RECAP
Current status

- Autonomous agents have three main parts:
  - representing and reasoning with actions and environment assuming simple goals and plans;
  - representing and understanding complex directives/goals;
  - synthesizing and/or verifying complex plan of actions given in some action execution language.
- We are in the first part and discussed representing and reasoning about actions in the simple language $\mathcal{A}$.
- Next Step: Extend $\mathcal{A}$ to allow expression of relationship between objects in an environment.
REPRESENTING THE ENVIRONMENT
Motivation

- What if fluents in the environment are related?
- Do we need to express it explicitly?
- Example 1: dead iff ¬ alive
  - Approach 1:
    * In our observations, whenever we say something about alive we must say the opposite about dead.
    * In addition we need to add:
      * shoot causes dead if loaded.
  - Approach 2: We just add ‘dead iff ¬ alive’ to the agents knowledge and let the agent think through it.
    i.e., need to extend \( \mathcal{A} \) (both syntax and semantics).
The language $\mathcal{AR}$ (Kartha and Lifschitz)

- Syntax: In addition to $\mathcal{A}$ we have classical domain constraints of the form:
  
  - $\text{always } \phi$

  where $\phi$ is a propositional formula.

- Illustration:
  
  - $\text{load causes loaded}$
  - $\text{shoot causes } \neg \text{alive if loaded}$
  - $\text{initially } \neg \text{loaded}$
  - $\text{initially alive}$
  - $\text{always dead } \Leftrightarrow \neg \text{alive}$

  We would like to conclude $\text{dead after load, shoot}$

- Semantics: Need to define the initial state and the transition function.
  We may have non-deterministic transition.
Semantics of $\mathcal{AR}$

- Valid state: a state that satisfies all the classical constraints.
- Only valid states are candidate for initial states.
- Transition function $\Phi$
  
  - With constraints transition could be non-deterministic.
    
    * $\textit{toss causes tossed}$
    * $\textit{always tossed} \iff \textit{heads} \oplus \textit{tails}$
    * If you do the action $\textit{toss}$ in the state $\emptyset$ then intuitively the resulting states may be $\{\textit{tossed, heads}\}$ or $\{\textit{tossed, tails}\}$.
    * Note: $\{\textit{tossed}\}$ is not a valid state. Transitions are from one valid state to one among a set of valid states.
  
  - How to define the transition:
    
    * Let $\sigma' \in \Phi(a, \sigma)$. Then $\sigma'$ must satisfy the following.
      
      - $\sigma'$ must be a valid state. (Note: $\sigma \cup E^+(a, \sigma) \setminus E^-(a, \sigma)$ may not be valid state.)
      - $E^+(a, \sigma) \subseteq \sigma'$ must be true.
Representing the environment

\[ E^-(a, \sigma) \cap \sigma' = \emptyset \text{ must be true.} \]
\[ \sigma' \text{ must be as close to } \sigma \text{ as possible.} \]

- Closeness: Which one between \( \sigma_1 \) and \( \sigma_2 \) is closer to \( \sigma \).
  \[ \sigma_1 \leq_{\sigma} \sigma_2 \text{ iff } ((\sigma_1 \setminus \sigma) \cup (\sigma \setminus \sigma_1)) \subseteq ((\sigma_2 \setminus \sigma) \cup (\sigma \setminus \sigma_2)) \]
- \( \Phi(a, \sigma) = \{\sigma' : E^+(a, \sigma) \subseteq \sigma', E^-(a, \sigma) \cap \sigma' = \emptyset, \sigma' \text{ is valid and there does not exist a valid } \sigma'' \text{, which satisfies } \sigma'' <_{\sigma} \sigma' \text{ and } E^+(a, \sigma) \subseteq \sigma'', E^-(a, \sigma) \cap \sigma'' = \emptyset \} \)
- Consider the action shoot and the state \( \sigma = \{\text{alive, loaded}\} \)
  \[ E^+(\text{shoot}, \sigma) = \emptyset; E^-(\text{shoot}, \sigma) = \{\text{alive}\} \]
  \[ (\sigma \cup E^+) \setminus E^- = \{\text{loaded}\}. \{\text{loaded}\} \text{ is not a valid state.} \]
  \[ \text{Let } \sigma' = \{\text{dead, loaded}\}. \sigma' \text{ is a valid state.} \]
  \[ E^+ \subseteq \sigma' \text{ and } E^- \cap \sigma' = \emptyset \]
  \[ \Phi(\text{shoot}, \sigma) = \{\sigma'\} \]