



# ARGUING ABOUT VOTING RULES

# Olivier Cailloux and Ulle Endriss

LAMSADE, Université Paris-Dauphine and ILLC, University of Amsterdam

# Main objects

- $A = \{a, b, c\}$  a set of alternatives
- $N = \{v_1, v_2, v_3\}$  a set of *voters*
- R a *profile*: maps voters to linear orders on A
- *Voting rule*: maps profiles to subsets of alternatives:  $[R \mapsto A \subseteq A]$

### Some example profiles

# Our objective

#### Scenario:

- Assume a commitee wants to choose a voting rule
- Many rules
- No best rule

We want to:

- automatically explain the outcome of rules
- based on their properties
- by referring to *examples*

# Example dialogue

Let's argue that in  $\mathbf{R}$ , a should win, using "unanimity", "symmetry" and "reinforcement"

**System** Consider election  $R_1$ , involving only voter  $v_1$ . Do you agree that a, enjoying unanimous support, should win this election?

User Yes, of course.

System Now consider election  $R_2$ , involving only voters  $v_2$  and  $v_3$ . Do you agree

that, for symmetry reasons, the outcome should be a three-way tie?

User Yes, that sounds reasonable.

**System** Observe that when we combine  $R_1$  and  $R_2$ , we obtain our election of interest,

namely R. Do you agree that in this combined election, as there was a three-

way tie in  $R_2$ ,  $R_1$  should be used to decide the winner?

User Yes, I do.

**System** To summarise, you agree that a should win for R.

# Language

- ullet A general language  $\mathcal L$  to talk about properties of voting rules
- Logic-based
- Can represent axioms and specific outcomes (such as  $[R \mapsto A]$ )
- Permits *arguments*: proof (in the language) from axioms to outcomes

**Theorem 1** (Completeness). Take some axioms of a voting rule. Encode in our language as J. Suppose some outsome  $[R \mapsto A]$  holds for every rule f satisfying those axioms. Then there exists a proof of  $[R \mapsto A]$  in our logic that is grounded in J.

# Example for axioms in $\mathcal{L}$

**REINF**:  $[R_1 \mapsto A_1] \land [R_2 \mapsto A_2] \rightarrow [R_1 \oplus R_2 \mapsto A_1 \cap A_2]$  with  $A_1 \cap A_2 \neq \emptyset$ 

**CANC**:  $\mathbf{R}$  such that  $[\forall (a,b): a > b \text{ as often as } b > a] \to [\mathbf{R} \mapsto \mathcal{A}]$ 

Anon:  $[R \mapsto A] \to [(R \circ \sigma) \mapsto A]$  for any permutation  $\sigma$ 

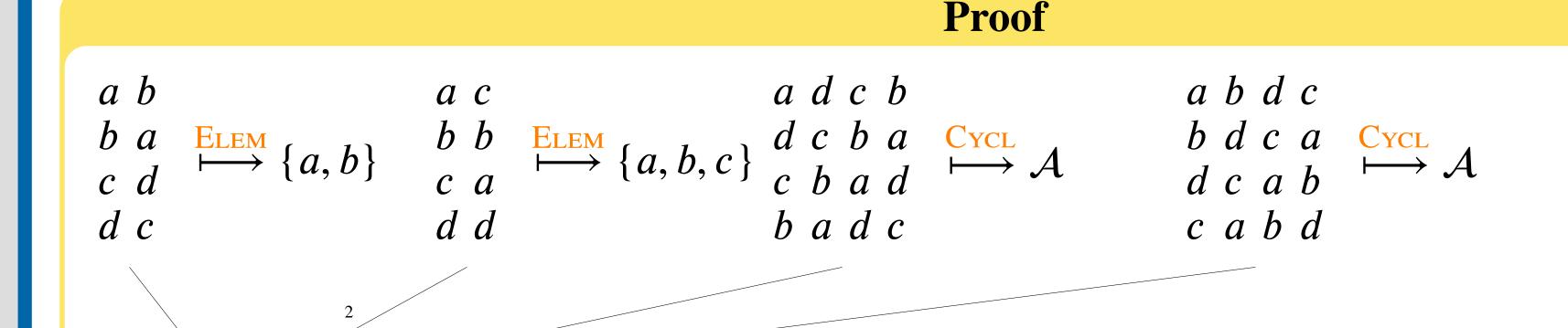
#### **Arguing for Borda**

We provide a concrete algorithm for explaining the outcomes of the Borda voting rule. Given any profile  $R^*$ :

- Let  $W \subseteq \mathcal{A}$  be the set of Borda winners
- The algorithm generates a proof that  $[\mathbf{R}^* \mapsto W]$
- Grounded in axioms: Reinf, Canc, Elem, Cycl

# Example: Arguing for Borda

Given 
$$\mathbf{R}^* = \begin{pmatrix} a & c \\ b & b \\ d & a \end{pmatrix}$$
, let's argue that  $[\mathbf{R}^* \mapsto \{a, b\}]$ 



 $\mathbf{R}' \longmapsto \{a,b\}$ 

REINF

 $\mathbf{R}' \oplus \overline{4\mathbf{R}^*} \oplus 4\mathbf{R}^* \longmapsto \{a, b\}$ 

 $R' \oplus \overline{4R^*} \oplus 4R^*$  must have the same winners as R', by Reinf and Canc

 $4\mathbf{R}^* \longmapsto \{a,b\}$ 

 $4R^*$  must have the same winners as  $R' \oplus \overline{4R^*} \oplus 4R^*$ , because  $R' \oplus \overline{4R^*}$  cancels

 $\mathbf{R}^* \longmapsto \{a, b\}$ 

Oh yeah

Olivier Cailloux and Ulle Endriss. *Arguing about Voting Rules*. In Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2016), IFAAMAS.