

The Power and Limits of Population Protocols

Michael Blondin



Université de
Sherbrooke

🕒 Determining babies' needs



① Determining babies' needs



🎯 Determining babies' needs



🎯 Determining babies' needs



🕒 Determining babies' needs



🕒 Determining babies' needs

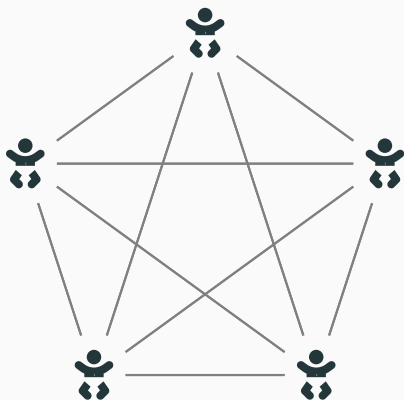


🕒 Determining babies' needs

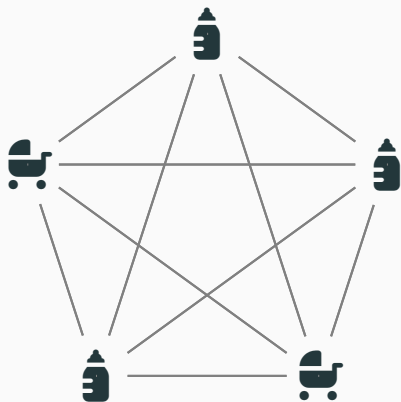


How to determine the majority?

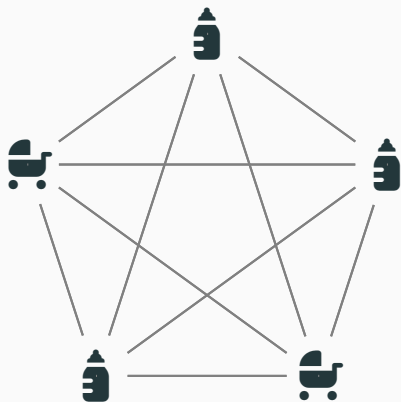
Population



Population



Population

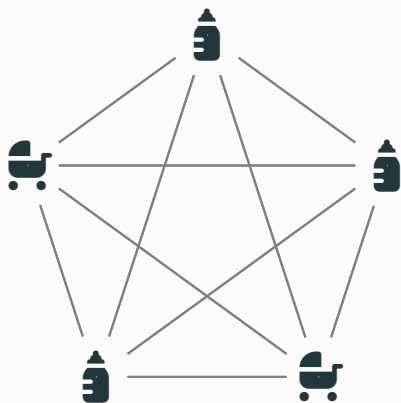


Informal rules

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

Majority vote protocol

Population



Formal rules

🍼 🛒 → 🍼 🛒

🍼 🛒 → 🍼 🍼

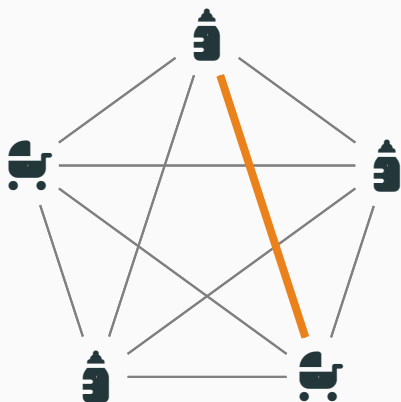
🛒 🍼 → 🛒 🛒

Informal rules

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

Majority vote protocol

Population



Formal rules

🍼 🛒 → 🍼 🛒

🍼 🛒 → 🍼 🍼

🛒 🍼 → 🛒 🛒

Informal rules

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

Majority vote protocol

Population



Formal rules

  $\rightarrow$  

  $\rightarrow$  

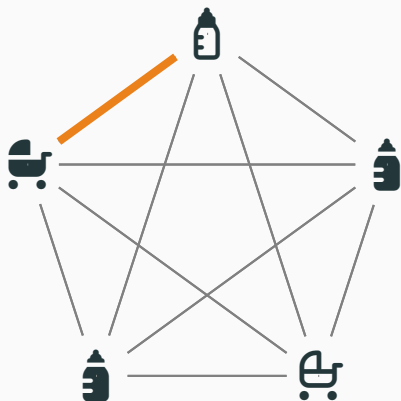
  $\rightarrow$  

Informal rules

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

Majority vote protocol

Population



Formal rules

  $\rightarrow$  

  $\rightarrow$  

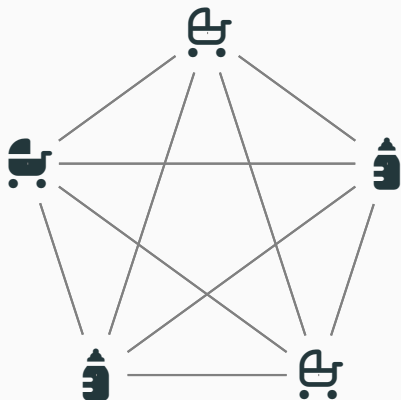
  $\rightarrow$  

Informal rules

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

Majority vote protocol

Population



Formal rules

👶 🛒 → 🍼 🛒

👶 🛒 → 🍼 🍼

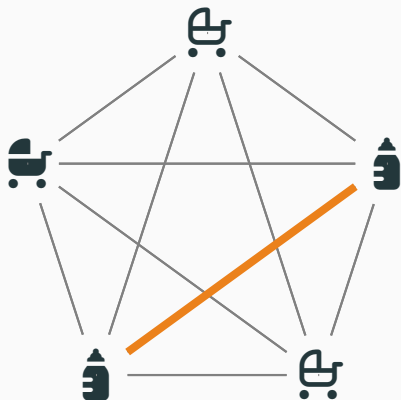
🛒 🍼 → 🛒 🛒

Informal rules

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

Majority vote protocol

Population



Formal rules

Active Active $\rightarrow$ Active Active

Active Passive $\rightarrow$ Active Active

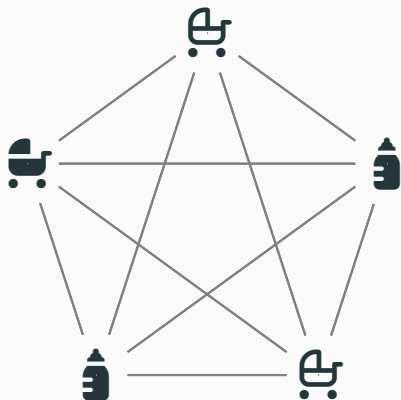
Passive Active $\rightarrow$ Passive Passive

Informal rules

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

Majority vote protocol

Population



Formal rules

🍼 🍼 → 🍼 🍼

🍼 🍼 → 🍼 🍼

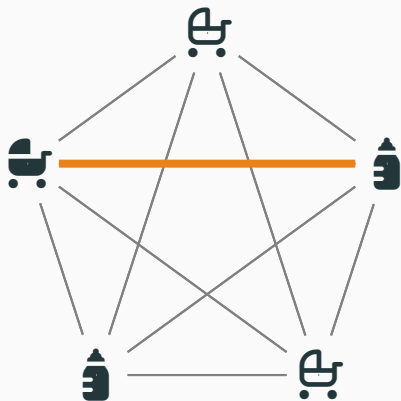
🍼 🍼 → 🍼 🍼

Informal rules

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

Majority vote protocol

Population



Formal rules

Active Stroller → Passive Stroller

Active Bottle → Passive Bottle

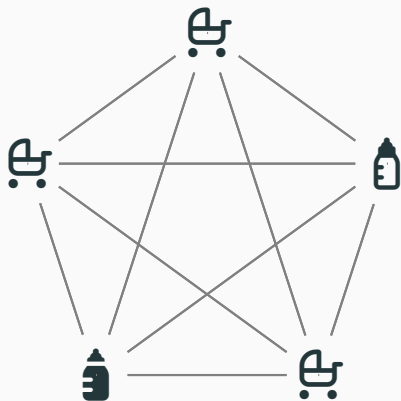
Active Bottle → Active Stroller

Informal rules

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

Majority vote protocol

Population



Formal rules

🍼 🛒 → 🍼 🛒

🍼 🛒 → 🍼 🍼

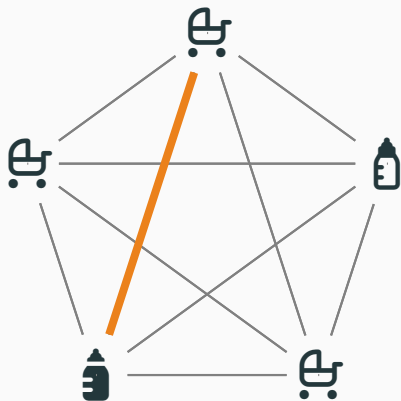
🛒 🍼 → 🛒 🛒

Informal rules

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

Majority vote protocol

Population



Formal rules

  $\rightarrow$  

  $\rightarrow$  

  $\rightarrow$  

Informal rules

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

Majority vote protocol

Population



Formal rules

🍼 🛒 → 🍼 🛒

🍼 🛒 → 🍼 🍼

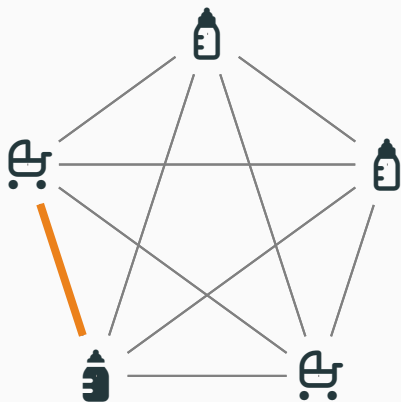
🛒 🍼 → 🛒 🛒

Informal rules

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

Majority vote protocol

Population



Formal rules

  $\rightarrow$  

  $\rightarrow$  

  $\rightarrow$  

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

$\rightarrow$

$\rightarrow$

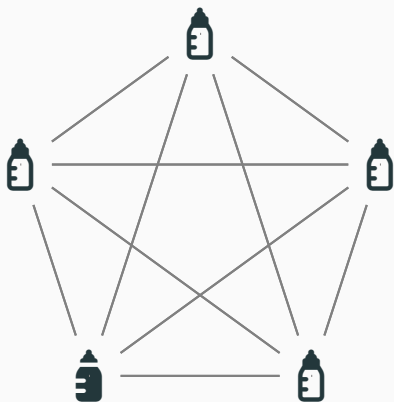
$\rightarrow$

Informal rules

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

Majority vote protocol

Population



Formal rules

  $\rightarrow$  
  $\rightarrow$  
  $\rightarrow$  

Informal rules

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

Short demo

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



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



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

- Introduced by Angluin, Aspö
- Dijkstra Prize of 2020

The challenge of computing in a highly dynamic environment.

BY OTHON MICHAEL 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 dynamism 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 dynamism and of the highly unpredictable. A great variety of modern networked systems are highly dynamic, but none are yet fully understood.



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

This talk:

1 The model



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

This talk:

- 1 The model
- 2 State complexity



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



This talk:

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



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

① Population protocols

The model

1 Population protocols: formal model

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

$$Q = \{ \text{👤}, \text{🛒}, \text{👤}, \text{🛒} \}$$

1 Population protocols: formal model

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

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

$$\iota(b) = \text{👤}$$

$$\iota(c) = \text{🛒}$$

1 Population protocols: formal model

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

$$o(\text{♂}) = o(\text{♀}) = 1$$

$$o(\text{♂}) = o(\text{♀}) = 0$$

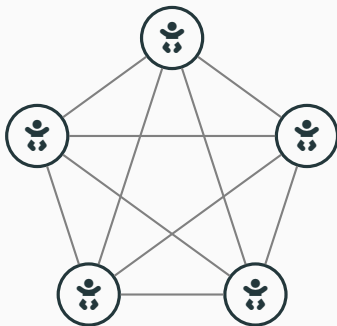
1 Population protocols: formal model

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



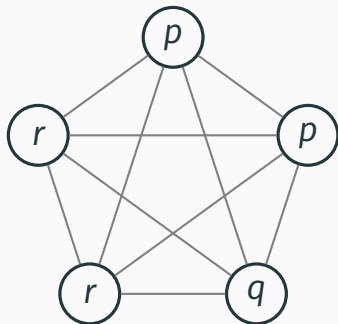
1 Population protocols: interactions

All agents can interact pairwise
(complete topology)



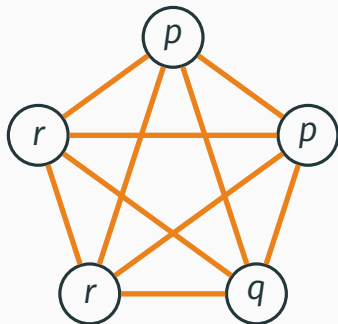
1 Population protocols: interactions

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



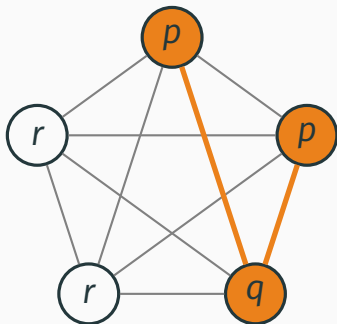
1 Population protocols: interactions

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



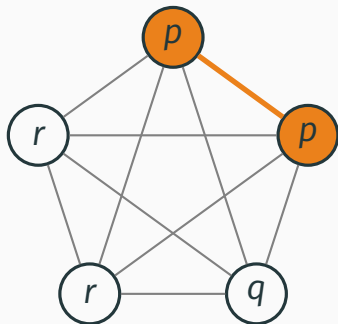
1 Population protocols: interactions

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



1 Population protocols: interactions

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



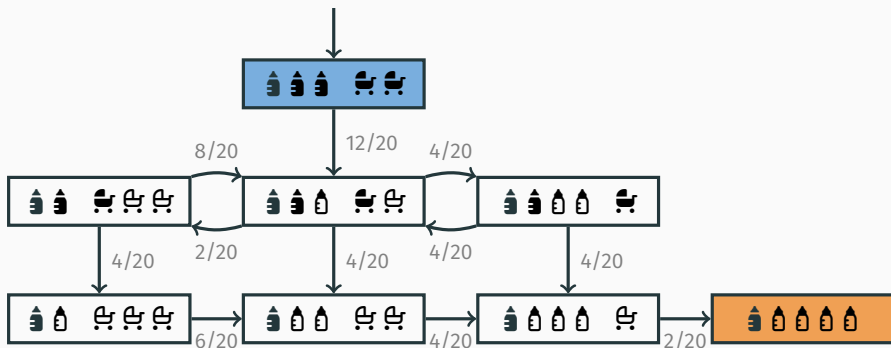
1 Population protocols: interactions

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

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

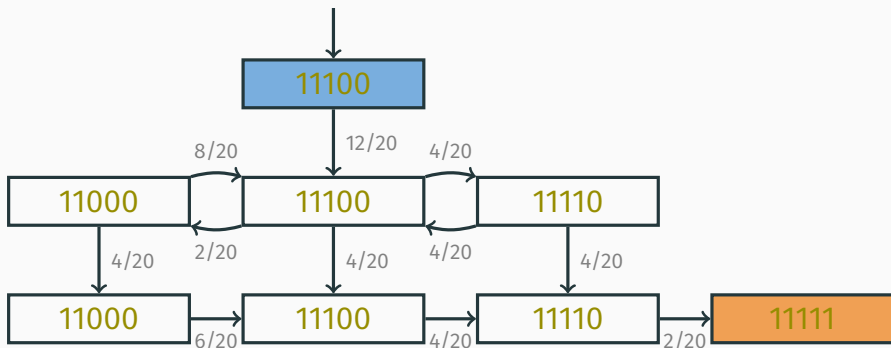
1 Computability: example

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



1 Computability: example

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



1 Computability: definition

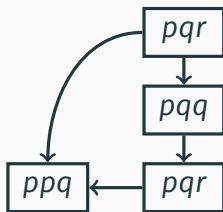
- A configuration C is **stable** if

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

1 Computability: definition

- A configuration C is **stable** if

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



$$p, r \rightarrow p, p$$

$$q, q \rightarrow q, r$$

$$q, r \rightarrow q, q$$

$$o(p) = 1$$

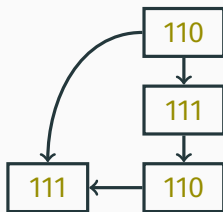
$$o(q) = 1$$

$$o(r) = 0$$

1 Computability: definition

- A configuration C is **stable** if

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



$$p, r \rightarrow p, p$$

$$q, q \rightarrow q, r$$

$$q, r \rightarrow q, q$$

$$o(p) = 1$$

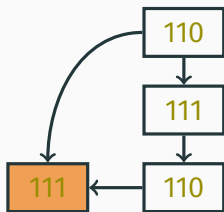
$$o(q) = 1$$

$$o(r) = 0$$

1 Computability: definition

- A configuration C is **stable** if

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



$$p, r \rightarrow p, p$$

$$q, q \rightarrow q, r$$

$$q, r \rightarrow q, q$$

$$o(p) = 1$$

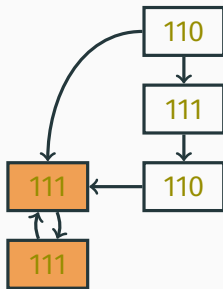
$$o(q) = 1$$

$$o(r) = 0$$

1 Computability: definition

- A configuration C is **stable** if

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



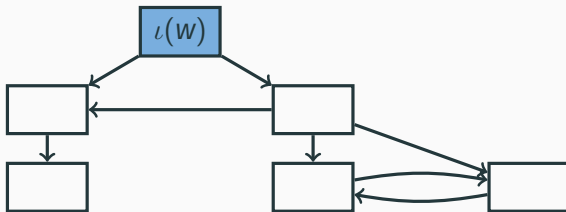
1 Computability: definition

- A configuration C is **stable** if

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

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

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



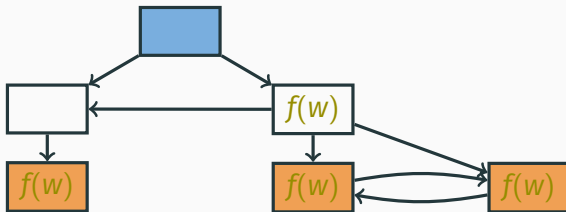
1 Computability: definition

- A configuration C is **stable** if

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

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

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



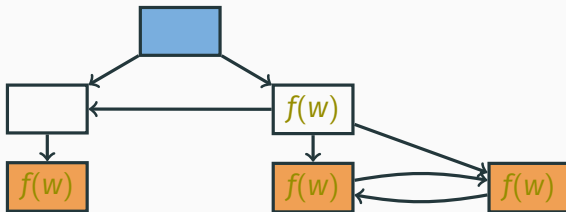
1 Computability: definition

- A configuration C is **stable** if

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

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

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



1 Computability: definition

- A configuration C is **stable** if

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

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

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

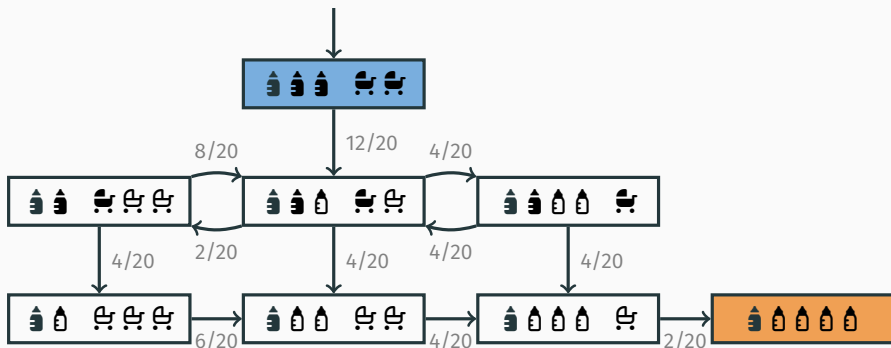
- A protocol **decides** $L \subseteq \Sigma^\oplus$ if it computes $f: \Sigma^\oplus \rightarrow \{0,1\}^\oplus$ s.t.

$$f(w) = 1 \dots 1 \quad \text{if } w \in L$$

$$f(w) = 0 \dots 0 \quad \text{if } w \notin L$$

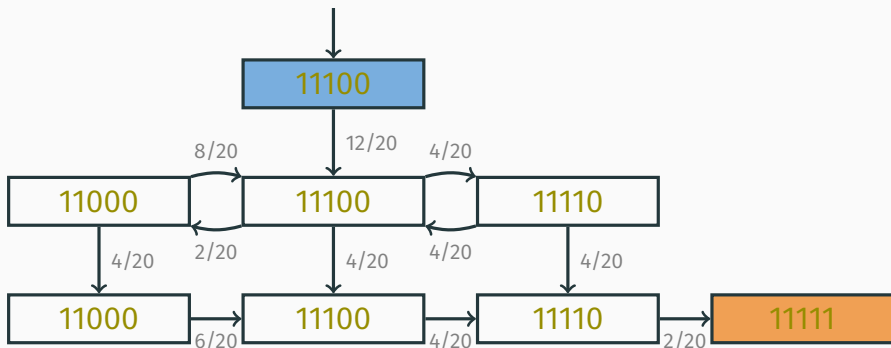
1 Computability: example (again)

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



1 Computability: example (again)

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



1 Computability: majority vote

Proposition

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

$\iota(b) =$				$\rightarrow$			$o(\text{bottle}) = 1$
$\iota(c) =$				$\rightarrow$			$o(\text{bottle}) = 1$
				$\rightarrow$			$o(\text{shopping cart}) = 0$
				$\rightarrow$			$o(\text{shopping cart}) = 0$

1 Computability: majority vote

Proposition

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

$\iota(b) = y_+$	$y_+ n_+ \rightarrow y_- n_-$	$o(y_+) = 1$
$\iota(c) = n_+$	$y_+ n_- \rightarrow y_+ y_-$	$o(y_-) = 1$
	$n_+ y_- \rightarrow n_+ n_-$	$o(n_+) = 0$
	$y_- n_- \rightarrow n_- n_-$	$o(n_-) = 0$

1 Computability: majority vote

Proposition

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

$\iota(b) = y_+$	$y_+ n_+ \rightarrow y_- n_-$	$o(y_+) = 1$
$\iota(c) = n_+$	$y_+ n_- \rightarrow y_+ y_-$	$o(y_-) = 1$
	$n_+ y_- \rightarrow n_+ n_-$	$o(n_+) = 0$
	$y_- n_- \rightarrow n_- n_-$	$o(n_-) = 0$

Proof

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

1 Computability: majority vote

Proposition

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

$\iota(b) = y_+$	$y_+ n_+ \rightarrow y_- n_-$	$o(y_+) = 1$
$\iota(c) = n_+$	$y_+ n_- \rightarrow y_+ y_-$	$o(y_-) = 1$
	$n_+ y_- \rightarrow n_+ n_-$	$o(n_+) = 0$
	$y_- n_- \rightarrow n_- n_-$	$o(n_-) = 0$

Proof

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

1 Computability: majority vote

Proposition

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

$\iota(b) = y_+$	$y_+ n_+ \rightarrow y_- n_-$	$o(y_+) = 1$
$\iota(c) = n_+$	$y_+ n_- \rightarrow y_+ y_-$	$o(y_-) = 1$
	$n_+ y_- \rightarrow n_+ n_-$	$o(n_+) = 0$
	$y_- n_- \rightarrow n_- n_-$	$o(n_-) = 0$

Proof

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

1 Expressiveness

Theorem

Angluin et al. (PODC'06)

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

1 Expressiveness

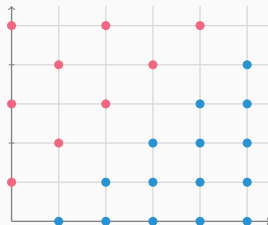
Theorem

Angluin et al. (PODC'06)

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

Presburger arithmetic:

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



1 Expressiveness

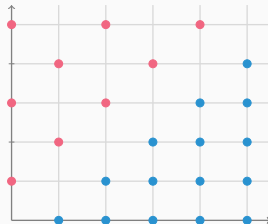
Theorem

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)



1 Expressiveness

Theorem

Angluin et al. (PODC'06)

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

Examples

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

1 Expressiveness

Theorem

Angluin et al. (PODC'06)

$L \subseteq \Sigma^+$ is decided by a population protocol iff L is semilinear

Examples

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

General case

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

1 Expressiveness

Theorem

Angluin et al. (PODC'06)

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

Examples

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

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

General case

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

② State complexity

**How many states
suffice/required?**

② State complexity: majority

Let $n := \# \text{ agents}$ 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

② State complexity: majority

Let $n := \# \text{ agents}$ 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 $|\# \text{ yes} - \# \text{ no}| = 1$

② State complexity: majority

Let $n := \# \text{ agents}$ 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)

2 State complexity: majority

Let $n := \# \text{ agents}$ 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	$\mathcal{O}(n \log n)$	$\leq \frac{1}{e^{c \cdot \epsilon^2 n}}$

2 State complexity: majority

Let $n := \# \text{ agents}$ 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

2 State complexity: majority

Let $n := \# \text{ agents}$ 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	$\mathcal{O}(n \log n)$	$\leq \frac{1}{e^{\epsilon} \cdot \epsilon^2 n}$
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

2 State complexity: majority

Let $n := \# \text{ agents}$ 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

② State complexity: threshold

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

- $Q = \{0, 1, \dots, m\}$
- $\iota(a) = 1, \iota(\sigma) = 0$ otherwise
- $o(m) = 1, o(q) = 0$ otherwise
- Transitions:
 - $p, q \rightarrow \min(p + q, m), p + q - \min(p + q, m)$
 - $m, q \rightarrow m, m$

② State complexity: threshold

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

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

$$2^i, 2^i \rightarrow 2^{i+1}, 0 \quad \text{if } i < k$$

$$2^k, q \rightarrow 2^k, 2^k$$

② State complexity: threshold

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

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

$$2^i, 2^i \rightarrow 2^{i+1}, 0 \quad \text{if } i < k$$

$$2^{i+1}, 0 \rightarrow 2^i, 2^i \quad \text{if } i < k - 1$$

$$\{2^i : i \in \text{bits}(m)\} \rightarrow \heartsuit, \dots, \heartsuit$$

$$\heartsuit, q \rightarrow \heartsuit, \heartsuit$$

② State complexity: threshold

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

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

$$2^i, 2^i \rightarrow 2^{i+1}, 0 \quad \text{if } i < k$$

$$2^{i+1}, 0 \rightarrow 2^i, 2^i \quad \text{if } i < k - 1$$

$$\{2^i : i \in \text{bits}(m)\} \rightarrow \heartsuit, \dots, \heartsuit$$

$$\heartsuit, q \rightarrow \heartsuit, \heartsuit$$

Emulated with $O(\log m)$ two-way transitions/states 8/14

② State complexity: threshold

Proposition

B., Esparza, Jaax (STACS'18)

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

2 State complexity: threshold

Proposition

B., Esparza, Jaax (STACS'18)

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

Theorem

Czerner (DISC'24)

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

2 State complexity: threshold

Proposition

B., Esparza, Jaax (STACS'18)

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

Theorem

Czerner (DISC'24)

$\exists m_1 < m_2 < \dots \in \mathbb{N}$ 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)

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

② State complexity: threshold

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

2 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)]

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



2 State complexity: linear inequalities

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

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

$$(\clubsuit, i, x), (s, j, y) \rightarrow (\clubsuit, c(i+j), x), (\heartsuit, i+j - c(i+j), y)$$

$$(\clubsuit, i, x), (\heartsuit, j, y) \rightarrow (\clubsuit, i, x \geq m), (\heartsuit, j, x \geq m)$$

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

2 State complexity: linear inequalities

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

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

$$(\clubsuit, i, x), (s, j, y) \rightarrow (\clubsuit, c(i+j), x), (\gtrless, i+j - c(i+j), y)$$

$$(\clubsuit, i, x), (\gtrless, j, y) \rightarrow (\clubsuit, i, x \geq m), (\gtrless, j, x \geq m)$$

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

② State complexity: linear inequalities

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

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

$$(\clubsuit, i, x), (s, j, y) \rightarrow (\clubsuit, c(i+j), x), (\heartsuit, i+j - c(i+j), y)$$

$$(\clubsuit, i, x), (\heartsuit, j, y) \rightarrow (\clubsuit, i, x \geq m), (\heartsuit, j, x \geq m)$$

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

2 State complexity: linear inequalities

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

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

$$(\clubsuit, i, x), (s, j, y) \rightarrow (\clubsuit, c(i+j), x), (\heartsuit, i+j - c(i+j), y)$$

$$(\clubsuit, i, x), (\heartsuit, j, y) \rightarrow (\clubsuit, i, x \geq m), (\heartsuit, j, x \geq m)$$

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

2 State complexity: linear inequalities

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

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

$$(\clubsuit, i, x), (s, j, y) \rightarrow (\clubsuit, c(i+j), x), (\gtrless, i+j - c(i+j), y)$$

$$(\clubsuit, i, x), (\gtrless, j, y) \rightarrow (\clubsuit, i, x \geq m), (\gtrless, j, x \geq m)$$

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

2 State complexity: linear inequalities

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

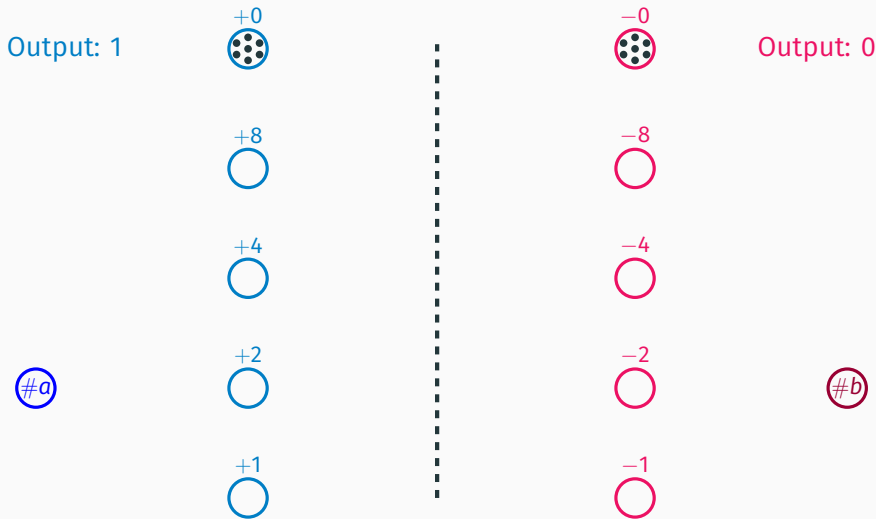
- $Q = \{\blacklozenge, \blacksquare\} \times [-B..B] \times \{0, 1\}$ with $B := \max(k_a, k_b, \dots, m)$
- $\iota(\sigma) = (\blacklozenge, k_{\sigma}, k_{\sigma} \geq m)$
- $o(s, i, x) = x$
- Transitions:

$$(\blacklozenge, i, x), (s, j, y) \rightarrow (\blacklozenge, c(i+j), x), (\blacksquare, i+j - c(i+j), y)$$

$$(\blacklozenge, i, x), (\blacksquare, j, y) \rightarrow (\blacklozenge, i, x \geq m), (\blacksquare, j, x \geq m)$$

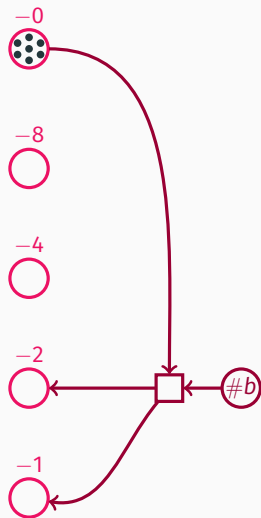
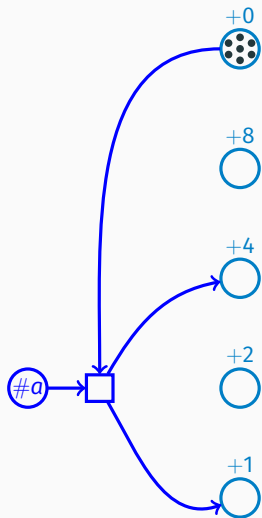
where $c(v) := \max(\min(v, B), -B)$

2 State complexity: linear inequalities



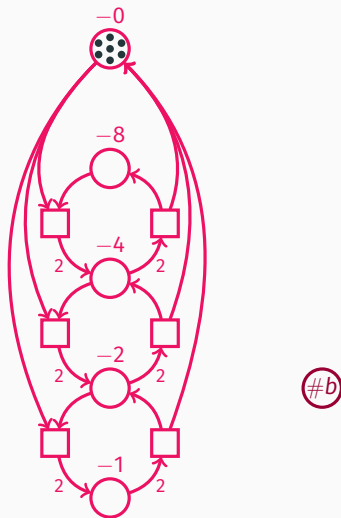
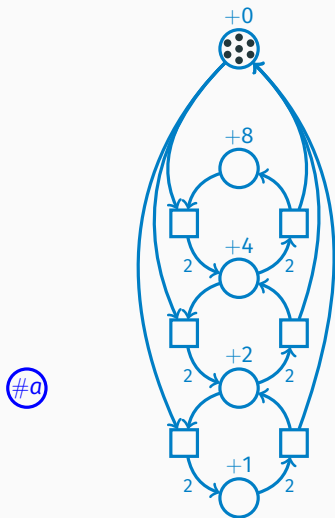
Protocol for $5|w|_a - 3|w|_b > 0$

2 State complexity: linear inequalities



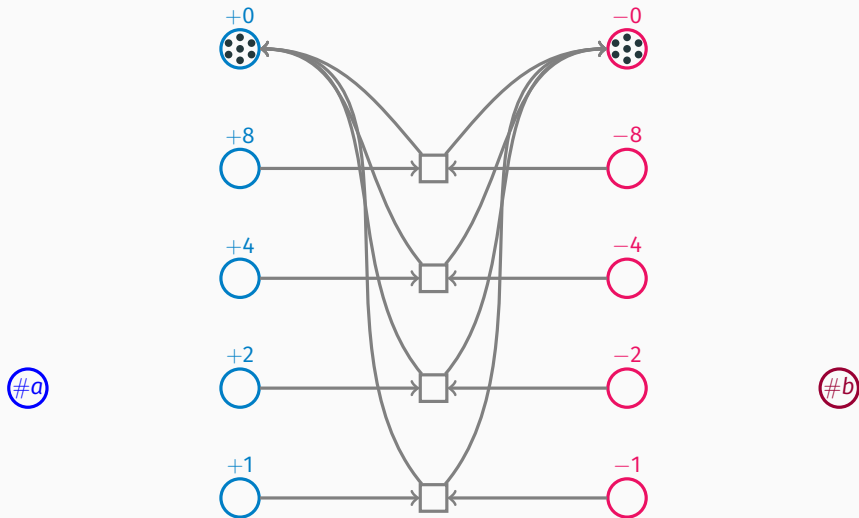
Protocol for $5|w|_a - 3|w|_b > 0$

2 State complexity: linear inequalities



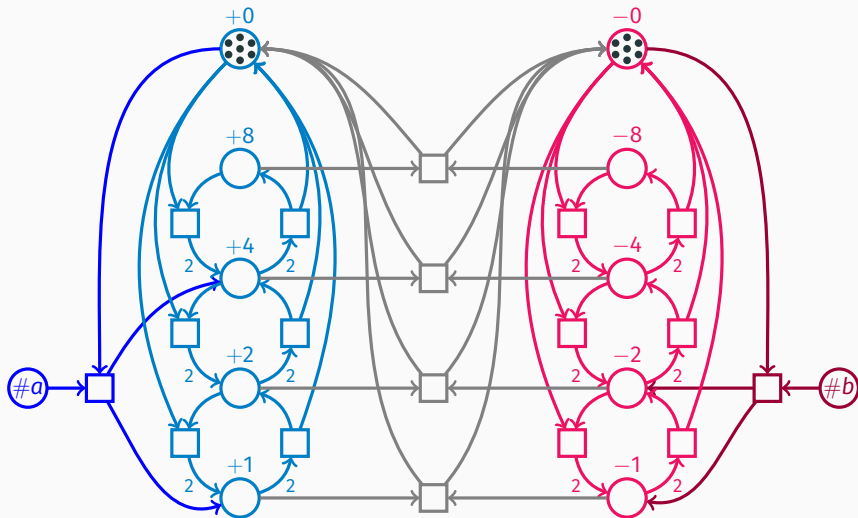
Protocol for $5|w|_a - 3|w|_b > 0$

2 State complexity: linear inequalities



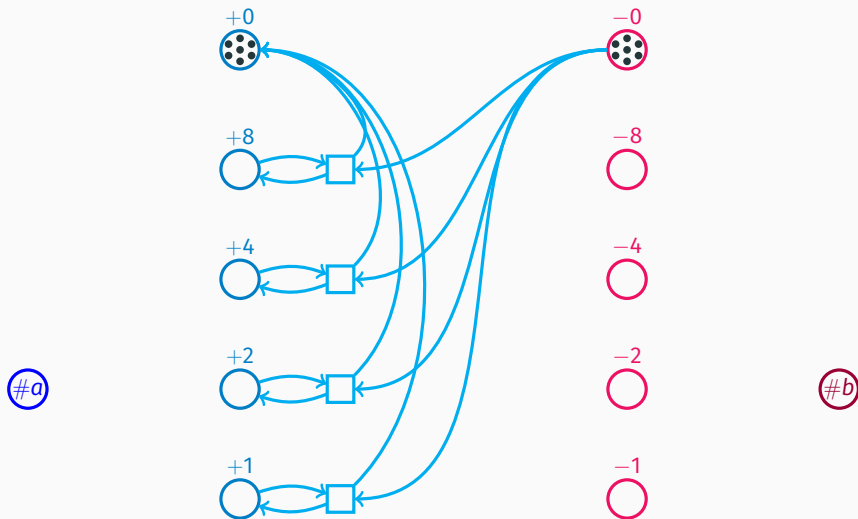
Protocol for $5|w|_a - 3|w|_b > 0$

② State complexity: linear inequalities



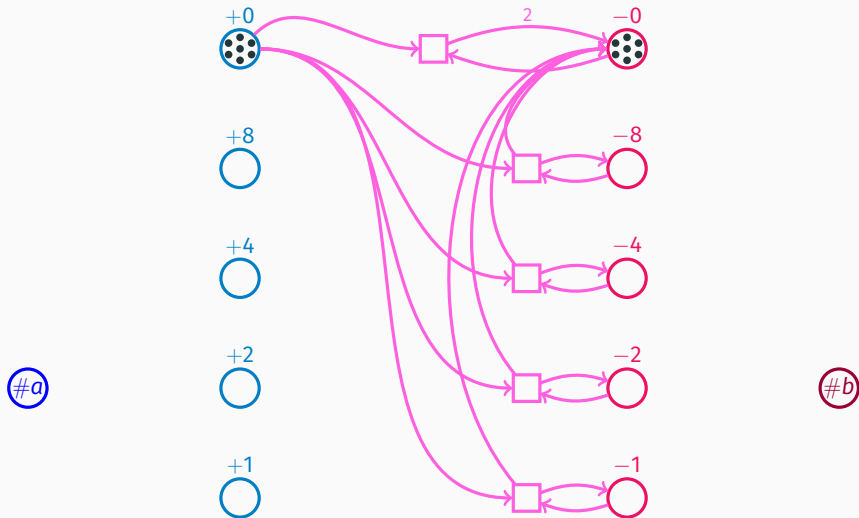
Protocol for $5|w|_a - 3|w|_b > 0$

2 State complexity: linear inequalities



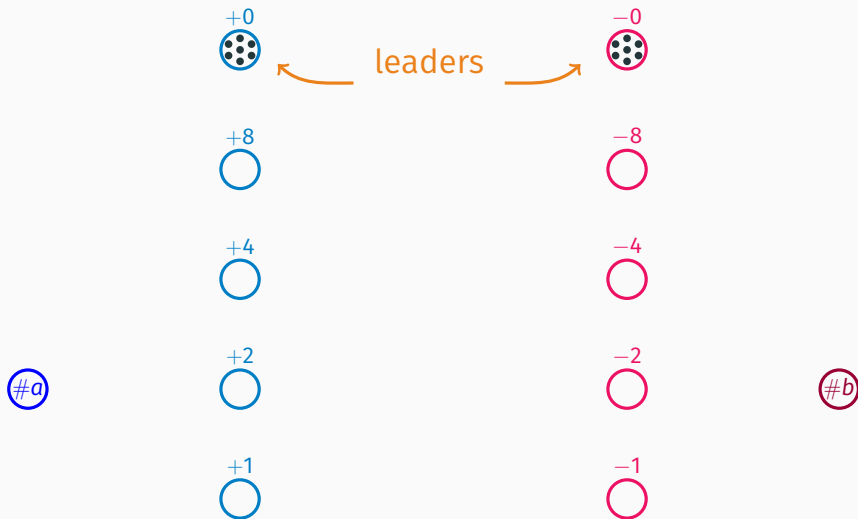
Protocol for $5|w|_a - 3|w|_b > 0$

2 State complexity: linear inequalities



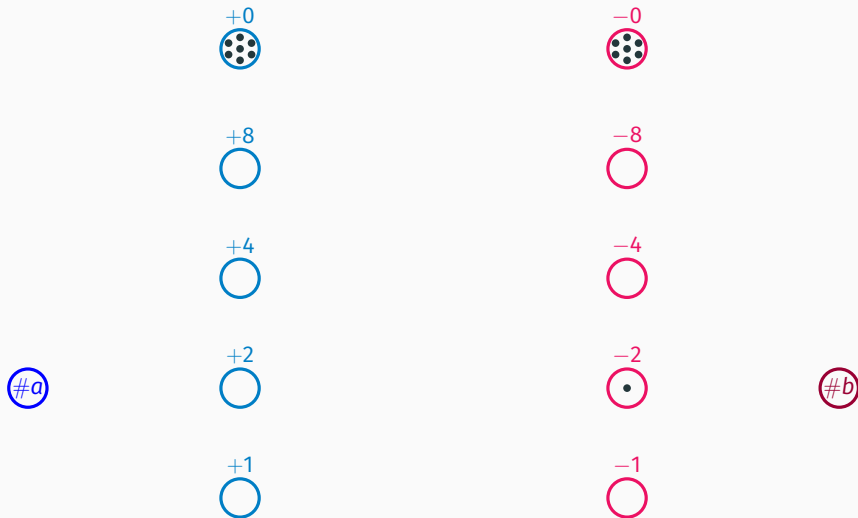
Protocol for $5|w|_a - 3|w|_b > 0$

2 State complexity: linear inequalities



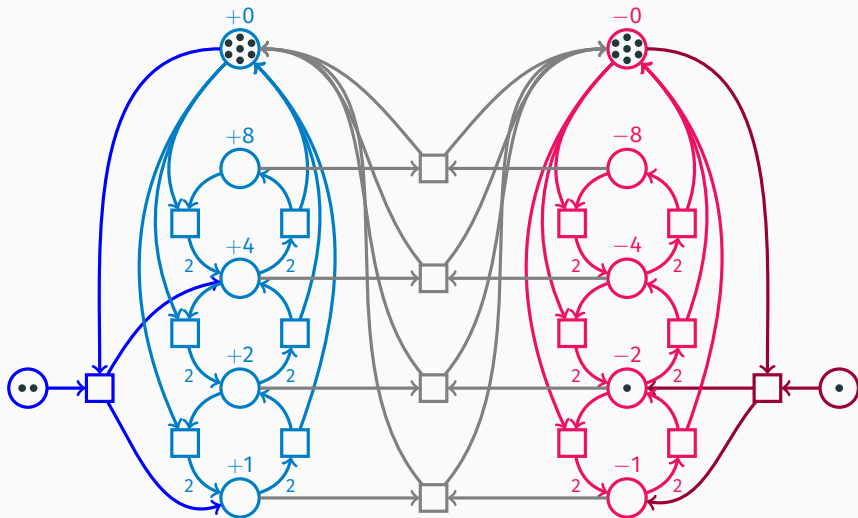
Protocol for $5|w|_a - 3|w|_b > 0$

2 State complexity: linear inequalities



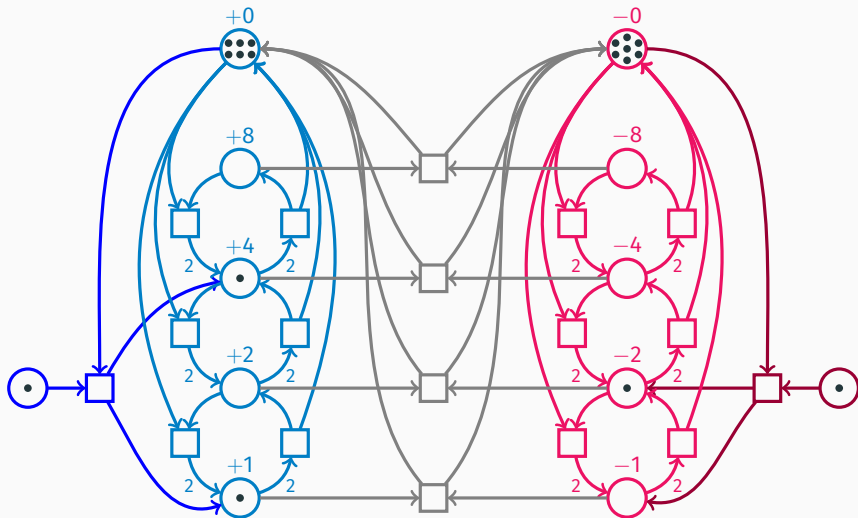
Protocol for $5|w|_a - 3|w|_b > 2$

② State complexity: linear inequalities



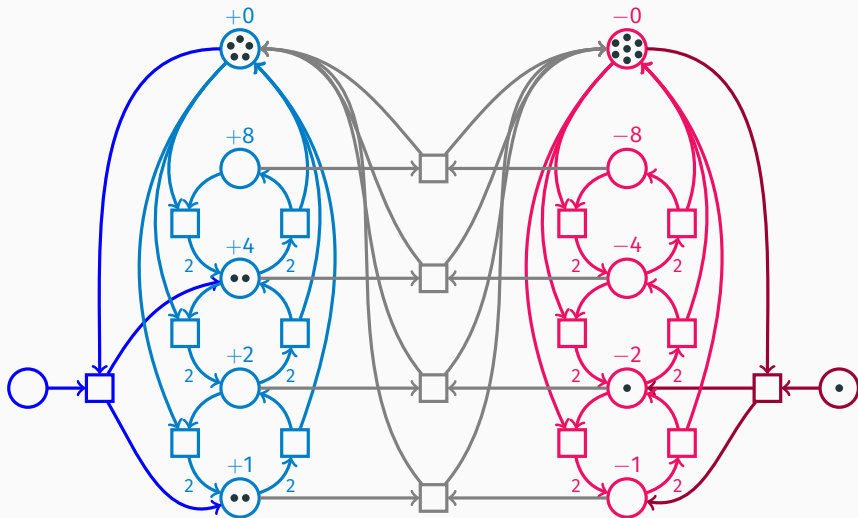
Protocol for $5|w|_a - 3|w|_b > 2$

② State complexity: linear inequalities



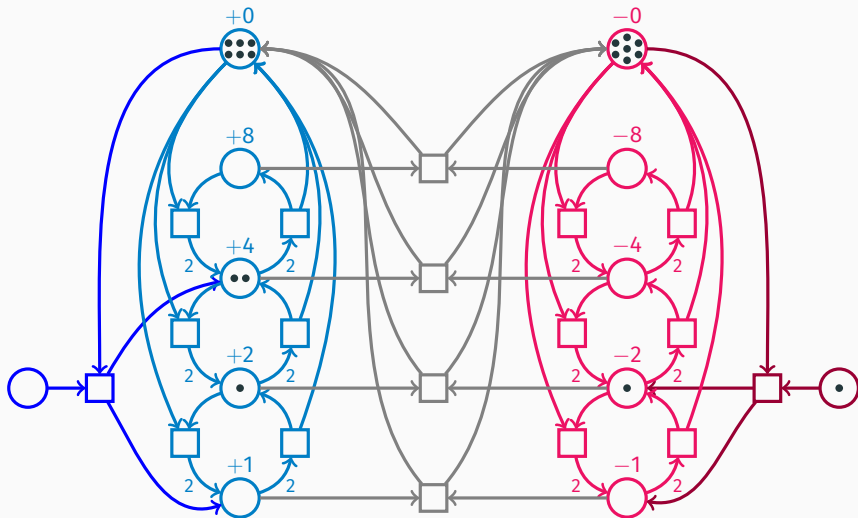
Protocol for $5|w|_a - 3|w|_b > 2$

② State complexity: linear inequalities



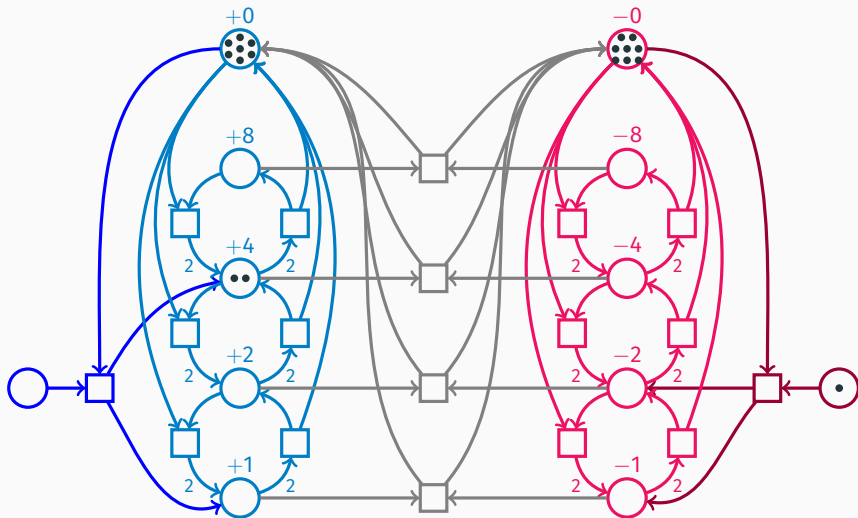
Protocol for $5|w|_a - 3|w|_b > 2$

② State complexity: linear inequalities



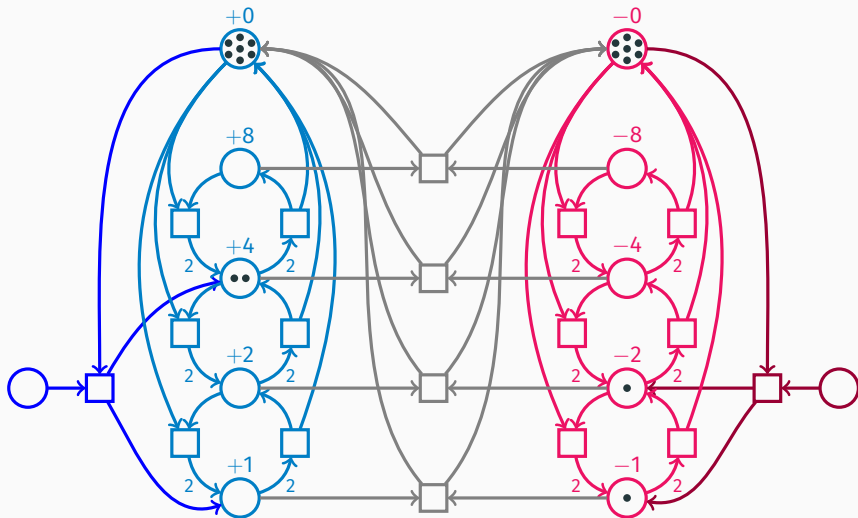
Protocol for $5|w|_a - 3|w|_b > 2$

② State complexity: linear inequalities



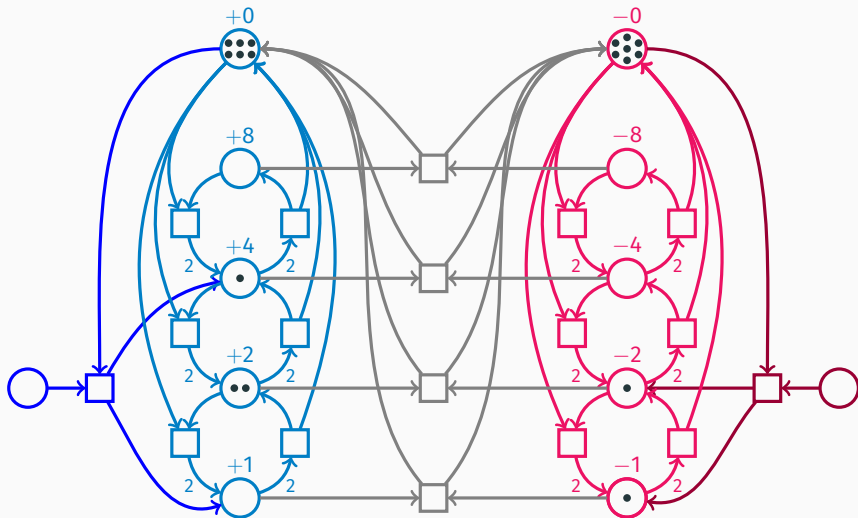
Protocol for $5|w|_a - 3|w|_b > 2$

② State complexity: linear inequalities



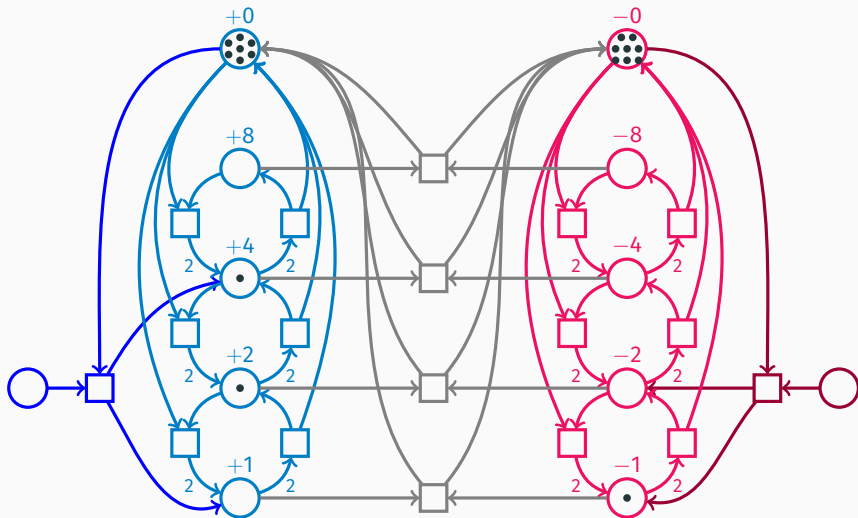
Protocol for $5|w|_a - 3|w|_b > 2$

② State complexity: linear inequalities



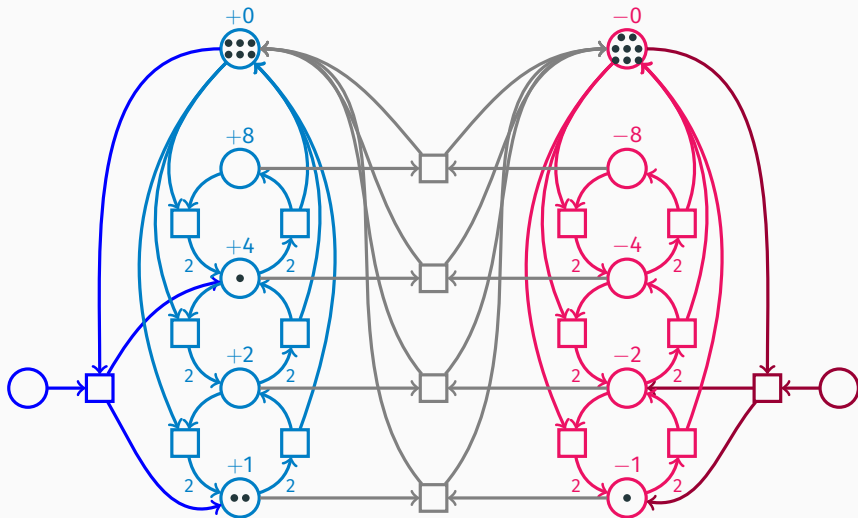
Protocol for $5|w|_a - 3|w|_b > 2$

② State complexity: linear inequalities



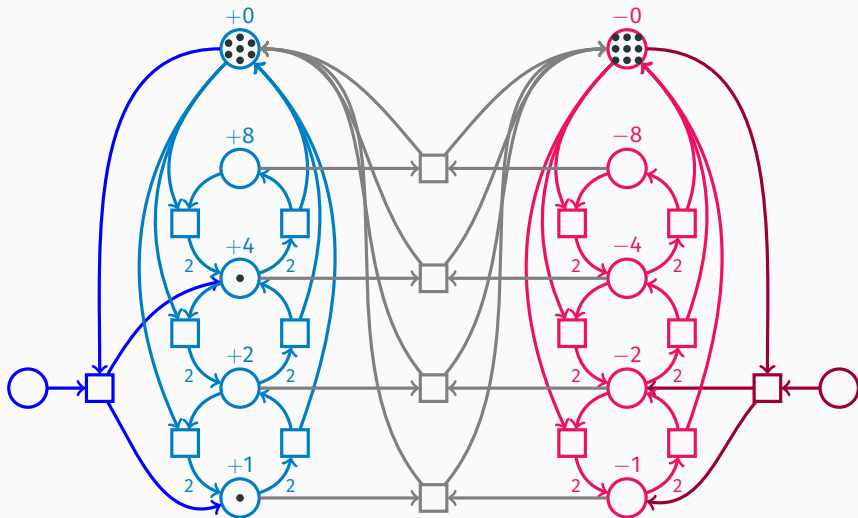
Protocol for $5|w|_a - 3|w|_b > 2$

② State complexity: linear inequalities



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

② State complexity: linear inequalities



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

② State complexity: linear inequalities

Theorem

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

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

② State complexity: linear inequalities

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 $\mathcal{O}(\text{poly}\|L\|)$ states

② State complexity: linear inequalities

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 $\mathcal{O}(\text{poly}\|L\|)$ states

*Extends to modulo constraints
and Boolean combinations*

2 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}(\text{poly}\ L\)$	$\Omega(2^n)$

2 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}(\text{poly}\ L\)$	$\Omega(2^n)$
Czerner et al. (SAND'22)	$\mathcal{O}(\text{poly}\ L\)$	$\mathcal{O}(\text{poly}\ L\ \cdot n^2)$

2 State complexity: general case

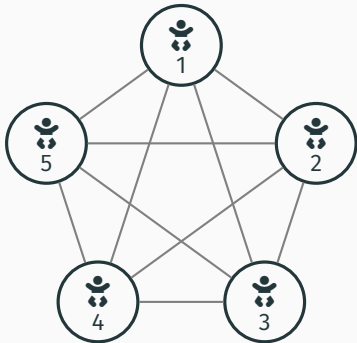
Protocol	# states	Expected time
Angluin et al. (PODC'04)	$\mathcal{O}(2^{\ L\ })$	$\mathcal{O}(\ L\ \cdot n^2 \log n)$
B. et al. (STACS'20)	$\mathcal{O}(\text{poly}\ L\)$	$\Omega(2^n)$
Czerner et al. (SAND'22)	$\mathcal{O}(\text{poly}\ L\)$	$\mathcal{O}(\text{poly}\ L\ \cdot n^2)$

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

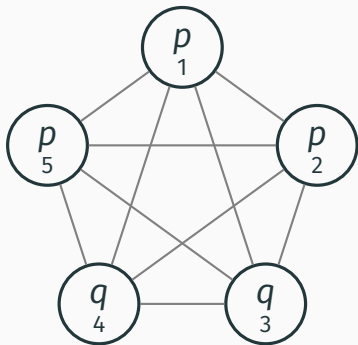
③ Protocols with order

**What happens if agents have
(unique) identifiers?**

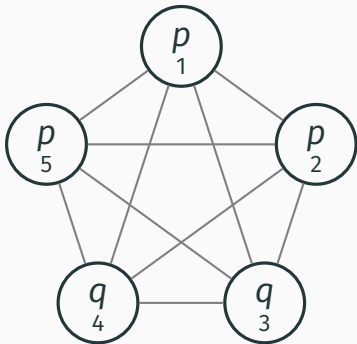
Agents with ordered IDs



Agents with ordered IDs



Agents with ordered IDs

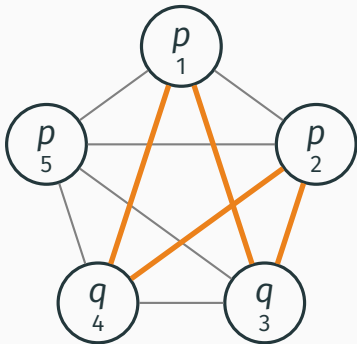


$$p, q \xrightarrow{\leq} p', q'$$

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

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

Agents with ordered IDs

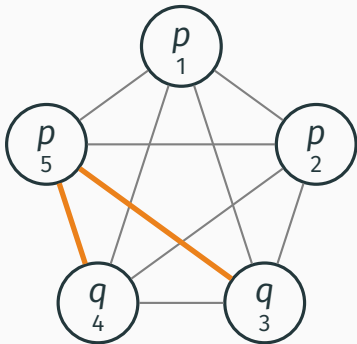


$$p, q \xrightarrow{\leq} p', q'$$

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

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

Agents with ordered IDs

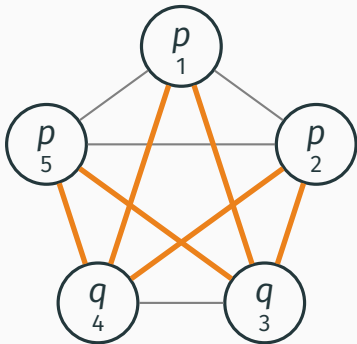


$$p, q \xrightarrow{\leq} p', q'$$

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

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

Agents with ordered IDs

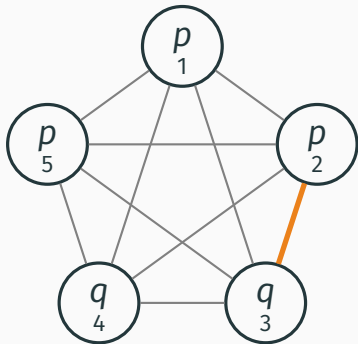


$$p, q \xrightarrow{\leq} p', q'$$

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

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

Agents with ordered IDs



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

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

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

Agents with ordered IDs: example

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

- $Q = \{\bullet_a, \bullet_b, \circ_a, \circ_b, \bullet_a, \bullet_b, \bullet_a, \bullet_b\} \times \{0, 1\}$
- $\iota(a) = \bullet_a, \iota(b) = \bullet_b$
- $O(s_x) = 1, O(s_x) = 0$
- Transitions:

$$\begin{aligned}
 \bullet_x s_y &\xrightarrow{\quad} \bullet_x s_y \\
 s_x \bullet_y &\xrightarrow{\quad} s_x \bullet_y \\
 s_x \bullet_y &\xrightarrow{\quad} s_x \circ_y \\
 \bullet_x s_y &\xrightarrow{\quad} \circ_x s_y
 \end{aligned}$$

$$\begin{aligned}
 \bullet_a \bullet_a &\xrightarrow{\quad} \bullet_a \bullet_a \\
 \bullet_x \bullet_b &\xrightarrow{\quad} \bullet_x \bullet_b \\
 \bullet_b \bullet_y &\xrightarrow{\quad} \bullet_b \bullet_y
 \end{aligned}$$

$$\begin{aligned}
 \bullet_x q &\xrightarrow{\quad} \bullet_x q \\
 \bullet_x q &\xrightarrow{\quad} \bullet_x q
 \end{aligned}$$

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:

$$p, q \rightarrow p', q$$

$$p, q \rightarrow p, q'$$

- Transitions:

$$\begin{array}{l} \bullet_x s_y \xrightarrow{\quad} \bullet_x s_y \\ s_x \bullet_y \xrightarrow{\quad} s_x \bullet_y \\ s_x \bullet_y \xrightarrow{\quad} s_x \circ_y \\ \bullet_x s_y \xrightarrow{\quad} \circ_x s_y \end{array}$$

$$\begin{array}{l} \bullet_a \bullet_a \xrightarrow{\quad} \bullet_a \bullet_a \\ \bullet_x \bullet_b \xrightarrow{\quad} \bullet_x \bullet_b \\ \bullet_b \bullet_y \xrightarrow{\quad} \bullet_b \bullet_y \end{array}$$

$$\begin{array}{l} \bullet_x q \xrightarrow{\quad} \bullet_x q \\ \bullet_x q \xrightarrow{\quad} \bullet_x q \end{array}$$

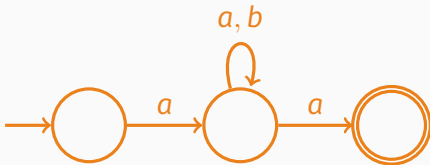
Agents with ordered IDs: example

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

$$p, q \rightarrow p', q$$

$$p, q \rightarrow p, q'$$

Unambiguous automaton for our imm.-obs. $\leftarrow$ -protocol:



ⓑ 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

ⓔ 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
 - $\text{LTL}[\mathbf{F}^{-1}, \mathbf{F}]$
 - $\text{FO}^2[<]$
 - $\Sigma_2[<] \cap \Pi_2[<]$
 - languages recognized by monoids from the variety DA

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
- partially-ordered
- $\text{LTL}[\mathbf{F}^{-1}, \mathbf{F}]$
- $\text{FO}^2[<]$
- $\Sigma_2[<] \cap \Pi_2[<]$
- 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.

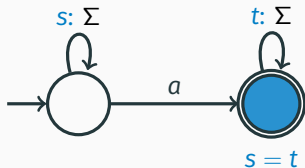
⊖ Agents with ordered IDs: expressiveness

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

ⓔ Agents with ordered IDs: expressiveness

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

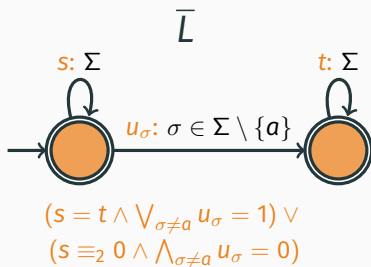
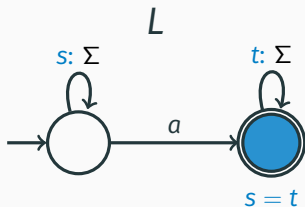
Partially-ordered Parikh automata:



ⓔ Agents with ordered IDs: expressiveness

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

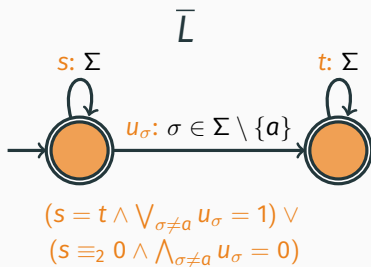
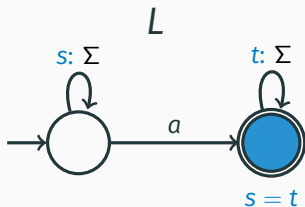
Partially-ordered Parikh automata:



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

Proposition

B. et al. (ICALP'26)

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

ⓔ Agents with ordered IDs: expressiveness

Proposition

B. et al. (ICALP'26)

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

Corollary

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

ⓔ Agents with ordered IDs: expressiveness

Community protocol :=

<-protocol + constant number of ID registers per agent

ⓔ Agents with ordered IDs: expressiveness

Community protocol $:=$

$\leftarrow$ -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

④ Protocols with data

**What happens if the input
alphabet is infinite?**

4 Infinite input alphabets

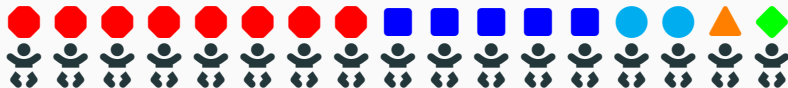
Majority vote with unboundedly many options

$$\mathbb{D} = \{\text{red octagon}, \text{blue square}, \text{cyan circle}, \text{orange triangle}, \text{green diamond}, \dots\}$$

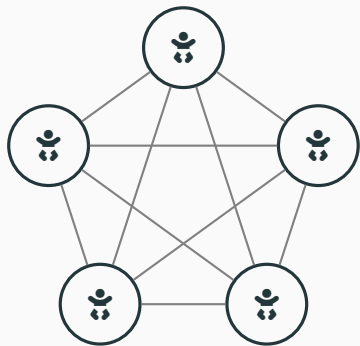
4 Infinite input alphabets

Majority vote with unboundedly many options

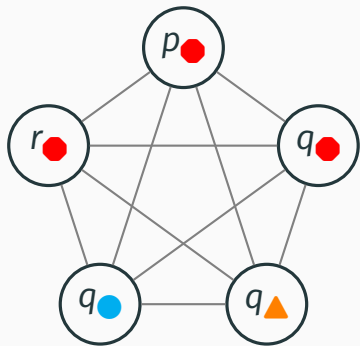
$$\mathbb{D} = \{\text{red octagon}, \text{blue square}, \text{blue circle}, \text{orange triangle}, \text{green diamond}, \dots\}$$



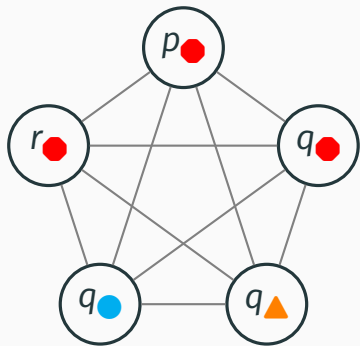
4 Infinite input alphabets



4 Infinite input alphabets



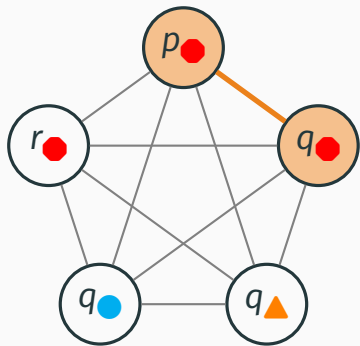
4 Infinite input alphabets



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

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

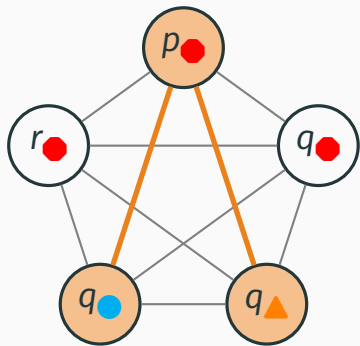
4 Infinite input alphabets



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

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

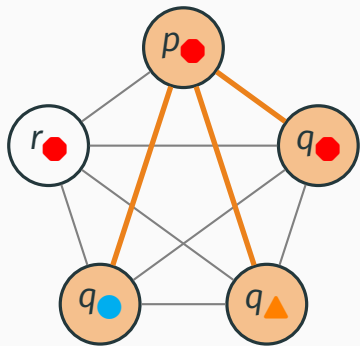
4 Infinite input alphabets



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

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

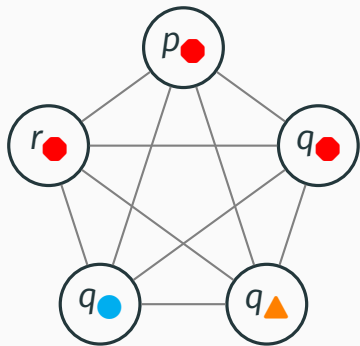
4 Infinite input alphabets



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

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

4 Infinite input alphabets



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

4 Infinite input alphabets: expressiveness

Theorem

B., Ladouceur (ICALP'23)

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

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

5 Conclusion

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 $\leq$ -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 $\leq$ -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 $\leftarrow$ -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 $\leq$ -protocols
 $\equiv$ star-free unambiguous languages
- 4 Majority with infinitely many options can be decided with $\mathcal{O}(1)$ states

5 Conclusion: some open questions

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? $\Omega(\|L\|)$?

5 Conclusion: some open questions

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? $\Omega(\|L\|)$?

5 Conclusion: some open questions

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? $\Omega(\|L\|)$?

5 Conclusion: some open questions

<-protocols and protocols with infinite alphabets:

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

5 Conclusion: some open questions

<-protocols and protocols with infinite alphabets:

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

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

Thank you!

Merci!