Abstractions in Probabilistic Planning

Maximilian Grüner

Computer Science Department, University of Basel

February 8, 2016

Overview

1. Introduction
2. Create an Abstraction
3. Make use of an Abstraction
4. Results
5. Outlook

What does the title mean

Probabilistic Planning Task

Idea

We have a system where the following holds:

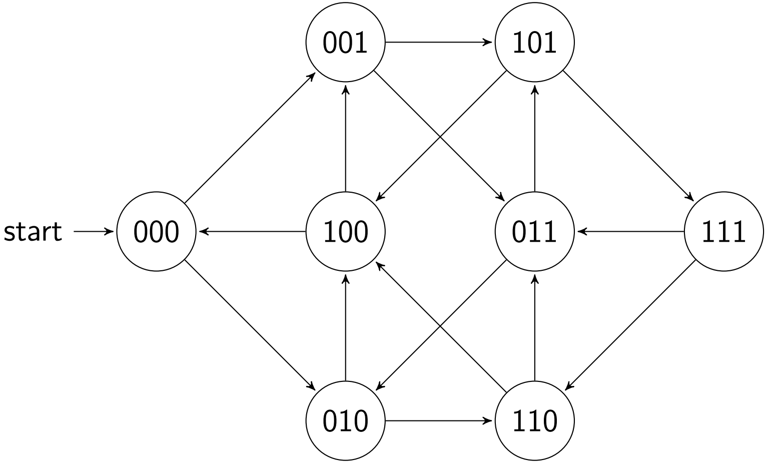
- States
- Transitions between states
- Transitions can be initiated via actions
- Transition are probabilistic

Probabilistic Planning Task

Idea (continuation)

- A formula indicates the optimal way through the system
- The number of actions we can take in a game are limited
- There is a start state

Probabilistic Planning Task

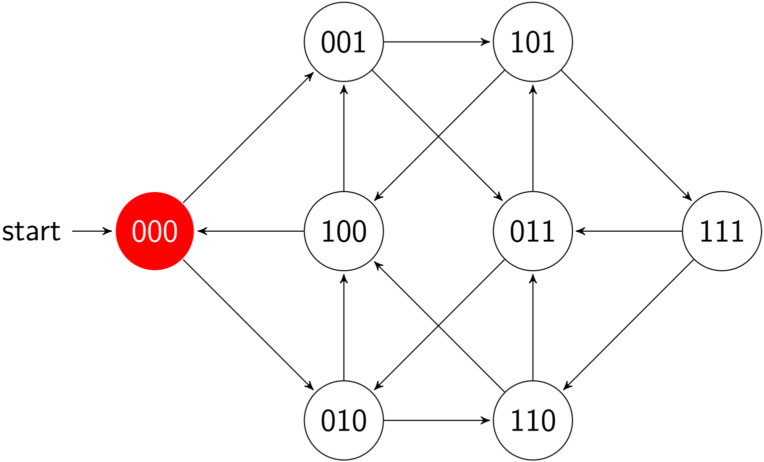


Probabilistic Planning Task

$r(s) = ([v_1]^I + [v_2]^I) \cdot [v_3]^I$

Reward:0

Step:0/3

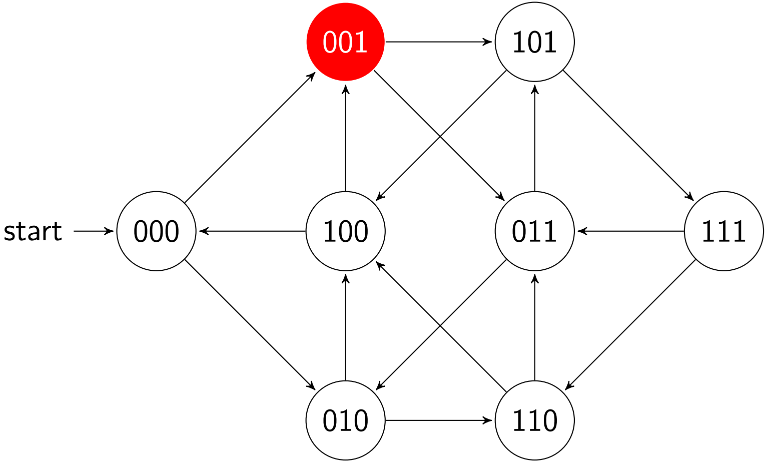


Probabilistic Planning Task

$(0 + 0) \cdot 0 = 0$

Reward:0

Step:1/3

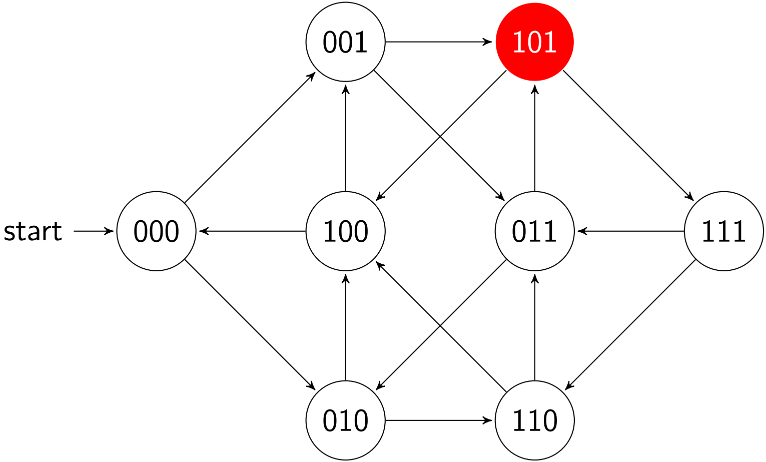


Probabilistic Planning Task

$(0 + 0) \cdot 1 = 0$

Reward:0

Step:2/3

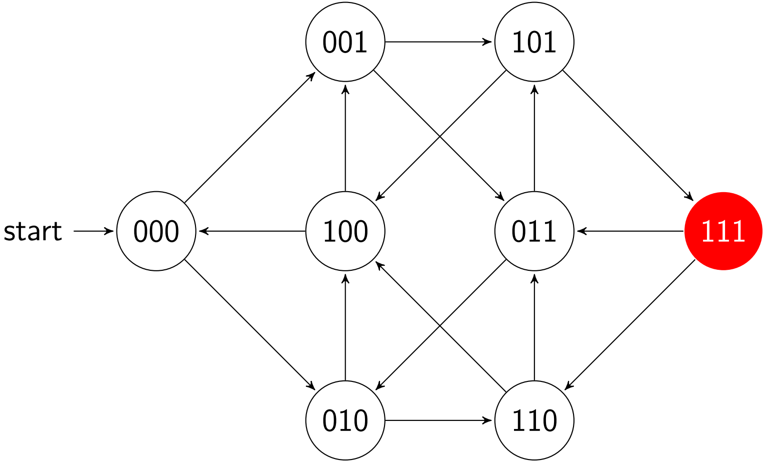


Probabilistic Planning Task

$(1 + 0) \cdot 1 = 1$

Reward:1

Step:3/3



Probabilistic Planning Task

There is a problem

Abstraction

Idea

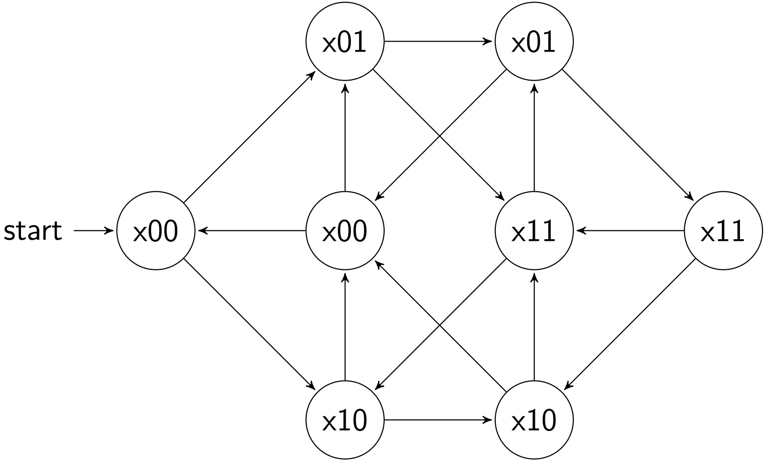
We have some function which limits the following:

- The original pattern
- The original transitions
- The original reward formulas
- The original starting point

Abstraction

Summarize with Pattern

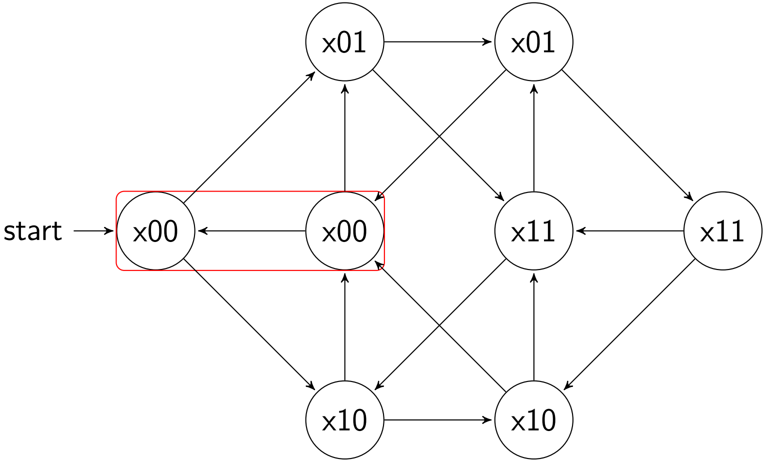
$P = \{v_2, v_3\}$



Abstraction

Summarize with Pattern

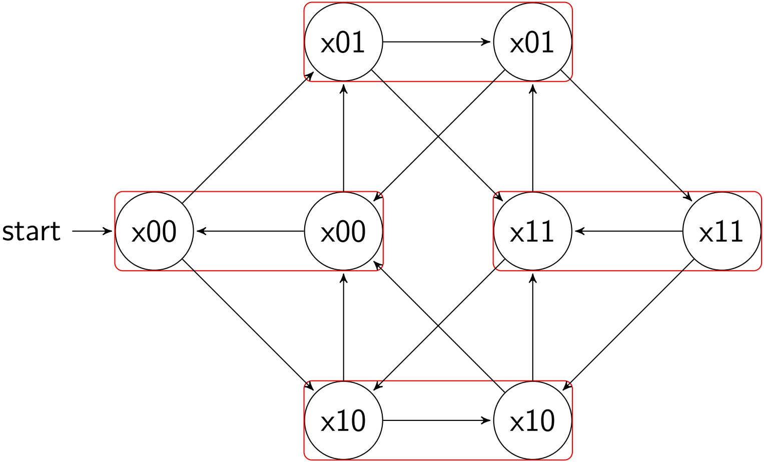
$P = \{v_2, v_3\}$



Abstraction

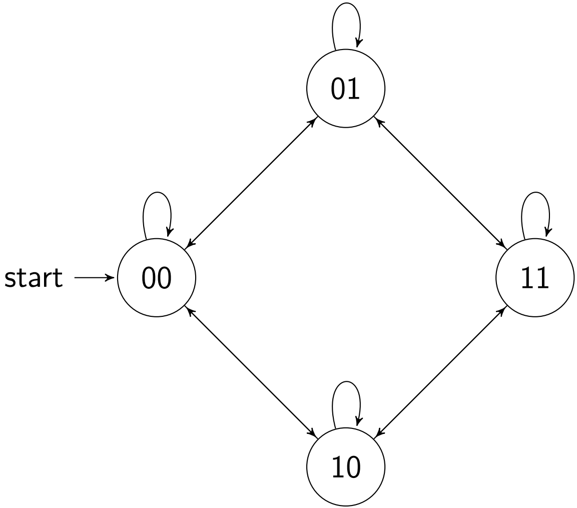
Summarize with Pattern

$P = \{v_2, v_3\}$



Abstractions

Result



Recapitulation

Probabilistic planning tasks can't in general be solved directly and we need to use more intelligent approaches to deal with them. One of these approaches is to create and abstraction.

What did I do

Create a Pattern

Boutilier and Dearden

- A robot should get coffee and take an umbrella with it if it rains.
- The reward is highest if it arrives dry with coffee and lowest if it arrives wet and without coffee.
- The robot gets coffee and ignores the fact whether it rains or not.

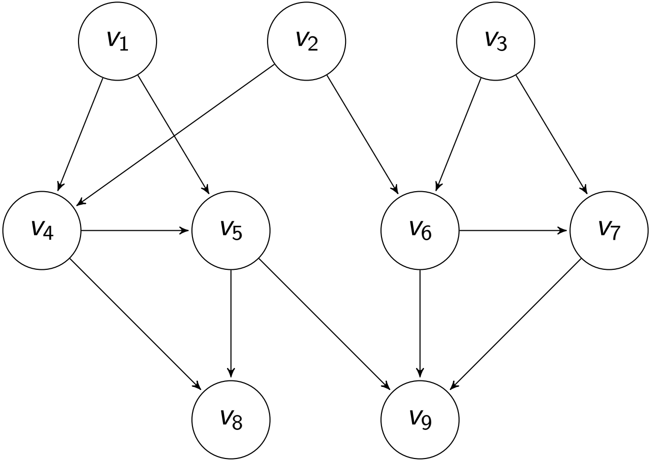
Create a Pattern

Dependency Graph

A variable is dependant on another variable if the probability of it being true or false depends on the aforementioned.

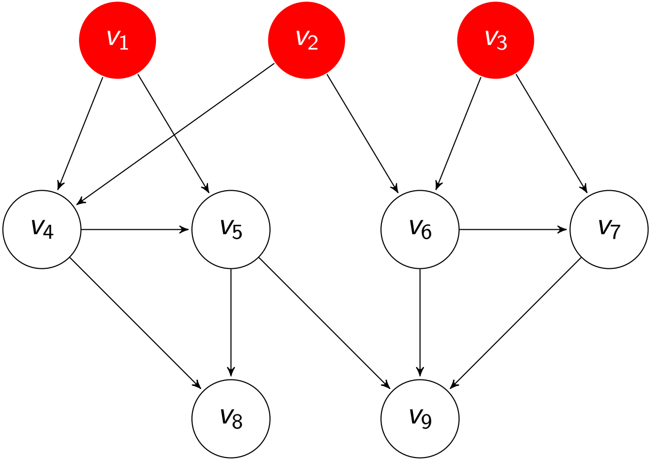
Create a Pattern

Dependency Graph



Create a Pattern

Dependency Graph



Create a Pattern

Determine Initial Set, Boutilier and Dearden

$$r(s) = ([v_1]^I + [v_2]^I) \cdot [v_3]^I$$

Patterns	Span
$\{v_1, v_2, v_3\}$	
$\{v_2, v_3\}$	
$\{v_1, v_2\}$	
$\{v_1, v_3\}$	
$\{v_3\}$	
$\{v_2\}$	
$\{v_1\}$	
$\{\}$	

Create a Pattern

Determine Initial Set, Boutilier and Dearden

$$r(s) = ([v_1]^I + [v_2]^I) \cdot [v_3]^I$$

Patterns	Span
$\{v_1, v_2, v_3\}$	
$\{v_2, v_3\}$	
$\{v_1, v_2\}$	
$\{v_1, v_3\}$	
$\{v_3\}$	
$\{v_2\}$	
$\{v_1\}$	
$\{\}$	

Create a Pattern

Determine Initial Set, Boutilier and Dearden

$$r(s) = ([v_1]^I + [v_2]^I) \cdot [v_3]^I$$

Patterns	Span
$\{v_1, v_2, v_3\}$	
$\{v_2, v_3\}$	
$\{v_1, v_2\}$	
$\{v_1, v_3\}$	
$\{v_3\}$	
$\{v_2\}$	
$\{v_1\}$	
$\{\}$	

$$\max(([v_1]^I + [v_2]^I) \cdot 1)$$
$$= 2$$

Create a Pattern

Determine Initial Set, Boutilier and Dearden

$$r(s) = ([v_1]^I + [v_2]^I) \cdot [v_3]^I$$

Patterns	Span
$\{v_1, v_2, v_3\}$	
$\{v_2, v_3\}$	
$\{v_1, v_2\}$	
$\{v_1, v_3\}$	
$\{v_3\}$	
$\{v_2\}$	
$\{v_1\}$	
$\{\}$	

$\max(([v_1]^I + [v_2]^I) \cdot 1)$

$= 2$

$\min(([v_1]^I + [v_2]^I) \cdot 1)$

$= 0$

Create a Pattern

Determine Initial Set, Boutilier and Dearden

$$r(s) = ([v_1]^I + [v_2]^I) \cdot [v_3]^I$$

Patterns	Span
$\{v_1, v_2, v_3\}$	
$\{v_2, v_3\}$	
$\{v_1, v_2\}$	
$\{v_1, v_3\}$	
$\{v_3\}$	2
$\{v_2\}$	
$\{v_1\}$	
$\{\}$	

$$\max(([v_1]^I + [v_2]^I) \cdot 1)$$

$$= 2$$

$$\min(([v_1]^I + [v_2]^I) \cdot 1)$$

$$= 0$$

$$span_{v_3=true} = 2$$

$$span_{v_3=false} = 0$$

$$\max(span_{v_3=true}, span_{v_3=false}) = 2$$

Create a Pattern

Determine Initial Set, Boutilier and Dearden

$$r(s) = ([v_1]^I + [v_2]^I) \cdot [v_3]^I$$

Patterns	Span
$\{v_1, v_2, v_3\}$	0
$\{v_2, v_3\}$	1
$\{v_1, v_2\}$	2
$\{v_1, v_3\}$	1
$\{v_3\}$	2
$\{v_2\}$	2
$\{v_1\}$	2
$\{\}$	2

$$\max(([v_1]^I + [v_2]^I) \cdot 1)$$

$$= 2$$

$$\min(([v_1]^I + [v_2]^I) \cdot 1)$$

$$= 0$$

$$span_{v_3=true} = 2$$

$$span_{v_3=false} = 0$$

$$\max(span_{v_3=true}, span_{v_3=false}) = 2$$

Create a Pattern

Determine Initial Set, Our Method

$$r(s) = ([v_1]^l + [v_2]^l) \cdot [v_3]^l$$

Patterns	Span
$\{v_1, v_2, v_3\}$	
$\{v_2, v_3\}$	
$\{v_1, v_2\}$	
$\{v_1, v_3\}$	
$\{v_3\}$	
$\{v_2\}$	
$\{v_1\}$	
$\{\}$	

Create a Pattern

Determine Initial Set, Our Method

$$r(s) = ([v_1]^I + [v_2]^I) \cdot [v_3]^I$$

Patterns	Span
$\{v_1, v_2, v_3\}$	
$\{v_2, v_3\}$	
$\{v_1, v_2\}$	
$\{v_1, v_3\}$	
$\{v_3\}$	
$\{v_2\}$	
$\{v_1\}$	
$\{\}$	

$$\frac{span_{v_3=true} + span_{v_3=false}}{2} = 1$$

Create a Pattern

Determine Initial Set, Our Method

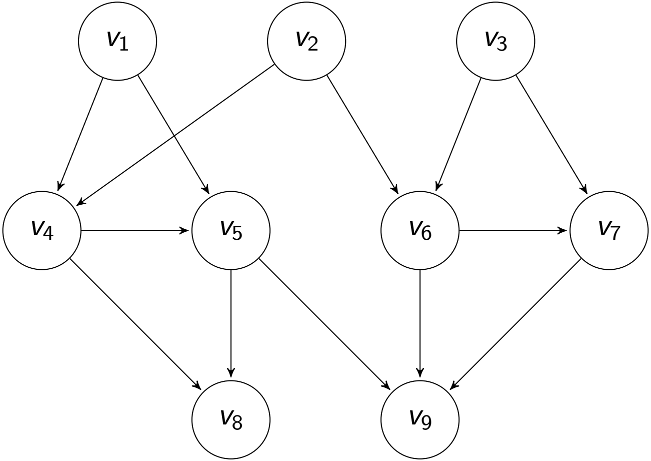
$$r(s) = ([v_1]^I + [v_2]^I) \cdot [v_3]^I$$

Patterns	Span
$\{v_1, v_2, v_3\}$	0
$\{v_2, v_3\}$	0.5
$\{v_1, v_2\}$	1
$\{v_1, v_3\}$	0.5
$\{v_3\}$	1
$\{v_2\}$	1.5
$\{v_1\}$	1.5
$\{\}$	2

$$\frac{span_{v_3=true} + span_{v_3=false}}{2} = 1$$

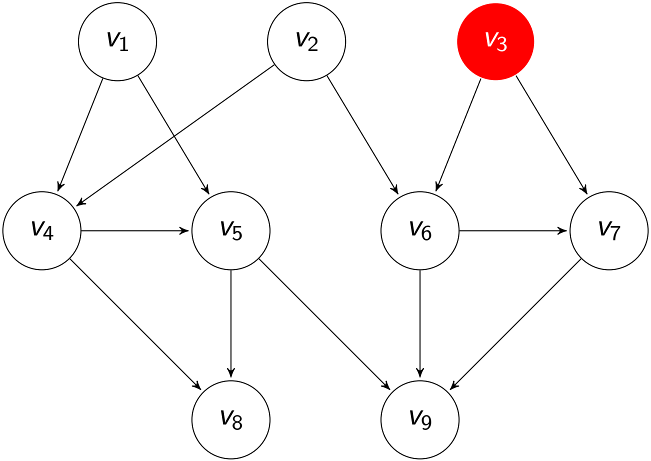
Create a Pattern

Search the Graph, Boutilier and Dearden



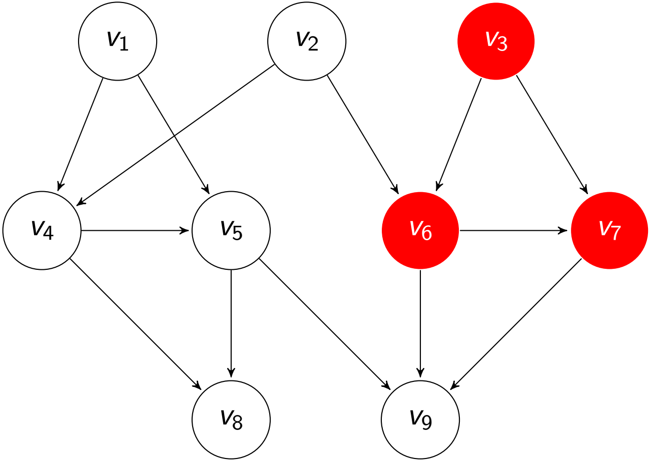
Create a Pattern

Search the Graph, Boutilier and Dearden



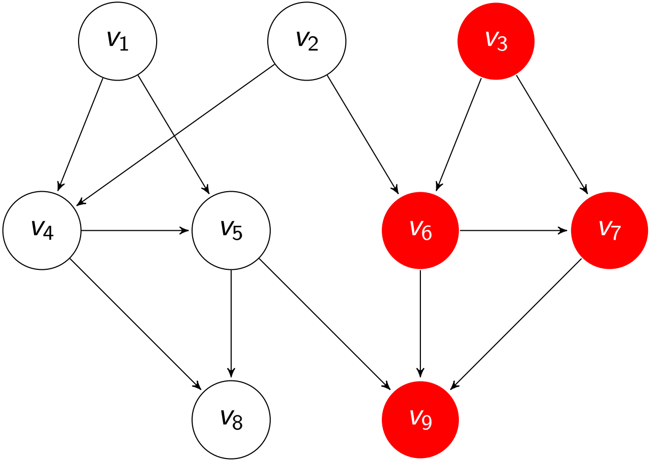
Create a Pattern

Search the Graph, Boutilier and Dearden



Create a Pattern

Search the Graph, Boutilier and Dearden



Create a Pattern

Structure of the Transition Formulas

Let's look at the transition formula for v_3

Create a Pattern

Structure of the Transition Formulas

Let's look at the transition formula for v_3

$$\begin{aligned} \text{If } v_6 &: 1 \\ \text{elif } v_7 &: 0.5 \\ \text{else } &0.3 \end{aligned}$$

Create a Pattern

Structure of the Transition Formulas

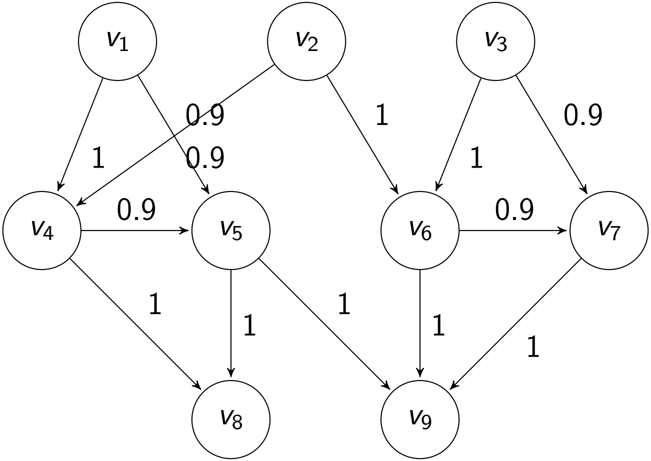
Let's look at the transition formula for v_3

$$\begin{aligned} \text{If } v_6 &: 1 \\ \text{elif } v_7 &: 0.5 \\ \text{else } &0.3 \end{aligned}$$

v_7 **only** matters if v_6 is false. The weight depends on the position.

Create a Pattern

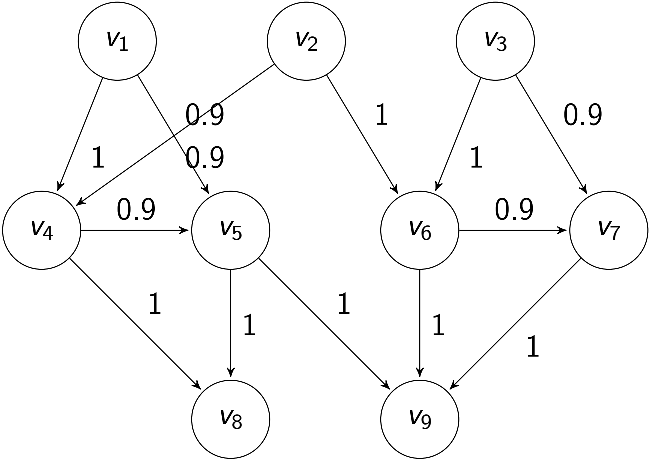
Search the Graph, Our Method



Create a Pattern

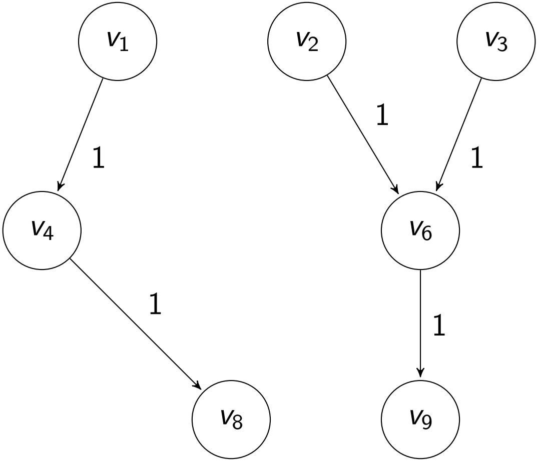
Search the Graph, Our Method

Remove all edges which weigh ≤ 0.9 and unnecessary nodes



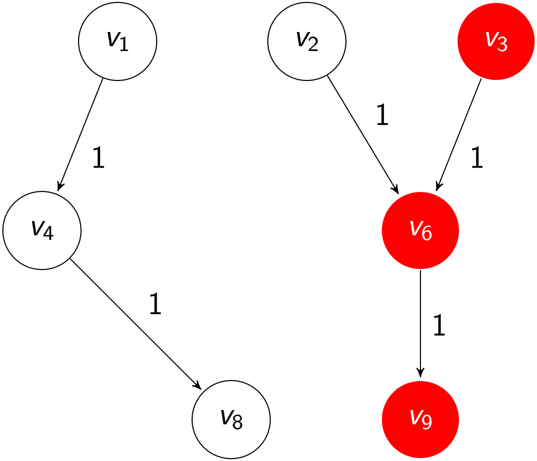
Create a Pattern

Search the Graph, Our Method



Create a Pattern

Search the Graph, Our Method



How do we solve this reduced task

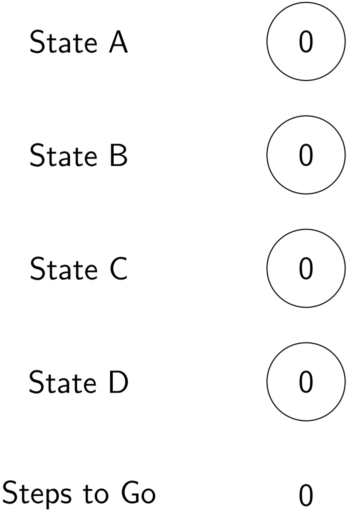
Heuristics

Idea

- We want to select the best set of actions
- We need to estimate the quality of a state
- The estimate needs to depend on the state on the steps left to go

Heuristics

Value Iteration



State A 0

State B 0

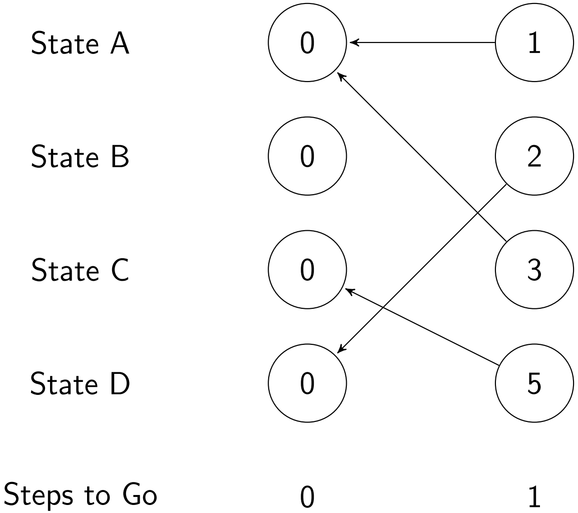
State C 0

State D 0

Steps to Go 0

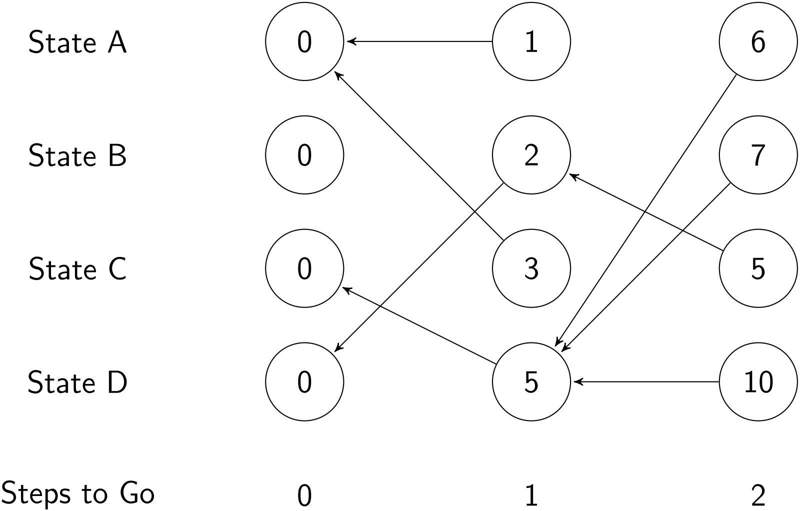
Heuristics

Value Iteration



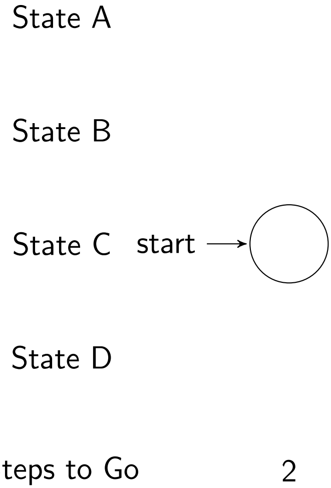
Heuristics

Value Iteration



State A	0	1	6
State B	0	2	7
State C	0	3	5
State D	0	5	10
Steps to Go	0	1	2

Let's invert it



State A

State B

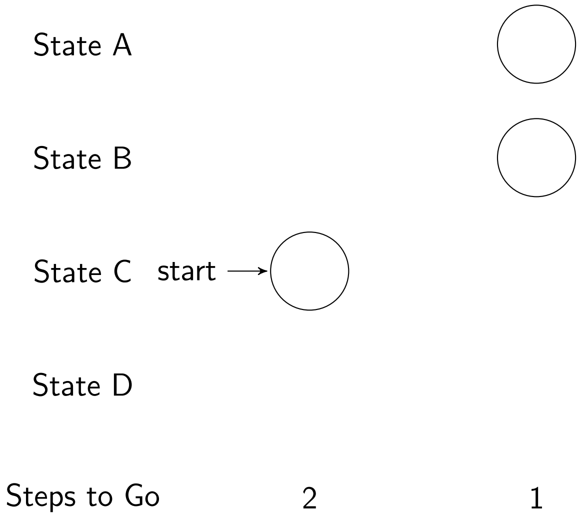
State C start

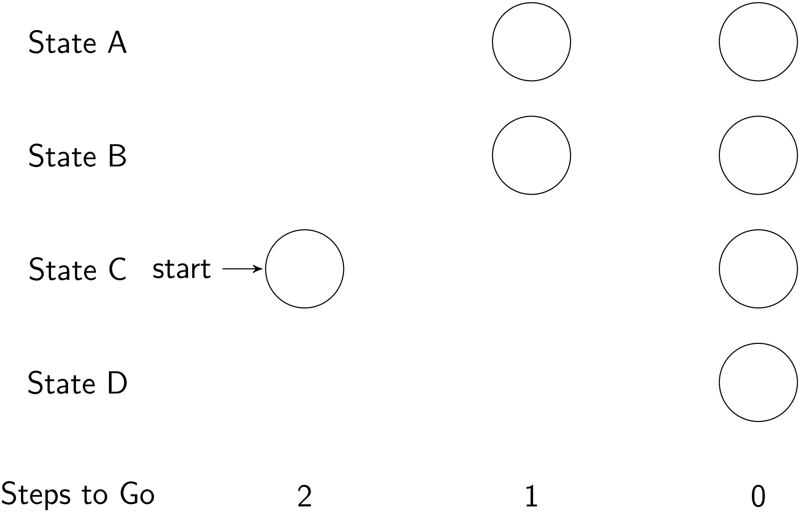
State D

Steps to Go 2

Heuristics

Value Iteration





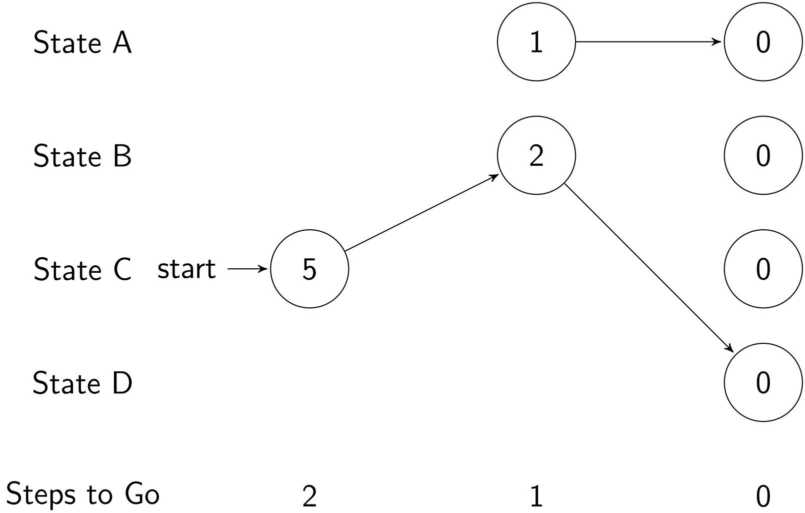
State A

State B

State C start

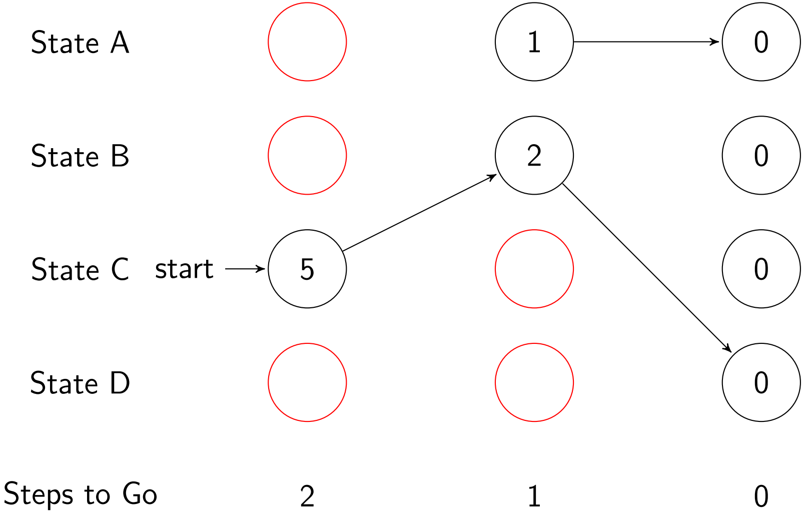
State D

Steps to Go 2 1 0


State A
State B
State C
State D
start
5
1
2
0
0
0
0
Steps to Go
2
1
0

Heuristics

Value Iteration



State A

State B

State C

State D

start

Steps to Go

What did I calculate

Results

	wildfire	triangle	academic	elevators	tamarisk	sysadmin	recon	game	traffic	crossing	skill	navigation	Total
UCTStar [IDS]	0.75	0.8	0.32	0.9	1.0	1.0	1.0	0.99	1.0	1.0	0.99	0.99	0.9
Standalone	0.91	0.2	0.46	0.29	0.64	0.82	0.0	0.63	0.0	0.57	0.44	0.31	0.44
Heuristic	0.35	0.63	0.39	0.34	0.14	0.57	0.96	0.54	0.34	0.4	0.48	0.95	0.51

Results

Good Patterns

	wildfire	triangle	academic	elevators	tamarisk	sysadmin	recon	game	traffic	crossing	skill	navigation	Total
UCTStar [IDS]	0.75	0.8	0.32	0.9	1.0	1.0	1.0	0.99	1.0	1.0	0.99	0.99	0.9
Standalone	0.91	0.2	0.46	0.29	0.64	0.82	0.0	0.63	0.0	0.57	0.44	0.31	0.44
Heuristic	0.35	0.63	0.39	0.34	0.14	0.57	0.96	0.54	0.34	0.4	0.48	0.95	0.51

Results

Bad Patterns

	wildfire	triangle	academic	elevators	tamarisk	sysadmin	recon	game	traffic	crossing	skill	navigation	Total
UCTStar [IDS]	0.75	0.8	0.32	0.9	1.0	1.0	1.0	0.99	1.0	1.0	0.99	0.99	0.9
Standalone	0.91	0.2	0.46	0.29	0.64	0.82	0.0	0.63	0.0	0.57	0.44	0.31	0.44
Heuristic	0.35	0.63	0.39	0.34	0.14	0.57	0.96	0.54	0.34	0.4	0.48	0.95	0.51

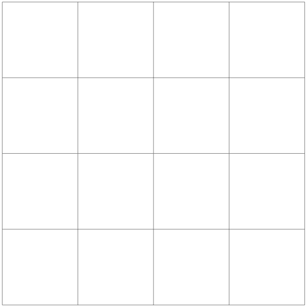
Results

Pathfinding

	wildfire	triangle	academic	elevators	tamarisk	sysadmin	recon	game	traffic	crossing	skill	navigation	Total
UCTStar [IDS]	0.75	0.8	0.32	0.9	1.0	1.0	1.0	0.99	1.0	1.0	0.99	0.99	0.9
Standalone	0.91	0.2	0.46	0.29	0.64	0.82	0.0	0.63	0.0	0.57	0.44	0.31	0.44
Heuristic	0.35	0.63	0.39	0.34	0.14	0.57	0.96	0.54	0.34	0.4	0.48	0.95	0.51

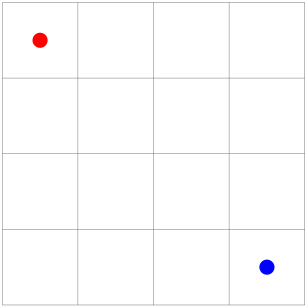
Can we do better

Invariant Synthesis



Let an agent move in this grid

Invariant Synthesis



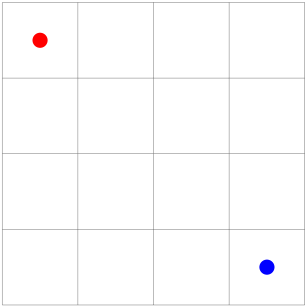
Boolean variables

1000 0000 0000 0000

0000 0000 0000 0001

State Space $= 2^{16}$

Invariant Synthesis



Non-Boolean variable

1

16

State Space $= 16$

Thank you for your time