Peregrine 2.0: Explaining Correctness of Population Protocols through Stage Graphs

Martin Helfrich

Javier Esparza, Stefan Jaax and Philipp J Meyer

Technical University of Munich

European Research Council
Established by the European Commission

Peregrine

Peregrine: [Blodin et al., CAV'2018]
tool for analysis and parameterized verification of population protocols

Formal model of distributed computation by collections of

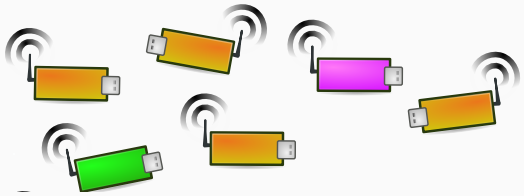
identical, finite-state, and mobile agents

like

Formal model of distributed computation by collections of

identical, finite-state, and mobile agents

like

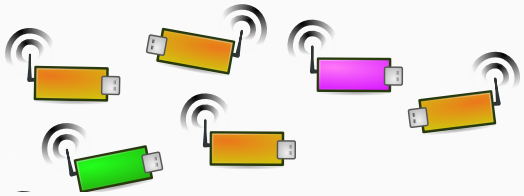


ad-hoc networks of mobile sensors

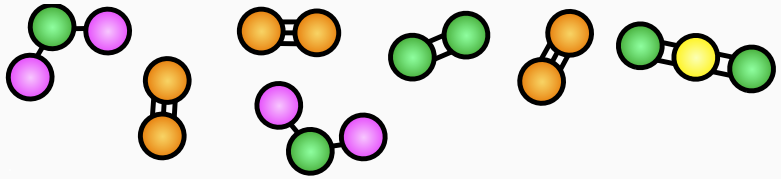
Formal model of distributed computation by collections of

identical, finite-state, and mobile agents

like



ad-hoc networks of mobile sensors

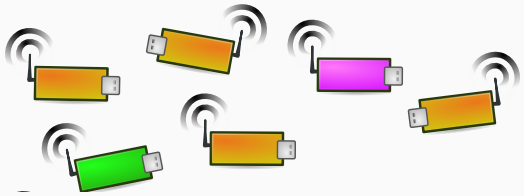


"soups" of molecules (Chemical Reaction Networks)

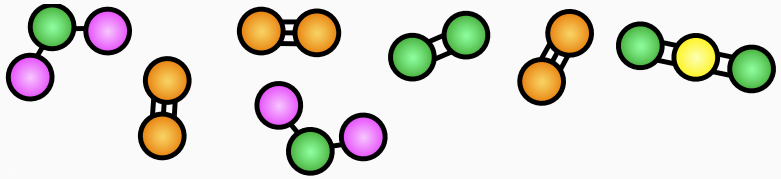
Formal model of distributed computation by collections of

identical, finite-state, and mobile agents

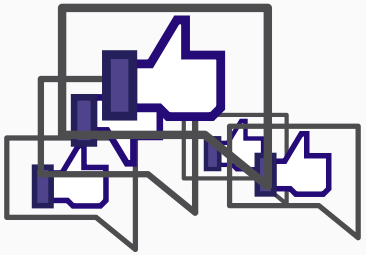
like



ad-hoc networks of mobile sensors



"soups" of molecules (Chemical Reaction Networks)

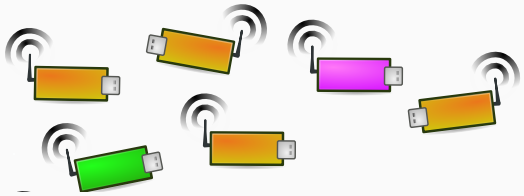


people in social networks

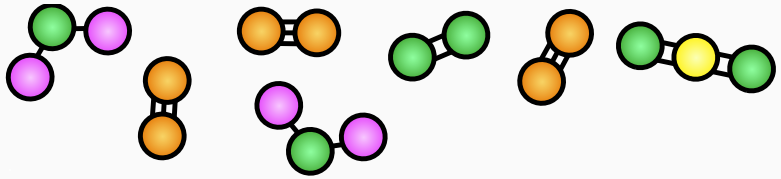
Formal model of distributed computation by collections of

identical, finite-state, and mobile agents

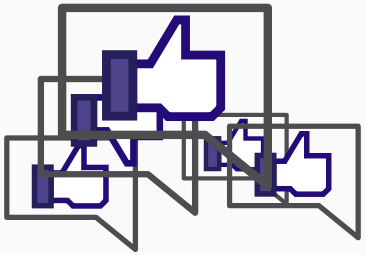
like



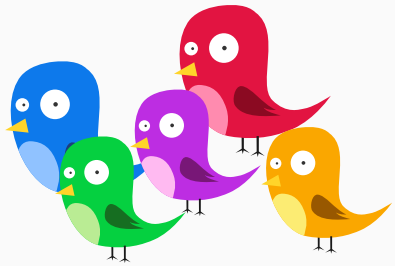
ad-hoc networks of mobile sensors



"soups" of molecules (Chemical Reaction Networks)

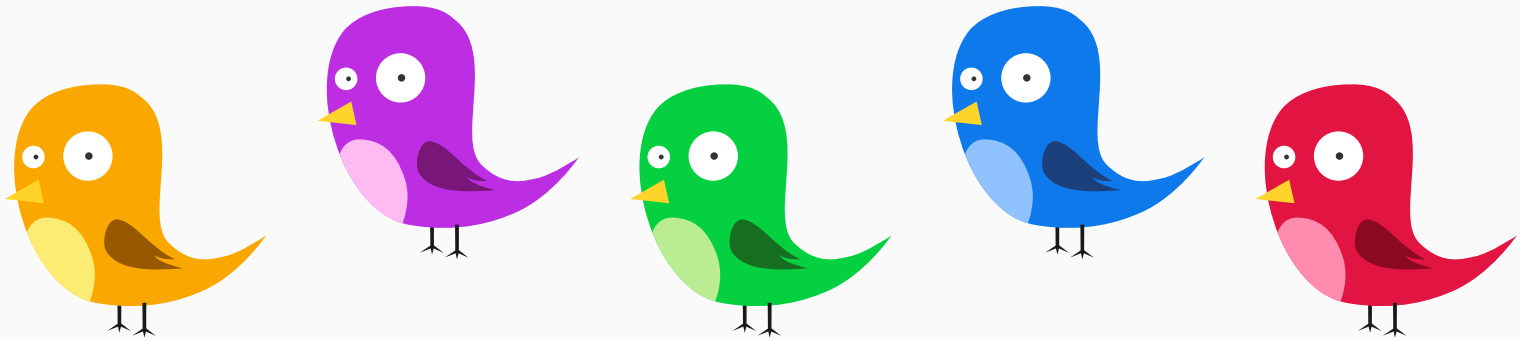


people in social networks



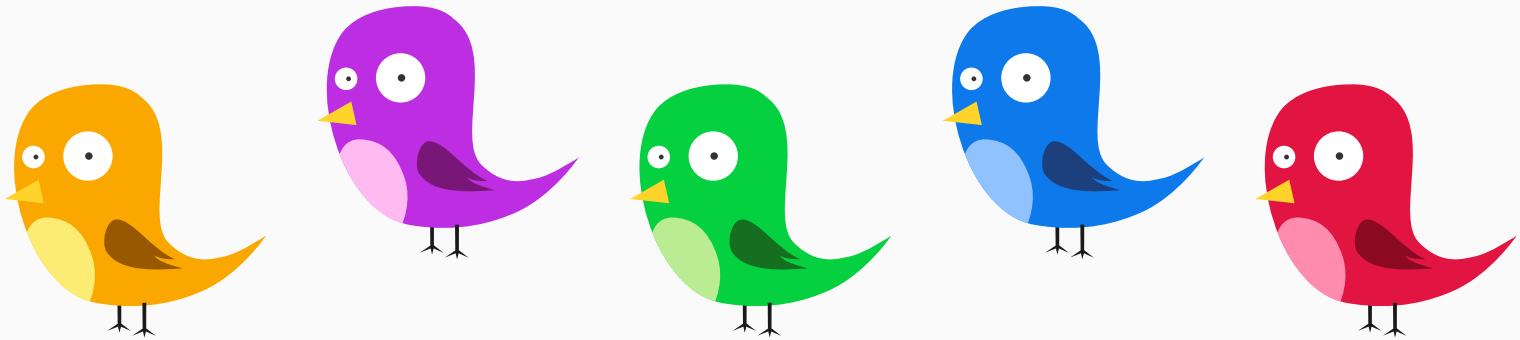
...and Birds!

Population Protocols: An Example



Population Protocols: An Example

- anonymous mobile agents with very few resources



Population Protocols: An Example

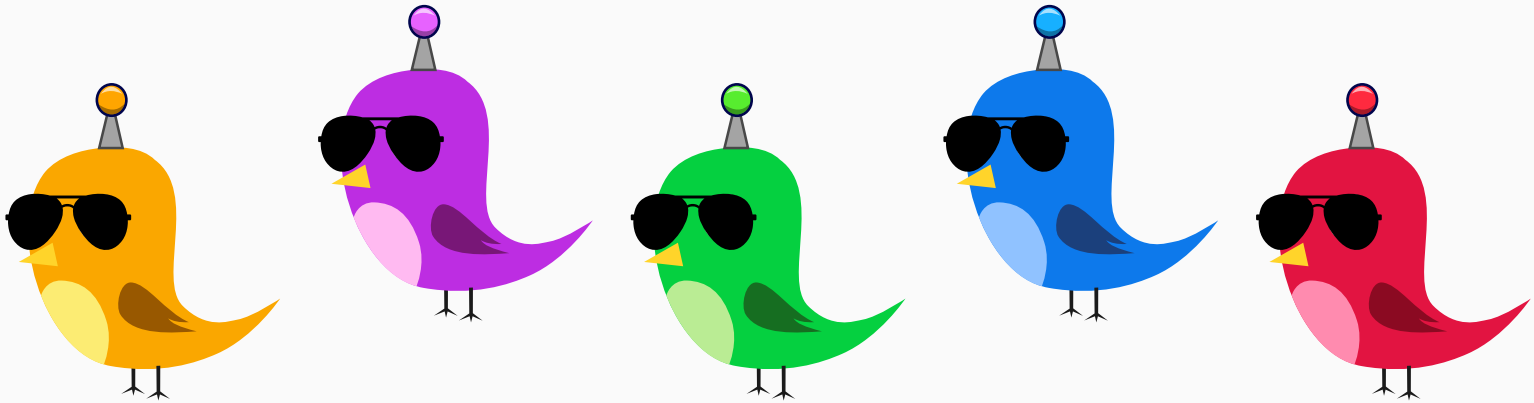
- anonymous mobile agents with very few resources

Population Protocols: An Example

- anonymous mobile agents with very few resources

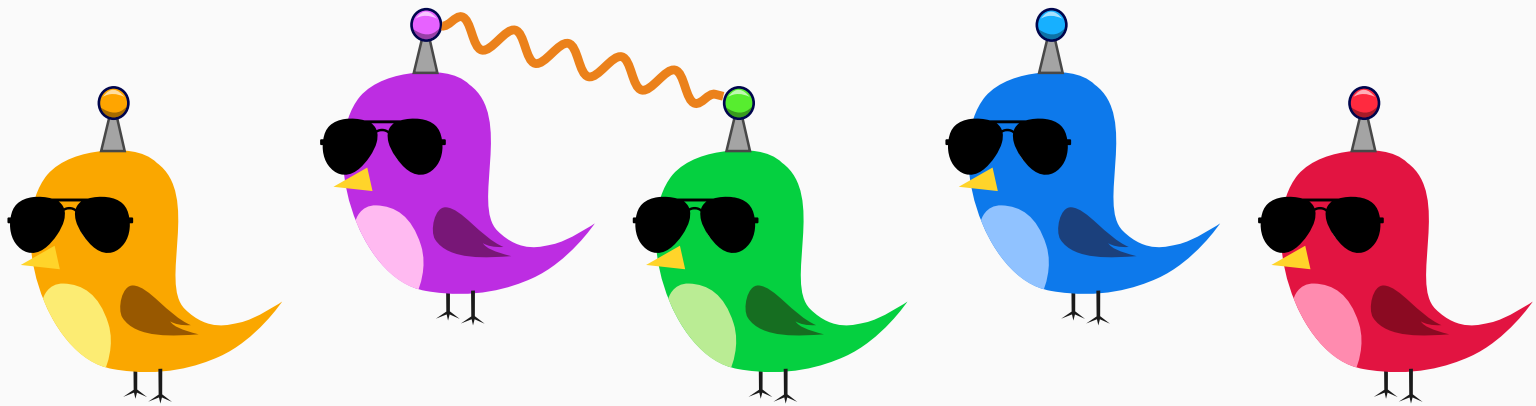
Population Protocols: An Example

- anonymous mobile agents with very few resources



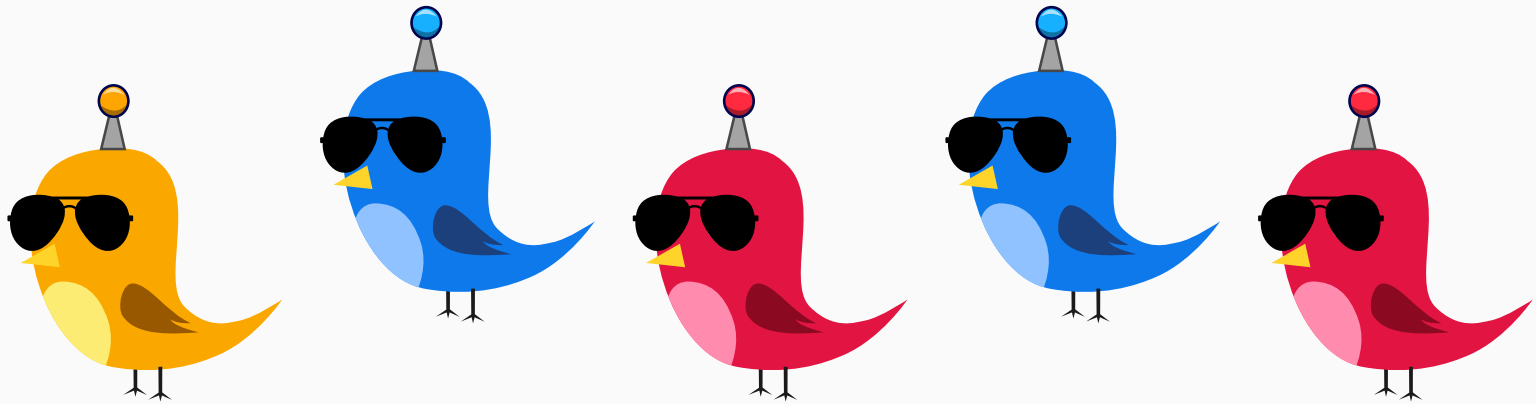
Population Protocols: An Example

- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions



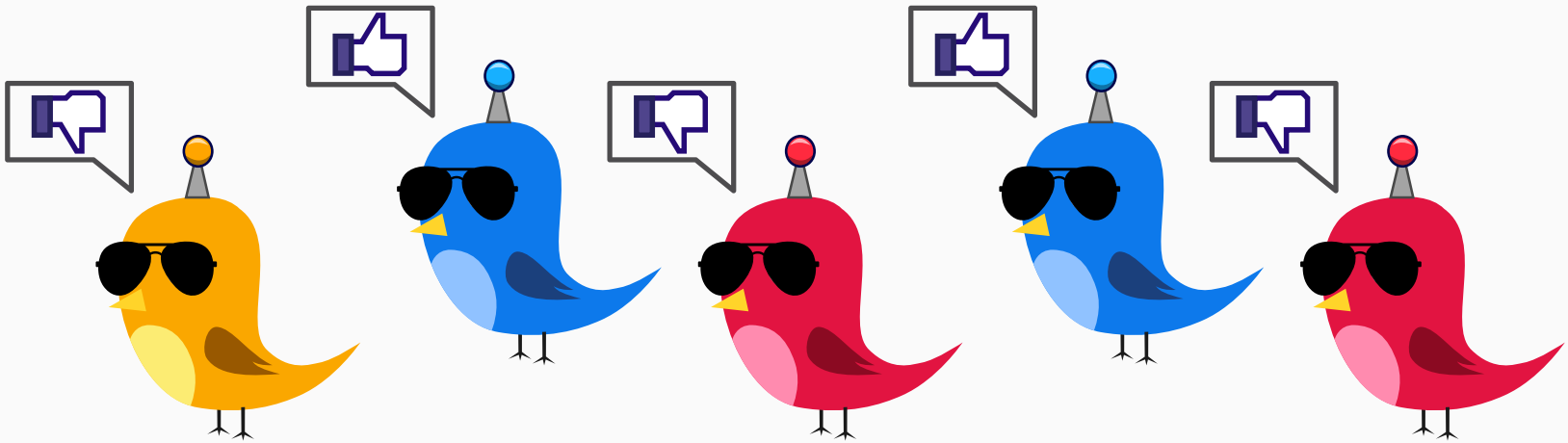
Population Protocols: An Example

- anonymous mobile agents with very few resources
- agents change states via random pairwise interactions



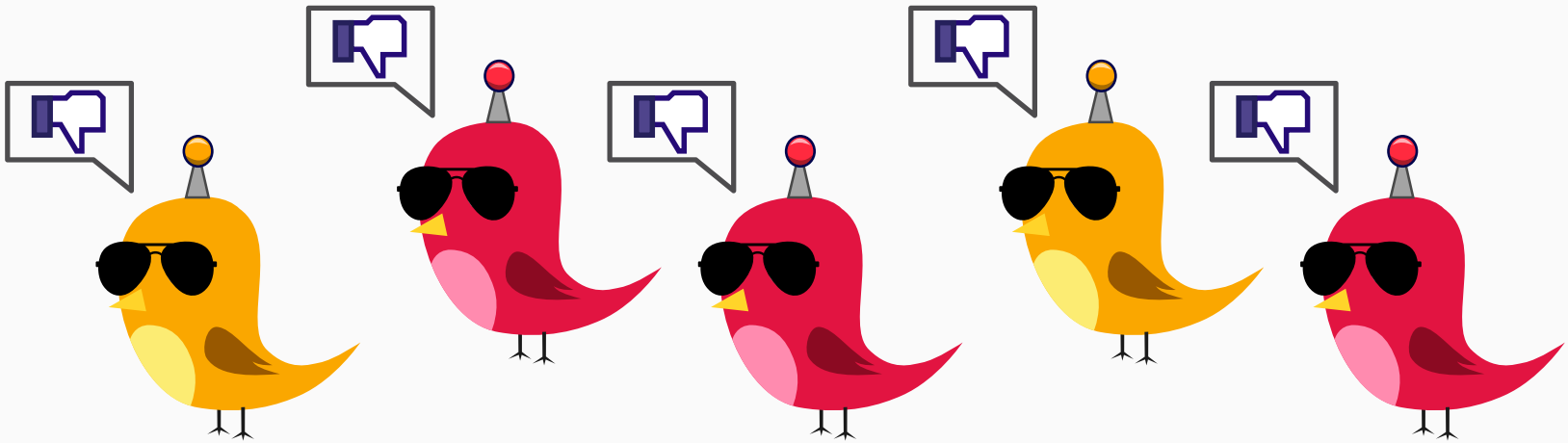
Population Protocols: An Example

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



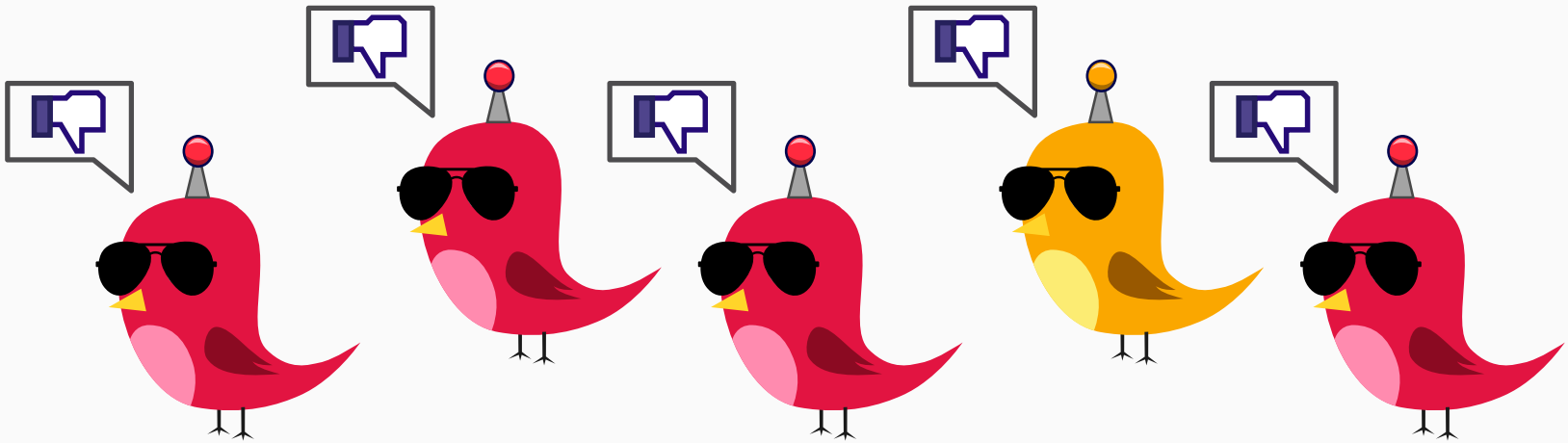
Population Protocols: An Example

- 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



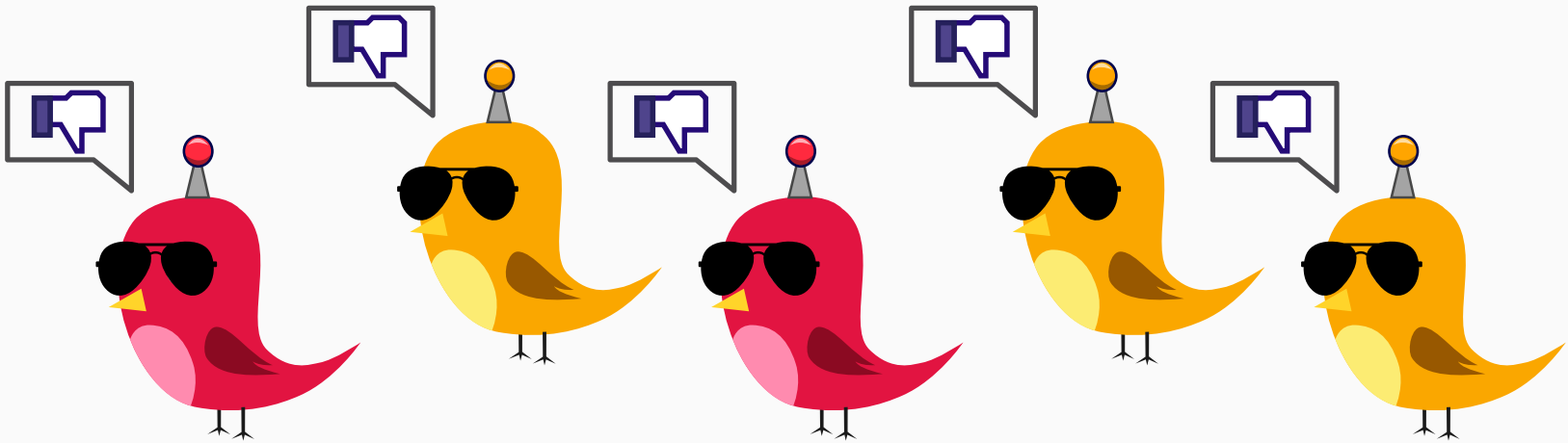
Population Protocols: An Example

- 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



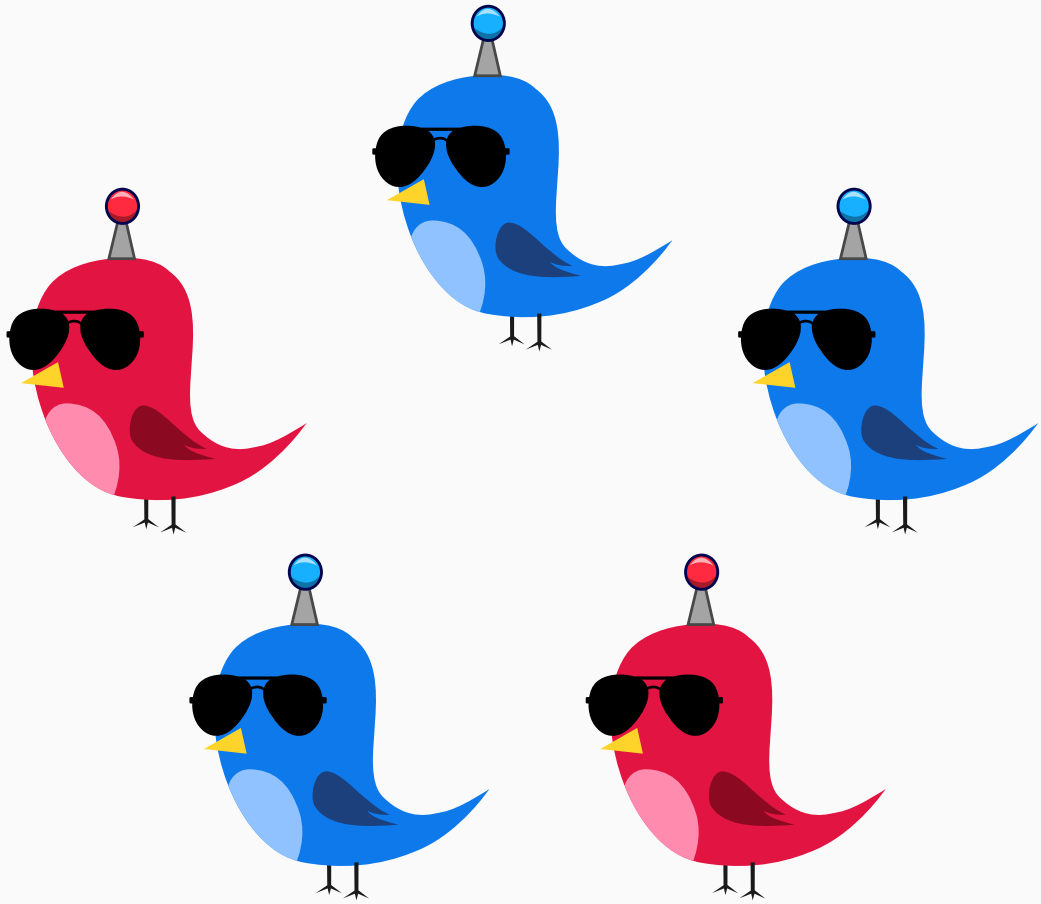
Population Protocols: An Example

- 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 voting protocol

At least as many blue birds as red birds?

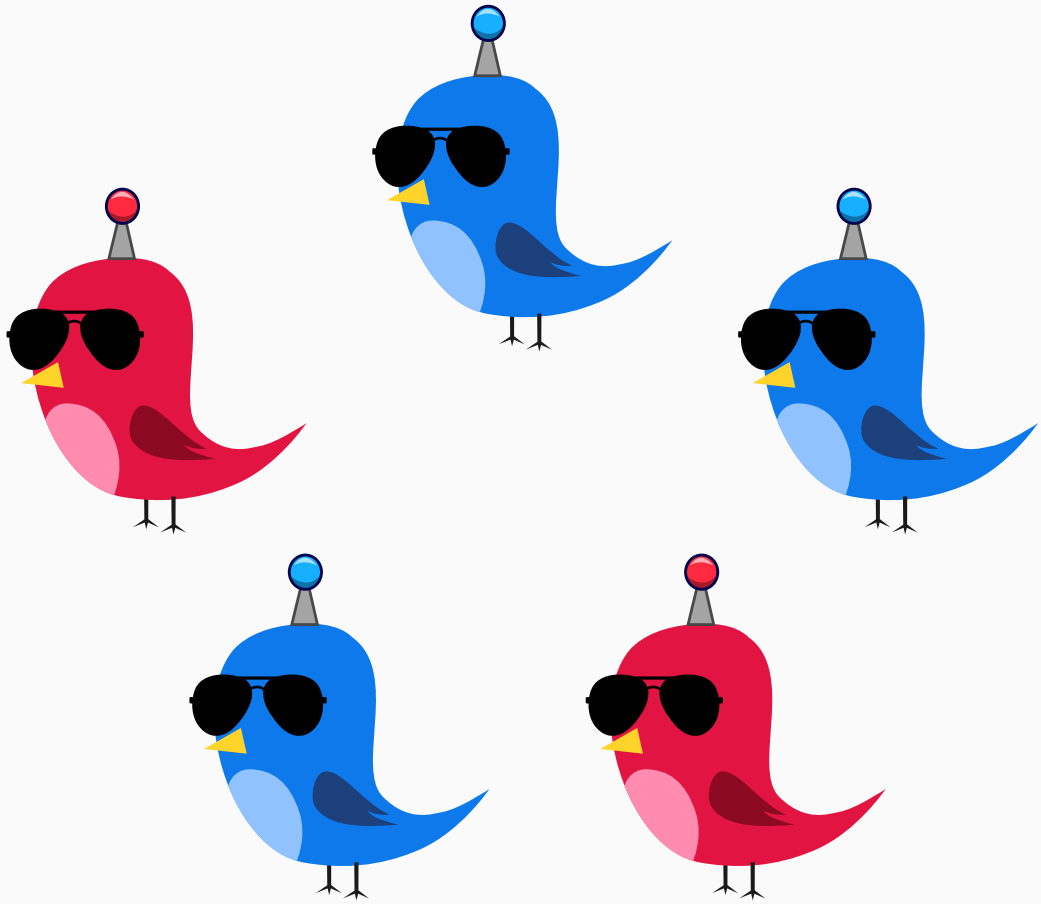


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

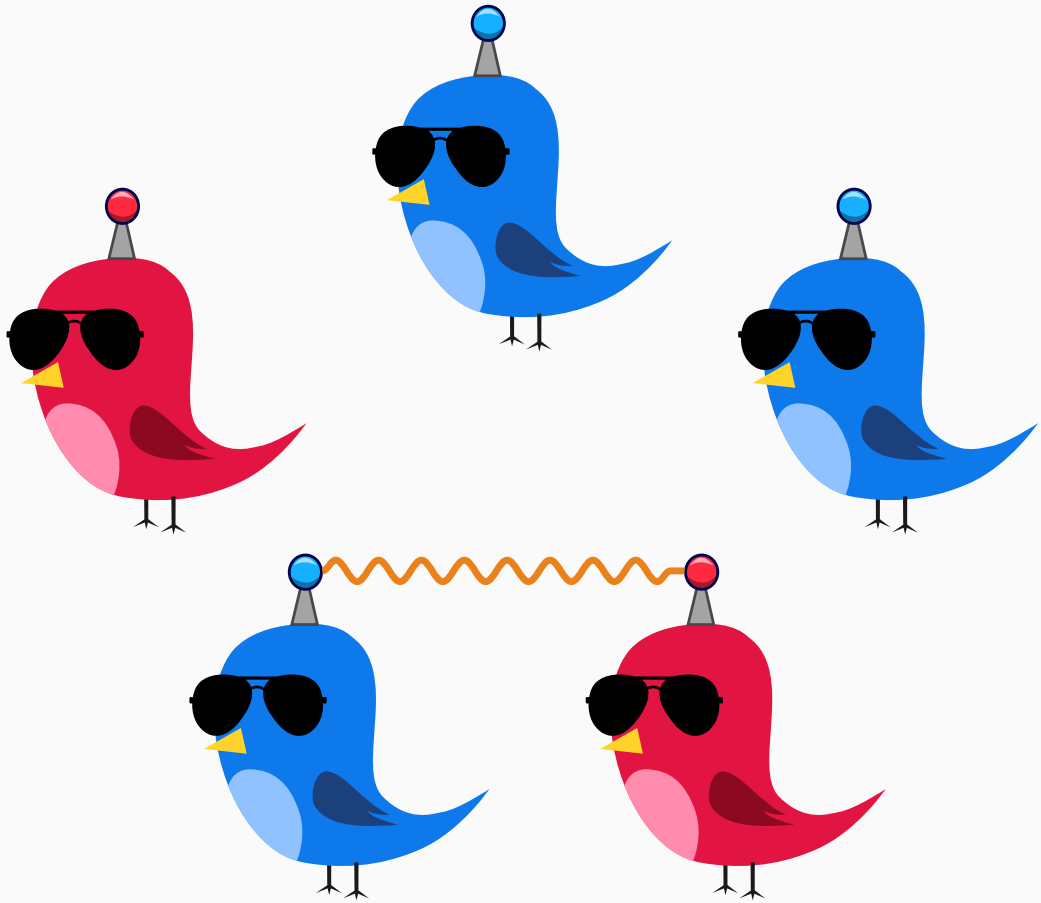


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

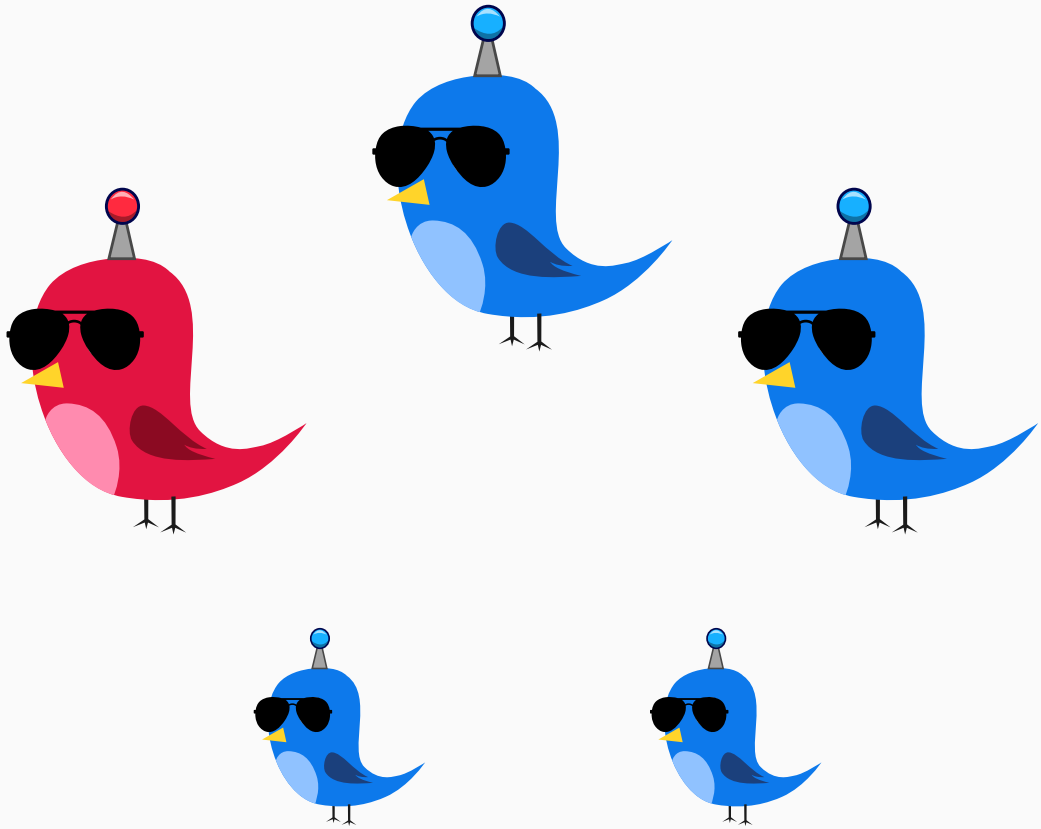


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

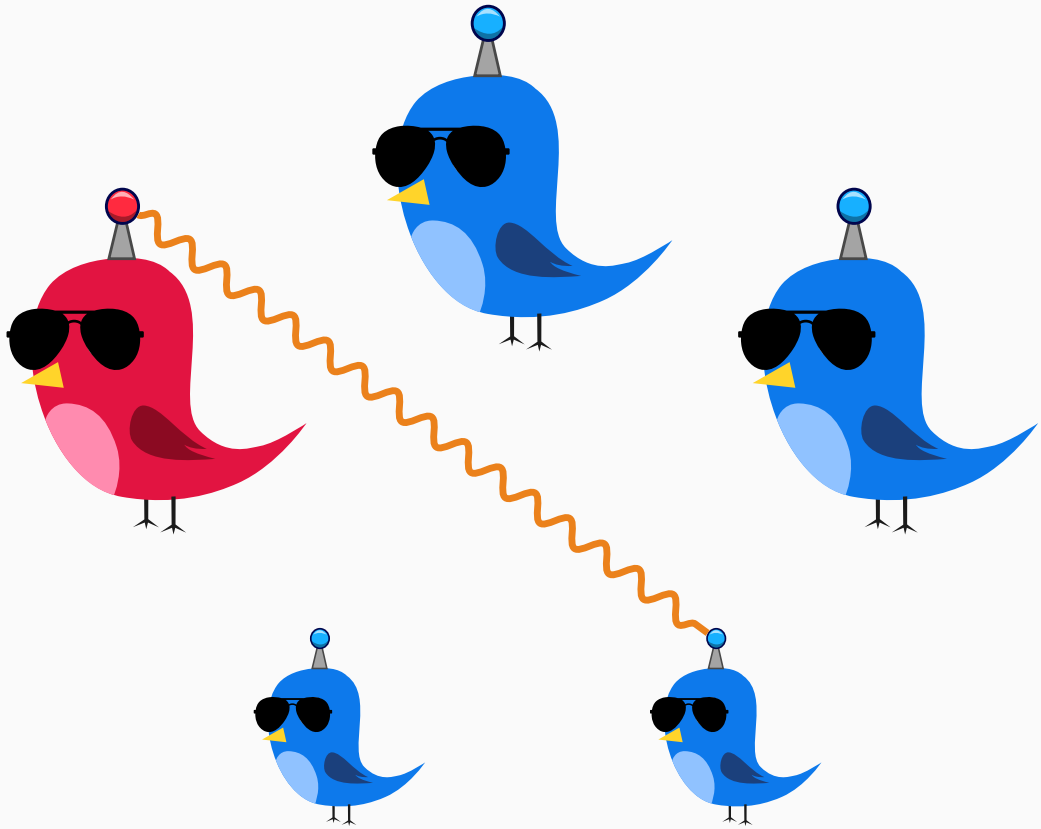


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

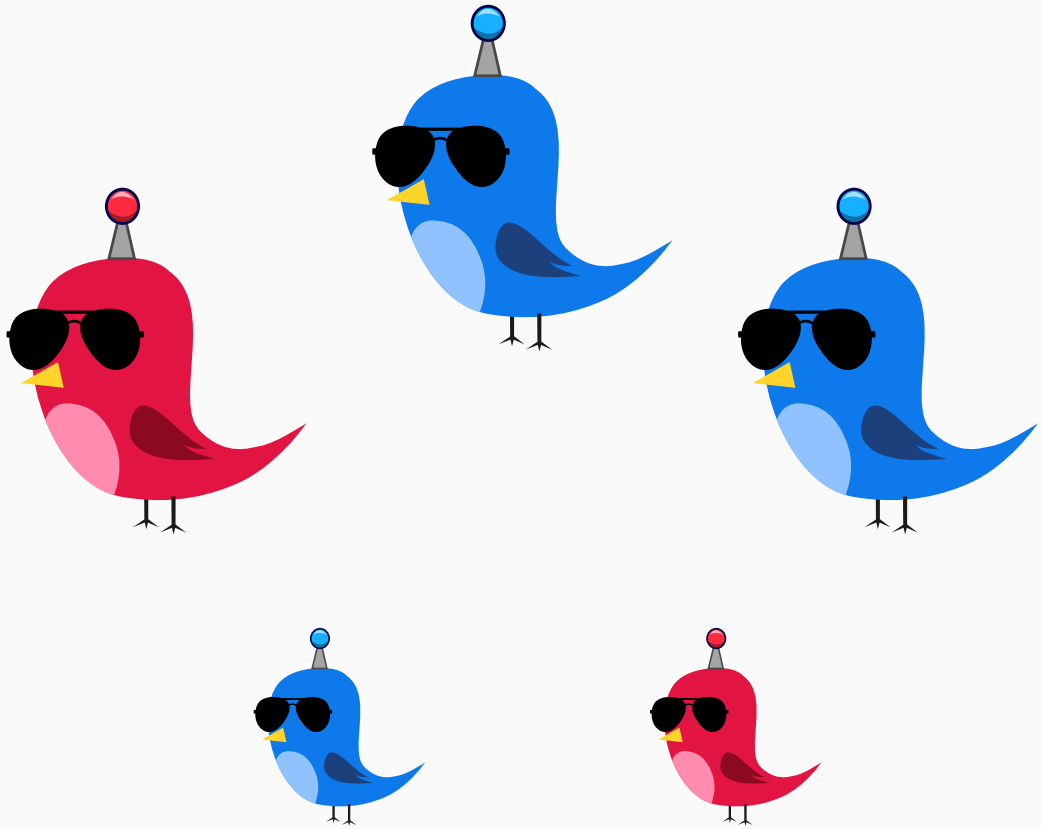


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

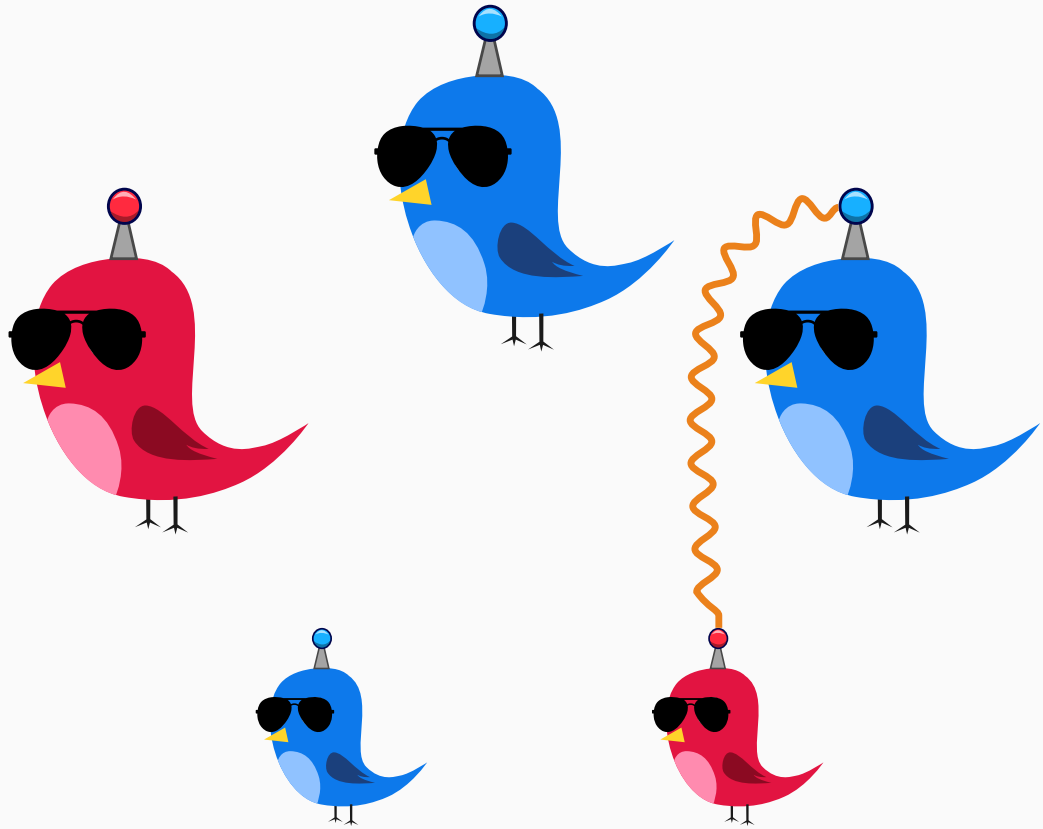


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

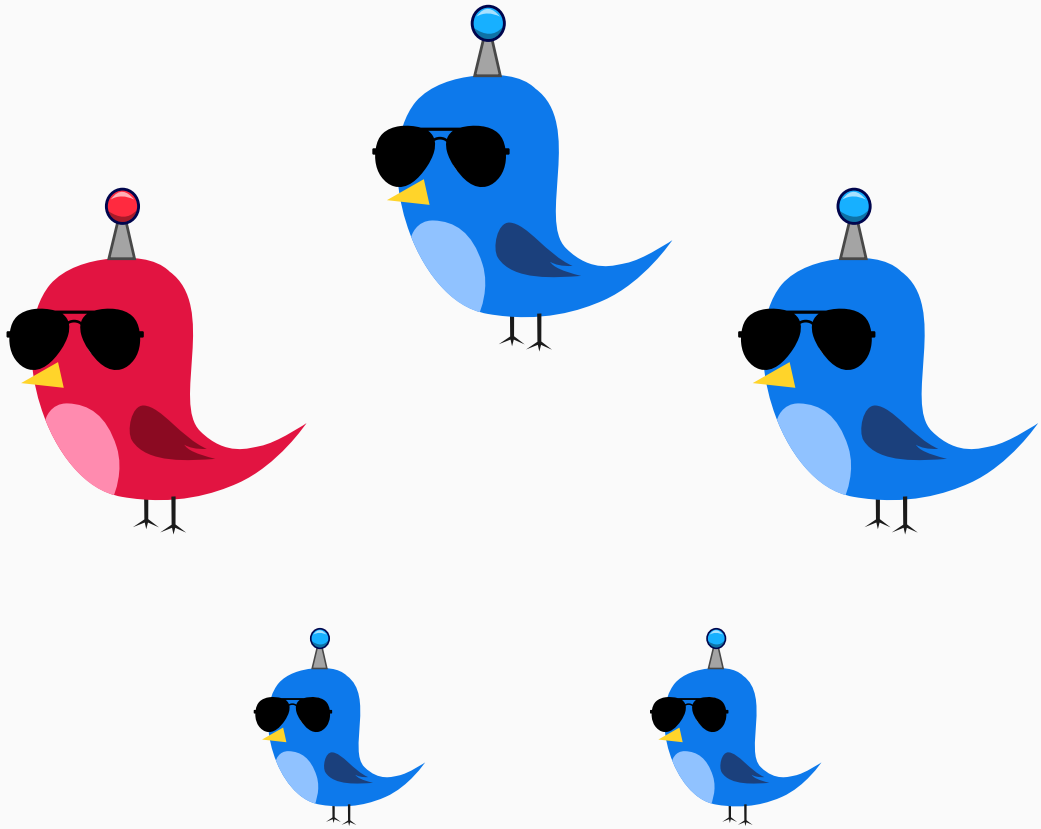


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

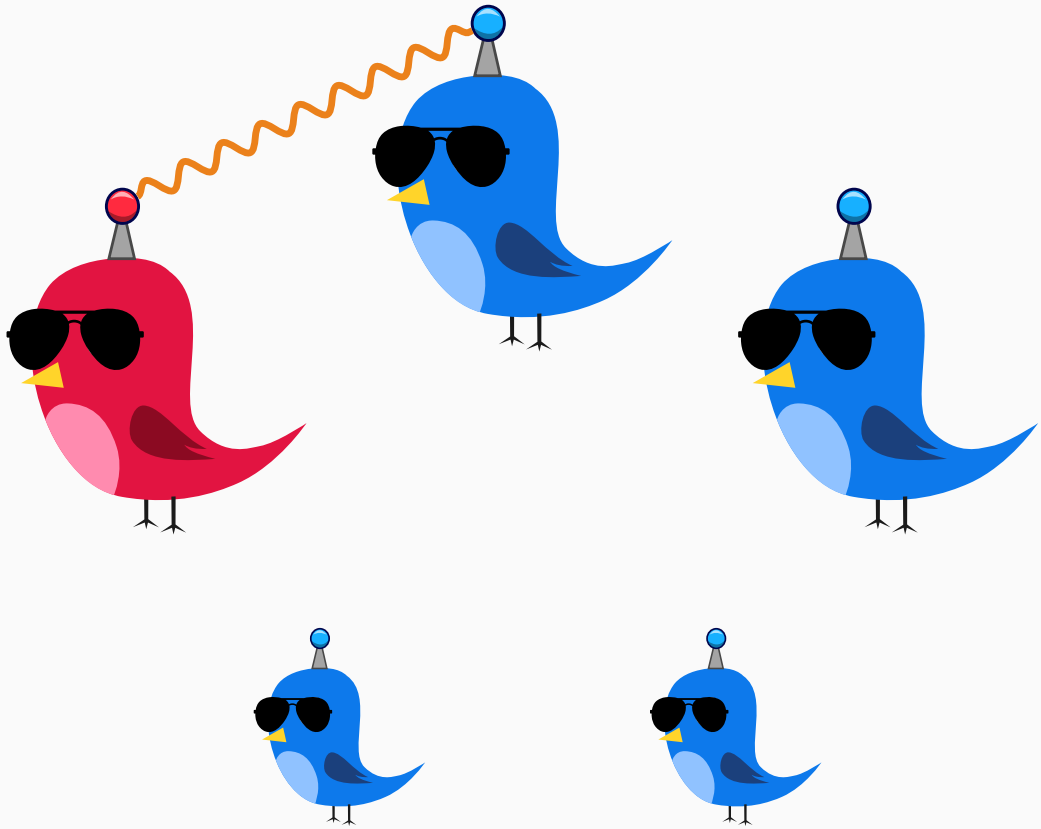


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

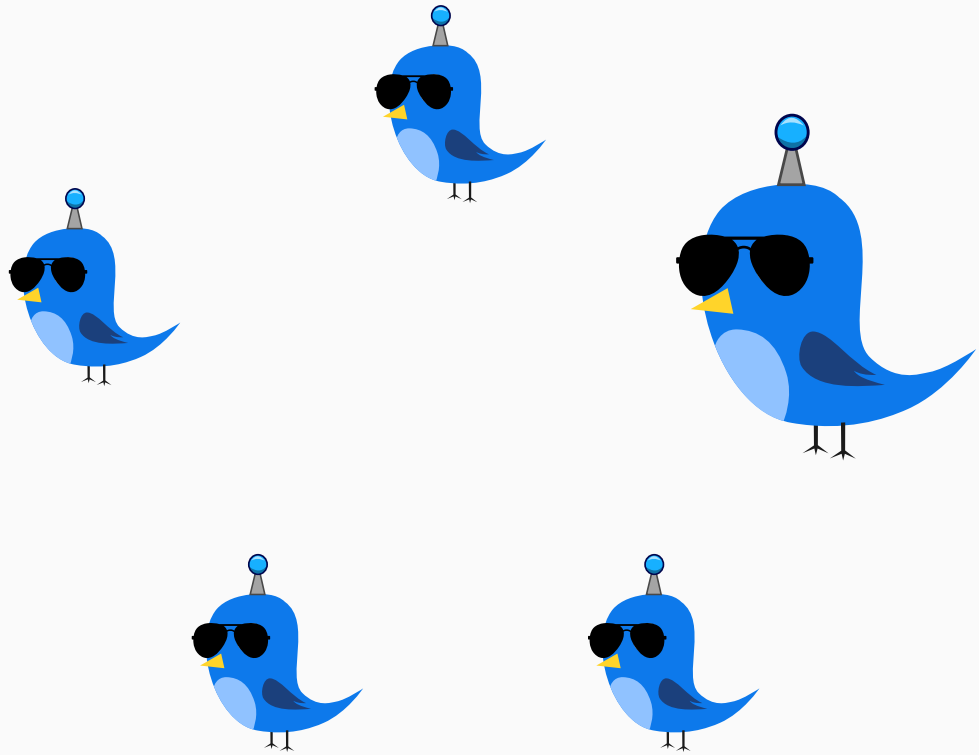


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

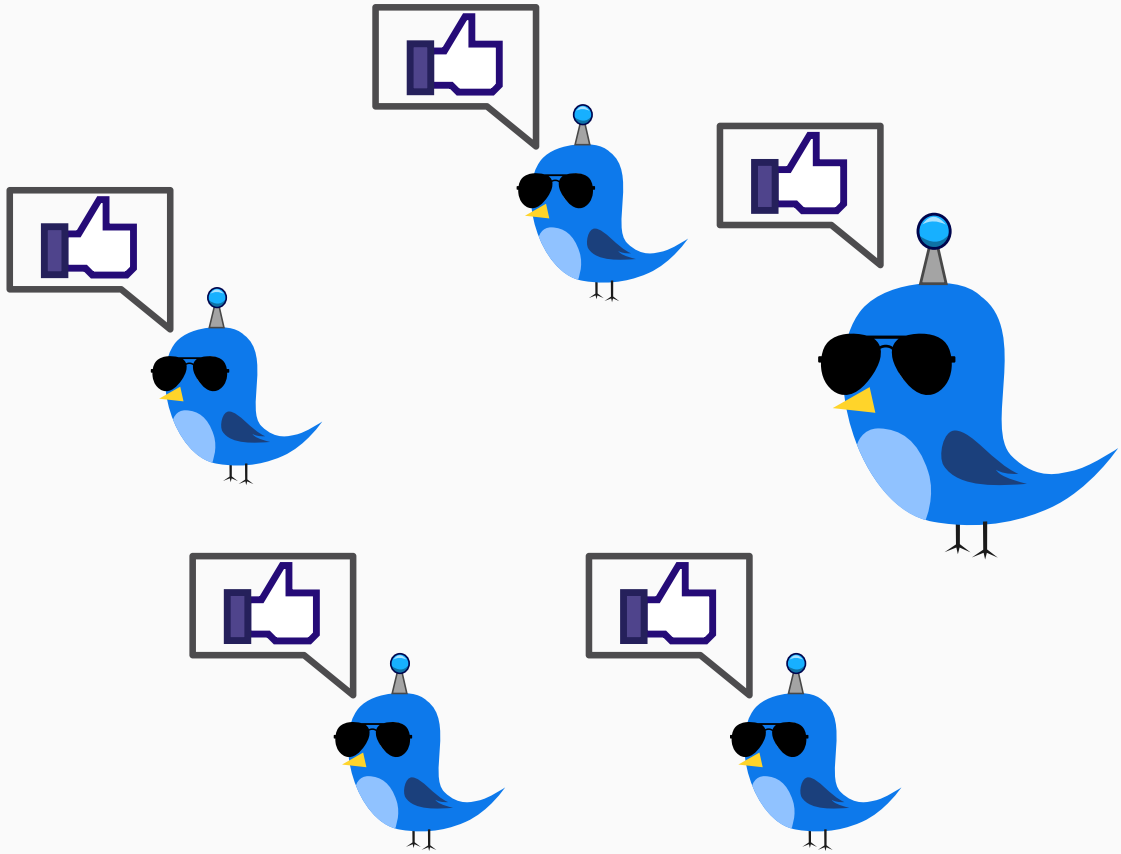


Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds



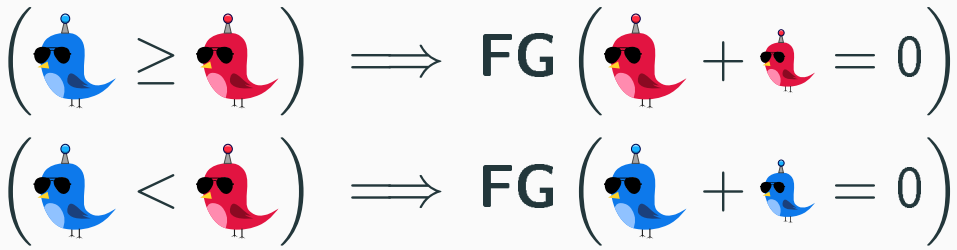
Example: majority voting protocol

At least as many blue birds as red birds?

Protocol:

- 4 states: blue/red, large/small
- Two large birds of different colors become small and blue
- Large birds convert small birds to their color
- Small blue birds convert small red birds

Correctness properties:

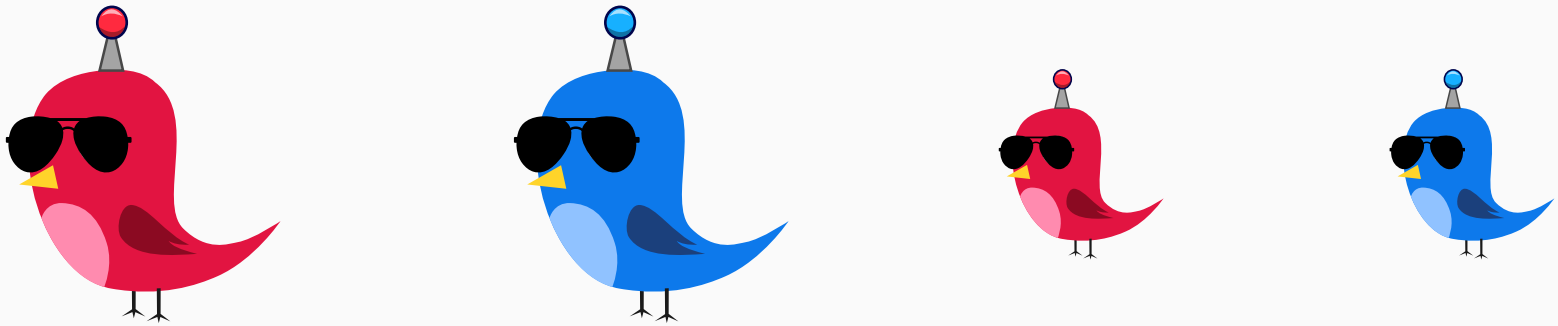


"Birds converge to color of majority."

Population Protocols: Formal Model

Angluin *et al.* PODC'04

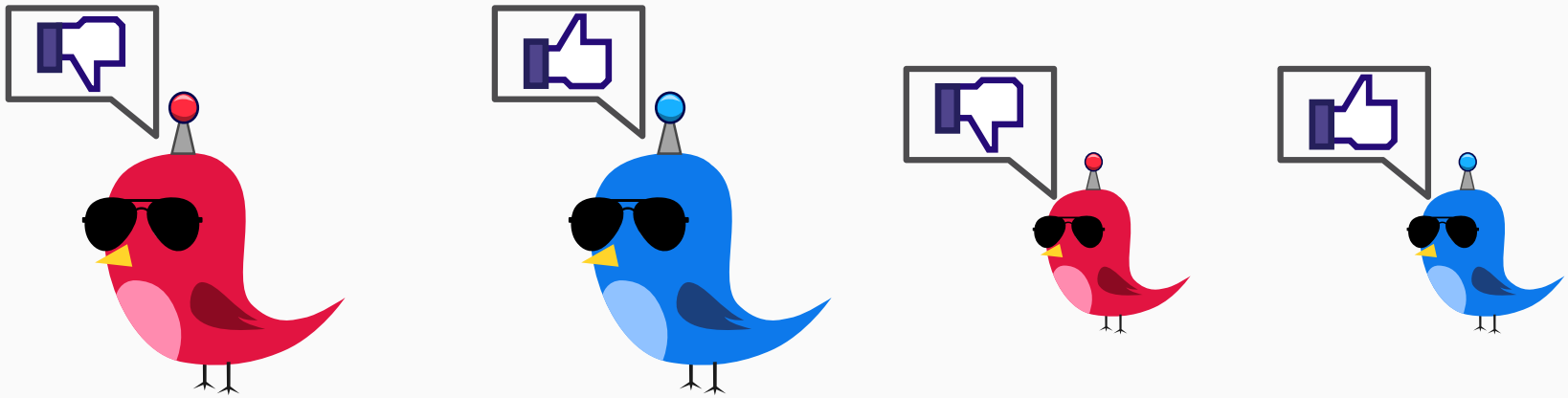
- States: finite set Q
- Opinions: $O : Q \to \{0, 1\}$
- Initial states: $I \subseteq Q$
- Transitions: $T \subseteq \bigcup_{k \geq 2} Q^{\langle k \rangle} \times Q^{\langle k \rangle}$



Population Protocols: Formal Model

Angluin *et al.* PODC'04

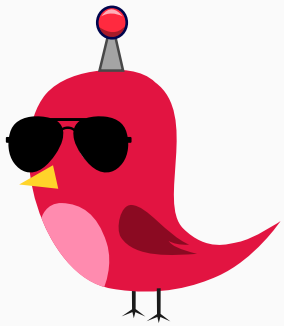
- States: finite set Q
- Opinions: $O : Q \to \{0, 1\}$
- Initial states: $I \subseteq Q$
- Transitions: $T \subseteq \bigcup_{k \geq 2} Q^{\langle k \rangle} \times Q^{\langle k \rangle}$



Population Protocols: Formal Model

Angluin *et al.* PODC'04

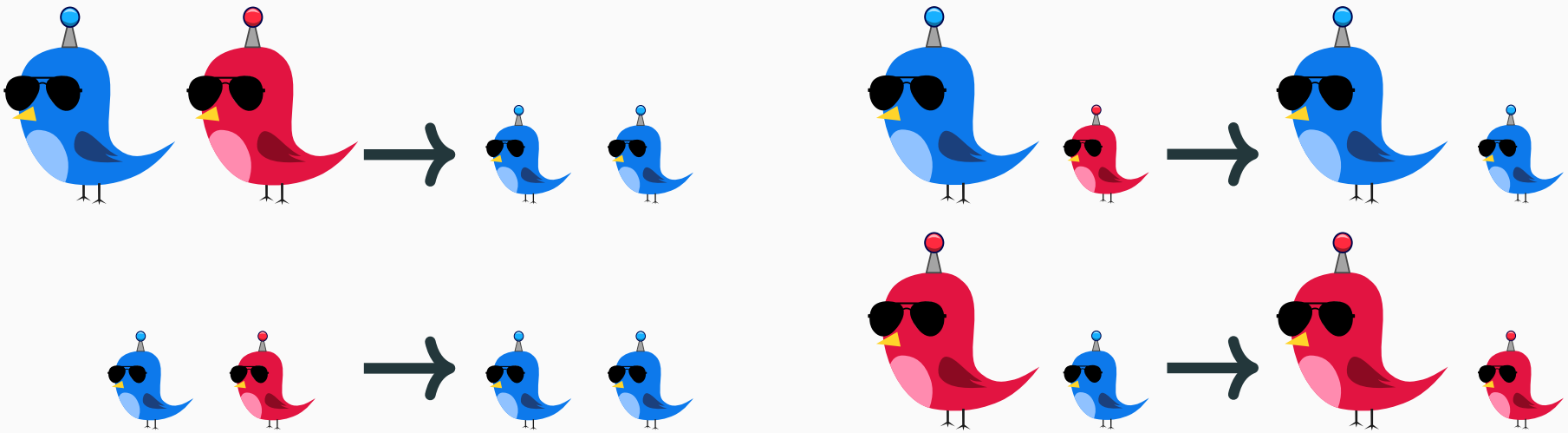
- States: finite set Q
- Opinions: $O : Q \to \{0, 1\}$
- Initial states: $I \subseteq Q$
- Transitions: $T \subseteq \bigcup_{k \geq 2} Q^{\langle k \rangle} \times Q^{\langle k \rangle}$



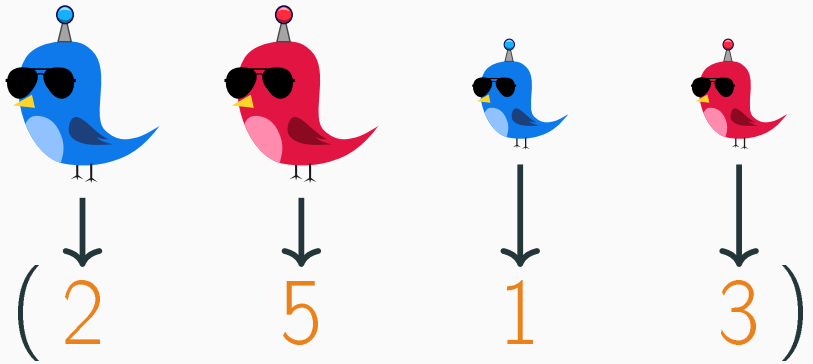
Population Protocols: Formal Model

Angluin *et al.* PODC'04

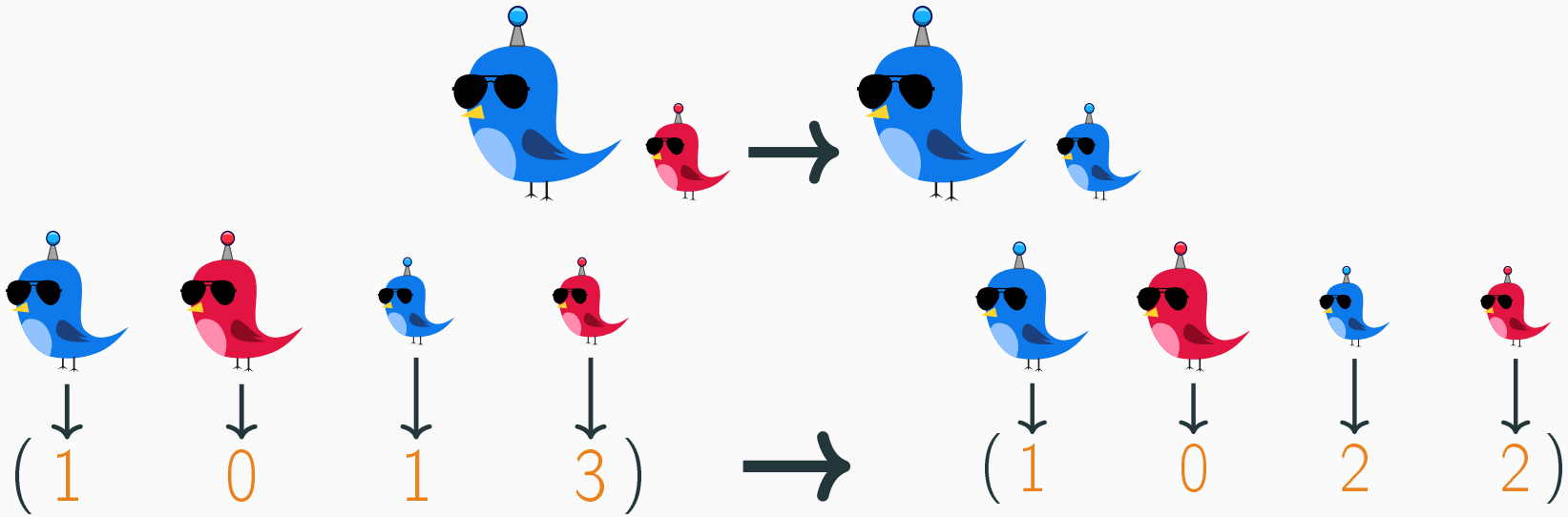
- States: finite set Q
- Opinions: $O : Q \to \{0, 1\}$
- Initial states: $I \subseteq Q$
- Transitions: $T \subseteq \bigcup_{k \geq 2} Q^{\langle k \rangle} \times Q^{\langle k \rangle}$



- Configurations: $Q \to \mathbb{N}$
- Transitions induce step relation $C \to C'$ between configurations

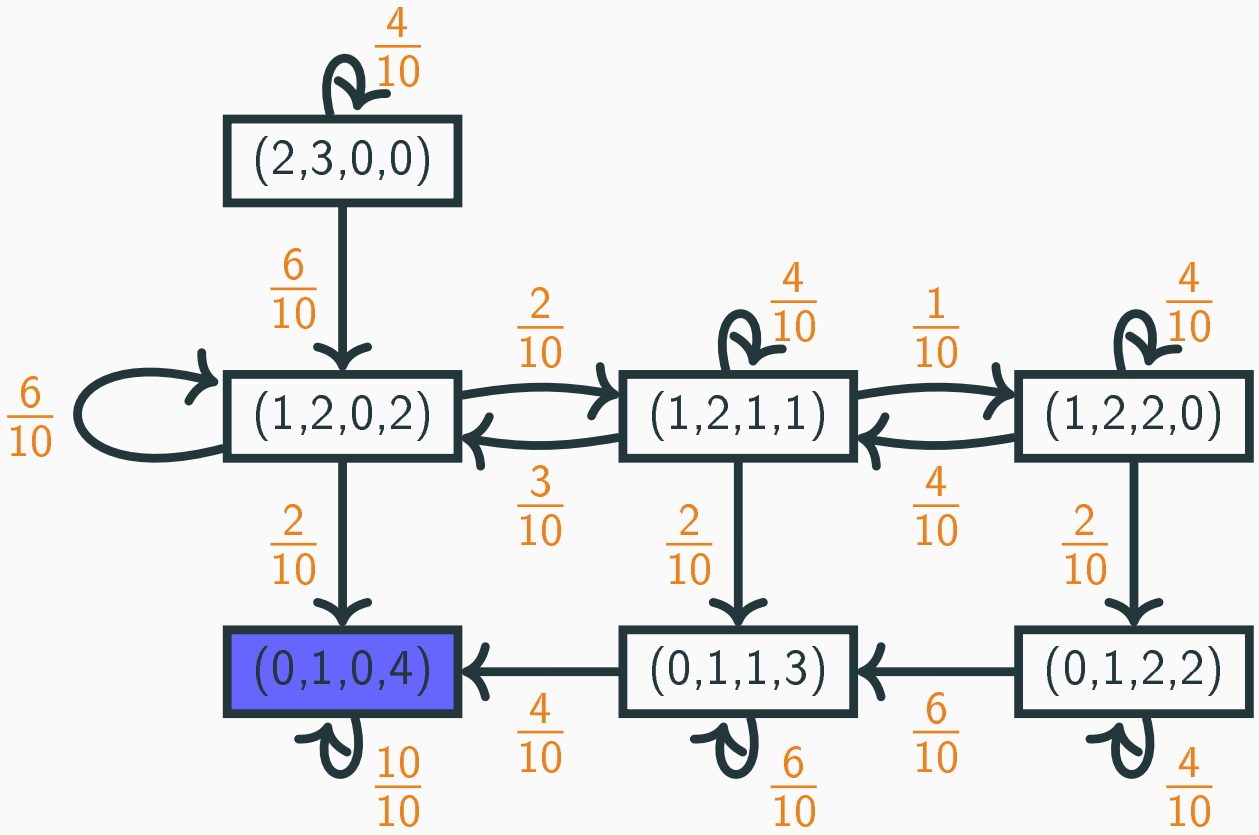


- Configurations: $Q \to \mathbb{N}$
- Transitions induce step relation $C \to C'$ between configurations



Underlying Markov chain for $(2, 3, 0, 0)$:

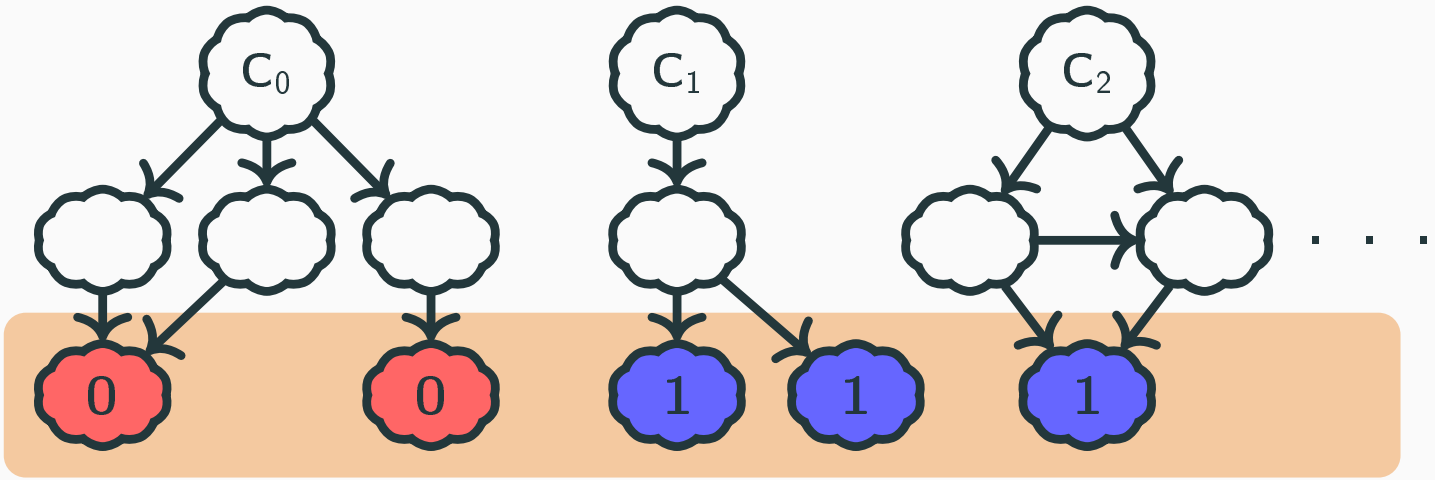
(pairs of agents are picked uniformly at random)



Population Protocols: Computing Predicates

Protocol computes φ: InitC $\to \{0,1\}$:

for every $C \in \mathsf{InitC}$, the runs starting at C reach **stable consensus** $\varphi(C)$ with probability 1.



Protocol computes $\varphi(C_0) = 0, \varphi(C_1) = 1, \varphi(C_2) = 1, \ldots$

Peregrine: Demo

Demo:

https://peregrine.model.in.tum.de/demo/

Peregrine: v2.0

New version brings:

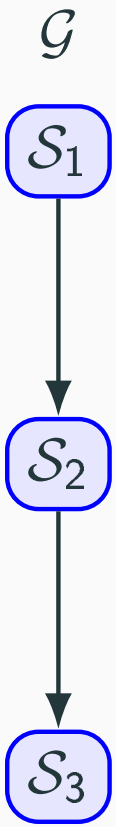
- automatic verification for even larger group of protocols
- utilizes stage graphs [Blodin et al., CAV'2020]

Idea

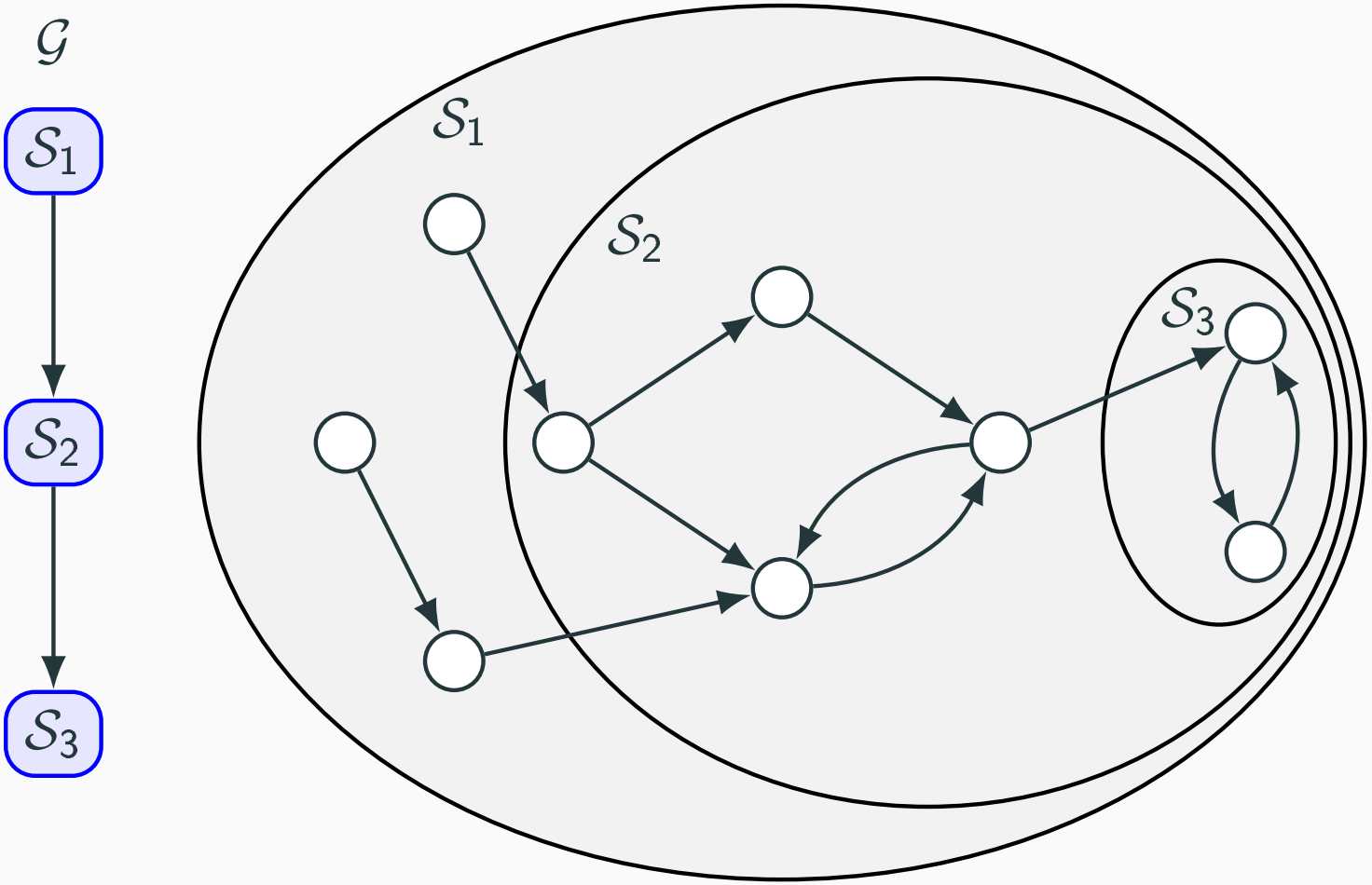
Most stable termination proofs are structured in stages: milestones trapping the system in increasingly smaller sets of configurations, until it gets trapped in the correct consensus

Stage Graphs

Stage Graphs

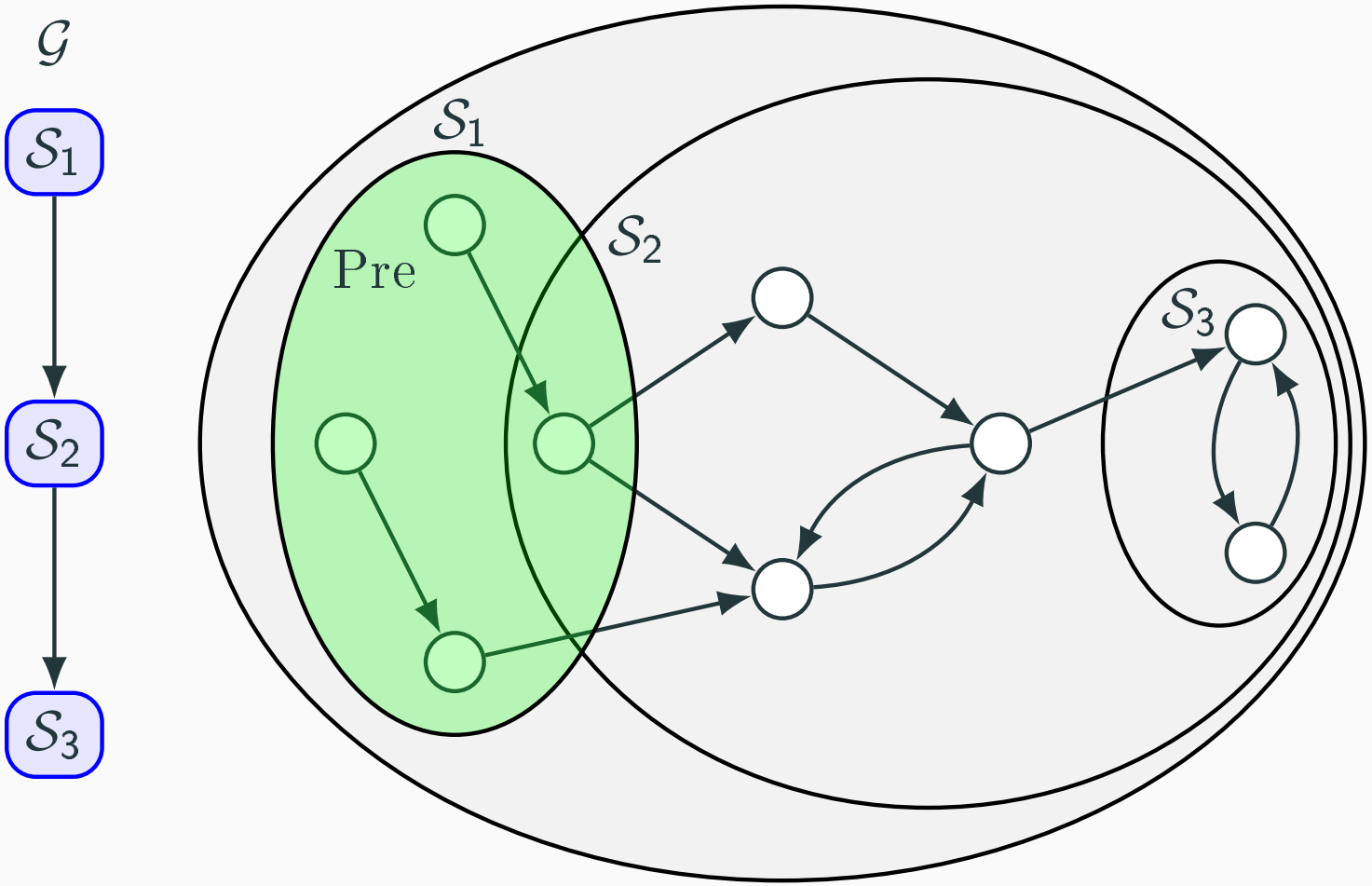


Stage Graphs



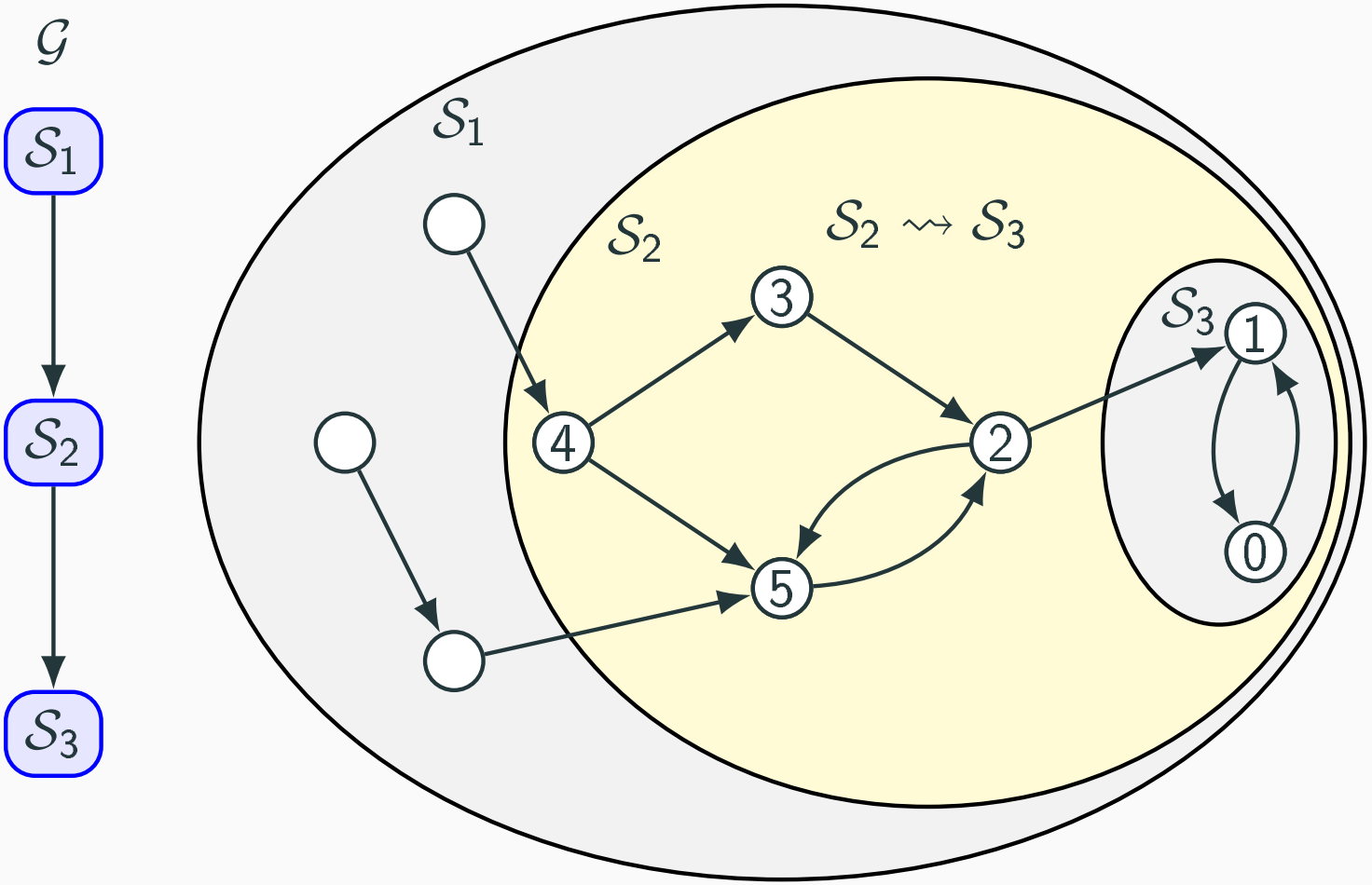
Stages $\mathcal{S}$ of $\mathcal{G}$ are inductive sets

Stage Graphs



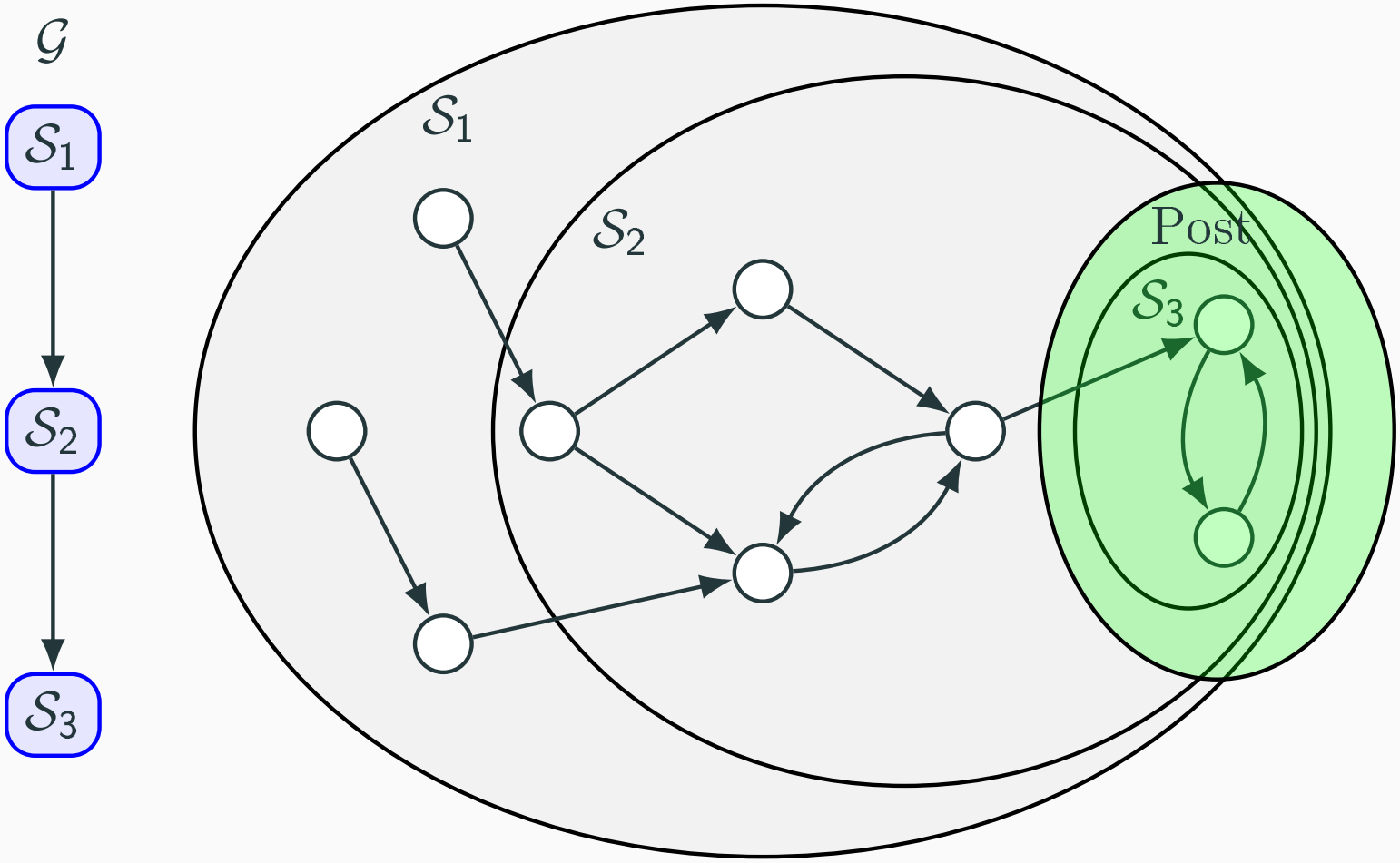
$$\text{Pre} \subseteq \bigcup_{\mathcal{S} \in \mathcal{G}} \mathcal{S}$$

Stage Graphs



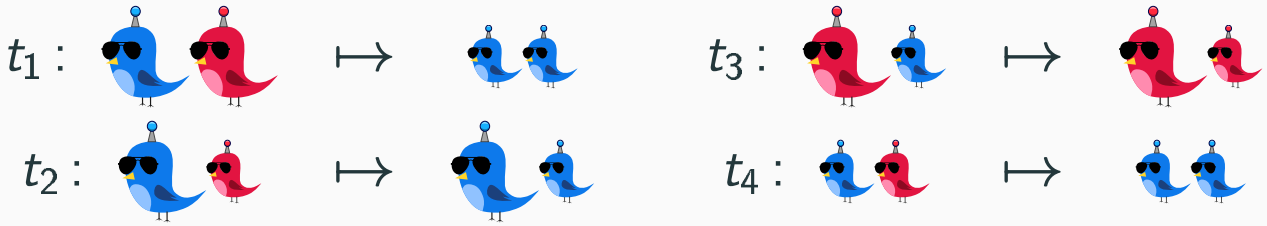
Certificates for non-terminal stages $\mathcal{S} \rightsquigarrow \underbrace{\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_k}_{\text{children of } \mathcal{S}}$

Stage Graphs

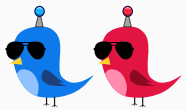


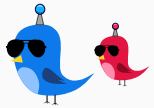
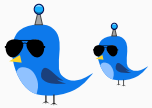
Terminal stages $\mathcal{S} \subseteq \mathrm{Post}$

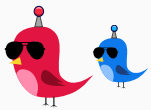
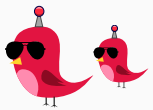
Stage Graph Example: majority voting protocol



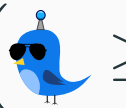
Stage Graph Example: majority voting protocol

t_1 :  $\mapsto$ 

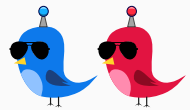
t_2 :  $\mapsto$ 

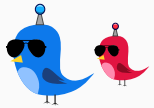
t_3 :  $\mapsto$ 

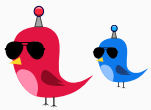
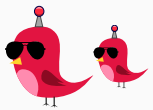
t_4 :  $\mapsto$ 

Stage graph for property $\left(\text{  } \geq \text{  } \right) \implies \mathbf{FG} \left(\text{  } + \text{  } = 0 \right)$

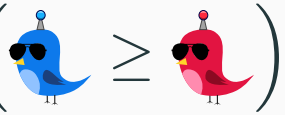
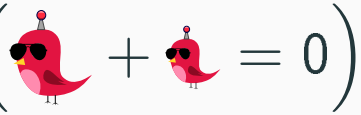
Stage Graph Example: majority voting protocol

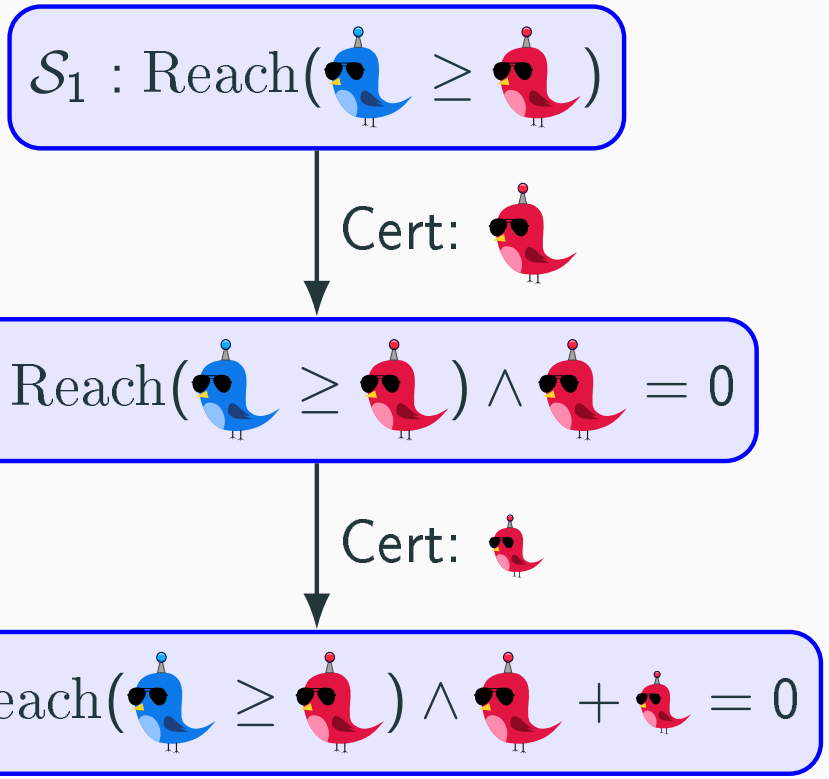
t_1 :  $\mapsto$ 

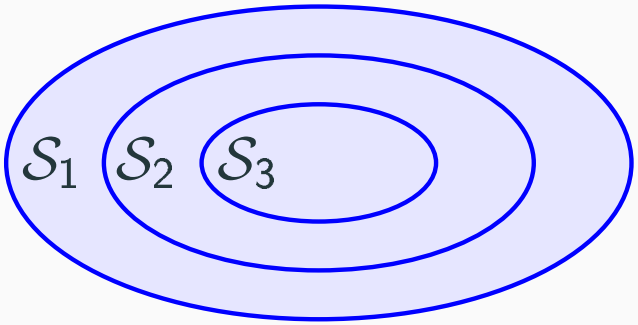
t_2 :  $\mapsto$ 

t_3 :  $\mapsto$ 

t_4 :  $\mapsto$ 

Stage graph for property  $\Longrightarrow$ **FG** 





Stage Graphs: theory

Soundness **CAV 2020**

If there is a stage graph for a property, then it holds.

Completeness **CAV 2020**

If a property holds, then there is a stage graph proving it.

Stage Graphs: theory

Soundness **CAV 2020**

If there is a stage graph for a prop[illegible]en it holds.

Completeness **CAV 2020**

If a property holds, then there is a stage graph proving it.

Stage Graphs: theory

Soundness **CAV 2020**

If there is a stage graph for a property, then it holds.

Completeness **CAV 2020**

If a property holds, then there is a stage graph proving it.

What about decidability?

$\rightarrow$ unknown (stages can be arbitrarily complicated!)

Stage Graphs: theory

Soundness **CAV 2020**

If there is a stage graph for a property, then it holds.

Completeness **CAV 2020**

If a property holds, then there is a stage graph proving it.

A Presburger stage graph is a stage graph where

- nodes are Presburger sets, and
- certificates are Presburger certificates.

$$C \in \mathcal{S} \iff \phi(C)$$

$$f(C) = a \iff \phi(C, a)$$

Stage Graphs: theory

Soundness **CAV 2020**

If there is a Presburger stage graph for a property, then it holds.

Completeness **CAV 2020**

If a property holds, then there is a Presburger stage graph proving it.

A Presburger stage graph is a stage graph where

- nodes are Presburger sets, and
- certificates are Presburger certificates.

$$C \in \mathcal{S} \iff \phi(C)$$

$$f(C) = a \iff \phi(C, a)$$

Stage Graphs: theory

Soundness **CAV 2020**

If there is a Presburger stage graph [illegible] property, then it holds.

HARD

Completeness **CAV 2020**

If a property holds, then there is a Presburger stage graph proving it.

A Presburger stage graph is a stage graph where

- nodes are Presburger sets, and
- certificates are Presburger certificates.

$$C \in \mathcal{S} \iff \phi(C)$$

$$f(C) = a \iff \phi(C, a)$$

Stage Graphs: theory

Soundness **CAV 2020**

If there is a Presburger stage graph for a property, then it holds.

Completeness **CAV 2020**

If a property holds, then there is a Presburger stage graph proving it.

Presburger stage graph can be independently checked!

$\rightarrow$ everything reduces to checking Presburger formulas

Stage Graphs: theory

Soundness **CAV 2020**

If there is a Presburger stage graph for a property, then it holds.

Completeness **CAV 2020**

If a property holds, then there is a Presburger stage graph proving it.

Decidability **CAV 2020**

It is decidable if a system satisfies a given stable termination property.

Stage Graphs: theory

Soundness **CAV 2020**

If there is a Presburger stage graph for a property, then it holds.

Completeness **CAV 2020**

If a property holds, then there is a Presburger stage graph proving it.

Decidability **CAV 2020**

It is decidable if a system satisfies a given stable termination property.

Problem: Presburger stage graphs might be huge (non-elementary)

$\rightarrow$ How can stage graphs help with automatic verification?

Stage Graphs: in practice

Ideas:

- Most systems have small stage graphs

Stage Graphs: in practice

Ideas:

- Most systems have small stage graphs
- Most systems "make progress" by "killing" transitions

Definition

A transition is dead if it can never be enabled again.

Stage Graphs: in practice

Ideas:

- Most systems have small stage graphs
- Most systems "make progress" by "killing" transitions
  $\rightarrow$ search for stages with more and more dead transitions

Definition

A transition is dead if it can never be enabled again.

Stage Graphs: in practice

Ideas:

- Most systems have small stage graphs
- Most systems "make progress" by "killing" transitions
  $\rightarrow$ search for stages with more and more dead transitions

Definition

A transition is dead if it can never be enabled again.

Algorithm:

SMT based semi-algorithm to automatically construct Presburger stage graphs

Peregrine: v2.0

New version brings:

- automatic verification for even larger group of protocols
- utilizes stage graphs [Blodin et al., CAV'2020]
- proof certificates

Peregrine: v2.0

New version brings:

- automatic verification for even larger group of protocols
- utilizes stage graphs [Blodin et al., CAV'2020]
- proof certificates
- speed bounds

Peregrine: v2.0

New version brings:

- automatic verification for even larger group of protocols
- utilizes stage graphs [Blodin et al., CAV'2020]
- proof certificates
- speed bounds
- interactive visualization of stage graphs

Peregrine: v2.0

New version brings:

- automatic verification for even larger group of protocols
- utilizes stage graphs [Blodin et al., CAV'2020]
- proof certificates
- speed bounds
- interactive visualization of stage graphs

$\Rightarrow$ helps user to understand protocol and its correctness

Peregrine: v2.0

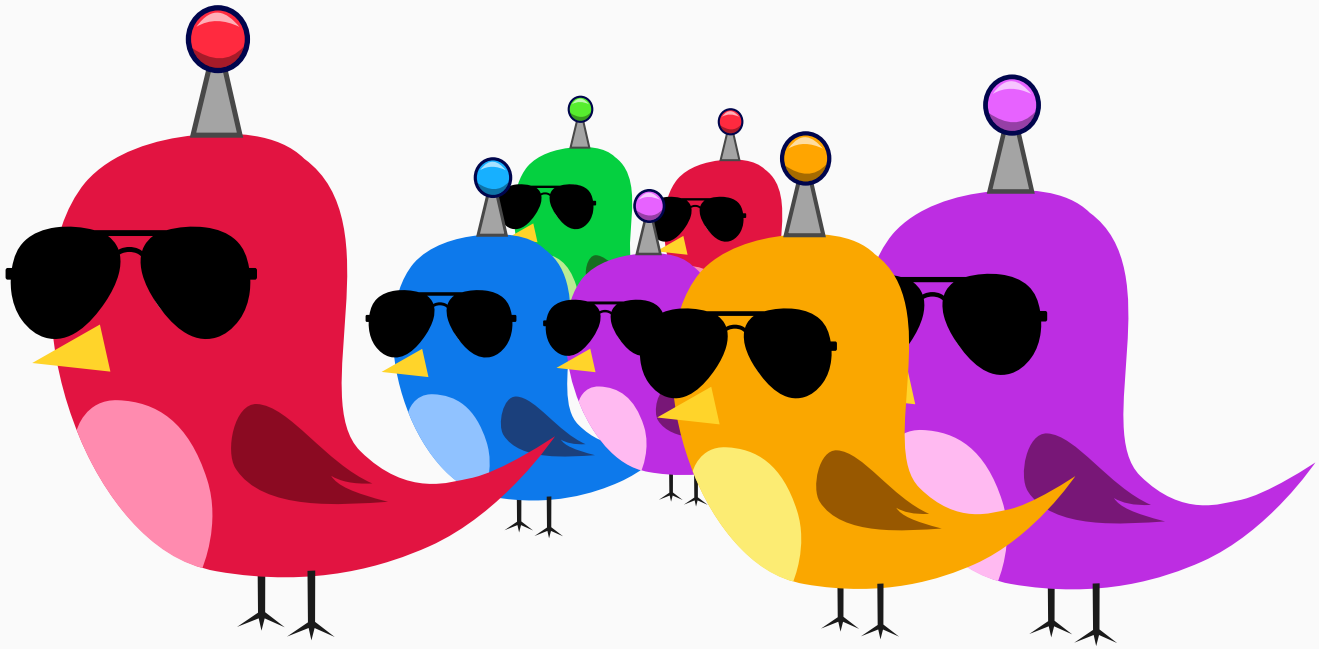
New version brings:

- automatic verification for even larger group of protocols
- utilizes stage graphs [Blodin et al., CAV'2020]
- proof certificates
- speed bounds
- interactive visualization of stage graphs

$\Rightarrow$ helps user to understand protocol and its correctness

Demo:

https://peregrine.model.in.tum.de/demo/



THANK YOU!