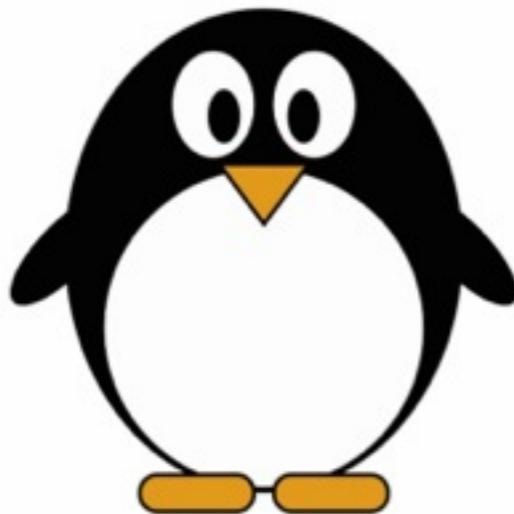


Computability in Population Protocols

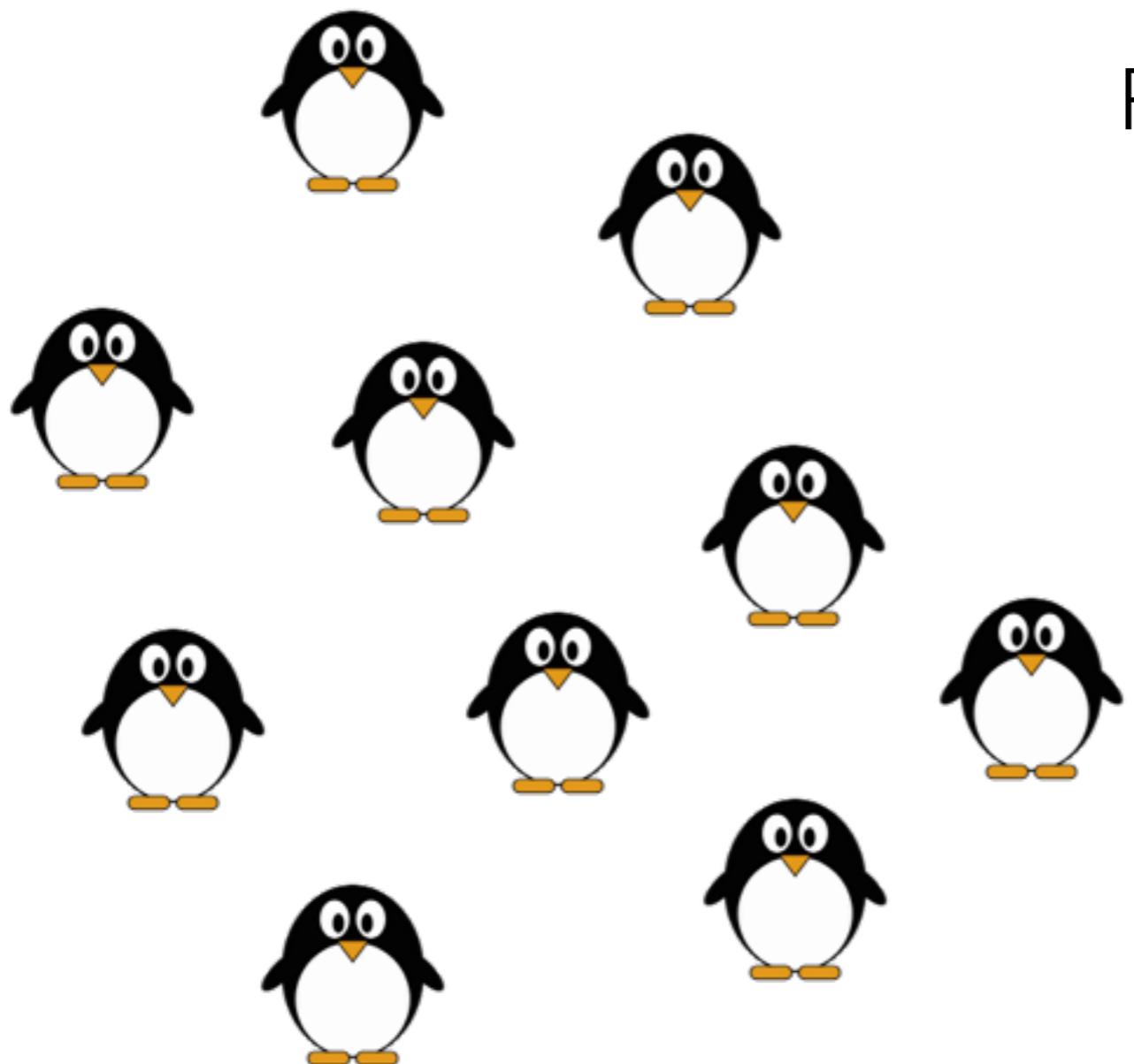
Peva Blanchard
EPFL 2014/2015

Population Protocol



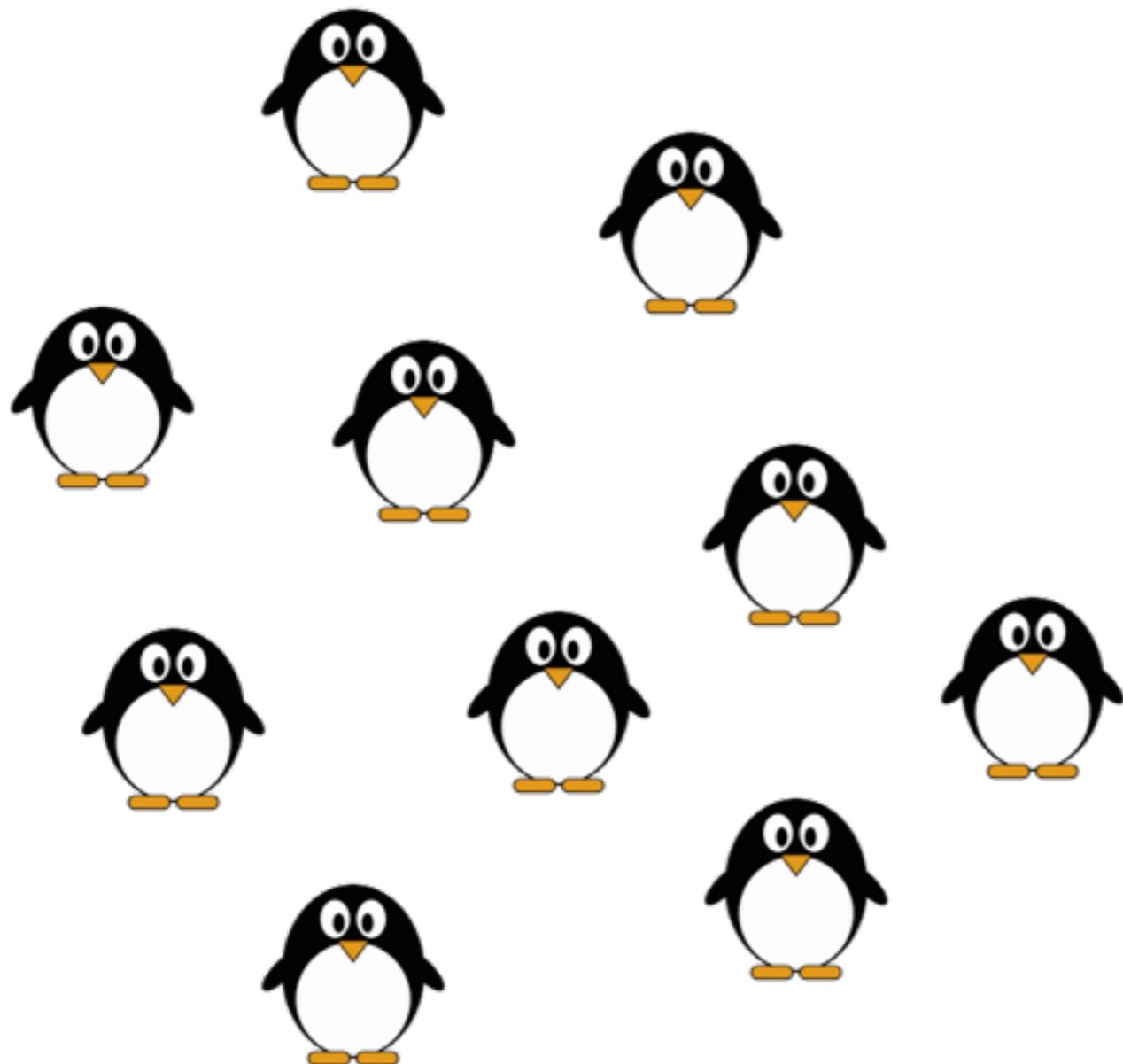
Agent : no id, small memory

Population Protocol



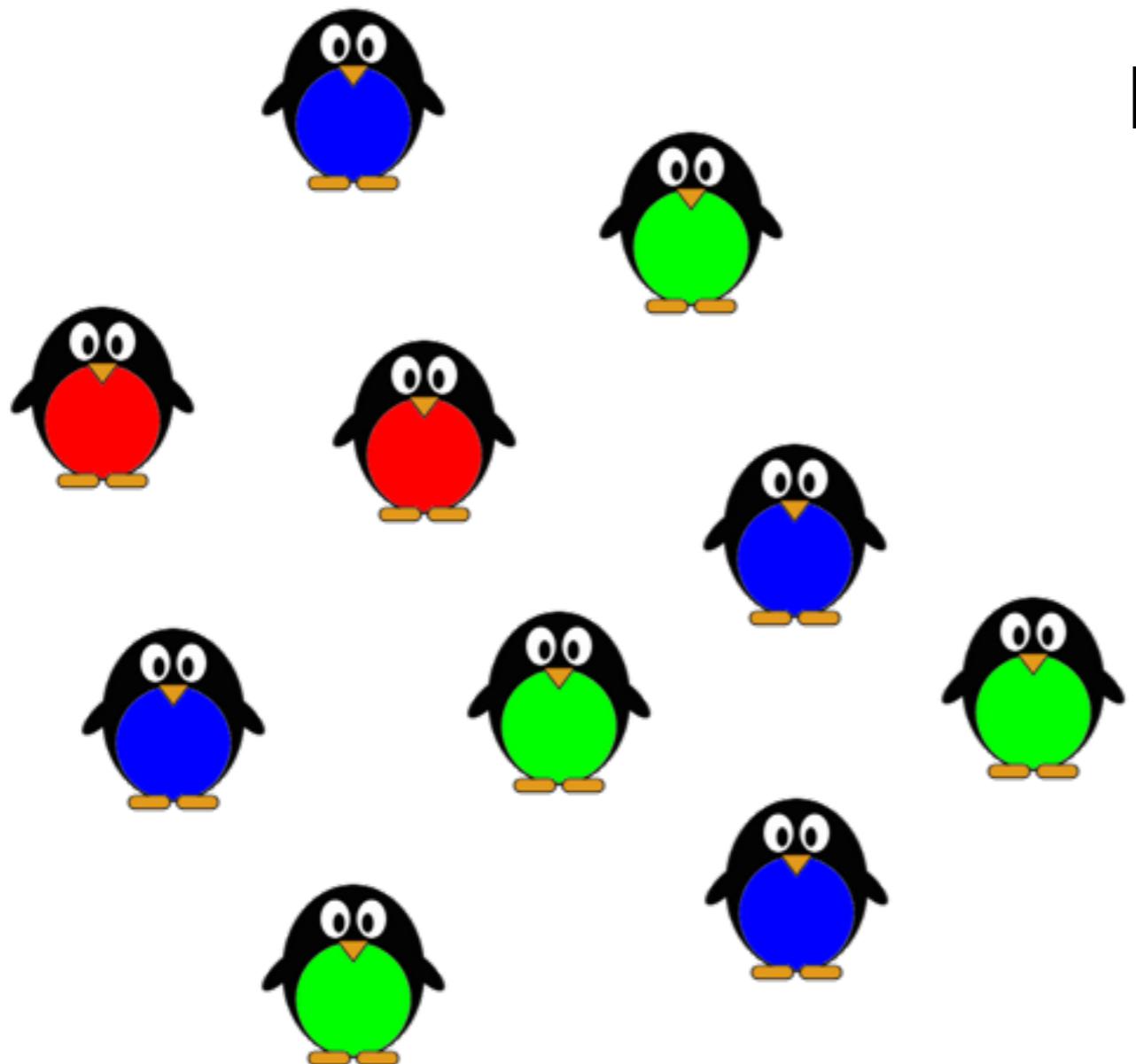
Population arbitrary size n

Population Protocol



Initially: sensors give data

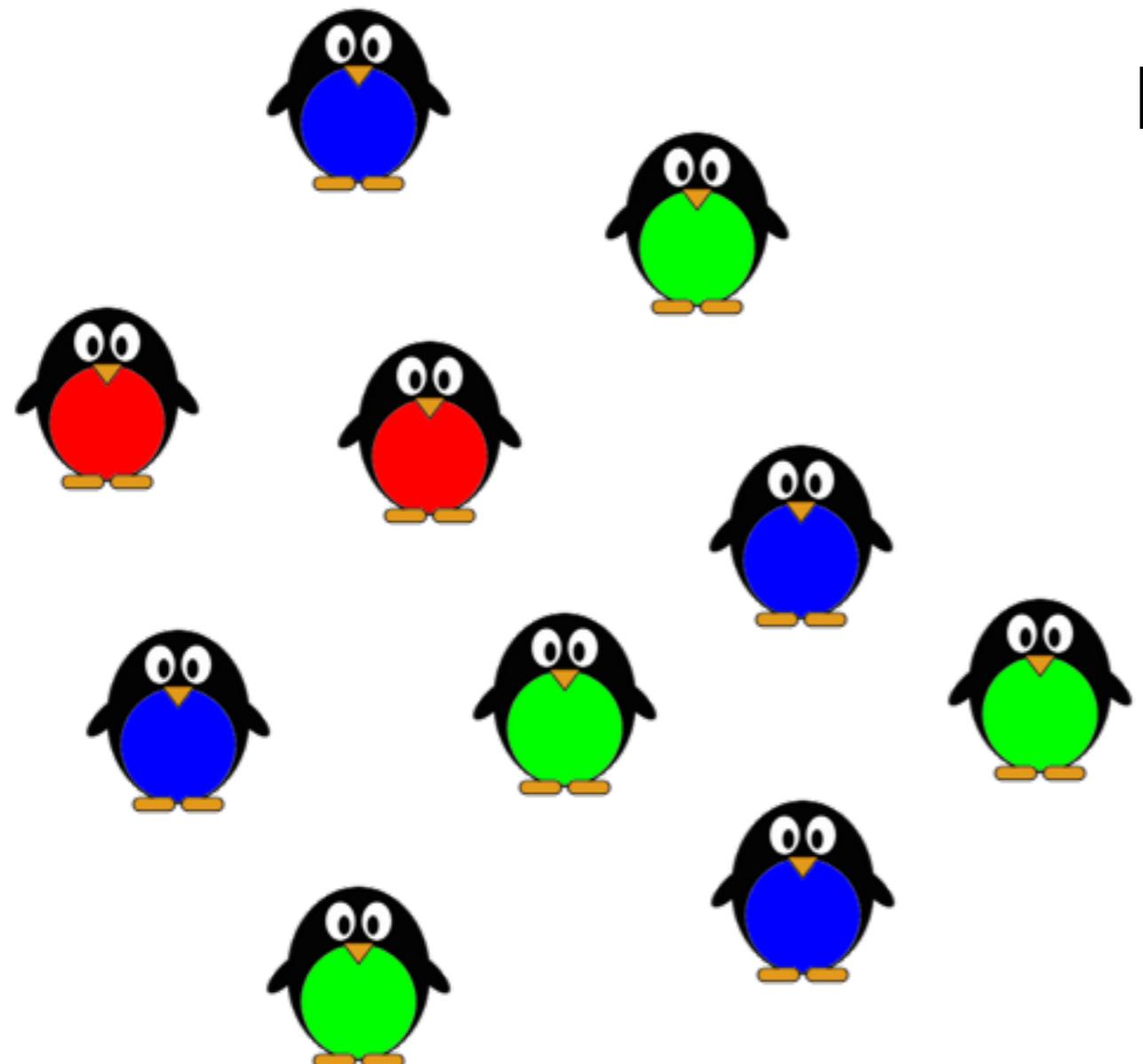
Population Protocol



Initially: sensors give data

red green blue

Population Protocol



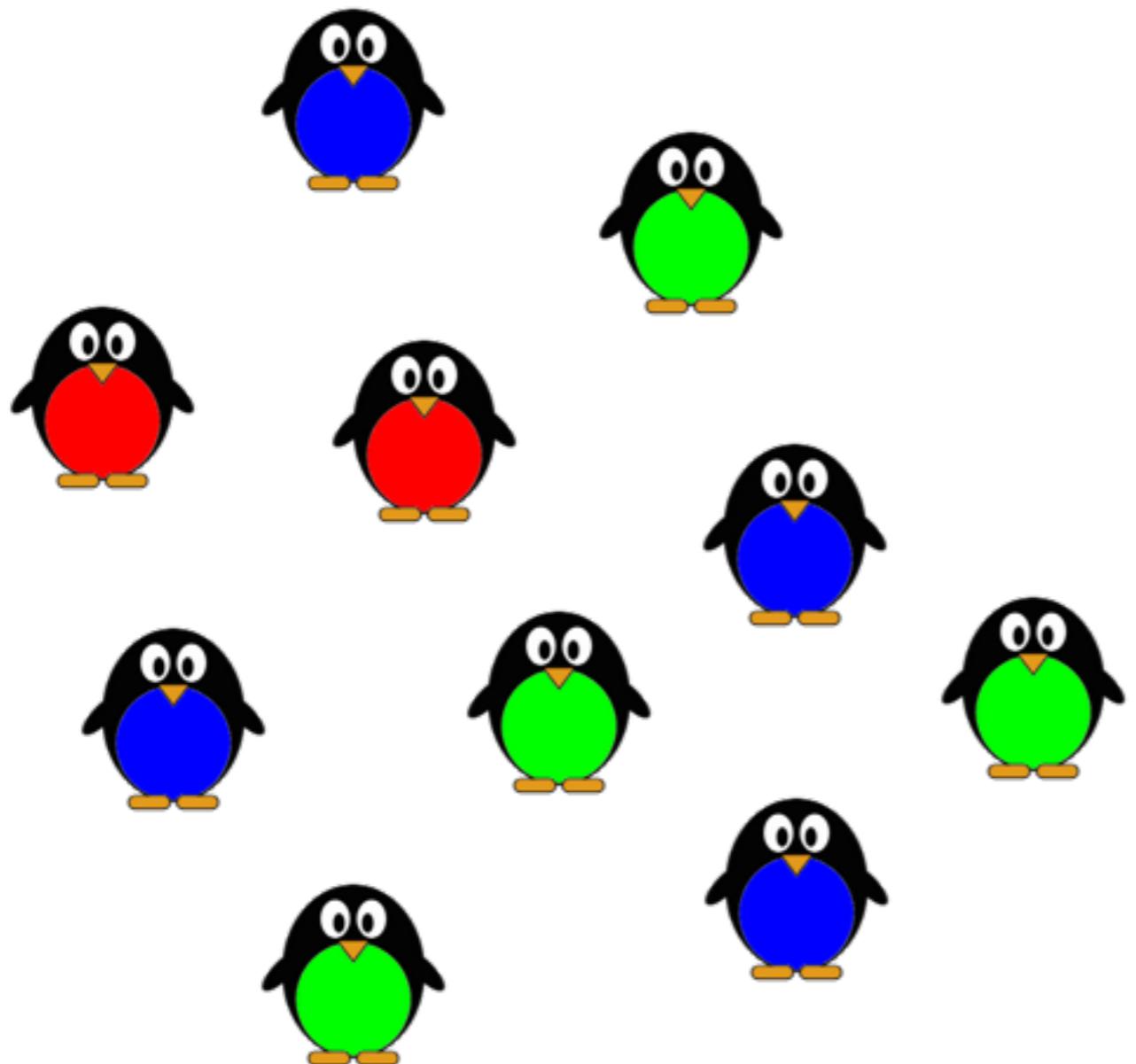
Initially: sensors give data

red green blue



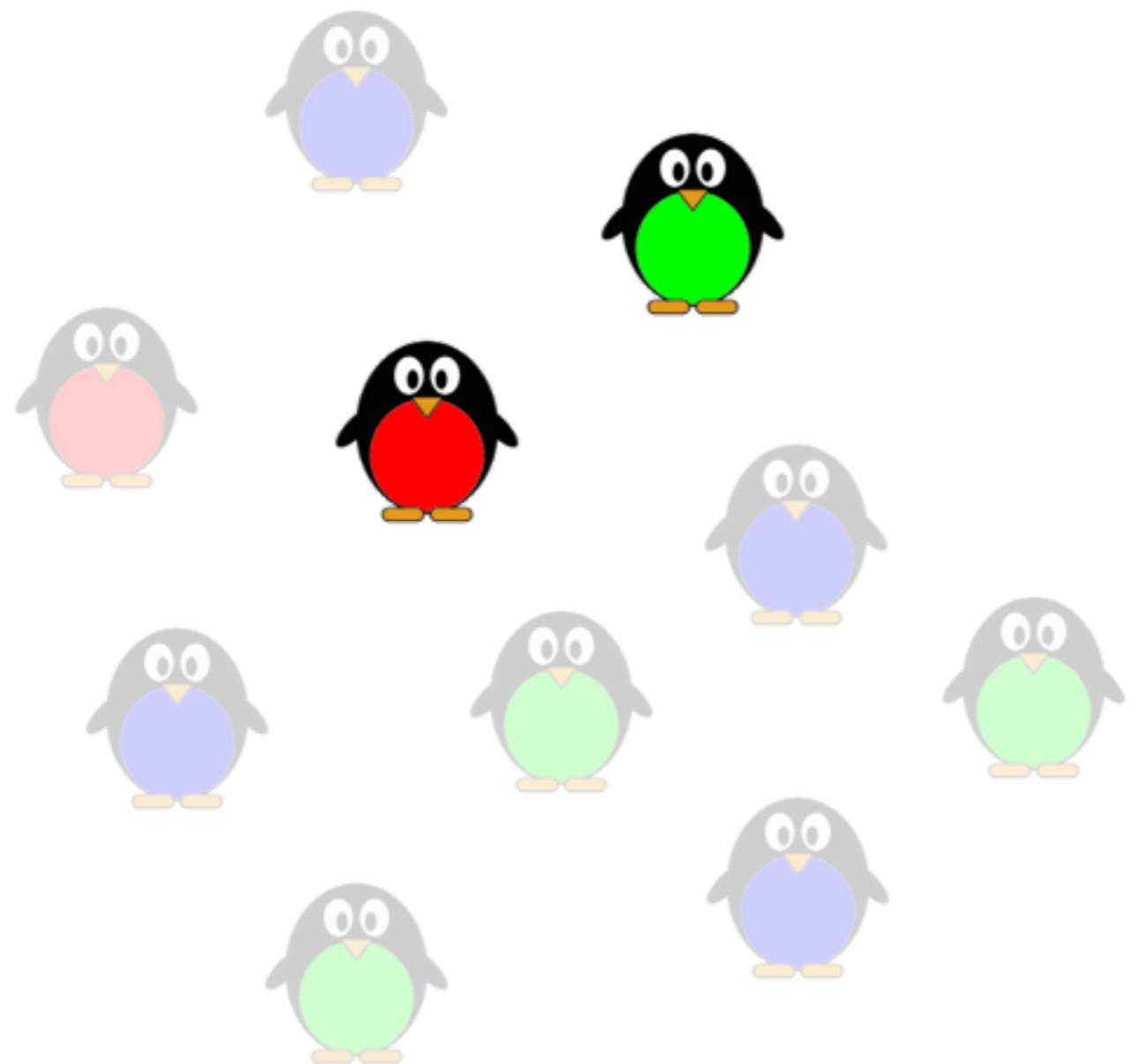
initial states

Population Protocol



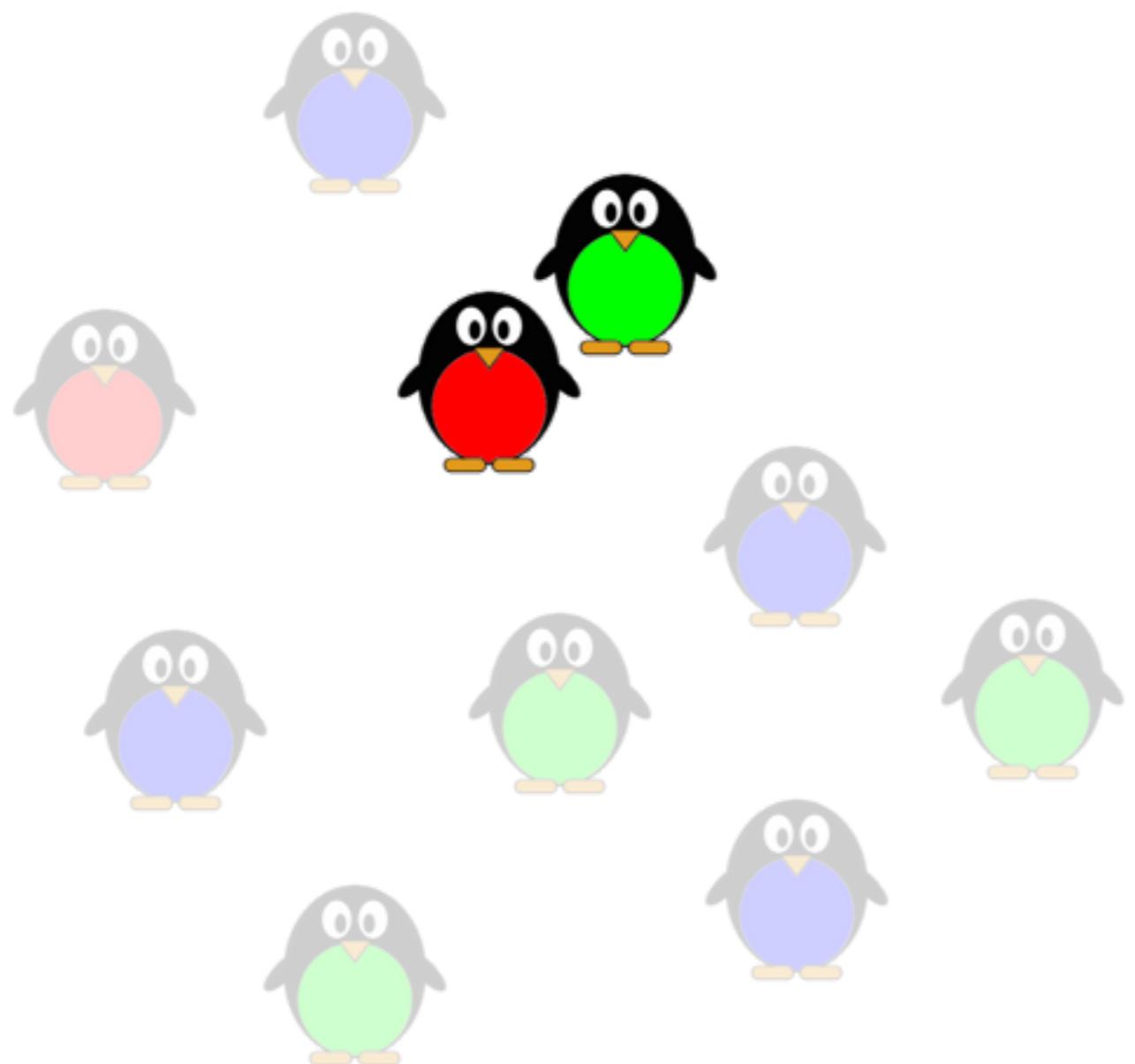
Agents move

Population Protocol



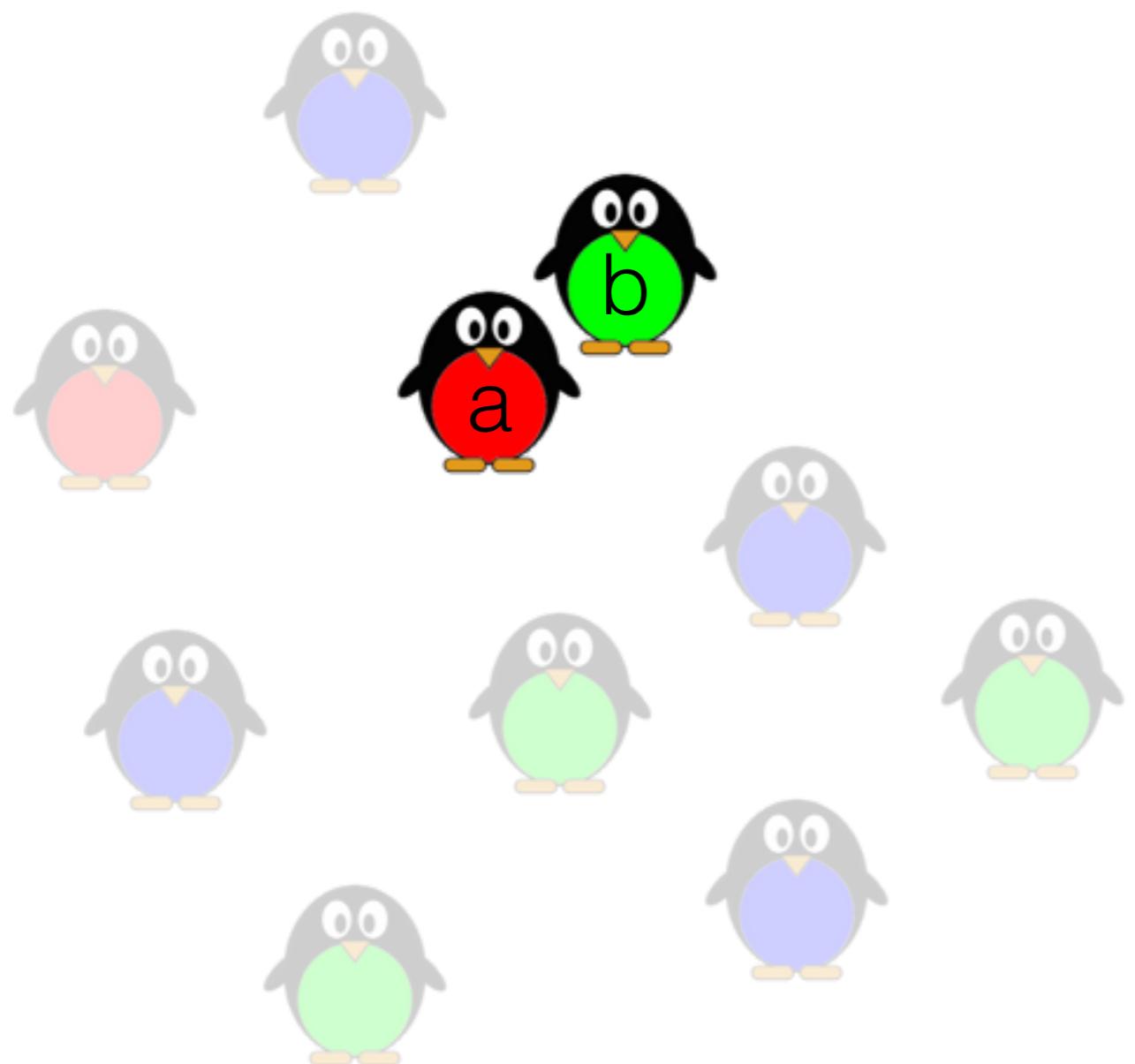
Agents move

Population Protocol



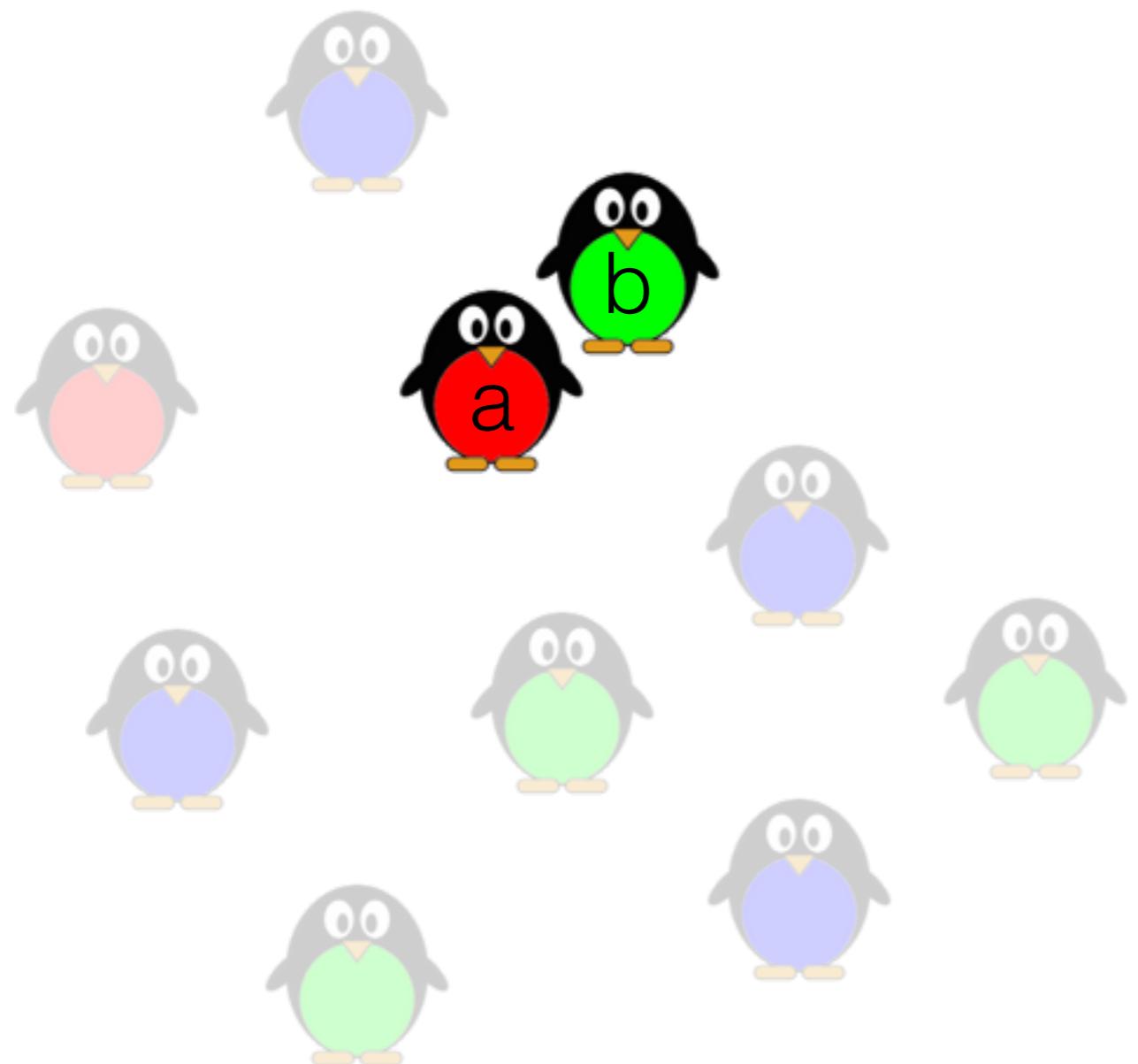
Agents move

Population Protocol



Agents move

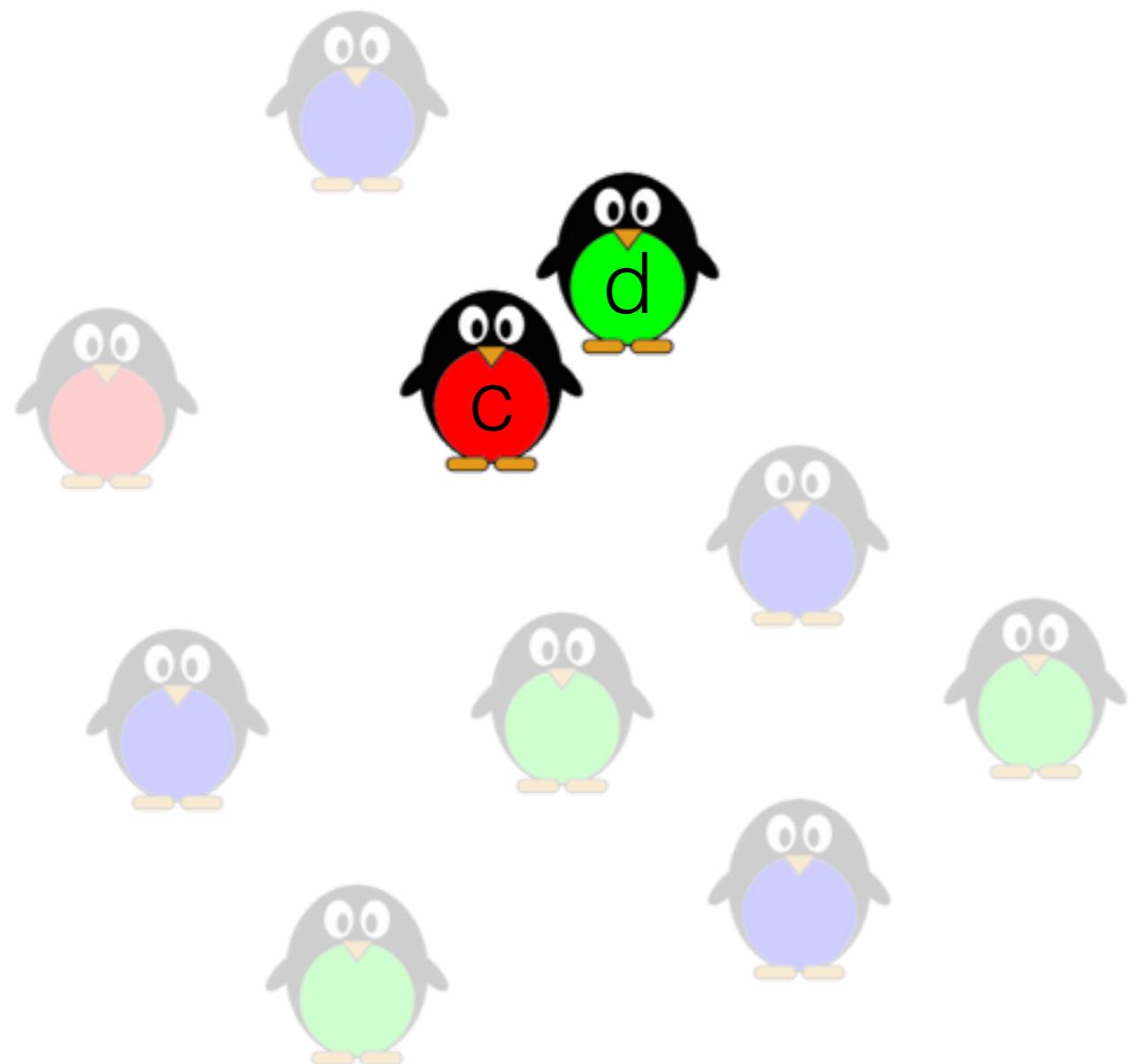
Population Protocol



Protocol rule



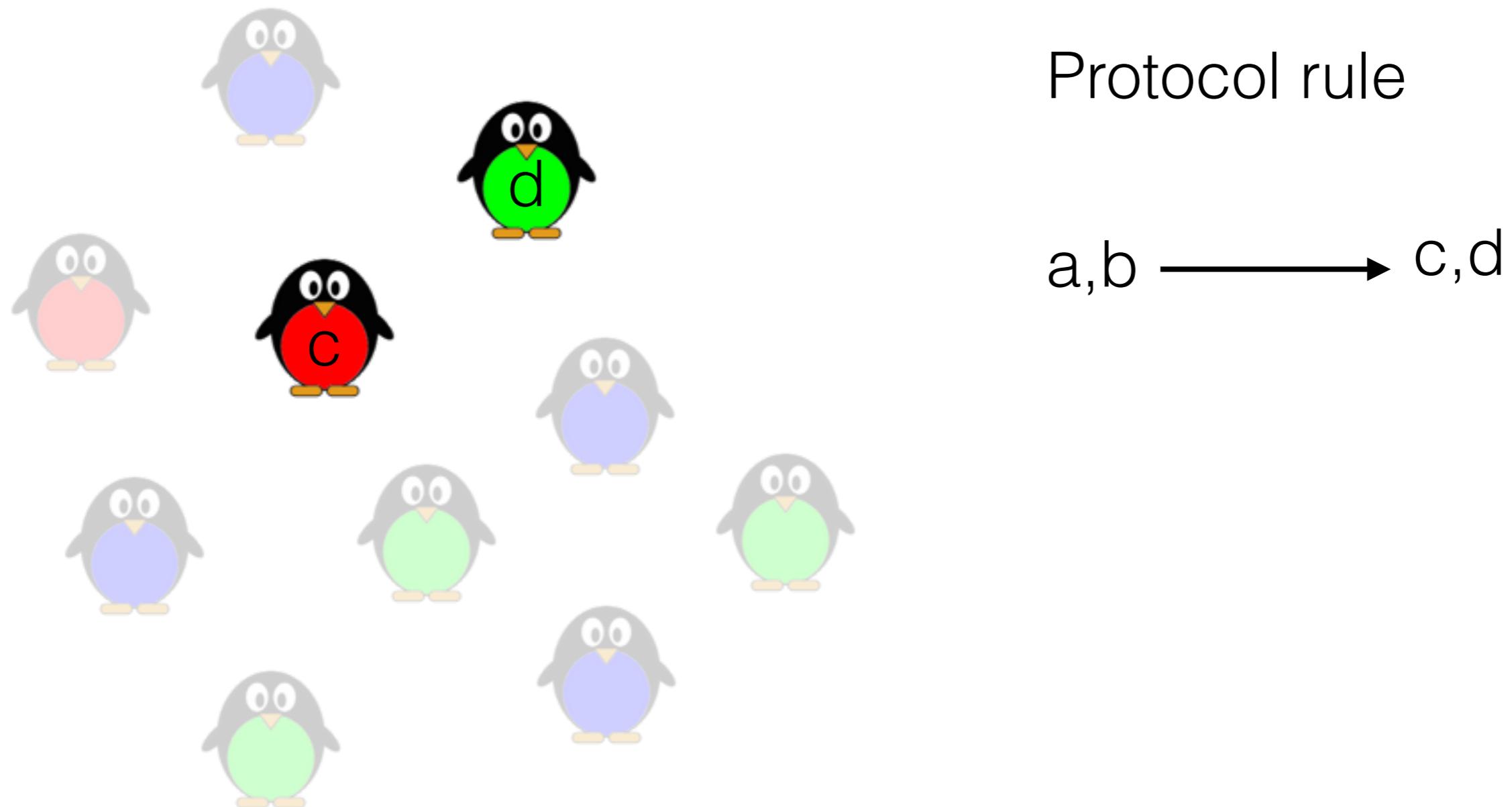
Population Protocol



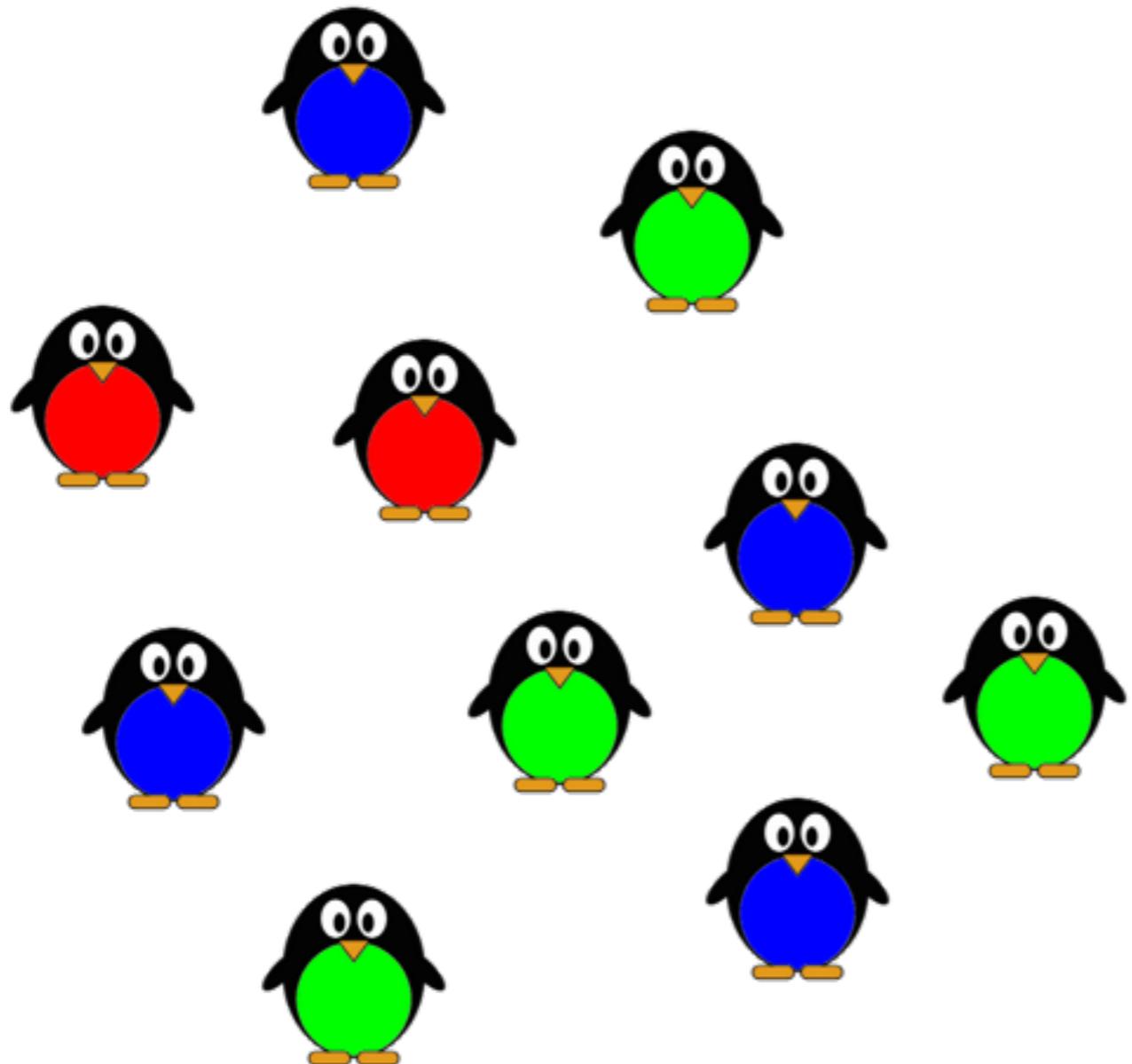
Protocol rule

a,b —————> c,d

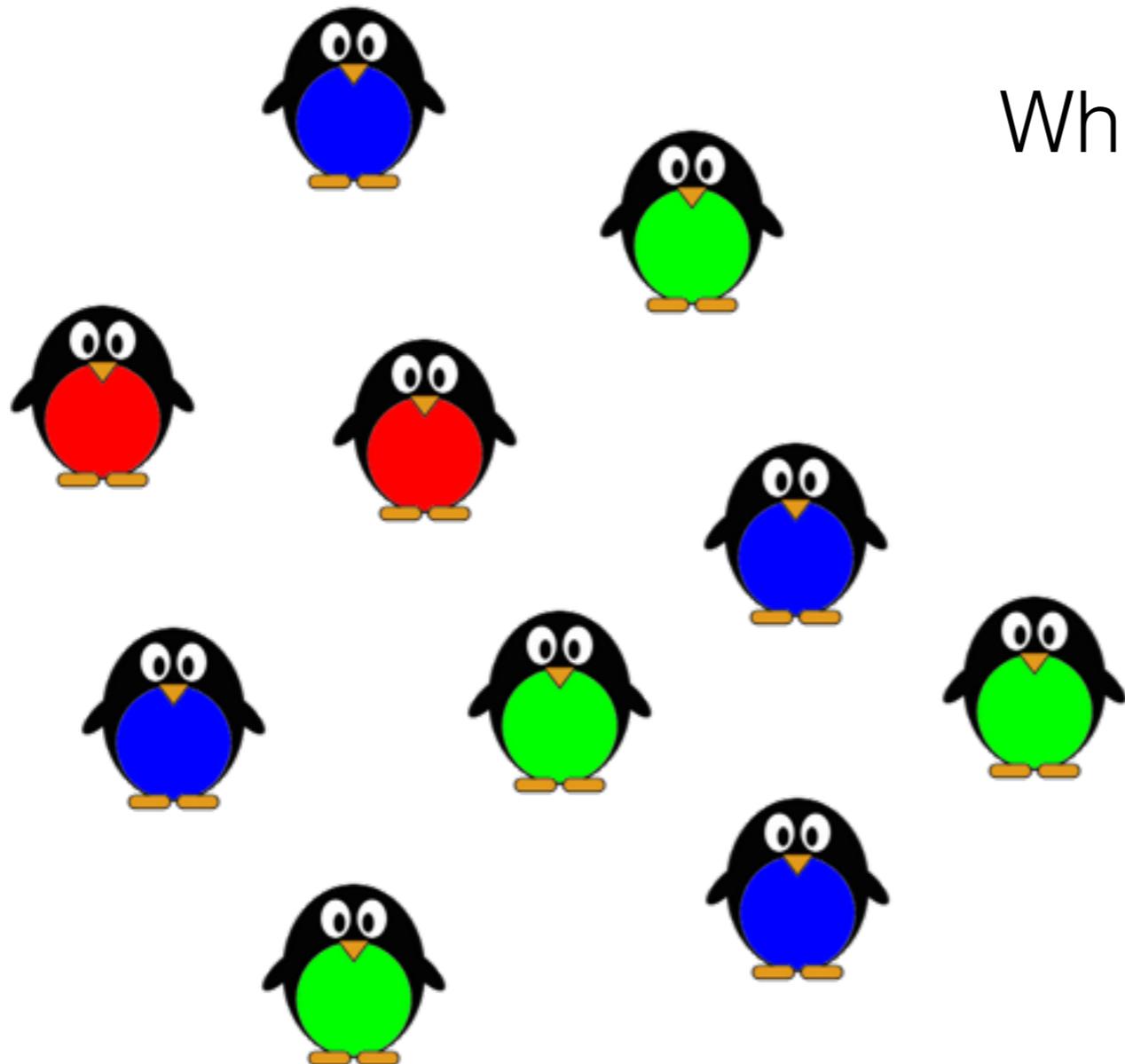
Population Protocol



What is computable ?

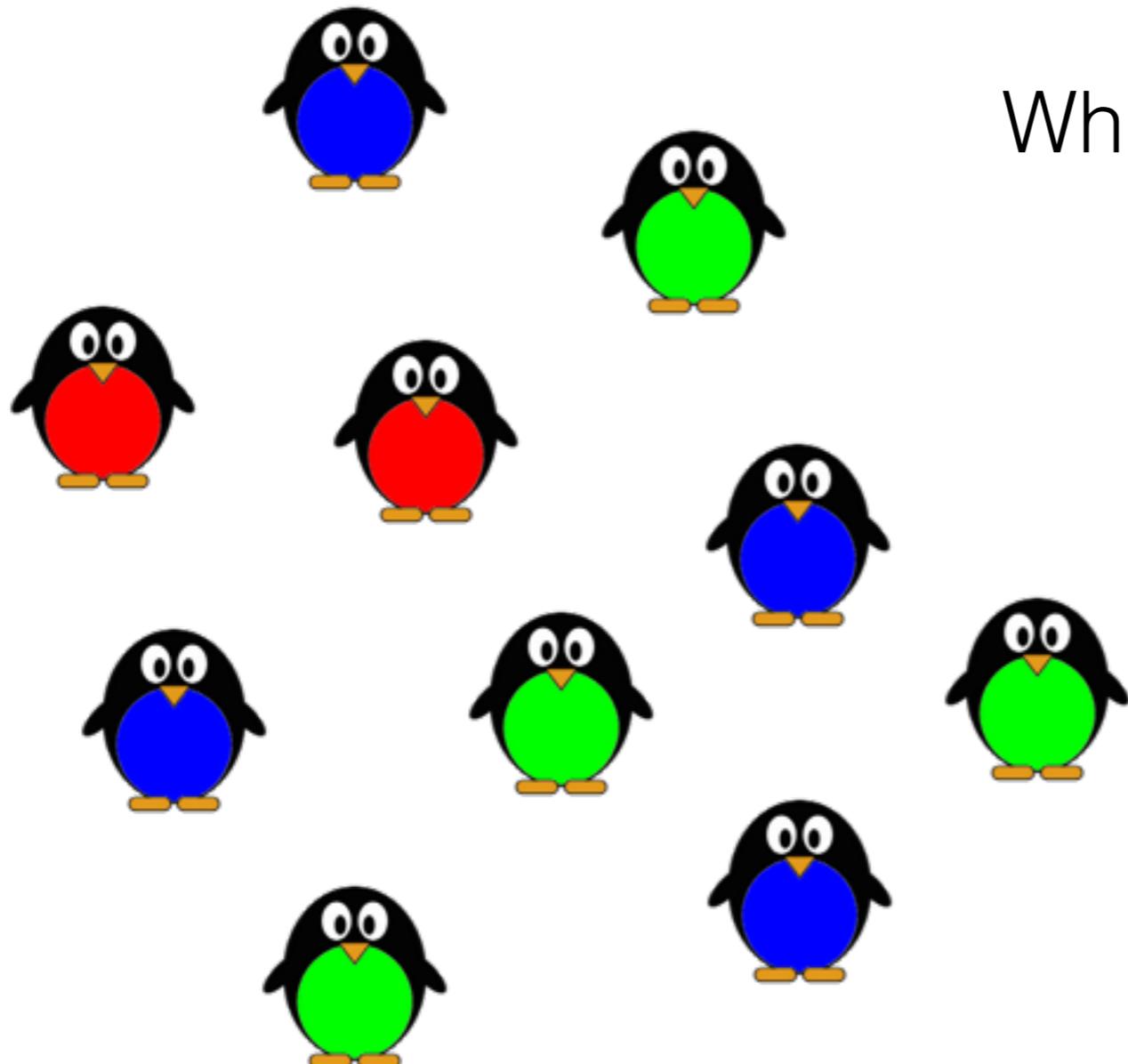


What is computable ?



What can the agents know about
the initial configuration ?

What is computable ?



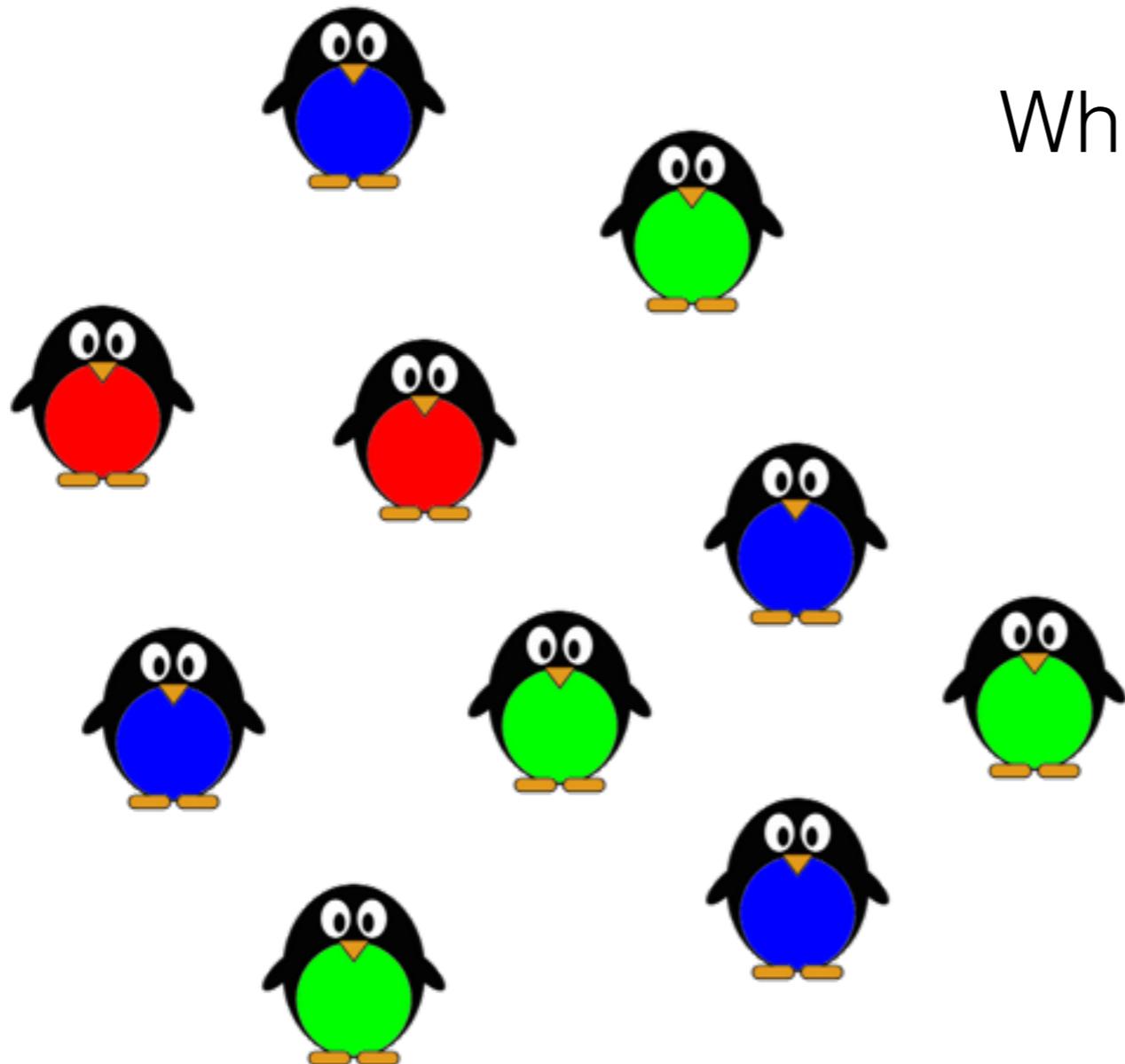
What can the agents know about
the initial configuration ?

No id



Only numbers of "species"

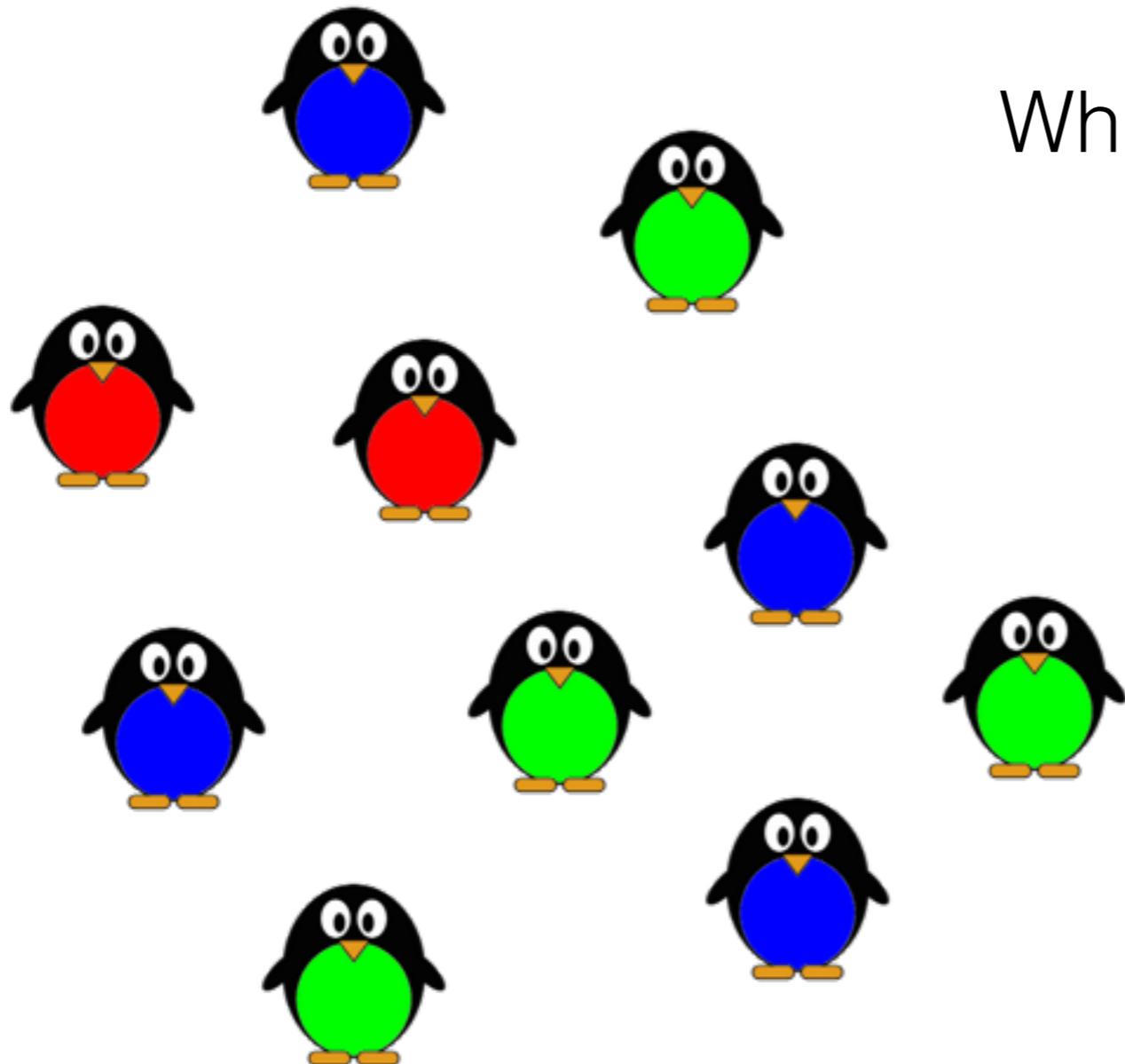
What is computable ?



What can the agents know about
the initial configuration ?

$$\#\text{green} \leq 4$$

What is computable ?

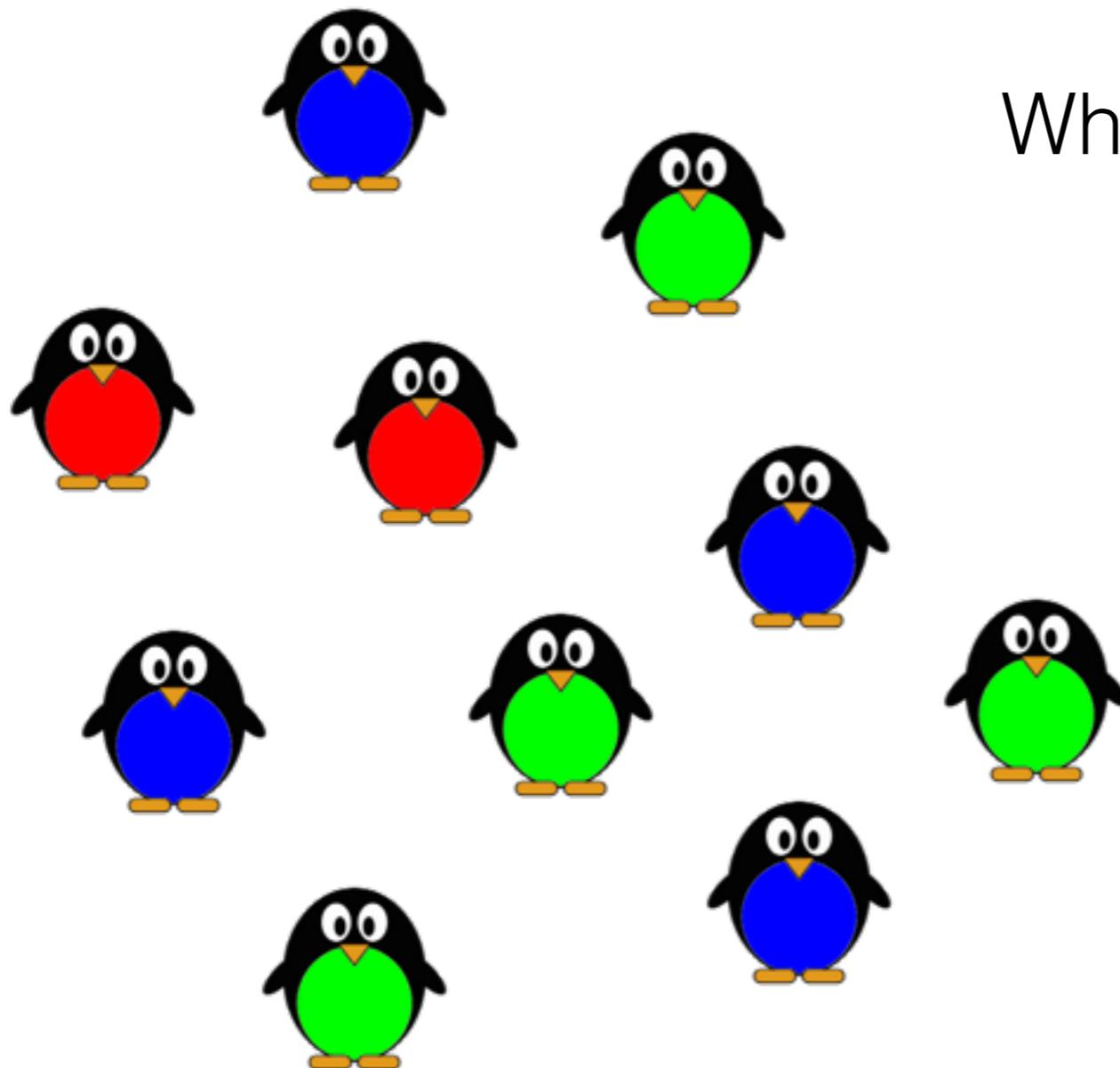


What can the agents know about
the initial configuration ?

$$\#\text{green} \leq 4$$

$$\#\text{green} \leq \#\text{blue}$$

What is computable ?



What can the agents know about
the initial configuration ?

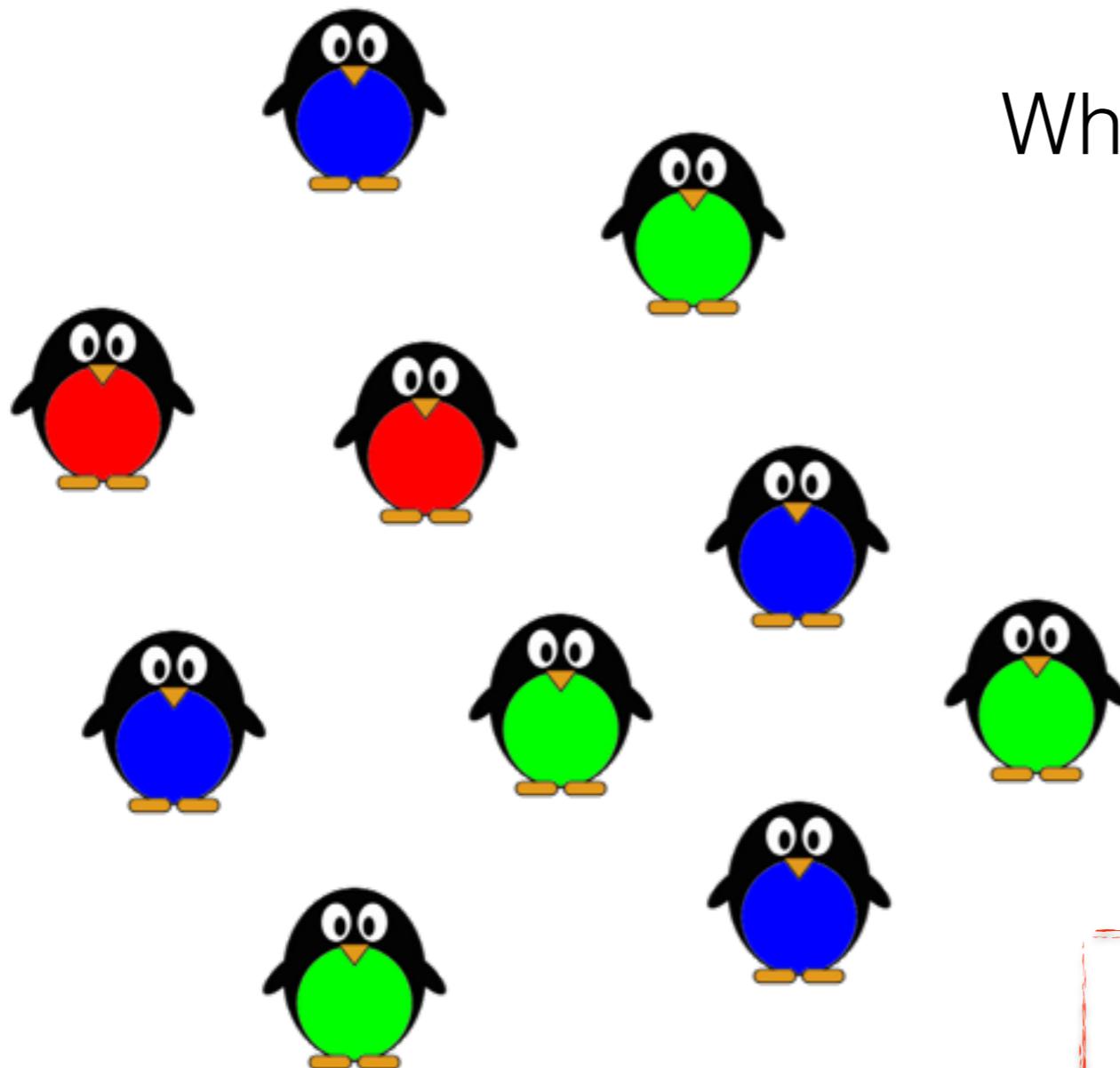
$$\#\text{green} \leq 4$$

$$\#\text{green} \leq \#\text{blue}$$

$$\#\text{green} \leq \#\text{blue} + 2\#\text{red}$$

$$\#\text{green} \leq \#\text{blue} \times \#\text{red}$$

What is computable ?



What can the agents know about
the initial configuration ?

$$\#\text{green} \leq 4$$

$$\#\text{green} \leq \#\text{blue}$$

$$\#\text{green} \leq \#\text{blue} + 2\#\text{red}$$

OK

$$\#\text{green} \leq \#\text{blue} \times \#\text{red}$$

NOT

$$X = \{\sigma_1, \dots, \sigma_k\}$$

the "species" (red,
green, blue, etc.)

$X = \{\sigma_1, \dots, \sigma_k\}$ the "species" (red,
green, blue, etc.) a_1, \dots, a_k, c integer coefficients

$X = \{\sigma_1, \dots, \sigma_k\}$ the "species" (red,
green, blue, etc.) a_1, \dots, a_k, c integer coefficients

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

$$X = \{\sigma_1, \dots, \sigma_k\}$$
 the "species" (red,
green, blue, etc.)
$$a_1, \dots, a_k, c$$
 integer coefficients

$$P : a_1 \cdot \#\sigma_1 + \dots + a_k \cdot \#\sigma_k < c$$



Number of agents with input σ_1

⋮



Number of agents with input σ_k

P computable

P computable

There exists a protocol A such that

P computable

There exists a protocol A such that

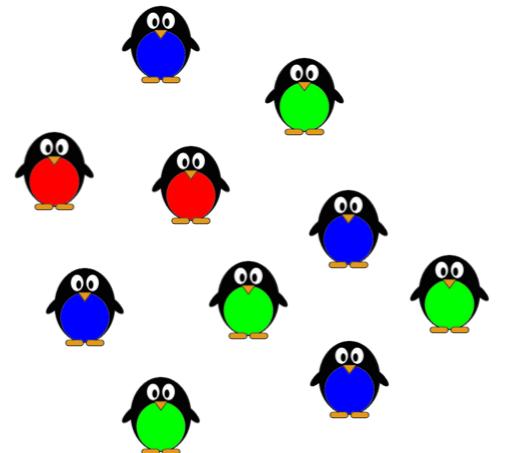
for any population size

P computable

There exists a protocol A such that

for any population size

for any input assignment



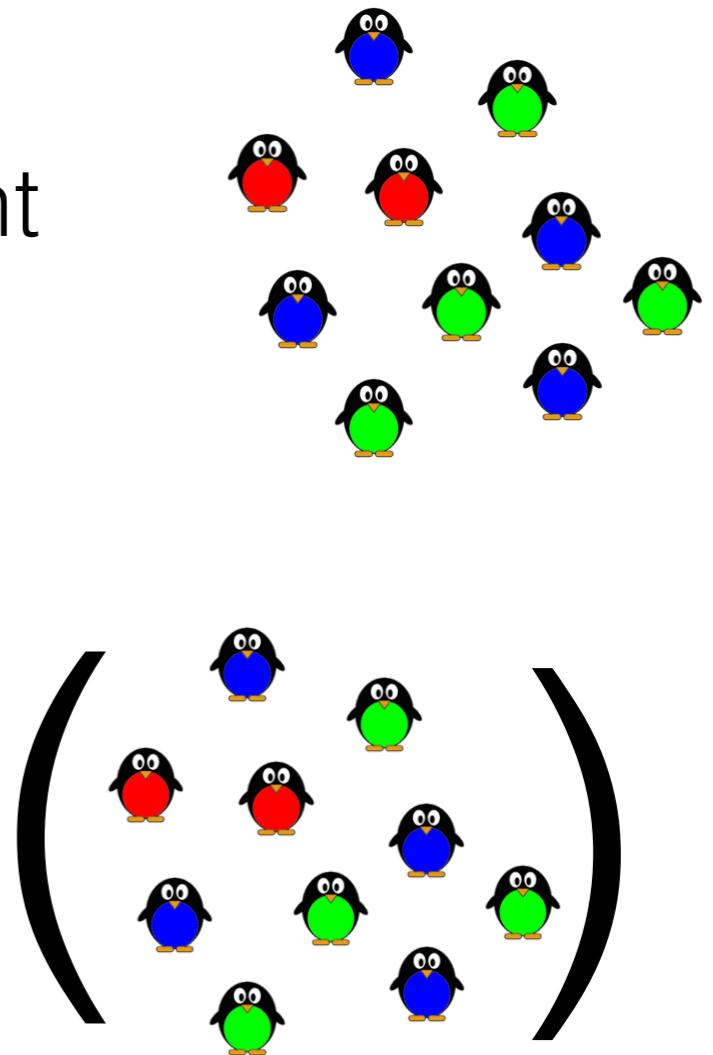
P computable

There exists a protocol A such that

for any population size

for any input assignment

eventually **all** agents output P



Idea

$$P : a_1 \cdot \# \sigma_1 + \cdots + a_k \cdot \# \sigma_k < c$$

Idea

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

- agent with input $\sigma_i \longrightarrow$ init. state with weight a_i

Idea

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

- agent with input $\sigma_i \longrightarrow$ init. state with weight a_i
- some leader collects the sum of weights

Idea

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

- agent with input $\sigma_i \longrightarrow$ init. state with weight a_i
- some leader collects the sum of weights
- leader can test the threshold

Idea

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

- agent with input $\sigma_i \longrightarrow$ init. state with weight a_i
 - some leader collects the sum of weights
 - leader can test the threshold
-
- A red curved arrow originates from the text "sum of weights" and points towards the text "leader can test the threshold". The word "counter field" is written in red, diagonal text next to the arrow.

Idea

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

- agent with input $\sigma_i \longrightarrow$ init. state with weight a_i
 - some leader collects the sum of weights
leader bit
 - leader can test the threshold
-
- The diagram illustrates the process of collecting weights. It shows three main components: 1) A blue oval labeled "leader" with a blue arrow pointing to it, representing the leader agent. 2) A red oval labeled "sum of weights" with a red arrow pointing to it, representing the total weight collected by the leader. 3) A red oval labeled "counter field" with a red curved arrow pointing to it, representing the mechanism for tracking the sum of weights. Arrows indicate the flow of information from the agents to the leader, and from the leader to the sum of weights and counter field.

$$P: a_1\cdot \#\sigma_1 + \cdots + a_k\cdot \#\sigma_k < c$$

$$\text{State space} \qquad \{0,1\}\times\{0,1\}\times\{-s,\ldots,s\}$$

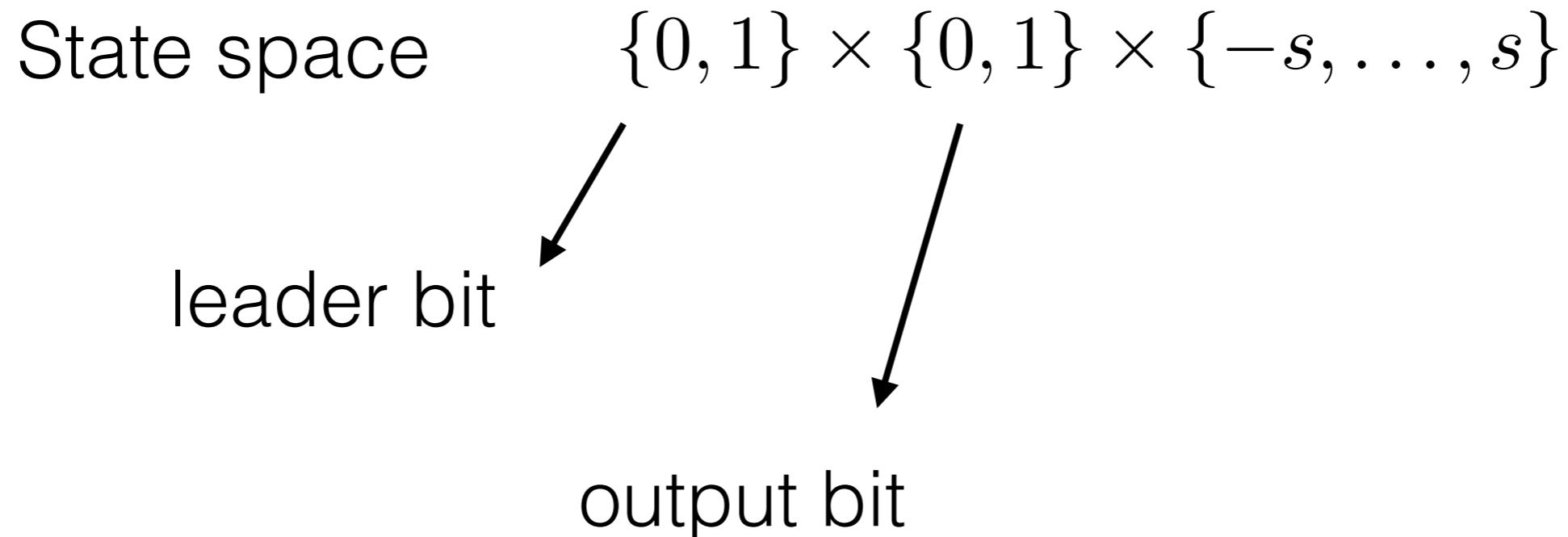
$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

State space $\{0, 1\} \times \{0, 1\} \times \{-s, \dots, s\}$

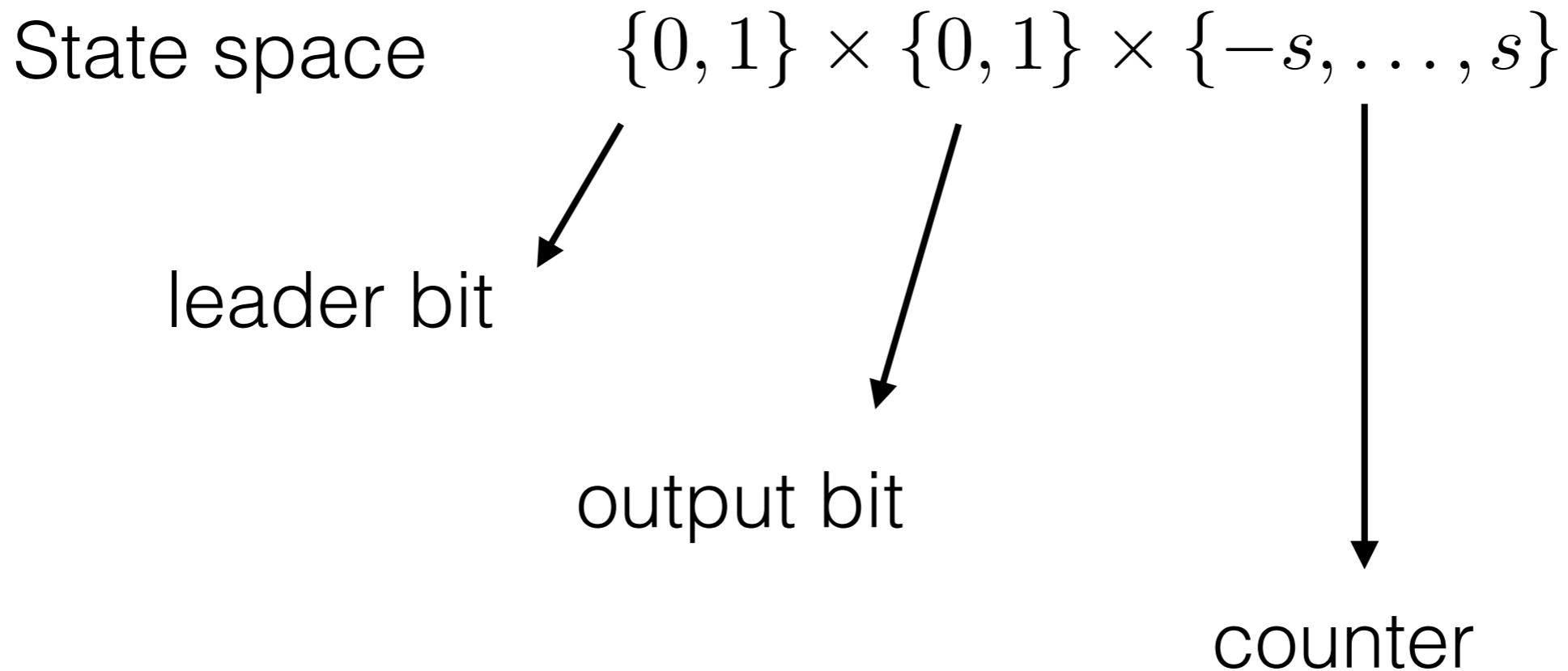
leader bit



$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$



$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$



$$s = \max\{|c| + 1, |a_1|, \dots, |a_k|\}$$

$$P : a_1 \cdot \# \sigma_1 + \cdots + a_k \cdot \# \sigma_k < c$$

$$\text{State space} \qquad \qquad \{0,1\} \times \{0,1\} \times \{-s,\dots,s\}$$

$$\text{initial state: } \sigma_i \longrightarrow (1,0,a_i)$$

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

State space $\{0, 1\} \times \{0, 1\} \times \{-s, \dots, s\}$

initial state : $\sigma_i \longrightarrow (1, 0, a_i)$

rules ($l \neq l'$) $(l, \cdot, u) \longrightarrow (1, b(u, u'), q(u, u'))$

$(l', \cdot, u') \longrightarrow (0, b(u, u'), r(u, u'))$

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

State space $\{0, 1\} \times \{0, 1\} \times \{-s, \dots, s\}$

initial state : $\sigma_i \longrightarrow (1, 0, a_i)$

rules ($l \neq l'$) $(l, \cdot, u) \longrightarrow (1, b(u, u'), q(u, u'))$

$(l', \cdot, u') \longrightarrow (0, b(u, u'), r(u, u'))$

$$q(u, u') = \max\{-s, \min\{s, u + u'\}\}$$

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

State space $\{0, 1\} \times \{0, 1\} \times \{-s, \dots, s\}$

initial state : $\sigma_i \longrightarrow (1, 0, a_i)$

rules ($l \neq l'$) $(l, \cdot, u) \longrightarrow (1, b(u, u'), q(u, u'))$

$(l', \cdot, u') \longrightarrow (0, b(u, u'), r(u, u'))$

truncated sum

$$q(u, u') = \max\{-s, \min\{s, u + u'\}\}$$

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

State space $\{0, 1\} \times \{0, 1\} \times \{-s, \dots, s\}$

initial state : $\sigma_i \longrightarrow (1, 0, a_i)$

rules ($l \neq l'$) $(l, \cdot, u) \longrightarrow (1, b(u, u'), q(u, u'))$
 $(l', \cdot, u') \longrightarrow (0, b(u, u'), r(u, u'))$

truncated sum
 $q(u, u') = \max\{-s, \min\{s, u + u'\}\}$

$r(u, u') = u + u' - q(u, u')$

remainder

$$P : a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

State space	$\{0, 1\} \times \{0, 1\} \times \{-s, \dots, s\}$
-------------	--

initial state :	$\sigma_i \longrightarrow (1, 0, a_i)$
-----------------	--

rules ($l \neq l'$)	$(l, \cdot, u) \longrightarrow (1, b(u, u'), q(u, u'))$
	$(l', \cdot, u') \longrightarrow (0, b(u, u'), r(u, u'))$

test threshold

$$b(u, u') = \begin{cases} 1 & \text{if } q(u, u') < c \\ 0 & \text{otherwise} \end{cases}$$

Proof strategy

A. Eventually a single leader

B. Eventually, the leader collects the value

$$\max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\}$$

C. Eventually, the agents produce correct outputs

Proof strategy

A. Eventually a single leader

B. Eventually, the leader collects the value

$$\max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\}$$

C. Eventually, the agents produce correct outputs

Proof strategy

A. Eventually a single leader

$$\begin{array}{ccc} (l, \cdot, u) & & (1, b(u, u'), q(u, u')) \\ (l \neq l') \quad \longrightarrow & & (0, b(u, u'), r(u, u')) \\ (l', \cdot, u') & & \end{array}$$

Proof strategy

leader, leader

leader, non-leader

non-leader, leader



A. Eventually a single leader

leader, non-leader

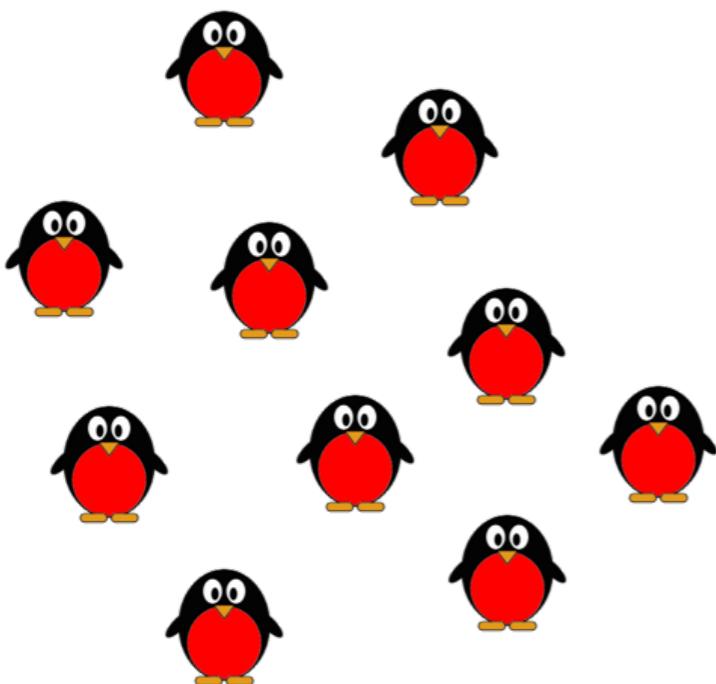
Proof strategy

leader, leader
leader, non-leader
non-leader, leader

→

leader, non-leader

A. Eventually a single leader



init. all leaders

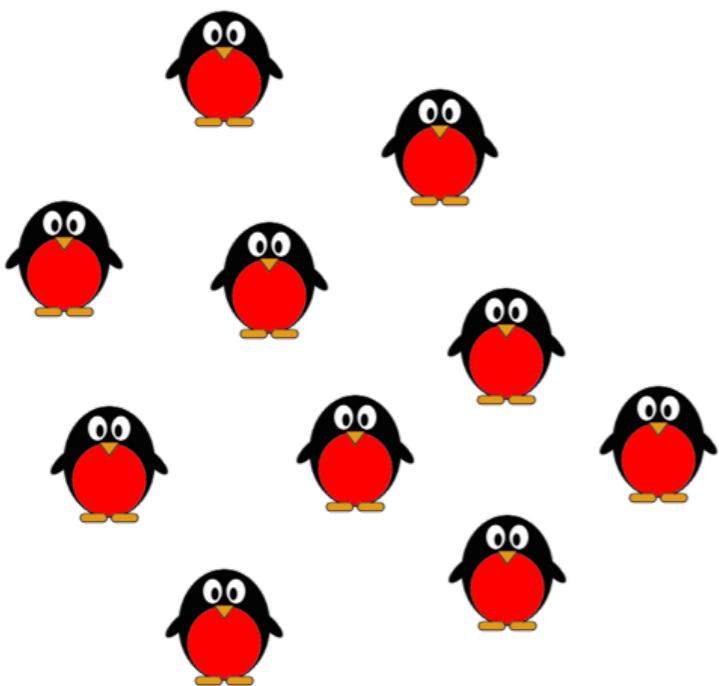
Proof strategy

leader, leader
leader, non-leader
non-leader, leader

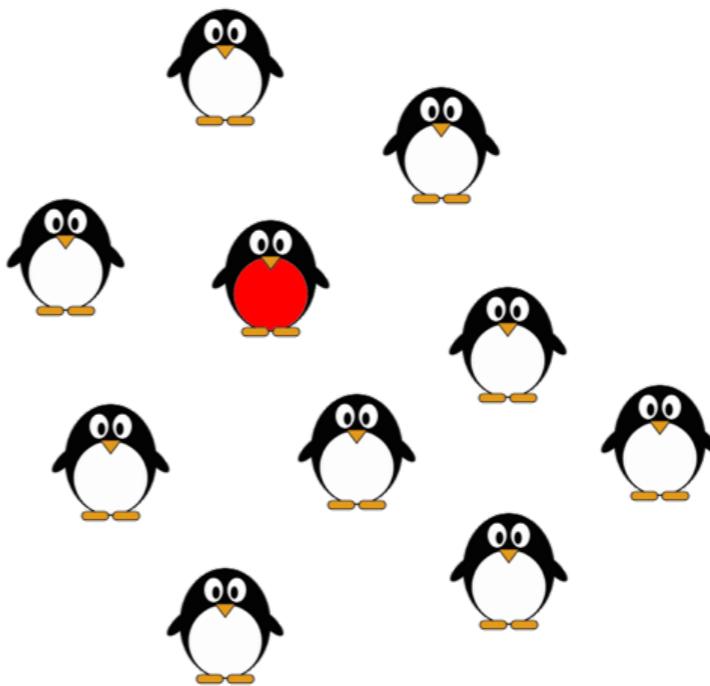
A. Eventually a single leader



leader, non-leader



init. all leaders

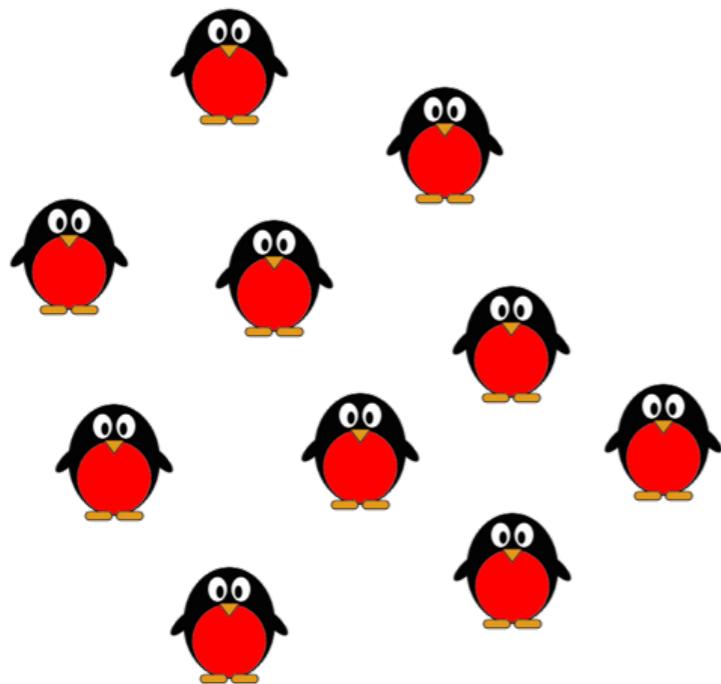


one survivor

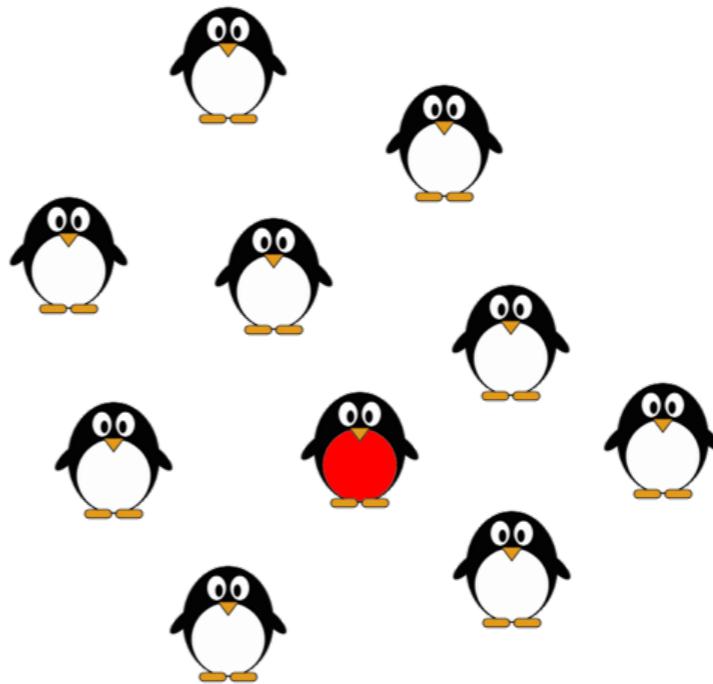
Proof strategy

leader, leader
leader, non-leader
non-leader, leader

A. Eventually a single leader



leader, non-leader



init. all leaders

one leader (which may change)

Proof strategy

A. Eventually ~~a single leader~~

B. Eventually, the leader collects the value

$$\max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\}$$

C. Eventually, the agents produce correct outputs

Proof strategy

B. Eventually, the leader ...

$$\begin{array}{ccc} (l, \cdot, u) & & (1, b(u, u'), q(u, u')) \\ (l \neq l') & \xrightarrow{\hspace{1cm}} & \\ (l', \cdot, u') & & (0, b(u, u'), r(u, u')) \end{array}$$

$$q(u, u') = \max\{-s, \min\{s, u + u'\}\}$$

$$r(u, u') = u + u' - q(u, u')$$

Proof strategy

B. Eventually, the leader ...

$$\begin{array}{ccc} (l, \cdot, u) & & (1, b(u, u'), q(u, u')) \\ (l \neq l') & \xrightarrow{\hspace{1cm}} & \\ (l', \cdot, u') & & (0, b(u, u'), r(u, u')) \end{array}$$

$$q(u, u') = \max\{-s, \min\{s, u + u'\}\}$$

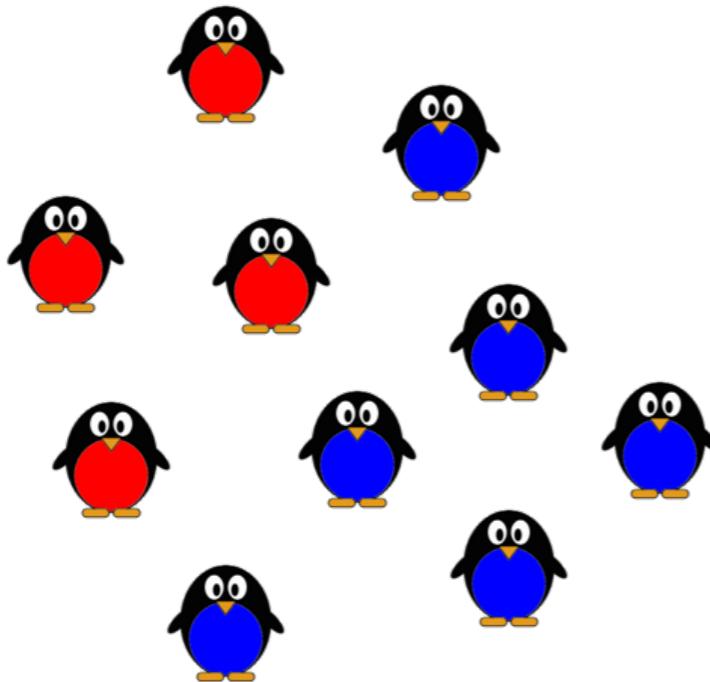
$$r(u, u') = u + u' - q(u, u')$$

$$q(u, u') + r(u, u') = u + u'$$

Proof strategy

B. Eventually, the leader ...

$$q(u, u') + r(u, u') = u + u'$$

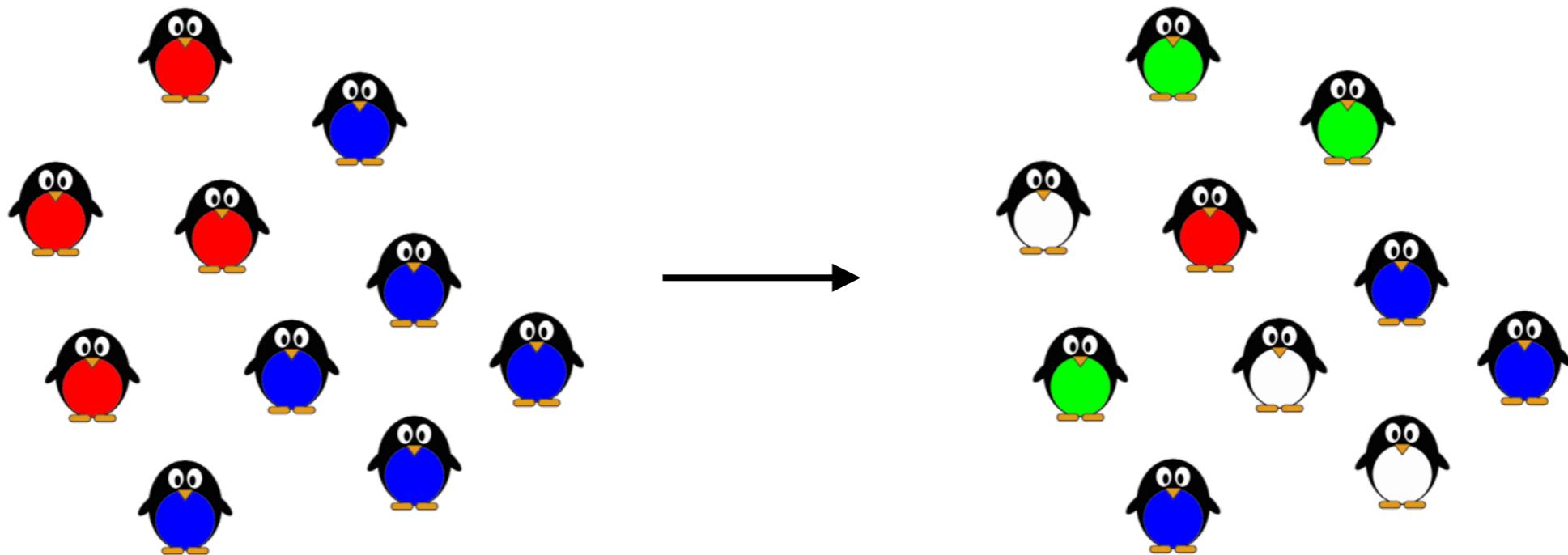


$$\sum a_i \cdot \#\sigma_i$$

Proof strategy

B. Eventually, the leader ...

$$q(u, u') + r(u, u') = u + u'$$

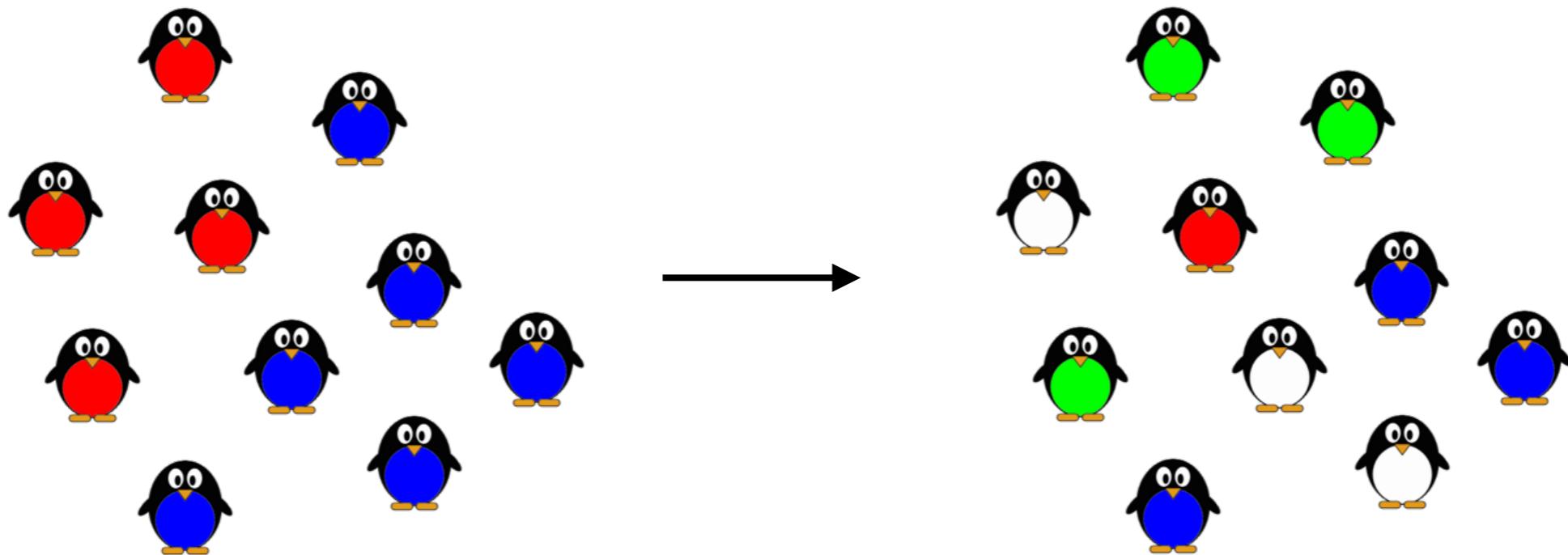


$$\sum a_i \cdot \#\sigma_i$$

Proof strategy

B. Eventually, the leader ...

$$q(u, u') + r(u, u') = u + u'$$



$$\sum a_i \cdot \#\sigma_i = \sum u_i$$

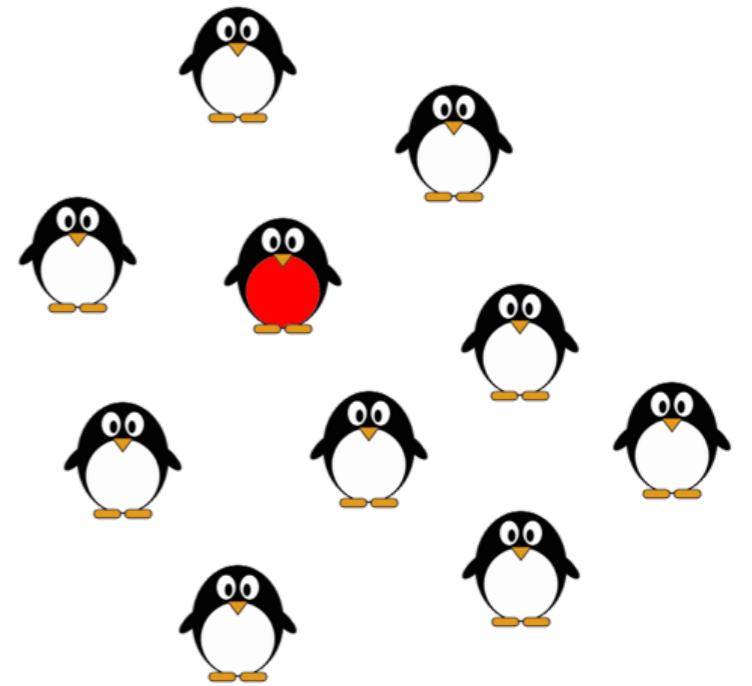
counter of agent i

Proof strategy

B. Eventually, the leader ...

Focus on suffix of execution

in each config C, a unique leader L



Proof strategy

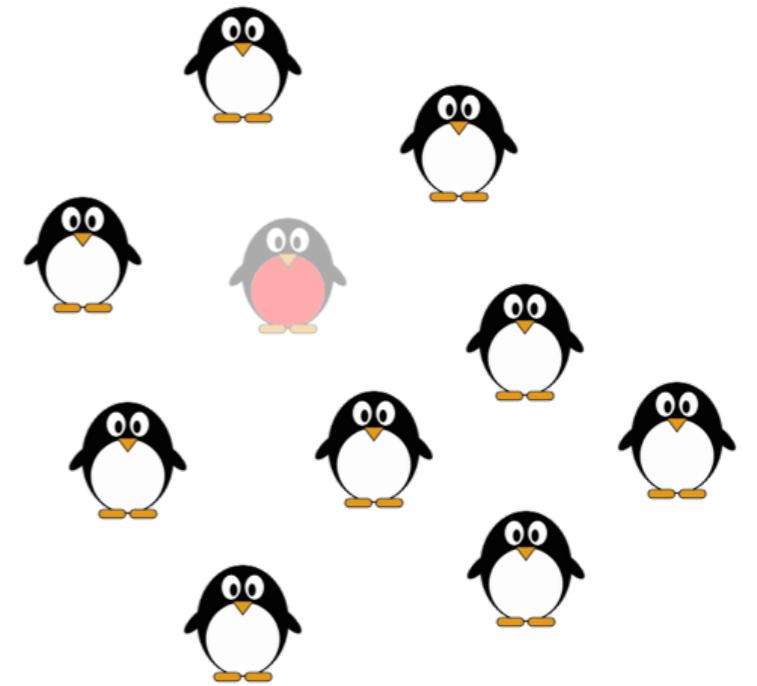
B. Eventually, the leader ...

Focus on suffix of execution

in each config C, a unique leader L

$$p(C) = \sum_{j \neq L} |u_j(C)|$$

measures the "mass" of the non-leaders



Proof strategy

B. Eventually, the leader ...

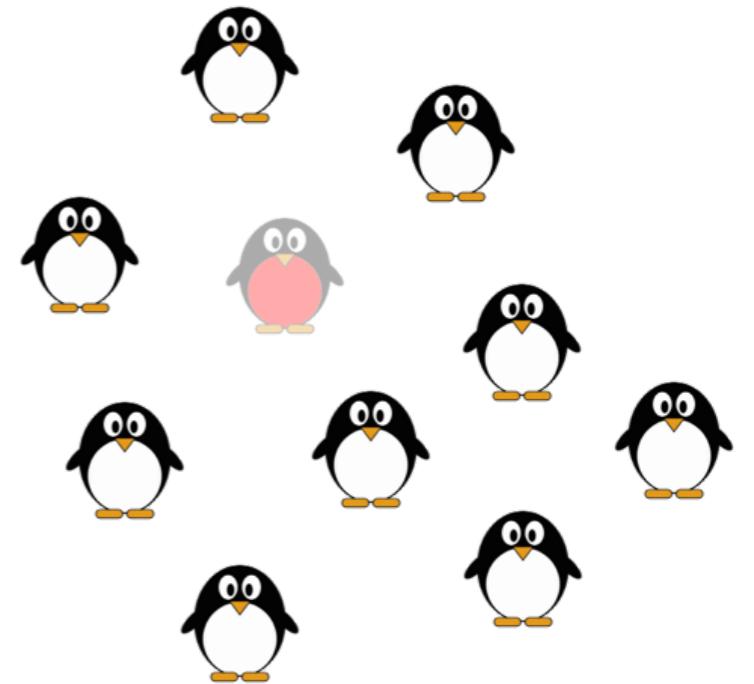
Focus on suffix of execution

in each config C, a unique leader L

$$p(C) = \sum_{j \neq L} |u_j(C)|$$

measures the "mass" of the non-leaders

show that p decreases
& eventually cst



Proof strategy

B. Eventually, the leader ...

$$u_L \rightarrow \max\{-s, \min\{s, u_L + u_j\}\}$$

$$u_j \quad u_L + u_j - \text{above}$$

$$p(C) = \sum_{j \neq L} |u_j(C)|$$

case 1: $-s \leq u_L \leq s$ and $u_j \geq 0$

$$u_L \rightarrow \min\{u_L + u_j, s\} = u_L + \min\{u_j, s - u_L\}$$

$$u_j \rightarrow u_j - \min\{u_j, s - u_L\}$$

Proof strategy

B. Eventually, the leader ...

$$\begin{array}{lcl} u_L & \rightarrow & \max\{-s, \min\{s, u_L + u_j\}\} \\ u_j & & u_L + u_j - \text{above} \end{array}$$

$$p(C) = \sum_{j \neq L} |u_j(C)|$$

case 1: $-s \leq u_L \leq s$ and $u_j \geq 0$

$$\begin{array}{lcl} u_L & \rightarrow & \min\{u_L + u_j, s\} = u_L + \min\{u_j, s - u_L\} \\ u_j & \rightarrow & u_j - \min\{u_j, s - u_L\} \end{array}$$

$|u_j|$ does not increase

p does not increase

Proof strategy

B. Eventually, the leader ...

$$u_L \rightarrow \max\{-s, \min\{s, u_L + u_j\}\}$$

$$u_j \quad u_L + u_j - \text{above}$$

$$p(C) = \sum_{j \neq L} |u_j(C)|$$

case 2: $-s \leq u_L \leq s$ and $u_j \leq 0$

$$u_L \quad \max\{u_L + u_j, -s\} = u_L - \min\{-u_j, s + u_L\}$$

$$u_j \quad \rightarrow \quad u_j + \min\{-u_j, s + u_L\}$$

Proof strategy

B. Eventually, the leader ...

$$\begin{array}{ll} u_L & \rightarrow \max\{-s, \min\{s, u_L + u_j\}\} \\ u_j & \quad u_L + u_j - \text{above} \end{array}$$

$$p(C) = \sum_{j \neq L} |u_j(C)|$$

case 2: $-s \leq u_L \leq s$ and $u_j \leq 0$

$$\begin{array}{ll} u_L & \max\{u_L + u_j, -s\} = u_L - \min\{-u_j, s + u_L\} \\ u_j & \rightarrow \boxed{u_j + \min\{-u_j, s + u_L\}} \end{array}$$

$|u_j|$ does not increase

p does not increase

Proof strategy

B. Eventually, the leader ...

Thus, eventually, p is constant

Proof strategy

B. Eventually, the leader ...

Thus, eventually, p is constant

Analyzing the previous equations

Proof strategy

B. Eventually, the leader ...

Thus, eventually, p is constant

Analyzing the previous equations

It is impossible from C to decrease p

iff

Proof strategy

B. Eventually, the leader ...

Thus, eventually, p is constant

Analyzing the previous equations

It is impossible from C to decrease p

iff

$$p = 0$$

or $u_L = s$ and $\forall j \neq L, u_j \geq 0$

or $u_L = -s$ and $\forall j \neq L, u_j \leq 0$

Proof strategy

B. Eventually, the leader ...

Thus, eventually, p is constant

Analyzing the previous equations

It is impossible from C to decrease p

iff

In all cases

$$\begin{aligned} p &= 0 \\ u_L &= \max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\} \end{aligned}$$

or

$$u_L = s \quad \text{and} \quad \forall j \neq L, u_j \geq 0$$

or

$$u_L = -s \quad \text{and} \quad \forall j \neq L, u_j \leq 0$$

Proof strategy

A. Eventually ~~a single leader~~

B. Eventually, the leader collects the value

$$\max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\}$$

C. Eventually, the agents produce correct outputs

Proof strategy

C. Eventually, correct outputs

$$u_L = \max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\}$$

Proof strategy

C. Eventually, correct outputs

$$u_L = \max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\}$$

If $\sum a_i \cdot \#\sigma_i < c$ then $u_L = \begin{cases} \sum a_i \cdot \#\sigma_i \\ -s \end{cases}$

If $\sum a_i \cdot \#\sigma_i \geq c$ then $u_L = \begin{cases} \sum a_i \cdot \#\sigma_i \\ s \end{cases}$

Proof strategy

C. Eventually, correct outputs

$$u_L = \max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\}$$

If $\sum a_i \cdot \#\sigma_i < c$ then $u_L = \begin{cases} \sum a_i \cdot \#\sigma_i \\ -s \end{cases}$

If $\sum a_i \cdot \#\sigma_i \geq c$ then $u_L = \begin{cases} \sum a_i \cdot \#\sigma_i \\ s \end{cases}$

Leader gets correct output

Proof strategy

C. Eventually, correct outputs

$$u_L = \max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\}$$

If $\sum a_i \cdot \#\sigma_i < c$ then $u_L = \begin{cases} \sum a_i \cdot \#\sigma_i \\ -s \end{cases}$

If $\sum a_i \cdot \#\sigma_i \geq c$ then $u_L = \begin{cases} \sum a_i \cdot \#\sigma_i \\ s \end{cases}$

Leader gets correct output

Others get correct output on meeting the leader

Proof strategy

C. Eventually, correct outputs

$$u_L = \max\{-s, \min\{s, \sum a_i \cdot \#\sigma_i\}\}$$

If $\sum a_i \cdot \#\sigma_i < c$ then $u_L = \begin{cases} \sum a_i \cdot \#\sigma_i \\ -s \end{cases}$

If $\sum a_i \cdot \#\sigma_i \geq c$ then $u_L = \begin{cases} \sum a_i \cdot \#\sigma_i \\ s \end{cases}$

EQP

Leader gets correct output

Others get correct output on meeting the leader

$$a_1\cdot \#\sigma_1+\cdots+a_k\cdot \#\sigma_k < c$$

$$a_1\cdot \#\sigma_1+\cdots+a_k\cdot \#\sigma_k < c$$

$$a_1\cdot \#\sigma_1+\cdots+a_k\cdot \#\sigma_k = c \mod m$$

$$a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

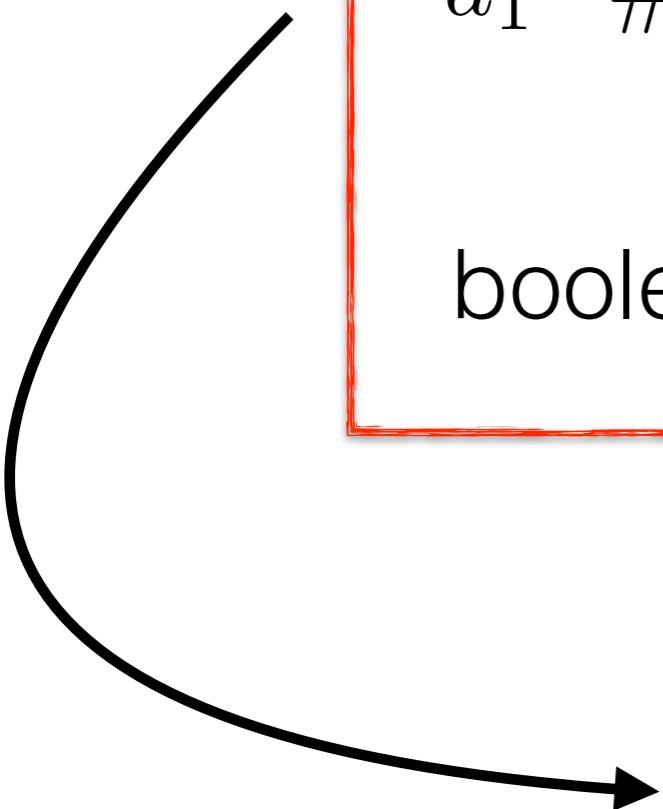
$$a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k = c \mod m$$

boolean combinations

$$a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

$$a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k = c \pmod{m}$$

boolean combinations

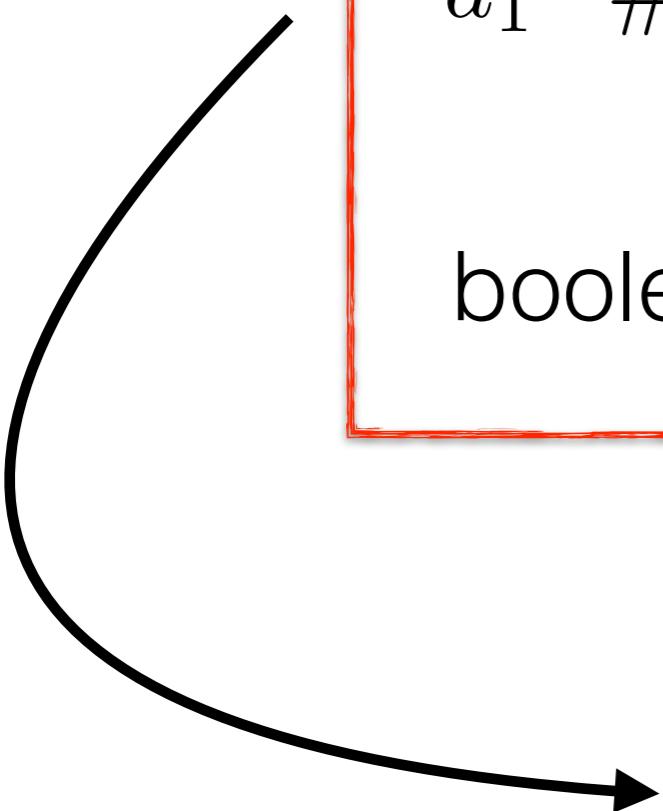


Presburger arithmetics

$$a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k < c$$

$$a_1 \cdot \#\sigma_1 + \cdots + a_k \cdot \#\sigma_k = c \pmod{m}$$

boolean combinations



Presburger arithmetics

(beware: integer coefficients)