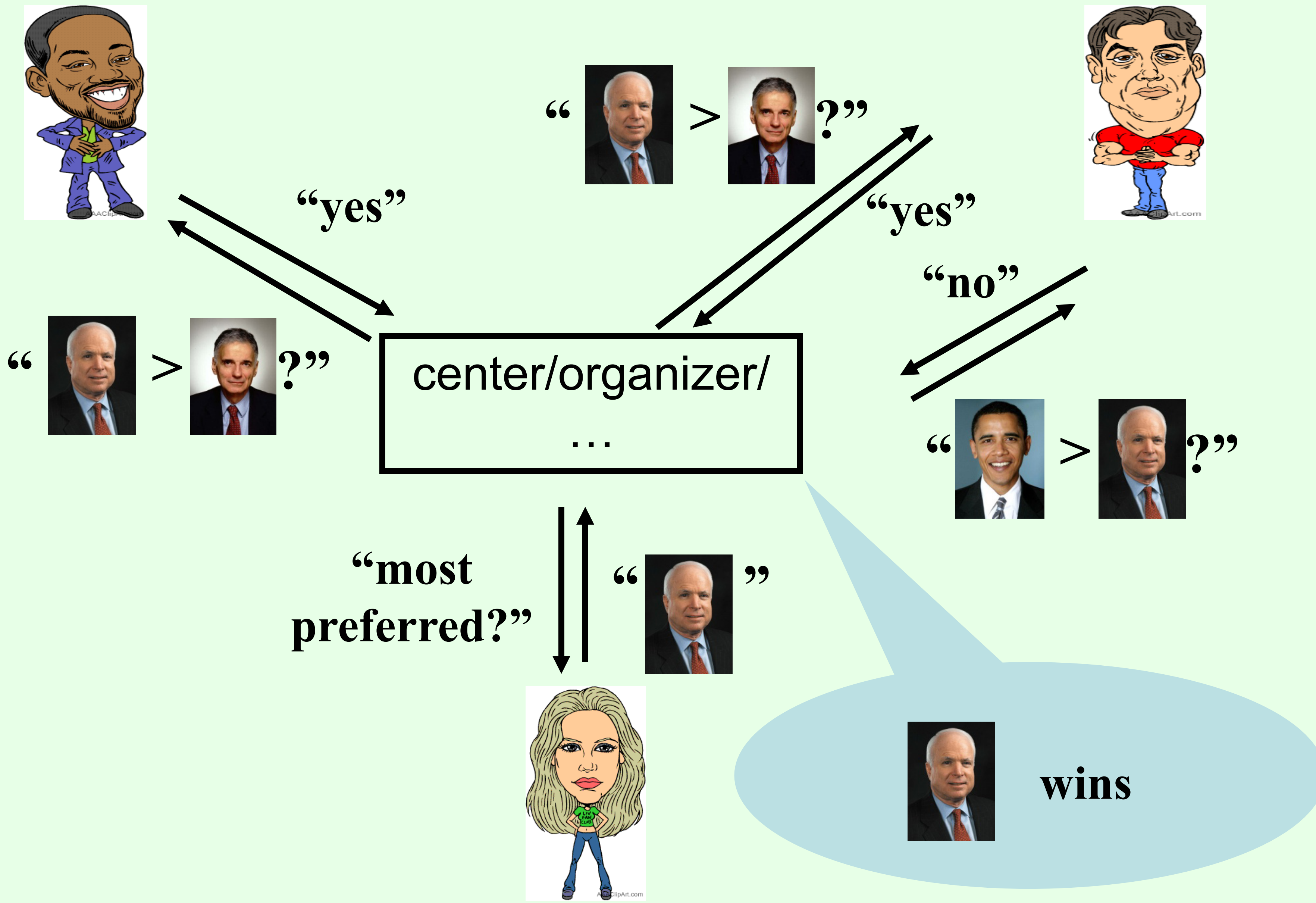
Preference elicitation

Vincent Conitzer
conitzer@cs.duke.edu

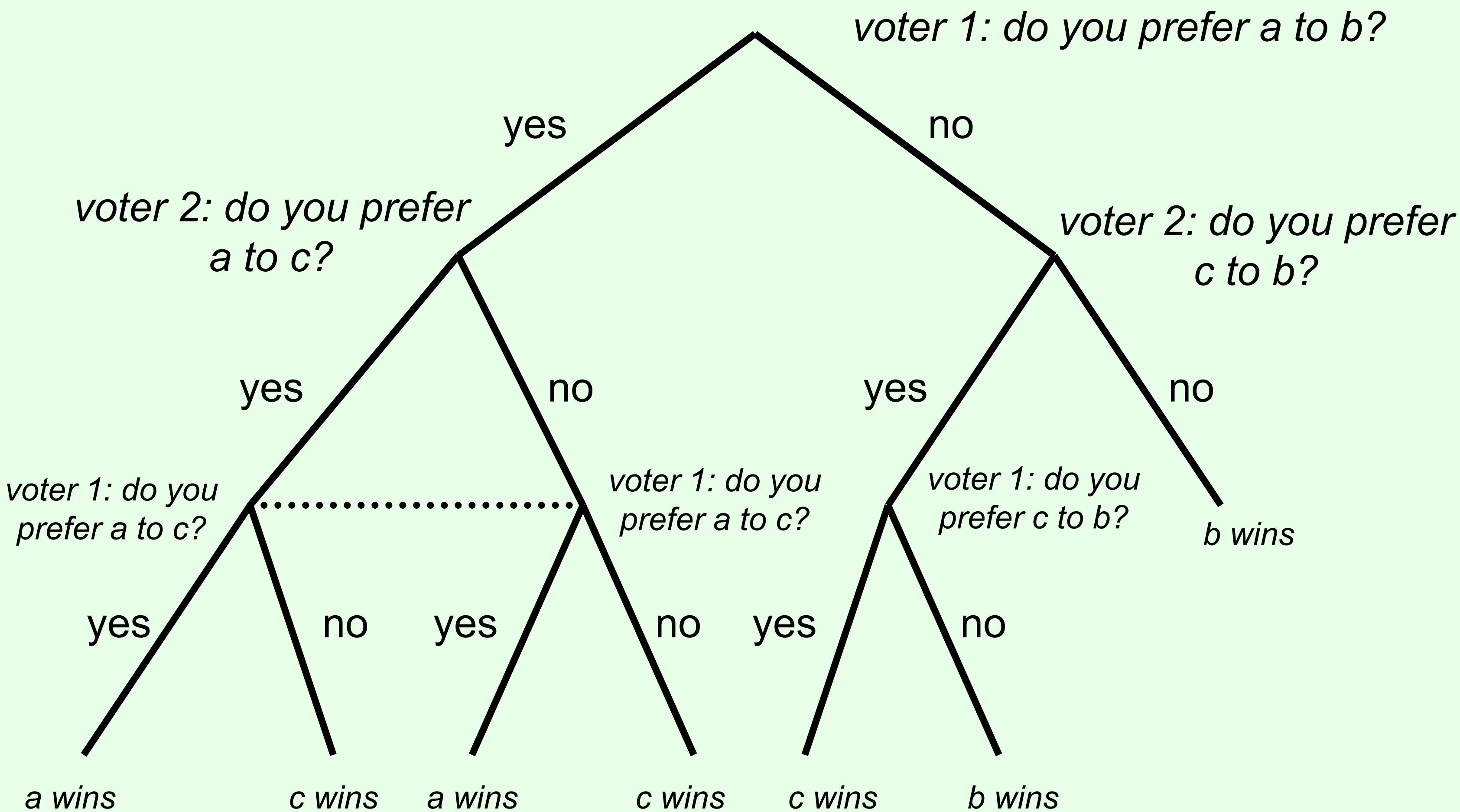
Excessive communication

- So far we have focused on revealing full preferences; often not practical
  - Rate/rank all the songs you've ever heard
  - Rate/rank all possible budget allocations
  - …
- Some ways to address this:
  - Directly model underlying structure
    - Directly express preferences over *aspects* of song (genre, instruments), *individual items* on the budget, …
    - Need to worry about interactions (voting in combinatorial domains)
  - Preference elicitation: ask voters selected queries about preferences, avoid asking for irrelevant information

Preference elicitation (elections)



Representing elicitation protocols



- Can choose to hide information from agents, but only insofar as it is not implied by queries we ask of them
- (What's inefficient about the above protocol?)

Strategic issues

- Consider the following protocol for eliciting a plurality winner with three voters

- Ask voter 1 for her most-preferred alternative t_1
- Ask voter 2 for his most-preferred alternative t_2
- If $t_1 = t_2$ then declare it the winner
- Otherwise, ask voter 3 if one of t_1 or t_2 is her most-preferred alternative
  - If yes, then declare that one the winner
  - If no, then toss a coin to decide between t_1 and t_2

- What opportunity for strategic manipulation does this provide voter 3?

Two possible goals

(Assume truthful reporting for the rest of this lecture)

1. Ask queries until we know the true winner under a given common voting rule (e.g., plurality)

2. Keep asking queries until we've found an alternative that is "good enough"

(allows us to get away with fewer queries)

(Note: 2 will still correspond to some voting rule)

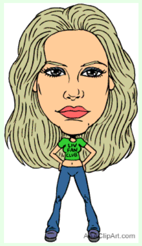
Elicitation algorithms for given rules

- Design elicitation algorithm to minimize queries for given rule
- What is a good elicitation algorithm for STV?
- What about Bucklin?

An elicitation algorithm for the Bucklin voting rule based on binary search

[C. & Sandholm 05]

- Alternatives: A B C D E F G H

- Top 4? {A B C D} {A B F G} {A C E H}
- Top 2? {A D} {B F} {C H}
- Top 3? {A C D} {B F G} {C E H}

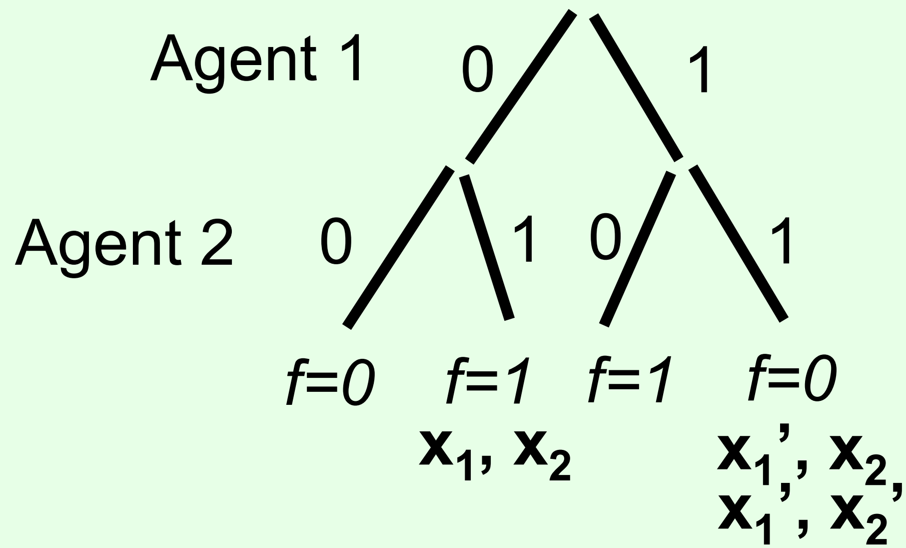
Total communication is nm + nm/2 + nm/4 + … ≤ 2nm bits
(n number of voters, m number of candidates)

Lower bounds on communication

- Communication complexity theory can be used to show lower bounds
  - "Any elicitation algorithm for rule r requires communication of at least N bits (in the worst case)"
- Voting [C. & Sandholm 05]
  - Bucklin requires at least on the order of nm bits
  - STV requires at least on the order of n log m bits
    - Natural algorithm uses on the order of $n(\log m)^2$ bits

How do we know that we have found the best elicitation protocol for a mechanism?

- Communication complexity theory: agent i holds input x_i, agents must communicate enough information to compute some $f(x_1, x_2, \ldots, x_n)$
- Consider the tree of all possible communications:
- Every input vector goes to some leaf



- If $x_1, \ldots, x_n$ goes to same leaf as $x_1', \ldots, x_n'$ then so must any mix of them (e.g., $x_1, x_2', x_3, \ldots, x_n'$)
- Only possible if f is same in all 2^n cases
- Suppose we have a fooling set of t input vectors that all give the same function value f_0, but for any two of them, there is a mix that gives a different value
- Then all vectors must go to different leaves $\Rightarrow$ tree depth must be $\geq \log(t)$

Example: plurality

- A fooling set:
- *a, a, b, b, c, c, a*
- *a, a, c, c, b, b, a*
- *b, b, a, a, c, c, a*
- *b, b, c, c, a, a, a*
- *c, c, a, a, b, b, a*
- *c, c, b, b, a, a, a*

any two can be mixed to get a different winner, e.g., *a, a, b, b, b, b, a*

a wins on all of these

can be extended to show an order *n* log *m* lower bound (best possible)

- Not a fooling set:
- *a, a, a, b*
- *c, a, a, a*

Finding a winner that is "good enough"

- If we have elicited enough information to determine that an alternative is a necessary winner, we can stop
- But what if we want to stop earlier?
- Suppose we have some measure of welfare (e.g., Borda score) and partially elicited preferences
- Max regret of an alternative *a*:

  How much worse than the best alternative could *a*'s welfare still be?
- Minimax regret alternative: an alternative that minimizes maximum regret
- Then ask queries to try to quickly reduce minimax regret
- [Lu & Boutilier 2011]

Example (Borda)

- Partial profile:
- Voter 1: *a>b>c*
- Voter 2: *a>b*
- Voter 3: *c>a*
- Max regret for *a*: 1 (against *c*, if vote 2 is *c>a>b* and vote 3 is *c>b>a*)
- Max regret for b: 4 (against *a*, if vote 2 is *a>c>b* and vote 3 is *c>a>b*)
- Max regret for *c*: 3 (against *a*, if vote 2 is *a>b>c* and vote 3 is *c>a>b* or *b>c>a*; or against *b*, if vote 2 is *a>b>c* and vote 3 is *b>c>a*)
- *a* is the minimax regret alternative