Lecture 17: Planning IV (Decision-Making II)

Professor Katie Driggs-Campbell

April 11, 2024

ECE484: Principles of Safe Autonomy

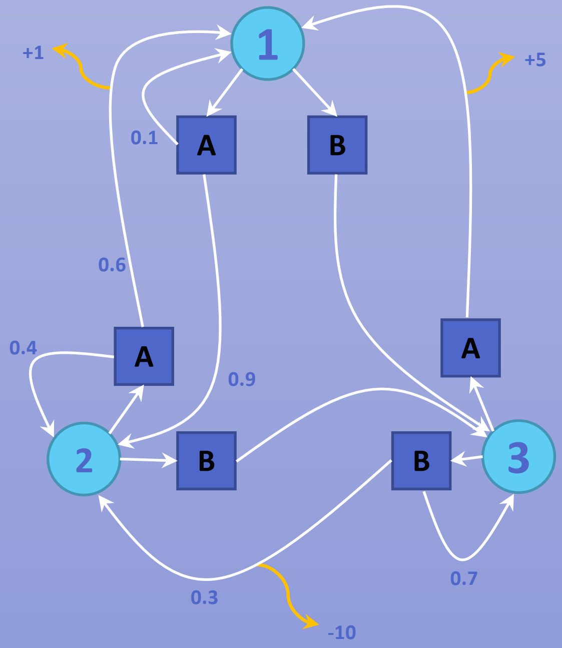
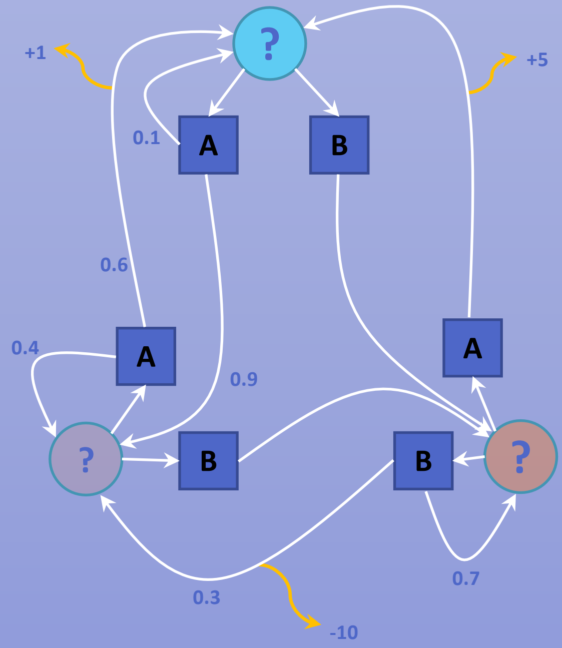
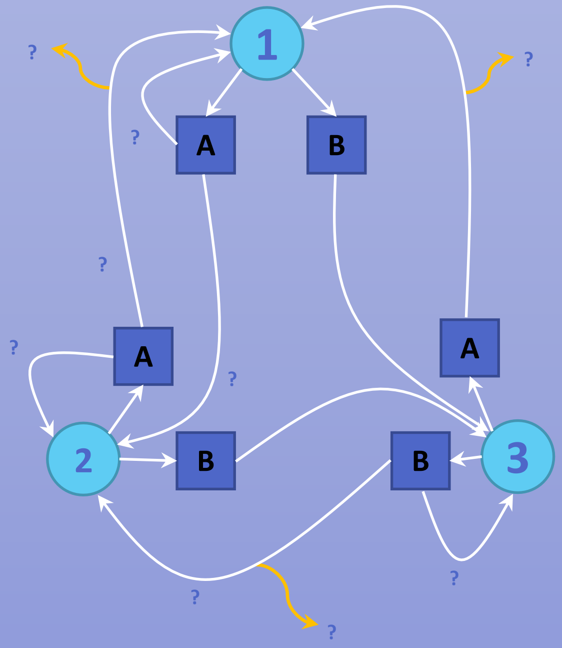
Administrivia

- Upcoming due dates:
  - Final Presentations in class on 4/23 and 4/25
  - Final Video due 5/3
- Exam on 4/18 at 7pm
  - Email me ASAP about conflict exams
  - Make reservation in testing center for DRES accommodations
  - No cheatsheet will be needed
  - CA review session on Friday 4/12 (HW party time)
  - In-class review session on Tuesday 4/16
- Prof. DC OH by appointment next week (4/16)
  - Otherwise in 260 CSL

Today's Plan

- Possible solutions for decision-making
- Markov Decision Processes
- MDP Policies and Value Iteration

Markov Models

Markov Decision Processes (MDP)	Partially Observable MDP	Reinforcement Learning
		
Uncertainty in effects of actions	Uncertainty in current state	Uncertainty in model

Decision-Making Policies

- We want to devise a scheme that generates actions to optimize the future payoff *in expectation*

Decision-Making Policies

- We want to devise a scheme that generates actions to optimize the future payoff *in expectation*
- Policy: $\pi : x_t \rightarrow u_t$
  - Maps states to actions
  - Can be low-level reactive algorithm or a long-term, high-level planner
  - May or may not be deterministic

Decision-Making Policies

- We want to devise a scheme that generates actions to optimize the future payoff *in expectation*
- Policy: $\pi : x_t \rightarrow u_t$
  - Maps states to actions
  - Can be low-level reactive algorithm or a long-term, high-level planner
  - May or may not be deterministic
- Typically, we want a policy that optimizes <u>future</u> payoff, considering optimal actions over a <u>planning (time) horizon</u>

MDP Policies

- Policies map states to actions

$$\pi: x \to u$$

- We want to find a policy that maximizes future pay off
  - Suppose $T = 1$: $\pi_1(x) = \operatorname{argmax}_u \ r(x, u)$

MDP Policies

- Policies map states to actions

  $$\pi: x \to u$$

- We want to find a policy that maximizes future pay off
  - Suppose $T = 1$: $\qquad \pi_1(x) = \text{argmax}_u \; r(x, u)$
- 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*

Expected Cumulative Payoff

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

Expected Cumulative Payoff

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

1. Greedy case: $T = 1$
   → Optimize next payoff

Expected Cumulative Payoff

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

1. Greedy case: $T = 1$
   → Optimize next payoff
2. Finite Horizon: $1 \leq T < \infty, (\gamma \leq 1)$
   → Optimize R_T for set time window

Expected Cumulative Payoff

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

1. Greedy case: $T = 1$
   → Optimize next payoff
2. Finite Horizon: $1 \leq T < \infty, (\gamma \leq 1)$
   → Optimize R_T for set time window
3. Infinite Horizon: $T = \infty, (\gamma < 1)$
   → Optimize R_∞ for all time
   If $|r| \leq r_{max}$, discounting guarantees R_∞ is finite

$$R_\infty \leq r_{max} + \gamma r_{max} + \gamma^2 r_{max} + \cdots = \frac{r_{max}}{1-\gamma}$$

Value Functions

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

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

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 $\hat{V}$
  - $\hat{V}(x) \leftarrow r_{min}, \forall x$

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 $\gamma < 1$

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:
  - $\pi(x) = \mathrm{argmax}_u \left[r(x,u) + \int \hat{V}(x')p(x'|x,u)dx' \right]$

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:
  - $\pi(x) = \mathrm{argmax}_u \left[r(x,u) + \int \hat{V}(x')p(x'|x,u)dx' \right]$
- Often, we use the discrete version:
  - $\pi(x) = \mathrm{argmax}_u \left[r(x,u) + \sum_{x'} \hat{V}(x')p(x'|x,u) \right]$

A Simple MDP

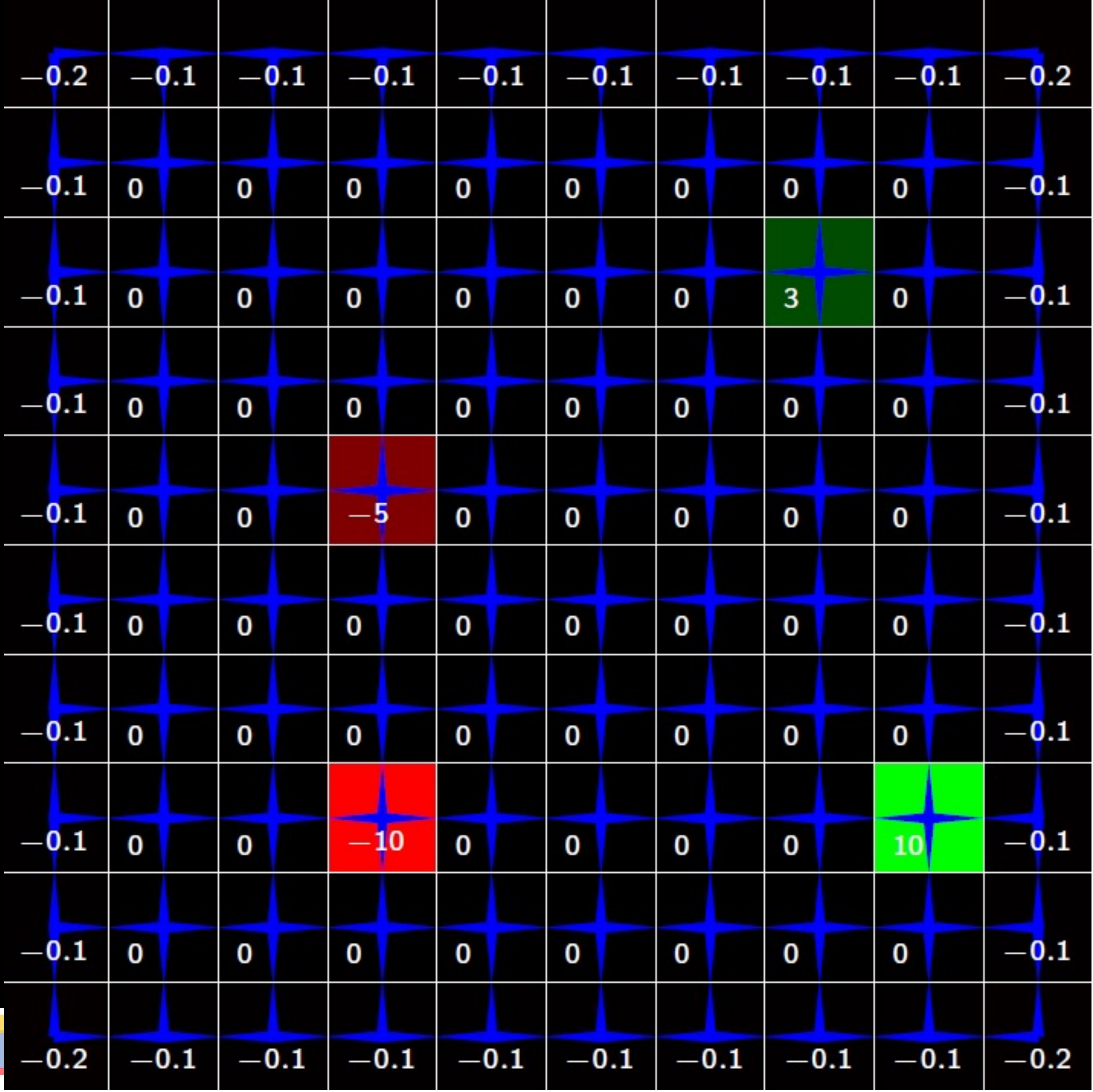
Example: Value Iteration Setup

Example: Value Iteration

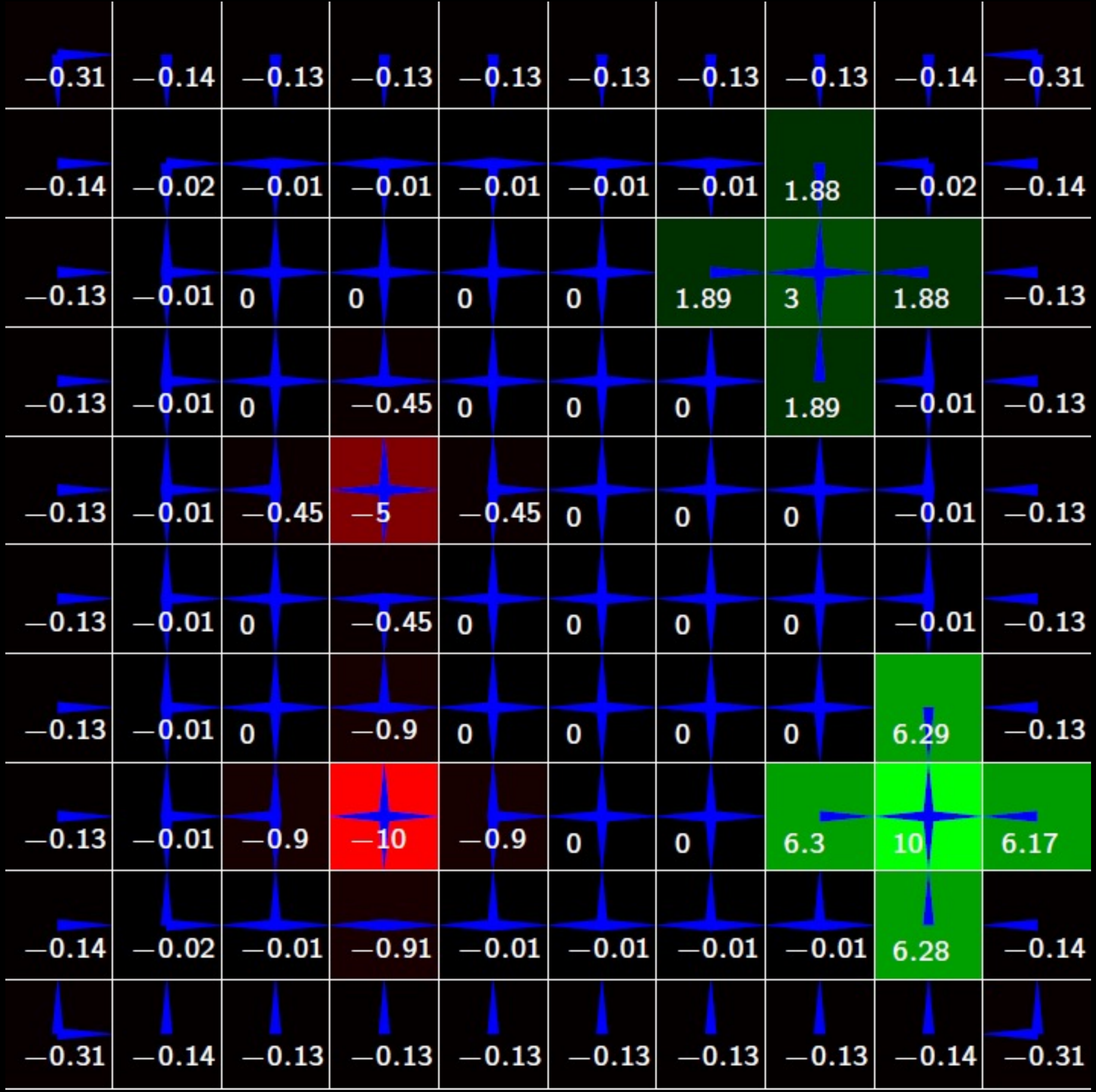
Example: Rewards, Values, Policy

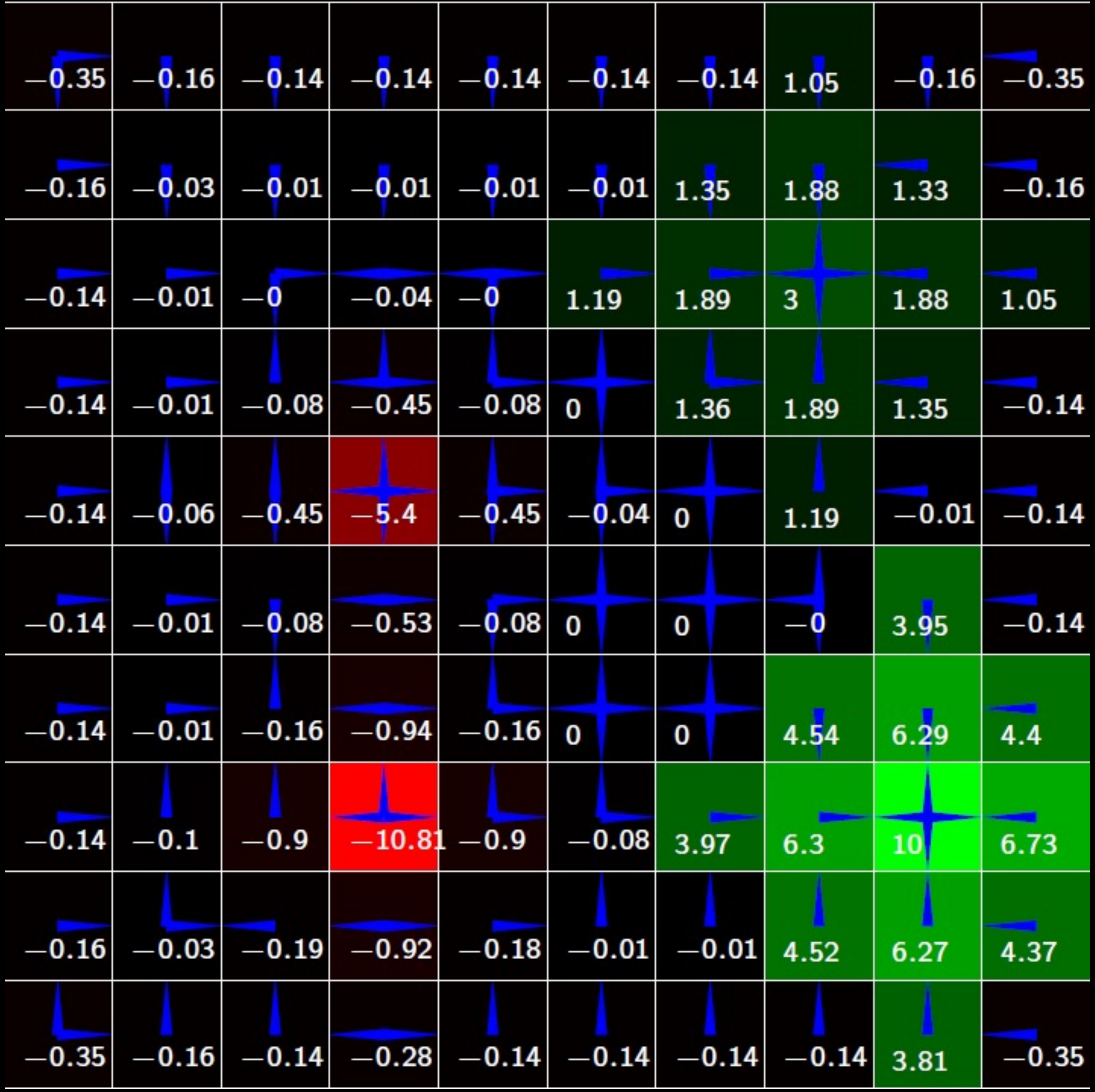
Grid world example

- 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

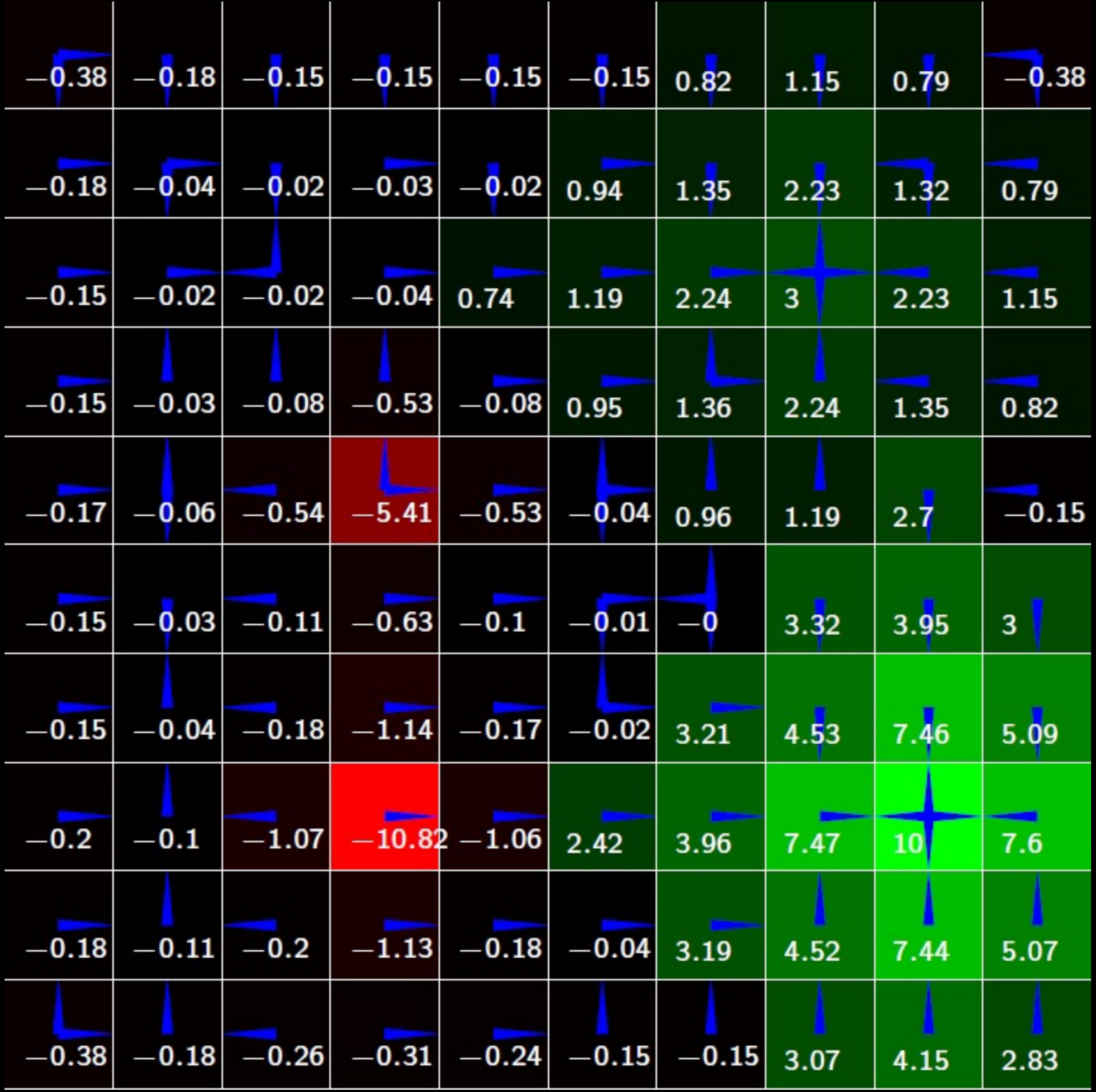


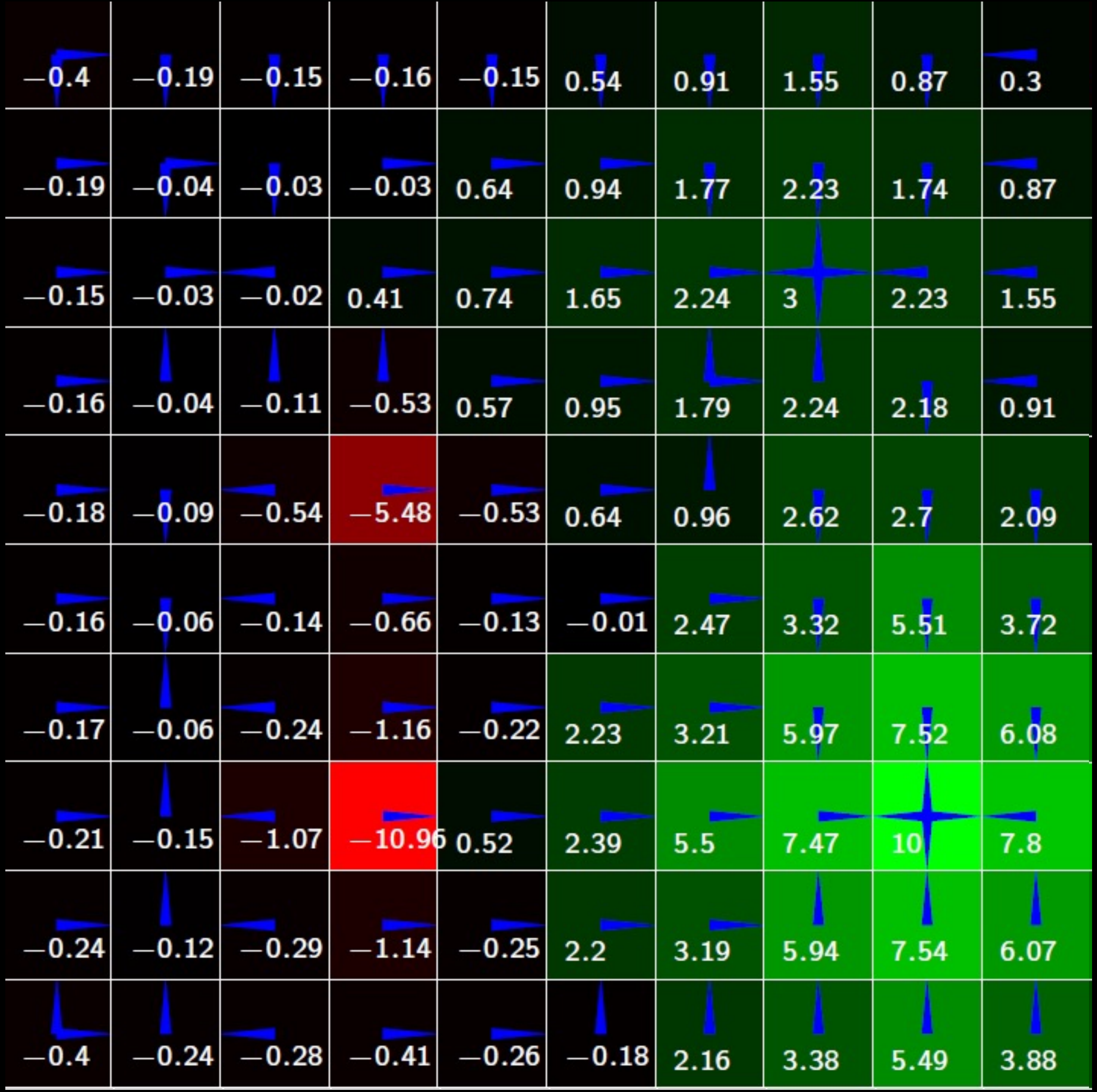
−0.2	−0.1	−0.1	−0.1	−0.1	−0.1	−0.1	−0.1	−0.1	−0.2
−0.1	0	0	0	0	0	0	0	0	−0.1
−0.1	0	0	0	0	0	0	3	0	−0.1
−0.1	0	0	0	0	0	0	0	0	−0.1
−0.1	0	0	−5	0	0	0	0	0	−0.1
−0.1	0	0	0	0	0	0	0	0	−0.1
−0.1	0	0	0	0	0	0	0	0	−0.1
−0.1	0	0	−10	0	0	0	0	10	−0.1
−0.1	0	0	0	0	0	0	0	0	−0.1
−0.2	−0.1	−0.1	−0.1	−0.1	−0.1	−0.1	−0.1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

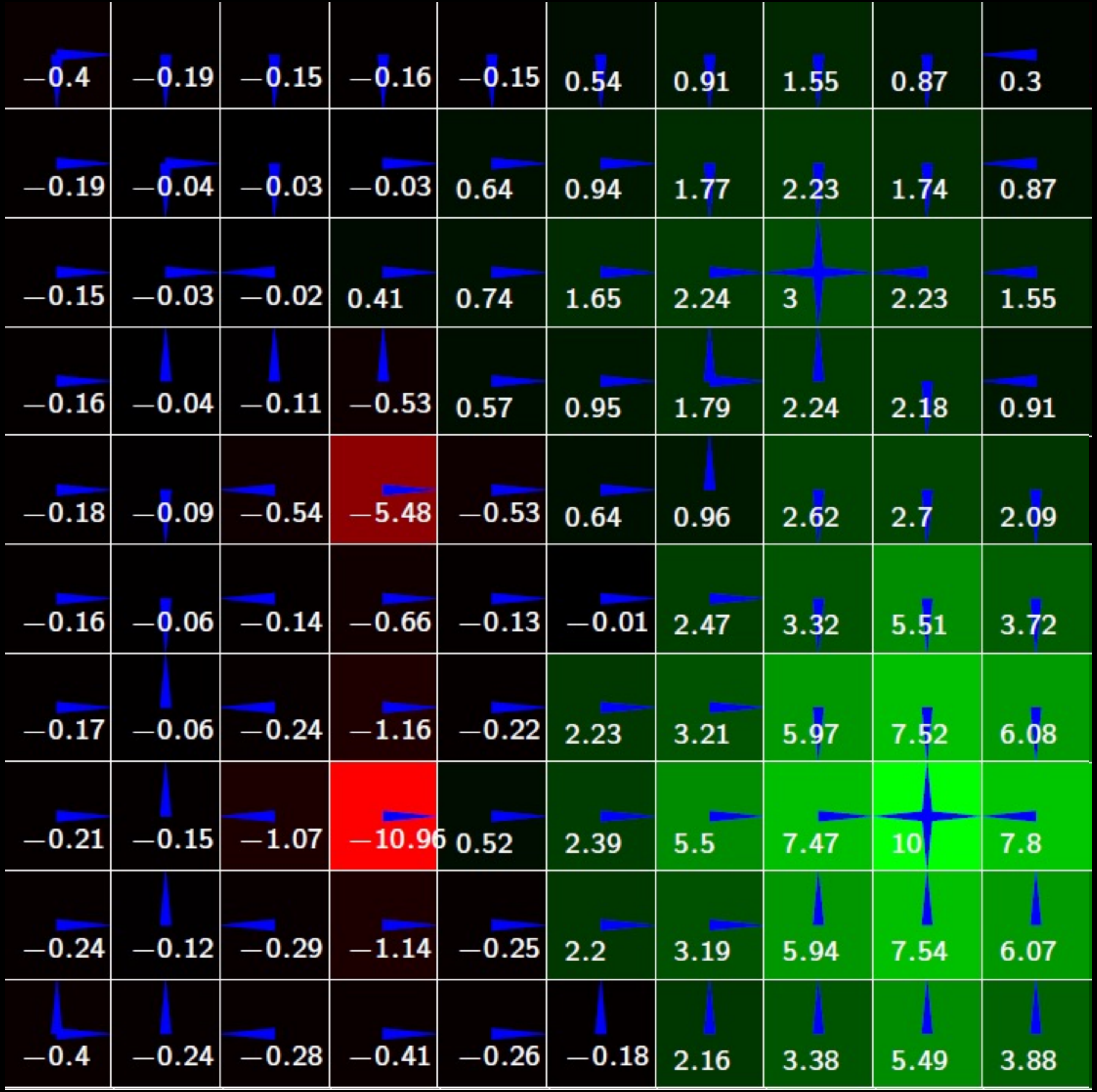

−0.31 −0.14 −0.13 −0.13 −0.13 −0.13 −0.13 −0.13 −0.14 −0.31
−0.14 −0.02 −0.01 −0.01 −0.01 −0.01 −0.01 1.88 −0.02 −0.14
−0.13 −0.01 0 0 0 0 1.89 3 1.88 −0.13
−0.13 −0.01 0 −0.45 0 0 0 1.89 −0.01 −0.13
−0.13 −0.01 −0.45 −5 −0.45 0 0 0 −0.01 −0.13
−0.13 −0.01 0 −0.45 0 0 0 0 −0.01 −0.13
−0.13 −0.01 0 −0.9 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



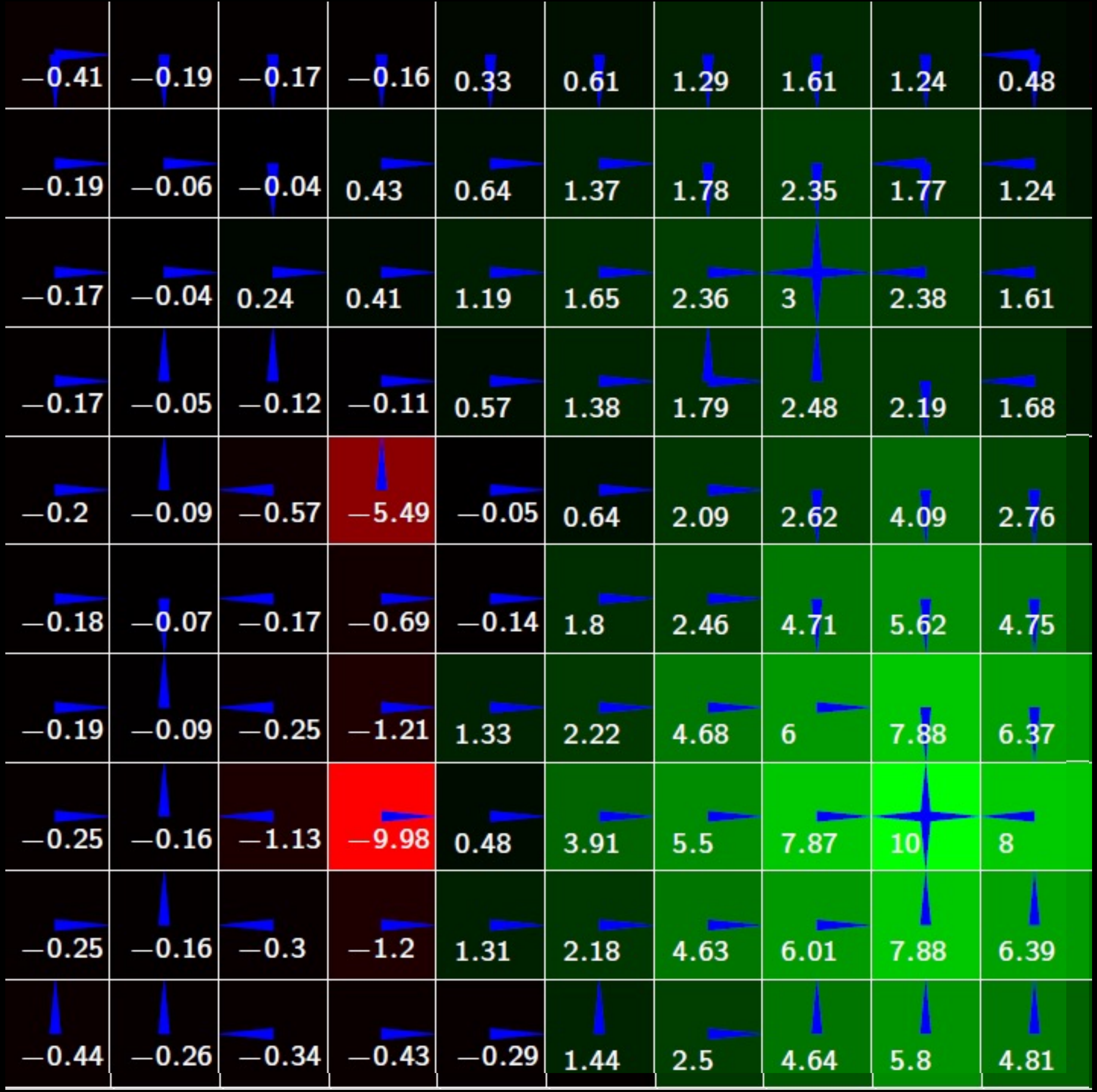
-0.35
-0.16
-0.14
-0.14
-0.14
-0.14
-0.14
1.05
-0.16
-0.35
-0.16
-0.03
-0.01
-0.01
-0.01
-0.01
1.35
1.88
1.33
-0.16
-0.14
-0.01
-0
-0.04
-0
1.19
1.89
3
1.88
1.05
-0.14
-0.01
-0.08
-0.45
-0.08
0
1.36
1.89
1.35
-0.14
-0.14
-0.06
-0.45
-5.4
-0.45
-0.04
0
1.19
-0.01
-0.14
-0.14
-0.01
-0.08
-0.53
-0.08
0
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

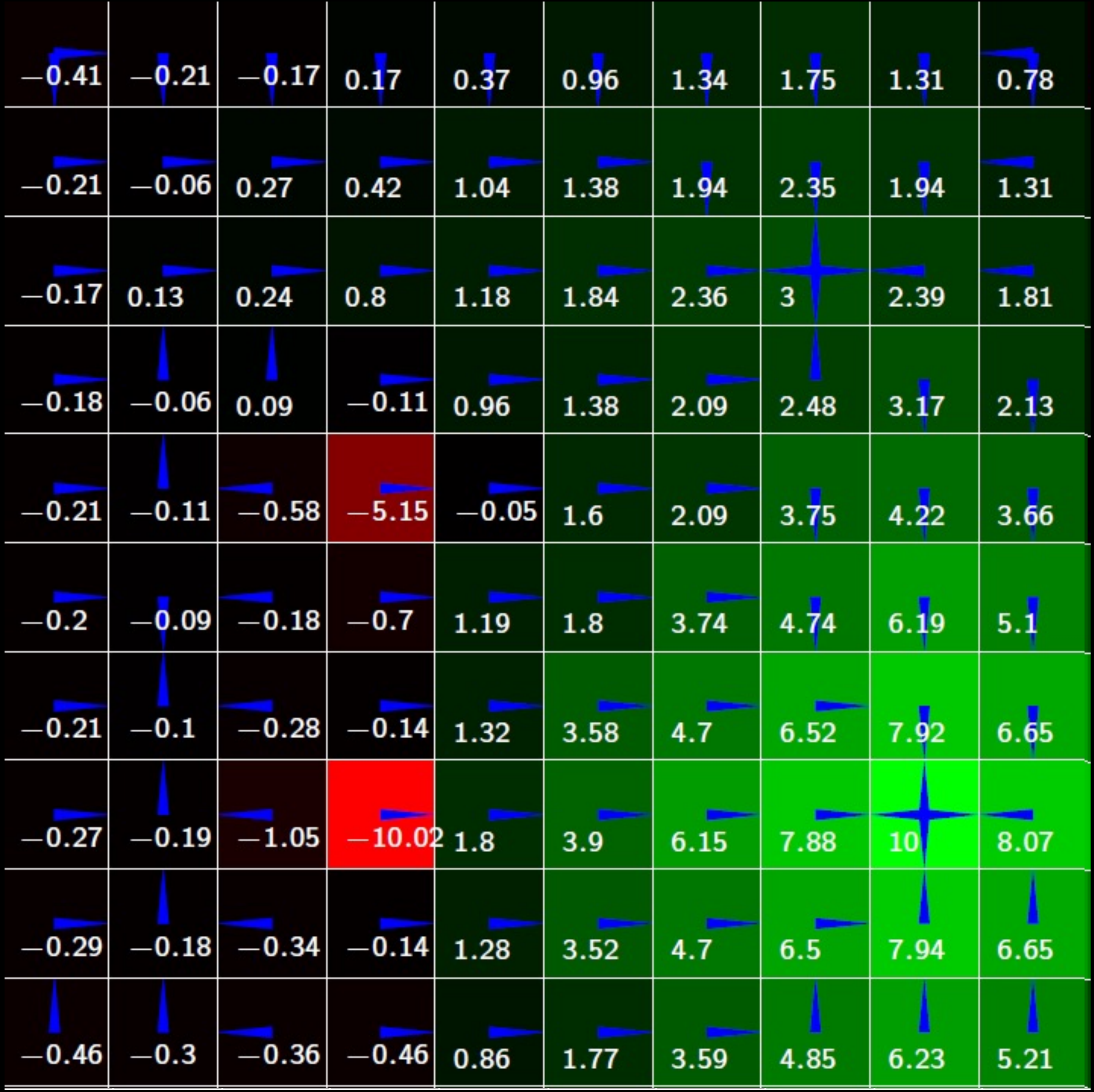

−0.38 −0.18 −0.15 −0.15 −0.15 −0.15 0.82 1.15 0.79 −0.38
−0.18 −0.04 −0.02 −0.03 −0.02 0.94 1.35 2.23 1.32 0.79
−0.15 −0.02 −0.02 −0.04 0.74 1.19 2.24 3 2.23 1.15
−0.15 −0.03 −0.08 −0.53 −0.08 0.95 1.36 2.24 1.35 0.82
−0.17 −0.06 −0.54 −5.41 −0.53 −0.04 0.96 1.19 2.7 −0.15
−0.15 −0.03 −0.11 −0.63 −0.1 −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

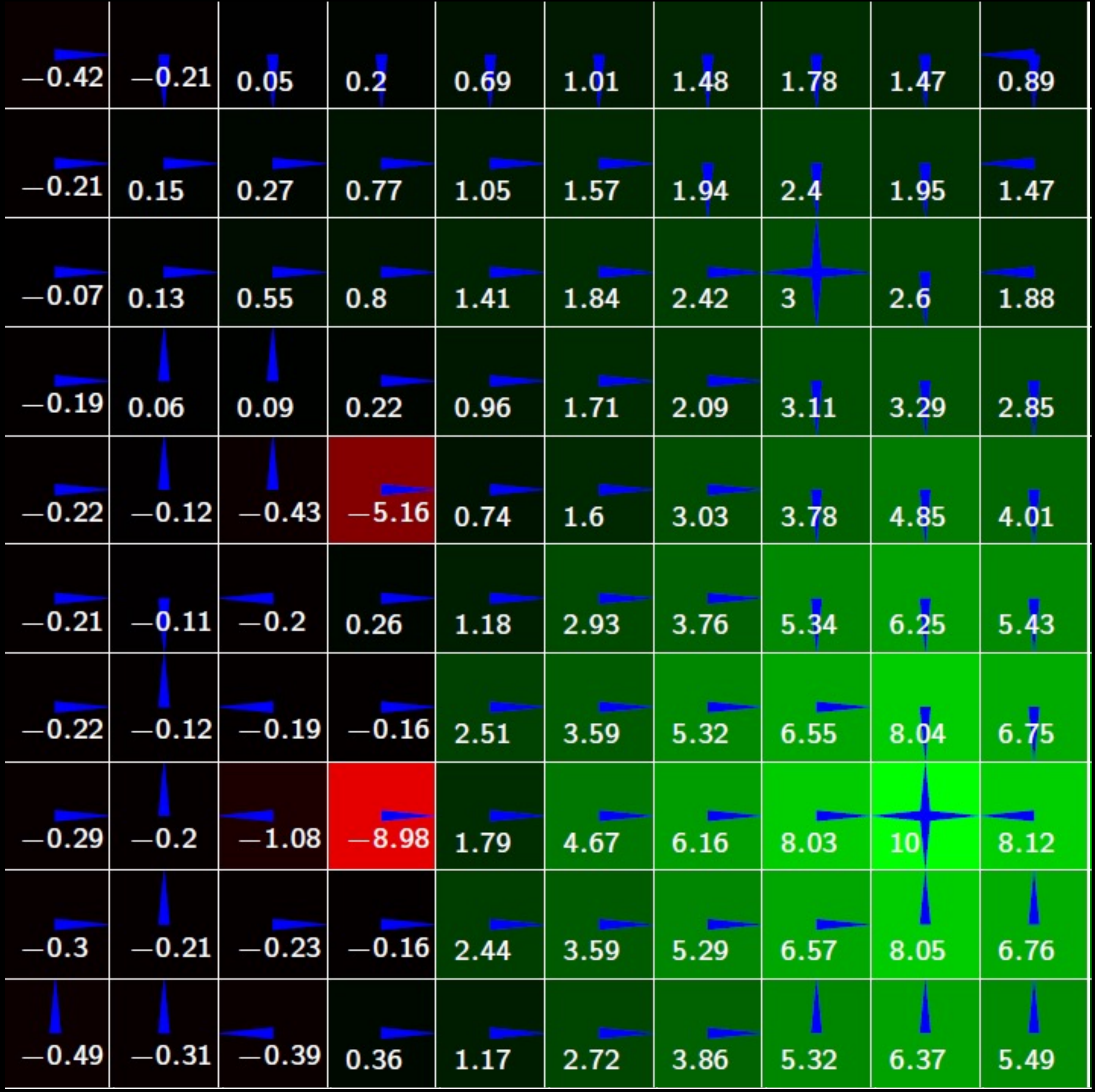

−0.4 −0.19 −0.15 −0.16 −0.15 0.54 0.91 1.55 0.87 0.3
−0.19 −0.04 −0.03 −0.03 0.64 0.94 1.77 2.23 1.74 0.87
−0.15 −0.03 −0.02 0.41 0.74 1.65 2.24 3 2.23 1.55
−0.16 −0.04 −0.11 −0.53 0.57 0.95 1.79 2.24 2.18 0.91
−0.18 −0.09 −0.54 −5.48 −0.53 0.64 0.96 2.62 2.7 2.09
−0.16 −0.06 −0.14 −0.66 −0.13 −0.01 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

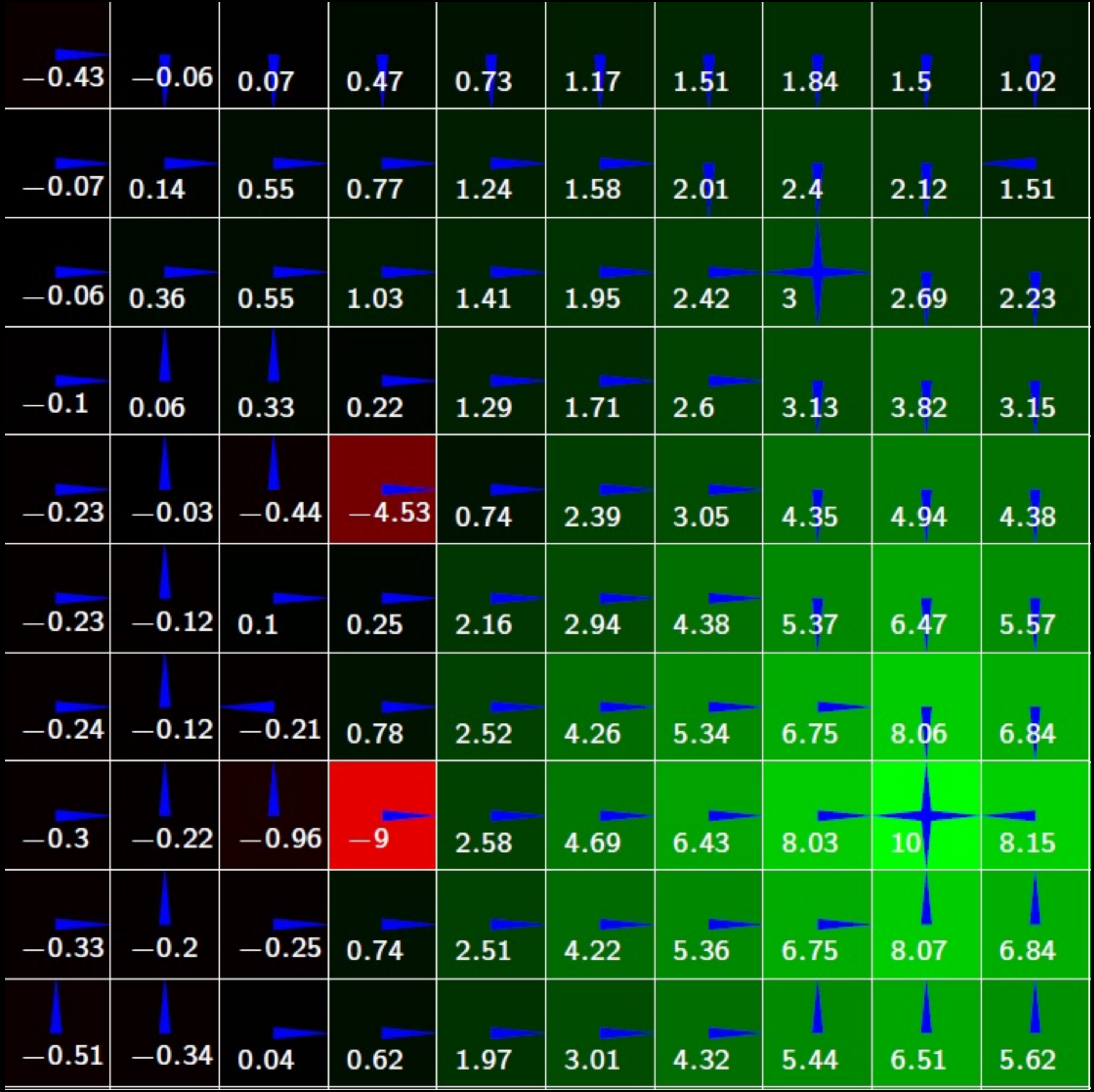


-0.4 -0.19 -0.15 -0.16 -0.15 0.54 0.91 1.55 0.87 0.3
-0.19 -0.04 -0.03 -0.03 0.64 0.94 1.77 2.23 1.74 0.87
-0.15 -0.03 -0.02 0.41 0.74 1.65 2.24 3 2.23 1.55
-0.16 -0.04 -0.11 -0.53 0.57 0.95 1.79 2.24 2.18 0.91
-0.18 -0.09 -0.54 -5.48 -0.53 0.64 0.96 2.62 2.7 2.09
-0.16 -0.06 -0.14 -0.66 -0.13 -0.01 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!

$\gamma = 0.9$

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

$\gamma = 0.5$

−0.28	−0.13	−0.12	−0.11	−0.09	−0.04	0.08	0.31	0.07	−0.19
−0.13	−0.01	0	0.02	0.07	0.18	0.46	1.11	0.45	0.07
−0.12	−0	0.01	0.04	0.15	0.42	1.12	3	1.11	0.31
−0.12	−0.01	−0.02	−0.24	0.05	0.19	0.47	1.12	0.48	0.09
−0.13	−0.02	−0.27	−5.12	−0.23	0.08	0.2	0.46	0.54	0.13
−0.12	−0.01	−0.04	−0.28	0.02	0.11	0.28	0.65	1.39	0.53
−0.12	−0.02	−0.06	−0.51	0.05	0.26	0.64	1.55	3.72	1.49
−0.13	−0.04	−0.53	−10.19	−0.33	0.5	1.39	3.72	10	3.74
−0.14	−0.03	−0.07	−0.51	0.04	0.25	0.63	1.55	3.72	1.49
−0.28	−0.14	−0.15	−0.18	−0.1	−0.01	0.16	0.54	1.32	0.43

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?

Return to Safety

How would you incorporate desirable behaviors and safety requirements into your AV using the various decision-making frameworks?

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

RSS Rules:

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

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:

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

Bonus MP4: Designing Decision-Logic

Task 1: Emergency Braking

Task 2: Stopping at a Red Light

Bonus MP4: Designing Decision-Logic

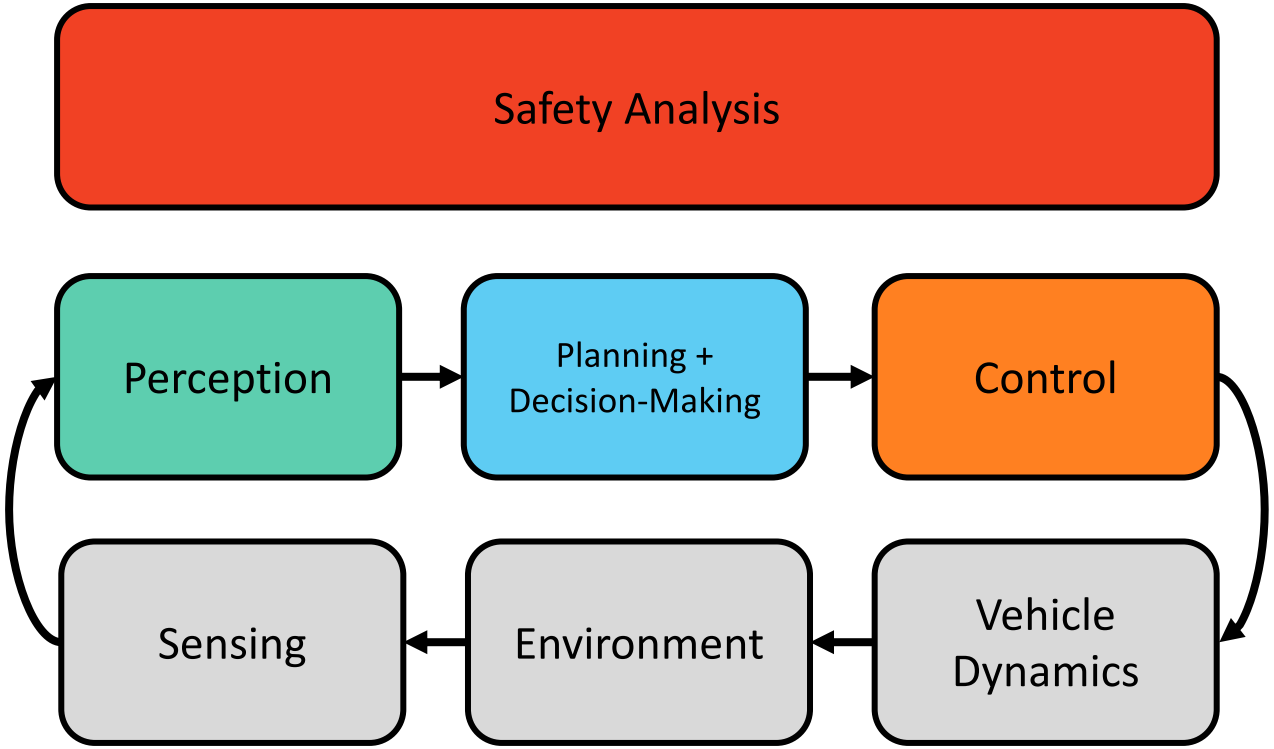
Validation

- Define test cases
- Gather samples through simulation
- Collect statistical evidence of safety

Verification

- Define uncertainty
- Compute reachable sets to assess safety and provide guarantees

Course Recap



Why are we here?

Know — Components of an autonomous system and safety standards.
→ How to use software modules for perception, planning, control, ROS, OpenCV, …

Do — Code and analyze algorithms for perception, localization, planning, control, & verification
→ Plan, propose, organize and execute a team project

Understand — Models, algorithms, data, biases, assumptions for building trustworthy autonomous systems
→ Theoretical properties of algorithms and their limitations

Get Inspired — Become the Isaac Newton of Autonomy
→ "To do things right, first you need love, then technique." – Antoni Gaudí