Verifying Planning Sketches in CTL_f^*

Anneline Daggelinckx

Brian Logan, Natasha Alechina

Utrecht University

March 2023

Sketch [Bonet and Geffner, 2021, Drexler et al., 2021]

- Planning
  - Find path from initial state to goal state
- Sketches express subgoal structure in a planning problem
- Sketches decompose problems into subproblems
- Class of domain instances with similar goal
  - e.g. clearing a block in the blocksworld

Sketch Rules

- Features
  - Numerical n: how many blocks are stacked
  - Boolean b: whether a block is being held
- Sketch consists of one or multiple sketch rules
- Sketch rule: $C \rightarrow E$
  - C: $n{>}0, n{=}0,\ b, \neg b$
  - E: $n{\uparrow}, n{\downarrow}, n{=},\ b, \neg b, b{=}$
  - not mentioned then value doesn't matter
- e.g. $clear(x)$[Bonet and Geffner, 2021]

$$\{\neg H, n > 0\} \rightarrow \{H, n\downarrow\} \qquad n = \text{number of blocks above } x$$
$$\{H, n > 0\} \rightarrow \{\neg H, n{=}\} \qquad H = \text{whether a block is being held}$$

Good Sketch

Definition (Good sketch)

For a sketch over a class of problems Q to be *good*, the following conditions should hold for all instances in Q;

1. For every state s of the instance, there should exist a sketch rule such that its condition is true in s and its effect can be reached from s, unless s is a goal state or a state from which it is impossible to reach the goal.
2. A sketch rule should not lead to a state from which the goal cannot be reached.
3. Every path consisting of a chain of rule applications should eventually reach the goal.
4. If it is possible to reach the goal from the initial state, there should exist at least one path consisting of a chain of rule applications.

Verifying Sketches

- Convert constraints of a good sketch to CTL_f^*
- E.g. constraint 2: A sketch rule should not lead to a state from which the goal cannot be reached.

$$\neg \bigvee_i \Big(\mathbf{EF}\big(C_i \wedge \mathbf{EF}(E_i \wedge \neg \mathbf{EF} goal)\big) \Big)$$

- Model check over transition systems of instances

Generating sketches

- [Drexler et al., 2022]
  - Generate feature pool [Bonet et al., 2019]
  - Encode constraints into answer set programming (ASP)
  - Return a desired number of solutions, using a set of PDDL instances
- Our work
  - Generate feature pool [Bonet et al., 2019]
  - Generate sketch pool
    - Combine all possible conditions and effects in rules
    - Combine rules in sketches
  - Verify sketches over a set of PDDL instances

Preliminary Evaluation

Done

- Implementation using NuSMV
- Verified sketches from [Drexler et al., 2022] for $blocks_{clear}$, $blocks_{on}$, *gripper*, *miconic*
- Generated and verified sketches, e.g. *gripper*

$s_0 : \{True\} \rightarrow \{n_g \uparrow\}$

$s_1 : \{True\} \rightarrow \{n_b \uparrow\}$

$s_2 : \{True\} \rightarrow \{n_a \downarrow\}$

$s_3 : \{n_a > 0\} \rightarrow \{n_a \downarrow\}$

$s_4 : \{True\} \rightarrow \{n_{\neg g} \downarrow\}$

$s_5 : \{n_{\neg g} > 0\} \rightarrow \{n_{\neg g} \downarrow\}$

$s_6 : \{True\} \rightarrow \{b\}$

$s_7 : \{\neg b\} \rightarrow \{b\}$

n_g = number of blocks at their goal destination

n_b = number of blocks in room b

n_a = number of blocks in room a

$n_{\neg g}$ = number of blocks not at their goal destination

b = whether there are balls that are not at their goal destination

Preliminary Evaluation

Done

- Implementation using NuSMV
- Verified sketches from [Drexler et al., 2022] for $blocks_{clear}$, $blocks_{on}$, *gripper*, *miconic*
- Generated and verified sketches, e.g. *gripper*
- Compared manually with sketches generated by Drexler et al.

To do/limitations

- Model check with bounded sketch width
- pre-filtering sketches/feature sets

References

Bonet, B., Frances, G., and Geffner, H. (2019).
Learning features and abstract actions for computing generalized plans.
In *Proc. AAAI 2019*, pages 2703–2710.

Bonet, B. and Geffner, H. (2021).
General policies, representations, and planning width.
In *Proc. AAAI 2021*, pages 11764–11773.

Drexler, D., Seipp, J., and Geffner, H. (2021).
Expressing and exploiting the common subgoal structure of classical planning domains using sketches: Extended version.
arXiv preprint arXiv:2105.04250.

Drexler, D., Seipp, J., and Geffner, H. (2022).
Learning sketches for decomposing planning problems into subproblems of bounded width: Extended version.
arXiv preprint arXiv:2203.14852.