نظریه بازیها
Game Theory

ارائه کننده: امیرحسین نیکوفرد
مهندسی برق و کامپیوتر دانشگاه خواجه نصیر

InfiniteDynamic Games

Material

- Dynamic Non-cooperative Game Theory: Second Edition
  - Chapter5: Sections 5:1–5:3.

InfiniteDynamic Games

- Zero sum games
- Non-zero sum games
- Infinite Games
- **Infinite Dynamic Games**
  - **Dynamic games in discrete time**
  - **Information structures**

Static v/s Dynamic(Recap)

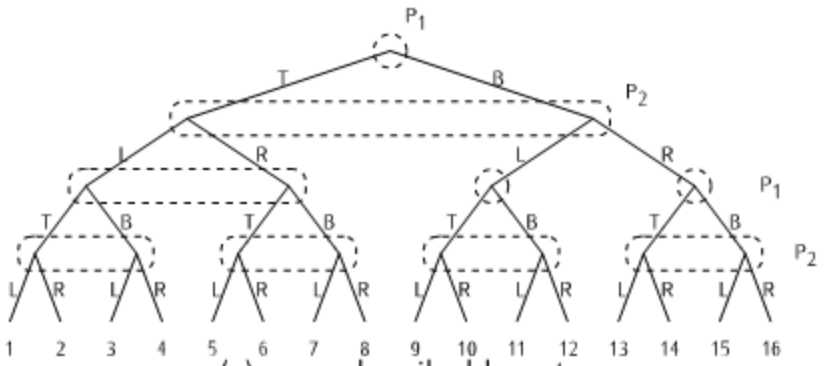
What does 'dynamic' refer to?

- Multi-stage games: Players gain dynamic information throughout the decision process
- Stages/Levels/Time: These games require a notion of time
  - Discrete-time Dynamic Games: Finite or countably infinite number of stages/levels of play
  - Continuous-time Dynamic Games: continuum of stages/levels of play

Dynamic games in discrete time

Consider a two-player multi-stage game in extensive form like the one in Figure (a) and suppose that we use the following notation:
For each stage, $k \in \{1,2,\ldots,K\}$

1. x_k denotes the node at which the game enters the kth stage,
2. u_k denotes the action of player P_1 at the kth stage,
3. d_k denotes the action of player P_2 at the kth stage.



(a) game described by a tree

Dynamic games in discrete time

In this case, the overall tree structure can be mathematically described by equations of the form:

$$\underbrace{x_{k+1}}_{\substack{\text{entry node}\\ \text{at stage } k+1}} = \underbrace{f_k}_{\substack{\text{"dynamics" at}\\ \text{stage } k}} \left(\underbrace{x_k}_{\substack{\text{entry node}\\ \text{at stage } k}} , \underbrace{u_k}_{\substack{P_1\text{'s action}\\ \text{at stage } k}} , \underbrace{d_k}_{\substack{P_2\text{'s action at}\\ \text{stage } k}} \right) \quad (1)$$

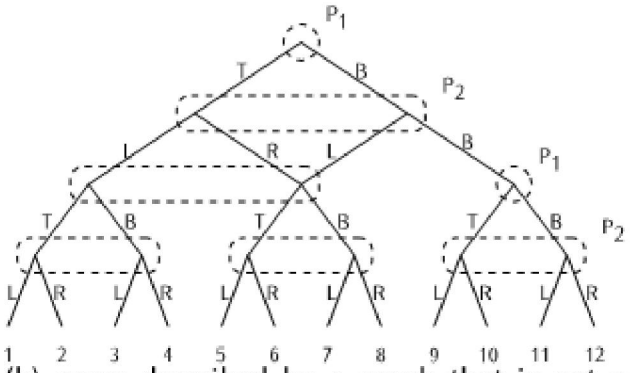
- Equation (1) describes the tree itself, but not the outcomes or the information sets.
- Evolution of Decision Process: dynamic state evolution with cost $J^i(x_k, u_k, d_k)$

Dynamic games in discrete time

This type of description actually allows for games that are more general that those that are typically described in extensive form. For example,

- games described by graphs that are not trees, such as the one in Figure (b);
- games with infinitely many stages ($k = \infty$);
- games with action spaces that are not finite sets.

Notation. A tree is (connected) graph that has no cycles.



(b) game described by a graph that is not a tree

Dynamic games in discrete time

Games whose evolution is represented by an equation such as (1) are called **dynamic games** and the equation (1) is often called the **dynamics of the game**. The set X where the state x_k takes values is called the **state-space** of the game.

- The outcome J_i for a particular player P_i, in a multi-stage game in extensive form like the one in Figure (a) is a function of the state of the game at the last stage K and the actions taken by the players at this stage: $J^i(x_k, u_k, d_k)$
- However, when the game is described by a graph that is not a tree, one may have different outcomes depending on how one got to the end of the game. In this case, the outcome J_i may depend on all the decisions made by both players from the start of the game:

$$J^i(x_1, u_1, d_1, \ldots, u_k, d_k)$$

Dynamic games in discrete time

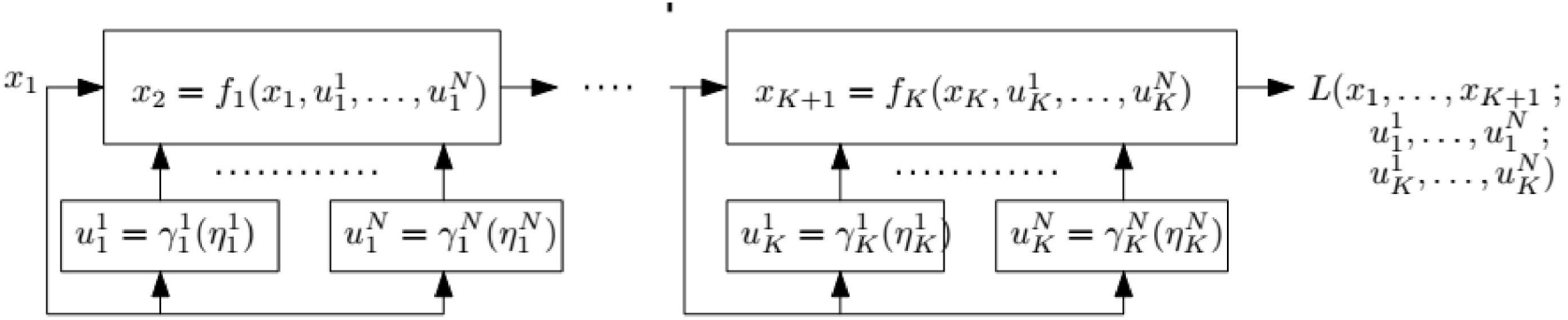
A dynamic game is said to have a **stage additive cost** when the outcome J_i to be minimized can be written as

$$\sum_{k=1}^{K} g_k^i(\mathrm{x}_k, \mathrm{u}_k, \mathrm{d}_k)$$

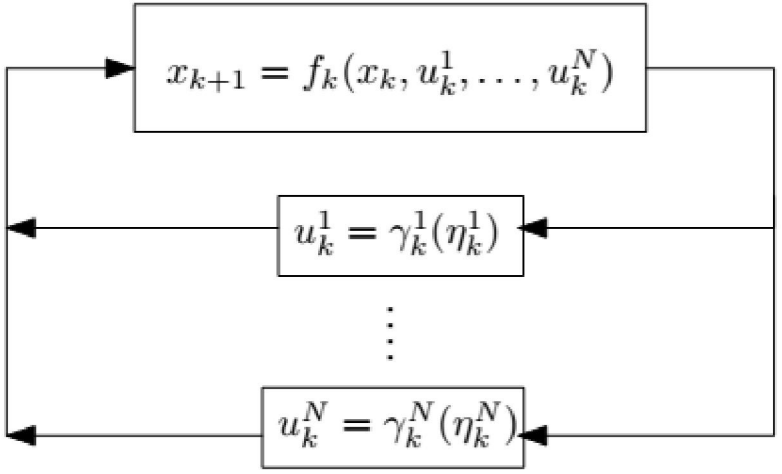
- when all g_k are equal to zero, except for the last one g_k, the game is said to have a **terminal cost**.
- Often one regards the stage index k as time and calls these **discrete-time dynamic games**, in contrast with differential games that take place in continuous times

Loop model

- We need to model the process:



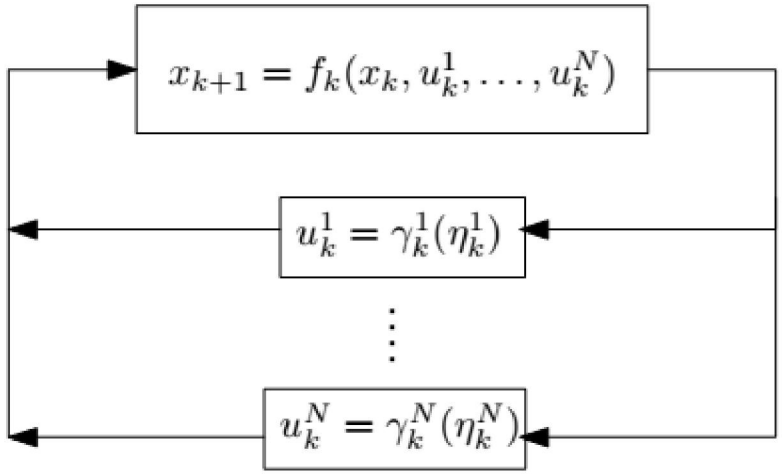
- Let's loop it around to get:



Loop model

- Basic elements of the loop model:
  - Set of Players: $P_i, i \in N, N := [1,\ldots,N]$
  - Stages: $k \in K, K := [1,\ldots,K]$
  - State Space: $x_k \in X \quad for \quad k \in K \cup K+1$
  - Action Space: $u_k^i \in U_k^i \quad , \quad u_k^{-i} \in U_k^{-i}$

$$U_k^{-i} := U_k^1 \times \ldots \times U_k^{i-1} \times U_k^{i+1} \times \ldots \times U_k^N, k \in K, i \in N$$



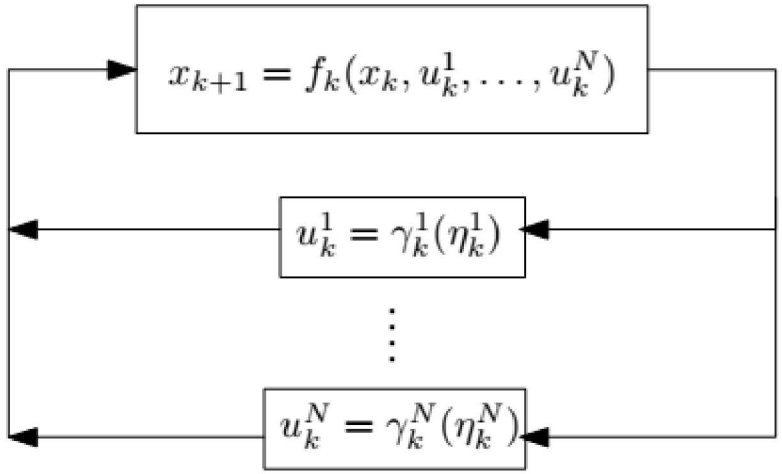
Loop model

- Information in the Loop Model:
  - Information Set: The information available to each player is

$$\eta_k^i \subseteq \{x_1,\ldots,x_k;u_1^1,\ldots,u_{k-1}^1;\ldots;u_1^N,\ldots,u_{k-1}^N\}, \quad i \in N$$

  - Information Space: $\eta_k^i \in N_k^i$

$$N_k^i \subseteq X^k \times U_1^1 \times \ldots \times U_{k-1}^1 \times \ldots \times U_1^N \times \ldots \times U_{k-1}^N$$



Loop model

- Functionals in the loop model:
  - State Equation: Describes the state evolution

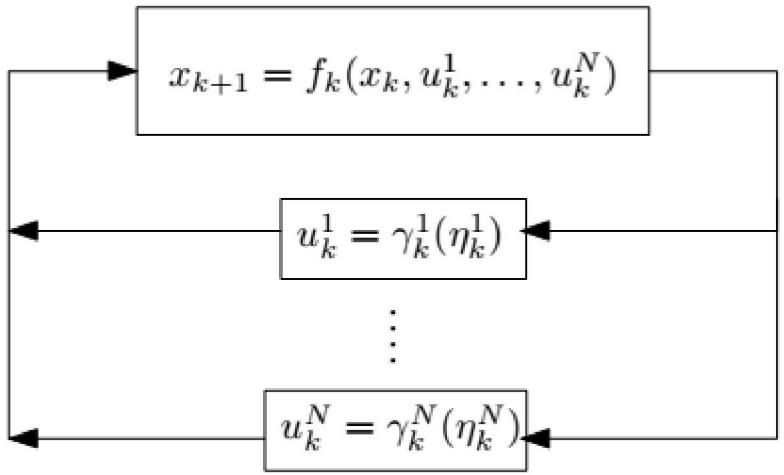
$$x_{k+1} = f_k(x_k, u_k^N, \ldots, u_k^N), \quad k \in K, x_k \in X$$

  - Strategies: P_i uses strategy $\gamma^i = \{\gamma_1^i, \ldots, \gamma_K^i\}$, $for\ \gamma^i \in \Gamma^i$

$$u_k^i = \gamma_k^i(\eta_k^i); \quad \gamma_k^i \in \Gamma_k^i$$

  - Cost Functional: The real-valued cost for P_i is given by

$$L^i(x_1, u_1^1, \ldots, u_1^N; \ldots; x_k, u_K^1, \ldots, u_K^N)$$



Example of a Nonzero sum Game

- Multi-agent Formation Game: N-robots, with first order dynamics

$$x_{k+1}^{i} = x_{k}^{i} + h\,u_{k}^{i}$$

begin from an initial position $x_1 = [x_1^1, \ldots, x_1^N]$

- Each robot tries to get to a position x_D^i, such that the position $x_D = [x_D^1, \ldots, x_D^N]$ corresponds to one where the N robots are evenly spaced1m apart, over K time instants.

Example of a Nonzero sum Game

Multi-agent Formation Game using the Loop Model:

- **Basic Elements:**
  - N players , K stages
  - State Space: $x_k^i \in R; x_k = [x_k^1, \ldots, x_k^N] \in R^N$
  - Action Space $u_k^i \in [-1,1], u_k^{-i} \in [-1,1]^{N-1}$
- **Information**: Suppose that each robot has a perfect measurement of its state, and transmits this information to all other robots instantaneously. Then

$$\eta_k^i = \{x_1, \ldots, x_k\}, \quad i \in N$$

- Missing actions?

  No

Example of a Nonzero sum Game

Multi-agent Formation Game using the Loop Model:

- **Functionals:**
  - **State Equation:**

$$x_{k+1} = x_k + h\,[u_k^1, \ldots, u_k^N]^T$$

  - **Strategies**: In general, $u_k^i = \gamma_k^i(\eta_k^i)$. One example could be

$$u_k^i = \sum_{j \in N} k_k^{ji} x_k^j$$

  - **Cost Function:**

$$L^i(\ldots) = \sum_{k=1}^{K+1} (x_k^i - x_D^i)^T Q_x (x_k^i - x_D^i) + \sum_{k=1}^{K} (u_k^i)^T Q_u (u_k^i)$$

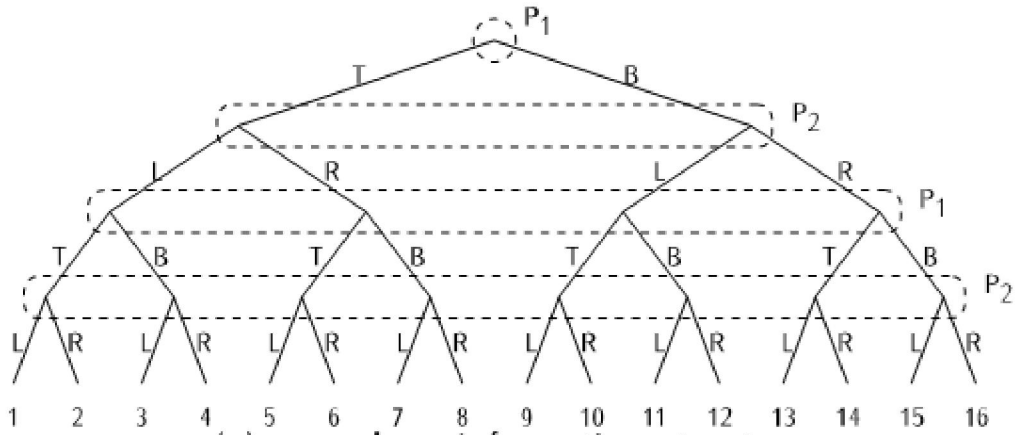
InfiniteDynamic Games

- Zero sum games
- Non-zero sum games
- Infinite Games
- **Infinite Dynamic Games**
  - Dynamic games in discrete time
  - **Information structures**

Information Structures

Dynamic games can have a wide range of information structures, but we will consider mostly some structures (such as open loop , (perfect) state feedback and etc.)

- In **open-loop (OL)** dynamic games, players do not gain any information as the game is played (other than the current stage) and must make their decisions solely based on a-priori information



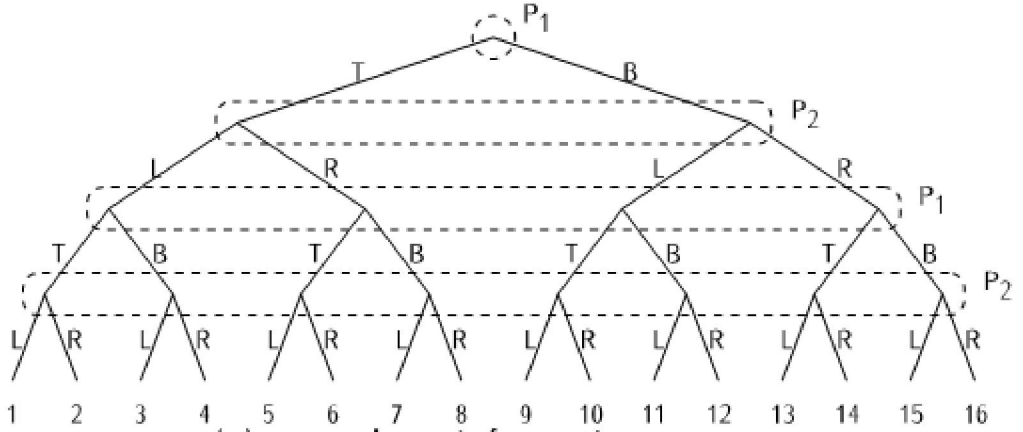
(a) open-loop information structure

Information Structures

In terms of an extensive form representation, each player has a single information set per stage, which contains all the nodes for that player at that stage, as in the game shown in Figure (a). For open-loop dynamic games, one typically represents *policies* as functions of the initial state.

$$\eta_k^i = x_1 \,, k \in K$$

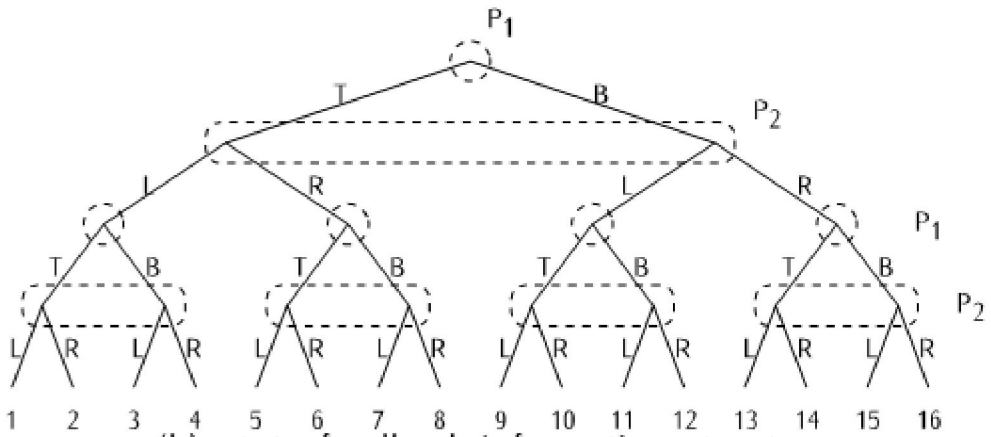
$$u_k^i = \gamma_k^{i(OL)}(\eta_k^i = x_1)$$



(a) open-loop information structure

Information Structures

In **(perfect) state-feedback (FB)** games, the players know exactly the state x_k of the game at the entry of the current stage and can use this information to choose their actions at that stage. However, they must make these decisions without knowing each others choice (i.e., we have simultaneous play at each stage).



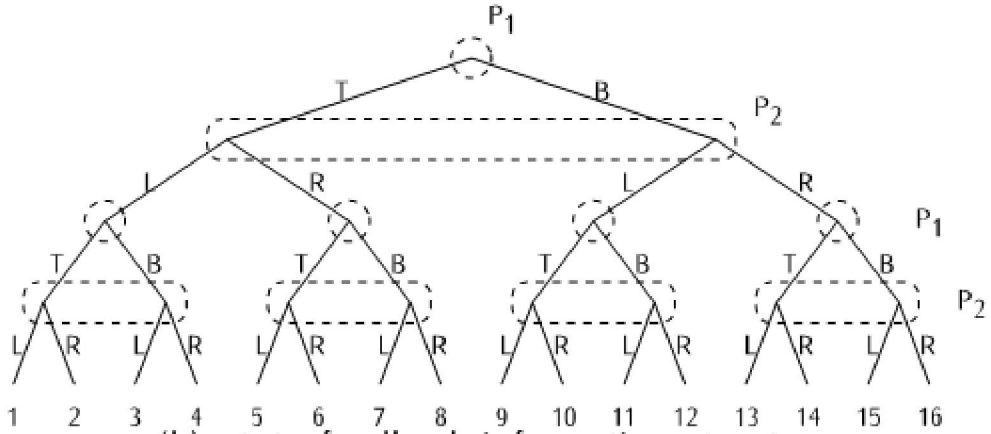
(b) state-feedback information structure

Information Structures

In terms of an extensive form representation, at each stage of the game there is exactly one information set for each entry-point to that stage, as in the game is shown in Figure (b). For state-feedback games, one typically represents *policies* as functions of the current state:

$$\eta_k^i = \{x_1, ..., x_k\}, k \in K$$

$$u_k^i = \gamma_k^{i(FB)}(\eta_k^i)$$



(b) state-feedback information structure

Information Structures

What if the full state is not available to the players?

- Observations (Measurements): $y_k^i \in Y_k^i$

$$y_k^i = h_k^i(x_k), \quad \mathrm{k} \in K, \mathrm{i} \in \mathrm{N}$$

- Information Structure (**Imperfect State Information**):

$$\eta_k^i \subseteq \{y_1^1,\ldots,y_k^1;\ldots;y_1^N,\ldots,y_k^N;$$
$$u_1^1,\ldots,u_k^1;\ldots;u_1^N,\ldots,u_k^N\}$$

- The qualifier "perfect" is sometime used to emphasize that the players know exactly the current state x_k, in contrast to situations in which the players only have access to estimates of x_k, possibly constructed from noisy measurements.

Information Structures (Example)

The robots use a low-quality GPS sensor to measure their positions:

- Observations (Measurements):

$$y_k^i = C_k^i x_k^i \ , \quad \mathrm{k} \in K, \mathrm{i} \in \mathrm{N}$$

- Information Pattern: Now, the robots transmit the measurements instantly to all other robots. Thus

$$\eta_k^i = \{y_1^1, \ldots.., y_k^1; \ldots; y_1^N, \ldots.., y_k^N\} \qquad i \in N$$

- Missing Actions?

No (under suitable assumptions)

Information Structures (Example)

Forgetful Players: Players forget some of the stored states/measurements/actions. For example: **Memoryless** players use only the current values of the state/observation to determine their actions.

- Information Set for Forgetful Players: In general -

$$\eta_k^i \subset \{y_1^1,....,y_k^1;\ldots;y_1^N,....,y_k^N\}$$

- Missing Actions? Possibly! Can be compensated by transmitting actions to other players as well.
- Example: $\eta_k^i = \{y_1^i, y_k^i\}, k \in K$

Information Structures (Example)

Delayed Information: The measured state/observation (or transmitted actions) are delayed by time steps.

- Information Set for Delayed Information: In general -

$$\eta_k^i \subseteq \{y_1^1,....,y_{k-\tau}^1;\ldots;y_1^N,....,y_{k-\tau}^N\}, \quad 1 \le \tau \le k-1$$

- Missing Actions? Possibly!

- Example: $\eta_k^i = \{y_1^1,....,y_{k-\tau}^1;\ldots;y_1^N,....,y_{k-\tau}^N\}$

Examples of Information Structures

A few typical information structures (IS):

- Open Loop IS: $\eta_k^i = x_1, k \in K$
- Closed-Loop Perfect State IS: $\eta_k^i = \{x_1, ..., x_k\}, k \in K$
- Closed-Loop Imperfect State IS: $\eta_k^i = \{y_1^i, ..., y_k^i\}, k \in K$
- Memoryless IS: $\eta_k^i = \{x_1, x_k\}, OR \quad \eta_k^i = \{y_1^i, y_k^i\}, k \in K$
- Feedback IS: $\eta_k^i = x_k, OR \quad \eta_k^i = y_k^i, k \in K$
- One step delayed IS:

$$\eta_k^i = \{x_1, \ldots, x_{k-1}\}, OR \quad \eta_k^i = \{y_1^i, \ldots, y_{k-1}^i\}, k \in K$$