Formal Analysis of Population Protocols

Michael Blondin

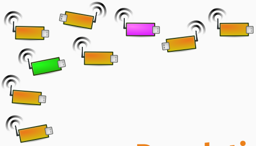
Joint work with Javier Esparza, Stefan Jaax,
Antonín Kučera and Philipp J. Meyer

Technical University of Munich

Overview

Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

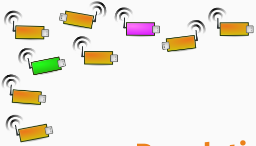
Overview



Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

Can model *e.g.* networks of passively mobile sensors and chemical reaction networks

Overview



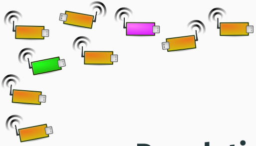
Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

Can model *e.g.* networks of passively mobile sensors and chemical reaction networks

Protocols compute predicates of the form $\varphi\colon \mathbb{N}^d \to \{0, 1\}$

e.g. if φ is unary, then $\varphi(n)$ is computed by n agents

Overview



Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

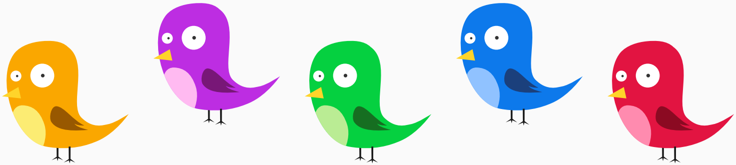
This talk: overview of recent advances on the formal analysis of population protocols

- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion

Population protocols

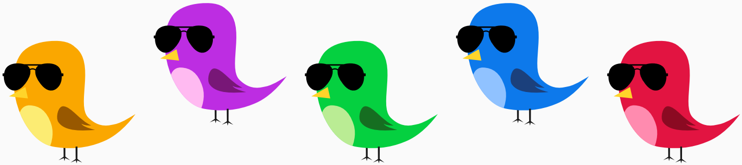
Angluin *et al.* PODC'04

- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion



Population protocols

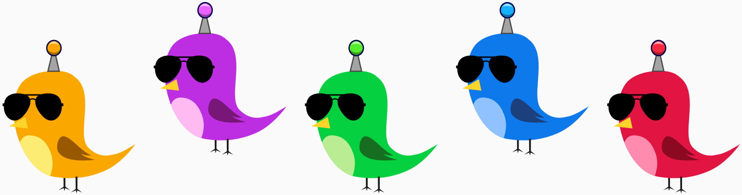
- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion



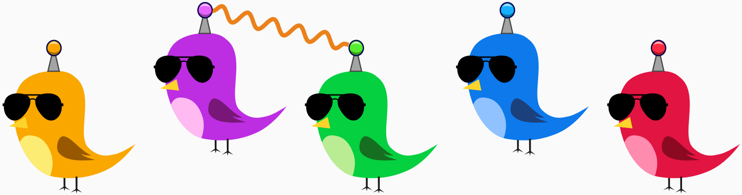
Population protocols

- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion

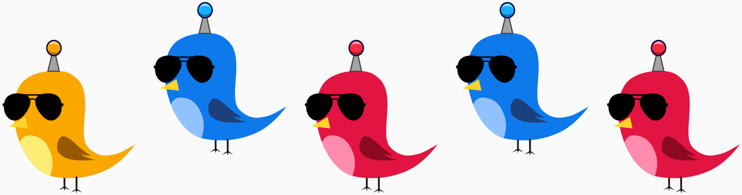
- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion



- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion



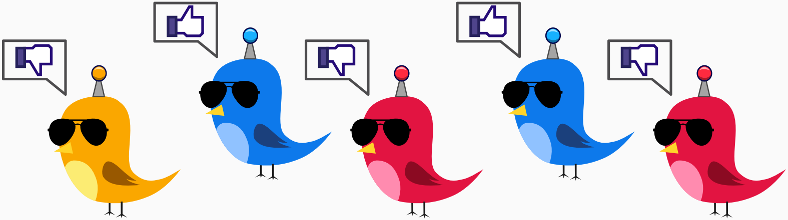
- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion



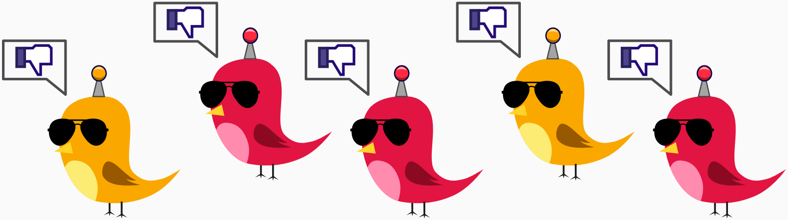
Population protocols

Angluin *et al.* PODC'04

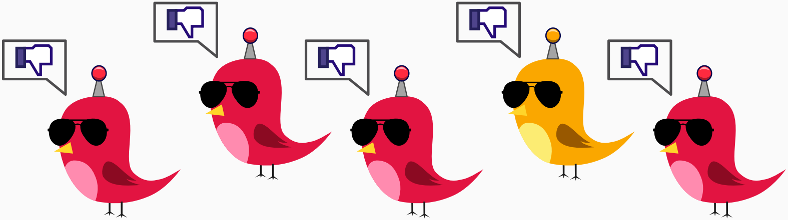
- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion



- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion



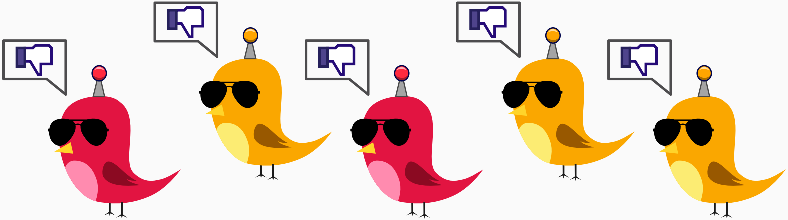
- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion



Population protocols

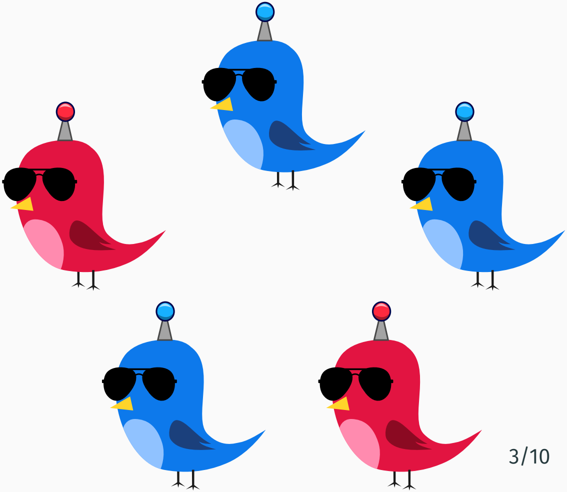
Angluin *et al.* PODC'04

- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions
- each agent has opinion true/false
- computes by stabilizing agents to some opinion



Example: majority protocol

At least as many blue birds than red birds?

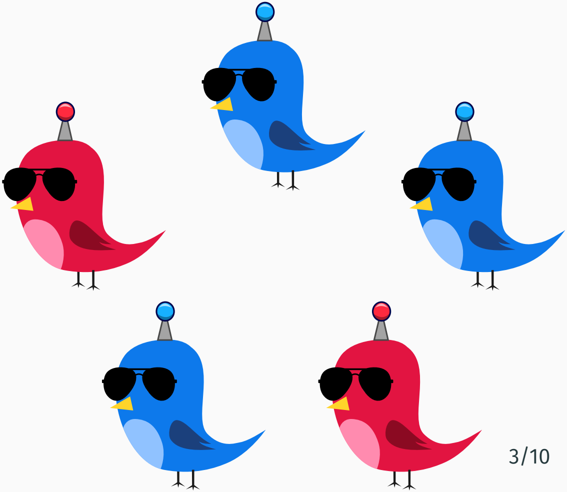


Example: majority protocol

At least as many blue birds than red birds?

Protocol:

- Two large birds of different colors become small and blue
- Large birds convert small birds to their color

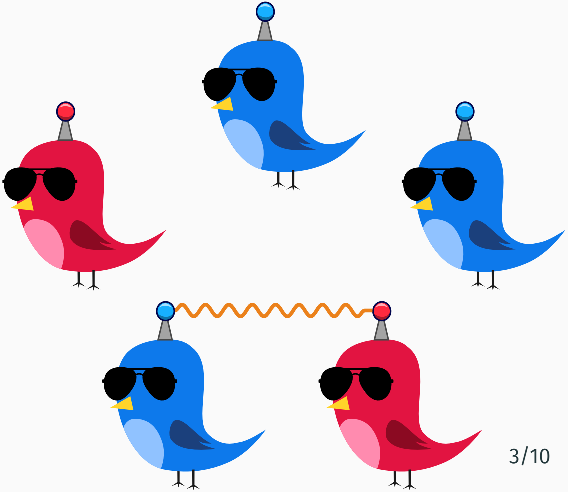


Example: majority protocol

At least as many blue birds than red birds?

Protocol:

- Two large birds of different colors become small and blue
- Large birds convert small birds to their color



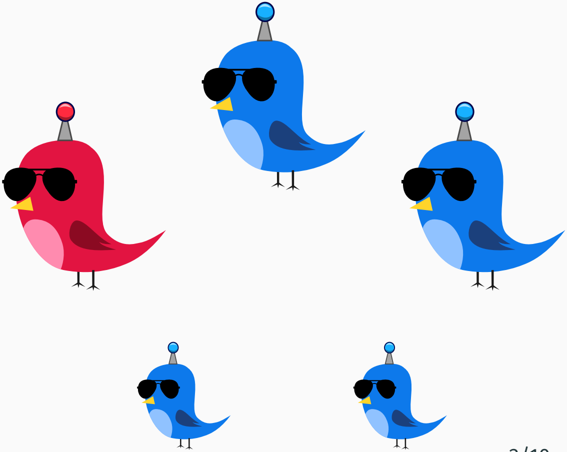
Example: majority protocol

At least as many blue birds than red birds?

Protocol:

- Two large birds of different colors become small and blue

- Large birds convert small birds to their color

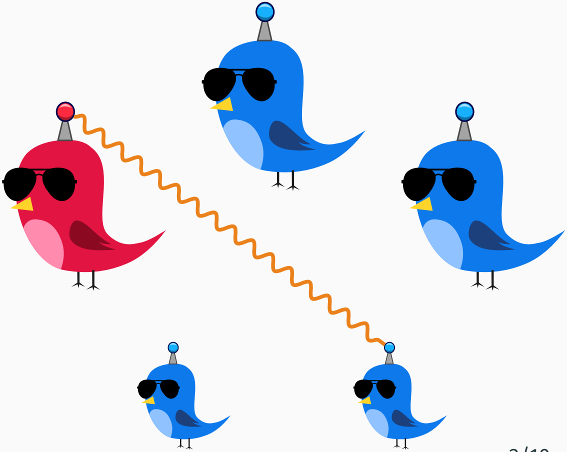


Example: majority protocol

At least as many blue birds than red birds?

Protocol:

- Two large birds of different colors become small and blue
- Large birds convert small birds to their color

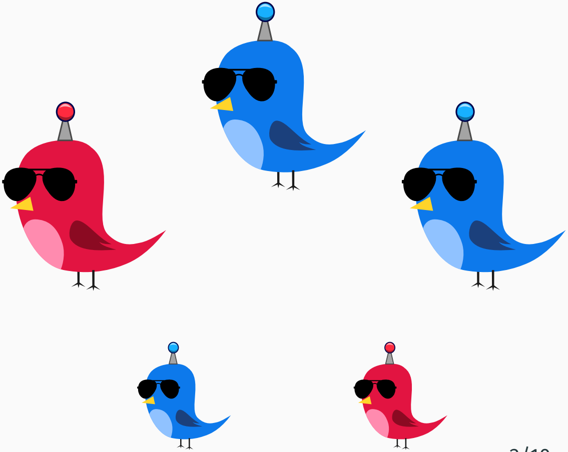


Example: majority protocol

At least as many blue birds than red birds?

Protocol:

- Two large birds of different colors become small and blue
- Large birds convert small birds to their color

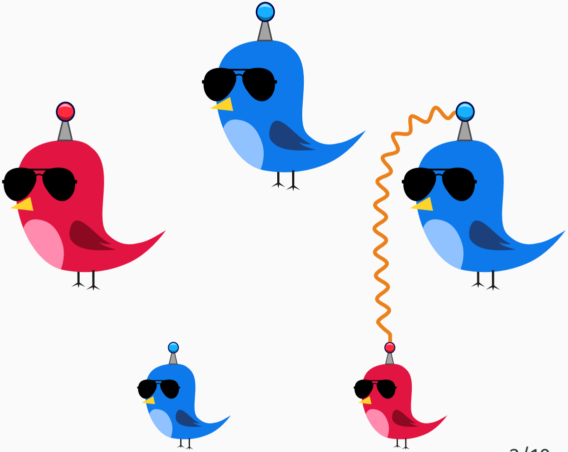


Example: majority protocol

At least as many blue birds than red birds?

Protocol:

- Two large birds of different colors become small and blue
- Large birds convert small birds to their color

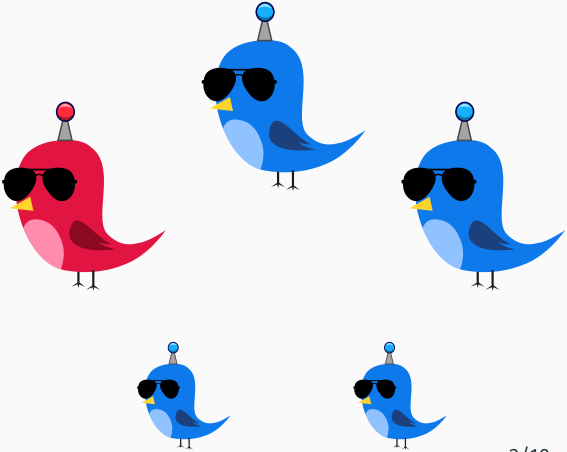


Example: majority protocol

At least as many blue birds than red birds?

Protocol:

- Two large birds of different colors become small and blue
- Large birds convert small birds to their color

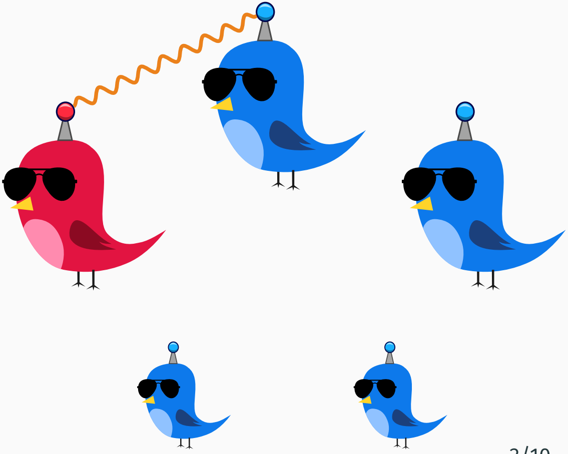


Example: majority protocol

At least as many blue birds than red birds?

Protocol:

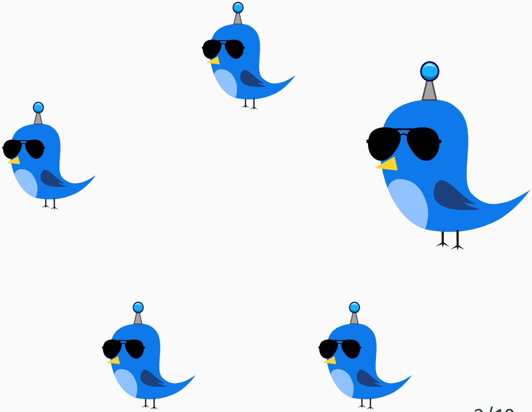
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color



At least as many blue birds than red birds?

Protocol:

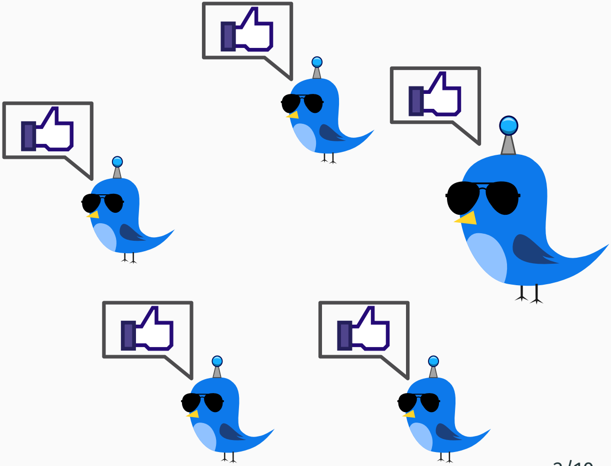
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color



At least as many blue birds than red birds?

Protocol:

- Two large birds of different colors become small and blue
- Large birds convert small birds to their color

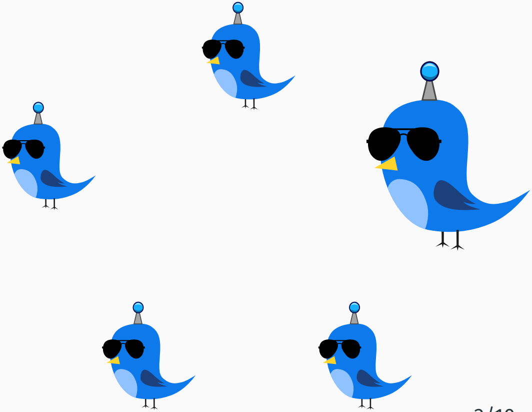


Example: majority protocol

At least as many blue birds than red birds?

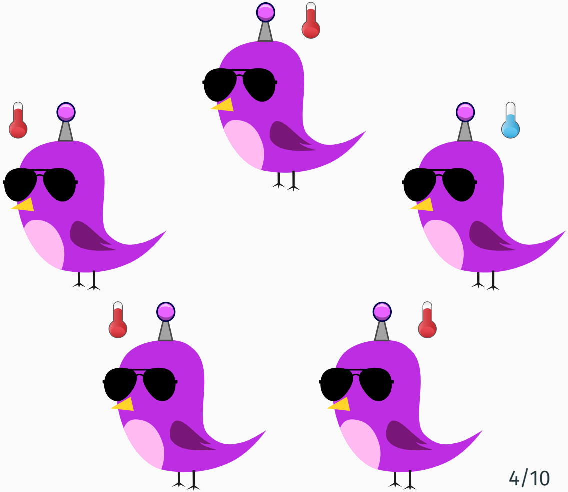
Protocol:

- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- **To break ties:** small blue birds convert small red birds



Example: threshold protocol

Are there at least 4 sick birds?

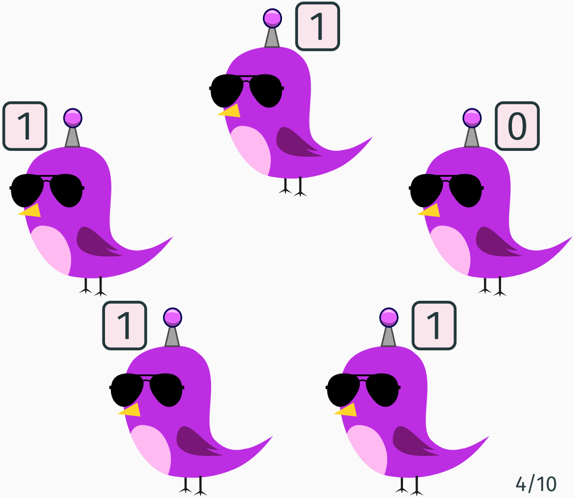


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$

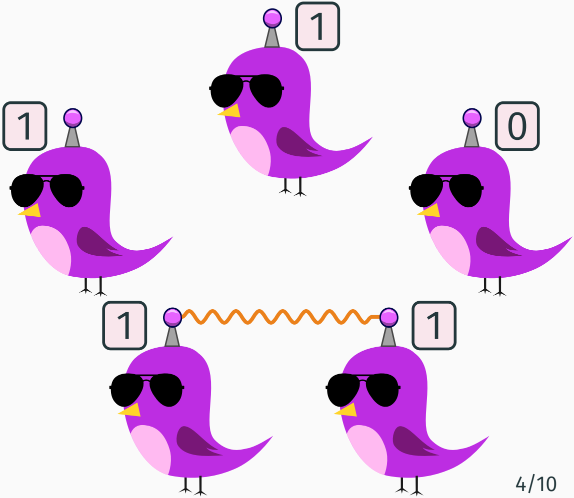


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

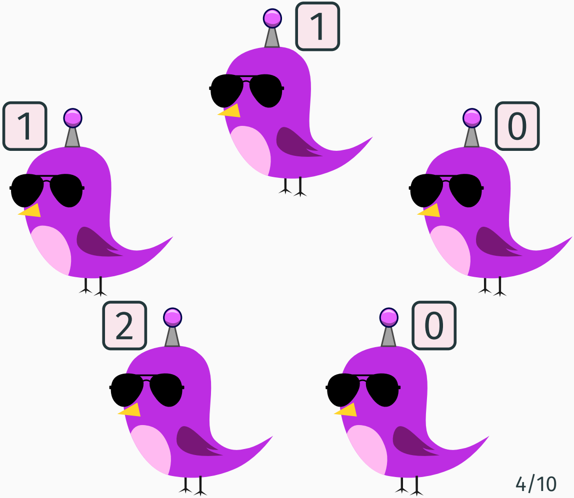
- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$



Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$

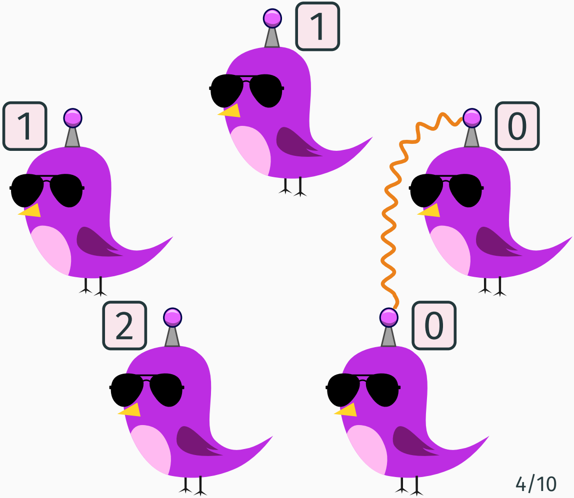


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m+n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$

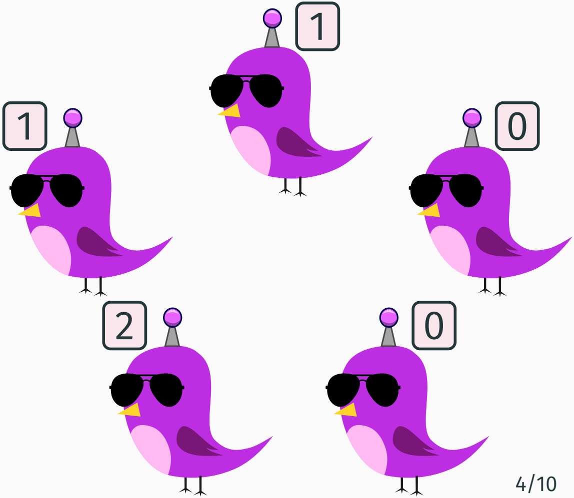


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$

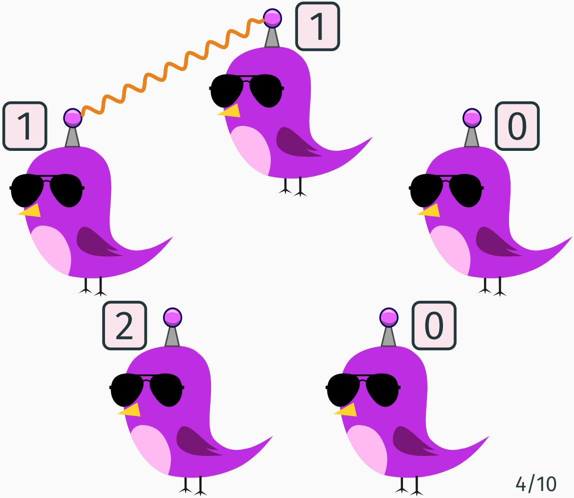


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$

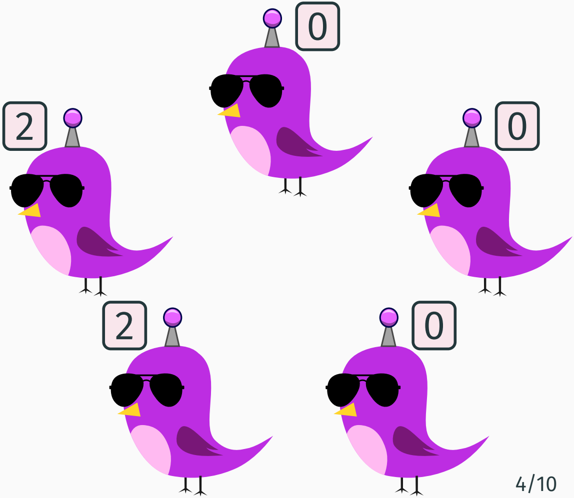


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$

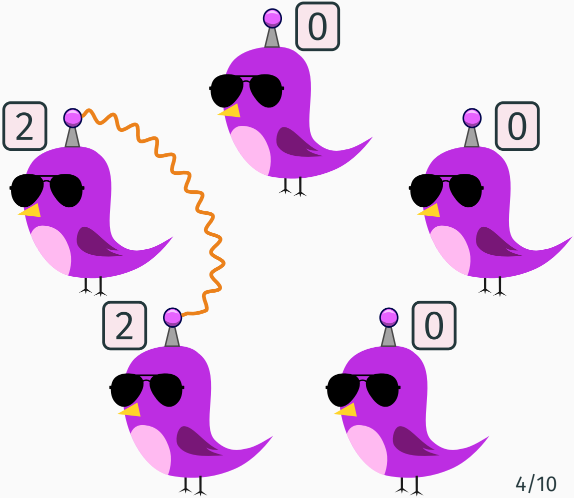


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

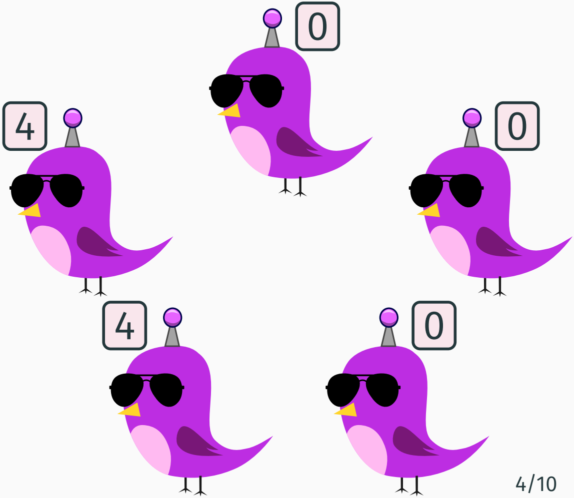
- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$



Are there at least 4 sick birds?

Protocol:

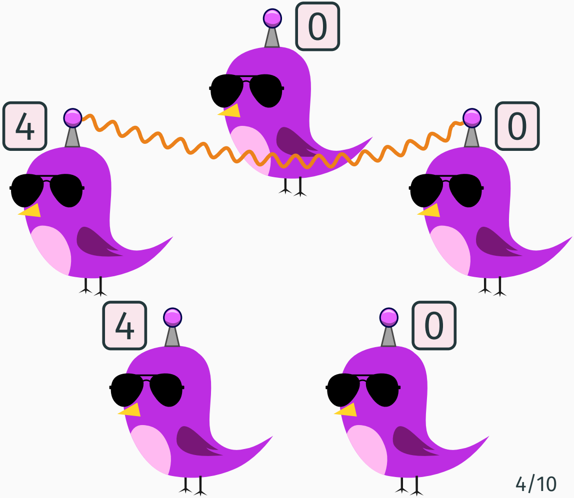
- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$



Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m+n, 0)$ if $m + n < 4$
- $(m, n) \mapsto (4, 4)$ if $m + n \geq 4$

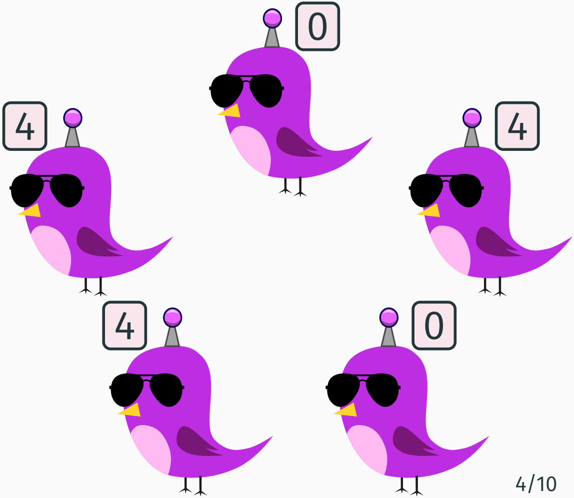


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m+n, 0)$ if $m+n < 4$
- $(m, n) \mapsto (4, 4)$ if $m+n \geq 4$

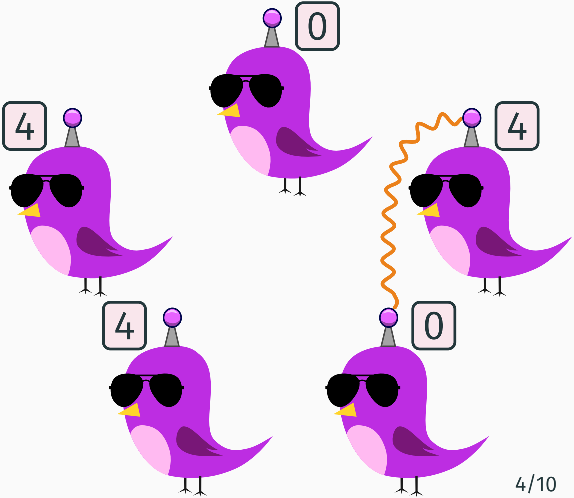


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

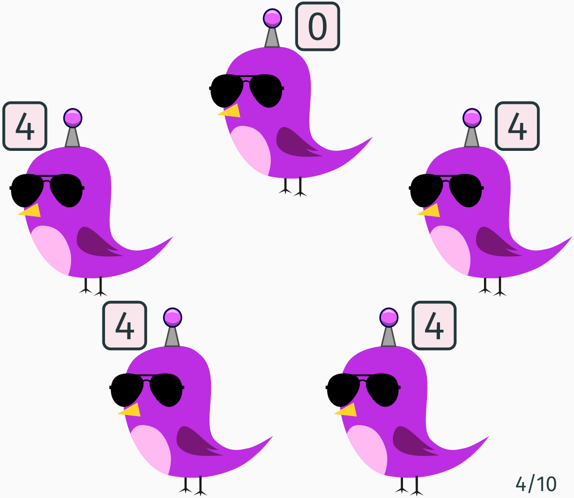
- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$



Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$ if $m + n < 4$
- $(m, n) \mapsto (4, 4)$ if $m + n \geq 4$

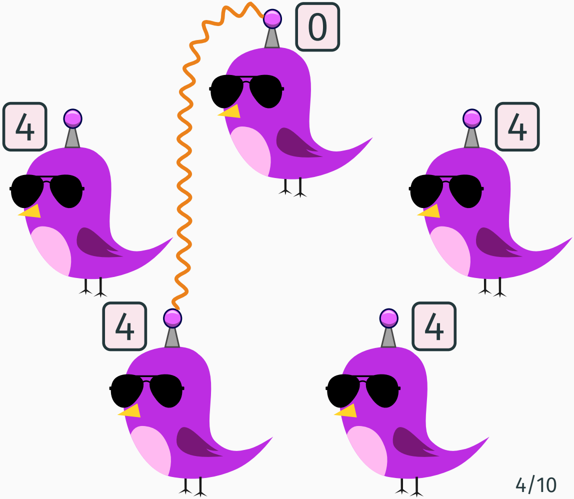


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 4$
- $(m, n) \mapsto (4, 4)$
  if $m + n \geq 4$

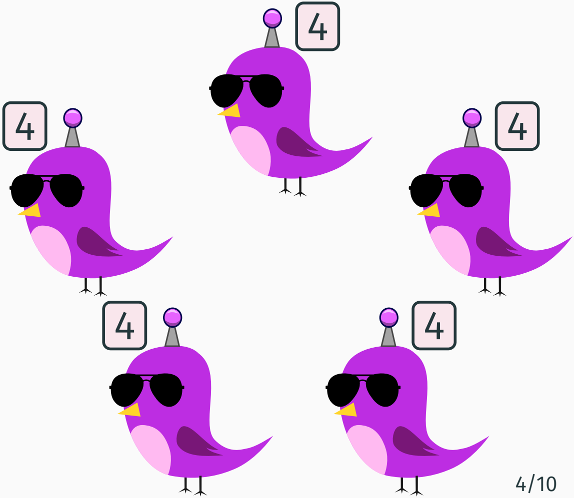


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$ if $m + n < 4$
- $(m, n) \mapsto (4, 4)$ if $m + n \geq 4$

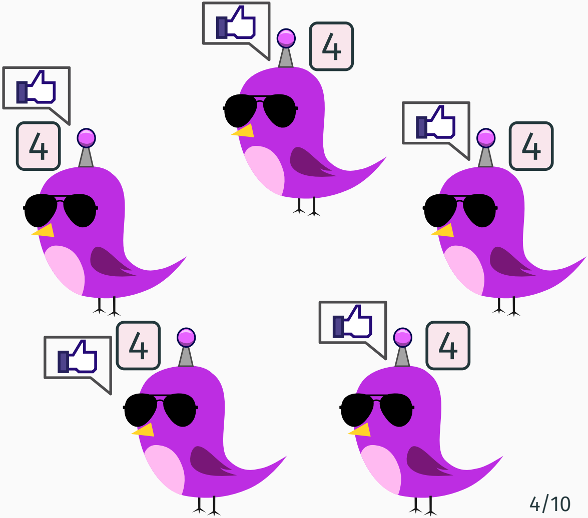


Example: threshold protocol

Are there at least 4 sick birds?

Protocol:

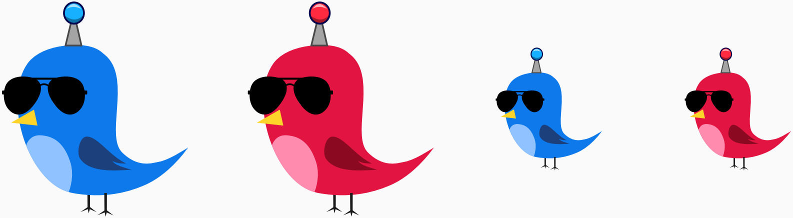
- Each bird is in a state of $\{0, 1, 2, 3, 4\}$
- Sick birds initially in state 1 and healthy birds in state 0
- $(m, n) \mapsto (m + n, 0)$ if $m + n < 4$
- $(m, n) \mapsto (4, 4)$ if $m + n \geq 4$



Demonstration

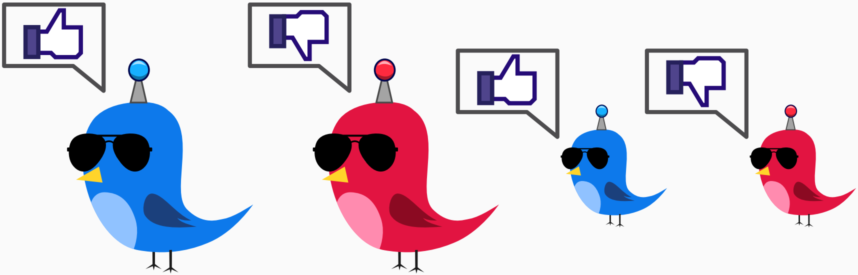
Population protocols: formal model

- *States*: finite set Q
- *Opinions*: $O : Q \to \{0, 1\}$
- *Initial states*: $I \subseteq Q$
- *Transitions*: $T \subseteq Q^2 \times Q^2$



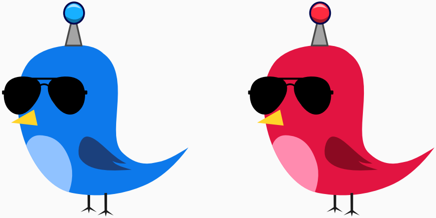
Population protocols: formal model

- *States*: finite set Q
- *Opinions*: $O : Q \to \{0, 1\}$
- *Initial states*: $I \subseteq Q$
- *Transitions*: $T \subseteq Q^2 \times Q^2$



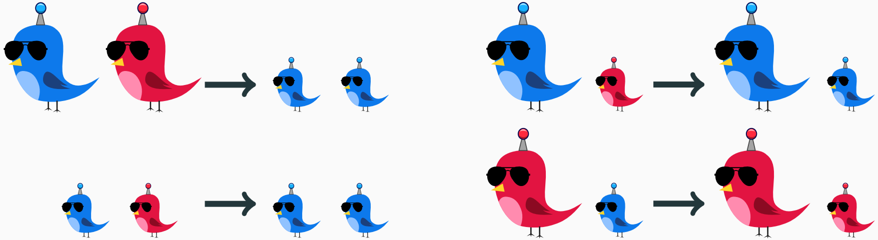
Population protocols: formal model

- *States*: finite set Q
- *Opinions*: $O : Q \to \{0, 1\}$
- *Initial states*: $I \subseteq Q$
- *Transitions*: $T \subseteq Q^2 \times Q^2$

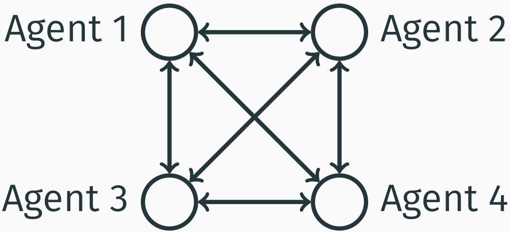


Population protocols: formal model

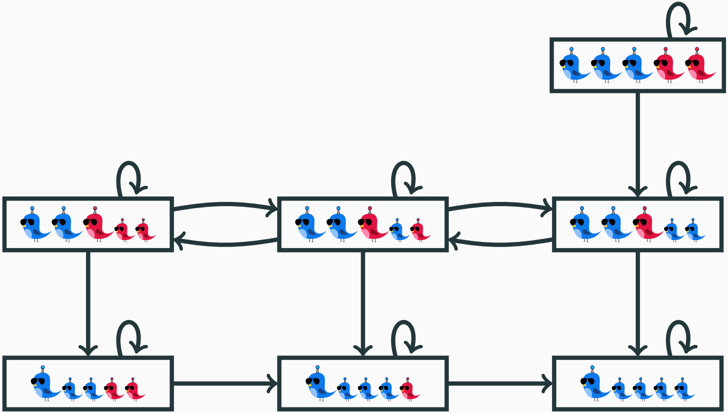
- *States*: finite set Q
- *Opinions*: $O : Q \to \{0, 1\}$
- *Initial states*: $I \subseteq Q$
- *Transitions*: $T \subseteq Q^2 \times Q^2$



Interaction graph:

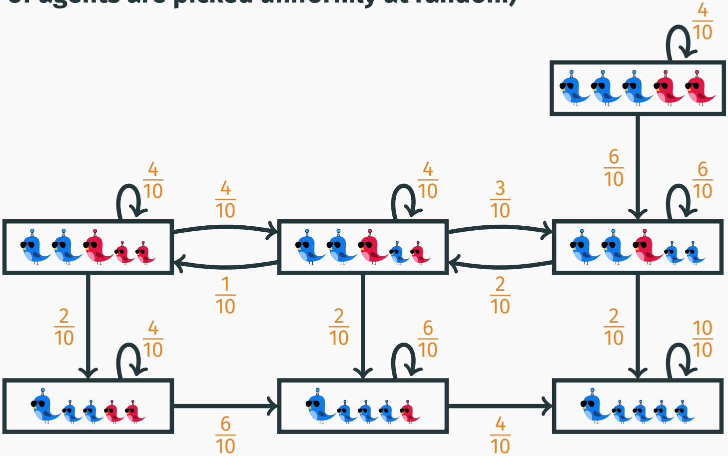


Reachability graph:

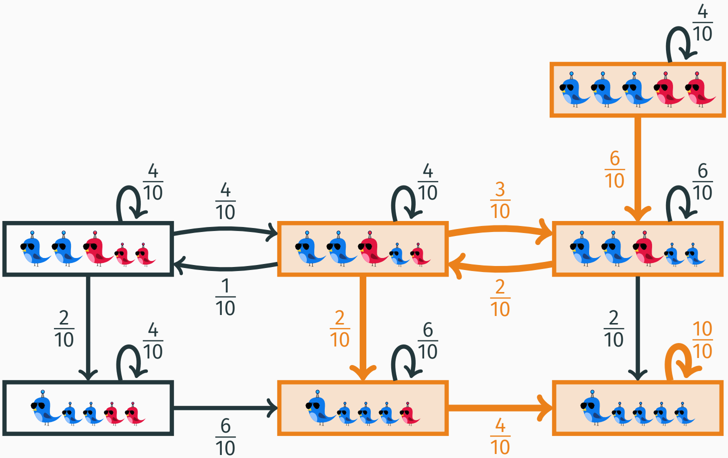


Underlying Markov chain:

(pairs of agents are picked uniformly at random)



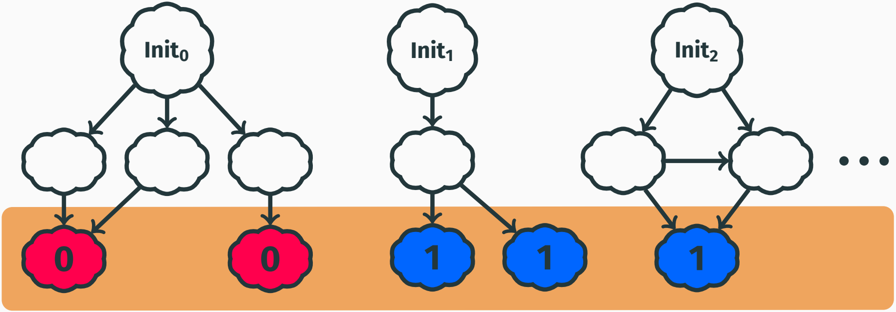
A *run* is an infinite path:



A protocol computes a predicate $\varphi\colon \mathbb{N}^I \to \{0,1\}$

if runs reach **common stable consensus**

with probability 1



A protocol computes a predicate $\varphi\colon \mathbb{N}^I \to \{0,1\}$

if runs reach **common stable consensus**

with probability 1

Expressive power **Angluin, Aspnes, Eisenstat PODC'06**

Population protocols compute precisely predicates definable in Presburger arithmetic, *i.e.* FO($\mathbb{N}$, $+$, $<$)

Other variants considered:

- Approximate protocols — *e.g.* Angluin, Aspnes, Eisenstat DISC'07
- Protocols with leaders — Angluin, Aspnes, Eisenstat Dist. Comput.'08
- Protocols with failures — Delporte-Gallet *et al.* DCOSS'06
- Trustful protocols — Bournez, Lefevre, Rabie DISC'13
- Mediated protocols, etc. — Michail, Chatzigiannakis, Spirakis TCS'11

Expressive power **Angluin, Aspnes, Eisenstat PODC'06**

Population protocols compute precisely predicates definable in Presburger arithmetic, *i.e.* $\mathrm{FO}(\mathbb{N}, +, <)$

Formal analysis of protocols

Protocols can become complex, even for B ≥ R:

Fast and Exact Majority in Population Protocols

Dan Alistarh
Microsoft Research

Rati Gelashvili*
MIT

Milan Vojnović
Microsoft Research

$$weight(x) = \begin{cases} |x| & \text{if } x \in StrongStates \text{ or } x \in WeakStates; \\ 1 & \text{if } x \in IntermediateStates. \end{cases}$$

$$sgn(x) = \begin{cases} 1 & \text{if } x \in \{+0, 1_d, \ldots, 1_1, 3, 5, \ldots, m\}; \\ -1 & \text{otherwise.} \end{cases}$$

$value(x) = sgn(x) \cdot weight(x)$

`/* Functions for rounding state interactions */`

$\phi(x) = -1_1$ if $x = -1$; 1_1 if $x = 1$; x, otherwise

$R_\downarrow(k) = \phi(k$ if k odd integer, $k - 1$ if k even$)$

$R_\uparrow(k) = \phi(k$ if k odd integer, $k + 1$ if k even$)$

$$Shift\text{-}to\text{-}Zero(x) = \begin{cases} -1_{j+1} & \text{if } x = -1_j \text{ for some index } j < d \\ 1_{j+1} & \text{if } x = 1_j \text{ for some index } j < d \\ x & \text{otherwise.} \end{cases}$$

$$Sign\text{-}to\text{-}Zero(x) = \begin{cases} +0 & \text{if } sgn(x) > 0 \\ -0 & \text{oherwise.} \end{cases}$$

procedure update$\langle x, y \rangle$
    if $(weight(x) > 0$ *and* $weight(y) > 1)$ ***or*** $(weight(y) > 0$ *and* $weight(x) > 1)$ **then**
        $x' \leftarrow R_\downarrow\left(\frac{value(x)+value(y)}{2}\right)$ and $y' \leftarrow R_\uparrow\left(\frac{value(x)+value(y)}{2}\right)$
    else if $weight(x) \cdot weight(y) = 0$ *and* $value(x) + value(y) > 0$ **then**
        if $weight(x) \neq 0$ **then** $x' \leftarrow Shift\text{-}to\text{-}Zero(x)$ and $y' \leftarrow Sign\text{-}to\text{-}Zero(x)$
        else $y' \leftarrow Shift\text{-}to\text{-}Zero(y)$ and $x' \leftarrow Sign\text{-}to\text{-}Zero(y)$
    else if $(x \in \{-1_d, +1_d\}$ *and* $weight(y) = 1$ *and* $sgn(x) \neq sgn(y))$ ***or***
            $(y \in \{-1_d, +1_d\}$ *and* $weight(x) = 1$ *and* $sgn(y) \neq sgn(x))$ **then**
        $x' \leftarrow -0$ and $y' \leftarrow +0$
    else
        $x' \leftarrow Shift\text{-}to\text{-}Zero(x)$ and $y' \leftarrow Shift\text{-}to\text{-}Zero(y)$

Protocols can become complex, even for B ≥ R:

Fast and Exact Majority in Population Protocols

Dan Alistarh
Microsoft Research

Rati Gelashvili*
MIT

Milan Vojnović
Microsoft Research

$$weight(x) = \begin{cases} |x| & \text{if } x \in StrongStates \text{ or } x \in WeakStates; \\ 1 & \text{if } x \in IntermediateStates. \end{cases}$$

$$sgn(x) = \begin{cases} 1 & \text{if } x \in \{+0, 1_d, \dots, 1_1, 3, 5, \dots, m\}; \\ -1 & \text{otherwise.} \end{cases}$$

$value(x) = sgn(x) \cdot weight(x)$

```
/* Functions for rounding state interactions */
```

$\phi(x) = -1_1$ if $x = -1$; 1_1 if $x = 1$; x, otherwise

$R_{\downarrow}(k) = \phi(k$ if k odd integer, $k-1$ if k even$)$

$R_{\uparrow}(k) = \phi(k$ if k odd integer, $k+1$ if k even$)$

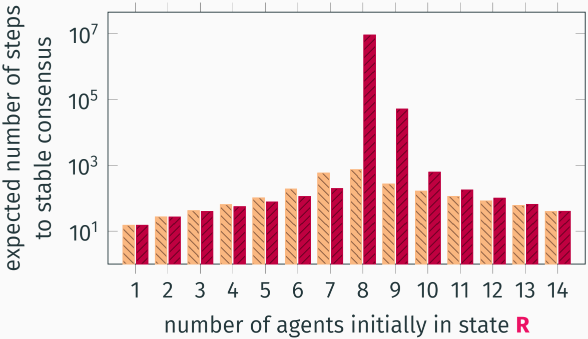
$$\textit{Shift-to-Zero}(x) = \begin{cases} -1_{j+1} & \text{if } x = -1_j \text{ for some index } j < d \\ 1_{j+1} & \text{if } x = 1_j \text{ for some index } j < d \\ x & \text{otherwise.} \end{cases}$$

$$\textit{Sign-to-Zero}(x) = \begin{cases} +0 & \text{if } sgn(x) > 0 \\ -0 & \text{oherwise.} \end{cases}$$

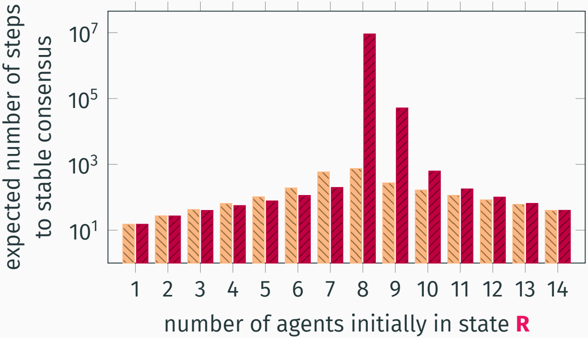
procedure update$\langle x, y \rangle$
  if $(weight(x) > 0$ *and* $weight(y) > 1)$ ***or*** $(weight(y) > 0$ *and* $weight(x) > 1)$ **then**
    $x' \leftarrow R_{\downarrow}\left(\frac{value(x)+value(y)}{2}\right)$ and $y' \leftarrow R_{\uparrow}\left(\frac{value(x)+value(y)}{2}\right)$
  else if $weight(x) \cdot weight(y) = 0$ *and* $value(x) + value(y) > 0$ **then**
    if $weight(x) \neq 0$ **then** $x' \leftarrow \textit{Shift-to-Zero}(x)$ and $y' \leftarrow \textit{Sign-to-Zero}(x)$
    else $y' \leftarrow \textit{Shift-to-Zero}(y)$ and $x' \leftarrow \textit{Sign-to-Zero}(y)$
  else if $(x \in \{-1_d, +1_d\}$ *and* $weight(y) = 1$ *and* $sgn(x) \neq sgn(y))$ ***or***
      $(y \in \{-1_d, +1_d\}$ *and* $weight(x) = 1$ *and* $sgn(y) \neq sgn(x))$ **then**
    $x' \leftarrow -0$ and $y' \leftarrow +0$
  else
    $x' \leftarrow \textit{Shift-to-Zero}(x)$ and $y' \leftarrow \textit{Shift-to-Zero}(y)$

How to verify correctness automatically?

Convergence speed may vary wildly, challenging to establish bounds



Convergence speed may vary wildly, challenging to establish bounds



How to derive asymptotic bounds automatically?

Formal analysis of protocols

Number of states corresponds to amount of memory, relevant to keep it minimal for embedded systems

- **B ≥ R** requires at least 4 states (Mertzios *et al.* ICALP'14)
- **X ≥ c** requires at most c + 1 states

Formal analysis of protocols

Number of states corresponds to amount of memory, relevant to keep it minimal for embedded systems

- **B ≥ R** requires at least 4 states (Mertzios *et al.* ICALP'14)
- **X ≥ c** requires at most c + 1 states

What is the state complexity of common predicates?

Formal analysis of protocols

1. **Automatic verification of correctness**
   - Decidability — Esparza, Ganty, Leroux, Majumdar CONCUR'15, FSTTCS'16
   - Towards efficient verification — B., Esparza, Jaax, Meyer PODC'17
   - Complete tool — B., Esparza, Jaax CAV'18
2. **Automatic analysis of convergence speed**
   - First procedure — B., Esparza, Kučera (submitted to CONCUR'18)
3. **State complexity of protocols w.r.t. predicates**
   - Study of linear inequalities — B., Esparza, Jaax STACS'18

Formal analysis of protocols

1. **Automatic verification of correctness**
   - Decidability — Esparza, Ganty, Leroux, Majumdar CONCUR'15, FSTTCS'16
   - Towards efficient verification — B., Esparza, Jaax, Meyer PODC'17
   - Complete tool — B., Esparza, Jaax CAV'18
2. **Automatic analysis of convergence speed**
   - First procedure — B., Esparza, Kučera (submitted to CONCUR'18)
3. **State complexity of protocols w.r.t. predicates**
   - Study of linear inequalities — B., Esparza, Jaax STACS'18

Existing verification tools:

- PAT: model checker with global fairness

  (Sun, Liu, Song Dong and Pang CAV'09)

- `bp-ver`: graph exploration

  (Chatzigiannakis, Michail and Spirakis SSS'10)

- Conversion to counter machines + `PRISM/Spin`

  (Clément, Delporte-Gallet, Fauconnier and Sighireanu ICDCS'11)

Verification: state of the art

Existing verification tools:

- PAT: model checker with global fairness

  (Sun, Liu, Song Dong and Pang CAV'09)

- `bp-ver`: graph exploration

  (Chatzigiannakis, Michail and Spirakis SSS'10)

- Conversion to counter machines + `PRISM`/`Spin`

  (Clément, Delporte-Gallet, Fauconnier and Sighireanu ICDCS'11)

Only for populations of fixed size!

Sometimes possible to verify all sizes:

- Verification with the interactive theorem prover Coq

(Deng and Monin TASE'09)

Sometimes possible to verify all sizes:

- Verification with the interactive theorem prover Coq

(Deng and Monin TASE'09)

Not automatic!

Sometimes possible to verify all sizes:

- Verification with the interactive theorem prover Coq

(Deng and Monin TASE'09)

Challenge: verifying automatically <u>all</u> sizes

Testing whether a protocol computes φ amounts to testing:

$$\neg\exists C, D: \quad \begin{aligned} & C \xrightarrow{*} D \wedge \\ & C \text{ is initial} \wedge \\ & D \text{ is bottom} \quad \wedge \\ & \text{opinion}(D) \neq \varphi(C) \end{aligned}$$

Testing whether a protocol computes φ amounts to testing:

$$\neg\exists C, D: \quad C \xrightarrow{*} D \land \\ C \text{ is initial} \land \\ D \text{ is bottom} \land \\ \text{opinion}(D) \neq \varphi(C)$$

As difficult as verification

Testing whether a protocol computes φ amounts to testing:

$$\neg\exists C, D: \quad C \overset{*}{\dashrightarrow} D \land C \text{ is initial} \land D \text{ is bottom} \land \text{opinion}(D) \neq \varphi(C)$$

Relaxed with Presburger-definable overapproximation

Testing whether a protocol computes φ amounts to testing:

$$\neg\exists C, D: \quad C \overset{*}{\dashrightarrow} D \wedge$$
$$C \text{ is initial } \wedge$$
$$D \text{ is bottom } \wedge$$
$$\text{opinion}(D) \neq \varphi(C)$$

Difficult to express

Testing whether a protocol computes φ amounts to testing:

$$\neg\exists C, D: \quad C \overset{*}{\dashrightarrow} D \land C \text{ is initial} \land D \text{ is terminal} \land \text{opinion}(D) \neq \varphi(C)$$

Most protocols are terminating!

Testing whether a protocol computes φ amounts to testing:

$$
\begin{aligned}
\neg\exists C, D: \quad & C \xrightarrow{*} D \;\land \\
& C \text{ is initial} \;\land \\
& D \text{ is terminal} \;\land \\
& \text{opinion}(D) \neq \varphi(C)
\end{aligned}
$$

Testable with an SMT solver

Testing whether a protocol computes φ amounts to testing:

$$\neg \exists C, D: \quad \begin{aligned} & C \overset{*}{\dashrightarrow} D \land \\ & C \text{ is initial} \land \\ & D \text{ is terminal} \land \\ & \text{opinion}(D) \neq \varphi(C) \end{aligned}$$

Protocol termination tested by structural analysis + SMT solving

Random variable *Steps*:

assigns to each run σ the smallest k s.t. σ_k in stable consensus

Maximal expected termination time

We are interested in $\mathit{time}\colon \mathbb{N} \to \mathbb{N}$ where

$$\mathit{time}(n) = \max\{\mathbb{E}_C[\mathit{Steps}] : C \text{ is initial and } |C| = n\}$$

Our approach:

- Most protocols are naturally designed in stages
- Construct these stages automatically
- Derive upper bounds on $time(n)$
  from stages structure

$\mathbf{B}, \mathbf{R} \mapsto \mathbf{b}, \mathbf{b}$

$\mathbf{B}, \mathbf{r} \mapsto \mathbf{B}, \mathbf{b}$

$\mathbf{R}, \mathbf{b} \mapsto \mathbf{R}, \mathbf{r}$

$\mathbf{b}, \mathbf{r} \mapsto \mathbf{b}, \mathbf{b}$

$$(\mathbf{B} \vee \mathbf{R}) \wedge \bigwedge_{q \notin \{\mathbf{B},\mathbf{R}\}} \neg q \xrightarrow{\mathcal{O}(1)} \Box\left(\mathbf{B} \wedge \bigwedge_{q \neq \mathbf{B}} \neg q\right)$$

$$(\mathbf{B} \vee \mathbf{R}) \wedge \bigwedge_{q \notin \{\mathbf{B},\mathbf{R}\}} \neg q \xrightarrow{\mathcal{O}(1)} \Box\left(\mathbf{R} \wedge \bigwedge_{q \neq \mathbf{R}} \neg q\right)$$

$$(\mathbf{B} \vee \mathbf{R}) \wedge \bigwedge_{q \notin \{\mathbf{B},\mathbf{R}\}} \neg q \xrightarrow{\mathcal{O}(n^2 \log n)} \Box(\neg\mathbf{B} \vee \neg\mathbf{R}) \wedge \mathbf{b} \wedge \neg\mathbf{b}!$$

$$\Box(\neg\mathbf{B} \vee \neg\mathbf{R}) \wedge \mathbf{b} \wedge \neg\mathbf{b}! \xrightarrow{\mathcal{O}(n^2 \log n)} \Box(\neg\mathbf{B} \wedge \neg\mathbf{R} \wedge \mathbf{b} \wedge \neg\mathbf{r})$$

$$\Box(\neg\mathbf{B} \vee \neg\mathbf{R}) \wedge \mathbf{b} \wedge \neg\mathbf{b}! \xrightarrow{\mathcal{O}(n^2 \log n)} \Box(\mathbf{B} \wedge \neg\mathbf{R} \wedge \mathbf{b} \wedge \neg\mathbf{r})$$

$$\Box(\neg\mathbf{B} \vee \neg\mathbf{R}) \wedge \mathbf{b} \wedge \neg\mathbf{b}! \xrightarrow{\mathcal{O}(\exp(n))} \Box(\neg\mathbf{B} \wedge \mathbf{R} \wedge \neg\mathbf{b} \wedge \mathbf{r})$$

- Prototype implemented in python™ + Microsoft Z3
- Can report: $\mathcal{O}(1), \mathcal{O}(n^2), \mathcal{O}(n^2 \log n), \mathcal{O}(n^3), \mathcal{O}(\text{poly}(n))$ or $\mathcal{O}(\exp(n))$
- Tested on various protocols from the literature

Peregrine: Haskell + Microsoft Z3 + JavaScript

`peregrine.model.in.tum.de`

- Design of protocols
- Manual and automatic simulation
- Statistics of properties such as termination time
- Automatic verification of correctness
- More to come!

Demonstration

Population protocols can be formally analyzed automatically:

- Verification of correctness
- Analysis of expected termination time
- Tool support!

Ongoing investigation of state complexity

Conclusion: future work (seeking for PhD students/Postdocs)

ERC Advanced Grant —

PaVeS: Parameterized Verification and Synthesis

- Goal: Develop proof and synthesis techniques for distributed algorithms working correctly for an arbitrary number of processes
- PI: Javier Esparza (esparza@in.tum.de), TU Munich
- Start of the project: Sept. 1, 2018
- Start of the PhDs/Postdocs: flexible, from Sept. 1, 2018 to about Sept. 1, 2019

Thank you!

Vielen Dank!