The Predicate Calculus in AI

Last time, we:

Motivated the use of Logic as a

representational *language* for AI

(Can derive new facts *syntactically* -
simply by pushing symbols around)

Described propositional logic

syntax

semantics (truth tables, anyway)

inference rules

Described first order predicate calculus:

Adds *terms*:

constants, functions, variables

and *quantifiers*:

"for all" and "there exists"

Went over some ways to encode English into FOPC

The Predicate Calculus in AI

What's a "wff"? (SYNTAX)

A *term* is

1. a constant
2. a variable
3. If *f* is an n-place function,
and t_1,.., t_n are terms, then

f(t_1,...,t_n) is a term

An *atom* is:

a predicate with terms for arguments, e.g.

P(t_1,...,t_n)

A *well-formed formula (wff)* is:

i. an atom
ii. if F and G are formulas, then

NOT F, F OR G, F AND G, F IMPLIES G and F EQUIV G

are formulas
iii. If F is a formula and *x* is a free variable in F,
the (FORALL x) F and (EXISTS x) F are formulas.

The Predicate Calculus in AI

SEMANTICS of First Order Predicate Calculus

A big advantage of FOPC: a well-defined semantics

When we write down a set of facts

about some DOMAIN in FOPC

We must establish a correspondence between our

constants, functions, and predicates

and things in the domain.

This is called an INTERPRETATION of the formulas

E.g., GC <-> Gary Cottrell
JB <-> Jellybean
Human(GC) <-> Human("Gary Cottrell")
Human(father(GC)) <-> Human("George Cottrell")

NOTE: A predicate picks out a SET in the domain:
e.g. the set of all humans

The Predicate Calculus in AI

Semantics of First Order Predicate Calculus

More formally, an INTERPRETATION of a formula F is:

A nonempty domain D and an assignment of "values" to every constant, function symbol, and Predicate as follows:

1. To each constant, we assign an element of D.

2. To each n-place function symbol, we assign a mapping from $\mathbf{D}^n$ to **D**

3. To each n-place predicate symbol, we assign a mapping from $\mathbf{D}^n$ to {**T,F**}

Now the SEMANTICS is:

A formula is (means) TRUE under an interpretation if

1. If G and H are evaluated, then NOT G, G AND H, G IMPLIES H, G OR H, etc. have values in the obvious way.

2. (FORALL X) G is T if G is T for every x in D, otherwise F

3. (EXISTS X) G is T if ONE x in D makes it T else F

The Predicate Calculus in AI

Semantics of First Order Predicate Calculus

A MODEL of the domain is:

an interpretation that makes our sentences TRUE.

[N.B.: One set of "facts" delineates a space of possible models!]

Simple, right? This is known in the biz as:

MODEL THEORETIC SEMANTICS

A TAUTOLOGY (or simply, VALID formula)

is a formula that is TRUE under ANY interpretation

A FALLACY (or, an INCONSISTENT formula)

is a formula that is FALSE under ANY interpretation

A formula is SATISFIABLE if

there is at least ONE interpretation that makes it TRUE

Note: ONE LINE in a truth table counts
as an interpretation!

The Predicate Calculus in AI

Inference

A formula G is a LOGICAL CONSEQUENCE of

a set of formulas F_1, F_2, ... F_N

iff any interpretation that makes

F_1 AND F_2 AND ... AND F_N true

ALSO makes G true

The DEDUCTION THEOREM:

G is a logical consequence of a set of formulas
F_1 AND F_2 AND ... F_n

iff [(F_1 AND F_2 AND ... F_n) IMPLIES G]

is a tautology!

More important for us (REFUTATION PROOF):

ALSO iff F_1 AND F_2 AND ... F_n AND (NOT G)

is a fallacy (or inconsistent)

The Predicate Calculus in AI

Resolution

Resolution is an inference rule that is both SOUND and COMPLETE

SOUND = only true facts (logical consequences) are inferred

COMPLETE = ALL facts that follow CAN be inferred

NOTE: Modus Ponens is sound but not complete -
I can’t infer everything with modus ponens.

So usually, to get completeness, we have a collection

of inference rules.

With resolution, we only need ONE.

Basically, resolution is a CANCELLING method:

given: A OR B
NOT A OR C
---
infer: B OR C

A and NOT A "cancelled" each other.

The Predicate Calculus in AI

The Resolution Principle

You have just seen the following inference rule in action:

Given two sentences in clause form:

If one clause contains **P** and the other **NOT P**,

remove these from the two clauses and

form the disjunction of the remaining literals

[Recall: A *literal* is a predicate or its negation.

complementary literals are ones of the form

P and **NOT P**

]

Hand written slides on moving to clause form

The Predicate Calculus in AI

Resolution in propositional logic

An Example

If we are in San Diego, then it is sunny.

If it is sunny, then it is warm.

If it is warm and sunny, then it is not winter.

It is winter.

Prove: We are not in San Diego

First: Pick predicate names: SD, WINTER, W, S.

Second: Write the above facts in logic:

SD $\rightarrow$ S

S $\rightarrow$ W

S AND W $\rightarrow$ NOT WINTER

WINTER

The Predicate Calculus in AI

An example

Next: Convert to clause form:

SD $\rightarrow$ S

NOT SD OR S

S AND W $\rightarrow$ NOT WINTER

1. Remove $\rightarrow$

NOT (S AND W) OR NOT WINTER

2. Move in NOT

(NOT S OR NOT W) OR NOT WINTER

NOT S OR NOT W OR NOT WINTER

3. Already in clause form!

The Predicate Calculus in AI

CAUTION!

Note: There DOES NOT EXIST a decision procedure

that will check the validity of formulas in Logic!!!!

HOWEVER: There are procedures which will check

if a formula is valid IF IT IS valid.

If a formula is INVALID,

these procedures will NEVER TERMINATE!!!!

<=> validity in FOPC is semi-decidable

[Church, 1936, Turing 1936]

This means that full-blown resolution may NEVER TERMINATE!

The Predicate Calculus in AI

Now what?

There are three approaches to this problem:

1) use an inference rule that is INCOMPLETE,

but will finish in finite time.

e.g., only use modus ponens

Issue: Picking the right set if inference rules for

your domain and knowing what you can’t prove

2) Use a subset of FOPC:

E.g. Horn clauses (PROLOG)

Issue: Is the language you picked expressive enough?

3) Stop the prover after some bounded time

and say "Don’t know".

Issue: How useful is this?