Lecture 17: Decision-Making cont. & Formal Safety

Professor Katie Driggs-Campbell

April 1, 2021

ECE484: Principles of Safe Autonomy

Administrivia - Schedule

- MP3 due next week 4/9
  - Demo on Thursday / Report due Friday
- MP4 due Friday 4/30
- Next week: Final Safety Lecture and MP4 Walkthrough
- No class on Tuesday 4/13
- No class on Tuesday 5/4 – extra office hours
- Guest Lectures on 4/15, 4/20, and 4/22
  - Attendance worth double
  - Google form to collect questions will be posted on discord
- Project presentations on 4/27 and 4/29
  - Information will be provided on 4/8 during lecture
- Report due on Friday 5/14 (last day of finals)
  - Rubric and guidelines to be posted soon

Administrivia - Oral Exam Protocol

- **Dates:** Monday 4/12 to Tuesday 5/4
- **Times:** 15-minute time slots on Mondays 5-6pm, Tuesdays 5-6pm, Wednesdays 11am-12pm, or by appointment (if needed)
  - Signup will be posted on discord – first come, first serve
- **Content:** Questions and rubric (with bonus allocation) released this weekend
  - You will be asked one of six questions, chosen by rolling a die
- **Protocol:** We will follow CBTF protocol
  - Log in with both computer and phone
  - Position phone so we can see screen and workspace
- **A word of caution:** If we suspect cheating, we will schedule a final exam

Today's Plan

- MDP Policies and Value Iteration
- Simple Example
- Introduction to Safety
- Responsibility Sensitive Safety

MDP Policies

- Policies map states to actions

$$\pi: x \rightarrow u$$

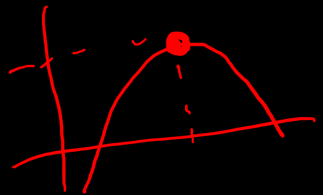
- We want to find a policy that maximizes future pay off
  - Suppose $T = 1$: $\pi_1(x) = \text{argmax}_u \; r(x, u)$
- We write the Value Function for given π:

$$V_1(x) = \gamma \max_u r(x, u)$$

- Generally, we want to find the sequence of actions that optimize the *expected cumulative discounted future payoff*

$$R_T = \mathbb{E}\left[\sum_{\tau=0}^{T} \gamma^{\tau} r_{t+\tau}\right]$$

Value Functions



Recall: $V_1(x) = \gamma \max_{u} r(x, u)$

For longer time horizons, we define V(x) recursively:

T = 2: pick an action that max sum V_1 + 1-step payoff

$$\pi_2(x) = \underset{u}{\arg\max}\left[r(x,u) + \int V_1(x')\, p(x' \mid x, u)\, dx'\right]$$

$$V_2(x) = \gamma \max_{u}\left[r(x,u) + \int V_1(x')\, p(x' \mid x, u)\, dx'\right]$$

For finite T:

$$\pi_T(x) = \underset{u}{\arg\max}\left[r(x,u) + \int V_{T-1}(x')\, p(x' \mid x, u)\, dx'\right]$$

$$V_T(x) = \gamma \max_{u}\left[r(x,u) + \int V_{T-1}(x')\, p(x' \mid x, u)\, dx'\right]$$

Value Functions

- In the infinite time horizon, we tend to reach equilibrium:

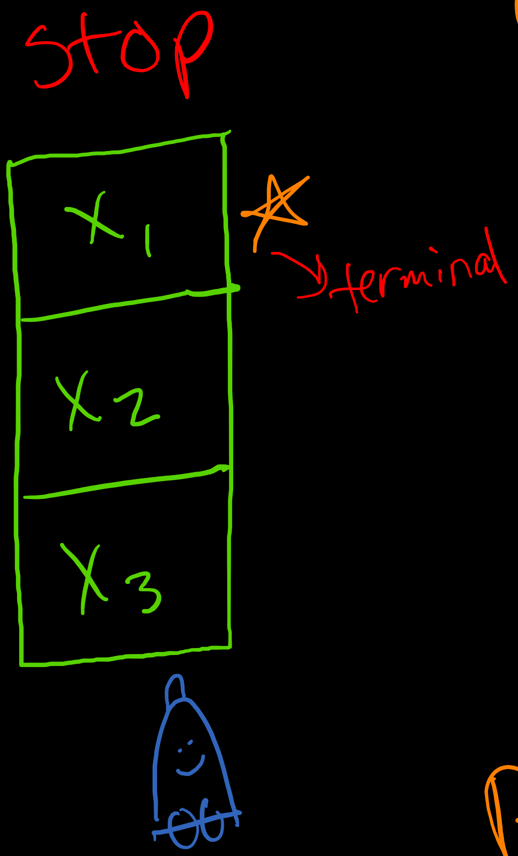
$$V_\infty(x) = \gamma \max_u \left[r(x,u) + \int V_\infty(x') p(x'|x,u)\, dx' \right]$$

- This is the *Bellman Equation*
  - Satisfying this is necessary and sufficient for an optimal policy

Computing the (Approximate) Value Function

- Initial guess for $\hat{V}$
  - $\hat{V}(x) \leftarrow r_{min}, \forall x$
- Successively update for increasing horizons
  - $\hat{V}(x) \leftarrow \gamma \max_u \left[r(x,u) + \int \hat{V}(x')p(x'|x,u)dx' \right]$
- Value iteration converges if $\gamma < 1$
- Given estimate $\hat{V}(x)$, policy is found:
  - $\pi(x) = \operatorname{argmax}_u \left[r(x,u) + \int \hat{V}(x')p(x'|x,u)dx' \right]$
- Often, we use the discrete version:
  - $\pi(x) = \operatorname{argmax}_u \left[r(x,u) + \sum_x' \hat{V}(x')p(x'|x,u) \right]$

Example: Create an MDP



$$(S, A, T, \gamma, R)$$

$$S = \{x_1, x_2, x_3\}$$

$$A = \{\uparrow, 0\}$$

$p(x' \mid x, u)$ is deterministic

- $\uparrow$ will move up $P=1$
- 0 will stop $P=1$

$$R = \begin{cases} r_s & \text{if } x_1 + u=0 \\ 0 & \text{ow} \end{cases}$$

Example: Value Iteration

$$\hat{V}_{iter}(x_i) = \gamma \max_u \left[r(x_i, u) + \sum_j \hat{V}_{iter-1}(x_j)\, p(x_j \mid x_i, u) \right]$$

vector rep: $\hat{V} = [\hat{V}(x_1) \;\; \hat{V}(x_2) \;\; \hat{V}(x_3)]$

→ init: $\hat{V} = [0 \;\; 0 \;\; 0]$

$\hat{V}_1(x_1) = \gamma \max_u (0 + 0,\; r_s + 0) = \gamma r_s$ (u=1, u=0)

$\hat{V}_1(x_2) = \gamma \max_u (0 + 0,\; 0 + 0) = 0$

$\hat{V}_1(x_3) = 0$

$\hat{V} = [\gamma r_s \;\; 0 \;\; 0]$

$\hat{V}_2(x_1) = \gamma r_s$

$\hat{V}_2(x_2) = \gamma \max_u (0 + \gamma r_s,\; 0 + 0) = \gamma^2 r_s$

$\hat{V}_2(x_3) = 0$

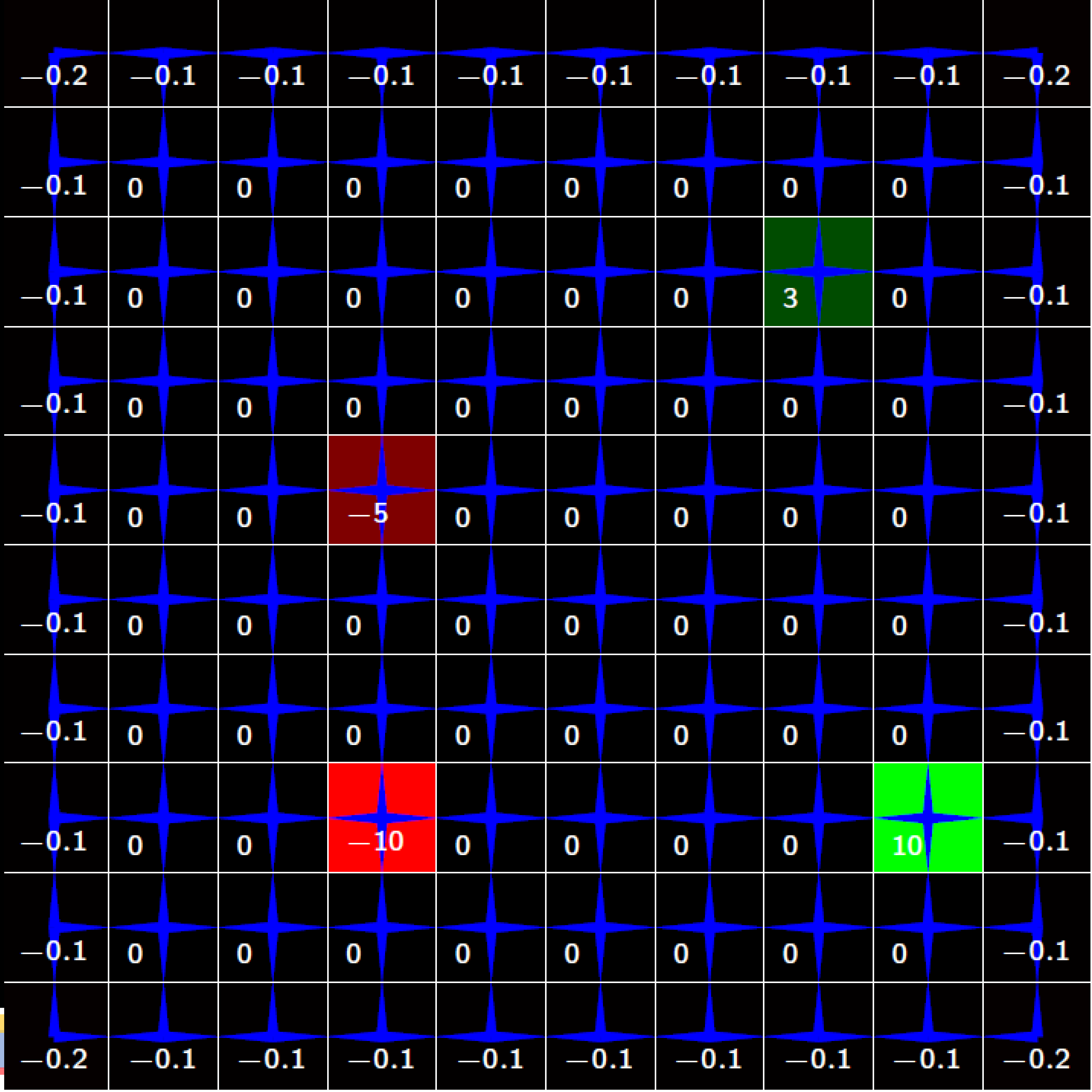
$\hat{V}_2 = [\gamma r_s \;\; \gamma^2 r_s \;\; 0]$

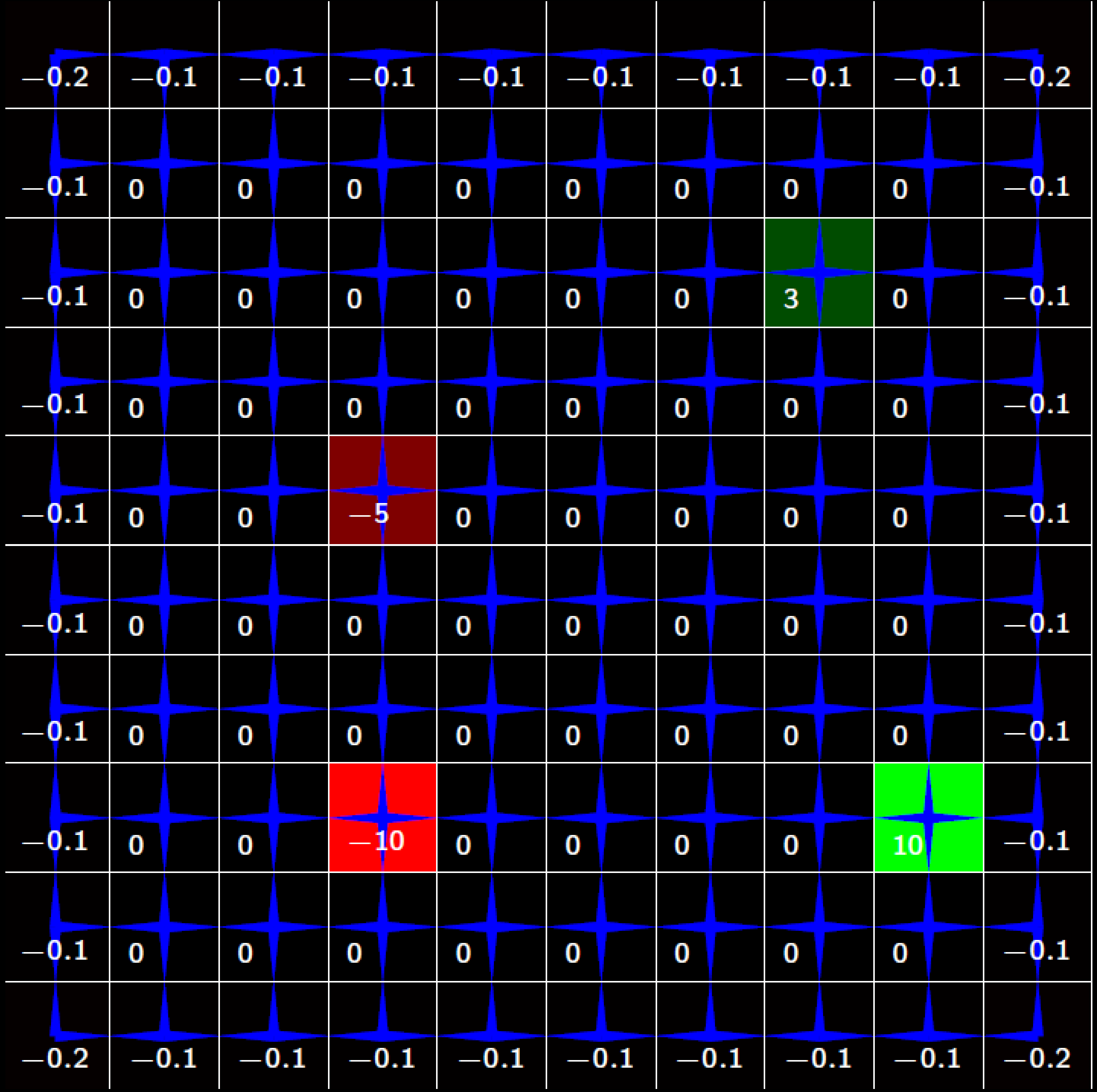
⋮

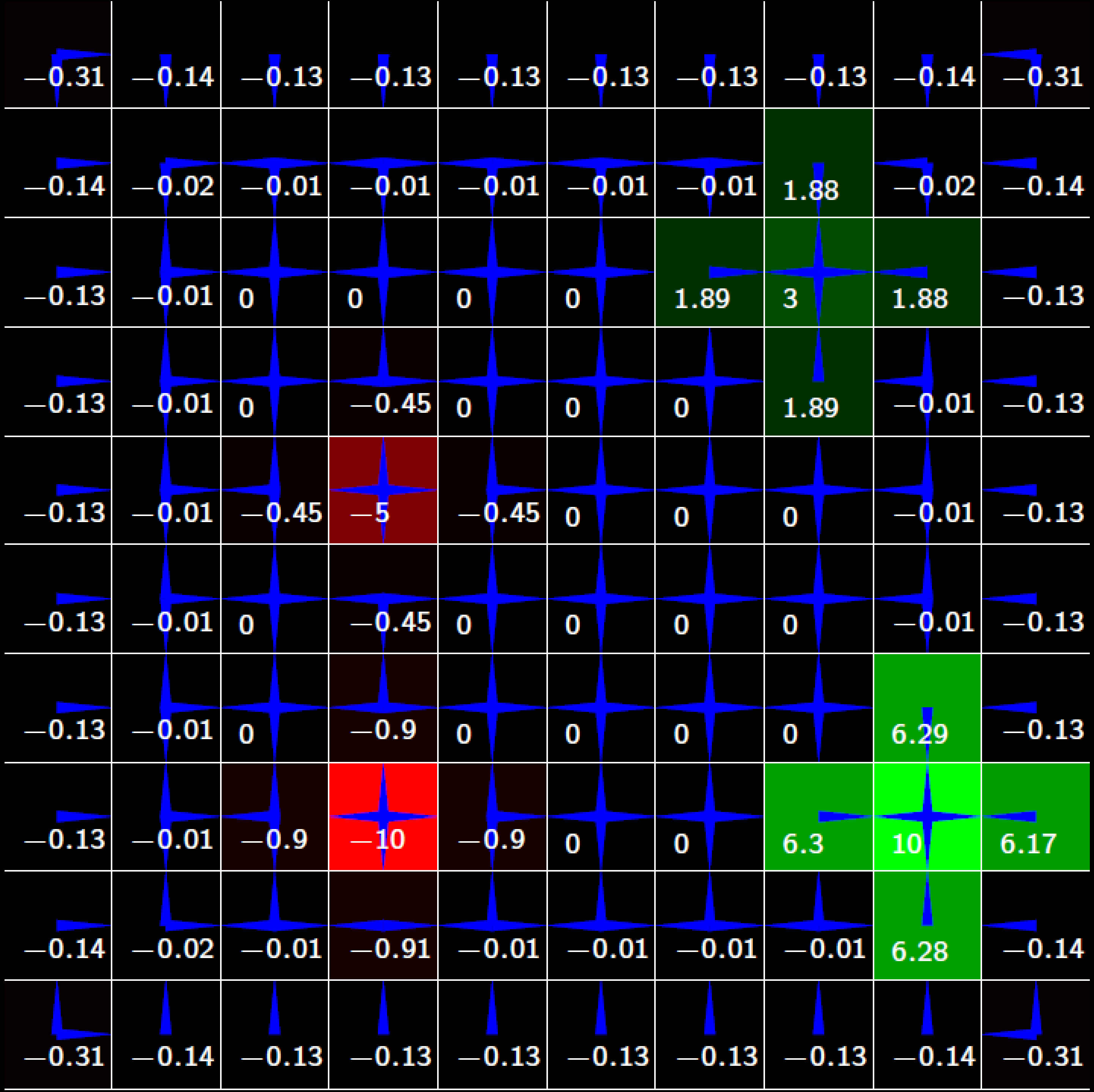
$\hat{V}_3 = [\gamma r_s \;\; \gamma^2 r_s \;\; \gamma^3 r_s]$

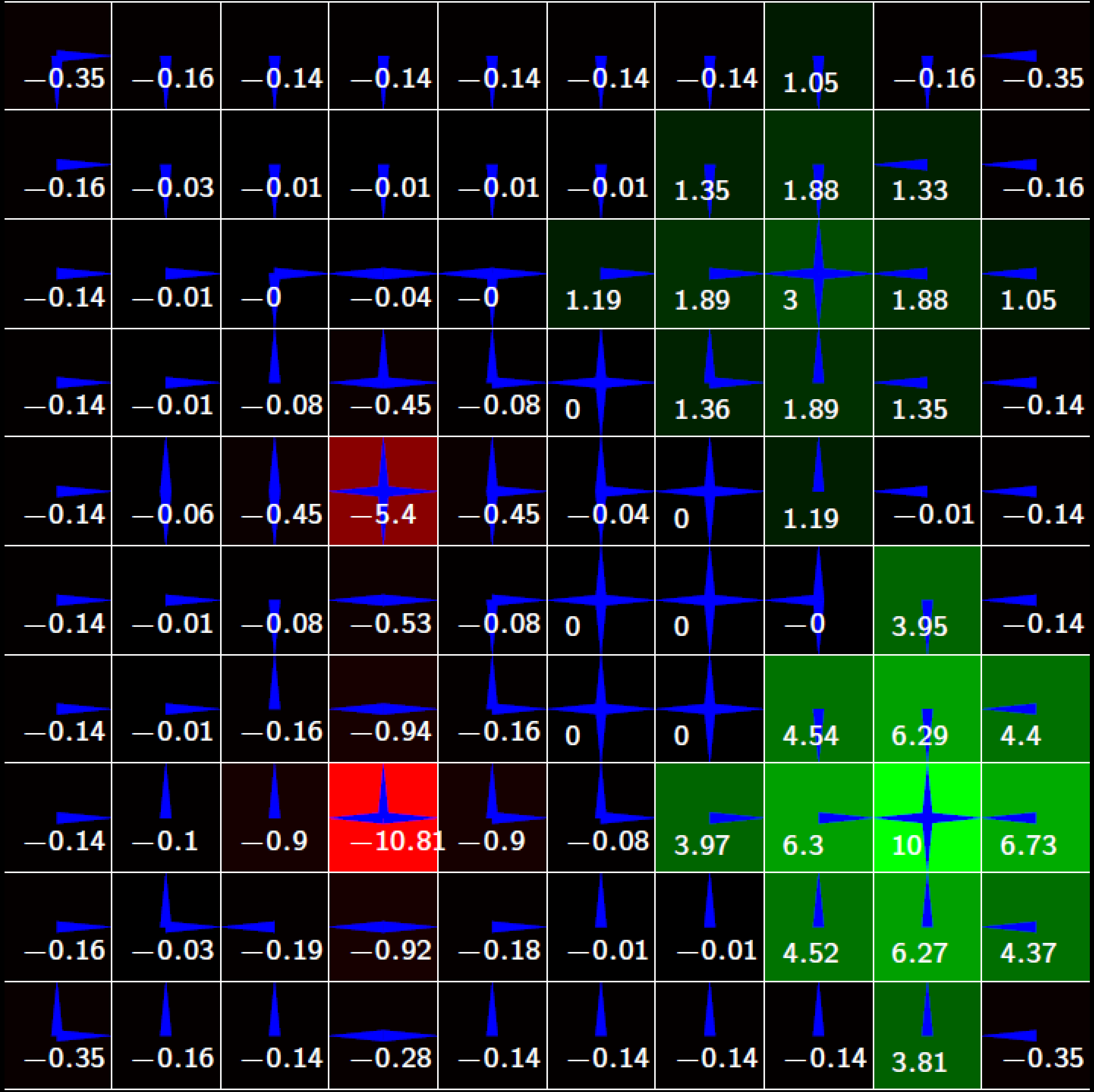
Grid world example

- States: cells in 10 × 10 grid
- Actions: up, down, left, right
- Transition model: 0.7 chance of moving in intended direction, uniform in other directions
- Reward:
  - two states with cost
  - two terminal states with rewards
  - −1 for wall crash
- Discount is 0.9

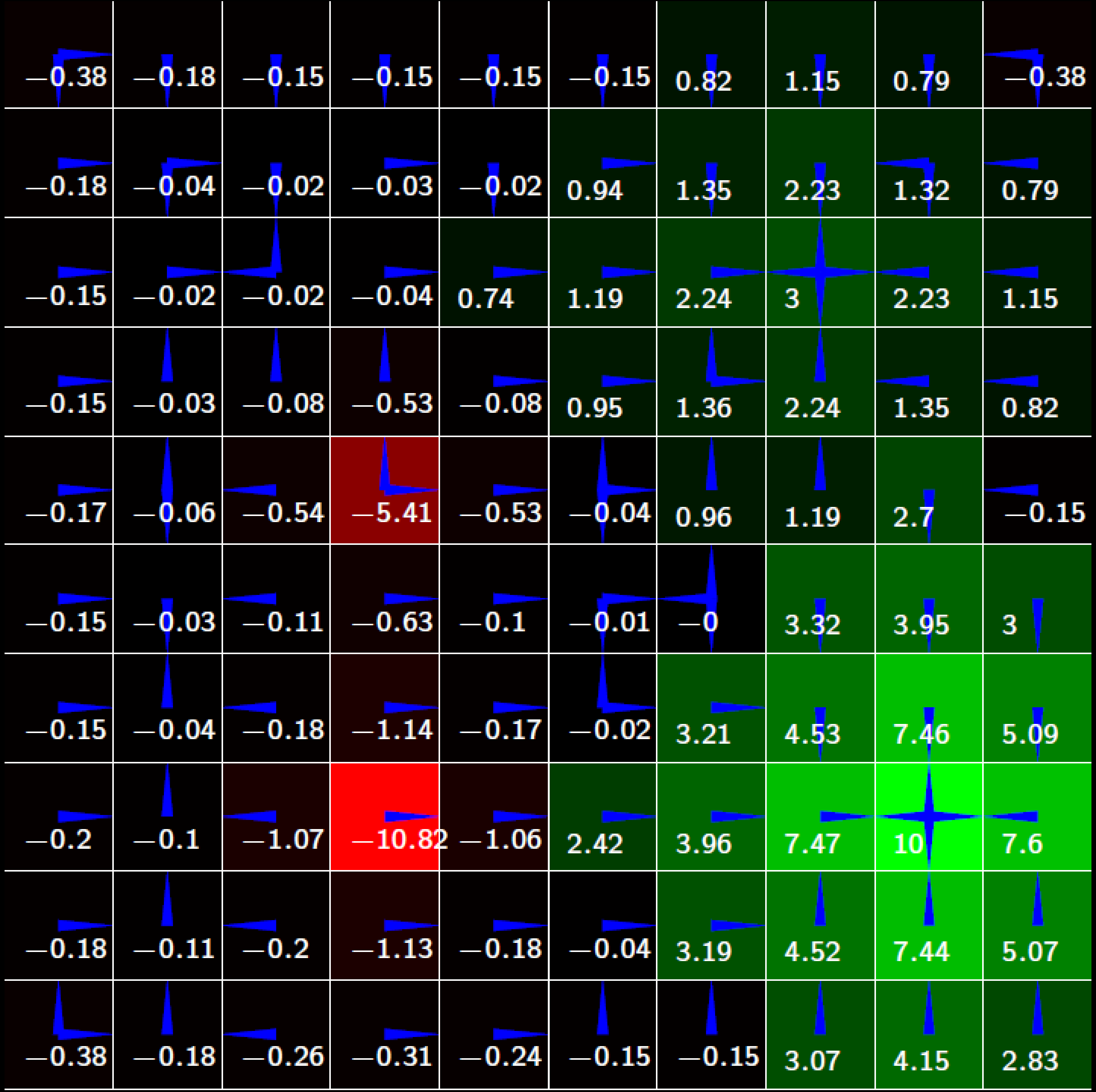



−0.2 −0.1 −0.1 −0.1 −0.1 −0.1 −0.1 −0.1 −0.1 −0.2
−0.1 0 0 0 0 0 0 0 0 −0.1
−0.1 0 0 0 0 0 0 3 0 −0.1
−0.1 0 0 0 0 0 0 0 0 −0.1
−0.1 0 0 −5 0 0 0 0 0 −0.1
−0.1 0 0 0 0 0 0 0 0 −0.1
−0.1 0 0 0 0 0 0 0 0 −0.1
−0.1 0 0 −10 0 0 0 0 10 −0.1
−0.1 0 0 0 0 0 0 0 0 −0.1
−0.2 −0.1 −0.1 −0.1 −0.1 −0.1 −0.1 −0.1 −0.1 −0.2

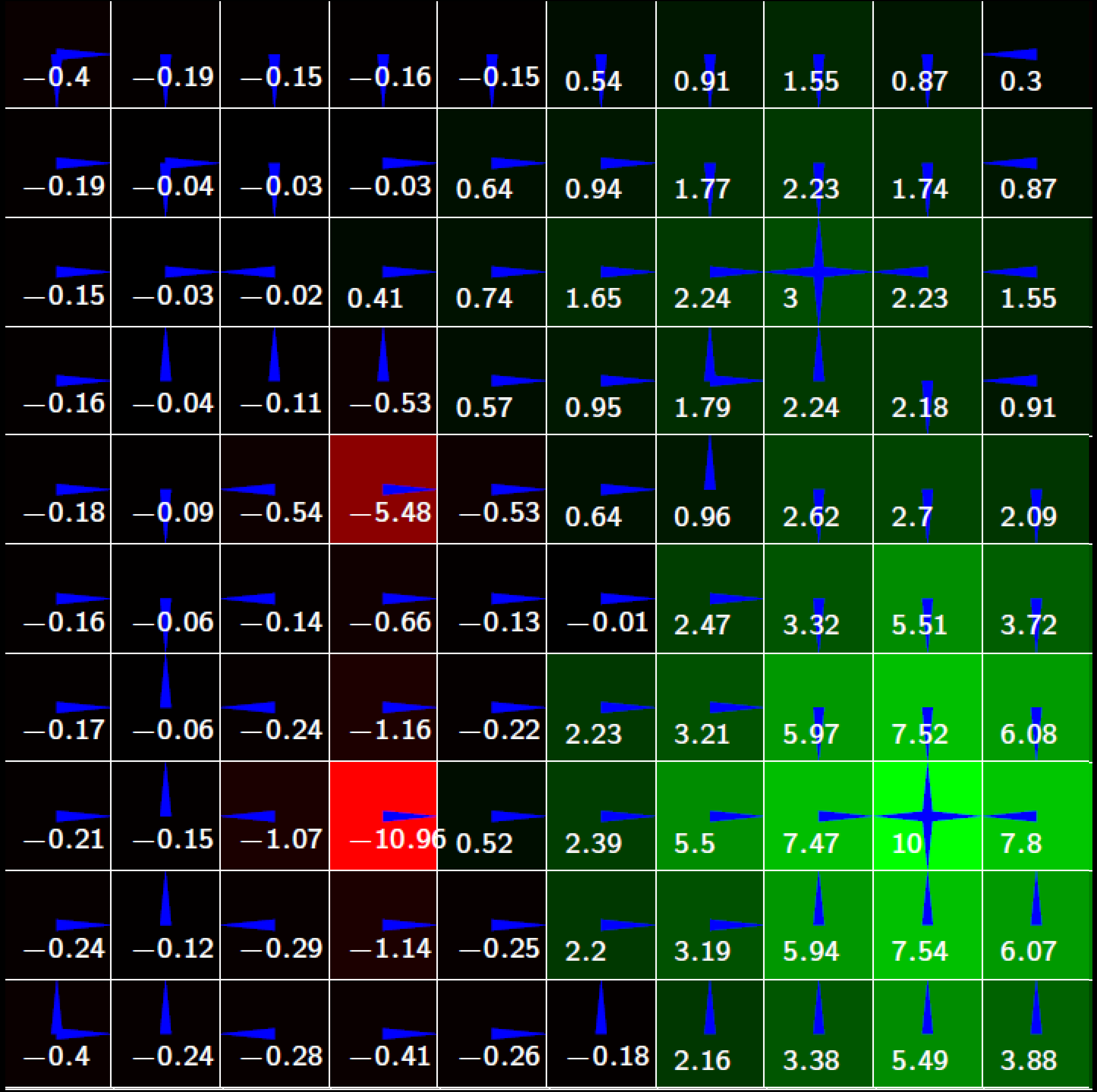

−0.31 −0.14 −0.13 −0.13 −0.13 −0.13 −0.13 −0.13 −0.14 −0.31
−0.14 −0.02 −0.01 −0.01 −0.01 −0.01 −0.01 1.88 −0.02 −0.14
−0.13 −0.01 0 0 0 0 1.89 3 1.88 −0.13
−0.13 −0.01 0 −0.45 0 0 0 1.89 −0.01 −0.13
−0.13 −0.01 −0.45 −5 −0.45 0 0 0 −0.01 −0.13
−0.13 −0.01 0 −0.45 0 0 0 0 −0.01 −0.13
−0.13 −0.01 0 −0.9 0 0 0 0 6.29 −0.13
−0.13 −0.01 −0.9 −10 −0.9 0 0 6.3 10 6.17
−0.14 −0.02 −0.01 −0.91 −0.01 −0.01 −0.01 −0.01 6.28 −0.14
−0.31 −0.14 −0.13 −0.13 −0.13 −0.13 −0.13 −0.13 −0.14 −0.31

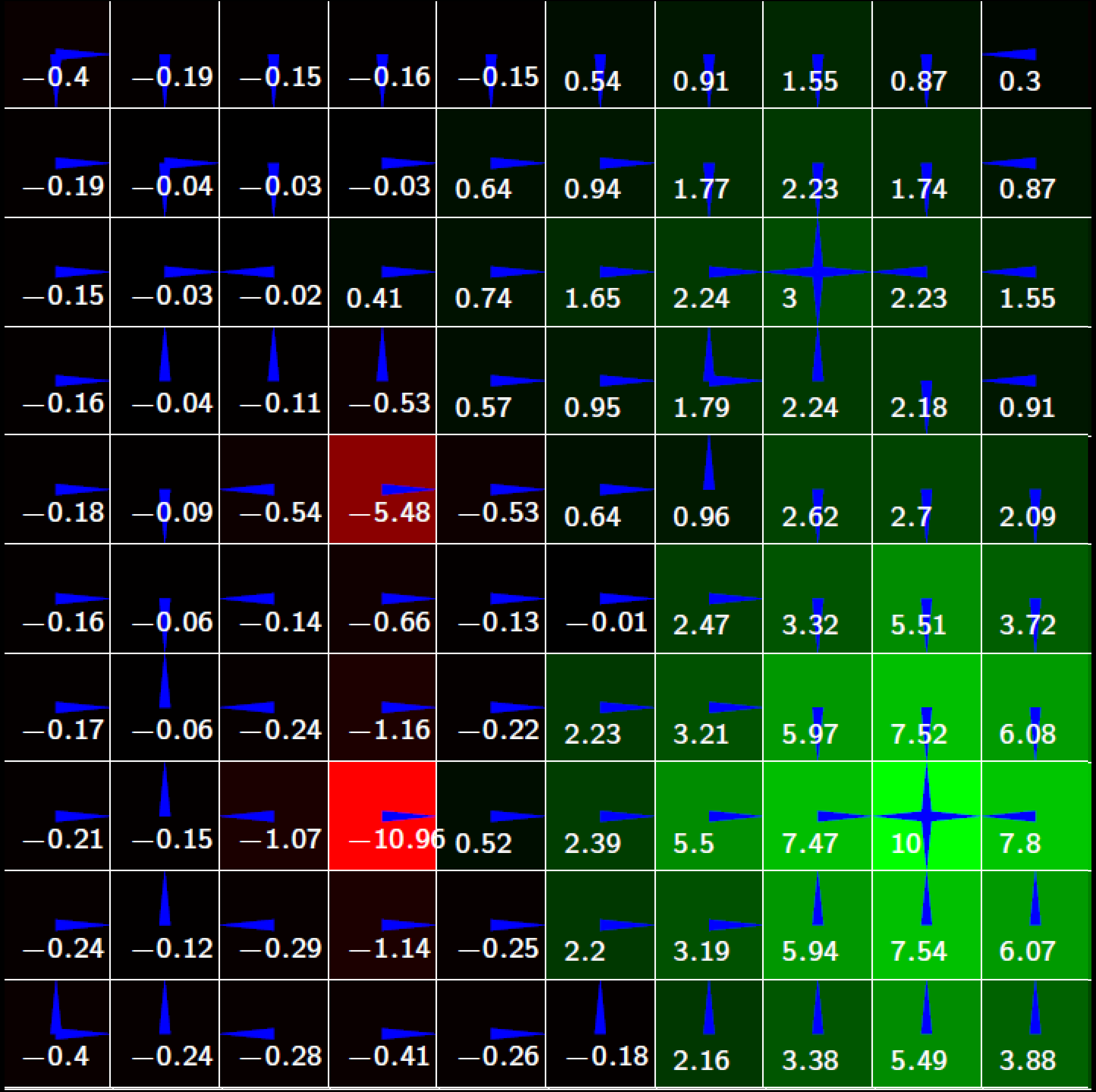


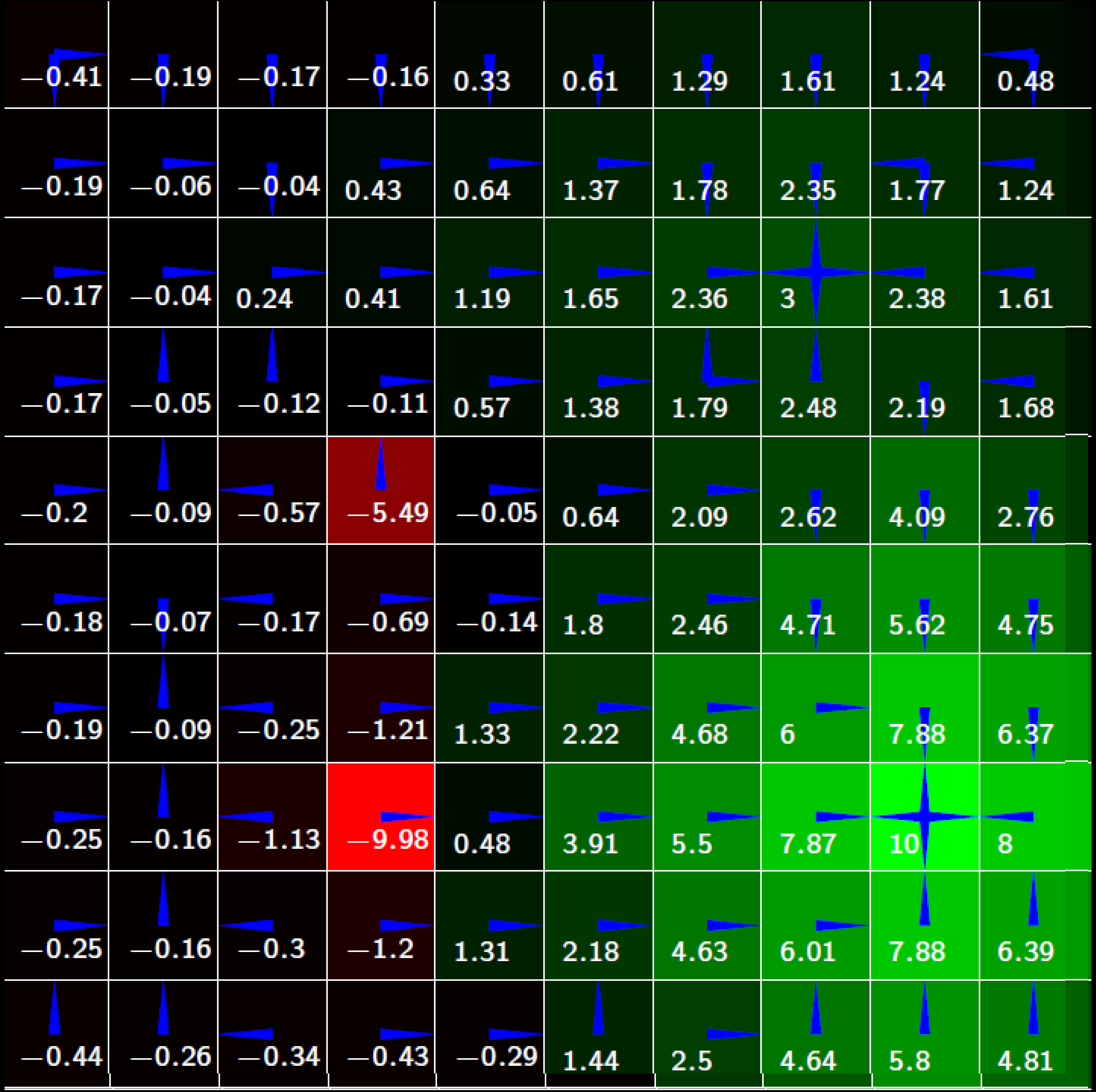
-0.35 -0.16 -0.14 -0.14 -0.14 -0.14 -0.14 1.05 -0.16 -0.35
-0.16 -0.03 -0.01 -0.01 -0.01 -0.01 1.35 1.88 1.33 -0.16
-0.14 -0.01 -0 -0.04 -0 1.19 1.89 3 1.88 1.05
-0.14 -0.01 -0.08 -0.45 -0.08 0 1.36 1.89 1.35 -0.14
-0.14 -0.06 -0.45 -5.4 -0.45 -0.04 0 1.19 -0.01 -0.14
-0.14 -0.01 -0.08 -0.53 -0.08 0 0 -0 3.95 -0.14
-0.14 -0.01 -0.16 -0.94 -0.16 0 0 4.54 6.29 4.4
-0.14 -0.1 -0.9 -10.81 -0.9 -0.08 3.97 6.3 10 6.73
-0.16 -0.03 -0.19 -0.92 -0.18 -0.01 -0.01 4.52 6.27 4.37
-0.35 -0.16 -0.14 -0.28 -0.14 -0.14 -0.14 -0.14 3.81 -0.35

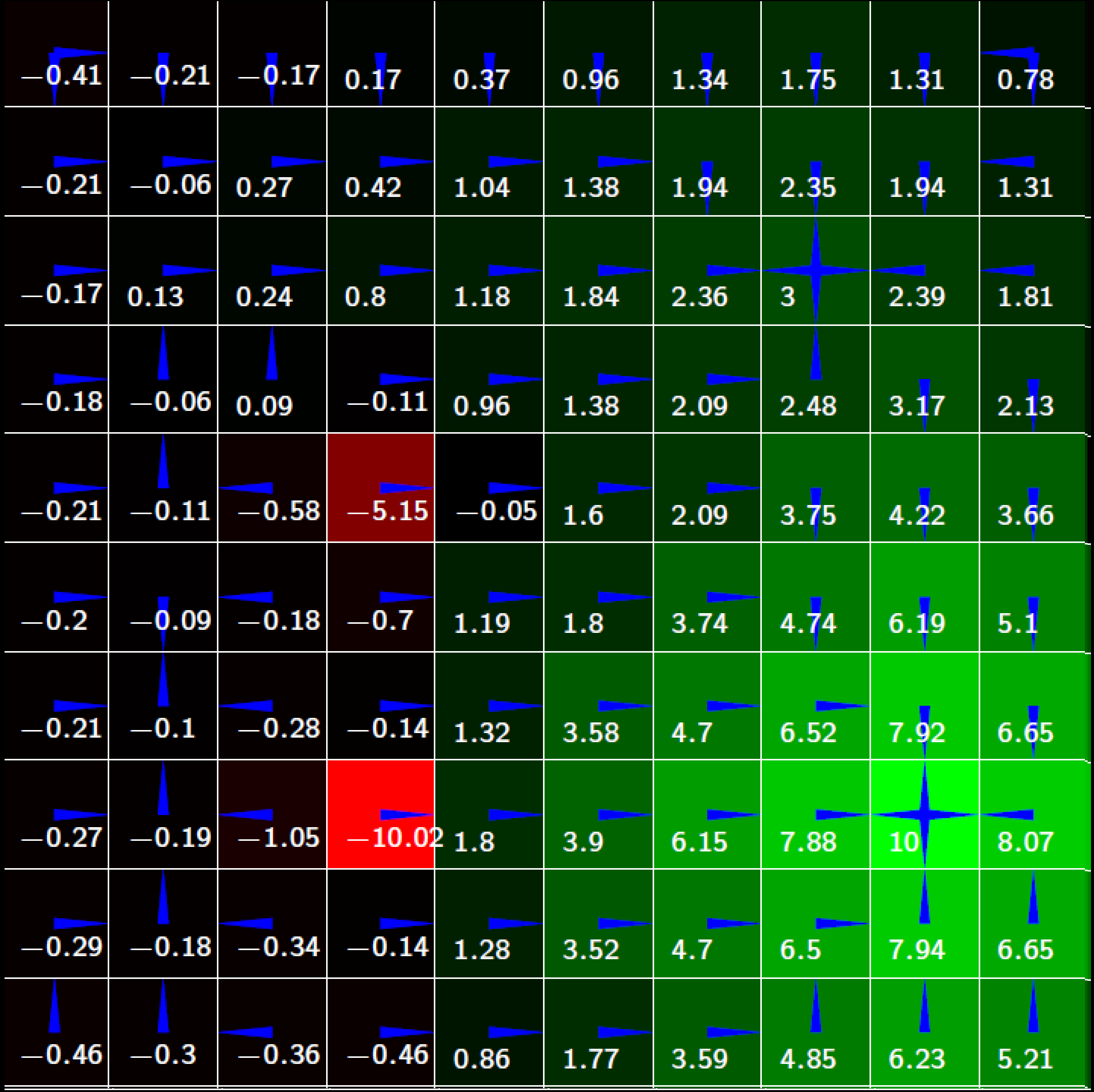


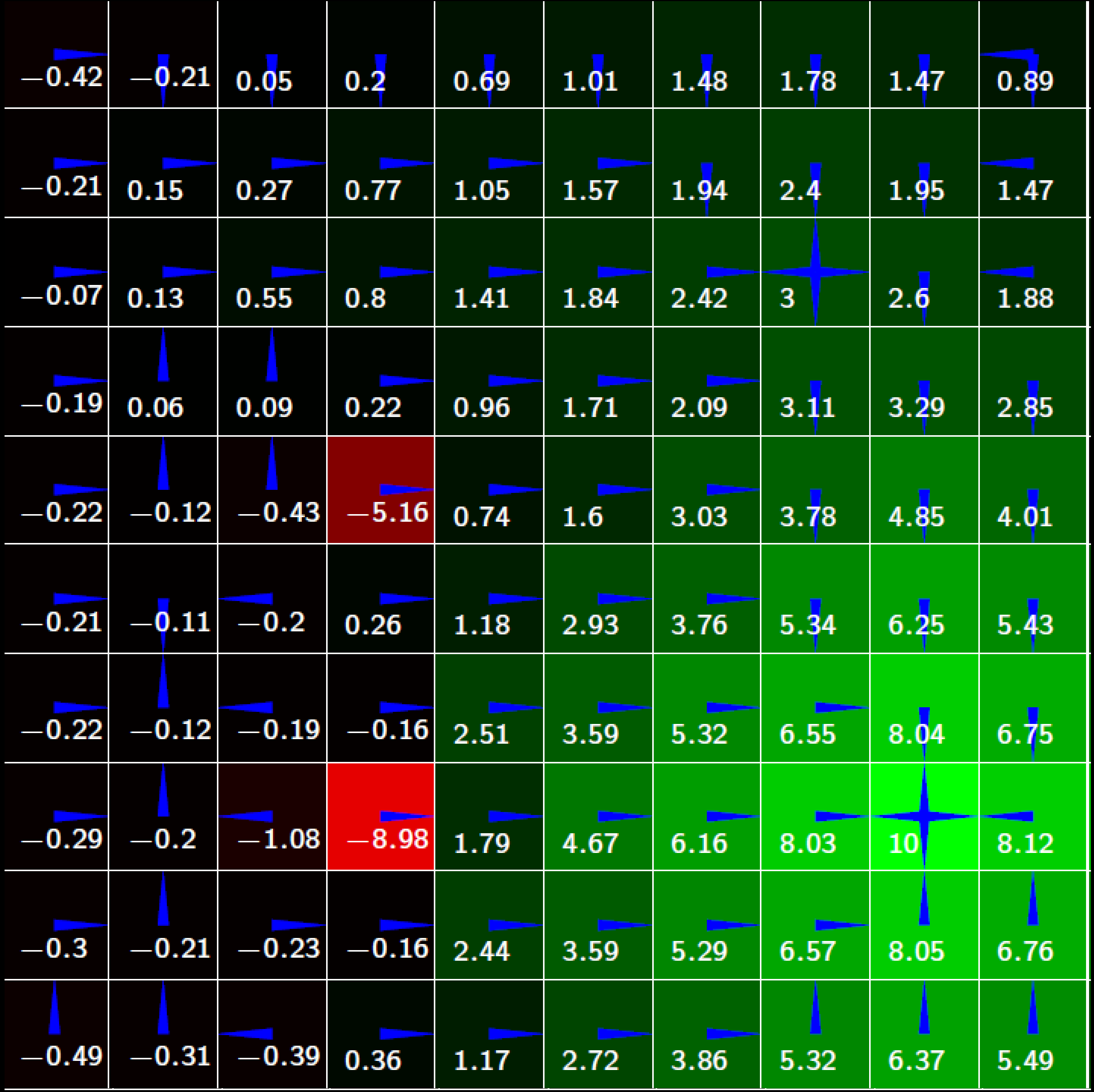
-0.38 -0.18 -0.15 -0.15 -0.15 -0.15 0.82 1.15 0.79 -0.38
-0.18 -0.04 -0.02 -0.03 -0.02 0.94 1.35 2.23 1.32 0.79
-0.15 -0.02 -0.02 -0.04 0.74 1.19 2.24 3 2.23 1.15
-0.15 -0.03 -0.08 -0.53 -0.08 0.95 1.36 2.24 1.35 0.82
-0.17 -0.06 -0.54 -5.41 -0.53 -0.04 0.96 1.19 2.7 -0.15
-0.15 -0.03 -0.11 -0.63 -0.1 -0.01 -0 3.32 3.95 3
-0.15 -0.04 -0.18 -1.14 -0.17 -0.02 3.21 4.53 7.46 5.09
-0.2 -0.1 -1.07 -10.82 -1.06 2.42 3.96 7.47 10 7.6
-0.18 -0.11 -0.2 -1.13 -0.18 -0.04 3.19 4.52 7.44 5.07
-0.38 -0.18 -0.26 -0.31 -0.24 -0.15 -0.15 3.07 4.15 2.83

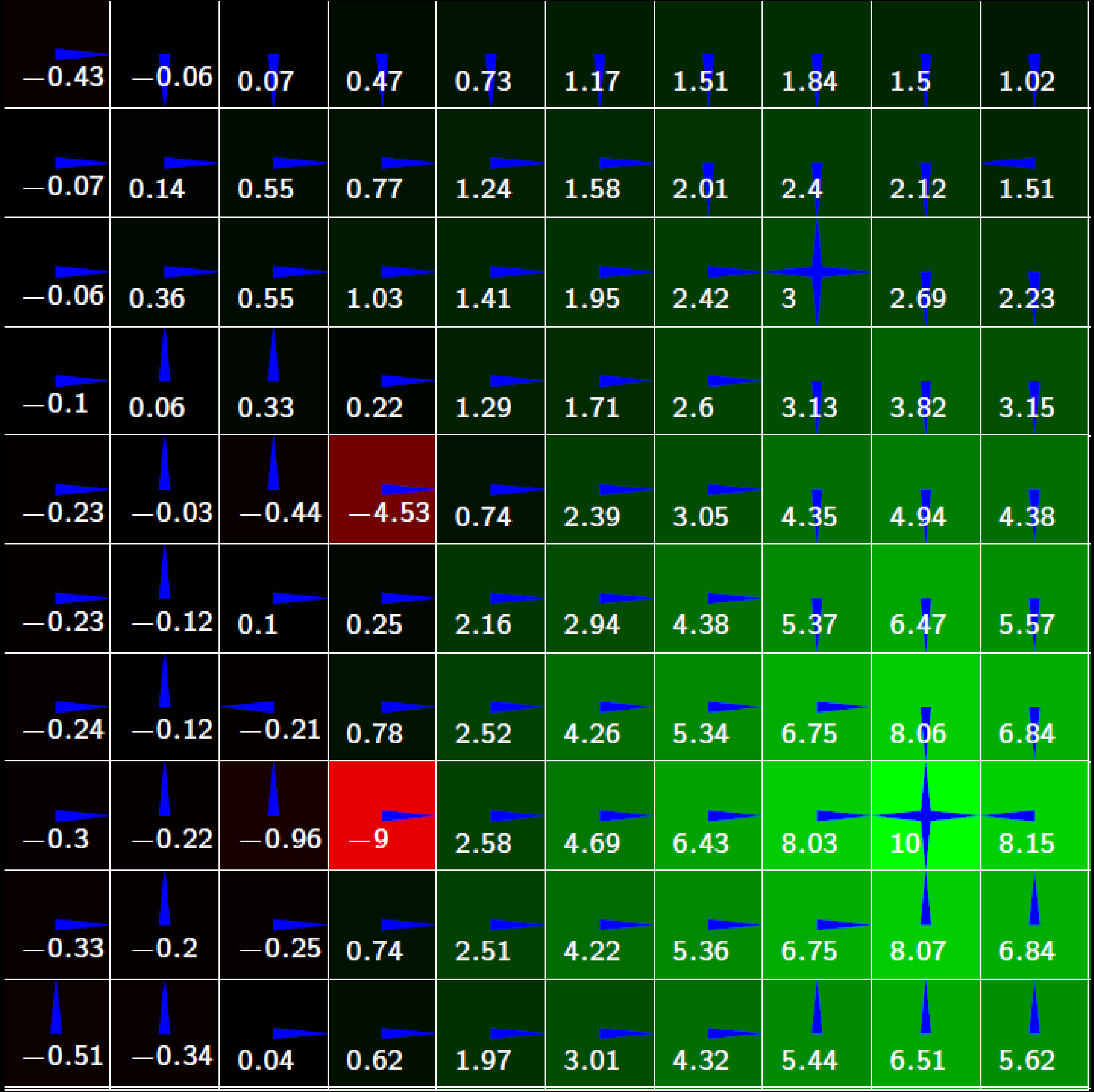

−0.4 −0.19 −0.15 −0.16 −0.15 0.54 0.91 1.55 0.87 0.3
−0.19 −0.04 −0.03 −0.03 0.64 0.94 1.77 2.23 1.74 0.87
−0.15 −0.03 −0.02 0.41 0.74 1.65 2.24 3 2.23 1.55
−0.16 −0.04 −0.11 −0.53 0.57 0.95 1.79 2.24 2.18 0.91
−0.18 −0.09 −0.54 −5.48 −0.53 0.64 0.96 2.62 2.7 2.09
−0.16 −0.06 −0.14 −0.66 −0.13 −0.01 2.47 3.32 5.51 3.72
−0.17 −0.06 −0.24 −1.16 −0.22 2.23 3.21 5.97 7.52 6.08
−0.21 −0.15 −1.07 −10.96 0.52 2.39 5.5 7.47 10 7.8
−0.24 −0.12 −0.29 −1.14 −0.25 2.2 3.19 5.94 7.54 6.07
−0.4 −0.24 −0.28 −0.41 −0.26 −0.18 2.16 3.38 5.49 3.88

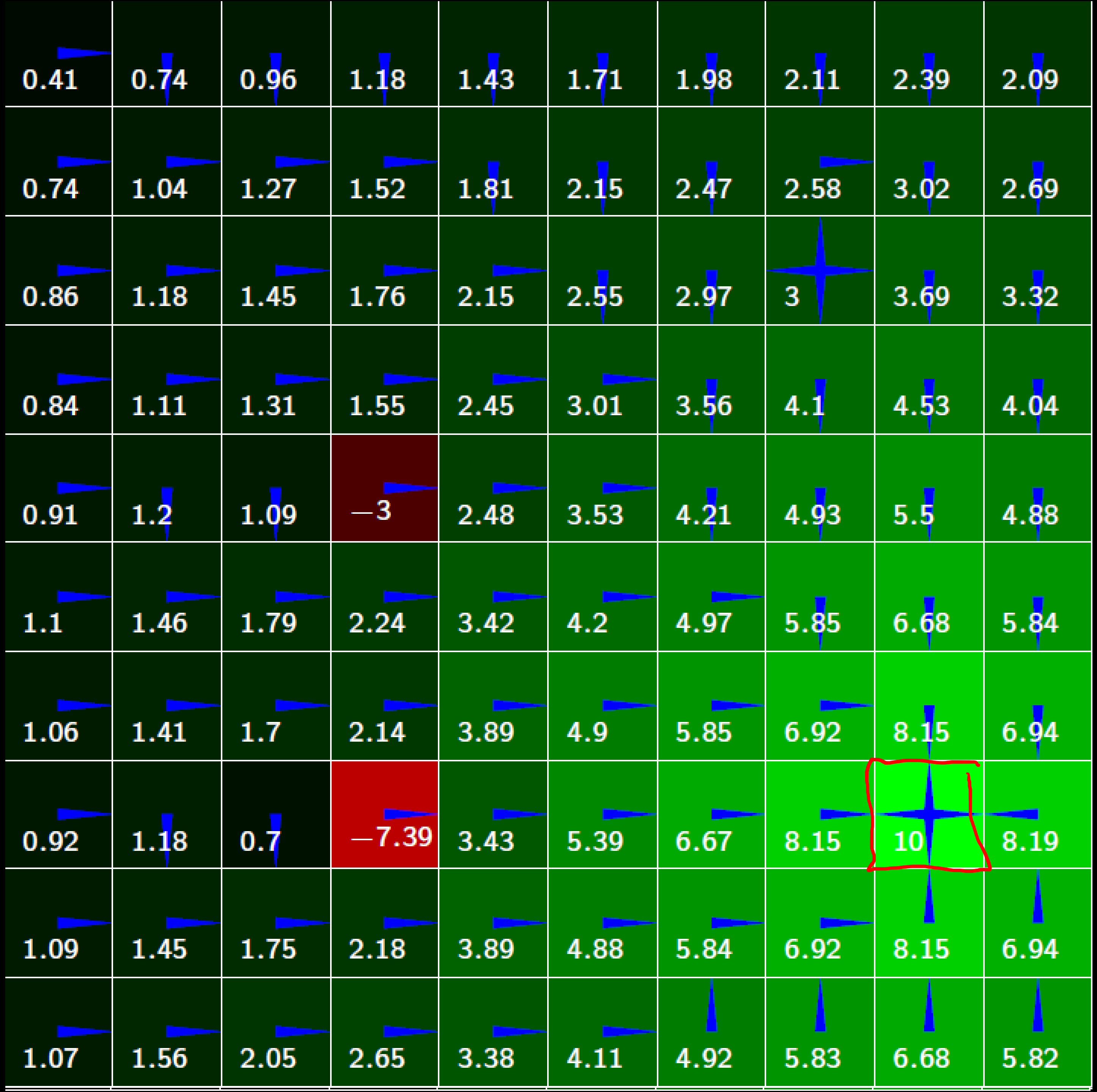

-0.4 -0.19 -0.15 -0.16 -0.15 0.54 0.91 1.55 0.87 0.3
-0.19 -0.04 -0.03 -0.03 0.64 0.94 1.77 2.23 1.74 0.87
-0.15 -0.03 -0.02 0.41 0.74 1.65 2.24 3 2.23 1.55
-0.16 -0.04 -0.11 -0.53 0.57 0.95 1.79 2.24 2.18 0.91
-0.18 -0.09 -0.54 -5.48 -0.53 0.64 0.96 2.62 2.7 2.09
-0.16 -0.06 -0.14 -0.66 -0.13 -0.01 2.47 3.32 5.51 3.72
-0.17 -0.06 -0.24 -1.16 -0.22 2.23 3.21 5.97 7.52 6.08
-0.21 -0.15 -1.07 -10.96 0.52 2.39 5.5 7.47 10 7.8
-0.24 -0.12 -0.29 -1.14 -0.25 2.2 3.19 5.94 7.54 6.07
-0.4 -0.24 -0.28 -0.41 -0.26 -0.18 2.16 3.38 5.49 3.88


-0.41 -0.19 -0.17 -0.16 0.33 0.61 1.29 1.61 1.24 0.48
-0.19 -0.06 -0.04 0.43 0.64 1.37 1.78 2.35 1.77 1.24
-0.17 -0.04 0.24 0.41 1.19 1.65 2.36 3 2.38 1.61
-0.17 -0.05 -0.12 -0.11 0.57 1.38 1.79 2.48 2.19 1.68
-0.2 -0.09 -0.57 -5.49 -0.05 0.64 2.09 2.62 4.09 2.76
-0.18 -0.07 -0.17 -0.69 -0.14 1.8 2.46 4.71 5.62 4.75
-0.19 -0.09 -0.25 -1.21 1.33 2.22 4.68 6 7.88 6.37
-0.25 -0.16 -1.13 -9.98 0.48 3.91 5.5 7.87 10 8
-0.25 -0.16 -0.3 -1.2 1.31 2.18 4.63 6.01 7.88 6.39
-0.44 -0.26 -0.34 -0.43 -0.29 1.44 2.5 4.64 5.8 4.81


−0.41 −0.21 −0.17 0.17 0.37 0.96 1.34 1.75 1.31 0.78
−0.21 −0.06 0.27 0.42 1.04 1.38 1.94 2.35 1.94 1.31
−0.17 0.13 0.24 0.8 1.18 1.84 2.36 3 2.39 1.81
−0.18 −0.06 0.09 −0.11 0.96 1.38 2.09 2.48 3.17 2.13
−0.21 −0.11 −0.58 −5.15 −0.05 1.6 2.09 3.75 4.22 3.66
−0.2 −0.09 −0.18 −0.7 1.19 1.8 3.74 4.74 6.19 5.1
−0.21 −0.1 −0.28 −0.14 1.32 3.58 4.7 6.52 7.92 6.65
−0.27 −0.19 −1.05 −10.02 1.8 3.9 6.15 7.88 10 8.07
−0.29 −0.18 −0.34 −0.14 1.28 3.52 4.7 6.5 7.94 6.65
−0.46 −0.3 −0.36 −0.46 0.86 1.77 3.59 4.85 6.23 5.21

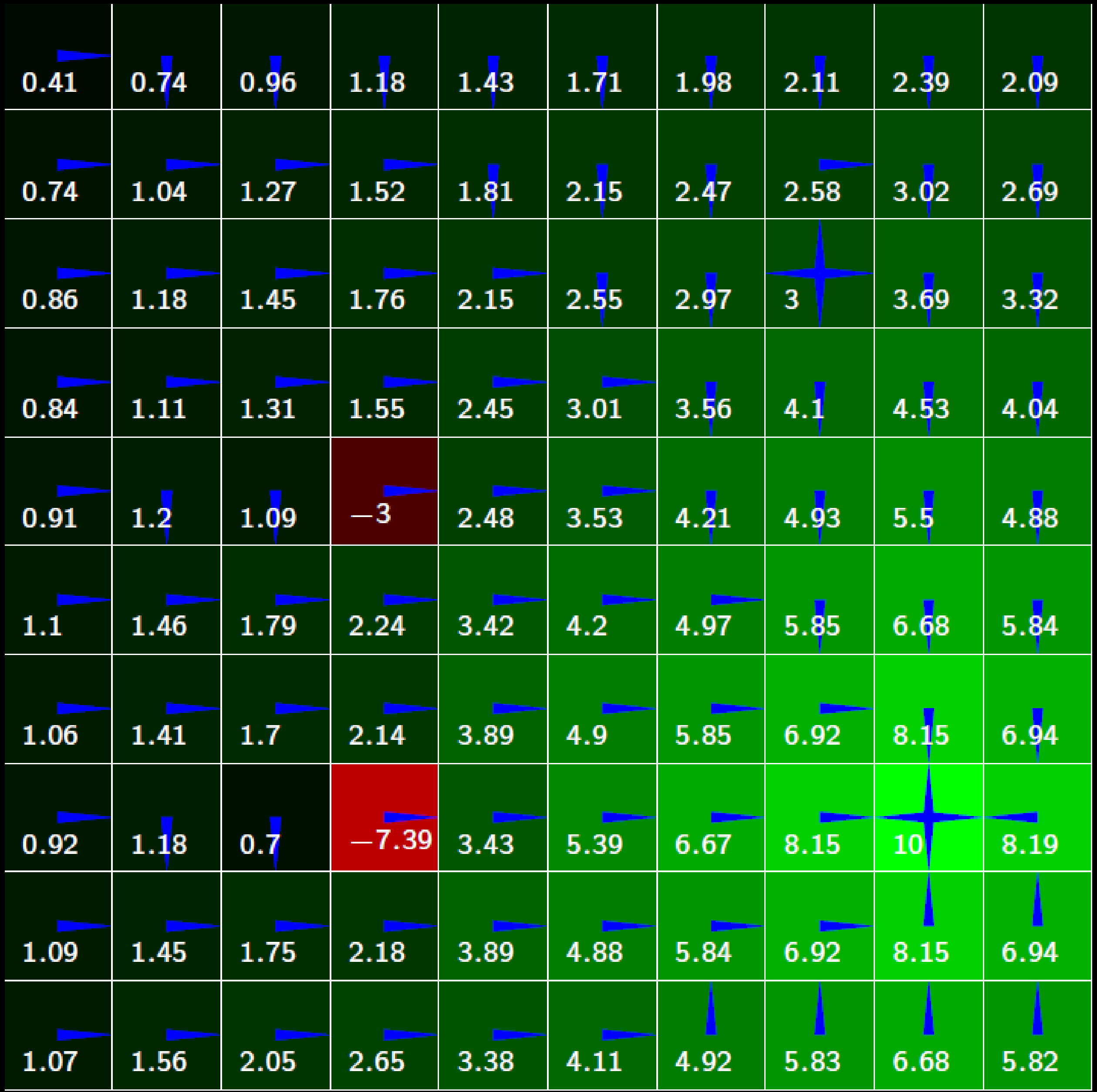

-0.42 -0.21 0.05 0.2 0.69 1.01 1.48 1.78 1.47 0.89
-0.21 0.15 0.27 0.77 1.05 1.57 1.94 2.4 1.95 1.47
-0.07 0.13 0.55 0.8 1.41 1.84 2.42 3 2.6 1.88
-0.19 0.06 0.09 0.22 0.96 1.71 2.09 3.11 3.29 2.85
-0.22 -0.12 -0.43 -5.16 0.74 1.6 3.03 3.78 4.85 4.01
-0.21 -0.11 -0.2 0.26 1.18 2.93 3.76 5.34 6.25 5.43
-0.22 -0.12 -0.19 -0.16 2.51 3.59 5.32 6.55 8.04 6.75
-0.29 -0.2 -1.08 -8.98 1.79 4.67 6.16 8.03 10 8.12
-0.3 -0.21 -0.23 -0.16 2.44 3.59 5.29 6.57 8.05 6.76
-0.49 -0.31 -0.39 0.36 1.17 2.72 3.86 5.32 6.37 5.49


−0.43 −0.06 0.07 0.47 0.73 1.17 1.51 1.84 1.5 1.02
−0.07 0.14 0.55 0.77 1.24 1.58 2.01 2.4 2.12 1.51
−0.06 0.36 0.55 1.03 1.41 1.95 2.42 3 2.69 2.23
−0.1 0.06 0.33 0.22 1.29 1.71 2.6 3.13 3.82 3.15
−0.23 −0.03 −0.44 −4.53 0.74 2.39 3.05 4.35 4.94 4.38
−0.23 −0.12 0.1 0.25 2.16 2.94 4.38 5.37 6.47 5.57
−0.24 −0.12 −0.21 0.78 2.52 4.26 5.34 6.75 8.06 6.84
−0.3 −0.22 −0.96 −9 2.58 4.69 6.43 8.03 10 8.15
−0.33 −0.2 −0.25 0.74 2.51 4.22 5.36 6.75 8.07 6.84
−0.51 −0.34 0.04 0.62 1.97 3.01 4.32 5.44 6.51 5.62

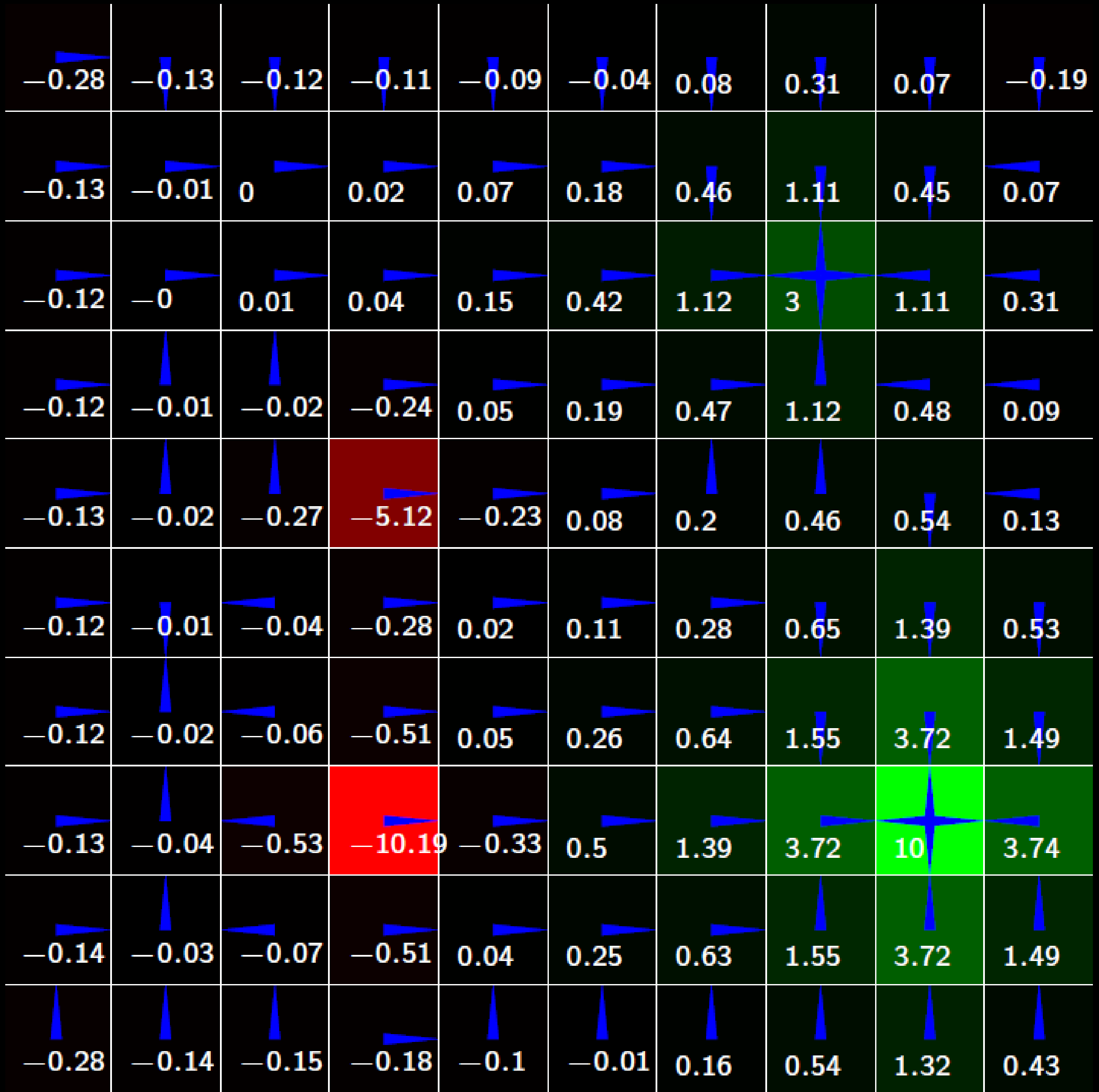

0.41 0.74 0.96 1.18 1.43 1.71 1.98 2.11 2.39 2.09
0.74 1.04 1.27 1.52 1.81 2.15 2.47 2.58 3.02 2.69
0.86 1.18 1.45 1.76 2.15 2.55 2.97 3 3.69 3.32
0.84 1.11 1.31 1.55 2.45 3.01 3.56 4.1 4.53 4.04
0.91 1.2 1.09 −3 2.48 3.53 4.21 4.93 5.5 4.88
1.1 1.46 1.79 2.24 3.42 4.2 4.97 5.85 6.68 5.84
1.06 1.41 1.7 2.14 3.89 4.9 5.85 6.92 8.15 6.94
0.92 1.18 0.7 −7.39 3.43 5.39 6.67 8.15 10 8.19
1.09 1.45 1.75 2.18 3.89 4.88 5.84 6.92 8.15 6.94
1.07 1.56 2.05 2.65 3.38 4.11 4.92 5.83 6.68 5.82

Converged!

$\gamma = 0.9$



$\gamma = 0.5$



Decision-Making Summary

- Given an MDP, we defined **Expected Cumulative Payoff,** which plays a key role in **optimizing actions over planning horizons**
- Used **value iteration** to determine the “value” of a particular state, which helps us determine the best action to take considering future payoff
- We generally assumed the transition and reward function are known exactly – but what if we don’t have access to this information?
  - Will post notes on basic Q-learning for RL!

Today's Plan

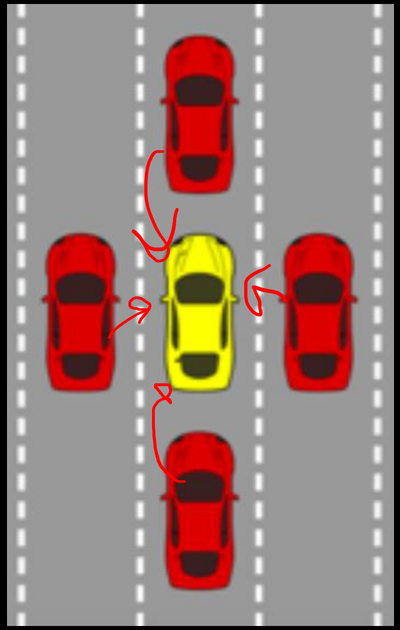
- MDP Policies and Value Iteration
- Simple Example
- Introduction to Safety
- Responsibility Sensitive Safety

(Safety) Challenges for AVs

- Each module is challenging to develop
- AVs are safety critical systems, but:
  - It is impossible to guarantee zero accidents
  - Statistical approaches are problematic
  - Any software or hardware update requires new certification
- Scalability remains a challenge
  - How to mass produce expensive computation and sensors?
  - How to ensure you can drive everywhere?

How to ensure multi-agent safety?

Absolute safety is impossible ☹



Statistical guarantees are infeasible ☹

Typically, to show p_1 likelihood of failure, we aim to gather at least $\frac{1}{p_1}$ samples

- Are all miles equal?
- 10 disengagements per 300 million miles is not enough
- Any change to the software will require a new data collection

On-Road Testing

- Data required to guarantee 10^{-9} probability of failure: 10^9 hours of driving or 30 billion miles
- However, this is insufficient to show the difference between an AV with error rate of p_1 and a perfect system
  - We want system with 10^{-9} accident rate per hour, but we have an AV system with 10^{-8} accident rate per hour.
  - If we drive for 10^8 hours, there is still a constant probability that the testing process is not telling us the true accident rate

→ Purely data-based approach for safety is naïve at best

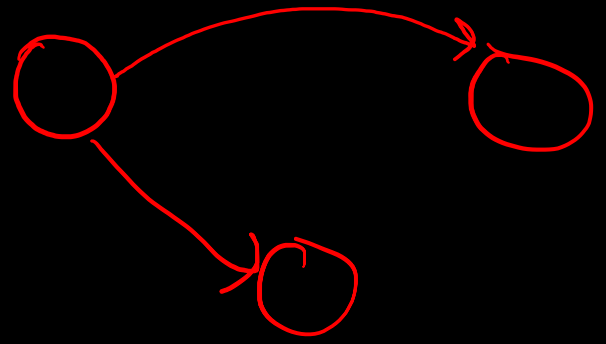
Safety Guarantees Automata

- An **automata** is a mathematical model for describing computations or processes evolving in discrete steps
- ***States*** can be discrete or continuous valued
- State ***transitions*** define how the states can change
  - Transitions can be ***non-deterministic:*** multiple next states from a single state
- No inherent notion of *time*, but each transition can be thought of as passage of a fixed amount of time

An automaton is a tuple $\mathcal{A}=\langle Q, \Theta, A, \mathcal{D}\rangle$ where

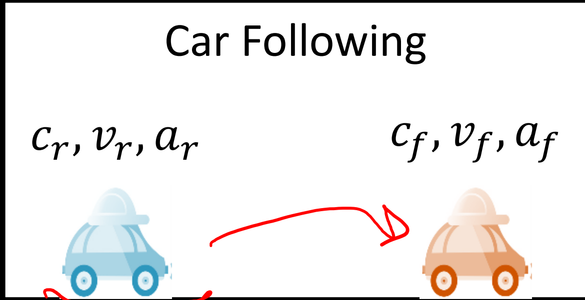
- Q is a set of states
- $\Theta \subseteq Q$ is the set of initial or start states
- A is a set of actions or labels
- $\mathcal{D} \subseteq Q \times A \times Q$ is the set of transitions

  → A transition can be thought of as a triple (u, a, u')



Safety Guarantees: Automata

- $a_f[t] = a_{min}$
- $v_f[t+1] = v_f[t] + a_f[t]\Delta$
- $c_f[t+1] = c_f[t] + v_f[t]\Delta + \frac{1}{2}a[t]\Delta^2$



- $a_r[t] \in [a_{min}, a_{max}]$
- $v_r[t+1] = v_r[t] + a_r[t]\Delta$
- $c_r[t+1] = c_r[t] + v_r[t]\Delta + \frac{1}{2}a_r[t]\Delta^2$

- $X = \{a_f, v_f, c_f, a_r, v_r, c_r\}$
- $Q = \mathbb{R}^6$
- $A = (a_1, a_2)$ the acceleration choices
- $\mathcal{D} \subseteq \mathbb{R}^6 \times A \times \mathbb{R}^6$

Executions, Reachability, & Invariants

- An **execution** of $\mathcal{A}$ is an alternating (possibly infinite) sequence of states s_t and actions u_t: $\alpha = s_0 u_1 s_1 u_2 s_2 \ldots$ such that:
  - $s_0 \in \Theta$
  - $\forall\, i$ in the sequence the transition function is applied $(s_i \xrightarrow{u_{i+1}} s_{i+1})$

$s_t = f(s_{t-1}, u_{t-1})$

- A state s is **reachable** if there exists an execution that ends at s

→ The set of reachable states is denoted by $Reach_A$

Formal Invariants

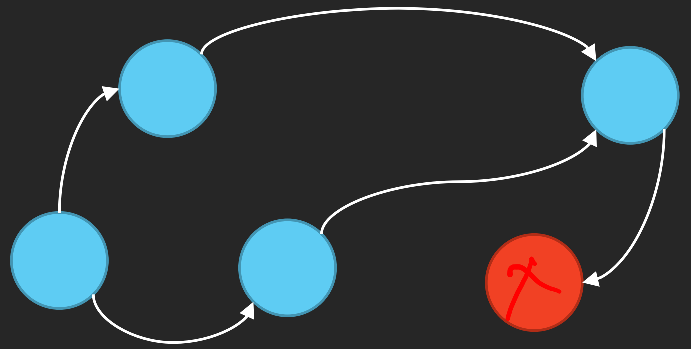
- What does it mean for I to hold "always" for $\mathcal{A}$?
  - I holds at all states along any execution $s_0 u_1 s_1 u_2 s_3$
  - I holds in all reachable states of $\mathcal{A}$
  - $\boldsymbol{Reach}_{\mathcal{A}} \subseteq [[\boldsymbol{I}]]$
- Invariants capture most properties that you will encounter in practice
  - safety: "aircraft always maintain separation"
  - bounded reaction time: "within 15 seconds of press, light must turn to walk"
- How to verify if I is an invariant?
  - Does there exist reachable state s such that $s \nvDash I$?

Reachability Problem

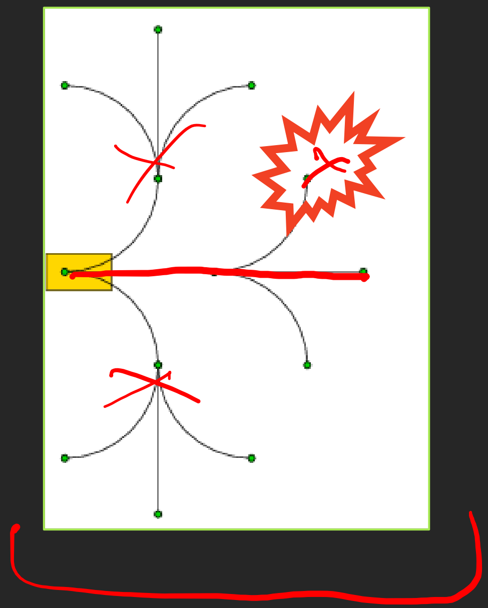
Automata

Given a directed graph $G = (V, E)$, and two sets of vertices $S, T \subseteq V$, T **is reachable from** S if there is a path from S to T.

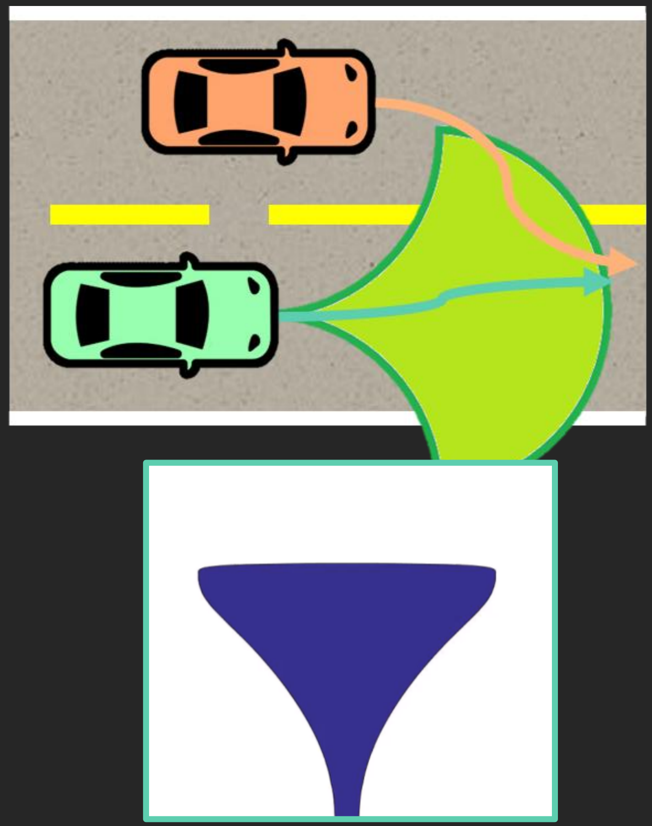
Reachability Problem (G, S, T):
is T is reachable from S in G?



Reachability Trees

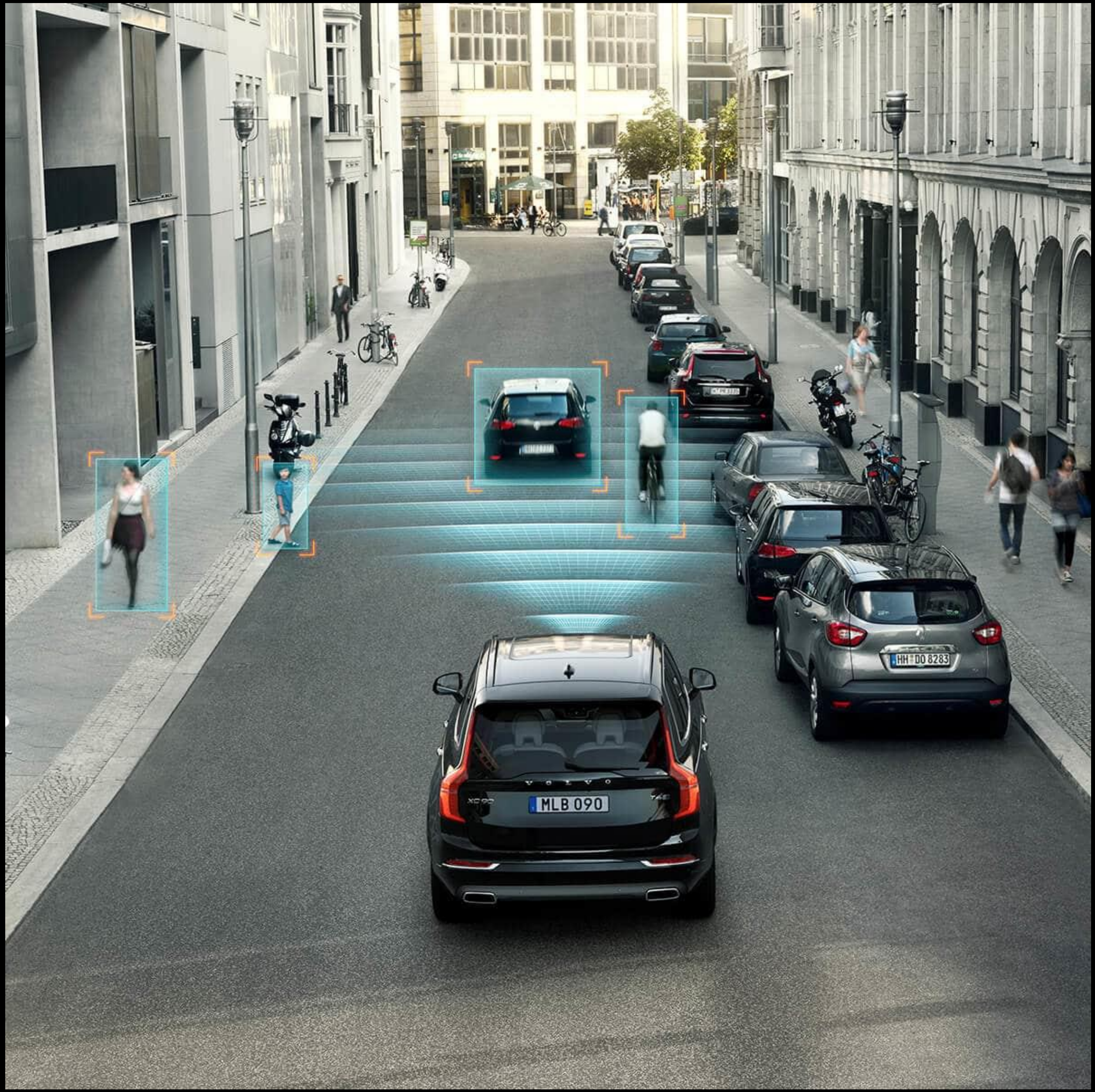


Reachable Sets



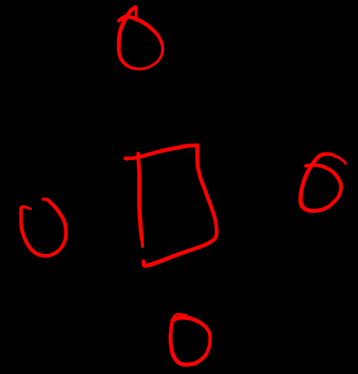
City Safety: Emergency Braking

Image Credit: Volvo



Responsibility Sensitive Safety

developed by Intel / MobilEye



Instead of looking at absolute safety, introduce a safety notion that depends on *responsibility*

→ AV should never be responsible for an accident

<u>RSS Rules:</u>

1. Keep a safe longitudinal distance from the car ahead.
2. Keep a safe lateral distance from the cars on your sides.
3. Respect right-of-way rules (multiple geometries, traffic lights, pedestrians, unstructured roads).
4. Be cautious of occluded areas.

RSS Example: Safe Following Distance

The longitudinal distance between a car (c_r) that drives behind another car (c_f) is safe w.r.t. a response time ρ if:

- c_f applies at most a_{max}^{brake}
- c_r will apply at most a_{max}^{accel} during response time
- After ρ, c_r will brake by at least a_{min}^{brake} until full stop

→ c_r will not collide with c_f

Remarks:

1. This is basic reachability!
2. The safe distance depends on a set of parameters that can be determined by regulation.
3. The parameters can be different for a robotic car and a human driver.
4. The parameters can be different for different road conditions.

RSS Example: Safe Following Distance

- Let v_r, v_f be the longitudinal velocities of the cars
- The minimal safe distance is:

$$d_{min} = \left[v_r \cdot \rho + \frac{1}{2} a_{max}^{\text{accel}} \cdot \rho^2 + \frac{\left(v_r + \rho \cdot a_{max}^{\text{accel}}\right)^2}{2a_{min}^{\text{brake}}} - \frac{v_f^2}{2a_{max}^{\text{brake}}} \right]_+$$

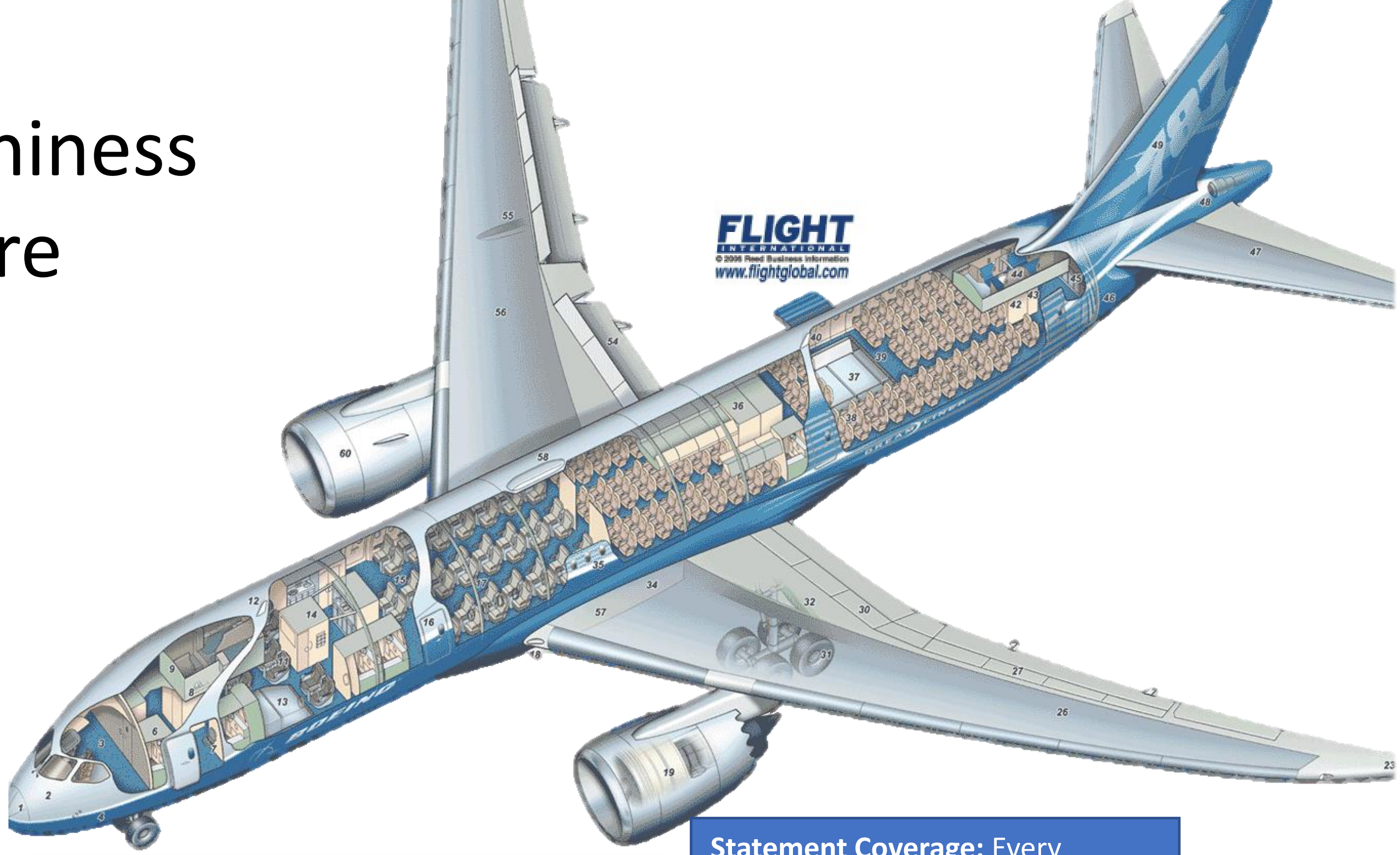
Summary

- Discussed the challenges with defining safety and introduced the ideas behind verification and formal guarantees for safety
  - Reachability is as key tool for guaranteeing specified properties
- While useful, often require well-defined specifications, require high-fidelity models (and assumptions), consider worst case scenarios, and/or only assess one piece of the puzzle
  - Making formal guarantees on the full autonomous stack remains a challenge
- Introduced RSS, a framework that aims to define safety in an intuitive manner

Extra Slides

Certifying airworthiness of aviation software

What fraction of the cost of developing a new aircraft is in SW?



DO178C

Primary document by which FAA & EASA approves software-based aerospace systems.

DAL establishes the rigor necessary to demonstrate compliance

Dev.Assurance Level (DAL)	Hazard Classification	Objectives
A	Catastrophic	71
B	Hazardous	69
C	Major	62
D	Minor	26
E	No Effect	0

Statement Coverage: Every statement of the source code must be covered by a test case

Condition Coverage: Every condition within a branch statement must be covered by a test case

"Special credits": For using formal methods based tools recently introduced

Safety Certification (cont)

- A component's DAL level is determined from a *safety assessment process and hazard analysis through examination of the effects of a failure condition in the system. The failure conditions are categorized by their effects on the aircraft, crew, and passengers.*
  - For example, Level A is assigned for "Catastrophic Outcome," and Level E is for "No Safety Effect."
- DAL level then establishes the rigor necessary to demonstrate compliance with DO-178C
- E.g., components that command, control, or monitor safety critical functions are Level A. The standard requires any Level A software to be tested to cover every statement, branch, and function call, and also to pass the Modified Condition Decision Coverage (MC/DC) tests
  - Requires that (i) each entry and exit point in the code be invoked, (ii) each decision take every possible value, and (iii) each condition in a decision take every possible value
  - For certain levels, DO-178C requires that the testing, verification, and validation be performed by a team that is independent of the software development team
- Again, certification is process-based and does not eliminate possibility of bad logic, interference in the the control code
- Dozens of commercial tools (e.g., MATLAB, Esterel, Cantata, VectorCAST, Rapita Systems, and CodeSonar) can support DO-178C certification by applying **formal verification**.

The formal verification problem

Example requirements:

Safety: “For all nominal behaviors of the car, the separation between the cars must be always > 1 m”

Efficiency: “For all nominal driver inputs, the air-fuel ratio must be in the range [1,4]”

Privacy: “Using GPS does not compromise user’s location”

Fairness: “Similar people’s loan approval are decided similarly”