Analyzing Vote Counting Algorithms Via Logic

Carsten Schürmann
IT University of Copenhagen

Joint work with

Bernhard Beckert
Karlsruhe Institute of Technology

Rajeev Gore
Australian National University

June 10, 2013

DemTech
Democratic Technology

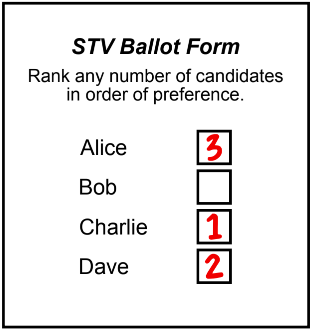
Single Transferable Vote

STV Ballot Form

Rank any number of candidates in order of preference.

Alice	3
Bob	
Charlie	1
Dave	2

Single Transferable Vote



“Standard” Version

0. Calculate the quota of votes.
1. Tally each ballot for its highest pref that is neither elected nor defeated.
   - Surplus votes go to next pref.
2. After all votes have been tallied:
   - If there are more cands. than seats, eliminate cand. with the fewest votes; transfer his votes and re-tally (go to 1).
   - If there are more seats than cands., then all remaining cands. are elected.

Single Transferable Vote

STV Ballot Form

Rank any number of candidates in order of preference.

Alice	3
Bob	
Charlie	1
Dave	2

"Standard" Version

0. Calculate the quota of votes.
1. Tally each ballot for its highest pref that is neither elected nor defeated.
   - Surplus votes go to next pref.
2. After all votes have been tallied:
   - If there are more cands. than seats, eliminate cand. with the fewest votes; transfer his votes and re-tally (go to 1).
   - If there are more seats than cands., then all remaining cands. are elected.

Many choices!

Many versions!

Example

Quota: $Q = \left\lfloor \frac{votes}{seats+1} \right\rfloor + 1$

Candidates: A, B, C, D

Seats: 2

Votes:

$$A > B > D$$
$$A > B > D$$
$$A > B > D$$
$$D > C$$
$$C > D$$

Example

Quota: $Q = \left\lfloor \frac{votes}{seats+1} \right\rfloor + 1$

Candidates: $A, \quad B, \quad C, \quad D$

$$Q = \left\lfloor \frac{5}{2+1} \right\rfloor + 1 = 2$$

Seats: 2

Votes:

$$A > B > D$$
$$A > B > D$$
$$A > B > D$$
$$D > C$$
$$C > D$$

DEMTECH
DEMOCRATIC TECHNOLOGY

Example

Quota: $Q = \left\lfloor \frac{votes}{seats+1} \right\rfloor + 1$

Candidates: $A, \quad B, \quad C, \quad D$

$$Q = \left\lfloor \frac{5}{2+1} \right\rfloor + 1 = 2$$

Seats: 2

Votes:

$A > B > D$ 1

$A > B > D$ 2

$A > B > D$ 3

$D > C$

$C > D$

Example

Quota: $Q = \left\lfloor \frac{votes}{seats+1} \right\rfloor + 1$

Candidates: $A, \quad B, \quad C, \quad D$

$$Q = \left\lfloor \frac{5}{2+1} \right\rfloor + 1 = 2$$

Seats: 2

Votes:

$$\begin{array}{ll} A > B > D & 1 \\ A > B > D & 2 \\ A > B > D & 3 \\ D > C & \\ C > D & \end{array}$$

Elected: A

Example

Quota: $Q = \left\lfloor \frac{votes}{seats+1} \right\rfloor + 1$

Candidates: $A, \quad B, \quad C, \quad D$

$$Q = \left\lfloor \frac{5}{2+1} \right\rfloor + 1 = 2$$

Seats: 2

Votes:

$\cancel{A > B > D}$ 1

$\cancel{A > B > D}$ 2

$A > B > D$ 3

$D > C$

$C > D$

Elected: A

Example

Quota: $Q = \left\lfloor \frac{votes}{seats+1} \right\rfloor + 1$

Candidates: $A, \quad B, \quad C, \quad D$

$Q = \left\lfloor \frac{5}{2+1} \right\rfloor + 1 = 2$

Seats: 2

Votes:

$\cancel{A > B > D}$

$\cancel{A > B > D}$

$\cancel{A} > B > D$

$D > C$

$C > D$

Elected: A

Example

Quota: $Q = \left\lfloor \frac{votes}{seats+1} \right\rfloor + 1$

Candidates: $A, \quad B, \quad C, \quad D$

$$Q = \left\lfloor \frac{5}{2+1} \right\rfloor + 1 = 2$$

Seats: 2

Votes:

$$\cancel{A > B > D}$$
$$\cancel{A > B > D}$$
$$\cancel{A} > \cancel{B} > D$$
$$D > C$$
$$C > D$$

Elected: A

Example

Quota: $Q = \left\lfloor \frac{votes}{seats+1} \right\rfloor + 1$

Candidates: $A, \quad B, \quad C, \quad D$

$Q = \left\lfloor \frac{5}{2+1} \right\rfloor + 1 = 2$

Seats: 2

Votes:

$\cancel{A > B > D}$

$\cancel{A > B > D}$

$\cancel{A} > \cancel{B} > D \quad 1$

$D > C \quad 2$

$C > D$

Elected: A

DEMTECH
DEMOCRATIC TECHNOLOGY

Example

Quota: $Q = \left\lfloor \frac{votes}{seats+1} \right\rfloor + 1$

Candidates: A, B, C, D

$Q = \left\lfloor \frac{5}{2+1} \right\rfloor + 1 = 2$

Seats: 2

Votes:

$\cancel{A > B > D}$

$\cancel{A > B > D}$

$\cancel{A} > \cancel{B} > D$ 1

$D > C$ 2

$C > D$

Elected: A, D

Declarative Properties of Voting Protocols

sort by:																						
	Majority	Mutual Majority Criterion	Condorcet	Strategic, Majority Condorcet	Condorcet loser	Smith/ ISDA	LIIA	IIA	Clone-proof	Mono-tone	CC	PC	Rever-sal sym-metry	Polytime/ Resolvable		Summable	ballot type	= ranks	>2 ranks	Later-no-harm/ Later-no-help		FBC:No favorite betrayal
Approval	Rated [nb 1]	No	No [nb 2]	Yes [nb 3]	No	No [nb 2]	Yes	Yes	Yes [nb 4]	Yes	Yes	Yes	Yes	O(N)	Yes	O(N)	appro-vals	Yes	No		[nb 5]	Yes
Borda count	No	No	No	No	Yes	No	No	No	No: teams	Yes	Yes	Yes	Yes	O(N)	Yes	O(N)	ranking	No	Yes	No	Yes	No
Copeland	Yes	Yes	Yes	Yes	Yes	Yes	No	No [nb 2]	teams, crowds	Yes	No [nb 2]	No [nb 2]	Yes	O(N^2)	No	O(N^2)	ranking	Yes	Yes	No [nb 2]	No	No [nb 2]
IRV (AV)	Yes	Yes	No [nb 2]	No	Yes	No [nb 2]	No	No	Yes	No	No	No	No	O(N^2)	Yes	O(N!)[nb 6]	ranking	No	Yes	Yes	Yes	No
Kemeny-Young	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No [nb 2]	No: spoilers	Yes	No [nb 2] [nb 7]	No [nb 2]	Yes	O(N!)	Yes	O(N^2) [nb 8]	ranking	Yes	Yes	No [nb 2]	No	No [nb 2]
Majority Judgment [nb 9]	Rated	No [nb 10]	No [nb 2]	Yes [nb 3]	No	No [nb 2]	Yes	Yes	Yes	Yes	No [nb 11]	No	Dep-ends [nb 12]	O(N)	Yes	O(N)[nb 13]	scores [nb 14]	Yes	Yes	No [nb 15]	Yes	Yes
Minimax	Yes	No	Yes [nb 16]	Yes	No	No	No	No [nb 2]	No: spoilers	Yes	No [nb 2]	No [nb 2]	No	O(N^2)	Yes	O(N^2)	ranking	Yes	Yes	No [nb 2] [nb 16]	No	No [nb 2]
Plurality	Yes	No	No [nb 2]	No	No	No [nb 2]	No	No	No: spoilers	Yes	Yes	Yes	No	O(N)	Yes	O(N)	single mark	NA	No	NA [nb 17]	NA [nb 17]	No
Range voting	No	No	No [nb 2]	Yes [nb 3]	No	No [nb 2]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	O(N)	Yes	O(N)	scores	Yes	Yes	No	Yes	Yes
Ranked pairs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No [nb 2]	Yes	Yes	No [nb 2]	No [nb 2]	Yes	O(N^4)	Yes	O(N^2)	ranking	Yes	Yes	No [nb 2]	No	No [nb 2]
Runoff voting	Yes	No	No [nb 2]	No	Yes	No [nb 2]	No	No	No: spoilers	No	No	No	No	O(N) [nb 18]	Yes	O(N) [nb 19]	single mark	NA	No [nb 20]	Yes	[nb 21]	No
Schulze	Yes	Yes	Yes	Yes	Yes	Yes	No	No [nb 2]	Yes	Yes	No [nb 2]	No [nb 2]	Yes	O(N^3)	Yes	O(N^2)	ranking	Yes	Yes	No [nb 2]	No	No [nb 2]

[https://en.wikipedia.org/wiki/Voting_system]

Declarative Properties of Voting Systems (cont'd)

Condorcet criterion

- The voting scheme always elects a candidate who, when compared with every other candidate, is preferred by more voters.

Declarative Properties of Voting Systems (cont'd)

Condorcet criterion

- The voting scheme always elects a candidate who, when compared with every other candidate, is preferred by more voters.

Monotonicity criterion

- A candidate x cannot be harmed if x is raised on some ballots without changing the orders of the other candidates.

Declarative Properties of Voting Systems (cont'd)

Condorcet criterion

- The voting scheme always elects a candidate who, when compared with every other candidate, is preferred by more voters.

Monotonicity criterion

- A candidate x cannot be harmed if x is raised on some ballots without changing the orders of the other candidates.

Majority criterion

- If one candidate is preferred by a majority (more than 50%) of voters, then that candidate must win.

Single Transferable Vote @CADE

Quote from CADE Bylaws (legal document)

```
Procedure STV

Elected <-- empty
T <-- Tbl                 {* Start with the original vote matrix *}
for E <-- 1 to K
    N' <-- N-E+1    {* Choose a winner among N' candidates *}
    T' <-- T        {* store the current vote matrix *}
    while (no candidate has a majority of 1st preferences)
        w <-- one weakest candidate
        for all candidates c {* remove all weakest candidates *}
            if c is equally weak as w
                Redistribute(c,T)
        end for
    end while
    win <-- the majority candidate
    Elected <-- append(Elected, [win])
    T <-- T'        {* restore back to N' candidates *}
    Redistribute(win, T)  {* remove winner & redistrb. votes *}
end for

End STV
```

What could go wrong?

Contributions

Celf – a Voting Algorithm Work Bench

Domain-Specific Declarative Criteria

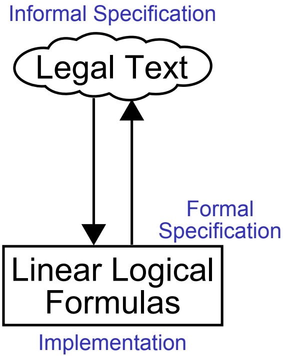
Bounded Model-Checking

Findings

Celf

A Voting Algorithm Work Bench

Linear Logical Voting Protocols [deYoung + CS '11]



Celf [Schack-Nielsen+CS '08]

- Law as specification
- Linear Inference
- Concise encodings
- Executlable proof search semantics
- Checkable certificates

Single Transferable Vote on a Single Slide

begin/1 :
$begin(S,H,U) \otimes$
$!(Q = U/(S+1)+1)$
$\multimap \{!quota(Q) \otimes$
$tally\text{-}votes(S,H,U)\}$

tally/1 :
$tally\text{-}votes(S,H,U) \otimes$
$uncounted\text{-}ballot(C,L) \otimes$
$hopeful(C,N) \otimes$
$!quota(Q) \otimes !(N+1 < Q)$
$\multimap \{counted\text{-}ballot(C,L) \otimes$
$hopeful(C,N+1) \otimes$
$tally\text{-}votes(S,H,U-1)\}$

tally/2 :
$tally\text{-}votes(S,H,U) \otimes$
$uncounted\text{-}ballot(C,L) \otimes$
$hopeful(C,N) \otimes$
$!quota(Q) \otimes !(N+1 \geq Q) \otimes$
$!(S \geq 1)$
$\multimap \{counted\text{-}ballot(C,L) \otimes$
$!elected(C) \otimes$
$tally\text{-}votes(S-1,H-1,U-1)\}$

tally/3 :
$tally\text{-}votes(S,H,U) \otimes$
$uncounted\text{-}ballot(C,[C' \mid L]) \otimes$
$(!elected(C) \oplus !defeated(C))$
$\multimap \{uncounted\text{-}ballot(C',L) \otimes$
$tally\text{-}votes(S,H,U)\}$

tally/4 :
$tally\text{-}votes(S,H,U) \otimes$
$uncounted\text{-}ballot(C,[\,]) \otimes$
$(!elected(C) \oplus !defeated(C))$
$\multimap \{tally\text{-}votes(S,H,U-1)\}$

tally/5 :
$tally\text{-}votes(S,H,0) \otimes$
$!(S < H)$
$\multimap \{defeat\text{-}min(S,H,0)\}$

tally/6 :
$tally\text{-}votes(S,H,0) \otimes$
$!(S \geq H)$
$\multimap \{!elect\text{-}all\}$

defeat-min/1 :
$defeat\text{-}min(S,H,M) \otimes$
$hopeful(C,N)$
$\multimap \{minimum(C,N) \otimes$
$defeat\text{-}min(S,H-1,M+1)\}$

defeat-min/2 :
$defeat\text{-}min(S,0,M)$
$\multimap \{defeat\text{-}min'(S,0,M)\}$

defeat-min′/1 :
$defeat\text{-}min'(S,H,M) \otimes$
$minimum(C_1,N_1) \otimes$
$minimum(C_2,N_2) \otimes$
$!(N_1 \leq N_2)$
$\multimap \{minimum(C_1,N_1) \otimes$
$hopeful(C_2,N_2) \otimes$
$defeat\text{-}min'(S,H+1,M-1)\}$

defeat-min′/2 :
$defeat\text{-}min'(S,H,1) \otimes$
$minimum(C,N)$
$\multimap \{!defeated(C) \otimes$
$transfer(C,N,S,H,0)\}$

transfer/1 :
$transfer(C,N,S,H,U) \otimes$
$counted\text{-}ballot(C,L)$
$\multimap \{uncounted\text{-}ballot(C,L) \otimes$
$transfer(C,N-1,S,H,U+1)\}$

transfer/2 :
$transfer(C,0,S,H,U)$
$\multimap \{tally\text{-}votes(S,H,U)\}$

elect-all/1 :
$!elect\text{-}all \otimes$
$hopeful(C,N)$
$\multimap \{!elected(C)\}$

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn't be reached by this vote,
then mark the ballot as counted and
update C's tally to $N+1$ votes and
tally the remaining $U-1$ ballots.

tally/1 :

$$
\begin{aligned}
& \mathit{tally\text{-}votes}(S, H, U) \otimes \\
& \mathit{uncounted\text{-}ballot}(C, L) \otimes \\
& \mathit{hopeful}(C, N) \otimes \\
& !\mathit{quota}(Q) \otimes !(N+1 < Q) \\
& \quad \multimap \{\mathit{counted\text{-}ballot}(C, L) \otimes \\
& \qquad \mathit{hopeful}(C, N+1) \otimes \\
& \qquad \mathit{tally\text{-}votes}(S, H, U-1)\}
\end{aligned}
$$

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading	tally/1 :
If we are tallying votes and	$\textit{tally-votes}(S, H, U) \otimes$
there is an uncounted vote for C and	$\textit{uncounted-ballot}(C, L) \otimes$
C is a hopeful with running tally N and	$\textit{hopeful}(C, N) \otimes$
the quota wouldn't be reached by this vote,	$!\textit{quota}(Q) \otimes !(N+1 < Q)$
then mark the ballot as counted and	$\multimap \{\textit{counted-ballot}(C, L) \otimes$
update C's tally to $N+1$ votes and	$\textit{hopeful}(C, N+1) \otimes$
tally the remaining $U-1$ ballots.	$\textit{tally-votes}(S, H, U-1)\}$

DEMTECH
DEMOCRATIC TECHNOLOGY

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn't be reached by this vote,
then mark the ballot as counted and
update C's tally to $N+1$ votes and
tally the remaining $U-1$ ballots.

tally/1 :

$$\begin{array}{l} \textit{tally-votes}(S,H,U) \otimes \\ \textit{uncounted-ballot}(C,L) \otimes \\ \textit{hopeful}(C,N) \otimes \\ !\textit{quota}(Q) \otimes !(N+1 < Q) \\ \quad \multimap \{\textit{counted-ballot}(C,L) \otimes \\ \qquad \textit{hopeful}(C,N+1) \otimes \\ \qquad \textit{tally-votes}(S,H,U-1)\} \end{array}$$

DEMTECH
DEMOCRATIC TECHNOLOGY

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn't be reached by this vote,
then mark the ballot as counted and
update C's tally to $N{+}1$ votes and
tally the remaining $U{-}1$ ballots.

tally/1 :

$$
\begin{array}{l}
\textit{tally-votes}(S,H,U) \otimes \\
\textit{uncounted-ballot}(C,L) \otimes \\
\textit{hopeful}(C,N) \otimes \\
!\textit{quota}(Q) \otimes\, !(N{+}1 < Q) \\
\quad \multimap \{\textit{counted-ballot}(C,L) \otimes \\
\qquad \textit{hopeful}(C,N{+}1) \otimes \\
\qquad \textit{tally-votes}(S,H,U{-}1)\}
\end{array}
$$

DEMTECH
DEMOCRATIC TECHNOLOGY

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn't be reached by this vote,
then mark the ballot as counted and
update C's tally to $N+1$ votes and
tally the remaining $U-1$ ballots.

tally/1 :

$$
\begin{aligned}
& \textit{tally-votes}(S, H, U) \otimes \\
& \textit{uncounted-ballot}(C, L) \otimes \\
& \textit{hopeful}(C, N) \otimes \\
& !\textit{quota}(Q) \otimes !(N+1 < Q) \\
& \quad \multimap \{\textit{counted-ballot}(C, L) \otimes \\
& \qquad \textit{hopeful}(C, N+1) \otimes \\
& \qquad \textit{tally-votes}(S, H, U-1)\}
\end{aligned}
$$

DEMTECH
DEMOCRATIC TECHNOLOGY

From Legal Text to Formal Specification

Legal Text

“Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated.”

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn’t be reached by this vote,
then mark the ballot as counted and
update C’s tally to $N+1$ votes and
tally the remaining $U-1$ ballots.

tally/1 :

$$
\begin{array}{l}
\mathit{tally\text{-}votes}(S,H,U) \otimes \\
\mathit{uncounted\text{-}ballot}(C,L) \otimes \\
\mathit{hopeful}(C,N) \otimes \\
!\mathit{quota}(Q) \otimes !(N+1 < Q) \\
\quad \multimap \{\mathit{counted\text{-}ballot}(C,L) \otimes \\
\qquad \mathit{hopeful}(C,N+1) \otimes \\
\qquad \mathit{tally\text{-}votes}(S,H,U-1)\}
\end{array}
$$

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn't be reached by this vote,
then mark the ballot as counted and
update C's tally to $N+1$ votes and
tally the remaining $U-1$ ballots.

tally/1 :

$$
\begin{array}{l}
\textit{tally-votes}(S, H, U) \otimes \\
\textit{uncounted-ballot}(C, L) \otimes \\
\textit{hopeful}(C, N) \otimes \\
!\textit{quota}(Q) \otimes !(N+1 < Q) \\
\quad \multimap \{\textit{counted-ballot}(C, L) \otimes \\
\qquad \textit{hopeful}(C, N+1) \otimes \\
\qquad \textit{tally-votes}(S, H, U-1)\}
\end{array}
$$

DEMTECH
DEMOCRATIC TECHNOLOGY

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn't be reached by this vote,
then mark the ballot as counted and
update C's tally to $N+1$ votes and
tally the remaining $U-1$ ballots.

tally/1 :

$$
\begin{array}{l}
\textit{tally-votes}(S, H, U) \otimes \\
\textit{uncounted-ballot}(C, L) \otimes \\
\textit{hopeful}(C, N) \otimes \\
!\textit{quota}(Q) \otimes !(N+1 < Q) \\
\quad \multimap \{\textit{counted-ballot}(C, L) \otimes \\
\qquad \textit{hopeful}(C, N+1) \otimes \\
\qquad \textit{tally-votes}(S, H, U-1)\}
\end{array}
$$

DEMTECH
DEMOCRATIC TECHNOLOGY

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn't be reached by this vote,
then mark the ballot as counted and
update C's tally to $N+1$ votes and
tally the remaining $U-1$ ballots.

tally/1 :

$$
\begin{array}{l}
\mathit{tally\text{-}votes}(S,H,U) \otimes \\
\mathit{uncounted\text{-}ballot}(C,L) \otimes \\
\mathit{hopeful}(C,N) \otimes \\
!\mathit{quota}(Q) \otimes !(N+1 < Q) \\
\quad \multimap \{\mathit{counted\text{-}ballot}(C,L) \otimes \\
\qquad \mathit{hopeful}(C,N+1) \otimes \\
\qquad \mathit{tally\text{-}votes}(S,H,U-1)\}
\end{array}
$$

- Correspondence between legal text and logical formula is plain!
- Linearity: count ballots only once and update running tallies.

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn't be reached by this vote,
then mark the ballot as counted and
update C's tally to $N+1$ votes and
tally the remaining $U-1$ ballots.

tally/1 :

$$
\begin{array}{l}
\textit{tally-votes}(S, H, U) \otimes \\
\textit{uncounted-ballot}(C, L) \otimes \\
\textit{hopeful}(C, N) \otimes \\
!\textit{quota}(Q) \otimes !(N+1 < Q) \\
\quad \multimap \{\textit{counted-ballot}(C, L) \otimes \\
\qquad \textit{hopeful}(C, N+1) \otimes \\
\qquad \textit{tally-votes}(S, H, U-1)\}
\end{array}
$$

- Correspondence between legal text and logical formula is plain!
- Linearity: count ballots only once and update running tallies.

DEMTECH
DEMOCRATIC TECHNOLOGY

From Legal Text to Formal Specification

Legal Text

"Tally the votes, assigning each ballot to its highest preference candidate who is neither elected nor defeated."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is a hopeful with running tally N and
the quota wouldn't be reached by this vote,
then mark the ballot as counted and
update C's tally to $N+1$ votes and
tally the remaining $U-1$ ballots.

tally/1 :

$$
\begin{array}{l}
\mathit{tally\text{-}votes}(S,H,U) \otimes \\
\mathit{uncounted\text{-}ballot}(C,L) \otimes \\
\mathit{hopeful}(C,N) \otimes \\
!\mathit{quota}(Q) \otimes !(N+1 < Q) \\
\quad \multimap \{\mathit{counted\text{-}ballot}(C,L) \otimes \\
\qquad \mathit{hopeful}(C,N+1) \otimes \\
\qquad \mathit{tally\text{-}votes}(S,H,U-1)\}
\end{array}
$$

- Correspondence between legal text and logical formula is plain!
- Linearity: count ballots only once and update running tallies.

DEMTECH
DEMOCRATIC TECHNOLOGY

Parametrization

QUOTA/DROOP, QUOTA/HARE, QUOTA/MAJORITY

How to compute the quota?

TIE

Shall ties be broken?

ZOMBIE

Resurrection of already eliminated candidates?

AUTOFILL

Automatic placement of remaining candidates on remaining seats?

NODEL

Keep votes from one iteration to the next?

Domain-Specific Declarative Criteria

Declarative Criteria

Electoral Systems

- Social choice functions
  - Society agrees on basic democratic principles
  - Optimization problem
  - Voting algorithm computes a ”good” approximation
- Challenges for preferentional voting schemes
- Intractability [Procaccia et al. ’08]
- Impossibility [Arrow ’51]
- *therefore formal verification IMPOSSIBLE in PRACTICE*

Declarative Criteria (cont'd)

Criterion 1

There are enough votes for each elected candidate
(ignoring preferences)

Criterion 2

- Election result is consistent with union U of preferences if U is consistent (ignoring number of votes)
- related to Pareto criterion: *If all voters rank X over Y, then Y should not win*
- See paper for details

DEMTECH
DEMOCRATIC TECHNOLOGY

Illustration of Criterion 1

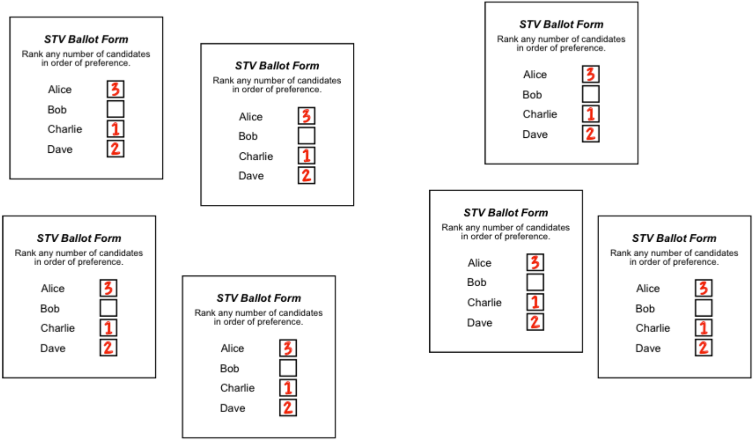
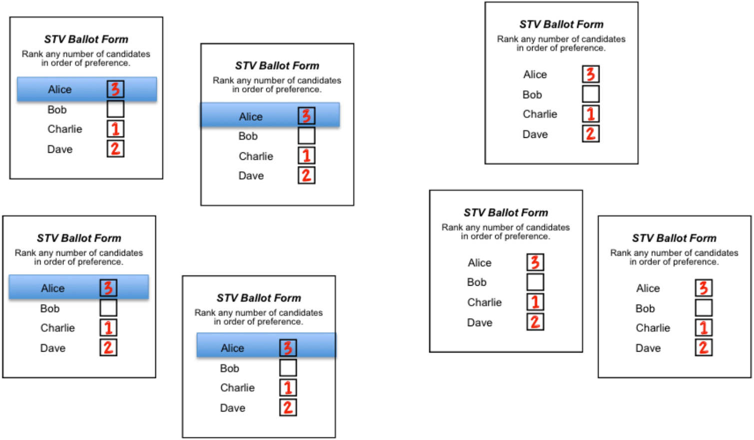


Illustration of Criterion 1



Alice Elected

Illustration of Criterion 1

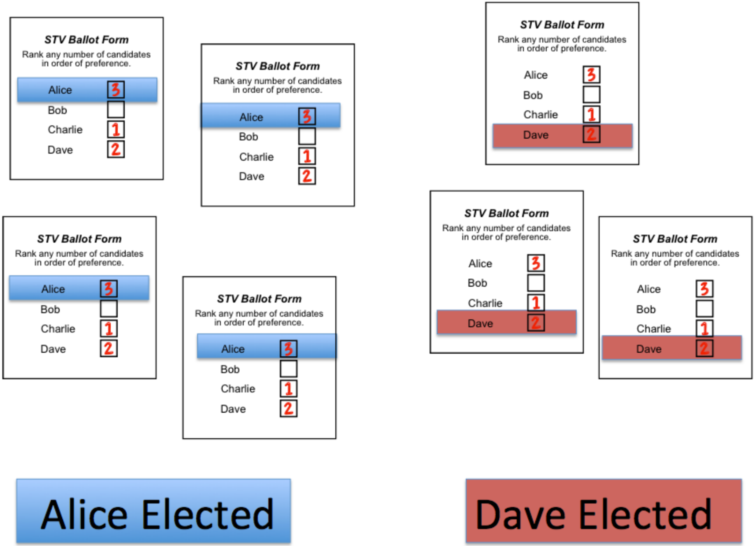
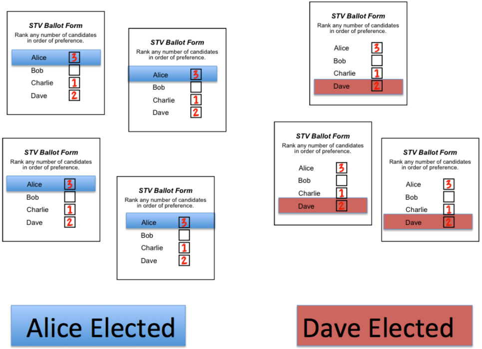


Illustration of Criterion 1

- i ranges over votes
- k ranges over seats
- j ranges over preferences

- $a[i]$ partition of votes
- $r[k]$ who got elected
- $b[i, j]$ ballot box



Declarative Criteria (cont'd)

Formalization of Criterion 1

$$
\begin{array}{l}
\exists a\Big(\\
\quad \forall i\big(1 \le i \le \mathtt{V} \to 0 \le a[i] \le \mathtt{S}\big) \wedge \\
\quad \forall i\big(1 \le i \le \mathtt{V} \to (a[i] \ne 0 \to r[a[i]] \ne 0\big) \wedge \\
\quad \forall i\big((1 \le i \le \mathtt{V} \wedge a[i] \ne 0) \to \exists j(1 \le j \le \mathtt{C} \wedge b[i,j] = r[a[i]])\big) \wedge \\
\quad \forall k\big((1 \le k \le \mathtt{S} \wedge r[k] \ne 0) \to \\
\qquad \exists count(count[0] = 0 \wedge \\
\qquad\quad \forall i(1 \le i \le \mathtt{V} \to (a[i] = k \to count[i] = count[i-1]+1) \wedge \\
\qquad\qquad\qquad\qquad (a[i] \ne k \to count[i] = count[i-1])) \wedge \\
\qquad count[\mathtt{V}] = \mathtt{Q})\big) \\
\quad \Big)
\end{array}
$$

Bounded Model-Checking

Bounded Model Checking Standard STV

Method

- Generate all possible ballot-boxes (up to certain bounds)

```
create-ballot/nil : create-ballot nil.
create-ballot/cons : create-ballot (cons C L)
      @- candidate C
      @- create-ballot L.
```

- Uses affine features of Celf
- Run STV
  (with QUOTA/DROOP, AUTOFILL, TIE and not ZOMBIE, NODEL)
- Check result in Z3 [Bjorner et al.]
- Ballot boxes up to small size checked

DEMTECH
DEMOCRATIC TECHNOLOGY

Differences CADE-STV / Standard STV

CADE-STV

Parameter Choices

- QUOTA/MAJORITY: $>$50% of votes (majority)
- TIE: random
- ZOMBIE and NODEL: Restart with original ballot-box (deleted votes and weakest candidates come back)
- AUTOFILL off: no automatic seating

Bounded Model Checking CADE-STV

Formalization of 1st Property in Z3

```
[[ And(a[i] >= 0, a[i] <= S) for i in range(V).

   Implies(And(a[i] != 0, a[i] == j), r[j] != 0)
     for i in range(V) for j in range(S+1),

   Implies(And(a[i] != 0, a[i] == pi),
                  Or([b[i][j] == r[pi] for j in range(C)]))
        for i in range(V) for pi in range(S+1),

   Implies(r[k] != 0,
     Exists(count,
       And(count[0] == 0, count[V] == Q,
       And([And(Implies(a[i] == k, count[i+1] == count[i]+1),
                Implies(a[i] != k, count[i+1] == count[i]))
                for i in range(V)])))))
       for k in range(S+1)
]]
```

Counter Example

Candidates: A, B, C, D

Seats: 2

Votes:

$$A > B > D$$
$$A > B > D$$
$$A > B > D$$
$$D > C$$
$$C > D$$

Counter Example

Candidates: $A, \quad B, \quad C, \quad D$ $\qquad Q = \left\lfloor \frac{5}{2} \right\rfloor + 1 = 3$

Seats: 2

Votes:

$$A > B > D$$
$$A > B > D$$
$$A > B > D$$
$$D > C$$
$$C > D$$

Counter Example

Candidates: $A, \quad B, \quad C, \quad D$ $\qquad Q = \left\lfloor \frac{5}{2} \right\rfloor + 1 = 3$

Seats: 2

Votes:

$$
\begin{array}{ll}
A > B > D & 1 \\
A > B > D & 2 \\
A > B > D & 3 \\
D > C & \\
C > D &
\end{array}
$$

Counter Example

Candidates: $A, \quad B, \quad C, \quad D$ $\qquad Q = \left\lfloor \frac{5}{2} \right\rfloor + 1 = 3$

Seats: 2

Votes:

$$\begin{array}{ll} A > B > D & 1 \\ A > B > D & 2 \\ A > B > D & 3 \\ D > C & \\ C > D & \end{array}$$

Elected: A

Counter Example

Candidates: $A, \quad B, \quad C, \quad D$ $\qquad Q = \left\lfloor \frac{5}{2} \right\rfloor + 1 = 3$

Seats: 2

Votes:

$$\cancel{A} > B > D \quad 1$$
$$\cancel{A} > B > D \quad 2$$
$$\cancel{A} > B > D \quad 3$$
$$D > C$$
$$C > D$$

Elected: A

Counter Example

Candidates: $A, \quad B, \quad C, \quad D$ $\qquad Q = \left\lfloor \frac{5}{2} \right\rfloor + 1 = 3$

Seats: 2

Votes:

$$\cancel{A} > B > D$$
$$\cancel{A} > B > D$$
$$\cancel{A} > B > D$$
$$D > C$$
$$C > D$$

Elected: A

Counter Example

Candidates: $A, \quad B, \quad C, \quad D$ $\qquad Q = \left\lfloor \frac{5}{2} \right\rfloor + 1 = 3$

Seats: 2

Votes:

$$\cancel{A} > B > D \quad 1$$
$$\cancel{A} > B > D \quad 2$$
$$\cancel{A} > B > D \quad 3$$
$$D > C$$
$$C > D$$

Elected: A

DemTech
Democratic Technology

Counter Example

Candidates: A, B, C, D $\quad Q = \left\lfloor \frac{5}{2} \right\rfloor + 1 = 3$

Seats: 2

Votes:

$$\cancel{A} > B > D \quad 1$$
$$\cancel{A} > B > D \quad 2$$
$$\cancel{A} > B > D \quad 3$$
$$D > C$$
$$C > D$$

Elected: A, B

Counter Example

Candidates: $A, \quad B, \quad C, \quad D$ $\qquad Q = \left\lfloor \frac{5}{2} \right\rfloor + 1 = 3$

Seats: 2

Votes:

$$\cancel{A} > B > D \quad 1$$
$$\cancel{A} > B > D \quad 2$$
$$\cancel{A} > B > D \quad 3$$
$$D > C$$
$$C > D$$

Elected: $A, \quad B$

No proportional representation!

Majority rules!

DEMTECH

DEMOCRATIC TECHNOLOGY

History

Concern *[D. Plaisted, 1996]*

[. . .] This means that the trustee nominees tend to be elected even if only a minority is happy with the scheme.

History

Concern *[D. Plaisted, 1996]*

> *[. . .] This means that the trustee nominees tend to be elected even if only a minority is happy with the scheme.*

"Solution"

1. QUOTA/MAJORITY, AUTOFILL off
   $\Longrightarrow$ No one elected against wishes of majority
2. Restart: NODEL and ZOMBIES
   $\Longrightarrow$ Seats get nevertheless filled (in practice)

History

Concern *[D. Plaisted, 1996]*

> *[. . .] This means that the trustee nominees tend to be elected even if only a minority is happy with the scheme.*

"Solution"

1. QUOTA/MAJORITY, AUTOFILL off
   $\Longrightarrow$ No one elected against wishes of majority
2. Restart: NODEL and ZOMBIES
   $\Longrightarrow$ Seats get nevertheless filled (in practice)

But . . .

Majority rules

Conclusions

Conclusions

Conclusion

- Support in reasoning about voting schemes needed
- Can be automated with bounded model checking
- Tailor-made properties for specific voting systems needed

Conclusions

Conclusion

- Support in reasoning about voting schemes needed
- Can be automated with bounded model checking
- Tailor-made properties for specific voting systems needed

Secondary Conclusion

- Technology drives the evolution of voting algorithms.