Certifying Voting Protocols

Carsten Schürmann

IT University of Copenhagen

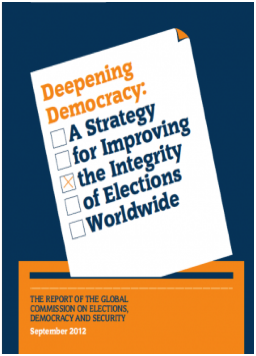
(technical part joined work with Henry deYoung)

ARSEC 2013

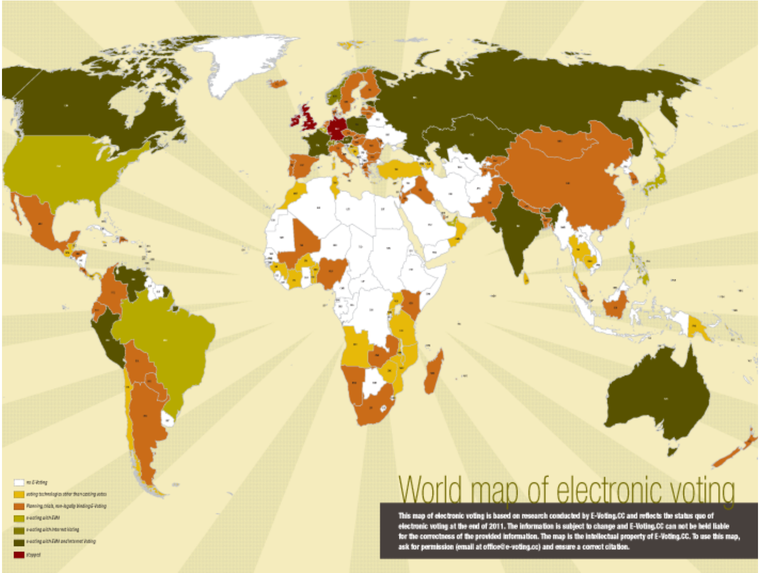
June 9, 2013

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
  - Integrity $\Leftrightarrow$ Secrecy
  - Attack surfaces
  - Software Independence
  - End to end verifiability
  - Programming language abstractions
- Selected Legal Challenges
  - Policy and law
- Selected Communication Challenges
  - Education and publication

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
  - Integrity $\Leftrightarrow$ Secrecy
  - 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

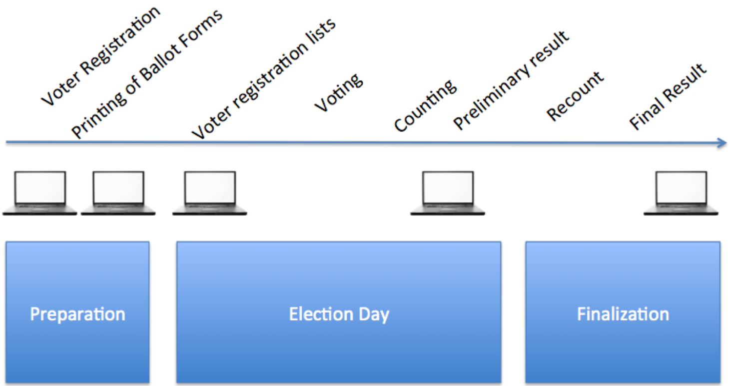
1. Case Study: Denmark

2. A Brief Introduction to Linear Logic

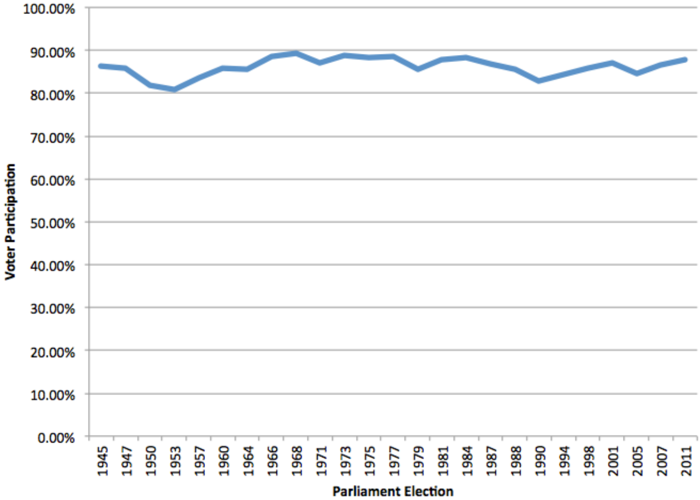
3. A Linear Logical Specification of Single Transferable Vote

Case Study: Denmark

A Bird's Eye View on the Electoral Process



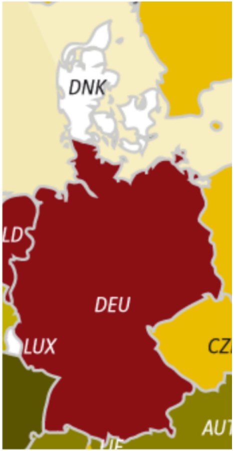
Managing Trust (Parliamentary Election)



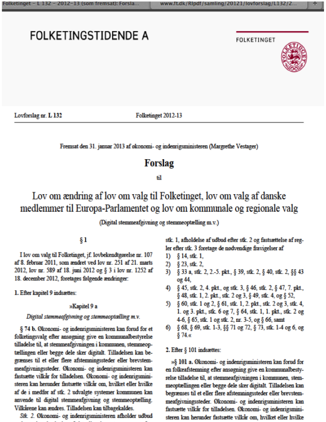
Technology Requires Law Changes... but It Ain't Easy

"The use of electronic voting machines in parliamentary elections is unconstitutional as long as it is not possible for citizens to exercise their right to inspect and verify the essential steps of the election.

[German Supreme Court March 3rd, 2009]



Denmark



Folketinget – L 132 – 2012-13 (som fremsat): Forsla... www.ft.dk/RIpdf/samling/20121/lovforslag/L132/2...

FOLKETINGSTIDENDE A

FOLKETINGET

Lovforslag nr. **L 132** Folketinget 2012-13

Fremsat den 31. januar 2013 af økonomi- og indenrigsministeren (Margrethe Vestager)

Forslag

til

Lov om ændring af lov om valg til Folketinget, lov om valg af danske medlemmer til Europa-Parlamentet og lov om kommunale og regionale valg

(Digital stemmeafgivning og stemmeoptælling m.v.)

§ 1

I lov om valg til Folketinget, jf. lovbekendtgørelse nr. 107 af 8. februar 2011, som ændret ved lov nr. 251 af 21. marts 2012, lov nr. 589 af 18. juni 2012 og § 3 i lov nr. 1252 af 18. december 2012, foretages følgende ændringer:

1. Efter kapitel 9 indsættes:

»Kapitel 9 a

Digital stemmeafgivning og stemmeoptælling m.v.

§ 74 b. Økonomi- og indenrigsministeren kan forud for et folketingsvalg efter ansøgning give en kommunalbestyrelse tilladelse til, at stemmeafgivningen i kommunen, stemmeoptællingen eller begge dele sker digitalt. Tilladelsen kan begrænses til et eller flere afstemningssteder eller brevstemmeafgivningssteder. Økonomi- og indenrigsministeren kan fastsætte vilkår for tilladelsen. Økonomi- og indenrigsministeren kan herunder fastsætte vilkår om, hvilket eller hvilke af de i medfør af stk. 2 udvalgte systemer kommunen kan anvende til digital stemmeafgivning og stemmeoptælling. Vilkårene kan ændres. Tilladelsen kan tilbagekaldes.

Stk. 2. Økonomi- og indenrigsministeren afholder udbud

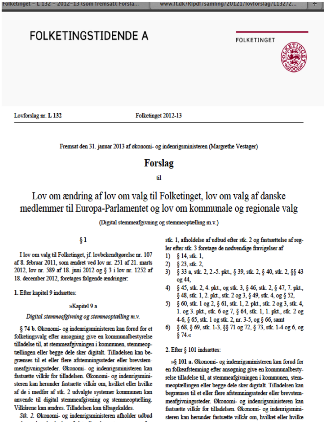
stk. 1, afholdelse af udbud efter stk. 2 og fastsættelse af regler efter stk. 3 foretage de nødvendige fravigelser af

1) § 14, stk. 1,
2) § 23, stk. 2,
3) § 33 a, stk. 2, 2.-5. pkt., § 39, stk. 2, § 40, stk. 2, §§ 43 og 44,
4) § 45, stk. 2, 4. pkt., og stk. 3, § 46, stk. 2, § 47, 7. pkt., § 48, stk. 1, 2. pkt., stk. 2 og 3, § 49, stk. 4, og § 52,
5) § 60, stk. 1 og 2, § 61, stk. 1, 2. pkt., stk. 2 og 3, stk. 4, 1. og 3. pkt., stk. 6 og 7, § 64, stk. 1, 1. pkt., stk. 2 og 4-6, § 65, stk. 1 og stk. 2, nr. 3-5, og § 66, samt
6) § 68, § 69, stk. 1-3, §§ 71 og 72, § 73, stk. 1-4 og 6, og § 74.«

2. Efter § 101 indsættes:

»**§ 101 a.** Økonomi- og indenrigsministeren kan forud for en folkeafstemning efter ansøgning give en kommunalbestyrelse tilladelse til, at stemmeafgivningen i kommunen, stemmeoptællingen eller begge dele sker digitalt. Tilladelsen kan begrænses til et eller flere afstemningssteder eller brevstemmeafgivningssteder. Økonomi- og indenrigsministeren kan fastsætte vilkår for tilladelsen. Økonomi- og indenrigsministeren kan herunder fastsætte vilkår om, hvilket eller hvilke

Denmark



Folketinget - L 132 - 2012-13 (som fremsat): Forsla... www.ft.dk/RIpdf/samling/20121/lovforslag/L132/2...

FOLKETINGSTIDENDE A

FOLKETINGET

Lovforslag nr. L 132 Folketinget 2012-13

Fremsat den 31. januar 2013 af økonomi- og indenrigsministeren (Margrethe Vestager)

Forslag

til

Lov om ændring af lov om valg til Folketinget, lov om valg af danske medlemmer til Europa-Parlamentet og lov om kommunale og regionale valg

(Digital stemmeafgivning og stemmeoptælling m.v.)

§ 1

I lov om valg til Folketinget, jf. lovbekendtgørelse nr. 107 af 8. februar 2011, som ændret ved lov nr. 251 af 21. marts 2012, lov nr. 589 af 18. juni 2012 og § 3 i lov nr. 1252 af 18. december 2012, foretages følgende ændringer:

1. Efter kapitel 9 indsættes:

»Kapitel 9 a

Digital stemmeafgivning og stemmeoptælling m.v.

§ 74 b. Økonomi- og indenrigsministeren kan forud for et folketingsvalg efter ansøgning give en kommunalbestyrelse tilladelse til, at stemmeafgivningen i kommunen, stemmeoptællingen eller begge dele sker digitalt. Tilladelsen kan begrænses til et eller flere afstemningssteder eller brevstemmeafgivningssteder. Økonomi- og indenrigsministeren kan fastsætte vilkår for tilladelsen. Økonomi- og indenrigsministeren kan herunder fastsætte vilkår om, hvilket eller hvilke af de i medfør af stk. 2 udvalgte systemer kommunen kan anvende til digital stemmeafgivning og stemmeoptælling. Vilkårene kan ændres. Tilladelsen kan tilbagekaldes.

Stk. 2. Økonomi- og indenrigsministeren afholder udbud

stk. 1, afholdelse af udbud efter stk. 2 og fastsættelse af regler efter stk. 3 foretage de nødvendige fravigelser af

1) § 14, stk. 1,
2) § 23, stk. 2,
3) § 33 a, stk. 2, 2.-5. pkt., § 39, stk. 2, § 40, stk. 2, §§ 43 og 44,
4) § 45, stk. 2, 4. pkt., og stk. 3, § 46, stk. 2, § 47, 7. pkt., § 48, stk. 1, 2. pkt., stk. 2 og 3, § 49, stk. 4, og § 52,
5) § 60, stk. 1 og 2, § 61, stk. 1, 2. pkt., stk. 2 og 3, stk. 4, 1. og 3. pkt., stk. 6 og 7, § 64, stk. 1, 1. pkt., stk. 2 og 4-6, § 65, stk. 1 og stk. 2, nr. 3-5, og § 66, samt
6) § 68, § 69, stk. 1-3, §§ 71 og 72, § 73, stk. 1-4 og 6, og § 74.«

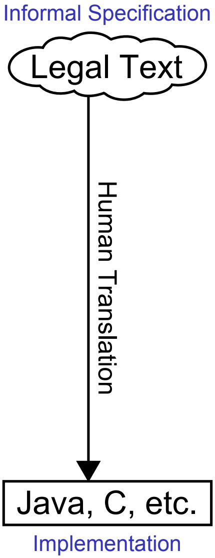
2. Efter § 101 indsættes:

»**§ 101 a.** Økonomi- og indenrigsministeren kan forud for en folkeafstemning efter ansøgning give en kommunalbestyrelse tilladelse til, at stemmeafgivningen i kommunen, stemmeoptællingen eller begge dele sker digitalt. Tilladelsen kan begrænses til et eller flere afstemningssteder eller brevstemmeafgivningssteder. Økonomi- og indenrigsministeren kan fastsætte vilkår for tilladelsen. Økonomi- og indenrigsministeren kan herunder fastsætte vilkår om, hvilket eller hvilke

Events

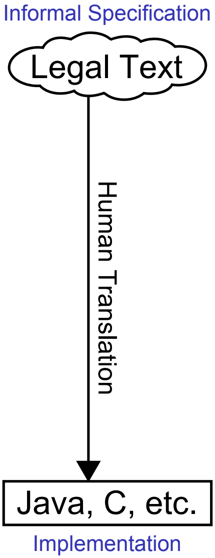
- Nov'12 Bill published
- Dec'12 Experts comment
- Jan'13 Ministry publishes rebuttal
- Feb'13 First hearing in parliament
- Mar'13 Expert hearing in parliament
- Apr'13 Majority in parliament against
- May'13 Oppostion proposes alternative
- to be continued

Tabulation Technology Denmark (since 1984)



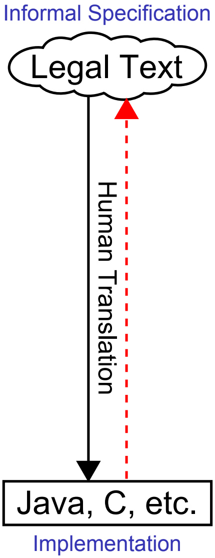
1. Translate legal text to imperative source code.

Tabulation Technology Denmark (since 1984)



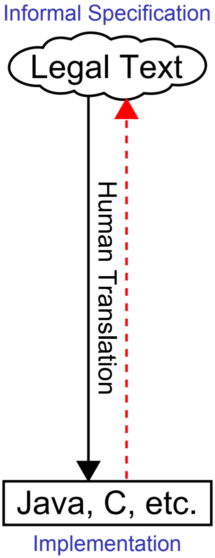
1. Translate legal text to imperative source code.
   - How to trust this?

Tabulation Technology Denmark (since 1984)



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

Tabulation Technology Denmark (since 1984)



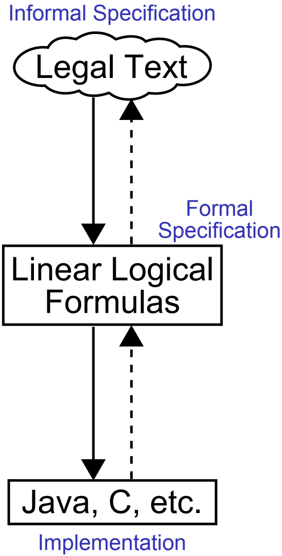
1. Translate legal text to imperative source code.
   - How to trust this?
2. Certify that code meets legal specification.
   - *Very* hard!

Is this approach really trustworthy?

Key Idea

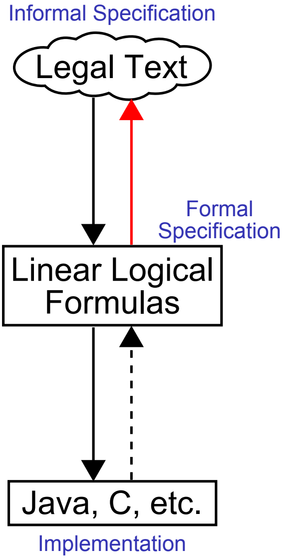
Formal logic, particularly linear logic, is well-suited to the trustworthy specification and implementation of voting protocols.

Our Approach to Electronic Elections



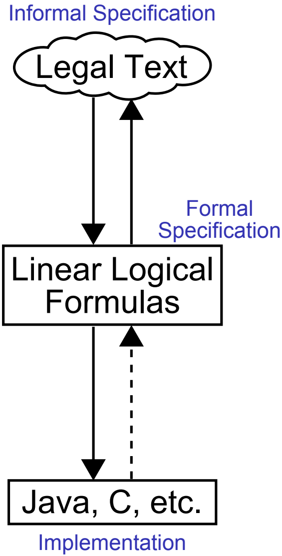
1. Translate legal text to logical formulas.
   - Algorithms at high level of abstraction

Our Approach to Electronic Elections



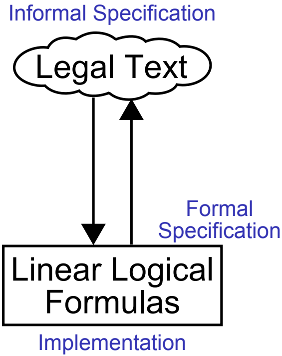
1. Translate legal text to logical formulas.
   - Algorithms at high level of abstraction
   - Much smaller gap from legal language!

Our Approach to Electronic Elections



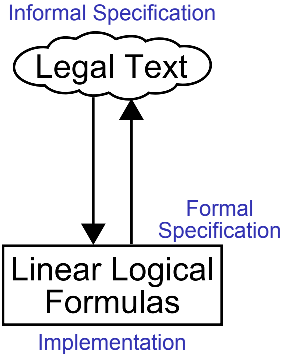
1. Translate legal text to logical formulas.
   - Algorithms at high level of abstraction
   - Much smaller gap from legal language!
2. Transliterate formulas to a logic program.
   - Formulas = source code

Our Approach to Electronic Elections



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!

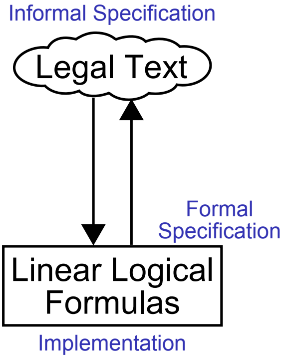
Our Approach to Electronic Elections



What must still be trusted?

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

Our Approach to Electronic Elections



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 are audit trails for free.
   - Use a simpler proof checker to validate proof objects.

Summary

Contributions:

- Full linear logical specifications of:
  - Single transferable vote (STV)
- Operational interpretation as Celf logic programs

Summary

Contributions:

- Full linear logical specifications of:
  - Single transferable vote (STV)
- Operational interpretation as Celf logic programs

Non-contributions:

- Focus is on *verified* elections, not *voter-verifiable* elections.
  - Complementary: E2E techniques detect errors at run time;
verified software minimizes run-time errors.
- Only operational correctness, not security properties
  - Linear logical voting specs should be readily amenable to meta-reasoning [Reed '09] about security; left to future work.

A Brief Introduction to Linear Logic

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.

Linear Logic

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.

Linear Logic

"Truth is a consumable resource."

- Assumptions must be used exactly once.

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.
- The logic of facts.

Linear Logic

"Truth is a consumable resource."

- Assumptions must be used exactly once.
- The logic of food.

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.
- The logic of facts.

Linear Logic

"Truth is a consumable resource."

- Assumptions must be used exactly once.
- The logic of ~~food~~ voting.

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.
- The logic of facts.

Linear Logic

"Truth is a consumable resource."

- Assumptions must be used exactly once.
- The logic of ~~food~~ voting.

Let's specify a voter check-in process:

- *Consume* an authorization card to prevent multiple check-ins.

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.
- The logic of facts.

Linear Logic

"Truth is a consumable resource."

- Assumptions must be used exactly once.
- The logic of ~~food~~ voting.

Let's specify a voter check-in process:

- *Consume* an authorization card to prevent multiple check-ins.
- Use linear implication, $A \multimap \{B\}$.
  - $A \multimap \{B\} \approx$ "*consume* resource A to produce B."

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.
- The logic of facts.

Linear Logic

"Truth is a consumable resource."

- Assumptions must be used exactly once.
- The logic of ~~food~~ voting.

Let's specify a voter check-in process:

- *Consume* an authorization card to prevent multiple check-ins.
- Use linear implication, $A \multimap \{B\}$.
  - $A \multimap \{B\} \approx$ "*consume* resource A to produce B."

$$\textit{voting-auth-card} \multimap \{\textit{blank-ballot}\}$$

"If I *give* an authorization card, then I *get* a blank ballot."

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.
- The logic of facts.

Linear Logic

"Truth is a consumable resource."

- Assumptions must be used exactly once.
- The logic of ~~food~~ voting.

Let's specify a voter check-in process:

- *Consume* an authorization card to prevent multiple check-ins.
- Use linear implication, $A \multimap \{B\}$.
  - $A \multimap \{B\} \approx$ "*consume* resource A to produce B."

$$\textit{voting-auth-card} \multimap \{\textit{blank-ballot}\}$$

"If I *give* an authorization card, then I *get* a blank ballot."

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.
- The logic of facts.

Linear Logic

"Truth is a consumable resource."

- Assumptions must be used exactly once.
- The logic of ~~food~~ voting.

Let's specify a voter check-in process:

- *Consume* an authorization card to prevent multiple check-ins.
- Use linear implication, $A \multimap \{B\}$.
  - $A \multimap \{B\} \approx$ "*consume* resource A to produce B."

$$\textit{voting-auth-card} \multimap \{\textit{blank-ballot}\}$$

"If I *give* an authorization card, then I *get* a blank ballot."

What is linear logic?

Traditional Logic

"Truth is free."

- Assumptions may be used any number of times.
- The logic of facts.

Linear Logic

"Truth is a consumable resource."

- Assumptions must be used exactly once.
- The logic of ~~food~~ voting.

Let's specify a voter check-in process:

- *Consume* an authorization card to prevent multiple check-ins.
- Use linear implication, $A \multimap \{B\}$.
  - $A \multimap \{B\} \approx$ "*consume* resource A to produce B."

$$\textit{voting-auth-card} \multimap \{\textit{blank-ballot}\}$$

"If I *give* an authorization card, then I *get* a blank ballot."

“May I please see your identification?”

$$\textit{voting-auth-card}\ \text{“and photo ID”} \multimap \{\textit{blank-ballot}\}$$

“If I give an auth. card and a photo ID, then I get a ballot.”

Problem: How to express a pair of resources?

“May I please see your identification?”

$$\mathit{voting\text{-}auth\text{-}card} \text{ “and photo ID”} \multimap \{\mathit{blank\text{-}ballot}\}$$

“If I give an auth. card and a photo ID, then I get a ballot.”

Problem: How to express a pair of resources?

Solution: Use simultaneous conjunction, $A \otimes B$.

- $A \otimes B \approx$ “both resources A and B”

“May I please see your identification?”

$$\textit{voting-auth-card} \otimes \textit{photo-ID} \multimap \{\textit{blank-ballot}\}$$

“If I give an auth. card and a photo ID, then I get a ballot.”

Problem: How to express a pair of resources?

Solution: Use simultaneous conjunction, $A \otimes B$.

- $A \otimes B \approx$ “both resources A and B”

"May I please see your identification?"

$$\textit{voting-auth-card} \otimes \textit{photo-ID} \multimap \{\textit{blank-ballot}\}$$

"If I give an auth. card and a photo ID, then I get a ballot."

Problem: How to express a pair of resources?

Solution: Use simultaneous conjunction, $A \otimes B$.

- $A \otimes B \approx$ "both resources A and B"

Problem: Linear implication incorrectly consumes the photo ID.
In linear logic, how do we only *show* the ID?

"May I please see your identification?"

$$\textit{voting-auth-card} \otimes \textit{photo-ID} \multimap \{\textit{blank-ballot}\}$$

"If I give an auth. card and a photo ID, then I get a ballot."

Problem: How to express a pair of resources?

Solution: Use simultaneous conjunction, $A \otimes B$.

- $A \otimes B \approx$ "both resources A and B"

Problem: Linear implication incorrectly consumes the photo ID. In linear logic, how do we only *show* the ID?

Solution: Use the unrestricted modality, $!A$.

- $!A \approx$ "a version of A that is never consumed."

“May I please see your identification?”

$$\textit{voting-auth-card} \otimes {!}\textit{photo-ID} \multimap \{\textit{blank-ballot}\}$$

“If I give an auth. card and show a photo ID, then I get a ballot.”

Problem: How to express a pair of resources?

Solution: Use simultaneous conjunction, $A \otimes B$.

- $A \otimes B \approx$ “both resources A and B”

Problem: Linear implication incorrectly consumes the photo ID.
In linear logic, how do we only *show* the ID?

Solution: Use the unrestricted modality, $!A$.

- $!A \approx$ “a version of A that is never consumed.”

Ensuring that the Card and ID Match

$$\textit{voting-auth-card} \otimes {!}\textit{photo-ID} \multimap \{\textit{blank-ballot}\}$$

"If I give an auth. card and show a photo ID, then I get a ballot."

Problem: Doesn't ensure that auth. card and photo ID match.

Ensuring that the Card and ID Match

$$\mathit{voting\text{-}auth\text{-}card} \otimes {!}\mathit{photo\text{-}ID} \multimap \{\mathit{blank\text{-}ballot}\}$$

"If I give an auth. card and show a photo ID, then I get a ballot."

Problem: Doesn't ensure that auth. card and photo ID match.

Solution: Use universal quantification, $\forall x.A$.

- Quantified variables are not resources.

Ensuring that the Card and ID Match

$$\forall v.\, \mathit{voting\text{-}auth\text{-}card}(v) \otimes !\mathit{photo\text{-}ID}(v) \multimap \{\mathit{blank\text{-}ballot}\}$$

"If I give an auth. card and show a matching ID, then I get a ballot."

Problem: Doesn't ensure that auth. card and photo ID match.

Solution: Use universal quantification, $\forall x.A$.

- Quantified variables are not resources.

Ensuring that the Card and ID Match

$$\textit{voting-auth-card}(V) \otimes\, !\textit{photo-ID}(V) \multimap \{\textit{blank-ballot}\}$$

"If I give an auth. card and show a matching ID, then I get a ballot."

Problem: Doesn't ensure that auth. card and photo ID match.

Solution: Use universal quantification, $\forall x.A$.

- Quantified variables are not resources.

Ensuring that the Card and ID Match

$$\textit{voting-auth-card}(V) \otimes !\textit{photo-ID}(V) \multimap \{\textit{blank-ballot}\}$$

"If I give an auth. card and show a matching ID, then I get a ballot."

Problem: Doesn't ensure that auth. card and photo ID match.

Solution: Use universal quantification, $\forall x.A$.

- Quantified variables are not resources.

Problem: Doesn't ensure that the auth. card and ID are mine.

Ensuring that the Card and ID Match

$$\textit{voting-auth-card}(V) \otimes\, !\textit{photo-ID}(V) \multimap \{\textit{blank-ballot}\}$$

"If I give an auth. card and show a matching ID, then I get a ballot."

Problem: Doesn't ensure that auth. card and photo ID match.

Solution: Use universal quantification, $\forall x.A$.

- Quantified variables are not resources.

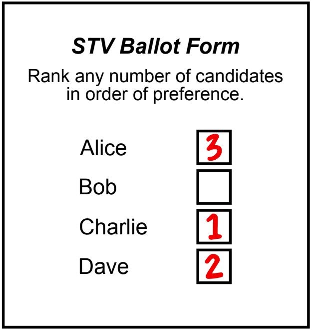
Problem: Doesn't ensure that the auth. card and ID are mine.

Future Work: Use possession modality [Garg^{+} '06], K has A.

- Location and secrecy of information?
  Homomorphic encryption?

A Linear Logical Specification of Single Transferable Vote

Single Transferable Vote (STV)



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.

Used in Australia and Ireland national elections

Single Transferable Vote on a Single Slide

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Single Transferable Vote on a Single Slide

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Single Transferable Vote on a Single Slide

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Single Transferable Vote on a Single Slide

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Aspect 1: 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 :

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

Aspect 1: 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 :

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

Aspect 1: 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 :

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

Aspect 1: 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 :

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

Aspect 1: 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 :

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

Aspect 1: 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 :

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

Aspect 1: 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 :

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

Aspect 1: 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 :

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

Aspect 1: 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.

Aspect 1: 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.

Aspect 1: 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 :

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

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

Aspect 1: From Legal Text to Formal Specification

Legal Text

"If a candidate reaches the quota, he is declared elected."

Detailed Reading

Otherwise, if we are tallying votes and there is an uncounted vote for C and C is a hopeful with running tally N and this vote would meet the quota and there is at least one seat left, then mark the ballot as counted and declare candidate C to be elected and tally the remaining $U-1$ ballots among the $H-1$ hopefuls and $S-1$ seats left.

tally/2 :

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

Aspect 1: From Legal Text to Formal Specification

Legal Text

"If a candidate reaches the quota, he is declared elected."

Detailed Reading

Otherwise, if we are tallying votes and there is an uncounted vote for C and C is a hopeful with running tally N and this vote would meet the quota and there is at least one seat left, then mark the ballot as counted and declare candidate C to be elected and tally the remaining $U{-}1$ ballots among the $H{-}1$ hopefuls and $S{-}1$ seats left.

tally/2 :

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

Aspect 1: From Legal Text to Formal Specification

Legal Text

"If a candidate reaches the quota, he is declared elected."

Detailed Reading

Otherwise, if we are tallying votes and there is an uncounted vote for C and C is a hopeful with running tally N and this vote would meet the quota and there is at least one seat left, then mark the ballot as counted and declare candidate C to be elected and tally the remaining $U{-}1$ ballots among the $H{-}1$ hopefuls and $S{-}1$ seats left.

tally/2 :

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

Aspect 1: From Legal Text to Formal Specification

Legal Text

"If a candidate reaches the quota, he is declared elected."

Detailed Reading

Otherwise, if we are tallying votes and there is an uncounted vote for C and C is a hopeful with running tally N and this vote would meet the quota and there is at least one seat left, then mark the ballot as counted and declare candidate C to be elected and tally the remaining $U-1$ ballots among the $H-1$ hopefuls and $S-1$ seats left.

tally/2 :

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

Aspect 1: From Legal Text to Formal Specification

Legal Text

"If a candidate reaches the quota, he is declared elected."

Detailed Reading

Otherwise, if we are tallying votes and there is an uncounted vote for C and C is a hopeful with running tally N and this vote would meet the quota and there is at least one seat left, then mark the ballot as counted and declare candidate C to be elected and tally the remaining $U-1$ ballots among the $H-1$ hopefuls and $S-1$ seats left.

tally/2 :

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

Aspect 1: From Legal Text to Formal Specification

Legal Text

"If a candidate reaches the quota, he is declared elected."

Detailed Reading

Otherwise, if we are tallying votes and there is an uncounted vote for C and C is a hopeful with running tally N and this vote would meet the quota and there is at least one seat left, then mark the ballot as counted and declare candidate C to be elected and tally the remaining $U-1$ ballots among the $H-1$ hopefuls and $S-1$ seats left.

tally/2 :

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

Aspect 1: From Legal Text to Formal Specification

Legal Text

"If a candidate reaches the quota, he is declared elected."

Detailed Reading

Otherwise, if we are tallying votes and there is an uncounted vote for C and C is a hopeful with running tally N and this vote would meet the quota and there is at least one seat left, then mark the ballot as counted and declare candidate C to be elected and tally the remaining $U-1$ ballots among the $H-1$ hopefuls and $S-1$ seats left.

tally/2 :

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

Aspect 1: From Legal Text to Formal Specification

Legal Text

"If a candidate reaches the quota, he is declared elected."

Detailed Reading

Otherwise, if we are tallying votes and there is an uncounted vote for C and C is a hopeful with running tally N and this vote would meet the quota and there is at least one seat left, then mark the ballot as counted and declare candidate C to be elected and tally the remaining $U{-}1$ ballots among the $H{-}1$ hopefuls and $S{-}1$ seats left.

tally/2 :

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

- ! modality: once declared elected, always declared elected.

Aspect 1: From Legal Text to Formal Specification

Legal Text

"Any surplus votes go to the next preference listed on the ballot."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is either elected or defeated,
then transfer it to the next pref. C' and
tally the remaining U ballots.

tally/3 :

$$\begin{array}{l} \textit{tally-votes}(S, H, U) \otimes \\ \textit{uncounted-ballot}(C, [C' \mid L]) \otimes \\ (!\textit{elected}(C) \oplus !\textit{defeated}(C)) \\ \quad \multimap \{\textit{uncounted-ballot}(C', L) \otimes \\ \qquad \textit{tally-votes}(S, H, U)\} \end{array}$$

Aspect 1: From Legal Text to Formal Specification

Legal Text

"Any surplus votes go to the next preference listed on the ballot."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is either elected or defeated,
then transfer it to the next pref. C' and
tally the remaining U ballots.

tally/3 :

$$\begin{array}{l} \textit{tally-votes}(S, H, U) \otimes \\ \textit{uncounted-ballot}(C, [C' \mid L]) \otimes \\ (!\textit{elected}(C) \oplus !\textit{defeated}(C)) \\ \quad \multimap \{\textit{uncounted-ballot}(C', L) \otimes \\ \qquad \textit{tally-votes}(S, H, U)\} \end{array}$$

Aspect 1: From Legal Text to Formal Specification

Legal Text

"Any surplus votes go to the next preference listed on the ballot."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is either elected or defeated,
then transfer it to the next pref. C' and
tally the remaining U ballots.

tally/3 :

$$
\begin{array}{l}
\textit{tally-votes}(S, H, U) \otimes \\
\textit{uncounted-ballot}(C, [C' \mid L]) \otimes \\
(!\textit{elected}(C) \oplus !\textit{defeated}(C)) \\
\quad \multimap \{\textit{uncounted-ballot}(C', L) \otimes \\
\qquad \textit{tally-votes}(S, H, U)\}
\end{array}
$$

Aspect 1: From Legal Text to Formal Specification

Legal Text

"Any surplus votes go to the next preference listed on the ballot."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is either elected or defeated,
then transfer it to the next pref. C' and
tally the remaining U ballots.

tally/3 :

$$
\begin{array}{l}
\mathit{tally\text{-}votes}(S, H, U) \otimes \\
\mathit{uncounted\text{-}ballot}(C, [C' \mid L]) \otimes \\
(!\mathit{elected}(C) \oplus !\mathit{defeated}(C)) \\
\quad \multimap \{\mathit{uncounted\text{-}ballot}(C', L) \otimes \\
\qquad \mathit{tally\text{-}votes}(S, H, U)\}
\end{array}
$$

Aspect 1: From Legal Text to Formal Specification

Legal Text

"Any surplus votes go to the next preference listed on the ballot."

Detailed Reading

If we are tallying votes and
there is an uncounted vote for C and
C is either elected or defeated,
then transfer it to the next pref. C' and
tally the remaining U ballots.

tally/3 :

$$\begin{array}{l} \mathit{tally\text{-}votes}(S, H, U) \otimes \\ \mathit{uncounted\text{-}ballot}(C, [C' \mid L]) \otimes \\ (!\mathit{elected}(C) \oplus !\mathit{defeated}(C)) \\ \quad \multimap \{\mathit{uncounted\text{-}ballot}(C', L) \otimes \\ \qquad \mathit{tally\text{-}votes}(S, H, U)\} \end{array}$$

Aspect 2: From Formal Specification to Implementation

tally/1 :

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

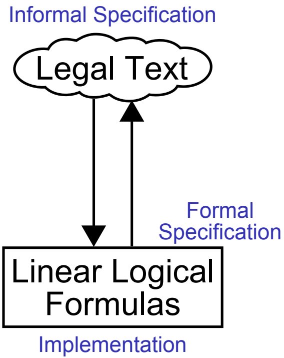
Celf Linear Logic Program

```
tally-votes S H (s U') *
uncounted-ballot C L *
hopeful C N *
!quota Q * !nat-less (s N) Q
   -o {counted-ballot C L *
       hopeful C (s N) *
       tally-votes S H U'}
```

- Transliteration of formula to logic programming syntax
- Complete Celf source code of STV available online at `http://www.itu.dk/~carsten/files/voteid2011.tgz`

Conclusion

Conclusion



Summary

Linear logic is well-suited to the trustworthy specification and implementation of voting protocols, including STV.

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
  - Come to my talk at CADE
- Awaiting access to the tabulation software for Denmark
- Future Work
  - Epistemic connectives to model identiy and secrecy.

Thank you.

www.demtech.dk

Twitter: @DemTechDK

begin/1 :

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

tally/1 :

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

tally/2 :

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

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

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

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

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

defeat-min/1 :

$$
\begin{array}{l}
\mathit{defeat\text{-}min}(S, H, M) \otimes \\
\mathit{hopeful}(C, N) \\
\quad \multimap \{\mathit{minimum}(C, N) \otimes \\
\qquad \mathit{defeat\text{-}min}(S, H{-}1, M{+}1)\}
\end{array}
$$

defeat-min/2 :

$$
\begin{array}{l}
\mathit{defeat\text{-}min}(S, 0, M) \\
\quad \multimap \{\mathit{defeat\text{-}min'}(S, 0, M)\}
\end{array}
$$

defeat-min′/1 :

$$
\begin{array}{l}
\mathit{defeat\text{-}min'}(S, H, M) \otimes \\
\mathit{minimum}(C_1, N_1) \otimes \\
\mathit{minimum}(C_2, N_2) \otimes \\
!(N_1 \leq N_2) \\
\quad \multimap \{\mathit{minimum}(C_1, N_1) \otimes \\
\qquad \mathit{hopeful}(C_2, N_2) \otimes \\
\qquad \mathit{defeat\text{-}min'}(S, H{+}1, M{-}1)\}
\end{array}
$$

defeat-min′/2 :

$$
\begin{array}{l}
\mathit{defeat\text{-}min'}(S, H, 1) \otimes \\
\mathit{minimum}(C, N) \\
\quad \multimap \{!\mathit{defeated}(C) \otimes \\
\qquad \mathit{transfer}(C, N, S, H, 0)\}
\end{array}
$$

transfer/1 :

$$\begin{array}{l} \mathit{transfer}(C, N, S, H, U) \otimes \\ \mathit{counted\text{-}ballot}(C, L) \\ \quad \multimap \{\mathit{uncounted\text{-}ballot}(C, L) \otimes \\ \qquad \mathit{transfer}(C, N{-}1, S, H, U{+}1)\} \end{array}$$

transfer/2 :

$$\begin{array}{l} \mathit{transfer}(C, 0, S, H, U) \\ \quad \multimap \{\mathit{tally\text{-}votes}(S, H, U)\} \end{array}$$

elect-all/1 :

$$\begin{array}{l} !\mathit{elect\text{-}all} \otimes \\ \mathit{hopeful}(C, N) \\ \quad \multimap \{!\mathit{elected}(C)\} \end{array}$$

Why not use traditional first-order logic?

To motivate the use of linear logic and illustrate its connectives, let's develop a specification of the voter check-in process.

- Let's try to use a traditional logical formula:

$$\textit{voting-auth-card} \rightarrow \textit{blank-ballot}$$

"If I have an authorization card, then I can have a blank ballot."

Why not use traditional first-order logic?

To motivate the use of linear logic and illustrate its connectives, let's develop a specification of the voter check-in process.

- Let's try to use a traditional logical formula:

$$\textit{voting-auth-card} \rightarrow \textit{blank-ballot}$$

"If I have an authorization card, then I can have a blank ballot."

Why not use traditional first-order logic?

To motivate the use of linear logic and illustrate its connectives, let's develop a specification of the voter check-in process.

- Let's try to use a traditional logical formula:

$$\textit{voting-auth-card} \rightarrow \textit{blank-ballot}$$

"If I have an authorization card, then I can have a blank ballot."

Why not use traditional first-order logic?

To motivate the use of linear logic and illustrate its connectives, let's develop a specification of the voter check-in process.

- Let's try to use a traditional logical formula:

$$\textit{voting-auth-card} \rightarrow \textit{blank-ballot}$$

"If I have an authorization card, then I can have a blank ballot."

Why not use traditional first-order logic?

To motivate the use of linear logic and illustrate its connectives, let's develop a specification of the voter check-in process.

- Let's try to use a traditional logical formula:

$$\textit{voting-auth-card} \rightarrow \textit{blank-ballot}$$

  "If I have an authorization card, then I can have a blank ballot."

- Not quite right. This formula allows the voter to keep his card:

$$\vdash \textit{voting-auth-card} \rightarrow \textit{blank-ballot} \wedge \textit{voting-auth-card}$$

Why not use traditional first-order logic?

To motivate the use of linear logic and illustrate its connectives, let's develop a specification of the voter check-in process.

- Let's try to use a traditional logical formula:

$$\textit{voting-auth-card} \rightarrow \textit{blank-ballot}$$

"If I have an authorization card, then I can have a blank ballot."

- Not quite right. This formula allows the voter to keep his card:

$$\vdash \textit{voting-auth-card} \rightarrow \textit{blank-ballot} \wedge \textit{voting-auth-card}$$

- Worse, it allows ballot stuffing:

$$\vdash \textit{voting-auth-card} \rightarrow \textit{blank-ballot} \wedge \cdots \wedge \textit{blank-ballot} \wedge \textit{voting-auth-card}$$

Why not use traditional first-order logic?

To motivate the use of linear logic and illustrate its connectives, let's develop a specification of the voter check-in process.

- Let's try to use a traditional logical formula:

$$\textit{voting-auth-card} \rightarrow \textit{blank-ballot}$$

"If I have an authorization card, then I can have a blank ballot."

- Not quite right. This formula allows the voter to keep his card:

$$\vdash \textit{voting-auth-card} \rightarrow \textit{blank-ballot} \wedge \textit{voting-auth-card}$$

- Worse, it allows ballot stuffing:

$$\vdash \textit{voting-auth-card} \rightarrow \textit{blank-ballot} \wedge \cdots \wedge \textit{blank-ballot} \wedge \textit{voting-auth-card}$$

Solution: Use linear logic!