Artificial Intelligence - Logical Agents

Knowledge Base & its agents

- Knowledge base (KB) is collection of sentences
- Each sentence is expressed in knowledge representation lang. & represents some assertion of the world.
- Sentences added & queried using TELL & ASK resp.
- Inferencing is the task of deriving sentences from old.
- Initial KB is called background knowledge.
- 2 Functions
  + MAKE-PERCEPT-SEQUENCE: Takes a percept & time and returns a sentence asserting that agent perceived the percept at the given time
  + MAKE-ACTION-QUEUE: Takes time as I/P & returns a sentence that asks what action should be performed at that time.

Logic

- Syntax of lang. tells whether a sentence is well-formed.
- The semantics of lang. defines the truth of each sentence wrt to each possible world.
- Logical Entailment - One sentence follows from another.
  + If α entails β, $\alpha \models \beta$.
  + If α entails β, then in every model where α is true, β is also true.
- If an inference algorithm i can derive α in KB, then $KB \vdash_i \alpha$., α is derived from KB by i,
- An inference algo that derives only entailed sentences is called sound / truth-preserving.
- An inference algo is complete if it can derived any sentence that is entailed.
- If KB is true in real world, then any sentence α derived by KB by a sound inference algo is also true in the real world.
- Grounding - rel^n, if any, bet^w logical reasoning process & real env. in which agent exists.

- Beliefs are not direct representation of a single percept, but a general rule - derived, perhaps, from perceptual experience but not identical to a stmnt of that experience.

Thanks.

- General rules likes this are produced by a sentence constructed process called as learning.

Propositional logic (PL)

- Consists of Atomic sentences with a single propositional symbol. - the symbol can be true/false.
- Complex sentences are constructed from simpler sentences using logical connectives which are
  - ¬ (not):
  - ∧ (and): A sent whose main connective is ∧ is called conjuction & parts as conjucts
  - ∨ (or): Sentence using is called disjuctions & parts as disjuncts
  - ⇒ (implies): Sentence is called implication, Part bfr ⇒ is called premise / antecedent & after as conclusion / consequent.
  - ⇔ (iff) : Sentence is biconditional.
- Semantics defines the rules for determining the truth of a sentence w.r.t. to particular model.
- In PL, model simply fixes the truth value - true/false for every PL symbol.
- Semantics for PL must specify how to compute truth value of any sentence, given a model.
- Logical equivalence: 2 sentences are logical equivalent if they are true for same set of models. written as $\alpha \Leftrightarrow \beta$ or $\alpha \equiv \beta$. Here $\alpha \models \beta$ & $\beta \models \alpha$.
- Valid sentence: A sentence is valid if it is true for all models: Also called tautologies
- Satisfiability: A sentence is satisfiable if it is true for some model.

Reasoning patterns in Propositional logic.

- Inference: Deriving chains of conclusions that lead to desired goals.
- Inference Rules: Patterns of Inference.
- Modus Ponens $\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$. Means whenever any sentence of forms $\alpha \Rightarrow \beta$ & α are given, β can be inferred
- And-Elimination: $\frac{\alpha \wedge \beta}{\alpha}$. From conjuctions, conjucts can be inferred.
- Proof: A sequence of appln of inference rules.
- Monotonicity: Set of entailed sentences can only be increased as info is added to KB.
  + Means the rules of inference can be applied whenever suitable premises are found in the KB; conclusion of rule must follow regardless of what else in KB.
- Resolution: Yields a complete inference algo when coupled with any complete search algo.
  + Can be used to either confirm / refute a sentence (process called refutation completeness) but not to enumerate a sentence.
  + Applies to only disjunction of literals.
  + Leads to complete inference procedure for all of propositional logic. (Every sentence of propositional logic is logically equivalent to conjuction of disjunction of literals).

* Search / inference algo are complete if they are able to ~~reach~~ find any reachable goal / ~~an~~ verify a sentence.

- Horn clause: A disjunction of literals of which at the most one literal is +ve
- Definite clause: only and only one ~~clau~~ literal in the clause is +ve.
  + Every Horn clause can be written as an implication whose premises is a conjuction

of +ve literals & whose conclusion is a +ve literal.
Eg. H.C. $(\neg L_{1,1} \vee \neg Breeze \vee B_{1,1})$ can be written as

$(L_{1,1} \wedge Breeze) \Rightarrow B_{1,1}$

+ A +ve literal is called head & -ve literals form body
+ A definite clause with no -ve literals simply asserts a given proposition - sometimes called as fact.
+ Inference done using Forward / Backward chaining algo.

An algo is
+ sound: every inference is essentially an application of some inference rule
+ complete: every entailed atomic sentence can be derived.
+ At fixed point: No new inferences are possible.

+ Forward chaining is data-driven while Backward chaining is goal driven.

Circuit Based Agents.
- A reflex agent with a state.
- Percepts are sequential circuits.
-
- X models in PL: Sets of truth values for propositional symbols

- AShish R. Gavande

Thanks.