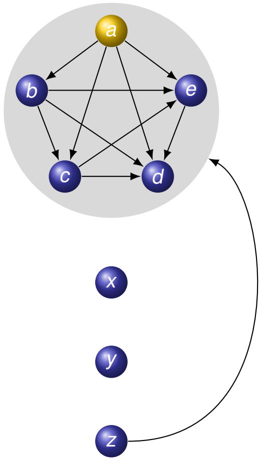
The Copeland Solution not Composition Consistent





The Copeland Solution not Composition Consistent





Summary

- Binary voting procedures and sophisticated voting
- Binary voting trees and the reduction method
- Each binary voting tree chooses from the Good set
- The Good set is computable by simple agendas
- The Banks set is computable by sophisticated agendas
- The Copeland set is not computable by binary trees