Epistemic Planning Tutorial

Semantic Approach: Coordination, Plan Execution, Games

October 19, 2020
ICAPS 2020, Virtual Nancy
Implicit Coordination
Implicit Coordination
[Engesser et al., M4M 2017]

**Scenario**

Bob wants to borrow Anne’s apartment while she is away. Anne can leave the key behind for Bob to pick up.
Implicit Coordination

Example

Plan for Bob to get the key

1. Anne leaves the key under the door mat.
2. Bob takes the key from under door mat.

Works from an omniscient observer’s perspective.

Does not work from Bob’s perspective.

Why? At execution time, Bob does not know where the key is.
Implicit Coordination

Example

Alternative plan for Bob to get the key

1. Anne leaves the key under the door mat.
2. Anne tells Bob that the key is under the door mat.
3. Bob takes the key from under door mat.

When it’s Bob’s turn, Bob knows that his action is applicable and makes progress towards the goal.

Terminology: If this is the case for all plan steps, the plan is called implicitly coordinated (IC).
Implicit Coordination

Why?

Make sure at plan time that at execution time (when following the plan) everybody knows that their actions are applicable and make progress towards the goal.
Implicit Coordination

How?

- When planning for someone else to do the next step, take that agent’s perspective first.
- In DEL:
  
  Taking perspective of agent $i = \text{constructing agent } i$’s associated local state.
Implicit Coordination

How?

- When planning for someone else to do the next step, take that agent’s perspective first.
- In DEL:
  Taking perspective of agent $i = $ constructing agent $i$’s associated local state.
Implicit Coordination in DEL

Unsuccessful Plan

Anne: put-under-mat

Bob: take-from-mat

Bob: take-from-mat

Key-bob

Key-mat

Key-flowerbed

Key-mat

Key-flowerbed
Implicit Coordination in DEL
Successful Plan with Communication

anne: put-under-mat

anne: announce-mat

bob: take-from-mat
Implicit Coordination in DEL
Successful Plan with Sensing

Note: This is a branching plan, but that’s okay.
Implicit Coordination

- **IC ⇒** after I’ve done my action, the next agent I expect to act *knows* that they can do what I expect them to do, and that this will make progress towards the goal.

- **IC \( \not\Rightarrow \)** they actually *intend* to do what I expect them to do!

~~ compatibility of plans?~~

~~ success of plan executions?~~
Lazy and Eager Agents
Assumptions in this subsection:

- Each agent finds an IC plan (possibly including other agents’ actions) by itself.
- At execution time, profile of IC plans is executed in an interleaved manner.
- **Successful** execution: finite and ending in a (stable) goal state.
- In case of **conflicting observations**: replan.
Lazy Agents

An agent is called lazy if it prefers another agents’ action.

Example task: Knock, knock! Who gets the door?

The goal for Anne and Bob is to have the door open. Both agents are capable of opening the door.

What happens if both agents are lazy?
Lazy Agents

An agent is called lazy if it prefers another agents’ action.

Example task: Knock, knock! Who gets the door?

The goal for Anne and Bob is to have the door open. Both agents are capable of opening the door.

What happens if both agents are lazy?

Unsuccessful empty execution $\sim\sim$ eager agents?
**Naively Eager Agents**

An agent is called **naively eager** if it prefers its own actions.

⇒ no more deadlocks, but ...  

**Example task: pulling the lever (I)**

The goal, for **Lisa** and **Ralph**, is to pull the lever either fully to the left or to the right. Lisa can only pull left while Ralph can only pull right.
Naively Eager Agents

An agent is called naively eager if it prefers its own actions.

⇝ no more deadlocks, but ...

Example task: pulling the lever (Ⅰ)

The goal, for Lisa and Ralph, is to pull the lever either fully to the left or to the right. Lisa can only pull left while Ralph can only pull right.
Naively Eager Agents

An agent is called naively eager if it prefers its own actions.

⇒ no more deadlocks, but …

Example task: pulling the lever (I)

The goal, for Lisa and Ralph, is to pull the lever either fully to the left or to the right. Lisa can only pull left while Ralph can only pull right.
Naively Eager Agents

An agent is called naively eager if it prefers its own actions.

⇝ no more deadlocks, but …

Example task: pulling the lever (I)

The goal, for Lisa and Ralph, is to pull the lever either fully to the left or to the right. Lisa can only pull left while Ralph can only pull right.
Naively Eager Agents

An agent is called naively eager if it prefers its own actions.

⇾ no more deadlocks, but …

Example task: pulling the lever (I)

The goal, for Lisa and Ralph, is to pull the lever either fully to the left or to the right. Lisa can only pull left while Ralph can only pull right.
**Naively Eager Agents**

An agent is called *naively eager* if it prefers *its own actions*.

↝ *no more deadlocks*, but …

**Example task: pulling the lever (I)**

The goal, for **Lisa** and **Ralph**, is to pull the lever either fully to the left or to the right. Lisa can only pull left while Ralph can only pull right.
Naively Eager Agents

An agent is called naively eager if it prefers its own actions.

⇝ no more deadlocks, but …

Example task: pulling the lever (I)

The goal, for Lisa and Ralph, is to pull the lever either fully to the left or to the right. Lisa can only pull left while Ralph can only pull right.

What happens if both agents are naively eager?
**Naively Eager Agents**

![Diagram of Naively Eager Agents]

Lisa

Ralph
Naively Eager Agents
Naively Eager Agents
Naively Eager Agents
Naively Eager Agents

- Many possible infinite executions
- Solution idea: optimality
  (only pull if lever is on “your” side)
An agent is called optimally eager if it prefers its own action among the optimal ones.

⇝ no more infinite executions if problem is uniformly observable, but ...

Example task: Pulling the lever (II)

Same problem as before, but Lisa only knows about the leftmost setting being a goal setting, while Ralph only knows about the rightmost setting being one.

What happens if both agents are optimally eager?
Optimally Eager Agents
**Optimally Eager Agents**

- Lisa
- Ralph

- or

Epistemic Game Playing
Optimally Eager Agents

or

Lisa

Ralph
Optimally Eager Agents
Optimally Eager Agents
Optimally Eager Agents
Optimally Eager Agents

- Problem: notion of optimality is subjective.
- Generally, cannot prevent infinite executions.
  ↞ increased reasoning capability?
  ↞ additional coordination mechanism (⇝ tokens)?
  ↞ special cases (⇝ MAPF/DU)?
Token Protocol
Idea: Use Tokens as a Coordination Mechanism

[Engesser et al., KR 2020]

- Introduce **token** only one agent can possess at a time.
- Only token owner may act or pass on the token.
Idea: Use Tokens as a Coordination Mechanism
Idea: Use Tokens as a Coordination Mechanism

Lisa

Ralph
Idea: Use Tokens as a Coordination Mechanism
Idea: Use Tokens as a Coordination Mechanism

Lisa

Ralph

T
Tokenization: Formalization and Results

Syntactic tokenization of planning tasks:

- Add token fluent, add token passing actions.
- Token possession becomes action precondition.

Theoretical results:

- ✓ If all agents act w.r.t. optimal maximal strong policies, all executions are finite.

- ✓ Tokenization preserves solutions provided agents can always identify to whom to pass the token.

- ✗ Otherwise, tokenization may destroy solvability.

- More details: Thorsten Engesser’s DMAP presentation on Thursday, Oct. 22 (session at 12:00 UTC)
MAPF under Destination Uncertainty
MAPF under Destination Uncertainty

[Nebel et al., JAIR 2019]

Robot and Human Meeting at Narrow Intersection – Problem

It is common knowledge that the human does not know the robot's goal (east or south) and the robot does not know the human's goal (west or south). The robot actually wants to go east. It cannot communicate with the human. Should the robot wait or should it go out of the way (south)?
It is common knowledge that

- It is common knowledge that

- [Nebel et al., JAIR 2019]
It is common knowledge that

- the human does not know the robot’s goal (east or south)
It is common knowledge that

- the human does not know the robot’s goal (east or south)
- the robot does not know the human’s goal (west or south)
MAPF under Destination Uncertainty

[Nebel et al., JAIR 2019]

Robot and Human Meeting at Narrow Intersection – Problem

- It is common knowledge that
  - the human does not know the robot’s goal (east or south)
  - the robot does not know the human’s goal (west or south)
- The robot actually want to go east.
MAPF under Destination Uncertainty
[Nebel et al., JAIR 2019]
Robot and Human Meeting at Narrow Intersection – Problem

- It is common knowledge that
  - the human does not know the robot’s goal (east or south)
  - the robot does not know the human’s goal (west or south)
- The robot actually want to go east.
- It cannot communicate with the human.
It is common knowledge that

- the human does not know the robot’s goal (east or south)
- the robot does not know the human’s goal (west or south)

The robot actually want to go east.

It cannot communicate with the human.

Should the robot wait or should it go out of the way (south)?
MAPF under Destination Uncertainty

[Nebel et al., JAIR 2019]

Robot and Human Meeting at Narrow Intersection – Solution

Going south is an advancement towards the goal

Case 1: Human wants to go west:

Human can walk directly to his goal (west) enabling the robot to reach both potential goals (placeholder)
Going south is an advancement towards the goal
Going south is an advancement towards the goal

Case 1: Human wants to go west:
Going south is an advancement towards the goal

Case 1: Human wants to go west:
  - Human can walk directly to his goal (west)
MAPF under Destination Uncertainty

[Nebel et al., JAIR 2019]

Robot and Human Meeting at Narrow Intersection – Solution

- Going south is an advancement towards the goal
- Case 1: Human wants to go west:
  - Human can walk directly to his goal (west)
  - enabling the robot to reach both potential goals
Going south is an advancement towards the goal

Case 2: Human wants to go south:
MAPF under Destination Uncertainty

[Nebel et al., JAIR 2019]

Robot and Human Meeting at Narrow Intersection – Solution

- Going south is an advancement towards the goal
- Case 2: Human wants to go south:
  - Human can go out of the way (west)
MAPF under Destination Uncertainty

[Nebel et al., JAIR 2019]
Robot and Human Meeting at Narrow Intersection – Solution

- Going south is an advancement towards the goal
- Case 2: Human wants to go south:
  - Human can go out of the way (west)
  - enabling the robot to reach both potential goals
Going south is an advancement towards the goal

Case 2: Human wants to go south:
  - Human can go out of the way (west)
  - enabling the robot to reach both potential goals
  - enabling the human to reach both potential goals
MAPF under Destination Uncertainty

Assumptions

- **Common goal** of all agents: everybody reaches its destination.
- All agents know their own destinations.
- For each agent, there exists a set of possible destinations, which are common knowledge.
- All agents plan and re-plan **without communicating**.
MAPF under Destination Uncertainty

Results

- **Guaranteed success with polynomial executions** if all agents plan
  - optimally, i.e., generate (worst-case) shortest plans;
  - conservatively, i.e., replan from the initial state using the executed actions as a prefix;
  - eagerly, i.e., always plan to act when they can act (respecting optimality and conservativity).

- The backbone of plans are **stepping stones**.
  (A stepping stone for agent $i$ is a state in which $i$ can move to each of its possible destinations, announce success, and afterwards, for each possible destination, there exists an $i$-strong plan to solve the resulting states.)

- Deciding whether an implicitly coordinated plan with execution cost $k$ or less exists is **PSPACE-complete**.
Epistemic Game Playing
Example: Hanabi
DEL vs. GDL-III
Problems When Using DEL to Specify Games
[Engesser et al., IJCAI 2018]

- Combinatorial explosion of action model sizes
- E.g., $2^n$ events for independent sensing of $n$ propositions

Alternative: Game Description Language with Imperfect Information and Introspection (GDL-III)
GDL-III Exponentially More Concise than DEL

Multiple independent observations in one GDL-III action:

\[
\text{sees}(ag, p\text{IsTrue}) :- \text{does}(ag, \text{sense}), \text{true}(p).
\]
\[
\text{sees}(ag, q\text{IsTrue}) :- \text{does}(ag, \text{sense}), \text{true}(q).
\]
\[
\text{sees}(ag, r\text{IsTrue}) :- \text{does}(ag, \text{sense}), \text{true}(r).
\]
Observation Token Inspired Edge-Conditions

Edge-conditions \((\varphi_i, \psi_i), i = 1, \ldots, N\), between events \(e, e'\).

Cond. under which \(ag\) makes obs. \(i\) in event \(e\)

Cond. under which \(ag\) makes obs. \(i\) in event \(e'\)

\[ ag : \ldots, (\varphi_i, \psi_i), \ldots \]

- Product update easy to adapt.
  
  \(((w, e) \sim (w', e') \text{ if } w \sim w' \text{ and for all } i \leq N \text{ it holds that } [w \models \varphi_i \iff w' \models \psi_i])\).

- Allows compiling GDL-III actions into DEL actions compactly.

(cf. also Bolander et al.’s Edge-Conditioned Event Models [2018])
**DEL vs. GDL-III**

- **Translation** between large fragments of GDL-III and DEL possible.
- Requires extending DEL with the functionality of observation tokens.
- Allows combining:
  - compact and convenient representation of GDL-III
  - semantics of DEL
Summary
Summary

- **Implicit coordination**: planning with perspective taking
- **Success of plan profile execution** depends on agent types and their knowledge.
- **Tokens** can help.
- **Special case** MAPF/DU
- **GDL-III and DEL**: similar expressiveness