Planning Techniques: POP

Dra. Mª Dolores Rodríguez Moreno

Objectives

Specific Objectives

- Understand POP techniques

Source

- D. Weld. An Introduction to Least Commitment Planning. AI Magazine, 1994
- Eva Onaindia De La Rivaherrera. Planificación Automática. Videos. UPV. https://media.upv.es/

Outline

- Motivation
- Definition
- POP tree
- UCPOP
- Conclusions

Motivation

- Having a totally ordered list of steps is restrictive
- Can we only do something if it's necessary?
- POP uses the principle of least commitment: never making a choice unless required to do so

Outline

- Motivation
- **Definition**
- POP tree
- UCPOP
- POP planners
- Conclusions

Definition (I)

- Perform PSS
- Partial Plan: $\bar{P} = (\bar{A}, \bar{O}, \bar{L}, \bar{OC}, \bar{UL})$ Action, Ordering constraints, causal Link, Open Conditions (flaws), Unsafe Links
- Backward search
- Each node has:
  - Partially instantiated actions
  - Set of constraints
  - Process stops if the solution is found, after several refinements (adding new operators)

Definition (II)

- The planning algorithm implements the *least commitment* technique
  - Only essential planning decisions are saved because it is not necessary to commit
  - The causal link structure is responsible for storing them
    - *3 fields: producer, consumer and the proposition*
  - As can be actions that threaten it, we can apply:
    - Demotion: add the restriction before the step that threats it
    - Promotion: add the restriction after the step that threats it
    - Separation: add the restriction to the variable binding
    - Confrontation: add the negation to the conditional effects

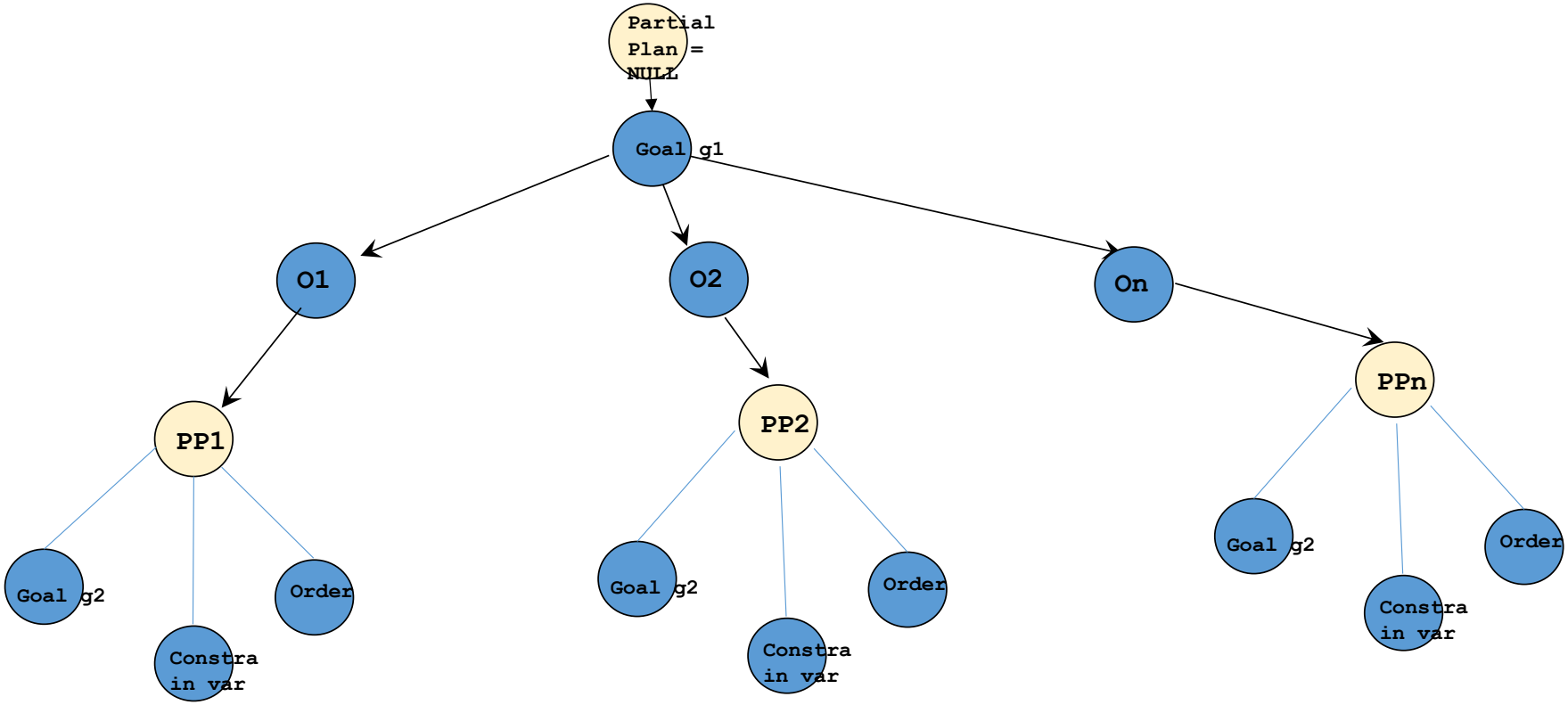
$A_p \xrightarrow{Q} A_c$

- Examples: UCPOP, Cassandra, ZENO, VHPOP

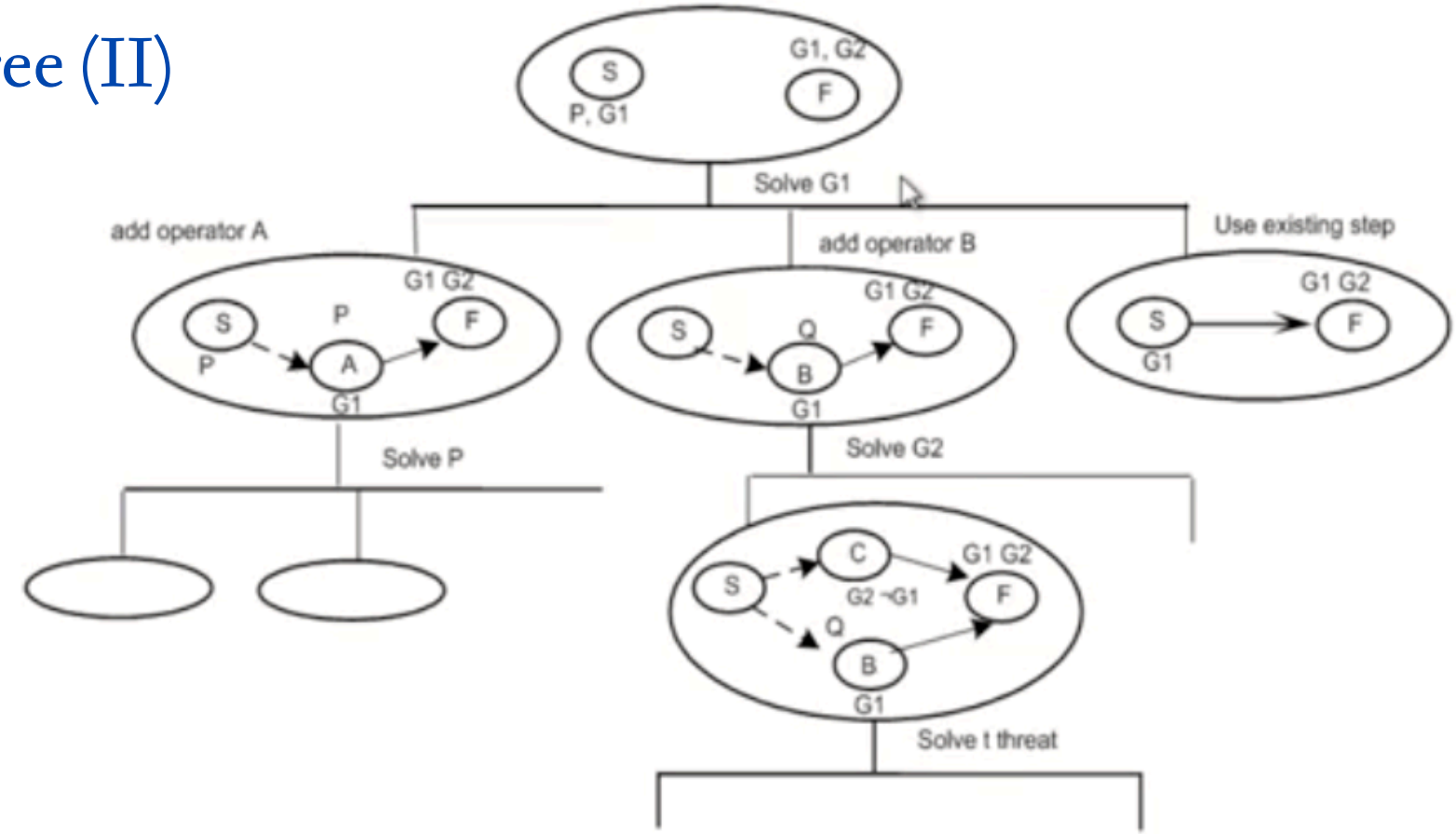
Outline

- Motivation
- Definition
- **POP tree**
- POP planners
- UCPOP
- Conclusions

POP tree (I)


Partial Plan = NULL
Goal g1
O1
O2
On
PP1
PP2
PPn
Goal g2
Constra in var
Order
Goal g2
Constra in var
Order
Goal g2
Constra in var
Order

POP tree (II)


S
P, G1
G1, G2
F
Solve G1
add operator A
add operator B
Use existing step
G1 G2
S
P
A
F
G1
G1 G2
S
Q
B
F
G1
G1 G2
S
F
G1
Solve P
Solve G2
C
G1 G2
S
G2 ¬G1
F
Q
B
G1
Solve t threat

POP tree (III)

- The initial plan is created from the initial state description and the goal description by creating two "pseudo-steps:"
  - Start
    - P: none
    - E: all positive literals defining the initial state
  - Finish
    - P: literals defining the conjunctive goal to be achieved
    - E: none
- Then creating the initial plan as: Start ---------> Finish
- Searching for a Solution in Plan Space

Outline

- Motivation
- **Definition**
- POP tree
- **POP planners**
- UCPOP
- Conclusions

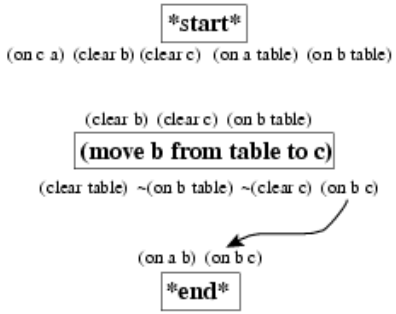
POP Planners

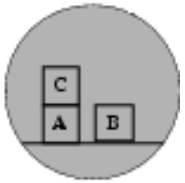
- UCPOP
- Cassandra
- ZENO
- VHPOP

Outline

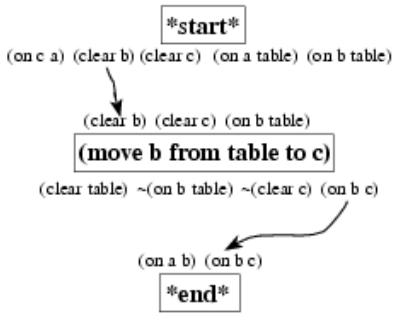
- Motivation
- Definition
- POP tree
- POP planners
- **UCPOP: example**
- Conclusions

UCPOP (I)



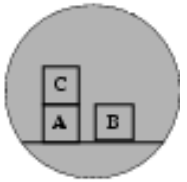


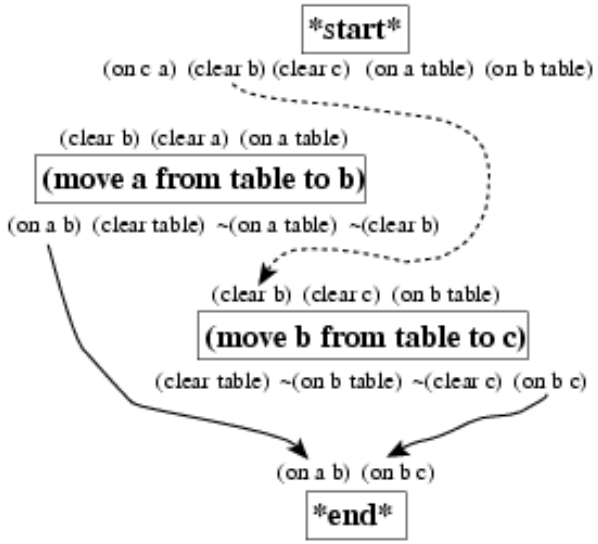
After adding a causal link to support `(on B C)`, the plan is as shown and **agenda** contains {`(clear B) (clear C) (on B Table) (on A B)`} as open propositions.



After adding a causal link to support `(clear B)`, the plan has two causal links and **agenda** is set to {`(clear C) (on B Table) (on A B)`}.

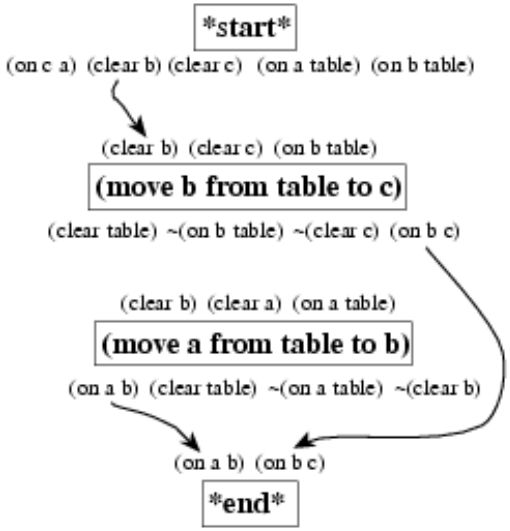
UCPOP (II)

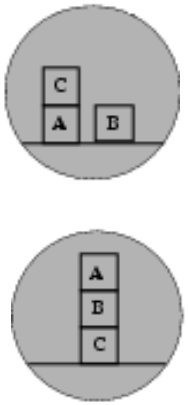




Since the `move-A` action could possibly precede the `move-B` action, it threatens the link labeled `(clear B)` as indicated by the dashed line.

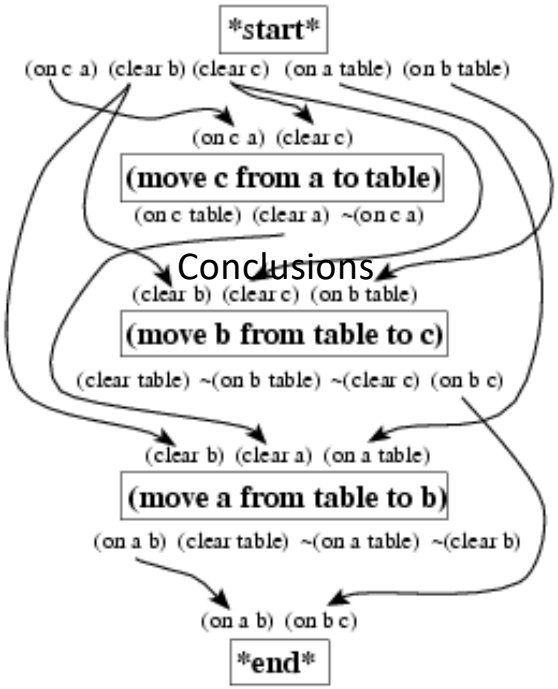
UCPOP (III)



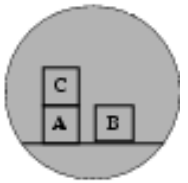


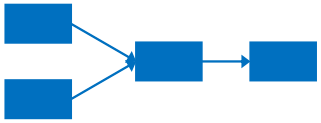
After promoting the threatening action, the plan's actions are totally ordered.

UCPOP (IV)



Conclusions





Outline

- Motivation
- Definition
- POP tree
- UCPOP
- **Conclusions**

Conclusions

- Idea
  - Work on several subgoals independently
  - Solve them with subplans
  - Combine the subplans
  - Flexibility in ordering the subplans
- *Least Commitment* strategy: delaying a choice during search
- Causal links lead to early pruning of portions of the search space because of irresolvable conflicts

Todo example

- Goal: Set the table, i.e., on(Tablecloth) ^ out(Glasses) ^ out(Plates) ^ out(Silverware)
- Initial State: clear(Table)
- Operators:
  - Lay-tablecloth

    P: clear(Table)

    E: on(Tablecloth), ~clear(Table)
  - Put-out(x)

    P: none

    E: out(x), ~clear(Table)