* **Agent -** Anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
  + **Rational agent**: One that does the “right thing”
* **Environment** - Part of the universe whose state we care about when designing agent. Affects what agent perceives and, is affected by agent’s actions
* **Percept -** Content an agent’s sensors are perceiving
* **Percept Sequence** - Complete history of everything the agent has ever perceived

An agent’s choice of action at any given instant can depend on its built-in knowledge and on the entire percept sequence observed to date, **but NOT on anything it hasn’t yet perceived**

* **Agent function -** Maps any given percept sequence to an action (abstract mathematical description)
  + **External characterization**: constructing a table that maps out what actions an agent takes given all possible percept sequences
  + **Internal characterization**: Agent function is implemented by an **agent program** (concrete implementation)
* **Consequentialism-** Evaluating an agent’s behavior by its consequences
  + **Performance measure:** Notion of desirability that evaluates any given sequence of environment states
    - Some agents have an explicit representation of the performance measure
    - Other designs the measure is implicit - **the agent may do the right thing but it does not know why**

[Generally], It’s better to design performance measures according to what one actually wants to be achieved in the environment, rather than according to how one thinks the agent should behave

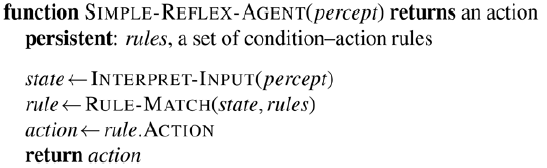
* **Rationality depends on:**
  + Performance measure that defines the criteria of success
  + Agent’s prior knowledge of the environment
  + Actions that the agent can perform
  + Agent’s percept sequence **to date**
* **Rational agent** - For each possible **percept sequence**, a rational agent should select an action that’s expected to maximize its **performance measure** given the evidence provided by the **percept sequence** and whatever built-in knowledge the agent has
  + **Rational is not omniscient** - Omniscient agents know the actual outcomes of their actions and can act accordingly (impossible in reality)
    - Rationality maximizes *expected* performance while perfection maximizes *actual*performance
  + **Rational agents gather information** - Actions that modify future percepts
    - Important component of rationality
    - Often gathered through exploration
  + **Rational agents learn -** Agent learns as much as possible from percepts
    - Agent’s prior knowledge of the environment can be modified
  + **Rational agents should be autonomous** - SHould learn what it can to compensate for partial or incorrect prior knowledge
    - If agent relies on prior knowledge from designer rather than its own percepts/learning it lacks autonomy
  + After enough experience in environment, the behavior of a rational agent can become independent of its prior knowledge
* **Task environment** - problems to which rational agents are the solutions
  + In designing an agent, the first step is ALWAYS to specify the task environment as fully as possible.
    - **P**erformance measure
    - **E**nvironment
    - **A**ctuators (of agent)
    - **S**ensors (of agent)
* **Observability (of Environments)**
  + If agent’s sensors give it access to the complete state of environment at each point in time, the task environment is **fully observable**
    - If sensors detect all aspects **relevant** to the choice of action, the task environment is *effectively* fully observable
    - Agent doesn’t need any internal state to keep track of the world
  + If the agent has no sensors at all, the environment is **unobservable**
* **Multiagency**
  + There’s a difference between entities that *might*be agents vs. entities that **must** be agents
  + Does Agent A have to treat B as an object, or agent?
    - The key distinction is whether B’s behavior is maximizing a performance measure whose value depends on Agent A’s behavior
  + **Competitive Multiagent Environment**
    - If Agent B is trying to maximize its performance measure which minimizes A’s performance measure
    - E.g. Chess
  + **Cooperative Multiagent Environment**
    - E.g. Taxi driving
      * Partially cooperative because avoiding collisions maximizes performance measure of all agents
      * Partially competitive because only one car can occupy parking space
  + Communication emerges as a rational behavior in multiagent environments
  + In some competitive environments randomized behavior is rational because it avoids predictability
* **Deterministic vs. Non-Deterministic**
  + If the next state of the environment is determined by the current state and action executed by the agent it is **deterministic**
  + If the environment is only partially observable it could *appear* to be **nondeterministic**
  + A model is **stochastic** if it explicitly deals with explicitly defined probabilities and **nondeterministic** if possibilities are listed without being quantified
* **Episodic vs. Sequential**
  + **Episodic:** agent’s experience is divided into atomic episodes
    - In each episode agent receives a percept and performs a single action
    - The next episode does NOTdepend on the actions taken in previous episodes.
    - Classification tasks are typically episodic
  + **Sequential:** Current decision could affect all future decisions
    - Short-term actions have long-term consequences
    - More complex because agent may have to think ahead
* **Static vs. Dynamic**
  + **Dynamic:** If the environment can change while agent is deliberating
    - Continuously asking agent what it wants to do
    - If agent hasn’t decided what to do, that counts as deciding to do nothing
    - If environment doesn’t change with passage of time but agent’s performance does, the environment is **semi-dynamic**
  + **Static:** Environment doesn’t change
    - Agent doesn’t need to keep observing world during deliberation
    - Agent doesn’t need to worry about time
* **Discrete** **vs. Continuous -** applies to the state of the environment, the way time is handled, and to the percepts and actions of the agent
  + **Discrete:** Finite number of distinct states
  + **Continuous:** Infinite range of values
* **Known vs. Unknown** - refers not to the environment but to the agent’s (or designer’s) state of knowledge about the laws of the environment
  + **Known environment:** the outcomes (or outcome probabilities for nondeterministic environments) for all actions are given
  + It’s possible for a *known* environment to be *partially* observable
    - E.g. Solitaire. Know rules but can’t see cards
  + Unknown environments can be fully observable
  + Performance measure itself can be unknown!
    - Maybe designer isn’t sure what it is
    - Performance measures can depend on the end user’s performances
      * In this case, the agent has to learn the user’s preferences (multiagent environment)

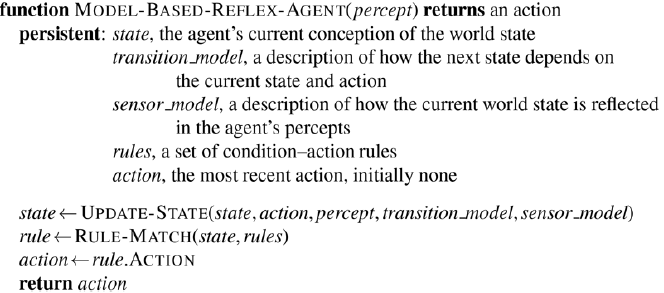
The hardest case is partially observable, multiagent, nondeterministic, sequential, dynamic, continuous and unknown*.*

* **Agent Architecture** - agent = architecture + program
  + Agent’s architecture must afford the operations the program contains
  + Architecture makes the percepts from sensors available to the program, runs the program, and feeds program’s action choices to the actuators
  + Difference between agent’s program (takes current percept as input) and agent function (which may depend on the entire percept history)
    - If agent’s actions need to depend on the entire percept sequence, agent has to remember the percepts
* **Table-Driven Agent -** Agent program that keeps track of the percept sequence and uses it to index into a lookup table of actions
  + **Why is this bad idea?** 
    - Let P = set of possible percepts and T be the lifetime of the agent (total number of percepts it’ll receive)
    - entries in the lookup
    - No physical agent in this universe have space to store the table

The key challenge for AI is to find out how to write programs that, to the extent possible, produce rational behavior from a smallish program rather than from a vast table.

* **Simple reflex agent** - select actions on the basis of the current percept, ignoring the rest of the percept history
  + Simple reflexive behavior is typically formed by **condition-action rules** or **if-then rules**.
  + Will only work if the correct decision can be made on the basis of JUST the current percept (i.e. the environment is fully observable)
  + Infinite loops are often unavoidable for simple reflex agents operating in partially observable environments
    - Escape is possible if the agent can randomize actions
  + Randomized simple reflex agent might outperform a deterministic simple reflex agent
  + In single-agent environments randomization is usually not rational



* **Model-based reflex agent -** Agent maintains *internal state* that depends on the percept history and reflects at least some of the unobserved aspects of the current state
  + Agent keeps track of the part of the world it can’t see right now
  + Most effective way to handle partial observability
  + To update internal state as time goes by need two kinds of knowledge
    - **Transition model:** knowledge about how the world works
      1. Effects of the agent’s actions
      2. How the world evolves independently of the agent
    - **Sensor model**: knowledge about how the state of the world is reflected in the agent’s percepts
      1. E.g. When car in front brakes, one or more illuminated red regions appear in the front camera image
  + **Model-based agent:** With **transition** and **sensor** models, agent can keep track of the world, limited only be agent’s sensors
  + Seldom possible for agent to determine current state of partially observable environment exactly
    - The agent usually just has a ‘best guess’
* **Goal-based agent:** Agent also needs goal information that describes situations that are desirable
  + Knowing current state of environment isn’t always enough to decide what to do
  + Fundamentally different from condition-action rules
    - Goal-based decision making involves consideration of the *future*
    - In reflex agent designs this information isn’t explicitly represented
      * Built-in rules map directly from percepts to actions
      * “The agent doesn’t know WHY it’s doing what it’s doing.”
  + More flexible because the knowledge that supports its decisions is represented EXPLICITLY and can be MODIFIED
* **Utility-based agent:** Goals only provide binary distinction between “happy” and “unhappy states”
  + What if some actions are ‘better’ than others?
  + More general performance measure should allow a comparison of different world states according to “how happy” they would make the agent
  + **Utility function**: Agent’s internalization of the performance measure
    - Performance measure assigns ‘scores’ to any given sequence of environment states to distinguish between more/less desirable
  + If internal utility function and external performance measure are in agreement, an agent that chooses to maximize its utility will be rational according to the external performance measure
    - Not the only way to be rational
  + There are cases where goals are inadequate but utility-based agents can still make rational decisions
    - Conflicting goals, only some of which can be achieved
      * Utility function specifies appropriate tradeoff
      * E.g. taxi driver’s goals of speed vs. safety
    - Several goals that the agent can aim for, none of which can be achieved with certainty
      * Utility provides a way in which the likelihood for success can be weighed against importance of goals
  + A **rational utility-based agent** chooses the action that maximizes **expected utility** of the action outcomes
    - Agent expects to derive, on average, given the probabilities and utilities of each outcome.
  + Any rational agent must behave as if it possesses a utility function whose expected value it tries to maximize
  + Agent that possesses an EXPLICIT utility function can make rational decisions with a general-purpose algorithm that doesn’t depend on the specific utility function being maximized
  + Utility-based agent programs handle the uncertainty inherent in nondeterministic or partially observable environments
  + Utility based agent must model and keep track of
    - It’s environment
    - Tasks that have involved a great deal of perception, representation, reasoning and learning
* **Learning Agents -** Any type (model, goal, utility-based, etc.) can be built as a learning agent or not.’
  + Learning allows the agent to operate in initially unknown environments and to become more competent than its initial knowledge alone
  + Four conceptual components
    - **Learning element:** responsible for making improvements
      * Design depends on design of performance element
      * “What kind of performance element will my agent use to do this once it has learned how?”
      * Learning mechanisms can be constructed to improve every part of the agent
      * Can make changes to any of the ‘knowledge’ components
      * Simplest case involves learning directly from percept sequence
      * Observation of pairs of successive states of the environment can allow the agent to learn “what did my actions do” and “how does the world evolve’ in response to actions
    - **Performance element:** responsible for selecting external actions
      * What we previously considered to be the entire agent (takes in percepts and decides on actions)
      * **Performance standard** distinguishes part of the incoming percept as a reward/penalty that provides feedback on the quality of the agent’s behavior
    - **Critic:** Provides feedback to learning element on how agent is doing based on *fixed performance standard* and determines how performance element should be modified to do better in the future
      * Necessary because percepts themselves provide no indication toward success
      * Important that the performance standard is *fixed*
      * Think of it as being outside the agent altogether because the agent MUST NOT modify performance standard to fit its own behavior
    - **Problem generator:** responsible for suggesting exploratory actions
      * Performance element ALONE will always only do what actions are best (given current knowledge)
      * If the agent explores suboptimal actions in the short-run, it might discover better long-term solution
      * Might identify certain parts of the model that are in need of improvement
  + Improving model components of a **model-based agent** so that they conform better with reality is almost always a good idea REGARDLESS of **performance standard**
* **Axes of Component Representations** 
  + **Expressiveness:** a more expressive representation can capture everything a less expressive representation can and more
    - **Atomic:** Each state of the world is indivisible, no internal structure
    - **Factored:** Splits up each state into a fixed set of **variables/attributes** each of which can have a **value**
      * 2 different atomic states have nothing in common, they are just different black boxes, but two different factored states can share variables/attributes.
    - **Structured:** Objects and their various relationships can be described explicitly
  + **Mapping of Concepts:** Mapping concepts to locations in physical memory (computer or brain)
    - **Localist:** 1-to-1 mapping between concepts and memory locations
      * Mapping from concept to memory location is arbitrary
    - **Distributed**: representation of a concept is spread over many memory locations, and each memory location s employed as part of the representation of multiple different concepts
      * More robust against noise and information loss