

CSE 303: Concepts and Tools for Software Development

Hal Perkins

Autumn 2008

Lecture S1— Societal Implications: Voting

Why are we doing this?

An educated computer-scientist should think about the broader implications of what they do.

Conversely, people not trained in computer-science may not be equipped to make ethical / practical decisions about relevant technology.

There are a million topics we could pick; elections are timely.

10% of your grade: a short paper on topic(s) we discuss or something related; more information later.

The plan

I'll share 5–10 minutes of thoughts, overview, examples, and questions.

We'll divide into small groups for 20+ minutes.

- Discuss questions of interest
- Bring up new examples (preferred) or discuss existing ones
- Pick a speaker to report back (1 minute)

We'll reconvene for whole-class discussion

- (1-minute presentations will guide us)

Voting

A mix of discussion and some “fascinating things” about voting.

Two pretty different issues:

1. What should the rules for an election be?
 - beside political and philosophical issues, there are mathematical and economic ones (and CS contributes a lot to discrete math)
2. What technology should we use to run elections well?
 - In particular, should we use computers to tally votes.
 - Clearly relevant (e.g., 2000 presidential election, 2004 Washington governor election, problems with electronic voting in various places)

The first is “more theory”,
the second is “mostly systems with some theory”.

Elections in general

- Individual preferences \rightarrow community choice
 - the arrow is the “social-choice function”
- There are many questions even before we decide how preferences are stated and the function is chosen:
 - Districts vs. proportional representation
 - Length of term, number of offices
 - Time between runoffs
 - Nomination process
 - ...
- To stick to a simple setting, let’s assume n choices, exactly one of which “wins”.

Some “bad” situations

Simple social-choice functions seem to do badly:

- plurality, majority, two-party nominations
 - issues: “wasted votes”, clones, ...
- voter’s paradoxes

Instant run-off voting (political issue: no deliberation)

- Still has shortcomings

Arrow’s Theorem (a Nobel prize in Economics): An impossibility result! (Cannot have a social-choice function that ranks ≥ 3 choices among ≥ 2 voters and *allows any input*, is *deterministic*, *allows any outcome*, *disallows dictators*, is *monotonic* (preferences only help), and *is consistent for all subsets*)

It's worse

And we've been assuming:

- each voter's vote is equally valuable (utilitarians would disagree)
- voter's preferences are "equally spaced"

Does it help to give each person 100 votes?

Should it be easy or hard to vote?

Bottom line: There is discrete math here; computational economics is a hot area.

Also: Political elections may never change, but there is on-line voting (often by computers) everywhere!

Ballot Properties

After deciding a voting method, there are still questions like “can votes be sold”, “should ballots be secret”, etc.

- Repudiation (secrecy)
- Auditability
- Accuracy
- Transparency (know they are accurate)
- Robust to small errors
- Robust to security attack

The paper-ballot method (properly executed) meets these goals, largely because we trust paper and physically observed locked boxes.

Electronic Voting

Why is going paperless dangerous?

Why is handing people a receipt showing who they voted for not a solution?

How are the security issues different than when writing desktop software?

How are the security issues different than when writing airplane code?

How are the security issues different than when writing banking code?