Logic and AI Programming:

Introduction to Logic
Introduction to Prolog

F. SADRI

Autumn Term

October – December

Course Material

Course material will be available via CATE.

Please print notes for the rest of the course and tutorial exercises as they appear on CATE.

CONTENTS

Introduction to logic and some of its applications in computer science

- Propositional logic
  - Syntax
  - Semantics (Truth Tables)
  - Rules of inference (Natural Deduction)
- Predicate logic
  - Syntax
  - Informal semantics
  - Rules of inference (Natural Deduction)
- Prolog programming
- Time permitting: Abduction

Books

background reading on logic

- Any book on logic will have useful examples.
- Richard Spencer-Smith, Logic and Prolog, Harvester Wheatsheaf, (The library has a number of copies)
- Jim Woodcock and Martin Loomes, Software Engineering Mathematics", Pitman Publishing

Books

Prolog

- Ivan Bratko, "Prolog programming for artificial intelligence“, Addison-Wesley, Third Edition, 2001 and later.

Assessments and Examination

- One Logic Coursework
- Two Prolog Lab Assessments
- One Examination in May:

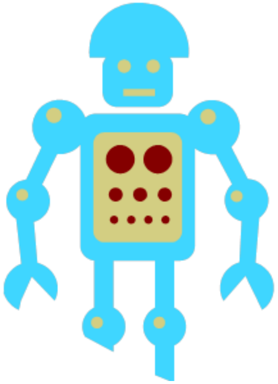
  Paper M1 (Program Design and Logic) will have:

  – two questions on Logic and

  – two questions on Object-Oriented Design

Logic has many applications in computing

Term 2:



- Logic-based Learning course
- Introduction to Artificial Intelligence course in term 2
- Database languages, e.g. relational calculus and some features of SQL

Datalog: emerging e.g. in data integration, declarative networking, information extraction, network monitoring, security and cloud computing

Logic has many applications in computing

- Basis of a family of programming languages, e.g. Prolog.
- Good mathematical foundation for reasoning about computer and information systems, and behaviour of programs.
- Software engineering: Formal specifications and formal verification of programs.

Logic has many applications in computing

How do you make sure a program is "correct"?

Review, again and again and ….

- Review the spec
- Review the design description
- Review the code

Test, again and again and ….

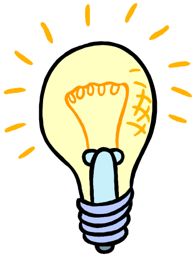
- Unit testing
- Integration testing
- Validation testing

Logic provides precise, unambiguous descriptions against which a system can be tested.

But that is not enough.

How many tests do you run to be sure the system is correct?

- Logic provides a way of proving the system correct and
- this can be automated too.



Logic is also useful more generally in life

- Clear thinking
- Judging validity of arguments and justification of conclusions
- Spotting inconsistencies
- Awareness and avoidance of ambiguities of natural language

Which of the following arguments are valid?

1) Advertisement for a computing book: If you don’t use computers you don’t need this book. But you are a person who uses computers. So you need this book.
2) If you work hard you will succeed. So if you do not succeed you have not worked hard.

Which of the following arguments are valid?

3) Heard in a radio interview with a well-known politician: All our problems have come from the EU. So nothing good has resulted for us from our membership of the EU.

4) We cannot win the war on poverty without spending money. So if we do spend money we will conquer poverty.

Another argument – Is it valid?

5) One of the old arguments of tobacco spokesmen against the claim that smoking causes lung cancer:

Lung cancer is more common among male smokers than it is among female smokers. If smoking were the cause of lung cancer, this would not be true. The fact that lung cancer is more common among male smokers means that it is caused by something in the male make-up. If follows that lung cancer is not caused by smoking, but something in the male make-up.

More reasoning exercises

1. All the trees in the park are flowering trees.
2. Some of the trees in the park are dogwoods.
3. All dogwoods in the park are flowering trees.

If the first two statements are true, the third statement is

A. true
B. false
C. uncertain

More reasoning exercises

1. All the tulips in Zoe's garden are white.
2. All the pansies in Zoe's garden are yellow.
3. All the flowers in Zoe's garden are either white or yellow

If the first two statements are true, the third statement is

A. true
B. false
C. uncertain

Fact 1:	All dogs like to run.
Fact 2:	Some dogs like to swim.
	Some dogs look like their masters.
Fact 3:	

If the first three statements are facts, which of the following statements must also be a fact?

I:	All dogs who like to swim look like their masters.
II:	Dogs who like to swim also like to run.
III:	Dogs who like to run do not look like their masters.

A. I only
B. II only
C. II and III only
D. None of the statements is a known fact.

Propositional Logic

- A good place to start.
- All logics are based on propositional logic.

Propositional Logic

Examples:

Passing the exams, the courseworks and the projects implies passing the MSc.

$(pass\_exams \wedge pass\_cwks \wedge pass\_projs) \rightarrow pass\_MSc$

$\wedge$: and

$\rightarrow$: implies (if-then)

You cannot retake the exams if you have not passed the courseworks or if you have cheated in the exams.

$(\neg$ pass_cwks $\vee$ cheat$) \rightarrow \neg$ retake_exams

$\neg$: **not** sometimes written as ~

$\vee$: **or**

Components of a logic

- Language:
  - alphabet : symbols
  - syntax : rules for putting together the symbols to make grammatically correct sentences.
- Semantics:

  meaning of the symbols and the sentences.
- Inference rules

The Propositional Language: Alphabet

- **Propositional symbols**

  e.g. pass_exams, p, q, r, s, p1

- **Logical connectives:**

  $\wedge$: **and (conjunction)**

  $\vee$: **(inclusive) or (disjunction)**

  $\neg$: **not (negation)**

  $\rightarrow$: **implication (if-then)**

  $\leftrightarrow$: **double implication (if and only if)**

The Propositional Language: Syntax of a grammatically correct sentence

(well formed formula, wff)

- A propositional symbol is a wff.
- If W, W1 and W2 are wffs then so are

  $\neg (W)$

  $(W1 \wedge W2)$

  $(W1 \vee W2)$

  $(W1 \rightarrow W2)$ $(W2 \leftarrow W1)$

  $(W1 \leftrightarrow W2)$

- There are no other wffs.

Examples

$(p \rightarrow q)$ is a wff

$((p \rightarrow q) \vee ((p \wedge r) \rightarrow \neg(s)))$ is a wff

$(r \wedge \vee t)$ is not a wff

$(p \neg \rightarrow q)$ is not a wff

Exercise:

Formulate arguments (1) and (2) at the beginning of the notes.

Some notes on simplifying syntax

- To avoid ending up with a large number of brackets one can drop the outermost brackets.

Examples:

$(\mathbf{p} \rightarrow \mathbf{q})$ can be written as $\mathbf{p} \rightarrow \mathbf{q}$

$((\mathbf{p} \rightarrow \mathbf{q}) \vee \mathbf{r})$ can be written as $(\mathbf{p} \rightarrow \mathbf{q}) \vee \mathbf{r}$.

- “¬ “ binds more closely than the other connectives. This can be used to drop some brackets.

Example

$(\neg (p) \wedge q) \rightarrow t$ can be written as

$(\neg p \wedge q) \rightarrow t$

- $\wedge$ and $\vee$ bind more closely than $\to$ and $\leftrightarrow$. This can be used to drop some brackets.

Examples:

$(\neg p \wedge q) \to t$ can be written as

$\neg p \wedge q \to t$.

$(p \wedge q) \to (r \vee s)$ can be written as

$p \wedge q \to r \vee s$.

Binding Strength of the Connectives

To avoid having to use many brackets, there is a convention of ordering the connectives.

Also :

- Order of precedence
- Binding priority

Binding Strength of the Connectives

Strongest	$\neg$
	$\wedge$
	$\vee$
	$\rightarrow$
Weakest	$\leftrightarrow$

Binding conventions: Examples

- $p \vee q \wedge r$ is understood as $p \vee (q \wedge r)$
- $\neg p \vee q$ is understood as $(\neg p) \vee q$
- $p \to q \leftrightarrow r$ is understood as $(p \to q) \leftrightarrow r$

I prefer the first and third bracketed versions. They are more clear, and having a few brackets is not much of a burden! Please don't write unreadable formulas like

$p \vee \neg q \to \neg r \leftrightarrow \neg\neg s \wedge t \vee \neg u$

Use brackets to remove ambiguity

Example:

$$P \to Q \to R$$

is ambiguous.

In general

$$P \to (Q \to R)$$

and

$$(P \to Q) \to R$$

are not equivalent (do not have the same meaning).

Binding conventions

So $p \to q \to r$

is a problem.

It needs brackets to disambiguate it.

But $p \wedge q \wedge r$ and $p \vee q \vee r$

are fine (to be discussed later).

Use brackets to remove ambiguity

Example:

Go to work and go to dinner or go to the cinema.

Exercise:

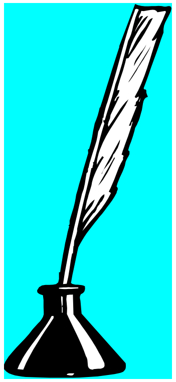
Spot the ambiguity.

Give two possible formulations of the sentence.

Go to work and go to dinner, or go to the cinema.

Go to work, and go to dinner or go to the cinema.

Exercise:
Which of the following are wffs?



Assume

p, q, r, sad, happy,

tall, rich, work_hard,

steal, borrow, and possess

are propositional symbols.

- $rich \rightarrow happy$
- $(p \vee q) \;\wedge\; (r \rightarrow p)$
- $p\vee \;\rightarrow\; q$
- $sad \;\rightarrow\; \neg happy$
- $\neg happy \;\leftarrow\; sad$

- rich $\rightarrow$ $\neg\neg$happy
- rich $\leftrightarrow$ (work_hard $\vee$ steal)
- (steal $\wedge$ $\vee$ borrow) $\rightarrow$ possess
- (steal $\vee$ borrow) $\rightarrow$ possess
- steal $\vee$ borrow $\rightarrow$ possess

- $(p \wedge q \rightarrow r) \wedge (\neg p \rightarrow \neg q)$
- $p \rightarrow \neg p$
- $p \wedge \neg p$

We will look at

- *Parse trees*
- *Principle connectives*
- *Subformulas*

in the lecture.

Notes on terminology

- $\neg$ is a **unary** operator.
- The other connectives are **binary** operators.
- $X \vee Y$ is called the **disjunction of X and Y**.
- $X \vee Y$ X and Y are **disjuncts.**
- $X \wedge Y$ is called the **conjunction of X and Y**.
- $X \wedge Y$ X and Y are **conjuncts.**
- $\neg X$ is called the **negation of X.**

Notes on terminology cntd.

- $A \rightarrow B$ is called an implication.

  A is called the antecedent,

  B is called the consequent.

A *Literal* is a proposition or the negation of a proposition.

Semantics

Provides

- The meaning of the simple (atomic) units
- Rules for putting together the meaning of the atomic units to form the meaning of the complex units (sentences).

Semantics specifies under what circumstances a sentence is *true* or *false*.

Example

John is not happy, but he is comfortable.

Represent as $\neg h \wedge c$

Four possible cases

h	**c**	**¬ h**	**¬ h ∧c**
T	T	F	F
T	F	F	F
F	T	T	T
F	F	T	F

T (truth) and
F (falsity)
are known as **truth values.**

Constructing Truth Tables for the connectives

A	**$\neg$ A**
T	F
F	T

A	**B**	**A ∧ B**
T	T	T
T	F	F
F	T	F
F	F	F

A	**B**	**A∨B**
T	T	T
T	F	T
F	T	T
F	F	F

A	**B**	**A→B**
T	T	T
T	F	F
F	T	T
F	F	T

A	B	$A \leftrightarrow B$
T	T	T
T	F	F
F	T	F
F	F	T

The truth value of a wff is uniquely determined by the truth value of its components.

Example:

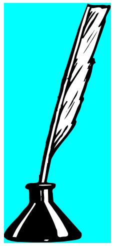
The truth table for the wff $(p \vee q) \wedge (r \rightarrow p)$

is as follows:

p	q	r	$p \vee q$	$r \to p$	$(p \vee q) \wedge (r \to p)$
T	T	T	T	T	T
T	T	F	T	T	T
T	F	T	T	T	T
T	F	F	T	T	T
F	T	T	T	F	F
F	T	F	T	T	T
F	F	T	F	F	F
F	F	F	F	T	F

Exercise:

How many rows will there be in a truth table for a wff containing n propositional symbols?



Exercise:

Go to work and go to dinner or go to the cinema.

The two wffs formulating the two interpretations of the above sentence are:

$\mathbf{w \wedge (d \vee c)}$

$\mathbf{(w \wedge d) \vee c}$

Under which interpretation(s) do the truth values of the two wffs differ?

Notes

The connective "$\vee$" stands for *inclusive* or, i.e.

$p \vee q$ is interpreted as true when either proposition is true or both are.

Often in English when we use "or" we intend *exclusive* or, e.g.

- I'll go shopping *or* I'll stay at home.
- We will meet at his house or at the club.

Notes cntd.

In this propositional language there is no connective for exclusive or, but we can still express the concept, e.g.

$$(\text{goShopping} \vee \text{stayHome}) \wedge \neg(\text{goShopping} \wedge \text{stayHome})$$

In general "p *exor* q" can be represented as:

$$(p \vee q) \wedge \neg(p \wedge q)$$

Exercise:

Draw the truth table of the first wff above.

Notes cntd.

- Law of excluded middle:

  A proposition (and consequently a wff) is either true or false – there is no middle ground, no "unknown".
- So propositional logic is a *2-valued* logic.
- There are other logics, including *3-valued* ones.
- SQL, for example, implements 3-valued logic, where comparisons with NULL, including that of another NULL gives *UNKNOWN*.

- A proposition (and consequently a wff) cannot be both true and false.

Exercise:

Draw the truth table for $A \wedge \neg A$.

Notes cntd.

- The interpretation of "$\rightarrow$" may be unintuitive sometimes.
- The semantics of "$\rightarrow$" is very simple in logic.
- $A \rightarrow B$ is simply the same as $\neg A \vee B$.
- In English we use " if .. then" in many different ways, and sometimes quite confusingly.
- Don't read $A \rightarrow B$ as "A causes B".

Notes cntd.

- For any wffs A and B, "$A \leftrightarrow B$" is true exactly when A and B have the same truth values, i.e. when they are both true or both false.

Some definitions

Definition :

A wff which evaluates to true in every interpretation of its constituent parts is called a **tautology**.

Example $\mathbf{A \vee \neg A}$

$\mathbf{A \rightarrow A}$

The two wffs above represent the **Law of excluded middle**.

Definition

A wff which evaluates to false in every interpretation of its constituent parts is called an **inconsistency (contradiction)**, or is said to be **inconsistent**.

- Example $\mathbf{A \wedge \neg A}$

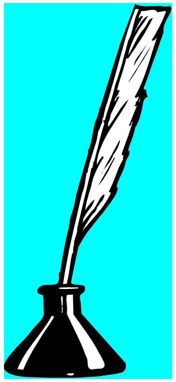
Definition

A wff which is neither a tautology, nor an inconsistency is a **contingency**, or is said to be **contingent**.

Exercise

For each of the following determine if it is a tautology, inconsistency or contingency by drawing the truth table.

a. $P \wedge (P \vee Q)$
b. $(P \vee Q) \wedge (P \rightarrow Q)$
c. $Q \wedge \neg P \wedge (P \vee (Q \rightarrow P))$
d. $(P \wedge (Q \vee P)) \leftrightarrow P$
e. $(P \rightarrow Q) \rightarrow (\neg P \vee Q)$
f. $((P \rightarrow Q) \wedge (R \rightarrow S) \wedge (P \vee R)) \rightarrow (Q \vee S)$



Definition: Equivalence

Two wffs are **equivalent** iff their truth values are the same under every interpretation.

A is equivalent to B is represented as

$\mathbf{A \equiv B}$.

"$\equiv$" is the metasymbol for equivalence.

Some useful equivalences

Double Negation Rule

$$\neg\neg A \equiv A$$

Implication Rule

$$A \rightarrow B \equiv \neg A \vee B$$

Some useful equivalences

Commutative Rules

$$A \wedge B \equiv B \wedge A$$

$$A \vee B \equiv B \vee A$$

Associative Rules

$$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$$

$$(A \vee B) \vee C \equiv A \vee (B \vee C)$$

Some useful equivalences

Idempotence

$$A \wedge A \equiv A$$

$$A \vee A \equiv A$$

Some useful equivalences

Distributive Rules

$$\mathbf{A \lor (B \land C) \equiv (A \lor B) \land (A \lor C)}$$

$$\mathbf{A \land (B \lor C) \equiv (A \land B) \lor (A \land C)}$$

De Morgan's Rules

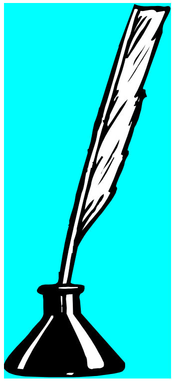
$$\mathbf{\neg(A \lor B) \equiv \neg A \land \neg B}$$

$$\mathbf{\neg(A \land B) \equiv \neg A \lor \neg B}$$

Some useful equivalences

$$\mathbf{A \leftrightarrow B \equiv}$$
$$\mathbf{(A \rightarrow B) \land (B \rightarrow A)}$$

Exercise



Show

$A \to B \equiv \neg(A \wedge \neg B)$

Example

I get an MSc $\to$ I get big salary $\equiv$

$\neg$(I get an MSc $\wedge$ $\neg$ I get big salary)

Exercises

Show

$$A \leftrightarrow B \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$$

Show

$$A \leftrightarrow B \equiv \neg A \leftrightarrow \neg B$$