Computational Social Choice and Incomplete Information

Phokion G. Kolaitis

UC Santa Cruz and IBM Research

Social Choice Theory

Definition:

"Social choice theory is the study of collective decision processes and procedures."

Stanford Encyclopedia of Philosophy – 2013

Selected Themes:

- How can individual votes, preferences, or judgments be aggregated into a collective (societal) output?
- What are the properties of different voting systems?
- When is non-dictatorial aggregation possible?

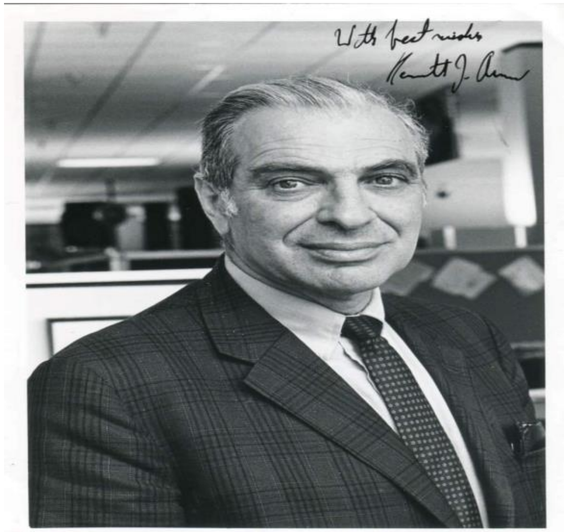
  (when is it the case that no individual voter can impose their preferences on the society?)

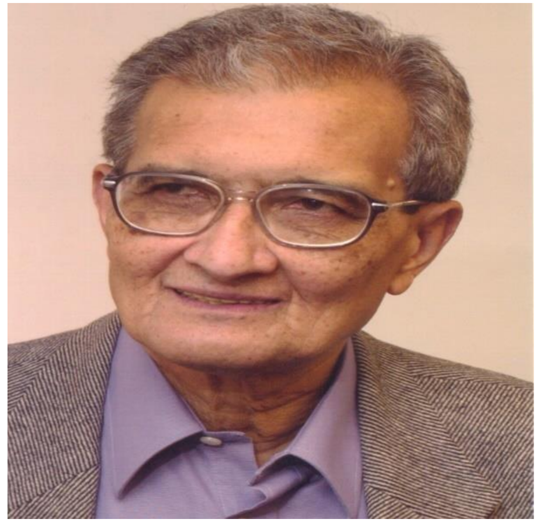
Very Brief History of Social Choice Theory

- Ramon Lull (1235-1315)
  Ars Electionis – pairwise majority voting

- Jean-Charles de Borda (1733-1799)
  Ranked preferential voting system – the Borda count

- Marquis de Condorcet (1743-1794)
  - A variant of pairwise majority voting
  - Discovered Condorcet’s paradox





Very Brief History of Social Choice Theory

- Kenneth Arrow (1921-2017)
  Arrow’s Impossibility Theorem

- Amartya Sen (1933 --)
  Social Choice and Welfare

- Eric Maskin (1950 --)
  Mechanism Design

Computational Social Choice

Definition:

Computational social choice is the study of algorithmic aspects of social choice theory.

- Meeting point of computer science, economics, social welfare

Selected Themes:

- Complexity of winner determination in elections
- How easy or difficult is it to manipulate an election?
- How to cope with uncertainty or incomplete information in voter preference?

Handbook of Computational Social Choice

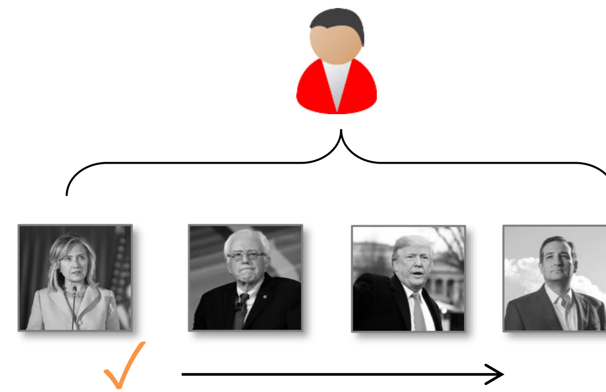
Edited by

F. Brandt, V. Conitzer, U. Endriss, J. Lang, A.D. Procaccia

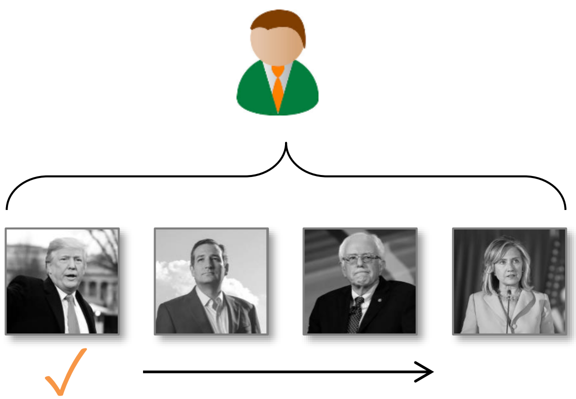
Cambridge University Press 2016, 529 pages.

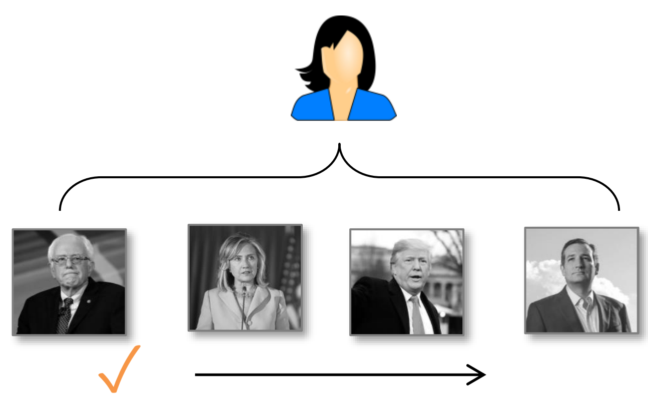
Elections

Candidates



Voters





Formal Model of Voting Rules

- Candidates: $c_1,\dots,c_m$;   Voters: $v_1,\dots,v_n$
- The preference of each voter is a linear order of the candidates
- A (preference) profile is a vector $(\succ_1,\dots,\succ_n)$ of linear orders over the candidates cast by the voters $v_1,\dots,v_n$
- A voting rule maps the profile to a set of winners
- Example: Positional scoring rule

v_i $\quad c_{i_1} \succ_i c_{i_2} \succ_i c_{i_3} \succ_i c_{i_4} \succ_i c_{i_5} \succ_i \cdots \succ_i c_{i_m}$

$s_1 \geq s_2 \geq s_3 \geq s_4 \geq s_5 \geq \cdots \geq s_m$

Winners: Candidates with maximum total score

Examples of Positional Scoring Rules

plurality

1	0	0	0	0	…	0
s_1	s_2	s_3	s_4	s_5		s_m

k-approval

1	…	1	0	0	…	0

(first k entries are 1: k)

veto

1	…	1	1	1	1	0

k-veto

1	…	1	1	0	…	0

(last k entries are 0: k)

Borda count

m-1	m-2	…	1	0

voter v_j

$$c_{j_1} \succ_j c_{j_2} \succ_j c_{i_3} \succ_j c_{j_4} \succ_j c_{j_5} \succ_j \dots \succ_j c_{i_m}$$

$$s_1 \geq s_2 \geq s_3 \geq s_4 \geq s_5 \geq \dots \geq s_m$$

Beyond Political Elections – Example 1

Eurovision Song Contest

- Scoring Vector

s = (12,10,8,7,6,5,4,3,2,1,0,…,0)

- The candidates are the songs
- The voters are the judges

Beyond Political Elections – Example 2

Formula One World Championship

- 21-23 races per year (Grands Prix)
- Scoring Vector

  s = (25, 18, 15, 12, 10, 8, 6, 4, 2, 1,0, …,0)
- The candidates are the drivers
- The voters are the races

Positional Scoring Rules- Recap

- Candidates $c_1,\ldots,c_m$ and Voters $v_1,\ldots,v_n$
- A preference profile is a vector $(\succ_1,\ldots,\succ_n)$ of linear orders over the candidates by the voters $v_1,\ldots,v_n$
- A positional scoring rule is a sequence of scoring vectors

  (one vector for each number of candidates)
  - A scoring vector of length m is a sequence

    $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_m$ of m natural numbers .
  - Voter v_j scores the candidates according to their position in the linear order $\succ_j$ of voter v_j.
- The scores each candidate receives are added up
- The winners are those getting a maximum sum of scores

Assumption about Positional Scoring Rules

A positional scoring rule r is a sequence $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m, \ldots$ of scoring vectors such that

- $\mathbf{r}_m = (s_1, s_2, \cdots, s_m)$, where $s_1, s_2, \cdots, s_m$ are natural numbers with $s_1 \geqslant s_2 \geqslant \cdots \geqslant s_m$, gcd = 1, and $s_1 > s_m = 0$.
- The function $m \to \mathbf{r}_m$ is efficiently computable.
- The scoring vector $\mathbf{r}_{m+1}$ is obtained from the scoring vector $\mathbf{r}_m$ by inserting a score in some position (purity property).

Incomplete Preferences

Fact: The preferences of voters may be incomplete

Question: How can incompleteness be modeled?

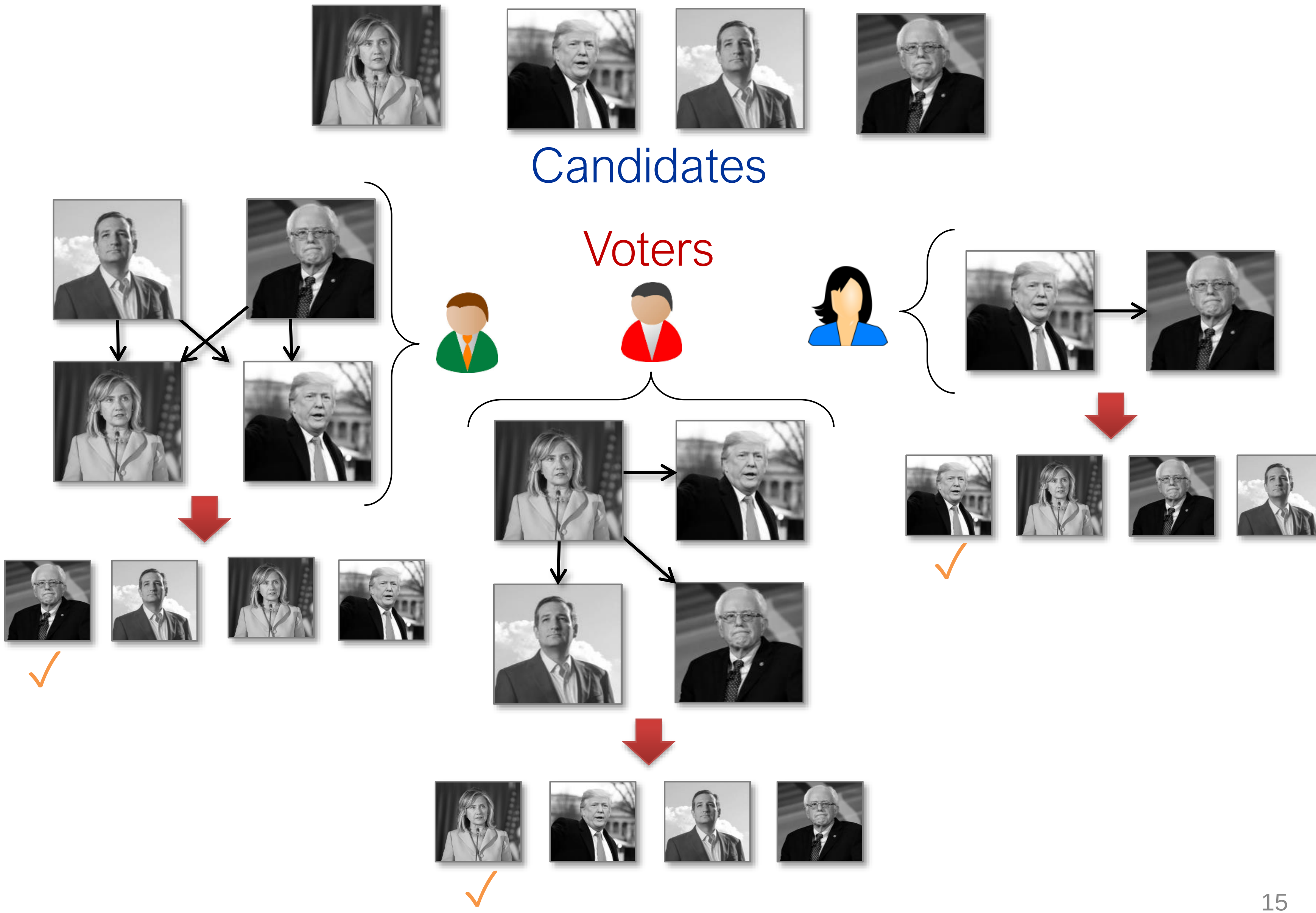
Answer: Use partial orders on the set of candidates

- Each voter casts a partial order
- Partial preference profile: vector $(\succ_1, \ldots, \succ_n)$ of partial orders over the candidates cast by the voters $v_1,\ldots,v_n$

Definition:

- A completion of a partial order $\succ$ is a linear order $\succ^*$ that extends the partial order $\succ$
- A completion of a partial preference profile is obtained by completing each partial order $\succ_j$ to a linear order $\succ^*_j$
  - Thus, $(\succ^*_1, \ldots, \succ^*_n)$ is a (complete) preference profile.

Completions of Incomplete Preferences



Necessary Winners & Possible Winners

- Partial preference profile: vector $(\succ_1, \ldots, \succ_n)$, where each $\succ_j$ is a partial order over the candidates.
- A completion of a partial preference profile $(\succ_1, \ldots, \succ_n)$ is a profile $(\succ^*_1, \ldots, \succ^*_n)$ obtained by completing each $\succ_j$ to a linear order $\succ^*_j$

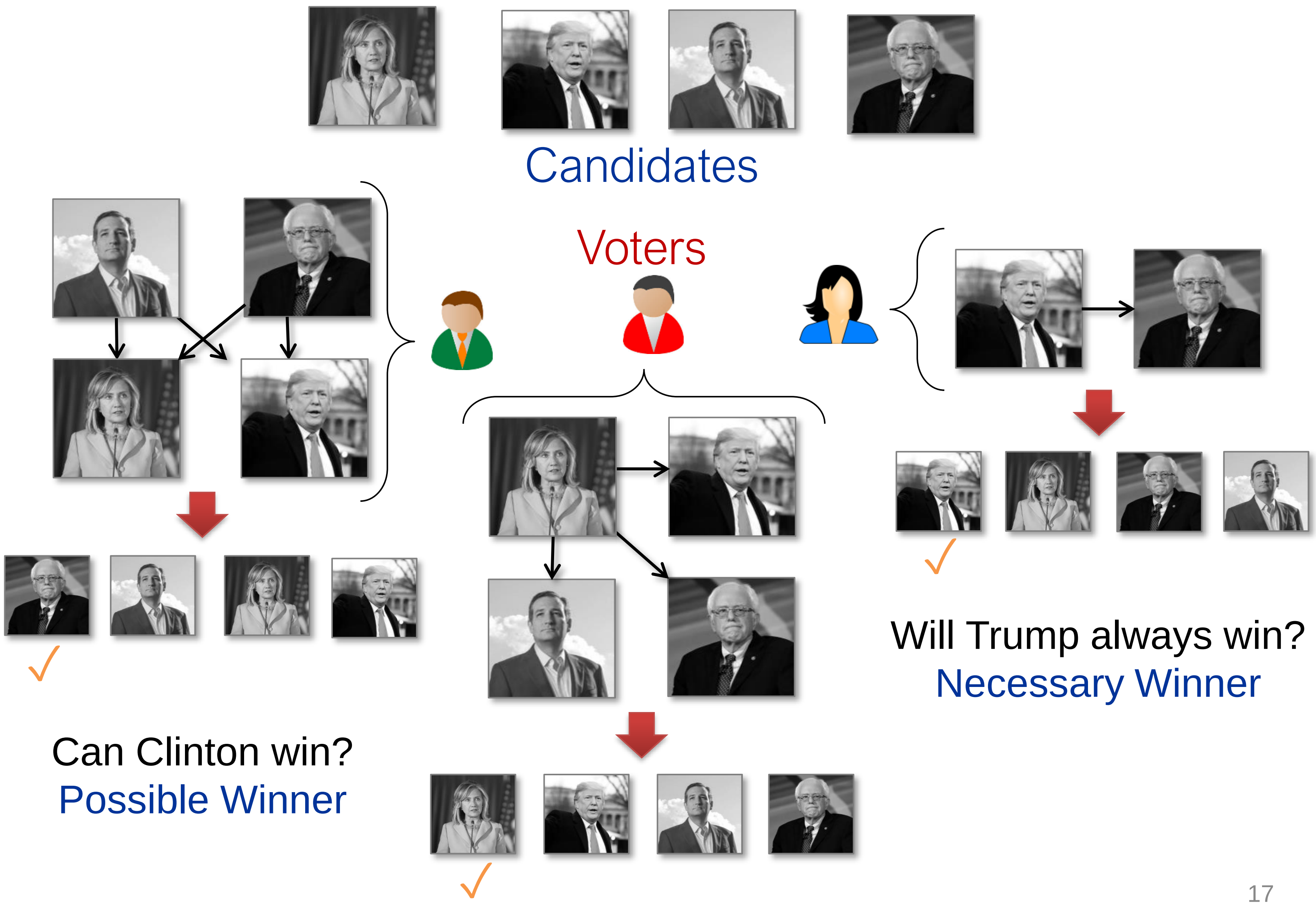
Fact: A partial profile may have exponentially many completions

Definition: Konczak & Lang - 2005

Given a partial preference profile **P**, a candidate c is a

- necessary winner if c wins in every completion;
- possible winner if c wins in at least one completion.

Necessary Winners & Possible Winners



Algorithmic Problems

Fix a positional scoring rule r

- The Necessary Winner Problem (NW) with respect to r
  Given a partial preference profile P and a candidate c, is c a necessary winner?
- The Possible Winner Problem (PW) with respect to r
  Given a partial preference profile P and a candidate c, is c a possible winner?

Question:

- Are there "good" algorithms for these decision problems?
- Can we avoid exhaustive search over all completions?

The Complexity of Necessary & Possible Winners

Konczak-Lang [2005], Betzler-Dorn [2010],
Xia-Conitzer [2011], Baumeister-Rothe [2012]

Classification Theorem

- The Necessary Winner Problem is in P, for every positional scoring rule.
- The Possible Winner Problem
  - is in P for the plurality rule and the veto rule;
  - is NP-complete for every other positional scoring rule.

the price of incompleteness

Social Choice in Context

- Elections and polls take place in a context
- There is information about candidates:
  - age, gender, education, net worth, position on issues, …
- There is information about voters:
  - age, gender, education, occupation, …
- Voters may have partial preferences:
  - They may be undecided between some candidates.

Definition: An election database is a relational database in which (partial) preferences of voters are incorporated.

Candidates

cand	party	net w	spouse
Clinton	D	$45M	Bill
Trump	R	$1.3B	Melania
Cruz	R	$3.5M	Heidi

BornIn

person	born
Clinton	Chicago
Trump	NYC
Cruz	Calgary

Voters

voter	age
Susan	45
David	62
James	29

Pref

poll	voter	partial preference
p1	Susan	Cl $\succ$ Tr , Cr $\succ$ Tr
p1	David	Tr $\succ$ Cr $\succ$ Cl
p1	James	Cl $\succ$ Tr

An Election Database

poll	voter	preference
p1	Susan	Cl $\succ$ Cr $\succ$ Tr
p1	David	Tr $\succ$ Cr $\succ$ Cl
p1	James	Cr $\succ$ Cl $\succ$ Tr

poll	voter	preference
p1	Susan	Cr $\succ$ Cl $\succ$ Tr
p1	David	Tr $\succ$ Cr $\succ$ Cl
p1	James	Cl $\succ$ Cr $\succ$ Tr

poll	voter	preference
p1	Susan	Cl $\succ$ Cr $\succ$ Tr
p1	David	Tr $\succ$ Cr $\succ$ Cl
p1	James	Cl $\succ$ Tr $\succ$ Cr

Completions
of
partial preferences

Election Databases

Question 1:

What is the semantics of queries posed against an election database?

Question 2:

What is the computational complexity of queries posed against an election database?

- Computational Social Choice Meets Databases
  Kimelfeld, K …, Stoyanovich - IJCAI 2018
- Query Evaluation in Election Databases
  Kimelfeld, K…, Tibi – PODS 2019

Examples of Queries

- Does a Republican always win?

$$q(\,):\ \exists x\,(\textsc{Winner}(x)\ \wedge \text{Party}(x,\text{'R'}))$$

- Which cities are guaranteed to have winners from?

$$q(x)\ :\ \exists y\,(\textsc{Winner}(y) \wedge \text{LivesIn}(y, x))$$

- Is there a winner of net worth < $1M?

$$q(\,):\ \exists x\,\exists w(\textsc{Winner}(x) \wedge \text{NetW}(x,w) \wedge w<1M)$$

- Are there two winners who differ on the pro-choice issue?

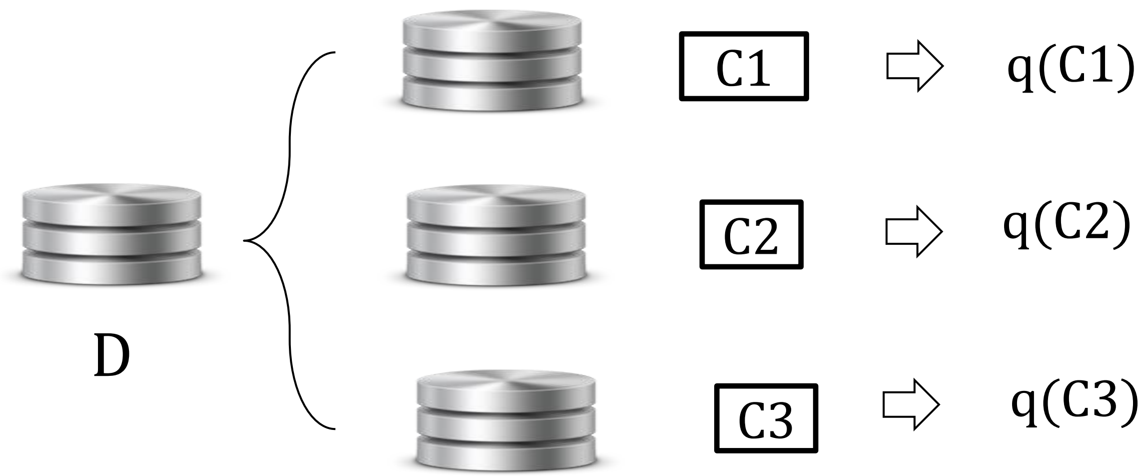
$$q(\,):\ \exists x\,\exists y\,(\textsc{Winner}(x) \wedge \textsc{Winner}(y) \wedge \text{Yes}(x,\text{'pc'}) \wedge \text{No}(y,\text{'pc'}))$$

Conjunctive Queries with Winner atom(s)

Necessary and Possible Answers to Queries

Definition: D a database, C partial profile, q a query that may involve the WINNER relation.

- A necessary answer to q is a tuple that belongs to q(C) for every completion C of D.
- A possible answer to q is a tuple that belongs to q(C) for at least one completion C of D.



$NA(q) = \cap\ q(Cj)$

$PA(q) = \cup\ q(Cj)$

Examples of Queries

- Does a Republican always win? [necessary]

  $$q(\,):\ \exists x\ (\mathrm{WINNER}(x) \wedge \mathrm{Party}(x,'R'))$$

- Which cities are guaranteed to have winners from?

  $$q(x):\ \exists y\ (\mathrm{WINNER}(y) \wedge \mathrm{LivesIn}(y,x))$$ [necessary]

- Is there a winner of net worth < $1M? [possible]

  $$q(\,):\ \exists x\ \exists w(\mathrm{WINNER}(x) \wedge \mathrm{NetW}(x,w) \wedge w<1M)$$

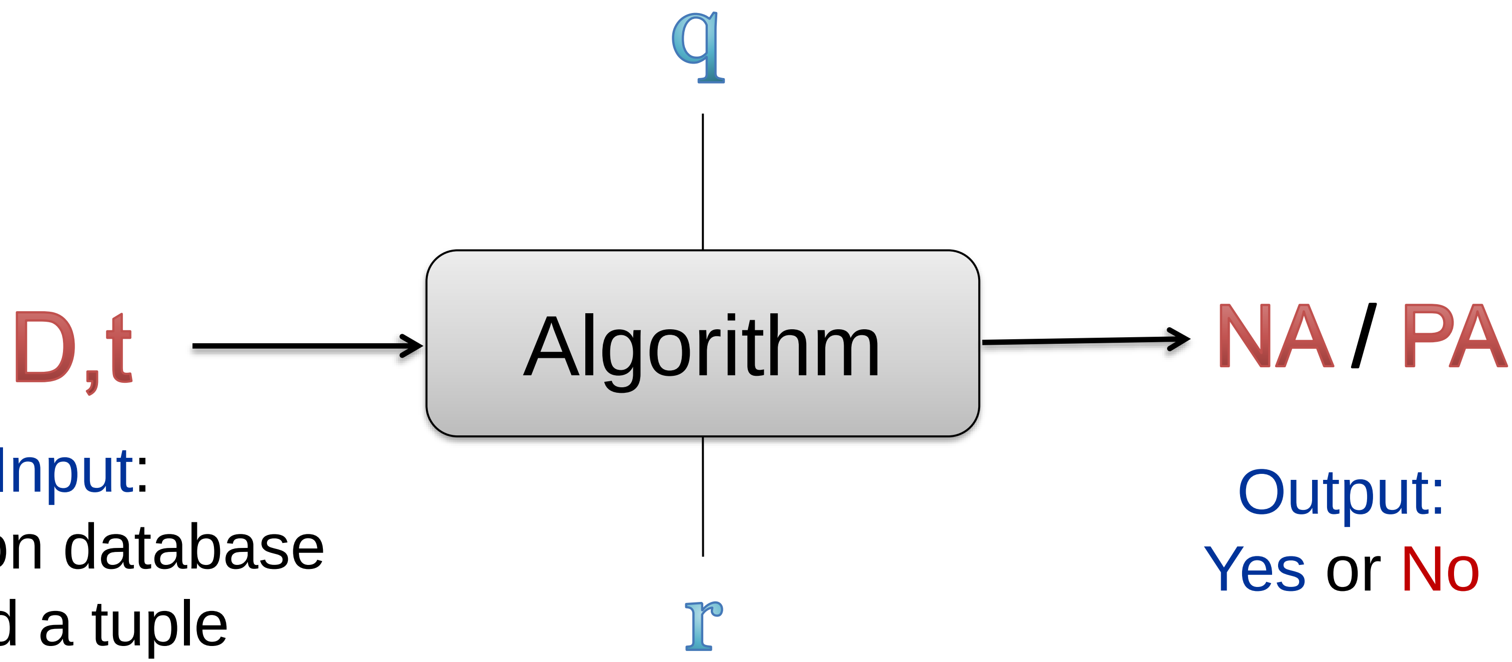
- Are there two winners who differ on the pro-choice issue?

  $$q(\,):\ \exists x\ \exists y\ (\mathrm{WINNER}(x) \wedge \mathrm{WINNER}(y) \wedge \mathrm{Yes}(x,'pc') \wedge \mathrm{No}(y,'pc'))$$

  [possible]

Necessary & Possible Answers: Data Complexity

Fixed Query



Input:
Election database and a tuple (partial profile + relations + tuple)

Output:
Yes or No

Fixed Voting Rule

Each pair (q, r) gives rise to two decision problems: NA and PA

Conjunctive Queries

Definition: A conjunctive query (CQ) is of the form

$$q(\mathbf{x}):\ \exists \boldsymbol{y}\ [\ \varphi_1(\mathbf{x},\mathbf{y}) \wedge \cdots \wedge\ \varphi_k(\mathbf{x},\mathbf{y})\ ],$$

where each $\varphi_i(\mathbf{x},\mathbf{y})$ is a WINNER atom or an atom from the DB

Example:

- $q(x)\ :\ \exists y\ (\text{Winner}(y) \wedge \text{LivesIn}(y, x))$
- $q(\ ):\ \exists y\ (\text{Winner}(y)\ \wedge \text{Party}(y,\text{'R'}))$

Fact:

- CQs are FAQs; also known as Select-Project-Join queries
- CQs are directly supported in SQL via the SELECT … FROM … WHERE … clause

Necessary Answers of Conjunctive Queries

Theorem: The following hold for the plurality and the veto rules:

- If is q a conjunctive query whose WINNER atoms are pairwise disconnected, then the Necessary Answers of q are in P.
- If is q a conjunctive query with two connected WINNER atoms and no repeated ordinary relations, then the Necessary Answers of q are coNP-complete.

Note: Sharp contrast between NW and NA for plurality and veto

- Necessary Winners are in P
- Necessary Answers of CQs can be coNP-complete.

Necessary Answers under Plurality and Veto

q( ) :- WINNER(x) , WINNER(y) , Relative(x,y)

coNP-complete

q( ) :- WINNER(x) , WINNER(y) , Supp(x,i) , Opp(y,i)

coNP-complete

Connected Winner atoms

---

q( ) :- WINNER(x) , WINNER(y) , Lives(x,'NY') , Works(y,'DC')

P

q( ) :- WINNER(x) , WINNER(y) , Supp(x,'proC') , Opp(y,'proC')

P

Disconnected Winner atoms

Necessary Answers Beyond Plurality and Veto

Theorem:

- The Necessary Answers Problem for the query

  $$q: \exists x(\mathrm{Winner}(x) \land \mathrm{R}(x))$$

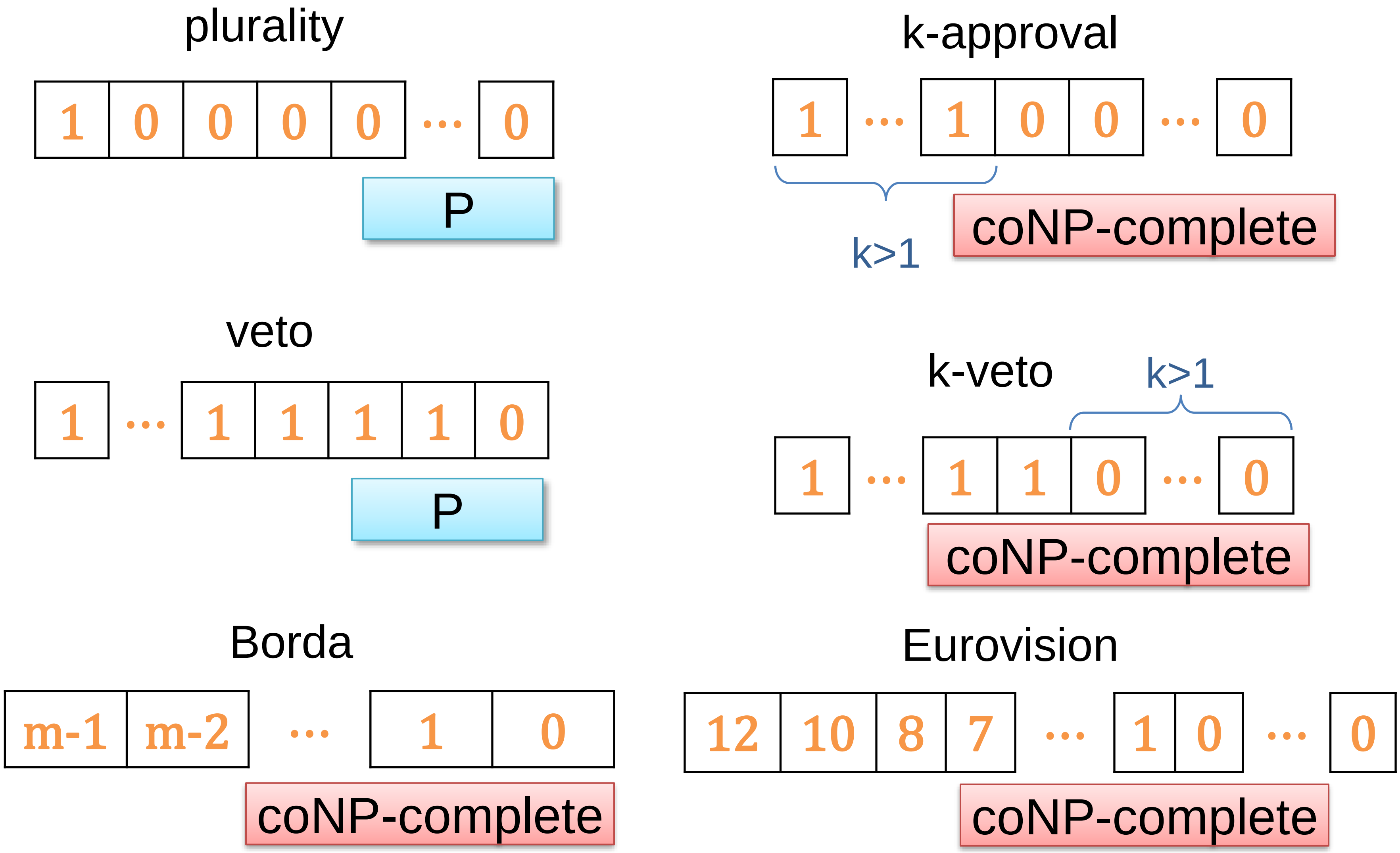
  is coNP-complete for every positional scoring rule other than plurality and veto.

- The Necessary Answers Problem for the query

  $$q: \exists x \exists y(\mathrm{Winner}(x) \land \mathrm{Winner}(y) \land \mathrm{T}(x,y))$$

  is coNP-complete for every positional scoring rule.

Necessary Answers of $\exists x\ (\mathrm{Winner}(\mathbf{x}) \wedge R(\mathbf{x}))$



Possible Answers for Plurality and Veto

- What can we say about the complexity of the Possible Answers to queries?
- Since the Possible Winner Problem is NP-complete for all positional scoring rules other than plurality and veto, the “best” we can hope for is the tractability for plurality and veto.

Theorem: For every conjunctive query q, the possible answers of q with respect to plurality and veto are in P.

Proof: Uses polynomial-time algorithms for polygamous matching, a generalization of the classical matching problem.

Computational Complexity Summary

	Necessary Winners	Necessary Answers
Plurality/Veto	Tractable	Tractable for disconnected CQs
		Hard for connected CQs
Other positional rules	Tractable	$\exists \mathbf{x}\ (\text{Winner}(\mathbf{x}) \wedge R(\mathbf{x}))$ Hard

	Possible Winners	Possible Answers
Plurality/Veto	Tractable	Tractable for CQs
Other positional rules	Hard	$\exists \mathbf{x}\ (\text{Winner}(\mathbf{x}) \wedge R(\mathbf{x}))$ Hard

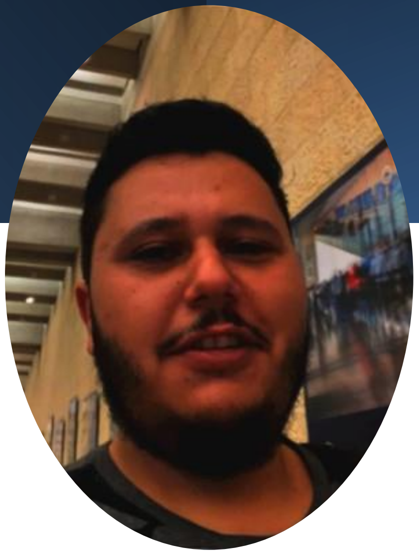
Concluding Remarks

- A new framework that augments computational choice with relational database context – interdisciplinary area of research
- From necessary/possible winners to necessary/possible answers to database queries.
- Take-home message:

  Context makes a difference, even for plurality and veto.

- Directions for future research:
  - Richer analysis: richer query languages; integrity constraints
  - Richer modeling: tie-breaking mechanisms; queries with multiple elections and/or multiple voting rules.
  - Approval voting (committee selection) and relational context.

Collaborators



Benny Kimelfeld
Technion

Julia Stoyanovich
New York University

Muhammad Tibi
Technion