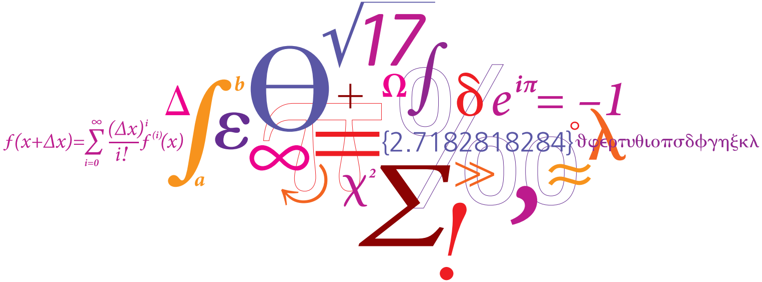
Don't Plan for the Unexpected: Planning Based on Plausibility Models

Thomas Bolander, DTU Informatics, Technical University of Denmark

Joint work with Mikkel Birkegaard Andersen and Martin Holm Jensen

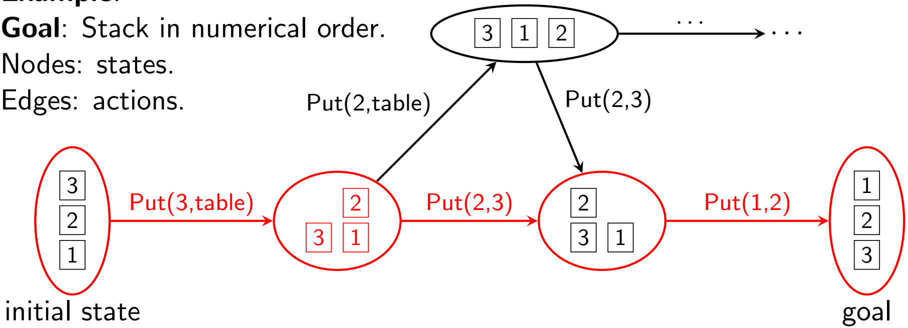


Automated planning

Automated planning (or, simply, **planning**):

- A central subfield of **artificial intelligence** (AI).
- Simplest version: Given an **initial state**, a **goal state** and some available **actions**, compute a sequence of actions (a **plan**) that leads you from the initial state to the goal state.

Example.
Goal: Stack in numerical order.
Nodes: states.
Edges: actions.



Main idea: Generalised planning framework

Essentially: A transition from **classical planning** based on propositional logic to planning based on **Dynamic Epistemic Logic** (**DEL**).

Publications:

1. Bolander and Andersen: *Epistemic planning for single- and multi-agent systems*. JANCL 2011.
2. Andersen, Bolander, Jensen: *Conditional Epistemic Planning*. JELIA 2012.
3. Andersen, Bolander, Jensen: *Dont' Plan for the unexpected: Planning based on Plausibility Models*. Submitted 2012.

Here we present the framework of 3, based on the *dynamic logic of doxastic actions* by Baltag and Smets + postconditions.

Main idea: Advantages

Generalises existing planning approaches by allowing:

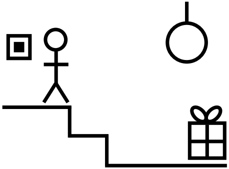
- Planning under partial observability and/or non-determinism with sensing actions. Observations can be action dependent.
- Multi-agent case: Planning including reasoning about other agents (essential to agent communication and collaboration).
- Provides a logical language for reasoning about plans.
- Plausibility models case: Allows for *plausibility planning* in which only the most plausible outcomes of actions are considered.

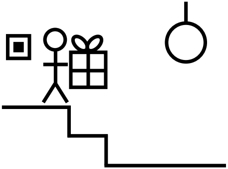
Audience

My expectations regarding the audience:

- Know a lot about dynamic epistemic logic (DEL).
- Might not know so much about automated planning from the mainstream AI perspective.

Running planning example

Initial state: 

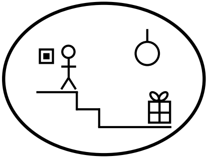
Goal: 

Actions:

- *push*: push the push-button light switch
- *desc*: descend stairs
- *asc*: ascend stairs
- *pick*: pickup Christimas gift

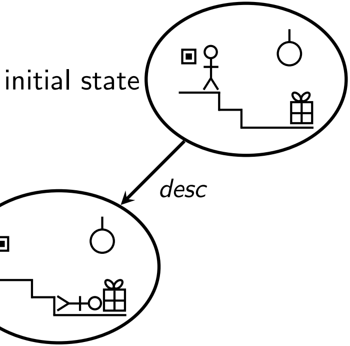
Deterministic planning

In classical planning, all actions are assumed to be **deterministic** (have unique outcomes).

initial state 

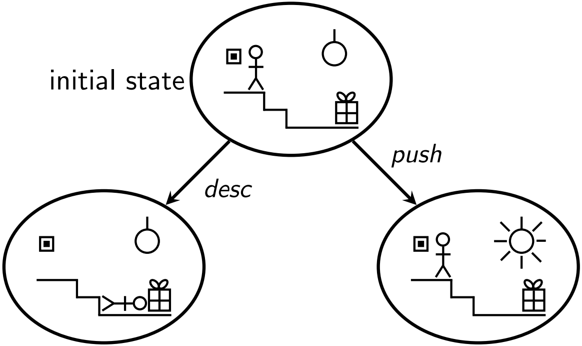
Deterministic planning

In classical planning, all actions are assumed to be **deterministic** (have unique outcomes).



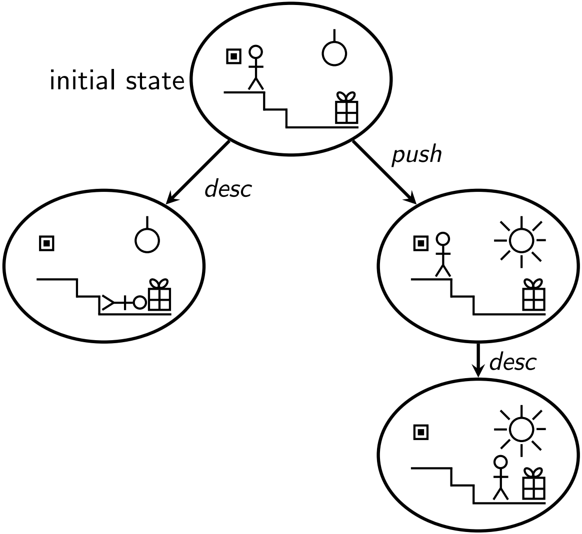
Deterministic planning

In classical planning, all actions are assumed to be **deterministic** (have unique outcomes).



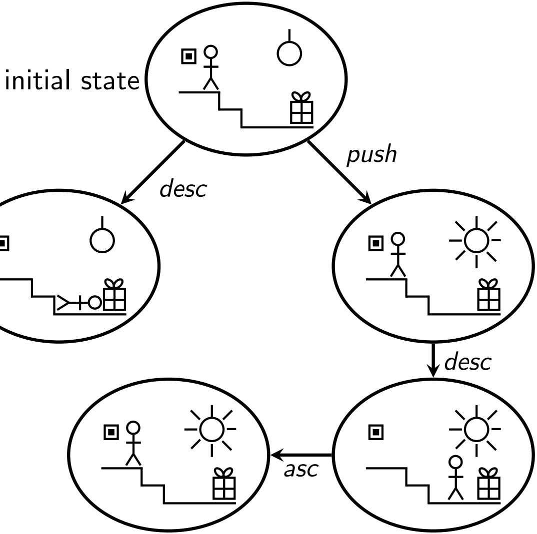
Deterministic planning

In classical planning, all actions are assumed to be **deterministic** (have unique outcomes).



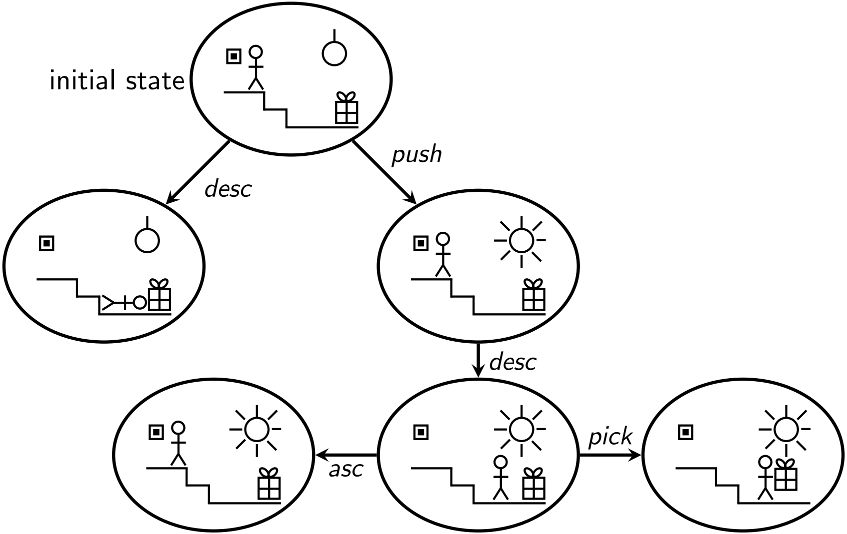
Deterministic planning

In classical planning, all actions are assumed to be **deterministic** (have unique outcomes).



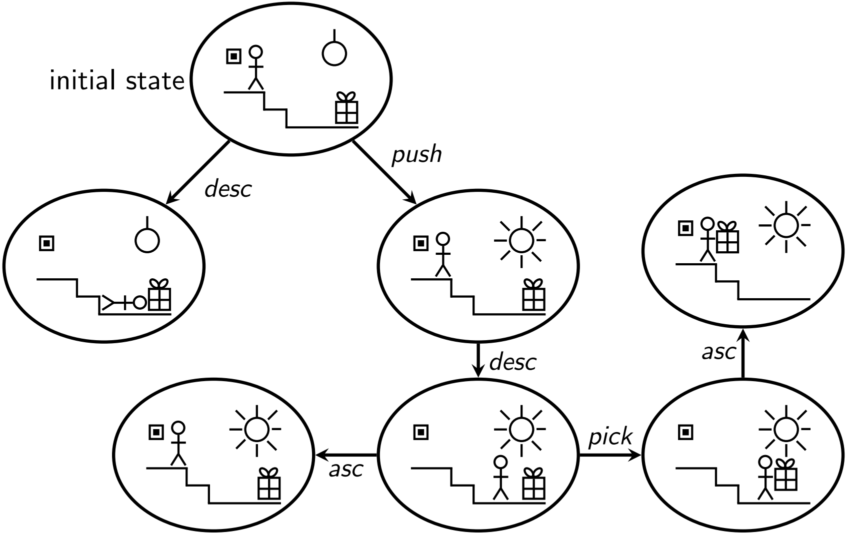
Deterministic planning

In classical planning, all actions are assumed to be **deterministic** (have unique outcomes).



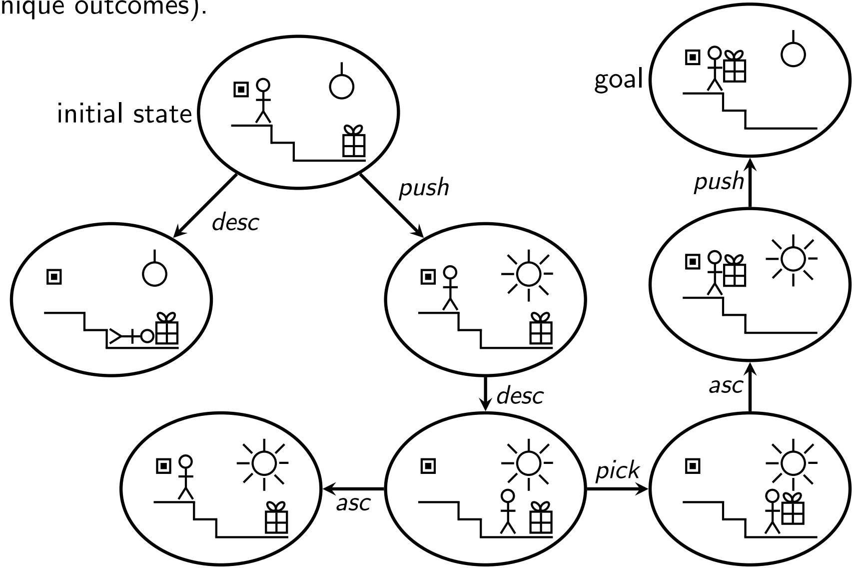
Deterministic planning

In classical planning, all actions are assumed to be **deterministic** (have unique outcomes).



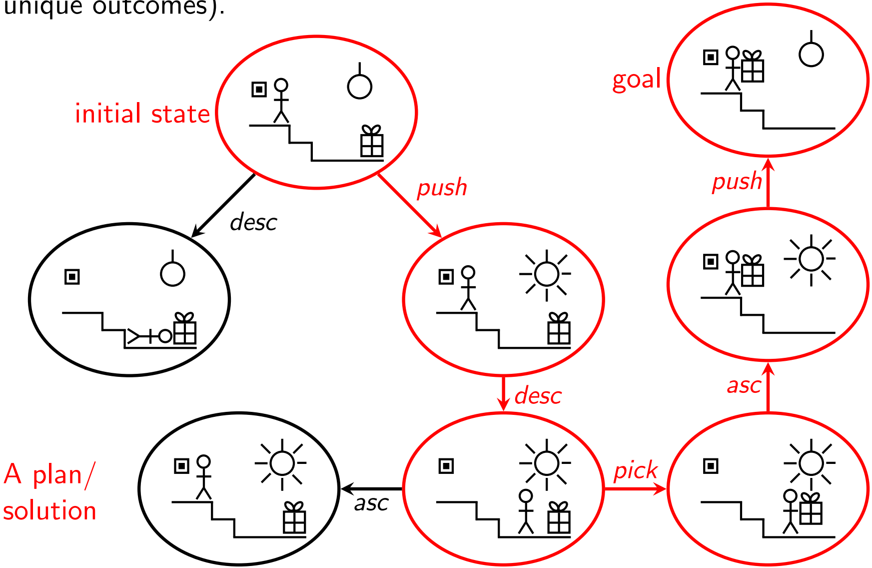
Deterministic planning

In classical planning, all actions are assumed to be **deterministic** (have unique outcomes).

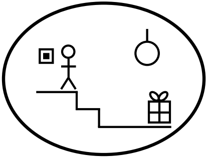


Deterministic planning

In classical planning, all actions are assumed to be **deterministic** (have unique outcomes).



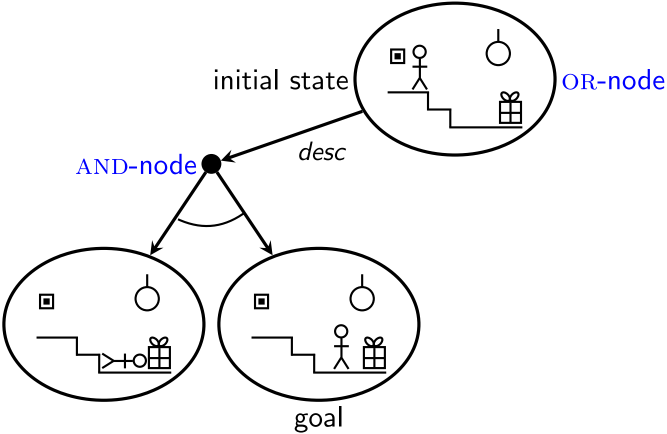
Introducing nondeterminism

initial state  OR-node

Simplified goal: Reach bottom unharmed.

Nondeterminism: AND-OR trees.

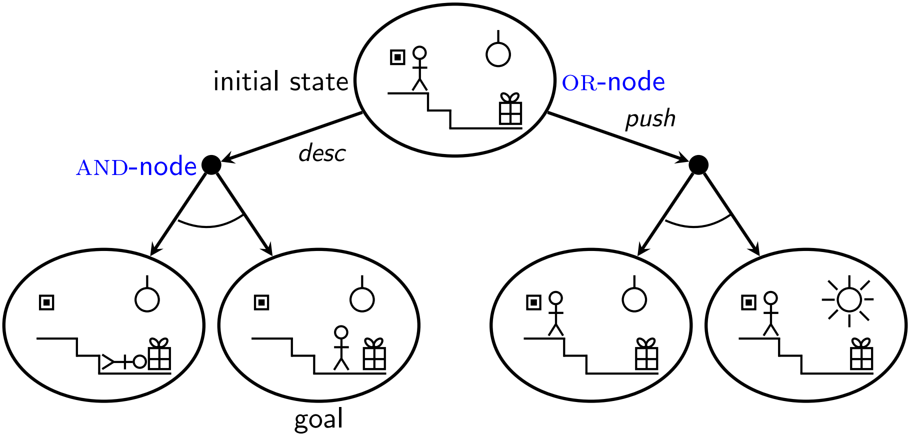
Introducing nondeterminism



Simplified goal: Reach bottom unharmed.

Nondeterminism: AND-OR trees.

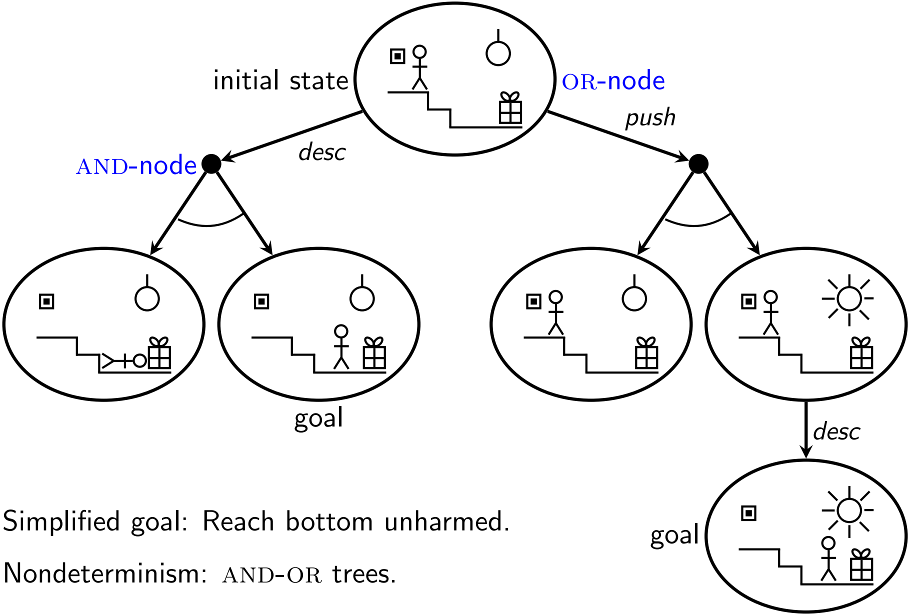
Introducing nondeterminism



Simplified goal: Reach bottom unharmed.

Nondeterminism: AND-OR trees.

Introducing nondeterminism



Simplified goal: Reach bottom unharmed.

Nondeterminism: AND-OR trees.

Conditional plans

When introducing nondeterminism, plans can no longer simply be sequences of actions. They must be conditional on the outcomes.

In the planning literature this is usually done by either:

1. **Policies**: A mapping from states to actions (like strategies in games).
2. **Conditional plans**: A plan language with sequential composition and an if-then-else construct (at least).

We have chosen 2.

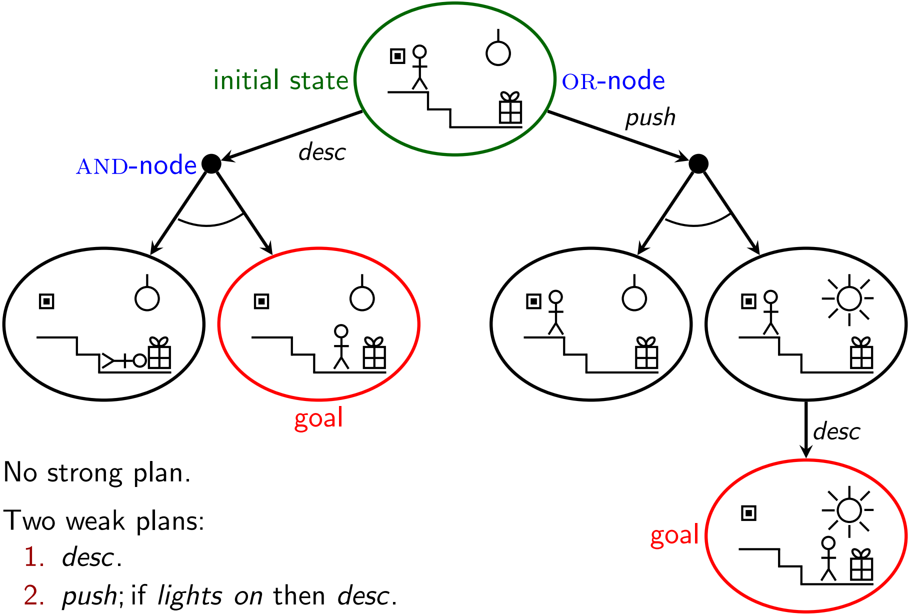
In both cases: A plan chooses an action at each (relevant) OR-node.

Weak and strong plans

Nondeterminism gives rise to multiple types of plans:

- **Strong plan**: The plan *necessarily* achieves the goal (goal reached on all possible execution paths induced by the plan).
- **Weak plan**: The plan *possibly* achieves the goal (goal reached on some execution path induced by the plan).

Weak and strong plans: example



No strong plan.

Two weak plans:

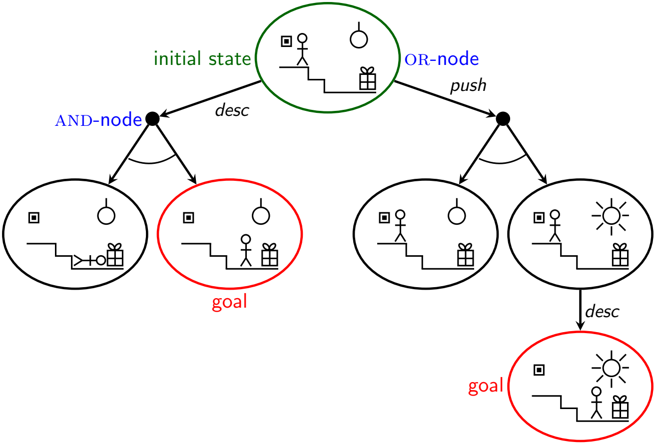
1. *desc*.
2. *push*; if *lights on* then *desc*.

Choosing between weak plans

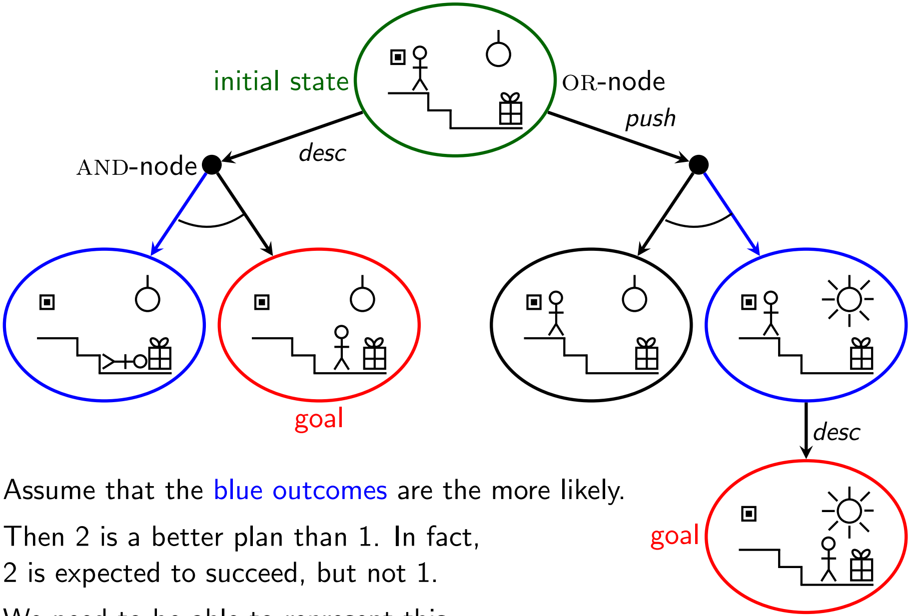
Consider again the two weak plans for descending unharmed:

1. *desc*.
2. *push*; if *lights on* then *desc*.

They are not necessarily equally good. Why?



Choosing between weak plans

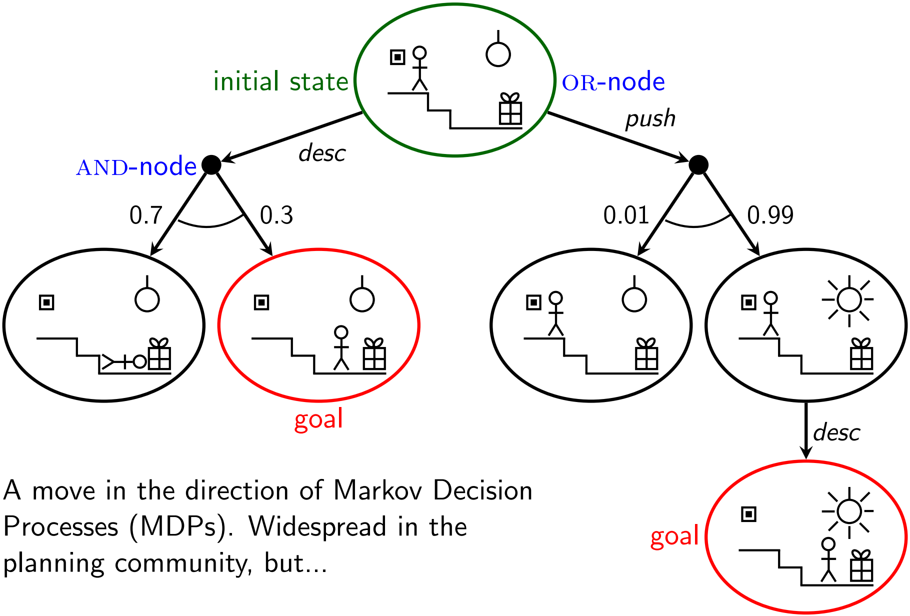


Assume that the blue outcomes are the more likely.

Then 2 is a better plan than 1. In fact, 2 is expected to succeed, but not 1.

We need to be able to represent this...

A possible solution: probabilities



A move in the direction of Markov Decision Processes (MDPs). Widespread in the planning community, but...

From probabilities to plausibilities

Disadvantages with probabilities:

- Where is the planning agent supposed to get them from?
- Heavy machinery when all we want to say is: "plan 2 more plausibly leads to success than 1" or "only 2 is expected to succeed".

From probabilities to plausibilities

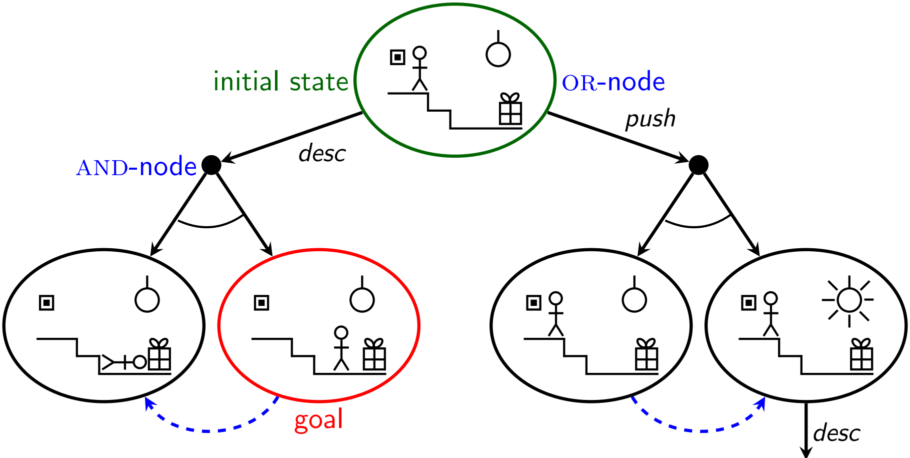
Disadvantages with probabilities:

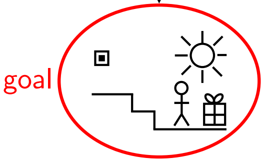
- Where is the planning agent supposed to get them from?
- Heavy machinery when all we want to say is: "plan 2 more plausibly leads to success than 1" or "only 2 is expected to succeed".

Alternative idea:

- Introduce a plausibility ordering on action outcomes...

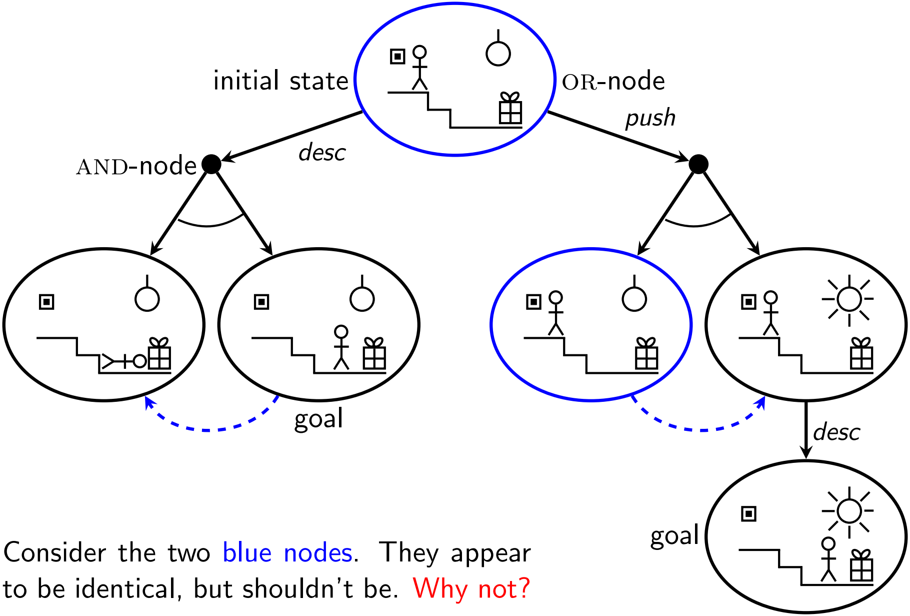
Introducing plausibility orders





Strong plausibility plan: Goal reached on all *most plausible* execution paths induced by the plan.

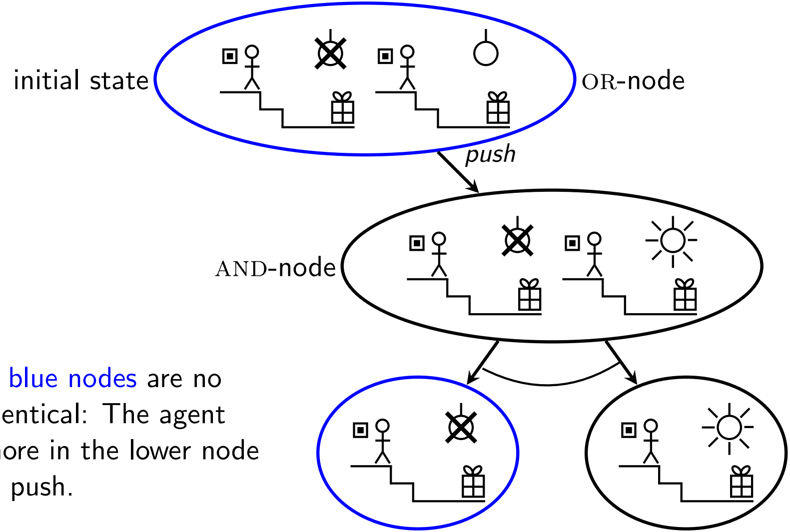
Example. Plan 2 from before is such a plan.



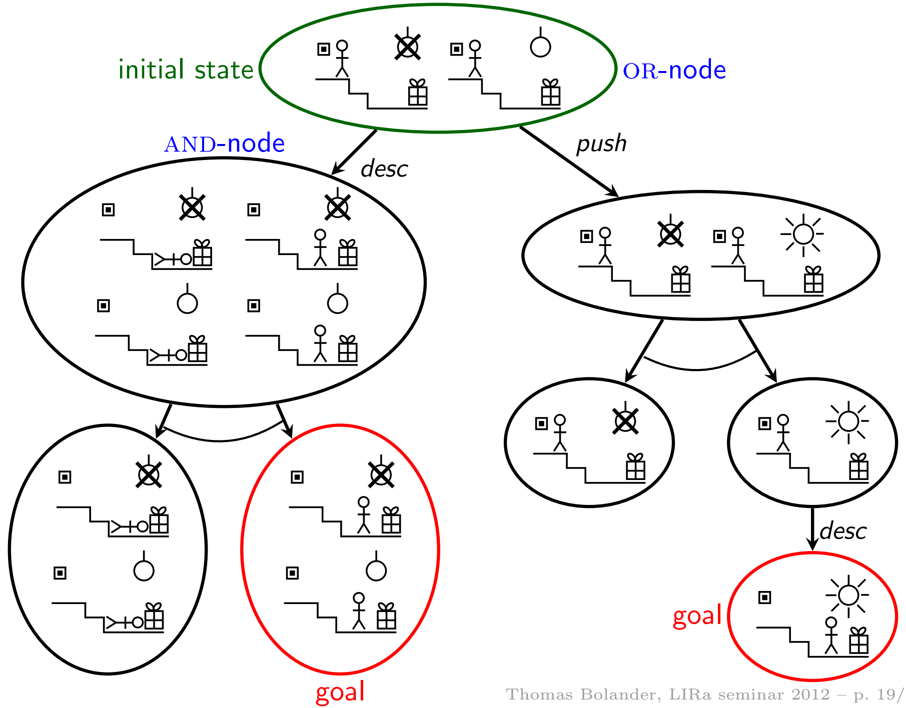
Consider the two blue nodes. They appear to be identical, but shouldn't be. Why not?

Introducing partial observability

We need to introduce partial observability. In automated planning this is often done via **belief states**: sets of indistinguishable states (essentially single-agent S5 models).



Now the blue nodes are no longer identical: The agent knows more in the lower node after the push.


initial state
OR-node
AND-node
desc
push
desc
goal
goal

Observations and DEL-approach

In automated planning, observations are usually dealt with using *action independent* observation models mapping states to belief states or to sets of observation variables.

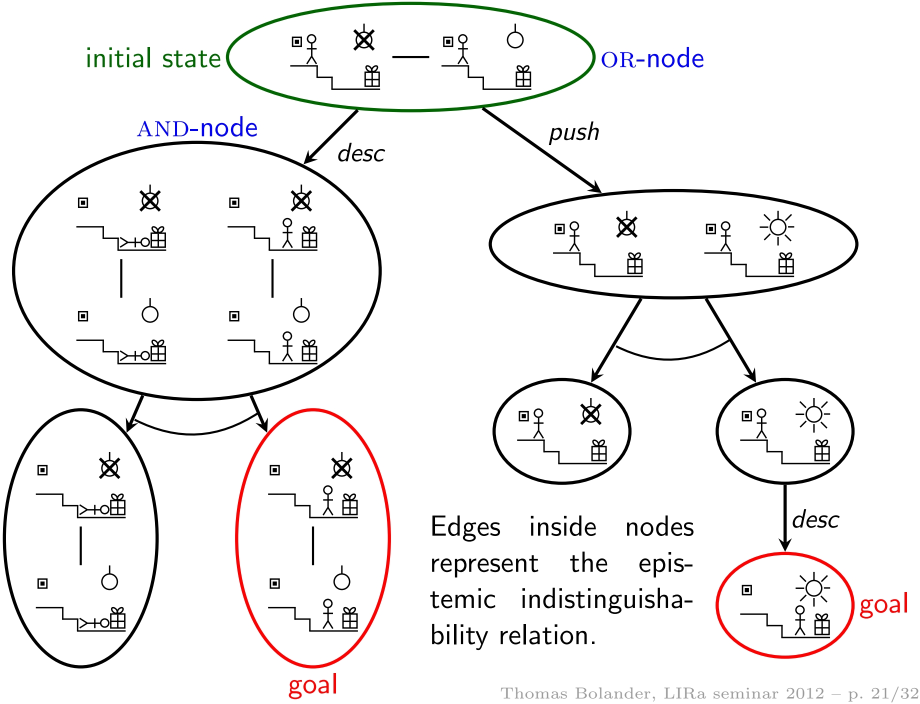
DEL offers a better solution:

- Let **states** be single-agent S5 models.
- Let **actions** be event models of DEL.

Then observations are taken care of by the event models, and observations become *action dependent*. More expressive.

Other advantages of a DEL-based approach:

- State space is induced by actions (event models) rather than being explicitly given in advance. Advantage over MDP-based approaches and more in line with classical planning.
- Generalises naturally to the plausibility and multi-agent cases.



Edges inside nodes represent the epistemic indistinguishability relation.

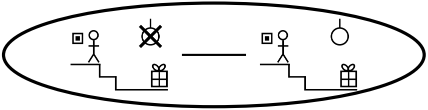
Running example more formally

Propositional letters:

t: top of stairs l: light on b: broken bulb u: unharmed

Initial state s_0 is:

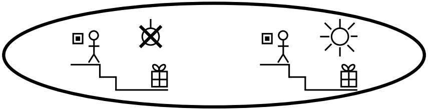
$t\underline{l}bu$ —— $t\underline{l}\underline{b}u$



Action *push* is: $\langle t \wedge b, \emptyset \rangle$ $\langle t \wedge \neg b, \{l \mapsto \neg l\} \rangle$

The **product update** $s_0 \otimes push$ is:

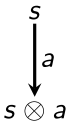
$t\underline{l}bu$ $tl\underline{b}u$



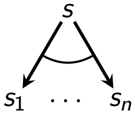
Construction of and-or trees in DEL-based setting

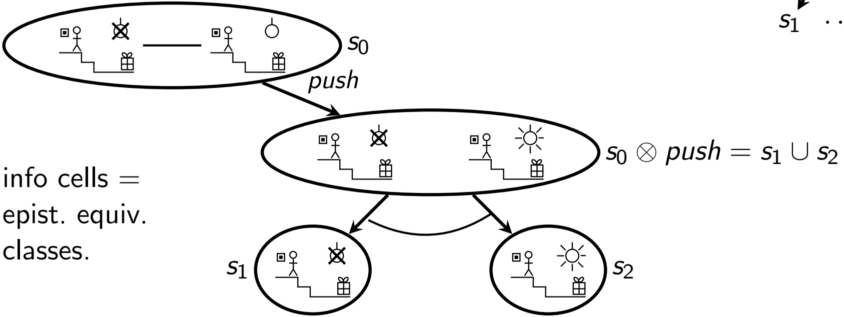
In the DEL-based setting, the AND-OR tree is built by starting with the initial state and repeatedly applying the following **tree expansion rules**:

- Expansion of an OR-node s: $s \xrightarrow{\text{choose } a}$



- Expansion of an AND-node s: $s \xrightarrow{s_1, \ldots, s_n \text{ are the info cells of } s}$





info cells = epist. equiv. classes.

Adding plausibilities

How about the plausibilities? First try: Add a plausibility ordering $\leq \subseteq \sim$. This is the (by now) standard approach.

means $w_1 \sim w_2$ and $w_1 \leq w_2$ (w_1 more plausible than w_2).

Adding plausibilities

How about the plausibilities? First try: Add a plausibility ordering $\leq \subseteq \sim$. This is the (by now) standard approach.

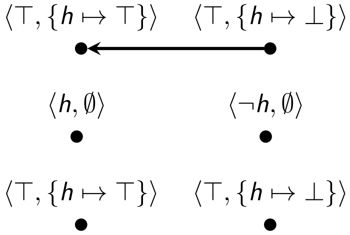
means $w_1 \sim w_2$ and $w_1 \leq w_2$ (w_1 more plausible than w_2).

Example. Consider a coin biased toward heads h.

Unobserved coin toss a_{toss} :

Observe coin a_{obs} :

Observed coin toss ($= a_{\text{toss}} \otimes a_{\text{obs}}$) $a_{\text{toss+obs}}$:



Adding plausibilities

How about the plausibilities? First try: Add a plausibility ordering $\leq \subseteq \sim$. This is the (by now) standard approach.

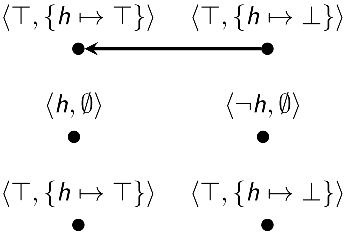
means $w_1 \sim w_2$ and $w_1 \leq w_2$ (w_1 more plausible than w_2).

Example. Consider a coin biased toward heads h.

Unobserved coin toss a_{toss} :

Observe coin a_{obs} :

Observed coin toss ($= a_{\text{toss}} \otimes a_{\text{obs}}$) $a_{\text{toss+obs}}$:



This is no good! When *considering* to perform an observed coin toss, I still need to know that $h \mapsto \top$ is more plausible than $h \mapsto \bot$.

Global plausibility

We need the plausibility relation to be **global**, **a priori**, representing the planning agent's prior beliefs about possible outcomes of future actions.

Hence, instead we take $\leq$ to be a **total** preorder.

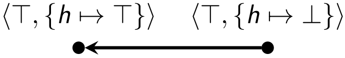
Now:

means $w_1 \sim w_2$ and $w_1 \leq w_2$.

means $w_1 \not\sim w_2$ and $w_1 \leq w_2$.

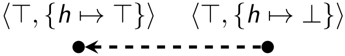
Global plausibility cont'd

Unobserved coin toss a_{toss} :



Observe coin a_{obs} :

Observed coin toss ($= a_{\text{toss}} \otimes a_{\text{obs}}$) $a_{\text{toss+obs}}$:

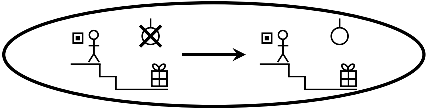


- a_{toss}: If I perform a_{toss} I will not come to know whether $h \mapsto \top$ or $h \mapsto \bot$ has happened, but I believe $h \mapsto \top$ to be most plausible.
- $a_{\text{toss+obs}}$: If I perform $a_{\text{toss+obs}}$ I **will** come to know whether $h \mapsto \top$ or $h \mapsto \bot$ has happened, and I currently believe $h \mapsto \top$ to be the most plausible.

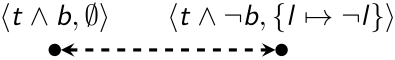
We can of course also define a local, a posteriori, plausibility relation: $\trianglelefteq := \leq \cap \sim$. This is the approach of Baltag and Smets [2006].

Example revisited

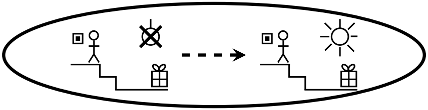
Initial state s_0 is:



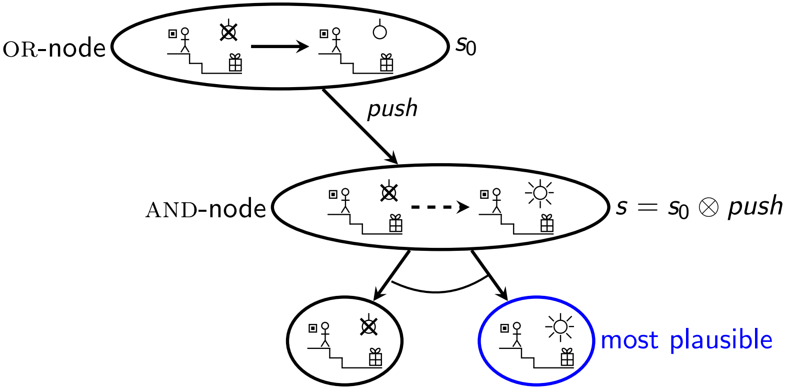
Action *push* is:



The **product update** $s_0 \otimes push$ is:



Most plausible children



Most plausible information cells of a node: The epistemic equivalence classes containing the most plausible worlds.

A child of an AND-node s is called **most plausible** if it is a most plausible information cell in s.

Strong plausibility planning: Goal reached on all most plausible executing paths induced by the plan.

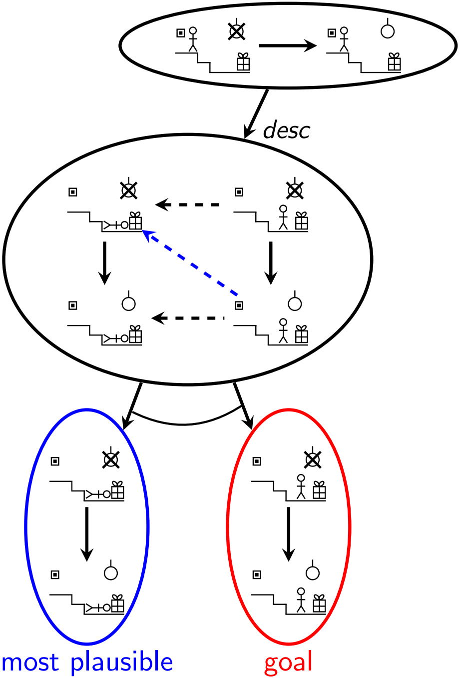
The blue edge is a consequence of choosing the **action-priority update rule** of Baltag and Smets.

Works here, but in certain scenarios using action-priority update in a setting with ontic actions (postconditions) produce counterintuitive consequences.

desc :

$\langle t \wedge \neg l, \{t \mapsto \bot, u \mapsto \bot\}\rangle$

$\langle t, \{t \mapsto \bot\}\rangle$



Some syntax—finally

We've mainly been talking semantics. The underlying syntax chosen is as follows.

Dynamic language:

$$\phi, \psi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid K_l \phi \mid B_g^{\psi} \phi \mid X\phi \mid [\mathcal{E}, e]\,\phi$$

where p is a propositional symbol, $\mathcal{E}$ is an event model and e a basic event in $\mathcal{E}$. K_l is the *local* knowledge modality, B_g^{ψ} the *global* conditional belief modality, and X a non-standard *localisation* modality.

Plan language:

$$\pi ::= \mathcal{E} \mid \text{skip} \mid \text{if } \phi \text{ then } \pi \text{ else } \pi \mid \pi ; \pi$$

where $\mathcal{E}$ is an event model from a fixed, finite **action library** (the actions available to the agent) and ϕ is a formula of the dynamic language.

Putting it all together

Combining the tree-expansion rules from before with a standard loop-check, we can construct an algorithm PLAN for solving planning problems in the plausibility-based framework. Solutions are plans expressed in the plan language.

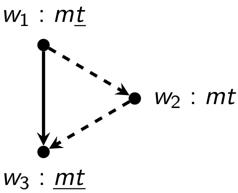
Main Theorem. PLAN *is a terminating, sound and complete algorithm for producing strong/weak/strong plausibility/weak plausibility plans to any given planning problem.*

Note: This is the single-agent case. The multi-agent case is known to be undecidable (Bolander and Andersen [2011]); even without postconditions (Bolander and Aucher [2013, to appear]).

Summing up

- Presented a planning framework based on the dynamic logic of doxastic actions (Baltag & Smets). Allows for plausibility planning: Planning only for the most plausible outcomes of your actions.
- Future work:
  - Generalise to multiple agents.
  - Generalise to more layers of plausibility (using degrees of belief).
  - Embed in replanning architecture: replan when not ending up in the expected (most plausible) information cells.
- Unsolved issues:
  - Action-priority update has counterintuitive consequences for some types of ontic action.
  - Choice of most plausible information cell is not always consistent with a quantitative interpretation.

Appendix



Trying to pay for a Friday Beer with a credit card. Agent believes it most plausible that no money is left on the card. Figure shows (her current beliefs about) the situation after having tried to pay.

- m: Money left on the account.
- t: Transaction goes through.

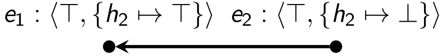
Transaction fails due to insufficient funds (w_3) $\leq$ transaction goes through (w_2) $\leq$ transaction fails due to machine failing (w_1).

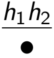
Most plausible info cell: $\{w_1, w_3\}$.

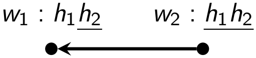
Appendix

Let there be given two identical coins, biased toward heads (h). Tossing them without observing the outcome is represented by the following event models, respectively:

$toss_1$: 

$toss_2$: 

s_0 : 

Now $s_0 \otimes toss_1 =$ 

And $s_0 \otimes toss_1 \otimes toss_2 =$