Sample solution for POP question.

**Iteration 1:** empty plan

- **consistency:** OK
- **not yet achieved:** \{7\text{clogged}, 7\text{armed}\}
  - choose: 7\text{clogged}
  - applicable existing operators: \{\text{start}\}
  - choose: \text{start}

**Iteration 2:**

- **consistency:** OK
- **not yet achieved:** \{7\text{armed}\}
  - choose: 7\text{armed}
  - applicable existing operators: \\emptyset
  - applicable new operators: \\{\text{dunk}\}

**Iteration 3:**

- **consistency:** dunk has effect clogged which threatens the protected literal 'clogged' solution 'clogged' will not work since that would mean ordering 'dunk' before 'start'.
- **promotion** does not work either for a similar reason (w & finish)
- **backtrack:** OK
  - last decision point: \(*\)
    - There was no other alternative
  - backtrack again: OK
    - previous decision point: \(*\)
      - There was no other alternative: 'start' was the only existing operator
Iteration 4

- Start
  - Flush
  - Not clogged & not armed
    - Finish

- Not yet achieved: \( T \mod 3 \)
- Choose: \( T \) armed
- Applicable existing operators: \( \notin \)
- Applicable new operators: \( \notin \)
- Choose: \( \text{dunk} \)

Iteration 5

- Start
  - Flush
  - Dunk
  - Not clogged & not armed
    - Finish

- Consistency: 'dunk' has effect clog, which threatens the protected literal 'clogged'
- Solutions: promote or demote the clobberer, namely 'dunk'
- Promotion: does not work, since 'dunk' would be ordered after 'finish', thus leads to an inconsistency
- Demotion: OK, order 'dunk' before '

Iteration 6

- Start
  - Not clogged
    - Dunk
      - Flush
      - Not clogged & not armed
        - Finish

- Consistency: OK
- Not yet achieved: \( \{ T \text{clogged} \} \)
- Choose: \( T \text{clogged} \)

- Applicable existing operators: \( \{ \text{start} \} \)
- Choose: \( \text{start} \)
At iteration 4, when choosing to add the 'chalk' operator to the plan, the algorithm only continued if 'start' and before 'finish' were not chosen. Since its 'temporal' operator did not matter, the plan only for all 'draw' on iteration 5, because the temporal order of the 'chalk' and 'draw' operators did not matter, the plan can be summarized in two ways: 'start-flush-dunk-flush' or 'finish-flush-dunk-flush'.

The plan is not totally ordered, namely Iteration 5.

Thus, the situation B, i.e., the current situation, is not yet checked.