Black Ninjas in the Dark: Forman Analysis of Population Protocols

Javier Esparza
Joint work with Michael Blondin, Pierre Ganty, Stefan Jaax, Antonín Kučera, Jérôme Leroux, Rupak Majumdar, Philipp J. Meyer, and Chana Weil-Kennedy
• Deaf Black Ninjas meet at a Zen garden in the dark
Deaf Black Ninjas in the Dark

- Deaf Black Ninjas meet at a Zen garden in the dark
- They must decide by majority to attack or not (no attack if tie)
Deaf Black Ninjas in the Dark

- Deaf Black Ninjas meet at a Zen garden in the dark
- They must decide by majority to attack or not (no attack if tie)
• Deaf Black Ninjas meet at a Zen garden in the dark
• They must decide by majority to attack or not (no attack if tie)
• How can they conduct the vote?
• Ninjas wander *randomly*, interacting when they bump into each other.
• Ninjas wander randomly, interacting when they bump into each other.
• Ninjas store their current estimation of the final outcome: **attack** or **don’t attack**.
Ninjas wander randomly, interacting when they bump into each other.

Ninjas store their current estimation of the final outcome: attack or don’t attack.

Additionally, they are active or passive.
Deaf Black Ninjas in the Dark

- Ninjas wander randomly, interacting when they bump into each other.
- Ninjas store their current estimation of the final outcome: **attack** or **don’t attack**.
- Additionally, they are active or passive.

*Initially: all ninjas active, estimation = own vote.*
Goal of voting protocol:

- eventually all ninjas reach the same estimation, and
- this estimation corresponds to the majority.
Goal of voting protocol:

• eventually all ninjas reach the same estimation, and
• this estimation corresponds to the majority.

Graphically:

• Initially more red ninjas $\rightarrow$ eventually all ninjas red.
• Initially more blue ninjas or tie $\rightarrow$ eventually all ninjas blue.
Majority protocol: Are there more red ninjas than blue ninjas?
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue.

![Diagram showing the change from active to passive ninjas]

![Diagram showing the conversion of passive ninjas to their color]

Majority protocol: Are there more red ninjas than blue ninjas?

• Active ninjas of opposite colors become passive and blue
Majority protocol: Are there more red ninjas than blue ninjas?

• Active ninjas of opposite colors become passive and blue
Majority protocol: Are there more **red ninjas** than **blue ninjas**?

- Active ninjas of opposite colors become passive and **blue**.
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue.
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue

- Active ninjas convert passive ninjas to their color
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue.

- Active ninjas convert passive ninjas to their color.
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue

- Active ninjas convert passive ninjas to their color
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue
- Active ninjas convert passive ninjas to their color
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue

- Active ninjas convert passive ninjas to their color
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue.
- Active ninjas convert passive ninjas to their color.
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue
- Active ninjas convert passive ninjas to their color
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue.
- Active ninjas convert passive ninjas to their color.
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue.

- Active ninjas convert passive ninjas to their color.
Majority protocol: Are there more red ninjas than blue ninjas?

- Active ninjas of opposite colors become passive and blue

- Active ninjas convert passive ninjas to their color

Go!
Sad story ...
• The first rule has no priority over the other two.
Majority protocol: Why?

- The first rule has no priority over the other two.
Majority protocol: Why?

- The first rule has no priority over the other two.
Majority protocol: Why?

- The first rule has no priority over the other two.
• The first rule has no priority over the other two.
• The first rule has no priority over the other two.
• The first rule has no priority over the other two.
Majority protocol: Why?

• The first rule has no priority over the other two.

NO CONSENSUS!
Sensei II’s protocol: Are there more red ninjas than blue ninjas?

Interaction rules:
Sensei II’s protocol: Are there more red ninjas than blue ninjas?

Interaction rules:

Passive blue ninjas convert passive red ninjas to their color
Sensei II’s protocol: Are there more red ninjas than blue ninjas?

Interaction rules:

Passive blue ninjas convert passive red ninjas to their color.
Sensei II’s protocol: Are there more red ninjas than blue ninjas?

Interaction rules:

Passive blue ninjas convert passive red ninjas to their color.
Sensei II’s protocol: Are there more red ninjas than blue ninjas?

Interaction rules:

Passive blue ninjas convert passive red ninjas to their color.

Sensei II Go!
Sensei II’s protocol: Are there more red ninjas than blue ninjas?

Expected number of steps to stable consensus for a population of 15 ninjas.
Very sad story...
Sensei III’s protocol

 meis = Attack majority  = Don’t attack majority = Tie

Interaction rules:

Go!
Sensei III’s protocol

Initial number of red ninjas

Expected number of steps to stable consensus for a population of 15 ninjas.
Sensei III’s questions

Formalization questions:

- What is a protocol?
- When is a protocol "correct"?
- When is a protocol "efficient"?
Verification questions:

• How do I check that my protocol is correct?
• How do I check that my protocol is efficient?
Sensei III’s questions

Expressivity questions:

- Are there protocols for other problems?
- How large is the smallest protocol for a problem?
- And the smallest efficient protocol?
Formal model of distributed computation by collections of **identical, finite-state, and mobile** agents like...
Population protocols

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
Population protocols

Angluin, Aspnes et al. PODC’04

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 ninjas!
Population protocols: formal model

Angluin, Aspnes et al. PODC’04

- **States:** finite set $Q$
- **Opinions:** $O : Q \rightarrow \{0, 1\}$
- **Initial states:** $I \subseteq Q$
- **Transitions:** $T \subseteq Q^2 \times Q^2$
• **States:** finite set $Q$
• **Opinions:** $O : Q \rightarrow \{0, 1\}$
• **Initial states:** $I \subseteq Q$
• **Transitions:** $T \subseteq Q^2 \times Q^2$
Population protocols: formal model

- **States:** finite set $Q$
- **Opinions:** $O : Q \rightarrow \{0, 1\}$
- **Initial states:** $I \subseteq Q$
- **Transitions:** $T \subseteq Q^2 \times Q^2$
• **States:** finite set \( Q \)
• **Opinions:** \( O : Q \rightarrow \{0, 1\} \)
• **Initial states:** \( I \subseteq Q \)
• **Transitions:** \( T \subseteq Q^2 \times Q^2 \)
• **States:** finite set $Q$
• **Opinions:** $O : Q \rightarrow \{0, 1\}$
• **Initial states:** $I \subseteq Q$
• **Transitions:** $T \subseteq Q^2 \times Q^2$
• **Configurations:** $Q \rightarrow \mathbb{N}$
Population protocols: formal model

- **States:** finite set $Q$
- **Opinions:** $O : Q \rightarrow \{0, 1\}$
- **Initial states:** $I \subseteq Q$
- **Transitions:** $T \subseteq Q^2 \times Q^2$
- **Configurations:** $Q \rightarrow \mathbb{N}$
- **Initial configurations:** $I \rightarrow \mathbb{N}$
Reachability graph for \((3, 2, 0, 0)\):
Population protocols: runs

Underlying Markov chain:
(pairs of agents are picked uniformly at random)
Run: infinite path from initial configuration
Protocol computes $\varphi : \text{InitC} \rightarrow \{0, 1\}$: for every $C \in \text{InitC}$, the runs starting at $C$ reach **stable consensus** $\varphi(C)$ with probability 1.
**Protocol computes** $\varphi : \text{InitC} \rightarrow \{0, 1\}$: for every $C \in \text{InitC}$, the runs starting at $C$ reach **stable consensus** $\varphi(C)$ with probability 1.

Protocol computes $\varphi(C_0) = 0$, $\varphi(C_1) = 1$, $\varphi(C_2) = 1$, …
Population protocols: computing predicates

Protocol computes $\varphi : \text{InitC} \rightarrow \{0, 1\}$:
for every $C \in \text{InitC}$, the runs starting at $C$ reach **stable consensus** $\varphi(C)$ with probability 1.

Protocol ill defined for $C_1$
Protocol computes $\varphi : \text{InitC} \rightarrow \{0, 1\}$: for every $C \in \text{InitC}$, the runs starting at $C$ reach **stable consensus** $\varphi(C)$ with probability 1.

Protocol ill defined for $C_1$ (Sensei I's problem)
A protocol is **well specified** if it computes some predicate.
A protocol is **well specified** if it computes some predicate

A protocol for a predicate $\phi$ is **correct** if it computes $\phi$ (in particular, correct protocols are well specified)
Sensei III’s questions

What predicates can we compute?
How fast can we compute them?
How succinctly can we compute them?
How can I check correctness?
How can I check efficiency?

To conclude ...
Angluin, Aspnes, Eisenstat Dist. Comp.’07

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

Proof: PPs compute all Presburger predicates
Since Presburger arithmetic has quantifier elimination, it suffices to:
Population protocols compute precisely the predicates definable in Presburger arithmetic, i.e. $\text{FO}(\mathbb{N}, +, <)$

Proof: PPs compute all Presburger predicates

Since Presburger arithmetic has quantifier elimination, it suffices to:

- Exhibit PPs for threshold and modulo predicates

\[
a_1x_1 + \cdots + a_n c_n \leq b \quad a_1x_1 + \cdots + a_n c_n \equiv b \pmod{c}
\]
Expressive power

Angluin, Aspnes, Eisenstat Dist. Comp.’07

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

Proof: PPs compute all Presburger predicates

Since Presburger arithmetic has quantifier elimination, it suffices to:

- Exhibit PPs for threshold and modulo predicates

$$a_1x_1 + \cdots + a_n c_n \leq b \quad a_1x_1 + \cdots + a_n c_n \equiv b \ (\text{mod } c)$$

- Prove that computable predicates are closed under negation and conjunction
# Expressive power

<table>
<thead>
<tr>
<th><strong>Angluin, Aspnes, Eisenstat Dist. Comp.’07</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Population protocols compute precisely the predicates definable in Presburger arithmetic, <em>i.e.</em> $\text{FO}(\mathbb{N}, +, &lt;)$</td>
</tr>
</tbody>
</table>

**Proof: PPs only compute Presburger predicates**

- Much harder!
Expressive power

**Angluin, Aspnes, Eisenstat Dist. Comp.’07**
Population protocols compute precisely the predicates definable in Presburger arithmetic, *i.e.* $\text{FO}(\mathbb{N}, +, <)$

**Proof: PPs only compute Presburger predicates**

- Much harder!
- Dist. Comp.’07 proof is “non-constructive”
**Expressive power**

<table>
<thead>
<tr>
<th>Angluin, Aspnes, Eisenstat Dist. Comp.’07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population protocols compute precisely the predicates definable in Presburger arithmetic, <em>i.e.</em> $\text{FO}({\mathbb{N}, +, &lt;})$</td>
</tr>
</tbody>
</table>

**Proof: PPs only compute Presburger predicates**

- Much harder!
- Dist. Comp.’07 proof is “non-constructive”
- “Constructive” proof by E., Ganty, Leroux, Majumdar Acta Inf.’17
Expressive power

**Angluin, Aspnes, Eisenstat Dist. Comp.’07**
Population protocols compute precisely the predicates definable in Presburger arithmetic, *i.e.* $\text{FO}(\mathbb{N}, +, <)$

**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
Sensei III’s questions

What predicates can we compute?
How fast can we compute them?
How succinctly can we compute them?
How can I check correctness?
How can I check efficiency?

To conclude ...
Efficiency measured by the expected number of interactions until stable consensus: $\text{Inter}(n)$
Efficiency measured by the expected number of interactions until stable consensus: $\text{Inter}(n)$

depends on the population size $n$
Efficiency measured by the expected number of interactions until stable consensus: $\text{Inter}(n)$

Depends on the population size $n$

In a natural model: expected (parallel) time to consensus satisfies

$$\text{Time}(n) = \frac{\text{Inter}(n)}{n}$$
<table>
<thead>
<tr>
<th><strong>Angluin, Aspnes et al., PODC’04</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Every Presburger predicate is computable by a protocol with $\text{Inter}(n) \in \mathcal{O}(n^2 \log n)$</td>
</tr>
<tr>
<td>Angluin, Aspnes et al.</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>Every Presburger predicate is computable by a protocol with $\text{Inter}(n) \in O(n^2 \log n)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Angluin, Aspnes, Eisenstat</th>
<th>Dist.Comp.’08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Every Presburger predicate is computable by a protocol with a leader and $\text{Inter}(n) \in O(n \log^{O(1)}(n))$</td>
<td></td>
</tr>
</tbody>
</table>
### Efficiency

<table>
<thead>
<tr>
<th><strong>Angluin, Aspnes et al.</strong> , <em>PODC’04</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Every Presburger predicate is computable by a protocol with ( \text{Inter}(n) \in \mathcal{O}(n^2 \log n) )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Angluin, Aspnes, Eisenstat</strong> Dist.Compo.’08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Every Presburger predicate is computable by a protocol with a leader and ( \text{Inter}(n) \in \mathcal{O}(n \log^{O(1)}(n)) )</td>
</tr>
</tbody>
</table>

**Open** whether \( \mathcal{O}(n \log^{O(1)}(n)) \) achievable without leaders.
Sensei III’s questions

What predicates can we compute?
How fast can we compute them?
How succinctly can we compute them?
How can I check correctness?
How can I check efficiency?

To conclude ...
Protocol for: Are there at least 4 sick ninjas?

Each ninja is in a state of $f_0; 1; 2; 3; 4$.

- Initially, sick ninjas in state $1$.
- Healthy ninjas in state $0$.

\[
\binom{m+n}{4,4} \text{ if } m+n < 4
\]

\[
\binom{m+n}{7} \text{ if } m+n \geq 4
\]
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \(\{0, 1, 2, 3, 4\}\)
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)
- \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \{0, 1, 2, 3, 4\}
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)
- \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \(\{0, 1, 2, 3, 4\}\)
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)
- \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \{0, 1, 2, 3, 4\}
- Initially, sick ninjas in state 1, healthy ninjas in state 0
  - \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)
  - \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

• Each ninja is in a state of \{0, 1, 2, 3, 4\}

• Initially, sick ninjas in state 1, healthy ninjas in state 0

• \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)

• \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \{0, 1, 2, 3, 4\}
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)
- \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \{0, 1, 2, 3, 4\}
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)
- \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \{0, 1, 2, 3, 4\}
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- \((m, n) \rightarrow (m + n, 0)\) if \(m + n < 4\)
- \((m, n) \rightarrow (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

• Each ninja is in a state of \{0, 1, 2, 3, 4\}

• Initially, sick ninjas in state 1, healthy ninjas in state 0

• \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)

• \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

• Each ninja is in a state of \{0, 1, 2, 3, 4\}

• Initially, sick ninjas in state 1, healthy ninjas in state 0

• \((m, n) \mapsto (m + n, 0)\)
  if \(m + n < 4\)

• \((m, n) \mapsto (4, 4)\)
  if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

• Each ninja is in a state of \{0, 1, 2, 3, 4\}

• Initially, sick ninjas in state 1, healthy ninjas in state 0

• \((m, n) \mapsto (m + n, 0)\)
  if \(m + n < 4\)

• \((m, n) \mapsto (4, 4)\)
  if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

• Each ninja is in a state of \{0, 1, 2, 3, 4\}

• Initially, sick ninjas in state 1, healthy ninjas in state 0

• \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)

• \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \{0, 1, 2, 3, 4\}
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)
- \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \{0, 1, 2, 3, 4\}
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)
- \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \{0, 1, 2, 3, 4\}
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)
- \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least 4 sick ninjas?

- Each ninja is in a state of \{0, 1, 2, 3, 4\}

- Initially, sick ninjas in state 1, healthy ninjas in state 0

- \((m, n) \mapsto (m + n, 0)\) if \(m + n < 4\)

- \((m, n) \mapsto (4, 4)\) if \(m + n \geq 4\)
Protocol for: Are there at least $2^\ell$ sick ninjas?

- Each ninja is in a state of $\{0, 1, \ldots, 2^\ell - 1, 2^\ell\}$
- Initially, sick ninjas in state 1, healthy ninjas in state 0

- $(m, n) \mapsto (m + n, 0)$
  if $m + n < 2^\ell$

- $(m, n) \mapsto (2^\ell, 2^\ell)$
  if $m + n \geq 2^\ell$
Protocol for: Are there at least $2^\ell$ sick ninjas?

- Each ninja is in a state of $\{0, 1, \ldots, 2^\ell - 1, 2^\ell\}$
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- $(m, n) \mapsto (m + n, 0)$ if $m + n < 2^\ell$
- $(m, n) \mapsto (2^\ell, 2^\ell)$ if $m + n \geq 2^\ell$

- Each ninja is in a state of $\{0, 2^0, \ldots, 2^{\ell-1}, 2^\ell\}$
- Initially, sick ninjas in state $2^0$, healthy ninjas in state 0
- $(2^m, 2^n) \mapsto (2^{m+1}, 0)$ if $m + 1 \leq \ell$
- $(2^\ell, n) \mapsto (2^\ell, 2^\ell)$

Can be generalized to non-powers of 2
Protocol for: Are there at least $2^{\ell}$ sick ninjas?

- Each ninja is in a state of $\{0, 1, \ldots, 2^{\ell} - 1, 2^{\ell}\}$
- Initially, sick ninjas in state 1, healthy ninjas in state 0
- $(m, n) \leftrightarrow (m + n, 0)$ if $m + n < 2^{\ell}$
- $(m, n) \leftrightarrow (2^{\ell}, 2^{\ell})$ if $m + n \geq 2^{\ell}$

- Each ninja is in a state of $\{0, 2^0, \ldots, 2^{\ell-1}, 2^{\ell}\}$
- Initially, sick ninjas in state $2^0$, healthy ninjas in state 0
- $(2^m, 2^m) \leftrightarrow (2^{m+1}, 0)$ if $m + 1 \leq \ell$
- $(2^{\ell}, n) \leftrightarrow (2^{\ell}, 2^{\ell})$
- Can be generalized to non-powers of 2
Succinctness

Just gave a protocol for $X \geq c$ with $O(\log c)$ states.
Succinctness

Just gave a protocol for $X \geq c$ with $O(\log c)$ states. Is $O(\log \log c)$ possible?
Just gave a protocol for $X \geq c$ with $O(\log c)$ states. Is $O(\log \log c)$ possible?

Not for every $c$ ...

**Blondin, E., Jaax STACS’18**

There exist infinitely many $c$ such that every protocol for $X \geq c$ has at least $(\log c)^{1/4}$ states.
Succinctness

Just gave a protocol for $X \geq c$ with $O(\log c)$ states.

Is $O(\log \log c)$ possible?

Not for every $c$ ...

Blondin, E., Jaax STACS’18

There exist infinitely many $c$ such that every protocol for $X \geq c$ has at least $(\log c)^{1/4}$ states

...but for some $c$, if we allow leaders:

Blondin, E., Jaax STACS’18

For infinitely many $c$ there is a protocol with two leaders and $O(\log \log c)$ states that computes $X \geq c$
Blondin, E., Jaax  STACS’18

For infinitely many $c$ there is a protocol with two leaders and $O(\log \log c)$ states that computes $X \geq c$

**Proof:**
For infinitely many $c$ there is a protocol with two leaders and $O(\log \log c)$ states that computes $X \geq c$.

**Proof:**

- **Mayr and Meyer ’82:** For every $n$ there is a commutative semigroup presentation and two elements $s, t$ such that the shortest word $\alpha$ leading from $s$ to $t$ (i.e., $t = s \alpha$) has length $|\alpha| \geq 2^{2^n}$.
Succinctness

Blondin, E., Jaax STACS’18

For infinitely many $c$ there is a protocol with two leaders and $O(\log \log c)$ states that computes $X \geq c$

Proof:

- **Mayr and Meyer ’82**: For every $n$ there is a commutative semigroup presentation and two elements $s, t$ such that the shortest word $\alpha$ leading from $s$ to $t$ (i.e., $t = s \alpha$) has length $|\alpha| \geq 2^{2^n}$

- Construct a protocol that “simulates” derivations in the semigroup
$O(\log \log c)$ without leaders?
$O(\log \log c)$ without leaders? Open
$O(\log \log c)$ without leaders? Open

And $O(\log \log \log c)$?
$O(\log \log c)$ without leaders? Open

And $O(\log \log \log c)$? Open
$O(\log \log c)$ without leaders? \textcolor{red}{Open}

And $O(\log \log \log c)$? \textcolor{red}{Open}

$O(\log |\varphi|)$ states for all $\varphi$?
$O(\log \log c)$ without leaders? Open

And $O(\log \log \log c)$? Open

$O(\log |\varphi|)$ states for all $\varphi$? Open
Sensei III’s questions

What predicates can we compute?
How fast can we compute them?
How succinctly can we compute them?
How can I check correctness?
How can I check efficiency?

To conclude …
Protocols can become complex, even for $B \geq R$:

Fast and Exact Majority in Population Protocols

Dan Alistarh  
Microsoft Research

Rati Gelashvili*  
MIT

Milan Vojnović  
Microsoft Research

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

2  \text{sgn}(x) = \begin{cases} 
1 & \text{if } x \in \{0, 1, \ldots, 11, 3, 5, \ldots, m\}; \\
-1 & \text{otherwise.} 
\end{cases}

3  \text{value}(x) = \text{sgn}(x) \cdot \text{weight}(x)

/* Functions for rounding state interactions */
4  \phi(x) = -1 \text{ if } x = -1; 1 \text{ if } x = 1, \text{ otherwise} 
5  R_1(k) = \phi(k \text{ if } k \text{ odd integer}, k - 1 \text{ if } k \text{ even}) 
6  R_1(k) = \phi(k \text{ if } k \text{ odd integer}, k + 1 \text{ if } k \text{ even})

7  \text{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}

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

9  \text{procedure } \text{update}(x, y)
10  \text{if } (\text{weight}(x) > 0 \text{ and } \text{weight}(y) > 1) \text{ or } (\text{weight}(y) > 0 \text{ and } \text{weight}(x) > 1) \text{ then}
11     x' \leftarrow R_1(\frac{\text{value}(x) + \text{value}(y)}{2}) \text{ and } y' \leftarrow R_1(\frac{\text{value}(x) + \text{value}(y)}{2}) 
12  \text{else if } \text{weight}(x) \cdot \text{weight}(y) = 0 \text{ and } \text{value}(x) + \text{value}(y) > 0 \text{ then}
13     \text{if } \text{weight}(x) \neq 0 \text{ then } x' \leftarrow \text{Shift-to-Zero}(x) \text{ and } y' \leftarrow \text{Sign-to-Zero}(x) 
14  \text{else } y' \leftarrow \text{Shift-to-Zero}(y) \text{ and } x' \leftarrow \text{Sign-to-Zero}(y)
15  \text{else if } (x \in \{-1, +1\} \text{ and } \text{weight}(y) = 1 \text{ and } \text{sgn}(x) \neq \text{sgn}(y)) \text{ or }
16      (y \in \{-1, +1\} \text{ and } \text{weight}(x) = 1 \text{ and } \text{sgn}(y) \neq \text{sgn}(x)) \text{ then}
17     x' \leftarrow -0 \text{ and } y' \leftarrow +0 
18  \text{else}
19     x' \leftarrow \text{Shift-to-Zero}(x) \text{ and } y' \leftarrow \text{Shift-to-Zero}(y)
```
Checking correctness

Protocols can become complex, even for $B \geq R$:

Fast and Exact Majority in Population Protocols

Dan Alistarh
Microsoft Research
Rati Gelashvili*
MIT
Milan Vojnović
Microsoft Research

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

2 $sgn(x) = \begin{cases} 1 & \text{if } x \in \{+0, 1_d, \ldots, 1_1, 3, 5, \ldots, m\}; \\ -1 & \text{otherwise}. \end{cases}$

3 $value(x) = sgn(x) \cdot weight(x)$

/* Functions for rounding state interactions */

4 $\phi(x) = -1_1 \text{ if } x = -1_1; 1_1 \text{ if } x = 1_1; x, \text{ otherwise}$

5 $R_1(k) = \phi(k \text{ if } k \text{ odd integer, } k-1 \text{ if } k \text{ even})$

6 $R_1(k) = \phi(k \text{ if } k \text{ odd integer, } k+1 \text{ if } k \text{ even})$

7 $Shift-to-Zero(x) = \begin{cases} -1_{j+1} & \text{if } x = -1_j \text{ for some index } j < d \\ 1_{j+1} & \text{if } x = 1_j \text{ for some index } j < d \\ x & \text{otherwise}. \end{cases}$

8 $Sign-to-Zero(x) = \begin{cases} +0 & \text{if } sgn(x) > 0 \\ -0 & \text{otherwise}. \end{cases}$

9 procedure update$(x, y)$

10 if (weight$(x) > 0$ and weight$(y) > 1$) or (weight$(y) > 0$ and weight$(x) > 1$) then

11 \quad $x' \leftarrow R_1\left(\frac{value(x) + value(y)}{2}\right)$ and $y' \leftarrow R_1\left(\frac{value(x) + value(y)}{2}\right)$

12 else if weight$(x) \cdot$ weight$(y) = 0$ and value$(x) +$ value$(y) > 0$ then

13 \quad if weight$(x) \neq 0$ then $x' \leftarrow Shift-to-Zero(x)$ and $y' \leftarrow Sign-to-Zero(x)$

14 \quad else $y' \leftarrow Shift-to-Zero(y)$ and $x' \leftarrow Sign-to-Zero(y)$

15 else if $(x \in \{-1_d, +1_d\} \text{ and } weight(y) = 1 \text{ and } sgn(x) \neq sgn(y)) \text{ or }$

16 \quad (y \in \{-1_d, +1_d\} \text{ and } weight(x) = 1 \text{ and } sgn(y) \neq sgn(x))$ then

17 \quad $x' \leftarrow -0$ and $y' \leftarrow +0$

18 else

19 \quad $x' \leftarrow Shift-to-Zero(x)$ and $y' \leftarrow Shift-to-Zero(y)$

How can we verify correctness automatically?
Model checkers:

• **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)
Checking correctness—Early days

Model checkers:

- **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!*
Theorem provers:

- Verification with the interactive theorem prover Coq (Deng and Monin TASE’09)

Only for populations of fixed size!

Not automatic!

Challenge: verifying automatically all sizes
Theorem provers:

- Verification with the interactive theorem prover **Coq**
  
  (Deng and Monin TASE’09)

*Not automatic!*
Theorem provers:

- Verification with the interactive theorem prover **Coq**
  
  (Deng and Monin TASE’09)

**Challenge:** verifying automatically all sizes
It is decidable if a population protocol is well specified (i.e., if it computes some predicate).
Checking correctness—Decidability

Initial confs.

$C_0$

$C_1$

$C_2$

...
Checking correctness—Decidability

Effectively Presburger set

$C_0 \rightarrow 0$

$C_1 \rightarrow 1$

$C_2 \rightarrow 1$

Initial confs.

Bottom confs.
Checking correctness—Decidability

Effectively Presburger set

\[
\begin{align*}
C_0 & = 0 \\
C_1 & = 1 \\
C_2 & = 1 \\
\vdots & \\
\end{align*}
\]

Initial confs.

Bottom confs.

Eilenberg and Schützenberger ’69:
Semilinear set
→ Presburger
Checking correctness—Decidability

Effectively Presburger set

Initial confs.

\[ \begin{align*}
C_0 & \quad 0 \\
C_1 & \quad 0 \\
C_2 & \quad 1 \\
\cdots & \\
\end{align*} \]

Bottom confs.

Eilenberg and Schützenberger ’69:
Semilinear set
→ Presburger

Leroux ’11:
Effectively semilinear
→ effectively Presburger
Effectively Presburger set

Eilenberg and Schützenberger ’69:
Semilinear set
→ Presburger

Leroux ’11:
Effectively semilinear
→ effectively Presburger

Reduction to the VAS reachability problem between Presburger sets
Effectively Presburger set


c_0

\[0\]

c_1

\[1\]

c_2

\[1\]

…

Initial confs.

Bottom confs.

Eilenberg and Schützenberger '69:
Semilinear set
→ Presburger

Leroux '11:
Effectively semilinear
→ effectively Presburger

Reduction to the VAS reachability problem between Presburger sets
⇒ Reduction to the VAS reachability problem (VAS engineering)
Checking correctness—Decidability

Effectively Presburger set

Effectively Presburger set

Eilenberg and Schützenberger ’69:
Semilinear set
→ Presburger

Leroux ’11:
Effectively semilinear
→ effectively Presburger

Reduction to the VAS reachability problem between Presburger sets
⇒ Reduction to the VAS reachability problem (VAS engineering)
⇒ Decidable (Mayr ’81, Kosaraju ’83).
It is decidable if a population protocol computes a given predicate (Presburger formula).
E., Ganty, Leroux, Majumdar  Acta Inf.'17

It is decidable if a population protocol computes a given predicate (Presburger formula).

There is an algorithm that returns the predicate computed by a well-specified protocol.
VAS reachability is reducible to the well-specification problem for population protocols
VAS reachability is reducible to the well-specification problem for population protocols.

→ Well specification is EXSPACE-hard, and all known algorithms for it have hyper-ackermannian complexity.
A class $\mathcal{P}$ of protocols is complete if for every Presburger predicate $\varphi$ some protocol in $\mathcal{P}$ computes $\varphi$. 
A class $\mathcal{P}$ of protocols is complete if for every Presburger predicate $\varphi$ some protocol in $\mathcal{P}$ computes $\varphi$.

**Goal:** Find a complete class of protocols verifiable in reasonable time.
The class of strongly silent protocols is complete, and its verification problem is in DP.
Checking correctness—Feasibility

Intel Core i7-4810MQ CPU and 16 GB of RAM.

| Protocol               | Predicate                        | |Q| | |T| | Time[s] |
|------------------------|----------------------------------|---|---|---|---|
| Majority[1]            | $x \geq y$                       | 4 | 4 | 0.1 |
| Approx. Majority[2]    | Not well-specified               | 3 | 4 | 0.1 |
| Broadcast[3]           | $x_1 \lor \ldots \lor x_n$       | 2 | 1 | 0.1 |
| Threshold[4]           | $\sum_i \alpha_i x_i < c$        | 76| 2148| 2375.9 |
| Remainder[5]           | $\sum_i \alpha_i x_i \mod 70 = 1$ | 72| 2555| 3176.5 |
| Sick ninjas[6]         | $x \geq 50$                      | 51| 1275| 181.6 |
| Sick ninjas[7]         | $x \geq 325$                     | 326| 649| 3470.8 |
| Poly-log sick ninjas   | $x \geq 8 \cdot 10^4$            | 66| 244| 12.79 |

The class of strongly silent protocols is complete, and its verification problem is in DP.

Mission accomplished?
The class of strongly silent protocols is complete, and its verification problem is in DP.

Mission accomplished?

Not yet. For some predicates no strongly silent succinct protocols are known.
A class $\mathcal{P}$ of protocols is **complete and succinct** if for every Presburger predicate $\varphi$ some protocol in $\mathcal{P}$ with $\log(|\varphi|)$ states computes $\varphi$.

A class $\mathcal{P}$ of protocols is **complete and efficient** if for every Presburger predicate $\varphi$ some protocol in $\mathcal{P}$ computes $\varphi$ in $\mathcal{O}(n^2 \log n)$ time.
Are strongly silent protocols complete and succinct?
Are strongly silent protocols complete and succinct?

Are strongly silent protocols complete and efficient?
Are strongly silent protocols complete and succinct?

Are strongly silent protocols complete and efficient?

What is the lowest expected time for a complete class of protocols?
Are strongly silent protocols complete and succinct?

Are strongly silent protocols complete and efficient?

What is the lowest expected time for a complete class of protocols?

...and for a complete and succinct class?
Are strongly silent protocols complete and succinct?

Are strongly silent protocols complete and efficient?

What is the lowest expected time for a complete class of protocols?

...and for a complete and succinct class?

...and for a complete and efficient class?
Are strongly silent protocols complete and succinct? Open
Are strongly silent protocols complete and efficient? Open
What is the lowest expected time for a complete class of protocols? Open
...and for a complete and succinct class? Open
...and for a complete and efficient class? Open
Sensei III’s questions

What predicates can we compute?
How fast can we compute them?
How succinctly can we compute them?
How can I check correctness?
How can I check efficiency?

To conclude ...
Our approach:

• Most protocols are naturally designed in stages

• Construct these stages automatically

• Derive upper bounds on $\text{Inter}(n)$ from stages structure
Checking expected termination time

Blondin, E., Kucera CONCUR’18

\[ (B \lor R) \land \bigwedge_{q \not\in \{B,R\}} \neg q \]

\[ O(1) \]

\[ O(n^2 \log n) \]

\[ O(n^2 \log n) \]

\[ O(n^2 \log n) \]

\[ O(\exp(n)) \]

\[ O(1) \]

\[ O(n^2 \log n) \]

\[ O(n^2 \log n) \]

\[ O(1) \]

\[ O(n^2 \log n) \]

\[ O(n^2 \log n) \]

\[ O(n^2 \log n) \]

\[ O(n^2 \log n) \]
• Prototype implemented in `python` + Microsoft Z3
• Prototype implemented in \texttt{python} + Microsoft Z3

• Can report: \( O(1), O(n^2), O(n^2 \log n), O(n^3), O(\text{poly}(n)) \)
  or \( O(\exp(n)) \)
• Prototype implemented in \texttt{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 $\mathcal{O}(\text{exp}(n))$

• \textbf{Decidability of checking $\text{Inte}(n) \geq f(n)$?}
  \textit{Open}
Sensei III’s questions

What predicates can we compute?
How fast can we compute them?
How succinctly can we compute them?
How can I check correctness?
How can I check efficiency?

To conclude ...
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!
Population protocols are a great model to study fundamental questions of distributed computation:

- Power of anonymous computation
- Network-independent algorithms
- Role of leaders
- Emergent behaviour and its limits
...and of formal verification:

- Verification of stochastic parameterized systems (parameterization, liveness under fairness)
- Automatic synthesis of parameterized systems
Join the team!

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

• Start of the project: Sept. 1, 2018

• Start of employment: flexible, from Sept. 1, 2018 to about Sept. 1, 2019
THANK YOU!
THANK YOU!