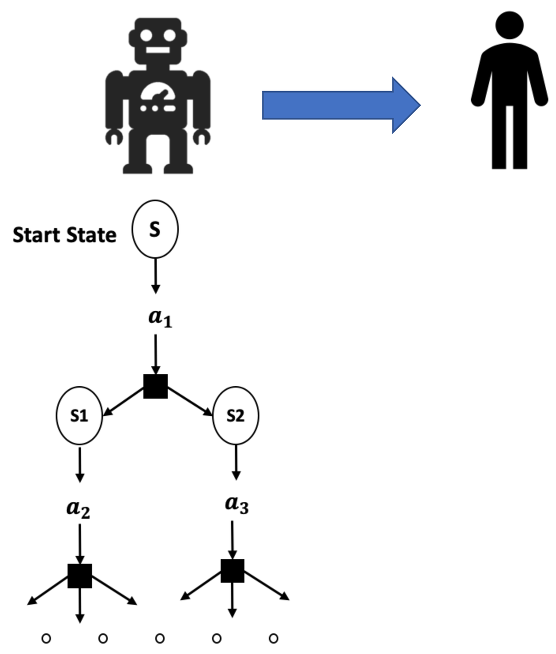
TLdR: Policy Summarization for Factored SSP Problems Using Temporal Abstractions

Sarath Sreedharan ▪ Siddharth Srivastava ▪ Subbarao Kambhampati

Collaborative Agents

- Ability to effectively collaborate with people is a prerequisite for real world agents
- We would expect such agents to be capable of effectively communicating their plans with users
- Particularly challenging in stochastic setting
    - Need to communicate policies



Existing Approaches

- Methods to simplify policy structure

Or

- **Summarize:**
    - Presenting state abstractions of the policy
    - Present actions from representative states

Our Method

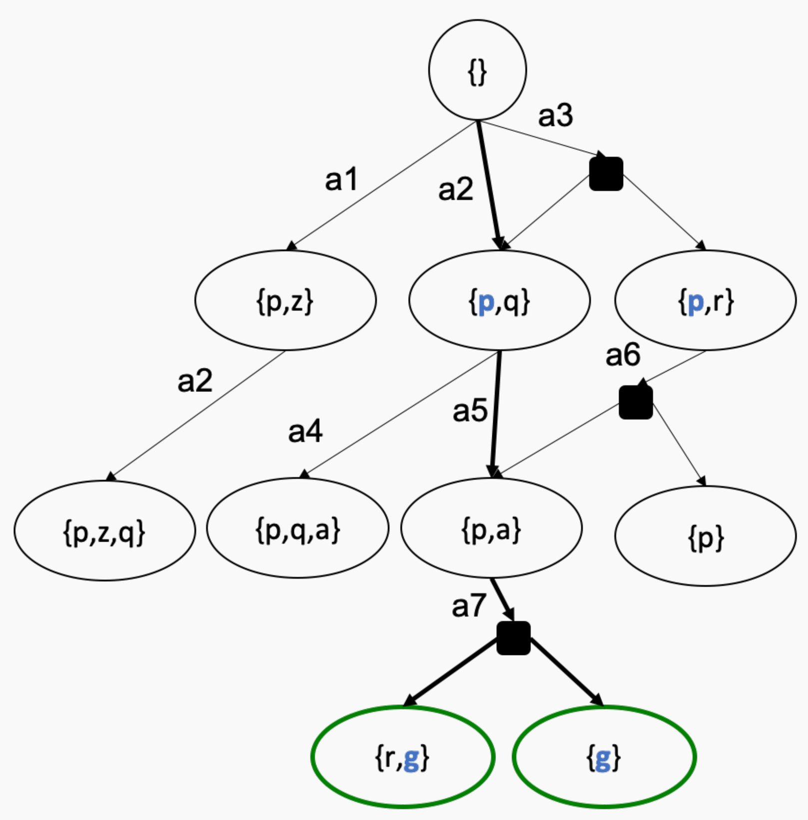
- Summary in terms of the subgoals to be achieved as part of the policy

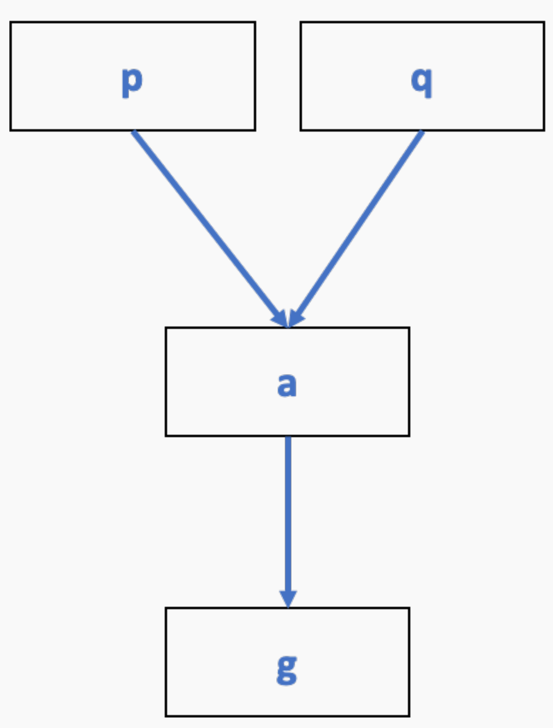
Challenge: Extract subgoals automatically for a given policy

Policy Landmark

- We focus on goal based MDP formulations with fixed initial states
- Subgoals identified by extracting landmarks
- Where landmarks in stochastic settings defined in terms of execution sequence that leads to the goal rather than in terms of plans
- For policy landmark we only consider the execution sequences feasible under the given policy

"Facts/formulae and their relative ordering that needs to be satisfied by every feasible trace possible under the policy that leads to the goal"





Extracting Landmarks

Convert it to landmark extraction for classical planning problem

- Constrain the model to only allow policy actions
- Generate an all outcome determinization
- Generate landmarks for determinized problem

- Further we can exploit the fact that most landmark extraction methods rely on delete relaxations – Consider a model with a simpler structure
    - Whose add effects are a union of all possible add effects

State: {(at_home), (uber_gift_card_1), (uber_gift_card_2)}
Action: walk_to_garage

Original Action:

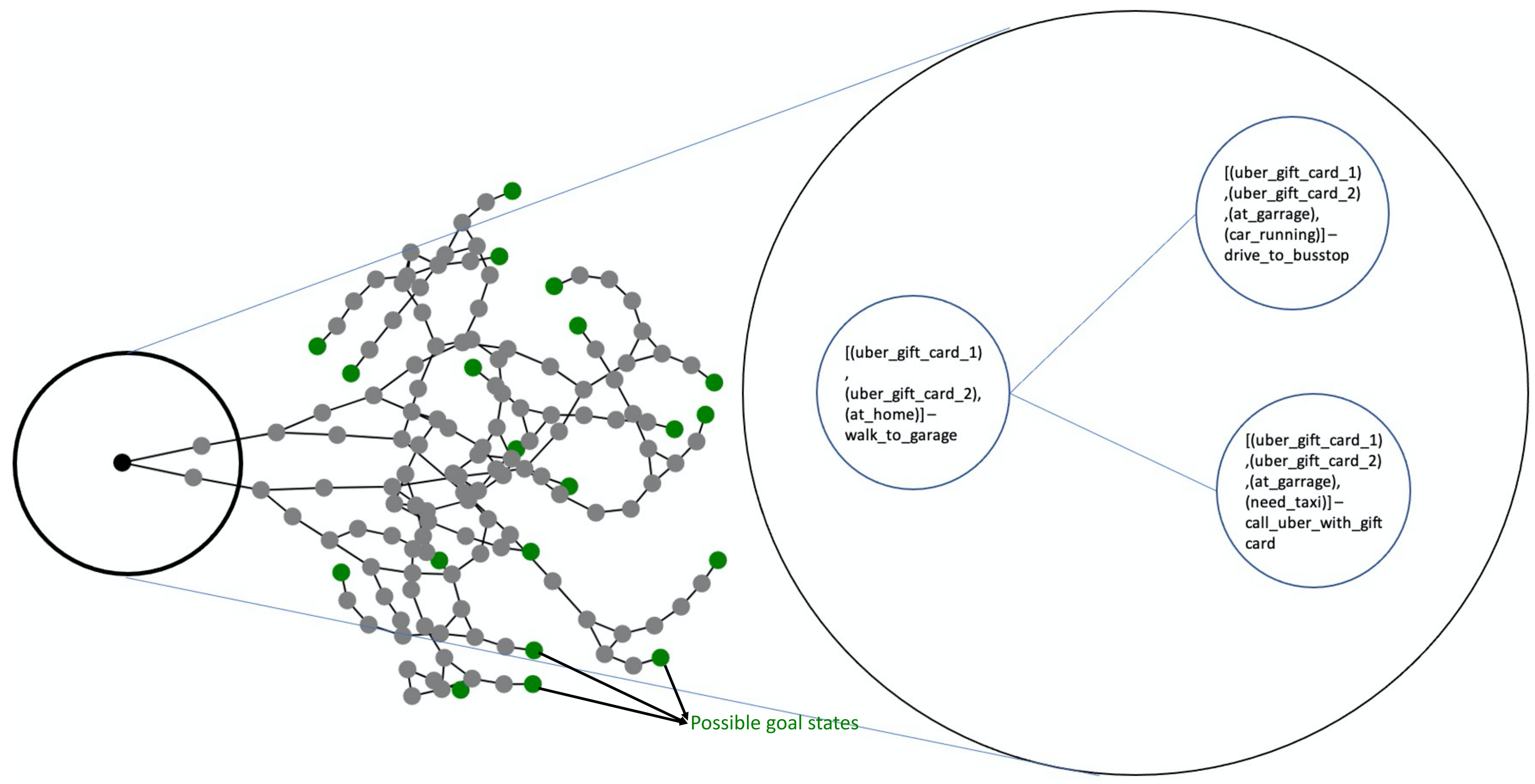
```
(:action walk_to_garage
:parameters ()
:precondition (and (at_home) )
:effect (and (at_garrage) (not (at_home))
              (probabilistic 1/2 (and (car_running))
                             1/2 (and
(need_taxi_to_busstop))))
)
```

Determinized Action:

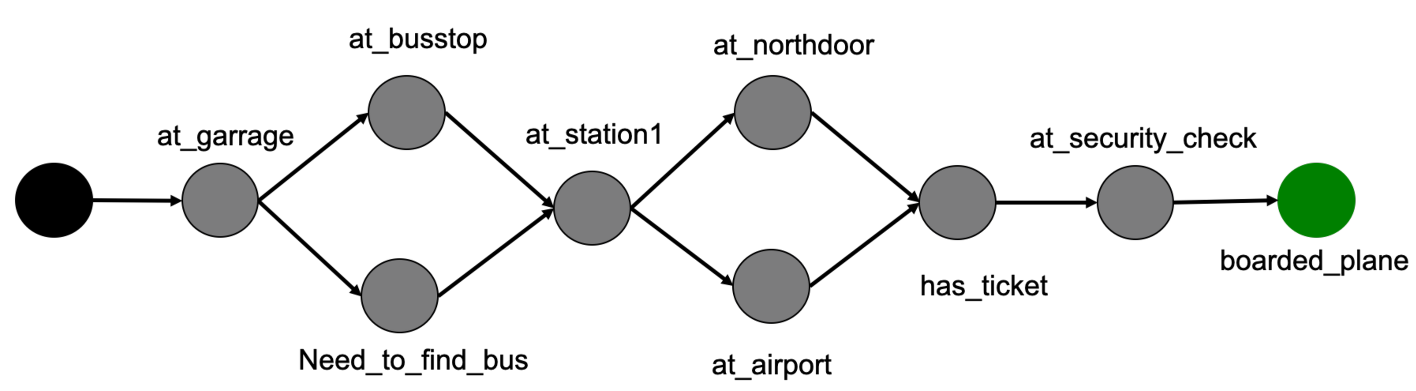
```
(:action ACT_1
:parameters ()
:precondition (and (at_home) (uber_gift_card_1)
                        (uber_gift_card_2))
:effect (and (at_garrage))
              (car_running)
             (need_taxi_to_busstop))
)
```

Example: Travel planning domain

- Consider a sample domain for generating a travel plan
- Domain consists of 30 grounded fluents and 27 grounded actions
- Partial policy consists of 100 states



Policy



Summary

Evaluation

User Studies

Hypothesis: Would people choose landmark facts as summary

- Users were presented with a policy graph with state details and asked to summarize based on subgoals
- 60 participants in total with 30 per scenario
- We made sure the participants understood the domain through a filtering question
- Every participant was shown 13 options and five to six facts out of which were landmarks
- We filtered 41 participant with 164 selections out of which 125 were landmarks (76.2%)

Empirical Evaluation

Domain	No of reachable states $\lvert \mathcal{R}^\pi(I) \rvert$	No of non-trivial policy landmarks $\lvert \Psi^\pi \rvert$	No of non-trivial model landmarks $\lvert \Psi^\mathcal{M} \rvert$
Ex-Blocksworld	104.8	8.8	5.2
Elevator	13.2	7	4
Tireworld	13.6	1.8	0.8
Tri-Tireworld	3973	6.25	0

Acknowledgement: This research is supported in part by ONR grants N00014-16-1-2892, N00014-18-1-2442, N00014-18-1-2840, N00014-9-1-2119, AFOSR grant FA9550-18-1-0067, DARPA SAIL-ON grant W911NF-19-2-0006, NSF grants 1936997 (C-ACCEL), 1844325 and 1909370, NASA grant NNX17AD06G, and a JP Morgan AI Faculty Research grant.