# Learning Generalized Unsolvability Heuristics for Classical Planning

Simon Ståhlberg[1], Guillem Francès[2] and Jendrik Seipp[1]

[1]Linköping University, Sweden

[2]Universitat Pompeu Fabra, Spain

IJCAI 2021

# Introduction

- Recently, interest in unsolvability heuristics for classical planning
- Tailored heuristics are successful for individual problems
- Learn generalized formulas for detecting unsolvable states
- Many unsolvable states are detectable in polynomial time
- We use Boolean features based on description logic to describe such states in a concise and generalized way
- Three different methods for learning such formulas

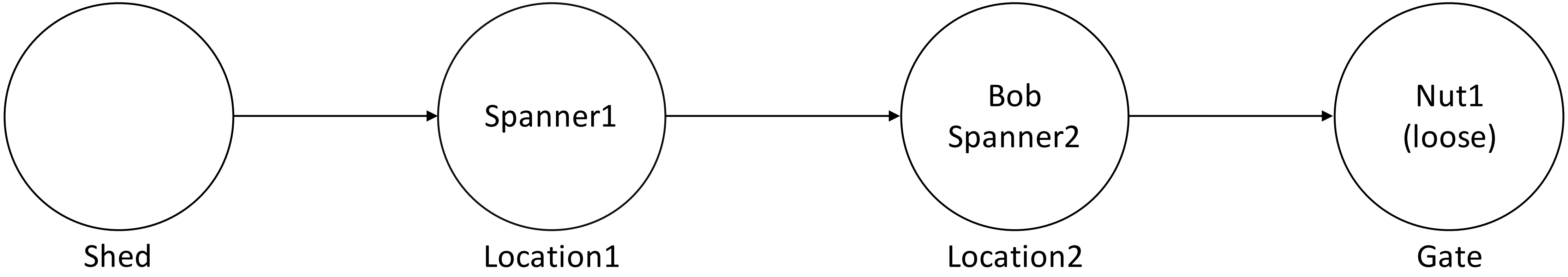
# Description Logic

- Use grammar defined by Description Logic (DL) in classical planning
- We are interested in the grammar, not inference
- DL separates schema (TBox) and data (ABox), similar to planning
  - TBox corresponds to the domain
    - Unary predicates in the domain are concepts
    - Binary predicates in the domain are roles
  - ABox corresponds to the problem
    - The universe is the set of objects in the problem

# Spanner

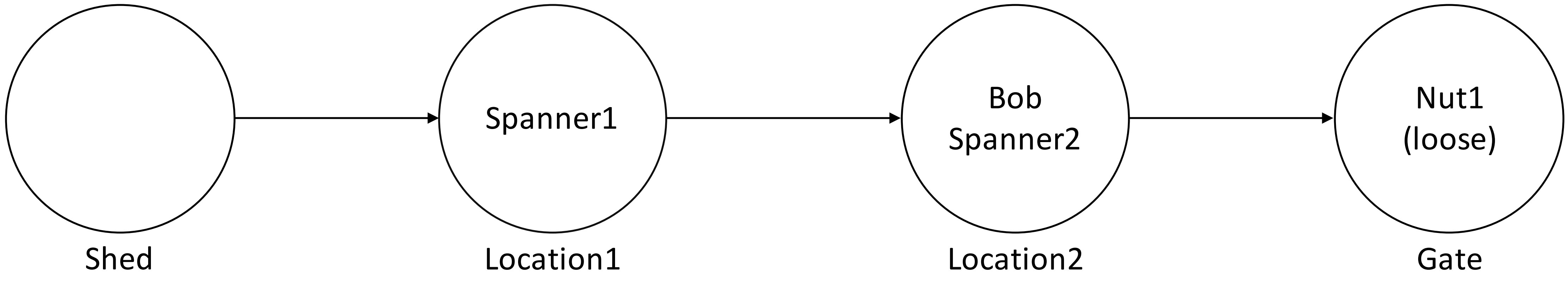
- Spanner as a running example
  - A single agent, Bob, must *tighten* all *loose nuts* at the gate with *spanners*
  - Bob lives in a shed and must walk to the gate
  - Along the one-way path there are *usable* spanners that he can *pick up*
  - All spanners becomes unusable after use
- Once Bob leaves a location, he cannot return
- Bob is in an unsolvable state if he leaves behind too many spanners
- Predicates: location/1, locatable/1, man/1, nut/1, spanner/1, at/2, carrying/2, useable/1, link/2, tightened/1, loose/1
- Actions: walk, pickup_spanner, tighten_nut

# Example




| *man* | *loose* | *usable* | *link* | *at* | *link*$^+$ |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 2 |
| {bob} | {nut1} | {spanner1, spanner2} | {(shed, location1), (location1, location2), (location2, gate)} | {(bob, location2), (spanner1, location1), (spanner2, location2), (nut1, gate)} | {(shed, location1), (shed, location2), (shed, gate), (location1, location2), (location1, gate), (location2, gate)} |

# Example




| $\exists at^{-1}.man$ | $\exists link^{+}.(\exists at^{-1}.man)$ | $\exists at.\left(\exists link^{+}.(\exists at^{-1}.man)\right)$ |
|---|---|---|
| 4 | 7 | 9 |
| {location2} | {location1, shed} | {spanner1} |

# Features

- Numerical features are composed by taking the cardinality, e.g.
  - $|usable|$
  - $|loose \sqcup \exists at. \left(\exists link^{+}.(\exists at^{-1}.man)\right)|$
- Boolean features are composed by comparing numerical features, e.g.
  - $|loose \sqcup \exists at. \left(\exists link^{+}.(\exists at^{-1}.man)\right)| > |usable|$
  - $|\exists at\_person.(\exists at\_car^{-1}.\top)| = 0$
  - Also, greater than zero

# Spanner

$$|loose \sqcup \exists at.\left(\exists link^{+}.(\exists at^{-1}.man)\right)| > |usable|$$

- The formula generalizes to the class of problems that can be produced by the generator
- The formula exploits certain properties of the generator:
  - The traversal graph from Bob's shed and the gate is a path graph
  - All spanners are initially *usable*
  - There is only one agent
- When Bob tightens a nut, both left- and righthand side decrements

# Pipeline

- **State labeling**
  - Input: a PDDL domain and a collection of PDDL problems
  - Output: a set of states labeled as solvable or unsolvable
- **Feature generation**
  - Input: a PDDL domain and the output from the previous step
  - Output: a set of Boolean features and their evaluations on all states (and their label)
- **Formula construction**
  - Input: the output from the previous step
  - Output: a DNF of Boolean features, we consider the three criteria
    - Perfect: holds if and only if state is an unsolvable state (not always possible)
    - Safe: holds if state is an unsolvable state
    - DecisionTree: maximizes F1 score

# Results

| | $h^{\mathrm{CG}}$ | $h^{\mathrm{CEA}}$ | $h^{\mathrm{SEQ}}$ | $h^1$ | $h^2$ | $h^3$ | $k$-consistency k=1 | k=2 | k=3 | $\mathbb{T}$-PERFECT prec | rec | C | L | $k$ | t | $\mathbb{T}$-SAFE prec | rec | C | L | $k$ | t | DECISIONTREE prec | rec | C | L | $k$ | t |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Barman | 0.00 | 0.88 | 0.39 | 0.88 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | – | – | – | – | – | – | 0.97 | 0.36 | 11 | 18 | 9 | 5h | 0.97 | 0.99 | 28 | 56 | 8 | 6s |
| Childsnack | 0.58 | 0.58 | 0.09 | 0.58 | 0.94 | 1.00 | 0.00 | 0.27 | 0.27 | – | – | – | – | – | – | 1.00 | 1.00 | 7 | 9 | 11 | 1h | 0.91 | 0.98 | 16 | 32 | 11 | 10s |
| Hiking | 0.00 | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 1.00 | 1 | 1 | 8 | 27m | 1.00 | 1.00 | 1 | 1 | 8 | 1m | 1.00 | 1.00 | 1 | 1 | 8 | 1s |
| Nomystery | 0.00 | 0.53 | 0.00 | 0.31 | 0.91 | 1.00 | 0.00 | 0.00 | 0.83 | – | – | – | – | – | – | 0.87 | 0.12 | 22 | 39 | 17 | 5h | 0.65 | 0.92 | 1 | 1 | 12 | 1s |
| Spanner | 0.05 | 0.05 | 0.00 | 0.05 | 0.13 | 0.31 | 0.00 | 0.00 | 0.01 | 1.00 | 1.00 | 1 | 1 | 13 | 5m | 1.00 | 1.00 | 1 | 1 | 13 | 4m | 1.00 | 1.00 | 1 | 1 | 13 | 1s |
| Woodworking | 1.00 | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.98 | 1.00 | 1 | 1 | 8 | 50s | 0.98 | 1.00 | 1 | 1 | 8 | 2m | 0.98 | 1.00 | 1 | 1 | 9 | 6s |

# Thanks for listening!