Representations for KBS: Logic: When Sound Deduction is Required

Spring Term 3/7/200
6.871 Knowledge Based Systems
Randall Davis Instructor
Howard Shrobe Lecturer

Syntax
Proofs
Semantics
Sound Inference and Complete Inference
What Properties hold?
The Language as a Representation
  Comprehensiveness
  Ambiguity
  Lack of Commitment
Compromises between generality and tractability
Some Applications in the logic spirit
Wrap-up, What pair of glasses is this?

What is Logic?

A "Universal" Language"

A formal Inference system that "preserves truth"
  Not a guarantee of correct answers.

General spirit is to "pour in the axioms" and grind out the theorems.

E.g. to do inference, start with axioms and grind out consequences.
  To plan, put in the planning axioms and grind out the plans.
  To do diagnosis, put in the description of the device and grind out the diagnosis.

A means for formally relating syntax to semantics (or denotation) or, put another way: The study of the relationship between <u>inference</u> and <u>entailment</u>.

Why A Universal Language?

- What can you say in Mycin-like rule languages?
  - There are "Contexts" meaning, roughly, objects
  - Rules refer to one of them
  - You can talk about the Values of Attributes of Objects
  - You can compare Object Attributes Values to Numeric Quantities
- What can't you express?
  - Relationships between the values of attributes of one object and the values of attributes of another object.
    - The location of the infectious agent is within the bloodstream of the host.
    - Infectious agent-1 infected host-1 is 2 days earlier than infectious agent-2 infected host-2.
  - Quantified things:
    - Every infections agent in host-1 is a coccus
    - There is an infectious agent in host-1 that is a coccus

Predicate Logic Syntax Solves These Problems

- The unit of representation is the <u>Statement</u> which is the application of a predicate to a set of arguments: John Loves Mary
- Building blocks:
  Constant Symbols, Variable Symbols, Function Symbols, Predicate Symbols
- A Term is:
  A Constant symbol: John
  A variable symbol: ?x
  The Application of a function symbol to set of terms: (Brother (Cousin John)))
- A basic Statement is:
  The application of a predicate symbol to a set of terms.
- Compound statements are formed by connecting Basic Statements using:
  Boolean Connectives: And, Or, Not, Implies
  Quantifiers: Forall, Thereis

Syntax Examples

(Loves Bob Mary)	Bob Loves Mary
(Loves (Father Bob) (Mother Judy))	The Father of Bob Loves the mother of Judy
(Implies (exercises Bob) (Healthy Bob))	If Bob exercises Bob is Healthy
(Forall (?y) (Thereis (?x) (Loves ?x ?y)))	Everybody has somebody who loves them
(Thereis (?y) (Forall (?x) (Loves ?x ?y))	There is somebody whom everybody loves

```
(Forall (?block ?robot ?t-1)
  (implies (and (cleartop ?block ?t-1)
                (handempty ?robot ?t-1))
           (and (possible-successor ?t-2 ?t-1)
                (results-from ?t-2 (pickup ?robot ?block ?t-1))
                (holding ?robot ?block ?t-2))))
```

at any time that the robot's hand is empty and there is nothing on top of the block, the robot can pick up the block and it will be holding the block in the resulting situation.

An Inference System

- A precise notion of **"Follows From"**
- Deduction Rules
  - 2 for each connective: An introduction rule and an elimination rule
  - And Elimination: From **(And A B)** you can deduce **A**
  - Modus Ponens:
    - From **(IMPLIES P Q)** and **P** you can deduce **Q**
  - Universal Instantiation:
    - **(FORALL (X) (P X))** you can deduce **(P A)** for any **A**
- Axioms: Statements that are given as *a priori* true
- A Proof is:
  A Sequence of statements, such that each element is either:
  An Axiom
  An Assumption warranted by a proof rule
  Or the results of applying a deduction rule to previous statements
- Theorem: Any conclusion of a proof.
  - Theorems are statements we're forced to believe if we believe the axioms

Natural Deduction Rules

Propositional Rules

And Introduction
i A
j B
k (And A B) (AI i j)

And Elimination
i (And A B)
j A (AE i)

Conditional Proof
i (Show (Implies P Q))
i+1 | P (assumption i)
j | Q
k (Implies P Q) (CP (j) (i+1))

Modus Ponens
i (Implies A B)
j A
k B (MP i j)

A statement can be copied from outside a box to inside the box, but not from the inside to the outside. Exactly like the lexical scoping rules of a programming language.

Natural Deduction Rules (2)

Or Introduction
i A
j (Or A B) (OI i)

Or Elimination
i (Or A B)
j (Show C)
k | A (assumption j)
l | C
m | B (assumption j)
n | C
o C (OE (i l n) (k m))

Double Negation Elimination
i (not (not A))
j A (DNE i)

Negation Introduction
i Show (Not A)
i+1 | A (assumption)
k | B
l | (not B)
m (not A) (NI (k l) (i+1))

Quantifier Rules

A Substitution of **a** for **x** in the statement (P x) is written [a/x](P x).
It means that every free occurrence of **x** is replaced by **a** in the statement (P x).
The substitution is only valid if no occurrence of **a** is "captured" (i.e. whenever **a** replaces a free occurrence of **x**, **a** should also be free in that context).

For Example:
[a/x](Forall (y)(P x y)) = (Forall (y) (P a y))

but [y/x](Forall (y)(P x y)) is not a valid substitution
Because this would give (Forall (y) (P y y)) "capturing" the substituted in y

Universal Instantiation
i (Forall (x) (P ...))
j [a/x](P ...) (UI i)

where a is any term at all
as long as [a/x] is a valid substitution.
variable

Universal Generalization
i (P ... x ...)
j (Forall (x) (P ... x ...)) (UG i)

where the sentence in line i contains no free variables introduced by existential instantiation and the x does not occur free in assumption within whose scope line i lies.

Quantifier Rules (2)

Existential Instantiation

i (Thereis (x) (P ...))
j [z/x](P ...) (EI i)
where z is a brand new variable

Existential Generalization

i (P ...)
j (Thereis (x)[x/a](P ...)) (EG i)
where a is any term at all

Alphabetic Variance
i (Q (x)(P ...))
j (Q (z)[z/x](P ...)) (AV i)
where Q is either quantifier and [z/x] is a valid substitution.

Derived Rules

Indirect Proof
i (Show A)
i+1 | (not A)
J | B
k | (not B)
l A (IP (j k) (i+1))

GIGO
i (not A)
j A
k C (GIGO i j)

Cut
i (Or A B)
j (not A)
k B (Cut i j)

Modus Tolens
i (Implies A B)
j (not B)
k (not A) (MT i j)

DeMorgan's Rules
(not (or A B)) = (and (not A) (not B))
(not (And A B)) = (or (not A) (not B)

Quantifier Negations
(not (For-all (x) P)) = (There-is (x) (not P))
(not (There-is (x) P)) = (For-all (x) (not P))

An Example Proof

1. Show (Implies (And P (Or Q R)) (Or (And P Q) (And P R)))	
2. | (And P (Or Q R))	Assumption motivated by 1
3. | P	And Elimination 2
4. | (Or Q R)	And Elimination 2
5. | Show (Implies Q (Or (And P Q) (And P R)))	trying for Or Elimination from 4
6. | | Q	Assumption motivated by 5
7. | | (And P Q)	And Introduction 3,6
8. | | (Or (And P Q) (And P R))	Or Introduction 7
9. | (Implies Q (Or (And P Q) (And P R)))	Conditional Proof 8 (6)
10. | Show (Implies R (Or (And P Q) (And P R)))	trying for Or Elimination from 4
11. | | R	Assumption motivated by 10
12. | | (And P R)	And Introduction 3,11
13. | (Implies R (Or (And P Q) (And P R)))	Conditional Proof 13 (11)
14. | (Or (And P Q) (And P R))	Or Elimination 4,9,13
15. (Implies (And P (Or Q R)) (Or (And P Q) (And P R)))	Conditional Proof 14 (2)

Another Proof

1. Show (Implies (Thereis (?y) (Forall (?x) (P ?x ?y)))
(Forall (?x) (Thereis (?y) (P ?x ?y))))
2. | (Thereis (?y) (Forall (?x) (P ?x ?y))) Assumption motivated by 1
3. | (Forall (?x) (P ?x a)) Existential Instantiation 2
4. | (P b a) Universal Instantiation 3
5. | (Thereis (?w) (P b ?w)) Existential Generalization 4
6. | (Thereis (?y) (P b ?y)) Alphabetic Variation 5
7. | (Forall (?z) (Thereis (?y) (P?z ?y))) Universal Generalization 6
8. | (Forall (?x) (Thereis (?y) (P ?x ?y))) Alphabetic Variation 7
9. | (Implies (Thereis (?y) (Forall (?x) (P ?x ?y)))
(Forall (?x) (Thereis (?y) (P ?x ?y)))) Conditional Proof 8 (2)

Bogus Proof

1. Show (Implies (Forall (?x) (Thereis (?y) (P ?x ?y)))
(Thereis (?y) (Forall (?x) (P ?x ?y))))

2. | (Forall (?x) (Thereis (?y) (P ?x ?y))) Assumption motivated by 1
3. | (Thereis (?y) (P b ?y)) Universal Instantiation 2
4. | (P b a) Existential Instantiation 3
5. | (Forall (?w) (P ?w a)) Universal Generalization 4
6. | (Forall (?x) (P ?x a)) Alphabetic Variation 5
7. | (Thereis (?z) (Forall (?x) (P ?x ?z))) Existential Generalization 6
8. | (Thereis (?y) (Forall (?x) (P ?x ?y))) Alphabetic Variation 7
9. (Implies (Forall (?x) (Thereis (?y) (P ?x ?y)))
(Thereis (?y) (Forall (?x) (P ?x ?y)))) Conditional Proof 8 (2)

Where is the bogon?

Normal Form

Notation: (Q (x) ...) means either (For-all (x) ...) or (There-is (x) ...)
(F .. x ..) or (H .. x ..) is some expression including a free occurence of x
G is some expression containing no free occurence of the variables of interest.

Quantifier Rules
(1) (Or (Q (x) (F .. x ..)) G) = (Q (x) (Or (F .. x ..) G))
(2) (And (Q (x) (F .. x ..)) G) = (Q (x) (And (F .. x ..) G))
(3) (And (For-all (x) (F .. x ..)) (For-all (x) (H .. x ..))) = (For-all (x) (And (F .. x ..) (H .. x ..)))
(4) (Or (There-is (x) (F .. x ..)) (There-is (x) (H .. x ..))) = (There-is (x) (Or (F .. x ..) (H .. x ..)))
(5) (Or (Q1 (x) (F .. x ..)) (Q2 (x) (H .. x ..))) = (Q1 (x) (Q2 (z) (Or (F .. x ..) (H .. z ..))))
(6) (And (Q1 (x) (F .. x ..)) (Q2 (x) (H .. x ..))) = (Q1 (x) (Q2 (z) (And (F .. x ..) (H .. z ..))))

Negation Rules
(1) (Not (Not A)) = A
(2) (Not (There-is (x) (F .. x ..))) = (For-all (x) (Not (F .. x ..)))
(3) (Not (For-all (x) (F .. x ..))) = (There-is (x) (Not (F .. x ..)))

DeMorgan's Laws
(1) (Not (And A B)) = (Or (Not A) (Not B))
(2) (Not (Or A B)) = (And (Not A) (Not B))

Distribution Laws

(1) (Or (And A B) (And C D)) = (And (Or A C) (Or A D) (Or B C) (Or B D))
(2) (And (Or A B) (Or C D)) = (Or (And A C) (And A D) (And B C) (And B D))

Implication
(Implies A B) = (Or (Not A) B)

Normal Form (2)

Order of Use of Identities:

1. Implication
2. Negations and deMorgan
3. Quantifers
4. Distribution

The end result is (Q (x) (Q (y) … (And (0r S1 S2 S2) (Or (S3 S4 ….)))
The S's are either positive or negative simple statements
The outermost Q's are all Forall and the inner Q's are Thereis.

Drop all the Thereis Q's replacing their variables by functions of the universally quantified variables.

Drop all the Forall Q's leaving the variables free.

Get a set of quantifier free disjunctions.

Resolution Rule

(Or P G)
(OR (Not Q) H)

If P and Q can be unified (with unifying substitution S)
then it is legal to infer
(OR G' H')

where G' and H' are the results of substituting Substitution S into G and H (respectively).

Unifying two statements means finding a substitution for the variables in the two statements such that the two statements are identical after the substitution is performed in each statement.

(P ?x ?x)	(P ?x a)	(P ?x a ?x)	(P ?x a ?x)
(P a b)	(P b ?y)	(P b ?y ?y)	(P ?y ?y a)
Not unifiable	(?x => b)	Not unifiable	(?x => a)
	(?y => a)		(?y => a)

Horn Clauses: A Restriction to Rule-like Form

Suppose that after normalization all clauses have at most 1 positive statement

(Or p (not q) (not r) (not s) (not t))
where any of p q r s t may contain with free variables

Then by implication identity this is the same as:

(implies (and q r s t) p)

Or rewriting as a <u>logical rule</u>:
If and q, r, s, t
Then p

Mechanizing Logical Rules

- Two natural chaining methods:
  - Forward, data driven, antecedent.
  - Backward, goal driven, consequent.
- Examples:

If (and (Parent ?x ?y) (Parent ?y ?z))
Then (Grandparent ?x ?z)

If (and (Grandparent ?x ?y) (Gender ?x male))
Then (Grandfather ?x ?y)

Examples

- Forward Chaining:

Assert (Parent Abe Ike)
Assert (Gender Abe male)
Assert (Parent Ike Jake)
Deduce (Grandparent Abe Jake)
Deduce (Grandfather Abe Jake)

- Backward Chaining:

Same Facts, plus (Father Abe Ishmael).
Goal (Grandfather Abe Jake)
Goal (Grandparent Abe Jake)
Goal (Parent Abe ?y) matches Ishmael, Ike
Goal (Parent Ishmael ?y) no matches, fail backup
Goal (Parent Ike ?y) matches Jake, rule succeeds
Goal (Gender Abe Male) succeeds, rule succeeds.

Basics of Semantics

- A means for formally assigning "meaning" to syntax.
- More precisely: Semantics is a way of specifying which inferences are sanctioned, i.e. "entailment".
- A Model
  - Domain (a mathematical set)
  - Interpretation, a function that maps:
    - Constant symbols of the syntax to elements of the domain.
    - Function symbols of the syntax to functions over the domain.
    - Predicate symbols of the syntax to predicates over the domain (note that this is a mapping from a set of arguments to true or false).
  - Through the obvious compositions, a statement is mapped to true or false.
  - A Model of a set of axioms is a model that maps all the axioms to True.

Validity and Provability

- Consider all models of a set of Axioms
  - Consider those statements that are mapped into true in all these models
  - These are called the Valid Statements.
- Consider the set of all statements provable by your inference system.
  - These are called the Theorems.
- Two different notions!
- We would like these to be the same sets of statements.
- If every valid statement is a theorem
  - The system is called complete
  - You can deduce what's true
- If every theorem is a valid statement
  - The system is called consistent.
  - You can make only sound deductions.
- A logicians day is made when a proof of soundness and completeness is obtained.

What Formal Properties Hold? Decidability

- Is there an algorithm for deciding whether or not something is a theorem?
- The British Museum algorithm:
  - A conceptual way to show that something is a theorem.
  - Generate all possible proofs in order of increasing length.
  - Stop when you get a proof with the intended theorem as its last line.
- What paradigm is this? How good is it?
- However:
  - This doesn't solve the problem of decidability.
  - Church, Turing and others showed that the language is rich enough to encode the workings of a Turing machine.
  - Does this seem contradictory?

What Formal Properties Hold? Completeness and Consistency

- Is the system consistent?
- Yes (Hilbert and Ackerman).
- Is the system complete?
- Yes (Godel).
- What happens if you add some interesting axioms? For example, those for arithmetic (or those for List structure: cons, car, cdr).
  - Godels' incompleteness theorem
  - You can't be both consistent and complete.
  - The language is rich enough to form a statement which says "I have no proof".

The Expressiveness of Logic

- Logic is intended to be a universal and therefore neutral formalism
  - You pick the constants, functions and predicates to represent your ontology.
  - Logic doesn't prevent you from doing what you want.
  - Logic doesn't guide you about how to do this.
- Logic is very expressive
  - Quantification
  - Any kinds of statements you want
- Logic can capture ambiguity and partial information:
  - Existential quantification
  - (Thereis (x) (p x))
  - Disjunction
  - (Or p q)

Using Logic in Practical Applications:

- Weaker logical systems seem to be strong enough in practice to be useful and yet still controllable.
- Logical Rule Languages are examples:
  - A1 & A2 & ... & An --> C1 & ...
  - The A's are called antecedents
  - The B's are called consequents
- Forward Chaining: As facts are asserted, the LHS of all rules are checked for a consistent matching set. If so the RHS is asserted.
- Backward chaining: If you want to prove the RHS, try to prove the antecedents in the LHS.
- Examples:
  - Prolog
  - Kee's rule system
  - ART
  - Joshua

When is Logic the Right Representation?

- Two Case Studies:
  - American Express Authorizer's Assistant
  - British Nationality Act
- Goal in both cases is Policy Distribution: The systematic and reproducible interpretation of the intended meaning of a set of rules or policies.
  - Credit Authorization
  - British Nationality Law
- Given the same data, the same conclusion should result.
- Human wisdom, common sense, and emotion are not useful or desirable.
- The corpus of policies is, in principle, complete and consistent.
- One can think of the policies as the axioms.
- Solving the problem involves soundly deducing a consequence of the set of axioms.

Amex AA Rationale

- Human nature works against the credit authorizer:
  - start-up transient to learn all the policies.
  - a conservative mindset
- The job is high pressure, boring and not particularly well compensated.
- High turnover
- Amex goal was to get uniform behavior in accordance with company policies.
  - The action to take is a logical consequence of the company's authorization rules.
  - Reasonably simple, logical rules worked well for this.
  - Prototype was more complicated than needed.

What are the "Logic Glasses" ?

- The statement is the appropriate unit of modularity.
- Information is correctly captured in the form of single independent statements and single independent inferences.
- Think about what's true and what follows from it. Sound inference is all. Denotation is king.
- Capture the logical statements appropriate to the domain, preserving their declarative spirit.
- Impose your ontology through the choice of predicates and functions.
- The logical inference system gives the declarative statements a procedural content (namely what inference rules they cause to run).
- The policies become active interpreters of the data.
- Control is secondary or to be ignored.

How well does this work in practice?

- Authorizer's assistant:
- Often's people's purchasing patterns are idiosyncratic.
  - Holidays, vacations, etc. need to be understood.
  - These were captured fairly well in the ART rule system.
- Speed of deduction was very important.
  - Control of reasoning was crucial.
  - Getting to the data was a critical success factor.
  - Accessing IBM "databases" was the "make or break" part.
- Has worked enormously well in practice.
  - 50% to 75% of Amex credit requests are automatically handled.
  - Good advice given in the rest.
- British Nationality Act:
  - Received Mindset: logic means correct, error free, "right the first time"
  - But: Exploratory logic is a perfectly reasonable mindset.

How Well Does It Work In Practice?

Nationality Act:

- Negation was very tricky.
- Counterfactuals and non-monotonicity are problems.
- Choosing the right predicates (i.e., picking the ontology) was hard and experimental in character.
- Logic provided no guidance for how to do this.