

PEAS: Performance measure, Environment, Actuators, Sensors (write more)

Different types of Agents

- Simple reflex agent: These agents select actions on the basis of the current percept, ignoring the rest of the percept history. These agents work only when the environment is fully observable.
- Model-based reflex agent: The knowledge about “how the world works” — whether implemented in simple Boolean circuits or in complete scientific theories — is called a model of the world. This agent uses a state which is a function of current state, action, percept and rules. It uses this state to decide the next action.
- Goal-based agent: It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.
- Utility-based agent: It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.
- General Learning Agent: A learning agent can be divided into four conceptual components
 - **learning element**, which is responsible for making improvements
 - **performance element**, which is responsible for selecting external actions. The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions.
 - The learning element uses feedback from the **critic** on how the agent is doing and determines how the performance element should be modified to do better in the future.
 - **Problem generator**, It is responsible for suggesting actions that will lead to new and informative experiences.

Ways to represent states and transition between them

- **Atomic representation**: a state (such as B or C) is a black box with no internal structure.
- **Factored representation**: a state consists of a vector of attribute values; values can be Boolean, realvalued, or one of a fixed set of symbols.
- **Structured representation**: a state includes objects, each of which may have attributes of its own as well as relationships to other objects.

Modelling paradigms for Intelligent Agents:

- Logic-based models: propositional logic, first-order logic; Applications: theorem proving, verification, reasoning. Think in terms of logical formulas and inference rules
- Variable-based models: CSPs, Bayesian networks; Applications: scheduling, tracking, medical diagnosis, etc. Think in terms of variables and factors
- State-based models: search problems, MDPs, games; Applications: route finding, game playing, etc. Think in terms of states, actions, and costs

Propositional Logic

- Well formed formula (wff): An atomic proposition is a wff
- Logical inference is used to create new sentences that logically follow from a given set of propositional logic sentences (Knowledge Base, KB).
- sound - inference rule does not create any contradictions
- complete - inference rule is able to produce every expression that logically follows from (is entailed by) the KB.

Interpretation for propositional logic is a mapping assigning a truth value to each of the simple sentences of the language.

- Satisfiability: We say that an interpretation i satisfies a sentence $\phi \iff$ it is true under that interpretation
- Validity: A sentence is valid \iff it is satisfied by every interpretation.
- Unsatisfiability: \iff it is not satisfied by any interpretation

We would like to know that given some sentences, whether other sentences are or are not logical conclusions. This relative property is known as logical entailment. Formally, a set of sentences Δ logically entails a sentence ϕ (written $\Delta \models \phi$) if and only if every interpretation that satisfies Δ also satisfies ϕ . For example, the set of sentences $\{p, p \vee q\}$ logically entails the sentence (q) .

CNF Form

- Eliminate \iff , replacing $\alpha \iff \beta$ with $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$.
- Eliminate \rightarrow , replacing $\alpha \rightarrow \beta$ with $\neg\alpha \vee \beta$
- Requires \neg to appear only in literals, so we “move \neg inwards” i.e. distribute it in a bracket.
- Distribute \vee over \wedge , so final looks like $(\alpha \vee \beta) \wedge (\alpha \vee \gamma) \iff (\alpha \vee (\beta \wedge \gamma))$

To show that $KB \models \alpha$, we show that $(KB \wedge \neg\alpha)$ is unsatisfiable

- $CNFSentence \rightarrow Clause_1 \wedge \dots \wedge Clause_n$
- $Clause \rightarrow Literal - 1 \wedge \dots \wedge Literal_m$
- $Literal \rightarrow Symbol | \neg Symbol$
- $Symbol \rightarrow \$P | Q | R | \dots \$$
- $DefiniteClauseForm \rightarrow (Symbol_1 \wedge \dots \wedge Symbol_l) \rightarrow Symbol$
- $GoalClauseForm \rightarrow (Symbol_1 \wedge \dots \wedge Symbol_l) \rightarrow False$
- $HornClauseForm \rightarrow DefiniteClauseForm | GoalClauseForm$

Forwards chaining: Recursively adding new sentences to the KB that can be inferred from the current KB till we reach the goal or no more sentences can be inferred.

Backwards chaining: Recursively adding new sentences to the KB which are required to prove the goal.

First Order Logic

- **Sentence** \rightarrow AtomicSentence | ComplexSentence; **AtomicSentence** \rightarrow Predicate | Predicate(Term, ...) | Term = Term
- **ComplexSentence** \rightarrow Sentence | \neg Sentence | Sentence \wedge Sentence | Sentence \vee Sentence | Sentence \rightarrow Sentence | Sentence \iff Sentence | Quantifier Variable, ... Sentence
- **Term** \rightarrow Constant | Variable | Function(Term, ...); **Constant** \rightarrow A | X_1 | John | ...; **Variable** \rightarrow x | y | z | ...
- **Predicate** \rightarrow True | False | After | Loves | ...; **Function** \rightarrow Mother | LeftLeg | ...; **Quantifier** $\rightarrow \forall$ | \exists

Interpretation - assigning symbols to objects / predicates / functions. A **model** in first-order logic consists of a set of objects and an interpretation that maps constant symbols to objects, function symbols to functions on those objects, and predicate symbols to relations.

Most of the time, use \rightarrow with \forall instead of \wedge and use \wedge with \exists instead of \rightarrow . Quantifiers of same type commute but of different type don't commute.

Inference

- Propositionalization : Eliminate quantifiers. In case of Universal Instantiation, replace variables with ground terms (no variables)
 - **Universal Instantiation**: $\forall v \alpha / (SUBST(\{v/g\}, \alpha))$ (v is variable and g is ground term). Can be applied many times.
 - **Existential Instantiation**: $\exists v \alpha / (SUBST(\{v/c\}, \alpha))$ (v is variable and c is constant). Called Skolemization, g is Skolem constant. Can only be applied once

Convert to CNF, Do Unification, Forward Chaining, Backward Chaining to get answer.

Knowledge Representation

- Knowledge comprises objects, events, time and beliefs. Needs general & flexible representation.
- **Reification**: Turning a predicate into an object. eg Basketball(b) is a predicate, which can be represented as a category $b \in \text{Basketball}$. or member of (b , Basketball)
- Categories can have objects as members, sub-categories (Children's Basketball) or super-categories (Ball). Members can have properties. Categories themselves can have properties
- **Partition** - Exhaustive decomposition using disjoint sub-categories
- **Things** - Objects that can be divided into distinct objects. eg. car. Count Nouns
- **Stuff** - Objects that can't be divided into distinct objects. eg. water. Mass Nouns
- Objects of Event Calculus - Events / actions, Fluents(aspects of the world that can change), Time points / intervals (to refer 'when' events happen)
- **Semantic Networks** - Representation of individual objects, categories & relations among objects
- **Truth Maintenance Systems** - With exceptions, the KB needs to be updated when new facts come to light and when new exceptions are discovered