Introduction to Artificial Intelligence

Masters DSC 1
MLDM 1

Elisa Fromont

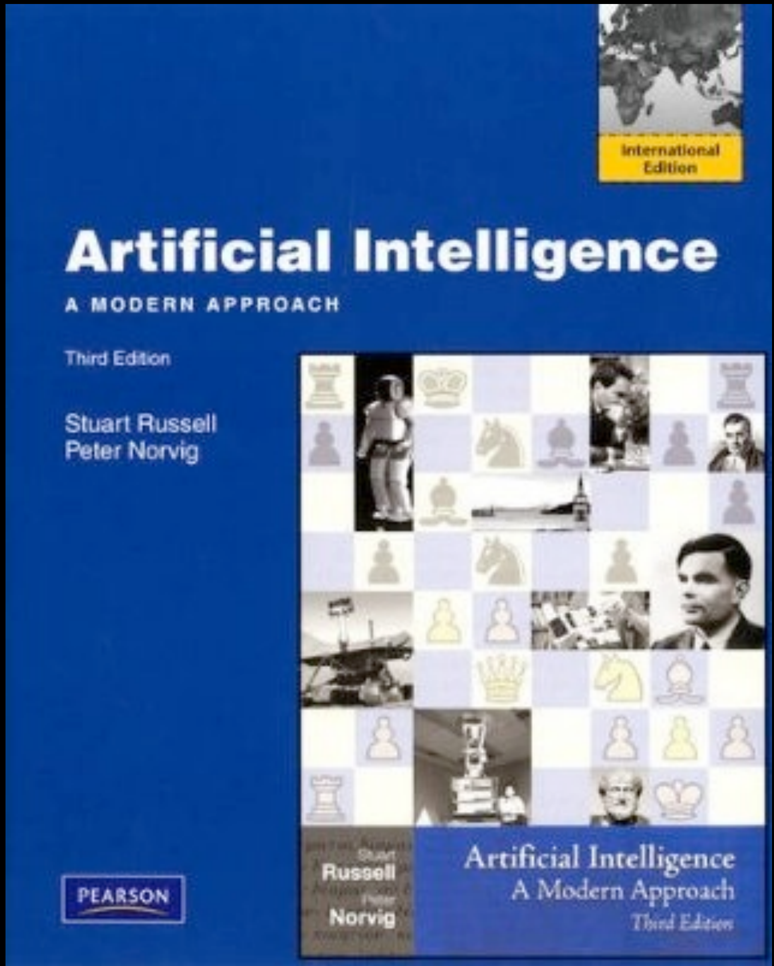
Who am I?

- Associate professor ("Maître de Conférences"), HDR at Jean Monnet University, Saint-Etienne, France. I work at the Hubert-Curien Lab in the Machine Learning team.

- Research domain (AI)
  - Machine Learning/Data Mining applied to computer vision
  - Data Mining for fraud and anomaly detection
  - Inductive databases
  - Inductive Logic Programming (ML+ logic)

- How to reach me ?
  - Mail : elisa.fromont@univ-st-etienne.fr
  - At the Hubert-Curien lab: office E103a
  - And during the lectures !

THE book



- Artificial Intelligence – A Modern Approach
- By Stuart Russel and Peter Norvig

Outline

1. Reasoning in logic (11h)
   1. Propositional logic (quick review)
   2. First-order logic (review)
      1. Properties, relations, functions, quantifiers, …
      2. Terms, sentences, wffs, axioms, theories, proofs, …
   3. FS Resolution

2. Intro to AI (13h)
3. Prolog Programming (12h)
4. Constraint Logic Programming LP (4h)
5. Practical sessions in Prolog and CLP (10h inverse classrooms)

\+ PROJECT

Project

- 2-player game in Prolog (me and an AI)
- Chose your own two player game but:
  - AI must need a search algorithm to play
  - AI must need heuristics to play better
  - State of games should be easily indentifiable

  (find ideas on boardgamegeek (look for 2 players abstract games) and in game shops)

- Group of 3 to 4 students
- Defense in January (15 minutes pres and 15 minutes questions)

REASONING IN LOGIC (PART 1 OF THE COURSE)

Why study logic?

- To look smarter while speaking… ☺
  - “It’s because I took the elevator that my feet hurt”

  (My son, 2 years old)

- For the second (more practical) part of the course
- To learn the great PROLOG programming language
- For knowledge representation and automatic reasoning in general
- For the Semantic Web and Multi Agent System courses (M2 for WI and M1 & M2 for MLDM)

Propositional Logic: review

Example

A man accused of a crime, hired an attorney whose statements were always admitted by the court as undisputable truth. The following exchange took place in court.
Prosecutor: "If the accused committed the crime, he had an accomplice.”
Lawyer: "That is not true!"

Did the lawyer help his client?

Propositional logic

- Logical constants: true, false
- Propositional symbols: P, Q, S, ... (atomic sentences)
- Wrapping parentheses: (…)
- Sentences are combined by connectives:

$\wedge$ ...and	[conjunction]
$\vee$ ...or	[disjunction]
$\Rightarrow$...implies	[implication / conditional]
$\Leftrightarrow$..is equivalent	[biconditional]
$\neg$ ...not	[negation]

- Literal: atomic sentence or negated atomic sentence

Examples of PL sentences

- $(P \wedge Q) \rightarrow R$
  "If it is hot and humid, then it is raining"
- $Q \rightarrow P$
  "If it is humid, then it is hot"
- Q
  "It is humid."
- A better way:
  Ho = "It is hot"
  Hu = "It is humid"
  R = "It is raining"

Propositional logic (PL)

- A simple language useful for showing key ideas and definitions
- User defines a set of propositional symbols, like P and Q.
- User defines the **semantics** of each propositional symbol:
  - P means "It is hot"
  - Q means "It is humid"
  - R means "It is raining"
- A sentence (well formed formula) is defined as follows:
  - A symbol is a sentence
  - If S is a sentence, then $\neg$S is a sentence
  - If S is a sentence, then (S) is a sentence
  - If S and T are sentences, then (S $\wedge$ T), (S $\vee$ T), (S => T), and (S $\leftrightarrow$ T) are sentences
  - A sentence results from a finite number of applications of the above rules

A BNF* grammar of sentences in propositional logic

```
S := <Sentence**> ;
<Sentence> := <AtomicSentence> | <ComplexSentence> ;
<AtomicSentence> := "TRUE" | "FALSE" |
                    "P" | "Q" | "S" ;
<ComplexSentence> := "(" <Sentence> ")" |
          <Sentence> <Connective> <Sentence> |
          "NOT" <Sentence> ;
<Connective> := "AND" | "OR" | "IMPLIES" | "EQUIVALENT" ;
```

**Backus-Naur Form*

*** the word "sentence" can be replaced by "proposition" everywhere. A "complex sentence" is also a "compound proposition".*

Some terms

- The meaning or semantics of a sentence determines its interpretation.
- Given the truth values of all symbols in a sentence, it can be "evaluated" to determine its truth value (True or False).
- A model for a KB is a "possible world" (assignment of truth values to propositional symbols) in which each sentence in the KB is True.

More terms

- A valid sentence or tautology is a sentence that is True under all interpretations, no matter what the world is actually like or what the semantics is. Example: “It’s raining or it’s not raining.”
- An inconsistent sentence or contradiction is a sentence that is False under all interpretations. The world is never like what it describes, as in “It’s raining and it’s not raining.”
- **P entails Q**, written P |= Q, means that whenever P is True, so is Q. In other words, all models of P are also models of Q.

Valuation in PL

- Evaluate all the sentences
- Evaluate all the complex sentences using for example the truth tables

- Ex : $A = \neg a \vee (b \Rightarrow (\ (a \wedge b) \Rightarrow \neg b))$

Truth tables

Val(A)	Val(B)	Val($A \wedge B$)	Val($A \vee B$)	Val($A \oplus B$)	Val($A \Rightarrow B$)	Val($A \Leftrightarrow B$)
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	0	0
1	1	1	1	0	1	1

Val(A)	Val($\neg$ A)
0	1
1	0

Evaluation of complex sentences

- Let A be a complex sentence with n atomic sentences
- We create a table with n columns et 2^n lines
- We open new columns for each sub-sentences in the complex sentence until we get back to A.

- Ex :

a	b	$\neg a$	$\neg b$	$a \wedge b$	$(a \wedge b) \Rightarrow \neg b$	$b \Rightarrow ((a \wedge b) \Rightarrow \neg b)$	A

Definitions

1. Two sentences A, B are equivalent iff for all possible valuation $v(A) = v(B)$. It is written $A \approx B$

2. Two sentences A, B are equivalent iff they have the same truth table!

3. A compound proposition is satisfiable if there is at least one truth assignment to the variables that makes the proposition true. In other words iff there exists a valuation v such that $v(A) = 1$.

Well-known equivalences 1

1. $\neg(\neg A) \approx A$
2. $(A \vee B) \vee C \approx A \vee (B \vee C)$
3. $(A \wedge B) \wedge C \approx A \wedge (B \wedge C)$

   associativity

4. $A \vee B \approx B \vee A$
5. $A \wedge B \approx B \wedge A$

   commutativity

6. $A \vee A \approx A$
7. $A \wedge A \approx A$

   idempotence

8. $A \vee (B \wedge C) \approx (A \vee C) \wedge (A \vee B)$
9. $A \wedge (B \vee C) \approx (A \wedge C) \vee (A \wedge B)$

   distributivity

10. $A \wedge (A \vee B) \approx A$
11. $A \vee (A \wedge B) \approx A$

    absorption

Well-known equivalences 2

1. $\neg (A \wedge B) \approx \neg A \vee \neg B$
2. $\neg (A \vee B) \approx \neg A \wedge \neg B$

De Morgan's laws

3. $A \Rightarrow B \approx \neg A \vee B$ **
4. $A \Leftrightarrow B \approx (\neg A \vee B) \wedge (\neg B \vee A)$
5. $A \oplus B \approx \neg(A \Leftrightarrow B)$

6. $(\neg A \vee A) \approx$ true (tautology)
7. $(\neg A \wedge A) \approx$ false (contradiction)

Remarks

- Whatever B, if A is false, A$\Rightarrow$B is true.
- Whatever A, if B is true, A$\Rightarrow$B is true.

- “if you are hungry, there is some chicken in the fridge”: what is the *contrapositive* ?

- You can win a new car or a 1000 euros cheque.
  - Ok, give me both !

Exercise 1

1. Write the sentences of the prosecutor and defender of Slide 8 in propositional logic. Can you answer the question of S8?
2. Show that the following sentences are tautologies:

   (sometimes using equivalences, sometimes using tables)

   1. $r \Rightarrow (s \Rightarrow r)$
   2. $(r \Rightarrow s) \Rightarrow ((s \Rightarrow t) \Rightarrow (r \Rightarrow t))$
   3. $r \Rightarrow (\neg r \Rightarrow s)$
   4. $(r \vee s) \Leftrightarrow ((r \Rightarrow s) \Rightarrow s)$

Solution for 2.2

- $(r \Rightarrow s) \Rightarrow ((s \Rightarrow t) \Rightarrow (r \Rightarrow t))$

$\approx (r \Rightarrow s) \Rightarrow (\neg(s \Rightarrow t) \vee (r \Rightarrow t))$

$\approx \neg (r \Rightarrow s) \vee (\neg(s \Rightarrow t) \vee (r \Rightarrow t))$

$\approx \neg (r \Rightarrow s) \vee (\neg(\neg s \vee t) \vee (\neg r \vee t))$

$\approx \neg (r \Rightarrow s) \vee ((s \wedge \neg t) \vee (\neg r \vee t))$

$\approx \neg (r \Rightarrow s) \vee ((\neg r \vee t \vee \neg t) \wedge (s \vee \neg r \vee t))$

$\approx \neg (r \Rightarrow s) \vee (\text{true} \wedge (s \vee \neg r \vee t))$

$\approx \neg (r \Rightarrow s) \vee ((s \vee \neg r \vee t))$

$\approx (r \wedge \neg s) \vee ((s \vee \neg r \vee t))$

$\approx (r \vee s \vee \neg r \vee t) \wedge (\neg s \vee s \vee \neg r \vee t)$

$\approx$ true (tautology)

Definitions…

- Definition: **A literal** is either an atom or a negation of an atom.
- Let ϕ = :(A $\wedge\neg$B). Then:
  - Atoms: atom(ϕ) = {A,B}
  - Literals: lit(ϕ) = {A, $\neg$B}

- Equivalent formulas can have different literals
  - $\phi = \neg(A \wedge \neg B) = \neg A \vee B$
  - Now lit(ϕ) = {$\neg$A, B}

Definitions…

- Definition: a term is a conjunction of literals
  - Example: $(A \wedge \neg B \wedge C)$

- Definition: a clause is a disjunction of literals
  - Example: $(A \vee \neg B \vee C)$
  - “ $\square$ “ is the empty clause such that
  - $\forall$ v, $v(\square) = 0$ (contradiction)

Negation Normal Form (NNF)

- Definition ¬A formula is said to be in Negation Normal Form (NNF) if it only contains $\neg$, $\vee$ and $\wedge$ connectives and only atoms can be negated.

- Examples:
  - $\phi_1 = \neg(A \vee \neg B)$ is not in NNF
  - $\phi_2 = \neg A \wedge B$ is in NNF

Converting to NNF

- Every formula can be converted to NNF in linear time:
  - Eliminate all connectives other than $\neg$, $\vee$ and $\wedge$
  - Use De Morgan and double-negation rules to push negations to the right
- Example: $\phi = \neg(A \rightarrow \neg B)$
  - Eliminate '$\rightarrow$': $\phi = \neg(\neg A \vee \neg B)$
  - Push negation using De Morgan: $\phi = (\neg\neg A \wedge \neg\neg B)$
  - Use Double negation rule: $\phi = (A \wedge B)$

Disjunctive Normal Form (DNF)

- Definition: A formula is said to be in Disjunctive Normal Form (DNF) if it is a disjunction of terms.
  - In other words, it is a formula of the form $\bigvee_i \left(\bigwedge_j l_{i,j} \right)$

    where $l_{i,j}$ is the j-th literal in the i-th term.

- Examples
  - $\phi = (A \land \neg B \land C) \lor (\neg A \land D) \lor (B)$ is in DNF

- DNF is a special case of NNF

Converting to DNF

- Every formula can be converted to DNF in exponential time and space:
  - Convert to NNF
  - Distribute disjunctions following the rule:
    $A \land (B \lor C) \approx ((A \land B) \lor (A \land C))$

- Example:
  - $\phi = (A \lor B) \land (\neg C \lor D) =$
    $((A \lor B) \land (\neg C)) \lor ((A \lor B) \land D) =$
    $(A \land \neg C) \lor (B \land \neg C) \lor (A \land D) \lor (B \land D)$

Conjunctive Normal Form or clausal form (CNF)

- Definition: A formula is said to be in Conjunctive Normal Form (CNF) if it is a conjunction of clauses.
  - In other words, it is a formula of the form $\bigwedge_i (\bigvee_j l_{i,j})$

    where $l_{i,j}$ is the j-th literal in the i-th term.
- Examples
  - $\phi = (A \vee \neg B \vee C) \wedge (\neg A \vee D) \wedge (B)$ is in CNF
- CNF is a special case of NNF
- NB : if all the variables appear in all the clauses, the CNF is in the canonical conjunctive normal form.

NB: Converting to CNF

- Every formula can be converted to CNF:
  - in exponential time and space with the same set of atoms
  - in linear time and space if new variables are added.
    - In this case the original and converted formulas are "equi-satisfiable".
    - This technique is called Tseitin's encoding.

Exercise 2

A safe with n locks can be open if the **n** locks are simultaneously opened. Five persons : a, b, c, d, e must be given keys corresponding to some of these locks. There are multiple exemplars of each keys.

How would you compute the smallest n such that each of the 5 persons is given a set of keys so that the safe can only be opened in the 3 following configurations:

- a and b are present together
- a, c et d are present together
- b, d et e are present together

Hint: use the equivalence between DNF and CNF

Exercise 3

$G = (A \Rightarrow ((B \wedge \neg A) \vee (\neg C \wedge A)))$

1. What is the truth table of this formula?
2. Deduce its canonical DNF
3. Deduce its canonical CNF

Where are we now?

We saw:

- The syntax of the propositional logic (P)
- How to evaluate a proposition/sentence

Now we need to learn:

- How to use the propositions in reasoning (= prove things)
- How to know if a reasoning is valid (= true under all interpretations)

Semantic deduction theorem

Let $S \neq \emptyset$ a set of sentences (KB) (S,B and $A \in$ LProp)

$$S \cup \{A\} \models B \quad \Leftrightarrow \quad S \models (A \Rightarrow B)$$

Proof:

Note that if $S \models A$ and $S \models (A \Rightarrow B)$ then $S \models B$

- <= (easy): $S \cup \{A\} \supset S$ so if $S \models (A \Rightarrow B)$ then $S \cup \{A\} \models (A \Rightarrow B)$.

Or $S \cup \{A\} \models A$ so (using the note) $S \cup \{A\} \models B$

- => (less easy by induction)

http://www.personal.kent.edu/~rmuhamma/Philosophy/Logic/Deduction/4-deductionTheorem.htm

Corollaries

Corollary 1 : (use of tautologies) :

$H_1, H_2, \ldots, H_n \models C \quad \Leftrightarrow \quad \varnothing \models H_1 \Rightarrow (H_2 \Rightarrow (\ldots(H_n \Rightarrow C))$

$\Leftrightarrow \varnothing \models (H1 \wedge H2 \wedge \ldots \wedge Hn) \Rightarrow C$

i.e $(H1 \wedge H2 \wedge \ldots \wedge Hn) \Rightarrow C$ is a *tautology* or a *theorem*

Corollary 2 : (reductio ad absurdum/proof by contradiction) :

$B \cup \{\neg A\} \models \text{false}$ iff $B \models ((\neg A) \Rightarrow \text{false})$ iff $(\approx)$ $B \models A$

(to deduce A from B can be done by proving that $B \cup \{\neg A\}$ is a contradiction)

Exercise 4

1. KB = { If I am clever then I will pass,
   If I will pass then I am clever,
   I am clever or I will pass }

Conclusion = I am clever and I will pass.
Is this reasoning valid?

2. Let the propositional sentences (KB):
   1. A : $(r \land g) \Rightarrow \neg q$
   2. B : $\neg b \Rightarrow (q \lor g)$
   3. C : $(c \land \neg r) \Rightarrow (g \land q)$
   4. D : $(g \lor q) \land \neg c \Rightarrow r \lor b$

   Show that we do not have {A, B, C, D} I= $((g \land \neg c) \Rightarrow b)$

Formal Systems (FS)

Definition: well-defined system of abstract thought based on the model of mathematic

Machine that can produce tautologies

Aim: study the reasoning process

Inference rules

Logical inference is used to create new sentences that logically follow from a given set of predicate calculus sentences (KB).

Soundness: **If KB |- Q then KB |= Q**

- If Q is derived from a set of sentences KB using a given set of rules of inference, then Q is entailed by KB.
- Hence, inference produces only real entailments, or any sentence that follows deductively from the premises is valid.

Completeness: **If KB |= Q then KB |- Q**

- If Q is entailed by a set of sentences KB, then Q can be derived from KB using the rules of inference.
- Hence, inference produces all entailments, or all valid sentences can be proved from the premises.

Sound rules of inference

- Here are some examples of sound rules of inference
  – *A rule is sound if its conclusion is true whenever the premise is true*
- Each can be shown to be sound using a truth table

RULE	PREMISE	CONCLUSION
Modus Ponens	$A, A \rightarrow B$	B
And Introduction	A, B	$A \wedge B$
And Elimination	$A \wedge B$	A
Double Negation	$\neg\neg A$	A
Unit Resolution	$A \vee B, \neg B$	A
Resolution	$A \vee B, \neg B \vee C$	$A \vee C$

Proving things

- A **proof** is a sequence of sentences, where each sentence is either a premise or a sentence derived from earlier sentences in the proof by one of the rules of inference.
- The last sentence is the **theorem** (also called goal or query) that we want to prove.
- Example for the "weather problem" given before.

1 Hu	Premise	"It is humid"
2 $Hu \rightarrow Ho$	Premise	"If it is humid, it is hot"
3 Ho	Modus Ponens(1,2)	"It is hot"
4 $(Ho \wedge Hu) \rightarrow R$	Premise	"If it's hot & humid, it's raining"
5 $Ho \wedge Hu$	And Introduction(1,3)	"It is hot and humid"
6 R	Modus Ponens(4,5)	"It is raining"

Entailment and derivation

- **Entailment: KB |= Q** **(cf. T12)**
  - Q is entailed by KB (a set of premises or assumptions) if and only if there is no logically possible world in which Q is false while all the premises in KB are true.
  - Or, stated positively, Q is entailed by KB if and only if the conclusion is true in every logically possible world (= model) in which all the premises in KB are true.
  - If KB is empty, $\models Q$ means: $\forall v, v(Q) = 1$ or Q is a tautology.
- **Derivation: KB |- Q**
  - We can derive Q from KB if there is a proof consisting of a sequence of valid inference steps starting from the premises in KB and resulting in Q

Horn sentences

- A **Horn sentence** or **Horn clause** has the form:

  $P1 \wedge P2 \wedge P3 \ldots \wedge Pn \rightarrow Q$

or alternatively

  $\neg P1 \vee \neg P2 \vee \neg P3 \ldots \vee \neg Pn \vee Q$

where Ps and Q are non-negated atoms

- To get a proof for Horn sentences, apply Modus Ponens repeatedly until nothing can be done

- We will use the Horn clause form later

Requirements of a Formal System

1. An alphabet of symbols
2. A set of finite strings of these symbols, the well-formed formulas (wffs).
3. Axioms (base), the definitions of the system.
4. Inference rules, which enable a wff to be deduced as the conclusion of a finite set of other wffs: axioms or other theorems of the logic system.
5. Completeness: every wff can either be proved or refuted.
6. The system must be sound: every theorem is a logically valid wff.

Resolution: how we do proofs in CNF

Converting predicate logic statements to CNF is very convenient, because it simplifies the process of trying to prove things automatically.

With fewer connectives and no quantifiers to worry about, there are fewer applicable equivalences or inference rules that can be used. In fact, we can do *a great deal* of automated proof work by using just **one** inference rule. This is called **Resolution**, and it is usually used in conjunction with Reasoning by Contradiction.

The **Resolution** inference rule is basically:
If, in CNF form, one clause contains X and another clause contains $\neg X$, then we can infer the union of those clauses, without the X and $\neg X$

FS Resolution for propositional logic (JA Robinson (60's))

- Language: C (clauses)
- Induction scheme:
  - Base: empty !!
  - Inference rule:

    a ∨ A, ¬a ∨ B |-- A∨B ← a : literal,

    A∨B : resolvent

Suppose that "a ∨ A, ¬a ∨ B" have been produced, FSR can produce A∨B »

Example of proof

Prove (in FSR) : $a \vee b \vee c \quad \neg a \vee b \vee c \quad \neg b \vee c \;\vdash c$

(the proof is written as a tree)

H: $a \vee b \vee c \qquad \neg a \vee b \vee c \qquad \neg b \vee c$

$b \vee c$

c

Is this $a \vee b \vee c \quad \neg a \vee b \vee c \quad \neg b \vee c \;\models c$ valid? (yes for proposition)

Use FSR for propositions

Is H |= c (H et c $\in$ P) valid?

1. Start from H $\cup$ {$\neg$ c} (deduction theo)
2. Create G $\approx$ H $\cup$ {$\neg$ c} (G $\in$ C) (where all formulas are in clausal form)
3. Produce $\square$ from G

To prove that a set is satisfiable (the contrary), you must prove that you cannot produce $\square$

Exercise 5

5.1
G1 = {$a \vee \neg b \vee c, \neg a \vee c, \neg e, a \vee b \vee e, \neg c \vee e$}
Show that G is contradictory

5.2
G2 = {$a \vee b \vee c, \neg c, \neg d, \neg a \vee \neg b \vee d$}
Show that G is satisfiable

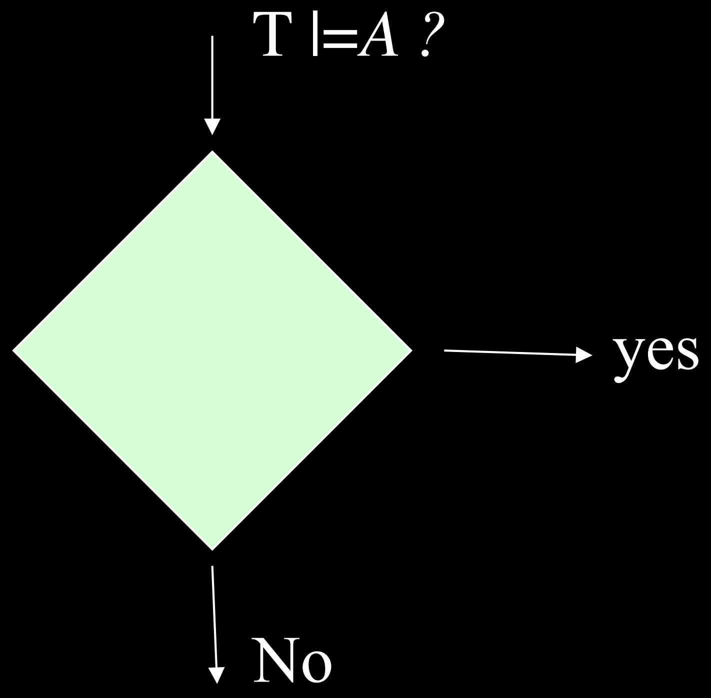
5.3

$$\left.\begin{array}{ll} 1. & (l \wedge t) \Rightarrow (m \vee w) \\ 2. & w \Rightarrow ((m \wedge \neg t) \vee \neg l) \\ 3. & (m \wedge t) \Rightarrow w \end{array}\right\} \quad l \Rightarrow \neg t$$

Show (with FSR) that the conclusion can be derived from 1,2 and 3

Decidability

- A formal logic is decidable if there is a mechanical (algorithmic) way of determining whether *A* follows from *T.*



Propositional logic is decidable: at worst we need only consider every row of the truth-table for which *T* is true, and there are only finitely many.

Problems with Propositional Logic

Propositional logic is a weak language

- Hard to identify “individuals” (e.g., Mary, 3)
- Can’t directly talk about properties of individuals or relations between individuals (e.g., “Bill is tall”)
- Generalizations, patterns, regularities can’t easily be represented (e.g., “all triangles have 3 sides”)
- First-Order Logic (abbreviated FOL or FOPC) is expressive enough to concisely represent this kind of information

FOL adds relations, variables, and quantifiers, e.g.,

- *“Every elephant is gray”:* ∀ x (elephant(x) → gray(x))
- *“There is a white alligator”:* ∃ x (alligator(X) ∧ white(X))

Example

- Consider the problem of representing the following information:
  - Every person is mortal.
  - Confucius is a person.
  - Confucius is mortal.
- How can these sentences be represented so that we can infer the third sentence from the first two?

Example II

- In PL we have to create propositional symbols to stand for all or part of each sentence. For example, we might have:
  P = “person”; Q = “mortal”; R = “Confucius”
- so the above 3 sentences are represented as:
  $P \rightarrow Q$; $R \rightarrow P$; $R \rightarrow Q$
- Although the third sentence is entailed by the first two, we needed an explicit symbol, R, to represent an individual, Confucius, who is a member of the classes “person” and “mortal”
- To represent other individuals we must introduce separate symbols for each one, with some way to represent the fact that all individuals who are “people” are also “mortal”

First-Order Logic (FOL): Review

First-order logic

- First-order logic (FOL) models the world in terms of
  - **Objects**, which are things with individual identities
    - Ex: Students, lectures, companies, cars ...
  - **Properties** of objects that distinguish them from other objects
    - Ex: blue, oval, even, large, ...
  - **Relations** that hold among sets of objects
    - Brother-of, bigger-than, outside, part-of, has-color, occurs-after, owns, visits, precedes, ...
  - **Functions**, which are a subset of relations where there is only one “value” for any given “input”
    - father-of, best-friend, second-half, one-more-than ...

User provides

- **Constant symbols**, which represent individuals in the world
  - mary, 3, green
- **Function symbols**, which map individuals to individuals
  - father-of(mary) = john
  - color-of(sky) = blue
- **Predicate symbols**, which map individuals to truth values
  - greater(5,3), green(grass) , color(grass, green)

FOL Provides

- **Variable symbols**
  - E.g., X, Y, Foo
- **Connectives**
  - Same as in PL: not (¬), and (∧), or (∨), implies (→), if and only if (biconditional ↔)
- **Quantifiers**
  - Universal **∀x** or **(Ax)**
  - Existential **∃x** or **(Ex)**

A BNF for FOL

```
S := <Sentence> ;
<Sentence> := <AtomicSentence> |
        <Sentence> <Connective> <Sentence> |
        <Quantifier> <Variable>,... <Sentence> |
        "NOT" <Sentence> |
        "(" <Sentence> ")";
<AtomicSentence> := <Predicate> "(" <Term>, ... ")" |
                <Term> "=" <Term>;
<Term> := <Function> "(" <Term>, ... ")" |
        <Constant> |
        <Variable>;
<Connective> := "AND" | "OR" | "IMPLIES" | "EQUIVALENT";
<Quantifier> := "EXISTS" | "FORALL" ;
<Constant> := "A" | "X1" | "John" | ... ;
<Variable> := "a" | "x" | "s" | ... ;
<Predicate> := "Before" | "HasColor" | "Raining" | ... ;
<Function> := "Mother" | "LeftLegOf" | ... ;
```

Sentences are built from terms and atoms

- A **term** (denoting a real-world individual) is a constant symbol, a variable symbol, or an n-place function of n terms.
  - x and $f(x_1, ..., x_n)$ are terms, where each x_i is a term.
  - A term with no variables is a **ground term**
- An **atomic sentence** (which has value true or false) is an n-place predicate of n terms
- A **complex sentence** is formed from atomic sentences connected by the logical connectives:
  - $\neg P$, $P \vee Q$, $P \wedge Q$, $P \rightarrow Q$, $P \leftrightarrow Q$ where P and Q are sentences
- A **quantified sentence** adds quantifiers $\forall$ and $\exists$
- A **well-formed formula (wff)** is a sentence containing no “free” variables. That is, all variables are “bound” by universal or existential quantifiers.
  - $(\forall x)P(x,y)$ has x bound as a universally quantified variable, but y is free.

Quantifiers

- Universal quantification
  - $(\forall x)P(x)$ means that P holds for all values of x in the domain associated with that variable
  - E.g., $(\forall x)$ dolphin(x) $\rightarrow$ mammal(x)
- Existential quantification
  - $(\exists x)P(x)$ means that P holds for some value of x in the domain associated with that variable
  - E.g., $(\exists x)$ mammal(x) $\wedge$ lays-eggs(x)
  - Permits one to make a statement about some object without naming it

Quantifiers

- Universal quantifiers are often used with "implies" to form "rules":
  $(\forall x)$ student(x) → smart(x) means "All students are smart"

- Universal quantification is *rarely* used to make blanket statements about every individual in the world:
  $(\forall x)$student(x)∧smart(x) means "Everyone in the world is a student and is smart" (there is no condition on x for the formula to be true)

- Existential quantifiers are usually used with "and" to specify a list of properties about an individual:
  $(\exists x)$ student(x) ∧ smart(x) means "There is a student who is smart"

- A common mistake is to represent this English sentence as the FOL sentence:
  $(\exists x)$ student(x) → smart(x)
  – But what happens when there is a person who is *not* a student?

Switching Quantifiers

- Switching the order of universal quantifiers *does not* change the meaning:
  - $(\forall x)(\forall y)P(x,y) \leftrightarrow (\forall y)(\forall x) P(x,y)$
- Similarly, you can switch the order of existential quantifiers:
  - $(\exists x)(\exists y)P(x,y) \leftrightarrow (\exists y)(\exists x) P(x,y)$
- Switching the order of universals and existentials *does* change meaning:
  - Everyone likes someone: $(\forall x)(\exists y)$ likes(x,y)
  - Someone is liked by everyone: $(\exists y)(\forall x)$ likes(x,y)

Connections between All and Exists

We can relate sentences involving $\forall$ and $\exists$ using De Morgan's laws:

- $(\forall x)\ \neg P(x) \leftrightarrow \neg(\exists x)\ P(x)$
- $\neg(\forall x)\ P \leftrightarrow (\exists x)\ \neg P(x)$
- $(\forall x)\ P(x) \leftrightarrow \neg\ (\exists x)\ \neg P(x)$
- $(\exists x)\ P(x) \leftrightarrow \neg(\forall x)\ \neg P(x)$

Occurrence of a variable

- An occurrence of a variable **x** in a formula **F** is the place where **x** appears in **F** without being immediately preceded by **∀** or **∃**

- An occurrence of x in F is either bound or free.

Open and Closed Formula

- **Scope of a Quantifier:** The scope of ∀x (resp. ∃x) in ∀x F (resp ∃x F) is F
- E.g. In ∀x(Rich(x)→Happy(x)) the scope of ∀x is (Rich(x)→Happy(x))
- In ∀x∃y(equal(mother(x),y)) the scope of ∀x is ∃y(equal(mother(x),y)) and the scope of ∃y is equal(mother(x),y)
- **Free and Bound Occurrence of a Variable:** A bound occurrence of a variable in a formula is an occurrence immediately following a quantifier or an occurrence within the scope of a quantifier, which has the same variable immediately after the quantifier.
- Any other variable is free.

Open and Closed:2

- **Ex**: In the formula $(\exists x P(x,y)) \wedge Q(x)$ the occurrence of x in $P(x,y)$ is bound but the occurrence of x in $Q(x)$ is free.
- All occurrences of x in $\exists x(P(x,y) \wedge Q(x))$ are bound.

- **A closed formula is a well-formed formula with no free occurrences of any variable**. An open formula is a formula that is not closed

- **Example:** $\forall x \exists y(equal(mother(x),y))$ is closed since all occurrences of x fall within the scope of $\forall x$ and all occurrences of y fall within the scope of $\exists y$.

Expressing uniqueness

- Sometimes we want to say that there is a single, unique object that satisfies a certain condition
- "There exists a unique x such that king(x) is true"
  - $\exists x\ king(x) \wedge \forall y\ (king(y) \rightarrow x{=}y)$
  - $\exists x\ king(x) \wedge \neg\exists y\ (king(y) \wedge x{\neq}y)$
  - $\exists!\ x\ king(x)$
- "Every country has exactly one ruler"
  - $\forall c\ country(c) \rightarrow \exists!\ r\ ruler(c,r)$

Translating English to FOL

Every gardener likes the sun.

$\forall x\ gardener(x) \rightarrow likes(x,sun)$

All purple mushrooms are poisonous.

$\forall x\ (mushroom(x) \wedge purple(x)) \rightarrow poisonous(x)$

No monkey is a soldier.

$\forall x\ monkey(x) \rightarrow \neg\ soldier(x)$

No purple mushroom is poisonous.

$\neg \exists x\ purple(x) \wedge mushroom(x) \wedge poisonous(x)$

$\forall x\ \ (mushroom(x) \wedge purple(x)) \rightarrow \neg poisonous(x)$

There are exactly two purple mushrooms.

$\exists x\ \exists y\ mushroom(x) \wedge purple(x) \wedge mushroom(y) \wedge purple(y) \wedge \neg(x=y) \wedge \forall z\ (mushroom(z) \wedge purple(z)) \rightarrow ((x=z) \vee (y=z))$

Clinton is not tall.

$\neg tall(clinton)$

Some lions do not drink coffee

$\exists x\ lion(x) \wedge \neg drink\_coffee(x)$

Y is above X iff Y is directly on top X or there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.

$\forall x\ \forall y\ above(x,y) \leftrightarrow (on(x,y) \vee\ \exists z\ (on(x,z) \wedge above(z,y)))$

Semantics of FOL

- **In FOL, there is no valuation without interpretation !**
- **Domain of interpretation M**: the set of all objects in the world (of interest)
- **Interpretation I**: includes
  - Assign each constant to an object in M
  - Define each function of n arguments as a mapping $M^n \Rightarrow M$
  - Define each predicate of n arguments as a mapping $M^n \Rightarrow \{T, F\}$ (note that a proposition is a predicate for which n= 0)
    - Therefore, every ground predicate with any instantiation will have a truth value
- **In general there is an infinite number of interpretations because |M| is infinite**

Example of interpretations

$$G = \forall x\ (R(x) \Rightarrow T(f(x),a))$$

D = {4,5}

a -> 4

$$f \to \begin{cases} 4 \to 5 \\ 5 \to 4 \end{cases}$$

$$R \to \begin{cases} 4 \to 1 \\ 5 \to 0 \end{cases}$$

$$T \to \begin{cases} (4,4) \to 1 \\ (4,5) \to 1 \\ (5,4) \to 1 \\ (5,5) \to 1 \end{cases}$$

D = {Jeanne, Marc, Zoe, Luc}

a -> Jeanne

f -> brother

R -> is-a-woman

T -> is-older-than

etc.

Evaluate an interpretation

- If G does not have any variables, the evaluation will be the one given by the functions/propositions/predicates
- If G has some variables (ex : P(x, f(b,y))), there will be one valuation per possible value for these variables!

- (∀x) P(x) is true iff P(x) is true under all interpretations
- (∃x) P(x) is true iff P(x) is true under some interpretation

- The logical connectives **~**, **∧**, **v**, **=>**, **<=>** behave as in PL

Definitions

- **Model**: an interpretation of a set of sentences such that every sentence is *True*
- **A sentence/formula is**
  - **satisfiable** if it is true under some interpretation
  - **valid** if it is true under all possible interpretations
  - **inconsistent** if there does not exist any interpretation under which the sentence is true
- **Logical consequence**: S |= X if all models of S are also models of X

Example

- Find possible models for $\forall x\ (P(x) \vee L(x))$

D = vehicles with engines	D = $\mathbb{N}$

Exercise 6

A1 : $\forall x \, \forall y \, \forall z \, ((P(x,y) \wedge P(y,z)) \Rightarrow P(x,z))$

A2 : $\forall x \, (P(a,x) \vee P(x,b))$

A3 : $\forall x \, (P(x,f(x)))$

1. Find two models of {A1,A2,A3}
2. Show that {A1,A2,A3} $\models \exists x \, P(x,a)$ is not valid?

Axioms, definitions and theorems

- **Axioms** are facts and rules that attempt to capture all of the (important) facts and concepts about a domain; axioms can be used to prove **theorems**
  - –Mathematicians don't want any unnecessary (dependent) axioms –ones that can be derived from other axioms
  - –Dependent axioms can make reasoning faster, however
  - –Choosing a good set of axioms for a domain is a kind of design problem
- A **definition** of a predicate is of the form "p(X) $\leftrightarrow$ …" and can be decomposed into two parts
  - –**Necessary** description: "p(x) $\rightarrow$ …"
  - –**Sufficient** description "p(x) $\leftarrow$ …"
  - –Some concepts don't have complete definitions (e.g., person(x))

Example

- Define father(x, y) by parent(x, y) and male(x)
  - parent(x, y) is a necessary (**but not sufficient**) description of father(x, y)
    - father(x, y) → parent(x, y)
  - parent(x, y) ∧ male(x) ∧ age(x, 35) is a **sufficient** (**but not necessary**) description of father(x, y):
    - father(x, y) ← parent(x, y) ∧ male(x) ∧ age(x, 35)
  - parent(x, y) ∧ male(x) is a **necessary and sufficient** description of father(x, y)
    - parent(x, y) ∧ male(x) ↔ father(x, y)

Main results (similar to PL)

- Compacity theorem: if all finite parts of KB (= set of sentences) have a model then KB has a model.
- Finitude theorem: H $\models$ A iff there exists a finite part B of H such that B $\models$ A

- Deduction theorem: let H be a set of sentences (formulas), and A and B two formulas such that A $\notin$ H

  H $\models$ A $\Rightarrow$ B iff H $\cup$ {A} $\models$ B
- Corollary (proof by contradiction) :

  H $\models$ A iff H $\cup$ {$\neg$A} is contradictory

RQ: Proofs in FOL

1. To prove that a formula is valid
   – Check all interpretations !

2. To prove that a formula is not valid
   – Find a counter-example (an interpretation for which the evaluation is false)

The second one seems easier…

Really check all interpretations?

- One useful way to represent formulas
  - **Clausal form**

- A theorem which helps
  - **Herbrand's Theorem**

- Use the "resolution" for FOL calculus (cf. formal systems)
  - Automatized the proofs.

A Procedure to obtain a Clausal Form from a formula

There are six stages to the conversion:

1. Remove $\boxed{\Rightarrow}$
2. De Morgan's to move negation to atomic propositions
   - Prenex form
3. Skolemizing (gets rid of $\boxed{\exists}$)
4. `Eliminating' universal quantifiers
5. Distributing AND over OR
6. Arrange into clauses and maybe reorder

Example

$$\forall x[big(x) \Rightarrow \neg(wet(x) \wedge \exists p[blue(p)])] \wedge \neg \forall q[orange(q)]$$

1. Remove implications (i.e. use the implication equivalence)

$$\forall x[\neg big(x) \vee \neg(wet(x) \wedge \exists p[blue(p)])] \wedge \neg \forall q[orange(q)]$$

2. Use De Morgan's to move negation to atomic propositions

$$\forall x[\neg big(x) \vee (\neg wet(x) \vee \neg \exists p[blue(p)])] \wedge \neg \forall q[orange(q)]$$

$$\forall x[\neg big(x) \vee (\neg wet(x) \vee \forall p[\neg blue(p)])] \wedge \exists q[\neg orange(q)]$$

Now the formula is in prenex form

Step 3: Skolemizing

We were here:

$$\forall x[\neg big(x) \vee (\neg wet(x) \vee \forall p[\neg blue(p)])] \wedge \exists q[\neg orange(q)]$$

In this step, we simply replace existentially quantified variables by constants.

Generally, if we have $\exists x[\ldots\text{something}(x)\ldots]$
we can replace this with $\text{something}(a34)$

I.e. we invent a specific individual for whom we can say that \`something' is true, and give it a unique name.

In this case, choosing any name that comes into my head, we get:

$$\forall x[\neg big(x) \vee (\neg wet(x) \vee \forall p[\neg blue(p)])] \wedge \neg orange(C3P0)$$

Step 4: Eliminate universal quantifiers

Since there are no longer any existential quantifiers, the order of the universal quantifiers doesn't matter. We may as well move them all the way to the left, like this:

$$\forall x \forall p [\neg big(x) \vee (\neg wet(x) \vee \neg blue(p))] \wedge \neg orange(C3P0)$$

Meanwhile, it is guaranteed that every variable here is universally quantified. So, why not just get rid of them, and we will take them as implicit

$$\neg big(x) \vee (\neg wet(x) \vee \neg blue(p)) \wedge \neg orange(C3P0)$$

Put the formula into CNF

$$\neg big(x) \vee (\neg wet(x) \vee \neg blue(p)) \wedge \neg orange(C3P0)$$

In this case we're fine already, since we don't have an AND within brackets.

It is already in Conjunctive normal form (CNF), we just have to tidy up:

$$(\neg big(x) \vee \neg blue(x) \vee \neg wet(p)) \wedge \neg orange(C3P0)$$

Now we have removed useless brackets, predicates within clauses are in alphabetical order, and the clauses themselves are in alphabetical order

About Skolemisation…

Actually Skolemization is *slightly* more complicated. Consider this:

$$\forall x[\exists y[mother(y,x)]]$$

If just replace the existential quantification with a constant we get

$$\forall x[mother(M302,x)]]$$

Which says M302 is everyone's mother (not what we wanted)

!!! Make the existential variable a *function* (called a Skolem function) of those universally quantified variables. So:

$$\forall x[mother(f1(x),x)]]$$

Here, f1(*k*) is the Skolem function, outputting the individual who is the mother of the input variable *k*.

Skolemization does not preserve the logical equivalence but if the formula is contradictory, so is the "skolemized" formula

Rules about skolemisation

- $(\exists y)\ (\forall x)\ likes(x,y) \xrightarrow{\text{skolemize}} (\forall x)\ likes(x,a)$

- $(\forall x)(\exists y)\ likes(x,y) \xrightarrow{\text{skolemize}} (\forall x)\ likes(x,f(x))$

- $(\forall x)\ (\forall y)\ (\exists z)\ P(x,y,z) \xrightarrow{\text{skolemize}} (\forall x)\ (\forall y)\ P(x,y,\ f(x,y))$
- …

Exercise 7

Consider the following axioms:

1. All hounds howl at night.
2. Anyone who has a cat will not have any mice.
3. Light sleepers do not have anything which howls at night.
4. John has a cat or a hound.
5. (Conclusion) If John is a light sleeper, then John does not have any mice.

- Translate them into first order logic and then find their clausal form

Rq on Ex 7

"Anyone who has a cat will not have any mice."

Correct translation: $\forall x\ \forall y\ [(have(x,y) \wedge cat(y)) \rightarrow \neg \exists z\ (have(x,z) \wedge mouse(z))]$

$\approx \forall x\ \forall y\ [(have(x,y) \wedge cat(y)) \rightarrow (\forall z\ (have(x,z) \rightarrow \neg mouse(z)))]$

$\approx \forall x\ \forall y\ [(have(x,y) \wedge cat(y)) \rightarrow (\forall z\ (mouse(z) \rightarrow \neg have(x,z)))]$

BUT $\forall x\ \forall y\ [(have(x,y) \wedge cat(y)) \rightarrow \forall z\ (\neg have(x,z) \wedge mouse(z))]$ IS NOT CORRECT because the second part is always false for every z that are not a mouse (this is TOO strong, the original sentence does not say anything about that)

Remember…

- <u>Satisfiability (Consistency)</u>
  - A formula G is satisfiable iff there exists an interpretation I such that G is evaluated to “T” (True) in I
  - I is then called a model of G and is said to satisfy G
- <u>Unsatisfiability (Inconsistency)</u>
  - G is inconsistent iff there exists no interpretation that satisfies G

Need for the theorem

- Proving satisfiability of a formula is better achieved by proving the unsatisfiability of its negation
- Proving unsatisfiability over a large set of interpretations is resource intensive
- Herbrands Theorem reduces the number of interpretations that need to be checked
  - Plays a fundamental role in Automated Theorem Proving

Herbrand Universe

- It is infeasible to consider all interpretations over all domains in order to prove unsatisfiability

- Instead, we try to fix a special domain (called a Herbrand universe) such that the formula, S, is unsatisfiable iff it is false under all the interpretations over this domain

Herbrand Universe (contd.)

- H_0 is the set of all constants in a set of clauses S
- If there are no constants in S, then H_0 will have a single constant, say $H_0 = \{a\}$
- For $i=1,2,3,\ldots$, let H_{i+1} be the union of H_i and set of all terms of the form $f^n(t_1,\ldots, t_n)$ for all n-place functions f in S, where t_j where $j=1,\ldots,n$ are members of the set H
- H_∞ is called the Herbrand universe of S

Herbrand Base

- Herbrand base or Atom Set of a set of clauses S: set of the *ground atoms* of the form $P^n(t_1, \ldots, t_n)$ for all n-place predicates P^n occuring in S, where $t_1, \ldots, t_n$ are elements of the Herbrand Universe of S

- A ground instance of a clause C of a set of clauses is a clause obtained by replacing variables in C by members of the Herbrand base of S

Example

$S = \{P(a), \neg P(x) \vee P(f(x)), Q(x)\}$

$H_0 = \{a\}$

$H_1 = \{a, f(a)\}$

$H_2 = \{a, f(a), f(f(a))\}$

$\vdots$

$H_\infty = \{a, f(a), f(f(a)), f(f(f(a))), \ldots\}$

Let, C = $Q(x)$

Here, $Q(a)$ and $Q(f(f(a)))$ are both ground instances of C

Atom Set (Herbrand base)

: $A = \{P(a), Q(a), P(f(a)), Q(f(a)), \ldots\}$

Example

Herbrand Universe of {P(f(x)),   R(a,g(y), b)}

Rq: missing fgfggggffgg in the herbrand universe…

Correction

$$
\begin{aligned}
H_0 &= \{a, b\} \\
H_1 &= \{a, b\} \cup \{f(a), f(b), g(a), g(b)\} \\
H_2 &= H_1 \cup \{f^2(a), f^2(b), g^2(a), g^2(b), f(g(a)), \\
&\qquad f(g(b)), g(f(a)), g(f(b))\} \\
&\vdots \\
H_i &= \{f^n(g^p(b)), f^k(g^\ell(a)) \mid \forall x \in \{n, p, k, \ell\}, x \leq i\}
\end{aligned}
$$

$$H_\infty = \{f^n(g^p(b)), f^k(g^\ell(a)) \mid \forall (n, p, k, \ell) \in \mathbb{N}^4\}.$$

Herbrand Interpretations

An Herbrand Interpretation is a set

$I = \{m_1, m_2, \ldots, m_n, \ldots\}$

where $m_j = A_j$ *or* $\neg A_j$ *(i.e.* A_j *is* set to true or false)

and $A = \{A_1, A_2, \ldots, A_n, \ldots\}$ *are taken in the Herbrand base (= atom set)*

H-Interpretations (contd.)

- Not all interpretations are H-Interpretations
- Given an interpretation I over a domain D, an H-Interpretation I* corresponding to I is an H-Interpretation that:
  - Has each element from the Herbrand Universe mapped to some element of D
  - Truth value of $P(h_1 ,\ldots, h_n)$ in I* must be same as that of $P(d_1 ,\ldots, d_n)$ in I

Example

$Let, S = \{P(x) \vee Q(x), R(f(y))\}$

$\Rightarrow$ Herbrand Universe $H = H_\infty = \{a, f(a), f(f(a)), \ldots\}$

$\Rightarrow$ Atom Set A is given by

$A = \{P(a), P(f(a)), P(f(f(a))), \ldots, Q(a), \ldots, R(a), \ldots\}$

$\Rightarrow$ Some Herbrand Interpretations are

$I_1 = \{P(a), P(f(a)), P(f(f(a))), \ldots, Q(a), \ldots, R(a), \ldots\}$

$I_2 = \{\neg P(a), \neg P(f(a)), \neg P(f(f(a))), \ldots, \neg Q(a), \ldots, \neg R(a), \ldots\}$

$I_3 = \{\neg P(a), \neg P(f(a)), \neg P(f(f(a))), \ldots, Q(a), \ldots, R(a), \ldots\}$

Use of H-Interpretations

- If an interpretation I satisfies a set of clauses S, over some domain D, then any one of the H-Interpretations I* corresponding to I will also satisfy S

- **A set of clauses S is unsatisfiable iff S is false under all H-Interpretations of S**

Herbrand's Theorem

A set S of clauses is unsatisfiable iff there is a finite *unsatisfiable* set S' of ground instances of clauses of S

Example

Let $S = \{P(x), \neg P(f(a))\}$

This set is unsatisfiable

Hence, by Herbrand's Theorem

there is a finite unsatisfiable set S' of ground

instances of clauses of S

One of these sets can be $S' = \{P(f(a)), \neg P(f(a))\}$

Exercise 8

Let H be the set of the 2 following formulas
$\forall x,\ P(a, x) \Rightarrow P(x, b)$
$\forall x\ \forall y\ \forall z\ ((P(x, y) \wedge P(y, z) \Rightarrow P(x, z))$

1. What is its Herbrand universe?

2. What are all the possible Herbrand interpretations? (there are 16 of them)

3. Find a Herbrand model. Deduce that H is satisfiable.

Decidability

$T \models F$

Proof Procedure
Query: does F follow from T

Algorithm terminate and answers “yes”

$T \not\models F$

Proof Procedure
Query: does F follow from T

Algorithm may or may not terminate and answers “no”

Predicate logic is only semi-decidable

Higher-order logic (HOL)

- FOL only allows to quantify over variables, and variables can only range over objects.
- HOL allows us to quantify over relations
- Example: (quantify over functions)

  "two functions are equal iff they produce the same value for all arguments"

  $\forall f\ \forall g\ (f = g) \leftrightarrow (\forall x\ f(x) = g(x))$

- Example: (quantify over predicates)

  $\forall r\ \text{transitive}(\ r\ ) \rightarrow (\forall xyz)\ r(x,y) \wedge r(y,z) \rightarrow r(x,z))$

- More expressive, but undecidable.

The return of FSR for 1st order logic

- Language : C (clauses)
- Induction scheme:
  - No axiom
  - Inference rule:

    $l_1 \vee D_1,\ \neg\ l_2 \vee D_2 \mid\text{--}\ D'_1 \vee D'_2$

  D_1, D_2 are clauses

  l_1, l_2 are literals with the same predicate symbol

  We want to prove that a set of clauses is contradictory (produce □)

  For the inference rule to be applicable, l_1 and l_2 must be unifiable

Adapt FSR for 1^{st} order logic L

Aim: do not lose the semantics of L <-> syntactic transformation

F: set of first order formulas

T1 : prenex form

F': formulas in prenex form

T2 : clausal form

F'': set of clauses

Cf. Slide 89

Use resolution

Substitution

- Given sentence S and substitution σ,
- Sσ denotes result of “plugging” σ into S
  - Example,
    - S = father(x, y)
    - σ = {x/John, y/Kim}
    - Subst(S, σ) = father(John, Kim)
      - Also written as Sσ = father(John, Kim)

Applying Substitution 1/2

- Given $\begin{cases} t & - \texttt{ a term} \\ \sigma & - \texttt{ a substitution} \end{cases}$

  "$t\sigma$" is the term resulting from applying substitution σ to term t.

- Small Examples:

  $\texttt{X\{X/a\} = a}$

  $\texttt{f(X)\{X/a\} = f(a)}$

Applying substitution 2/2

- **Example:** Using $t = \mathtt{f(\ a,\ h(Y,b),\ X\ )}$

  $\mathtt{f(\ a,\ h(Y,b),\ X\ )}\{\mathtt{X/b}\} = \mathtt{f(\ a,\ h(Y,b),\ b\ )}$

  $\mathtt{f(\ a,\ h(Y,b),\ X\ )}\{\mathtt{X/b\ Y/f(Z)}\} = \mathtt{f(\ a,\ h(f(Z),b),\ b\ )}$

  $\mathtt{f(\ a,\ h(Y,b),\ X\ )}\{\mathtt{X/Z\ Y/f(Z,a)}\} = \mathtt{f(\ a,\ h(f(Z,a),b),\ Z\ )}$

  $\mathtt{f(\ a,\ h(Y,b),\ X\ )}\{\mathtt{W/Z}\} = \mathtt{f(\ a,\ h(Y,b),\ X\ )}$

- σ need not include all variables in t;
  σ can include variables not in t.

Semantics of a substitution

- All its elements are applied simultaneously.

- Example: The application of the substitution {x/g(y), y/h(z), z/x} to

f(x, y, g(z), w) is f(g(y), h(z), g(x), w),
and not f(g(h(x)), h(x), g(x), w).

- The order of the elements in a substitution list is irrelevant

Composition of Substitutions

- Composition:
  $\sigma \circ \theta$ is *composition* of substitutions σ , θ.
  For any term t, $t[\sigma \circ \theta] = (t\sigma)\theta$.

- Example:

$$\begin{aligned}
\texttt{f(X)[\{X/Z\}} \circ \texttt{\{Z/a\}]} &= \texttt{(f(X)\{X/Z\})\{Z/a\}} \\
&= \texttt{f(Z)\{Z/a\}} \\
&= \texttt{f(a)}
\end{aligned}$$

- $\sigma \circ \theta$ is a *substitution* (usually)

- Eg:
  $\texttt{[\{X/a\}} \circ \texttt{\{Y/b\}]} = \texttt{\{X/a,} \quad \texttt{Y/b\}}$
  $\texttt{[\{X/Z\}} \circ \texttt{\{Z/a\}]} = \texttt{\{X/a,} \quad \texttt{Z/a\}}$

Unification

- To apply the resolution rule, we need to find a pair of complementary literals.
- Unification is a method for making two literals identical.
- If two literals can be made identical, they are unifiable.

- t_1 and t_2 are *unified by* σ iff

  $t_1\sigma = t_2\sigma$.

  Then σ is called a *unifer*

  t_1 and t_2 are *unifiable*

- Examples:

t_1	t_2	unifer		term
f(b,c)	f(b,c)	{}		f(b,c)
f(X,b)	f(a,Y)	{X/a	Y/b}	f(a,b)
f(a,b)	f(c,d)	*		
f(a,b)	f(X,X)	*		
f(X,a)	f(Y,Y)	{X/a	Y/a}	f(a,a)
f(g(U),d)	f(X,U)	{U/d	X/g(d)}	f(g(d),d)
f(X)	f(g(X))	*		
f(X,g(X))	f(Y,Y)	*		
f(X)	f(Y)	{X/Y}		f(Y)

- Unifier for $\mathtt{t}_1 = \mathtt{f(X)}$ and $\mathtt{t}_2 = \mathtt{f(Y)}$

θ	$\mathtt{t}_1\theta = \mathtt{t}_2\theta =$
{X/Y}	f(Y)
{Y/X}	f(X)
{Y/a X/a}	f(a)
{Y/g(b,Z) X/g(b,Z)}	f(g(b(Z)))
{X/Y W/f(q,Z)}	f(Y)

- {Y/X} and {X/Y} make sense, but
  {Y/a X/a} has irrelevant constant
  {X/Y W/g} has irrelevant binding (W)

Exercise 9: compare unifiers

- Let A= p(g(y),f(x,h(x),y)) and B = p(x,f(g(z),w,z)), be 2 terms

1. Is the following substitution a unifier of the terms? What is the result of the unification?

   $\sigma 1$ = { x/ g(f(a)), y / f(a), w/ h(g(f(a))), z/ f(a) }

2. Same questions with:

   $\sigma 2$ = { x/ g(z), w/ h(g(z)), y/z }

3. One of the unifiers is more general than the other, which one?

Sol Ex 9

- A= p(g(y),f(x,h(x),y)) and B = p(x,f(g(z),w,z))

1. $\sigma 1$ = { x/ g(f(a)), y / f(a), w/ h(g(f(a))), z/ f(a) }

$A\sigma 1$ = p(g(f(a)),f(g(f(a)), h(g(f(a))), f(a)))
$B\sigma 1$ = p(g(f(a)),f(g(f(a)), h(g(f(a))), f(a)))
So $A\sigma 1 = B\sigma 1$ so $\sigma 1$ is a unifier

2. $\sigma 2$ = { x/ g(z), w/ h(g(z)), y/z }

$A\sigma 2$ = p(g(z),f(g(z), h(g(z)),z))
$B\sigma 2$ = p(g(z),f(g(z), h(g(z)),z))
So $A\sigma 2 = B\sigma 2$ so $\sigma 2$ is a unifier

3. Now $\sigma 1 = \sigma 2$ o {z/ f(a) } so $\sigma 2$ is more “general” than $\sigma 1$

Quest for Best Unifier

- Wish list:
  - No irrelevant constants
    So {Y/X} prefered over { Y/a, X/a }
  - No irrelevant bindings
    So {Y/X} prefered over { Y/X, W/f(4,Z) }

- Spse λ_1 has constant where λ_2 has variable
  (Eg, λ_1 = { X/a, Y/a }, λ_2 = {X/Y})
  Then $\exists$ subsitution μ s.t. $\lambda_2 \circ \mu = \lambda_1$
  (Eg, μ = {Y/a}: {X/Y}$\circ${Y/a} = { X/a, Y/a })

- Spse λ_1 has extra binding over λ_2
  (Eg, λ_1 = {X/a, Y/b}, λ_2 = {X/a})
  Then $\exists$ subsitution μ s.t. $\lambda_2 \circ \mu = \lambda_1$
  (Eg, μ = {Y/b}: {X/a}$\circ${Y/b} = {X/a, Y/b})

Most General Unifier

- σ is a *mgu* for t_1 and t_2 iff
  - σ unifies t_1 and t_2, and
  - $\forall\ \mu$: unifier of t_1 and t_2,
    $\exists$ subsitution, θ, s.t. $\sigma \circ \theta = \mu$.
    (Ie, for all terms t, $t\mu = (t\sigma)\theta$.)

MGU

- Example: $\sigma = \{\texttt{X/Y}\}$ is mgu for $\texttt{f(X)}$ and $\texttt{f(Y)}$.
  Consider unifier $\mu = \{\texttt{X/a} \quad \texttt{Y/a}\}$.
  Use substitution $\theta = \{\texttt{Y/a}\}$:

$$\begin{aligned}
\texttt{f(X)}\mu &= \texttt{f(X)}\{\texttt{X/a Y/a}\} \\
&= \texttt{f(a)}
\end{aligned}$$

$$\begin{aligned}
\texttt{f(X)}[\sigma \circ \theta] &= (\texttt{f(X)}\sigma) \quad \theta \\
&= (\texttt{f(X)}\{\texttt{X/Y}\})\theta \\
&= \texttt{f(Y)}\{\texttt{Y/a}\} \\
&= \texttt{f(a)}
\end{aligned}$$

Similarly, $\texttt{f(Y)}\mu = \texttt{f(a)} = \texttt{f(Y)}[\sigma \circ \theta]$

(μ is NOT a mgu, as $\neg\exists\theta'$ s.t. $\mu \circ \theta' = \sigma$!)

```
Recursive Procedure MGU (x,y)
    If  x=y ⇒ Return ()
    If  Variable(x) ⇒ Return( MguVar(x,y) )
    If  Variable(y) ⇒ Return( MguVar(y,x) )
    If  Constant(x) or Constant(y) ⇒ Return( False )
    If  Not(Length(x) = Length(y)) ⇒ Return( False )
    g ← []
    For  i = 1..Length(x)  do
      s ← MGU( Part(x,i), Part(y,i) )
      g ← Compose(g,s)
      x ← Substitute(x,g)
      y ← Substitute(y,g)
    Return( g )
End MGU


Procedure MguVar (v,e)
    If    Includes(v,e) ⇒ Return( False )
    Else  Return( [v/e] )
End
```

Notes on MGU

- If two terms are unifiable, then there exists a MGU
- There can be more than one MGU, but they differ only in variable names
- Not every unifier is MGU
- A MGU uses constants only as necessary

Ex 10: unification and MGUs

Let f(x,y,z) and f(x,a,x) be 2 terms. Let the 4 following substitutions be:

- $\sigma1$ = { (x, b), (y, a), (z, b) }
- $\sigma2$ = { (y, a), (z, x) }
- $\sigma3$ = { (x, y), (y, a), (z, y) }
- $\sigma4$ = { (x, z), (y, a), (z, x) }

1. Which substitutions σi are unifiers of the 2 terms?
2. Which substitutions are MGUs of the 2 terms?
3. Apply the MGU algorithm and verify your answers.

Exercise 11: MGU

Find the MGU for the following sets of terms (one for each if it exists)

1. { f(a, x, h(x)), f(a, y, y) }
2. { g(x, g(y, z)), g(g(a, b), x), g(x, g(a ,z)) }
3. { g(y, h(h(x))), g(h(a), y) }
4. { f(x, y, w), f(h(g(v, y)), g(v, w), h(a)), f(h(z), y, h(v)) }

Sol Ex 11.4

1. f(x, y, w),
2. f(h(g(v, y)), g(v, w), h(a)),
3. f(h(z), y, h(v))

MGU of 1 and 2: σ1 = { x / h(g(v, g(v, h(a)))), y/ g(v, h(a)), w/ h(a) }
1σ1 = 2σ1 = f(h(g(v, g(v, h(a)))), g(v, h(a)), h(a))

MGU of 2σ1 and 3 = σ2 = { x / h(g(a, g(a, h(a)))), y/ g(a, h(a)), w/ h(a), z/g(a, g(a, h(a))), v/a }

So MGU = { x / h(g(a, g(a, h(a)))), y/ g(a, h(a)), w/ h(a), z/ g(a, g(a, h(a))), v/a }

Resolution algorithm

Theorem: First-order **resolution is sound**. That is,for any set of clauses Φ and for any clause φ,

$\Phi \vdash \varphi$ implies $\Phi \models \varphi$.

Theorem: First-order resolution **is complete for refutation**. That is, for any set of clauses Φ,

$\Phi \models \square$ implies $\Phi \vdash \square$

Algorithm: Resolution refutation in predicate logic.

1. Transform the set of axioms A in clausal form and get S0
2. Negate the theorem (conclusion), transform the negated theorem in clausal form and add it to S0

   3. **repeat**

   3.1. Select two clauses C1 and C2 from S

   3.2. Compute R = {Res(C1,C2)}

   3.3. **if** $\square \notin R$

   then add R to S

   until $\square \in R$ **or** there are no two other clauses that resolve **or** a predefined amount of effort has been exhausted

4. **if** $\square \in R$

   then the theorem is proven

5. **else**

   if there are no two other clauses that resolve

   then it is not a theorem

   else there is not a definite conclusion

end.

Example

- Horses are faster than dogs and there is a greyhound that is faster than every rabbit. We know that Harry is a horse and that Ralph is a rabbit. Derive that Harry is faster than Ralph.

- Horse(x) Greyhound(y) Dog(y) Rabbit(z) Faster(y,z)

$$\forall x\ \forall y\ \mathrm{Horse}(x) \land \mathrm{Dog}(y) \Rightarrow \mathrm{Faster}(x,y)$$

$$\exists y\ \mathrm{Greyhound}(y) \land (\forall z\ \mathrm{Rabbit}(z) \Rightarrow \mathrm{Faster}(y,z))$$

$$\mathrm{Horse}(\mathrm{Harry})$$

$$\mathrm{Rabbit}(\mathrm{Ralph})$$

$$\forall y\ \mathrm{Greyhound}(y) \Rightarrow \mathrm{Dog}(y)$$

$$\forall x\ \forall y\ \forall z\ \mathrm{Faster}(x,y) \land \mathrm{Faster}(y,z) \Rightarrow \mathrm{Faster}(x,z)$$

A1. $\forall x\ \forall y\ Horse(x) \wedge Dog(y) \Rightarrow Faster(x,y)$

A2. $\exists y\ Greyhound(y) \wedge (\forall z\ Rabbit(z) \Rightarrow Faster(y,z))$

A3. $Horse(Harry)$

A4. $Rabbit(Ralph)$

A5. $\forall y\ Greyhound(y) \Rightarrow Dog(y)$

A6. $\forall x\ \forall y\ \forall z\ Faster(x,y) \wedge Faster(y,z) \Rightarrow Faster(x,z)$

T $Faster(Harry,Ralph)$

```
C1. ¬Horse(x1) ∨ ¬Dog(y1) ∨ Faster(x1,y1)
C2. Greyhound(Greg)
C2' ¬Rabbit(z1) ∨ Faster(Greg,z1)
C3. Horse(Harry)
C4. Rabbit(Ralph)
C5. ¬Greyhound(y2) ∨ Dog(y2)
C6. ¬Faster(x2,y3) ∨ ¬Faster(y3,z2) ∨ Faster(x2,z2)
C7. ¬Faster(Harry,Ralph)
```

Proof [illegible]

C7 C6

$x/Harry$ $z/Ralph$

C8 $\sim Faster(Harry, y) \vee \sim Faster(y, Ralph)$

C1 C3 $x/Harry$

C9 $\sim Dog(y) \vee Faster(Harry, y)$

C5

C10 $\sim Greyhound(y) \vee Faster(Harry, y)$

C2

C11 $Faster(Harry, Greg)$ $y/Greg$

C8

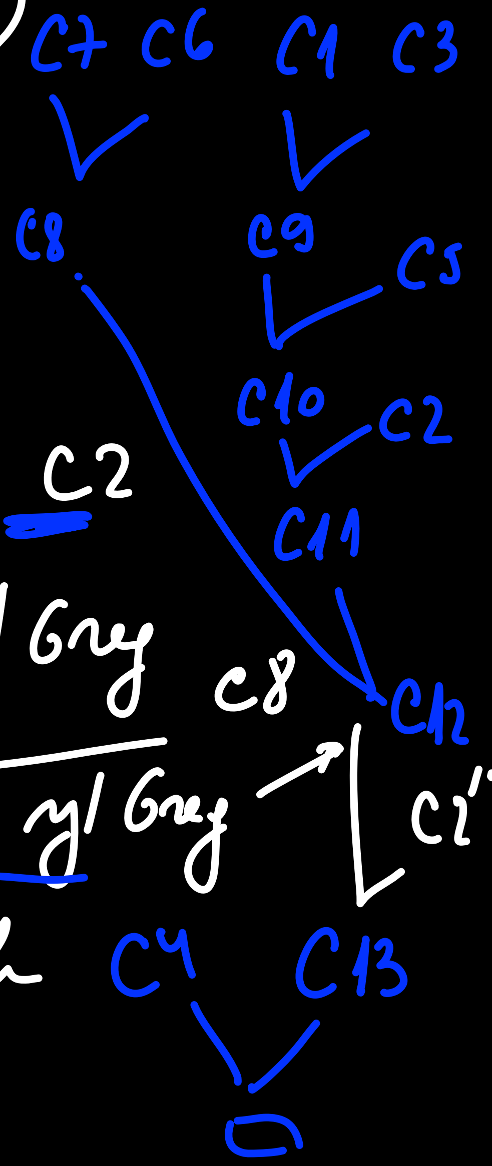
$\sim Faster(Greg, Ralph)$ $y/Greg$

C2'

$\sim Rabbit(Ralph)$ $z/Ralph$

C4

$\square$



Exercise 12 (resolution)

1. $H = \{P(a), \forall x\ (P(x) \Rightarrow Q\ (s(x))), \forall x\ (Q(x) \Rightarrow P(s(x)))\}$

   Show that we can produce $P(s(s(s(s(a)))))$ from H using FSR

2. Is the reasoning about the hound who howls at night (slide) valid?

Exercise 14: Is this valid?

1. All stamp collectors born in Italy eat olives
2. No president is a stamp collector
3. Anyone born in Italy eats olives or is a stamp collector
4. Every president who is not a stamp collector is not born in Italy

C: All Italians eat olives