# Formalization of Emergence in Multi-agent Systems

**Ba Linh Luong**[1], Yong Meng Teo[1], Claudia Szabo[2]
[1] National University of Singapore
[2] University of Adelaide

21 May 2013

ACM SIGSIM Conference on
Principles of Advanced Discrete Simulation (PADS)
Montreal, Canada

# Outline

- Motivation
- Objective
- Related work
- Grammar-based Approach
  - Formalization
  - Emergent Property States
  - Example: Boids Model
- Evaluation
- Summary

# Motivation

- **Emergence**: *system properties that cannot be derived from the properties of the individual entities*
  - Desirable or undesirable
- **Challenges**
  - Advance understanding of emergence
  - Lack of consensus on emergence
- **Propose Formalization**
  - Set of emergent property states
  - Reason about cause-and-effect of emergence

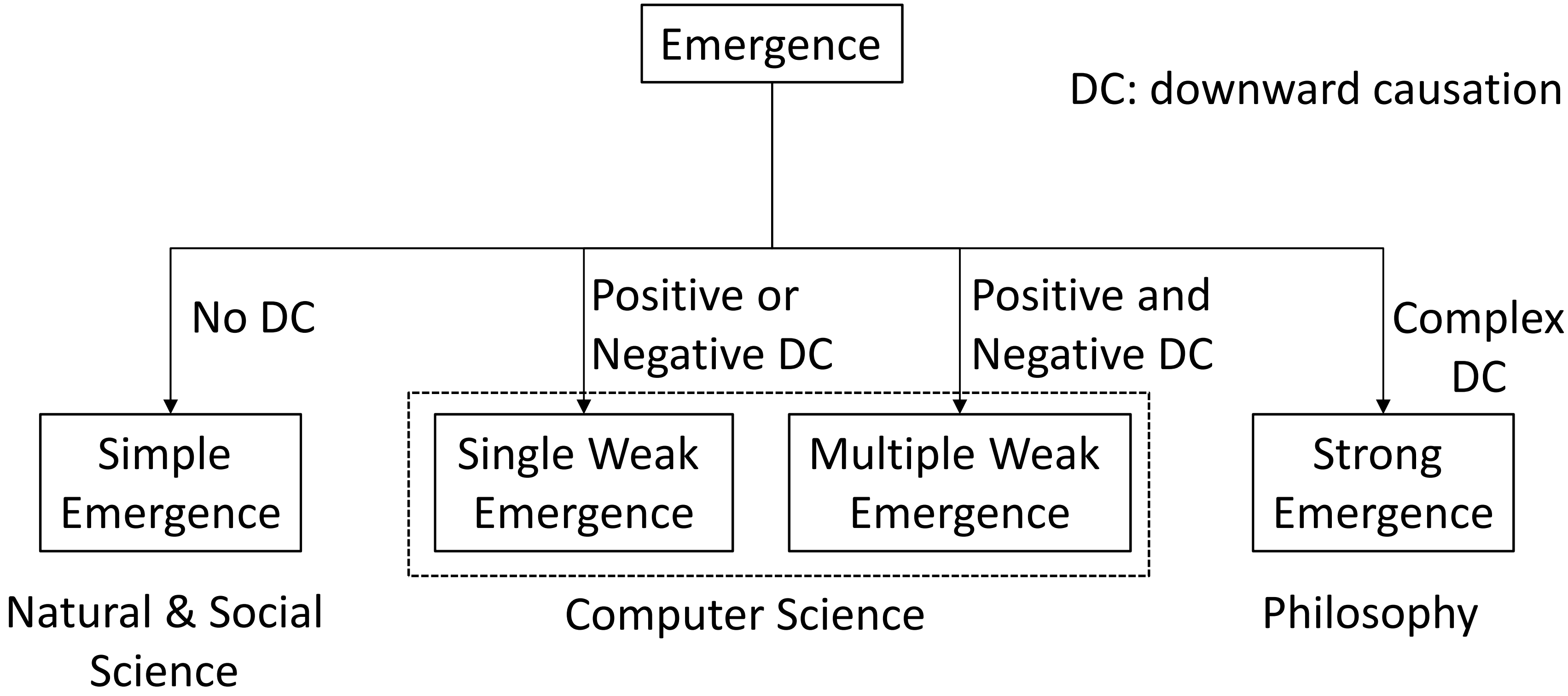
# Objective

A *formal approach* for determining the *set of emergent property states* in a given system.

# Emergence Perspectives

| | Perspective | How |
|---|---|---|
| Philosophy [6, 30] | Surprise - Limitations of our knowledge | Observer with correct scale |
| Natural & Social Science [2, 11, 16] | Observer-independent | Self-organization, hierarchy |
| Computer Science [6, 8, 22] | Derived from entity interactions (weak emergence) | Simulation |

# Types of Emergence



# Emergence Formalization

| Approach | Prior Knowledge | Analysis |
| --- | --- | --- |
| Variable-based [15, 26, 35] | required | post-mortem |
| Event-based [10] | required | post-mortem |
| Grammar-based [22] | not required | on-the-fly |

# Outline

- Motivation
- Objective
- Related work
- Grammar-based Approach
  - Formalization
  - Emergent Property States
  - Example: Boids Model
- Evaluation
- Summary

# Grammar-based Approach

- Kubik's approach*: "The whole is greater than the sum of its parts."*

- Main idea: determine the set of system states that are in the whole but not in the sum ($\mathrm{L}_\xi$)

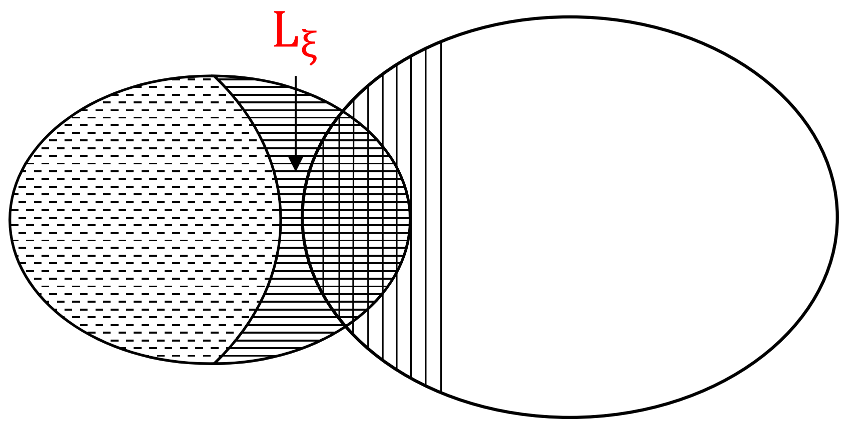
$$\mathrm{L}_\xi = \mathrm{L}_{\mathrm{whole}} \setminus \mathrm{L}_{\mathrm{sum}}$$

  - $\mathrm{L}_{\mathrm{whole}}$: set of all system states obtained by simulation
  - $\mathrm{L}_{\mathrm{sum}}$ : set of all permutations of states of individual parts

# Limitations

- Suffers from state-space explosion ($\mathrm{L}_{\mathrm{whole}}, \mathrm{L}_{\mathrm{sum}}$)
  [ next slide ]
- Cannot model agent types
  [ introduce agent type, $A_{ij}$ type i ($1 \leq i \leq m$) ]
- No support for mobile agents
  [ define mobility as attributes of agents:
  $P_i = P_{i\_mobile} \cup P_{i\_others}$ ]
- Closed systems with fixed number of agents
  [ agents can enter and leave system ]

# Proposed Approach – Reduce State Space




$L^{NI}_{whole}$ $L^{I}_{whole}$ $L^{P}_{sum}$ $L^{NP}_{sum}$

$$L_{whole} = L^{I}_{whole} \cup L^{NI}_{whole}$$

$$L_{sum} = L^{P}_{sum} \cup L^{NP}_{sum}$$

$$L_{\xi} = L^{I}_{whole} \setminus L^{P}_{sum}$$

I: interest
NI: not interest
P: possible
NP: not possible

# Proposed Formalization

A system consisting of m agent types and n agents $A_{11}, \ldots, A_{mn_m}$ ($n = n_1 + \ldots + n_m$, $n_i$ agents of type i) interacting in an environment (2D grid) consisting of c cells is defined as

$$\mathbf{GBS} = (V_A, V_E, A_{11}, \ldots, A_{mn_m}, S(0))$$

- $\mathbf{A_{ij}}$: agent type i ($1 \leq i \leq m$) of instance j ($1 \leq j \leq n_i$)
- $\mathbf{V_A} = \bigcup_{i=1}^{m} V_{A_i}$: set of possible agent states for all agent types, and $\mathbf{V_{A_i}}$ denotes the set of possible states for agents of type i
- $\mathbf{V_E}$: set of possible cell states
- $\mathbf{V} = V_A \cup V_E$
- $\mathbf{S(t)} \in V^{c+n}$ : system state at time t

# Environment

- Cell (e)
  - $\mathbf{V_e}$: set of possible states of cell e
  - $\mathbf{s_e(t)} \in V_e$: state at time t

- Environment (E)
  - $\mathbf{V_E} = \bigcup_{e=1}^{c} V_e$
  - $\mathbf{S_E(t)} \in {V_E}^c$ : state at time t

# Agent

- Agent $A_{ij}$ ($1 \leq i \leq m$, $1 \leq j \leq n$), is defined as:

$$\mathbf{A_{ij} = (P_i, R_i, s_{ij}(0))}$$

  - $\mathbf{P_i}$: set of attributes for agents of type i

    $P_i = P_{i\_mobile} \cup P_{i\_others}$

    $\mathbf{P_{i\_mobile}} = \{x | x \text{ is an attribute that models mobility}\}$

  - $\mathbf{R_i}$: set of behavior rules for agents of type i

    $R_i = R_{i\_mobile} \cup R_{i\_others}$

    $\mathbf{R_i}$: $V_{A_i} \rightarrow V_{A_i}$ // $V_{A_i}$: set of possible states for agents of type i

  - $\mathbf{s_{ij}(t)} \in V_{A_i}$: state of $A_{ij}$ at time t

# Emergent Property States

- Set of emergent property states

$$\mathbf{L}_{\boldsymbol{\xi}} = \mathbf{L}^{\mathbf{I}}_{\mathbf{whole}} \setminus \mathbf{L}^{\mathbf{P}}_{\mathbf{sum}}$$

- Set of system states with agent coordination (GROUP)

$$\mathbf{L}^{\mathbf{I}}_{\mathbf{whole}} = \{\mathbf{w} \in \mathbf{V}^{\mathbf{c+n}} \mid \mathbf{S(0)} \Rightarrow^{*}_{\mathbf{GROUP}} \mathbf{w}\}$$

- Sum of states of individual agents

$$\mathrm{L_{sum}} = \text{superimpose}(\mathrm{L}(\mathrm{A}_{11}), \ldots, \mathrm{L}(\mathrm{A}_{\mathrm{mn_m}}))$$

$$\boldsymbol{L_{sum}} \Rightarrow^{\textbf{constraints}} \boldsymbol{L^{P}_{sum}}$$

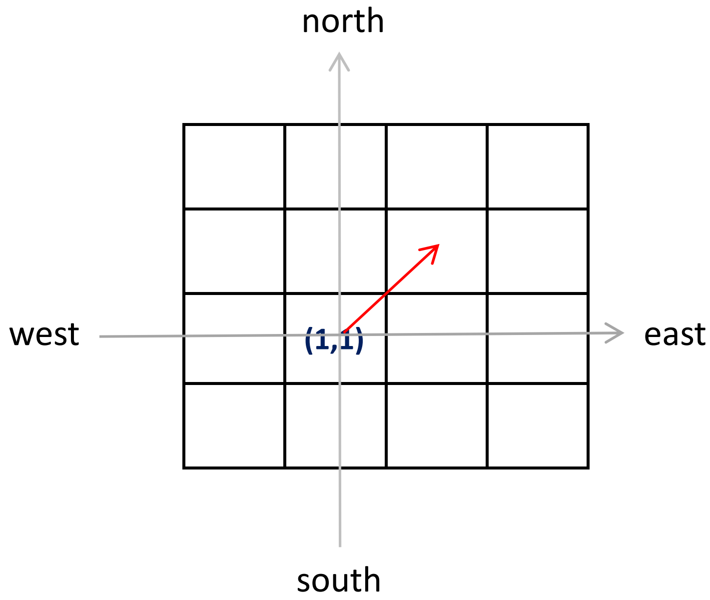
# Example

- Boids model [Reynolds87]
  - Separation (collision avoidance)
  - Alignment
  - Cohesion

- Two types of birds, five ducks and five geese, moving on a 8 x 8 grid
- Maximum speed: ducks (2 cells/step), geese (3 cells/step)
- Birds re-enter the system when they pass the grid edges

# Vector Representation of Velocity

| Direction | Speed | | |
|---|---|---|---|
| | 0 | 1 | 2 |
| **North** | (0,0) | (0,1) | (0,2) |
| **North-east** | (0,0) | (1,1) | (1,2), (2,1), (2,2) |
| **East** | (0,0) | (1,0) | (2,0) |
| **South-east** | (0,0) | (1,-1) | (1,-2), (2,-1), (2,-2) |
| **South** | (0,0) | (0,-1) | (0,-2) |
| **South-west** | (0,0) | (-1,-1) | (-1,-2), (-2,-1), (-2,-2) |
| **West** | (0,0) | (-1,0) | (-2,0) |
| **North-west** | (0,0) | (-1,1) | (-1,2), (-2,1), (-2,2) |



# Agents – Ducks

- Duck instance $A_{1j}$ $(1 \leq j \leq 5)$ is defined as

$$\mathbf{A_{1j} = (P_1, R_1, s_{1j}(0))}$$

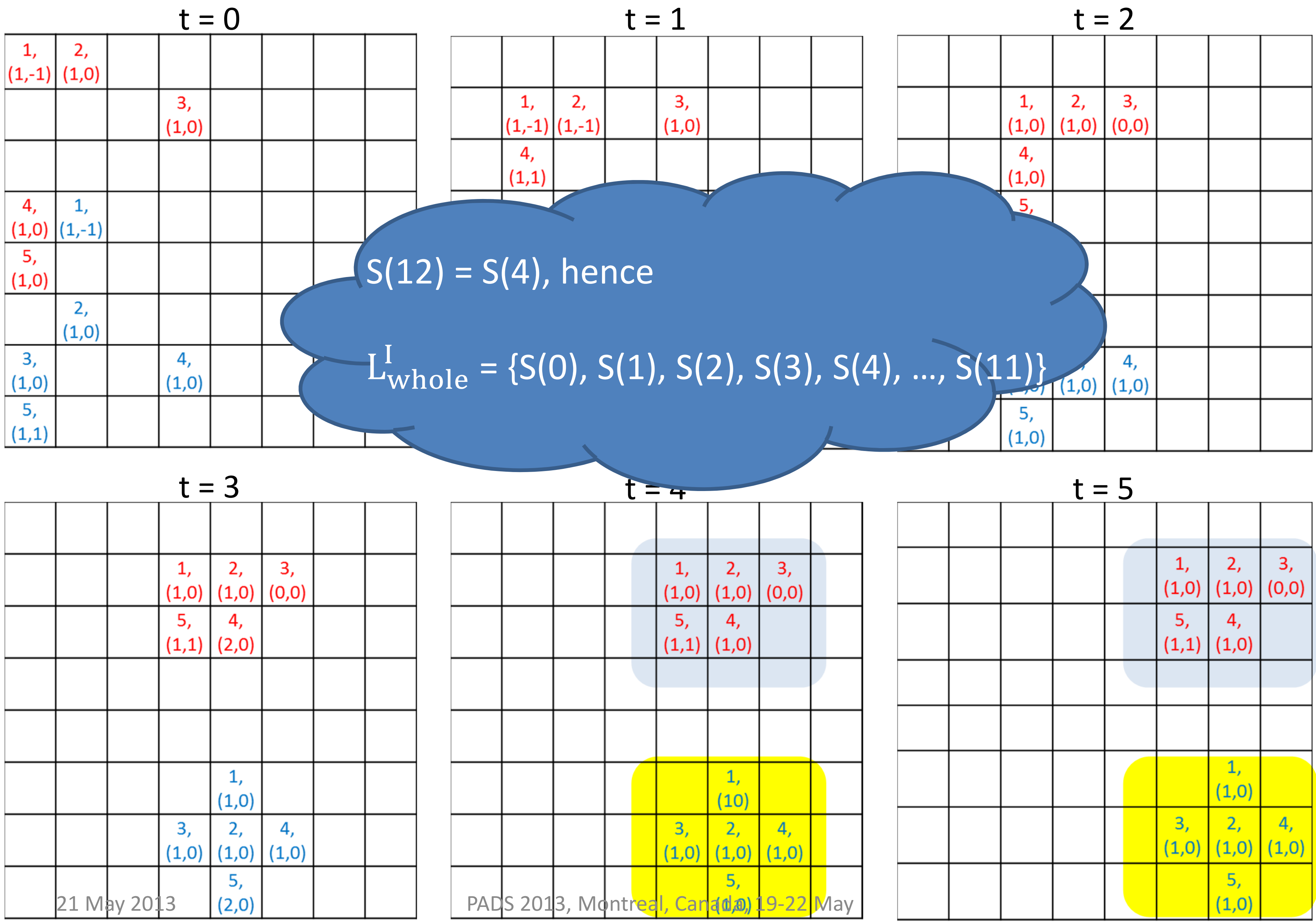
  - $P_1 = P_{1\_mobile} \cup P_{1\_others}$
    $\mathbf{P_{1\_mobile}} = \{position(g_{1j}), velocity(v_{1j})\}$, $\mathbf{P_{1\_others}} = \emptyset$
  - $\mathbf{V_{A_1}} = \{(x,y) \mid 1 \leq x \leq 8;\ 1 \leq y \leq 8\} \times \{(\alpha,\beta) \mid -2 \leq \alpha \leq 2;\ -2 \leq \beta \leq 2\}$
  - $R_1 = R_{1\_mobile} \cup R_{1\_others}$
    $\mathbf{R_{1\_mobile}}, \mathbf{R_{1\_others}} = \emptyset$
  - $\mathbf{s_{1j}(t)} \in V_{A_1}$

# Behavior Rules for Ducks – $R_{1\_mobile}$

- For duck instance $A_{1j}$ $(1 \leq j \leq 5)$ at time t with position $g_{1j}(t)$ and velocity $v_{1j}(t)$:
  - **Update position:**
    $g_{1j}(t+1) = g_{1j}(t) + v_{1j}(t+1)$
  - **Update velocity:**
    $$v_{1j}(t+1) = v_{1j}(t) + \text{separation}(A_{1j}) + \text{alignment}(A_{1j}) + \text{cohesion}(A_{1j})$$

**Similarly for Geese!**

# $L^{I}_{whole}$



# Emergent Property States

- $L_{\xi} = L^{I}_{whole} \setminus L_{sum}$ = {S(1), S(2), S(3), S(4), …, S(11)}

- **Flocking -** at least 4 birds of the same type fly together
  - Together - each bird has at least one immediate neighbor of the same type

1. known emergent states: S(2), S(3), S(4), …, S(11)
2. unknown emergent state: S(1)

# Experimental Results

- Java simulator
- Equal numbers of ducks and geese

| number of birds | number of states | | | $\frac{L_{\xi}}{L^{I}_{whole}}$ |
|---|---|---|---|---|
| | $L^{I}_{whole}$ | $L_{sum}$ | $L_{\xi}$ | |
| 4 | 13 | 767 | 6 | 0.46 |
| 6 | 18 | 70,118 | 12 | 0.67 |
| 8 | 13 | 509,103 | 9 | 0.69 |
| 10 | 26 | 13,314,006 | 23 | 0.88 |

# Summary

- Grammar-based set-theoretic approach
  - Reduce state space
  - Without a priori knowledge of emergence
  - Agents of different types, mobile agents, and open systems

- Example of boids model

- Open issues: reduce state space, reasoning of emergent property states

# Q & A

Thank You!

[teoym,luongbal]@comp.nus.edu.sg
claudia.szabo@adelaide.edu.au

Y. M. Teo, B. L. Luong, C. Szabo, **Formalization of Emergence in Multi-agent Systems**, ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, Montreal, Canada, May 19-22, 2013.

# Separation Rule

- Goal: avoid collision with nearby birds
- How: if duck **b** is close to another bird **a**, i.e. within ε cells, then **b** flies away from **a**

$$\text{separtion(b)} = \sum_{distance(a,b) \leq \varepsilon} b.position - a.position$$

```
separation(boid b)
   vector c = 0;
   for each boid a
        if |a.position - b.position| ≤ ε then
           c = c - (a.position - b.position)
 return c
```

# Alignment Rule

- Goal: fly as fast as nearby ducks
- How: change velocity of duck b λ% towards the average velocity of its neighbor ducks

$$\text{alignment(b)} = \left(\left(\sum_{\substack{duck(a)\\ neighbor(a,b)}}^{k} a.velocity\right)/k - b.velocity\right)/\lambda$$

```
Alignment(boid b)
  vector c = 0;
  integer k = 0;
  for each neighbor duck a
      k = k + 1;
      c = c + a.velocity;
   endfor
   c = c / k;
return (c - b.velocity) / λ
```

# Cohesion Rule

- Goal: stay close to nearby ducks
- How: move duck b γ% towards the center of its neighbor ducks

$$\text{cohesion(b)} = \Big(\Big(\sum_{\substack{duck(a)\\ neighbor(a,b)}}^{k} a.position\Big)/k - b.position\Big)/\gamma$$

```
Cohesion(boid b)
    vector c = 0;
    integer k = 0;
    for each neighbor duck a
        k = k + 1;
        c = c+ a.position;
    endfor
    c = c / k;
return (c - b.position) / γ
```

# $L_{sum}$

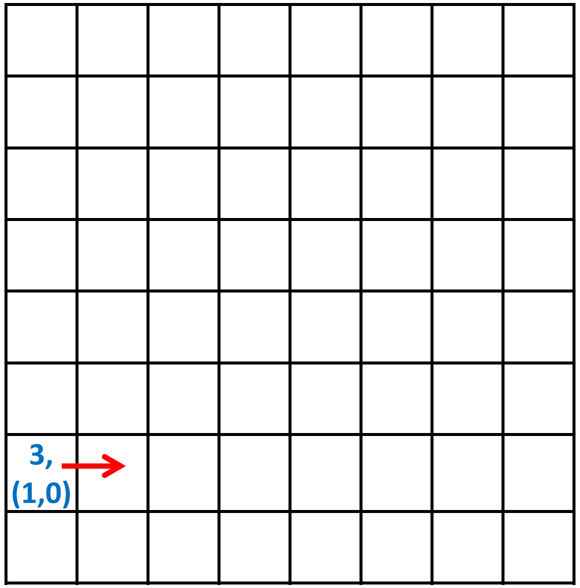
- $L_{sum}$ = superimpose(L($A_{11}$), …, L($A_{15}$), L($A_{21}$), …, L($A_{25}$))

- For illustration, consider two geese:

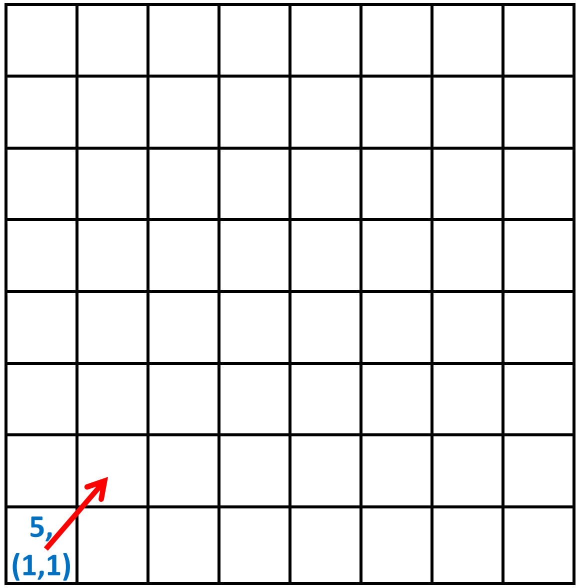
  $L_{sum}$ = superimpose(L($A_{23}$), L($A_{25}$))

  = L($A_{23}$) superimpose (L($A_{25}$)) U L($A_{25}$) superimpose (L($A_{23}$))

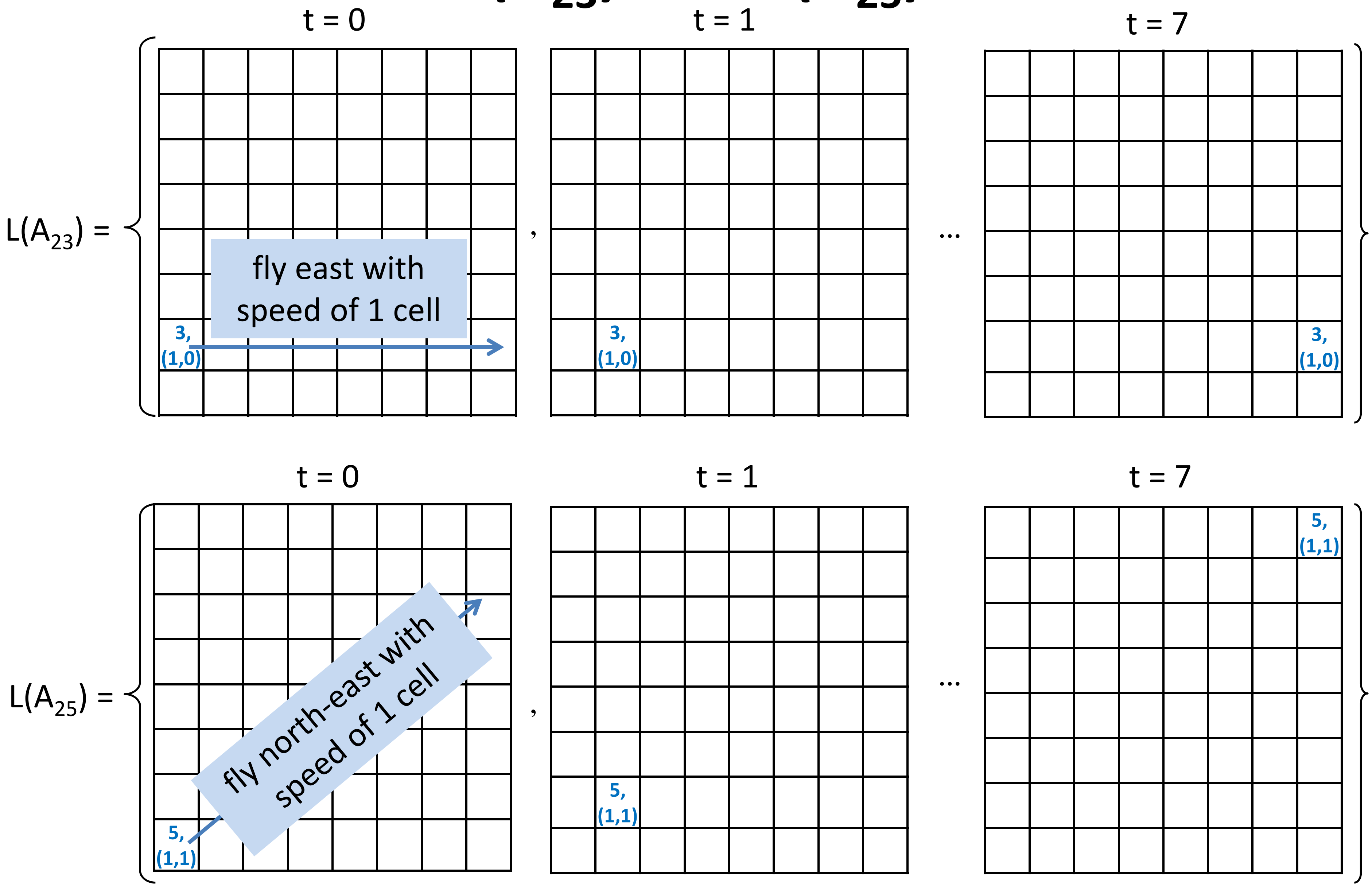
**$s_{23}$(0)**




**$s_{25}$(0)**



# $L(A_{23})$ and $L(A_{25})$



# $L_{sum}$ = superimpose(L($A_{23}$), L($A_{25}$))