MDP

Yuetong Liu

February 2021

A king in the chess game would like find the shortest path between 2 squares on a chessboard. It starts from a origin square and move according to the moving rule(move exactly one square horizontally, vertically, or diagonally). In each step, it gets a reward with respect to its current state. The process ends when the king arrives at the terminal square. High-level decisions about how to pick these moves are made by a reinforcement learning agent based on the its rewards.

To make a simple example, we assume that the chess board has 2*2 squares, comprising a small state set S = (0,0),(0,1), (1,0), (1,1).

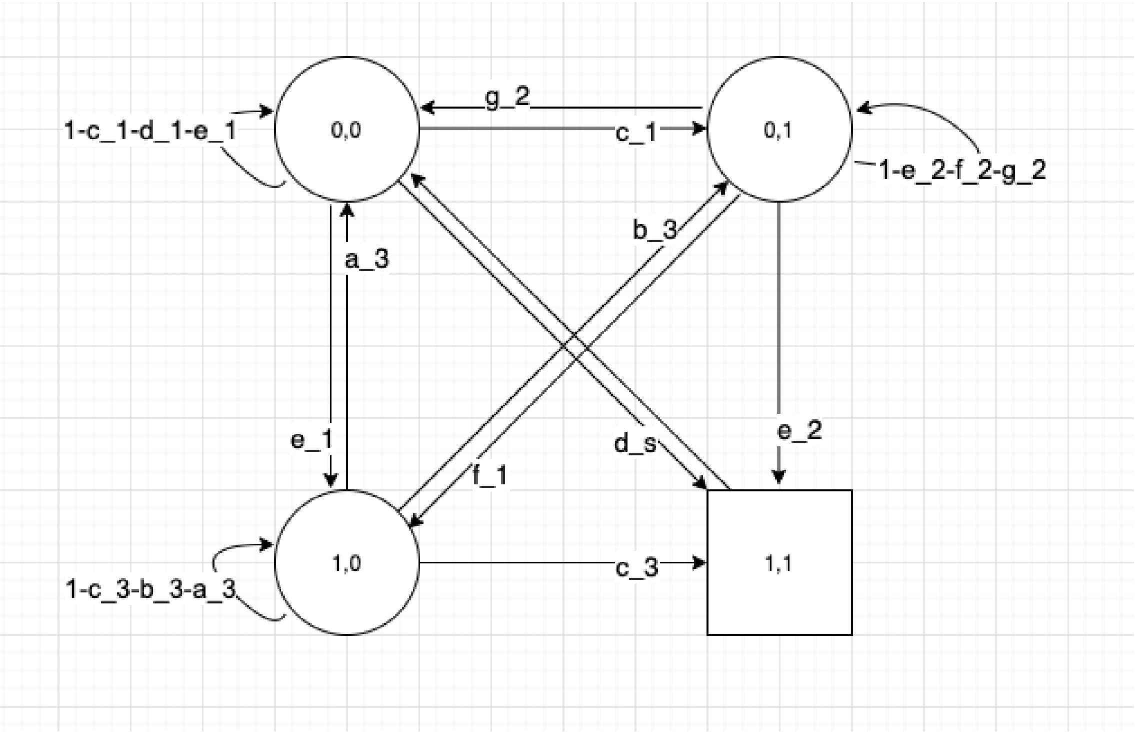
In each state, the king can move exactly one square horizontally, vertically, or diagonally. When the king hit the boundary on the chessboard, it will circle back to the successor state. Therefore the action sets are then A(0,0) = Right, Down Right, Down, Circle, A(0,1) = Down, Down left, Left, Circle. A(1,0) = Right, Up right, Up, Circle

The rewards are -1 most of the time, but become 0 when the agent goes to the terminal state.

This is finite MDP question, and we can write down the transition probabilities and the expected rewards, with dynamics as indicated in the table.

S	a	s'	p(s',s,a)	r(s, a,s')
(0,0)	Right	(0,1)	c_s	-1
(0,0)	Down right	(1,1)	d_s	0
(0,0)	Down	(1,0)	e_s	-1
(0,0)	Circle	(0,0)	$1-c_s - d_s - e_s$	-1
(0,1)	Down	(1,1)	e_1	0
(0,1)	Down left	(1,0)	f_1	-1
(0,1)	Left	(0,0)	g_1	-1
(0,1)	Circle	(0,1)	$1-e_1 - f_1 - g_1$	-1
(1,0)	Right	(1,1)	c_2	0
(1,0)	Up right	(0,1)	b_2	-1
(1,0)	Up	(0,0)	a_2	-1
(1,0)	Circle	(1,0)	$1-c_2 - b_2 - a_2$	-1

Here is the MDP transition graph.


0,0
0,1
1,0
1,1
1-c_1-d_1-e_1
g_2
c_1
1-e_2-f_2-g_2
b_3
a_3
e_1
d_s
e_2
f_1
1-c_3-b_3-a_3
c_3