ON COMPUTING CONFORMANT PLANS USING CLASSICAL PLANNERS: A GENERATE AND COMPLETE APPROACH

Khoi Nguyen, Vien Tran,
Tran Cao Son and Enrico Pontelli
New Mexico State University

Take Home Message

- A novel approach to conformant planning
  - Simple
  - Coverage: 93% solved instances compared to 58% of state-of-the-art planners.

Outline

- Conformant Planning Problem
- Intuition
- Generate-and-Complete Approach
- Result
- Conclusion

Conformant Planning Problem

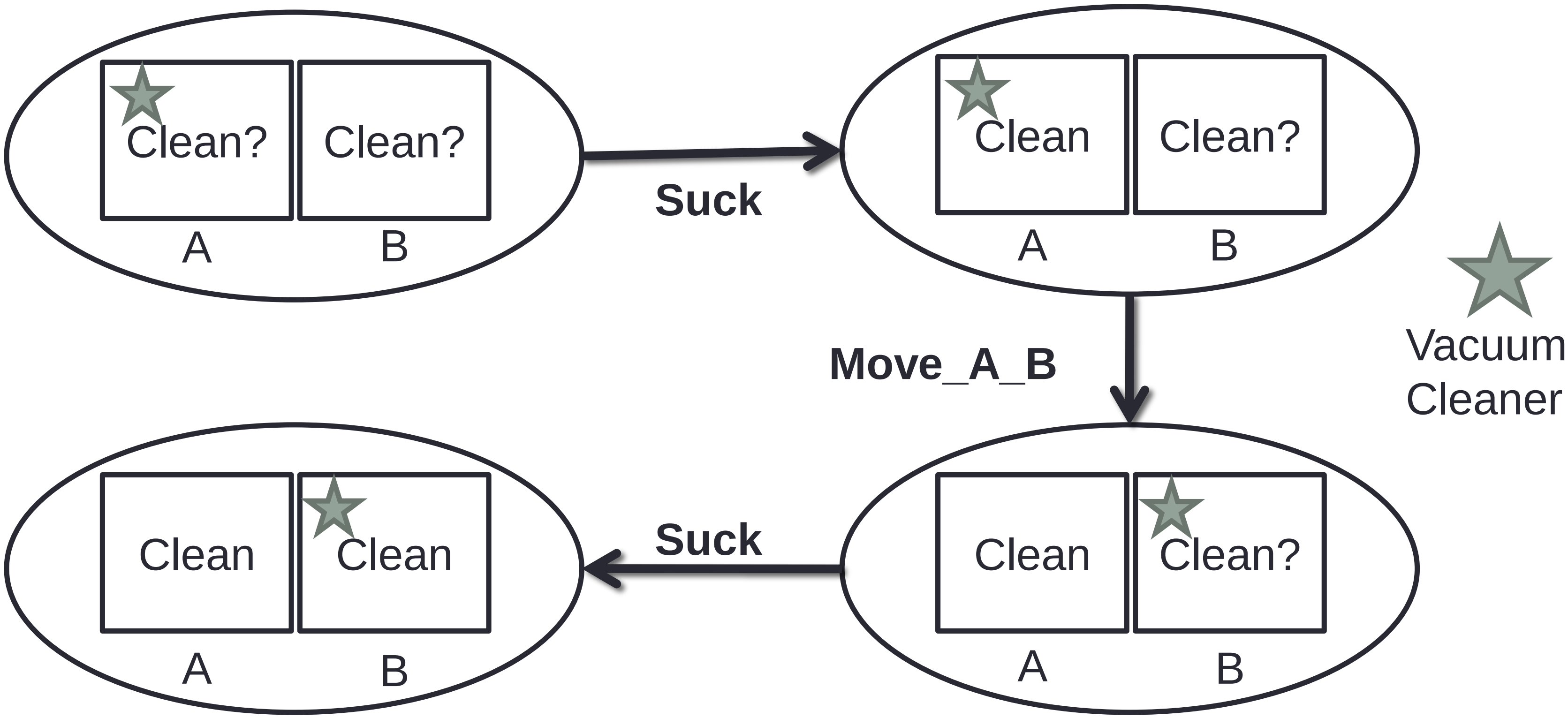
- Planning under uncertainty and no observation
  - Incomplete initial state
  - Conformant plan: a sequential plan that achieves *goal* from all possible initial states of the world

Conformant Planning Problem

- Given: planning problem *P* = < *F*, *O*, *I*, *G*> where
  - *F* is a set of propositions
  - *O* is a set of operators
  - *I* is the incomplete initial state
  - *G* is the goal
- Problem: Computing a plan that achieves *G* from all possible initial states of the world satisfying *I*

Vacuum Cleaner World

F = {at(A), at(B), clean(A), clean(B)}
O = {Move_A_B, Move_B_A, Suck}
I = {at(A), one_of(clean(A), not clean(A)), one_of(clean(B), not clean(B))}
G = {clean(A), clean(B)}



Conformant Planning Problem

- Conformant Planning Problem could be viewed as belief state space search (State-of-the-art approach).

- Belief state = Set of states = Set of possible worlds

Previous works

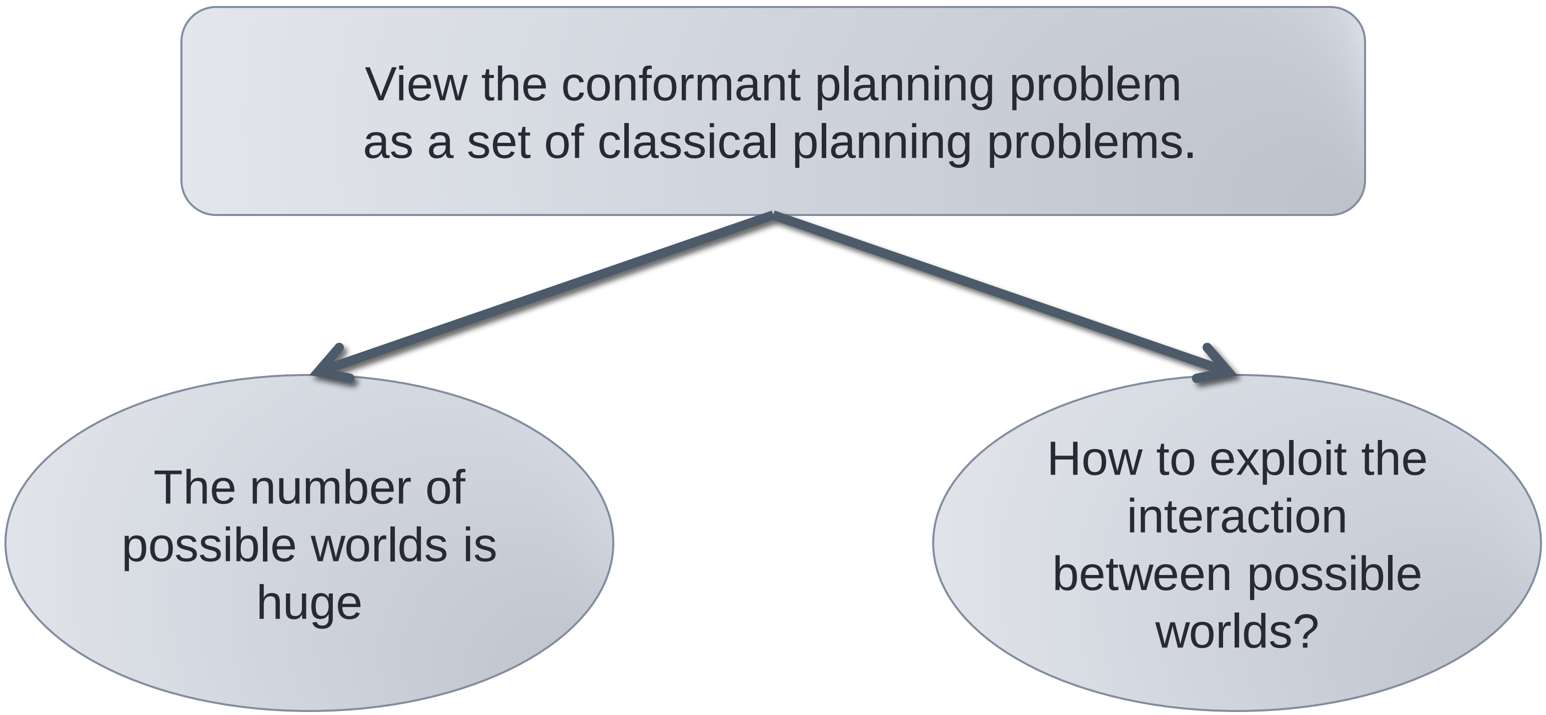
- Belief state space search: CFF, POND, DNF, CNF
- Approximation state space search: CpA
- Translation to classical planning: t0
- Other approaches
  - Generate and test approach: C_PLAN
  - Fragment approach: FRAG_PLAN

Challenge

- Consider a planning problem with n fluents.
  - Classical planning (0 unknown fluents):
    - Number of possible initial worlds is 1.
    - Size of states space 2^n.
  - Conformant planning (k unknown fluents):
    - Number of possible initial worlds is 2^k.
    - Size of belief states space 2^{2^n} (HUGE!).

Intuition

- Alternative approach to belief state search approach?
  - Generate and test approach (C_PLAN)
  - Fragment approach (FRAG_PLAN)



Intuition

- Four possible worlds:
  - s_0: Both squares are clean → []
  - s_1: Only A is dirty → [Suck]
  - s_2: Only B is dirty → [Suck, move_A_B, Suck]
  - s_3: Both squares are dirty → [Suck, move_A_B, Suck]
- Conformant plan: [Suck, move_A_B, Suck]

Our Approach (1)

View the conformant planning problem as a set of classical planning problems.

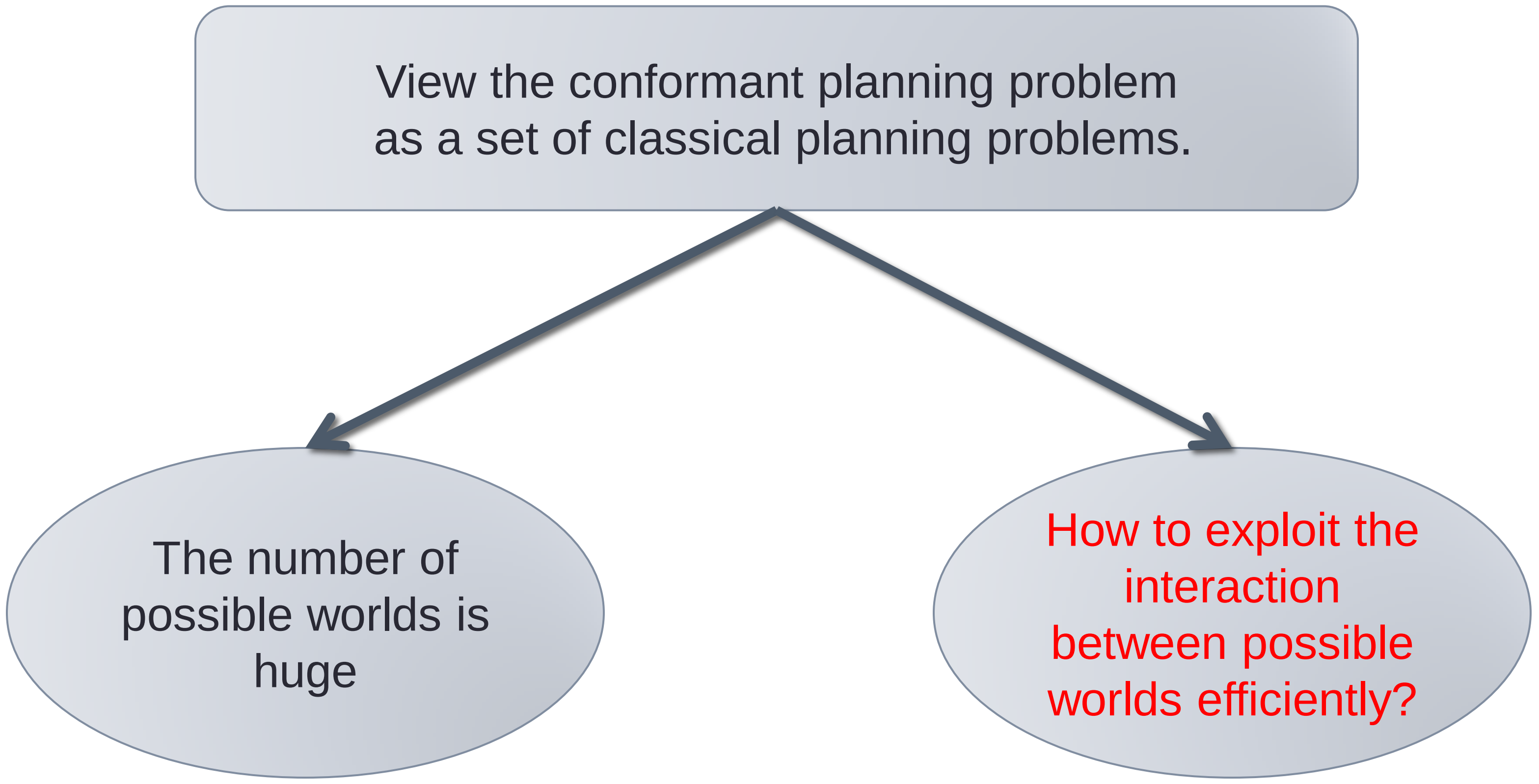
The number of possible worlds is huge

How to exploit the interaction between possible worlds efficiently?

Reduction the Size of Initial Belief State

- One-of reduction technique (CpAh 2008) could be used to reduce the number of possible worlds.
- In the vacuum cleaner world:
  - Originally, 4 possible initial states
  - After one-of reduction, 2 possible initial states

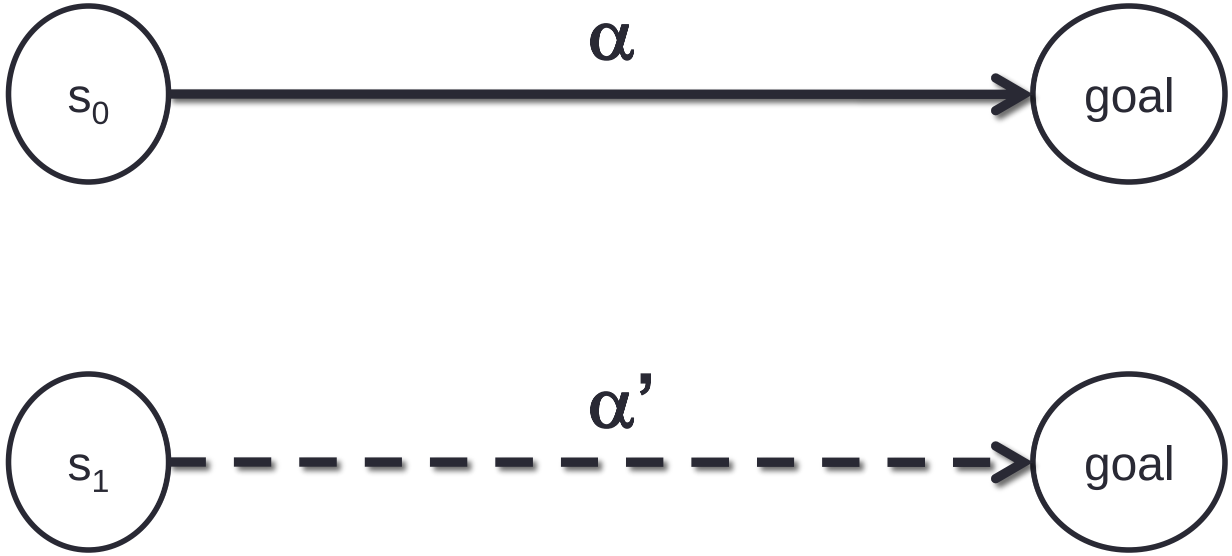
Our Approach (2)



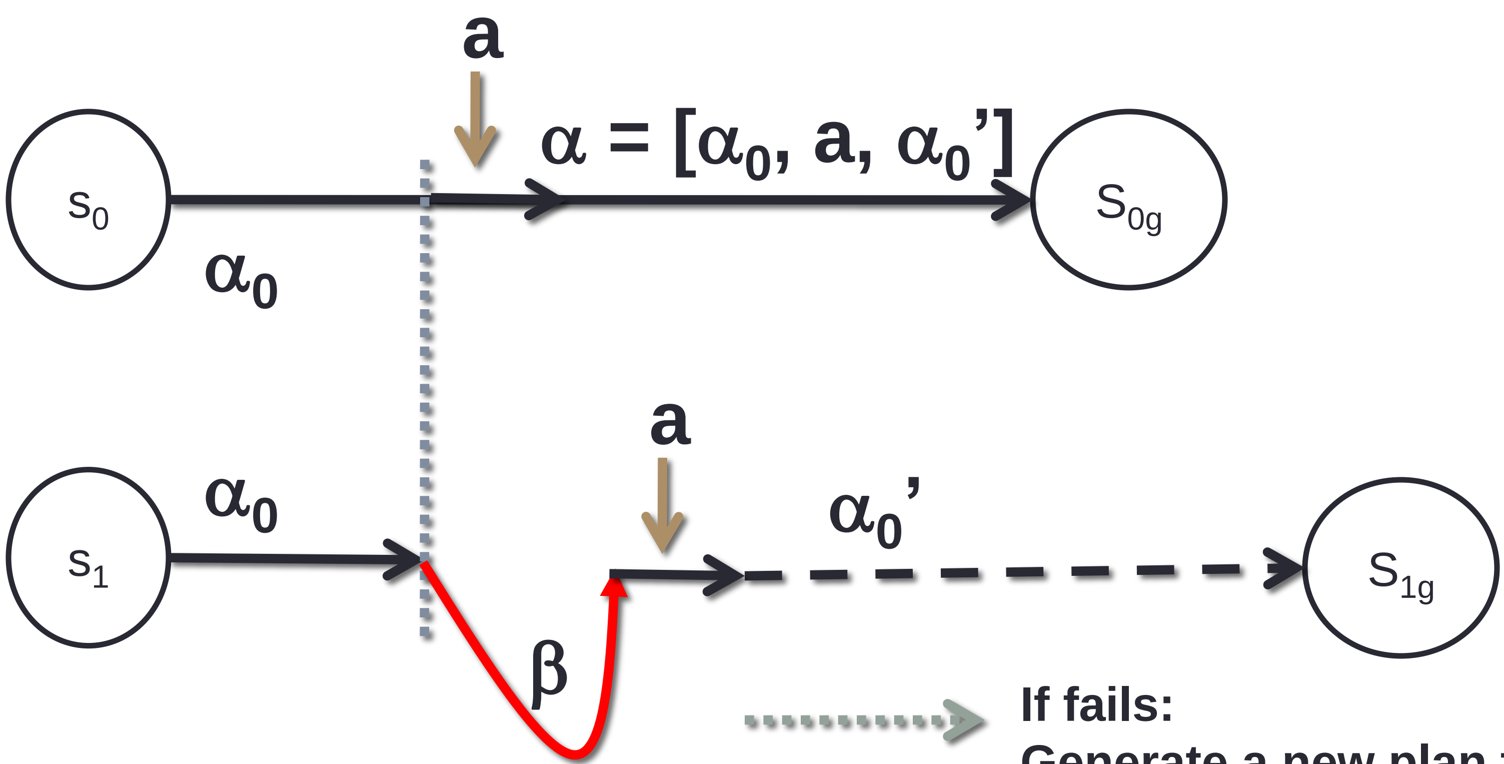
Generate and Complete (Idea)

- Given a plan α for one possible world, how do we modify it to obtain a plan for another world?
- Insert actions into α to create a new plan α' that
  - achieves the goal from the new possible world
  - has the following desirable properties:
    - α' is executable in the new possible world
    - α' maintains the same effect of each action in both worlds

Generate and Complete (Illustration)



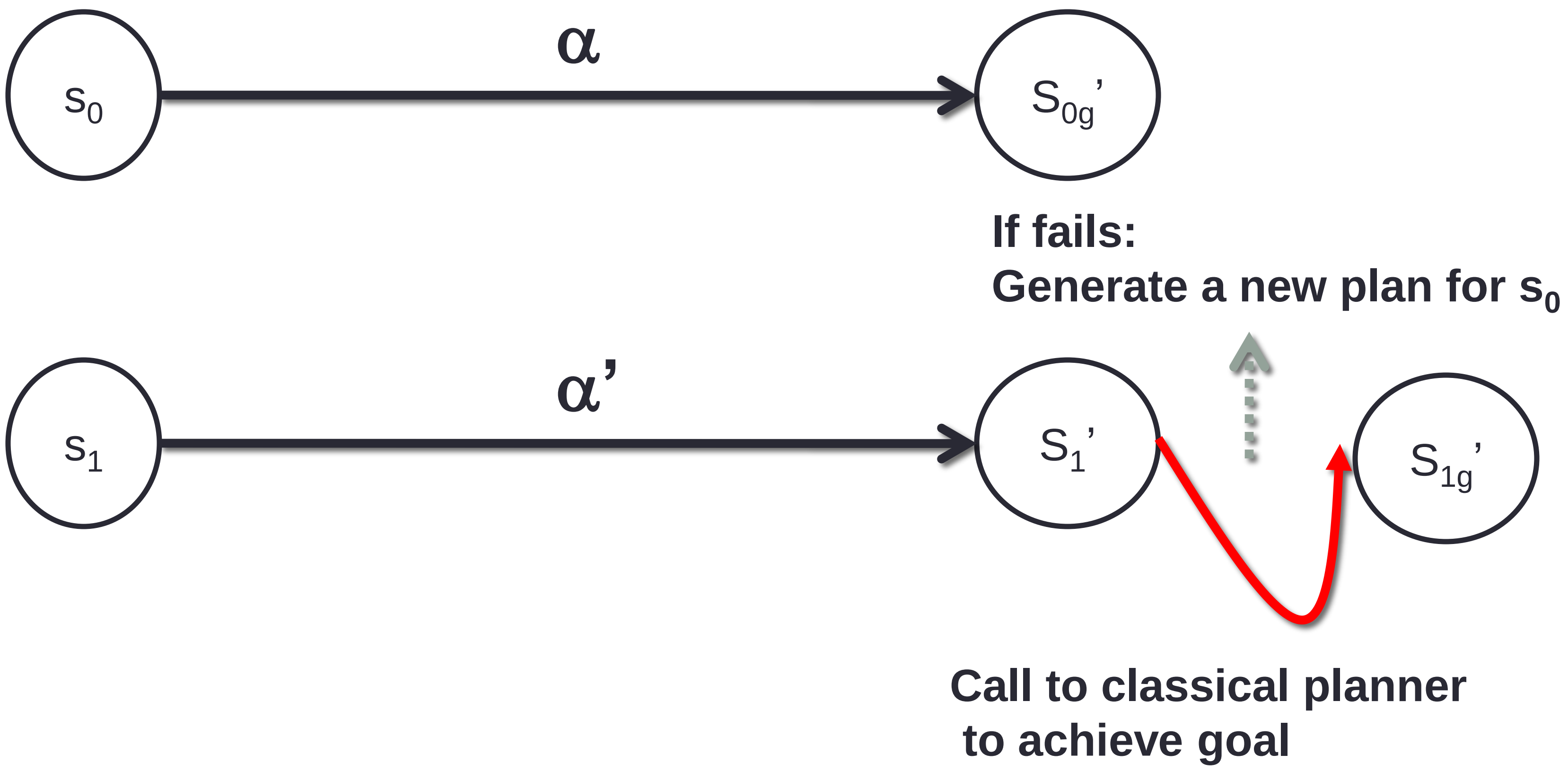
Generate and Complete (Illustration)



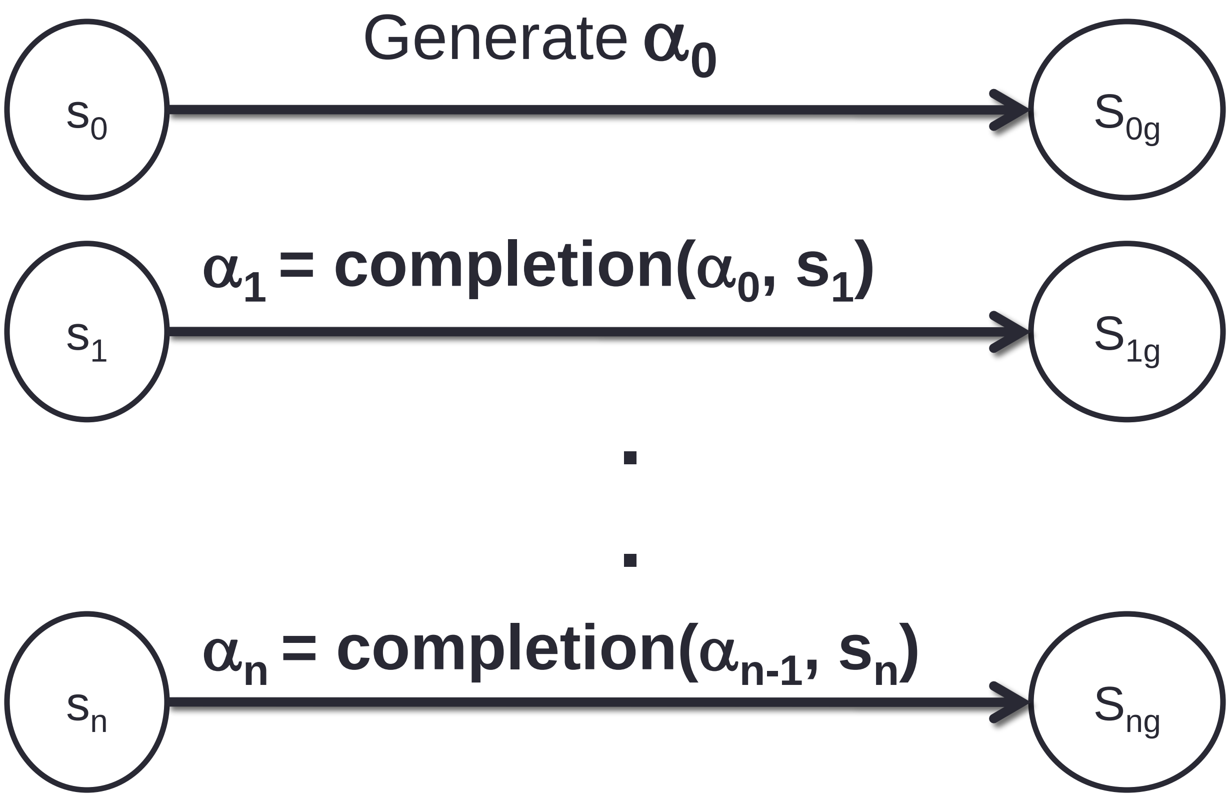
If fails:
Generate a new plan for s_0

Call to classical planner:
_ β achieves precondition of a
_ β maintains effects of a in both worlds

Generate and Complete (Illustration)

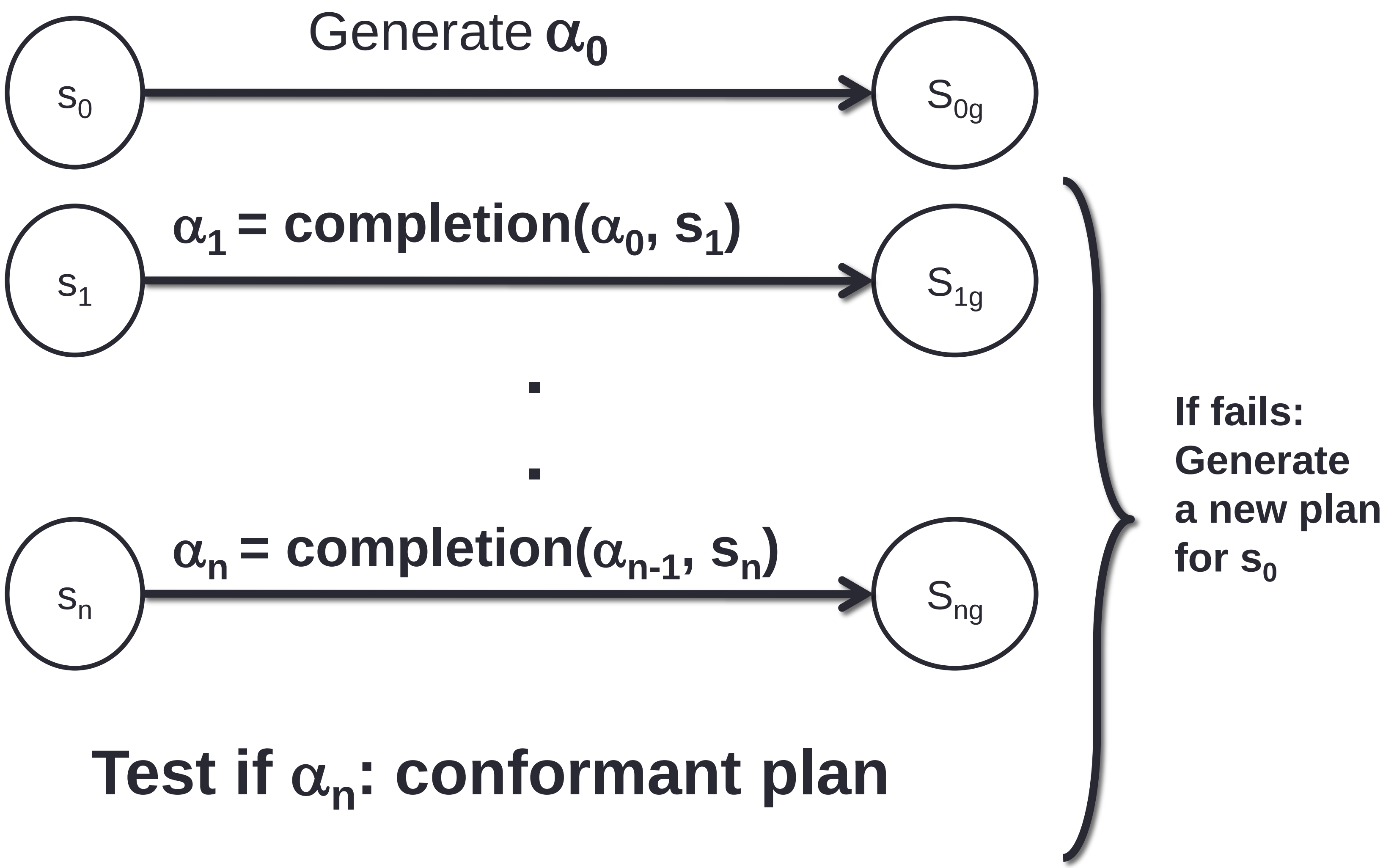


Generate and Complete (Illustration)



Test if α_n: conformant plan

Generate and complete approach



Test if α_n: conformant plan

Generate and Complete Approach

- Advantages:
  - Searching in state space instead of belief state space.
  - Divide conformant planning process into many classical planning phases.
  - Exploit the advantage of classical planners.
- Disadvantages:
  - Incomplete method?
    - Classical planners only return subsets of all possible plans.
    - Weak complete.

Result

- We implemented our algorithm on top of the classical planner LAMA2008. Our planner is named GC[LAMA].
- Compare GC[LAMA] to state-of-the-art conformant planners:
  - CpAh: Winner 2008
  - t0: Winner 2006
  - DNF
- Benchmarks include 777 instances from:
  - IPCs 2006 and 2008.
  - CFF
  - t0
  - Finite-State-Controllers
- Time out: 30 minutes

Result

- Coverage.

Planners	#Solved Instances/ #All Instances
GC[LAMA]	93%
DNF	58%
t0	50%
CpAh	46%

- GC[LAMA] solves many problems that no other planners were able to solve before.

Result (Overall)

Domain (#Instances)	GC[LAMA]	CpAh	t0	DNF
Blocks(4)	**4**	3	3	3
Coins(30)	**30**	20	20	20
Comm(25)	**25**	**25**	**25**	**25**
Sortnet(15)	**15**	**15**	9	**15**
Sortnum(30)	**25**	4	7	6
Uts(30)	**30**	**30**	**30**	**30**
Uts-cycle(30)	**15**	11	7	12
Raoskeys(30)	**3**	2	2	2
Forest(9)	**9**	2	8	2
Bomb(16)	**16**	15	**16**	15
Cube(35)	**35**	14	27	20

Result (IPC Domains) (Time/Length)

Instance	GC[LAMA]	CpAh	t0	DNF
Blw-03	**1.3**/266	20.4/205	48.51/80	307/325
Blw-04	**29.5**/1384	AB	AB	AB
Coins-10	0.037/36	**0.03**/48	0.04/26	0.20/27
Coins-30	**1.0**/1107	AB	AB	AB
Comm-15	0.1/97	2.29/95	**0.092**/110	3.43/125
Comm-25	**0.8**/294	1222/389	1.55/453	1797/501
Uts-cycle-03	0.04/3	**0.01**/3	0.14/3	0.01/3
Uts-cycle-15	**0.11**/12	AB	AB	AB
Raoskeys-02	0.05/38	0.26/32	**0.02**/21	0.09/39
Raokeys-04	**16.78**/163	AB	AB	AB
Forest-09	**183.8**/963	AB	AB	AB

(Time in seconds, AB means aborted due to insufficient memory)

Results (Finite-State Controllers Domain)

Instance	GC[LAMA]	CpAh	t0	DNF
Hall-A-ui-q2-1x10	**0.01**/46	39/77	1.09/56	0.29/58
Hall-A-ui-q2-1x20	**0.04**/86	AB	7.2/116	0.7/118
Hall-A-ui-q4-3x2	**0.05**/19	AB	TO	AB
Hall-A-ui-q4-4x2	**0.04**/42	AB	TO	AB
Hall-R-ui-q1-1x10	**0.01**/138	828/93	0.42/109	680/91
Hall-R-ui-q1-2x2	**0.0**/24	AB	TO	124/25
Hall-R-ui-q1-3x3	**0.01**/63	AB	TO	AB
Hall-R-ui-q1-4x4	**0.01**/110	AB	TO	AB
Marker-enc1-q2-5x4	**0.37**/25	AB	TO	124/25
Marker-enc1-q2-7x5	**11.1**/31	AB	TO	AB
Marker-enc1-q2-8x5	**11.5**/39	AB	TO	AB
Marker-enc1-q2-9x5	**0.02**/45	AB	TO	AB

(AB: abort, TO: timeout)

Conclusion

- We develop a novel approach to conformant planning:
  - Simple.
  - Exploit classical planners.
- We develop a conformant planner GC[LAMA] :
  - Very good coverage.
  - Excellent performance.

Future work

- Apply conformant planning to solve other problems:
  - Cellular automata
  - Contingent planning
- Investigate the ordering of possible initial worlds
- Completeness

Thank you for listening

Question?