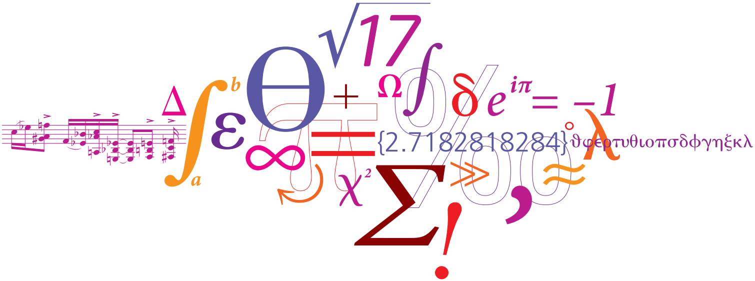
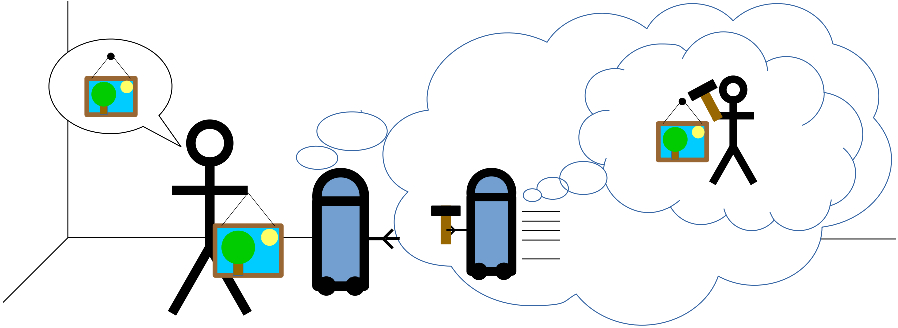
Epistemic Planning With Implicit Coordination

Thomas Bolander, DTU Compute, Technical University of Denmark
Joint work with Thorsten Engesser, Robert Mattmüller and Bernhard Nebel from Uni Freiburg



DTU Compute
Department of Applied Mathematics and Computer Science

Example: The helpful household robot



Essential features:

- **No instructions** are given to the robot.
- **Multi-agent planning**: The robot plans for both its own actions and the actions of the human.
- It does **(dynamic) epistemic reasoning**: It knows that the human doesn't know the location of the hammer, and plans to inform him.
- It is **altruistic**: Seeks to minimise the number of actions the human has to execute.

The problem we wish to solve

We are interested in decentralised multi-agent planning where:

- The agents form a **single coalition** with a **joint goal**.
- Agents may differ arbitrarily in **uncertainty about initial state** and **partial observability of actions** (including higher-order uncertainty).
- Plans are computed *by* **all agents,** *for* **all agents**.
- **Sequential execution**: At every time step during plan execution, one action is randomly chosen among the agents who wish to act.
- No explicit coordination/negotiation/commitments/requests. Coordination is achieved **implicitly** via observing action outcomes (e.g. ontic actions or announcement).

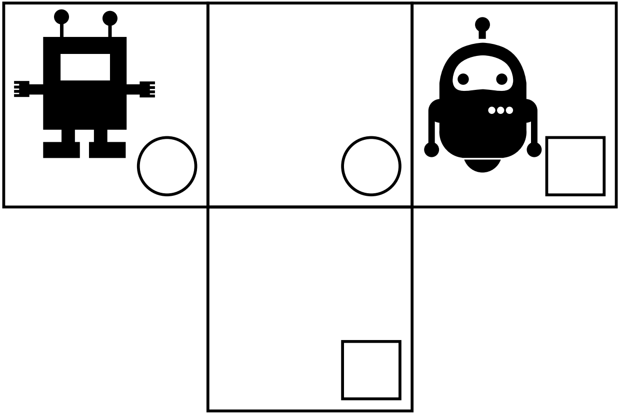
We call it *epistemic planning with implicit coordination*.

Based on the paper "Cooperative Epistemic Multi-Agent Planning With Implicit Coordination" [Engesser et al., 2015] + additional unpublished work.

Another example: Implicit robot coordination under partial observability

Joint goal: Both robots get to their respective goal cells.

They can move one cell at a time. A cell can only contain one robot. Both robots only know the location of their own goal cell.



A simpler example: Stealing a diamond

And now, finally, some technicalities...

Setting: Multi-agent planning under higher-order partial observability.

Natural formal framework: **Dynamic epistemic logic** (**DEL**) [Baltag et al., 1998]. We use DEL with postconditions [van Ditmarsch and Kooi, 2008].

Language:

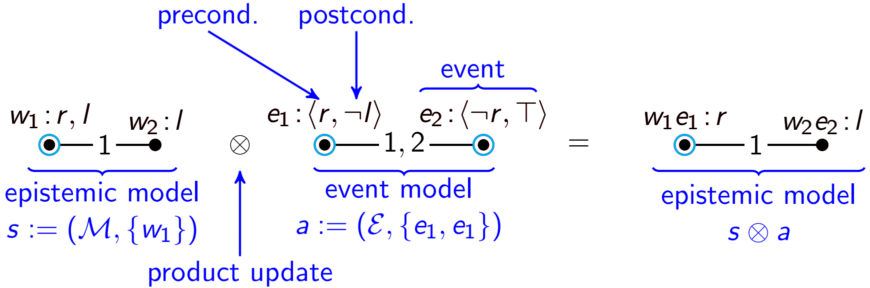
$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid K_i\phi \mid C\phi \mid (a)\phi,$$

where a is an (epistemic) action (to be defined later).

- $K_i\phi$ is read "agent i knows that ϕ".
- $C\phi$ is read "it is common knowledge that ϕ".
- $(a)\phi$ is read "action a is applicable and will result in ϕ holding".

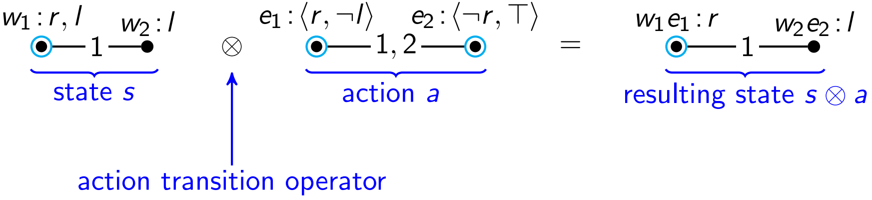
DEL by example: Cutting the red wire

I'm agent 0, my partner in crime is agent 1. r: The red wire is the power cable for the alarm. l: The alarm is activated. h: Have diamond. All indistinguishability relations are equivalence relations (S5).



- **Designated worlds/events** marked by ⊙.
- $s \models Cl \wedge K_0 r \wedge \neg K_1 r \wedge K_0 \neg K_1 r$. (Truth in a model means truth in all designated worlds)
- **Event model**: the action of cutting the red wire.
- $s \otimes a \models K_0 \neg l \wedge \neg K_1 \neg l \wedge K_0 \neg K_1 \neg l$.

Planning interpretation of DEL



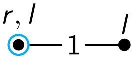
- **States**: Epistemic models.
- **Actions**: Event models.
- **Result of applying an action in a state**: Product update of state with action.
- **Semantics**: $s \models (a)\phi$ iff a is applicable in s and $s \otimes a \models \phi$.
- **Example**: $s \models (a)(\neg l \wedge \neg K_1 \neg l)$.

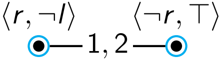
Planning to get the diamond

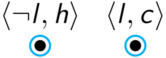
Definition. A **planning task** is $\Pi = (s_0, A, \omega, \phi_g)$ where

- s_0 is the **initial state**: an epistemic model.
- A is the **action library**: a finite set of event models called **actions**.
- $\omega : A \to Ag$ is an **owner** function: specifies who "owns" each action, that is, is able to execute it.
- ϕ_g is a **goal formula**: a formula of epistemic logic.

Example

- $s_0 =$ 

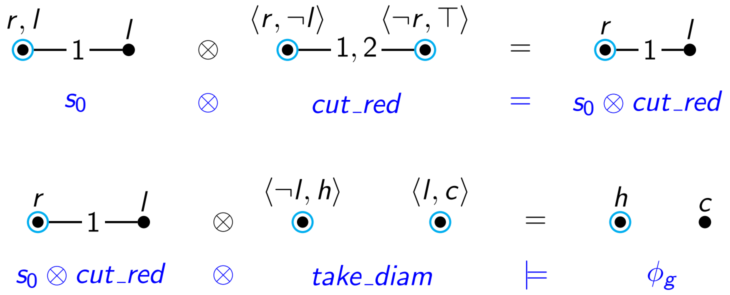
- $A = \{cut\_red, take\_diam\}$
- $\omega(cut\_red) = 0$; $\omega(take\_dia) = 1$
- $cut\_red =$ 

- $take\_diam =$ 

(where c: get caught)
- $\phi_g = h$

Example continued

Consider again the planning task Π from the previous slide (actions are *cut_red* and *take_diam*, goal is $\phi_g = h$). A plan exists for Π exists: $(cut\_red, take\_diam)$, since



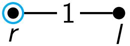
Expressed syntactically:

$$s_0 \models (cut\_red)(take\_diam)\phi_g.$$

This reads: "Executing the plan $(cut\_red, take\_diam)$ in the init. state s_0 leads to the goal ϕ_g being satisfied." But not **implicitly coordinated**...

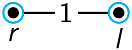
Local states and perspective shifts

Consider the state s after the red wire has been cut:

$s =$ 

s is the **global state** of the system after the wire has been cut (a state with a single designated world).

But s is not the **local state** of agent 1 in this situation. The **associated local state** of agent 1, s^1, is achieved by closing under the indistinguishability relation of 1:

$s^1 =$ 

We have $s \models \neg l$ and $s^0 \models \neg l$ but $s^1 \not\models \neg l$. Hence agent 1 does not know that it is safe to take the diamond.

Agent 0 can in $s^0 = s$ make a **change of perspective** to agent 1, that is, compute s^1, and conclude that agent 1 will not take the diamond.

Example continued

- Agent 0 knows the plan $(cut\_red, take\_diam)$ works: $s_0 \models K_0(cut\_red)(take\_diam)\phi_g$.
- Agent 1 does not know the plan works, and agent 0 knows this: $s_0 \models \neg K_1(cut\_red)(take\_diam)\phi_g \wedge K_0(\neg K_1(cut\_red)(take\_diam)\phi_g)$.
- Even after the wire has been cut, agent 1 does not know she can achieve the goal by *take_diam*: $s_0 \models (cut\_red)\neg K_1(take\_diam)\phi_g$.

Consider adding an announcement action $tell\_\neg l$ with $\omega(tell\_\neg l) = 0$. Then:

- Agent 0 knows the plan $(cut\_red, tell\_\neg l, take\_diam)$ works: $s_0 \models K_0(cut\_red)(tell\_\neg l)(cut\_diam)\phi_g$.
- Agent 1 still does not know the plan works: $s_0 \models \neg K_1(cut\_red)(tell\_\neg l)(take\_diam)\phi_g$.
- But agent 1 will know **in due time**, and agent 0 knows this: $s_0 \models K_0(cut\_red)(tell\_\neg l)K_1(take\_diam)\phi_g$.

Implicitly coordinated sequential plans

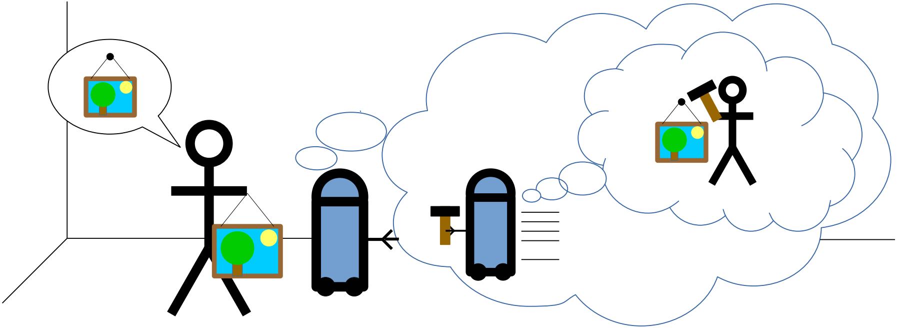
Definition. Given a planning taks $\Pi = (s_0, A, \omega, \phi_g)$, an **implicitly coordinated plan** is a sequence $\pi = (a_1, \dots, a_n)$ of action from A such that

$$s_0 \models K_{\omega(a_1)}(a_1)K_{\omega(a_2)}(a_2)\cdots K_{\omega(a_n)}(a_n)\phi_g.$$

In words: *The owner of the first action a_1 knows that a_1 is initially applicable and will lead to a situation where the owner of the second action a_2 knows that a_2 is applicable and will lead to a situation where... the owner of the nth action an knows that a_n is applicable and will lead to the goal being satisfied.*

Example. For the diamond stealing task, $(cut\_red, take\_diam)$ is not an implicitly coordinated plan, but $(cut\_red, tell\_\neg l, take\_diam)$ is.

Household robot example



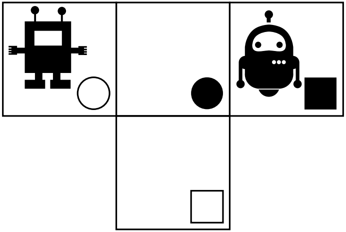
$$s_0 \models K_r(get\_hammer)K_h(hang\_up\_picture)\phi_g$$

$$s_0 \models K_r(tell\_hammer\_location)K_h(get\_hammer)K_h(hang\_up\_picture)\phi_g$$

If the robot is **eager** to help, it will prefer implicitly coordinated plans in which it itself acts whenever possible. If it is **altruistic** it will try to minimise the actions of the human.

From sequential plans to policies

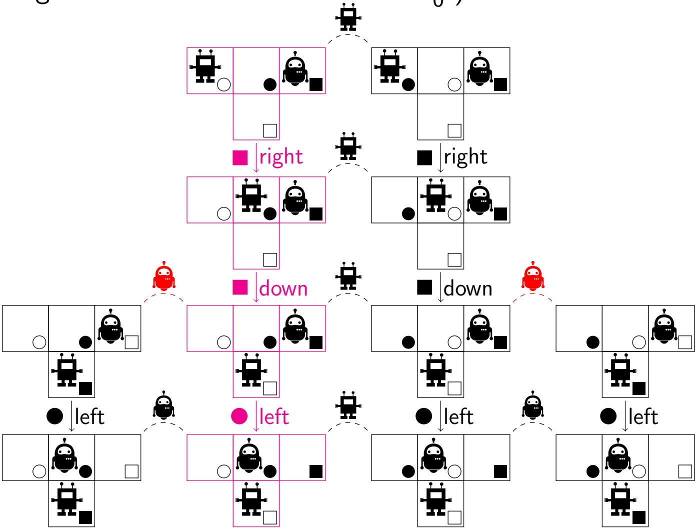
Sequential plans are not in general sufficient.



We need to define policies: mappings from states to actions...

Implicitly coordinated policies by example

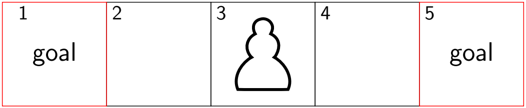
Below: Initial segment of the execution tree of an implicitly coordinated policy for the square robot (that is, an implicitly coordinated policy for the planning task where the initial state is $s_0^{🤖}$).



Policy profiles

When agents are implicitly coordinating, each agent independently forms an implicitly coordinated policy to reach the goal. A **policy profile** is a family of profiles, one for each agent.

Example. Two agents, L and R. L can only move the chess piece left, R only right. The chess piece has to be moved to a goal square. The goal squares are square 1 and 5, and this is common knowledge.

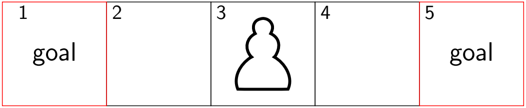


Example policy profile consisting of implicitly coordinated plans:

- Policy/plan of agent L: $(moveL, moveL)$.
- Policy/plan of agent R: $(moveL, moveL)$.

Note that $\omega(MoveL) = L$.

Agent types



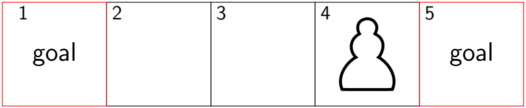
Lazy agents. An agent i is **lazy** if actions in $\{a \mid \omega(a) \neq i\}$ always take precedence in its choice of policy. A policy profile for the chess problem made by lazy agents leads to a **deadlock** (unsuccessful execution).

Eager agents. An agent i is **eager** if actions in $\{a \mid \omega(a) = i\}$ always take precedence in its choice of policy. A policy profile for the chess problem made by eager agents can result in a “livelock” (infinite unsuccessful execution).

Altruistic agents. An agent i is **altruistic** if it always chooses policies that minimise the worst-case number of actions in $\{a \mid \omega(a) \neq i\}$. A policy profile made by altruistic agents can also result in a “livelock”.

Compare with the household robot problem.

Intelligently eager agents

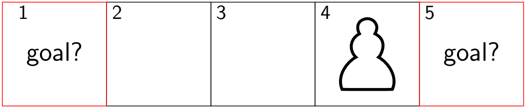


Intelligently eager agents. An agent i is **intelligently eager** if it always chooses a policy of minimal (perspective-sensitive) worst-cases execution length, and among those policies, the actions in $\{a \mid \omega(a) = i\}$ take precedence.

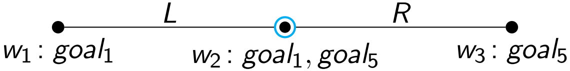
Success!: Any execution of a policy profile for the chess problem made by intelligently eager agents is successful.

So will intelligently eager agents always be successful in implicit coordination?...

Chess problem under partial observability



Consider the chess problem from before, but where initially L only knows that square 1 is a goal, and agent R only knows that square 5 is a goal:



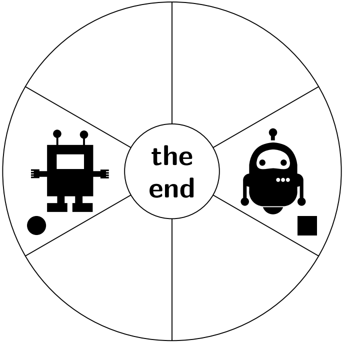
In this case, even policies made by intelligently eager agents can result in infinite unsuccessful executions.

Our only positive result so far then becomes:

Theorem. Let Π be a planning task with uniform observability (all agents share the same indistinguishability relation). Then any execution of a policy profile made by intelligently eager agents will be successful.

Future work

- Meta-reasoning: If R moves the chess piece to the right and L knows that agent R is intelligently eager, L can infer that there is a goal to the right.
- Ensuring successful executions through announcements: If R plans to announce $goal_5$ before going right (and vice versa for agent L), any execution will be successful.



References I

Baltag, A., Moss, L. S. and Solecki, S. (1998).
The Logic of Public Announcements and Common Knowledge and Private Suspicions.
In Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98), (Gilboa, I., ed.), pp. 43–56, Morgan Kaufmann.

Engesser, T., Bolander, T., Mattmüller, R. and Nebel, B. (2015).
Cooperative Epistemic Multi-Agent Planning With Implicit Coordination.
In Distributed and Multi-Agent Planning (DMAP-15) pp. 68–76,.

van Benthem, J., Gerbrandy, J. and Pacuit, E. (2007).
Merging frameworks for interaction: DEL and ETL.
In Proceedings of the 11th conference on Theoretical aspects of rationality and knowledge TARK '07 pp. 72–81, ACM, New York, NY, USA.

van Ditmarsch, H. and Kooi, B. (2008).
Semantic Results for Ontic and Epistemic Change.
In Logic and the Foundation of Game and Decision Theory (LOFT 7), (Bonanno, G., van der Hoek, W. and Wooldridge, M., eds), Texts in Logic and Games 3 pp. 87–117, Amsterdam University Press.