Linköping University

Automated Planning

Goal Count:
A Simple Domain-Independent Heuristic Function

Jonas Kvarnström
Department of Computer and Information Science
Linköping University

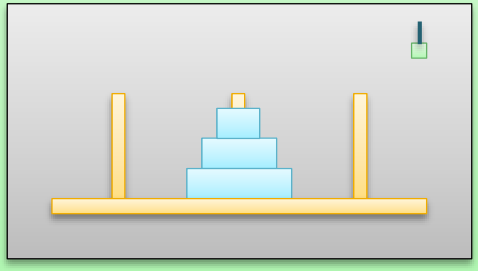
jonas.kvarnstrom@liu.se – 2019

Heuristics given Structured States

- In planning, we often 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 572,342,104,485,172,012
  - The goal is to be in one of the 10^47 states in S_g={ s[482,293], s[482,294], … }
  - Should we try action A297,295,283,291 leading to state 572,342,104,485,172,016?
  - Or maybe action A297,295,283,292 leading to state 572,342,104,485,175,201?

- **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* $\{on(A,B), \neg handempty\}$
    - PDDL: (and (on A B) (not (handempty)))

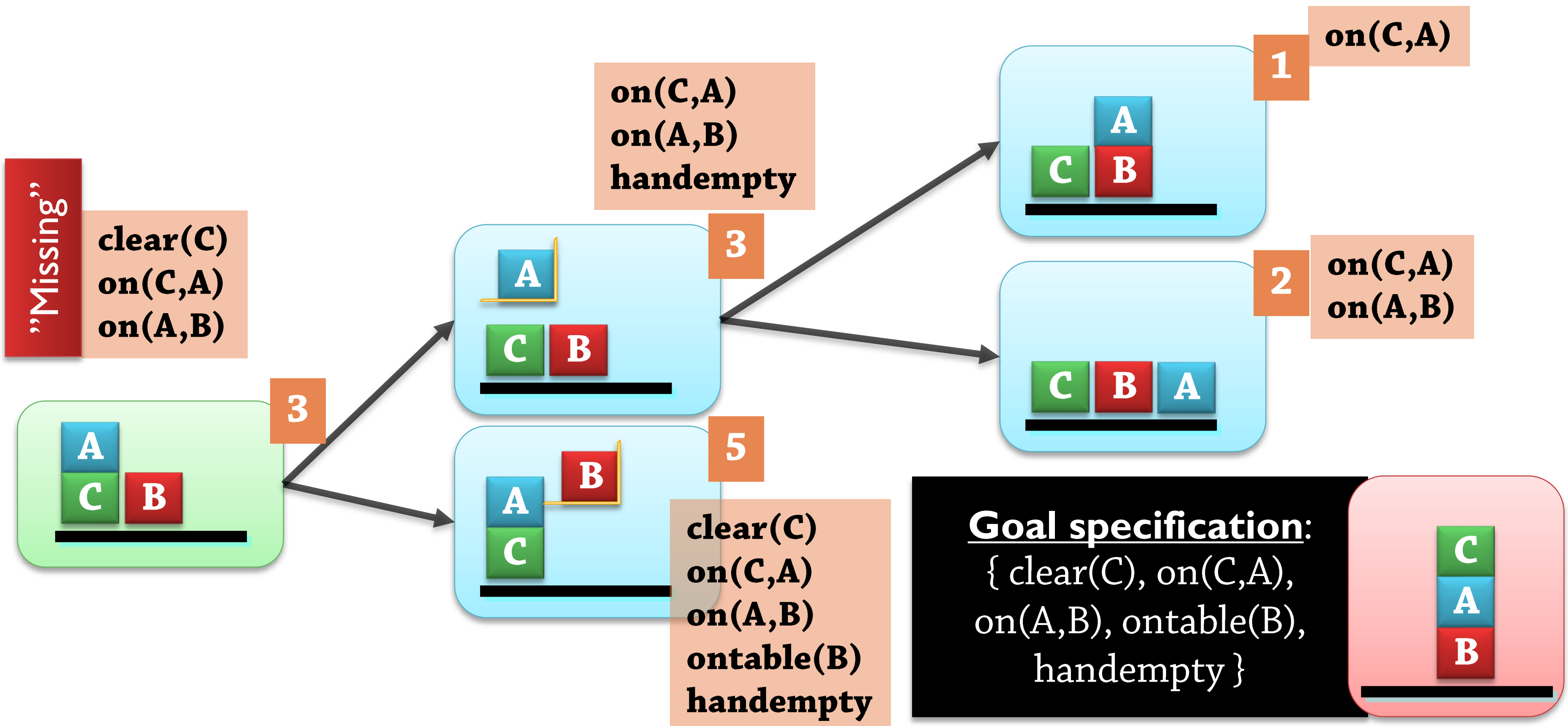
- An **intuitive** idea for $h(s)$:
  - Try to estimate the number of **actions** required to **reach the goal** from s
    - Should be ***related to***
      how many **goal facts** are not yet achieved in s
  - Let $h(s)$ = **number of goal literals** that are **not achieved** in s
    - $h(s) = |(g^{+} - s) \cup (g^{-} \cap s)|$
    - (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:
unstack(A,C)
stack(A,B)
pickup(C)
stack(C,A)



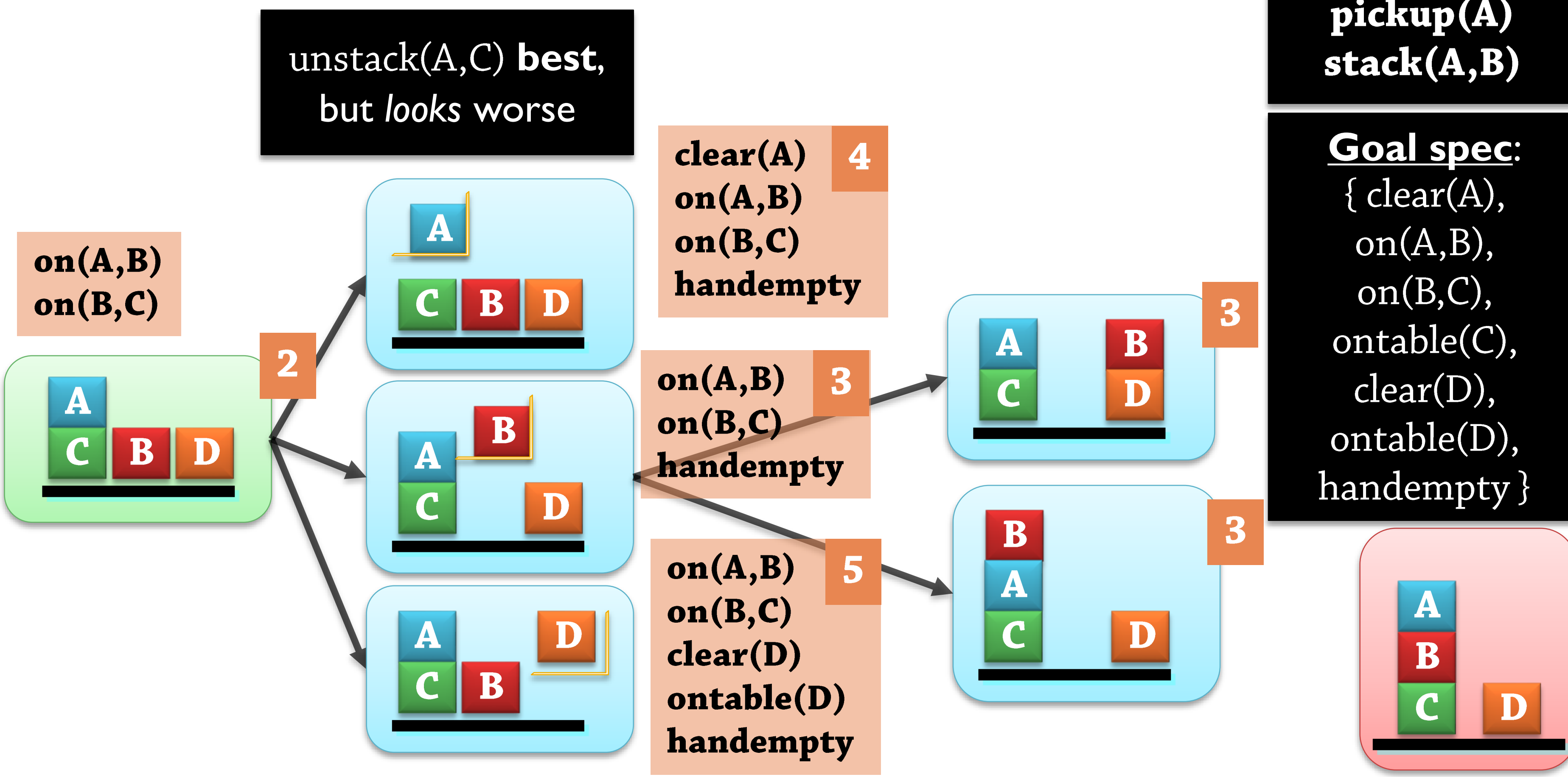
Goal specification:
{ clear(C), on(C,A), on(A,B), ontable(B), handempty }

Counting Remaining Goals (2)

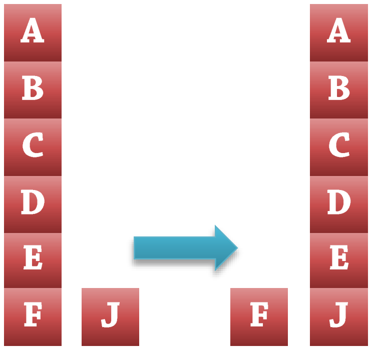
jonkv@ida

- A **perfect** solution? No!
  - We must often "**unachieve**" individual goal literals to get closer to a goal state!

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



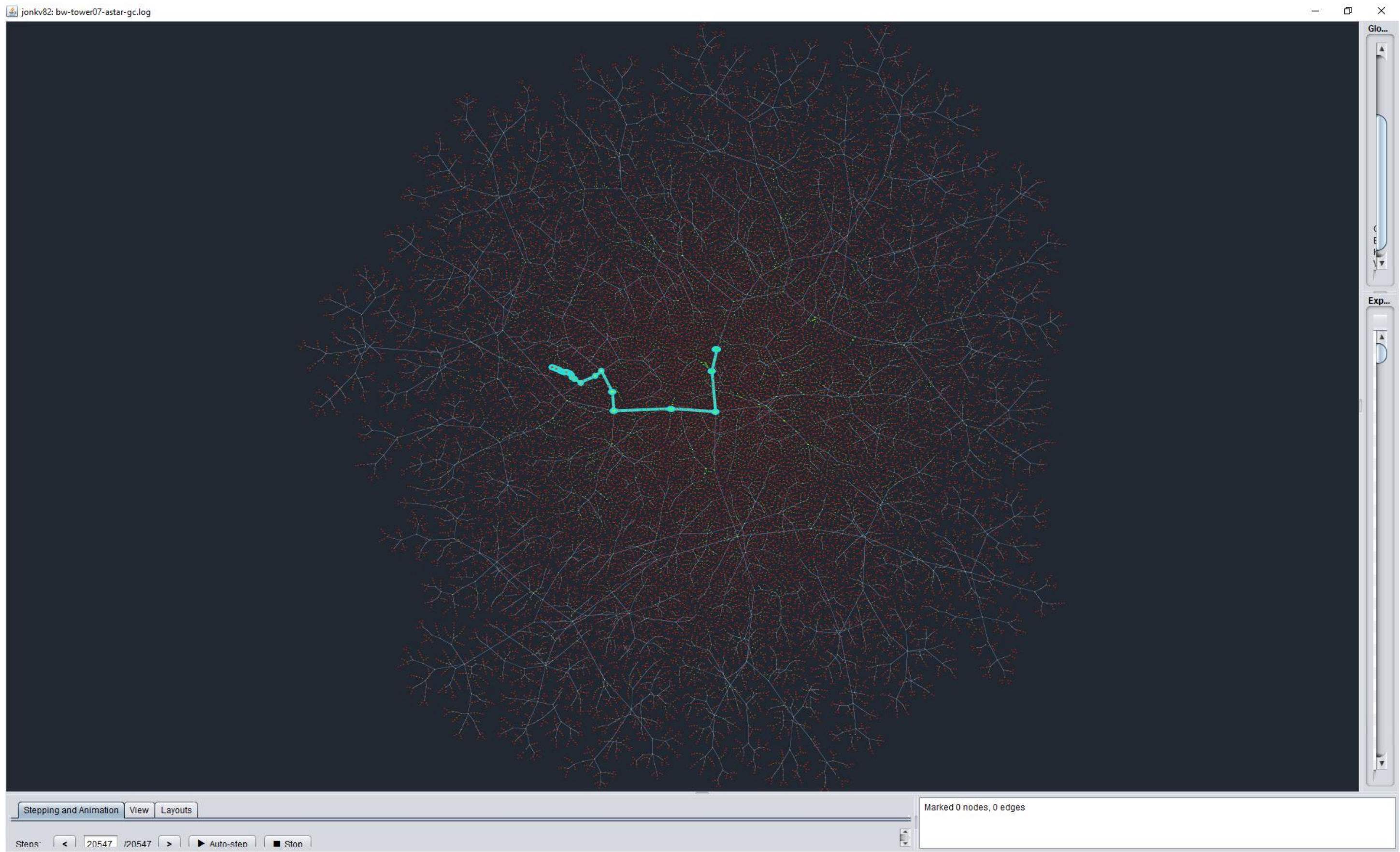
bw-tower07-astar-gc: Only 7 blocks, A* search, based on goal count



18 actions in π

States:
6463 calculated,
3222 visited

(With Dijkstra, 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

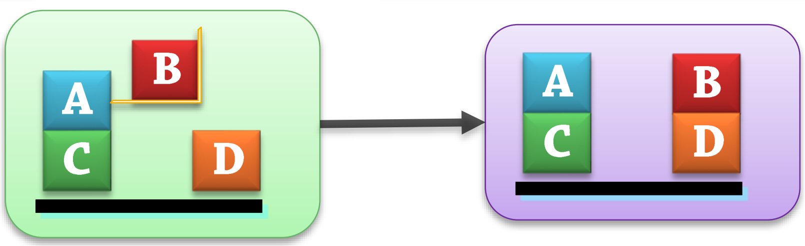
Not very *informative!*

Counting Remaining Goals (3)

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

Missing:
clear(B)
on(B,D)
handempty

3 literals missing, can be fixed with a single action



Goal spec:
{ clear(A),
on(A,C),
ontable(C),
clear(B),
on(B,D),
ontable(D),
handempty }

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

Counting Remaining Goals (4): Analysis

- What we see from 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 sufficient!

Example Statistics

- Planning Competition 2011: Elevators domain, problem 1
  - A* with goal count heuristics
    - States: 108'922'864 generated, gave up
  - LAMA 2011 planner, good heuristics, other strategy:
    - Solution: 79 steps, 369 cost
    - States: 13236 generated, 425 evaluated/expanded
- Elevators, problem 5
  - LAMA 2011 planner:
    - Solution: 112 steps, 523 cost
    - States: 41811 generated, 1317 evaluated/expanded
- Elevators, problem 20
  - LAMA 2011 planner:
    - Solution: 354 steps, 2182 cost
    - States: 1'364'657 generated, 14985 evaluated/expand

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…