Algorithmes et protocoles de population :

Que peux-t-on calculer avec une population d'oiseaux ?

Johanne Cohen

PRiSM/CNRS, Versailles, France.

Comment étudier un réseau mobile ?

Première Approche :

1. Choisir son modèle et son problème à résoudre ;
2. Réaliser des jolis algorithmes ;
3. Etudier les performances des algorithmes selon des scénarii (théoriquement ou pratiquement)

Deuxième Approche : [Angluin et al DISC 2004]

1. Définir des modèles TRÈS simples
2. Définir ce qu'on peut calculer dans ces modèles
3. Comparer les différents modèles.

Caractéristiques des réseaux mobiles.

- Quels sont les entités/agents des réseaux ?
  portables, robots, capteurs
- Quel est le degré du synchronisme du réseau ?
  synchrone ou asynchrone,
- Quelles sont les contraintes physiques/communications ?
  la puissance des batteries, qualité de la transmission, puissance d'émission....
- Quelles sont les contraintes sur la mobilité ?
  types de mouvement (probabiliste ou prédéterminé), limite sur la vitesse de déplacement
- . . .

Introduction des protocoles de population

Un modèle (le $\oplus$ simple possible) pour les réseaux de capteurs où

- Quels sont les entités/agents des réseaux ?

  automate à état fini.

- Quel est le degré du synchronisme du réseau ?

  asynchronisme total

- Quelles sont les contraintes physiques/communications ?

  aucune infrastructure et aucun identifiant
  simple communication entre deux agents

- Quelles sont les contraintes sur la mobilité ?

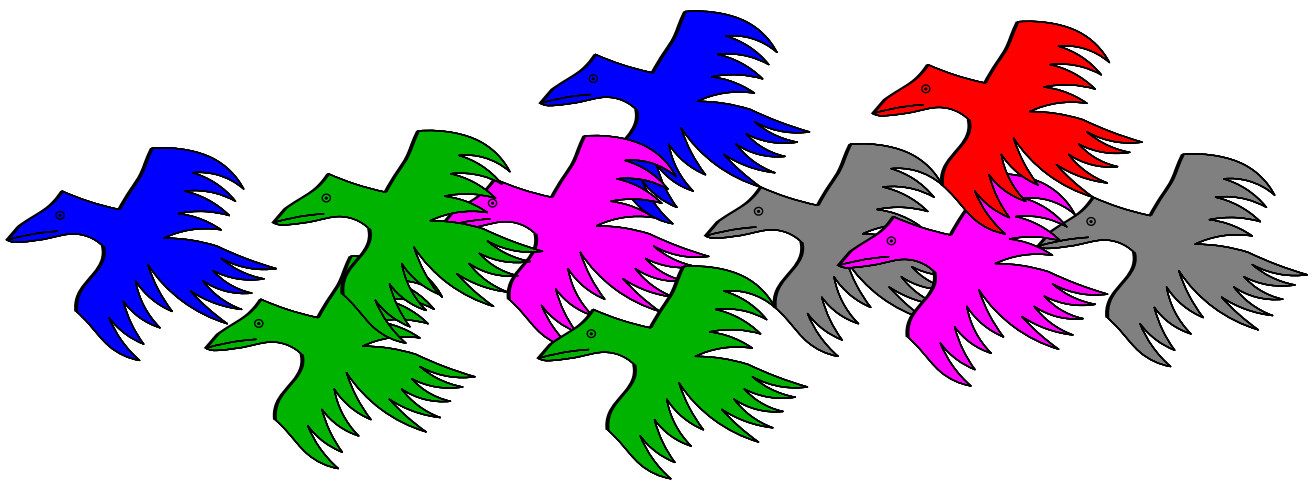
  interaction occationnelle entre deux agents possibles.

Exemple [Angluin et Al] : oiseaux.

- Un capteur sur chaque oiseau de la nuée.
- Les capteurs de 2 oiseaux peuvent interagir ensemble quand ils sont proches
- Peut-on programmer les capteurs pour décider si

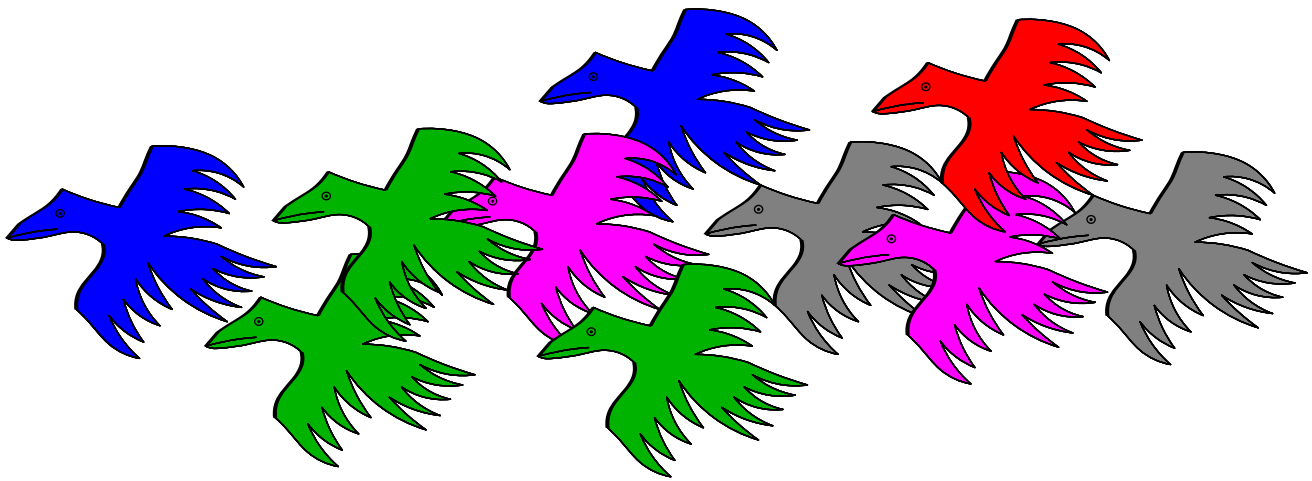
  au moins 5 oiseaux ont de la fièvre ?

  au moins 5% oiseaux ont de la fièvre ?



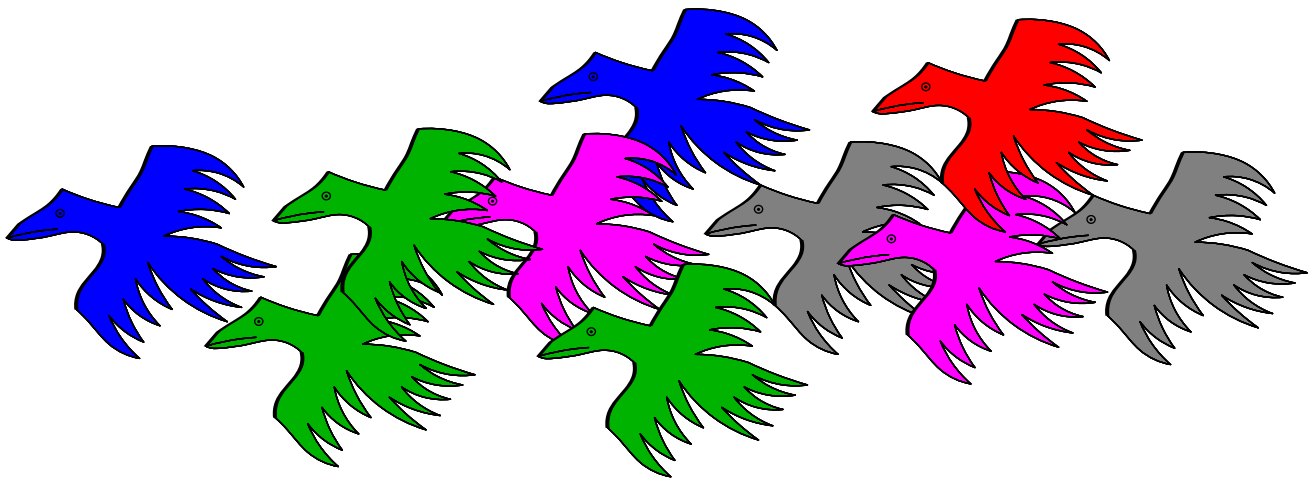
Exemple [Angluin et Al] : oiseaux.

- Un capteur sur chaque oiseau de la nuée.
- Les capteurs de 2 oiseaux peuvent interagir ensemble quand ils sont proches
- Peut-on programmer les capteurs pour décider si
  - au moins 5 oiseaux ont de la fièvre ?
  - au moins 5% oiseaux ont de la fièvre ?



Exemple [Angluin et Al] : oiseaux.

- Un capteur sur chaque oiseau de la nuée.
- Les capteurs de 2 oiseaux peuvent interagir ensemble quand ils sont proches
- Peut-on programmer les capteurs pour décider si
  - au moins 5 oiseaux ont de la fièvre ?
  - au moins 5% oiseaux ont de la fièvre ?



Plan de l'exposé

Protocoles classiques de population.
- Motivation
- Définitions
- Quelque algorithmes
- La puissance de calcul

Protocole de population ayant une population infinie
- Motivation
- Calcul de nombres irrationnels

Discussions et conclusions

Le modèle

Ensemble fini d'agents identiques (automates finis)

Interaction entre deux agents proches

Changement d'état des agents suite à l'interaction.

Hypothèse d'équité : toutes les interactions par paire se réalisent probablement.

Idée : les interactions se réalisent de façon indépendante et se définissent par une loi de probabilité

Comment calculer ?

1. Codage de l'entrée : donner un état initial à chaque agent ;
2. Exécution de l'algorithme ;
3. Décodage de la sortie (interprétation des états des agents).

Exemple simple : calculer le *OR*

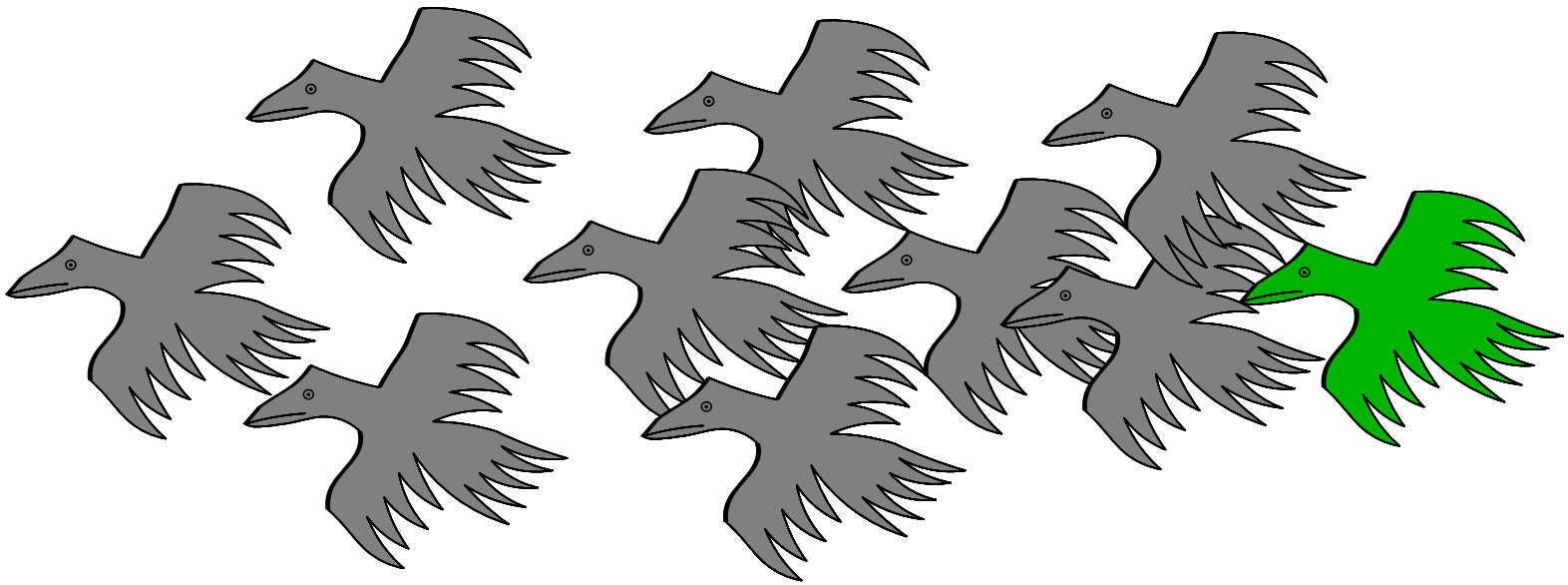
Objectif : Calculer le *OR* d'un ensemble de bits.

- Etats : $\{0, 1\}$
- Entrée : chaque agent contient un bit.
- la fonction de transition :

$$\begin{aligned} 0,\ 1 &\longrightarrow 1,\ 1 \\ 0,\ 0 &\longrightarrow 0,\ 0 \\ 1,\ 0 &\longrightarrow 1,\ 1 \\ 1,\ 1 &\longrightarrow 1,\ 1 \end{aligned}$$

- Sortie : l'état de chaque agent.

Exemple simple : calculer le *OR*

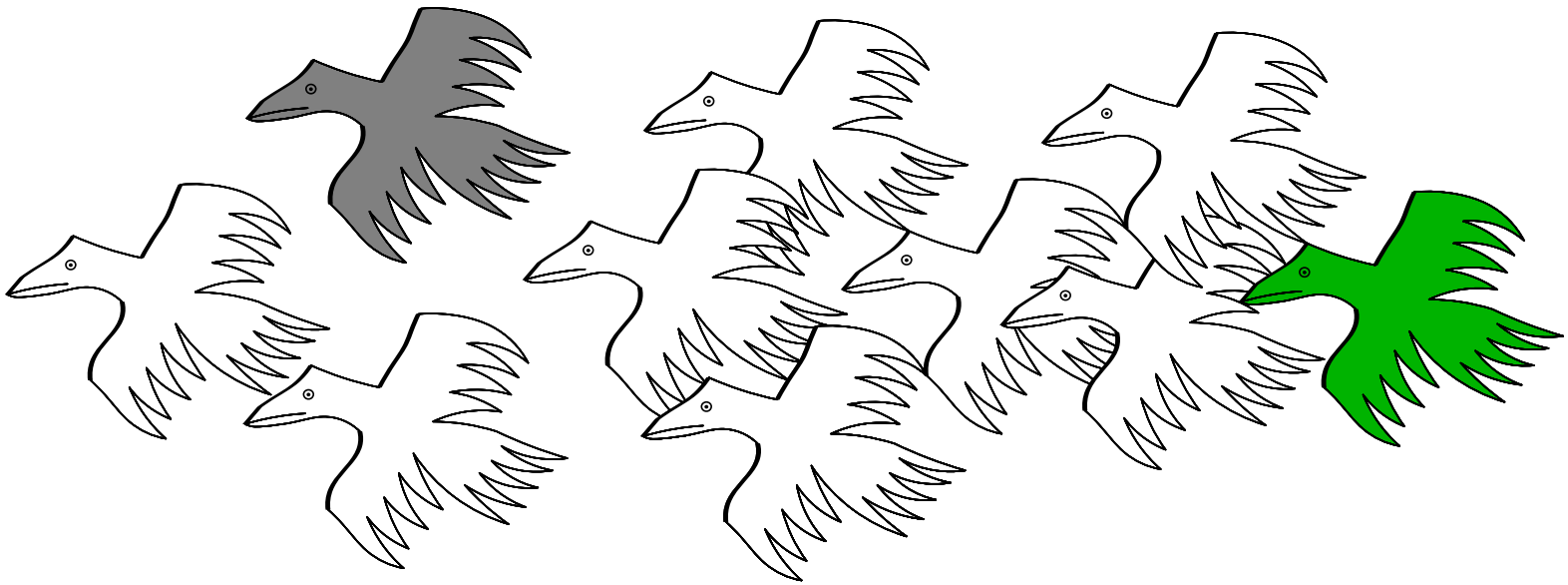


1 = 

0 = 

$$
\begin{aligned}
0,\ 1 &\rightarrow 1,\ 1 \\
0,\ 0 &\rightarrow 0,\ 0 \\
1,\ 0 &\rightarrow 1,\ 1 \\
1,\ 1 &\rightarrow 1,\ 1
\end{aligned}
$$

Exemple simple : calculer le *OR*

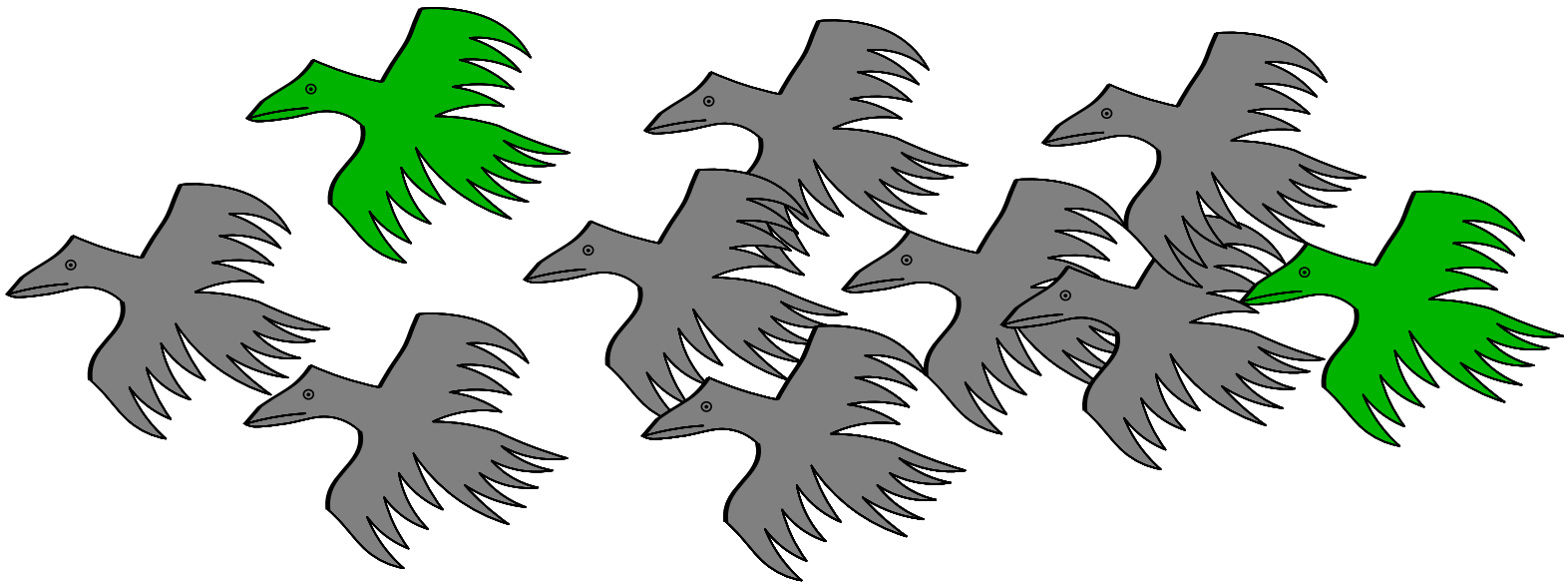


1 = 

0 = 

$$0, 1 \rightarrow 1, 1$$
$$0, 0 \rightarrow 0, 0$$
$$1, 0 \rightarrow 1, 1$$
$$1, 1 \rightarrow 1, 1$$

Exemple simple : calculer le *OR*

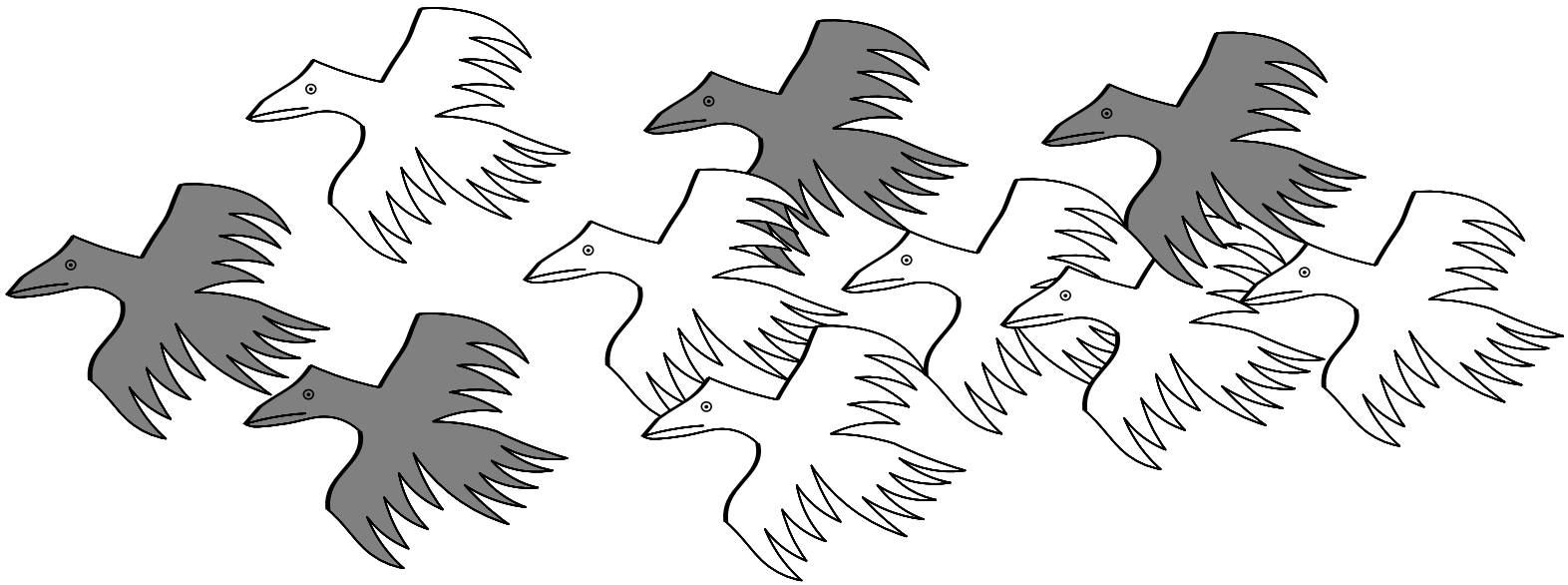


1 = 

0 = 

$$
\begin{aligned}
0,\ 1 &\rightarrow 1,\ 1 \\
0,\ 0 &\rightarrow 0,\ 0 \\
1,\ 0 &\rightarrow 1,\ 1 \\
1,\ 1 &\rightarrow 1,\ 1
\end{aligned}
$$

Exemple simple : calculer le *OR*

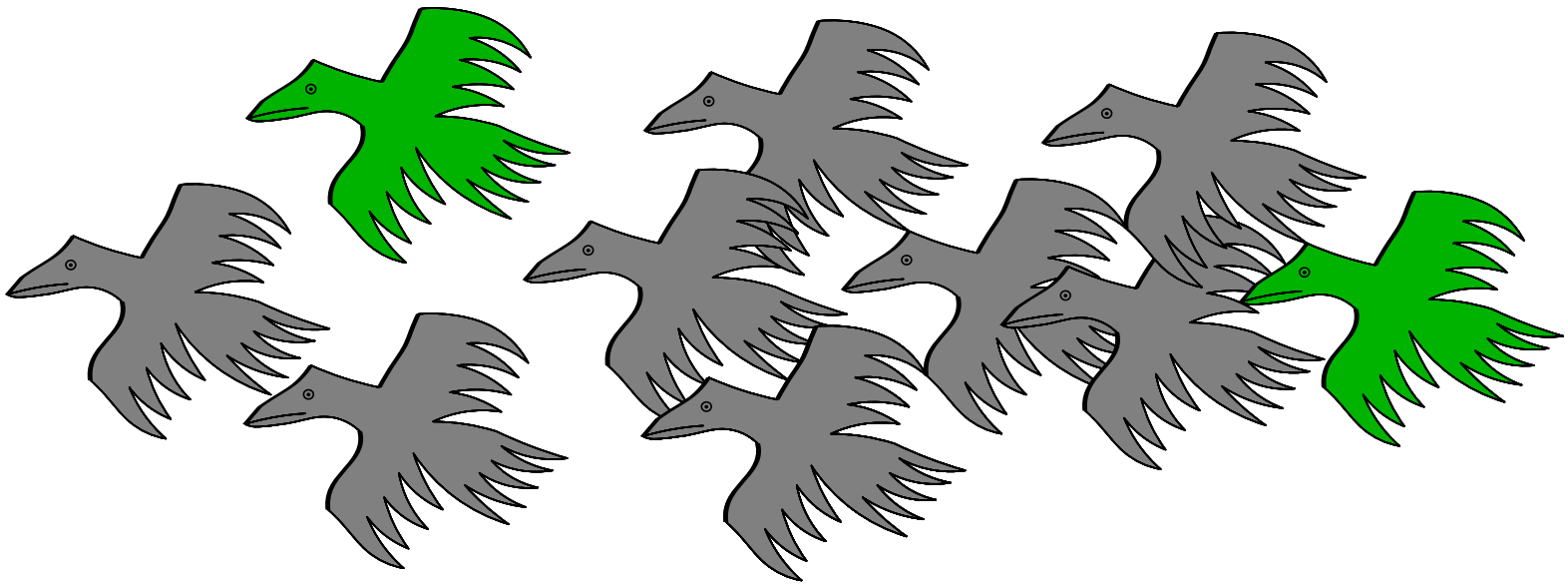


1 = 

0 = 

$$
\begin{aligned}
0,\ 1 &\rightarrow 1,\ 1 \\
0,\ 0 &\rightarrow 0,\ 0 \\
1,\ 0 &\rightarrow 1,\ 1 \\
1,\ 1 &\rightarrow 1,\ 1
\end{aligned}
$$

Exemple simple : calculer le *OR*

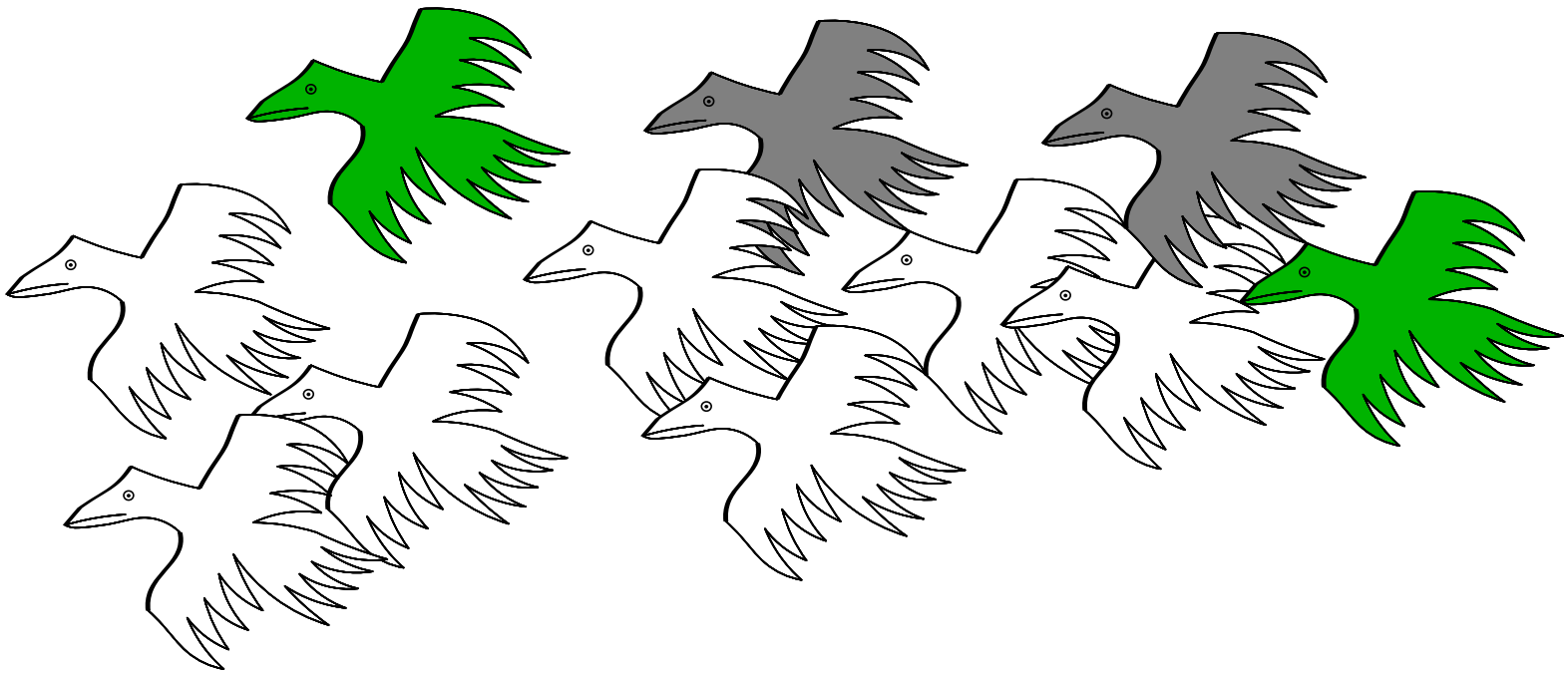


1 = 

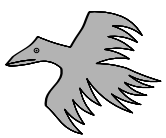
0 = 

$$
\begin{aligned}
0,\ 1 &\rightarrow 1,\ 1 \\
0,\ 0 &\rightarrow 0,\ 0 \\
1,\ 0 &\rightarrow 1,\ 1 \\
1,\ 1 &\rightarrow 1,\ 1
\end{aligned}
$$

Exemple simple : calculer le *OR*

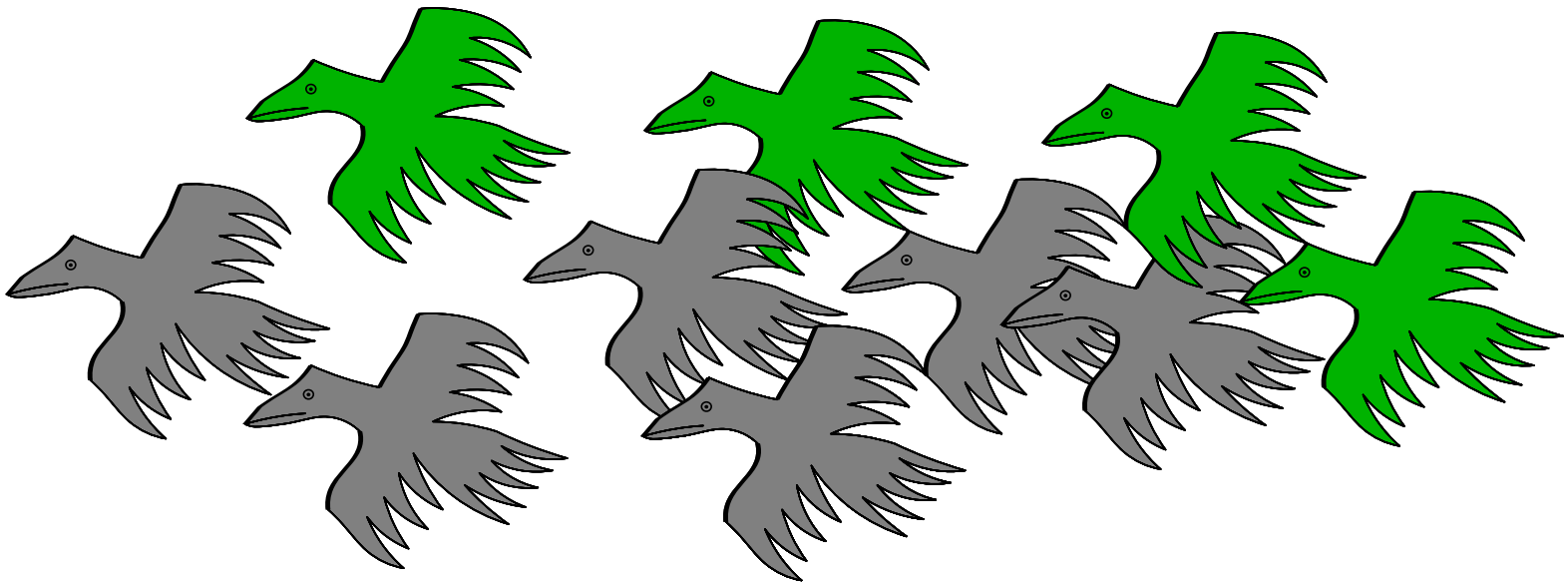


1 = 

0 = 

$$
\begin{aligned}
0,\ 1 &\rightarrow 1,\ 1 \\
0,\ 0 &\rightarrow 0,\ 0 \\
1,\ 0 &\rightarrow 1,\ 1 \\
1,\ 1 &\rightarrow 1,\ 1
\end{aligned}
$$

Exemple simple : calculer le *OR*

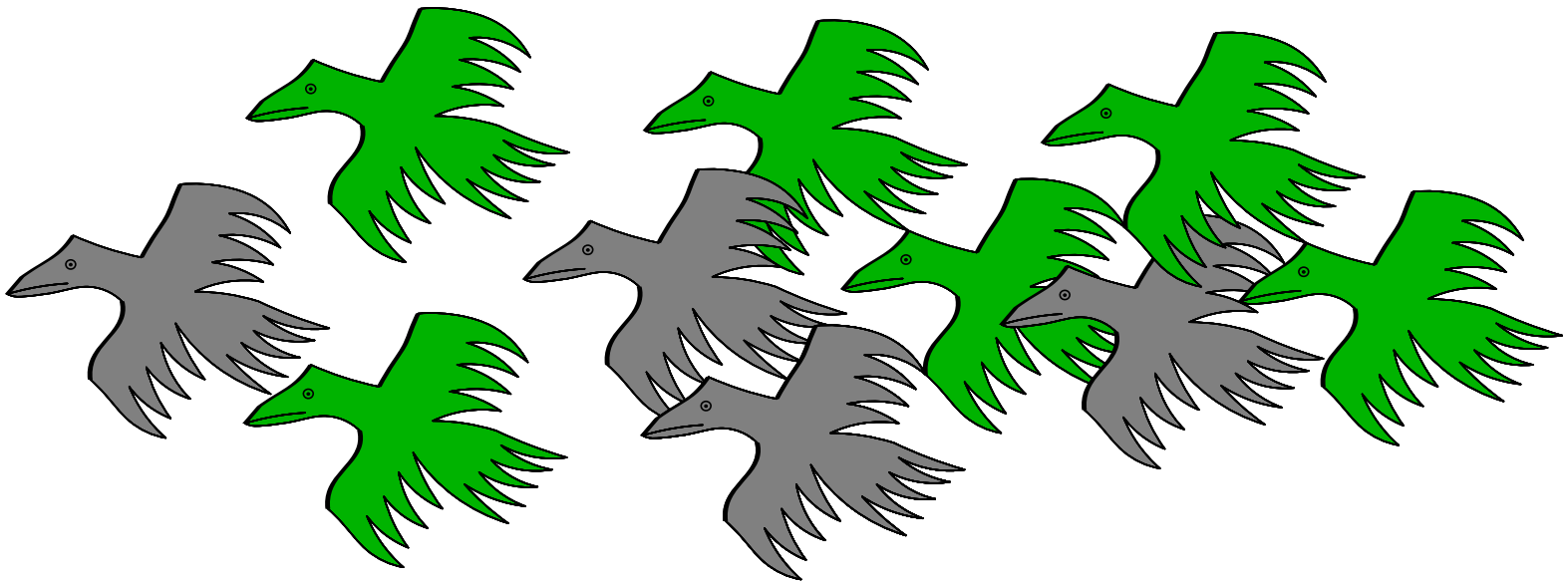


$1 =$ 

$0 =$ 

$$
\begin{aligned}
0,\ 1 &\rightarrow 1,\ 1 \\
0,\ 0 &\rightarrow 0,\ 0 \\
1,\ 0 &\rightarrow 1,\ 1 \\
1,\ 1 &\rightarrow 1,\ 1
\end{aligned}
$$

Exemple simple : calculer le *OR*

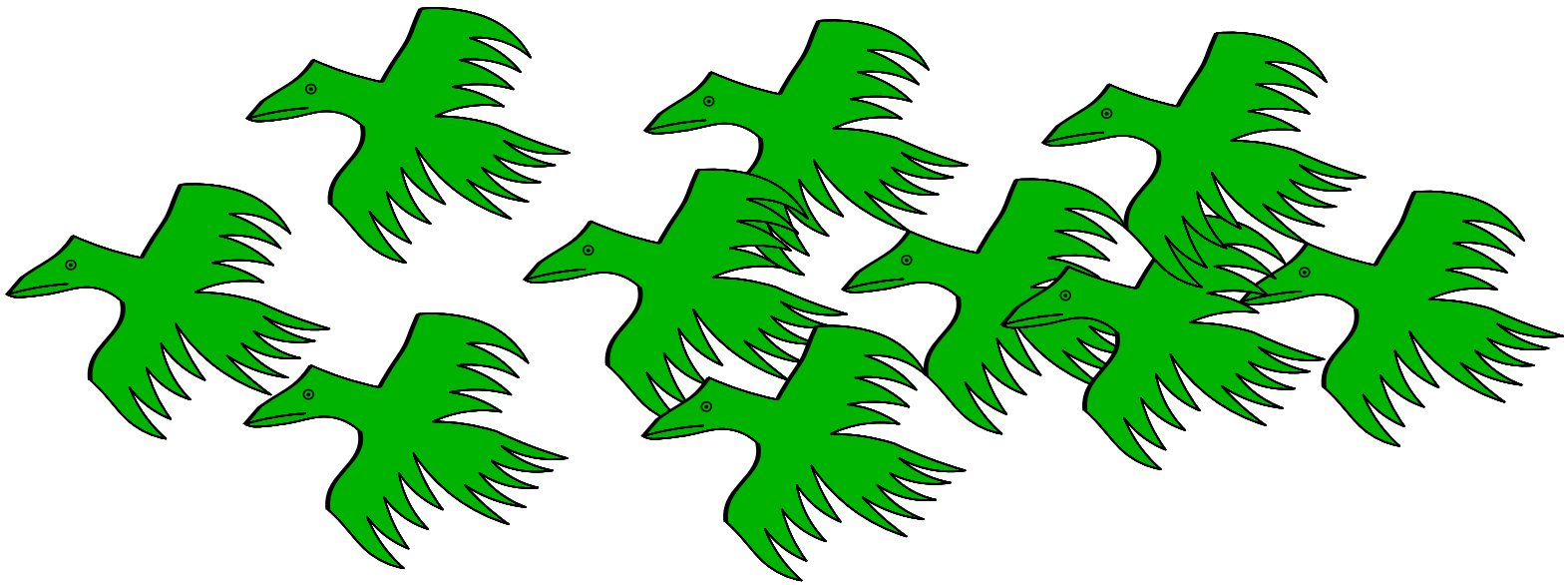


$1 =$ 

$0 =$ 

$$
\begin{aligned}
0,\ 1 &\rightarrow 1,\ 1 \\
0,\ 0 &\rightarrow 0,\ 0 \\
1,\ 0 &\rightarrow 1,\ 1 \\
1,\ 1 &\rightarrow 1,\ 1
\end{aligned}
$$

Exemple simple : calculer le *OR*



$1 =$ 

$0 =$ 

$$
\begin{aligned}
0,\ 1 &\rightarrow 1,\ 1 \\
0,\ 0 &\rightarrow 0,\ 0 \\
1,\ 0 &\rightarrow 1,\ 1 \\
1,\ 1 &\rightarrow 1,\ 1
\end{aligned}
$$

Définition d'un algorithme

L'algorithme est composé :

1. d'un ensemble fini d'états
2. de règles de transition
3. d'une fonction de codage des entrées.
4. d'une fonction d'interprétation des états des agents

Remarque : L'algorithme doit être indépendant de la taille de la population.

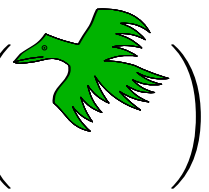
Protocoles classiques de population :

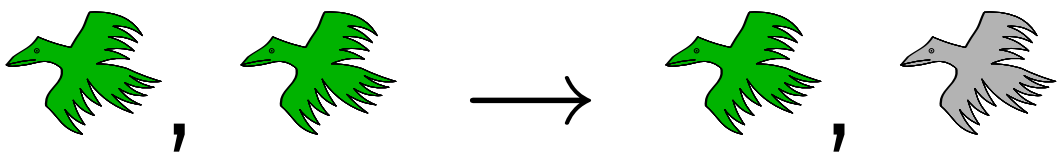
Algorithmes

D'autres algorithmes

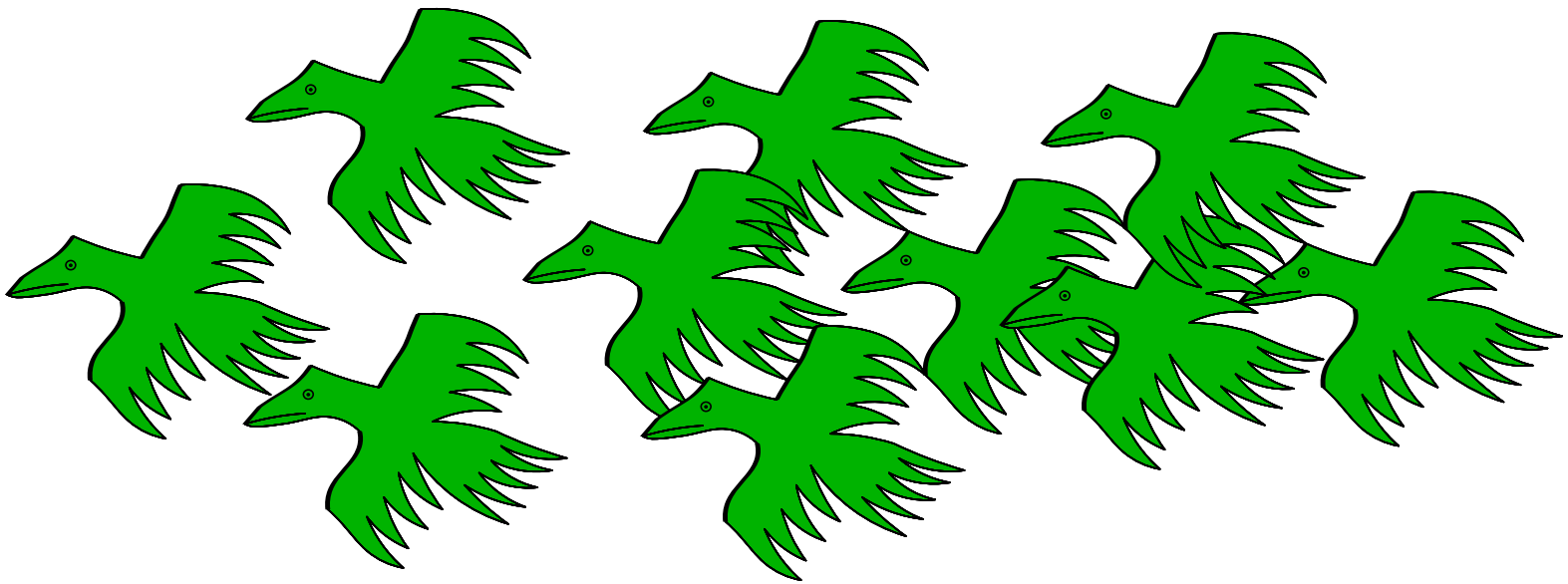
- Calculer le *OR*, le *AND* d'un ensemble de bits.
- Calculer d'un chef
- Calculer la majorité
- Calculer un seuil.
- Calculer si X% d'agents est dans un état particulier
- Réaliser les opérations arithmétiques : $+$, $-$, $\mod$

Calcul d'un chef

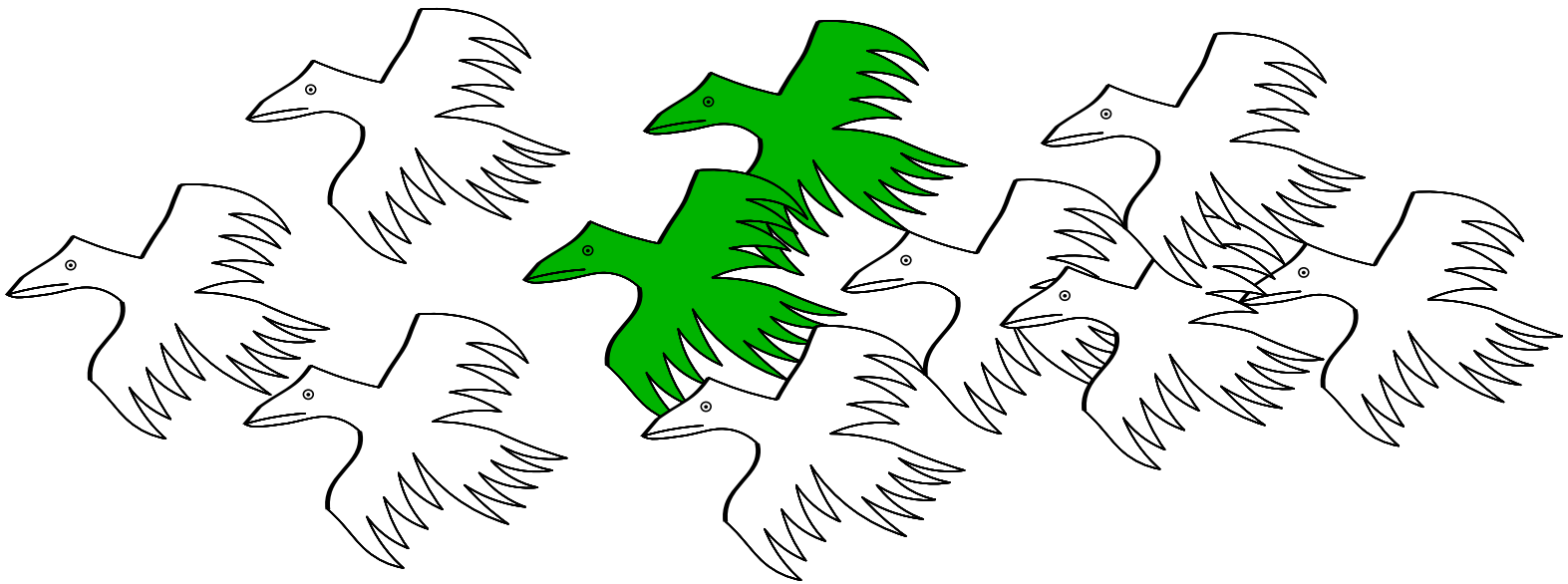
- Entrée : tous les agents ont le même état 
- Sortie : un seul agent sera le chef (état differencié)
- la fonction de transition :



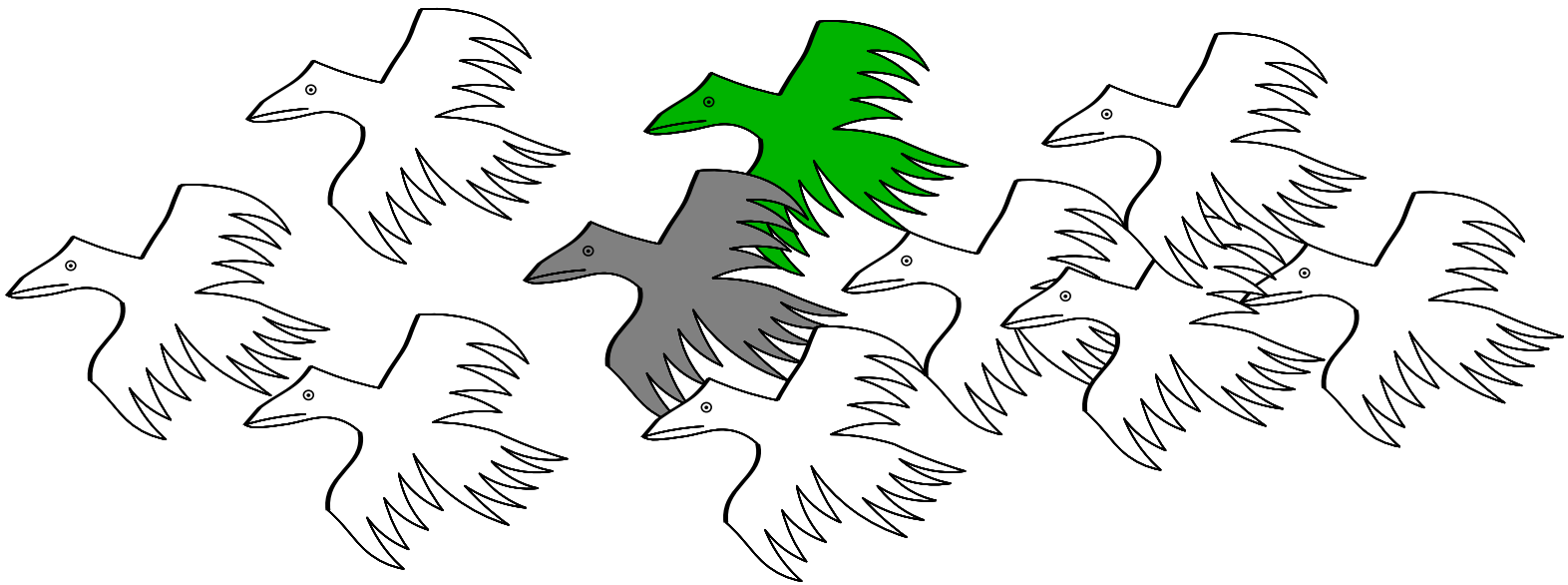
Calcul d'un chef



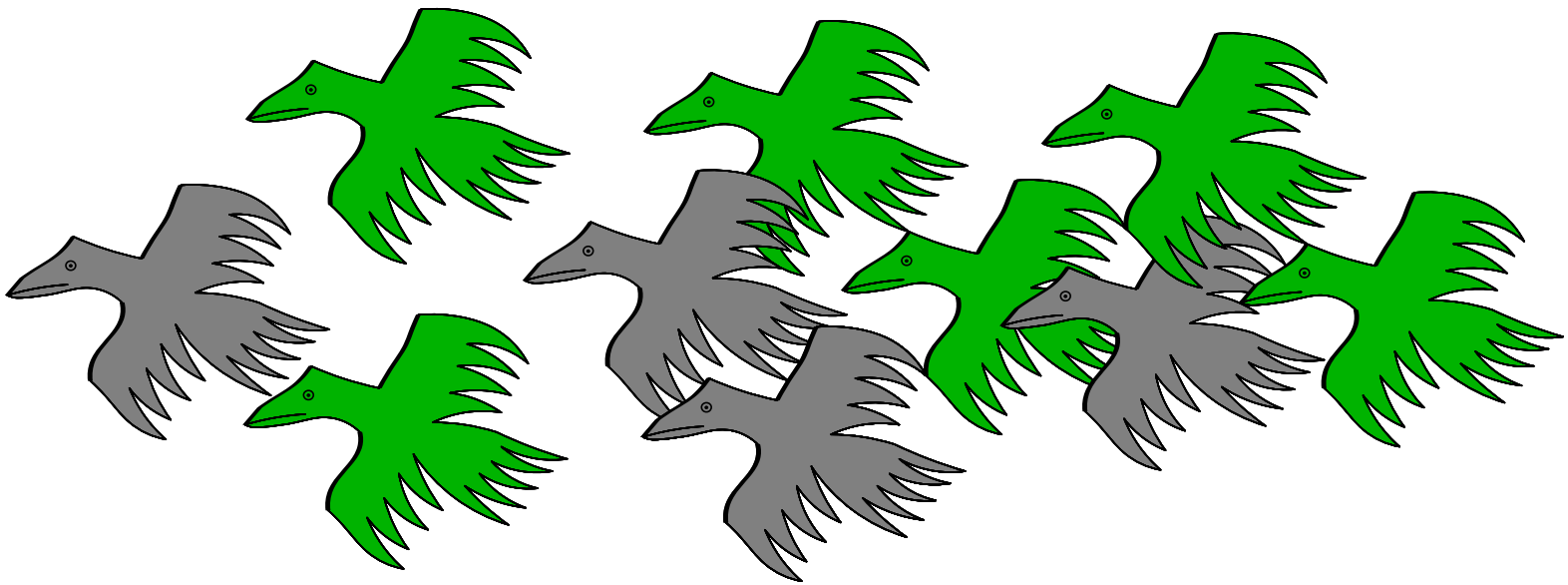
Calcul d'un chef



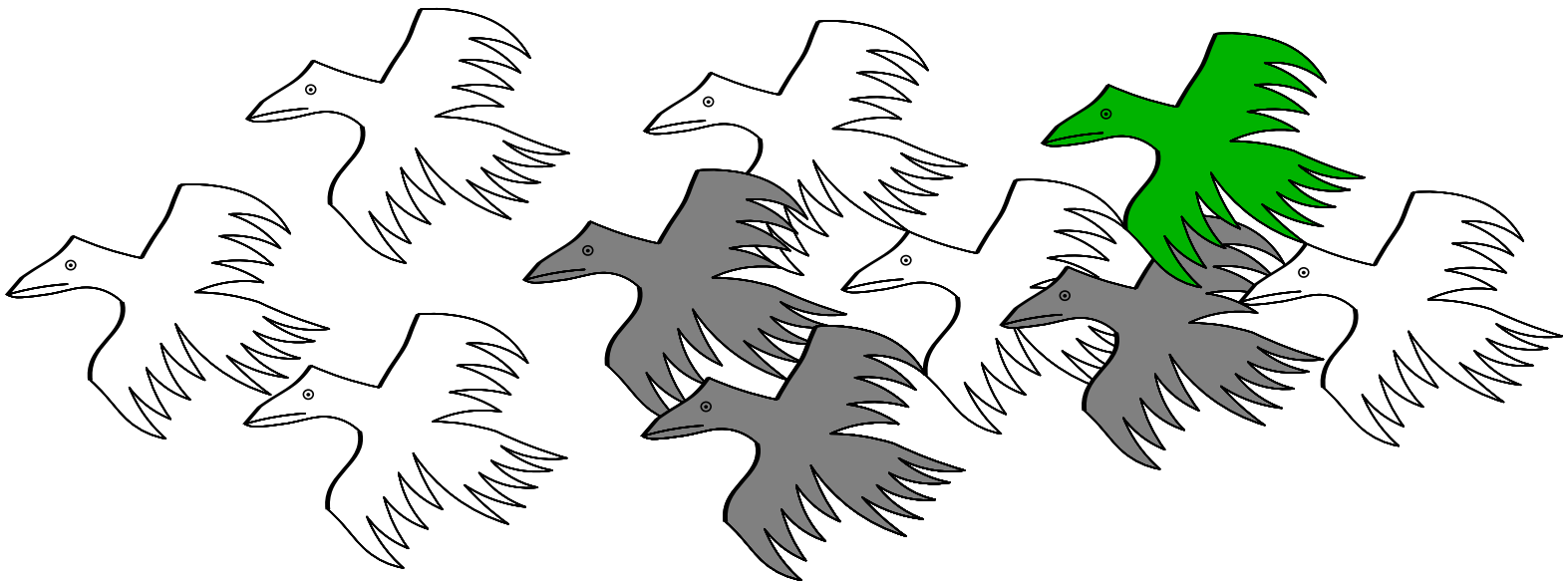
Calcul d'un chef



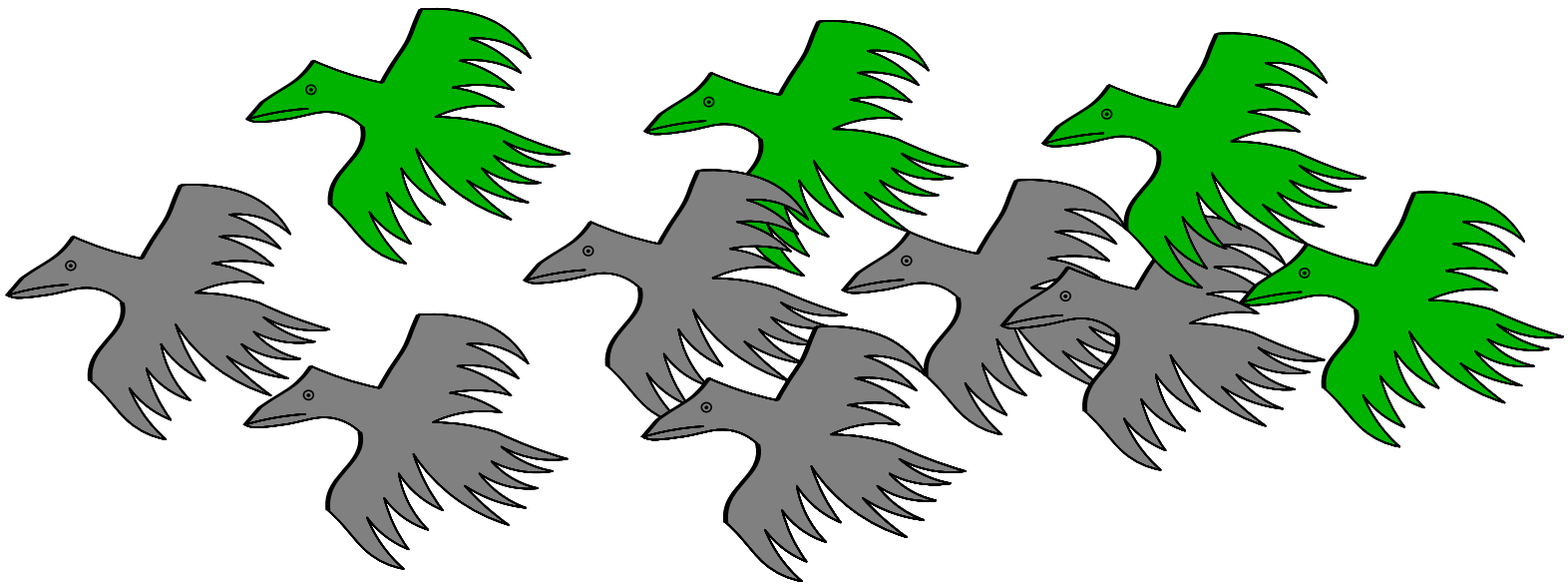
Calcul d'un chef



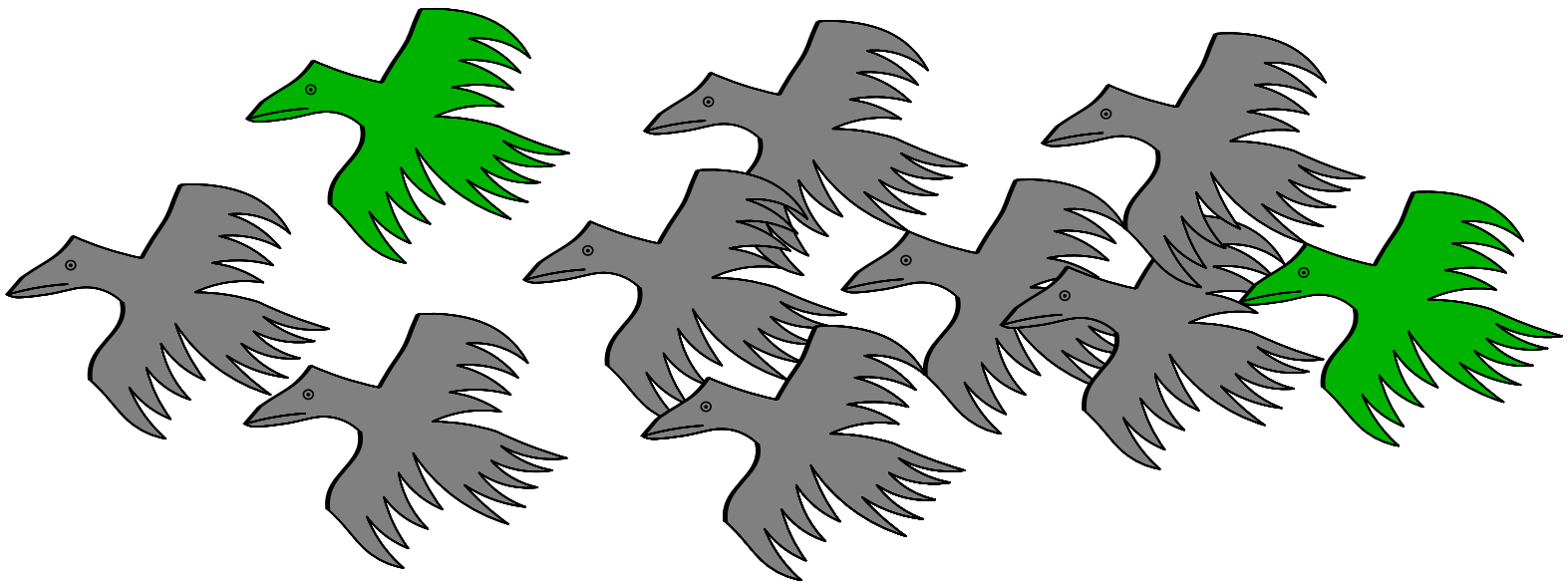
Calcul d'un chef



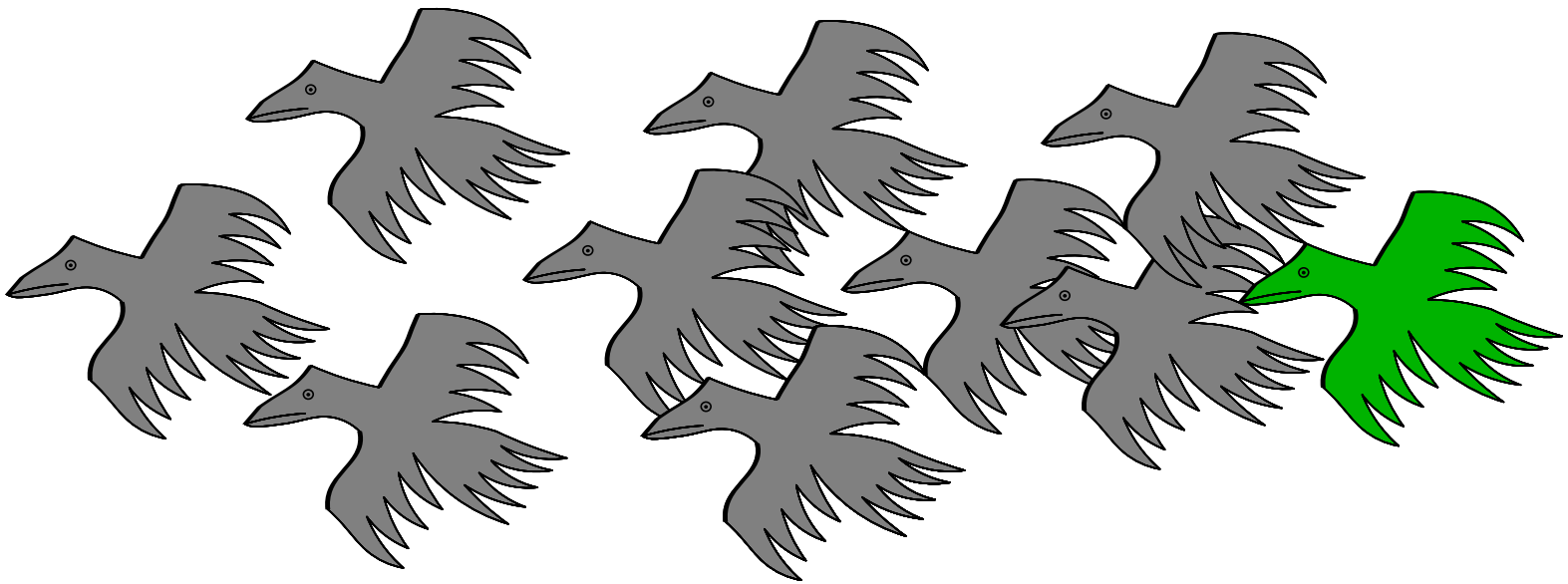
Calcul d'un chef



Calcul d'un chef



Calcul d'un chef



Calcul de la majorité

- Entrée : Ensemble d'éléments
- Algorithme : Déterminer si #  > #  .
- Etats : {  ,  , *Yes*, *No*}.

- Règles :

Calcul de la majorité

- Entrée : Ensemble d'éléments
- Algorithme : Déterminer si #  > #  .
- Etats : {  ,  , *Yes*, *No*}.

   ,  $\rightarrow$ *No*, *No*

- Règles :

élimination de  et de 

Calcul de la majorité

- Entrée : Ensemble d'éléments
- Algorithme : Déterminer si $\#$  $> \#$  .
- Etats : {  ,  , *Yes*, *No*}.
- Règles :  ,  $\rightarrow$ *No*, *No*
   , *No* $\rightarrow$  , *Yes*

existence d'un  et la réponse = oui

Calcul de la majorité

- Entrée : Ensemble d'éléments
- Algorithme : Déterminer si #  > #  .
- Etats : {  ,  , *Yes*, *No*}.
- Règles :

 ,  → *No*, *No*

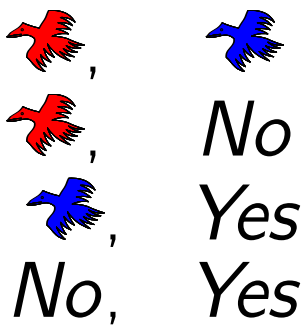
 , *No* →  , *Yes*

 , *Yes* →  , *No*

existence d'un  et la réponse = non

Calcul de la majorité

- Entrée : Ensemble d'éléments
- Algorithme : Déterminer si #  > #  .
- Etats : {  ,  , *Yes*, *No*}.
- Règles :



la réponse = non

Le prédicat du seuil

- Entrée : les agents sont dans les états 0 ou 1
- Algorithme : déterminer si au moins 5 agents sont dans 1
- Etats : $\{0, 1, 2, 3, 4, 5\}$
- Règles :
- Sortie : finalement tous les agents seront dans l'état 5 si au départ 5 agents étaient dans l'état 1

Le prédicat du seuil

- Entrée : les agents sont dans les états 0 ou 1
- Algorithme : déterminer si au moins 5 agents sont dans 1
- Etats : $\{0, 1, 2, 3, 4, 5\}$
- Règles :

$$
\begin{array}{llcll}
0, & 1 & \longrightarrow & 1, & 0 \\
1, & 1 & \longrightarrow & 2, & 0 \\
2, & 1 & \longrightarrow & 3, & 0 \\
3, & 1 & \longrightarrow & 4, & 0 \\
4, & 1 & \longrightarrow & 5, & 0 \\
5, & j & \longrightarrow & 5, & 5 \\
i, & j & \longrightarrow & i, & j \quad \text{dans les autres cas.}
\end{array}
$$

- Sortie : finalement tous les agents seront dans l'état 5 si au départ 5 agents étaient dans l'état 1

Déterminer s'il y a plus 5% de 1

- Entrée : les agents sont dans les états 0 ou 1
- Algorithme : déterminer si au moins 5% agents sont dans 1
- Etats : $\{0, 1, 2, 3, 4, 5\}$
- Règles :

Déterminer s'il y a plus 5% de 1

- Entrée : les agents sont dans les états 0 ou 1
- Algorithme : déterminer si au moins 5% agents sont dans 1
- Etats : $\{0, 1, 2, 3, 4, 5\}$
- Règles :
  - Identique au protocole de la majorité,
  - A l'exception que l'état 1 peut supprimer 19 états 0.

Déterminer s'il y a plus 5% de 1

- Entrée : les agents sont dans les états 0 ou 1
- Algorithme : déterminer si au moins 5% agents sont dans 1
- Etats : $\{0, 1, 2, 3, 4, 5\}$
- Règles :
  - Identique au protocole de la majorité,
  - A l'exception que l'état 1 peut supprimer 19 états 0.

Remarque : il existe un algorithme qui détermine si le pourcentage de $1 \geq Y\%$

Les algorithmes

- Calculer le *OR*, le *AND* d'un ensemble de bits.
- Calculer d'un chef
- Calculer la majorité
- Calculer un seuil.
- Calculer si X% d'agents est dans un état particulier
- Réaliser les opérations arithmétiques : $+$, $-$, $\mod$

Protocoles classiques de population :

puissance de calcul

Les prédicats

Un prédicat a comme sortie YES ou NO.

Un prédicat doit être symétrique
(l'ordre des entrées n'a aucune importance).

Donc un prédicat doit être écrit comme $P(x_1, x_2, ..., x_k)$ où

$k =$ le nombre initial d'états.

$x_i =$ le nombre d'agents initialisés à l'état i.

Les prédicats calculables

Theorem (Angluin et al 2006)

Un prédicat est calculable par un protocole de population si et seulement si il satisfait une de ces conditions

- $\sum_{i=1}^{k} c_i x_i \geq a$, *où* a, c_i *sont des constantes entières*
- $\sum_{i=1}^{k} c_i x_i \equiv a \pmod{b}$ *où* a, b *et* c_i *sont des constantes entières*
- *une combinaison linéaire des prédicats ci-dessus.*

▸ Bonus

Les prédicats calculables

Theorem (Angluin et al 2006)

Un prédicat est calculable par un protocole de population si et seulement si il satisfait une de ces conditions

- $\sum_{i=1}^{k} c_i x_i \geq a$, *où* a, c_i *sont des constantes entières*

  Généralisation des algorithmes calculant le prédicat seuil et la majorité.

- $\sum_{i=1}^{k} c_i x_i \equiv a \pmod{b}$ *où* a, b *et* c_i *sont des constantes entières*

  Généralisation de l'algorithme calculant mod .

- *une combinaison linéaire des prédicats ci-dessus.*

▸ Bonus

Autre caractérisation de cette arithmétique

Un prédicat est calculable par un protocole de population si et seulement si il peut être exprimé en logique de premier ordre en utilisant des variables. et des symboles

$$+, 0, 1, \vee, \wedge, \neg, \forall, \exists, =, <, (,)$$

(Arithmétique de Presburger [1929])

Remarque : aucune multiplication.

Exemples :

- majorité : $x_1 < x_2$
- divisible par 3 : $\exists y : y + y + y = x_1$
- au moins 40% de 1 :$\neg(x1 + x1 < x0 + x0 + x0)$.

En résumé.

Nous connaissons exactement quels sont les prédicats calculables dans ce modèle classique.

- Ils sont des combinaisons booléennes de prédicats composés de comparaisons et de modulo.
- Ils sont semi-linéaires.
- Ils sont dans l'arithmétique de Presburger.

Remarque : Autoriser les probabilités dans les règles de transition ne change pas la puissance de calcul. [Bournez et Al 2008]

Variante des protocoles de population

Le modèle classique est maintenant bien étudié.

Il existe des variantes dans la littérature :

1. Les interactions correspondent à des graphes
2. Des erreurs peuvent apparaître dans les communications
3. Les communications peuvent être uni-directionnelles

Protocole de population ayant une population infinie

Protocole de population ayant une population infinie

Inspiré par la théorie des jeux évolutionnaires :
en considérant les interactions comme des jeux

Hypothèses :

- une population infinie (au lieu d'un GRAND nombre d'agents)
- notion de pourcentage d'agents dans un état (au lieu de les compter)

Un exemple

Etats : $\{+, -\}$.
Règles d'interactions

$$\begin{array}{lcl} ++ & \rightarrow & +- \\ +- & \rightarrow & ++ \\ -+ & \rightarrow & ++ \\ -- & \rightarrow & +- \end{array}$$

Que peut-on dire sur

$$p_+ = \frac{nombre\ de\ +}{nombre\ total\ de\ +\ et\ de\ -} ?$$

Quand le nombre d'agents est fixé ?

- Les règles de transition décrivent une chaine homogène de Markov avec 2^n états.
- La configuration $(-, -, \cdots, -)$ est quittée en une seule étape.
- Autrement, les autres configurations sont atteignables avec une probabilité positive.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

Quand le nombre d'agents est fixé ?

La distribution de $+$ et $-$ va converger (chaine ergodique)

Quand le nombre d'agents est fixé ?

La distribution de $+$ et $-$ va converger

vers une distribution stationnaire de la chaine de Markov.

Quand le nombre d'agents est fixé ?

La distribution de $+$ et $-$ va converger

vers une distribution stationnaire de la chaine de Markov.

Donc, $E[p_+]$ va converger vers une valeur rationnelle.

Quand le nombre d'agents est infini ?

$$
\begin{aligned}
++ &\longrightarrow +- \\
+- &\longrightarrow ++ \\
-+ &\longrightarrow ++ \\
-- &\longrightarrow +-
\end{aligned}
$$

Convergence du système ? Vers quelle valeur de p_+ ?

1. Approche informelle.

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

- Moyenne de la création de $+$

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

- Moyenne de la création de $+$

  $-1 * p_+^2$

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

- Moyenne de la création de $+$

  $-1 * p_+^2 \ +1 * p_+(1 - p_+)$

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

- Moyenne de la création de $+$

$$-1 * p_+^2 \; +1 * p_+(1-p_+) + 1 * p_+(1-p_+)$$

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

- Moyenne de la création de $+$

$$-1 * p_+^2 \; +1 * p_+(1-p_+) + 1 * p_+(1-p_+) \; +1 * (1-p_+)^2$$

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

- Moyenne de la création de $+$ $(= 1 - 2p_+^2\ )$

$$-1 * p_+^2\ +1 * p_+(1-p_+) + 1 * p_+(1-p_+)\ +1 * (1-p_+)^2$$

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

- Moyenne de la création de $+$ $(= 1 - 2p_+^2)$

doit tendre vers 0 (si convergence)

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

- Moyenne de la création de $+$ $(= 1 - 2p_+^2)$

doit tendre vers 0 (si convergence)

- Donc

$$1 - 2p_+^2 = 0$$

Approche informelle.

$$
\begin{array}{lcl}
++ & \rightarrow & +- \\
+- & \rightarrow & ++ \\
-+ & \rightarrow & ++ \\
-- & \rightarrow & +-
\end{array}
$$

- Moyenne de la création de $+$ $(= 1 - 2p_+^2)$

doit tendre vers 0 (si convergence)

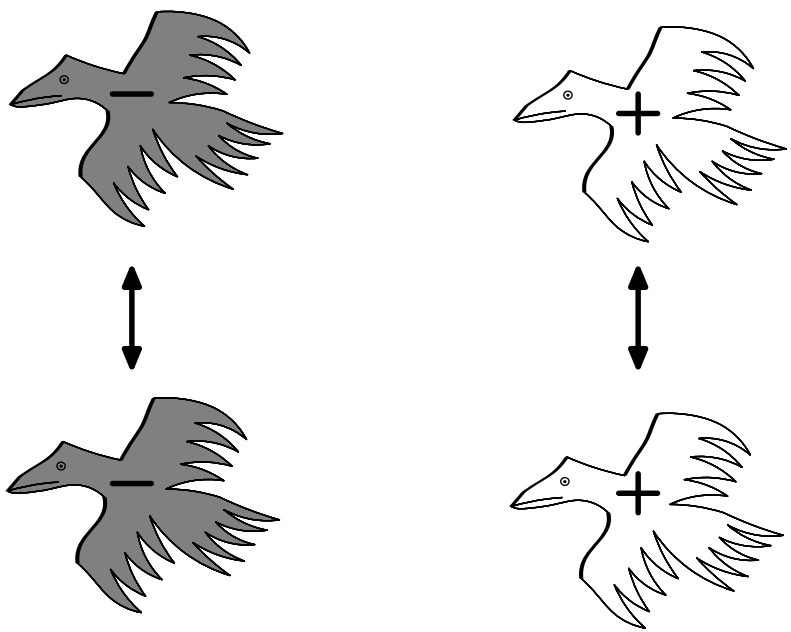
- Donc

$$p_+ = \frac{\sqrt{2}}{2}$$

Cas pathologique

- Configuration : autant de $+$ que de $-$
- A chaque étape,
  - les $+$ s'interagissent entre eux (et les $-$ aussi)
  - $p_+ = 1/2$

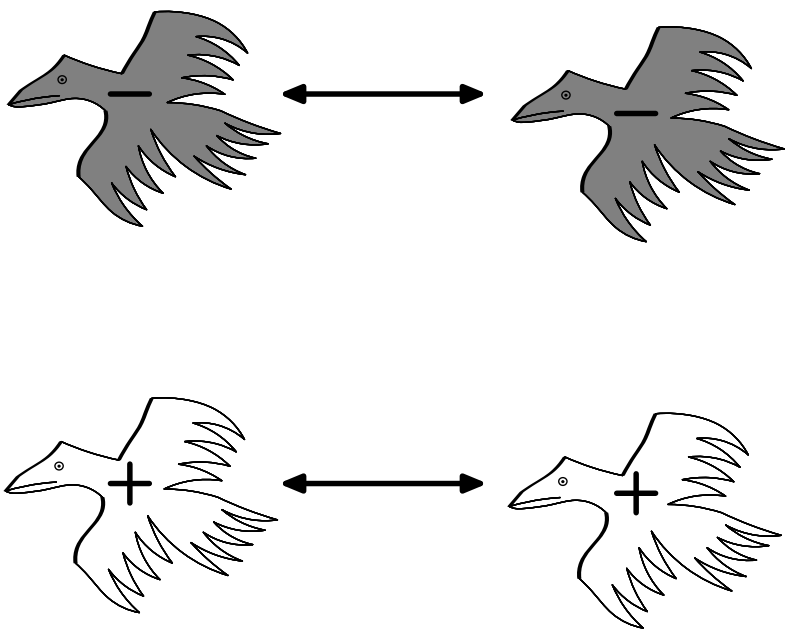
$$
\begin{aligned}
++ &\longrightarrow +- \\
+- &\longrightarrow ++ \\
-+ &\longrightarrow ++ \\
-- &\longrightarrow +-
\end{aligned}
$$



Cas pathologique

- Configuration : autant de $+$ que de $-$
- A chaque étape,
  - les $+$ s'interagissent entre eux (et les $-$ aussi)
  - $p_+ = 1/2$

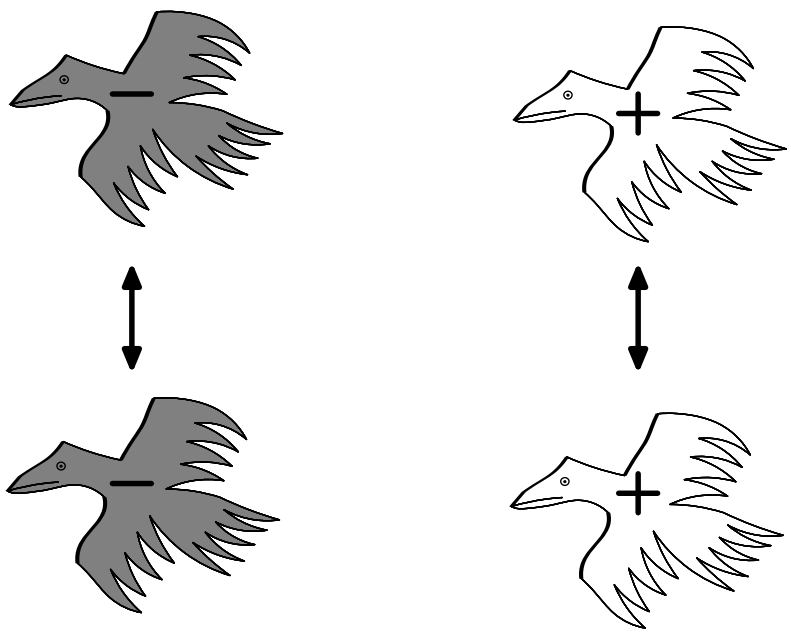
$$
\begin{aligned}
++ &\longrightarrow +- \\
+- &\longrightarrow ++ \\
-+ &\longrightarrow ++ \\
-- &\longrightarrow +-
\end{aligned}
$$



Cas pathologique

- Configuration : autant de $+$ que de $-$
- A chaque étape,
  - les $+$ s'interagissent entre eux (et les $-$ aussi)
  - $p_+ = 1/2$

$$
\begin{aligned}
++ &\longrightarrow +- \\
+- &\longrightarrow ++ \\
-+ &\longrightarrow ++ \\
-- &\longrightarrow +-
\end{aligned}
$$



Où est l'erreur ?

$$
\begin{aligned}
++ &\rightarrow +- \\
+- &\rightarrow ++ \\
-+ &\rightarrow ++ \\
-- &\rightarrow +-
\end{aligned}
$$

la probabilité que la règle 1 se réalise est

$$p_+ \times (p_+ - \epsilon) \text{ et non } p_+ \times p_+$$

2. Approche Expérimentale

3. Approche formelle

Approche formelle : convergence

Théorème : Nous avons pour tout t,

$$p_+(\lfloor nt \rfloor) = \frac{\sqrt{2}}{2} + Z_n(t),$$

où quand n tend vers l'infini, $Z_n(t)$ converge en loi vers une solution d'équation ordinaire différentielle

$$dX(t) = (1 - 2X^2)dt.$$

Théorème : Nous avons, pour tout t,

$$p_+(\lfloor nt \rfloor) = \frac{\sqrt{2}}{2} + \frac{1}{\sqrt{n}} A_n(t),$$

où $A_n(t)$ converge en loi vers une solution d'équation ordinaire différentielle stochastique (processus de Orstein-Uhlenbeck)

$$dX(t) = -2\sqrt{2}X(t)dt + dB(t),$$

et aussi vers une gaussienne $\mathcal{N}(0, \frac{\sqrt{2}}{8})$ quant t tend vers l'infini.

Preuve : Application du théorème sur l'approximation de la diffusion de Stroock and Varadhan 79.

Protocole de population ayant une population infinie :

la puissance de calcul ?

Calculer avec une grande population

- Il existe un protocole qui **calcule** un nombre réel $\frac{\sqrt{2}}{2}$.

  Travail en collaboration avec O. Bournez, P. Chassaing, L. Gerin, X. Koegler.

- Quel nombre réel peut-on calculer ?
  - Nombre réels entre 0 et 1
  - Nombre réels algébriques.

Calculer avec une grande population

- Il existe un protocole qui **calcule** un nombre réel $\frac{\sqrt{2}}{2}$.

  Travail en collaboration avec O. Bournez, P. Chassaing, L. Gerin, X. Koegler.

- Quel nombre réel peut-on calculer ?
  - Nombre réels entre 0 et 1
  - Nombre réels algébriques.

Calculer avec une grande population

- Il existe un protocole qui **calcule** un nombre réel $\frac{\sqrt{2}}{2}$.

  Travail en collaboration avec O. Bournez, P. Chassaing, L. Gerin, X. Koegler.

- Quel nombre réel peut-on calculer ?
  - Nombre réels entre 0 et 1
  - Nombre réels algébriques.

Calculer avec une grande population

- Il existe un protocole qui **calcule** un nombre réel $\frac{\sqrt{2}}{2}$.

  Travail en collaboration avec O. Bournez, P. Chassaing, L. Gerin, X. Koegler.

- Quel nombre réel peut-on calculer ?
  - Nombre réels entre 0 et 1
  - Nombre réels algébriques.

Peut-on calculer tous les nombres algébriques entre 0 et 1 ?

Discussions

Pour l'instant, il n'existe pas de formalisation précise des nombres réels calculables.

- Ils peuvent être algébriques ou rationnels
- Ils sont clos pour [Koegler 2008]
  - le produit,
  - la somme,
  - la racine carrée
  - . . .

Travaux courants et futurs

- Caractériser précisément les nombres réellement calculables
- Adapter ces protocoles à la théorie des jeux évolutionnaires
  - Peut-on caratériser les protocoles de populations avec des jeux ? ou inversement ?
  - impact des règles utilisées : règles symétriques, différenciation entre éméteur/récepteur ...

Références

- Dans les protocoles de polulations classiques : travail de
  - Dana Angluin, James Aspnes, Melody Chan, Carole Delporte-Gallet, Zoë Diamadi, David Eisenstat, Michael J. Fischer, Hugues Fauconnier, Rachid Guerraoui, Hong Jiang, René Peralta, Eric Ruppert

- Dans les protocoles de polulations à large population :
  - Travail en cours de
    - Olivier Bournez, Philippe Chassaing, Jérémie Chalopin, Johanne Cohen, Pierre Fraigniaud, Lucas Gerin, Xavier Koegler.

Calcul de la racine carrée

Theorem

A partir d'un protocole calculant p_2, il existe un protocole qui calcule $\sqrt{p_2}$

$$
\begin{array}{lcl}
(1,1)(.,1) & \rightarrow & (1,1) \\
(1,1)(.,0) & \rightarrow & 3/4(1,1); 1/4(0,1) \\
(1,0)(0,0) & \rightarrow & 1/2(1,0); 1/2(0,0) \\
(1,0)(0,1) & \rightarrow & 1/4(1,0); 3/4(0,0) \\
(1,0)(1,0) & \rightarrow & 1/4(1,0); 3/4(0,0) \\
(1,0)(1,1) & \rightarrow & (1,0) \\
(0,1)(1,.) & \rightarrow & 1/2(1,1); 1/2(0,1) \\
(0,1)(0,.) & \rightarrow & 1/4(1,1); 3/4(0,1) \\
(0,0)(1,.) & \rightarrow & 1/4(1,1); 3/4(0,1) \\
(0,0)(0,.) & \rightarrow & (0,0)
\end{array}
$$

$$p_1^{n+1} - p_1^n = -1/4((p_1^n)^2 - p_2)$$

$$\lim_{n \rightarrow \infty} p_1^n = \sqrt{p_2}$$

Calcul de la racine carrée

Theorem

A partir d'un protocole calculant p_2, il existe un protocole qui calcule $\sqrt{p_2}$

$$
\begin{array}{lcl}
(1,1)(.,1) & \rightarrow & (1,1) \\
(1,1)(.,0) & \rightarrow & 3/4(1,1); 1/4(0,1) \\
(1,0)(0,0) & \rightarrow & 1/2(1,0); 1/2(0,0) \\
(1,0)(0,1) & \rightarrow & 1/4(1,0); 3/4(0,0) \\
(1,0)(1,0) & \rightarrow & 1/4(1,0); 3/4(0,0) \\
(1,0)(1,1) & \rightarrow & (1,0) \\
(0,1)(1,.) & \rightarrow & 1/2(1,1); 1/2(0,1) \\
(0,1)(0,.) & \rightarrow & 1/4(1,1); 3/4(0,1) \\
(0,0)(1,.) & \rightarrow & 1/4(1,1); 3/4(0,1) \\
(0,0)(0,.) & \rightarrow & (0,0)
\end{array}
$$

$$p_1^{n+1} - p_1^n = -1/4((p_1^n)^2 - p_2)$$

$$\lim_{n \rightarrow \infty} p_1^n = \sqrt{p_2}$$

Calcul du produit

Theorem

A partir 2 protocoles de population calculant p_2 et p_3, il existe un protocole de population calculant $p_2 \times p_3$.

$$
\begin{array}{lcl}
(.,1,1)(.,.,1) & \rightarrow & (1,1,1) \\
(.,1,0)(.,.,1) & \rightarrow & (1,1,0) \\
(.,1,1)(.,.,0) & \rightarrow & (0,1,1) \\
(.,1,0)(.,.,0) & \rightarrow & (0,1,0) \\
(.,0,1)(.,.,.) & \rightarrow & (0,0,1) \\
(.,0,0)(.,.,.) & \rightarrow & (0,0,0)
\end{array}
$$

$$p^{n+1}_{(1,.,.)} - p^{n}_{(1,.,.)} = (p_2 p_3) - p^{n}_{1}$$

◂ Back

Calcul du produit

Theorem

A partir 2 protocoles de population calculant p_2 et p_3, il existe un protocole de population calculant $p_2 \times p_3$.

$$
\begin{array}{lcl}
(.,1,1)(.,.,1) & \longrightarrow & (1,1,1) \\
(.,1,0)(.,.,1) & \longrightarrow & (1,1,0) \\
(.,1,1)(.,.,0) & \longrightarrow & (0,1,1) \\
(.,1,0)(.,.,0) & \longrightarrow & (0,1,0) \\
(.,0,1)(.,.,.) & \longrightarrow & (0,0,1) \\
(.,0,0)(.,.,.) & \longrightarrow & (0,0,0)
\end{array}
$$

$$p^{n+1}_{(1,.,.)} - p^{n}_{(1,.,.)} = (p_2 p_3) - p^{n}_{1}$$

◂ Back

Calcul du produit

Theorem

A partir 2 protocoles de population calculant p_2 et p_3, il existe un protocole de population calculant $p_2 \times p_3$.

$$\begin{array}{rcl}
(.,1,1)(.,.,1) & \longrightarrow & (1,1,1) \\
(.,1,0)(.,.,1) & \longrightarrow & (1,1,0) \\
(.,1,1)(.,.,0) & \longrightarrow & (0,1,1) \\
(.,1,0)(.,.,0) & \longrightarrow & (0,1,0) \\
(.,0,1)(.,.,.) & \longrightarrow & (0,0,1) \\
(.,0,0)(.,.,.) & \longrightarrow & (0,0,0)
\end{array}$$

$$p^{n+1}_{(1,.,.)} - p^{n}_{(1,.,.)} = (p_2 p_3) - p^{n}_{1}$$

◂ Back

Calcul de la demi-somme

Theorem

A partir 2 protocoles de population calculant p_2 et p_3, il existe un protocole de population calculant $(p_2 + p_3)/2$.

$$
\begin{array}{lcl}
(1,1,1)(.,.,.) & \rightarrow & (1,1,1) \\
(0,1,1)(.,.,.) & \rightarrow & (1,1,1) \\
(1,0,1)(.,.,.) & \rightarrow & 1/2(1,0,1); 1/2(0,0,1) \\
(0,0,1)(.,.,.) & \rightarrow & 1/2(1,0,1); 1/2(0,0,1) \\
(1,1,0)(.,.,.) & \rightarrow & 1/2(1,1,0); 1/2(0,1,0) \\
(0,1,0)(.,.,.) & \rightarrow & 1/2(1,1,0); 1/2(0,1,0) \\
(1,0,0)(.,.,.) & \rightarrow & (0,0,0) \\
(0,0,0)(.,.,.) & \rightarrow & (0,0,0)
\end{array}
$$

$$p_1^{n+1} - p_1^n = 1/2(p_2^n + p_3^n) - p_1^n$$

◂ Back

Calcul de la demi-somme

Theorem

A partir 2 protocoles de population calculant p_2 et p_3, il existe un protocole de population calculant $(p_2 + p_3)/2$.

$$
\begin{array}{lcl}
(1,1,1)(.,.,.) & \rightarrow & (1,1,1) \\
(0,1,1)(.,.,.) & \rightarrow & (1,1,1) \\
(1,0,1)(.,.,.) & \rightarrow & 1/2(1,0,1); 1/2(0,0,1) \\
(0,0,1)(.,.,.) & \rightarrow & 1/2(1,0,1); 1/2(0,0,1) \\
(1,1,0)(.,.,.) & \rightarrow & 1/2(1,1,0); 1/2(0,1,0) \\
(0,1,0)(.,.,.) & \rightarrow & 1/2(1,1,0); 1/2(0,1,0) \\
(1,0,0)(.,.,.) & \rightarrow & (0,0,0) \\
(0,0,0)(.,.,.) & \rightarrow & (0,0,0)
\end{array}
$$

$$p_1^{n+1} - p_1^n = 1/2(p_2^n + p_3^n) - p_1^n$$

◂ Back

Calcul de la demi-somme

Theorem

A partir 2 protocoles de population calculant p_2 et p_3, il existe un protocole de population calculant $(p_2 + p_3)/2$.

$$
\begin{array}{lcl}
(1,1,1)(.,.,.) & \rightarrow & (1,1,1) \\
(0,1,1)(.,.,.) & \rightarrow & (1,1,1) \\
(1,0,1)(.,.,.) & \rightarrow & 1/2(1,0,1); 1/2(0,0,1) \\
(0,0,1)(.,.,.) & \rightarrow & 1/2(1,0,1); 1/2(0,0,1) \\
(1,1,0)(.,.,.) & \rightarrow & 1/2(1,1,0); 1/2(0,1,0) \\
(0,1,0)(.,.,.) & \rightarrow & 1/2(1,1,0); 1/2(0,1,0) \\
(1,0,0)(.,.,.) & \rightarrow & (0,0,0) \\
(0,0,0)(.,.,.) & \rightarrow & (0,0,0)
\end{array}
$$

$$p_1^{n+1} - p_1^n = 1/2(p_2^n + p_3^n) - p_1^n$$

◂ Back

How to Compute $\sum_{i=1}^{k} c_i x_i \geq a$

Input convention : each agent with ith input symbol starts in state c_i.
Each agent has a leader bit.
Let $m = \max(|a| + 1, |c_1|, ..., |c_k|)$.
Each agent also stores a value from $-m, -m+1, ..., m-1, m$.
If a leader meets a non-leader, their values change as follows :

$$\begin{array}{ll} x, y \rightarrow x + y, 0 & \text{if } 0 \leq x + y \leq m \\ x, y \rightarrow m, x + y - m & \text{if } x + y > m \\ x, y \rightarrow -m, x + y + m & \text{if } x + y < -m \end{array}$$

(In each case first agent on right hand side is the leader.)

Each agent also remembers output of last leader it met.

Correctness

Sum of agents' values is invariant.

Sum is eventually gathered into the unique leader
(up to maximum absolute value of m) :

If $sum > m$, leader has value m $\Rightarrow$ Output Yes.
If $sum < -m$, leader has value -m $\Rightarrow$ Output No.
If $-m \leq sum \leq m$, leader's value is the actual sum $\Rightarrow$ Output depends on sum.

In each case, leader knows output and tells everyone else.

◂ Back

How to Compute $\sum_{i=1}^{k} c_i x_i \equiv a \pmod{b}$

Very similar (and easier) :
gather sum into one agent, then disseminate answer.

Back