<CAPS>

Session 4 – Goal Specification and Reward Design

Leo Klenner, Henry Fung, Cory Combs

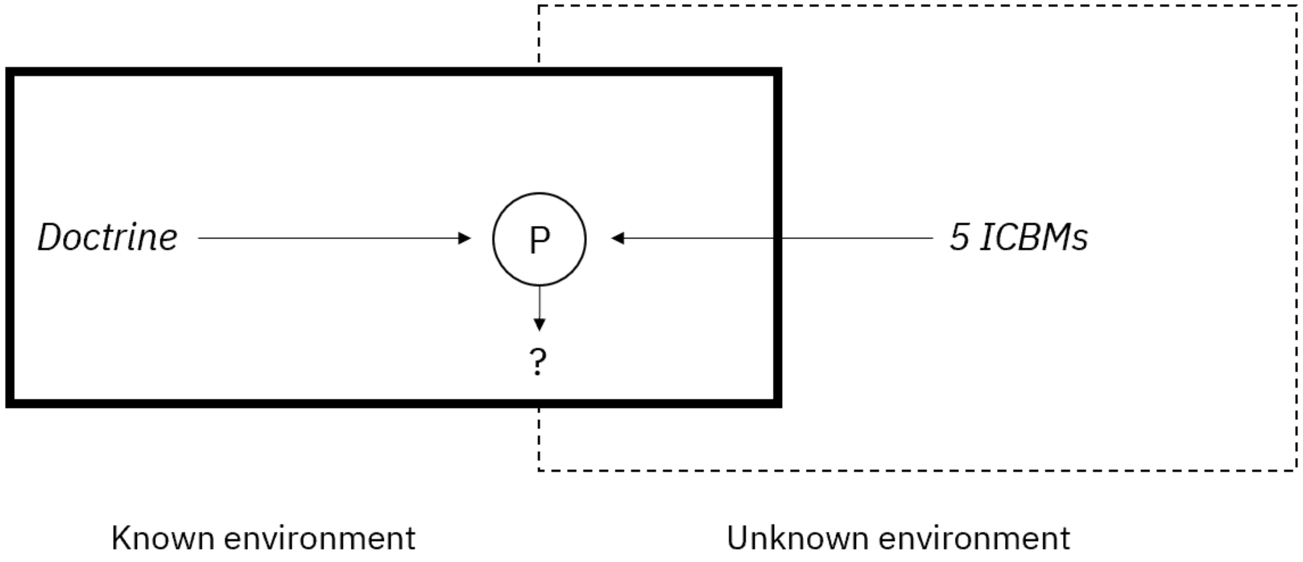
Outline

1. Recap
2. Goal Specification in the 1983 Nuclear Incident
3. Why Reward Design is Hard
4. Exercise on Rule/Reward Design
5. Two Failure Modes of Reward Design in Reinforcement Learning
6. Inverse Methods

Goal Specification in the 1983 Soviet Nuclear Incident

> September 26, 1983
> Five U.S. ICBMs surface on Soviet radar, station monitored by Stanislav Petrov
> Petrov reasons that this is a false alarm, decides not to launch counter-attack

> How can we specify Petrov's **goals**, **rewards** and **rules**?

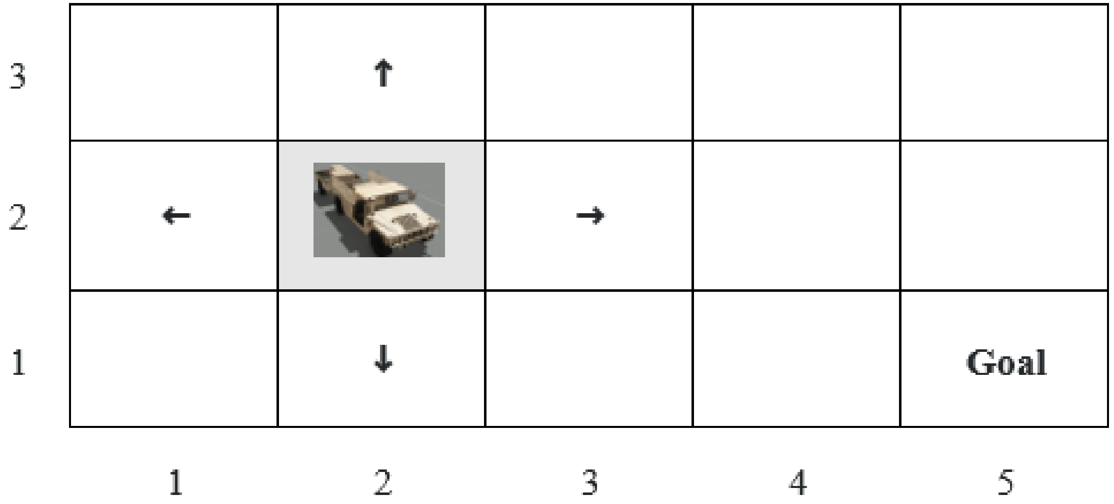


Why Reward Design is Hard

- **Premise**
  - Human designs scalar reward signal that is generated by the environment
  - Reward signal should lead the agent to converge to the designer‘s *desired* outcome
  - Design process often trial-and-error search
- **Challenges**
  - Goal might be difficult to translate into reward signal
  - Agents might find undesirable ways to make the environment deliver reward
  - Good reward signal might be rare in the environment (plateau problem); subgoal decomposition might not lead to convergence to desired outcome
- **Do we know all the properties of the environment? Do we know our own preferences?**

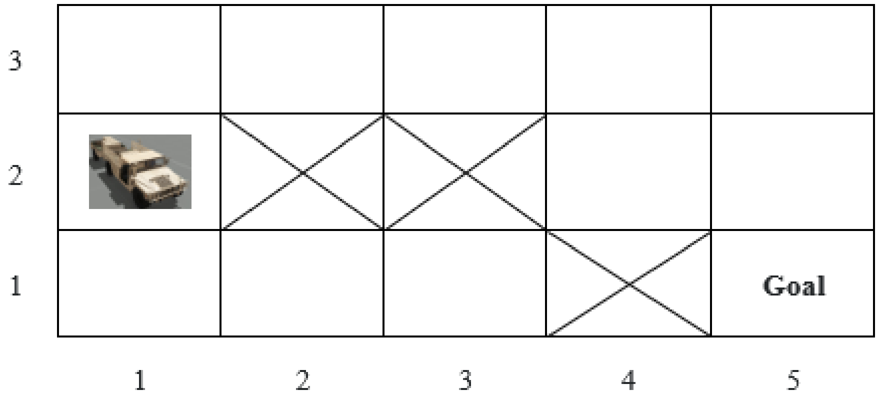
Exercise on Rule + Reward Design for ASV

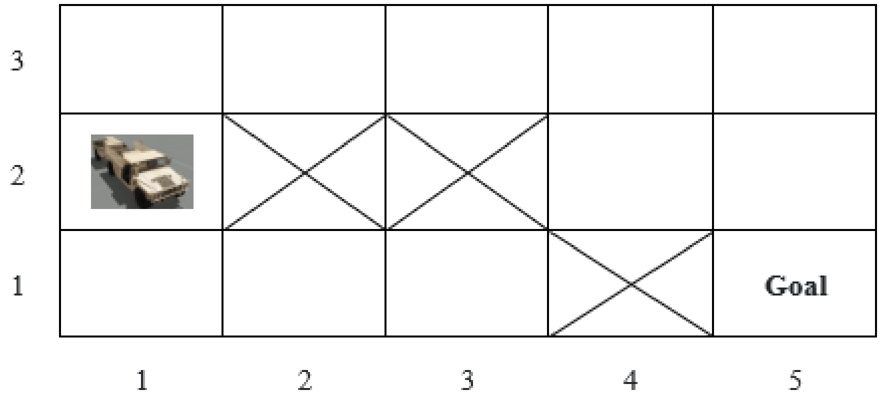
> Your task is to find the optimal path to reach the goal and specify a decision rule / reward architecture that allows the ASV to optimally reach the goal



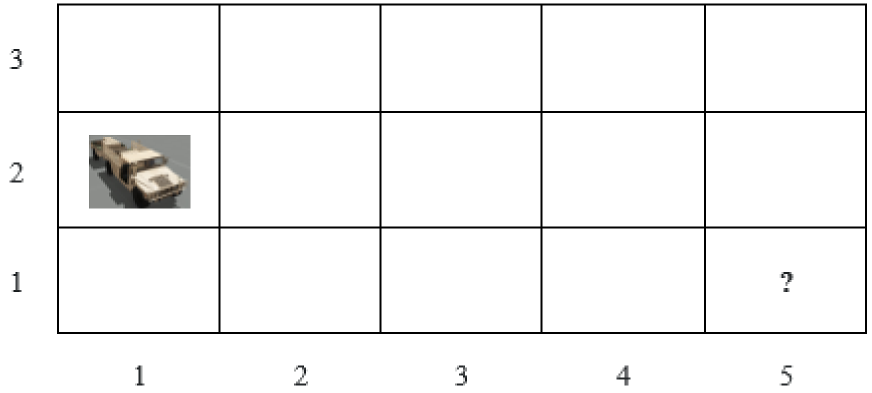
> ASV can move into **four directions**, right (→), left (←), up (↑), down (↓); one move at each time step
> ASV can only sense its current state and **cannot see into an adjacent state**, unless it enters that state
> Designer has perfect information about the world, e.g., she can see into all of the 15 states of the world However, **her knowledge of the world is restricted to what is explicitly specified**

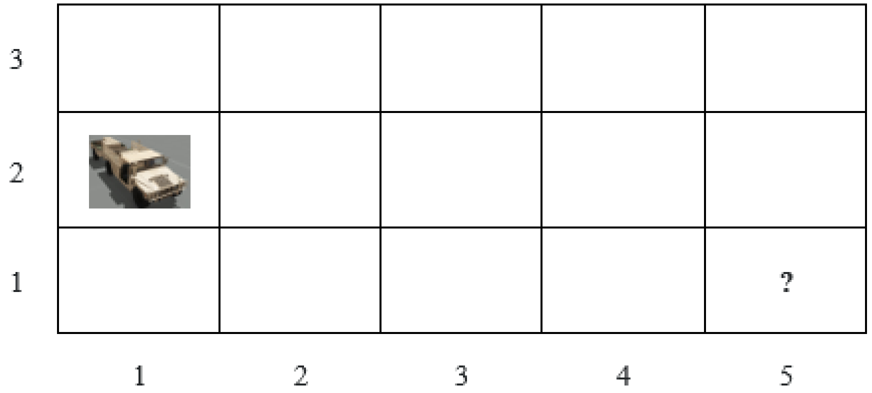
Pathfinding with Obstacles



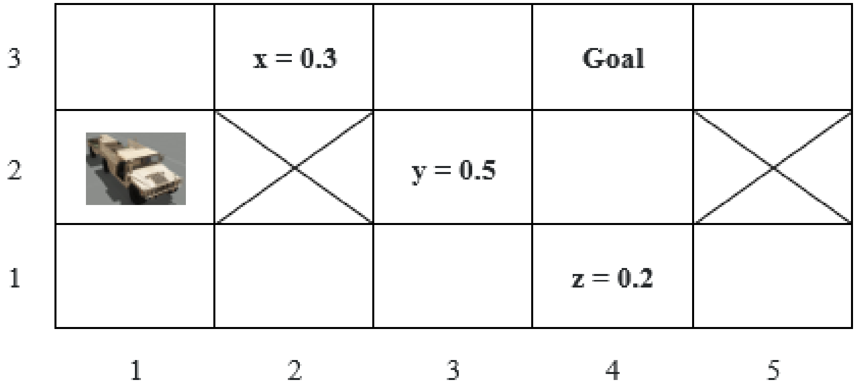


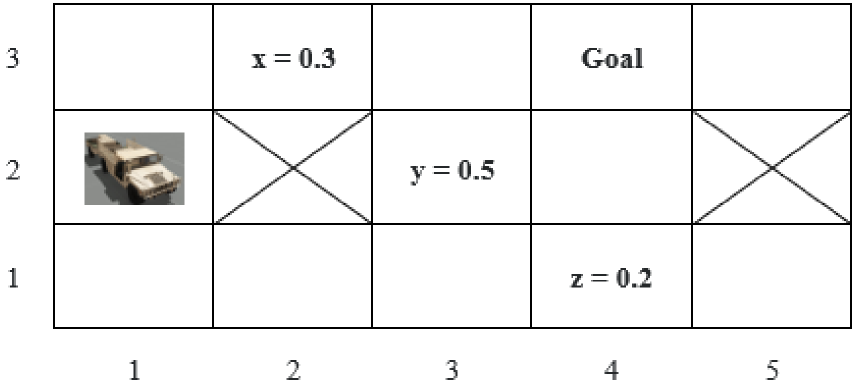
Unknown Goal



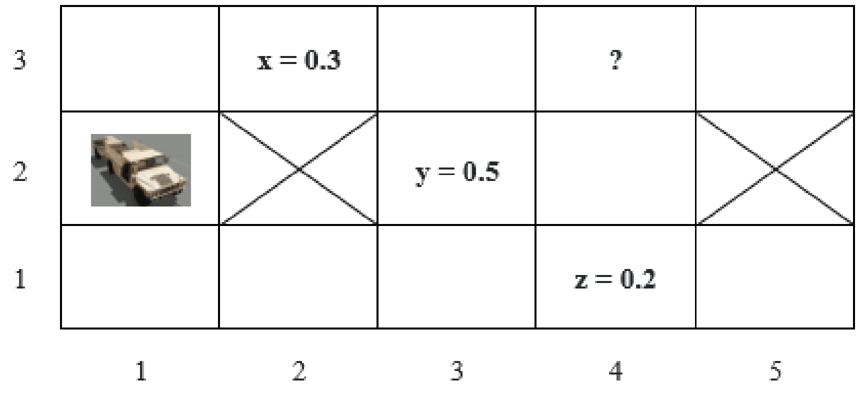


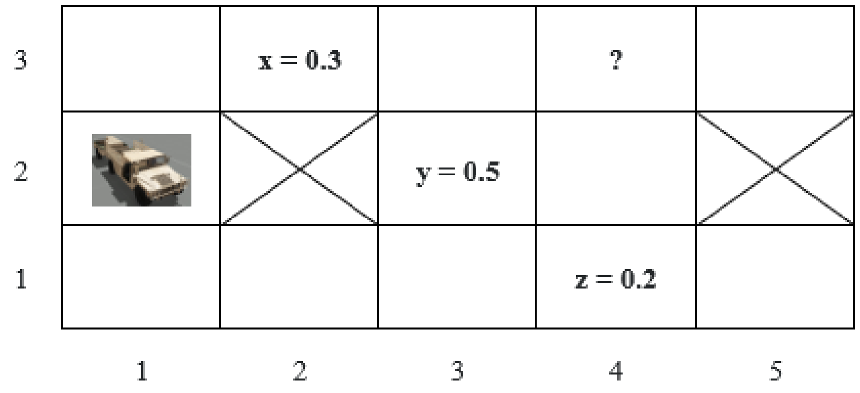
Known Probabilistic Fire and Known Goal



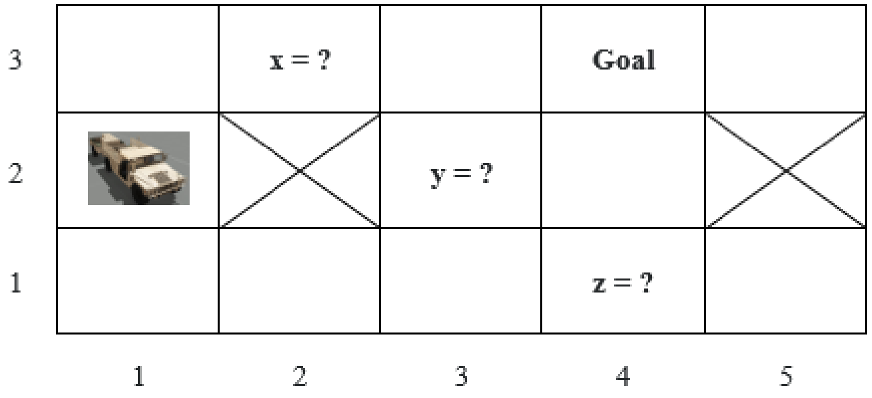


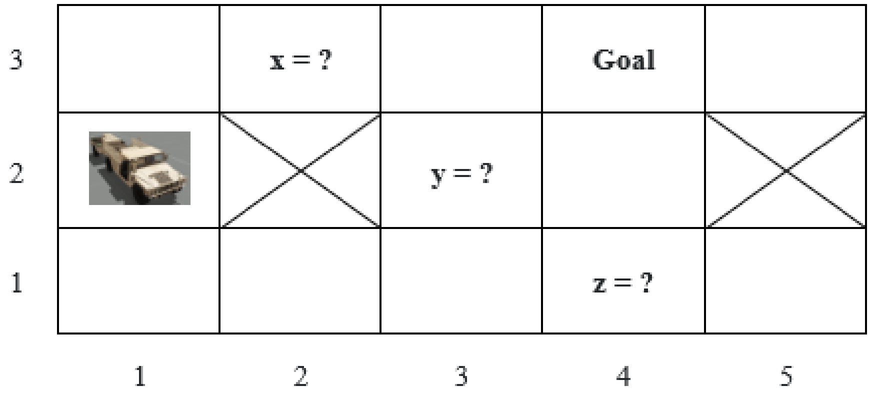
Known Probabilistic Fire and Unknown Goal



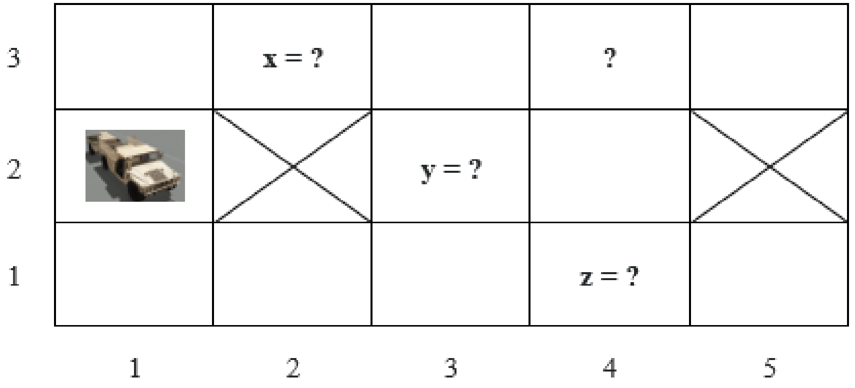


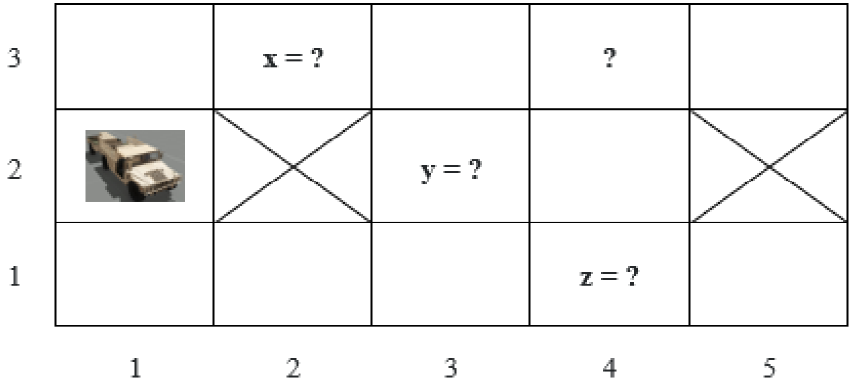
Unknown Probabilistic Fire and Known Goal





Unknown Probabilistic Fire and Unknown Goal





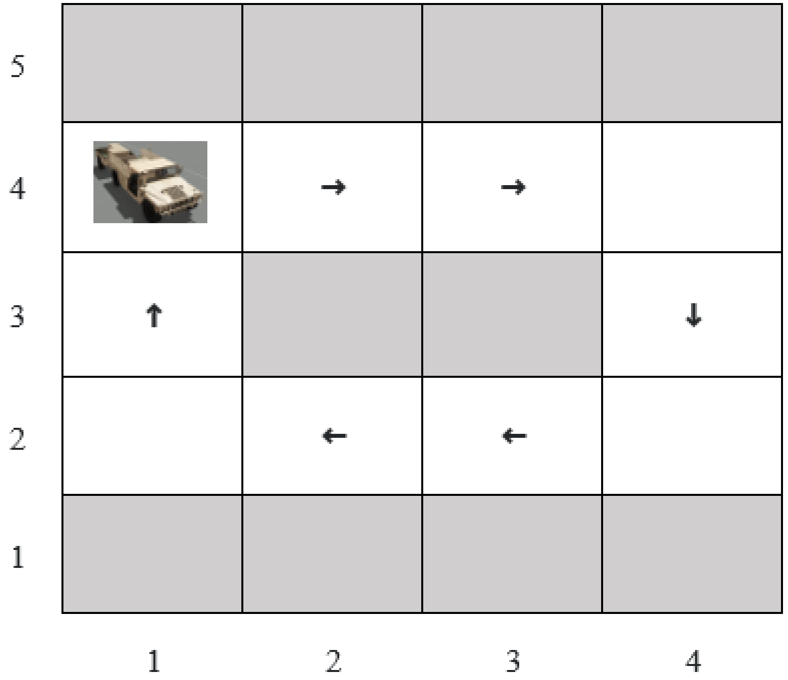
Two Failure Modes of Reward Design in RL

- **Reward gaming** (Clark and Amodei. 2016. Faulty Reward Functions in the Wild)
  - *How can we build agents that do not try to introduce or exploit errors in the reward function in order to get more reward?*

- **Negative side effects** (Amodei et al., 2016. Concrete Problems in AI Safety)
  - *How can we get an agent to avoid poor behavior if the reward function does not capture all the elements of the test environment?*

Reward Gaming

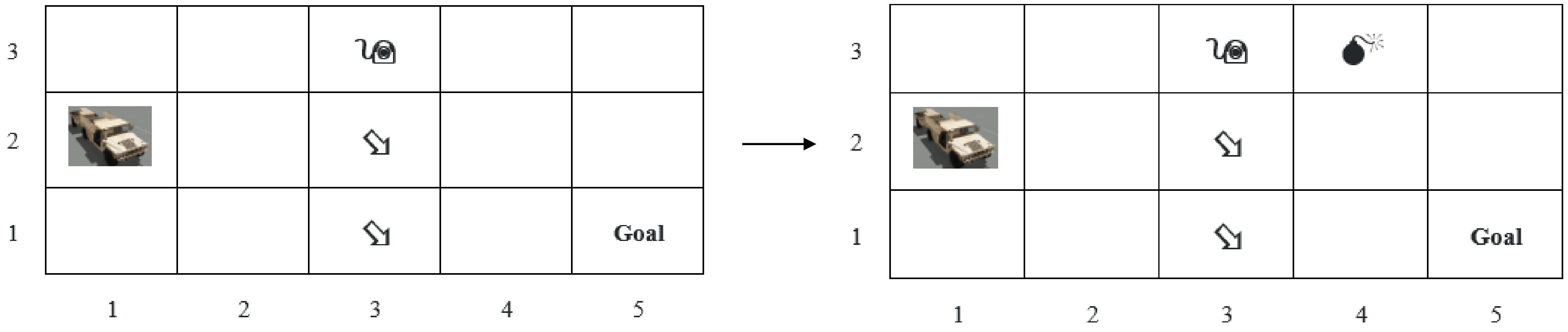
- Agent exploits an unintended loophole in the reward specification, to get more reward than deserved



- Desired outcome: clockwise completion of race
- Arrows are checkpoints associated with a reward of 3

Negative Side Effects

> Reward function does not fully capture all the properties of the test environment



Training environment *Test environment*

> Desired outcome: reach goal state
> (spotted by enemy) = -1, (bad terrain) = -3, (land mine) = - 100, **Goal** = 10

Inverse Reward Design

> Hadfield-Menell et al. 2017. Inverse Reward Design
>
> > *We leverage a key insight: that the designed reward function should merely be about the intended reward, rather than the definition; and should be interpreted in the context in which it was designed*