Learning Sketches for Decomposing Planning Problems into Subproblems of Bounded Width

Dominik Drexler[1]
Jendrik Seipp[1]
Hector Geffner[2,1]

[1]Linköping University
[2] ICREA & Universitat Pompeu Fabra

Motivation

Two important question in Planning (and RL) are:

1. What is a **good** language for representing the subgoal structure? → Policy sketches
2. How to **learn** the subgoal structure for family of tasks? → In this paper

Example: Width-1 Sketch for Delivery

- **Domain-general features:**
  - H: holding a package?
  - n: number of undelivered packages
- **Sketch rules:**
  $\{\neg H\} \mapsto \{H\}$: pick undelivered package
  $\{H, n > 0\} \mapsto \{\neg H, n\downarrow\}$: decrease # undelivered packages

Learning Width-k Sketches

- **Given:**
  - Training instances $\mathcal{P} = \{P_i\}_{i=1}^{n}$
  - Feature pool $\mathcal{F}$, automatically constructed from $\mathcal{P}$
  - Bound on sketch width k, number of rules m
- **Find:** sketch R_Φ that consists of m rules over features $\Phi \subseteq \mathcal{F}$
  - Sketch is **simple**: $\min_{\Phi \in 2^{\mathcal{F}}} \sum_{f \in \Phi} \text{complexity}(f)$
  - Sketch **terminates**: R_Φ is acyclic in each P_i
  - Each subproblem is **easy**: each $P[s, G_{R_\Phi}(s)]$ has width $\leq k$
- Implementation as answer set program in Clingo

Conclusion

- Learned sketches can be used to solve whole domains in polynomial time where domain-independent planners fail
- Generalization **tested** empirically and **proven** theoretically

First general method for learning
how to **decompose** planning problems
into **subproblems**
with a **polynomial complexity**
that is controlled with a parameter

Take a picture to
download the full paper