The Power and Limits of Population Protocols

Michael Blondin

Université de Sherbrooke

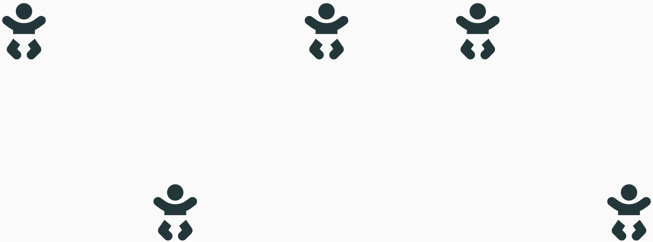
0 Determining babies’ needs

0 Determining babies' needs

0 Determining babies' needs

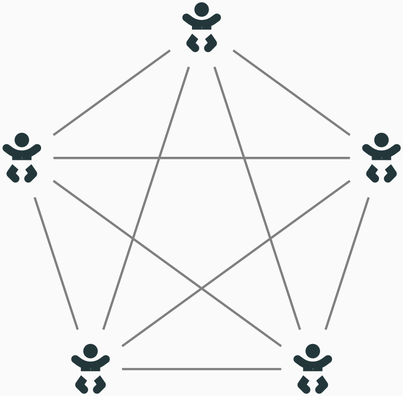
Determining babies' needs

0 Determining babies' needs

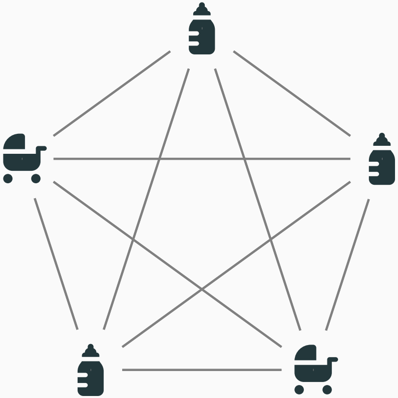


How to determine the majority?

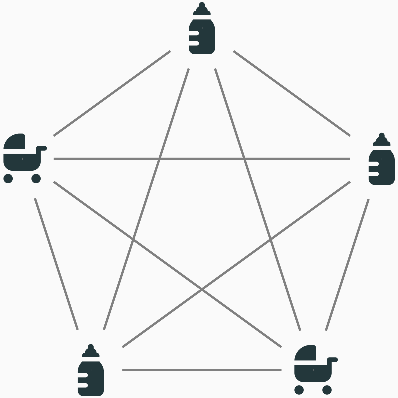
Population



Population



Population

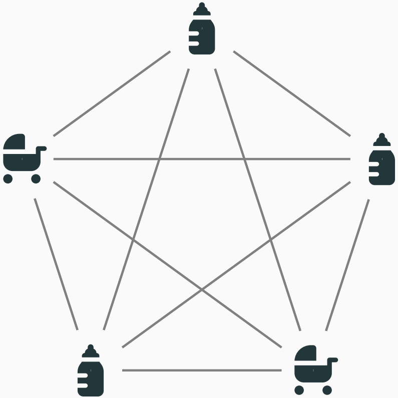


Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Majority vote protocol

Population

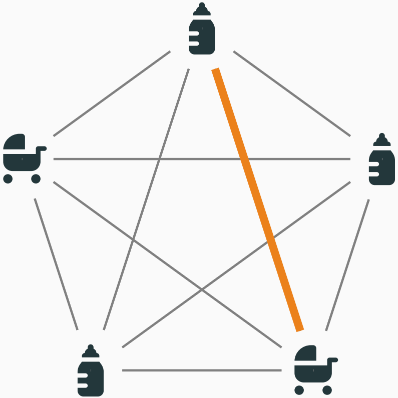


Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Population

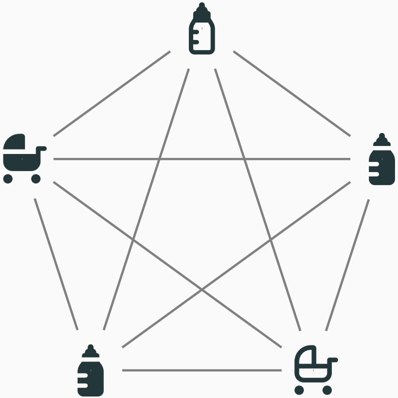


Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Population



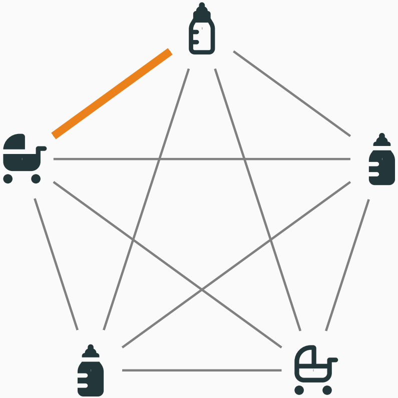
Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Majority vote protocol

Population



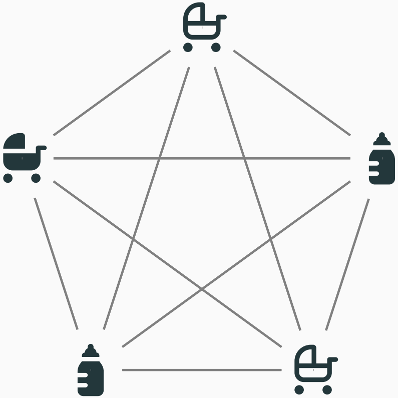
Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Majority vote protocol

Population

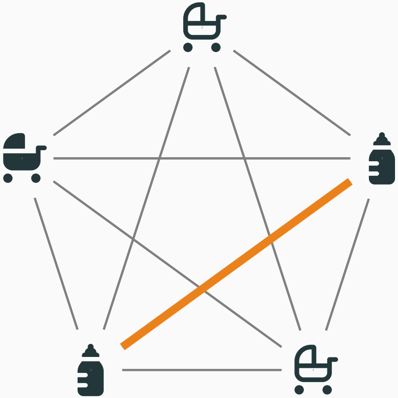


Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Population



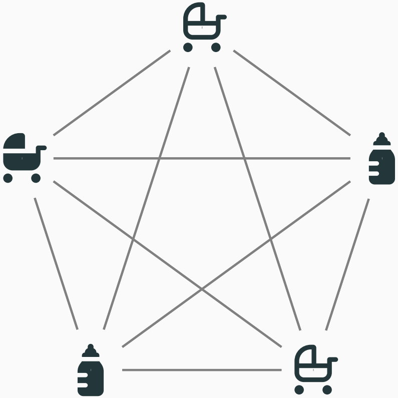
Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Majority vote protocol

Population



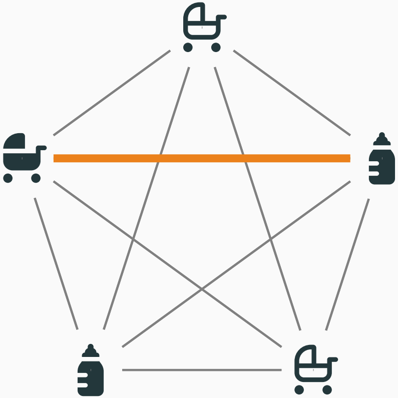
Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Majority vote protocol

Population

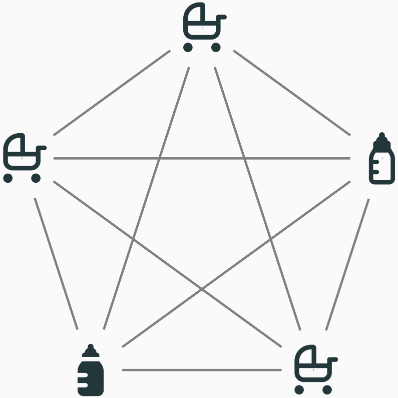


Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Population



Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Majority vote protocol

Population

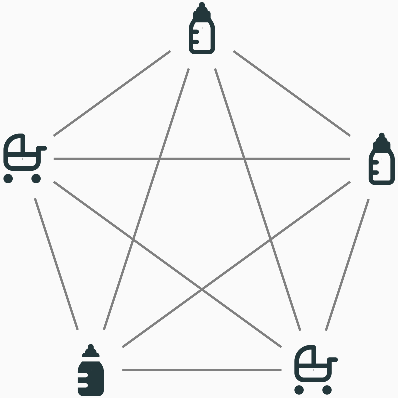
Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Majority vote protocol

Population



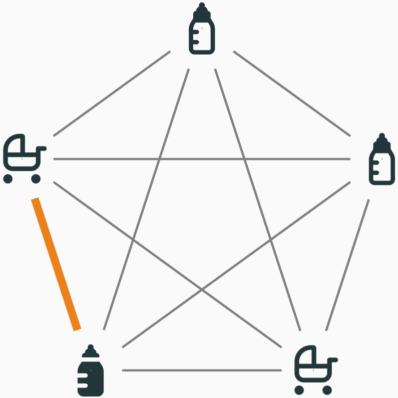
Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Majority vote protocol

Population



Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Population

Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Majority vote protocol

Population

Formal rules

Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

Population

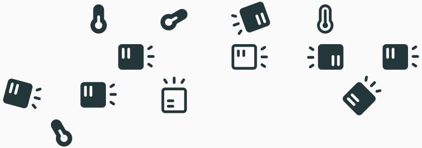
Formal rules

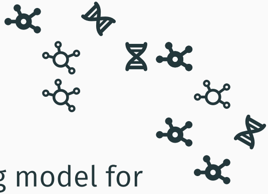
Informal rules

- Opposite active agents become passive
- Active agents convince passive agents

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

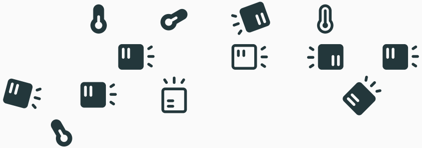
0 Overview

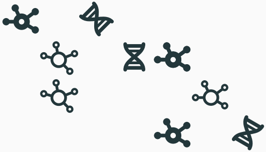




Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

- Introduced by Angluin, Aspnes, Diamadi, Fischer, Peralta (PODC'04)

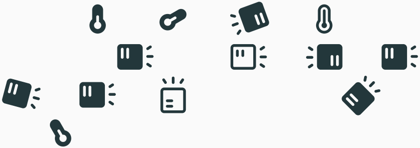


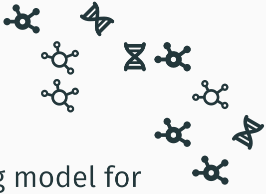


Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

- Introduced by Angluin, Aspnes, Diamadi, Fischer, Peralta (PODC'04)
- Dijkstra Prize of 2020

⓿ Overview

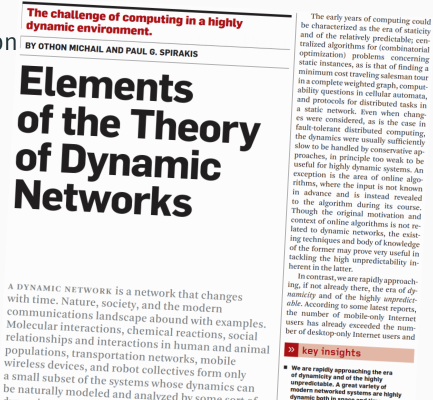




Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

- Introduced by Angluin, Aspn

- Dijkstra Prize of 2020



The challenge of computing in a highly dynamic environment.

BY OTHON MICHAIL AND PAUL G. SPIRAKIS

Elements of the Theory of Dynamic Networks

A DYNAMIC NETWORK is a network that changes with time. Nature, society, and the modern communications landscape abound with examples. Molecular interactions, chemical reactions, social relationships and interactions in human and animal populations, transportation networks, mobile wireless devices, and robot collectives form only a small subset of the systems whose dynamics can be naturally modeled and analyzed by some sort of

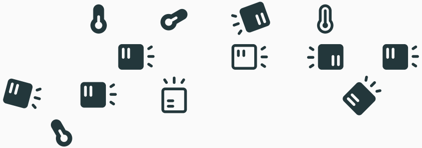
The early years of computing could be characterized as the era of staticity and of the relatively predictable; centralized algorithms for (combinatorial optimization) problems concerning static instances, as is that of finding a minimum cost traveling salesman tour in a complete weighted graph, computability questions in cellular automata, and protocols for distributed tasks in a static network. Even when changes were considered, as is the case in fault-tolerant distributed computing, the dynamics were usually sufficiently slow to be handled by conservative approaches, in principle too weak to be useful for highly dynamic systems. An exception is the area of online algorithms, where the input is not known in advance and is instead revealed to the algorithm during its course. Though the original motivation and context of online algorithms is not related to dynamic networks, the existing techniques and body of knowledge of the former may prove very useful in tackling the high unpredictability inherent in the latter.

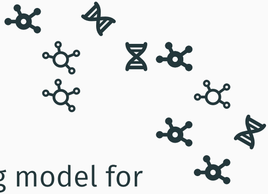
In contrast, we are rapidly approaching, if not already there, the era of *dynamicity* and of the highly *unpredictable*. According to some latest reports, the number of mobile-only Internet users has already exceeded the number of desktop-only Internet users and

» key insights

- We are rapidly approaching the era of dynamicity and of the highly unpredictable. A great variety of modern networked systems are highly dynamic both in space and

0 Overview



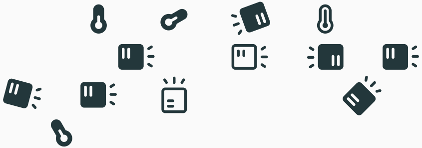


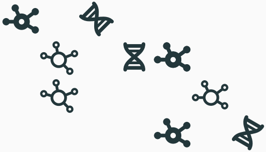
Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

This talk:

1. The model

0 Overview



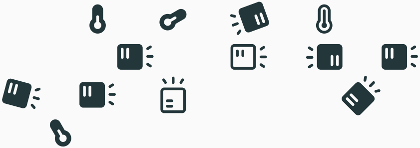


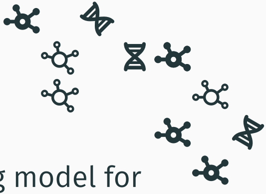
Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

This talk:

1. The model
2. State complexity

0 Overview

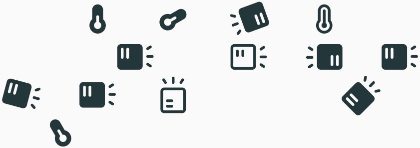


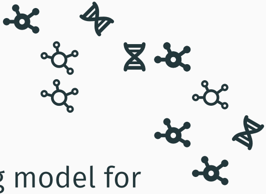


Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

This talk:

1. The model
2. State complexity
3. Extension of the model: ordered identifiers



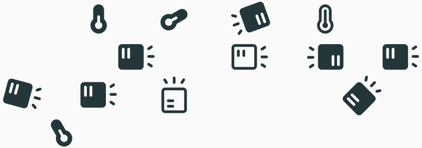


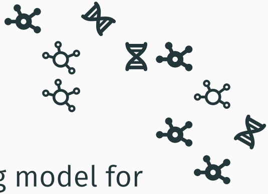
Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

This talk:

1. The model
2. State complexity
3. Extension of the model: ordered identifiers
4. Extension of the model: infinite alphabets

0 Overview

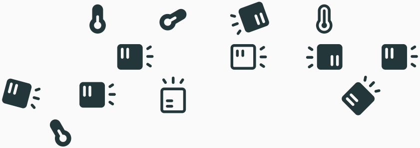


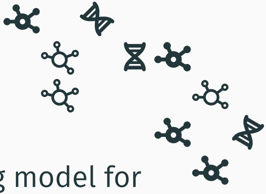


Population protocols: distributed computing model for massive networks of passively mobile finite-state agents

This talk:

1. The model
2. State complexity
3. Extension of the model: ordered identifiers
4. Extension of the model: infinite alphabets
5. Conclusion and open questions





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

This talk:

1. The model
2. State complexity
3. Extension of the model: ordered identifiers
4. Extension of the model: infinite alphabets
5. Conclusion and open questions

❶ Population protocols

The model

❶ Population protocols: formal model

- *States:* finite set Q
- *Input mapping:* $\iota \colon \Sigma \to Q$ with finite Σ
- *Output mapping:* $o \colon Q \to \Gamma$ with finite Γ
- *Transitions:* $\to \subseteq Q^2 \times Q^2$

$Q = \{$ , , ,  $\}$

❶ Population protocols: formal model

- *States:* finite set Q
- *Input mapping:* $\iota\colon \Sigma \to Q$ with finite Σ
- *Output mapping:* $o\colon Q \to \Gamma$ with finite Γ
- *Transitions:* $\to \subseteq Q^2 \times Q^2$

$$\Sigma = \{b, c\}$$

$\iota(b) =$ 

$\iota(c) =$ 

❶ Population protocols: formal model

- *States:* finite set Q
- *Input mapping:* $\iota \colon \Sigma \to Q$ with finite Σ
- *Output mapping:* $o \colon Q \to \Gamma$ with finite Γ
- *Transitions:* $\to \subseteq Q^2 \times Q^2$

$o($  $) = o($  $) = 1$

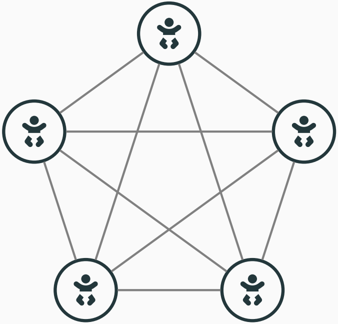
$o($  $) = o($  $) = 0$

❶ Population protocols: formal model

- *States:* finite set Q
- *Input mapping:* $\iota\colon \Sigma \to Q$ with finite Σ
- *Output mapping:* $o\colon Q \to \Gamma$ with finite Γ
- *Transitions:* $\rightarrow \subseteq Q^2 \times Q^2$

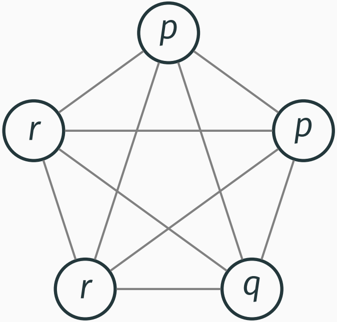
All agents can interact pairwise

(complete topology)



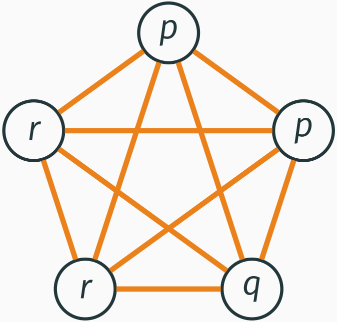
❶ Population protocols: interactions

$$\mathbb{P}[\text{fire } p, q \to p', q' \text{ in } C] = \begin{cases} \dfrac{2 \cdot C(p) \cdot C(q)}{n^2 - n} & \text{if } p \neq q \\[2ex] \dfrac{C(p) \cdot (C(p) - 1)}{n^2 - n} & \text{if } p = q \end{cases}$$



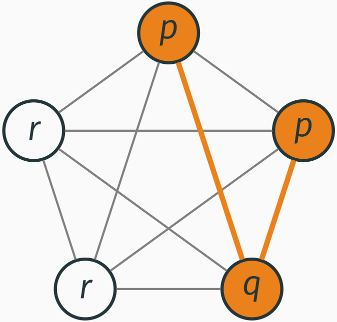
❶ Population protocols: interactions

$$\mathbb{P}[\text{fire } p, q \to p', q' \text{ in } C] = \begin{cases} \dfrac{2 \cdot C(p) \cdot C(q)}{n^2 - n} & \text{if } p \neq q \\ \dfrac{C(p) \cdot (C(p) - 1)}{n^2 - n} & \text{if } p = q \end{cases}$$



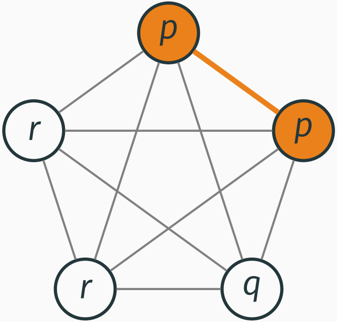
❶ Population protocols: interactions

$$\mathbb{P}[\text{fire } p, q \to p', q' \text{ in } C] = \begin{cases} \dfrac{2 \cdot C(p) \cdot C(q)}{n^2 - n} & \text{if } p \neq q \\[2ex] \dfrac{C(p) \cdot (C(p) - 1)}{n^2 - n} & \text{if } p = q \end{cases}$$



❶ Population protocols: interactions

$$\mathbb{P}[\text{fire } p, q \to p', q' \text{ in } C] = \begin{cases} \dfrac{2 \cdot C(p) \cdot C(q)}{n^2 - n} & \text{if } p \neq q \\[2ex] \dfrac{C(p) \cdot (C(p) - 1)}{n^2 - n} & \text{if } p = q \end{cases}$$



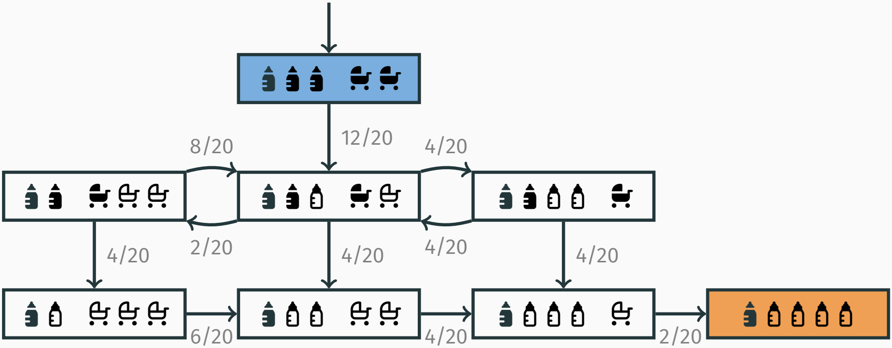
❶ Population protocols: interactions

$$\mathbb{P}[\text{fire } p, q \to p', q' \text{ in } C] = \begin{cases} \dfrac{2 \cdot C(p) \cdot C(q)}{n^2 - n} & \text{if } p \neq q \\[2ex] \dfrac{C(p) \cdot (C(p) - 1)}{n^2 - n} & \text{if } p = q \end{cases}$$

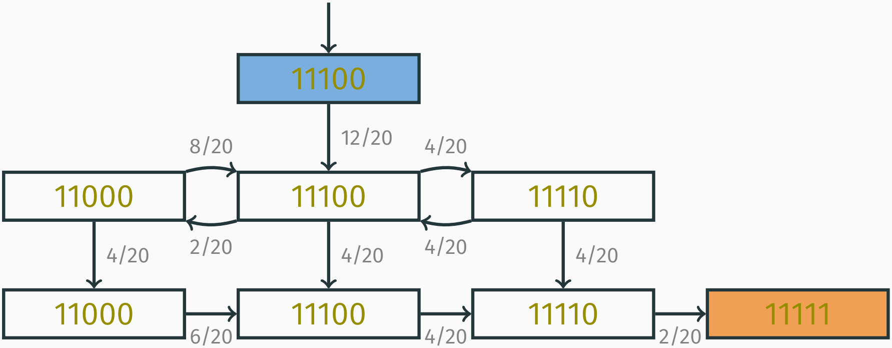
$$\mathbb{P}[C \to C'] = \sum_{t \text{ s.t. } C \xrightarrow{t} C'} \mathbb{P}[\text{fire } t \text{ in } C]$$

❶ Computability: example

Markov chain from configuration $C = \iota(bbbcc)$:



Markov chain from configuration $C = \iota(bbbcc)$:



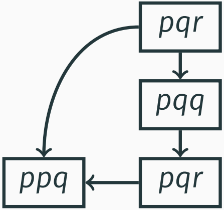
- A configuration C is ***stable*** if

$$C \to^* C' \text{ implies } o(C) = o(C')$$

1 Computability: definition

- A configuration C is ***stable*** if

$$C \to^* C' \text{ implies } o(C) = o(C')$$

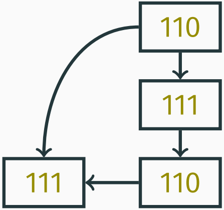


$$\begin{aligned} p, r &\to p, p \\ q, q &\to q, r \\ q, r &\to q, q \end{aligned}$$

$$\begin{aligned} o(p) &= 1 \\ o(q) &= 1 \\ o(r) &= 0 \end{aligned}$$

- A configuration C is ***stable*** if

$$C \to^* C' \text{ implies } o(C) = o(C')$$

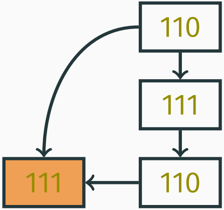


$$\begin{array}{rcl} p, r & \to & p, p \\ q, q & \to & q, r \\ q, r & \to & q, q \end{array}$$

$$\begin{array}{rcl} o(p) & = & 1 \\ o(q) & = & 1 \\ o(r) & = & 0 \end{array}$$

- A configuration C is ***stable*** if

$$C \rightarrow^* C' \text{ implies } o(C) = o(C')$$



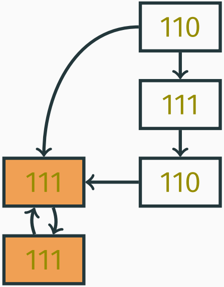
$$\begin{aligned} p, r &\rightarrow p, p \\ q, q &\rightarrow q, r \\ q, r &\rightarrow q, q \end{aligned}$$

$$\begin{aligned} o(p) &= 1 \\ o(q) &= 1 \\ o(r) &= 0 \end{aligned}$$

❶ Computability: definition

- A configuration C is ***stable*** if

$$C \to^* C' \text{ implies } o(C) = o(C')$$

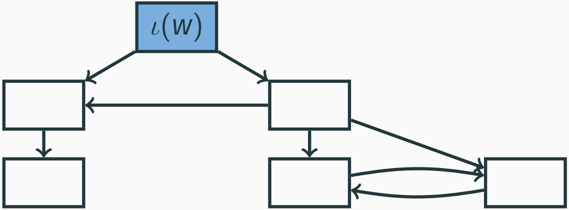


- A configuration C is ***stable*** if

$$C \rightarrow^* C' \text{ implies } o(C) = o(C')$$

- A protocol ***computes*** $f\colon \Sigma^{\oplus} \rightarrow \Gamma^{\oplus}$ if

$$\forall w \in \Sigma^{\oplus} \quad \forall \iota(w) \rightarrow^* C$$

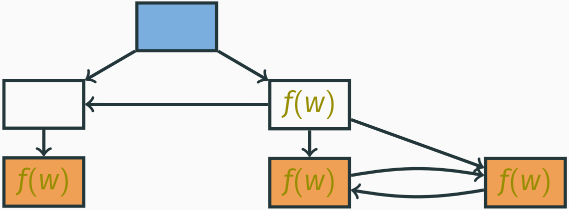


- A configuration C is ***stable*** if

$$C \rightarrow^* C' \text{ implies } o(C) = o(C')$$

- A protocol ***computes*** $f\colon \Sigma^\oplus \rightarrow \Gamma^\oplus$ if

$$\forall w \in \Sigma^\oplus \quad \forall \iota(w) \rightarrow^* C \quad \exists \text{ stable } D \quad C \rightarrow^* D \text{ and } o(D) = f(w)$$



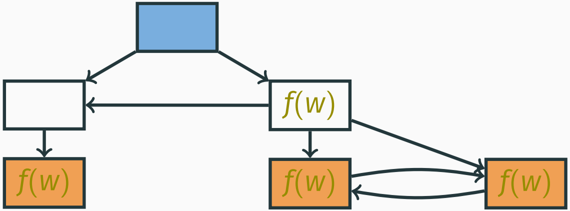
❶ Computability: definition

- A configuration C is ***stable*** if

$$C \rightarrow^* C' \text{ implies } o(C) = o(C')$$

- A protocol ***computes*** $f \colon \Sigma^\oplus \rightarrow \Gamma^\oplus$ if

$$\forall w \in \Sigma^\oplus \qquad \mathbb{P}\left[\iota(w) \rightarrow^* o^{-1}(f(w)) \rightarrow o^{-1}(f(w)) \rightarrow \cdots\right] = 1$$



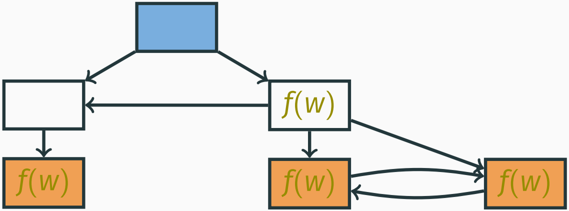
❶ Computability: definition

- A configuration C is ***stable*** if

$$C \rightarrow^* C' \text{ implies } o(C) = o(C')$$

- A protocol ***computes*** $f\colon \Sigma^{\oplus} \rightarrow \Gamma^{\oplus}$ if

$$\forall w \in \Sigma^{\oplus} \qquad \mathbb{P}\left[\iota(w) \models \mathsf{FG}\ o^{-1}(f(w))\right] = 1$$



- A configuration C is ***stable*** if

$$C \rightarrow^* C' \text{ implies } o(C) = o(C')$$

- A protocol ***computes*** $f \colon \Sigma^{\oplus} \rightarrow \Gamma^{\oplus}$ if

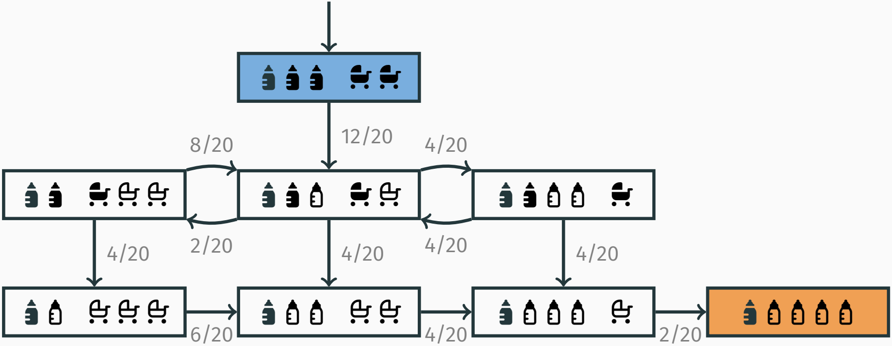
$$\forall w \in \Sigma^{\oplus} \qquad \mathbb{P}\left[\iota(w) \rightarrow^* o^{-1}(f(w)) \rightarrow o^{-1}(f(w)) \rightarrow \cdots\right] = 1$$

- A protocol ***decides*** $L \subseteq \Sigma^{\oplus}$ if it computes $f \colon \Sigma^{\oplus} \rightarrow \{0,1\}^{\oplus}$ s.t.

$$\begin{aligned} f(w) &= 1 \cdots 1 \quad \text{if } w \in L \\ f(w) &= 0 \cdots 0 \quad \text{if } w \notin L \end{aligned}$$

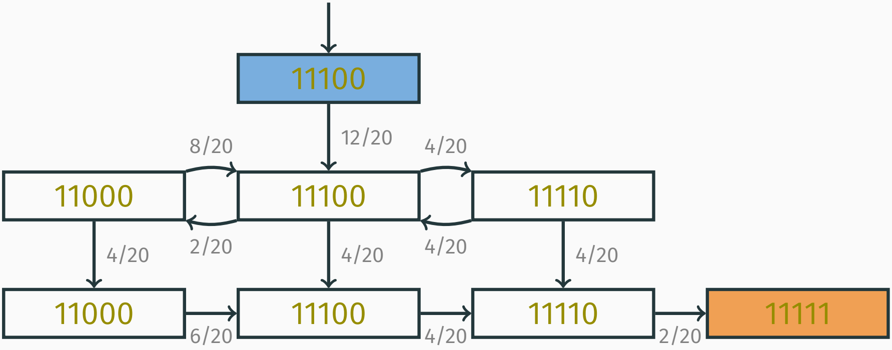
❶ Computability: example (again)

Markov chain from configuration $C = \iota(bbbcc)$:



❶ Computability: example (again)

Markov chain from configuration $C = \iota(bbbcc)$:



❶ Computability: majority vote

Proposition

This protocol decides $Maj := \{w \in \{b, c\}^{\oplus} : |w|_b > |w|_c\}$:

Proposition

This protocol decides $Maj := \{w \in \{b, c\}^{\oplus} : |w|_b > |w|_c\}$:

$$
\begin{array}{rcl}
\iota(b) & = & y_+ \\
\iota(c) & = & n_+
\end{array}
\qquad
\begin{array}{rcl}
y_+\, n_+ & \rightarrow & y_-\, n_- \\
y_+\, n_- & \rightarrow & y_+\, y_- \\
n_+\, y_- & \rightarrow & n_+\, n_- \\
y_-\, n_- & \rightarrow & n_-\, n_-
\end{array}
\qquad
\begin{array}{rcl}
o(y_+) & = & 1 \\
o(y_-) & = & 1 \\
o(n_+) & = & 0 \\
o(n_-) & = & 0
\end{array}
$$

❶ Computability: majority vote

Proposition

This protocol decides $Maj := \{w \in \{b, c\}^{\oplus} : |w|_b > |w|_c\}$:

$$\begin{array}{rcl} \iota(b) & = & y_+ \\ \iota(c) & = & n_+ \end{array} \qquad \begin{array}{rcl} y_+\, n_+ & \to & y_-\, n_- \\ y_+\, n_- & \to & y_+\, y_- \\ n_+\, y_- & \to & n_+\, n_- \\ y_-\, n_- & \to & n_-\, n_- \end{array} \qquad \begin{array}{rcl} o(y_+) & = & 1 \\ o(y_-) & = & 1 \\ o(n_+) & = & 0 \\ o(n_-) & = & 0 \end{array}$$

Proof

- Invariant: $\iota(w) \to^* C$ implies $C(y_+) - C(n_+) = |w|_b - |w|_c$

❶ Computability: majority vote

Proposition

This protocol decides $Maj := \{w \in \{b,c\}^{\oplus} : |w|_b > |w|_c\}$:

$$\begin{aligned} \iota(b) &= y_+ \\ \iota(c) &= n_+ \end{aligned} \qquad \begin{aligned} y_+\, n_+ &\to y_-\, n_- \\ y_+\, n_- &\to y_+\, y_- \\ n_+\, y_- &\to n_+\, n_- \\ y_-\, n_- &\to n_-\, n_- \end{aligned} \qquad \begin{aligned} o(y_+) &= 1 \\ o(y_-) &= 1 \\ o(n_+) &= 0 \\ o(n_-) &= 0 \end{aligned}$$

Proof

- Invariant: $\iota(w) \to^* C$ implies $C(y_+) - C(n_+) = |w|_b - |w|_c$
- Almost surely: losing side gets fully passive

Proposition

This protocol decides $Maj \coloneqq \{w \in \{b, c\}^{\oplus} : |w|_b > |w|_c\}$:

$$
\begin{array}{rcl}
\iota(b) & = & y_+ \\
\iota(c) & = & n_+
\end{array}
\qquad
\begin{array}{rcl}
y_+\, n_+ & \to & y_-\, n_- \\
y_+\, n_- & \to & y_+\, y_- \\
n_+\, y_- & \to & n_+\, n_- \\
y_-\, n_- & \to & n_-\, n_-
\end{array}
\qquad
\begin{array}{rcl}
o(y_+) & = & 1 \\
o(y_-) & = & 1 \\
o(n_+) & = & 0 \\
o(n_-) & = & 0
\end{array}
$$

Proof

- Invariant: $\iota(w) \to^* C$ implies $C(y_+) - C(n_+) = |w|_b - |w|_c$
- Almost surely: losing side gets fully passive
- Almost surely: winning side converts other side

❶ Expressiveness

Theorem **Angluin et al. (PODC'06)**

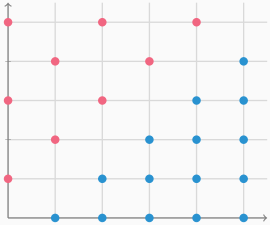
$L \subseteq \Sigma^{\oplus}$ is decided by a population protocol iff L is semilinear

Theorem **Angluin et al. (PODC'06)**

$L \subseteq \Sigma^{\oplus}$ is decided by a population protocol iff L is semilinear

Presburger arithmetic:

$\mathrm{FO}(\mathbb{N}, +, \leq)$

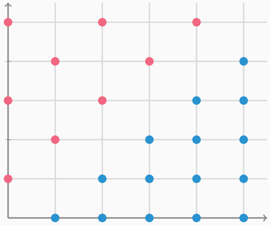


Theorem **Angluin et al. (PODC'06)**

$L \subseteq \Sigma^{\oplus}$ is decided by a population protocol iff L is semilinear

Same as commutative images of regular/context-free languages

Parikh (JACM '66)



Theorem **Angluin et al. (PODC'06)**

$L \subseteq \Sigma^{\oplus}$ is decided by a population protocol iff L is semilinear

Examples

- $|w|_b \geq |w|_c$
- $|w|_a \equiv |w|_b \pmod 2$
- $5|w|_a - 3|w|_b + |w|_c \geq 2 \land |w|_a \equiv 0 \pmod 7$

Theorem **Angluin et al. (PODC'06)**

$L \subseteq \Sigma^{\oplus}$ is decided by a population protocol iff L is semilinear

Examples

- $|w|_b \geq |w|_c$
- $|w|_a \equiv |w|_b \pmod 2$
- $5|w|_a - 3|w|_b + |w|_c \geq 2 \land |w|_a \equiv 0 \pmod 7$

General case

Bool. comb. of $\sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \geq m$ and $\sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \equiv m \pmod q$

Theorem — Angluin et al. (PODC'06)

$L \subseteq \Sigma^{\oplus}$ is decided by a population protocol iff L is semilinear

$\geq 2^{\|L\|}$ states

Examples

- $|w|_b \geq |w|_c$
- $|w|_a \equiv |w|_b \pmod 2$
- $5|w|_a - 3|w|_b + |w|_c \geq 2 \land |w|_a \equiv 0 \pmod 7$

General case

Bool. comb. of $\sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \geq m$ and $\sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \equiv m \pmod q$

❷ State complexity

How many states suffice/required?

❷ State complexity: majority

Let n := # agents and $\epsilon := |\#\text{ yes} - \#\text{ no}|/n$

Protocol	# states	Expected time	Err. prob.
Mertzios et al. (ICALP'14)	4	$\mathcal{O}(\frac{1}{\epsilon} \cdot n \log n)$	0

❷ State complexity: majority

Let $n := \#$ agents and $\epsilon := |\# \text{yes} - \# \text{no}|/n$

Protocol	# states	Expected time	Err. prob.
Mertzios et al. (ICALP'14)	4	$\mathcal{O}(\frac{1}{\epsilon} \cdot n \log n)$	0

$\mathcal{O}(n^2 \log n)$ if $|\#\text{yes} - \#\text{no}| = 1$

❷ State complexity: majority

Let $n :=$ # agents and $\epsilon := |\#\text{ yes} - \#\text{ no}|/n$

Protocol	# states	Expected time	Err. prob.
Mertzios et al. (ICALP'14)	4	$\mathcal{O}(\frac{1}{\epsilon} \cdot n \log n)$	0

Impossible to do better
(without errors)

❷ State complexity: majority

Let $n :=$ # agents and $\epsilon := |\#\text{ yes} - \#\text{ no}|/n$

Protocol	# states	Expected time	Err. prob.
Mertzios et al. (ICALP'14)	4	$\mathcal{O}(\frac{1}{\epsilon} \cdot n \log n)$	0
Angluin et al. (DISC'07)	3	$\mathcal{O}(n \log n)$	$\leq \frac{1}{e^{c \cdot \epsilon^2 n}}$

❷ State complexity: majority

Let $n :=$ # agents and $\epsilon := |\#\text{ yes} - \#\text{ no}|/n$

Protocol	**# states**	**Expected time**	**Err. prob.**
Mertzios et al. (ICALP'14)	4	$\mathcal{O}(\frac{1}{\epsilon} \cdot n \log n)$	0
Angluin et al. (DISC'07)	3	$\mathcal{O}(n \log n)$	$\leq \frac{1}{e^{c \cdot \epsilon^2 n}}$
Alistarh et al. (PODC'15)	$\frac{1}{\epsilon} + \mathcal{O}(\log \frac{1}{\epsilon} \cdot \log n)$	$\mathcal{O}(\log \frac{1}{\epsilon} \cdot n \log n)$	0

❷ State complexity: majority

Let $n :=$ # agents and $\epsilon := |\text{\# yes} - \text{\# no}|/n$

Protocol	**# states**	**Expected time**	**Err. prob.**
Mertzios et al. (ICALP'14)	4	$\mathcal{O}(\frac{1}{\epsilon} \cdot n \log n)$	0
Angluin et al. (DISC'07)	3	$\mathcal{O}(n \log n)$	$\leq \frac{1}{e^{c \cdot \epsilon^2 n}}$
Alistarh et al. (PODC'15)	$\frac{1}{\epsilon} + \mathcal{O}(\log \frac{1}{\epsilon} \cdot \log n)$	$\mathcal{O}(\log \frac{1}{\epsilon} \cdot n \log n)$	0
Alistarh et al. (SODA'18)	$\mathcal{O}(\log n)$	$\mathcal{O}(n \log^2 n)$	0
Berenbrink et al. (DISC'18)	$\mathcal{O}(\log n)$	$\mathcal{O}(n \log^{5/3} n)$	0
Ben-Nun et al. (PODC'20)	$\mathcal{O}(\log n)$	$\mathcal{O}(n \log^{3/2} n)$	0

❷ State complexity: majority

Let $n :=$ # agents and $\epsilon := |\#\text{ yes} - \#\text{ no}|/n$

Protocol	**# states**	**Expected time**	**Err. prob.**
Mertzios et al. (ICALP'14)	4	$\mathcal{O}(\frac{1}{\epsilon} \cdot n \log n)$	0
Angluin et al. (DISC'07)	3	$\mathcal{O}(n \log n)$	$\leq \frac{1}{e^{c \cdot \epsilon^2 n}}$
Alistarh et al. (PODC'15)	$\frac{1}{\epsilon} + \mathcal{O}(\log \frac{1}{\epsilon} \cdot \log n)$	$\mathcal{O}(\log \frac{1}{\epsilon} \cdot n \log n)$	0
Alistarh et al. (SODA'18)	$\mathcal{O}(\log n)$	$\mathcal{O}(n \log^2 n)$	0
Berenbrink et al. (DISC'18)	$\mathcal{O}(\log n)$	$\mathcal{O}(n \log^{5/3} n)$	0
Ben-Nun et al. (PODC'20)	$\mathcal{O}(\log n)$	$\mathcal{O}(n \log^{3/2} n)$	0
Doty et al. (FOCS'21)	$\mathcal{O}(\log n)$	$\mathcal{O}(n \log n)$	0

❷ State complexity: threshold

This protocol decides $\{w \in \Sigma^{\oplus} : |w|_a \geq m\}$:

- $Q = \{0, 1, \ldots, m\}$
- $\iota(a) = 1, \;\; \iota(\sigma) = 0$ otherwise
- $o(m) = 1, \;\; o(q) = 0$ otherwise
- Transitions:

$$p, q \rightarrow \min(p+q, m), p+q-\min(p+q, m)$$

$$m, q \rightarrow m, m$$

This protocol decides $\{w \in \Sigma^{\oplus} : |w|_a \geq 2^k\}$:

- $Q = \{0, 1, 2, 4, \ldots, 2^k\}$
- $\iota(a) = 1, \;\; \iota(\sigma) = 0$ otherwise
- $o(2^k) = 1, \;\; o(q) = 0$ otherwise
- Transitions:

$$\begin{aligned} 2^i, 2^i &\to 2^{i+1}, 0 && \text{if } i < k \\ 2^k, q &\to 2^k, 2^k \end{aligned}$$

❷ State complexity: threshold

This protocol decides $\{w \in \Sigma^{\oplus} : |w|_a \geq m\}$:

- $Q = \{0, 1, 2, 4, \ldots, 2^{\lfloor \log m \rfloor}, \heartsuit\}$
- $\iota(a) = 1, \;\; \iota(\sigma) = 0$ otherwise
- $o(\heartsuit) = 1, \;\; o(q) = 0$ otherwise
- Transitions:

$$
\begin{aligned}
2^i, 2^i &\to 2^{i+1}, 0 && \text{if } i < k \\
2^{i+1}, 0 &\to 2^i, 2^i && \text{if } i < k - 1 \\
\{2^i : i \in bits(m)\} &\to \heartsuit, \ldots, \heartsuit && \\
\heartsuit, q &\to \heartsuit, \heartsuit &&
\end{aligned}
$$

❷ State complexity: threshold

This protocol decides $\{w \in \Sigma^{\oplus} : |w|_a \geq m\}$:

- $Q = \{0, 1, 2, 4, \ldots, 2^{\lfloor \log m \rfloor}, \heartsuit\}$
- $\iota(a) = 1, \quad \iota(\sigma) = 0$ otherwise
- $o(\heartsuit) = 1, \quad o(q) = 0$ otherwise
- Transitions:

$$\begin{aligned} 2^i, 2^i &\to 2^{i+1}, 0 && \text{if } i < k \\ 2^{i+1}, 0 &\to 2^i, 2^i && \text{if } i < k-1 \\ \{2^i : i \in bits(m)\} &\to \heartsuit, \ldots, \heartsuit \\ \heartsuit, q &\to \heartsuit, \heartsuit \end{aligned}$$

Emulated with $O(\log m)$ two-way transitions/states

Proposition **B., Esparza, Jaax (STACS'18)**

If $\mathcal{P}_1, \mathcal{P}_2, \ldots$ decide $|w|_a \geq m$, then $\overset{\infty}{\exists} m$ with $|\mathcal{P}_m| \geq (\log m)^{1/4}$

❷ State complexity: threshold

Proposition — B., Esparza, Jaax (STACS'18)

If $\mathcal{P}_1, \mathcal{P}_2, \ldots$ decide $|w|_a \geq m$, then $\overset{\infty}{\exists} m$ with $|\mathcal{P}_m| \geq (\log m)^{1/4}$

Theorem — Czerner (DISC'24)

$\exists m_1 < m_2 < \cdots \in \mathbb{N}$ s.t. $|w|_a \geq m_n$ is decided by a family of protocols with $\mathcal{O}(\log \log m_n)$ states

❷ State complexity: threshold

Proposition — **B., Esparza, Jaax (STACS'18)**

If $\mathcal{P}_1, \mathcal{P}_2, \ldots$ decide $|w|_a \geq m$, then $\overset{\infty}{\exists} m$ with $|\mathcal{P}_m| \geq (\log m)^{1/4}$

Theorem — **Czerner (DISC'24)**

$\exists m_1 < m_2 < \cdots \in \mathbb{N}$ s.t. $|w|_a \geq m_n$ is decided by a family of protocols with $\mathcal{O}(\log \log m_n)$ states

Theorem — **Czerner, Esparza, Leroux (Distributed Comput.'23)**

$\forall m_1 < m_2 < \cdots \in \mathbb{N}$, any family of protocols for $|w|_a \geq m_n$ has $\Omega(\log \log m_n)$ states

❷ State complexity: threshold

	All thresholds	Family of thresholds
Upper bound	$\mathcal{O}(\log m)$	$\mathcal{O}(\log \log m_n)$
Lower bound	$\Omega(\log^{1/4-\epsilon} m)$	$\Omega(\log \log m_n)$

❷ State complexity: threshold

	All thresholds	Family of thresholds
Upper bound	$\mathcal{O}(\log m)$	$\mathcal{O}(\log \log m_n)$
Lower bound	$\Omega(\log^{1/4-\epsilon} m)$	$\Omega(\log \log m_n)$

Holds even if ***leaders*** are allowed [+ Leroux (PODC'22)]

$$w \mapsto C \odot \iota(w)$$

❷ State complexity: linear inequalities

This protocol decides $L \coloneqq \{w \in \Sigma^{\oplus} : \sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \geq m\}$:

- $Q = \{\text{♛}, \text{♙}\} \times [-B..B] \times \{0, 1\}$ with $B \coloneqq \max(k_a, k_b, \ldots, m)$
- $\iota(\sigma) = (\text{♛}, k_\sigma, k_\sigma \geq m)$
- $o(s, i, x) = x$
- Transitions:

$$(\text{♛}, i, x), (s, j, y) \to (\text{♛}, c(i+j), x), (\text{♙}, i + j - c(i+j), y)$$
$$(\text{♛}, i, x), (\text{♙}, j, y) \to (\text{♛}, i, x \geq m), (\text{♙}, j, x \geq m)$$

where $c(v) \coloneqq \max(\min(v, B), -B)$

❷ State complexity: linear inequalities

This protocol decides $L \coloneqq \{w \in \Sigma^{\oplus} : \sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \geq m\}$:

- $Q = \{\text{♛}, \text{♙}\} \times [-B..B] \times \{0, 1\}$ with $B \coloneqq \max(k_a, k_b, \ldots, m)$
- $\iota(\sigma) = (\text{♛}, k_\sigma, k_\sigma \geq m)$
- $o(s, i, x) = x$
- Transitions:

$$(\text{♛}, i, x), (s, j, y) \rightarrow (\text{♛}, c(i + j), x), (\text{♙}, i + j - c(i + j), y)$$
$$(\text{♛}, i, x), (\text{♙}, j, y) \rightarrow (\text{♛}, i, x \geq m), (\text{♙}, j, x \geq m)$$

where $c(v) \coloneqq \max(\min(v, B), -B)$

❷ State complexity: linear inequalities

This protocol decides $L \coloneqq \{w \in \Sigma^{\oplus} : \sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \geq m\}$:

- $Q = \{\text{♛}, \text{♙}\} \times [-B..B] \times \{0, 1\}$ with $B \coloneqq \max(k_a, k_b, \ldots, m)$
- $\iota(\sigma) = (\text{♛}, k_\sigma, k_\sigma \geq m)$
- $o(s, i, x) = x$
- Transitions:

$$(\text{♛}, i, x), (s, j, y) \rightarrow (\text{♛}, c(i+j), x), (\text{♙}, i+j-c(i+j), y)$$
$$(\text{♛}, i, x), (\text{♙}, j, y) \rightarrow (\text{♛}, i, x \geq m), (\text{♙}, j, x \geq m)$$

where $c(v) \coloneqq \max(\min(v, B), -B)$

❷ State complexity: linear inequalities

This protocol decides $L \coloneqq \{w \in \Sigma^{\oplus} : \sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \geq m\}$:

- $Q = \{♛, ♙\} \times [-B..B] \times \{0, 1\}$ with $B \coloneqq \max(k_a, k_b, \ldots, m)$
- $\iota(\sigma) = (♛, k_\sigma, k_\sigma \geq m)$
- $o(s, i, x) = x$
- Transitions:

$$(♛, i, x), (s, j, y) \rightarrow (♛, c(i+j), x), (♙, i+j-c(i+j), y)$$
$$(♛, i, x), (♙, j, y) \rightarrow (♛, i, x \geq m), (♙, j, x \geq m)$$

where $c(v) \coloneqq \max(\min(v, B), -B)$

❷ State complexity: linear inequalities

This protocol decides $L := \{w \in \Sigma^{\oplus} : \sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \geq m\}$:

- $Q = \{\text{♛}, \text{♙}\} \times [-B..B] \times \{0, 1\}$ with $B := \max(k_a, k_b, \ldots, m)$
- $\iota(\sigma) = (\text{♛}, k_\sigma, k_\sigma \geq m)$
- $o(s, i, x) = x$
- Transitions:

$$(\text{♛}, i, x), (s, j, y) \to (\text{♛}, c(i+j), x), (\text{♙}, i + j - c(i+j), y)$$
$$(\text{♛}, i, x), (\text{♙}, j, y) \to (\text{♛}, i, x \geq m), (\text{♙}, j, x \geq m)$$

where $c(v) := \max(\min(v, B), -B)$

❷ State complexity: linear inequalities

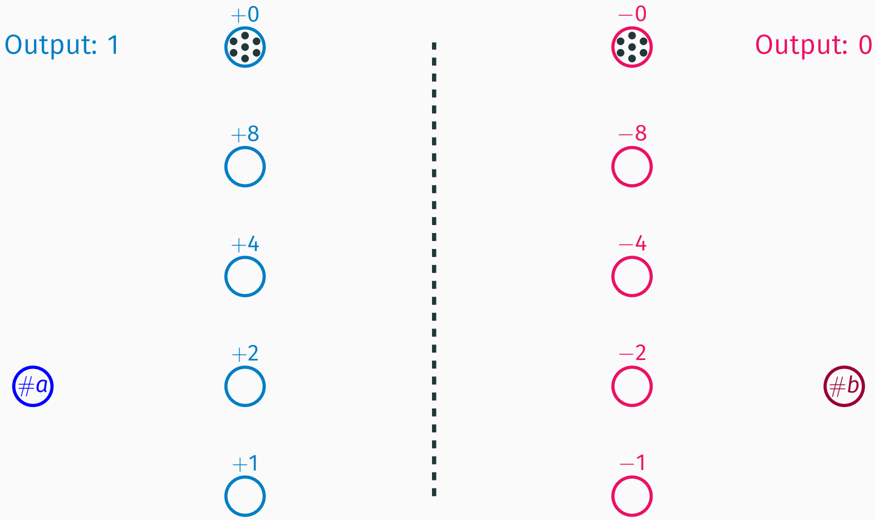
This protocol decides $L \coloneqq \{w \in \Sigma^{\oplus} : \sum_{\sigma \in \Sigma} k_\sigma |w|_\sigma \geq m\}$:

- $Q = \{\text{♛}, \text{♙}\} \times [-B..B] \times \{0, 1\}$ with $B \coloneqq \max(k_a, k_b, \ldots, m)$
- $\iota(\sigma) = (\text{♛}, k_\sigma, k_\sigma \geq m)$
- $o(s, i, x) = x$
- Transitions:

$$(\text{♛}, i, x), (s, j, y) \to (\text{♛}, c(i+j), x), (\text{♙}, i+j-c(i+j), y)$$
$$(\text{♛}, i, x), (\text{♙}, j, y) \to (\text{♛}, i, x \geq m), (\text{♙}, j, x \geq m)$$

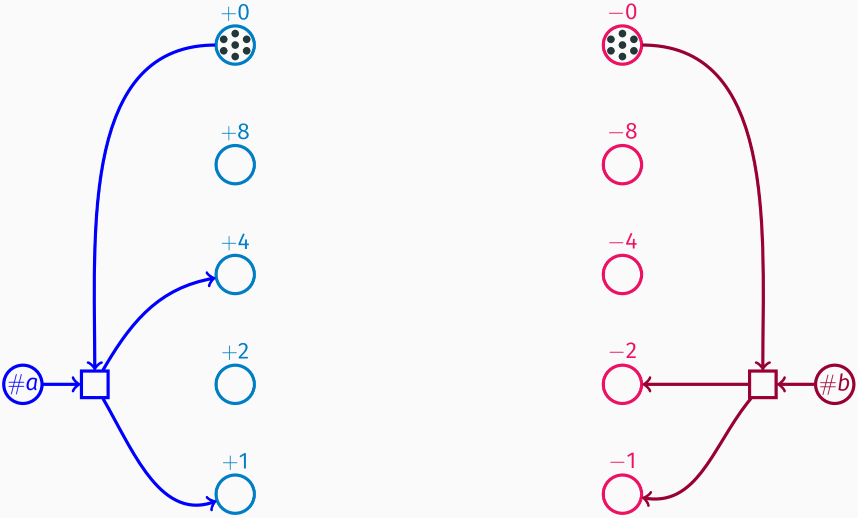
where $c(v) \coloneqq \max(\min(v, B), -B)$

❷ State complexity: linear inequalities

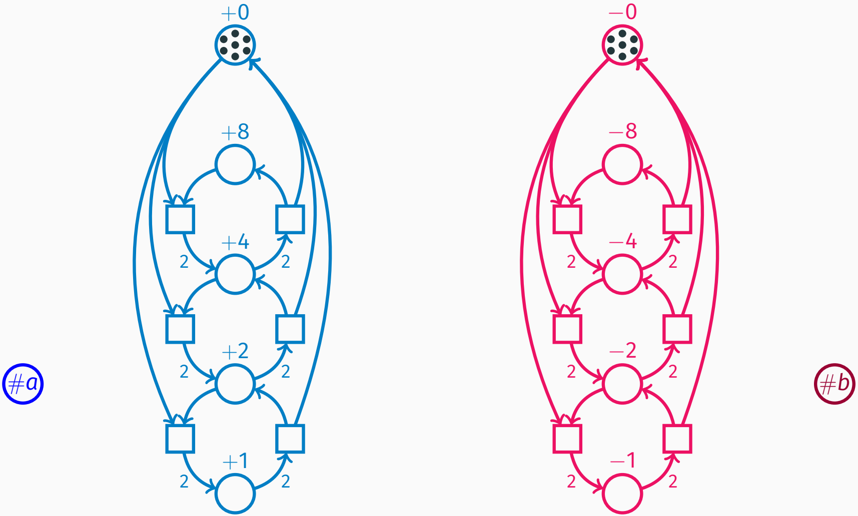


Protocol for $5|w|_a - 3|w|_b > 0$

❷ State complexity: linear inequalities

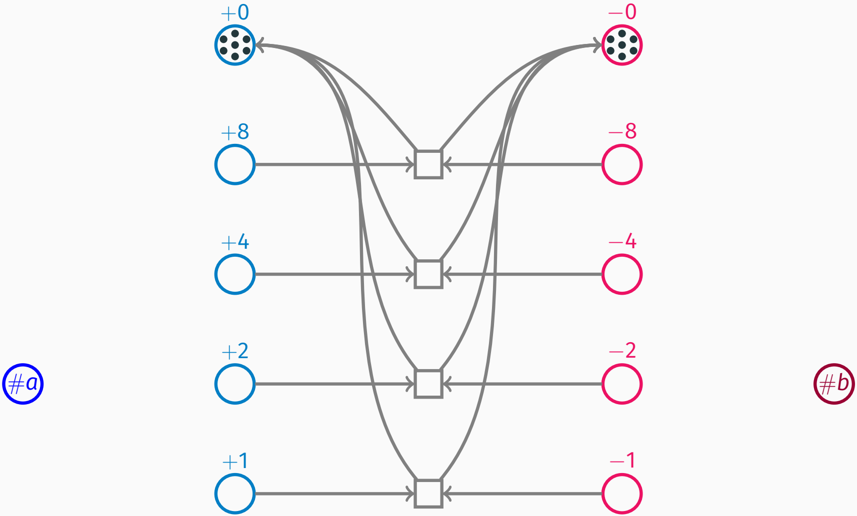


Protocol for $5|w|_a - 3|w|_b > 0$



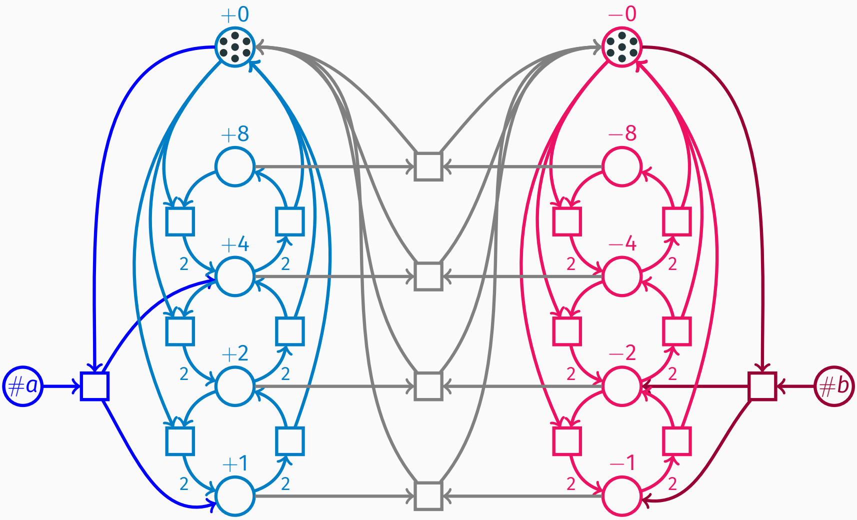
Protocol for $5|w|_a - 3|w|_b > 0$

❷ State complexity: linear inequalities



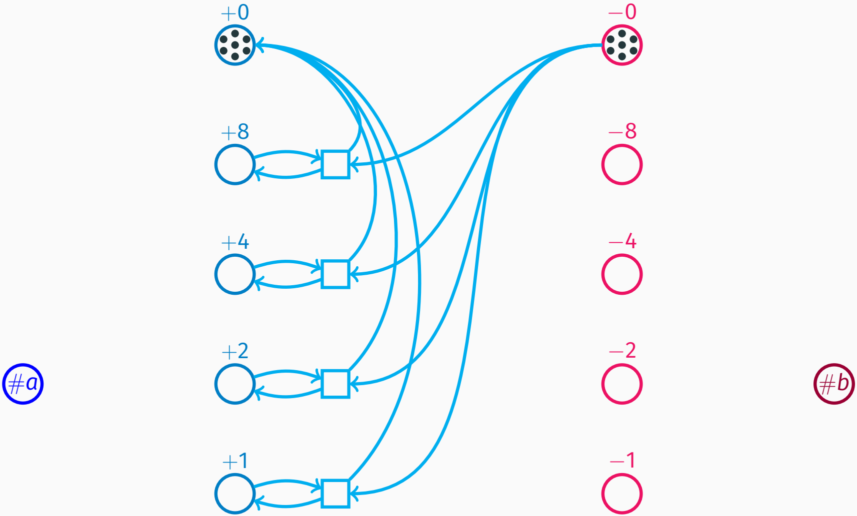
Protocol for $5|w|_a - 3|w|_b > 0$

❷ State complexity: linear inequalities



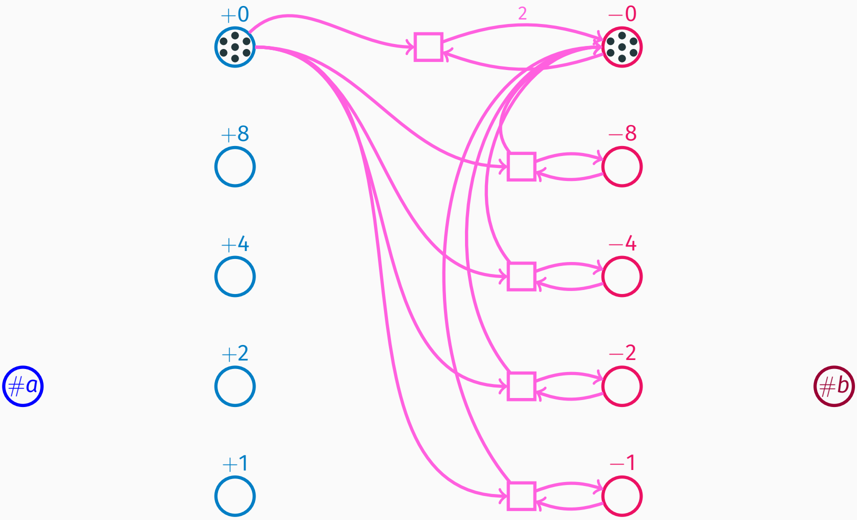
Protocol for $5|w|_a - 3|w|_b > 0$

❷ State complexity: linear inequalities



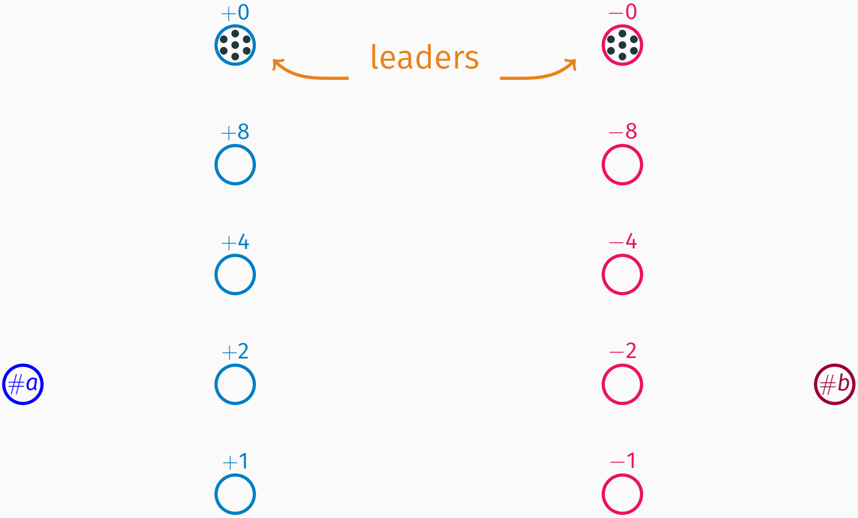
Protocol for $5|w|_a - 3|w|_b > 0$

❷ State complexity: linear inequalities



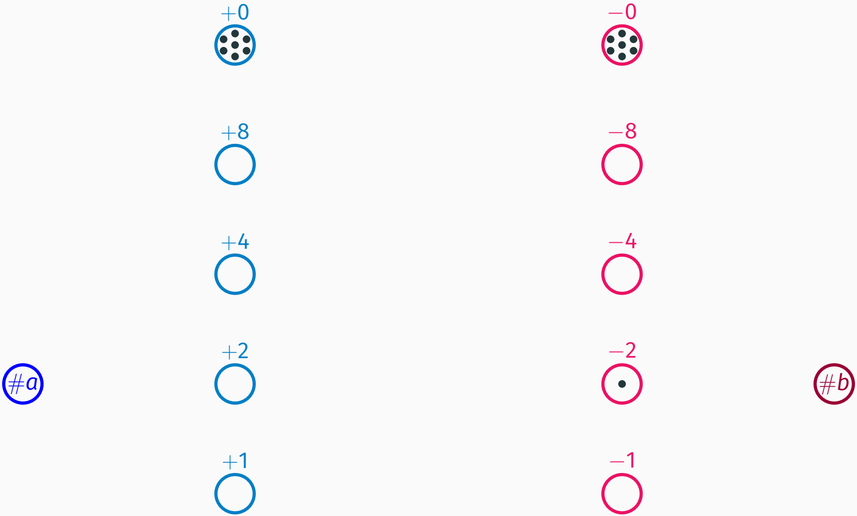
Protocol for $5|w|_a - 3|w|_b > 0$

❷ State complexity: linear inequalities



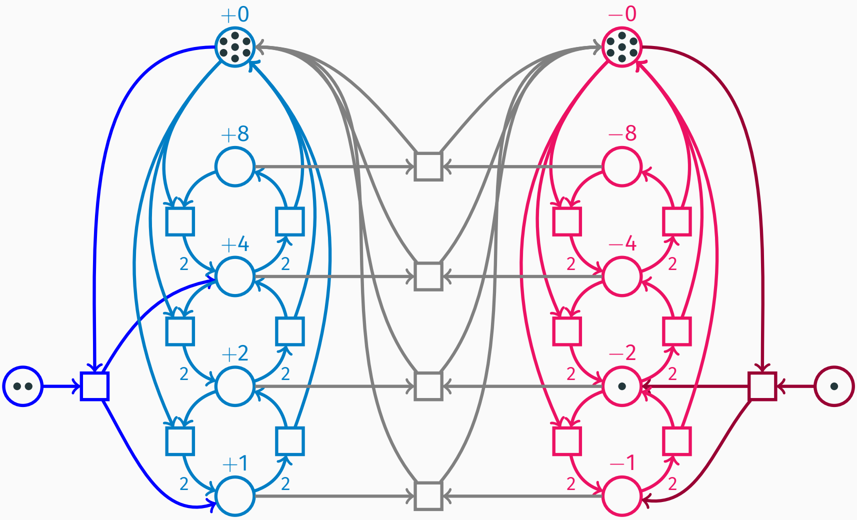
Protocol for $5|w|_a - 3|w|_b > 0$

❷ State complexity: linear inequalities

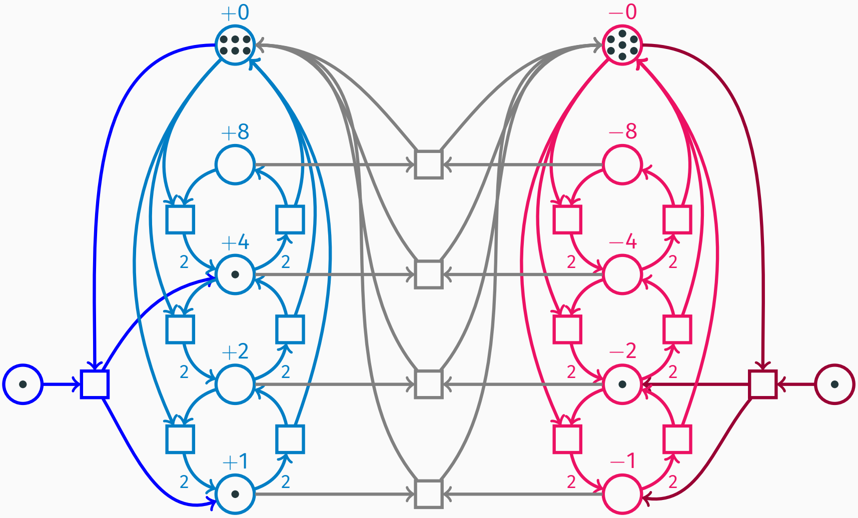


Protocol for $5|w|_a - 3|w|_b > 2$

❷ State complexity: linear inequalities

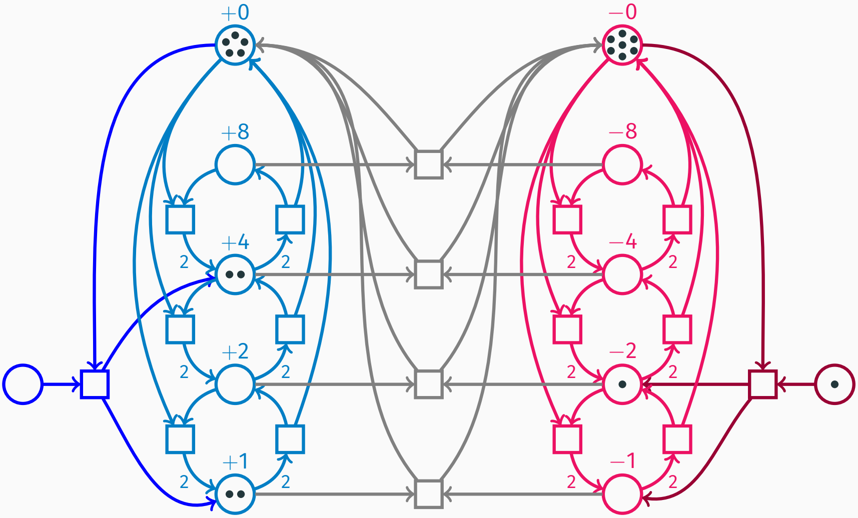


Protocol for $5|w|_a - 3|w|_b > 2$



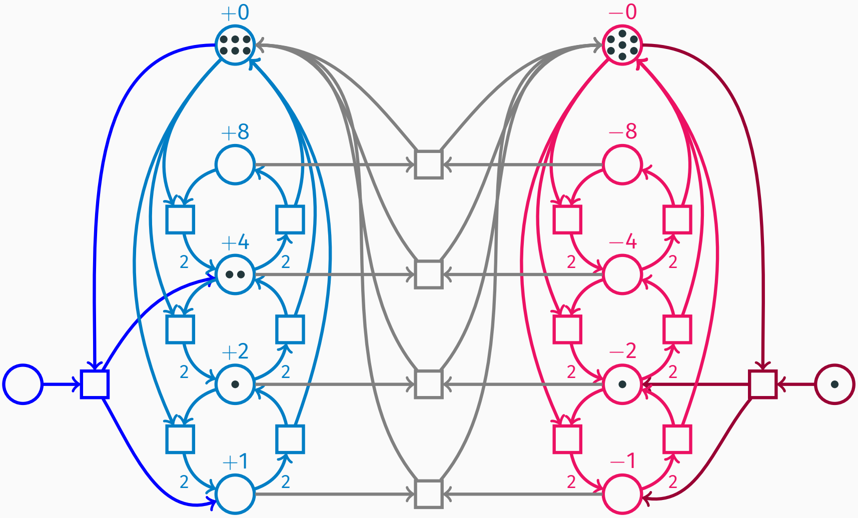
Protocol for $5|w|_a - 3|w|_b > 2$

❷ State complexity: linear inequalities



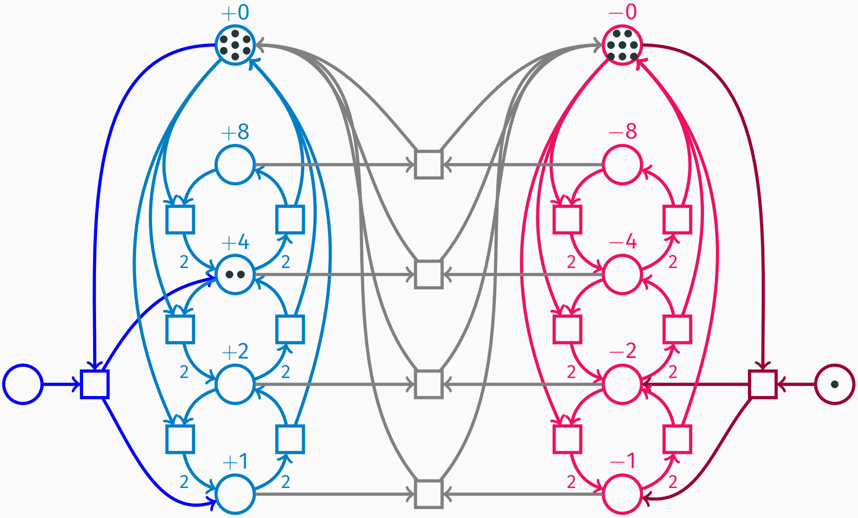
Protocol for $5|w|_a - 3|w|_b > 2$

❷ State complexity: linear inequalities



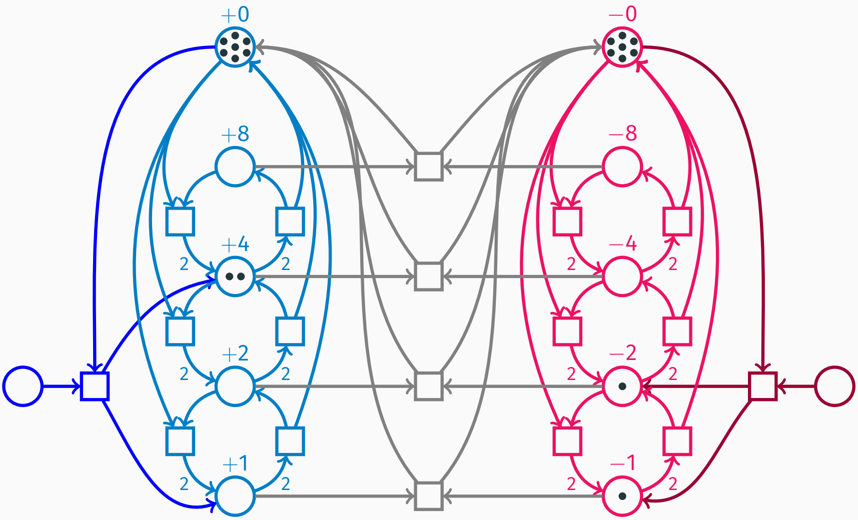
Protocol for $5|w|_a - 3|w|_b > 2$

❷ State complexity: linear inequalities



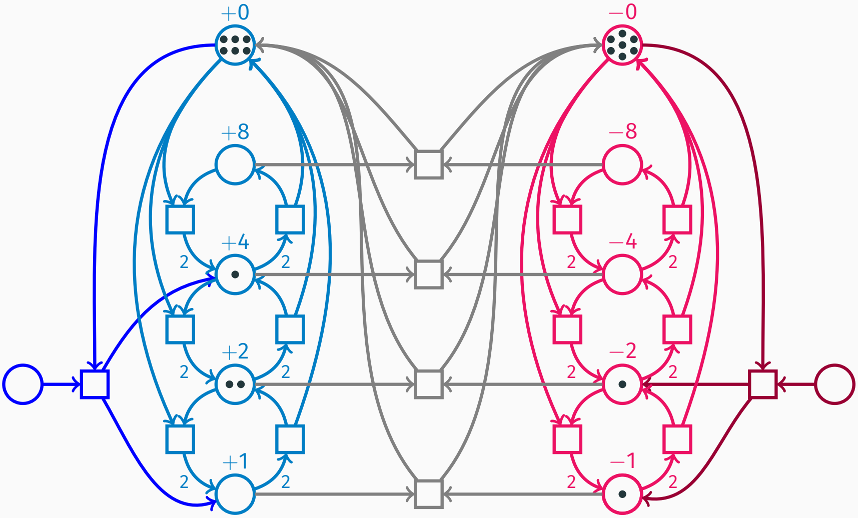
Protocol for $5|w|_a - 3|w|_b > 2$

❷ State complexity: linear inequalities



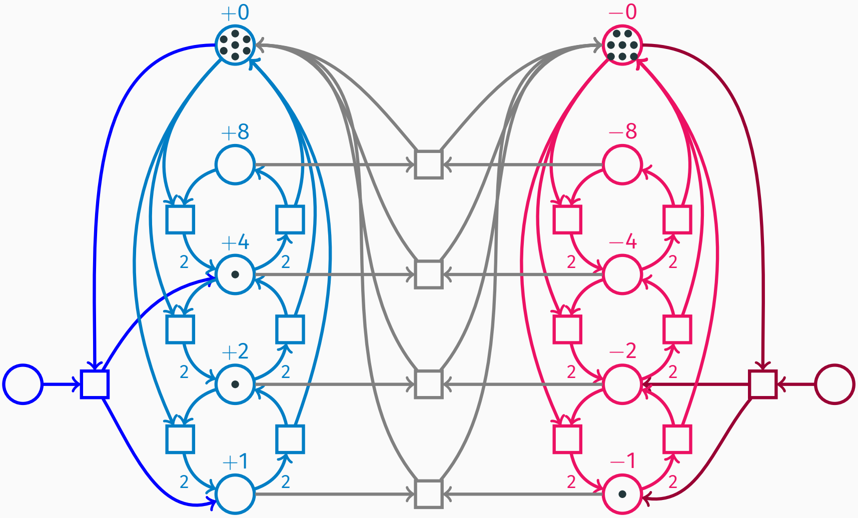
Protocol for $5|w|_a - 3|w|_b > 2$

❷ State complexity: linear inequalities



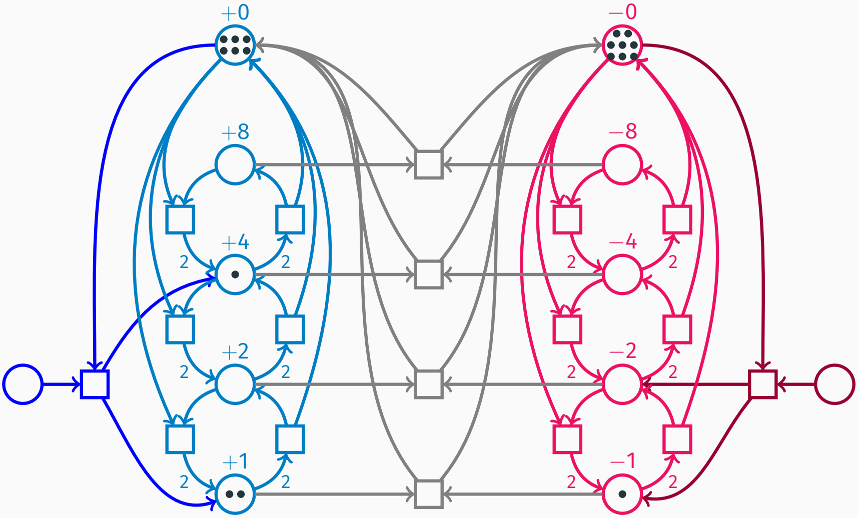
Protocol for $5|w|_a - 3|w|_b > 2$

❷ State complexity: linear inequalities

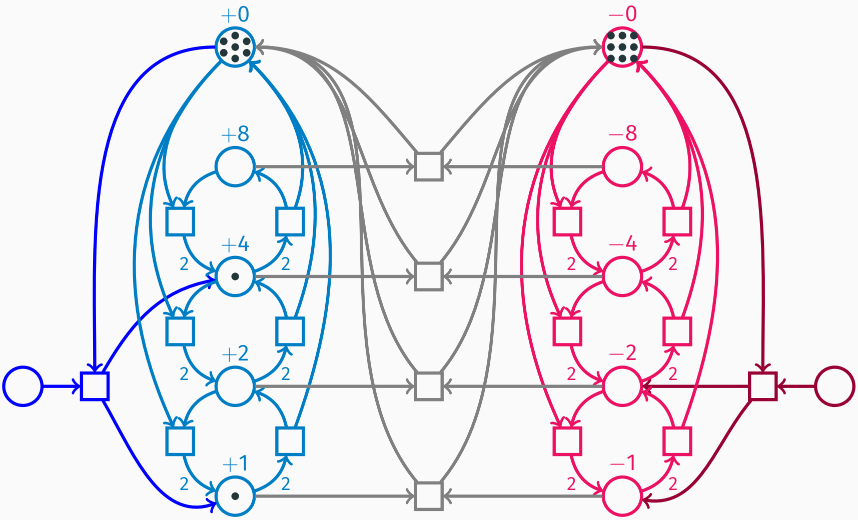


Protocol for $5|w|_a - 3|w|_b > 2$

❷ State complexity: linear inequalities



Protocol for $5|w|_a - 3|w|_b > 2$



Protocol for $5|w|_a - 3|w|_b > 2$

Theorem **B., Esparza, Genest, Helfrich, Jaax (STACS'20)**

- $\mathcal{O}(\|L\|)$ leaders suffice
- Leaders can be removed with a polynomial blow-up

Theorem B., Esparza, Genest, Helfrich, Jaax (STACS'20)

- $\mathcal{O}(\|L\|)$ leaders suffice
- Leaders can be removed with a polynomial blow-up

Corollary

A linear inequality language L can be decided with $\mathcal{O}(\mathrm{poly}\|L\|)$ states

Theorem B., Esparza, Genest, Helfrich, Jaax (STACS'20)

- $\mathcal{O}(\|L\|)$ leaders suffice
- Leaders can be removed with a polynomial blow-up

Corollary

A linear inequality language L can be decided with $\mathcal{O}(\mathrm{poly}\|L\|)$ states

Extends to modulo constraints
and Boolean combinations

❷ State complexity: general case

Protocol	# states	Expected time
Angluin et al. (PODC'04)	$\mathcal{O}(2^{\|L\|})$	$\mathcal{O}(\|L\| \cdot n^2 \log n)$
B. et al. (STACS'20)	$\mathcal{O}(\mathrm{poly}\|L\|)$	$\Omega(2^n)$

❷ State complexity: general case

Protocol	# states	Expected time
Angluin et al. (PODC'04)	$\mathcal{O}(2^{\|L\|})$	$\mathcal{O}(\|L\| \cdot n^2 \log n)$
B. et al. (STACS'20)	$\mathcal{O}(\mathrm{poly}\|L\|)$	$\Omega(2^n)$
Czerner et al. (SAND'22)	$\mathcal{O}(\mathrm{poly}\|L\|)$	$\mathcal{O}(\mathrm{poly}\|L\| \cdot n^2)$

❷ State complexity: general case

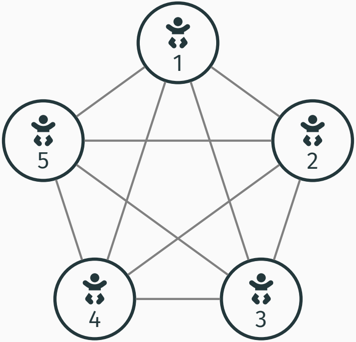
Protocol	# states	Expected time
Angluin et al. (PODC'04)	$\mathcal{O}(2^{\|L\|})$	$\mathcal{O}(\|L\| \cdot n^2 \log n)$
B. et al. (STACS'20)	$\mathcal{O}(\mathrm{poly}\|L\|)$	$\Omega(2^n)$
Czerner et al. (SAND'22)	$\mathcal{O}(\cancel{\mathrm{poly}}\|L\|)$	$\mathcal{O}(\mathrm{poly}\|L\| \cdot n^2)$

if $n \in \Omega(\|L\|)$

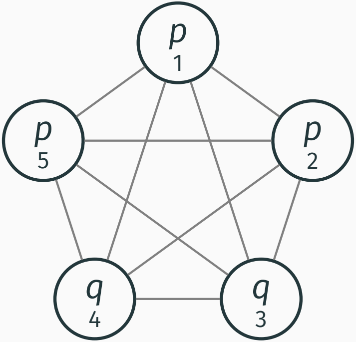
❸ Protocols with order

What happens if agents have (unique) identifiers?

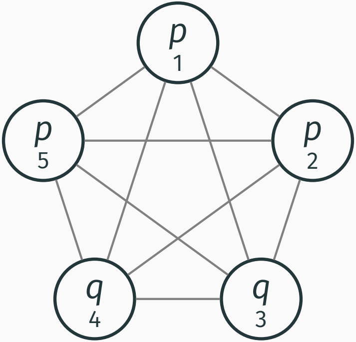
❸ Agents with ordered IDs



❸ Agents with ordered IDs



❸ Agents with ordered IDs

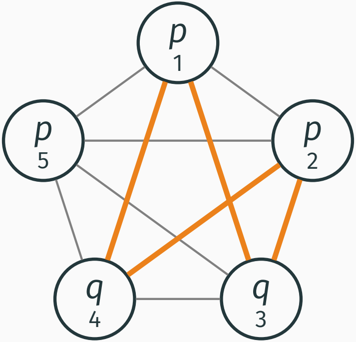


$$p, q \xrightarrow{<} p', q'$$

$$p, q \xrightarrow{>} p', q'$$

$$p, q \xrightarrow{+1} p', q'$$

❸ Agents with ordered IDs

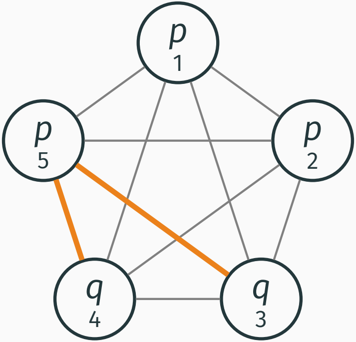


$$p, q \xrightarrow{<} p', q'$$

$$p, q \xrightarrow{>} p', q'$$

$$p, q \xrightarrow{+1} p', q'$$

❸ Agents with ordered IDs

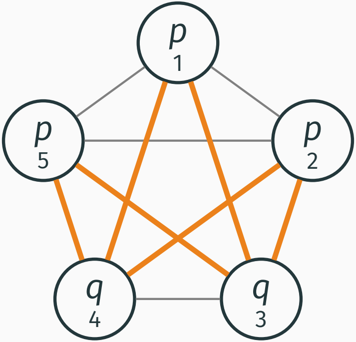


$$p, q \xrightarrow{<} p', q'$$

$$p, q \xrightarrow{>} p', q'$$

$$p, q \xrightarrow{+1} p', q'$$

❸ Agents with ordered IDs

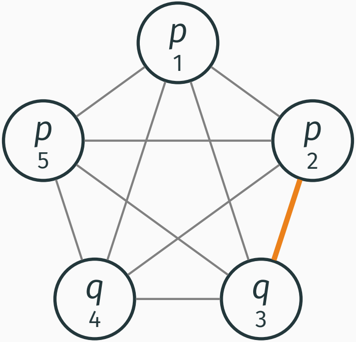


$$p, q \xrightarrow{<} p', q'$$

$$p, q \xrightarrow{>} p', q'$$

$$p, q \xrightarrow{+1} p', q'$$

❸ Agents with ordered IDs



$$p, q \xrightarrow{<} p', q'$$

$$p, q \xrightarrow{>} p', q'$$

$$p, q \xrightarrow{+1} p', q'$$

❸ Agents with ordered IDs: example

This protocol decides $\{w \in \{a, b\}^+ : w \text{ starts and ends with } a\}$:

- $Q = \left\{s_x : s \in \{\LEFTcircle, \Circle, \CIRCLE, \RIGHTcircle\}, x \in \{a, b\}\right\} \times \{\textcolor{magenta}{0}, \textcolor{blue}{1}\}$
- $\iota(a) = \textcolor{blue}{\CIRCLE_a}$, $\iota(b) = \textcolor{magenta}{\CIRCLE_b}$
- $O(\textcolor{blue}{s_x}) = \textcolor{blue}{1}$, $O(\textcolor{magenta}{s_x}) = \textcolor{magenta}{0}$
- Transitions:

$$\begin{array}{l|l|l}
\CIRCLE_x \; s_y \xrightarrow{\leq} \LEFTcircle_x \; s_y & \LEFTcircle_a \; \RIGHTcircle_a \xrightarrow{\leq} \textcolor{blue}{\LEFTcircle_a} \; \RIGHTcircle_a & \textcolor{blue}{\LEFTcircle_x} \; q \xrightarrow{\leq} \textcolor{blue}{\LEFTcircle_x} \; \textcolor{blue}{q} \\
s_x \; \CIRCLE_y \xrightarrow{\leq} s_x \; \RIGHTcircle_y & \LEFTcircle_x \; \RIGHTcircle_b \xrightarrow{\leq} \textcolor{magenta}{\LEFTcircle_x} \; \RIGHTcircle_b & \textcolor{magenta}{\LEFTcircle_x} \; q \xrightarrow{\leq} \textcolor{magenta}{\LEFTcircle_x} \; \textcolor{magenta}{q} \\
s_x \; \LEFTcircle_y \xrightarrow{\leq} s_x \; \Circle_y & \LEFTcircle_b \; \RIGHTcircle_y \xrightarrow{\leq} \textcolor{magenta}{\LEFTcircle_b} \; \RIGHTcircle_y & \\
\RIGHTcircle_x \; s_y \xrightarrow{\leq} \Circle_x \; s_y & &
\end{array}$$

❸ Agents with ordered IDs: example

A protocol is *immediate-observation* if a single agent can change in an interaction, i.e. transitions are of this form:

$$p, q \longrightarrow p', q$$
$$p, q \longrightarrow p, q'$$

- Transitions:

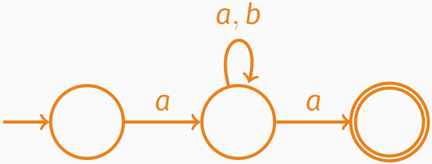
$$\begin{array}{l|l|l}
\text{●}_x\ s_y \overset{\leq}{\to} \text{◐}_x\ s_y & \text{◐}_a\ \text{◑}_a \overset{\leq}{\to} {\color{blue}\text{◐}_a}\ \text{◑}_a & {\color{blue}\text{◐}_x}\ q \overset{\leq}{\to} {\color{blue}\text{◐}_x\ q} \\
s_x\ \text{●}_y \overset{\leq}{\to} s_x\ \text{◑}_y & \text{◐}_x\ \text{◑}_b \overset{\leq}{\to} {\color{magenta}\text{◐}_x}\ \text{◑}_b & {\color{magenta}\text{◐}_x}\ q \overset{\leq}{\to} {\color{magenta}\text{◐}_x\ q} \\
s_x\ \text{◐}_y \overset{\leq}{\to} s_x\ \text{○}_y & \text{◐}_b\ \text{◑}_y \overset{\leq}{\to} {\color{magenta}\text{◐}_b}\ \text{◑}_y & \\
\text{◑}_x\ s_y \overset{\leq}{\to} \text{○}_x\ s_y & &
\end{array}$$

❸ Agents with ordered IDs: example

A protocol is *immediate-observation* if a single agent can change in an interaction, i.e. transitions are of this form:

$$p, q \longrightarrow p', q$$
$$p, q \longrightarrow p, q'$$

Unambiguous automaton for our imm.-obs. $<$-protocol:



Theorem — B., Cadilhac, Courchesne, Guillou, Mascle, Vialard (ICALP'26)

$L \subseteq \Sigma^+$ is decided by an immediate-observation $<$-protocol iff it is unambiguous star-free

❸ Agents with ordered IDs: expressiveness

Theorem B., Cadilhac, Courchesne, Guillou, Mascle, Vialard (ICALP'26)

$L \subseteq \Sigma^+$ is decided by an immediate-observation $<$-protocol iff it is unambiguous star-free

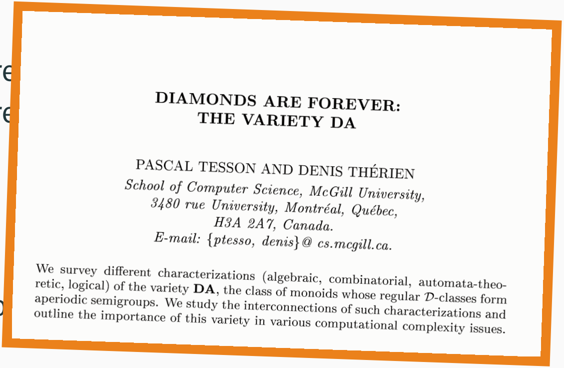
- partially-ordered unambiguous automata
- partially-ordered two-way deterministic automata
- $\mathrm{LTL}[\mathbf{F}^{-1}, \mathbf{F}]$
- $\mathrm{FO}^2[<]$
- $\Sigma_2[<] \cap \Pi_2[<]$
- languages recognized by monoids from the variety DA

❸ Agents with ordered IDs: expressiveness

Theorem **B., Cadilhac, Courchesne, Guillou, Mascle, Vialard (ICALP'26)**

$L \subseteq \Sigma^+$ is decided by an immediate-observation $<$-protocol iff it is unambiguous star-free

- partially-ordere
- partially-ordere
- LTL[$\mathbf{F}^{-1}, \mathbf{F}$]
- $\mathrm{FO}^2[<]$
- $\Sigma_2[<] \cap \Pi_2[<]$
- languages reco


DIAMONDS ARE FOREVER:
THE VARIETY DA

PASCAL TESSON AND DENIS THÉRIEN
School of Computer Science, McGill University,
3480 rue University, Montréal, Québec,
H3A 2A7, Canada.
E-mail: {ptesso, denis}@ cs.mcgill.ca.

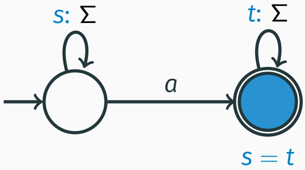
We survey different characterizations (algebraic, combinatorial, automata-theoretic, logical) of the variety **DA**, the class of monoids whose regular $\mathcal{D}$-classes form aperiodic semigroups. We study the interconnections of such characterizations and outline the importance of this variety in various computational complexity issues.

❸ Agents with ordered IDs: expressiveness

Median language: $L := \bigcup_{n \in \mathbb{N}} \Sigma^n a \Sigma^n$

Median language: $L := \bigcup_{n \in \mathbb{N}} \Sigma^n a \Sigma^n$

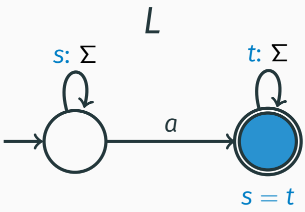
Partially-ordered Parikh automata:

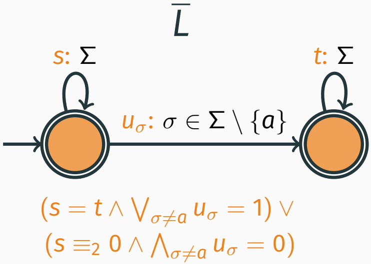


❸ Agents with ordered IDs: expressiveness

Median language: $L := \bigcup_{n \in \mathbb{N}} \Sigma^n a \Sigma^n$

Partially-ordered Parikh automata:

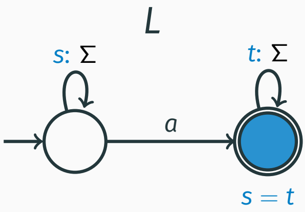


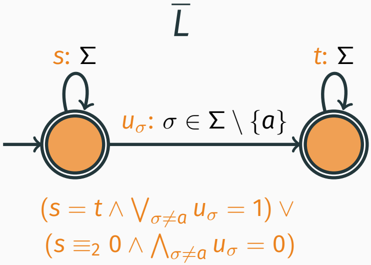


❸ Agents with ordered IDs: expressiveness

Median language: $L := \bigcup_{n \in \mathbb{N}} \Sigma^n a \Sigma^n$

Partially-ordered Parikh automata:





Theorem **B. et al. (ICALP'26)**

If L and $\overline{L}$ are both recognized by p.o. Parikh automata, then L is decided by a $<$-protocol

❸ Agents with ordered IDs: expressiveness

Proposition **B. et al. (ICALP'26)**

- Any $L \in$ NSPACE(n) is decidable by an imm. obs. $+1$-protocol
- Any L decided by a $\{+1, <\}$-protocol belongs to NSPACE(n)

Proposition **B. et al. (ICALP'26)**

- Any $L \in \mathrm{NSPACE}(n)$ is decidable by an imm. obs. $+1$-protocol
- Any L decided by a $\{+1, <\}$-protocol belongs to $\mathrm{NSPACE}(n)$

Corollary

$+1$-protocols $\equiv$ imm. obs. $+1$-protocols $\equiv \mathrm{NSPACE}(n)$

Community protocol $:=$

$<$-protocol $+$ constant number of ID registers per agent

Community protocol $:=$

$<$-protocol $+$ constant number of ID registers per agent

Theorem **Guerraoui, Ruppert (ICALP'09)**

Community protocols $\equiv$ NSPACE$(n \log n)$. Moreover, they can resist to Byzantine failures of a constant number of agents

❹ Protocols with data

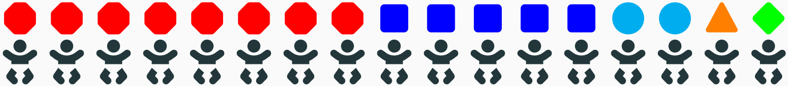
What happens if the input alphabet is infinite?

Majority vote with unboundedly many options

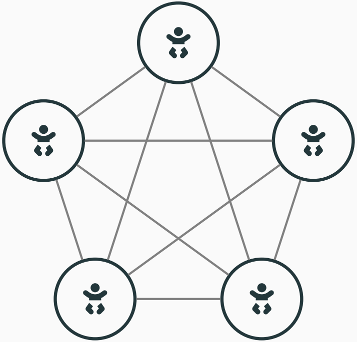
$\mathbb{D} = \{$  $, \ldots \}$

Majority vote with unboundedly many options

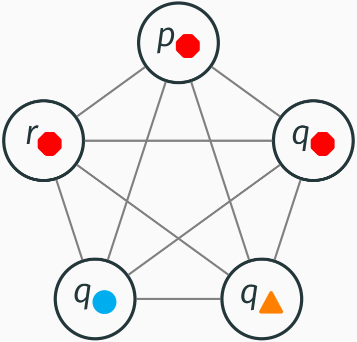
$$\mathbb{D} = \{ \ldots \}$$



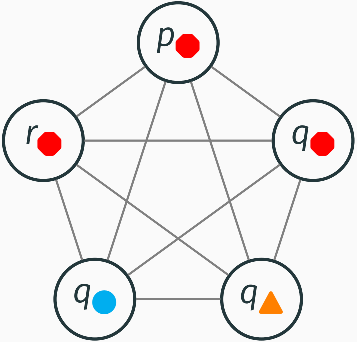
❹ Infinite input alphabets



❹ Infinite input alphabets



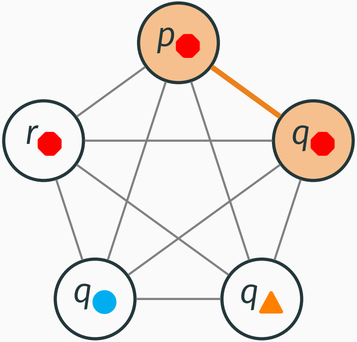
❹ Infinite input alphabets



$$p, q \xrightarrow{=} p', q'$$

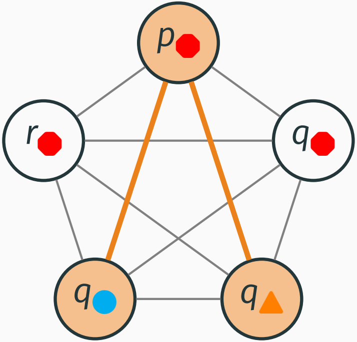
$$p, q \xrightarrow{\neq} p', q'$$

❹ Infinite input alphabets



$$p, q \xrightarrow{=} p', q'$$

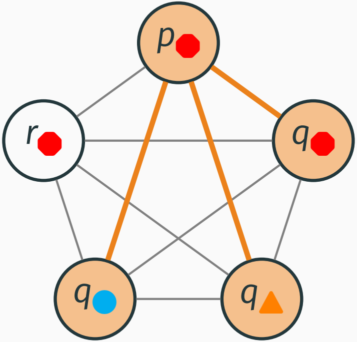
$$p, q \xrightarrow{\neq} p', q'$$



$$p, q \xrightarrow{=} p', q'$$

$$p, q \xrightarrow{\neq} p', q'$$

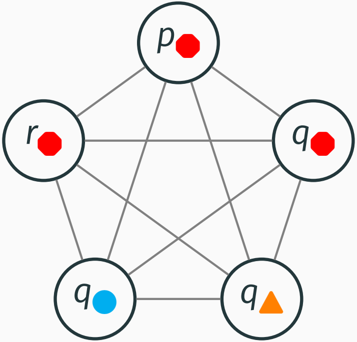
❹ Infinite input alphabets



$$p, q \xrightarrow{=} p', q'$$

$$p, q \xrightarrow{\neq} p', q'$$

❹ Infinite input alphabets



- Datum from $\mathbb{D}$ is immutable
- Input/output mappings depend only on Q

Theorem — **B., Ladouceur (ICALP'23)**

- Possible to compute "$\exists d \in \mathbb{D} : d$ has the majority"
- Imm.-obs. protocols compute Boolean combinations of

$$\overset{\text{distinct}}{\exists} d_1, \ldots, d_n \in \mathbb{D} : \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} \#(d_i, q_j) \in I_{i,j}$$

5 Conclusion

1. Population protocols: distributed computing by stable consensus of agents with little power
2. Succinct protocols, space vs. time trade-offs
3. Immediate-observation $<$-protocols $\equiv$ star-free unambiguous languages
4. Majority with infinitely many options can be decided with $\mathcal{O}(1)$ states

5 Conclusion: summary

1. Population protocols: distributed computing by stable consensus of agents with little power

2. Succinct protocols, space vs. time trade-offs

3. Immediate-observation $<$-protocols $\equiv$ star-free unambiguous languages

4. Majority with infinitely many options can be decided with $\mathcal{O}(1)$ states

❶ Population protocols: distributed computing by stable consensus of agents with little power

❷ Succinct protocols, space vs. time trade-offs

❸ Immediate-observation $<$-protocols $\equiv$ star-free unambiguous languages

❹ Majority with infinitely many options can be decided with $\mathcal{O}(1)$ states

1. Population protocols: distributed computing by stable consensus of agents with little power

2. Succinct protocols, space vs. time trade-offs

3. Immediate-observation $<$-protocols $\equiv$ star-free unambiguous languages

4. Majority with infinitely many options can be decided with $\mathcal{O}(1)$ states

State complexity:

- *Majority*: Is there an $\Omega(\log n)$ unconditional lower bound on (monotone) protocols with expected time $\mathcal{O}(n \log^c n)$?
- *Modulo*: Is there a family deciding $|w|_a \equiv 0 \pmod{m_n}$ with $\mathcal{O}(\log \log m_n)$ states?
- *General case*: exact state complexity? $\mathcal{O}(\|L\|)$ unconditionally? $\Omega(\|L\|)$?

State complexity:

- *Majority*: Is there an $\Omega(\log n)$ unconditional lower bound on (monotone) protocols with expected time $\mathcal{O}(n \log^c n)$?
- *Modulo*: Is there a family deciding $|w|_a \equiv 0 \pmod{m_n}$ with $\mathcal{O}(\log \log m_n)$ states?
- *General case*: exact state complexity? $\mathcal{O}(\|L\|)$ unconditionally? $\Omega(\|L\|)$?

State complexity:

- *Majority*: Is there an $\Omega(\log n)$ unconditional lower bound on (monotone) protocols with expected time $\mathcal{O}(n \log^c n)$?
- *Modulo*: Is there a family deciding $|w|_a \equiv 0 \pmod{m_n}$ with $\mathcal{O}(\log \log m_n)$ states?
- *General case*: exact state complexity? $\mathcal{O}(\|L\|)$ unconditionally? $\Omega(\|L\|)$?

$<$-protocols and protocols with infinite alphabets:

- Expressiveness?
- State complexity?
- Expected time complexity?

<-protocols and protocols with infinite alphabets:

- Expressiveness?
- State complexity?
- Expected time complexity?

Are unambiguous p.o. Parikh automata
closed under complementation?

Thank you!

Merci!