Dynamic Epistemic Logic

5. Epistemic Planning

Albert-Ludwigs-Universität Freiburg

Bernhard Nebel and Robert Mattmüller

June 19th, 2019

Prelude: Two Extensions

Epistemic Planning

Remark: Epistemic planning can also be based on formalisms other than DEL. We only focus on DEL here, though.

Before we begin: We first want to introduce to extensions to our DEL models:

- Multipointed models
- Action models with ontic effects

Multipointed Models

So far: State and action models only had a unique designated world/event.

- The actual world
- The event that actually takes place

Now: We also allow state and action models with more than one designated world/event.

- The set of worlds that may be the actual world (from some agent's perspective)
- The set of events that may actually take place (nondeterministically)

Multipointed Models

Definition (closure under indistinguishability)

Let $\mathcal{M} = (S, \sim, V)$ be an epistemic model, $S' \subseteq S$, and $a \in A$. Then S' is closed under indistinguishability of agent a if $s \in S'$ and $s \sim_a s'$ implies $s' \in S'$ for all $s, s' \in S$.

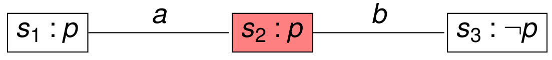
Definition (multipointed epistemic model)

Let $\mathcal{M} = (S, \sim, V)$ be an epistemic model, and $\emptyset \neq S_\mathrm{d} \subseteq S$. Then $(\mathcal{M}, S_\mathrm{d})$ is a multipointed model. If $S_\mathrm{d} = \{s\}$, then $(\mathcal{M}, S_\mathrm{d})$ is a global state. If S_d is closed under indistinguishability for some agent $a \in A$, then $(\mathcal{M}, S_\mathrm{d})$ is local for agent a. Given a global state $(\mathcal{M}, \{s\})$, the associated local state for agent a is the model $(\mathcal{M}, \{s\})^a = (\mathcal{M}, \{s' \in S \mid s' \sim_a s\})$. Similarly, $(\mathcal{M}, S_\mathrm{d})^a = (\mathcal{M}, S'_\mathrm{d})$, for $S'_\mathrm{d} = \{s' \in S \mid s' \sim_a s \text{ for some } s \in S_\mathrm{d}\}$.

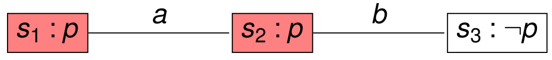
Multipointed Models

Example

Global state $(\mathcal{M}, \{s_2\})$:



Associated local state $(\mathcal{M}, \{s_1, s_2\})$ for agent a:



Multipointed Models

Note: Definition of bisimulation has to be adapted to relate designated worlds in one model to designated worlds in the other model. (Homework.)

Definition (truth condition in multipointed models)

Given a formula φ (from any of the logics defined earlier) and a multipointed model $(\mathcal{M}, S_\mathrm{d})$, we define:

$$\mathcal{M}, S_\mathrm{d} \models \varphi \quad \text{iff} \quad \mathcal{M}, s \models \varphi \text{ for all } s \in S_\mathrm{d}.$$

Note: If $(\mathcal{M}, S_\mathrm{d})$ is local for some agent a, then $\mathcal{M}, S_\mathrm{d} \models K_a\varphi$ iff $\mathcal{M}, S_\mathrm{d} \models \varphi$.

Multipointed Models

Definition (multipointed action model)

Let $M = (E, \sim, pre)$ be an action model and $\emptyset \neq E_\mathrm{d} \subseteq E$. Then we call (M, E_d) a multipointed action model.

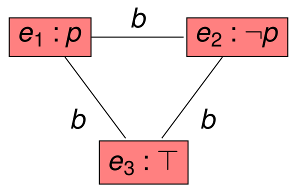
Note: Definitions of closure under indistinguishability, local/global/associated local (action) models similar to those for multipointed epistemic models. Adaptation of definition of bisimulations/emulations also similar.

Remark: Multipointed action models show up if

- an action is actually nondeterministic, or
- an action appears nondeterministic from some agent's perspective.

Example (Nondeterministic action)

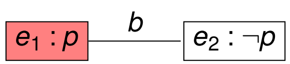
Action model (Mayread, $\{e_1, e_2, e_3\}$):



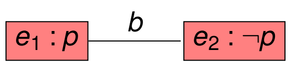
Alice may or may not read the letter, nondeterministically.

Example (Seemingly nondeterministic action)

Action model (Read, e_1):



Associated action model (Read, $\{e_1, e_2\}$) for agent b:



Although the Read action is deterministic (in every state, only one of the events can possibly take place), it appears nondeterministic to agent b, since he does not know which event occurs.

Further remarks:

- Action (M, E_{d}) behaves like nondeterministic action $(M, e_1) \cup \cdots \cup (M, e_n)$ for $E_{\mathrm{d}} = \{e_1, \ldots, e_n\}$.
- Better examples of nondeterministic actions, like coin tossing, possible once we have ontic effects (see below).

Ontic Effects

So far: Actions only affect knowledge (via announcements, other forms of communication, sensing, . . .).

Now: We also want actions to change ontic facts (opening a door, tossing a coin, toggling a switch, moving from A to B, . . .).

Ontic Effects

Definition (action model with ontic effects)

An action model with ontic effects $M = (E, \sim, pre, eff)$ is an action model $(E, \sim, pre)$ together with a function $eff : E \to \mathcal{L}_K$, where for all $e \in E$, $eff(e)$ is a conjunction of atoms and negated atoms from P.

Example

$eff(e) = p \wedge q \wedge \neg r \wedge \neg x$ means that event e makes p and q true and r and x false.

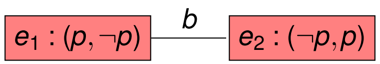
Note: This corresponds to add and delete lists in STRIPS planning.

Ontic Effects

Graphical notation: Label $e : (\varphi, \psi)$ means that the event is named e, and that $pre(e) = \varphi$ and $eff(e) = \psi$.

Example (Toggling a switch)

The truth value of p is complemented. Agent a sees p, agent b does not.



Example (Tossing a coin)

A coin is tossed (p means heads, $\neg p$ means tails). The coin toss happens in public.

Ontic Effects

In order to correctly reflect ontic effects in our semantics, we need to make the product update take them into account.

Definition (Product update)

Let $\mathcal{M} = (S, \sim, V)$ be an epistemic state with designated worlds $S_d \subseteq S$, and let $M = (E, \sim, pre, eff)$ be an action model with designated events $E_d \subseteq E$. Then the product update $(\mathcal{M}, S_d) \otimes (M, E_d)$ is the epistemic state $\mathcal{M}' = (S', \sim', V')$ with with designated worlds $S'_d \subseteq S'$, where:

- $S' = \{(s, e) \in S \times E \mid \mathcal{M}, s \models pre(e)\}$,
- $(s, e) \sim'_a (t, \varepsilon)$ iff $s \sim_a t$ and $e \sim_a \varepsilon$, for $a \in A$,
- $(s, e) \in V'_p$ iff $(s \in V_p$ and $eff(e) \not\models \neg p)$ or $eff(e) \models p$, for all $p \in P$, and
- $(s, e) \in S'_d$ iff $s \in S_d$ and $e \in E_d$.

Ontic Effects

Definition (applicability)

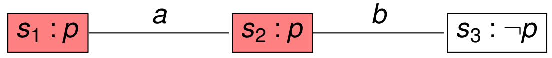
Action (M, E_d) is applicable in local state $(\mathcal{M}, S_d)$ iff, for all $s \in S_d$, there is at least one $e \in E$ with $\mathcal{M}, s \models pre(e)$.

Everything else stays more or less the same (well, except for action bisimulations and emulations, . . .).

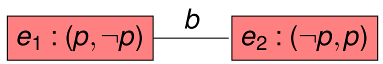
Ontic Effects

Example

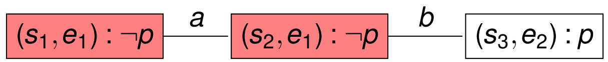
Initially, a knows p and considers it possible that b does not know p.



We then apply the toggling action.



Resulting epistemic state: like initially, but with p toggled.



Ontic Effects

Recall the funniest joke in the world:

> *Two hunters are out in the woods when one of them collapses. He doesn't seem to be breathing and his eyes are glazed. The other guy whips out his phone and calls the emergency services. He gasps, "My friend is dead! What can I do?" The operator says, "Calm down. I can help. First, let's make sure he's really dead." There is a silence; then a gun shot is heard. Back on the phone, the guy says, "OK, now what?"*

Homework:

- DEL action model for the "epistemic reading" of making sure he's really dead?
- DEL action model for the "ontic reading" of making sure he's really dead?

Classical Planning

Classical Planning

Roadmap:

- Review of classical and nondeterministic planning without epistemic aspects
  (this section)
- Extension towards epistemic planning
  (next section)

Note: Rest of this section based on slides of AI Planning course, WS 2018/2019, chapters 1, 2, and 14 (`http://gki.informatik.uni-freiburg.de/teaching/ws1819/aip/aip01.pdf`, `http://gki.informatik.uni-freiburg.de/teaching/ws1819/aip/aip02.pdf`, `http://gki.informatik.uni-freiburg.de/teaching/ws1819/aip/aip14.pdf`). Extended slides can be found there.

What is planning?

Planning

"Planning is the art and practice of thinking before acting."

— Patrik Haslum

- intelligent decision making: What actions to take?
- general-purpose problem representation
- algorithms for solving any problem expressible in the representation

Transition systems

Definition (transition system)

A transition system is a 5-tuple $\mathcal{T} = (S, L, T, s_0, S_\star)$ where

- S is a finite set of states,
- L is a finite set of (transition) labels,
- $T \subseteq S \times L \times S$ is the transition relation,
- $s_0 \in S$ is the initial state, and
- $S_\star \subseteq S$ is the set of goal states.

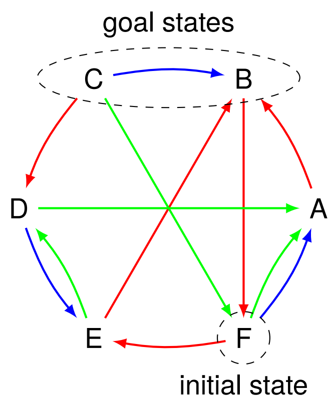
We say that $\mathcal{T}$ has the transition (s, ℓ, s') if $(s, \ell, s') \in T$.

We also write this $s \xrightarrow{\ell} s'$, or $s \rightarrow s'$ when not interested in ℓ.

$\mathcal{T}$ is called deterministic if for all states s and labels ℓ, there is at most one state s' with $s \xrightarrow{\ell} s'$.

Transition systems: example

Transition systems are often depicted as directed arc-labeled graphs with marks to indicate the initial state and goal states.



Transition system terminology

We use common graph theory terms for transition systems:

- s' successor of s if $s \to s'$
- s predecessor of s' if $s \to s'$
- s' reachable from s if there exists a sequence of transitions from s to s'.

Compact representations

- Classical (i. e., deterministic) planning is in essence the problem of finding solutions in huge transition systems.
- The transition systems we are usually interested in are too large to explicitly enumerate all states or transitions.
- Hence, the input to a planning algorithm must be given in a more concise form.

State variables

How to represent huge state sets without enumerating them?

- represent different aspects of the world in terms of different state variables

⇝ a state is a valuation of state variables

- n state variables with m possible values each induce m^n different states

⇝ exponentially more compact than “flat” representations

Boolean state variables

Problem:

- How to succinctly represent transitions and goal states?

Idea: Use propositional logic

- state variables: propositional variables (0 or 1)
- goal states: defined by a propositional formula
- transitions: defined by actions given by
  - precondition: when is the action applicable?
  - effect: how does it change the valuation?

Note: general finite-domain state variables can be compactly encoded as Boolean variables

Operators

Transitions for state sets described by propositions P can be concisely represented as operators or actions $o = \langle pre, eff \rangle$ where

- the precondition pre is a propositional formula over P describing the set of states in which the transition can be taken (states in which a transition starts), and
- the effect eff describes how the resulting successor states are obtained from the state where the transitions is taken (where the transition goes).

Effects (for deterministic operators)

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Definition (effects)

(Deterministic) effects are recursively defined as follows:

- If $p \in P$ is a state variable, then p and $\neg p$ are effects (atomic effect).
- If $eff_1, \ldots, eff_n$ are effects, then $eff_1 \wedge \cdots \wedge eff_n$ is an effect (conjunctive effect).
  The special case with $n = 0$ is the empty effect $\top$.
- If pre is a propositional formula and eff is an effect, then $pre \rhd eff$ is an effect (conditional effect).

Atomic effects p and $\neg p$ are best understood as assignments $p := 1$ and $p := 0$, respectively.

Operator semantics

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Definition (changes caused by an operator)

For each effect eff and state s, we define the change set of eff in s, written $[eff]_s$, as the following set of literals:

- $[p]_s = \{p\}$ and $[\neg p]_s = \{\neg p\}$ for atomic effects p, $\neg p$
- $[eff_1 \wedge \cdots \wedge eff_n]_s = [eff_1]_s \cup \cdots \cup [eff_n]_s$
- $[pre \rhd eff]_s = [eff]_s$ if $s \models pre$ and $[pre \rhd eff]_s = \emptyset$ otherwise

Definition (applicable operators)

Operator $\langle pre, eff \rangle$ is applicable in a state s iff $s \models pre$ and $[eff]_s$ is consistent (i. e., does not contain two complementary literals).

Operator semantics (ctd.)

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Definition (successor state)

The successor state $app_o(s)$ of s with respect to operator $o = \langle pre, eff \rangle$ is the state s' with $s' \models [eff]_s$ and $s'(p) = s(p)$ for all state variables p not mentioned in $[eff]_s$.
This is defined only if o is applicable in s.

Deterministic planning tasks

Definition (deterministic planning task)

A deterministic planning task is a 4-tuple $\Pi = (P, I, \mathrm{Act}, \gamma)$ where

- P is a finite set of state variables (propositions),
- I is a valuation over P called the initial state,
- Act is a finite set of operators over P, and
- γ is a formula over P called the goal.

Mapping planning tasks to transition systems

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Definition (induced transition system of a planning task)

Every planning task $\Pi = (P, I, \mathrm{Act}, \gamma)$ induces a corresponding deterministic transition system $\mathcal{T}(\Pi) = (S, L, T, s_0, S_\star)$:

- S is the set of all valuations of P,
- L is the set of operators Act,
- $T = \{(s, o, s') \mid s \in S,\ o \text{ applicable in } s,\ s' = app_o(s)\}$,
- $s_0 = I$, and
- $S_\star = \{s \in S \mid s \models \gamma\}$

Planning tasks: terminology

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

- Terminology for transitions systems is also applied to the planning tasks that induce them.
- A sequence of operators that forms a goal path of $\mathcal{T}(\Pi)$ is called a plan of Π.

Planning

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

By planning, we mean the following two algorithmic problems:

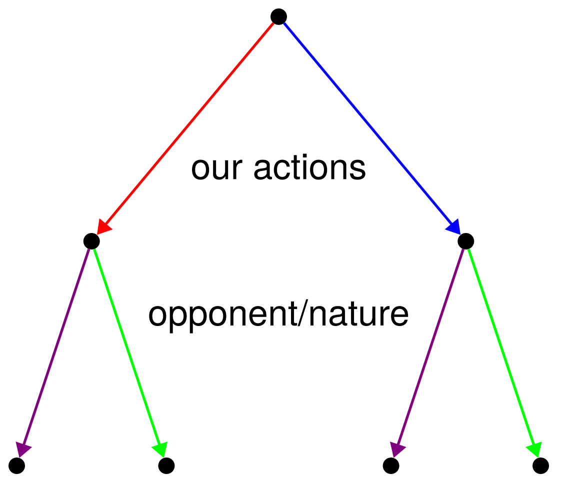
Definition (satisficing planning)

Given: a planning task Π
Output: a plan for Π, or **unsolvable** if no plan for Π exists

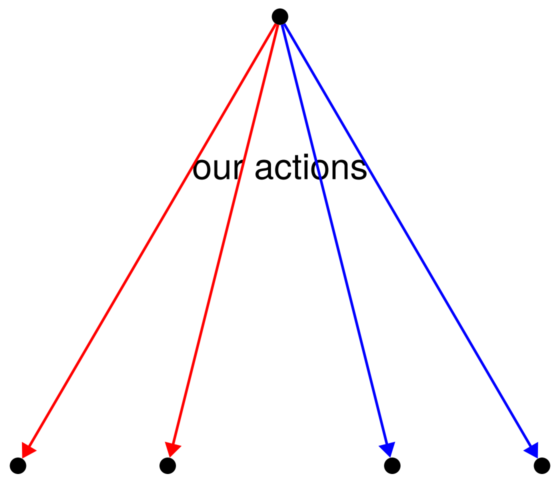
Definition (optimal planning)

Given: a planning task Π
Output: a plan for Π with minimal length among all plans for Π, or **unsolvable** if no plan for Π exists

Nondeterminism



Nondeterminism



Nondeterministic operators

Definition (nondeterministic operator)

A nondeterministic operator is a pair $o = (pre, Eff)$, where

- pre is a conjunction of atoms (the precondition), and
- $Eff = \{eff_1, \dots, eff_n\}$ is a finite set of possible effects of o, each eff_i being a conjunction of atomic finite-domain effects.

Nondeterministic operators

Definition (nondeterministic operator application)

Let $o = (pre, Eff)$ be a nondeterministic operator and s a state.

Applicability of o in s is definied as in the deterministic case, i.e., o is applicable in s iff $s \models pre$ and the change set of each effect $eff \in Eff$ is consistent.

If o is applicable in s, then the application of o in s leads to one of the states in the set $app_o(s) := \{app_{(pre,eff)}(s) \mid eff \in Eff\}$ nondeterministically.

Nondeterministic planning tasks and transition systems

Nondeterministic planning task: Like a deterministic planning task, but now possibly with nondeterministic actions.

Induced transition system: Like before, but now possibly with nondeterministic transitions.

What is a plan?

In nondeterministic planning, plans are more complicated objects than in the deterministic case:

The best action to take may depend on nondeterministic effects of previous operators.

Nondeterministic plans thus often require branching. Sometimes, they even require looping.

Here: Only consider branching, no looping.

Nondeterministic plans: formal definition

Definition (strategy)

Let $\Pi = (P, I, \mathrm{Act}, \gamma)$ be a nondeterministic planning task with state set S and goal states $S_\star$.

A strategy for Π is a function $\pi : S_\pi \to \mathrm{Act}$ for some subset $S_\pi \subseteq S$ such that for all states $s \in S_\pi$ the action $\pi(s)$ is applicable in s.

The set of states reachable in $\mathcal{T}(\Pi)$ starting in state s and following π is denoted by $S_\pi(s)$.

Nondeterministic plans: formal definition

Definition (weak, closed, proper, and acyclic strategies)

Let $\Pi = (P, I, \mathrm{Act}, \gamma)$ be a nondeterministic planning task with state set S and goal states $S_\star$, and let π be a strategy for Π. Then π is called

- proper iff $S_\pi(s') \cap S_\star \neq \emptyset$ for all $s' \in S_\pi(s_0)$, and
- acyclic iff there is no state $s' \in S_\pi(s_0)$ such that s' is reachable from s' following π in a strictly positive number of steps.

Definition

Let $\Pi = (P, I, \mathrm{Act}, \gamma)$ be a nondeterministic planning task with state set S and goal states $S_\star$.
A strategy for Π is called a strong plan if it is proper and acyclic.

Strong Planning

Definition (strong planning)

Given: a nondeterministic planning task Π
Output: a strong plan for Π, or **unsolvable** if no strong plan for Π exists

From Classical to Epistemic Planning

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Summary: Classical planning on one slide:

- Given:
  - Initial world state
  - Goal description
  - Available actions
- Wanted:
  - Plan leading from initial state to goal state
- Assumptions:
  - Single agent
  - Full observability
  - Deterministic actions
  - Static and discrete environment
  - Reachability goal
  - ...

From Classical to Epistemic Planning

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Algorithmic techniques successful in (satisficing) classical planning:

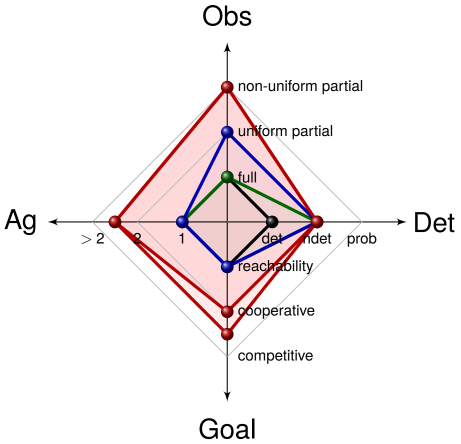
Mainly state-space search

- guided by goal-distance heuristics
- based on delete relaxation,
- abstractions, and
- landmarks,
- enhanced with pruning techniques (helpful actions, commutativity, symmetry),
- as well as invariants, causal relationships, decoupling techniques, ...

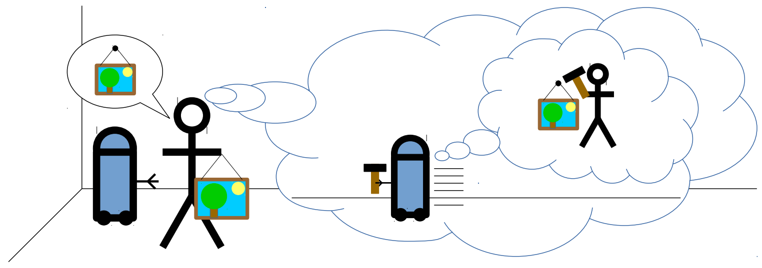
Prelude: Two Extensions
Classical Planning
Epistemic Planning
Epistemic Planning
Summary

From Classical to Epistemic Planning

Classical, FOND, POND, epistemic planning, . . .



Example: Robot Collaborating with Human



- Epistemic planning useful if we want the agents to coordinate implicitly

Cooperative Epistemic Planning

Cooperative epistemic planning:

- Task: Collaboratively reach joint goal
- Challenge: Required knowledge and capabilities distributed among agents
- Idea: Communication / coordination as part of the plan

Epistemic Planning

Roadmap of the remainder of this lecture:

- Definition of epistemic planning tasks and objectives
- Centralized vs. implicitly coordinated plans
- Linear vs. branching plans
- Hardness of epistemic planning (?)
- Algorithms for epistemic planning, tractable fragments, heuristics (?)
- Execution of (profiles of) epistemic plans (?)

Epistemic Planning

Good and relatively recent overview of the state of the art in epistemic planning:
Baral et al., Epistemic Planning (Dagstuhl Seminar 17231), Dagstuhl Reports, Vol. 7, Issue 6 (2017),
`http://drops.dagstuhl.de/opus/volltexte/2017/8285/`

Intro to DEL-based epistemic planning:
Bolander, A Gentle Introduction to Epistemic Planning: The DEL Approach (2017),
`https://arxiv.org/abs/1703.02192`

Cooperative Epistemic Planning Tasks

From now on: Multi-pointed models, ontic effects.
Fix a finite set of agents A.

Definition

A cooperative epistemic planning task $\Pi = (P, I, \text{Act}, \gamma, \omega)$ consists of

- a finite set of state variables (atomic propositions) P,
- an initial global epistemic state $I = (\mathcal{M}_0, s_0)$ over P,
- a finite set Act of epistemic actions over P,
- a goal formula $\gamma \in \mathcal{L}_{KC}$ over P, and
- an owner function $\omega : \text{Act} \to A$, such that each action $\alpha \in \text{Act}$ is local for $\omega(\alpha)$.

Assumption: Act is common knowledge among all agents.

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Centralized Sequential Plans

An epistemic model $(\mathcal{M}, S_\text{d})$ is a goal state iff $(\mathcal{M}, S_\text{d}) \models \gamma$ iff $(\mathcal{M}, s) \models \gamma$ for all $s \in S_\text{d}$.

Terminology: In the following, we abbreviate "cooperative epistemic planning task" as "planning task".

Definition (Centralized sequential epistemic plan)

A centralized sequential (or linear) epistemic plan for a planning task $\Pi = (P, I, \text{Act}, \gamma, \omega)$ is a sequence of actions from Act, $\pi = \alpha_1, \dots, \alpha_n$ such that

- for each $i = 1, \dots, n$, action α_i is applicable in $I \otimes \alpha_1 \otimes \dots \otimes \alpha_{i-1}$, and
- $I \otimes \alpha_1 \otimes \dots \otimes \alpha_n \models \gamma$.

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Centralized Sequential Plans

In order to simplify some future proofs and to highlight the simplicity to the definition of implicitly coordinated sequential plans (see below), we give an equivalent definition of centralized sequential epistemic plans:

Proposition

Let $\Pi = (P, I, \text{Act}, \gamma, \omega)$ be a planning task and $\pi = \alpha_1, \dots, \alpha_n$ be a sequence of actions from Act. Then π is a centralized sequential epistemic plan for Π iff

- *$n = 0$ and $I \models \gamma$, or*
- *$n > 0$ and α_1 is applicable in I and $\alpha_2, \dots, \alpha_n$ is a centralized sequential epistemic plan for $\Pi' = (P, I \otimes \alpha_1, \text{Act}, \gamma, \omega)$.* □

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Centralized Sequential Plans

For convenience, we add a new modality as an abbreviation to $\mathcal{L}_{KC[]}$:

Definition

Modality $(\!(\alpha)\!)$ is defined such that, for all formulas $\varphi \in \mathcal{L}_{KC[]}$, we have

$$(\!(\alpha)\!)\varphi \quad \equiv \quad \langle\alpha\rangle\top \wedge [\alpha]\varphi$$

Truth condition:

$\mathcal{M}, s \models (\!(\alpha)\!)\varphi$ iff α is applicable in $\mathcal{M}, s$ and $(\mathcal{M}, s) \otimes \alpha \models \varphi$

Centralized Sequential Plans

Proposition

Let $\Pi = (P, I, \mathrm{Act}, \gamma, \omega)$ be a planning task and $\pi = \alpha_1, \dots, \alpha_n$ a sequence of actions from Act. Then π is a centralized sequential epistemic plan for Π if and only if $I \models (\!(\alpha_1)\!)(\!(\alpha_2)\!) \dots (\!(\alpha_n)\!)\gamma$.

Proof.

Induction on plan length n.

- Base case ($n = 0$): Then π is a centralized sequential epistemic plan for Π iff $I \models \gamma$.
- Inductive case ($n > 0$): [...]

Centralized Sequential Plans

Proof (ctd.)

- Inductive case ($n > 0$): Let $\pi = \alpha_1, \dots, \alpha_n$.

  Then π is a centralized sequential epistemic plan for Π iff (previous proposition)

  α_1 is applicable in I and $\alpha_2, \dots, \alpha_n$ is a centralized sequential epistemic plan for $\Pi' = (P, I \otimes \alpha_1, \mathrm{Act}, \gamma, \omega)$ iff (induction hypothesis!)

  α_1 is applicable in I and $I \otimes \alpha_1 \models (\!(\alpha_2)\!) \dots (\!(\alpha_n)\!)\gamma$ iff (truth condition of $(\!(\cdot)\!)$)

  $I \models (\!(\alpha_1)\!)(\!(\alpha_2)\!) \dots (\!(\alpha_n)\!)\gamma$. □

Plantime vs. Runtime Indistinguishability

Definition (plantime vs. runtime indistinguishability)

Let $M = (E, \sim, pre, eff)$, $E_\mathrm{d} \subseteq E$, and assume that (M, E_d) is local to some agent $a \in A$. Let $e_1, e_2 \in E_\mathrm{d}$. Then e_1 and e_2 are called runtime indistinguishable for agent a if $e_1 \sim_a e_2$. Otherwise (if $e_1 \not\sim_a e_2$), they are runtime distinguishable for a, but plantime indistinguishable for a.

Above, we defined plantime and runtime indistinguishability of events. Plantime and runtime indistinguishability of worlds in epistemic states can be defined similarly.

Plantime vs. Runtime Indistinguishability

Example (for $A = \{a\}$)

Model (Before, $\{s_1, s_2\}$):

$s_1 : p$ —a— $s_2 : \neg p$

Worlds s_1 and s_2 both plantime and runtime indistinguishable to agent a.

Action model ($\text{Read}_a, \{e_1, e_2\}$):

$e_1 : (p, \top)$ $e_2 : (\neg p, \top)$

Events e_1 and e_2 plantime indistinguishable, but runtime distinguishable to agent a.

Plantime vs. Runtime Indistinguishability

Example (ctd.)

Model $(\text{After}, S_d) = (\text{Before}, \{s_1, s_2\}) \otimes (\text{Read}_a, \{e_1, e_2\})$:

$(s_1, e_1) : p$ $(s_2, e_2) : \neg p$

Worlds (s_1, e_1) and (s_2, e_2) plantime indistinguishable, but runtime distinguishable to agent a.

Indistinguishability of Action Models

We assume that, for a given planning task Π, the set of actions Act is common knowledge among the agents. This does not imply that agents can always correctly identify the actions executed by other agents.

This can lead to problematic situations: Assume that the only way to achieve the goal is (1) agent a performs action α_p, followed by (2) agent b performing action α_γ. Now, if agent b cannot distinguish between agent a performing action α_p, and agent a performing a different, useless action α_q, then even after agent a has performed the good action α_p, agent b does not know that his action α_γ is applicable and leads to the goal!

Indistinguishability of Action Models

Example

Let $\Pi = (P, I, \text{Act}, \gamma, \omega)$ be the following planning task:

- Atomic propositions: $P = \{p, q, r\}$
- Initial state: $I =$ $s_0 : \neg p, \neg q, \neg r$
- Goal: $\gamma = r$
- Actions and owners:
  - $\alpha_p =$ $e_p : (\top, p)$ —b— $e_q : (\top, q)$ $\omega(\alpha_p) = a$
  - $\alpha_q =$ $e_p : (\top, p)$ —b— $e_q : (\top, q)$ $\omega(\alpha_q) = a$
  - $\alpha_\gamma =$ $e_\gamma : (p, r)$ $\omega(\alpha_\gamma) = b$

Act $= \{\alpha_p, \alpha_q, \alpha_\gamma\}$.
Agent b cannot distinguish between α_p and α_q at runtime.
Both are initially applicable.

Indistinguishability of Action Models

Example (ctd.)

$I \otimes \alpha_p =$

$(s_0, e_p) : p, \neg q, \neg r$ —b— $(s_0, e_q) : \neg p, q, \neg r$

Then $I \otimes \alpha_p \models p \wedge \neg K_b p$.

Note connection to earlier discussion of action and knowledge axiom: $I \models K_b[\alpha_p]p$, but $I \not\models [\alpha_p]K_b p$.

$I \otimes \alpha_p \otimes \alpha_\gamma =$

$(s_0, e_p, e_\gamma) : p, \neg q, r$

Then $I \otimes \alpha_p \otimes \alpha_\gamma \models \gamma$.
I. e., $(\alpha_p, \alpha_\gamma)$ is a centralized sequential epistemic plan.

Indistinguishability of Action Models

Example (ctd.)

This is the case although, in $I \otimes \alpha_p$, agent $b = \omega(\alpha_\gamma)$ does not event know that α_γ is applicable.

- Is this reasonable?
- Agents can always observe that an action occurs. But not necessarily which action!
- Problem here arises because agent b can mistake action α_p for α_q. This, however, can only happen if $\alpha_p, \alpha_q \in \mathrm{Act}$.

Indistinguishability of Action Models

If $\alpha_q \notin \mathrm{Act}$, agent b could use this knowledge to infer that the action applied by agent a must have been action α_p, and hence infer that p holds in $I \otimes \alpha_p$.

To avoid a "side-channel attack" of agent b, we define:

Definition

An action set Act is closed if, for all $(M, E_\mathrm{d}) \in \mathrm{Act}$ and $e \in E$ (of M), there is an $E'_\mathrm{d} \subseteq E$ with $e \in E'_\mathrm{d}$ and $(M, E'_\mathrm{d}) \in \mathrm{Act}$.

In the example, we had $(M, E_\mathrm{d}) = \alpha_p$, $e = e_q$, and $(M, E'_\mathrm{d}) = \alpha_q$.

From now on, we assume that all action sets Act are closed.

Indistinguishability of Action Models

Example (ctd.)

Back to the plan $\pi = (\alpha_p, \alpha_\gamma)$.

- If π is computed by a centralized instance and then distributed among the agents for execution:
  $\rightsquigarrow$ okay
- If agents need to coordinate themselves:
  $\rightsquigarrow$ not okay

Remedy:

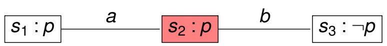
- Add an action announce$_p$ = $e_p : (p, \top)$ with $\omega(\text{announce}_p) = a$ to the planning task and
- change the definition of epistemic plans to enforce agent b to know that α_γ is applicable before applying it.

Implicitly Coordinated Sequential Plans

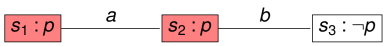
Recall (local perspective of an agent):
If $(\mathcal{M}, S_d)$ is an epistemic state and a is an agent, then $(\mathcal{M}, S_d)^a = (\mathcal{M}, S'_d)$ is agent a's associated local state, where $S'_d = \{s' \in S \mid s' \sim_a s \text{ for some } s \in S_d\}$).

Example

Global state $(\mathcal{M}, \{s_2\})$:



Associated local state $(\mathcal{M}, \{s_2\})^a$ for agent a:



Implicitly Coordinated Sequential Plans

In an implicitly coordinated plan, an agent knows that their supposed action is applicable and makes progress towards the goal.

Definition (Implicitly coordinated sequential plan)

Let $\Pi = (P, I, \mathrm{Act}, \gamma, \omega)$ be a planning task and $\pi = \alpha_1, \ldots, \alpha_n$ a sequence of actions from Act. Then π is an implicitly coordinated sequential epistemic plan (ICSEP) for Π iff either

- $n = 0$ and $I \models \gamma$, or
- $n > 0$ and α_1 is applicable in $I^{\omega(\alpha_1)}$ and $\alpha_2, \ldots, \alpha_n$ is a ICSEP for $\Pi' = (P, I^{\omega(\alpha_1)} \otimes \alpha_1, \mathrm{Act}, \gamma, \omega)$.

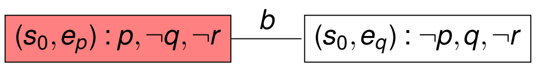
Implicitly Coordinated Sequential Plans

Recall the previous example:

1. $(\alpha_p, \alpha_\gamma)$ is not an ICSEP.
2. $(\alpha_p, \mathrm{announce}_p, \alpha_\gamma)$ is an ICSEP.

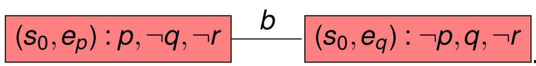
Ad (1): $I = \boxed{s_0 : \neg p, \neg q, \neg r} = I^{\omega(\alpha_p)}$.

$I^{\omega(\alpha_p)} \otimes \alpha_p =$



$=: (\mathcal{M}^1, S^1_d)$.

Now, is α_γ an ICSEP starting in $(\mathcal{M}^1, S^1_d)$? No, since α_γ with precondition p is not applicable in $(\mathcal{M}^1, S^1_d)^{\omega(\alpha_\gamma)} = (\mathcal{M}^1, S^1_d)^b =$



.

Implicitly Coordinated Sequential Plans

Ad (2): Is $\mathrm{announce}_p$ applicable in $(\mathcal{M}^1, S^1_d)^{\omega(\mathrm{announce}_p)} = (\mathcal{M}^1, S^1_d)^a$? YES!

$(\mathcal{M}^1, S^1_d)^a \otimes \mathrm{announce}_p = \boxed{(s_0, e_p, e_p) : p, \neg q, \neg r} =: (\mathcal{M}^2, S^2_d)$.

Still to show: α_γ is an ICSEP starting in $(\mathcal{M}^2, S^2_d)$.
This is clear, since $(\mathcal{M}^2, S^2_d)^{\omega(\alpha_\gamma)} = (\mathcal{M}^2, S^2_d)^b = (\mathcal{M}^2, S^2_d) \models p$.

Then $(\mathcal{M}^2, S^2_d)^b \otimes \alpha_\gamma = \boxed{(s_0, e_p, e_p, e_\gamma) : p, \neg q, r} =: (\mathcal{M}^3, S^3_d)$.

Now, the empty plan is an ICSEP for $(\mathcal{M}^3, S^3_d)$, since $(\mathcal{M}^3, S^3_d) \models \gamma$.

Implicitly Coordinated Sequential Plans

A simple lemma we will need in a moment.

Proposition (knowledge and associated local states)

$(\mathcal{M}, S_d)^a \models \varphi$ *iff* $\mathcal{M}, S_d \models K_a\varphi$.

Proof.

$$
\begin{aligned}
(\mathcal{M}, S_d)^a \models \varphi \quad &\text{iff} \quad (\mathcal{M}, \{s' \mid s' \sim_a s \text{ for some } s \in S_d\}) \models \varphi \\
&\text{iff} \quad \mathcal{M}, s' \models \varphi \text{ f.a. } s' \text{ s.t. ex. } s \in S_d \text{ s.t. } s' \sim_a s \\
&\text{iff} \quad \mathcal{M}, s \models K_a\varphi \text{ for all } s \in S_d \\
&\text{iff} \quad \mathcal{M}, S_d \models K_a\varphi \qquad \square
\end{aligned}
$$

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Implicitly Coordinated Sequential Plans

Proposition

Let $\Pi = (P, I, \text{Act}, \gamma, \omega)$ be a planning task and $\pi = \alpha_1, \dots, \alpha_n$ a sequence of actions from Act. Then π is an implicitly coordinated sequential epistemic plan for Π if and only if $I \models K_{\omega(\alpha_1)}(\!(\alpha_1)\!)K_{\omega(\alpha_2)}(\!(\alpha_2)\!)\dots K_{\omega(\alpha_n)}(\!(\alpha_n)\!)\gamma$.

Proof.

Induction on plan length n.

- Base case ($n = 0$): Then π is an implicitly coordinated sequential epistemic plan iff $I \models \gamma$.
- Inductive case ($n > 0$): [...]

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Implicitly Coordinated Sequential Plans

Proof (ctd.)

- Inductive case ($n > 0$): Let $\pi = \alpha_1, \dots, \alpha_n$.

  Then π is an implicitly coordinated epistemic plan for Π iff (definition)

  α_1 is applicable in $I^{\omega(\alpha_1)}$ and $\alpha_2, \dots, \alpha_n$ is an implicitly coordinated epistemic plan for $\Pi' = (P, I^{\omega(\alpha_1)} \otimes \alpha_1, \text{Act}, \gamma, \omega)$ iff (induction hypothesis!)

  α_1 is applicable in $I^{\omega(\alpha_1)}$ and $I^{\omega(\alpha_1)} \otimes \alpha_1 \models K_{\omega(\alpha_2)}(\!(\alpha_2)\!)\dots K_{\omega(\alpha_n)}(\!(\alpha_n)\!)\gamma$ iff (truth condition of $(\!(\cdot)\!)$)

  $I^{\omega(\alpha_1)} \models (\!(\alpha_1)\!)K_{\omega(\alpha_2)}(\!(\alpha_2)\!)\dots K_{\omega(\alpha_n)}(\!(\alpha_n)\!)\gamma$ iff (knowledge and associated local states)

  $I \models K_{\omega(\alpha_1)}(\!(\alpha_1)\!)K_{\omega(\alpha_2)}(\!(\alpha_2)\!)\dots K_{\omega(\alpha_n)}(\!(\alpha_n)\!)\gamma$. $\square$

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Implicitly Coordinated Sequential Plans

So, are we now happy with the definition of implicitly coordinated sequential epistemic plans?

Example

$\Pi = (P, I, \text{Act}, \gamma, \omega)$ with:

- $P = \{p, q, r, t\}$
- $I =$ $s_0 : p, \neg q, \neg r, \neg t$
- $\gamma = t$
- $\text{Act} = \{\alpha_1, \alpha_2, \alpha_3\}$ with
  - $\alpha_1 =$ $e_{11} : \langle p, \neg p \wedge q \rangle$ —a— $e_{12} : \langle p, \neg p \wedge r \rangle$ $\quad \omega(\alpha_1) = a$
  - $\alpha_2 =$ $e_2 : \langle q, t \rangle$ $\quad \omega(\alpha_2) = b$
  - $\alpha_3 =$ $e_3 : \langle r, t \rangle$ $\quad \omega(\alpha_3) = b$

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Implicitly Coordinated Sequential Plans

Example (ctd.)

There is no ICSEP for this planning task. Reason: if there were one, it would have to start with α_1 (nothing else is applicable).

Then, $I^{\omega(\alpha_1)} \otimes \alpha_1$ =

$(s_0, e_{11}) : \neg p, q, \neg r, \neg t \overset{a}{—} (s_0, e_{12}) : \neg p, \neg q, r, \neg t$ $=: (\mathcal{M}^1, S^1_{\mathrm{d}})$.

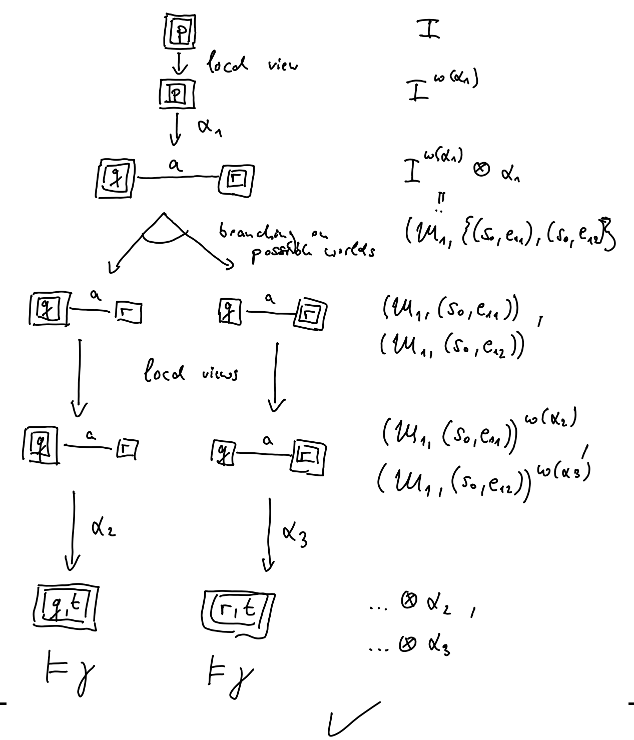
In $(\mathcal{M}^1, S^1_{\mathrm{d}})$, none of the available actions is applicable, and it is not a goal state.

This task would be solvable with a branching or conditional plan: start with α_1, and depending on the outcome, continue with α_2 or α_3. This would even be implicitly coordinated in the sense that at each point in the plan, the agent to move knows that it can move and that this leads to progress.

Branching Plans

We can again define centralized and implicitly coordinated branching plans.

Next slide: Example



Branching Plans

Definition

A global epistemic state is a single-pointed epistemic state.

Idea for formal definition of branching plans: Formalize them as policies mapping global states to sets of actions.

Side note: The obvious thing to do would be to use policies mapping global state to actions, not to sets of actions. So, why do we need sets here? See following example.

Branching Plans

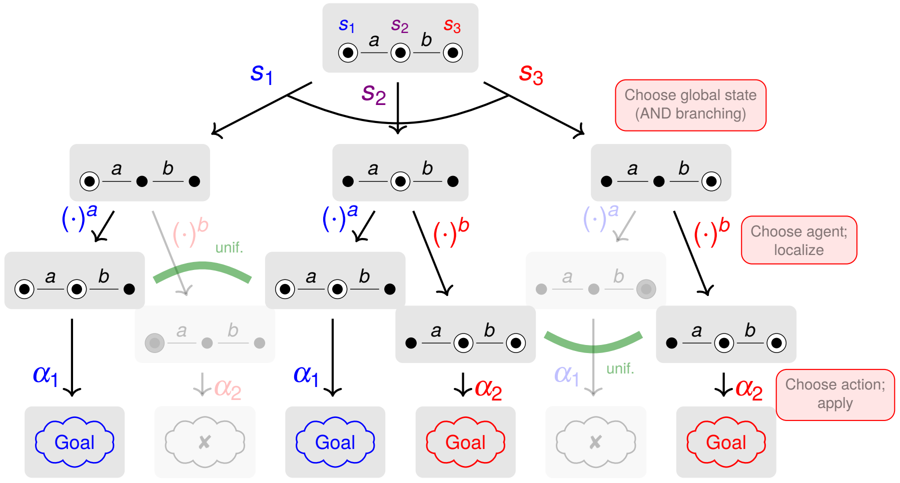
Example

Assume propositions p and q and worlds s_1, s_2, s_3 such that $V_p = \{s_1, s_2\}$ and $V_q = \{s_2, s_3\}$.

Assume actions α_1 with single event $e_1 : (p, \gamma)$ and owner $\omega(\alpha_1) = a$, and α_2 with single event $e_2 : (q, \gamma)$ and owner $\omega(\alpha_2) = b$.

I. e., α_1 is applicable in s_1 and s_2, and α_2 is applicable in s_2 and s_3.

Further, assume that we want to find a plan for the multi-pointed model at the top of the following figure. [...]



AND-OR Search

- AND: Solve an arbitrary state $(\mathcal{M}, S_d)$ by solving all global states $(\mathcal{M}, s)$ with $s \in S_d$.
- OR: Solve a global state $(\mathcal{M}, s)$ by finding an agent a and an action α with $\omega(\alpha) = a$, and solving $(\mathcal{M}, s)^a \otimes \alpha$.

Branching Plans

Example (ctd.)

We have to assign $\{\alpha_1, \alpha_2\}$ to s_2. Both α_1 or α_2 alone would achieve the goal.

But since agent a cannot distinguish between s_1 and s_2, also her policy should not be able to distinguish between s_1 and s_2. Since her policy prescribes α_1 in s_1, it should also prescribe α_1 in s_2 (uniformity!).

Similarly, since agent b cannot distinguish between s_2 and s_3, also his policy should not be able to distinguish between s_2 and s_3. Since his policy prescribes α_2 in s_3, it should also prescribe α_2 in s_2 (uniformity!).

Branching Plans

Notation: In the following, we use s not only to refer to worlds in epistemic models, but also to (single-pointed) epistemic models themselves. Will be clear from the context.

Definition (Policy)

Let $\Pi = (P, I, \mathrm{Act}, \gamma, \omega)$ be an epistemic planning task and let S^{gl} be the set of global epistemic states of Π. Then a policy is a mapping $\pi : S^{\mathrm{gl}} \to 2^{\mathrm{Act}}$ such that:

- Applicability (APP): for all $s \in S^{\mathrm{gl}}$ and all $\alpha \in \pi(s)$, α is applicable in s.
- Determinism (DET): for all $s \in S^{\mathrm{gl}}$ and all $\alpha, \alpha' \in \pi(s)$ with $\omega(\alpha) = \omega(\alpha')$, we have $\alpha = \alpha'$.
- Uniformity (UNIF): for all $s, t \in S^{\mathrm{gl}}$ and all $\alpha \in \pi(s)$ with $s^{\omega(\alpha)} = t^{\omega(\alpha)}$, we have $\alpha \in \pi(t)$.

Branching Plans

Note:
APP and UNIF together imply knowledge of preconditions (KOP): for all $s \in S^{gl}$ and all $\alpha \in \pi(s)$, α is applicable in $s^{\omega(\alpha)}$, i. e., agents supposed to act know that their action is applicable.

Branching Plans

UNI FREIBURG

This definition does not require yet that the goal be reached.
For that, we need some preliminaries first:

Definition (Globals)

For any state $(\mathcal{M}, S_d)$, we let
$\text{Globals}(\mathcal{M}, S_d) = \{(\mathcal{M}, s) \mid s \in S_d\}$.

Usually, a policy π is only considered to be a solution to a planning task if it is closed in the sense that π is defined for all non-goal states reachable following π.

Here: distinguish between closedness refering to all states reachable from a centralized perspective, and closedness refering to all states considered reachable when tracking perspective shifts.

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Branching Plans

UNI FREIBURG

To that end, distinguish between centralized and perspective-sensitive successor functions.

Definition (Successor function)

- A successor function is a function $\sigma : S^{gl} \times \text{Act} \to 2^{S^{gl}}$.
- The centralized successor function is the function $\sigma_{cen}(s, \alpha) = \text{Globals}(s \otimes \alpha)$.
- The perspective-sensitive (or implicitly coordinated) successor function is the function $\sigma_{ps}(s, \alpha) = \text{Globals}(s^{\omega(\alpha)} \otimes \alpha)$.

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Branching Plans

UNI FREIBURG

Intuition:

- σ_{cen} only generates the successors that are objectively possible.
- σ_{ps} also generates successors that are only subjectively considered possible by the respective agent.

Definition (Strong policy)

Let $\Pi = (P, I, \text{Act}, \gamma, \omega)$ be an epistemic planning task and σ a successor function. Then a policy π is called a strong policy for Π with respect to σ if

- Finiteness: π is finite.
- Foundedness: for all $s \in \text{Globals}(I)$, $s \models \gamma$ or $\pi(s) \neq \emptyset$.
- Closedness: for all $(s, \text{Act}') \in \pi$, $\alpha \in \text{Act}'$, $s' \in \sigma(s, \alpha)$, we have $s' \models \gamma$ or $\pi(s') \neq \emptyset$.

Prelude: Two Extensions
Classical Planning
Epistemic Planning
Summary

Branching Plans

Note: No need to explicitly require acyclicity, since transition system is already acyclic (as long as we do not view bisimilar states as identical).

Definition (Centralized and implicitly coordinated branching epistemic plans)

A strong plan with respect to σ_{cen} is called a centralized branching epistemic plan, and a strong plan with respect to σ_{ps} is called an implicitly coordinated branching epistemic plan.

Branching Plans

Proposition

Let π be an implicitly coordinated branching epistemic plan for a planning task $\Pi = (P, I, \text{Act}, \gamma, \omega)$ and let s be a non-goal successor state of I by following π. Then there is an agent $a \in A$ such that $\pi(s)$ contains at least one of agent a's actions and π is an implicitly coordinated branching epistemic plan for $\Pi' = (P, s^a, \text{Act}, \gamma, \omega)$. □

Example

The policy represented by the plan tree above is an implicitly coordinated branching epistemic plan.

Summary

Summary

- Multipointed models and ontic effects
- Review of classical planning
- Centralied vs. implicitly coordinated plans
- Sequential vs. branching plans