Planning Techniques: TO

Dra. Mª Dolores Rodríguez Moreno

Objectives

Specific Objectives

- Understand TOP technique

Source

- D. Weld. An Introduction to Least Commitment Planning. AI Magazine, 1994
- Veloso et al. Integrating Planning and Learning: The PRODIGY Architecture, 1995

Outline

- **Definition**
- Prodigy planner
- Conclusions

Definition (I)

- The solution is an **ordered** sequence of actions
- A plan is *valid* if the preconditions of every operator are satisfied before the execution of the operator
- It maintains a total-order plan
- Overcommitted, introduces constraints not necessary
- Given a problem, most planning algorithms start with empty plan and modify it until a solution plan is found

Definition (II)

- A plan may be modified by inserting a new operator, imposing a constraint on the order of operators in the plan, or instantiating a variable in the description of an operator
- The plans considered during the search for a solution are called incomplete plans
- Each incomplete plan may be viewed as a node in the search space of the planning algorithm
- Modifying a current incomplete plan corresponds to expanding a node
- The branching factor of search is determined by the number of possible modifications of the current plan

Definition (III)

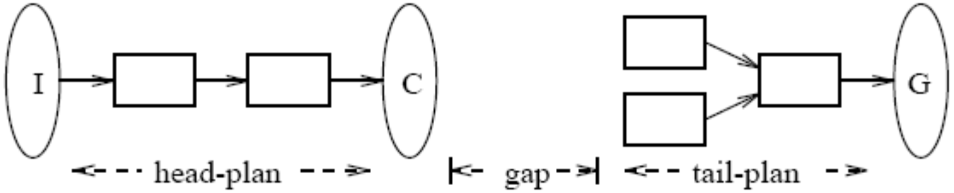
- Distinguishing TO must be separated from the distinction of SSS and PSS based planners
  - SSS: each search state corresponds to a state in the world
  - PSS: each search state corresponds to a plan
- Generally, TO is associated to SSS: Prodigy or VVPLAN, although there are PSS: INTERPLAN or WARPLAN

Outline

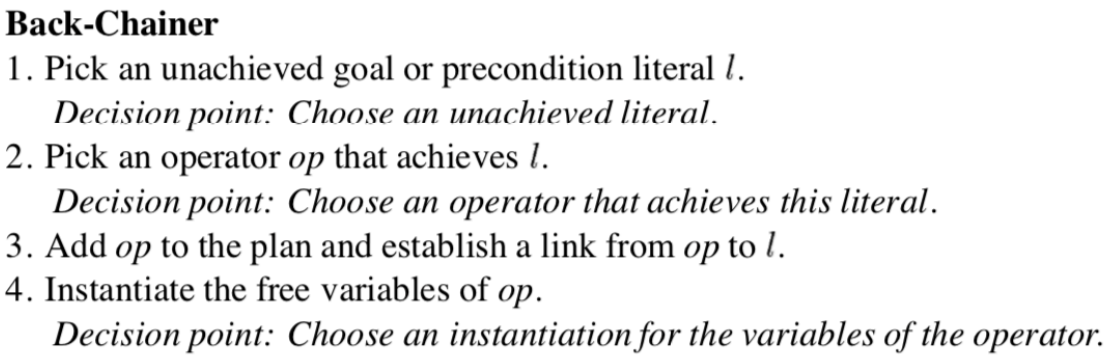
- Definition
- **Prodigy planner**
- Conclusions

TO: Prodigy 4.0 (I)

- Nonlinear, domain independent planner that includes learning (Carnegie Mellon)
- An incomplete plan consists of:
  - The tail-plan is built using a backward chaining algorithm, which starts on the goals and adds operators one by one, to achieve the goal and the preconditions of the operators in the tail-plan that are not satisfied in the current state (pending goals)
  - The head-plan: applies operators in the tail-plan (*valid total-order plan)*



TO: Prodigy 4.0 (II)



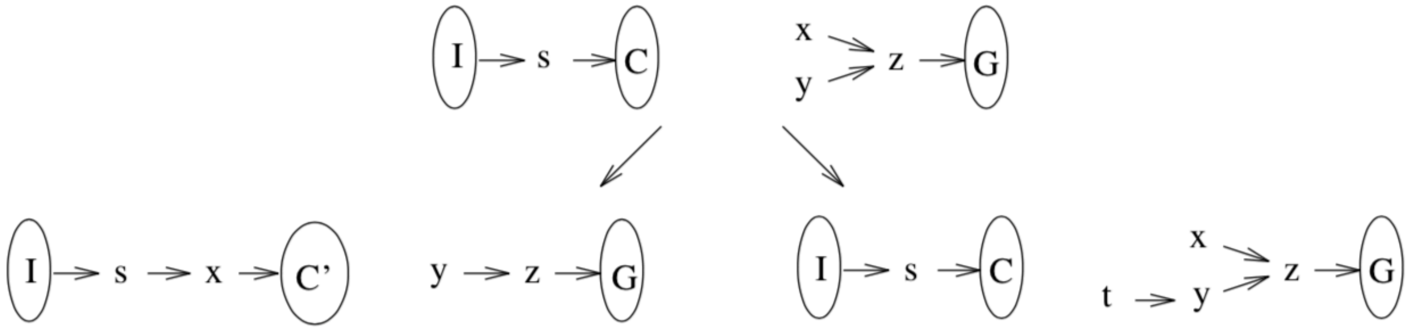
Back-Chainer

1. Pick an unachieved goal or precondition literal l.

   Decision point: Choose an unachieved literal.
2. Pick an operator *op* that achieves l.

   Decision point: Choose an operator that achieves this literal.
3. Add *op* to the plan and establish a link from *op* to l.
4. Instantiate the free variables of *op*.

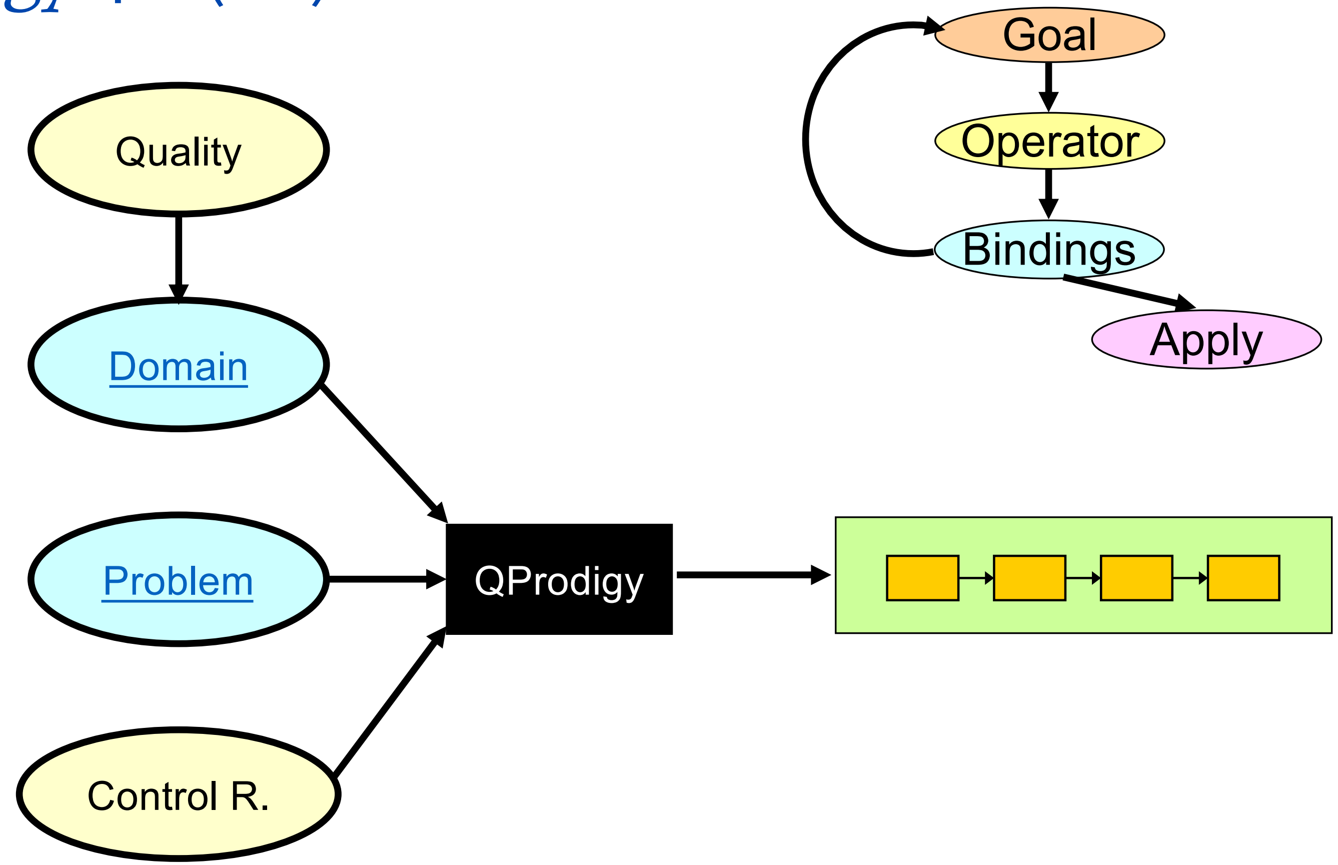
   Decision point: Choose an instantiation for the variables of the operator.



Applying an operator (moving it to the head)

Adding an operator to the tail-plan

TO: Prodigy 4.0 (III)



Conclusions

- Definition
- Prodigy planner
- **Conclusions**

Conclusions

- Forward/backward state-space searches are forms of totally ordered plan search
- Explore only strictly linear sequences of actions