LANGAGES FORMELS, CALCULABILITÉ, COMPLEXITÉ

Calculabilité distribuée : Protocoles de population

Yoann Bourse

2009-2010 : Semestre 1

Plan de la présentation

Principe

Nous étudions ici un modèle de calcul distribué : **les protocoles de population**.

- Agents mobiles que l'expérimentateur ne controle pas.
- Non différenciés
- Faibles ressources : communication courte portée, petite mémoire.

Eléments bon marché et répandus :
Capteurs sur des animaux, puces RFID...

Principe

Nous étudions ici un modèle de calcul distribué : **les protocoles de population**.

- Agents mobiles que l'expérimentateur ne controle pas.
- Non différenciés
- Faibles ressources : communication courte portée, petite mémoire.

Eléments bon marché et répandus :
Capteurs sur des animaux, puces RFID...

Principe

Nous étudions ici un modèle de calcul distribué : **les protocoles de population**.

- Agents mobiles que l'expérimentateur ne controle pas.
- Non différenciés
- Faibles ressources : communication courte portée, petite mémoire.

Eléments bon marché et répandus :
Capteurs sur des animaux, puces RFID...

Principe

Nous étudions ici un modèle de calcul distribué : **les protocoles de population**.

- Agents mobiles que l'expérimentateur ne controle pas.
- Non différenciés
- Faibles ressources : communication courte portée, petite mémoire.

Eléments bon marché et répandus :
Capteurs sur des animaux, puces RFID...

Principe

Nous étudions ici un modèle de calcul distribué : **les protocoles de population**.

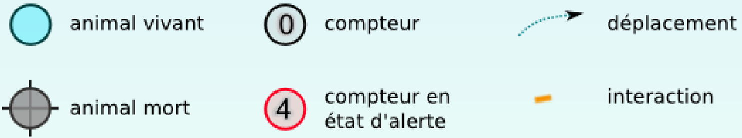
- Agents mobiles que l'expérimentateur ne controle pas.
- Non différenciés
- Faibles ressources : communication courte portée, petite mémoire.

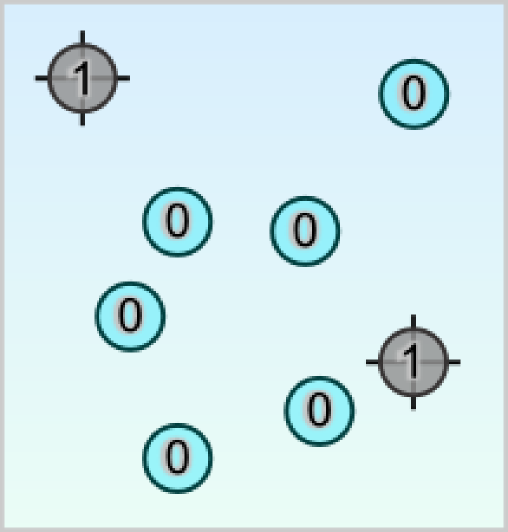
Eléments bon marché et répandus :
Capteurs sur des animaux, puces RFID...

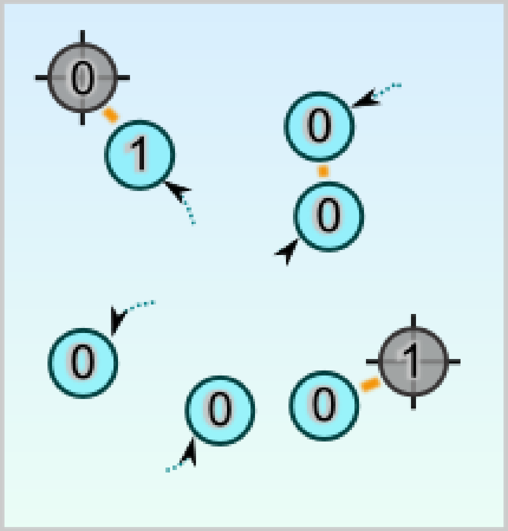
Compteur et surveillance animale

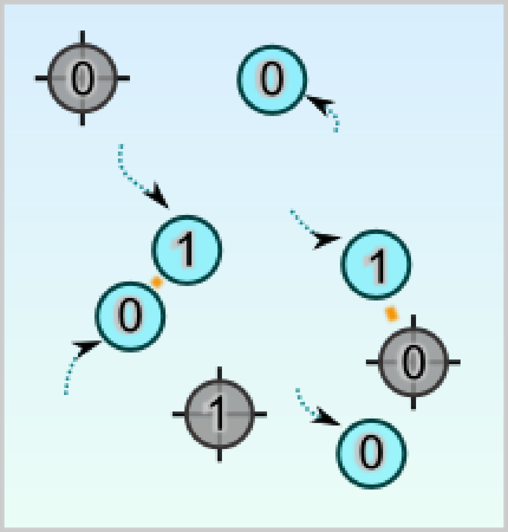
Objectif : Compter le nombre de morts dans une population animale.

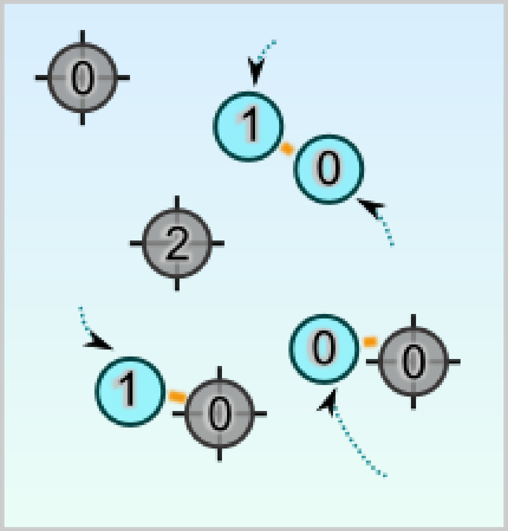
Terminaison : Etat d'alerte à n morts.

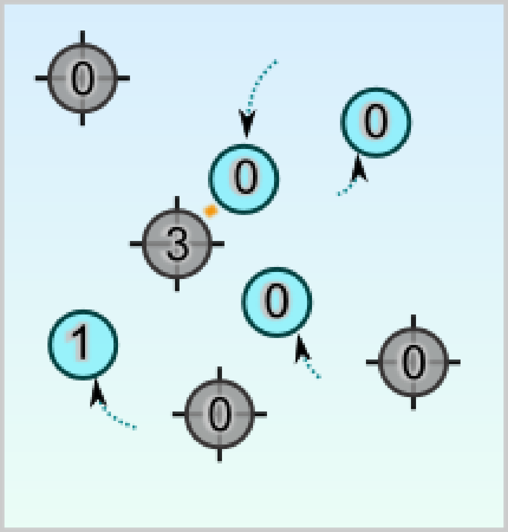


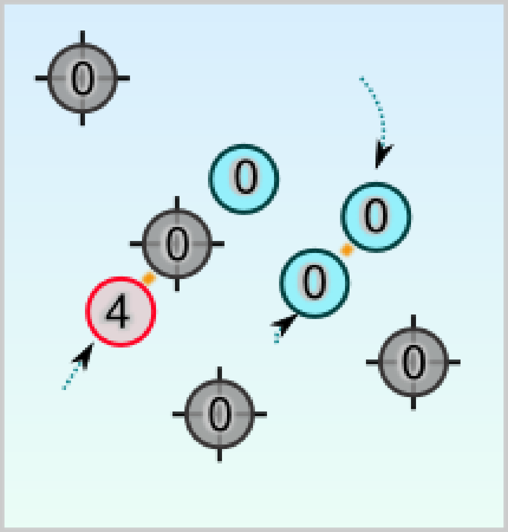

1
0
0
0
0
1
0
0

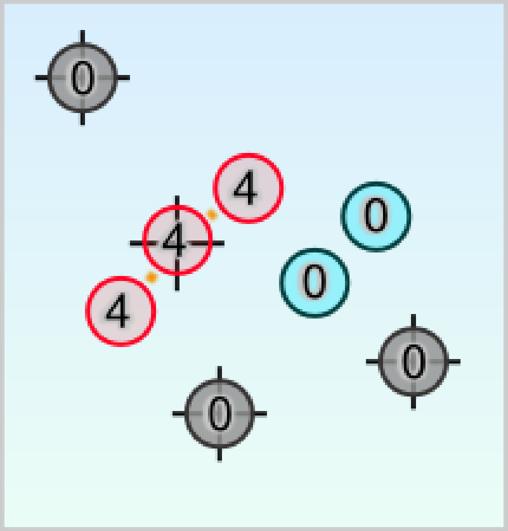

0
1
0
0
0
1
0
0

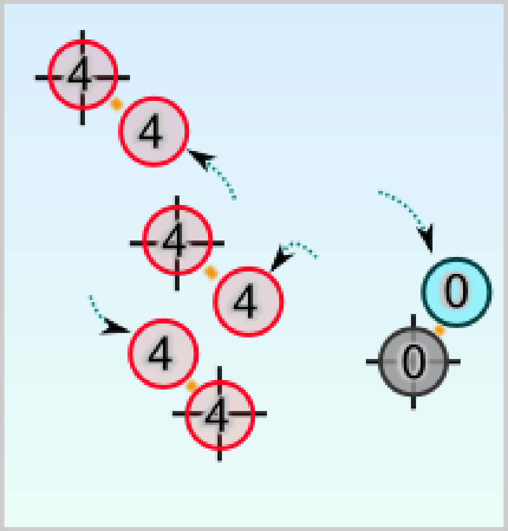

0
0
1
1
0
0
1
0


0
1
0
2
1
0
0
0


0
0
0
3
0
1
0
0
0


0
0
0
0
0
0
4
0
0


0
4
0
4
0
4
0
0


4
4
4
4
4
4
0
0

Comparaison, utilisation pour l'homme

Objectif : Déterminer le sexe majoritaire des clients d'un magasin

- Meme sexe, pas de changement :
  (♂,♂)⇒(♂,♂) ; (♀,♀)⇒(♀,♀)
- Sexes opposés s'annulent deux à deux :
  (♂,♀)⇒(○,○) ; (♀,♂)⇒(○,○)
- Etat neutre :
  (♂,○)⇒(♂,○) ; (♀,○)⇒(♀,○)...

Terminaison : Lecture de la puce du dernier client (meneur).

Comparaison, utilisation pour l'homme

Objectif : Déterminer le sexe majoritaire des clients d'un magasin

- Meme sexe, pas de changement :
  (♂,♂)⇒(♂,♂) ; (♀,♀)⇒(♀,♀)
- Sexes opposés s'annulent deux à deux :
  (♂,♀)⇒(○,○) ; (♀,♂)⇒(○,○)
- Etat neutre :
  (♂,○)⇒(♂,○) ; (♀,○)⇒(♀,○)...

Terminaison : Lecture de la puce du dernier client (meneur).

Comparaison, utilisation pour l'homme

Objectif : Déterminer le sexe majoritaire des clients d'un magasin

- Meme sexe, pas de changement :
  (♂,♂)⇒(♂,♂) ; (♀,♀)⇒(♀,♀)
- Sexes opposés s'annulent deux à deux :
  (♂,♀)⇒(○,○) ; (♀,♂)⇒(○,○)
- Etat neutre :
  (♂,○)⇒(♂,○) ; (♀,○)⇒(♀,○)...

Terminaison : Lecture de la puce du dernier client (meneur).

Comparaison, utilisation pour l'homme

Objectif : Déterminer le sexe majoritaire des clients d'un magasin

- Meme sexe, pas de changement :
  (♂,♂)⇒(♂,♂) ; (♀,♀)⇒(♀,♀)
- Sexes opposés s'annulent deux à deux :
  (♂,♀)⇒(○,○) ; (♀,♂)⇒(○,○)
- Etat neutre :
  (♂,○)⇒(♂,○) ; (♀,○)⇒(♀,○)...

Terminaison : Lecture de la puce du dernier client (meneur).

Calculs parallèles, combinaisons booléennes

Travail sur des paires de symboles.

Par exemple, on peut distinguer adolescent $\bullet$ et adulte $\bigotimes$.

La sortie correspond au résultat attendu si le meneur final est ♀$\wedge\bullet$.

Formalisation

Sur une population $\mathcal{P}$ de n agents.

Protocole de population

- Alphabet d'entrée A_e et de sortie A_s.
- Ensemble d'états Q.
- Fonctions d'entrée et de sortie.
- Fonction de transition $\delta : Q \times Q \rightarrow Q \times Q$.

Configuration

Fonction $C : \mathcal{P} \rightarrow Q$ associant à chaque agent un état.

Relation d'interaction

Formalisation

Sur une population $\mathcal{P}$ de n agents.

Protocole de population

- Alphabet d'entrée A_e et de sortie A_s.
- Ensemble d'états Q.
- Fonctions d'entrée et de sortie.
- Fonction de transition $\delta : Q \times Q \rightarrow Q \times Q$.

Configuration

Fonction $C : \mathcal{P} \rightarrow Q$ associant à chaque agent un état.

Relation d'interaction

Formalisation

Sur une population $\mathcal{P}$ de n agents.

Protocole de population

- Alphabet d'entrée A_e et de sortie A_s.
- Ensemble d'états Q.
- Fonctions d'entrée et de sortie.
- Fonction de transition $\delta : Q \times Q \longrightarrow Q \times Q$.

Configuration

Fonction $C : \mathcal{P} \longrightarrow Q$ associant à chaque agent un état.

Relation d'interaction

Formalisation

Sur une population $\mathcal{P}$ de n agents.

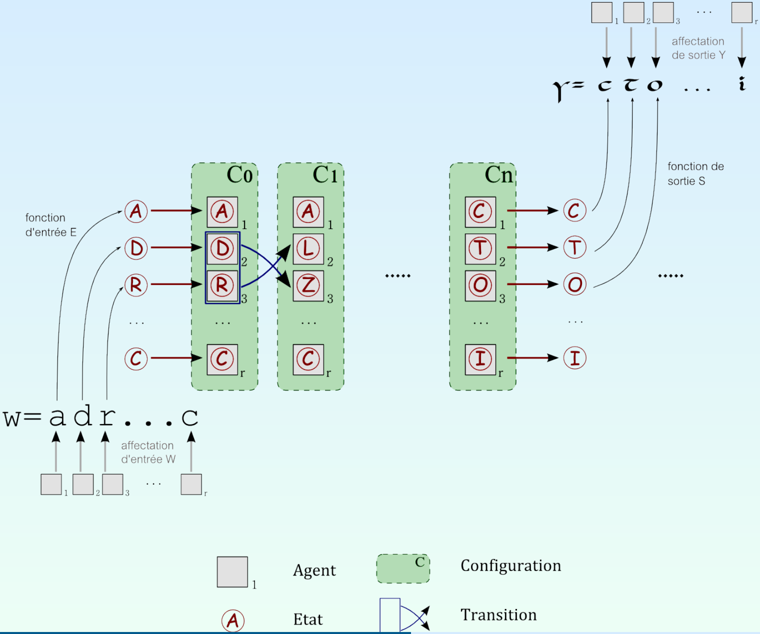
Protocole de population

- Alphabet d'entrée A_e et de sortie A_s.
- Ensemble d'états Q.
- Fonctions d'entrée et de sortie.
- Fonction de transition $\delta : Q \times Q \longrightarrow Q \times Q$.

Configuration

Fonction $C : \mathcal{P} \longrightarrow Q$ associant à chaque agent un état.

Relation d'interaction


affectation
de sortie Y
C0
C1
Cn
fonction de
sortie S
fonction
d'entrée E
w=adr...c
affectation
d'entrée W
Agent
Configuration
Etat
Transition

Calcul

Suite de transitions de la forme $C \longrightarrow C'$ telle que si C apparait un nombre infini de fois, alors C' aussi.

Configuration stable

Pour toute configuration C' accessible depuis C, le mot de sortie reste le même.
Le calcul est dit convergent.

Calcul d'une relation

Si pour tout mot d'entrée w tous les calculs possibles convergent, le protocole calcule la relation $R(w, y)$:
Il existe un calcul d'entrée w se stabilisant sur la sortie y.

Calcul

Suite de transitions de la forme $C \longrightarrow C'$ telle que si C apparait un nombre infini de fois, alors C' aussi.

Configuration stable

Pour toute configuration C' accessible depuis C, le mot de sortie reste le même.
Le calcul est dit convergent.

Calcul d'une relation

Si pour tout mot d'entrée w tous les calculs possibles convergent, le protocole calcule la relation $R(w, y)$:
Il existe un calcul d'entrée w se stabilisant sur la sortie y.

Calcul

Suite de transitions de la forme $C \longrightarrow C'$ telle que si C apparait un nombre infini de fois, alors C' aussi.

Configuration stable

Pour toute configuration C' accessible depuis C, le mot de sortie reste le même.
Le calcul est dit convergent.

Calcul d'une relation

Si pour tout mot d'entrée w tous les calculs possibles convergent, le protocole calcule la relation $R(w, y)$:
Il existe un calcul d'entrée w se stabilisant sur la sortie y.

Et en particulier, pour les formules logiques :

Prédicat

Fonction ϕ_F associée à la formule logique F de variables libres $(X_1 \ldots X_n)$ qui à chaque uplet de $(u_1 \ldots u_n)$ associe la **valeur de vérité de F** quand les X_i sont interprétés par les u_i.

Calcul d'un prédicat

Pour toute entrée $(u_1, \ldots, u_n)$, la sortie du protocole est $11\ldots1$ si et seulement si $\phi_F(u_1, \ldots, u_n) = 1$ et $00\ldots0$ si et seulement si $\phi_F(u_1, \ldots, u_n) = 0$.

Il existe diverses conventions d'entrée et de sortie.

Et en particulier, pour les formules logiques :

Prédicat

Fonction ϕ_F associée à la formule logique F de variables libres $(X_1 \ldots X_n)$ qui à chaque uplet de $(u_1 \ldots u_n)$ associe la **valeur de vérité de F** quand les X_i sont interprétés par les u_i.

Calcul d'un prédicat

Pour toute entrée $(u_1, \ldots, u_n)$, la sortie du protocole est $11\ldots1$ si et seulement si $\phi_F(u_1, \ldots, u_n) = 1$ et $00\ldots0$ si et seulement si $\phi_F(u_1, \ldots, u_n) = 0$.

Il existe diverses conventions d'entrée et de sortie.

Arithmétique de Presburger

Quelle est la portée calculatoire du modèle ?

Arithmétique de Presburger

C'est l'arithmétique classique (Peano) sans les axiomes de la multiplication : $(\mathbb{N}, +)$ en est un modèle.

Lemme

Tout prédicat peut s'écrire sans quantificateurs $(\exists, \forall)$ si on ajoute la relation de congruence $\equiv_m$.

Arithmétique de Presburger

Quelle est la portée calculatoire du modèle ?

Arithmétique de Presburger

C'est l'arithmétique classique (Peano) sans les axiomes de la multiplication : $(\mathbb{N}, +)$ en est un modèle.

Lemme

Tout prédicat peut s'écrire sans quantificateurs $(\exists, \forall)$ si on ajoute la relation de congruence $\equiv_m$.

Arithmétique de Presburger

Quelle est la portée calculatoire du modèle ?

Arithmétique de Presburger

C'est l'arithmétique classique (Peano) sans les axiomes de la multiplication : $(\mathbb{N}, +)$ en est un modèle.

Lemme

Tout prédicat peut s'écrire sans quantificateurs $(\exists, \forall)$ si on ajoute la relation de congruence $\equiv_m$.

Calculabilité

Calculabilité de l'arithmétique de Presburger

L'ensemble des prédicats calculables par des protocoles de population est exactement l'arithmétique de Presburger.

Montré par Angluin & Al, 2004-06.

Complexité

Automate conjugant

La paire d'agents interagissant parmi les paires possible est choisie aléatoirement.

Complexité des calculs

Tout prédicat calculable stablement par un protocole de population est calculable avec la probabilité 1 par un automate conjugant en un nombre total de $O(n^2 log(n))$ interactions.

Complexité

Automate conjugant

La paire d'agents interagissant parmi les paires possible est choisie aléatoirement.

Complexité des calculs

Tout prédicat calculable stablement par un protocole de population est calculable avec la probabilité 1 par un automate conjugant en un nombre total de $O(n^2 log(n))$ interactions.

Simulation des machines de Turing

Un protocole de population est similaire à une machine à bande finie dont les cases vivent leur existence séparément.

Simulation d'une machine de Turing

- Complexité spatiale logarithmique sur un alphabet unaire
- Complexité temporelle $O(n^d)$ dans le pire des cas

Simulable sur des entrées de taille inférieures à n par un protocole sur une population de n agents : $\forall c$

- Complexité temporelle de $O(n^{d+2} log(n) + n^{2d+c+1})$
- Probabilité d'erreur de $O(n^{-c} log(n))$.

Simulation des machines de Turing

Un protocole de population est similaire à une machine à bande finie dont les cases vivent leur existence séparément.

Simulation d'une machine de Turing

- Complexité spatiale logarithmique sur un alphabet unaire
- Complexité temporelle $O(n^d)$ dans le pire des cas

Simulable sur des entrées de taille inférieures à n par un protocole sur une population de n agents : $\forall c$

- Complexité temporelle de $O(n^{d+2} log(n) + n^{2d+c+1})$
- Probabilité d'erreur de $O(n^{-c} log(n))$.

Conclusion

- Un modèle relativement puissant
- Omniprésent (peu coûteux)
- Capable de prélever et traiter des données de l'environnement en direct