Representation & Logic

Representation

- We need a way to enter "world facts" into the computer in such a manner that the computer can ***reason*** ("make inferences") with and about them
- normal English is insufficient
  - too hard currently
    - ambiguity
    - how do we draw inferences in natural languages?

Physical Symbol Hypothesis (again)

- Intelligence can be achieved by
  - symbols that ***represent*** the significant aspects of the given problem domain
  - ***operations*** on these basic & compound symbols that generate potential solutions
  - ***search*** to find the a solution among solutions
- We've looked at #3; now we examine #1 & #2

Requirements for an AI language

- Handle ***qualitative*** *knowledge*
- Allow inference
  - inference rules save us from ***ex****plicitly* writing down every fact ("deductive database")
- Allow representation of general principles (*rules*) and specific situations (*facts*)
- capture complex situations (time, change, etc.)
- support ***meta-level*** reasoning
  - analyzing one's knowledge, reasoning, learning, etc.
  - stepping outside the system

First-order predicate calculus (FOPC)

- The **core** representation language
- In terms of representation, it is well-defined (mathematical logic)
- In terms of reasoning, it is
  - ***sound*****:** inferences are correct
  - ***complete*****:** all possible inferences can be *mechanically* (syntactically) produced
- **Note:** one can & often does use FOPC as a ***representation language*** while using a more efficient (but less sound & complete) ***reasoning system***

Russell & logical atomism

- The belief that *"the world can be analyzed into a number of separate things with relations and so forth"* (1918)
  - in opposition to a sort of holism which holds that not everything can be analyzed into parts & put back together to form the original whole
- **Methodology:** take complex entities & dissolve them into simple atoms
  - we take a seemingly complex thing & **enumerate** all of its properties & relationships

Language

- Problem: what are the atoms?
- Solution: a logically perfect (ideal) language
  - one-to-one mapping between ***facts*** in the world & "words" (***symbols***)
    - thus there is no ambiguity & no inter-dependence regarding facts
  - relations between facts
- **Two categories**
  - atoms, relationships
  - logical connectives: and, or, if-then, not, etc.

Propositional calculus

- Rather than jumping right into FOPC, we begin with propositional calculus
- FOPC's little brother
  - No quantification
  - No equality

“Data types”

- Propositions
  - Boolean-valued
  - P, Q, R,…
    - statements about the world
    - R : it’s-raining-now
    - needn’t be a single letter
- Truth symbols
  - true, false
  - same meaning as in English

Connectives

- and ($\wedge$)
- or ($\vee$)
- implies ($\Rightarrow$)
- equivalent ($\Leftrightarrow$)
- not ($\neg$)
- used to combine simple statements into more complex ones

Truth tables

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
False	*False*	*True*	*False*	*False*	*True*	*True*
False	*True*	*True*	*False*	*True*	*True*	*False*
True	*False*	*False*	*False*	*True*	*False*	*False*
True	*True*	*False*	*True*	*True*	*True*	*True*

Well-formed formulae (**wffs**)

- Sentences
  - just like in a programming language, there are rules (***syntax***) for legally creating compound statements
  - remember: we're always stating a truth about the world,
    - hence every **wff** is something that has a *Boolean value* (it is either a true or a false statement about the world)

Syntax rules

- Propositions (P, Q, R, …) are wffs
- Truth symbols (true, false) are wffs
- If A is a wff, so are $\neg A$ and (A)
- If A and B are wffs, so are
  - $A \wedge B$
  - $A \vee B$
  - $A \Rightarrow B$
  - $A \Leftrightarrow B$

Interpretation example

- $[(P \vee Q) \wedge R] \Rightarrow (S \Leftrightarrow V)$
- First, we need an ***interpretation***
  - truth values for our "atomic" sentences
  - P : T; Q : F; R : T; S : F; V : T
- Then evaluate
  - $P \vee Q$: T
  - $(P \vee Q) \wedge R$: T
  - $S \Leftrightarrow V$: F
  - whole thing : F

Connectives

- Think of connectives as functions that take truth values as their arguments and return a truth value
- The output of these functions is determined by the previous truth tables
- Just like a normal function that maps inputs to outputs;
  - *in this case, since the possible values are relatively few,* ***we can enumerate all of them***

Are these WFFs?

- $P \; Q \; R$
- $(P \wedge Q) \vee (R \vee S)$
- $P \Rightarrow \vee (Q \wedge R)$

Example of k-rep in prop calc

- R : “It is raining”
- B : “Take the bus to class”
- W : “Walk to class”
- Some things to tell our agent
  - $R \Rightarrow B$ (“If it is raining, (then) take the bus to class”)
  - $\neg R \Rightarrow W$ (“If it is not raining, (then) walk to class”)
- Ideally, we would like our agent to sense that it is raining & then decide to take the bus

Validity

- A wff is ***valid*** if it is true under all possible interpretations (i.e., all possible "variable settings") [use truth table to show this]
  - **$P \vee \neg P$ is valid**
    - if P is true, then the whole sentence is true
    - if P is false, then ~P is true and the whole sentence is true
  - **$(P \wedge \neg Q) \vee (\neg P \wedge Q)$ isn't valid**
    - when P is true & Q is true, the sentence isn't true
    - in order to not be valid, there only need exist one counter-example
  - **valid** is also called a ***tautology***

Satisfiable

- A wff is ***satisfiable*** if some interpretation makes it true
- Examples:
  - P is satisfiable
    - simply let P be true
  - $P \wedge \neg P$ is not satisifiable
    - if P is true, $\neg P$ is false, the whole sentence is false
    - if P is false, the whole sentence is false
  - $P \Rightarrow Q$ is satisfiable
    - several ways: P is true, Q is true; etc.
  - A wff that cannot be satisfied is called a ***contradiction***

Soundness

What is soundness?

- An inference procedure is **sound** if it only generates entailed wffs
  - a wff is *entailed* if it is necessarily true given the previously true wffs
  - "necessarily true" means it is true given the previously true wffs ***on any interpretation*** (on any truth assignment to the symbols)
  - this is written as **K**B **|=** A
    - for example, {A ⇒ B, A} |= B
  - examples of sound inference procedures are: modus ponens, resolution, and-introduction, etc.
    - the wffs they generate a **true under any interpretation**

Why do we care about soundness?

- Sound inference procedures are ***truth-preserving***
  - none of the wffs produced by the inference procedure contradict any of the given wffs or any of the other derived wffs
  - all the wffs produced are ***consistent*** with all the wffs given or generated
  - thus, any ***model*** for the original set of wffs is also a model for the derived set of wffs
  - we **can write this as**: "For every KB **|-** A, **K**B **|=** A"

Model

What is a model?

- A model is an ***interpretation*** that makes all the wffs in a set **true**
  - for example, a model for $\{A \wedge B, \neg B \vee C\}$ is
    - A : true, B : true, C : true
    - note: there may be more than one model
  - thus, KB |= A means every model of KB is also a model of A
    - every assignment of truth values to the wffs in KB that make all of the wffs in KB true, also make A true

What is an interpretation?

- An **interpretation** is the assignment of ***facts*** to symbols (or: proposition letters)
  - a fact is taken to be either true or false about the world
  - thus, by providing an interpretation, we also provide the *truth value* of each of symbol
  - example
    - P : it-is-raining-here-now
    - since this is either a true statement about the world or a false statement, the *value* of P is either true or false

Completeness

Completeness

- We have shown what it means to be a sound inference procedure: we only generate entailed wffs
- One other question we can ask is whether using our inference procedure we can generate ***all*** of the entailed wffs
- If we are able to do so, we say that our inference procedure is ***complete***

What is completeness?

- An inference procedure is complete if it can find a proof for any sentence that is ***entailed***
  - that is, that it can generate all the wffs consistent with the “givens” using it’s “operations”
- What is complete?
  - Are truth tables complete?
  - When are the inference rules in some set of rules complete?

Truth tables

- **Truth tables are sound and complete**
  - they enumerate every combination of truth values
    - as the number of literals increases, the size of the truth table grows exponentially ($2^{(\# \text{ of literals})}$)
  - thus, they will be able to "prove" every entailed wff (using the definitions of the connectives)
    - for a truth table, a proof is simply the truth table itself
  - they are sound because they simply enumerate all of the truth possibilities

Inference rules

- The inference rules are rather *ad hoc*
- They are sound (they only derive entailed wffs), but they aren't complete
  - for example, they cannot prove that de Morgan's law is valid
    - $\neg(A \vee B) \Rightarrow (\neg A \wedge \neg B)$
    - $\neg(A \wedge B) \Rightarrow (\neg A \vee \neg B)$
  - **solution:**
    - add more inference rules (how many are enough?),
    - use truth tables (too tedious),
    - use a different inference procedure

Direction

- We want to devise methods for **deducing new facts** that logically follow from old facts ***regardless*** of the interpretation
  - i.e., things that are ***necessarily*** true, rather than ***possibly*** true
  - we will use valid propositions (tautologies) to produce new wffs;
    - since tautologies don't change the truth "mapping" of the original wff, the new wff will have the same **"mapping"**

Review

- Propositional calculus is a precise way to tell our computer facts about the world
- ***Syntax*** says what is a “grammatical” sentence
- ***Semantics*** says whether or not a wff is true, given the truth values of our “primitive”/atomic propositions (compositional semantics)
  - truth tables define the semantics of our five connectives ($\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, $\neg$)
- ***Interpretations*** are how we (users) tell the computer the truth value of the primitive propositions

Deduction in Propositional calculus

Deduction in Propositional calculus

- Inference rules allow us to deduce new wffs from known ones
- Notation

<given wffs that match these patterns>

<we can deduce this>

And-elimination

- Given: $A_1 \wedge A_2 \wedge \ldots \wedge A_n$
- We can deduce: A_i
- If a conjunct is true, so is each individual wff that it is composed of

And-introduction

- Given: $A_1, A_2, \ldots, A_n$
- We can deduce: $A_1 \land A_2 \land \ldots \land A_n$
- if we know a bunch of wffs are true, their conjunctive combination is true

Double-negation elimination

- Given: $\neg\neg A$
- We can deduce: A
- Two negations cancel out
  - think of -(-9) = 9

Double-negation introduction

- Given: A
- We can deduce: $\neg\neg A$

Or-introduction

- Given: A
- We can deduce: $A \vee B$
- If A is true, then $A \vee B$ is also, for any B

Modus Ponens

- Given: A $\Rightarrow$ B, and also given A
- We can deduce: B
- Alternatively:
  - Given: $\neg$A $\vee$ B, and also given A
  - B
- If we "believe" a rule, and we know the the antecedent is true, we can deduce that the conclusion is true

Unit resolution

- Given: $A \vee B$, and also given $\neg B$
- We can deduce: A
- Alternate form
  - $\neg A \Rightarrow B, \neg B$
  - A
- Really, just a variant of modus ponens
- If at least one of two wffs is true (A or B) & we know one is false, then the other must be true

Resolution [hard one]

- Given: $A \vee B$, and also $\neg B \vee C$
- We can deduce: $A \vee C$
- Alternatively:
  - Given: $\neg A \Rightarrow B$, and also $B \Rightarrow C$
  - $\neg A \Rightarrow C$
- Case analysis on the possible values of B

Proof as a search task

- ***State representation:*** a list of wffs that are true
- ***Operators:*** our inference rules
- ***Start state:*** our “givens” (what is true initially)
- ***Goal state:*** the wff to prove is in our state’s list of known wffs

Proof form

- Write down and number (for reference) all the "givens"
- Generate new sentences using ***inference rules***
  - justify by listing the rule used & the numbers of the wffs used
  - can use previously deduced wffs, not limited to the givens
  - give a number to each newly deduced wff
- When the desired wff (that which is to be shown) is generated, we're done
  - **question:** what is our search strategy?

Sources

- Computer Science, University of Wisconsin, Madison.