C3: Planning with Consistent Causal Chains

Nir Lipovetzky
Departamento de Tecnología
Universitat Pompeu Fabra
08003 Barcelona, SPAIN
nir.lipovetzky@upf.edu

Miquel Ramírez
Departamento de Tecnología
Universitat Pompeu Fabra
08003 Barcelona, SPAIN
miquel.ramirez@upf.edu

Hector Geffner
Departamento de Tecnología
ICREA & Universitat Pompeu Fabra
08003 Barcelona, SPAIN
hector.geffner@upf.edu

The automatic derivation of informative heuristic functions has been a key development in modern domain-independent planning. Heuristic functions provide the search for plans with a sense of direction that allows large problems to be solved quite effectively. Heuristic search planners, on the other hand, are not transparent: it is not clear why and when they will work, it is not simple to explain, as humans do, why certain actions are selected and others are discarded, and most important of all, it is not simple to improve them in spite of their known limitations.

The C3 planner approaches the problem of inference in planning from a different perspective. Rather than relying on the extraction and use of heuristic functions or reductions into SAT or CSPs, C3 prunes 'bad actions' by appealing to the notion of *consistent causal chains* (Lipovetzky, Ramirez, and Geffner 2008). A causal chain $a_0, p_1, a_1, p_2, a_2, \ldots, p_n, a_n$ is a sequence of actions a_i and fluents p_i such that a_0 is an action applicable in the current state, and p_{i+1} is a precondition of action a_{i+1} that is added by the action a_i. A causal chain that terminates with the action $a_n = End$ is called a *path* and connects the action a_0 with the goal. In principle, an action a_0 can be deemed as relevant to the goal if there is a *path* starting with a_0. This is a standard notion of relevance. However, by enforcing *the semantics of the causal links* a_i, p_i, a_{i+1} in such paths (Tate 1977; McAllester and Rosenblitt 1991), it is possible to propagate side effects along such chains and detect in polynomial time that some of these chains cannot be part of any plan. The causal chains and paths that can be shown to be logically impossible, are said to be *inconsistent*, while an action a_0 in a state s is deemed inconsistent if it does not start a consistent path leading to the goal. Inconsistent actions can then be pruned.

This pruning rule is simple but powerful: the planner C1, which is a plain backtracking forward-state search planner with a version of this pruning rule turns out to solve as many benchmark problems as the effective Enforced Hill Climbing (EHC) search of FF (Hoffmann and Nebel 2001), many of them backtrack-free (Lipovetzky, Ramirez, and Geffner 2008). In this version of inconsistency pruning, the consistency pruning criterion is applied to the *minimal paths* rather than to all paths. A path $a_0, p_1, a_1, p_2, a_2, \ldots, p_n, a_n$ is minimal if the supporters a_i of the fluents p_{i+1} in the path are 'best' according to a simple criterion: a_i is a best supporter for p_{i+1} in a state s if a_i is an action that adds p_{i+1} and no action that adds p_{i+1} has a smaller $h_{max}(a_i)$ value (Bonet and Geffner 2001).

For example, in the initial state s_0 of the Tower-n domain (Vidal and Geffner 2005), that represents a simple class of Block-World problems, where n blocks 1, 2, ..., n on the table must be arranged into a single tower with block i on block $i+1$ for all $i < n$, there are paths of the form:

$$t_1: \quad pick(i), hold(i), stack(i, i{+}1), on(i, i{+}1), End$$

$$t_2: \quad pick(i), hold(i), stack(i, j), on(i, j), unstack(i, j), hold(i), on(i, i{+}1), End$$

The first path is a minimal path for any $i < n$, but the second path is not as the action $unstack(i, j)$ is not a best supporter of the fluent $hold(i)$ in s_0 (the best supporter for $hold(i)$ in s_0 is $pick(i)$). Moreover, the first path is *consistent* only for $i = n-1$, meaning that *the only action that is consistent* in this state, is the action $pick(n-1)$. Notice that in this example, all the actions $pick(i)$ are applicable and helpful according to FF.

The planner C3 is a refined version of the simple, forward-state C1 planner above that combines a backtrack search with consistency pruning. More details on the ideas underlying the C3 planner can be found in (Lipovetzky, Ramirez, and Geffner 2008).

References

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence* 129(1–2):5–33.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14:253–302.

Lipovetzky, N.; Ramirez, M.; and Geffner, H. 2008. Structural inference in planning: Scaling up without heuristic estimators. Submitted.

McAllester, D., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *Proceedings of AAAI-91*, 634–639. Anaheim, CA: AAAI Press.

Tate, A. 1977. Generating project networks. In *Proc. IJCAI*, 888–893.

Vidal, V., and Geffner, H. 2005. Solving simple planning problems with more inference and no search. In *Proc. of the 11th*

Int. Conf. on Principles and Practice of Constraint Programming (CP-05). Springer.