Knowledge Representation

CS 486/686: Introduction to Artificial Intelligence

Outline

- Knowledge-based agents
- Logics in general
- Propositional Logic& Reasoning
- First Order Logic

Introduction

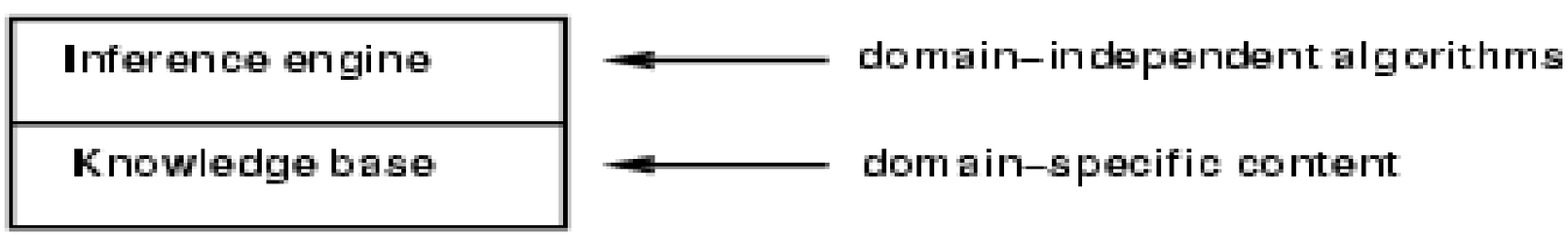
- So far we have taken the following approach
  - Figure out exactly what the problem is (problem definition)
  - Design or pick an algorithm to solve the problem (search algorithm)
  - Execute the program

Knowledge-Based Agents

- An alternative approach
  - Identify the knowledge needed to solve the problem
  - Write down this knowledge in some language
  - Use logical consequences to solve the problem

Knowledge-Based Agents

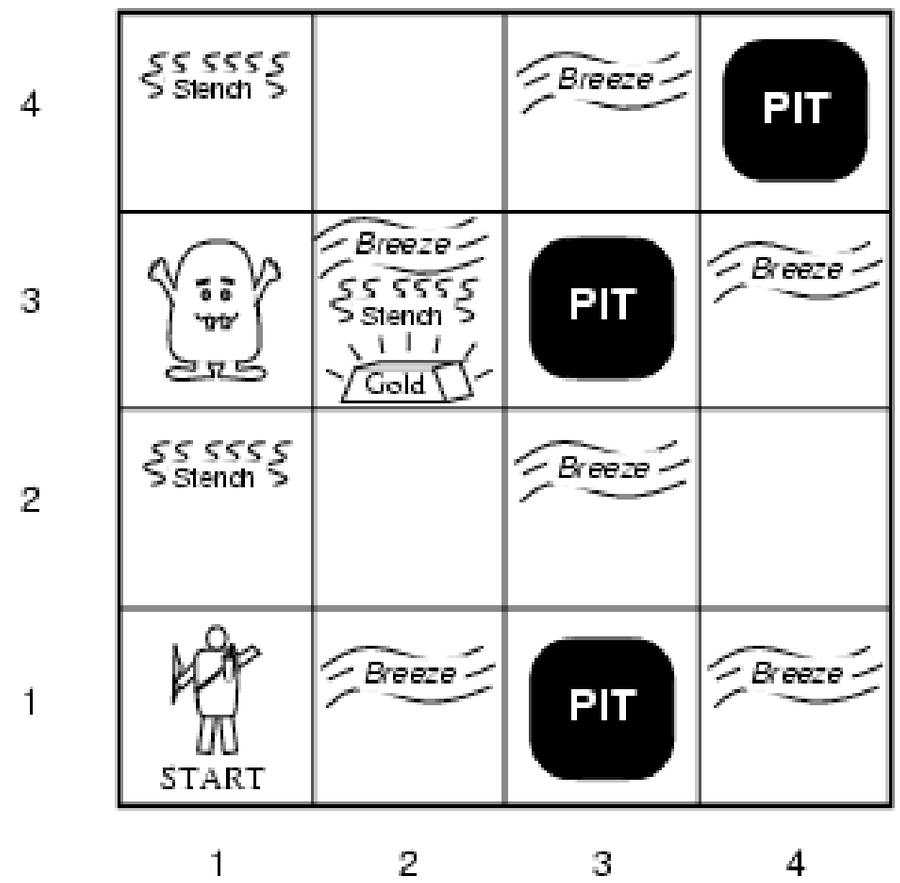
- Ideally
  - We tell the agent what it needs to know
  - The agent infers what to do and how to do it
- Agent has two parts
  - **Knowledge base**: Set of facts expressed in a formal standard language
  - **Inference engine**: Rules for deducing new facts



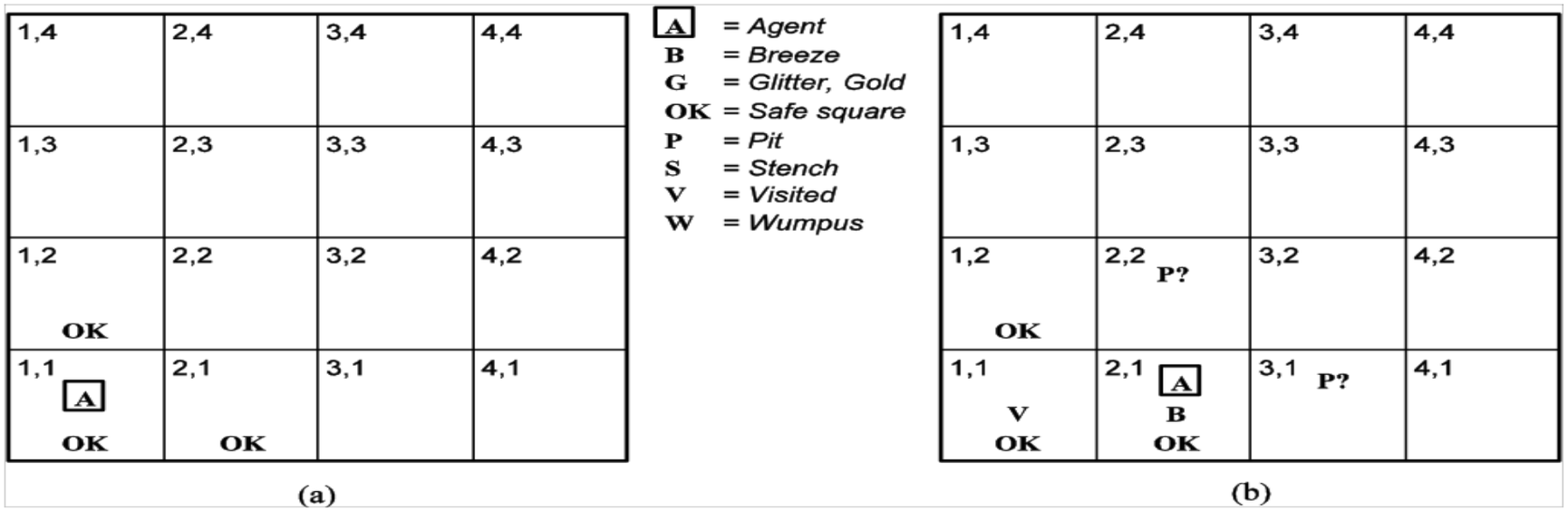
An Example: Wumpus World

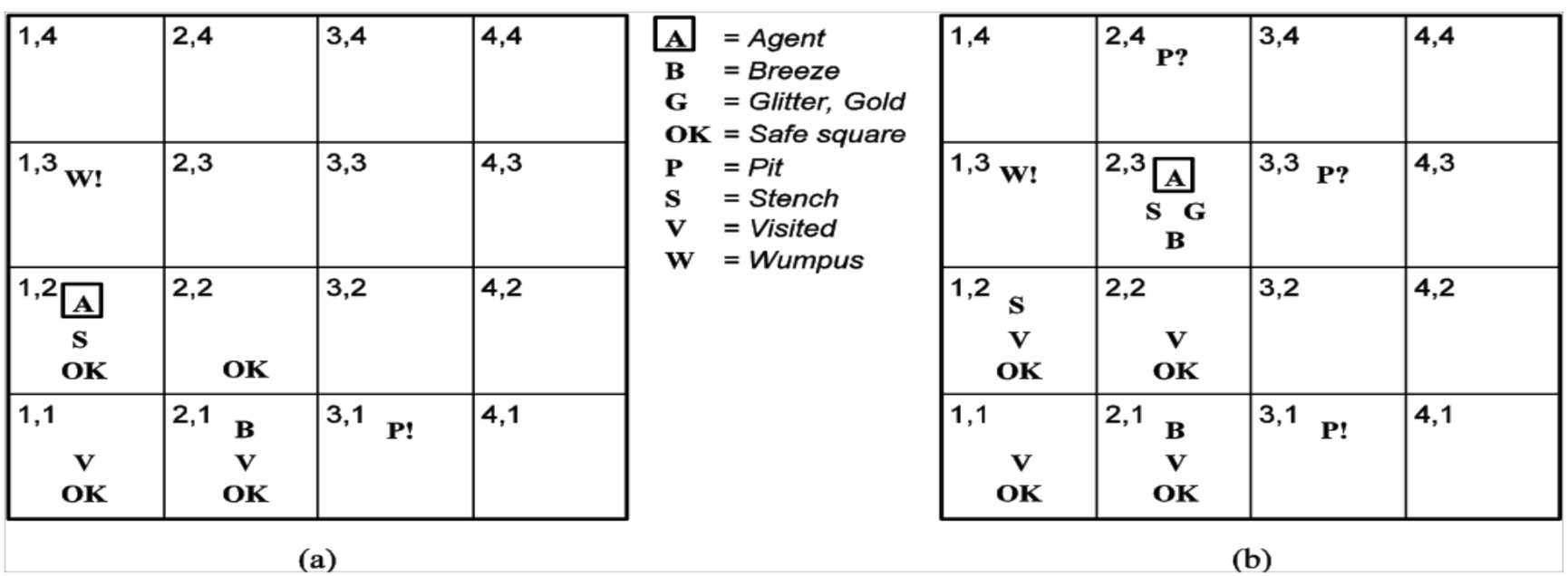
- **Goal**:
  - Get gold back to start without falling into a pit or getting eaten by the wumpus
- **Environment**
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square

- **Sensors**: Stench, Breeze, Glitter, Bump, Scream
- **Actuators**: Left turn, Right turn, Forward, Grab, Release, Shoot



Wumpus World





Outline

- Knowledge-based agents
- Logics in general
- Propositional Logic& Reasoning
- First Order Logic

What Is A Logic?

- Logic
  - A formal language for representing information so that conclusions can be drawn
- Logics have 2 components
  - **Syntax**: defines the sentences of the language
  - **Semantics**: defines the meaning of the sentences

Entailment

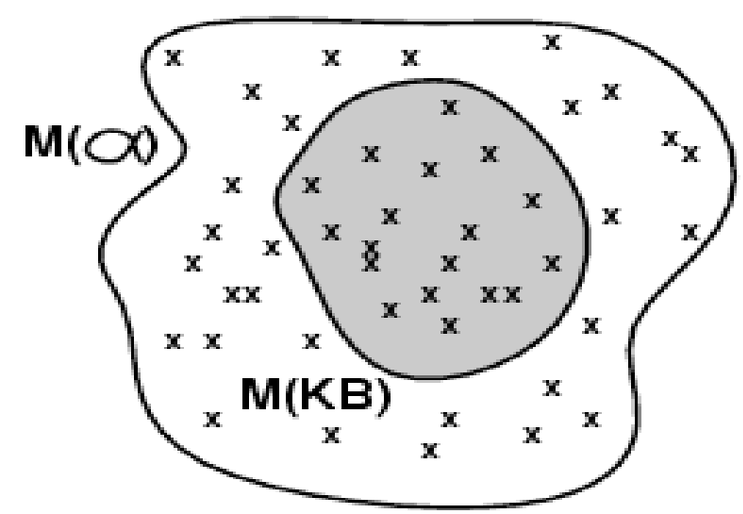
- Entailment means that “one thing follows from another”
  - KB |= α
- Knowledge base (KB) **entails** sentence α if and only if α is true in all possible worlds where KB is true
- Example:
  - KB: I finished the AI assignment. I am happy
  - α: I finished the AI assignment and I am happy.

Models

- A model is a formal “possible world” where a sentence can be evaluated
  - m is a model of sentence α if α is true in m
- M(α) is the set of all models of α
- KB |=α if and only of M(KB)⊆M(α)

KB: I finished the AI homework and I did not sleep last night

α : I finished the AI homework



Inference

- Given a KB, we want to be able to draw conclusions from it
- **Inference procedure:** KB $\vdash_i \alpha$
  - Sentence α can be derived from KB by inference algorithm
- Desired properties:
  - **Soundness:** the procedure only infers true statements
    - If KB $\vdash_i \alpha$ then KB $\models \alpha$
  - **Completeness**: the procedure can generate all true statements
    - IF KB $\models \alpha$ then it is true that KB $\vdash_i \alpha$

Outline

- Knowledge-based agents
- Logics in general
- Propositional Logic & Reasoning
- First Order Logic

Propositional Logic

- **Atomic Symbols**: P, Q, R,...
  - Each symbol stands for a proposition that can be either True or False
- **Logical Connectives**
  - $\neg$ (negation)
  - $\vee$ (or)
  - $\wedge$ (and)
  - $\Rightarrow$ (implies)
  - $\Leftrightarrow$ (if and only if, equivalence)

Inference: Propositional Logic

- Using truth tables is
  - **Sound**: direct definition of entailment
  - **Complete**: works for any KB and α and always terminates

- But...
  - Really inefficient
  - If there are n symbols, then there are 2^n models

Inference Rules

- Given a KB we want to derive conclusions
  - Proof: sequence of inference rule applications

Modus Ponens

$$\frac{\alpha, \alpha \Rightarrow \beta}{\beta}$$

Resolution

$$\frac{\alpha \vee \beta, \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

And Elimination

$$\frac{\alpha \wedge \beta}{\alpha}$$

Unit Resolution

$$\frac{\alpha \vee \beta, \neg\beta}{\alpha}$$

Resolution

- Resolution is a **sound** and **complete** inference rule
  - Any complete search algorithm, applying only the resolution rule, can derive any conclusion entailed by any knowledge base in propositional logic.

Caveat: Given that 〈 is true, we can not automatically generate 〈 V® is true. However, we can find the answer to the question "Is 〈 V® true".

Conjunctive Normal Form

- Resolution is applied to clauses of the form $\alpha \vee \beta \vee \ldots \vee \gamma$
- Any clause in propositional logic is logically equivalent to a clause in CNF
  - conjunction of disjunctions
  - eg. $(P \vee \neg Q \vee R) \wedge (\neg Q \vee A \vee B) \wedge \ldots$

Converting to CNF

1. Eliminate $\Leftrightarrow$, replacing $P \Leftrightarrow Q$ with $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$
2. Eliminate $\Rightarrow$, replacing $P \Rightarrow Q$ with $\neg P \vee Q$
3. Move "$\neg$" inwards, using $\neg(\neg P) = P$, $\neg(P \wedge Q) = \neg P \vee \neg Q$ and $\neg(P \vee Q) = \neg P \wedge \neg Q$
4. Distribute $\vee$ over $\wedge$ where possible

Resolution Algorithm

- Recall: To show KB|=α, we show that (KB∧¬α) is unsatisfiable
- **Resolution Algorithm:**
  - Convert (KB∧¬α) to CNF
  - For every pair of clauses that contain complementary literals
    - Apply resolution to produce a new clause
    - Add new clause to set of clauses
    - Continue until
      - No new clauses are being added (KB does not entail α) or
      - Two clauses resolve to produce empty clause (KB|=α)

Complexity of Inference

- Inference for propositional logic is NP-complete
- If all clauses are **Horn clauses**, then inference is linear in size of KB!
  - Horn clause: Disjunction of literals where at most one literal is positive
    - $\neg P \vee Q \vee \neg R$ is a Horn clause
    - $P \vee Q \vee R$ is not a Horn clause
  - Every Horn clauses establishes exactly one new fact
    - $\neg P \vee Q \vee \neg R \Leftrightarrow (P \wedge R) \Rightarrow Q$
    - We add all new facts in n passes

Forward Chaining

- When a new sentence α is added to the KB
  - Look for all sentences that share literals with α
  - Perform resolution
  - Add new sentence to KB and continue
- Forward chaining is
  - Data-driven
  - Eager: new facts are inferred as soon as possible

Backward Chaining

- When a query q is asked of the KB
  - If q is in the KB, return True
  - Otherwise, use resolution for q with other sentences in the KB and continue from result
- Backward chaining is
  - Goal driven: Centers reasoning around query being asked
  - Lazy: new facts are inferred only when needed

Forward vs Backward

- Which is better? That depends!
- Backward Chaining:
  - Does not grow the KB as much
  - Focused on proof so is generally more efficient
  - Does nothing until a question is asked
  - Typically used in proofs by contradiction

Forward vs Backward

- Forward Chaining
  - Extends the KB and improves understanding of the world
  - Typically used in tasks where the focus is on providing a model of the world

Outline

- Knowledge-based agents
- Logics in general
- Propositional Logic & Reasoning
- First Order Logic

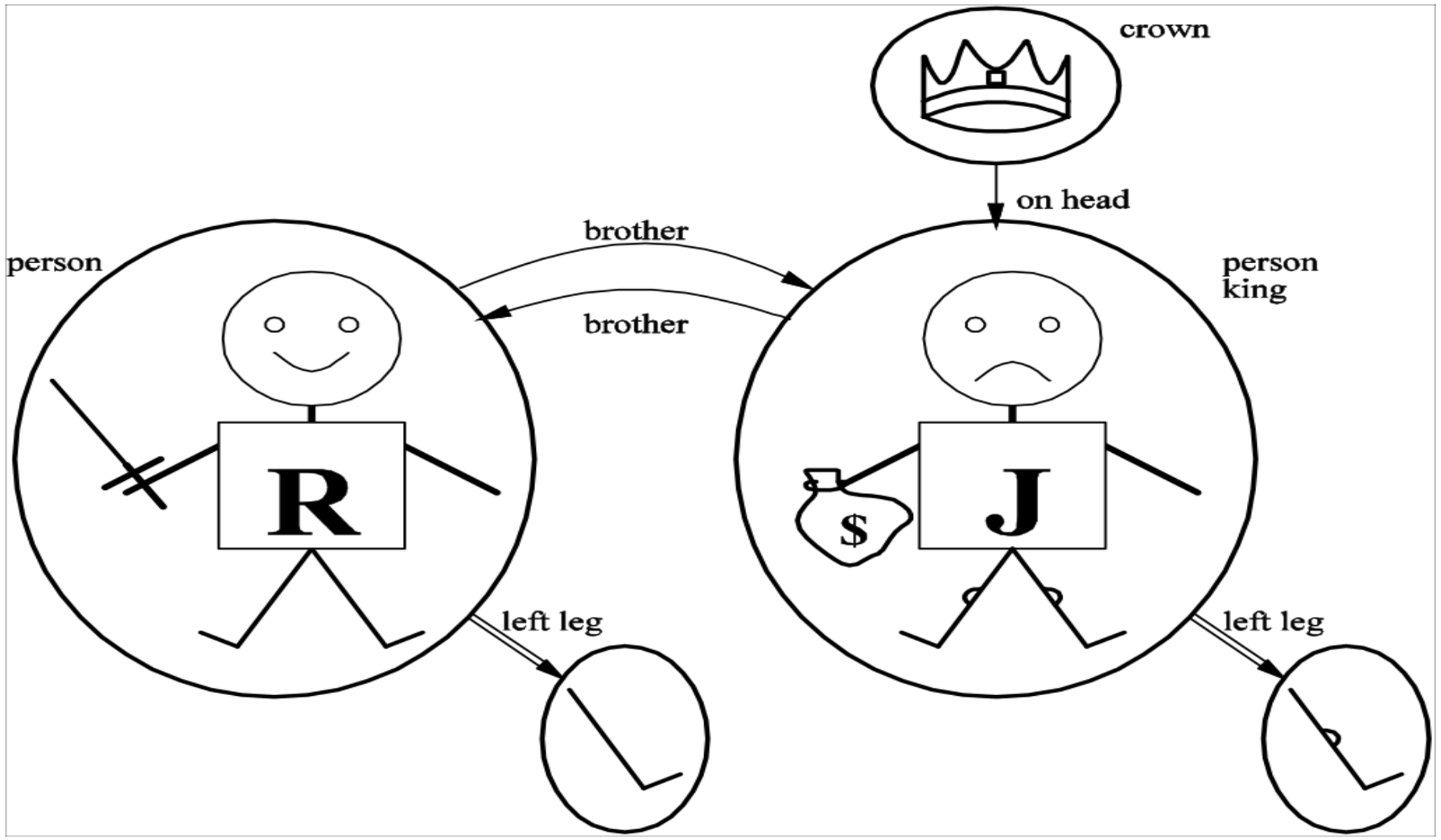
First Order Logic

- New elements
  - Predicates
    - Define objects, properties, relationships
  - Quantifiers
    - ∀ (for all), ∃ (there exists) are used in statements that apply to a class of objects
- Example: $\forall x\ On(x, Table) \Rightarrow Fruit(x)$

Sentences

- Terms
  - Constants, variables, function($term_1$,…,$term_n$)
- Atomic Sentences
  - Predicate($term_1$,$term_2$), $term_1$=$term_2$
- Complex Sentences
  - Combine atomic sentences with connectives
  - Likes(Alice, IceCream)∧Likes(Bob, IceCream)

Semantics


crown
on head
brother
brother
person
person
king
R
J
$
left leg
left leg

Inference and FOL

- We know how to do inference in Propositional Logic: find α such that KB|=α
  - Is it possible to use these techniques for FOL?
  - Have to handle quantifiers, predicates, functions, ...

Universal Instantiation

- Given sentence $\forall x\ P(x) \wedge Q(x) \Rightarrow R(x)$ then we want to infer $P(John) \wedge Q(John) \Rightarrow R(John)$ and $P(Anne) \wedge P(Anne) \Rightarrow R(Anne)$ and ...

<u>Universal Instantiation (UI)</u>

v is a variable

$$\frac{\forall v \alpha}{\mathrm{SUBST}(\{v/g\}\alpha)}$$

$\langle$ is a sentence

Substitute g for all occurrences of v in $\langle$

g is a ground term*

Existential Instantiation

- For any sentence α, variable v and constant symbol K **that does not appear anywhere in the KB**

Skolum Constant

$$\frac{\exists v \alpha}{\mathrm{SUBST}(\{x/K\}, \alpha)}$$

<u>Example</u>

∃ x Crown(x) yields

Crown(C_1) (C_1 is a new constant)

Reduction to Propositional Inference

- Suppose the KB contained the following
  - $\forall x\ Cat(x) \wedge Orange(x) \Rightarrow Cute(x)$
  - Orange(Kitty)
  - Cat(Kitty)
  - Sister(Kitty, Katy)

- Instantiating the universal sentence in all possible ways we have a new KB:
  - $Cat(Kitty) \wedge Orange(Kitty) \Rightarrow Cute(Kitty)$
  - $Cat(Katy) \wedge Orange(Katy) \Rightarrow Cute(Katy)$
  - Cat(Kitty)
  - Sister(Kitty, Katy)

- The new KB is in propositional form. The symbols are
  - Cat(Kitty), Cat(Katy), Orange(Kitty), Cute(Katy), Sister(Kitty,Katy), …

Reduction Continued

- Every FOL KB can be propositionalized
  - Transformed into propositional logic
- This preserves entailment
  - A ground sentence is entailed by the new KB if and only if it was entailed in the original KB
- Thus we can apply resolution (sound and complete) and return the result?

Reduction Continued

- **Problem**: Works if α is entailed by the KB but it loops forever if α is not entailed
- **Theorem**: (Turing 1936, Church 1936) Entailment in FOL is semi-decidable.
  - Algorithms exist that say yes to every entailed sentence
  - No algorithm exists that says no to every unentailed sentence

Reduction Continued

- **Theorem**: (Turing 1936, Church 1936) Entailment in FOL is semi-decidable.
- **Proof Intuition**
  - Can write infinitely expandable statements.
  - Even with IDS, never know when to stop expanding and give up.

Can we reason with FOL?

- Problem is with universal instantiation
  - Generates many irrelevant sentences due to substitutions
- Workaround: Unification (See R&N 9.2)

Conclusion

- Syntax, semantics, entailment and inference
- Propositional logic and FOL
- Understand how forward-chaining, backward-chaining and resolution work