# Knowledge-Based Agent

- an agent that
  - can form representations of the world
  - uses a process of inference to derive new representations of the world
  - uses the new representations to deduce what to do

# Knowledge-Based Agents

- KB approach
  - aims to implement an agent that knows about its world and reasons about its possible courses of action
  - agent accepts new tasks in the form of explicit goals
    - can achieve competence by learning and updating knowledge

# Knowledge-Based Agent

- central component - its knowledge base
  - representations of facts about the world
  - individual representation = sentence
  - sentence expressed in some Representation Language
- use TELL & ASK (relating to KB)
  - add new sentences
  - query what is known
- answer to ask should follow from what has been told
  - KB does not make up things as it goes

# Knowledge-Based Agent

- maintains KB
  - may initially contain background knowledge
- when Agent program is called
  - it tells KB what it perceives (not the same as user telling the KB something)
  - asks the KB what action to perform
- uses logical reasoning to prove which action is better to answer query
- agent should then perform chosen action

# Knowledge-Based Agent

- details of KRL and inference hidden in two functions
- Make-Percept-Sentence
  - takes percept and time
  - returns sentence - fact that the agent perceived the percept at a given time
- Make-Action-Query
  - takes time
  - returns sentence suitable for asking what action should be performed at that time

# Knowledge-Based Agent

- design given is similar to agent with internal state
- because of definitions of Tell and Ask, a KB Agent is not a program for calculating actions based on internal state variables
Knowledge-Based Agent

- can be described at three levels
  - knowledge (epistemological) level
  - logical level
  - implementation level

Knowledge-Based Agent

- Knowledge Level (Epistemological Level)
  - most abstract
  - what the agent knows
    - know that Eyre Square is two way system
  - if Tell and Ask work correctly, then can work at this level

- Logical Level
  - knowledge is encoded into sentences
    - Oneway(EyreSquare)

Knowledge-Based Agent

- Implementation Level
  - level that runs on agents architecture
  - physical representations of sentences at logical level
  - sentence represented by string in list of strings
    - Oneway(EyreSquare) represented by the string
    - or by complex set of pointers connecting machine addresses corresponding to individual symbols
  - choice of implementation influences performance of agent

Representation, Reasoning, Logic

- basic elements of agent design
- Knowledge Representation
  - object is to express knowledge in computer tractable form
  - Consider syntax and semantics
- syntax
  - possible configurations that constitute sentences
  - usually, how sentences are represented on printed page
    - with agent, representation is inside computer
    - can think of electron pattern in memory

Representation

- semantics
  - determine facts in the world to which sentence refers
  - each sentence makes a claim
  - when a particular configuration exists, the agent believes the corresponding sentence
- example
  - syntax says if x and y are numbers, then X >=Y is a sentence about numbers
  - semantics says that X>=Y is false if Y is bigger than X

Representation

- if syntax and semantics defined precisely
  - language is a logic (broad interpretation)
- from syntax & semantics
  - derive an inference mechanism for Agent using that language
- how?
  - semantics - determine fact to which sentence refers
  - distinguish between facts and their representation
    - facts - part of the world
    - representation - encoded to be stored in the agent
Representation

- all reasoning mechanisms work on representation, not actual facts
- reasoning is a process of constructing new physical configurations from old ones
- proper reasoning should ensure that new configurations represent facts that actually follow from the facts that the old configurations represent

Representation

- example: From the facts
  - solar system obeys laws of gravitation
  - current arrangement of sun, planets and other bodies
- it follows Pluto will eventually spin off into the void
- agent could end with representation
  - Pluto arrives in Galway
- consider Monty Python & Holy Grail

Representation

- Entailment must be preserved
  - generate sentences that are necessarily true, given that old sentences are true
  - $KB \models \alpha$
- inference procedure can do 1 of 2 things
  - given $KB$, can generate new sentences $\alpha$ that purport to be entailed by $KB$
  - given $KB$ & $\alpha$, can report if $\alpha$ is entailed by $KB$

Representation

- inference procedure $i$ can be described by the $\alpha$ it derives
  - Should be sound/truth preserving
- if $i$ can derive $\alpha$ from $KB$, then write
  - $KB \vdash i \alpha$
  - $\alpha$ is derived from $KB$ by $i$
  - record of operation of a sound $i$ called a proof

Representation, Reasoning, Logic

- $i$ is complete if
  - can find a proof for any entailed $\alpha$ (haystack)
- sound inference is desirable
  - key is to have inference steps respect semantics of $\alpha$ they operate upon
  - given a $KB$, inference steps should only derive new $\alpha$ that represent facts that follow from the facts represented by $KB$
- from semantics, can extract proof theory
  - specifies which reasoning steps are sound

Representation, Reasoning, Logic

- consider
  - $E = mc^2$
  - can show that a new sentence can be generated
  - $ET = mc^2T$
  - logical languages are like this
    - but they must deal with everything we might want to represent
    - not just numbers
Representation

• consider programming languages and natural languages

• PL
  – good for describing algorithms and data structures
    • use a 8x8 array to represent a chess board
    • World[2,2] ← Pawn – no problem
    • What about
      – “There is a pawn in some square”

Representation

• PL
  – completely describes state of computer and how it changes during execution
  – would like to support case where full information not available, only know some of the possibilities
  – PL not expressive enough

• NL
  – is expressive
  – meets communication rather than representation

Representation

– good way for someone to get to know something
– sharing of knowledge without explicit representation
  • Look!
  • Don’t want this to represent an exact piece of knowledge
– suffer from ambiguity (small cats and dogs)

• Good KRL
  – combine advantages of PL and NL
  – expressive and concise
  – unambiguous and context independent
  – effective, new inferences from sentences

Representation

• main ideas
  – how a precise formal language can represent knowledge
  – how mechanical procedures can act on expressions in that language to perform reasoning

  • the fundamental concepts remain the same no matter what language is used

Semantics

• in logic
  – meaning of $\alpha$ is what it states about the world
• how do we establish the meaning
  – the correspondence between sentence and fact
• usually up to the person who wrote the sentence
  – they provide an interpretation
    • a sentence does not mean something by itself
    • spies like us
• in practice, all KRL impose systematic relationship between $\alpha$ and fact

Semantics

• compositional
  – meaning of a sentence is function of the meaning of its parts
• once $\alpha$ is given interpretation by semantics
  – can be true or false
  – depends on interpretation and actual state of world

A sentence is true under a particular interpretation if the state of affairs it represents is the case
Inference

- any process by which conclusion is reached
- sound reasoning
  - logical inference or deduction
  - implements entailment between sentences
- necessarily true sentence
  - valid if and only if all possible interpretations in all possible worlds
  - “There is a stench in [1,1] or there is not a stench in [1,1]”

Inference

- “There is an open area in the square in front of me or there is a wall in the square in front of me”
  - only valid under the assumption that every square has either a wall or an open area in it
- satisfiable sentence
  - if and only if there is at least one possible world for which it is true
  - “there is a wumpus at [1,2]”
  - self-contradictory sentences are unsatisfiable

Inference in Computers

- computer suffers from 2 handicaps
  - does not necessarily know your interpretation for sentences
  - knows nothing about the world except what is in KB
  - is [2,2] ok
    - cannot reason informally, only use what is in KB
    - it must show that “if KB is true then [2,2] is OK” is valid

Inference in Computers

- formal inference is powerful
  - no limit to the complexity of the sentences it can handle
  - KB may be conjunction of 000s of sentences describing laws of gravity and current state of solar system
  - can be used to derive valid conclusions even when computer does not know the interpretation you are using
  - reports only valid conclusions

Validity and inference

- use truth table to test valid sentences
- one row for each possible combination of truth values for symbols
  - ((P ∨ H) ∧ ¬H) ⇒ P
- is this valid?
- If P=Wumpus in [3,1], H=Wumpus in [2,2]
  - learn (P ∨ H) and ¬H then what?

Validity and inference

- important point
  - if machine has some premises and a possible conclusion
  - it can test if conclusion is valid
  - build truth table
  - every row true, then conclusion is true
- computer has no access to world independently
  - inference procedure must work regardless
### Models

- sentence true under particular interpretation
  - world is a model of that sentence under that interpretation
- back to entailment
  - \( \alpha \) is entailed if the models of KB are all models of \( \alpha \)
- may consider worlds unsuitable base for formal system
  - use models as mathematical objects (model = mapping)

### Rules of inference

- inference rule
  - pattern of inference reoccurs
  - soundness shown once and for all
  - rule established, don’t need to use truth table to prove again
- notation \( \alpha \vdash \beta \) to say \( \beta \) is derived from \( \alpha \)
  - denotes inference rule - not sentence
  - if pattern above line is matched - premise true

#### 7 Rules of Inference

- **Modus Ponens (Implication Elimination)**
  - from an implication and the premise of the implication, you can infer the conclusion
  \[
  \frac{\alpha \rightarrow \beta, \alpha}{\beta}
  \]

- **And Elimination**
  - from a conjunction, you can infer any of the conjuncts
  \[
  \frac{\alpha_1 \land \alpha_2 \land \ldots \land \alpha_n}{\alpha_i}
  \]

- **Double Negation elimination**
  - from a doubly negated sentence, you can infer a positive sentence
  \[
  \frac{\neg \neg \alpha}{\alpha}
  \]

- **Unit Resolution**
  - from a disjunction, if one of the disjuncts is false, you can infer the other to be true
  \[
  \frac{\alpha \lor \beta, \neg \beta \lor \gamma}{\alpha \lor \gamma}
  \]

- **Resolution**
  - because \( \beta \) cannot be both true and false, one of the other disjuncts must be true in one of the premises. OR implication is transitive
  \[
  \frac{\alpha \lor \beta, \neg \beta \lor \gamma}{\alpha \lor \gamma}
  \]
Rules of Inference

• inference rule is sound
  – conclusion is true in all cases where premises are true

• verify soundness by using a truth table
  – logical proof
  – sequence of applications of inference rules
  – start with sentences in KB
  – finish with sentence to prove

Agent for Wumpus World

• example for situation shown in Figure 6.4a

Wumpus Agent

– KB contains
  \[ \neg S_{1,1}, \neg B_{1,1}, \neg S_{2,1} \rightarrow B_{3,1}, S_{1,2} \rightarrow \neg B_{1,2} \]

– Agent starts with rules

  • example,
    • R₁: \( \neg S_{1,1} \rightarrow \neg W_{1,1} \land \neg W_{1,2} \land \neg W_{2,1} \)
    • R₂: \( \neg S_{2,1} \rightarrow \neg W_{1,1} \land \neg W_{1,2} \land \neg W_{2,1} \land \neg W_{1,3} \)
    • R₃: \( \neg S_{1,2} \rightarrow \neg W_{1,1} \land \neg W_{1,2} \land \neg W_{2,1} \land \neg W_{1,3} \land \neg W_{2,2} \)

• another useful fact
  \[ \neg S_{1,2} \rightarrow W_{1,2} \lor W_{2,2} \lor W_{1,1} \]

– given these sentences, deduce where the wumpus is

  – construct truth table for KB \( \Rightarrow W_{1,3} \) to show sentence is valid
  – 2¹² = 4096 rows in truth table
  – use inference rules
  – by elimination show \( W_{1,3} \) is true

Wumpus Agent

• 1 Apply Modus Ponens to \( \neg S_{1,1} \) and R1
  \[ \neg W_{1,1} \land \neg W_{1,2} \land \neg W_{2,1} \]

• 2 Apply And Elimination
  \[ \neg W_{1,1} \lor \neg W_{1,2} \lor \neg W_{2,1} \]

• 3 Apply Modus Ponens to \( \neg S_{2,1} \) and R2
  then apply And Eliminations to get
  \[ \neg W_{2,2} \land \neg W_{1,3} \land \neg W_{1,1} \]

• 4 Apply Modus Ponens to \( S_{1,2} \) and R4
  \[ W_{1,3} \lor W_{1,2} \lor W_{2,2} \lor W_{1,1} \]

• 5 Apply Unit Resolution
  \[ \alpha is W_{1,3} \lor W_{1,2} \lor W_{2,2} and \ beta is W_{1,3} \]
  \[ get W_{1,3} \lor W_{1,2} \]

• 6 Apply unit resolution
  \[ \alpha is W_{1,3} \lor W_{1,2} and \ beta is W_{2,2} \]
  \[ W_{1,3} \lor W_{1,2} \]

• 7 one more resolution
  \[ \alpha is W_{1,3} and \ beta is W_{1,2} \]
  \[ get W_{1,3} \]
Knowledge into Action

- propositional logic
  - infer knowledge - where is wumpus
  - only useful if it helps Agent act
- need additional rules
  - relate to current state of world to actions to take
  - $A[1,1] \wedge \text{East}[A] \wedge W[2,1] \Rightarrow \neg\text{Forward}$

Problems with Agent

- number of propositions
- using current notation, to state
  - don’t go forward if agent in front of you
  - requires 64 rules (16 squares x 4 directions)
- large number of rules slows agent down
- taxing to set out all rules
- agent moves - world changes
  - $A[1,1] \text{ false } A[1,2] \text{ true }$
  - may not be able to forget $A[1,1]$

- if Agent is to remember
  - need different propositional symbols for each location
- difficulties
  - how long is game
  - have to write time dependent versions of each rule
- Main Problem
  - only one representational device - proposition
  - 100 steps requires 6400 rules
- first order logic (objects and relations)