The Barman-HTN Domain for IPC 2020

Max Waters, Lin Padgham, Sebastian Sardina
RMIT University, Melbourne, Australia
max.waters@rmit.edu.au, sebastian.sardina@rmit.edu.au, lin.padgham@rmit.edu.au

Abstract

The *Barman-HTN* domain is an HTN decomposition of the IPC *Barman* domain. It extends *Barman* with a task network that guides cocktail creation while retaining the primitive operators that made the original so challenging for the delete-relaxation heuristic.

The *Barman-HTN* domain is an HTN extension of the well-known IPC *Barman* domain (López, Celorrio, and Olaya 2015), in which cocktails must be prepared out of various ingredients. Solving a *Barman* instance requires careful management of limited resources. For example, shot glasses and shakers must be clean and empty before use, but filling one with an ingredient deletes both of these conditions, meaning that it must be emptied and cleaned before re-use. This makes *Barman* particularly challenging for planners that use the delete-relaxation heuristic: as action preconditions are frequently deleted and can only be restored by executing further actions, delete-relaxation tends to produce overly optimistic estimates.

This property of the domain operators, combined with recent interest in the delete-relaxation heuristic in HTN planners (Höller, Bercher, and Behnke 2020) and the fact that the cocktail construction task can be naturally decomposed into subtasks, suggest that *Barman* is a suitable basis for an HTN benchmark domain.

Barman-HTN extends *Barman* with an HDDL (Höller et al. 2020) task network that guides the pouring and mixing of ingredients and also provides careful resource management. For example, the method `MakeCocktail` (Figure 1) decomposes the task of mixing and shaking a cocktail. The first subtask, `AchieveCleanShaker`, ensures that the shaker is clean and empty, and the two instances of `AchieveContainsShakerIngredient` ensure that it contains the required ingredients. The next two steps, `AchieveHolding` and `AchieveHandEmpty`, ensure that one hand is empty and the other is holding the shaker. These subtasks satisfy the preconditions of the final step, the action `shake`, which results in the shaker containing the cocktail.

Resource management is handled by tasks and methods that bring about a required condition from any given state. For example, the task `AchieveHolding(?h, ?c)` produces the condition `holding(?h, ?c)`, and is decomposed by two methods. If hand `?h` is already holding container `?c`, then the empty method `AchieveHoldingNull` is applicable. Otherwise, `PickUp` (Figure 1) decomposes the task into `AchieveHandEmpty(?h)` and `AchieveOnTable(?c)`, that satisfy the preconditions of the primitive action `grasp(?h, ?c)` by ensuring that `?h` is empty and `?c` can be picked up, respectively. The task network contains other such decompositions for resource management tasks such as cleaning glasses and shakers, and emptying hands.

```
(:method MakeCocktail
    :parameters (?s - shaker ?c - cocktail
       ?i_1, i_2 - ingredient ?h_1, ?h_2 - hand)
    :task (AchieveContainsShakerCocktail ?s ?c)
    :precondition (and
       (cocktailPart1 ?c ?i_1) (cocktailPart2 ?c ?i_2)
       (not (= ?h_1 ?h_2)))
    :ordered-subtasks (and
       (AchieveCleanShaker ?s)
       (AchieveContainsShakerIngredient ?s ?i_1)
       (AchieveContainsShakerIngredient ?s ?i_2)
       (AchieveHolding ?h_1 ?s)
       (AchieveHandEmpty ?h_2)
       (shake ?c ?i_1 ?i_2 ?s ?h_1 ?h_2)))

(:method PickUp
    :parameters (?h - hand ?c - container)
    :task (AchieveHolding ?h ?c)
    :precondition (not (holding ?h ?c))
    :ordered-subtasks (and
       (AchieveHandEmpty ?h) (AchieveOnTable ?c)
       (grasp ?h ?c)))
```

Figure 1: The `MakeCocktail` and `Pickup` methods.

References

Höller, D.; Behnke, G.; Bercher, P.; Biundo, S.; Fiorino, H.; Pellier, D.; and Alford, R. 2020. HDDL: An Extension to PDDL for Expressing Hierarchical Planning Problems. In *AAAI 2020*, 9883–9891. AAAI Press.

Höller, D.; Bercher, P.; and Behnke, G. 2020. Delete- and Ordering-Relaxation Heuristics for HTN Planning. In *IJCAI 2020*, 4076–4083. IJCAI.

López, C. L.; Celorrio, S. J.; and Olaya, A. G. 2015. The deterministic part of the seventh International Planning Competition. *Artificial Intelligence* 223: 82–119.