AI Planning: A Primer and Survey (Preliminary Report)

Dillon Z. Chen   Pulkit Verma   Siddharth Srivastava   Michael Katz   Sylvie Thiébaux

PRL@AAAI-2025
Philadelphia, PA, USA

[In this talk:]

Generalisation in AI Planning

Dillon Z. Chen   Pulkit Verma   Siddharth Srivastava   Michael Katz   Sylvie Thiébaux





PRL@AAAI-2025
Philadelphia, PA, USA

Special acknowledgements:
Sheila McIlraith and Tom Silver

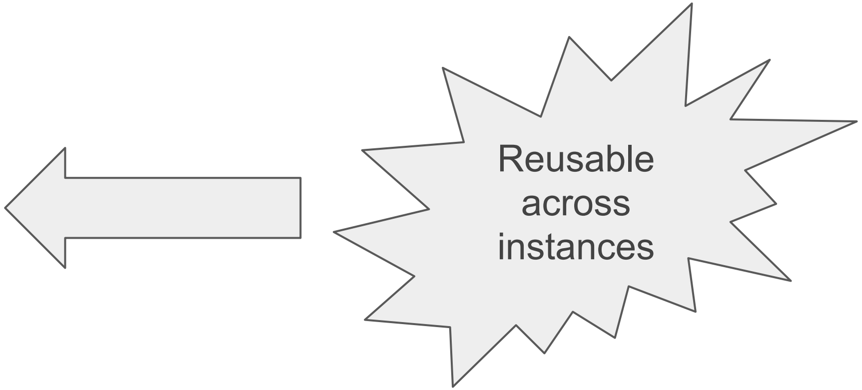
What is AI Planning?

- sequential decision making problems (SDMs) via formal, modelling languages
- Most common: PDDL [1, 2]
    - Many other languages: PPDDL, RDDL, GOLOG etc.
- Used to specify physics, preferences and objectives to agents

[1] McDermott, D.; Ghallab, M.; Howe, A. E.; Knoblock, C. A.; Ram, A.; Veloso, M. M.; Weld, D. S.; and Wilkins, D. E. 1998. *PDDL-The Planning Domain Definition Language.*
[2] Haslum, P.; Lipovetzky, N.; Magazzeni, D.; and Muise, C. 2019. *An Introduction to the Planning Domain Definition Language.*

AI Planning

- *Idea*: first-order logic for defining states and reusable actions
- An SDM can be modelled via
  - domain *D*:
    - first-order predicates (states)
    - action schema (transitions)
  - instance *I*:
    - objects
    - initial state
    - goal condition
- A solution for <*D*, *I*> is a goal achieving policy



AI Planning

- reusable action example

```
pickup(agent, object, location)

        pre: at(agent, location), at(object, location), capacity-available(agent)

        eff: at(agent, location), !at(object, location), !capacity-available(agent), holding(agent, object)
```

- can use across problems with different
  - numbers of objects
  - types of objects
  - low level action executions

AI Planning and 3 forms of Generalisation

1. Generalised Planning
    - learning to generalise across number of objects
2. Learning for Planning
    - learning to generalise across different goals
3. Learning Planning Models
    - learning to generalise across domains

(1) Generalised Planning

"Model-driven generalisation"

The Setup

- A Generalised Planning (GP) problem can be described [3, 4, 5] by a
  - domain *D (predicates, schemata)*
  - set of possibly infinitely many instances $\boldsymbol{I} = \{I_1, \ldots, I_n (, \ldots)\}$

May have arbitrarily many objects, e.g. $|I_i| = i$

- A solution for <*D, **I***> is a **single** policy that solves every instance in $\boldsymbol{I}$

[3] Hector J. Levesque: *Planning with Loops*. IJCAI 2005
[4] Siddharth Srivastava, Neil Immerman, Shlomo Zilberstein: *Learning Generalized Plans Using Abstract Counting*. AAAI 2008
[5] Sergio Jiménez Celorrio, Javier Segovia-Aguas, Anders Jonsson: *A review of generalized planning*. KER 2019

The Setup

- e.g. place all the dirty objects on the floor into the laundry basket

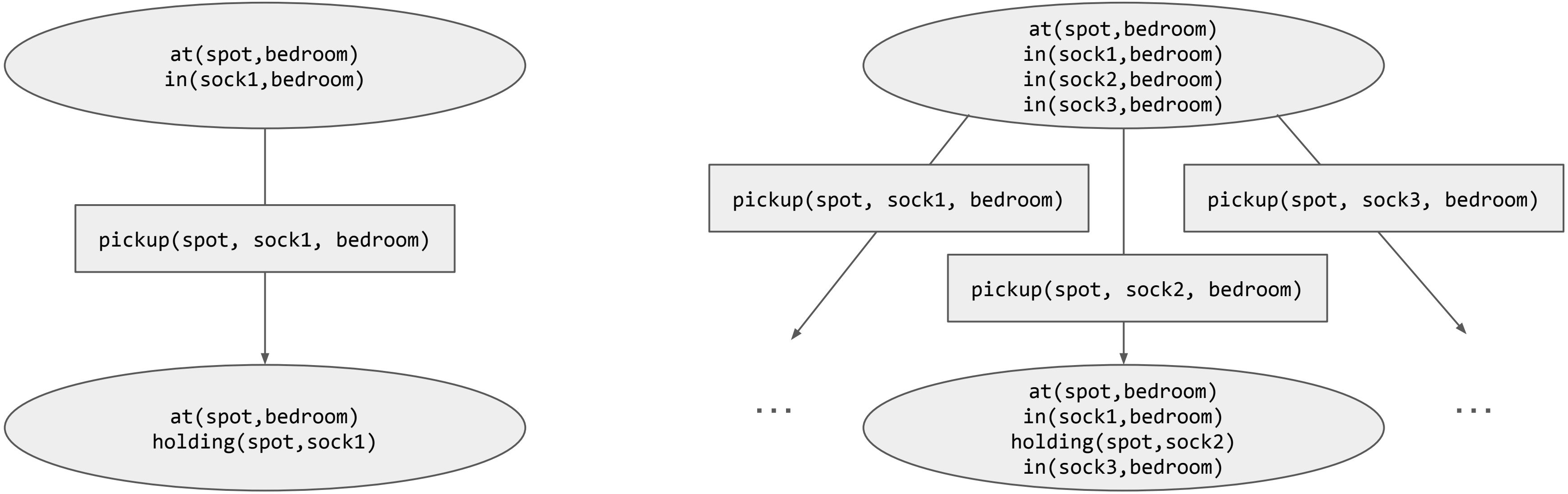
Day 1

Day 10

How to synthesise a policy across instances?

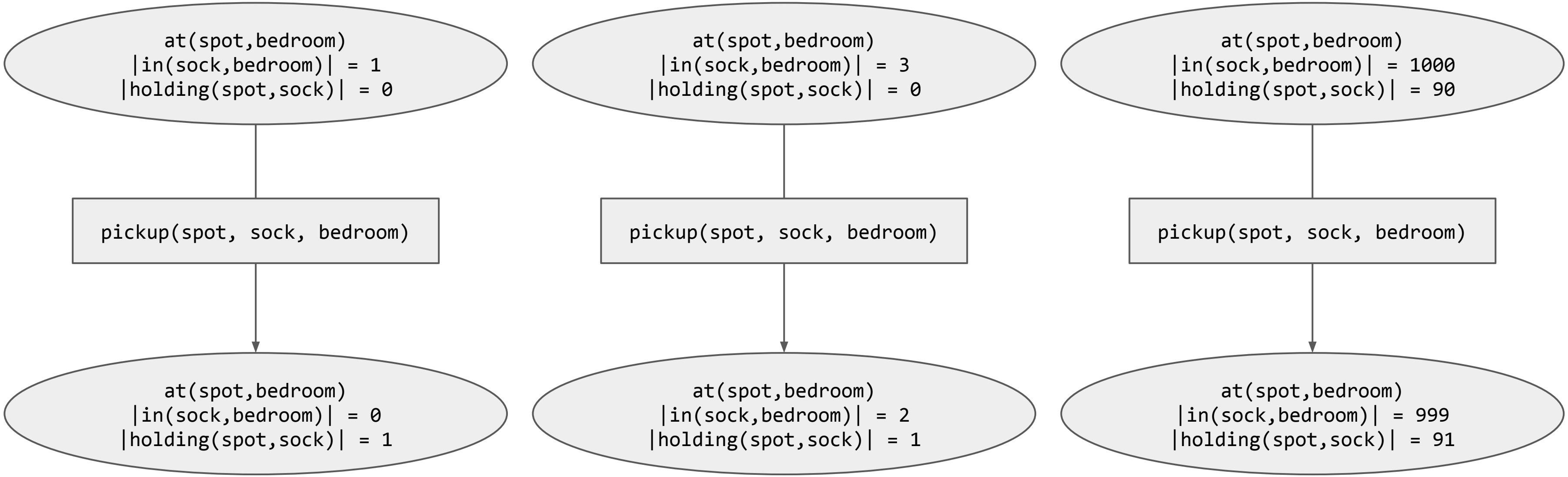
Methodologies

- many different instances = many different transitions, … what can we do?



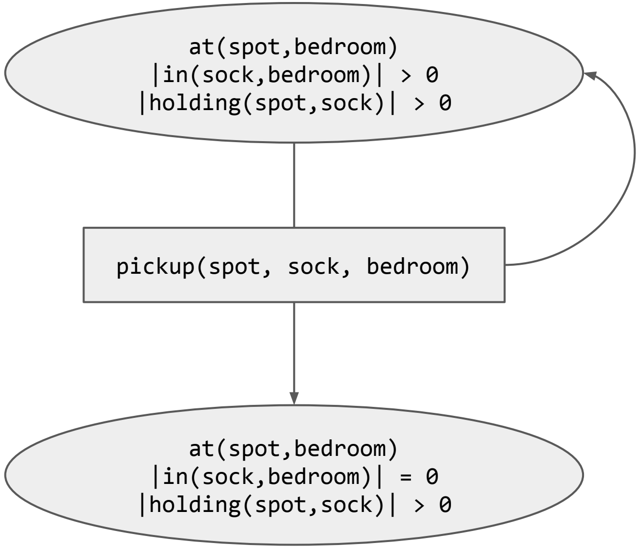
Methodologies – Abstraction is All You Need

- we can abstract equivalent objects away:



Methodologies – Abstraction is All You Need

- we can even abstract numbers away via qualitative states [5]



[6] Siddharth Srivastava, Shlomo Zilberstein, Neil Immerman, Hector Geffner: *Qualitative Numeric Planning*. AAAI 2011

Methodologies – Abstraction is All You Need

learn an abstraction of a **set** of instances into

a **single** nondeterministic planning instance

[6] Siddharth Srivastava, Shlomo Zilberstein, Neil Immerman, Hector Geffner: *Qualitative Numeric Planning*. AAAI 2011
[7] León Illanes, Sheila A. McIlraith: *Generalized Planning via Abstraction: Arbitrary Numbers of Objects*. AAAI 2019
[8] Blai Bonet, Hector Geffner: *Qualitative Numeric Planning: Reductions and Complexity*. J. Artif. Intell. Res. 69: 923-961 (2020)

Many other details left out…

- How do we learn an abstraction?

- How can we check whether a policy is sound and complete?

- What is the complexity of solving such problems?

(2) Learning for Planning

“Data-driven generalisation”

GP drawbacks – and how L4P can help

1. (implicitly/explicitly) assumes certain initial state and goal structures
    - learn from data
2. methods are “0 or 1” – either works or fails
    - statistical approaches allow for leeway
3. no single approach encapsulates PTIME domains (yet)
    - fall back on search

Conversely: models help learning

- efficiently generated training data
- additional relational, inductive biases
- more efficient search engines

Methodologies – Policies

- Learn reusable policies
  - learn (deep) action scorers
    - Examples: ASNets – Toyer et al. (AAAI-18, JAIR-20)
  - learn (deep) global value functions
    - Examples: Muninn – Ståhlberg et al. (ICAPS-22, KR-23)
  - policies as code
    - Examples: Silver et al., (AAAI-24)
  - learn symbolic rules
    - Examples: Khardon (AIJ-1999); Martín and Geffner (KR-00, JAIR-04); Gretton and Thiébaux (UAI-04); Yoon et al. (JMLR-08); Bonet et al. (AAAI-19); Francès et al. (AAAI-21)

Methodologies – Heuristics

- Learn heuristics for heuristic search
    - Lessons from deep learning:
        - Rich Sutton's *Bitter Lesson (2019)*: scale from learning and search
        - Large Language Models: scale from train-time and test-time compute
    - Search can handle mistakes and is complete
    - Learn from supervision
        - Examples: Yoon et al. (JMLR-08); STRIPS-HGN – Shen et al. (ICAPS-20); GHNs – Karia and Srivastava (AAAI-21); GOOSE – Chen et al. (ICAPS-24)
    - Learn with Reinforcement Learning
        - Examples: Micheli and Valentini (AAAI-2021); Gehring et al. (ICAPS-22);

Methodologies – Meta Knowledge

- Learn meta knowledge
  - learn problem transformations: can simplify instances
    - Examples: Gnad et al. (AAAI-19); Silver et al. (AAAI-21)
  - learn planner portfolios: select planner that performs best for each instance
    - Examples: Sievers et al. (AAAI-19); Ma et al. (AAAI-20); Ferber and Seipp (AAAI-22)

(3) Learning Planning Models

Why learn Planning Models?

- Improves sample efficiency, learning speed, and safety of agents.
- Reusable across instances with similar dynamics (domain-general vs. task-specific models).
- Automates the requirement to hand-code the structure needed for planning.

Methodologies – Abstract Traces

- From Passive State-Action Traces:
  - Learn models from observed state-action transitions.
  - Learn preconditions and effects by analyzing state transitions.
  - Examples: ARMS – Yang et al. (AIJ 2007), LOCM - Cresswell et al. (ICAPS 2009), LOUGA - Kučera and Barták (KMAIS 2018), FAMA - Aineto et al. (AIJ 2019), etc.

Methodologies – Image Traces

- From Passive Image Traces:
  - Learn models from image-based state representations.
  - Learn preconditions and effects by converting image transitions to symbolic transitions.
  - Examples: Asai et al. (2022); Xi, Gould, and Thiebaux (2024); Campari et al. (2022).

Methodologies – Directed Exploration

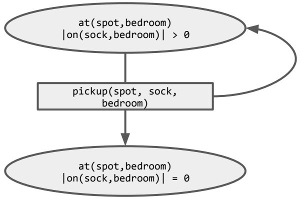
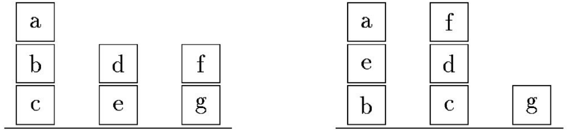
- From Directed Exploration:
  - Actively explore the environment to generate the observations.

  - Directed Search: What action should I execute more to acquire more samples?
    - Examples: EXPO – Gil (ICML'93); IRALe – Rodrigues et al. (ILP'11); ILM - Ng and Petrick (IJCAI'19); OLAM - Lamanna et al. (IJCAI'21); GLIB – Chitnis et al. (AAAI'21)

  - Active Querying: Ask the agent what would happen if they had to execute a plan from a given state.
    - Examples: AIA - Verma et al. (AAAI'21); DAAISy - Nayyar et al. (AAAI'22); CLaP - Karia et al. (ICAPS'24)

AI Planning: A Primer and Survey (Preliminary Report)

Dillon Z. Chen   Pulkit Verma   Siddharth Srivastava   Michael Katz   Sylvie Thiébaux

<table>
<tr><th>Generalisation Subfield</th><th>Form of Generalisation</th><th>Methodologies</th></tr>
<tr><td>Generalised Planning
(model-driven generalisation)</td><td>across number of objects
</td><td>Abstraction is All You Need

</td></tr>
<tr><td>Learning for Planning
(data-driven generalisation)</td><td>across different goals

</td><td>Learn knowledge as
policies
heuristics
transformations/portfolios</td></tr>
<tr><td>Learning Planning Models</td><td>across different domains

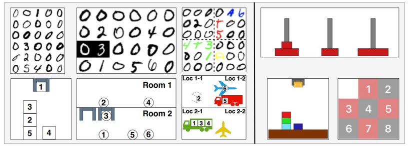



image by Xi, Gould, and Thiebaux (ICAPS-24)</td><td>Learn models from
abstract traces (passive)
sensory data (passive)
directory exploration (active)</td></tr>
</table>