E0 205 Mathematical logic and theorem proving

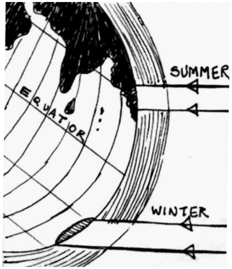
Deepak D'Souza and Kamal Lodaya

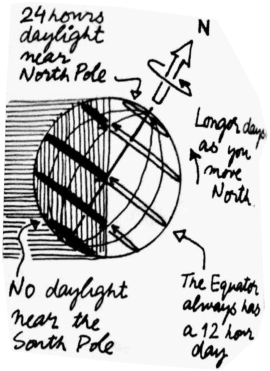
January 2025

Outline

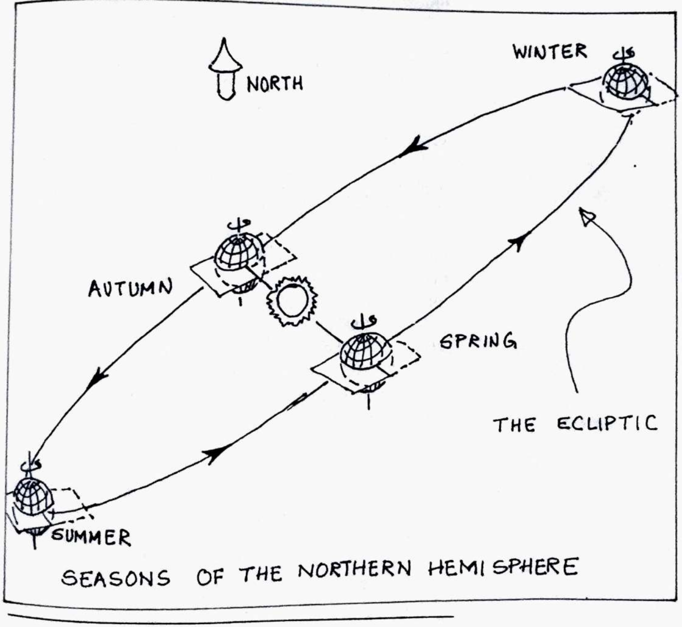
Why is it 14°C in the morning these days?

- Naive science: Earth is farther away from Sun
- Geography: Sun overhead on Tropic of Capricorn on 21 December
- Physics: Temperature is distribution of sunlight over area



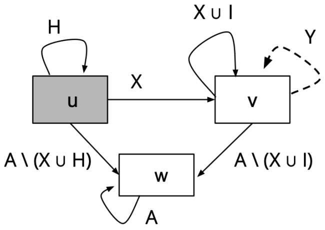

24 hours daylight near North Pole
N
Longer days as you move North
No daylight near the South Pole
The Equator always has a 12 hour day

- You rotate on Earth from West to East
- To you, Sun appears to rise in East and set in West
- Tilt in axis of rotating Earth gives longer days and shorter nights in summer



Drawings from *Signs of the zodiac*, by Deepak Khemani, BGVS

Introducing a new agent (Conan Doyle 1894)



At the Reichenbach falls in *The final problem* (see *The memoirs of Sherlock Holmes*), the arch-criminal Moriarty (H) wants to deceive Dr Watson (X) away from his friend Holmes by saying that there is a lady (Y, introduced by dashed lines) who is ill at the inn they are staying and needs his help. Because Watson would not believe Moriarty, he forges a letter from the innkeeper (I), and Watson gets deceived. Thus Watson believes the innkeeper knows of the existence of the lady, whereas Moriarty knows that the innkeeper knows nothing.

Who came first: Mahavira or Buddha? (Lodaya 2024)

- Here is how you may find some dates introduced in a school history textbook:

1. 15th century BCE: Vedas (Hinduism)
2. 6th/5th century BCE: Vardhman Mahavira (Jainism)
3. 6th/5th century BCE: Gautam Buddha (Buddhism)

- A total order is used but no explanation is provided for the dates
- When prompted, O1-Preview provided an explanation for ordering Mahavira before Buddha
- When prompted, O1-Preview provided an explanation for ordering Buddha before Mahavira

Broadening the domain

Here is a more refined picture known to historians.

- 15th-11th century BCE: Rig veda (Rig vedic Sanskrit)
- 12th-8th century: Sama veda
- 11th-8th century: Yajur veda
- 10th-8th century: Atharva veda (includes kala)
- 7th-6th century: Zarathustra (mazdayasna, heaven, hell), early upanishads
- 6th century: Parshva (nigantha, moksha), Sanjaya Belathiputta (ajnana), Kassapa (akriya)
- 6th century: Kaccayana (sassata), Ajita Kesakambali (charvaka)
- 5th century BCE: Makhkhali Gosala (ajivika), Vardhman Mahavira (nigantha), Gautam Buddha

What justification can be provided for these dates?

Development of ideas BCE

- 15th-8th century: Vedas (sacrifices, referred to later)
- 6th century: Parshva (23rd tirthankara, fourfold way with asteya, satya, aparigraha and ahimsa)
- 6th century: Sanjaya Belathiputta, Kassapa, Kaccayana, Ajita Kesakambali (referred to by later philosophers)
- 5th century: Makhkhali Gosala (moksha after 84 lakh births), Vardhman Mahavira refer to each other, their followers call them 24th tirthankara, digambar aparigraha, confusion between Ajivika and Nigantha
- 5th century: Mahavira (fivefold way with brahmacharya, good act karma can pay for bad karma, even across births)
- 5th century: Gautam Buddha (hard asceticism before Middle Way, celibacy, karma as intention)

Social sciences require higher-order ideas and definitions, which may have multiple interpretations

Geometry

A logic is a formal language equipped with a method for making inferences in this language.

P1: Side AB = Side DE (a “formula” in the formal language)

P2: Side AC = Side DF (another formula)

P3: $\angle BAC = \angle EDF$ (another formula)

C: $\therefore \triangle ABC \equiv \triangle DEF$ (formula prefixed “therefore”)

Domain of discourse is geometry.

We say “from the finite set of premisses $P1$, $P2$, $P3$, derive that the conclusion C holds”.
In short, “from $P1$, $P2$, $P3$, derive C”.
In short, $P1, P2, P3 \vdash C$.

Formal geometry

- Weird symbols like $\angle$ and $\triangle$ are used, we have to be told what they mean.
- Words like “Side” in English (any natural language) are used in a technical sense.
- We are also told when from a given set of formulas (premisses), we can derive a formula (the conclusion), using the inference method $\vdash$.

Vocabulary (Ramanujam 2021)

Question (Symbols $sqr, -, /, 1$)
Evaluate $x^2 - 1/x - 1$ *at* $x = 1$.

Question (Symbols $lim, \rightarrow, sqr, -, /, 1$)
Evaluate $\lim_{x \rightarrow 1} x^2 - 1/x - 1$.
Domain of discourse is reals.

Question (Symbols $odd, even, +$)
Show that sum of two odd numbers is even.
Domain of discourse is integers.

Question (Symbols $\sin, \cos, \tan, +, -, \times, /, sqrt, \mathbb{N}$)
Solve $2 \sin x - 2\sqrt{3} \cos x - \sqrt{3} \tan x + 3 = 0$.
Domain of discourse is angles.

Types (Ramanujam 2021)

- In $x + 3 = 5$, solve for variable x.
- In $x + y = 5$, x can now take many values.
- In $x + y = y + x$, x can be any number whatsoever.
- In $x = k$, k a constant. A variable is a constant?

- Is $x^2 - 2 = 0$ true? Depends on what x ranges over.
- Is the angle sum property true? For angles in the plane.
- Is multiplication repeated addition? For integer multipliers.
- In general $Th \models C$. Formula consequence of a theory.

- In factoring, polynomials are expressions.
- In graphing, polynomials are functions.
- In finding roots, polynomials are equations.
- In college math, polynomials are elements of a ring.

Programming

A programming language is a formal language equipped with rules which specify how a program in this programming language is to be executed. What does this program do?

```
pre {list V i}    /* int *i points to a list of items V */
int *j = null;
while (i != null) {
    int *k = i->next;
    i->next = j;
    j = i;
    i = k;
}
post {list W j}    /* int *j points to a list of items W */
```

Tool jargon

- A theorem prover is a procedure for checking whether a derivation of the form $P1, P2, \ldots, Pn \vdash C$ is allowed or not. $P1, P2, \ldots, Pn, C$ are formulas.
- A model checker is a procedure for checking whether a consequence of the form $Th \models C$ is valid or not. Here C is a formula, but Th is a theory, for example, it could be integer arithmetic, or it could be a program with statements $S1\,S2 \ldots Sn$.
- A sat solver is a procedure for checking whether a finite theory is consistent, $Con\ Th$ (no contradiction derivable from Th). The solver finds a way of assigning to variables of formulas in Th so that it can be made true.
- These procedures are computational, but they may fail to terminate; the tool user may have to manually terminate it.

Direct or indirect

- Why such a convoluted way? Why not just take a program and check what is the (partial) function that it computes?
- Or why not take a program and check whether giving it the inputs $I1, I2, I3$ will produce the output O?

Theorem (Turing 1936)

There is no (terminating) algorithm which can check the two questions above.

We can, of course, start running the program to check this (or simulate it in some way) but that defeats the very idea of verification. Also, how do we simulate the program on every possible input from an infinite set of values?

Note: Alan Turing's parents lived in Chhatrapur, Odisha, in 1911. He was born in London in June 1912.

The verification process

- So we need to look at the system (which has a finite implementation) and reason about whether some possibilities can happen or not, and from this infer whether the specification is met. In short, we need to use logic.
- From Turing's theorem, we know that it is not possible to always get an answer to the verification question.
- A large finite system may have an astronomical number of states; analyzing it might take a very long time or may require the tool to use a very large amount of space. So the tool may crash, or may have to be crashed!
- Then we build a coarser model of the system and run the tool on the model. Hopefully that will not crash.

Propositional logic (de Morgan 1847, Boole 1854)

- Atomic formulas are symbols from alphabet S, which we do not analyze any further.
- Formulas are built up from atomic formulas using the Boolean operations $\wedge, \vee, \neg$.

SYNTAX (BACKUS-NAUR/CHOMSKY/CONTEXT-FREE GRAMMAR)

$p, q ::= a \in S \mid (\neg p) \mid (p \vee q) \mid (p \wedge q)$

$(p \rightarrow q) \overset{\text{def}}{=} ((\neg p) \vee q)$; $(p \leftrightarrow q) \overset{\text{def}}{=} ((p \rightarrow q) \wedge (q \rightarrow p))$. We can construct a parse tree for any formula.

Example: Let a be the proposition "Augustus de Morgan was born in Madura, Madras presidency".
Let b be the proposition "Augustus de Morgan was born in 1806".
Examine the formulas $(a \wedge b)$, $(\neg a)$, $(\neg b)$, $((\neg a) \vee (\neg b))$, $(a \rightarrow b)$, $(b \rightarrow a)$.

Predicate logic with quantifiers (Frege 1879)

- Names like *Odd* and *Even* that we use in math are called predicates (in one variable) and serve as an alphabet S.
  Example: $Odd(x)$ has truth value when variable x is given an integer value. $Born(a)$ has truth value of individual a being born.

  So we have two types of syntactic expressions in predicate logic.
- Terms are built up from variables and constant symbols using function symbols (eg, numbers with $+, \times$). The value of a term is in a domain of discourse such as the integers.
- Atomic formulas are built up from terms by applying an n-ary predicate symbol to n terms. For example, the binary relational operators $<, \leq, >, \geq, =, \neq$ return boolean value.
- Formulas are built up from atomic formulas using the Boolean operations $\wedge, \vee, \neg$ and quantifiers $(\forall x), (\exists x)$.
  A formula can only take a boolean value T or F.

Logic of programs (Floyd 1968, Hoare 1969)

- To verify a program, we associate a correctness triple $\{p\}$ C $\{q\}$ with every construct C in the program. The precondition p and the postcondition q are formulas in a logic.
  eg, $\{x > 0\}$ x = x-1; $\{x \geq 0\}$
- Burstall 1969 pointed out that you need to do induction on the data type (eg, lists)
- Manna and Waldinger 1985, 1990 present theories for inducting over data types
- For example, in proving programs with lists, we might use the inductive predicate *list* taking a sequence of values (the "data" in the list) and a natural number (an "index" or "position" in the list):
  $\mathit{list}\ \varepsilon\ i \overset{\text{def}}{=} i = \mathit{nil}$, and
  $\mathit{list}\ aV\ i \overset{\text{def}}{=} (\mathit{head}(i) = a) \wedge (\mathit{list}\ V\ \mathit{tail}(i))$
- Symbols include unary functions *head* and *tail*, they form terms with a list index as argument.

Broadening with a Floyd-Hoare invariant

$\star$ is a conjunction which separates a list into two disjoint parts ($\wedge$ has no such requirement), giving precise assertions (Pym-O'Hearn, Reynolds 1999)

```
pre $\{list\ V\ \mathrm{i}\}$   /* i points to $V$ */
j = null;
inv $\{\exists W, X((list\ W\ \mathrm{i} \star list\ X\ \mathrm{j}) \wedge V^R = W^R X)\}$
while (i != null) {
    $\{\exists a, W, X((list\ aW\ \mathrm{i} \star list\ X\ \mathrm{j}) \wedge V^R = (aW)^R X)\}$
    k = i->next;
    i->next = j;
    j = i;
    i = k;
inv $\{\exists W, X((list\ W\ \mathrm{i} \star list\ X\ \mathrm{j}) \wedge V^R = W^R X)\}$
}
post $\{list\ V^R\ \mathrm{j}\}$   /* j points to reversal of $V$ */
```

Intermediate assertions complete the proof

inv $\{\exists a, W, X((\mathit{list}\ aW\ \mathrm{i} \star \mathit{list}\ X\ \mathrm{j}) \wedge V^R = (aW)^R X)\}$
$\Longrightarrow \{\exists a, W, X, k((\mathrm{i} \mapsto a, k \star \mathit{list}\ W\ k \star \mathit{list}\ X\ \mathrm{j}) \wedge V^R = (aW)^R X)\}$

```
    k = i->next;
```

assert $\{\exists a, W, X((\mathrm{i} \mapsto a, \mathrm{k} \star \mathit{list}\ W\ \mathrm{k} \star \mathit{list}\ X\ \mathrm{j}) \wedge V^R = (aW)^R X)\}$

```
    i->next = j;
```

assert $\{\exists a, W, X((\mathrm{i} \mapsto a, \mathrm{j} \star \mathit{list}\ W\ \mathrm{k} \star \mathit{list}\ X\ \mathrm{j}) \wedge V^R = (aW)^R X)\}$
$\Longrightarrow \{\exists a, W, X((\mathit{list}\ W\ \mathrm{k} \star \mathit{list}\ aX\ \mathrm{i}) \wedge V^R = W^R aX)\}$
$\Longrightarrow \{\exists W, X((\mathit{list}\ W\ \mathrm{k} \star \mathit{list}\ X\ \mathrm{i}) \wedge V^R = W^R X)\}$

```
    j = i; i = k;
```

inv $\{\exists W, X((\mathit{list}\ W\ \mathrm{i} \star \mathit{list}\ X\ \mathrm{j}) \wedge V^R = W^R X)\}$

Theorem (Cook 1978)

Given a predicate language which can express loop invariants and intermediate assertions, checking correctness in Floyd-Hoare (Pym-O'Hearn-Reynolds) logic of programs reduces to theorem proving in predicate logic (with separating conjunction).