Deciding a Slater Winner is Complete for Parallel Access to NP

Michael Lampis
LAMSADE

STACS 2022
Mar 16th 2022

One-Slide Summary

- Context: complexity of voting rules
  - How to pick “most popular” out of n candidates?
  - Hard to define for $n \geq 3$!
  - Many voting rules have been proposed.
- In this talk: **Slater Rule**
  - Def: c is winner if number of head-to-head matchups that need to be reversed to make c best is minimum.
- Question: Determine complexity of following
  - Given list of voter preferences and candidate c, is c a Slater winner?
- Previous bounds: NP-hard, in Θ_2^p.
- **This talk**: Problem is Θ_2^p-complete.

Thanks

Many thanks to Jérôme Lang for suggesting this problem and guessing the correct solution!

The Slater Rule

An example

Angela

Five candidates for President of STACS

An example

Angela

Boris

Five candidates for President of STACS

An example

Angela

Boris

Christine

Five candidates for President of STACS

An example

Angela

Boris

Christine

Donald

Five candidates for President of STACS

An example

Angela

Boris

Christine

Donald

Emmanuel

Five candidates for President of STACS

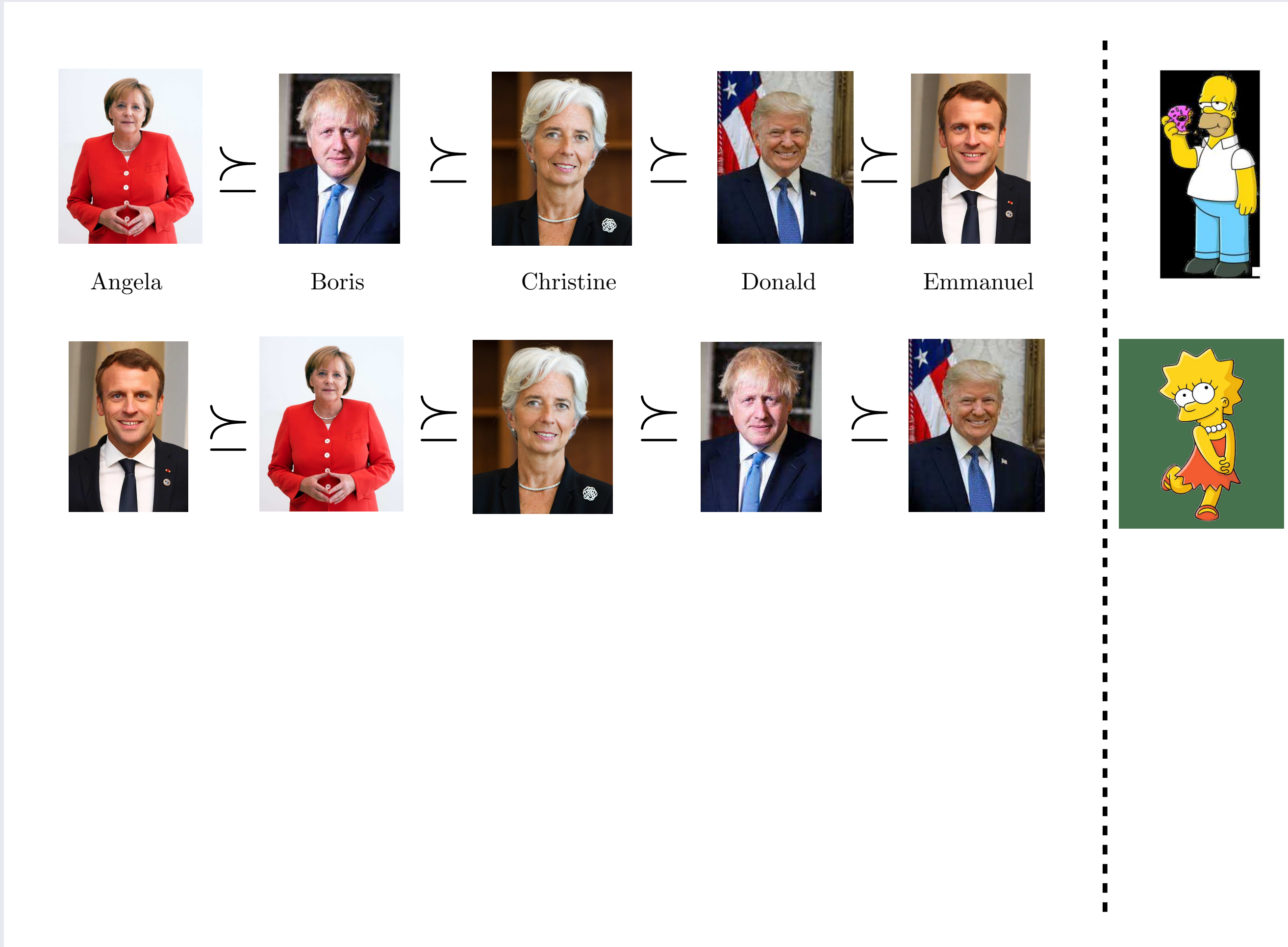
An example

 $\succeq$  $\succeq$  $\succeq$  $\succeq$  

Angela Boris Christine Donald Emmanuel

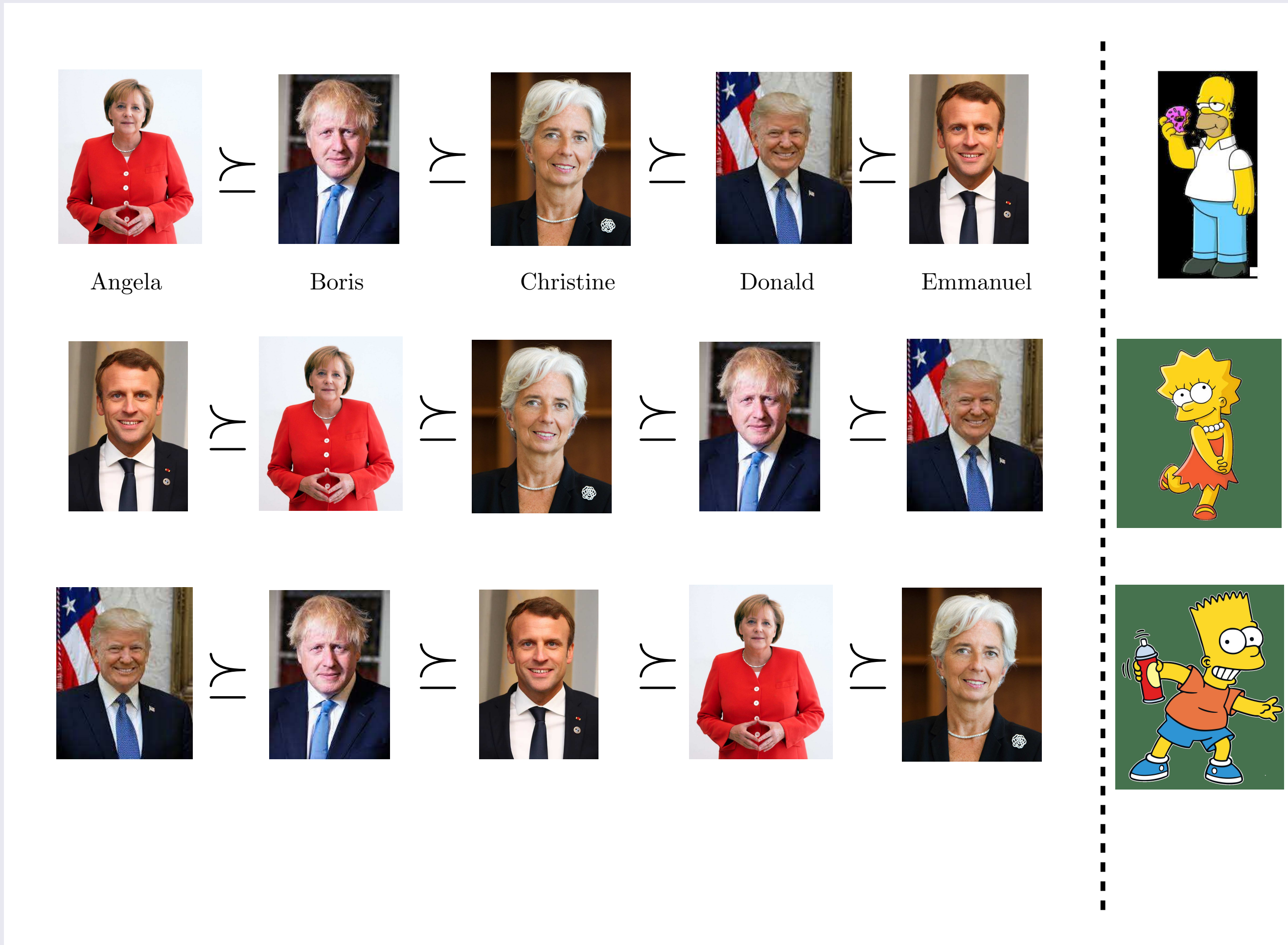
Three voters with distinct preferences

An example



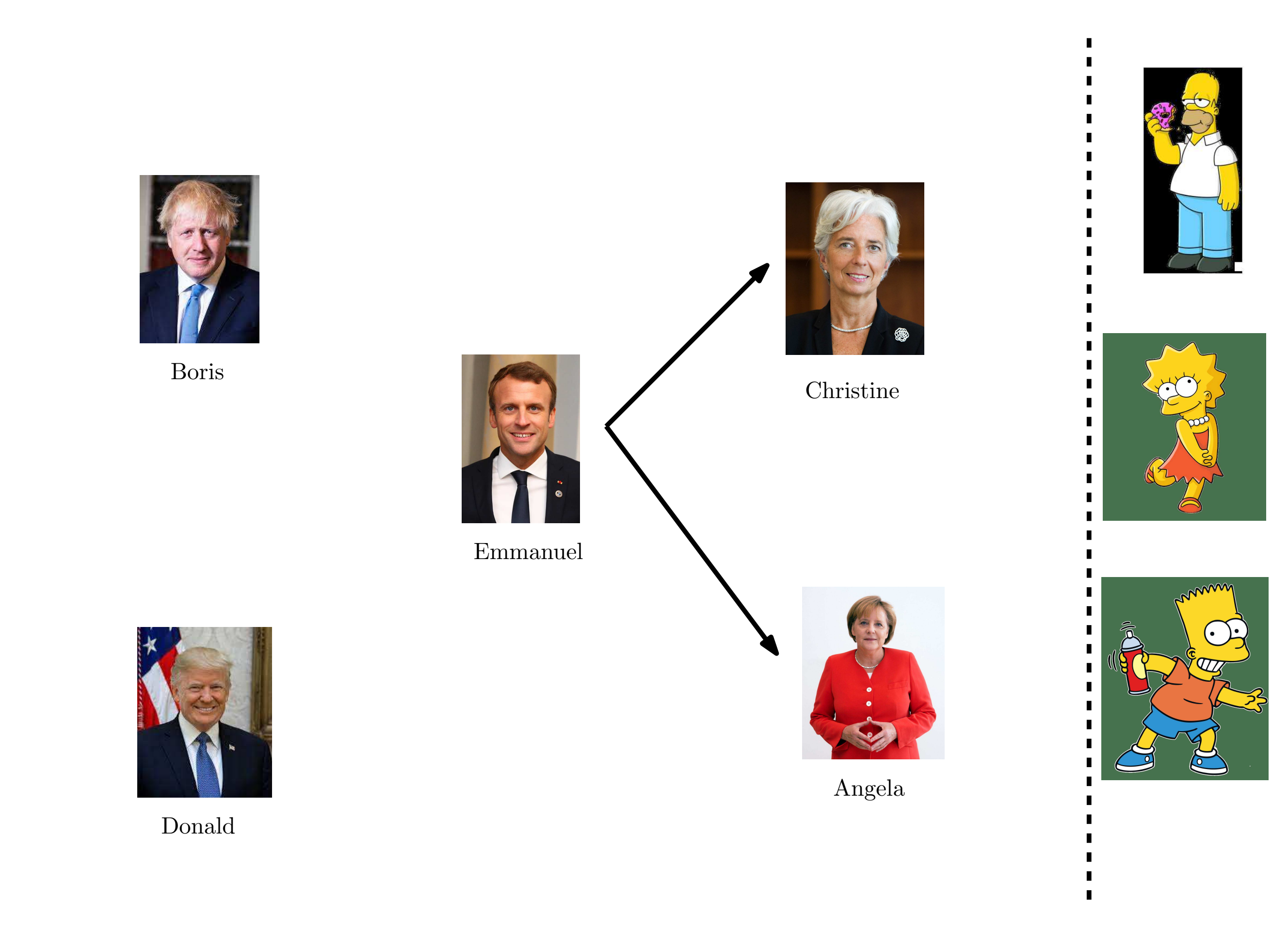
Three voters with distinct preferences

An example



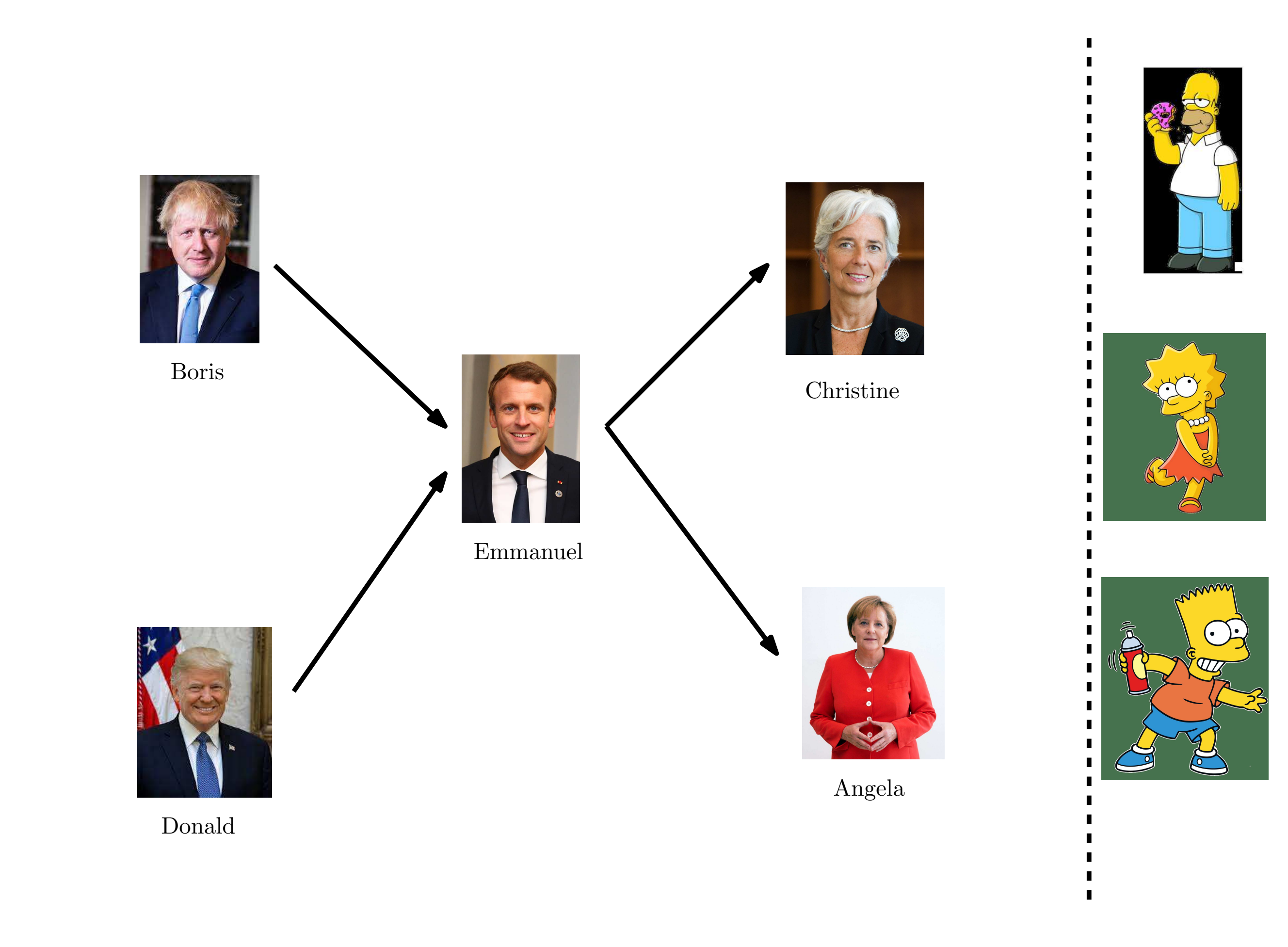
Three voters with distinct preferences

An example



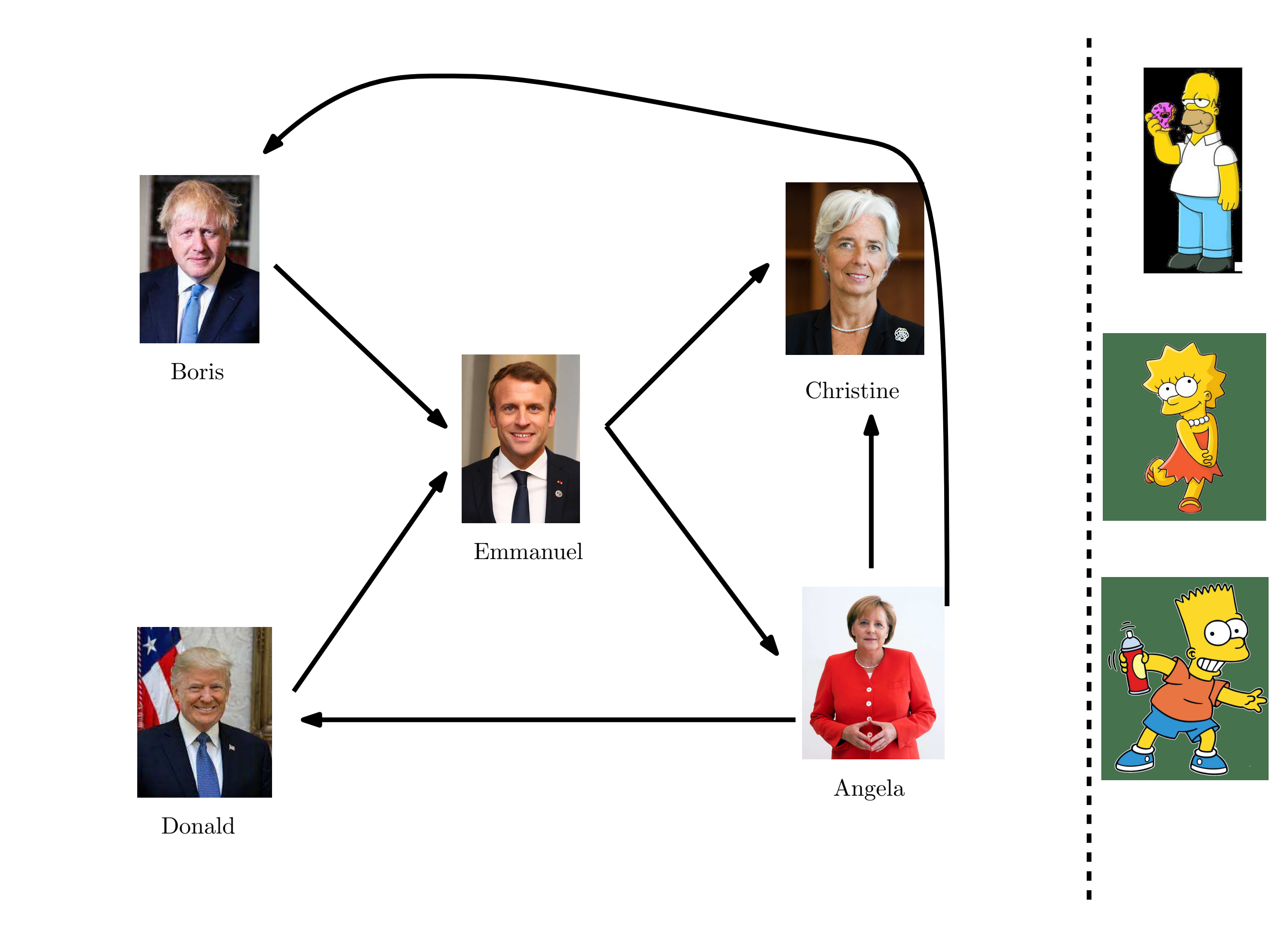
Head-to-Head results

An example



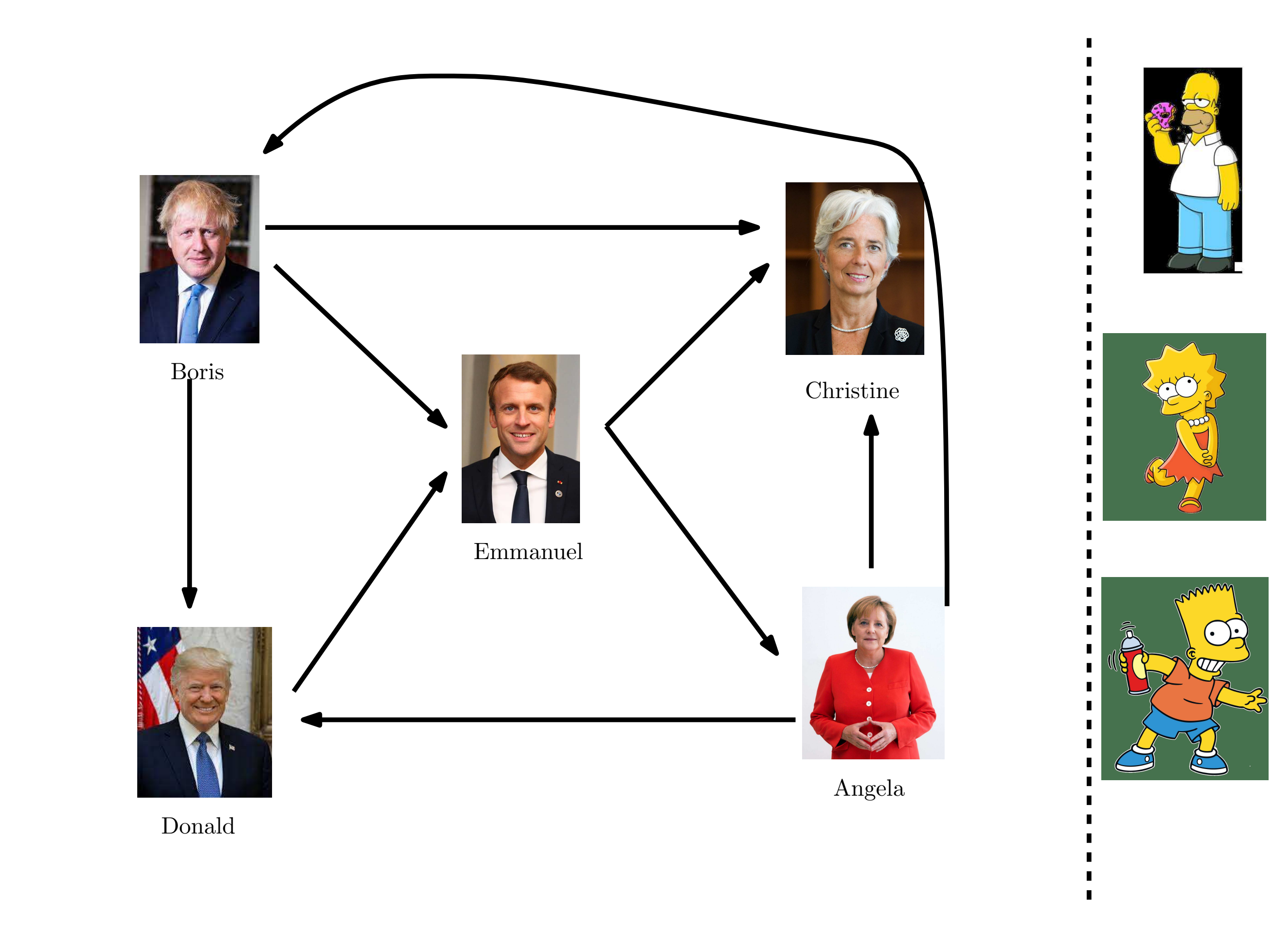
Head-to-Head results

An example



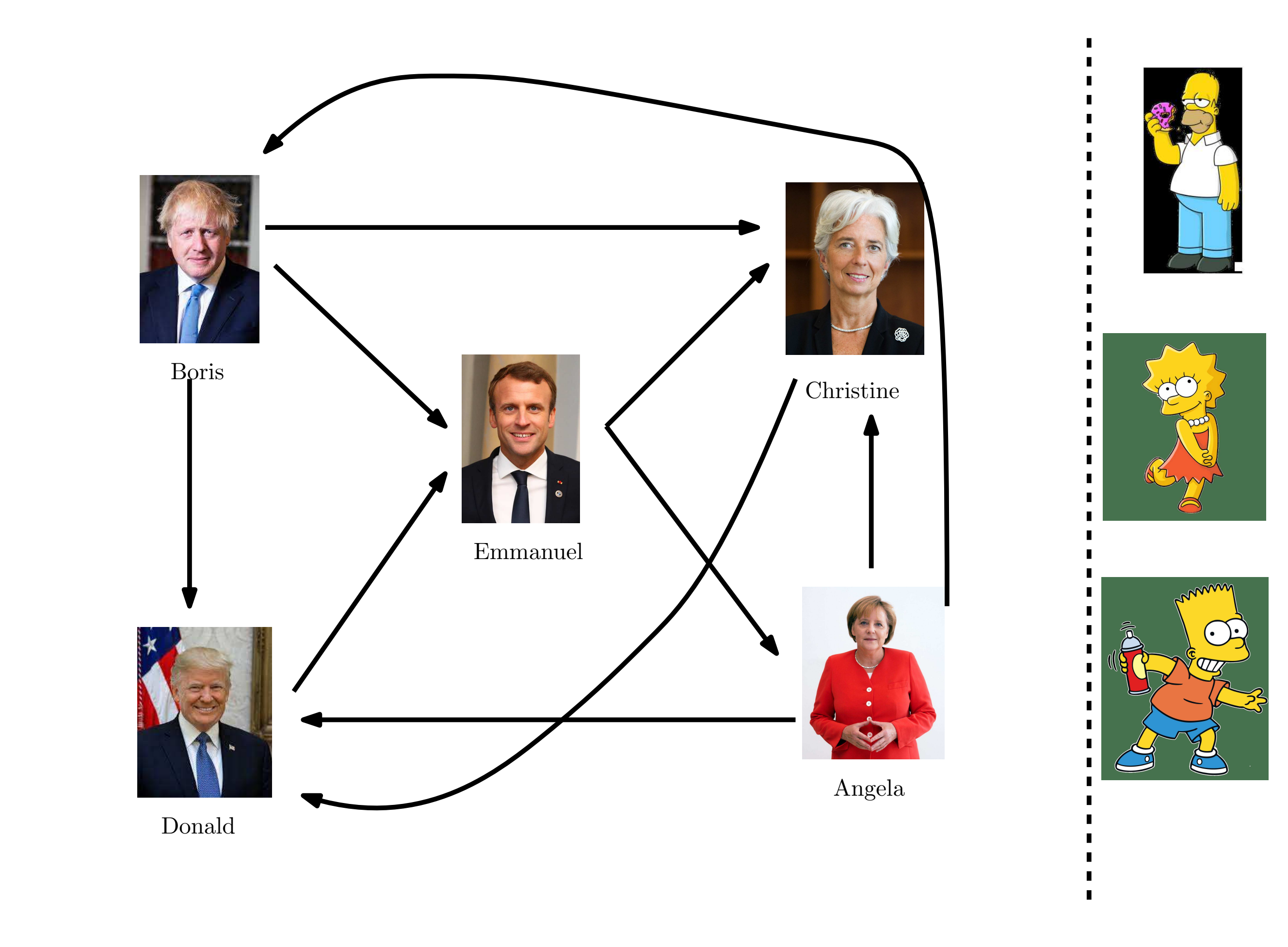
Head-to-Head results

An example



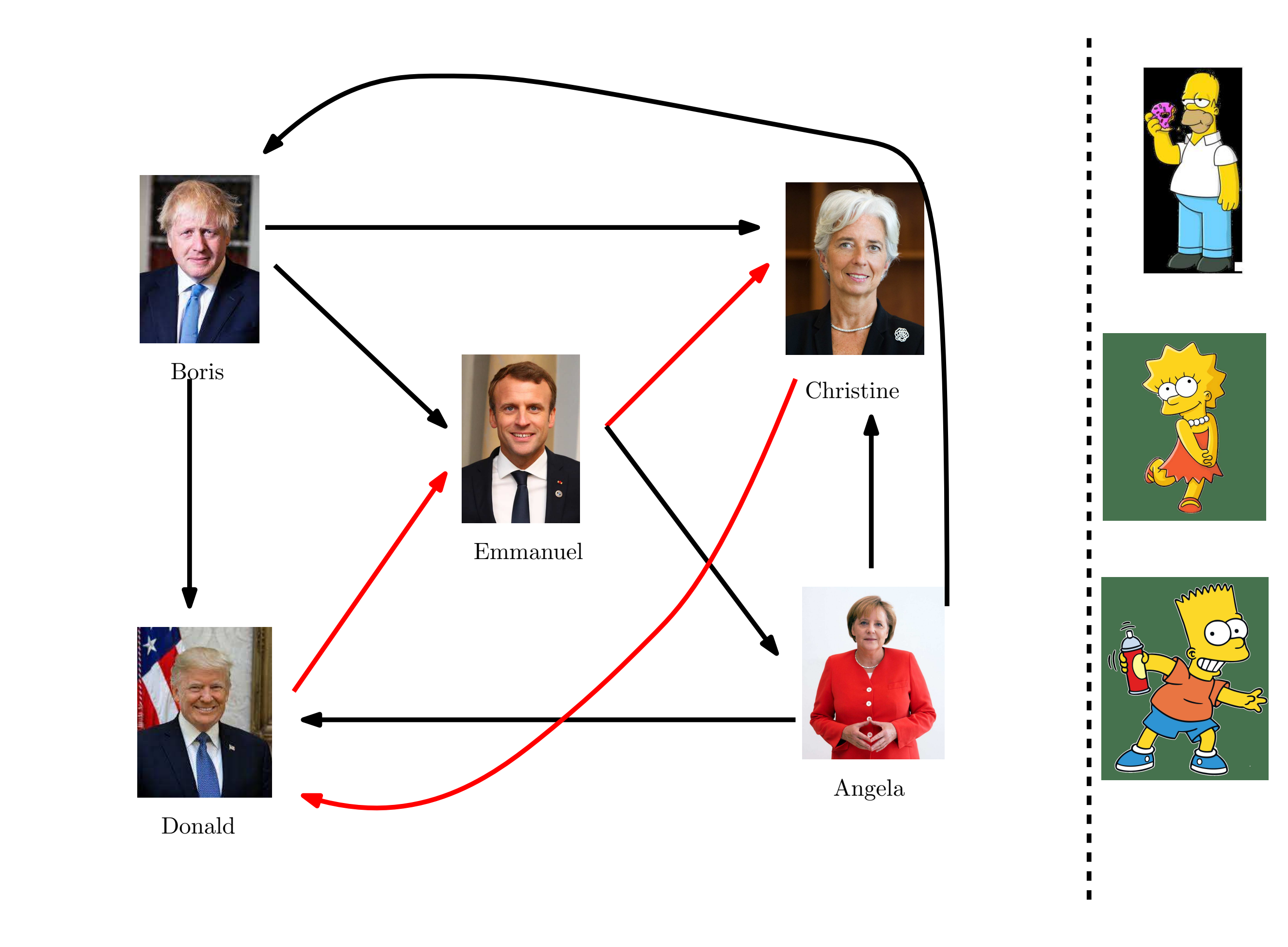
Head-to-Head results

An example



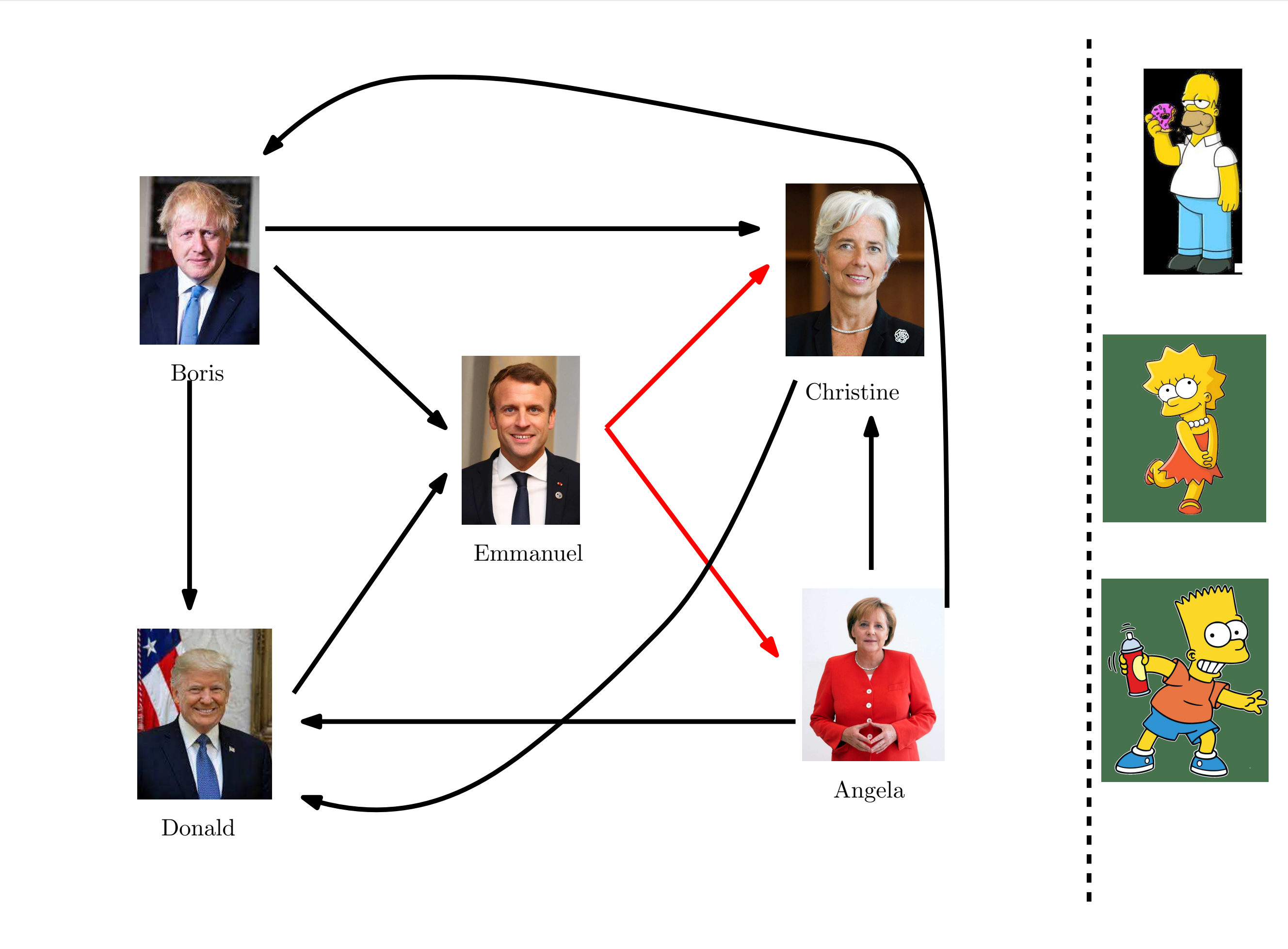
Head-to-Head results

An example



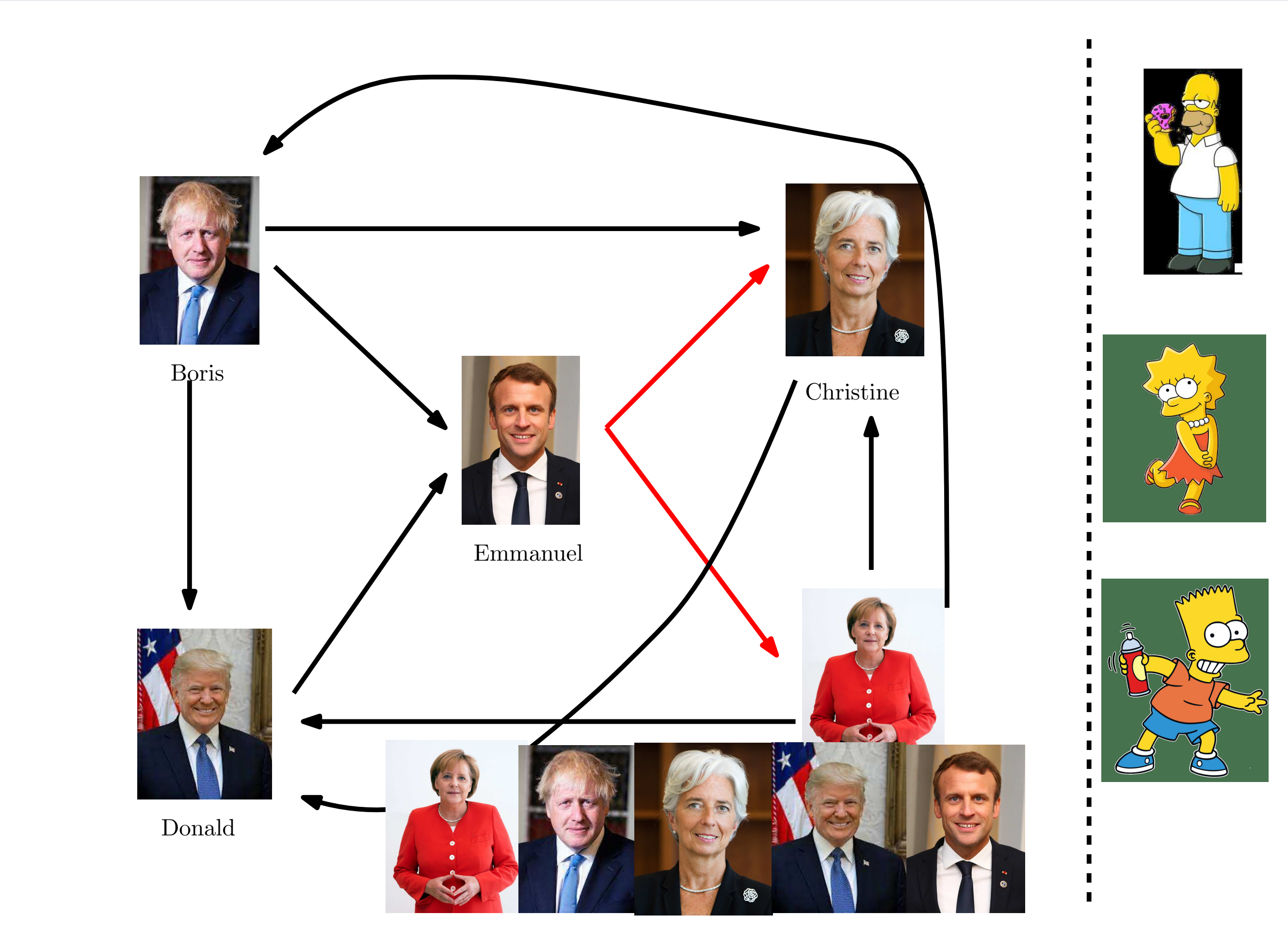
These results contain contradictions!

An example



Can be repaired by flipping two arcs.

An example



Angela's Slater score is 2.

The Slater Rule

Input:

- n candidates and m voters.
- For each voter a total ranking of all candidates.

Head-to-Head Graph

- A vertex for each candidate.
- Arc $u \to v$ if u beats v.

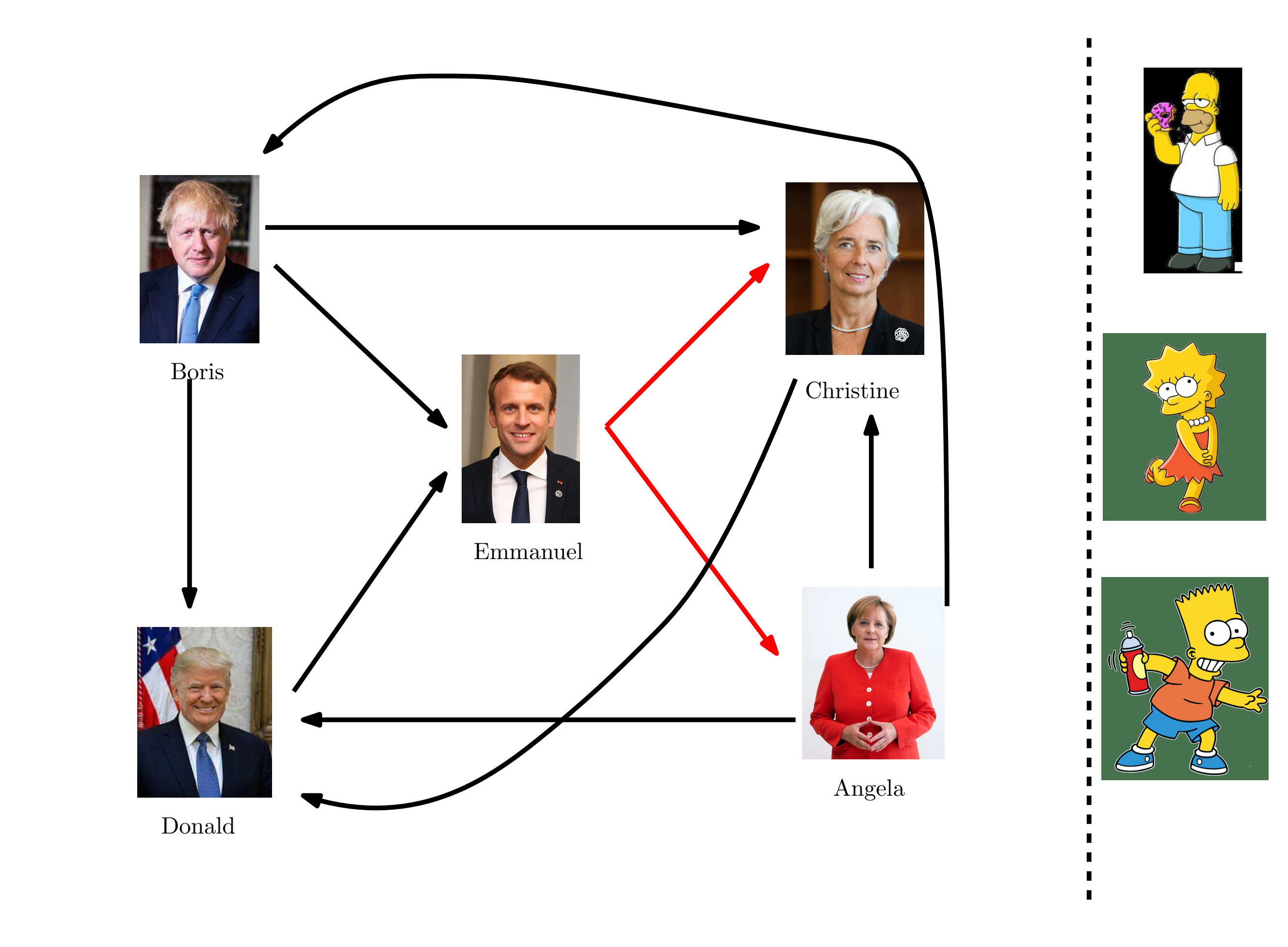
Slater Score of u:

- Minimum number of arcs that need to be reversed so that u is winner and ranking is globally consistent.

Slater Winner:

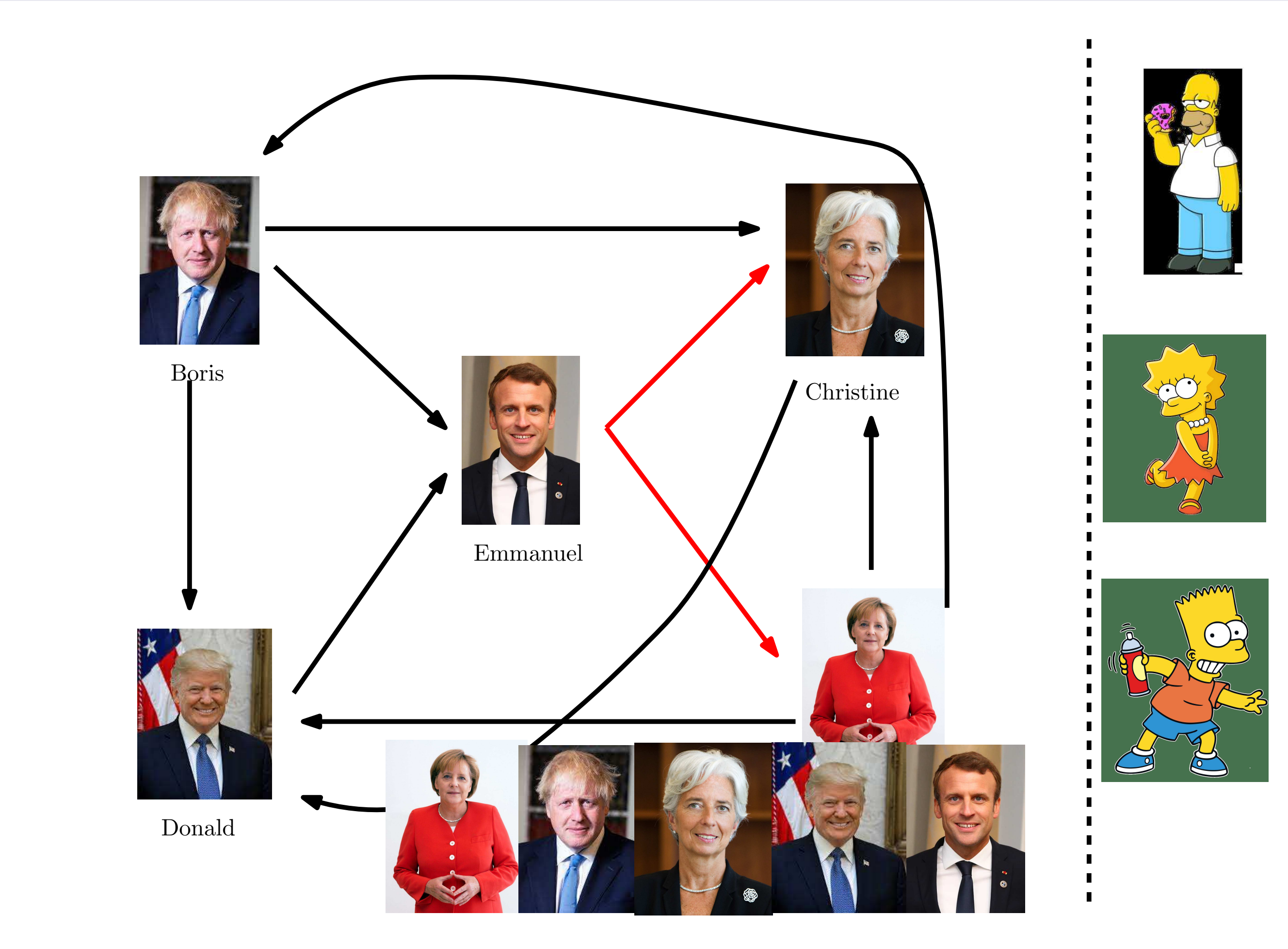
- Candidate with minimum Slater score.

Back to example



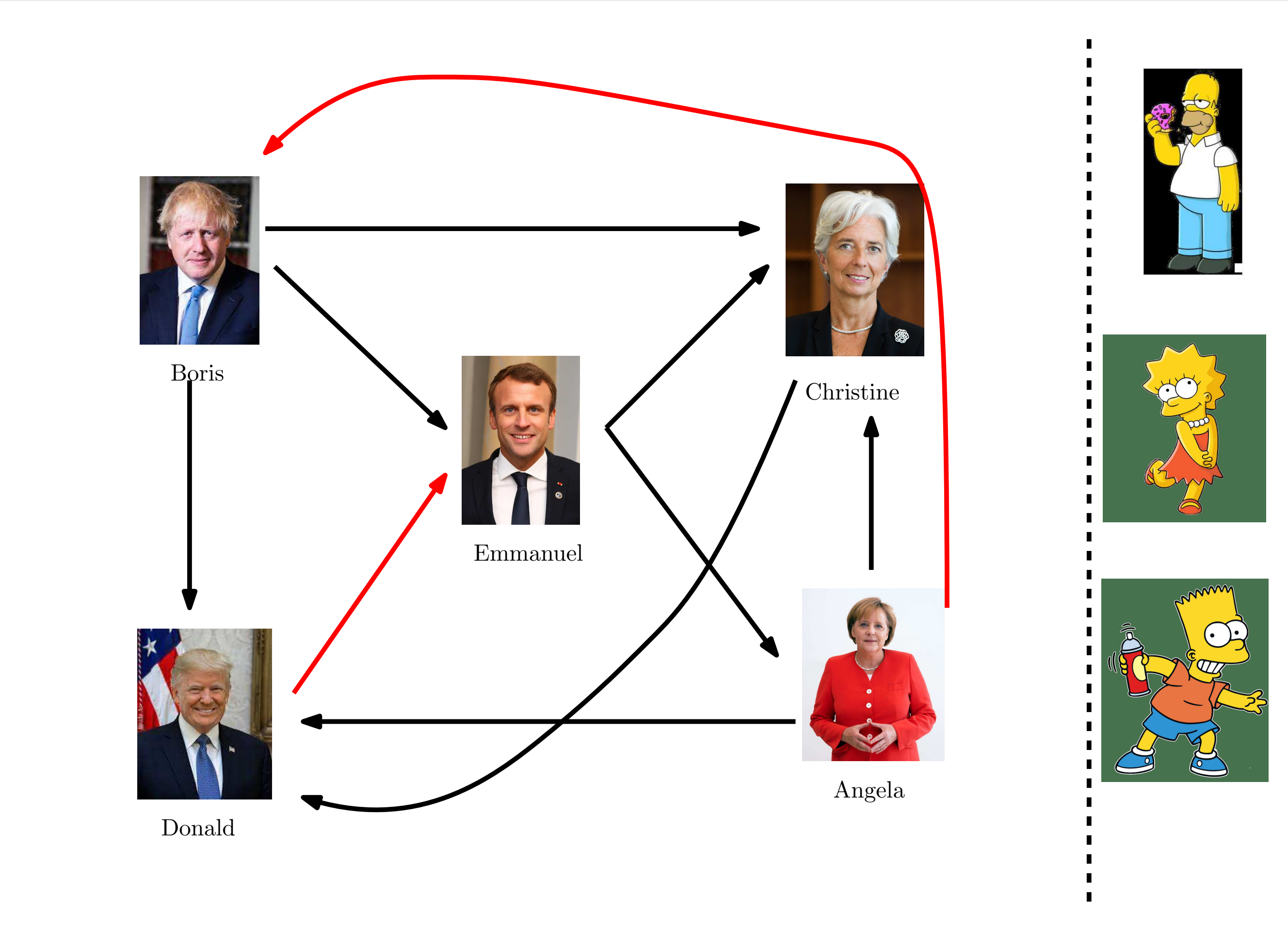
Angela has a score of 2.

Back to example



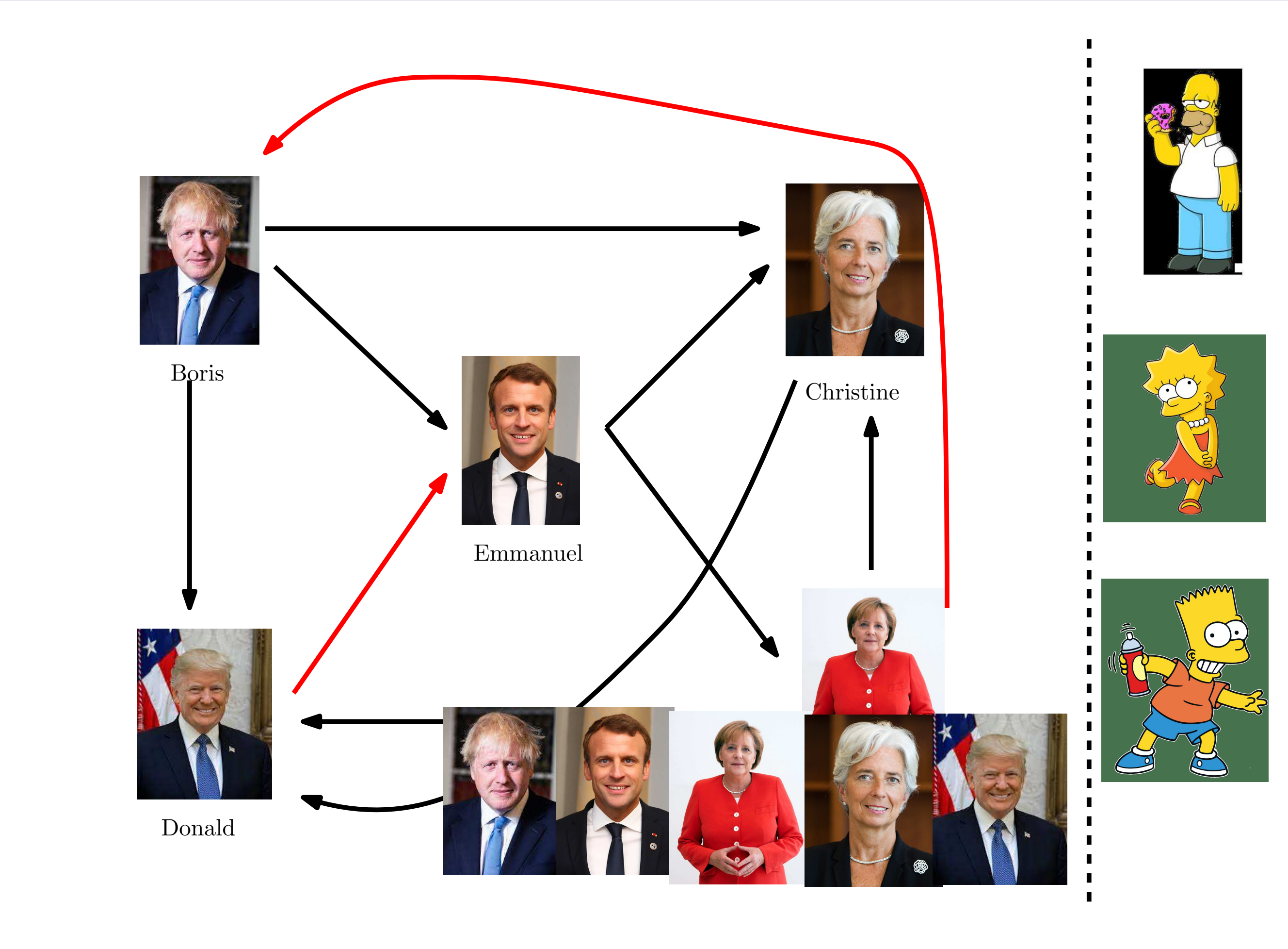
Angela has a score of 2.

Back to example



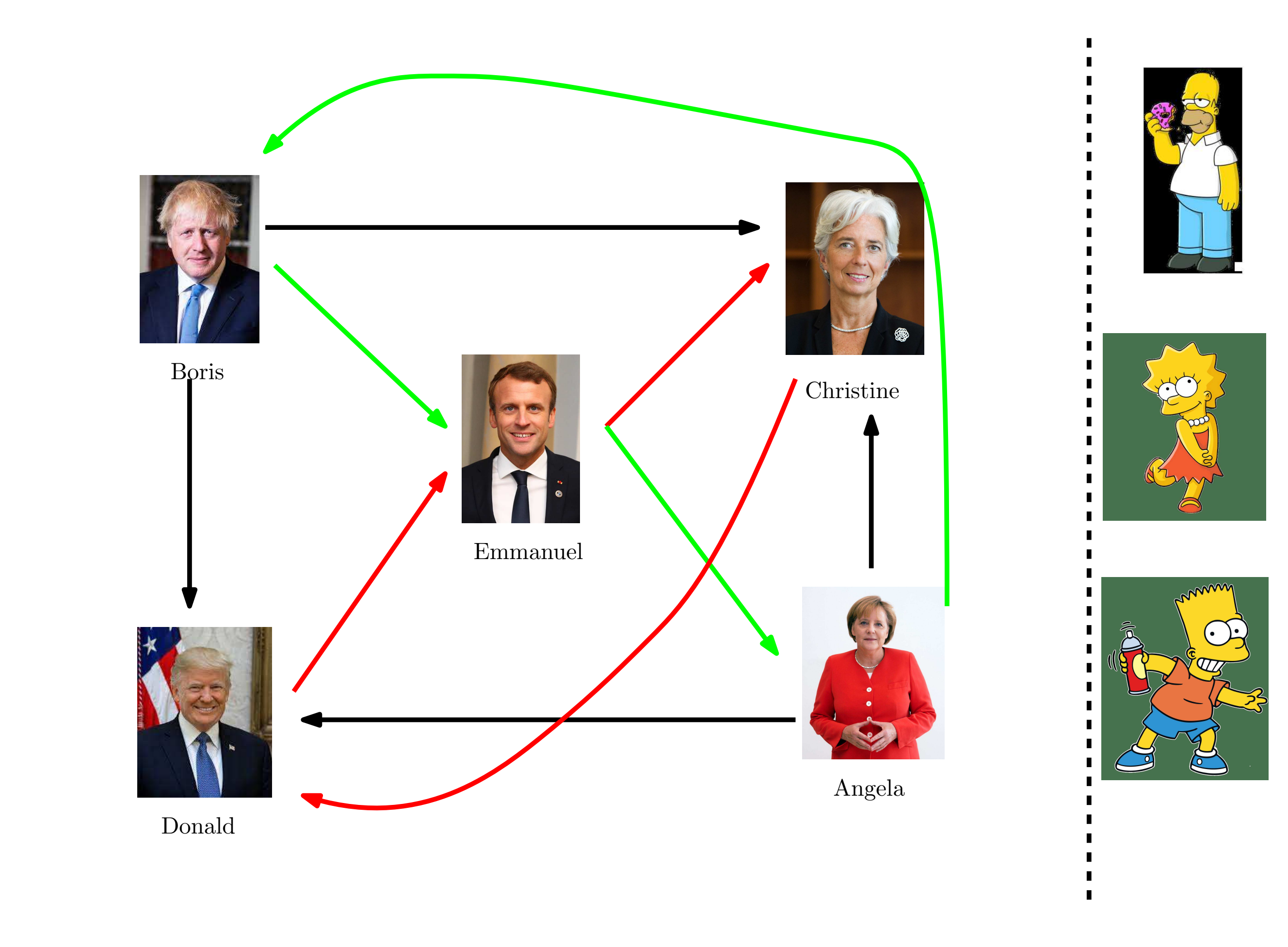
Boris also has a score of 2.

Back to example



Boris also has a score of 2.

Back to example



2 is best possible, so they are both Slater winners.

Complexity Considerations

Basic Decision Problem:

- Is v a Slater winner?

- If odd number of voters $\rightarrow$ graph is a **Tournament**.
- Problem seems at least as hard as FAST.
- Is it in NP?
- How do we prove someone's score$= k$?
- How do we prove no one has a better score?

Complexity Classes

Some Problems

Consider following variations of the problem:

- Input: T, v, k. Is v's score $\leq k$?
- Input: T, v, k. Is v's score $\geq k$?
- Input: T, v, k. Is v's score $= k$?

- Input: T, v, u, k. Is v's score $\leq u$'s score?
- Input: T, v, k. Is v's score $\leq$ everyone else's?

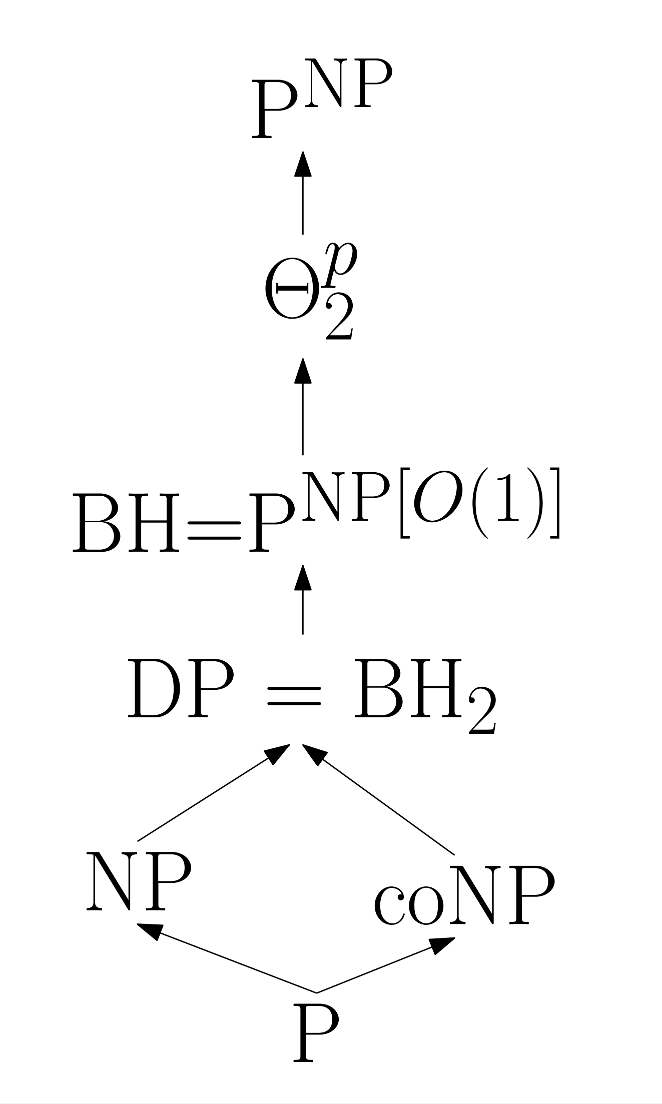
Some Problems

Consider following variations of the problem:

- Input: T, v, k. Is v's score $\leq k$?
- Input: T, v, k. Is v's score $\geq k$?
- Input: T, v, k. Is v's score $= k$?
- Input: T, v, u, k. Is v's score $\leq u$'s score?
- Input: T, v, k. Is v's score $\leq$ everyone else's?

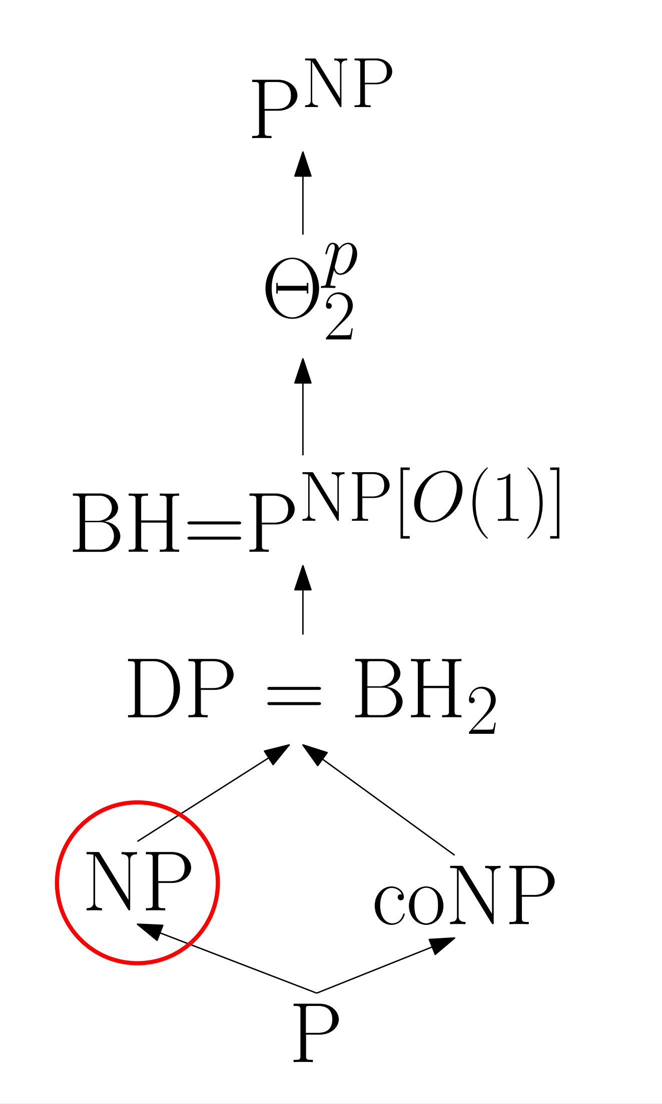
- Problems **kind of** poly-time equivalent.
  - If one is in P, others are in P.
- Are they **really** equivalent?
  - Can I transform an instance of one into an equivalent instance of the other? (Karp reduction)

- Reminder of some classes



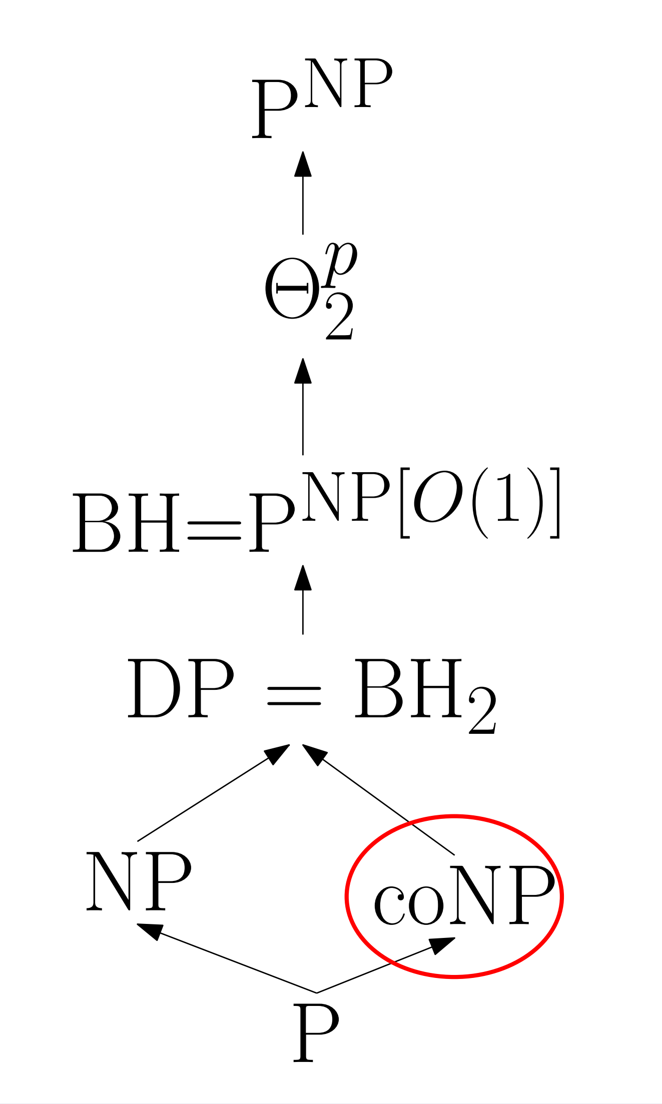
- Reminder of some classes

- **NP**: Problems with a Yes certificate
- Example: 3-Coloring
- Example: is Slater score of u at most k?

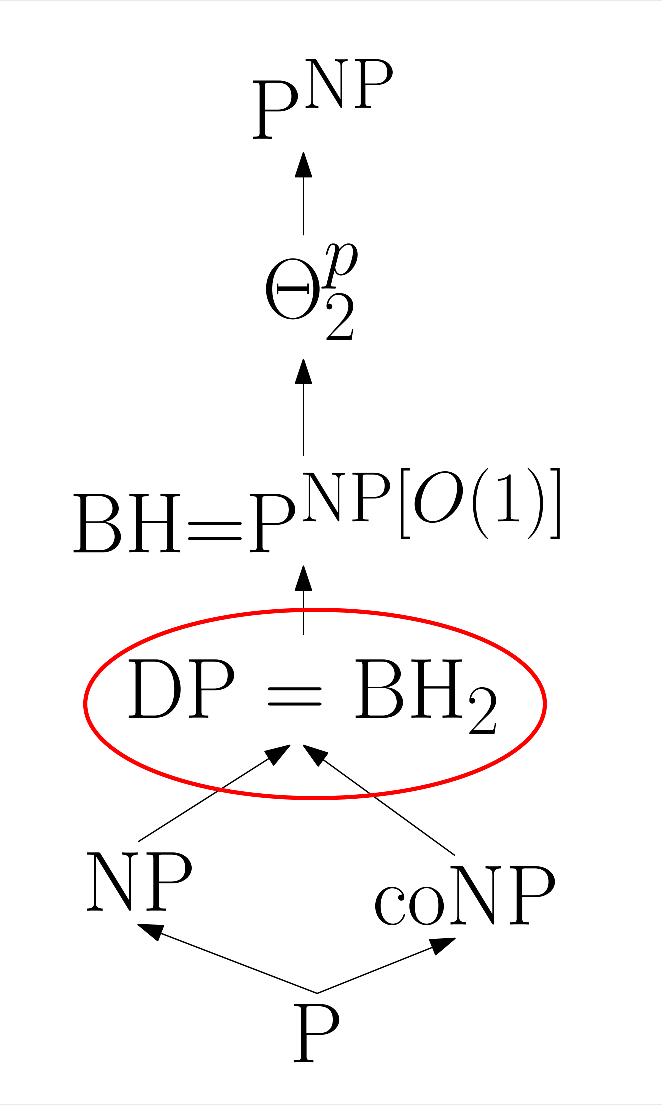


- Reminder of some classes

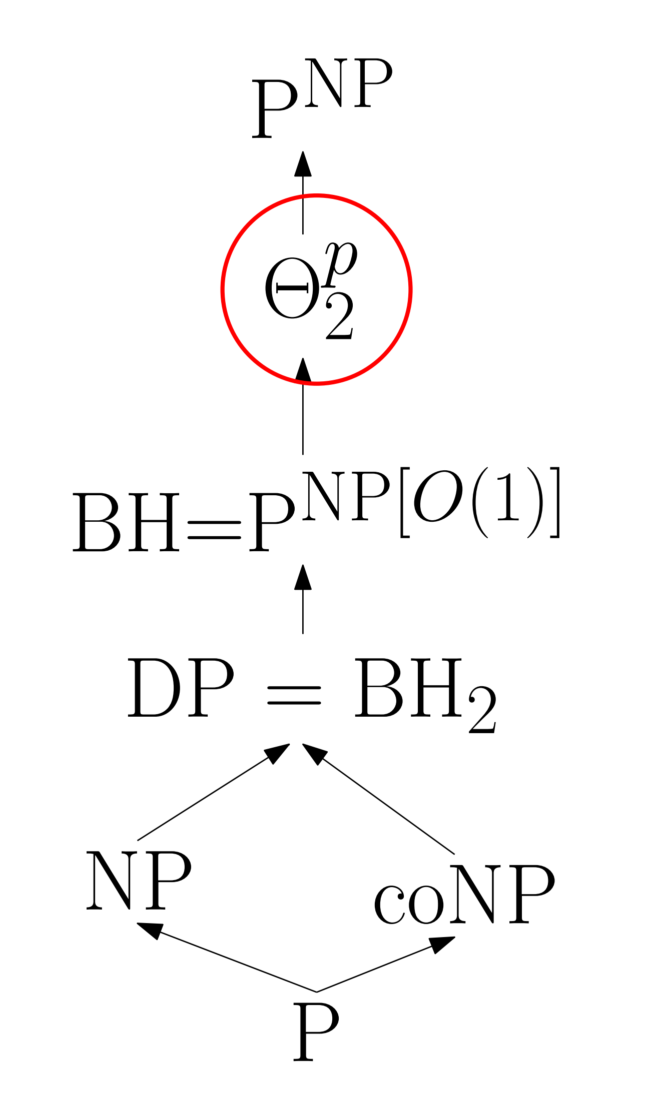
- **coNP**: Problems with a No certificate
- Example: Formula Equivalence
- Example: is Slater score of u at least k?



- Reminder of some classes
- What about: is Slater score of u **exactly** k?
- **DP**: Intersection of a problem in NP with a problem in coNP.
- Essentially: P with **two** calls to an NP oracle.



- Reminder of some classes
- Problem: Is Slater score of $u \leq$ Slater score of v?
- How many calls to NP oracle needed?
- Note: problem is in P^{NP}.
- Actually: problem is in Θ_2^p.



Parallel Access to NP

The class Θ_2^p

- $\mathsf{P}^{\mathrm{NP}[\log n]}$
  - P with $\log n$ calls to an NP oracle
- $\mathsf{P}_{||}^{\mathrm{NP}}$
  - P with $n^{O(1)}$ **non-adaptive** calls to an NP oracle
- L^{NP}
  - L with $n^{O(1)}$ calls to an NP oracle

Parallel Access to NP and Elections

- Many election systems are **complete** for Θ_2^p
  - Dodgson [Hemaspaandra, Hemaspaandra, Rothe, J.ACM'97]
  - Young [Rothe, Spakowski, Vogel, TCS'03]
  - **Kemeny** [Hemaspaandra, Spakowski, Vogel, TCS'05]

SLATER WINNER $\in \Theta_2^p$

- Compute Angela's score, best score with binary search ($O(\log n)$) oracle calls)
- Compute everyone's score with $n^{O(1)}$ non-adaptive calls.

SLATER WINNER is NP-hard (under Turing reductions)

- If we could find Slater winner in P, we could solve FAST.

Main Result: SLATER WINNER is Θ_2^p-complete.

Reductions

Not so FAST!

Background: FEEDBACK ARC SET ON TOURNAMENTS

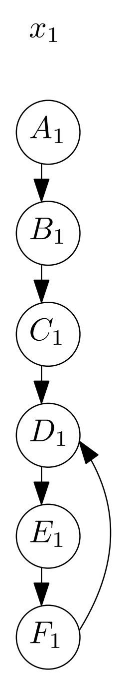
- Problem with interesting history.
- FAS is easily NP-complete (from VC$\rightarrow$FVS).
- Whether still NP-complete on **Tournaments** open for a long time!
  - Conjectured NP-complete by [Bang-Jensen, Thomassen, SIDMA'92]
  - **Almost** proved (via randomized reduction) by [Ailon, Charikar, Newman, STOC'05]
  - **Proved** (derandomized) by [Alon SIDMA'06] and [Charbit, Thomassé, Yeo Comb. Prob. Comp. '06]

Not so FAST!

Background: FEEDBACK ARC SET ON TOURNAMENTS

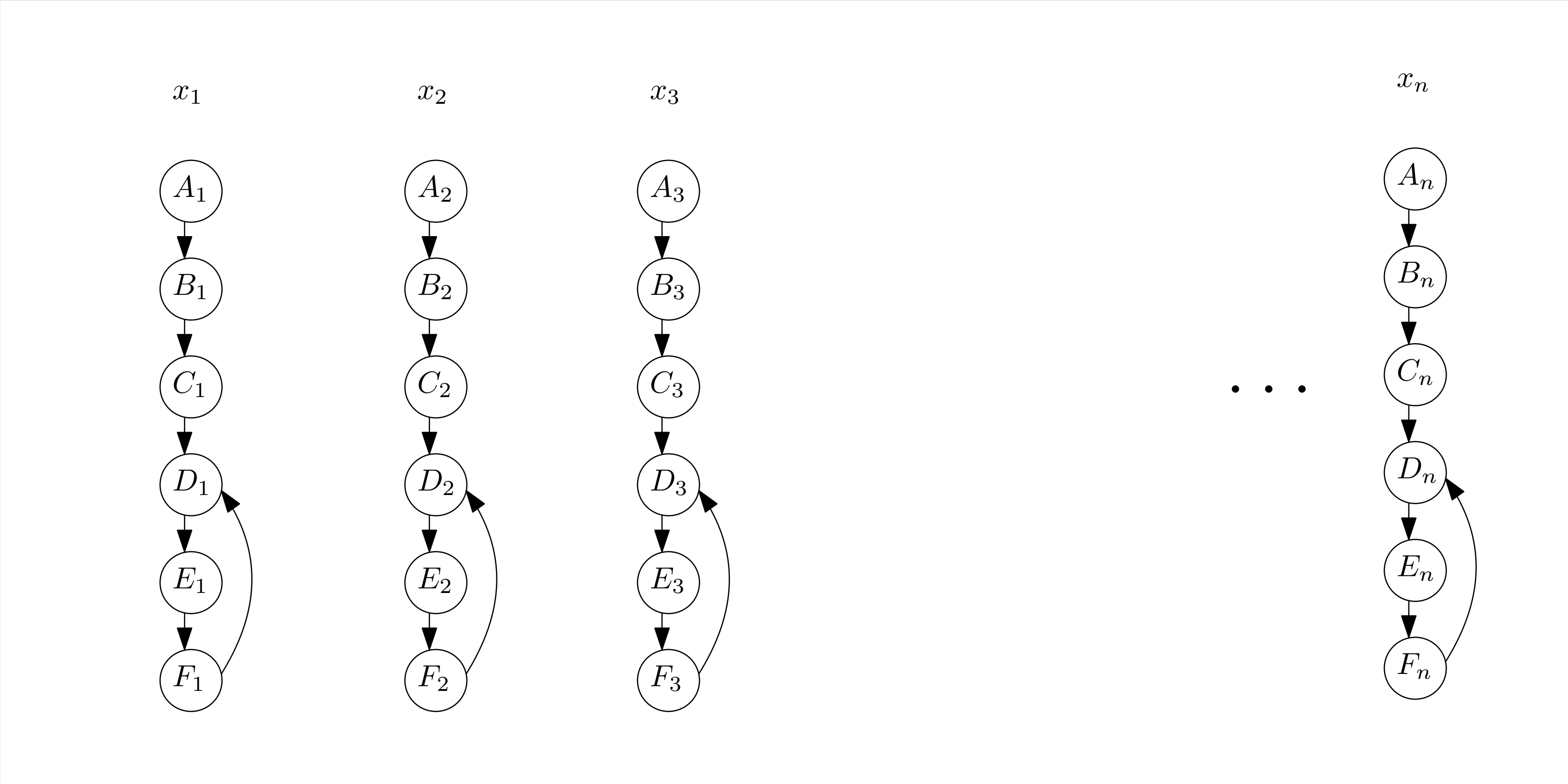
- Problem with interesting history.
- FAS is easily NP-complete (from VC$\rightarrow$FVS).
- Whether still NP-complete on **Tournaments** open for a long time!
  - Conjectured NP-complete by [Bang-Jensen, Thomassen, SIDMA'92]
  - **Almost** proved (via randomized reduction) by [Ailon, Charikar, Newman, STOC'05]
  - **Proved** (derandomized) by [Alon SIDMA'06] and [Charbit, Thomassé, Yeo Comb. Prob. Comp. '06]
- **Reproved from scratch** by [Conitzer AAAI'06]!

Conitzer's reduction – Setup



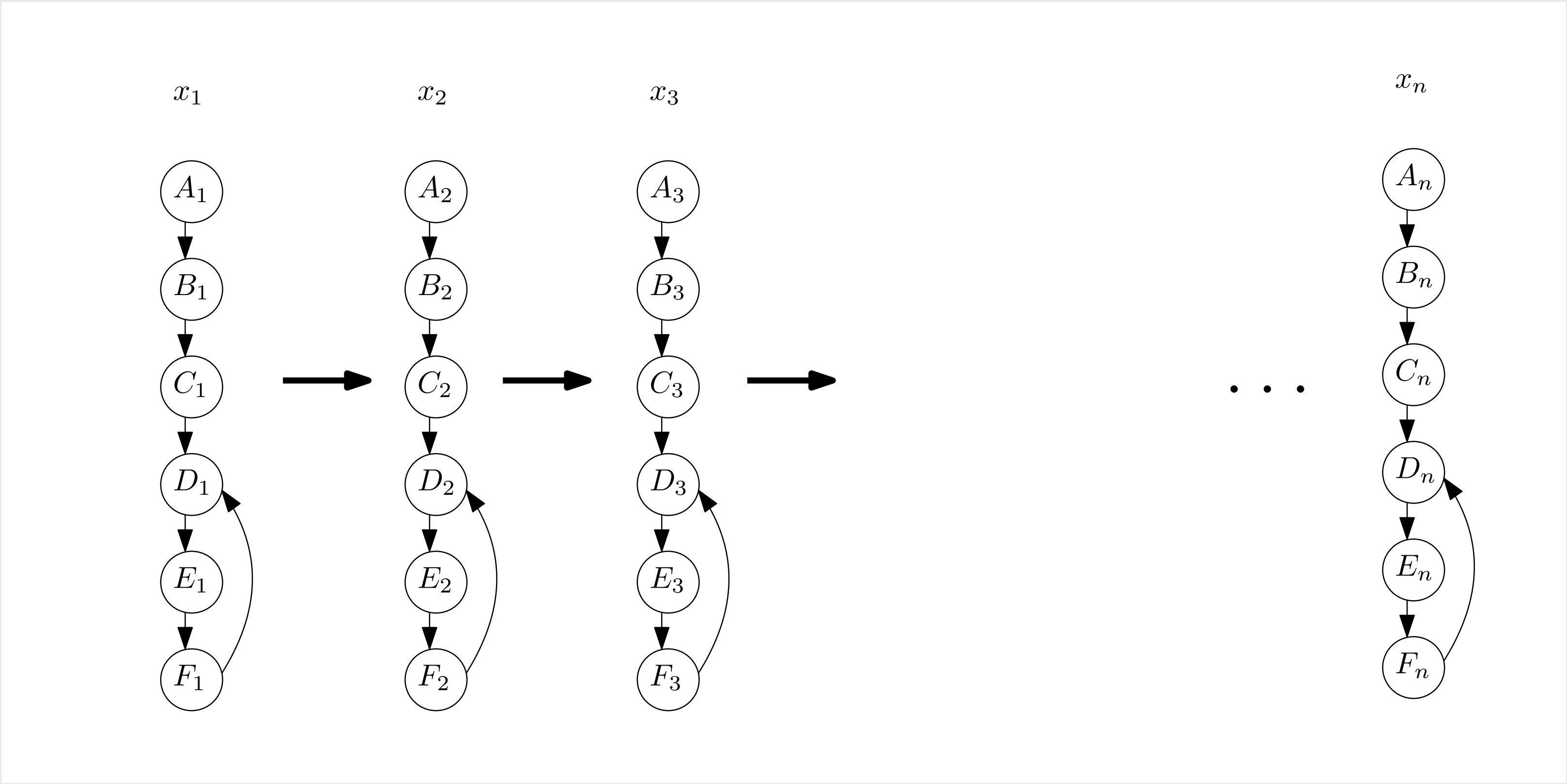
- Start from a 3-SAT formula with n variables.
- Make six **large** groups for each variable.

Conitzer's reduction – Setup



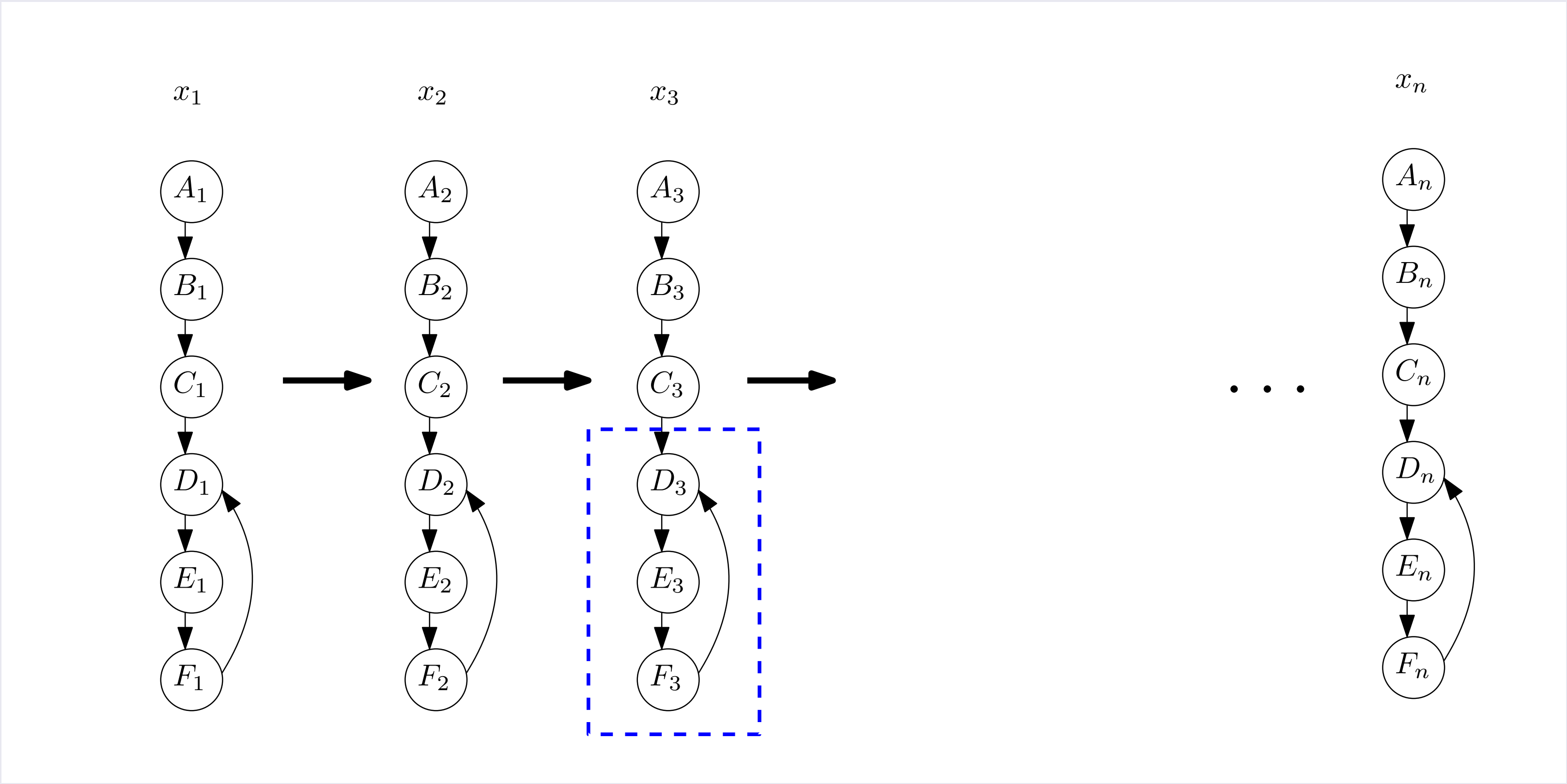
- Start from a 3-SAT formula with n variables.
- Make six **large** groups for each variable.

Conitzer's reduction – Setup



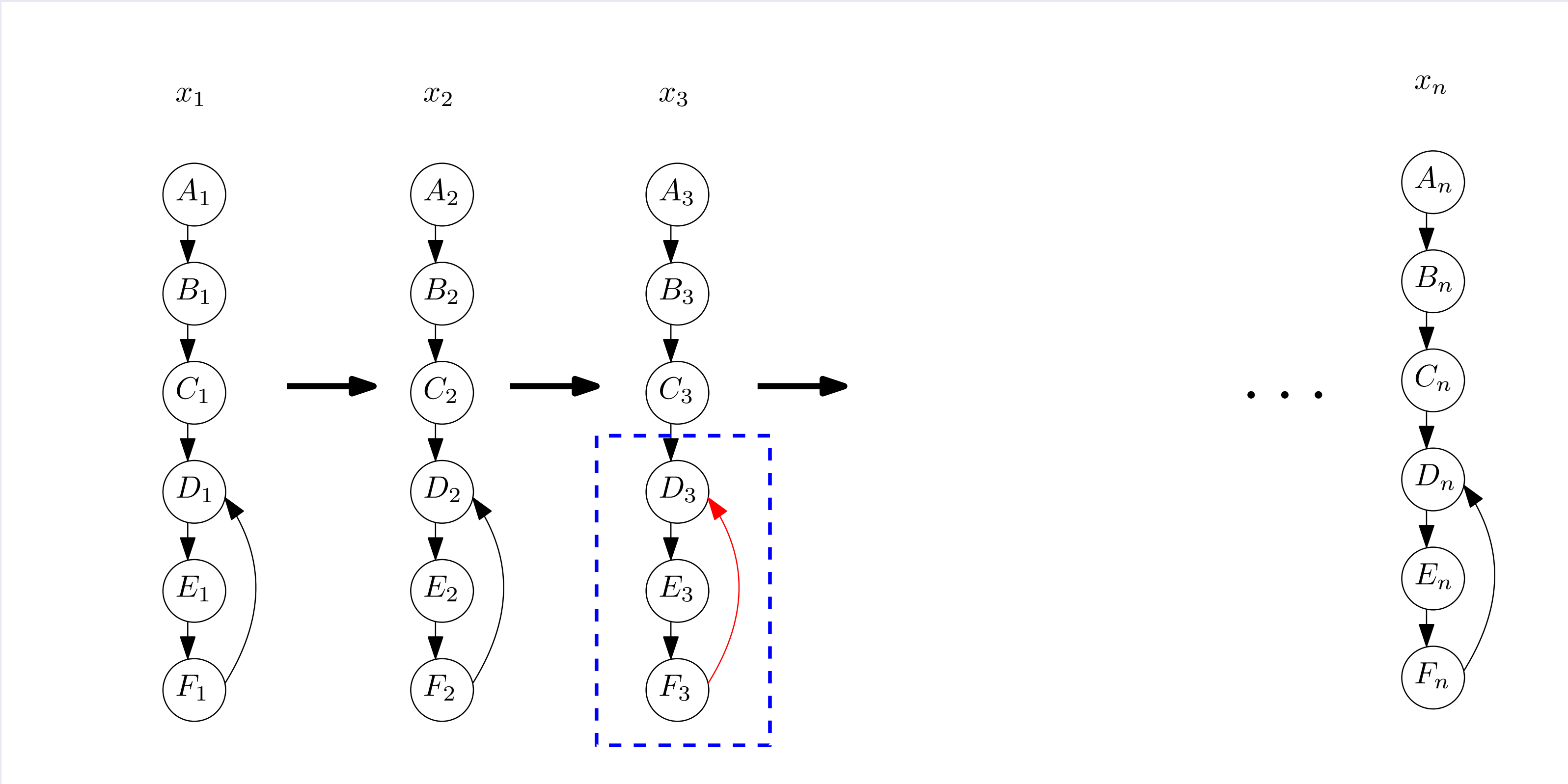
- Order variable groups linearly.
- Now we only have a choice **inside** each group.

Conitzer's reduction – Setup



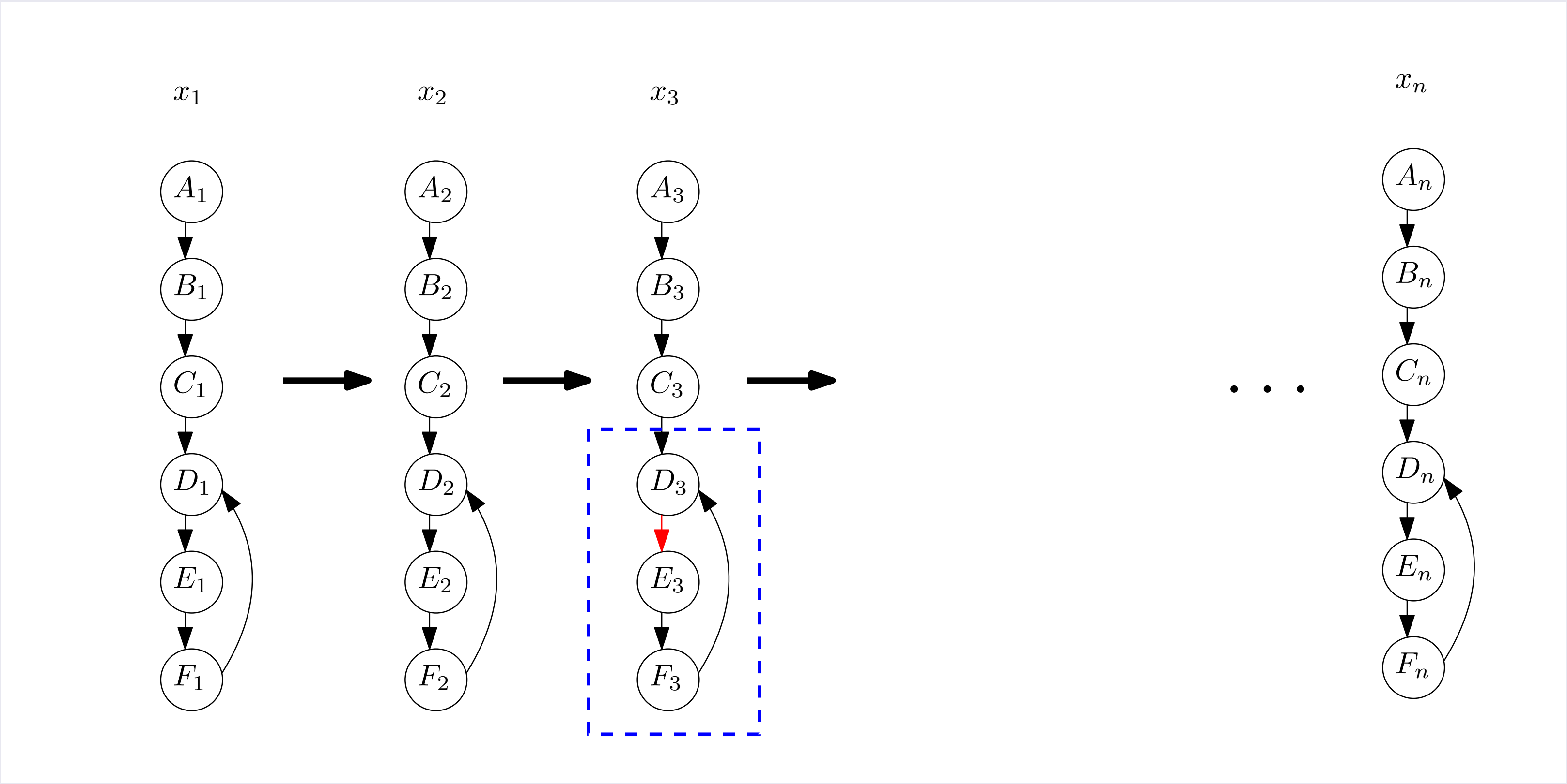
- Order variable groups linearly.
- Now we only have a choice **inside** each group.

Conitzer's reduction – Setup



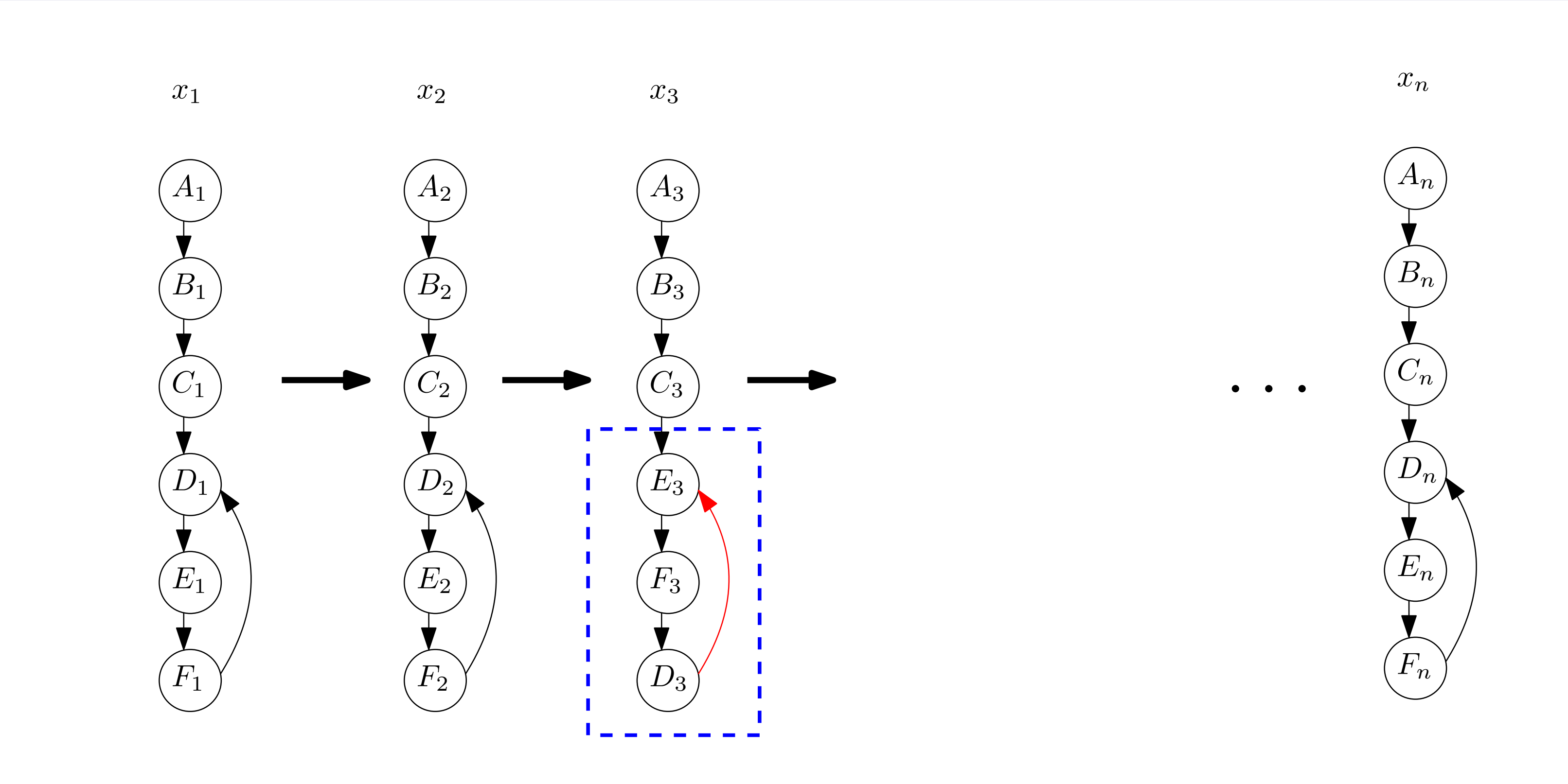
- Three reasonable choices.
  - $D \to E \to F$
  - $E \to F \to D$
  - $F \to D \to E$

Conitzer's reduction – Setup



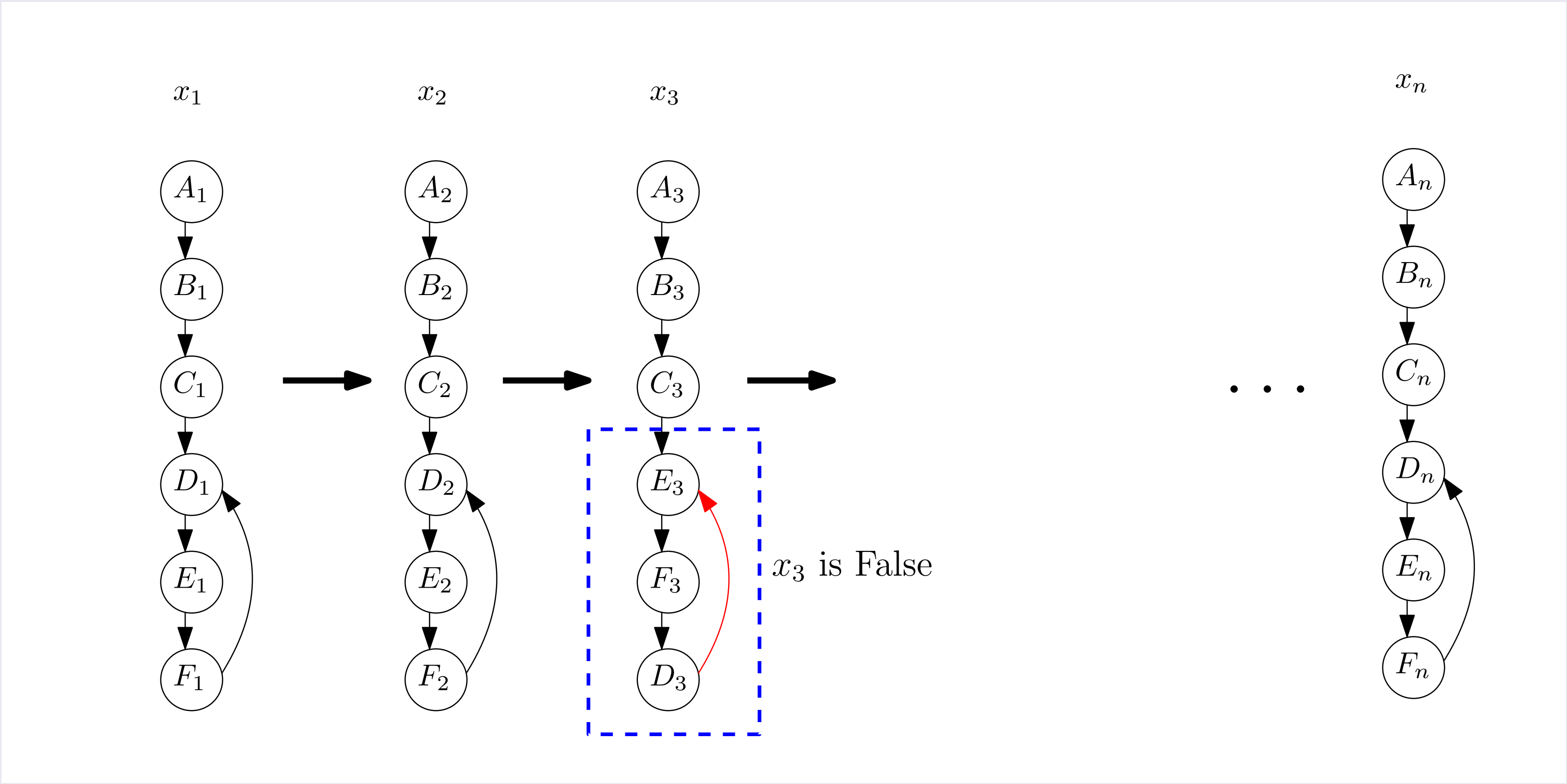
- Three reasonable choices.
  - $D \rightarrow E \rightarrow F$
  - $E \rightarrow F \rightarrow D$
  - $F \rightarrow D \rightarrow E$

Conitzer's reduction – Setup



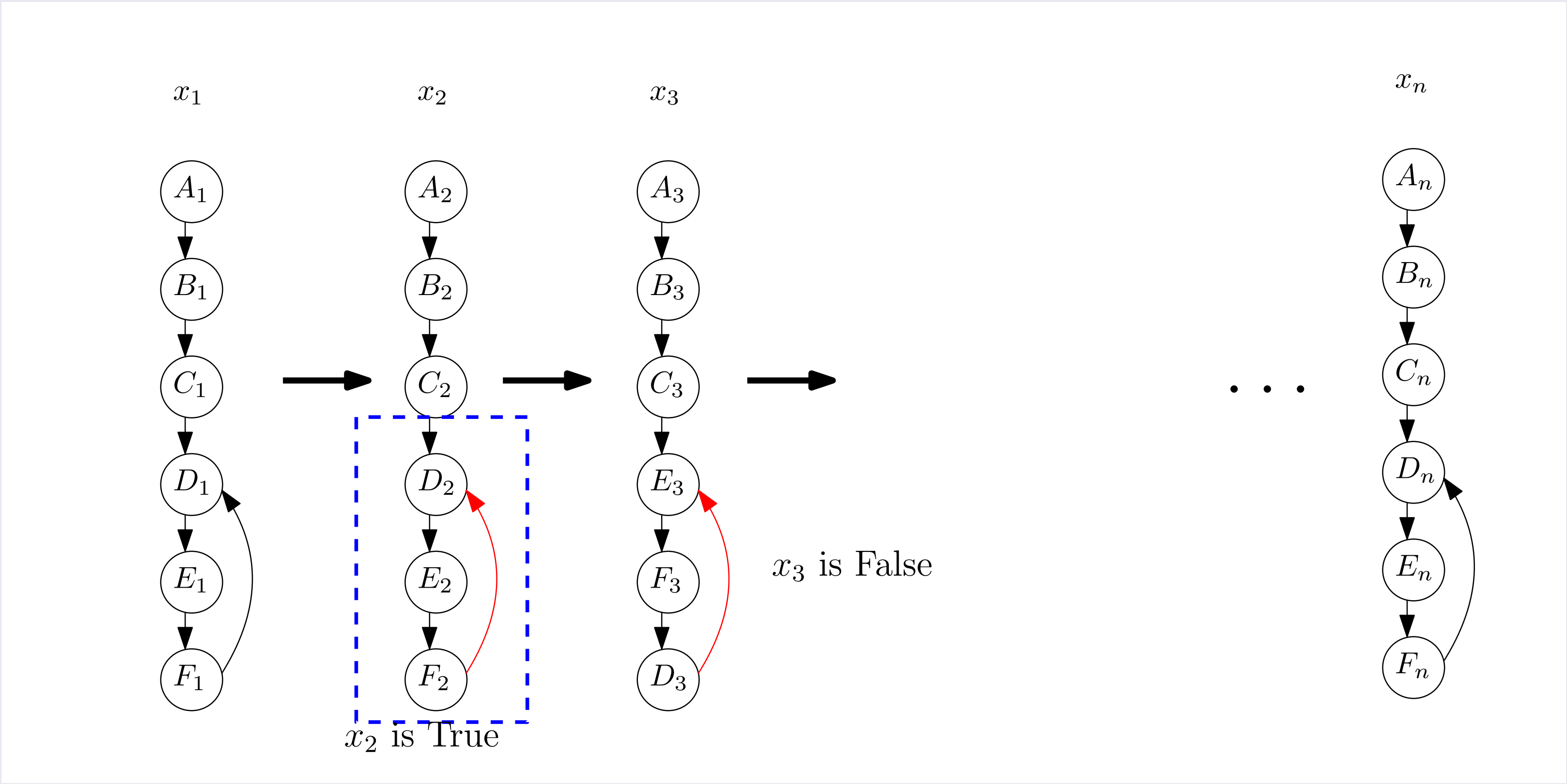
- Three reasonable choices.
  - $D \rightarrow E \rightarrow F$
  - $E \rightarrow F \rightarrow D$
  - $F \rightarrow D \rightarrow E$

Conitzer's reduction – Setup



- Convention:
  - $D \rightarrow E \rightarrow F$: Variable is True
  - $E \rightarrow F \rightarrow D$: Variable is False

Conitzer's reduction – Setup

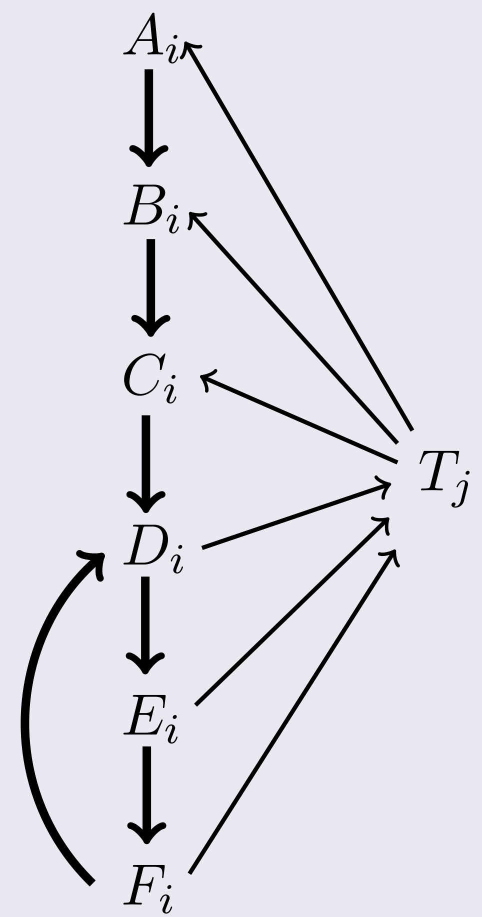


- Convention:
  - $D \to E \to F$: Variable is True
  - $E \to F \to D$: Variable is False

Conitzer's reduction – Validation

- Represent each clause with a vertex T_j
- Encode variable incidence via arcs to gadget

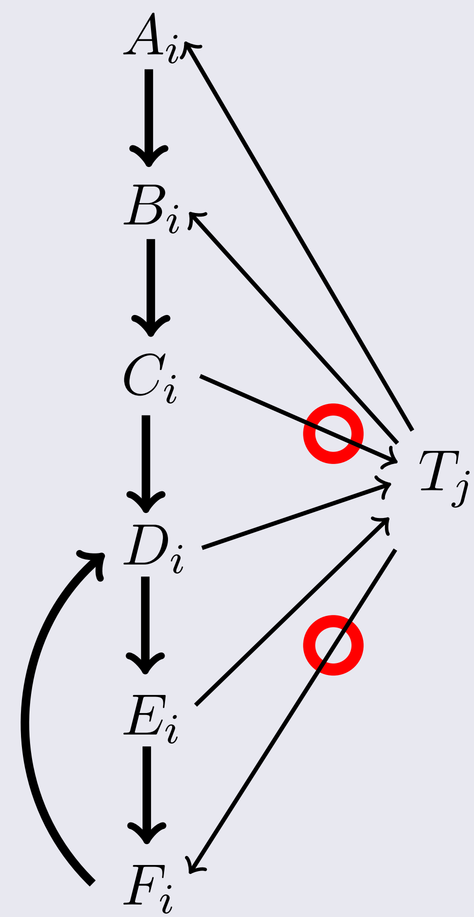
- Variable doesn't appear in clause
- Better to keep T_j before or after gadget.



Conitzer's reduction – Validation

- Represent each clause with a vertex T_j
- Encode variable incidence via arcs to gadget

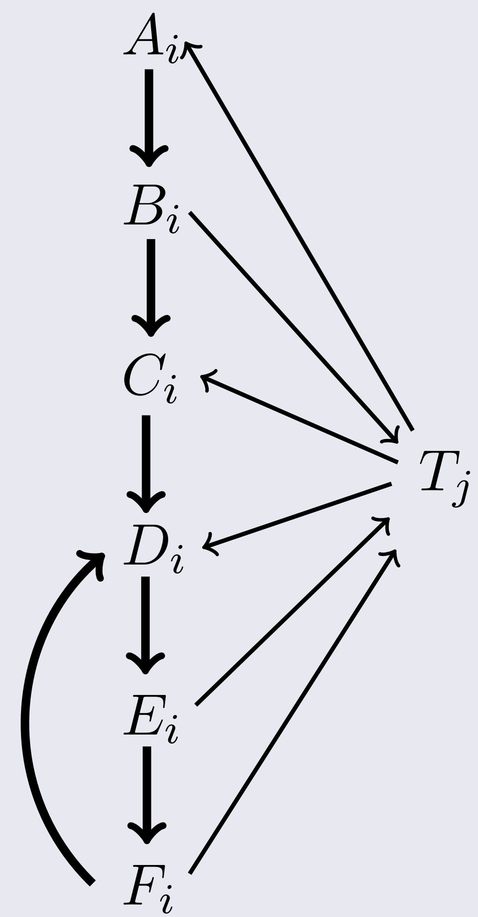
- Variable appears **positive** in clause
- Better to keep T_j **inside** gadget **right before** F, assuming F is last.



Conitzer's reduction – Validation

- Represent each clause with a vertex T_j
- Encode variable incidence via arcs to gadget

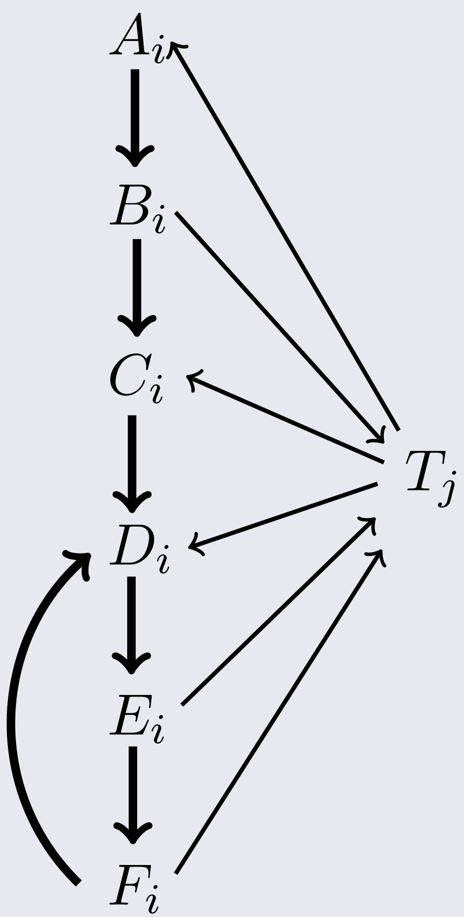
- Variable appears **negative** in clause
- Better to keep T_j **inside** gadget **right before** D, assuming D is last.



- Represent each clause with a vertex T_j
- Encode variable incidence via arcs to gadget

Remaining ideas:

- Variable groups are so large that:
  - Must respect variable structure.
  - Clause ordering is irrelevant. Only clause satisfaction matters.
- Formula satisfiable $\Leftrightarrow$ FAST$\leq k$



This reduction

New ideas to obtain Θ_2^p-completeness for SLATER WINNER

- Start reduction from MAX MODEL:
  - Input: CNF formula ϕ with a distinguished variable x_n
  - Question: Is there a **Maximum Weight** satisfying assignment of ϕ that sets x_n to True?
  - Prototypical Θ_2^p-complete problem.

This reduction

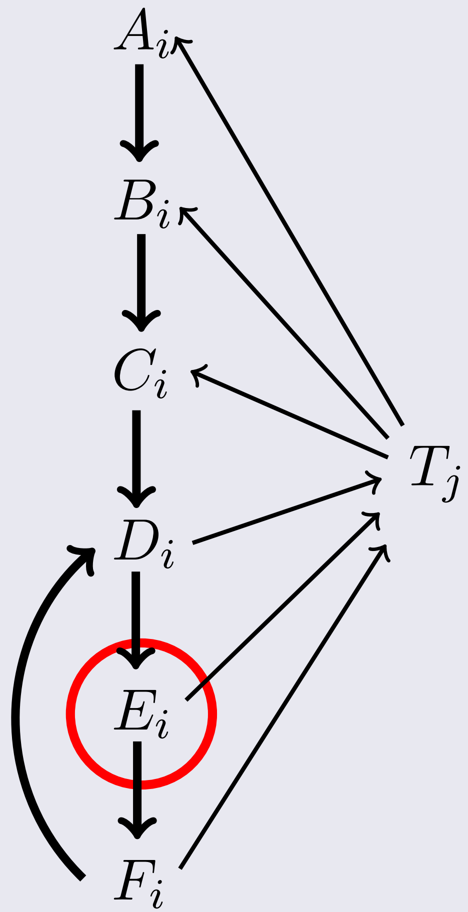
New ideas to obtain Θ_2^p-completeness for SLATER WINNER

- Start reduction from MAX MODEL:
  - Input: CNF formula ϕ with a distinguished variable x_n
  - Question: Is there a **Maximum Weight** satisfying assignment of ϕ that sets x_n to True?
  - Prototypical Θ_2^p-complete problem.
- Modify reduction so that:
  - Assignment weight is taken into account. More True variables $\Rightarrow$ smaller FAS
  - Setting x_n to True is more important than setting another variable to True…
  - …but less important than setting **two** other variables to True.

Dauphine | PSL UNIVERSITÉ PARIS

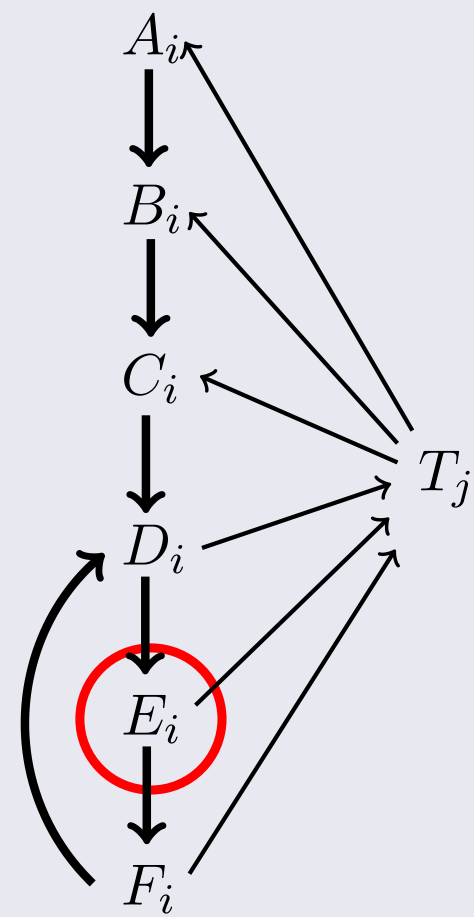
This reduction continued

- Main idea: add 2 vertices to group E_i
  - Arc $F \to D$ is now less heavy than $D \to E$ and $E \to F$
  - $\Rightarrow$ slightly better FAS if we order $D \to E \to F$
  - This corresponds to x_i set to True

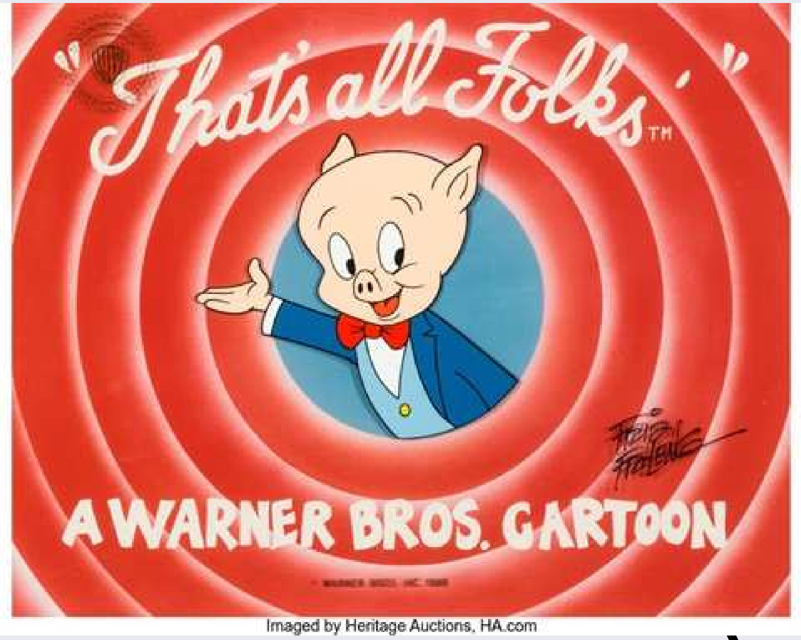


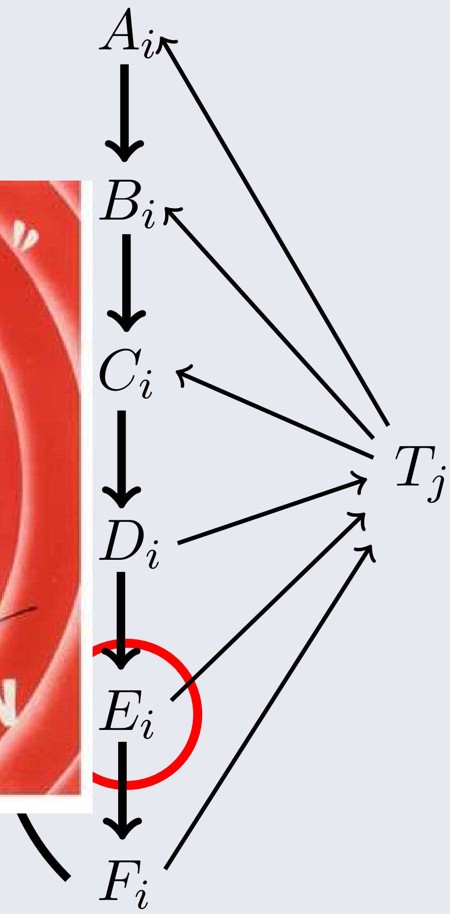
This reduction continued

- Main idea: add 2 vertices to group E_i
- For the group of x_n add 3 vertices to E_n
  - Setting x_n to True is more important than one other variable, less important than two others.
- Optimal FAS $\Leftrightarrow$ Max Weight Sat assignment which sets x_n to True if possible.
- Slater winner reflected in configuration for x_n.



- Main idea: add 2 vertices to group E_i
- For the group of x_n add 3 vertices to E_n
    - Setting x_n to
      than one oth
      tant than two
- Optimal FAS $\Leftrightarrow$
  ment which sets
- Slater winner refl
  x_n.





Conclusions

Conclusions

- Slater is **another** election system complete for Θ_2^p
  - This class seems to nicely capture key ideas in social choice!
- Strengthening: still Θ_2^p-complete for 7 voters!
  - Following ideas of [Bachmeier et al. JCSS'19]
- Open problem:
  - What about 3 or 5 voters?

Conclusions

- Slater is **another** election system complete for Θ_2^p
  - This class seems to nicely capture key ideas in social choice!
- Strengthening: still Θ_2^p-complete for 7 voters!
  - Following ideas of [Bachmeier et al. JCSS'19]
- Open problem:
  - What about 3 or 5 voters?

Thank you!