Knowledge, Reasoning and Planning

1. Organization, Dates & Topics

Malte Helmert, Tanja Schindler and Travis Rivera Petit

University of Basel

March 2, 2026

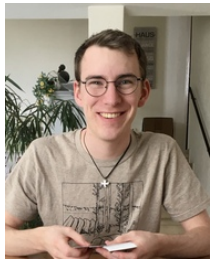
Organization

Organizers and Supervisors

Malte Helmert

Tanja Schindler

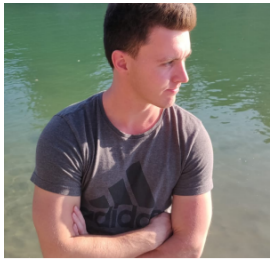
Travis Rivera Petit



Clemens Büchner

Remo Christen

Gustavo Delazeri



Simon Dold

Claudia Grundke

Florian Pommerening

Gabriele Röger

David Speck

Organizers

Organizers

Tanja Schindler (main contact)

- email: `tanja.schindler@unibas.ch`
- office: Spiegelgasse 1, room 04.005

Travis Rivera Petit (main contact)

- email: `travis.riverapetit@unibas.ch`
- office: Spiegelgasse 5, room 04.001

Malte Helmert

- email: `malte.helmert@unibas.ch`
- office: Spiegelgasse 1, room 06.004

Other Supervisors

- Spiegelgasse 1, room 04.005
  - Gabriele Röger (gabriele.roeger@unibas.ch)
  - Florian Pommerening (florian.pommerening@unibas.ch)
  - Clemens Büchner (clemens.buechner@unibas.ch)
- Spiegelgasse 5, room 04.003
  - David Speck (davidjakob.speck@unibas.ch)
- Spiegelgasse 5, room 04.001
  - Remo Christen (remo.christen@unibas.ch)
  - Simon Dold (simon.dold@unibas.ch)
  - Gustavo Delazeri (gustavo.delazeri@unibas.ch)
  - Claudia Grundke (claudia.grundke@unibas.ch)

Target Group and Prerequisites

Target Group

- MSc students of Computer Science and similar subjects

Prerequisites

- ideally: completed the course “Planning and Optimization”

Capacity Limit

Capacity: ≤ 20 students in this seminar

Priority rules:

- 1st priority: students who have already successfully completed the course "Planning and Optimization"
- 2nd priority: students who are enrolled in the MSc computer science programme with major in machine intelligence
- provided that they have registered for the seminar until February 20, 2026
- otherwise first-come first-served

Format

Seminar Format

- theoretical part + project
- 6 ECTS points
- assessment: graded (1.0–6.0)

Objectives

Objectives

Theoretical part:

- reading and understanding scientific literature
- preparing and presenting scientific talks
- conducting scientific discussions with peers
- writing and discussing scientific reports

Project:

- practical project related to the seminar topic
- analysing and discussing experimental results

Requirements (I)

Presentation (25%)

- 20 min. presentation + 5 min Q&A
- includes one slide on the plan for the project
- send draft to supervisor one week before the talk
- send final slides the evening before the talk

Requirements (II)

Written Report (25%)

- 10–12 pages (excluding references) in LaTeX
- initial version will be reviewed by another student
- only the final version is graded
- due: April 15 (initial version)/May 6 (final version)

Peer Review (15%)

- give feedback on the report of another student
- single-blind
- 1–2 pages
- due: April 26

Requirements (III)

Project (25%)

- project related to the seminar assignment
- agree on the topic with your supervisor by March 27
- due: May 13

Project Presentation (10%)

- 5 min. presentation including a live demo

Active Participation

- attend each meeting
- participate in the discussions

Grading

Grading

- Presentation: 25%
- Written Report: 25%
- Peer Review: 15%
- Project: 25%
- Project Presentation: 10%

- each part is individually graded (1.0–6.0)

Plagiarism

Plagiarism

- plagiarism: presenting work or ideas of others as your own
  - this includes AI-generated content!
- consequence: failing the seminar
- if in doubt: ask us in advance!

Repeat offenders can be excluded from the study program.

Course Links

Seminar Homepage

https://dmi.unibas.ch/en/studies/computer-science/course-offer-spring-2026/77779-seminar-knowledge-reasoning-and-planning/

- seminar description
- slides
- examples for good talk slides, reports, and reviews
- link to ADAM workspace

ADAM Workspace

- additional material that we cannot share publicly
- link to Discord server

Discord Server

- place to interact with us and with each other

Schedule (Part 1)

Mon	Mar 02	12:15–14:00	organization, dates & topics
Fri	Mar 06	$\leq$ 11:59	mail us $\geq$ 3 topic wishes
Fri	Mar 06	$\leq$ 23:59	you receive a topic
Mon	Mar 09	12:15–14:00	intro on reports, reviews & presentations
Mon	Mar 16		no meeting
Mon	Mar 23	12:15–14:00	seminar presentations
Fri	Mar 27	$\leq$ 15:00	agree on a project topic with your mentor
Mon	Mar 30	12:15–14:00	seminar presentations
Mon	Apr 06		no meeting (Easter)
Mon	Apr 13	12:15–14:00	seminar presentations

Mail to

`tanja.schindler@unibas.ch`, `travis.riverapetit@unibas.ch`
CC: `malte.helmert@unibas.ch`

Schedule (Part 2)

Wed	Apr 15	$\leq$ 23:59	mail us your report draft
Mon	Apr 20	12:15–14:00	seminar presentations
Sun	Apr 26	$\leq$ 23:59	mail us your peer review
Mon	Apr 27	12:15–14:00	seminar presentations
Mon	May 04	12:15–14:00	seminar presentations
Wed	May 06	$\leq$ 23:59	mail us the final version of your report
Mon	May 11	12:15–14:00	seminar presentations
Wed	May 13	$\leq$ 23:59	final commit of your project
Mon	May 18	12:15–14:00	project presentations
Mon	May 25		no meeting (Pentecost)

Mail to

`tanja.schindler@unibas.ch`, `travis.riverapetit@unibas.ch`
CC: `malte.helmert@unibas.ch`

Questions about the Organization

Questions?

Topics

1. Search Algorithms for Planning with Bounded Suboptimality

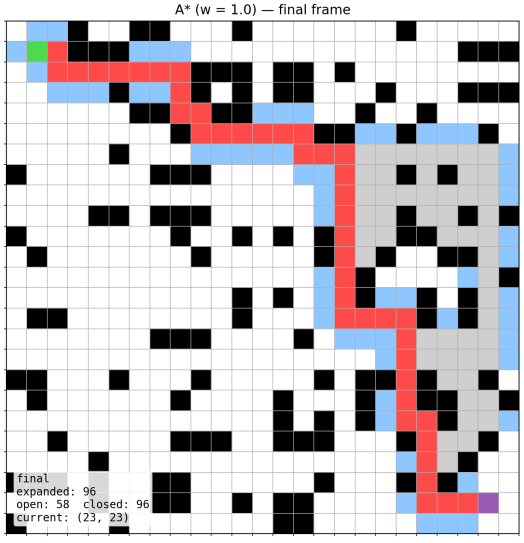


Figure: A*

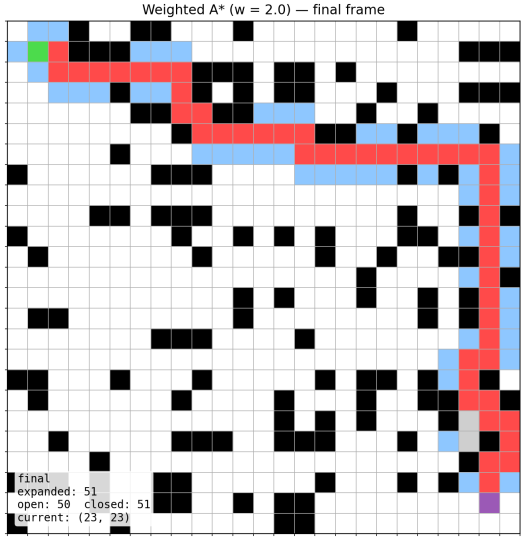
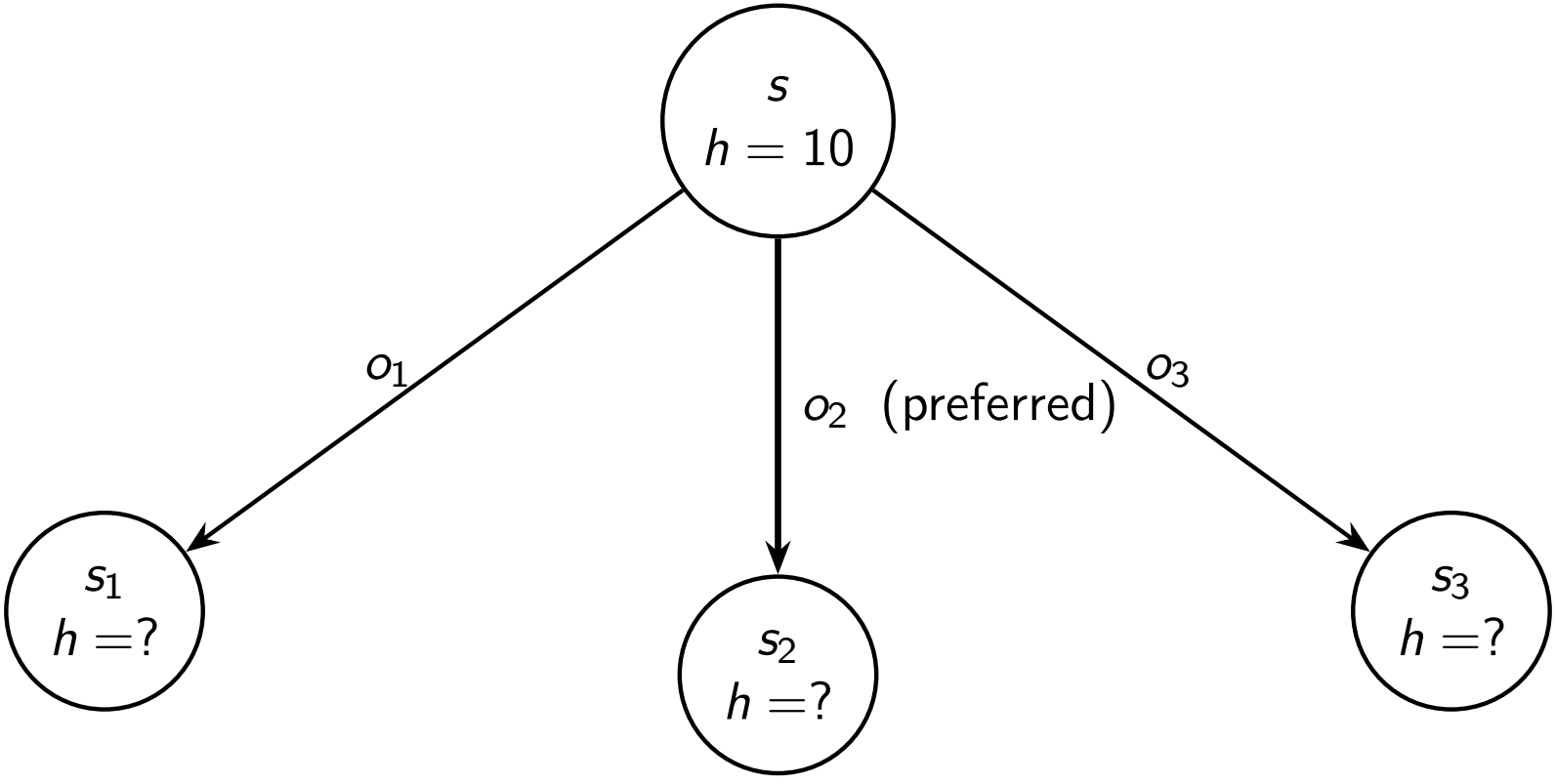


Figure: Weighted A* (w = 2.0)

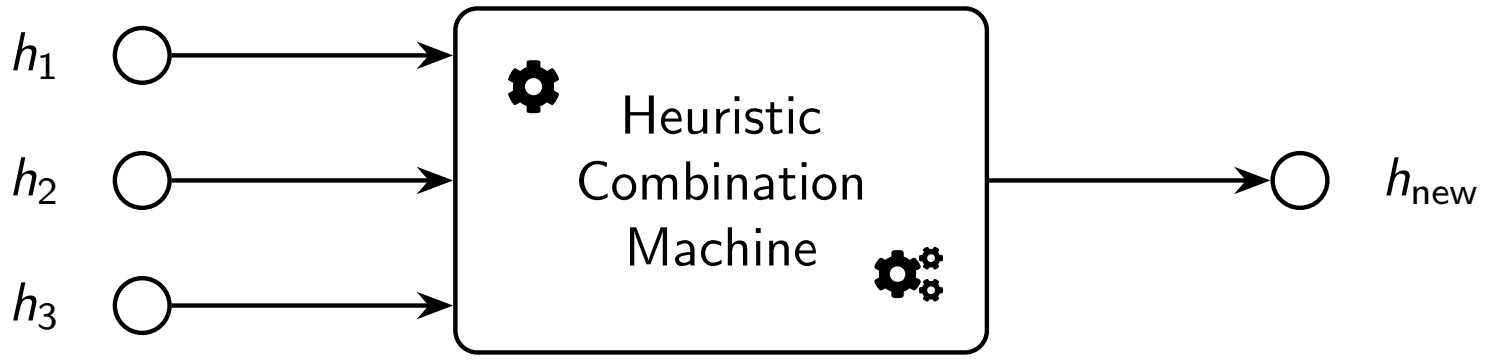
- "I want to find a plan that is at most a factor k away from being optimal"
- Focal search, Weighted A*, etc.

2. Preferred Operators and Deferred Evaluation



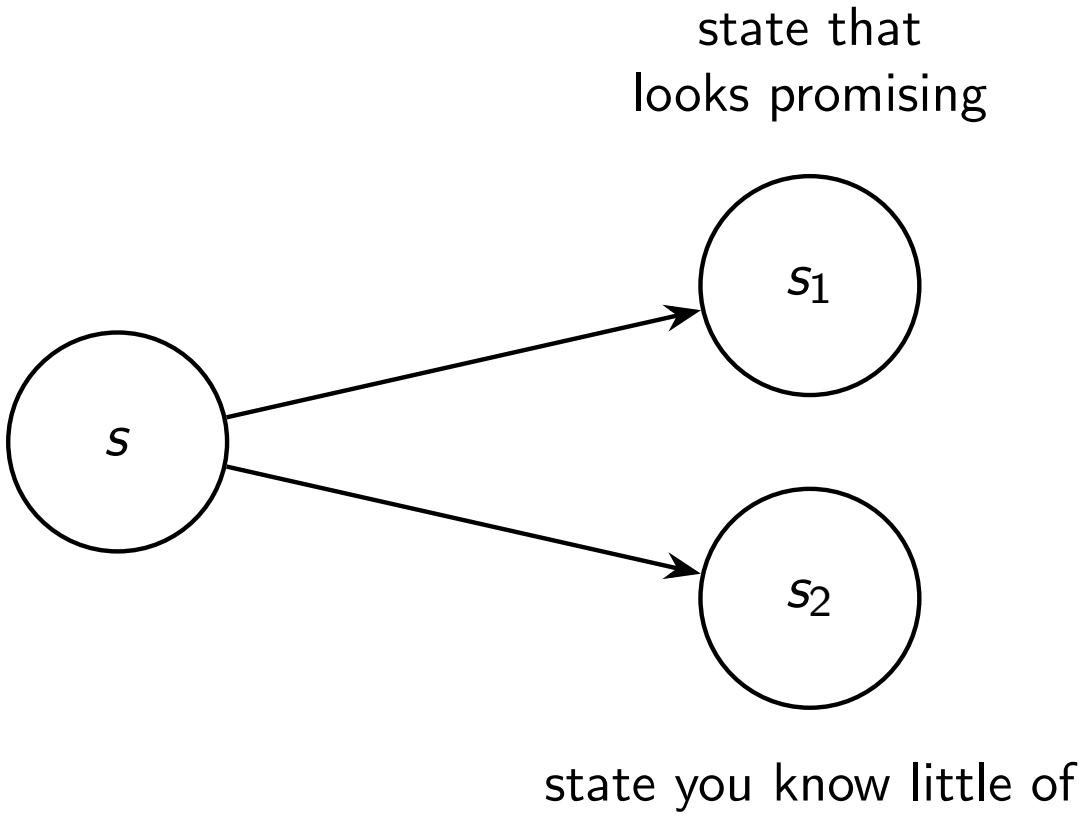
- Deferred evaluation: use parent's h value
- Preferred operators: prioritize operators liked by the heuristic

3. Satisficing Search with Multiple Heuristics



- different heuristics good for different search space regions
- Scorpion Maidu, winner of the last international planning competition alternates between six open lists

4. Width- and Novelty-Based Search



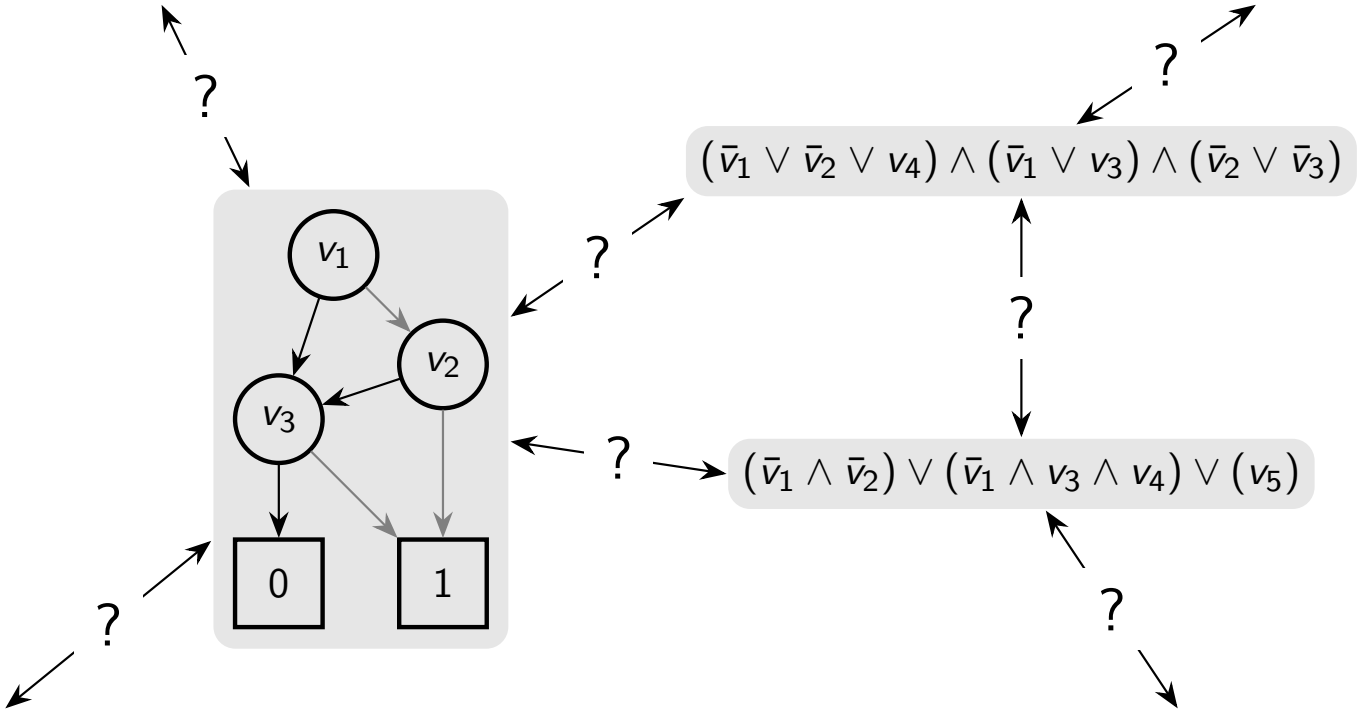
- exploration vs. exploitation tradeoff
- "novelty" in planning

5. Planning as Symbolic Search

```
while goal-states ∩ visited-states ≠ ∅:
  if no applicable operators:
    return unsolvable
  visited-states.apply(operators)
return plan
```

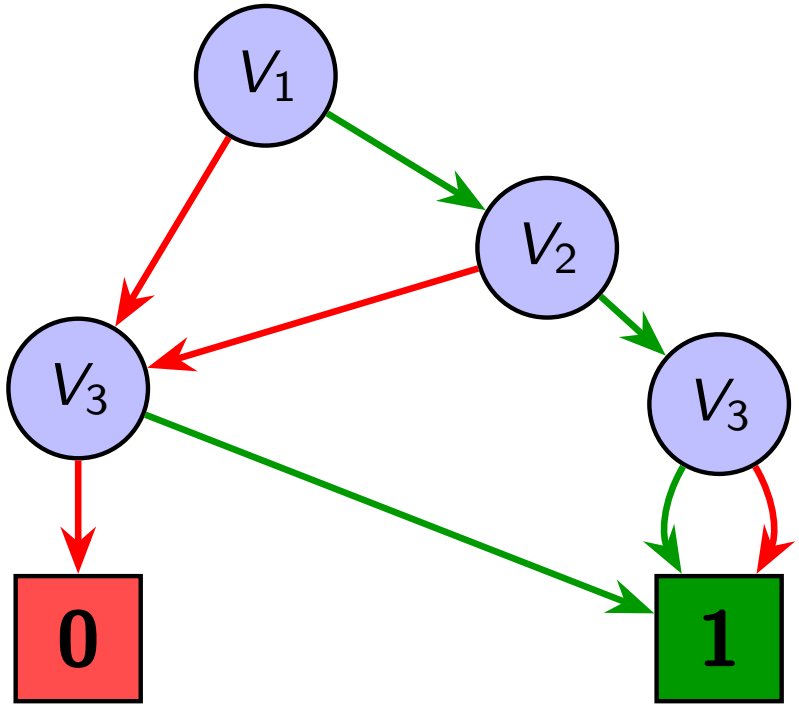
- keep track of all visited states using a compact representation
- simulate a standard search algorithm like BFS

6. Knowledge Compilation



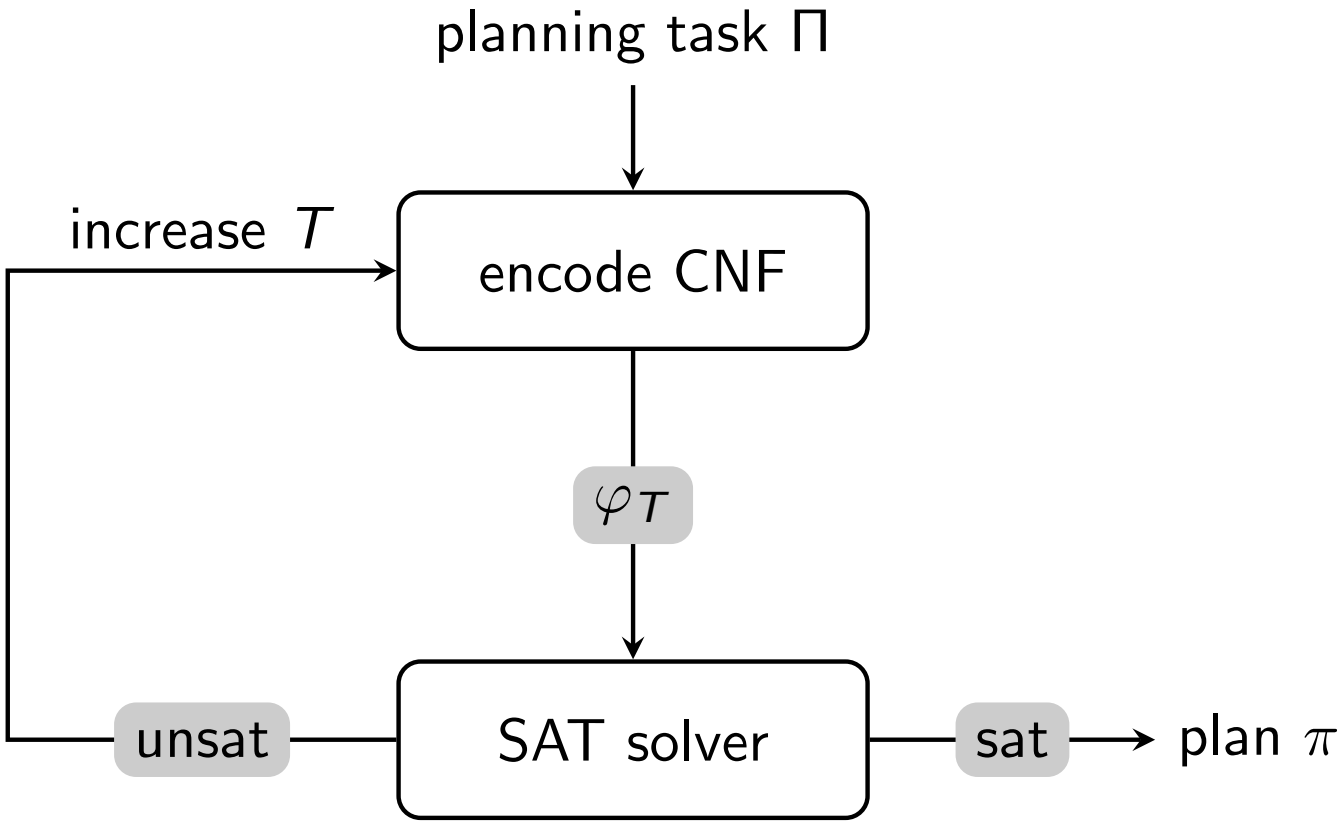
- How can knowledge be represented succinctly?
- Which queries can be answered efficiently?
- Which operations can be performed efficiently?

7. Decision Diagrams



- compact representation of logical formulas
- BDDs, ZDDs, relaxed decision diagrams, restricted decision diagrams

8. Planning as SAT



- encode plan existence within horizon T as Boolean satisfiability problem
- different encodings: sequential, $\forall$-step, $\exists$-step, ...

9. Planning with Property Directed Reachability

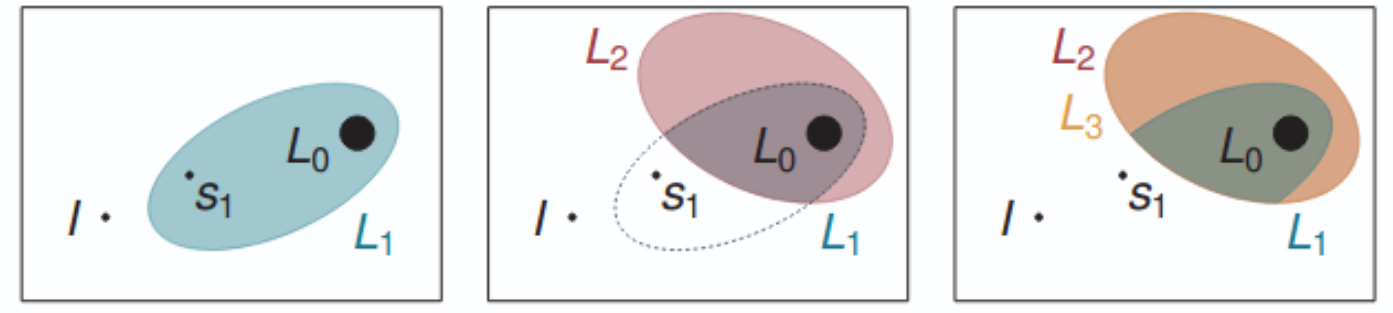


Figure: Eriksson and Helmert (2020)

- Use propositional formulas to overapproximate set of states within k steps from the goal.
- Refine overapproximation using counterexamples.

10. Tractable Boolean Satisfaction Problems

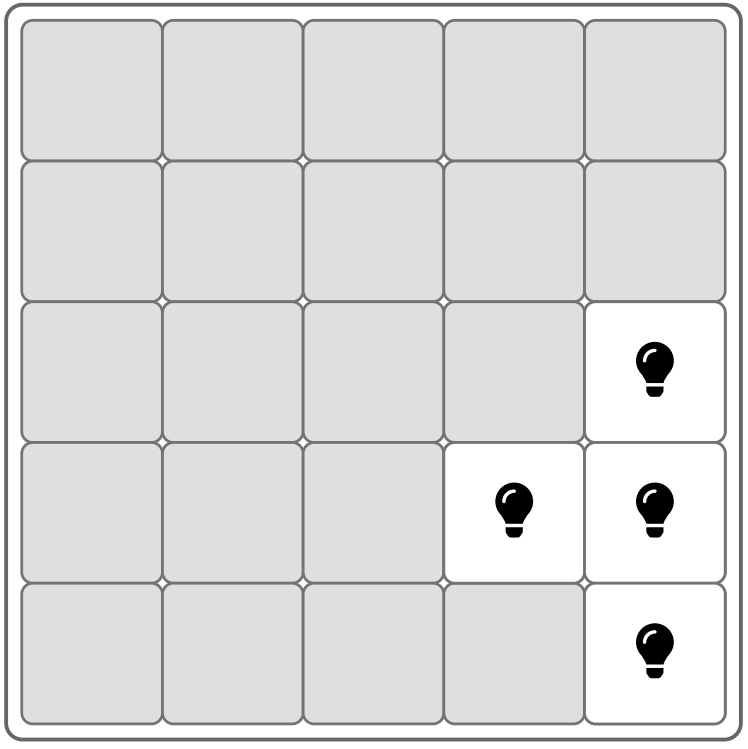


Figure: Lights Out puzzle

- some boolean constraint satisfaction problems are tractable
- Which ones? Why?
- relationship to SAT

11. SAT Solving

Is the following formula satisfiable?

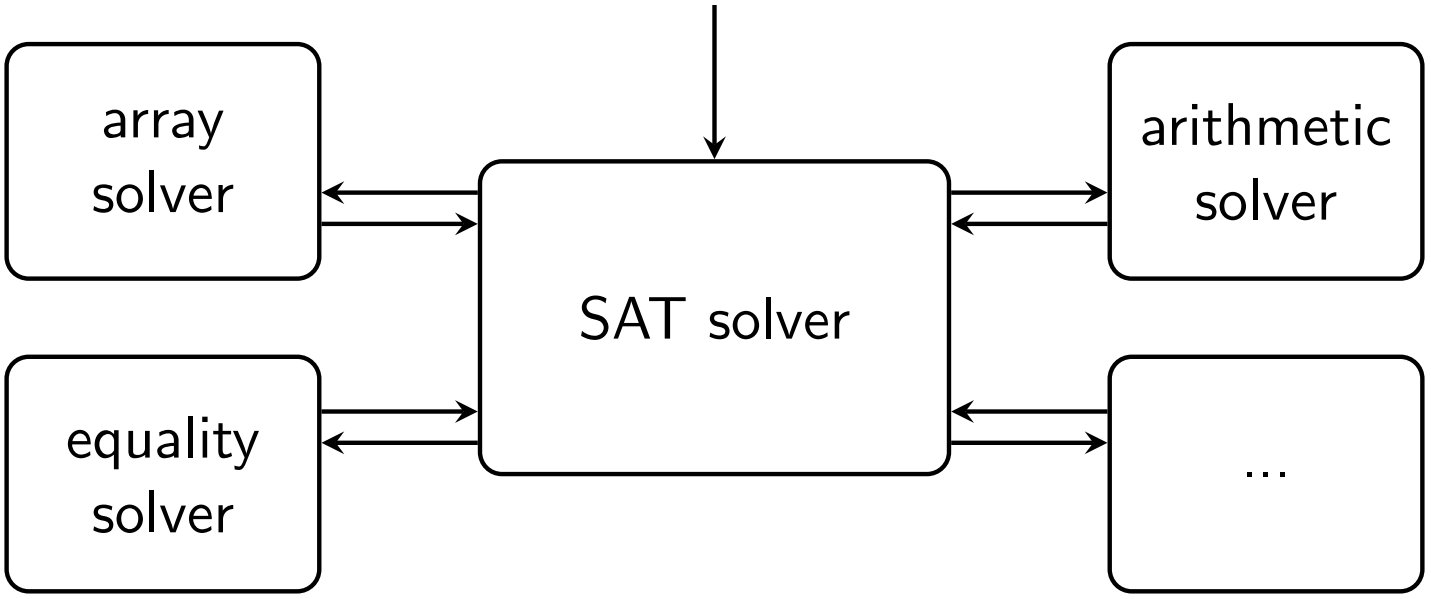
$$(p \vee q \vee \neg r) \wedge (\neg p \vee r) \wedge (\neg q) \wedge (p \vee r)$$

- SAT is NP-complete
- SAT solvers work well in practice
- basic algorithm DPLL/CDCL

12. SMT Solving

Is the following formula satisfiable modulo theory T?

$$(a = 0) \wedge (b = 0 \vee b = c) \wedge (c = a - 2) \wedge (f(a) \neq f(b))$$



- basic algorithm DPLL(T)/CDCL(T)
- decision procedures

13. Plan Improvement

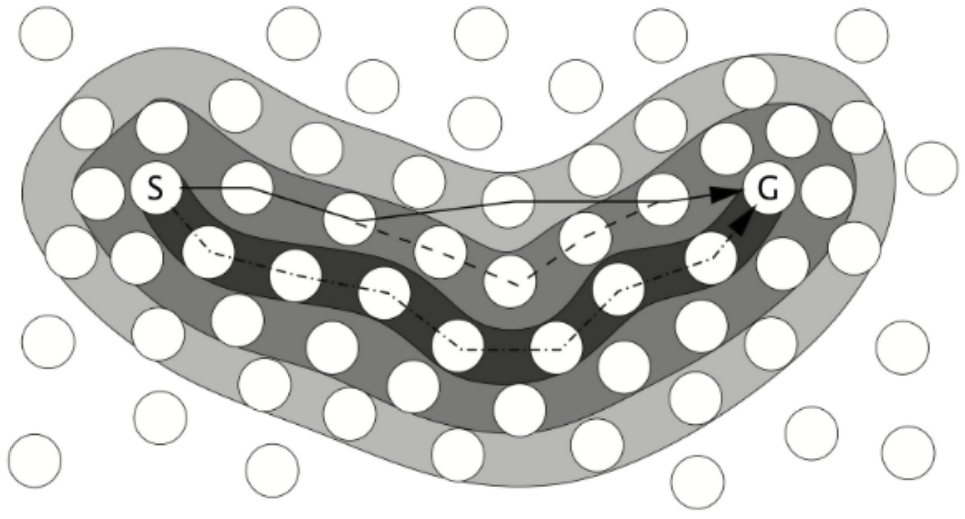
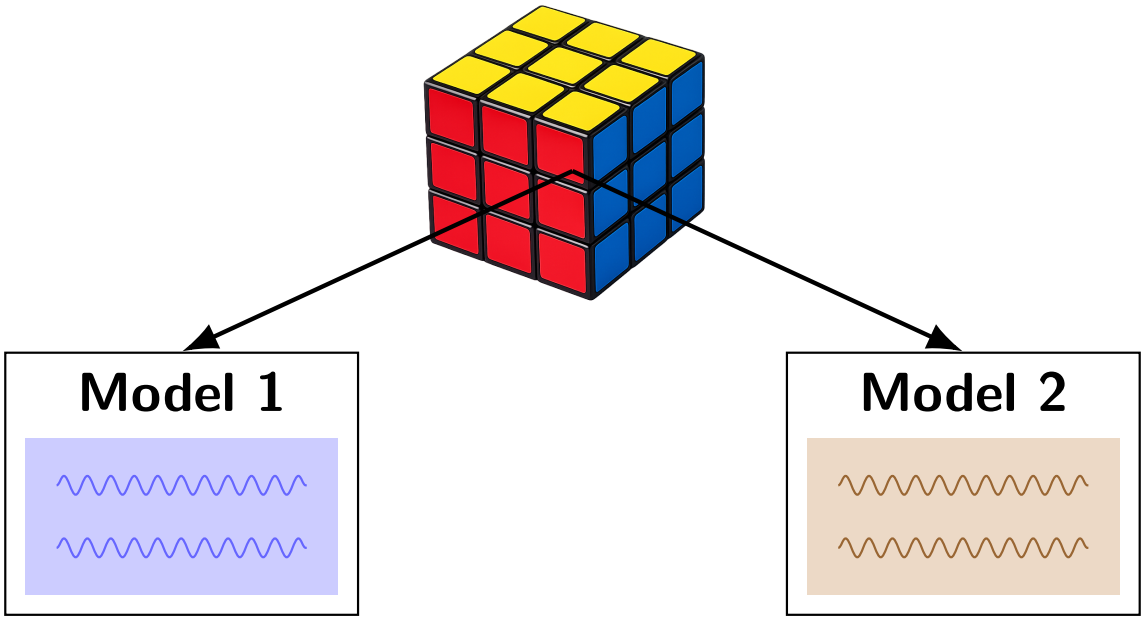


Figure: Basil Kohler (2012)

- start with a suboptimal plan
- apply an iterative algorithm that improves its quality

14. PDDL Modeling



- impact on performance and heuristic values
- compactness of representation
- axioms, conditional effects
- pros and cons of redundancy

15. Grounding PDDL Models

$$\text{PDDL} \rightarrow \langle \mathcal{V}, \mathcal{O}, s_0, \gamma \rangle$$

- planning problems are often defined using the first-order language PDDL
- most planners translate (i.e., ground) PDDL problems into the type of planning tasks you are familiar with
- How does this work?

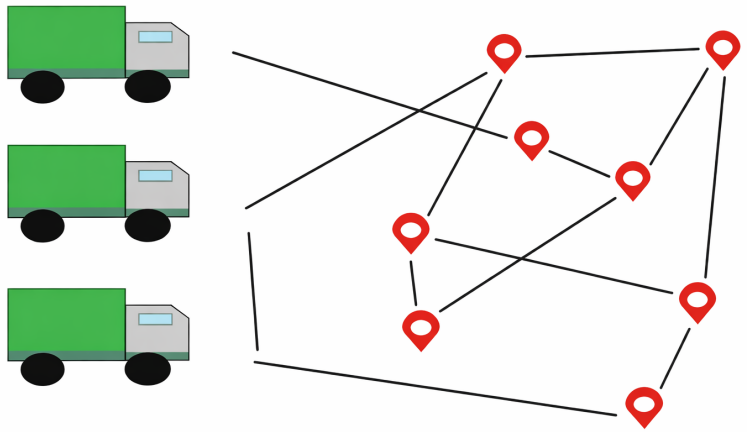
16. Logic Programs and ASP

```
topic(programming; logic; ai).

prereq(ai, programming).
prereq(ai, logic).

tired(logic).

skip(T) :- tired(T).
skip(T) :- prereq(T,P), not study(P).

study(T) :- has-book(T), not skip(T).
```

- model complex domains with rules and constraints
- reason about recursive conditions and transitive closure

17. Invariant Synthesis for Classical Planning

invariant synthesis: automatically find things that are true in every reachable state



In every reachable state:

- no two trucks are in the same position
- all trucks are in exactly one position
- there are exactly three trucks
- ...

18. Proof Calculi

$*54{\cdot}43.\quad \vdash :. \alpha, \beta \in 1 . \supset : \alpha \cap \beta = \Lambda . \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54{\cdot}26 . \supset \vdash :. \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

[*51·231] $\equiv . \iota`x \cap \iota`y = \Lambda .$

[*13·12] $\equiv . \alpha \cap \beta = \Lambda \qquad (1)$

$\vdash . (1) . *11{\cdot}11{\cdot}35 . \supset$

$\vdash :. (\exists x, y) . \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda \qquad (2)$

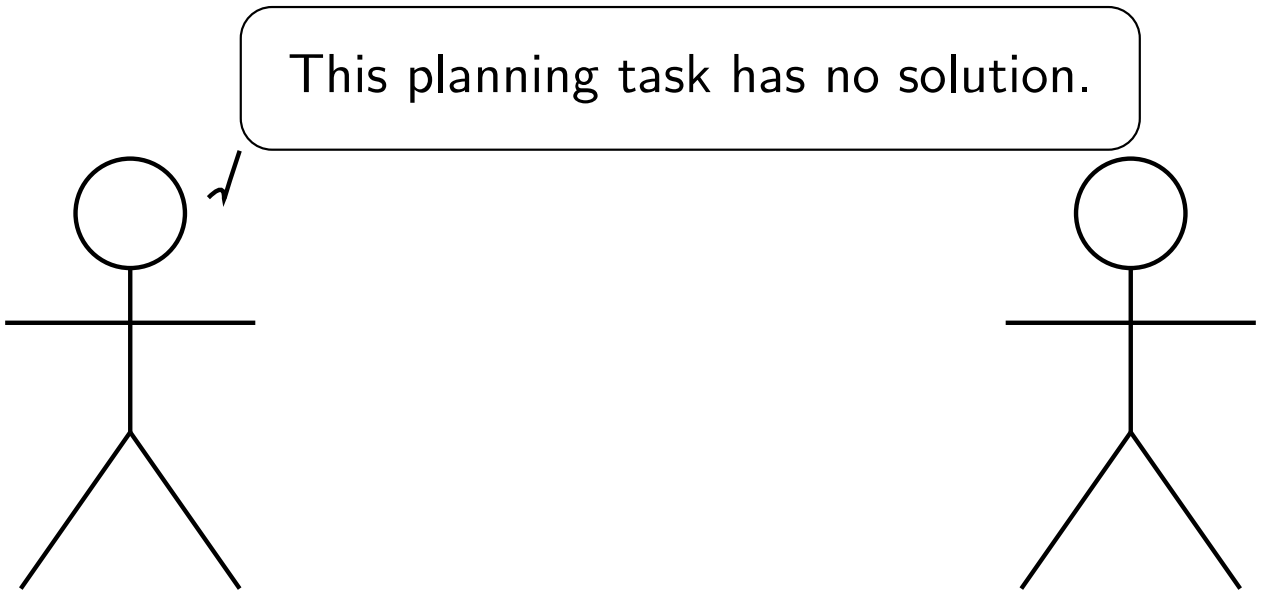
$\vdash . (2) . *11{\cdot}54 . *52{\cdot}1 . \supset \vdash . \text{Prop}$

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

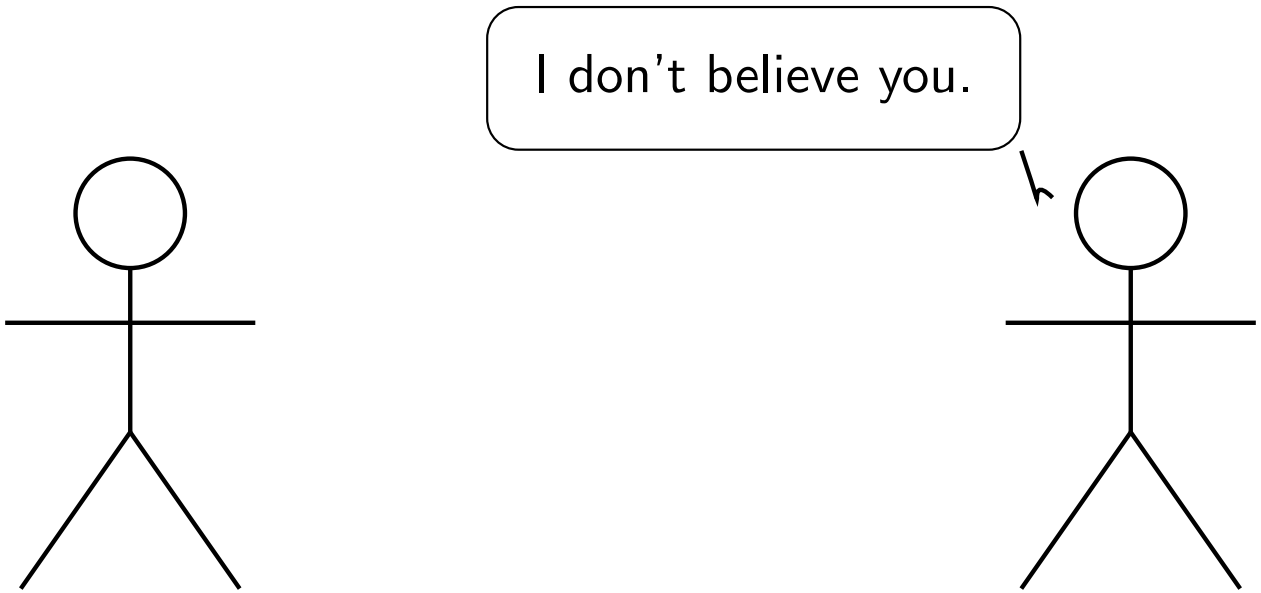
Figure: Principia Mathematica, p. 379

- What is a proof?
- Are there algorithms that can automatically prove theorems for us?

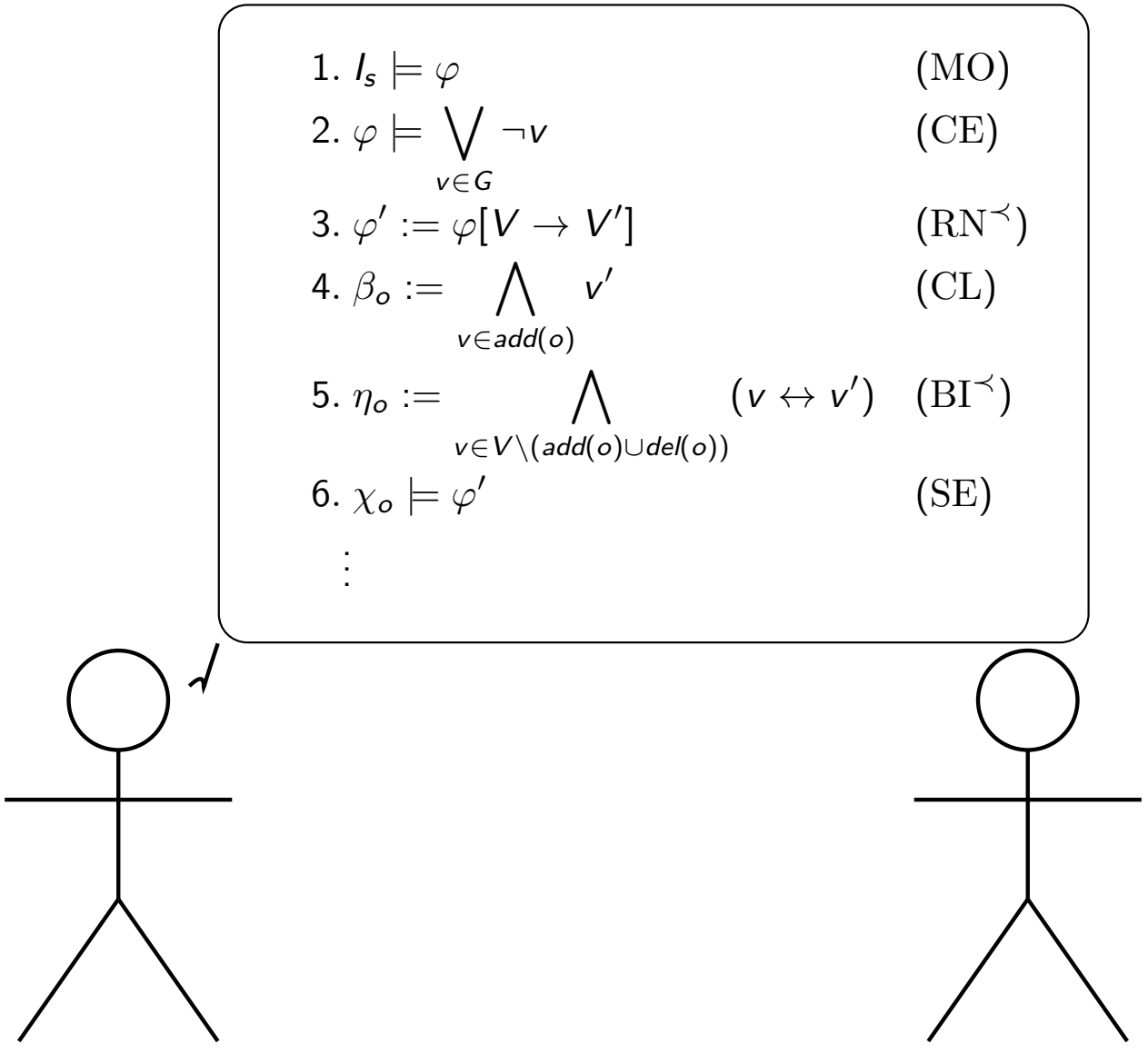
19. Unsolvability Certificates for Planning



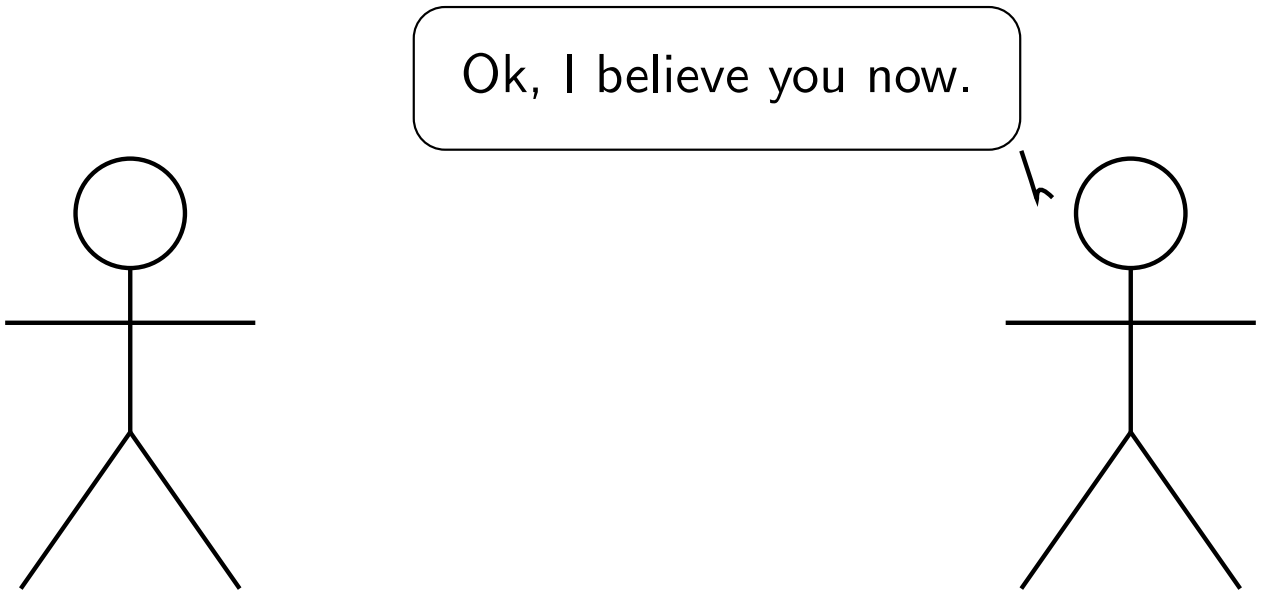
19. Unsolvability Certificates for Planning



19. Unsolvability Certificates for Planning



19. Unsolvability Certificates for Planning

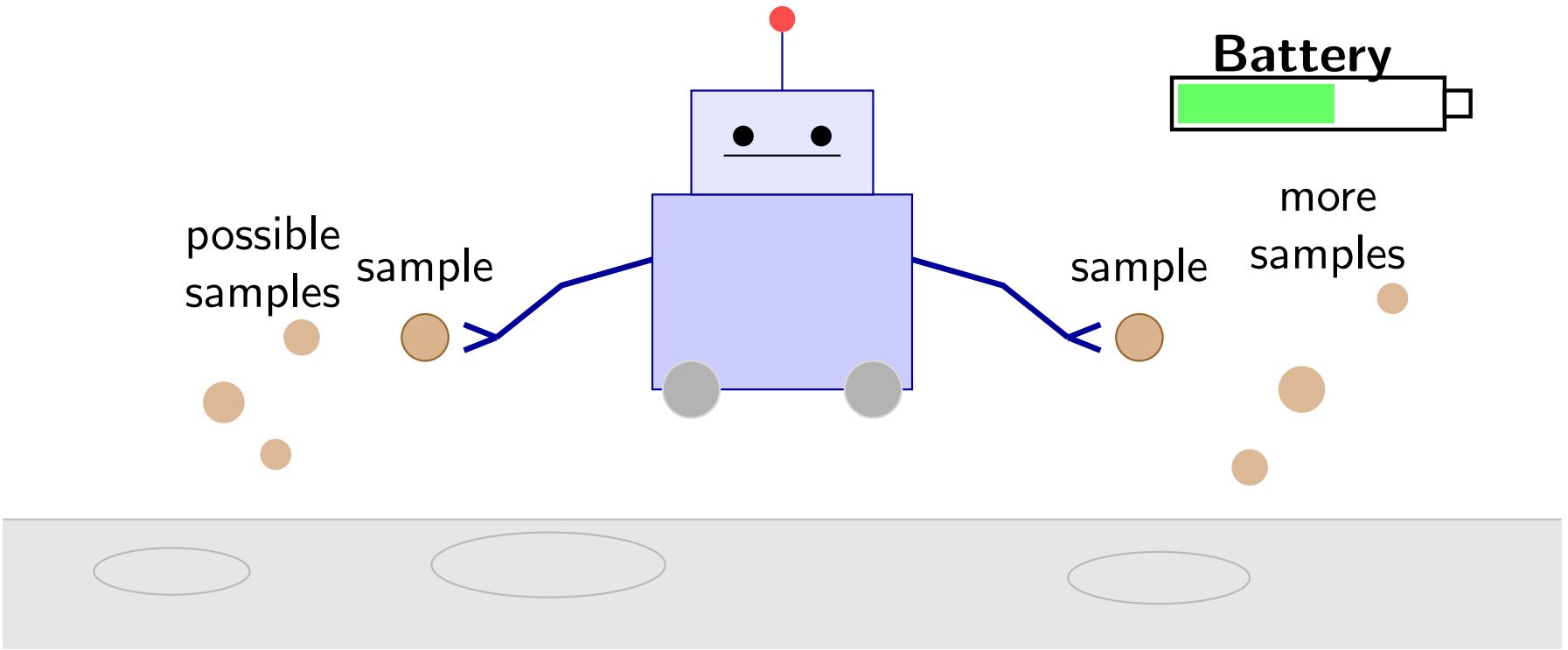


20. Unsolvability with Parity Constraint Invariants and Monovariants

Figure: The Fifteen Puzzle

- algorithms that automatically detect unsolvability using parity and monotonicity arguments

21. Oversubscription Planning



- e.g., maximize the amount of samples collected without running out of battery
- oversubscription planning = find a plan that maximizes utility given a budget

22. FOND Planning

Classical Planning

start-engine

pre: engine-off $\wedge$ has-key
eff: engine-on

FOND Planning

start-engine

pre: engine-off $\wedge$ has-key
eff: engine-on **or** key-jammed

- **f**ully **o**bservable, **n**on-**d**eterministic planning
- non-deterministic operator effects

23. Numeric Planning with MIP Compilation

drive-Basel-Freiburg

pre: $\text{in-Basel} \wedge \text{fuel} \geq 30 \wedge \text{has-passport}$
eff: $\neg\ \text{in-Basel} \wedge \text{in-Freiburg} \wedge \text{fuel} := \text{fuel} - 30$

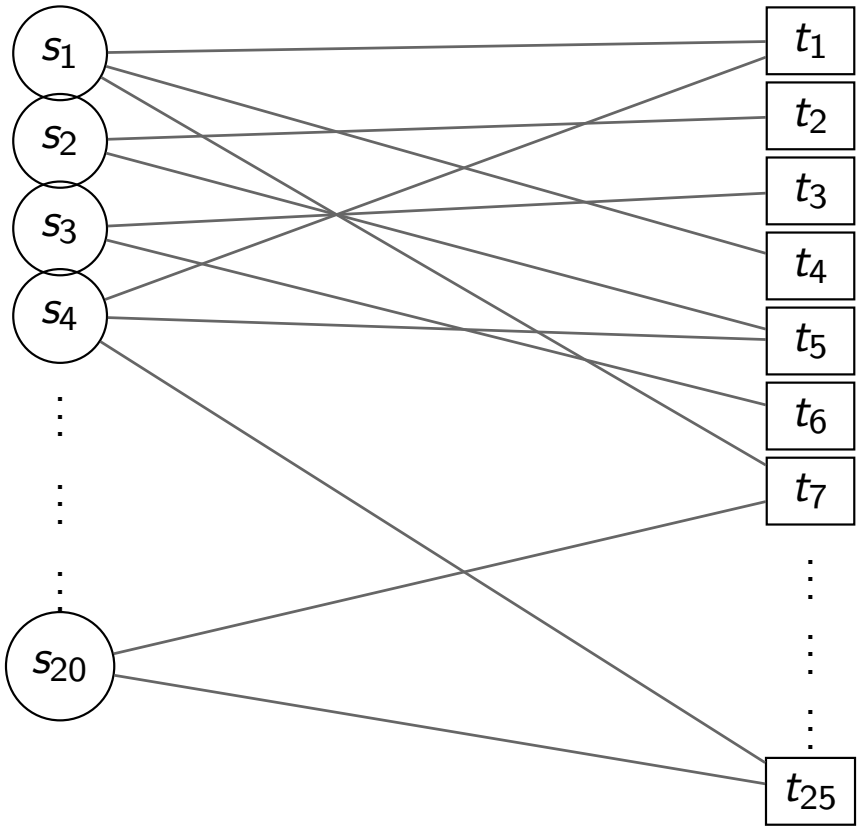
- planning with numeric variables
- compile the problem into a mixed integer program (MIP)
- solve it with a MIP solver

24. LP Solving

$$\max_{\mathbf{x}\in\mathbb{R}^d} \mathbf{a}^\top \mathbf{x} \text{ subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b}$$

- ubiquitous in AI and related fields
- How do we solve these problems?
- How do Gurobi, HiGHS, SoPLEX etc. work?

25. LP Solvers for Bipartite Graph Matching



- 20 students, $\geq$ 3 topics each: can you find a matching such that no two students get the same topic?
- IP and LP bipartite graph matching
- Complexity and comparison to standard algorithms