Practical Encodings of Factored Deterministic POMDPs into Probabilistic Planning

Patrik Haslum[1], Abdallah Saffidine[2]

[1]Australian National University, Canberra, Australia
[2] The University of New South Wales, Sydney, Australia

May 2, 2019

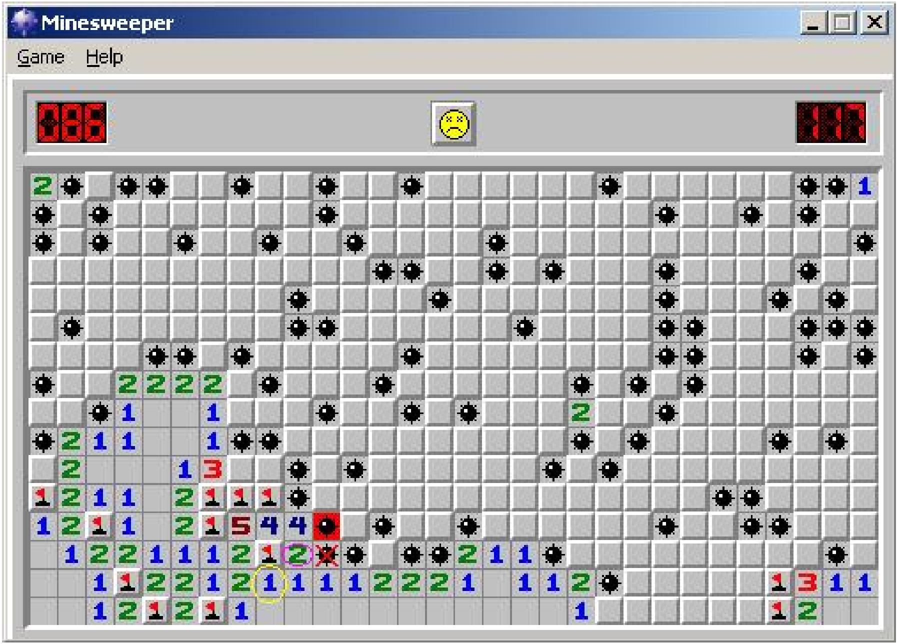

Player
CPU

A story about Chinese Dark Chess

A story about Chinese Dark Chess

Representation choices matter!

A Markov Decision Process


Minesweeper
Game
Help


MASTERMIND


reshuffle
19:44
Score: 0
change background

Some single-agent domain models

		Observability	
		Full	Partial
Control over state transitions	No	Markov Chain	Hidden Markov Model
	Yes	Markov Decision Process	Partially Observable MDP

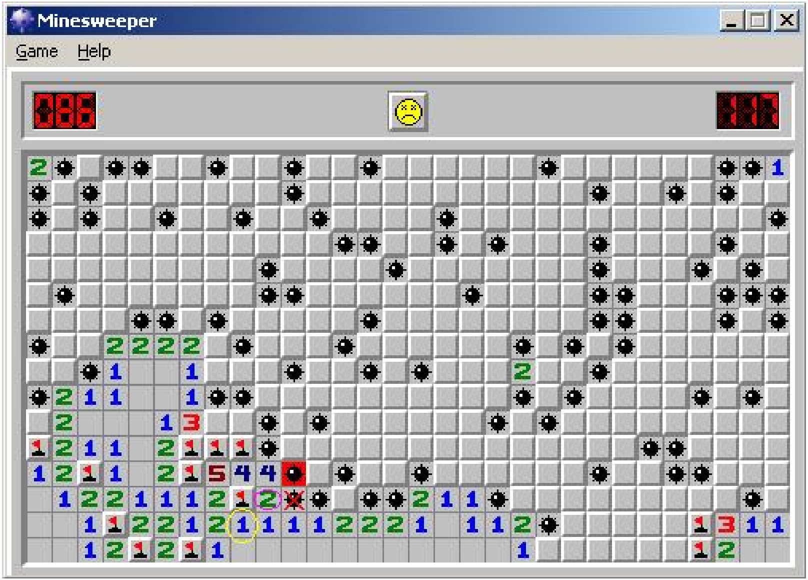
Some single-agent domain models

		Observability	
		Full	Partial
Control over state transitions	No	Markov Chain	Hidden Markov Model
	Yes	Markov Decision Process	Partially Observable MDP

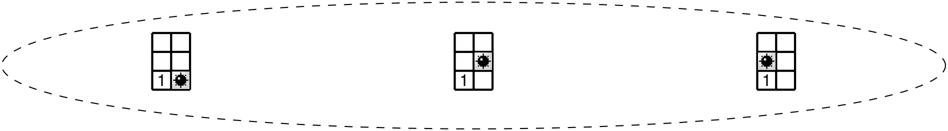
Today's concern

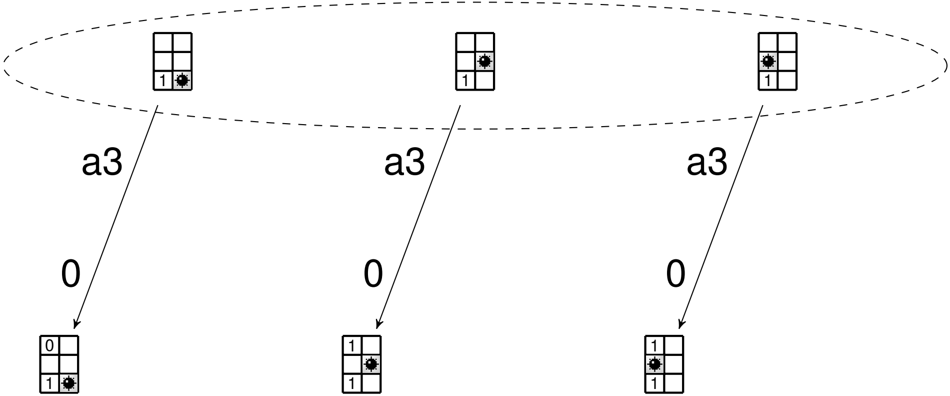
- deterministic POMDPs
- factored representations

A deterministic POMDP

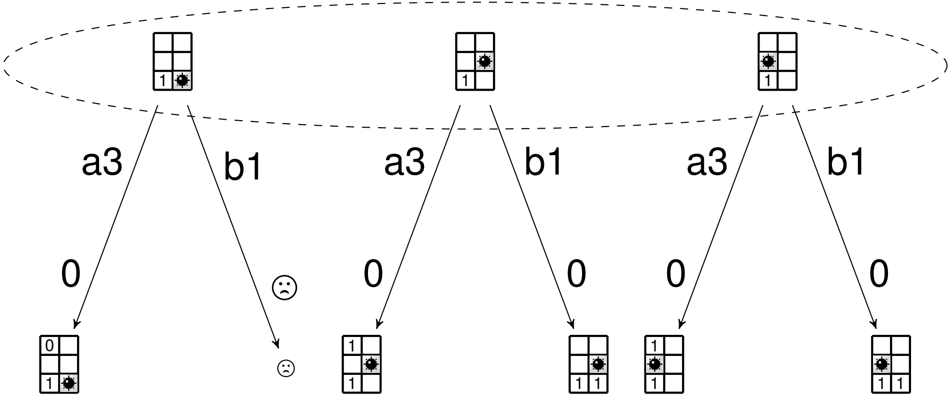


1	

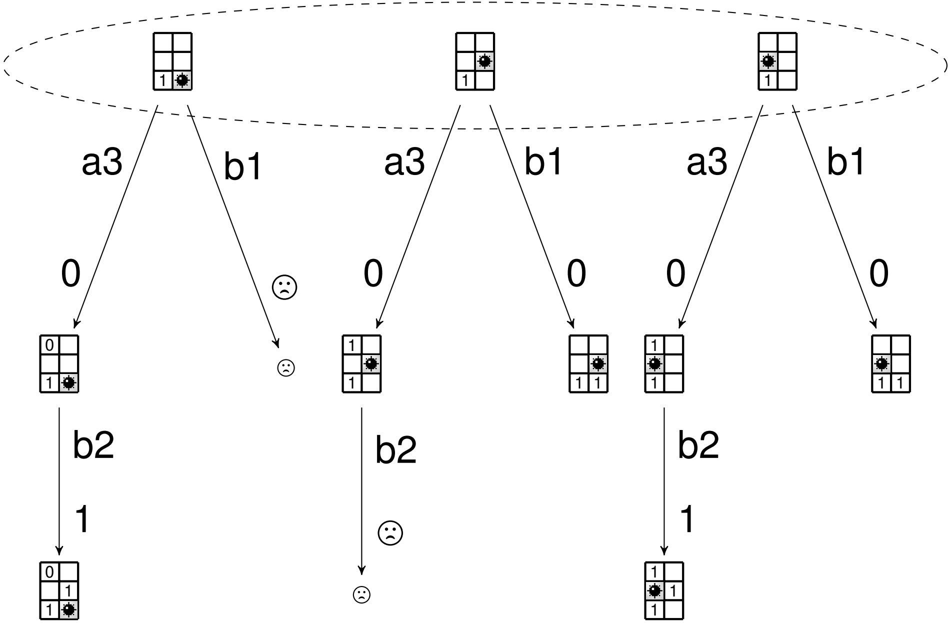

1
1
1



1
1
1
a3
a3
a3
0
0
0
0
1
1
1
1
1



1
1
1
a3
b1
a3
b1
a3
b1
0
☹
0
0
0
0
0
1
☹
1
1
1 1
1
1
1 1


a3
b1
a3
b1
a3
b1
0
0
0
0
0
b2
b2
b2
1
1

Reducing POMDPs to MDPs

- Transform a POMDP P into an MDP M_P
- With equivalent optimal values

Reducing POMDPs to MDPs

- Transform a POMDP P into an MDP M_P
- With equivalent optimal values
- Mapping policies for M_P to policies for P.

Reducing POMDPs to MDPs

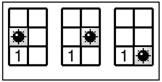
- Transform a POMDP P into an MDP M_P
- With equivalent optimal values
- Mapping policies for M_P to policies for P.
- $\rightarrow$ reuse mature MDP technology
- $\rightarrow$ provide complexity bounds

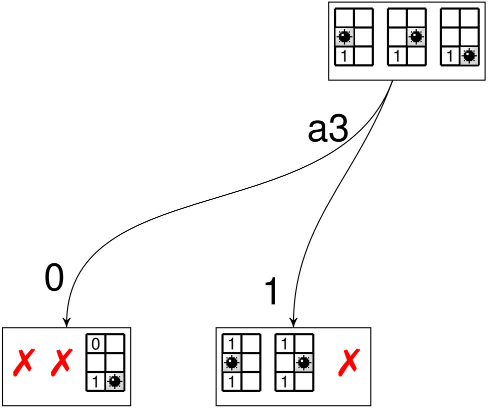
Littman's encoding (1996)

- Transform a det-POMDP P into an MDP L_P
- Each state of L_P is a table with one entry per state of P describing the evolution of that state.

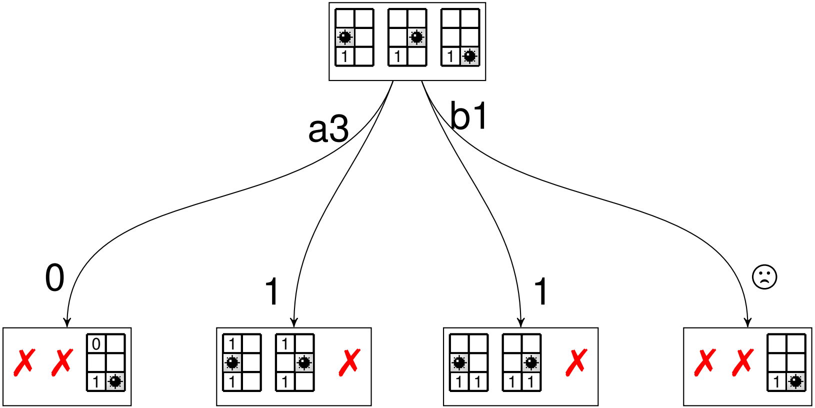
Littman's encoding (1996)

- Transform a det-POMDP P into an MDP L_P
- Each state of L_P is a table with one entry per state of P describing the evolution of that state.
- Σ states for $P \rightarrow \mathcal{O}((1+\Sigma)^{\Sigma})$ states for L_P

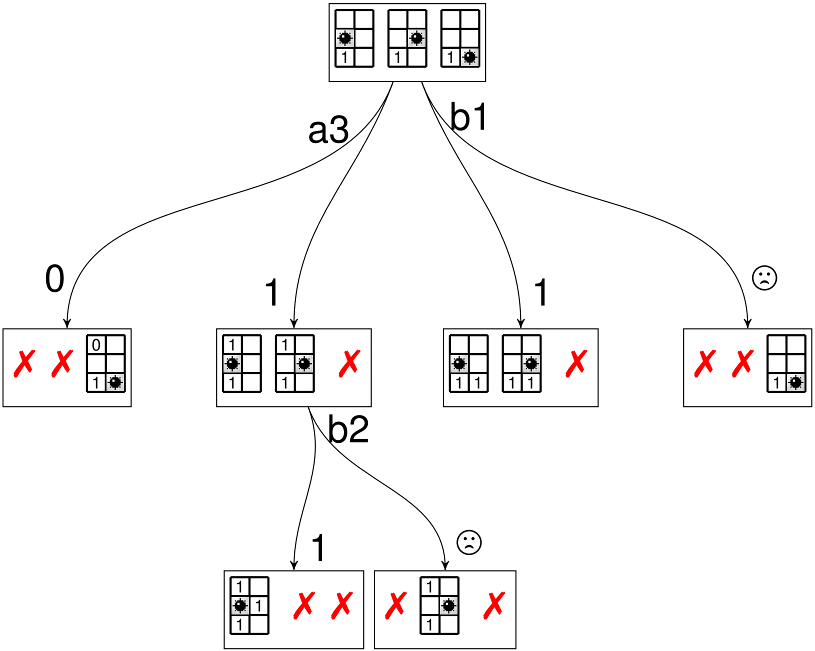

1
1
1



1
1
1
a3
0
1
0
1
1
1
1
1



a3
b1
0
1
1
☹


a3
b1
0
1
1
b2
1

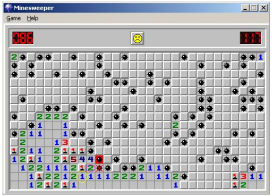
Littman's encoding

- Transform a det-POMDP P into an MDP L_P
- Each state of L_P is a table with one entry per state of P describing the evolution of that state.
- Σ states for $P \to \mathscr{O}((1+\Sigma)^{\Sigma})$ states for L_P

Littman's encoding

- Transform a det-POMDP P into an MDP L_P
- Each state of L_P is a table with one entry per state of P describing the evolution of that state.
- Σ states for $P \rightarrow \mathcal{O}((1+\Sigma)^{\Sigma})$ states for L_P

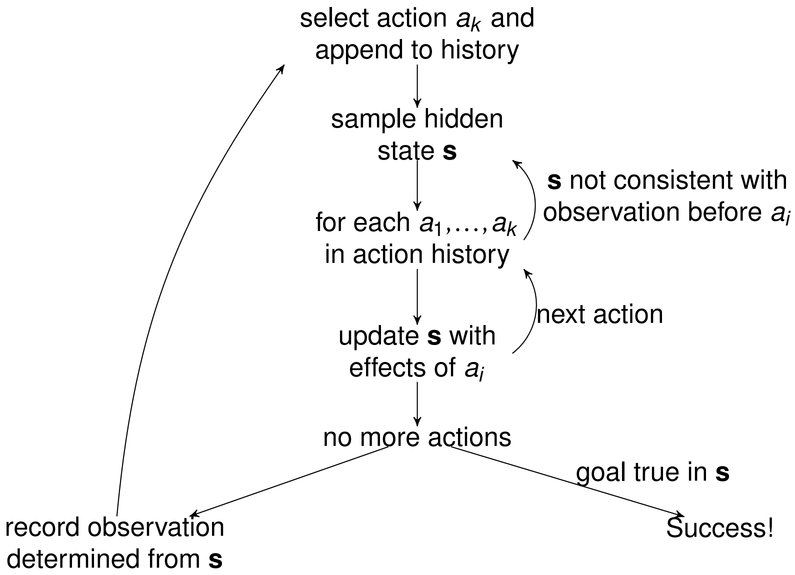
How satisfactory is that?

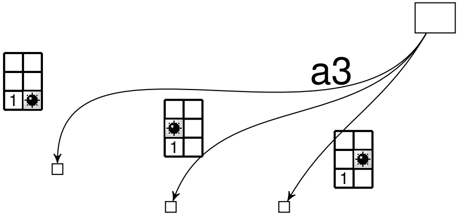


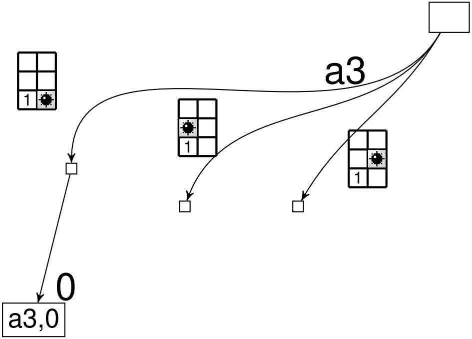
History-based encoding (new(?))

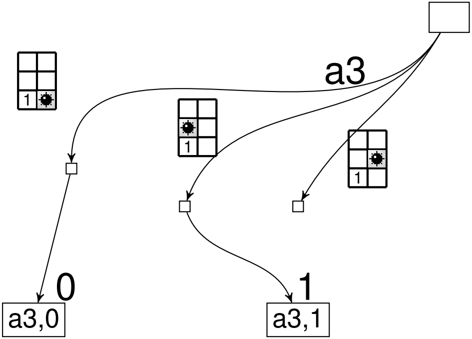
- Transform a det-POMDP P into an MDP H_P
- Each state of H_P is a table encoding the history of action/observation performed in P.

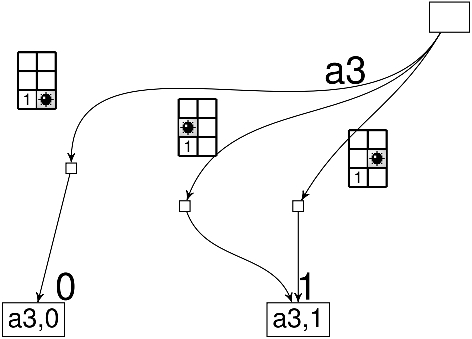
History-based encoding

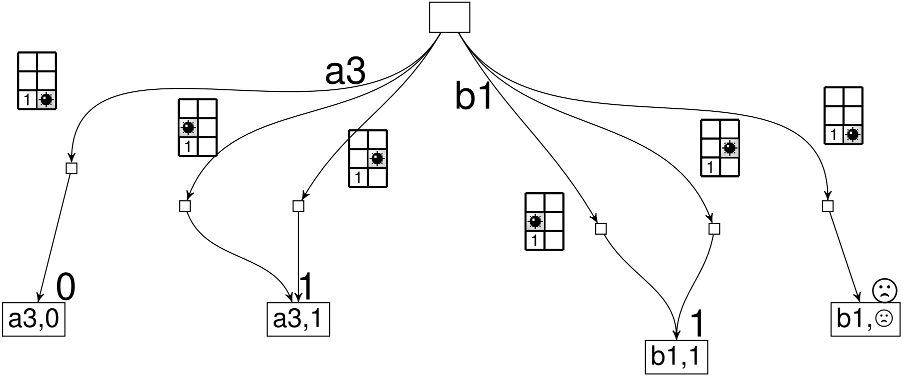


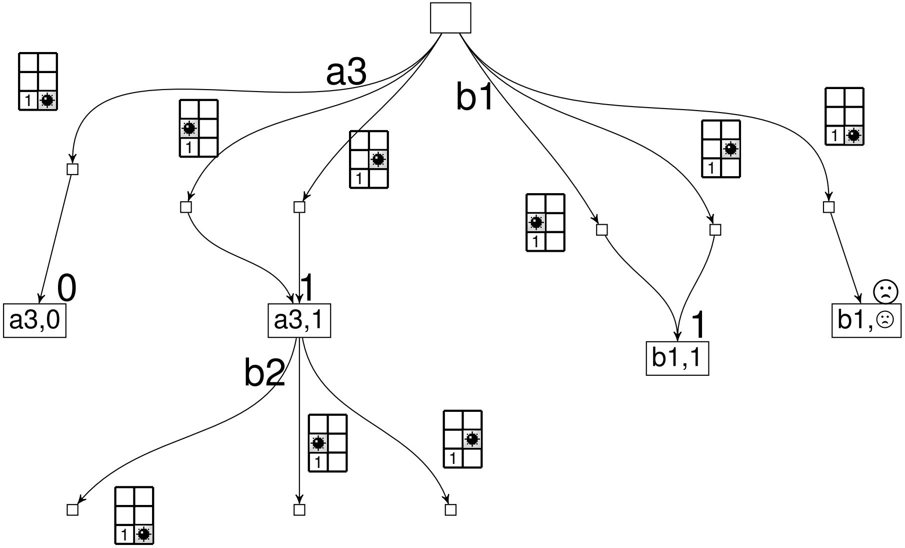

a3

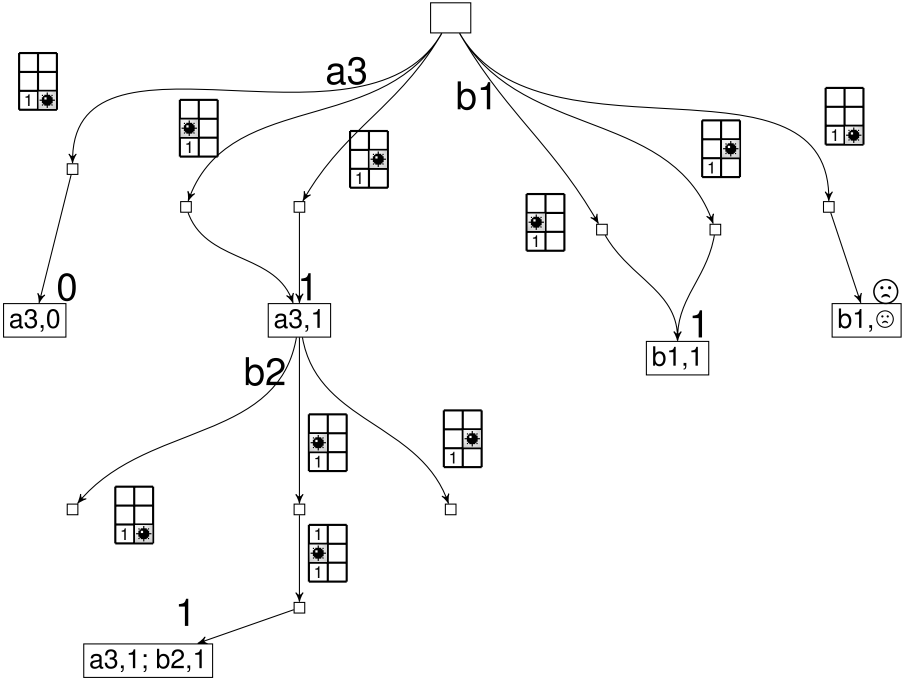

a3
1
1
1
0
a3,0


a3
1
1
1
0
1
a3,0
a3,1

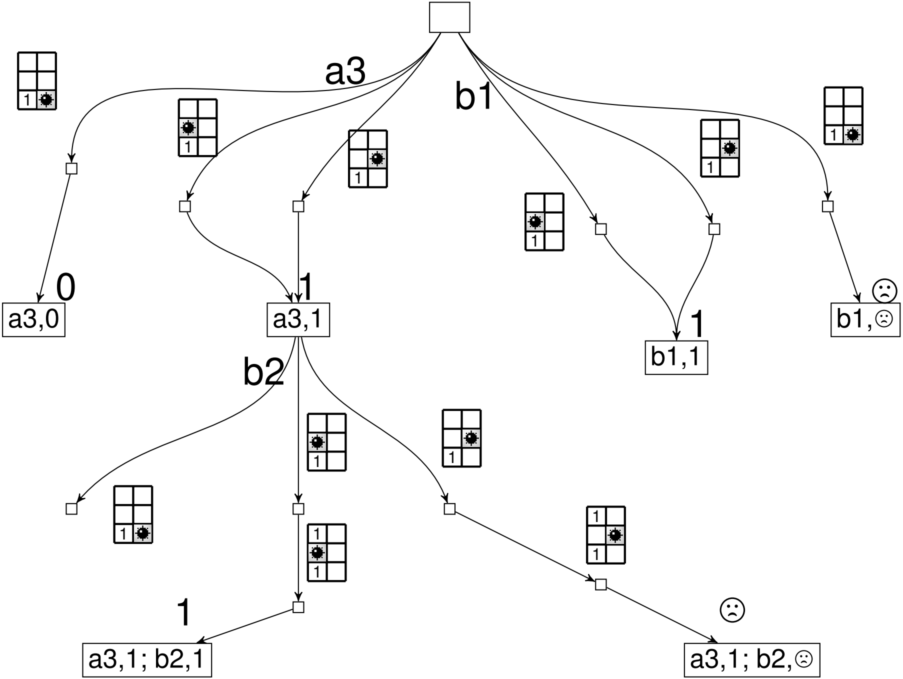

a3
1
1
1
0
1
a3,0
a3,1

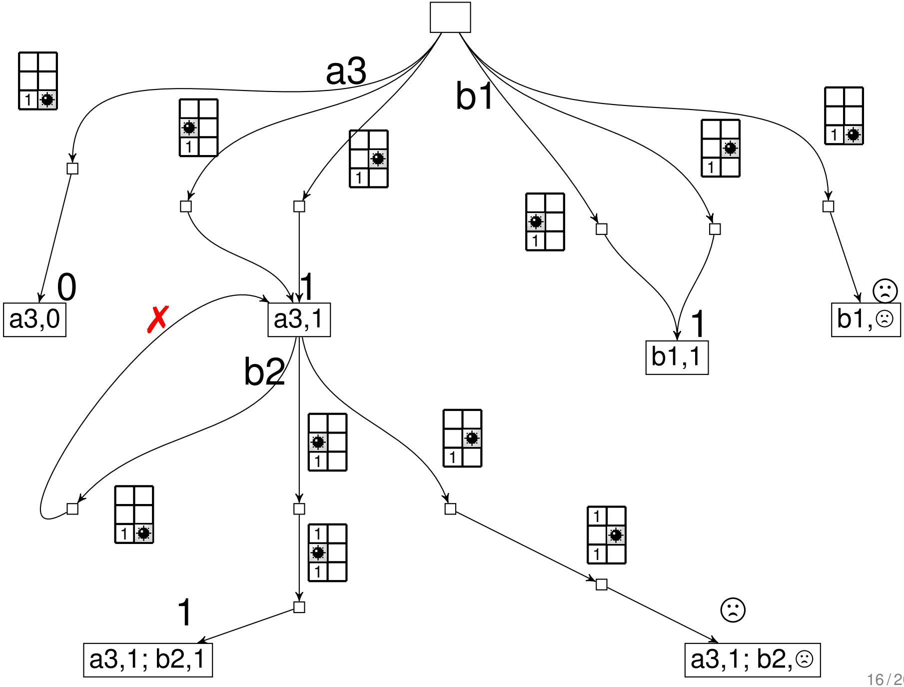

a3
b1
1
1
1
1
1
1
0
1
1
☹
a3,0
a3,1
b1,1
b1,☹


a3
b1
0
1
1
a3,0
a3,1
b1,1
b1,☹
b2



a3
b1
0
a3,0
1
a3,1
1
b1,1
☹
b1,☹
b2
1
a3,1; b2,1


a3
b1
0
a3,0
1
a3,1
1
b1,1
☹
b1,☹
b2
1
a3,1; b2,1
☹
a3,1; b2,☹


a3
b1
0
a3,0
✗
1
a3,1
1
b1,1
☹
b1,☹
b2
1
a3,1; b2,1
☹
a3,1; b2,☹

History-based encoding

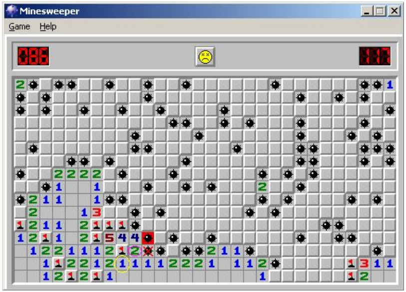
- Transform a det-POMDP P into an MDP H_P
- Each state of H_P is a table encoding the history of action/observation performed in P.
- A actions-observations for P, horizon $H \rightarrow \mathcal{O}((A)^{H+c})$ states for H_P

Size considerations

Repr.	Encoding	Number of States
Flat	POMDP	Σ
Flat	MDP: Littman	$(1+\Sigma)^{\Sigma}$
Flat	MDP: History	A^{H+c}
Factored	POMDP	2^V
Factored	MDP: Littman	$(1+2^V)^{2^V}$
Factored	MDP: History	A^{H+c}

A actions and observations, V state variables, c small constant
Σ states, H horizon

Det-POMDPs with polynomial depth



Sanity check: we reprove PSPACE membership.

Conclusion

Discussion point

- Factored vs explicit representations

Conclusion

Discussion point

- Factored vs explicit representations

Limitations → won't fix

- Deterministic effects
- No adversary

Conclusion

Discussion point

- Factored vs explicit representations

Limitations → won't fix

- Deterministic effects
- No adversary

Future Work

- Extend to discounted rewards
- Experiments in Troubleshooting domains
- Check the literature on the complexity of factored POMDPs