Knowledge Representation

Warning: The following slides contain symbols that the mathematically sensitive may find disturbing

Introduction

- What is Knowledge?
  - Facts/data that have meaning for an agent
  - Facts: sentences about the world that may be true or false
  - Internal descriptions model external states and events
- Why is it hard to represent?
  - Difficulty is in finding the most appropriate representation that fits the world and delivers correct results when it is manipulated by an inference system
- Knowledge base
  - a set of representations of facts about the world

Knowledge Levels

- Knowledge level
  - epistemological level
  - knowledge about the world
- Logical level
  - knowledge encoded in sentences
- Implementation level
  - physical representations of sentences

Representation Characteristics

- Expressive adequacy
  - sufficiently precise, complete, clear
- Representational power
  - coverage, flexibility, extensibility
- Inferential adequacy
  - manipulation ability for reasoning
- Inferential efficiency
  - ability to control reasoning
- Expliciteness
  - clarity, tracing, explanation

Main Schemes

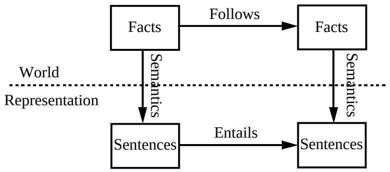
- Declarative
  - Logic (Propositional, FOPL)
  - Associative Networks
  - Models
- Imperative
  - Procedural
- Mixed
  - Frames
  - Production Systems

Some terminology

- Entailment
  - generates new sentences that are necessarilly true, from old ones
    - mirrors one fact following from another
    - KB |= α
- Sound (truth preserving)
  - inferences procedure only produces entailed sentences
- Derivation
  - inference procedure can be described by the sentences it can derive
  - KB |- α
- Proof
  - record of sound inference procedure
- Complete
  - finds procedure for all entailed sentences

Representation & Logic

- Object of KR
  - to express knowledge in a computer tractable form - that performs well
- KR Language:
  - Syntax
    - describes possible configurations that constitute sentences
  - Semantics
    - determines facts to which sentences refer



Semantics

- Meaning
  - what it states about the world
    - "it has *this* form not *that* form"
- Composition
  - Sentence meaning is a function of the meaning of its parts
- Interpretation
  - how the meaning of a sentence corresponds to the facts in the world
  - a sentence does not mean anything *in itself*!
  - Systematic
  - True or False

Inference

- Any process by which conclusions are reached
- Valid sentence
  - true under all possible interpretations in all possible worlds
- Satisfiable sentence
  - true under some interpretation in some world
- Unsatisfiable sentence
  - false under all possible interpretations in all possible worlds

Propositional Logic

- Syntax
  - symbols
    - constants: *True, False*
    - propositions symbols: P, Q, etc
    - connectives: ^, v, ~, =>, <=>
  - sentences
    - constants and propositions are sentences
    - complex sentences can be formed from simple (atomic) ones via the connectives
      - conjunction: P^Q
      - disjunction: PvQ
      - implication: P => Q
      - equivalence: P <=> Q
      - negation: ~P
    - precedence: ~, ^, v, =>, <=>

Logics

- Summary
  - syntax:
    - how to make sentences
  - semantics:
    - systematic constraints on how sentences relate to states of affairs
  - proof theory
    - rules for deducing entailments
- Propositional logic
- Predicate logic
- Ontological commitments
- Epistemological commitments

Propositional Logic (2)

- Semantics
  - interpretation of the proposition symbols - can mean anything
  - *True* and *False* have fixed meaning
  - complex sentence meaning derived from meaning of parts
- Truth tables

P	Q	~P	P^Q	PvQ	P=>Q	P<=>Q
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

Propositional Logic (3)

- Validity & Inference
- Models

Rules of Inference

- Modus Ponens
  - P => Q, P |- Q
- And-elimination
  - from a conjunction infer any of the conjuncts
    - A1^A2^. . . .^An |- Ai
- And-introduction
  - from a list of sentences infer their conjunction
    - A1, A2, A3 |- A1^A2^A3
- Or-introduction
  - from a sentence infer its disjunction with anything
    - A1 |- A1vA2vA3 . . .

Valid Sentences

P ^ (Q ^ R)	<=>	(P ^ Q) ^ R	Assoc. Conj.
P v (Q v R)	<=>	(P v Q) v R	Assoc. Disj.
P ^ Q	<=>	Q ^ P	Comm. Conj
P v Q	<=>	Q v P	Comm. Disj
P ^ (Q v R)	<=>	(P ^ Q) v (P ^ R)	
P v (Q ^ R)	<=>	(P v Q) ^ (P v R)	
~(P ^ Q)	<=>	~P v ~Q	de Morgan
~(P v Q)	<=>	~P ^ ~Q	de Morgan
P => Q	<=>	~Q => ~P	Contraposition
~~P	<=>	P	
P => Q	<=>	~P v Q	
P <=> Q	<=>	(P => Q) ^ (Q => P)	
P <=> Q	<=>	(P ^ Q) v (~P ^ ~Q)	
P ^ ~P	<=>	*False*	
P v ~P	<=>	*True*	

Rules of Inference (2)

- Double-negation-elimination
  - from a doubly negated sentence infer a positive one
    - ~~P |- P
- Unit Resolution
  - if one disjunct is false infer that the other is true
    - PvQ, ~Q |- P
- Resolution
  - a sentence cannot be both true and false so one of the other disjuncts must be true (or implication is transitive)
    - PvQ, ~QvR |- PvR
    - ~P => Q, Q => R |- ~P => R

Bits and Pieces

- Substitution
  - From a formula A in which occurs once or more a propositional variable v infer what results from substituting any formula B for v in A
- Complexity
  - NP-complete, though hard problems quite rare
- Monotonicity
  - adding new sentences does not change entailment
  - **if** KB1 |= a **then** (KB1 U KB2) |= a
- Propositional logic is sound and complete
- Problems
  - too many rules
  - no variables

First Order Predicate Logic

Components of FOPL

- Objects
  - properties and relations
- Constant symbols
  - *a, b, c*
- Variable symbols
  - *A, B, C*
- Function symbols
  - *sine, age, name*
- Predicate symbols
  - *likes, borrows*
- Terms
  - constants, variables and functions
- Sentences
  - predicates and quantifiers

Components of FOPL (2)

- Tuple
  - ordered set of objects
  - provides the *arity* of a function or predicate
- Universal quantification (∀)
  - specifies that a predicate refers to *All* variable values in its scope
- Existential quantification(∃)
  - specifies that a predicate refers to *Some* variable values in its scope
- Atomic formula
  - a predicate applied to a tuple of terms

Well Formed Formulae

- Atomic formulae are wff's
- If W, W1 and W2 are wff's - so are:
  - W and ~W
  - W1 ^ W2 and W1 v W2
  - W1 => W2 and W1 <=> W2
  - (QX)W
    - where (QX) is any quantified
- Scope
  - in (QX)W the scope of (QX) is W
- Binding
  - in a wff the occurence of a variable X is bound iff:
    - it is in the scope of a quantifier (QX), or
    - it is in the quantifier (QX)
- Sentence
  - wff with all variables completely bound

Theorem Proving

- Well developed with long history in AI
- Theorem proved from a set of axioms using inference rules:

```
...............
...............
...............       axioms (facts)
-------------
-------------         inference rules
. . . . . . . . .
. . . . . . . . .
. . . . . . . . .     steps in proof
xxxxxxxxx             theorem
```

- To prove theorem S
  - derive many new premises
    - search process
    - combinatorial problems (ie exponential)

Instantiation

- Universal instantiation (UI)
  - $\forall$Xg(X) |- g[X/t] where t, a term, is <u>free</u> for X in g
  - in general: t is free for X in g iff X does not occur within the scope of a quantifier of some variable in t
- Existential instantiation (EI)
  - $\exists$Xg(X) |- g[X/p(Y1, Y2, ... Yn)]
  - p is a new function constant where Y1 .... Yn are free variables in g

Theorem Proving (2)

If S can be found, it may take many search steps throught the space of possible premises

If S is **not** provable ie does **not** derive from the axioms, then the search will not terminate

Thus, first order predicate logic is **semidecidable**

This is **Decuctive** logic

Induction

- Extracting rules from examples
  - Given the facts: island(malta)
    island(ireland)
    island(tahiti)
    island(iceland)

  infer: $\forall X\{island(X)\}$

  ie everything is an island

The type of inference needed for learning.

It is only **plausible** inference. Counter examples might be found which would invalidate the inference. In that case clustering methods might be used to group the axioms.

Abduction

Make rules about results:

(A => B), B |- A

ie if A implies B and B is true then A is true

Again this is a **plausible** inference, not a legal inference [Because B = T and A = F is allowed by A => B]

eg: seriously-ill(Student) => not-in-lecture(Student)

is a reasonable statement, but the abductive inference that a student is seriously ill *because* they are not in the lecture is perhaps rather trusting! :-)

We use this sort of reasoning when results are known and "causes" are to be inferred. This is typical in many diagnostic expert systems.

An example of Deduction

Sherlock lists the facts:

1. Simon is in the library
2. John is in the library
3. A gun is in the library
4. John is not breathing

and some rules of inference

5. If a person is not breathing they are dead.
6. A gun is a weapon.
7. A weapon and a dead person indicate a crime.

Prove that a crime has been commited in the library

Logic for KB's

We can easily build a **database** of facts. We simple store lots of items like:

(it is cloudy)

However, a **knowledge base** requires more. we need facilities for manipulation, refinement and generation of new knowledge from old. Logic has a very well defined framework for this.

Example

Facts: on(component-A tray)
on(component-B tray)
inside(tray oven)

Rules of inference:
inside(X oven) => temp(X v-hot)
on(X Y) ^ inside(Y Z) => inside(X Z)

Logic for KB's (2)

Using the inference rules we can generate:
New knowledge:

temp(tray v-hot)
inside(component-A oven)
inside(component-B oven)
temp(component-A v-hot)
temp(component-B v-hot)

This is **forward chaining** - the facts are generated at assertion time ie forward consequences are explored. If we query the KB and ask:

temp(component-A v-hot)?

we immediately get TRUE as the answer because a direct match is available.

Backward chaining generates new axioms at query time. No new KB material would be produced *until* a query is given

eg temp(tray v-hot)

This would produce a **sub-goal** (inside(tray oven)), which would return TRUE. From this it can be inferred that the answer to the query is TRUE

KB's & DB's

•Multiple use of facts

– Inference: produce new facts (using given facts and rules of inference) Deductive & inductive
– Access: explanations, question-answering systems.
– Matching: in order to classify types, confirm hypotheses, decompose inputs, correct pattern errors. Different levels of matching
syntactic (form)
parmetric (variables)
semantic (function)

•Incompleteness

A KB is usually incomplete. This affects many issues: its design, organisation, and the knowledge acquisition process.

New knowledge can be acquired by: interaction with experts in their environment, self diagnosis or discovery, and learning.

It should be said that in present day expert systems, acquiring new knowledge is a maintenance task, in the same way it is for any large program.

Semantic Nets

Developed to represent meaning of english words

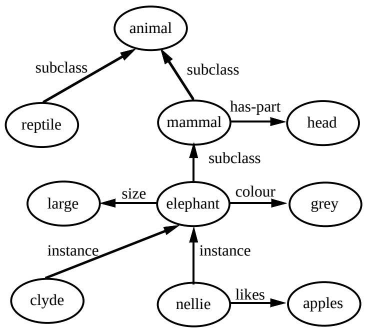
Knowledge represented as a graph:

Nodes are concepts
Links are relations: subclass
instance
has-part, etc.

Often represented in terms of set theory - in order to make the semantics of the relations clear

Precursor to Object Orientation

Semantic Nets (2)



Frames

A variant of semantic networks.

All information relevant to a particular concept is stored in a single complex entity - a **Frame**

Permits the use of **defaults**: to indicate *typical* values

```
Mammal:
   subclass:       Animal
   has-part:       head
Elephant:
   subclass:       Mammal
   colour:         grey
   size:           large
Nellie
  instance:        Elephant
  likes:           apples
```