NatSTV: Towards Verification of Natural Strategic Ability

Mateusz Kamiński, Damian Kurpiewski
Wojtek Jamroga

ZTSRiO 08.05.2025

Motivation

Analyzing voting protocols

Voting / e-voting protocol

- Cryptography
- Procedures
- Attacker/intruder
- Security
- Human factor

Human factor

- Makes a mistake
- Ignores instructions
- Skips the procedure, because it’s too complex, time-consuming, hard to understand…
- Can affect the security of the system

Analyzing the voting system

01

Create the (simplified) model of the system

02

Focus on the voter's behavior and her point of view

03

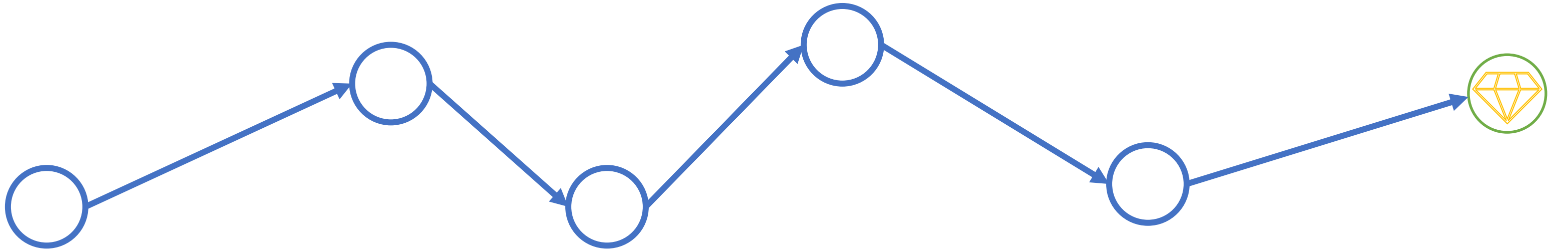
Describe requirements using ATL/NatATL formulae

04

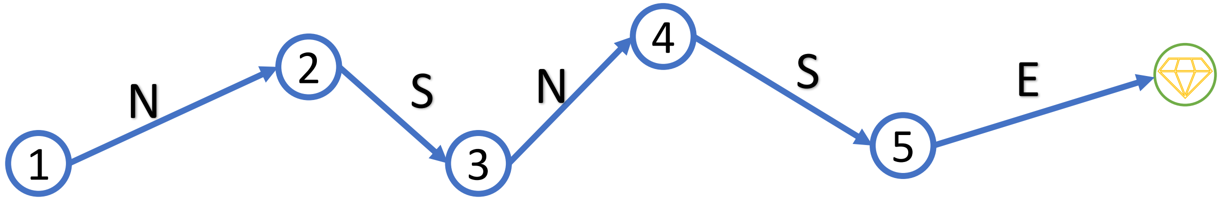
Create natural strategies for the voter (and other agents)

Strategy

- A plan
- A path in the model



Strategy description



- 1: N
- 2: S
- 3: N
- 4: S
- 5: E

Classic strategy

- Complex
- Long
- Easy for the computer
- Hard for the human

Natural Strategy

Conditional plan

Decisions are based on some observations

Based on the human behavior

Natural Strategy for the Voter

1. Out of the polling station -> go to the polling station

2. Empty ballot -> fill your ballot

3. Filled ballot -> cast your vote

Strategy in reality

- Understand the rules of the voting procedure
- Check if your vote is correct
- Verify that your vote has been counted correctly
- Sign-in to the e-voting system
- And much more …

Background

Logics and strategies

ATL: What Agents Can Achieve

- ATL: Alternating-time Temporal Logic [Alur et al. 1997-2002]
- Temporal logic meets game theory
- Main idea: <u>cooperation modalities</u>
- $\langle\langle \boldsymbol{A} \rangle\rangle \boldsymbol{\phi}$**: coalition** $\boldsymbol{A}$ **has a collective strategy to enforce** $\boldsymbol{\phi}$
- ϕ can include temporal operators: X (next), F (sometime in the future), G (always in the future), U (strong until)

Example Formula

- $\langle\langle \boldsymbol{Client} \rangle\rangle \boldsymbol{F}\ \boldsymbol{ticket}$
- Client can eventually buy a ticket

Strategy

A *strategy* of agent $a \in \mathbb{A}\mathrm{gt}$ is a conditional plan that specifies what a is going to do in every possible situation.

Formally, a perfect information memoryless strategy for a can be represented by a function $s_a\colon St \rightarrow Act$ satisfying $s_a(q) \in d_a(q)$ for each $q \in St$.

Strategy

A *strategy* of agent $a \in \mathbb{A}\mathrm{gt}$ is a conditional plan that specifies what a is going to do in every possible situation.

Formally, a perfect information memoryless strategy for a can be represented by a function $s_a : St \to Act$ satisfying $s_a(q) \in d_a(q)$ for each $q \in St$.

An *imperfect information memoryless strategy* additionally satisfies $s_a(q) = s_a(q')$ whenever $q \sim_a q'$

Natural ATL

- Strategies in a form of a set of simple conditions: guarded actions
- Strategy complexity represented as the total lengths of guards in the strategy
- $\langle\langle A \rangle\rangle^{\leq k}\phi$: **coalition A has a collective strategy of size less or equal than k to enforce ϕ**
- $\langle\langle Client \rangle\rangle^{\leq 10} F\ ticket$
- Client can buy a ticket by a strategy of complexity at most 10

Example Strategy

1. $\neg ticket \wedge \neg selected \wedge \neg paid \wedge \neg error \rightarrow \boldsymbol{select}$

2. $selected \rightarrow \boldsymbol{pay}$

3. $\top \rightarrow \boldsymbol{idle}$

Example Strategy Complexity

1. $\neg ticket \land \neg selected \land \neg paid \land \neg error \rightarrow \boldsymbol{select}$

$$\boldsymbol{cost = 11}$$

2. $selected \rightarrow \boldsymbol{pay}$

$$\boldsymbol{cost = 1}$$

3. $\top \rightarrow \boldsymbol{idle}$

$$\boldsymbol{cost = 1}$$

Complexity: $\mathbf{11 + 1 + 1 = 13}$

Case Study

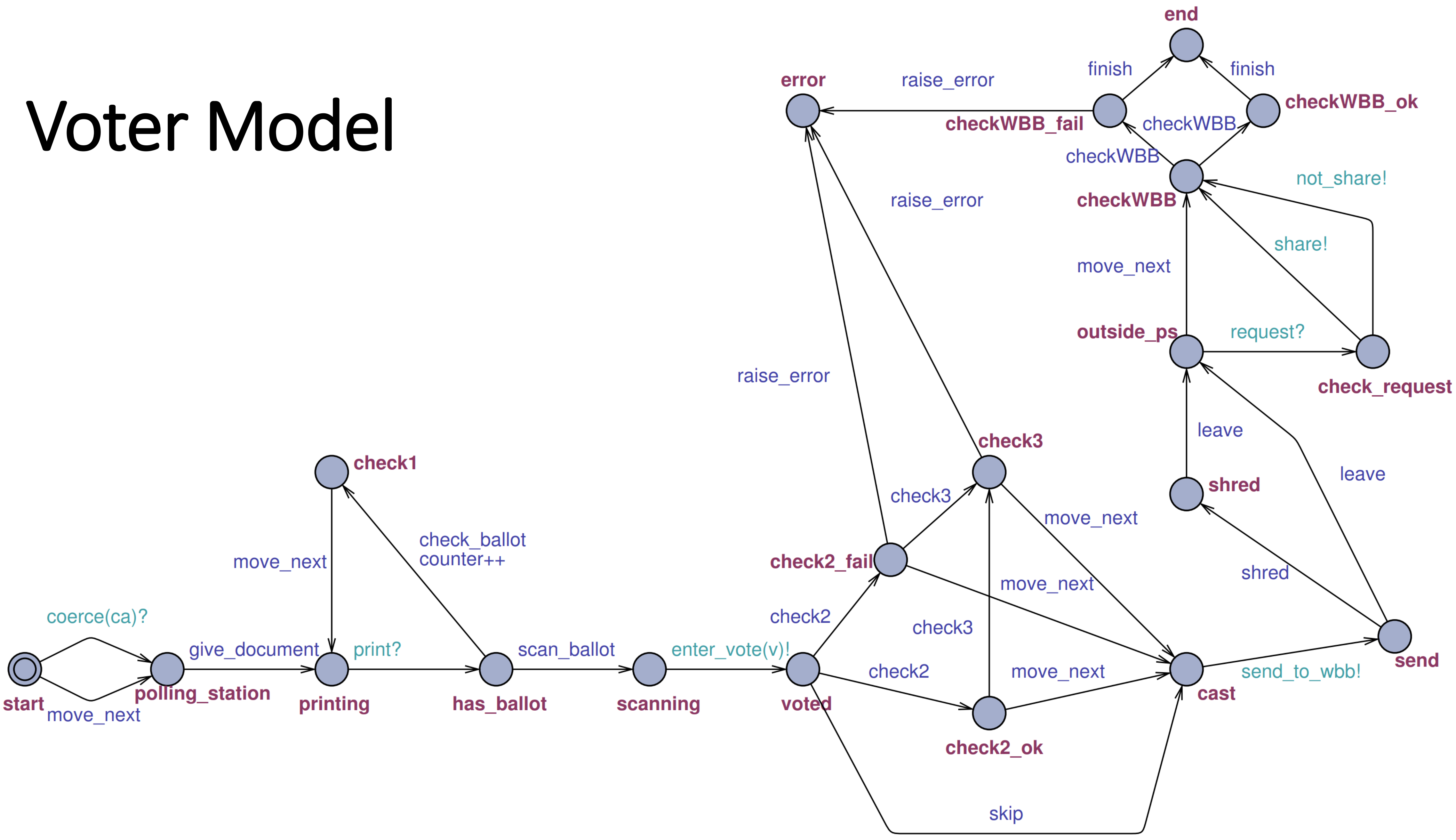
vVote voting system

ZTSRiO 08.05.2025

Example case study: vVote

- Implementation of $Prêt\ á\ Voter$ protocol
- Used for remote voting and voting of handicapped persons in the Australian state of Victoria elections in November 2014
- **Main idea**: encoding the vote using a randomized candidate list

Voter Model



end
finish
finish
error
raise_error
checkWBB_ok
checkWBB_fail
checkWBB
checkWBB
not_share!
raise_error
checkWBB
share!
move_next
outside_ps
request?
raise_error
check_request
leave
check3
leave
check1
shred
check3
move_next
check_ballot
counter++
move_next
check2_fail
move_next
shred
check2
check3
coerce(ca)?
give_document
print?
scan_ballot
enter_vote(v)!
send
check2
move_next
send_to_wbb!
start
polling_station
printing
has_ballot
scanning
voted
cast
move_next
check2_ok
skip

$$\varphi_1 = \langle\langle voter \rangle\rangle^{\leqslant k} \mathrm{F}(\textsf{checkWBB\_ok} \vee \textsf{checkWBB\_fail})$$

$$
\begin{array}{rl}
(1) & \textsf{start} \vee \textsf{check2\_ok} \vee \textsf{check2\_fail} \vee \textsf{outside\_ps} \rightsquigarrow \mathit{move\_next} \\
(2) & \textsf{polling\_station} \rightsquigarrow \mathit{give\_document} \\
(3) & \textsf{has\_ballot} \rightsquigarrow \mathit{scan\_ballot} \\
(4) & \textsf{scanning} \rightsquigarrow \mathit{enter\_vote}(v) \\
(5) & \textsf{voted} \rightsquigarrow \mathit{check2} \\
(6) & \textsf{cast} \rightsquigarrow \mathit{send\_to\_wbb} \\
(7) & \textsf{send} \rightsquigarrow \mathit{shred} \\
(8) & \textsf{shred} \rightsquigarrow \mathit{leave} \\
(9) & \textsf{check\_request} \rightsquigarrow \mathit{not\_share} \\
(10) & \textsf{checkWBB} \rightsquigarrow \mathit{checkWBB} \\
(11) & \top \rightsquigarrow \star
\end{array}
$$

Complexity

- 11 guarded commands
- (1) $\text{start} \vee check2\_ok \vee check2\_fail \vee outside\_ps$: cost 7
- Other guarded commands cost 1
- Total complexity: $1 * 10 + 7 * 1 = \mathbf{17}$
- <u>The formula φ_1 is true with any k of 17 and more</u>

Example construction of the strategy for φ_1

(1) $\mathsf{has\_ballot} \rightsquigarrow \mathit{scan\_ballot}$

(2) $\neg\mathsf{has\_ballot} \wedge \mathsf{scanning} \rightsquigarrow \mathit{enter\_vote}$

(3) $\neg\mathsf{has\_ballot} \wedge \neg\mathsf{scanning} \wedge \mathsf{voted} \rightsquigarrow \mathit{check2}$

(4) $\neg\mathsf{has\_ballot} \wedge \neg\mathsf{scanning} \wedge \neg\mathsf{voted} \wedge (\mathsf{check2\_ok} \vee \mathsf{check2\_fail}) \rightsquigarrow \mathit{move\_next}$

(5) $\neg\mathsf{has\_ballot} \wedge \neg\mathsf{scanning} \wedge \neg\mathsf{voted} \wedge \neg(\mathsf{check2\_ok} \vee \mathsf{check2\_fail}) \wedge \mathsf{cast} \rightsquigarrow \mathit{send\_to\_wbb}$

(6) $\neg\mathsf{has\_ballot} \wedge \neg\mathsf{scanning} \wedge \neg\mathsf{voted} \wedge \neg(\mathsf{check2\_ok} \vee \mathsf{check2\_fail}) \wedge \neg\mathsf{cast} \wedge \mathsf{send} \rightsquigarrow \mathit{shred}$

(7) $\neg\mathsf{has\_ballot} \wedge \neg\mathsf{scanning} \wedge \neg\mathsf{voted} \wedge \neg(\mathsf{check2\_ok} \vee \mathsf{check2\_fail}) \wedge \neg\mathsf{cast} \wedge \neg\mathsf{send} \wedge \mathsf{shred} \rightsquigarrow \mathit{leave}$

(8) $\neg\mathsf{has\_ballot} \wedge \neg\mathsf{scanning} \wedge \neg\mathsf{voted} \wedge \neg(\mathsf{check2\_ok} \vee \mathsf{check2\_fail}) \wedge \neg\mathsf{cast} \wedge \neg\mathsf{send} \wedge \neg\mathsf{shred} \wedge \mathsf{check\_request} \rightsquigarrow \mathit{not\_share}$

(9) $\neg\mathsf{has\_ballot} \wedge \neg\mathsf{scanning} \wedge \neg\mathsf{voted} \wedge \neg(\mathsf{check2\_ok} \vee \mathsf{check2\_fail}) \wedge \neg\mathsf{cast} \wedge \neg\mathsf{send} \wedge \neg\mathsf{shred} \wedge \neg\mathsf{check\_request} \wedge \mathsf{checkWBB} \rightsquigarrow \mathit{checkWBB}$

(10) $\neg\mathsf{has\_ballot} \wedge \neg\mathsf{scanning} \wedge \neg\mathsf{voted} \wedge \neg(\mathsf{check2\_ok} \vee \mathsf{check2\_fail}) \wedge \neg\mathsf{cast} \wedge \neg\mathsf{send} \wedge \neg\mathsf{shred} \wedge \neg\mathsf{check\_request} \wedge \neg\mathsf{checkWBB} \rightsquigarrow \star$

Challenges

Problems to solve

Finding (one of possibly many) natural strategy for the given formulae (if the strategy exists)

Minimizing the representation/complexity of the found strategy

Problems to solve

Finding (one of possibly many) natural strategy for the given formulae (if the strategy exists)

Minimazing the representation/complexity of the found strategy

Strategy representation example 1

q1	q2	q3	q4	act
1	0	0	0	A
0	1	1	0	B
0	1	0	0	C

Strategy representation example 1

q1	q2	q3	q4	act
1	0	0	0	A
0	1	1	0	B
0	1	0	0	C

After reduction:

q1	q3	act
1		A
	1	B
		C

Strategy representation example 1

q1	q2	q3	q4	act
1	0	0	0	A
0	1	1	0	B
0	1	0	0	C

After reduction:

q1	q3	act
1		A
	1	B
		C

Natural strategy:

1. $q1 \rightarrow \boldsymbol{A}$
2. $q3 \rightarrow \boldsymbol{B}$
3. $\top \rightarrow \boldsymbol{C}$

Strategy representation example 2

q1	q2	q3	q4	act
1	0	0	0	A
0	1	0	1	A
1	1	0	0	B
0	1	1	0	B

Strategy representation example 2

q1	q2	q3	q4	act
1	0	0	0	A
0	1	0	1	A
1	1	0	0	B
0	1	1	0	B

After reduction:

q1	q2	q4	act
1	1		B
1			A
		1	A
			B

Strategy representation example 2

q1	q2	q3	q4	act
1	0	0	0	A
0	1	0	1	A
1	1	0	0	B
0	1	1	0	B

After reduction:

q1	q2	q4	act
1	1		B
1			A
		1	A
			B

Natural strategy:

1. $q1 \wedge q2 \rightarrow \boldsymbol{B}$
2. $q1 \vee q4 \rightarrow \boldsymbol{A}$
3. $\top \rightarrow \boldsymbol{B}$

Experimental evaluation

Simple Voting with 2 candidates

$$\phi_1 \equiv \langle\langle c \rangle\rangle \Box((finish_k \wedge vote_{i,j}) \rightarrow pun_i)$$
$$\phi_2 \equiv \langle\langle v_i \rangle\rangle \Box(\neg K_c vote_{i,j})$$
$$\phi_3 \equiv \langle\langle v_i \rangle\rangle \Box(finish_i \rightarrow vote_{i,j} \wedge \neg K_c vote_{i,j})$$

#V	**Model generation**	ϕ_1				ϕ_2				ϕ_3			
		General synthesis	**Natural synthesis**	**Compl. raw**	**Compl. optimized**	**General synthesis**	**Natural synthesis**	**Compl. raw**	**Compl. optimized**	**General synthesis**	**Natural synthesis**	**Compl. raw**	**Compl. optimized**
1	0.03	<0.01	<0.01	156	26	<0.01	<0.01	9	3	<0.01	<0.01	9	3
2	0.05	<0.01	<0.01	991	131	<0.01	<0.01	9	3	<0.01	<0.01	9	3
3	0.21	0.15	0.15	4516	512	0.01	0.04	9	3	0.02	0.03	9	3
4	5.89	5.25	5.48	18043	1831	0.02	0.02	9	3	0.04	0.05	9	3
5	254.98	memout				25.02	10.15	9	3	28.56	12.68	9	3
6	timeout	-	-	-	-	-	-	-	-	-	-	-	-

Natural strategy for ϕ_3 found by STV

1. $\neg vote_{1,2} \rightarrow \boldsymbol{vote_2}$

$$\boldsymbol{cost\ = 2}$$

2. $\top \rightarrow \boldsymbol{idle}$

$$\boldsymbol{cost\ =\ 1}$$

Complexity: $\mathbf{2 + 1 = 3}$

vVote with 2 candidates

$$\phi_4 \equiv \langle\langle v_1 \rangle\rangle \Diamond (checkWBB\_ok \vee checkWBB\_notok)$$
$$\phi_5 \equiv \langle\langle v_1, c \rangle\rangle \Diamond (vote_{1,1} \wedge K_c vote_{1,1}).$$

#V	gen.	ϕ_4				ϕ_5			
		Strat.	Strat. nat.	Cmp. std.	Cmp. red.	Strat.	Strat. nat.	Cmp. std.	Cmp. red.
1	0.04	<0.01	0.06	797	39	<0.01	0.02	863	42
2	0.24	<0.01	0.26	2170	124	<0.01	0.04	851	38
3	9.02	0.43	0.54	2105	122	0.22	0.41	851	38
4	526.16	29.55	21.83	2170	124	18.64	18.81	851	38
5	timeout	-	-	-	-	-	-	-	-

Conclusions

- It’s not enough that a voter has a strategy – **complexity** is important
- Natural Strategy complexity helps to estimate the mental difficulty
- Other important factors exists: time, money, etc.
- The presented methodology can be applied outside the e-voting domain
- STV can be used to find natural strategy, if one exists

Thank You