Course "Automated Planning: Theory and Practice" Chapter 09: Goal Count - A simple Domain Independent Heuristic Function

Teacher:	Marco Roveri - `marco.roveri@unitn.it`
M.S. Course:	Artificial Intelligence Systems (LM)
A.A.:	2025-2026
Where:	DISI, University of Trento
URL:	`https://shorturl.at/A81hf`

Last updated: Wednesday 15th October, 2025

Terms of Use and Copyright

The material (text, figures) in these slides is authored by Jonas Kvarnström and Marco Roveri.

Heuristics given Structured States

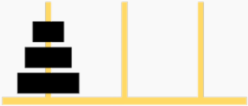
- In planning, we aften want domain-independent heuristics
  - Should work for any planning domain - how?
- Take advantage of structured high-level representation!

Plain state transition system

- We are in state 572242104485172012
- The goal is to be in one of the 10^{47} states in $S_g = \{s[482293], s[482294], \ldots\}$
- Should we try action $A297295283291$ leading to state 572342104485172016?
- Or may be action $A297295283292$ leading to state 572342104485175202?

Classical Representation

- We are in a state where disk 1 is on top of disk 2
- The goal is for all disks to be on peg C
- Should we try `(take B)`, leading to a state where we are holding disk 1?
- ...

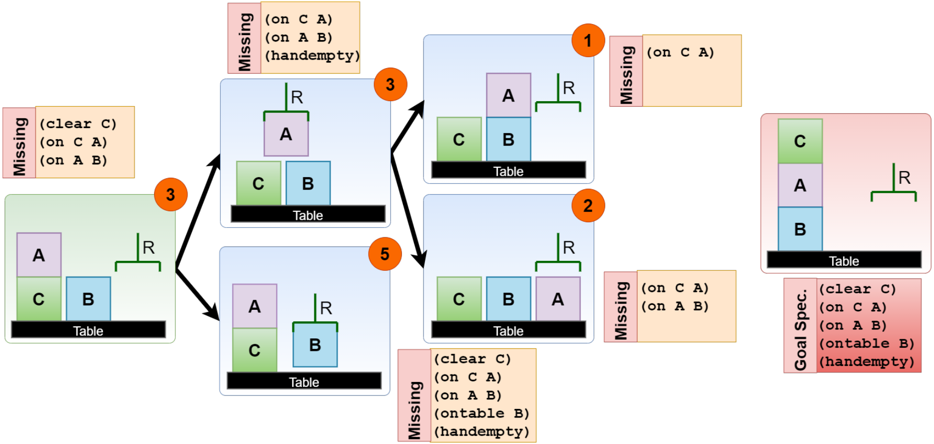


An Intuitive Heuristic

- Assumptions
  - *Forward state space planning*: Nodes n are states s
  - Classical expressivity: goal is a set of *ground literals*
    `{ (on A B), (not (handempty)) }`
- An intuitive idea for $h(s)$:
  - Try to estimate the number of actions required to reach the goal from s
    - Should be related to how many goal facts are not yet achieved in s
  - Let $h(s)$ = number of goal literals that are not achieved in s
    - $h(s) = |(g^{+} \setminus s) \cup (g^{-} \setminus s)|$
    - Not the expected cost to achieve those goals!
- An associated search strategy:
  - Let's use Greedy Best First Search!

Counting Remaining Goals

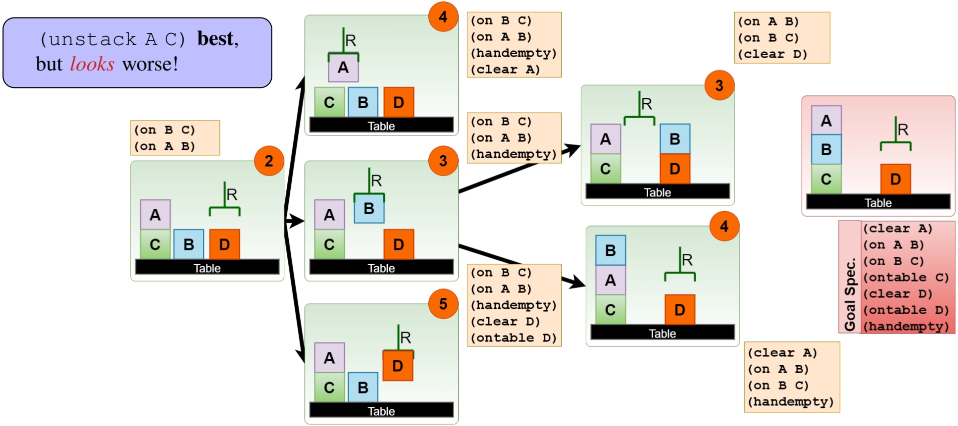
- Count the number of "missing" goal literals!



Optimal plan: (unstack A C) (stack A B) (pickup C)(stack C A)

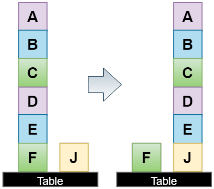
Counting Remaining Goals (cont.)

- A perfect solution? No!
  - We must often unachieve individual goal literals to get closer to the goal!

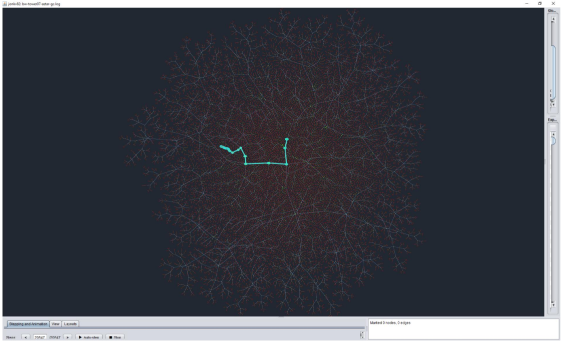


Optimal plan: (unstack A C) (putdown A) (pickup B) (stack B C) (pickup A) (stack A B)

Example: Block World - A* search, based on goal count!



18 actions in π
States: 6463 calculated, 3222 visited
(With Dijkastra 43150/33436 - improved but we can do better!)

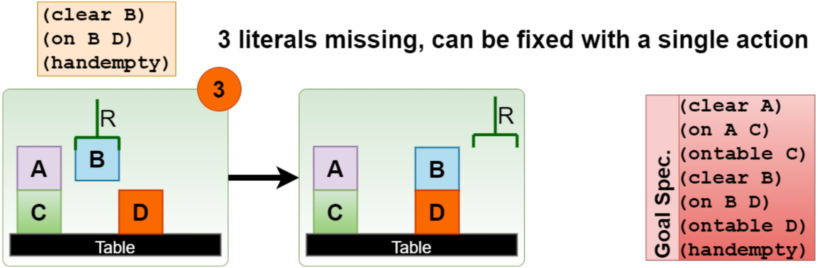


- $h(s_0) = 1$: Only one "missing" literal
- For a long time, all useful successors appear to increase remaining cost
  - Removing a block that must be moved
- And many useless successors appear to decrease remaining cost
  - Building towers that will need to be torn down

Not very informative

Counting Remaining Goals (cont.)

- Admissible? No!
  - (Doesn't matter in our chosen search strategy!)



- Can we make it admissible?
  - Yes: Divide by the maximum number of facts modified by any action!

Counting Remaining Goals: Analysis

- What we see for this example...
  - Not very much: All heuristics have weaknesses!

Even the best planners will make "strange" choices, visit tens, hundreds or even thousands of "unproductive" nodes for every action in the final plan!

The heuristic should make sure we don't need to visit millions, billions or even trillions of "unproductive" nodes for every action in the final plan!

- But a thorough empirical analysis would tell us:
  - This heuristic is far from being sufficient!

Example Statistics: Planning competition 2011

- Elevators domain, problem 1
  - A^* with goal counter heuristics
    - States: 108,922,864 generated, ... gave up!
  - LAMA 2011 planner, good heuristics, other strategy:
    - Solution: 79 steps, 369 cost
    - States: 13,236 generated, 425 evaluated/expanded
- Elevators domain, problem 5
  - LAMA 2011 planner
    - Solution: 112 steps, 523 cost
    - States: 41,811 generated, 1,317 evaluated/expanded
- Elevators domain, problem 20
  - LAMA 2011 planner
    - Solution: 354 steps, 2182 cost
    - States: 1,364,657 generated, 14,984 evaluated/expanded

http://www.plg.inf.uc3m.es/ipc2011-deterministic/
https://www.icaps-conference.org/competitions/

- Important insight
  - Even a state-of-the-art planner can't go *directly* to a goal state!
  - Generates *many* more states than those actually on the path to the goal...

References I

[1] Hector Geffner and Blai Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2013. ISBN 9781608459698. doi: 10.2200/S00513ED1V01Y201306AIM022. URL `https://doi.org/10.2200/S00513ED1V01Y201306AIM022`.

[2] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated planning - theory and practice*. Elsevier, 2004. ISBN 978-1-55860-856-6.

[3] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016. ISBN 978-1-107-03727-4. URL `http://www.cambridge.org/de/academic/subjects/computer-science/artificial-intelligence-and-natural-language-processing/automated-planning-and-acting?format=HB`.

[4] Patrik Haslum, Nir Lipovetzky, Daniele Magazzeni, and Christian Muise. *An Introduction to the Planning Domain Definition Language*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2019. doi: 10.2200/S00900ED2V01Y201902AIM042. URL `https://doi.org/10.2200/S00900ED2V01Y201902AIM042`.