# Certifying Voting Protocols

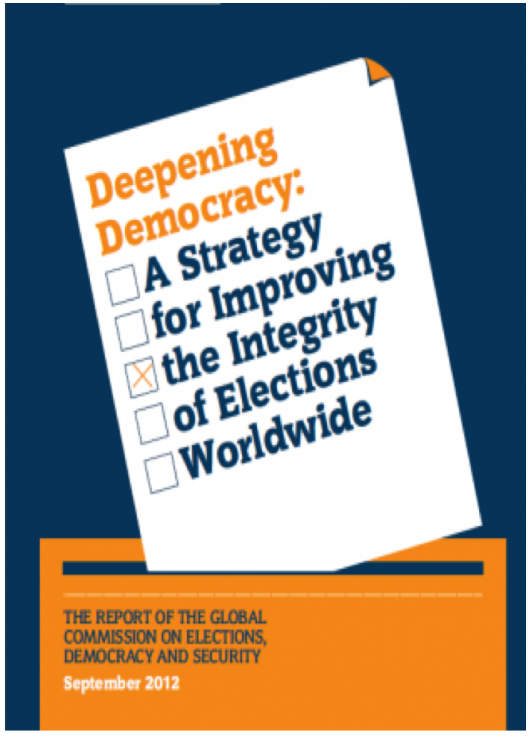
Carsten Schürmann

DemTech
IT University of Copenhagen

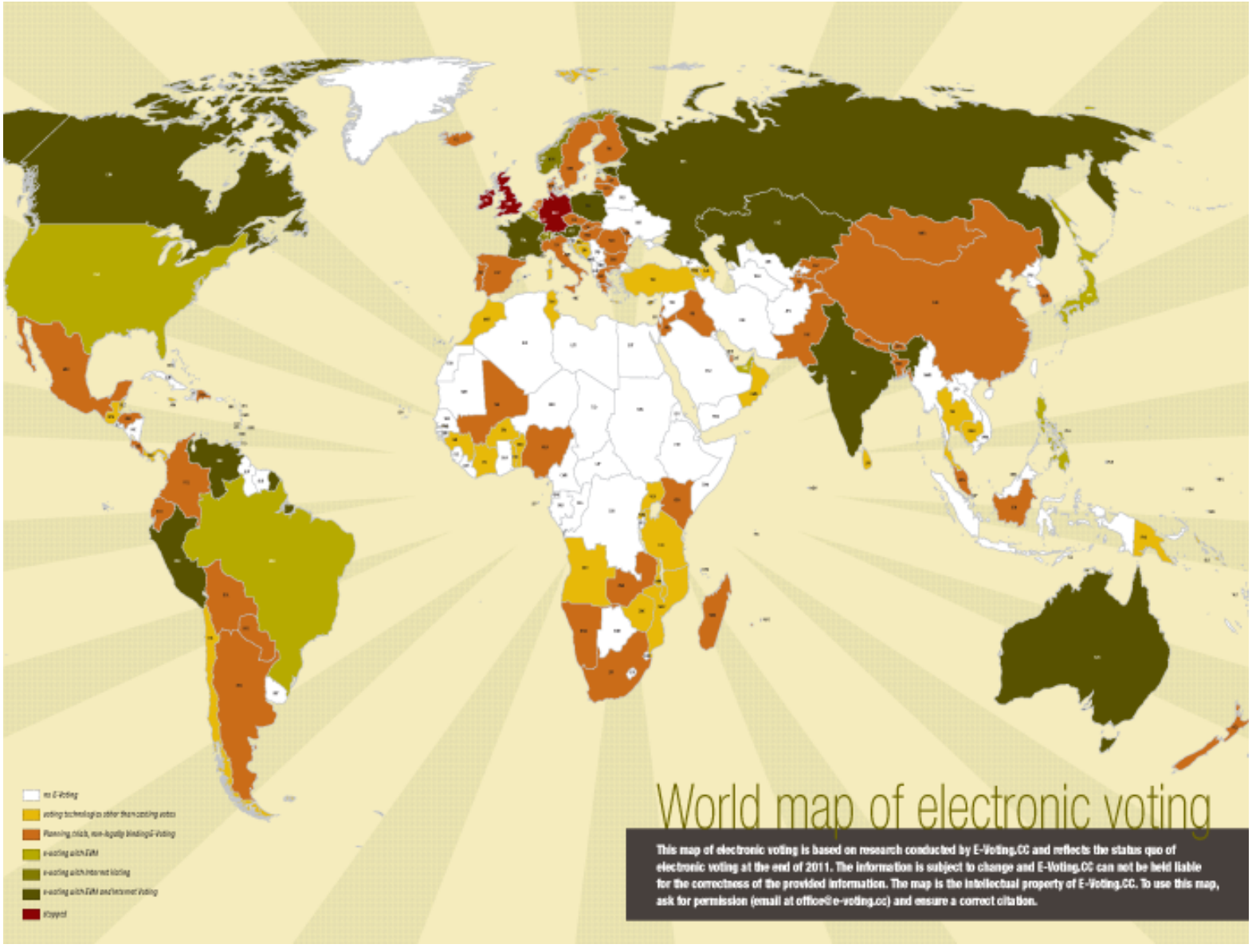
May 28, 2013

# Kofi Annan Report 2012

- 196 countries in the world
- 185 of those held national elections since 2000
- Active construction of Electoral Management Bodies
- often using technology



# International Evoting Map



# Information Technology and the Electoral Process

**Principle**

The goal of designing election processes must always be to achieve credible elections that are acceptable. Information technology should only be used in the electoral process, if it can be satisfactorily argued that it it preserves or creates *trust* in the electoral process.

# Trust

- Voter value and trust system
  - Trust in bureaucracy
  - Trust in public control
  - Trust in judges
- Voting culture and rituals
- Formal verification
- Voter verifiable paper trails
- Auditing procedures and policies
- Classification: Administrative, cultural, mechanical, procedural, cryptographic

# Cyber Security Challenges

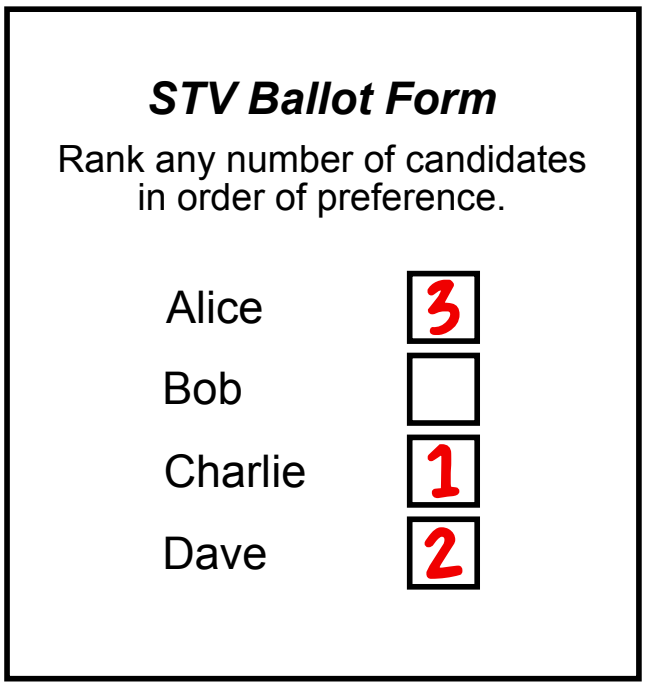
- Selected Administrative Challenges
  - Voter registration and polling stations
  - Election day
  - Out of country voting
  - *Tabulation*
  - Transmission of results
  - *Tracking and solving disputes*
- Selected Technological Challenges
  - Pervasive uses of complex ICT
  - Attack surfaces
  - Software Independence
  - End to end verifiability
  - *Programming language abstractions*
- Selected Legal Challenges
  - Policy and law
- Selected Communication Challenges
  - Education and publication

# In This Talk: Certifying Voting Protocols

- Programming Language Abstractions
- Tabulation
- Tracking and solving disputes

1. Single Transferable Vote (STV)
2. A Logical Specification of Single Transferable Vote
3. Summary

# Electoral Law




## Outline of STV Protocol:

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.

# How and Who Translates this into Software?

- It's just pseudo code

  What operational semantics?

- It's a specification

  How do we know that the specification is ok?

  What democratic properties shall the electoral system have?

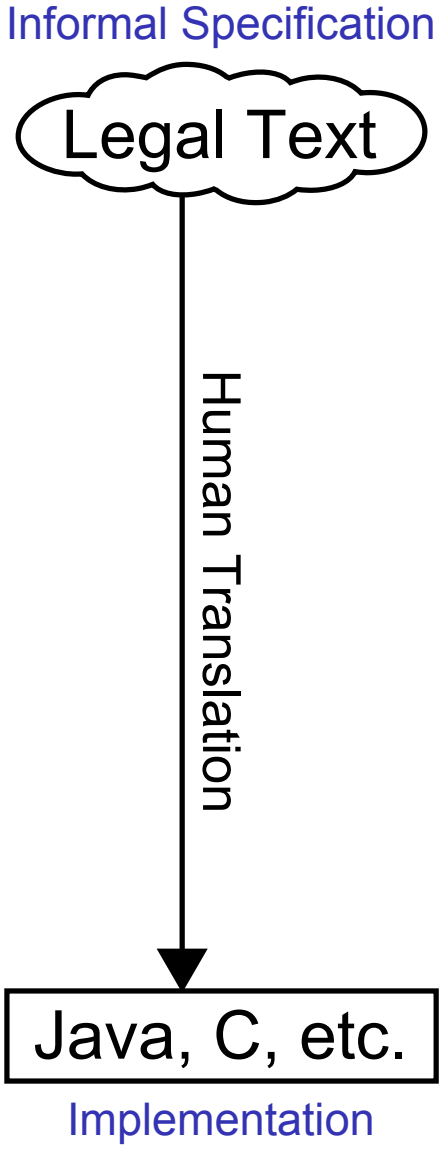
- It is legally binding

  Can we fix errors in the protocol?

  How do we we resolve inconsistencies?

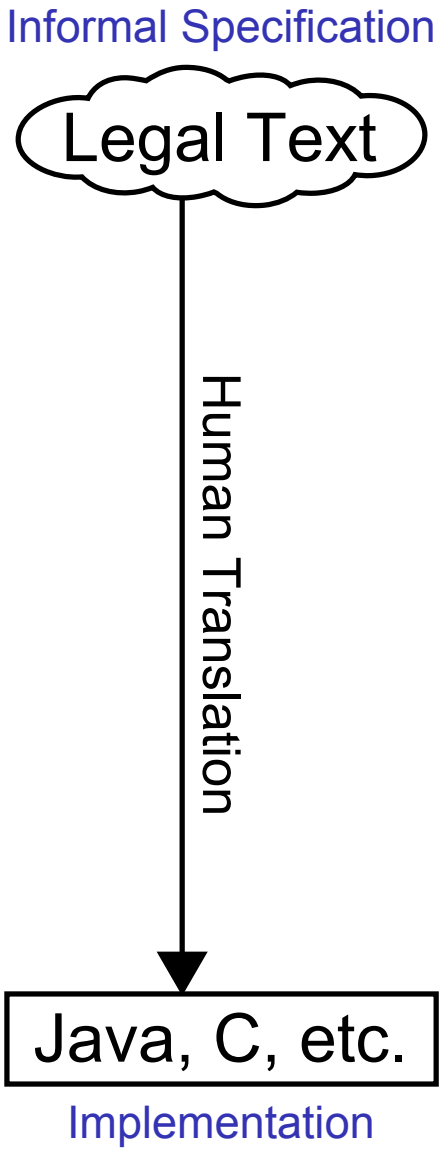
  How do we we fill holes?

# Current Approach to Programming Voting Protocols



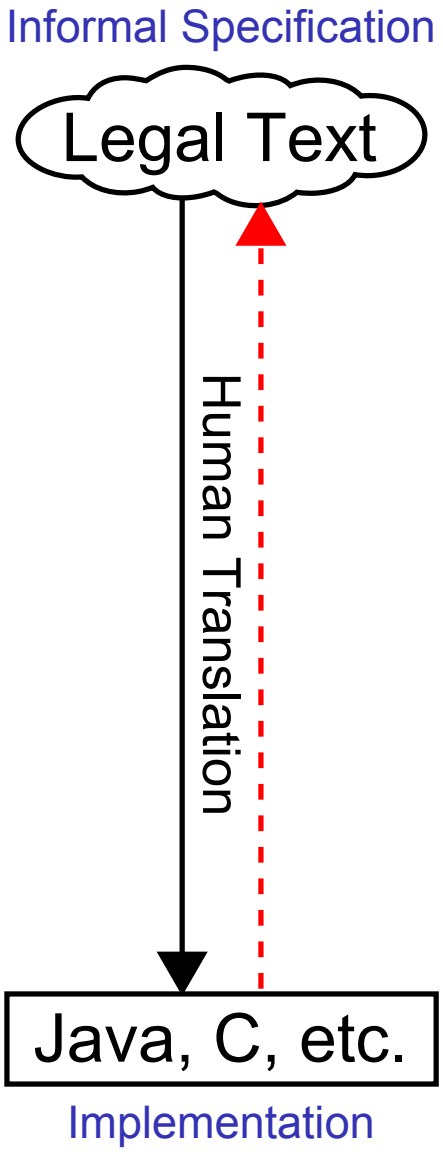

1. Translate legal text to imperative source code.

# Current Approach to Programming Voting Protocols



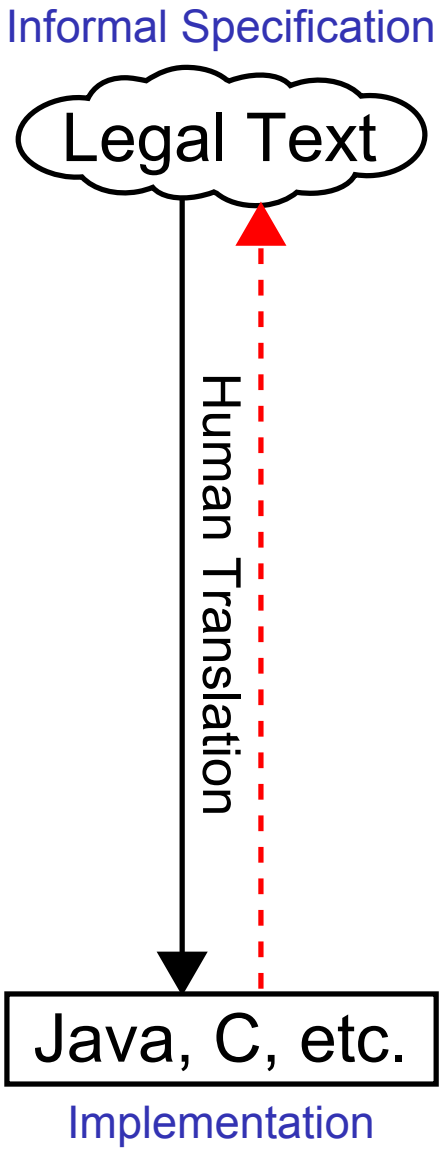

1. Translate legal text to imperative source code.
   - How to trust this?

# Current Approach to Programming Voting Protocols




1. Translate legal text to imperative source code.
   - How to trust this?
2. Certify that code meets legal specification.
   - *Very* hard!

# Current Approach to Programming Voting Protocols




1. Translate legal text to imperative source code.
   - How to trust this?
2. Certify that code meets legal specification.
   - *Very* hard!

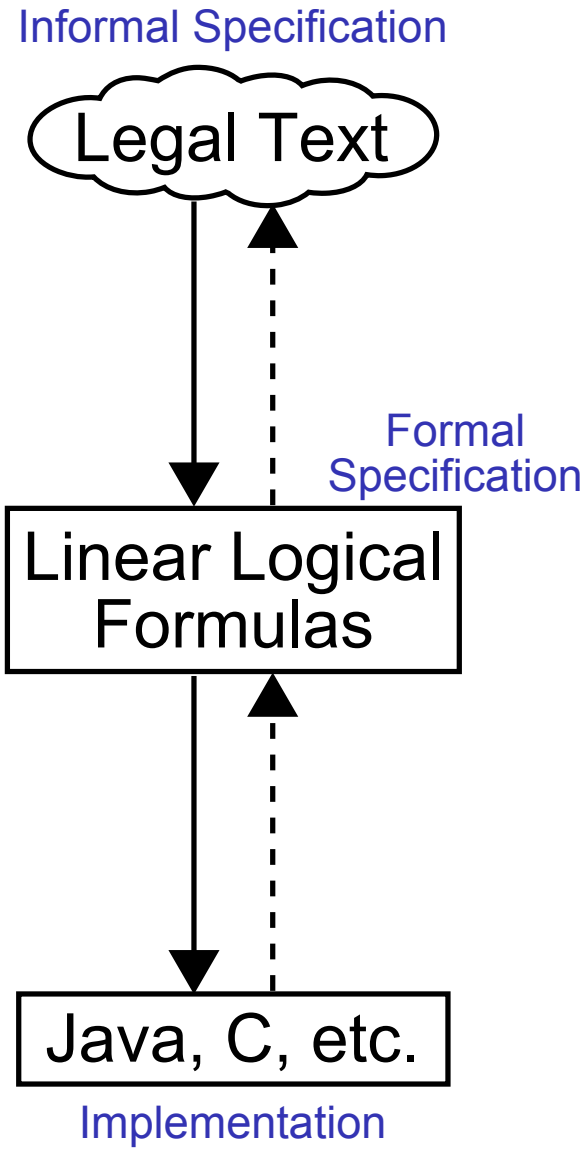
*Is this approach really trustworthy?*

## Key Idea

Formal logic, particularly linear logic, is well-suited to the trustworthy specification and implementation of voting protocols.

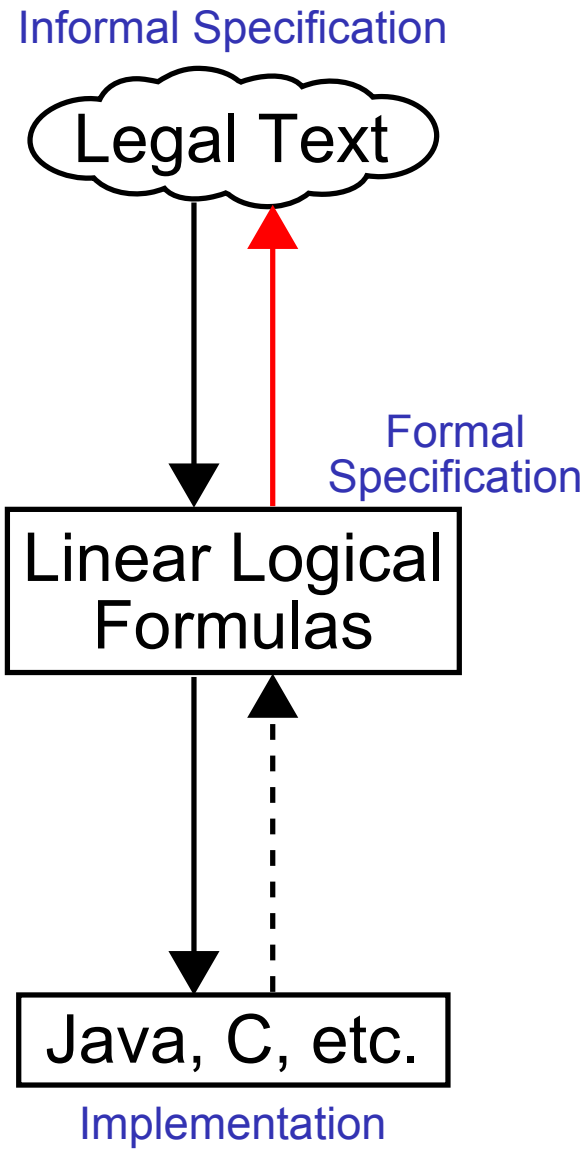
(joined work with Henry deYoung, CMU)

# Certifying Voting Protocols



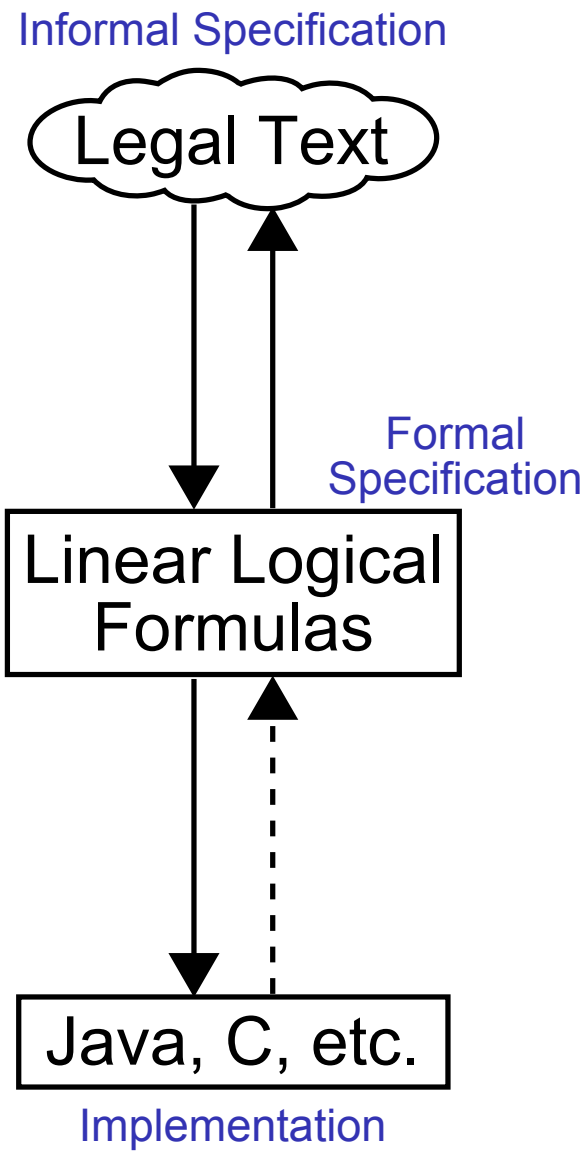

1. Translate legal text to logical formulas.
   - Algorithms at high level of abstraction

# Certifying Voting Protocols



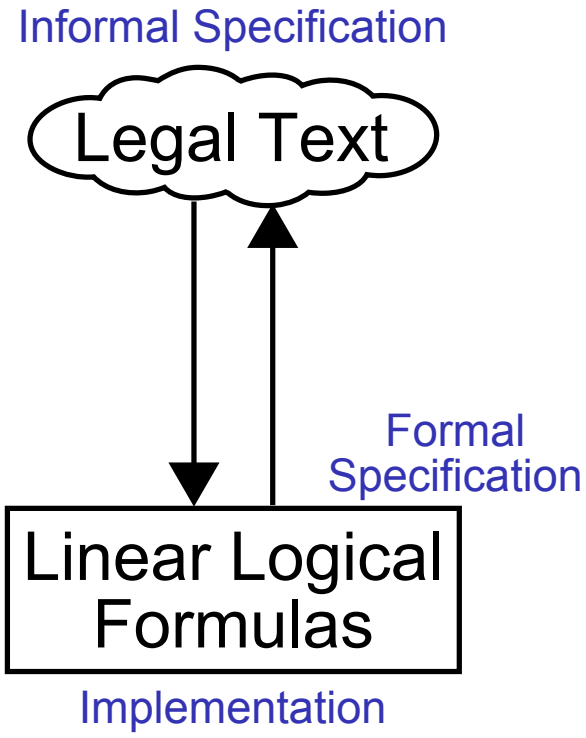

1. Translate legal text to logical formulas.
   - Algorithms at high level of abstraction
   - Much smaller gap from legal language!

# Certifying Voting Protocols



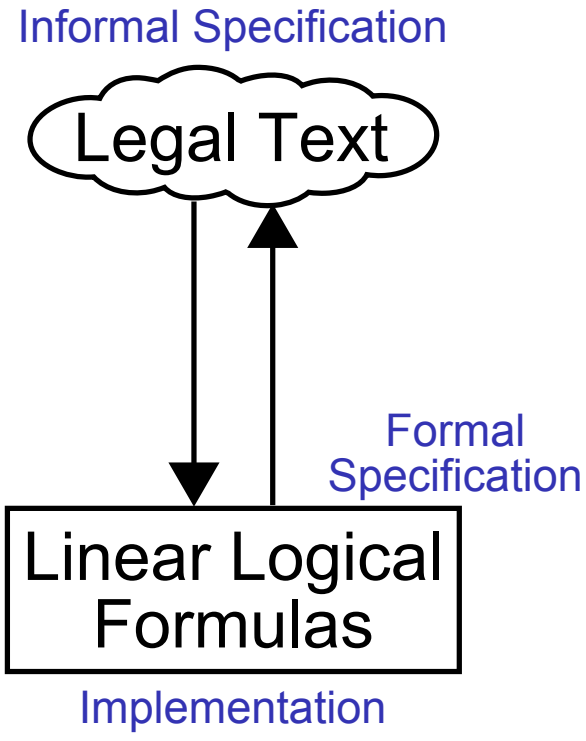

1. Translate legal text to logical formulas.
   - Algorithms at high level of abstraction
   - Much smaller gap from legal language!
2. Transliterate formulas to a logic program.
   - Formulas = source code

# Certifying Voting Protocols




1. Translate legal text to logical formulas.
   - Algorithms at high level of abstraction
   - Much smaller gap from legal language!
2. Transliterate formulas to a logic program.
   - Formulas = source code
   - No further translation necessary!

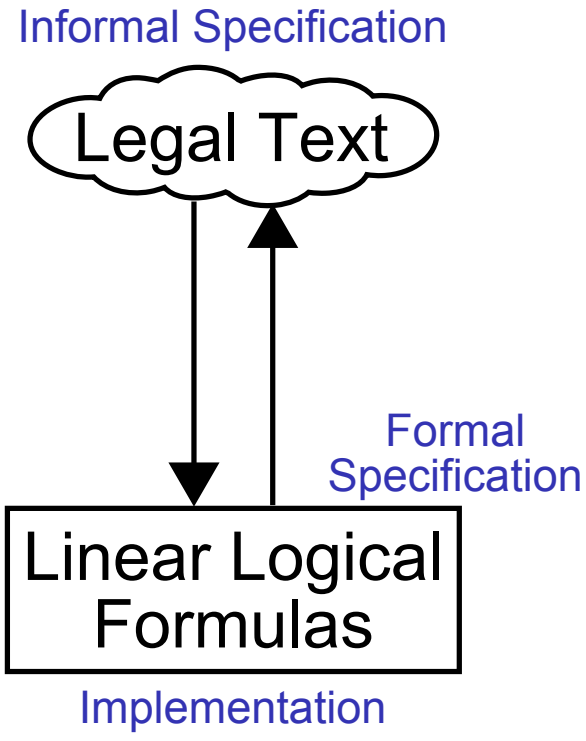
# Certifying Voting Protocols




## What must still be trusted?

1. Translation to logical formulas
   - Much smaller gap from legal language — more trustworthy!

# Certifying Voting Protocols




## What must still be trusted?

1. Translation to logical formulas
   - Much smaller gap from legal language — more trustworthy!
2. Correctness of logic programming engine
   - Equal to or easier than trusting compiler
   - Proof witnesses as certificates
   - Certificates as audit trails
   - Use a simpler proof checker to validate proof objects

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

# 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

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

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

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

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

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

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

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

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 :

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

- 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 :

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

- 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 :

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

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

# Conclusion

# Conclusion

- Elections are safety critical systems
- Decisions regarding trust are never only technical
- Even experts get it wrong
  - CADE-STV implements not STV but majority rule
  - Over 15 years in use, designed by mathematicians and logicians
  - Used for other professional meetings as well.
- Future Work
  - Epistemic connectives to model identiy and secrecy.

Thank you.