Goal-Clobbering Avoidance in Non-Linear Planners

Rujith de Silva
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213-3891
desilva+@cmu.edu

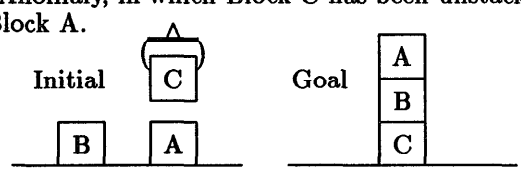
A central issue in non-linear planning is the ordering of operators so as to avoid undesirable interactions between their effects. The Modal Truth Criterion (Chapman 1987) states the conditions under which these interactions will occur. Non-linear planners use the Criterion, directly or indirectly, to promote or demote operators, or to co-designate variables, so as to avoid interactions.

This abstract describes a method, called *Goal Clobbering Avoidance* (GCA), to avoid some interactions in a partially-ordered plan by promoting or demoting a sequence of operators, rather than individual operators. Effectively, it simultaneously applies the Modal Truth Criterion to all operators in the sequence, using pre-compiled information about the domain.

GCA will be illustrated in the familiar Blocksworld domain, with the operators

⟨put-down ?block⟩	on table
⟨pick-up ?block⟩	from table
⟨stack ?blockA ?blockB⟩	
⟨unstack ?blockA ?blockB⟩	

Consider the following problem, related to Sussman's Anomaly, in which Block C has been unstacked from Block A.



Furthermore, suppose a partially-ordered plan has been built as shown towards solving the problem.

(on A B)	(on B C)
⟨stack A B⟩	⟨stack B C⟩
(holding A)	(holding B)
⟨pick-up A⟩	⟨pick-up B⟩
(arm-empty)	
⟨put-down C⟩	

This plan has a large number of interactions involving (arm-empty), (clear B) and (clear C), and it is not immediately clear how to promote or demote the operators to achieve the desired goals.

Consider the goals and operators under (on A B) in relation to the sibling goal (on B C). If (on A B) is achieved first, then *any* plan that achieves (on B C) will dis-achieve (on A B). Similarly, any such plan will also dis-achieve (holding A).

Hence applying ⟨pick-up A⟩ before ⟨stack B C⟩ is pointless, as its effects will be clobbered by the later achievement of (on B C). However, applying ⟨put-down C⟩ is not pointless, as (arm-empty) is *not* dis-achieved by (on B C). Therefore the planner can split the left-hand branch by constraining ⟨pick-up A⟩ and ⟨stack A B⟩ to occur after the achievement of (on B C). Furthermore, it can do this *before* even deciding how to achieve (on B C), as it makes use of the following judgements expressing properties of *all* possible ways of achieving (on B C):

Goal-clobbering by (on ?B ?C) of (on ?A ?B)
Goal-clobbering by (on ?B ?C) of (holding ?A)

The first states that any plan, justified or not, whose final operator achieves (on ?B ?C) must result in a final state in which (on ?A ?B) is not true. Similarly for the second. Conditional goal-clobberings that constrain the initial states in which they are applicable also exist.

GCA has yielded large savings in planning-time in numerous domains on the totally-ordered non-linear planner PRODIGY(Etzioni 1991). I am currently evaluating its performance in partially-ordered planners. In addition, I am automating the derivation of goal-clobbering judgements, which explicitly state provable properties of the domain, by extending the work done by (Etzioni 1991).

References

Chapman, D. 1987. Planning for conjunctive goals. *Artificial Intelligence* 32:333–377.

Etzioni, O. 1991. STATIC: A problem-space compiler for PRODIGY. In Proceedings of the Eighth National Conference on Artificial Intelligence, 533–540.