CS228 : Logic for Computer Science

Module 1: Propositional Logic

Lecture 1: Syntax

Instructor: S. Akshay

IIT Bombay, India

Logistics

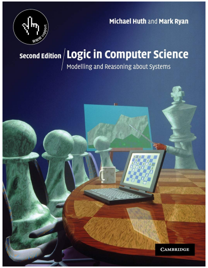
- Course webpage: `https://www.cse.iitb.ac.in/~akshayss/courses/cs228`

Logistics

- Course webpage: `https://www.cse.iitb.ac.in/~akshayss/courses/cs228`
- Piazza invites sent!

Logistics

- Course webpage: `https://www.cse.iitb.ac.in/~akshayss/courses/cs228`
- Piazza invites sent!
- Textbook: For the first part of the course, we will use Huth and Ryan's book on Logic in Computer Science.



Module 1: Logic, the language/calculus for reasoning

A whole zoo of symbolic logics out there!

We will go from simple to more complicated

- Propositional Logic

Module 1: Logic, the language/calculus for reasoning

A whole zoo of symbolic logics out there!

We will go from simple to more complicated

- Propositional Logic
- Predicate Logic
  - First order logic

Module 1: Logic, the language/calculus for reasoning

A whole zoo of symbolic logics out there!

We will go from simple to more complicated

- Propositional Logic
- Predicate Logic
  - First order logic
  - Monadic second order logic

Module 1: Logic, the language/calculus for reasoning

A whole zoo of symbolic logics out there!

We will go from simple to more complicated

- Propositional Logic
- Predicate Logic
    - First order logic
    - Monadic second order logic
    - Higher order logics

Module 1: Logic, the language/calculus for reasoning

A whole zoo of symbolic logics out there!

We will go from simple to more complicated

- Propositional Logic
- Predicate Logic
  - First order logic
  - Monadic second order logic
  - Higher order logics
- Temporal logics

Module 1: Logic, the language/calculus for reasoning

A whole zoo of symbolic logics out there!

We will go from simple to more complicated

- Propositional Logic
- Predicate Logic
  - First order logic
  - Monadic second order logic
  - Higher order logics
- Temporal logics
- Modal logics

Topic 1.1

Propositional logic: Syntax

An example beyond good and evil

In a country, every person is either good or evil. Good people always tell the truth and evil people always lie.

An example beyond good and evil

In a country, every person is either good or evil. Good people always tell the truth and evil people always lie.

You meet two citizens A and B. $\boxed{A \text{ says, “I am evil or } B \text{ is good”.}}$
What are A and B?

An example beyond good and evil

In a country, every person is either good or evil. Good people always tell the truth and evil people always lie.

You meet two citizens A and B. $\boxed{A \text{ says, “I am evil or } B \text{ is good”.}}$
What are A and B? **Exercise 1.1: Work this out!**

An example beyond good and evil

In a country, every person is either good or evil. Good people always tell the truth and evil people always lie.

You meet two citizens A and B. $\boxed{A \text{ says, "I am evil or } B \text{ is good".}}$
What are A and B? **Exercise 1.1: Work this out! How do we find all possibilities?**

An example beyond good and evil

In a country, every person is either good or evil. Good people always tell the truth and evil people always lie.

You meet two citizens A and B. $\boxed{A \text{ says, "I am evil or } B \text{ is good".}}$
What are A and B? **Exercise 1.1: Work this out! How do we find all possibilities?**

This lecture: How to formalize this reasoning?

- Step 1: Introduce symbols!
  - Let p_A denote the statement: A is good. It is either true or false.
  - Similarly, let p_B denote "B is good."

An example beyond good and evil

In a country, every person is either good or evil. Good people always tell the truth and evil people always lie.

You meet two citizens A and B. $\boxed{A \text{ says, ``I am evil or } B \text{ is good''.}}$
What are A and B? **Exercise 1.1: Work this out! How do we find all possibilities?**

This lecture: How to formalize this reasoning?

- Step 1: Introduce symbols!
  - Let p_A denote the statement: A is good. It is either true or false.
  - Similarly, let p_B denote "B is good."
- Step 2: Encode puzzle using the symbols: $\underbrace{(\neg p_A \vee p_B)}_{\text{Statement of A}}$

An example beyond good and evil

In a country, every person is either good or evil. Good people always tell the truth and evil people always lie.

You meet two citizens A and B. $\boxed{A \text{ says, ``I am evil or } B \text{ is good''.}}$
What are A and B? **Exercise 1.1: Work this out! How do we find all possibilities?**

This lecture: How to formalize this reasoning?

- Step 1: Introduce symbols!
  - Let p_A denote the statement: A is good. It is either true or false.
  - Similarly, let p_B denote "B is good."
- Step 2: Encode puzzle using the symbols: $\underbrace{(\neg p_A \vee p_B)}_{\text{Statement of A}} \leftrightarrow p_A$

An example beyond good and evil

In a country, every person is either good or evil. Good people always tell the truth and evil people always lie.

You meet two citizens A and B. $\boxed{A \text{ says, “I am evil or } B \text{ is good”.}}$
What are A and B? **Exercise 1.1: Work this out! How do we find all possibilities?**

This lecture: How to formalize this reasoning?

- Step 1: Introduce symbols!
  - Let p_A denote the statement: A is good. It is either true or false.
  - Similarly, let p_B denote "B is good."
- Step 2: Encode puzzle using the symbols: $\underbrace{(\neg p_A \vee p_B)}_{\text{Statement of A}} \leftrightarrow p_A$
- Step 3: Solve the puzzle: Assign truth values to p_A and p_B such that the above "formula" holds. (how? trial and error?)

From English to Math

So in this example we went from an English puzzle to a symbolic logic problem.

What does a language have?

- Symbols: good, A, B, is, not

From English to Math

So in this example we went from an English puzzle to a symbolic logic problem.

What does a language have?

- Symbols: good, A, B, is, not
- Syntax: Grammar

From English to Math

So in this example we went from an English puzzle to a symbolic logic problem.

What does a language have?

- Symbols: good, A, B, is, not
- Syntax: Grammar
- Semantics: Meaning

From English to Math

So in this example we went from an English puzzle to a symbolic logic problem.

What does a language have?

- Symbols: good, A, B, is, not
- Syntax: Grammar
- Semantics: Meaning

But any natural language has so much ambiguity!

From English to Math

So in this example we went from an English puzzle to a symbolic logic problem.

What does a language have?

- Symbols: good, A, B, is, not
- Syntax: Grammar
- Semantics: Meaning

But any natural language has so much ambiguity!

"I saw the man with the telescope."

A Symbolic logic

The logic is over a list of propositions. Some examples:

- A is good.
- This tree is green
- The fridge is broken
- ... *many more*

A Symbolic logic

The logic is over a list of propositions. Some examples:

- A is good.
- This tree is green
- The fridge is broken
- ... *many more*
    - A proposition: any sentence for which we can answer if it is True or False.

A Symbolic logic

The logic is over a list of propositions. Some examples:

- A is good. p_A
- This tree is green q
- The fridge is broken r
- ... *many more*
  - A proposition: any sentence for which we can answer if it is True or False.
  - We do not care what each one says: we can give them a symbol.

A Symbolic logic

The logic is over a list of propositions. Some examples:

- A is good. p_A
- This tree is green q
- The fridge is broken r
- ... *many more*
  - A proposition: any sentence for which we can answer if it is True or False.
  - We do not care what each one says: we can give them a symbol.
  - Not a proposition: "May the force be with you!"

A Symbolic logic

The logic is over a list of propositions. Some examples:

- A is good. p_A
- This tree is green q
- The fridge is broken r
- ... *many more*
  - A proposition: any sentence for which we can answer if it is True or False.
  - We do not care what each one says: we can give them a symbol.
  - Not a proposition: "May the force be with you!"

Exercise 1.2

- Give two examples of sentences that are not propositions.
- Give three examples of propositions.

Propositions as Boolean variables

We start with a set **AP** of finite or countably many (atomic) propositions.

Propositions as Boolean variables

We start with a set **AP** of finite or countably many (atomic) propositions.

- Since **AP** is countable, we assume that they are indexed.

$$\mathbf{AP} = \{p_1, p_2, \dots\}$$

Propositions as Boolean variables

We start with a set **AP** of finite or countably many (atomic) propositions.

- Since **AP** is countable, we assume that they are indexed.

$$\mathbf{AP} = \{p_1, p_2, \ldots\}$$

- These are just names/symbols without inherent meaning

Propositions as Boolean variables

We start with a set **AP** of finite or countably many (atomic) propositions.

- Since **AP** is countable, we assume that they are indexed.

$$\mathbf{AP} = \{p_1, p_2, \ldots\}$$

- These are just names/symbols without inherent meaning
- We may also use $p, q, r, .., x, y, z$ to denote them

Propositions as Boolean variables

We start with a set **AP** of finite or countably many (atomic) propositions.

- Since **AP** is countable, we assume that they are indexed.

$$\mathbf{AP} = \{p_1, p_2, \dots\}$$

- These are just names/symbols without inherent meaning
- We may also use $p, q, r, .., x, y, z$ to denote them
- Propositions are Boolean variables (as they take true or false as value!)

Topic 1.2

Propositional Formulas

Logical connectives to build propositions

Starting from “atomic” propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another,

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together,

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together, e.g., A is good & it is raining.

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together, e.g., A is good & it is raining.
- at least one of the two statements are true,

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together, e.g., A is good & it is raining.
- at least one of the two statements are true, e.g., A is good or it is raining.
- Implication: if A is true then B is also true,

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together, e.g., A is good & it is raining.
- at least one of the two statements are true, e.g., A is good or it is raining.
- Implication: if A is true then B is also true, e.g., If it rains then trees will grow.

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together, e.g., A is good & it is raining.
- at least one of the two statements are true, e.g., A is good or it is raining.
- Implication: if A is true then B is also true, e.g., If it rains then trees will grow.
- Equivalence: truth value of two statements are same,

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together, e.g., A is good & it is raining.
- at least one of the two statements are true, e.g., A is good or it is raining.
- Implication: if A is true then B is also true, e.g., If it rains then trees will grow.
- Equivalence: truth value of two statements are same, e.g., water is blue if and only if sky is black.

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together, e.g., A is good & it is raining.
- at least one of the two statements are true, e.g., A is good or it is raining.
- Implication: if A is true then B is also true, e.g., If it rains then trees will grow.
- Equivalence: truth value of two statements are same, e.g., water is blue if and only if sky is black.

We need True and False also!

- always true:

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together, e.g., A is good & it is raining.
- at least one of the two statements are true, e.g., A is good or it is raining.
- Implication: if A is true then B is also true, e.g., If it rains then trees will grow.
- Equivalence: truth value of two statements are same, e.g., water is blue if and only if sky is black.

We need True and False also!

- always true: e.g., If A is good, then A is good.
- always false:

Logical connectives to build propositions

Starting from "atomic" propositions, we can combine them as follows:

Building new propositions from existing ones

- a statement that says negation of another, e.g., A is not good.
- two statements are true together, e.g., A is good & it is raining.
- at least one of the two statements are true, e.g., A is good or it is raining.
- Implication: if A is true then B is also true, e.g., If it rains then trees will grow.
- Equivalence: truth value of two statements are same, e.g., water is blue if and only if sky is black.

We need True and False also!

- always true: e.g., If A is good, then A is good.
- always false: e.g., A is good and A is bad.

Logical connectives

formal name	symbol	read as
true	$\top$	top
false	$\bot$	bot

Logical connectives

formal name	symbol	read as
true	$\top$	top
false	$\bot$	bot
negation	$\neg$	not
conjunction	$\wedge$	and
disjunction	$\vee$	or

Logical connectives

formal name	symbol	read as
true	$\top$	top
false	$\bot$	bot
negation	$\neg$	not
conjunction	$\wedge$	and
disjunction	$\vee$	or
implication	$\rightarrow$	implies
equivalence	$\leftrightarrow$	iff
exclusive or	$\oplus$	xor

Logical connectives

formal name	symbol	read as
true	$\top$	top
false	$\bot$	bot
negation	$\neg$	not
conjunction	$\wedge$	and
disjunction	$\vee$	or
implication	$\rightarrow$	implies
equivalence	$\leftrightarrow$	iff
exclusive or	$\oplus$	xor
open parenthesis	(	
close parenthesis	)	

Logical connectives

formal name	symbol	read as	
true	$\top$	top	0-ary symbols
false	$\bot$	bot	
negation	$\neg$	not	unary symbols
conjunction	$\wedge$	and	binary symbols
disjunction	$\vee$	or	
implication	$\rightarrow$	implies	
equivalence	$\leftrightarrow$	iff	
exclusive or	$\oplus$	xor	
open parenthesis	(		punctuation
close parenthesis	)		

- We assume that the logical connectives are not in **AP**.

Logical connectives

formal name	symbol	read as	
true	$\top$	top	0-ary symbols
false	$\bot$	bot	
negation	$\neg$	not	unary symbols
conjunction	$\wedge$	and	binary symbols
disjunction	$\vee$	or	
implication	$\rightarrow$	implies	
equivalence	$\leftrightarrow$	iff	
exclusive or	$\oplus$	xor	
open parenthesis	(		punctuation
close parenthesis	)		

- We assume that the logical connectives are not in **AP**.
- Do we need all these connectives? Which can be derived from others?

Propositional formulas

A propositional formula is a finite string containing symbols in **AP** and logical connectives. The set of all propositional formulas over **AP** is denoted **Prop**.

Propositional formulas

A propositional formula is a finite string containing symbols in **AP** and logical connectives. The set of all propositional formulas over **AP** is denoted **Prop**. Well, not every string...

Propositional formulas

A propositional formula is a finite string containing symbols in **AP** and logical connectives. The set of all propositional formulas over **AP** is denoted **Prop**. Well, not every string...

Formal Definition

The set of propositional formulas is the smallest set **Prop** such that

Propositional formulas

A propositional formula is a finite string containing symbols in **AP** and logical connectives. The set of all propositional formulas over **AP** is denoted **Prop**. Well, not every string...

Formal Definition

The set of propositional formulas is the smallest set **Prop** such that

- $\top, \bot \in$ **Prop**
- if $p \in$ **AP** then $p \in$ **Prop**
- if $F \in$ **Prop** then $\neg F \in$ **Prop**
- if $\circ$ is a binary symbol and $F, G \in$ **Prop** then $(F \circ G) \in$ **Prop**

Propositional formulas

A propositional formula is a finite string containing symbols in **AP** and logical connectives. The set of all propositional formulas over **AP** is denoted **Prop**. Well, not every string...

Formal Definition (Note! This is an inductive definition)

The set of propositional formulas is the smallest set **Prop** such that

- $\top, \bot \in$ **Prop**
- if $p \in$ **AP** then $p \in$ **Prop**
- if $F \in$ **Prop** then $\neg F \in$ **Prop**
- if $\circ$ is a binary symbol and $F, G \in$ **Prop** then $(F \circ G) \in$ **Prop**

Propositional formulas

Another way to write this definition

$F \in \mathbf{Prop}$ if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in \mathbf{AP}$.

Propositional formulas

Another way to write this definition – Have you see this way of defining before?

$F \in$ **Prop** if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in \mathbf{AP}$.

Propositional formulas

Another way to write this definition . This is called the Backus Naur Form.

$F \in \mathbf{Prop}$ if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in \mathbf{AP}$.

Propositional formulas

Another way to write this definition . This is called the Backus Naur Form.

$F \in \textbf{Prop}$ if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in \textbf{AP}$. Any such formula is also called well-formed.

Propositional formulas

Another way to write this definition . This is called the Backus Naur Form.

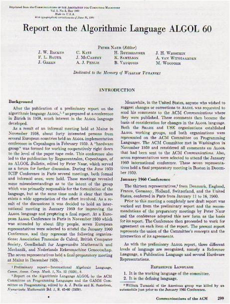
$F \in \textbf{Prop}$ if

$$F ::= p \mid \top \mid \bot \mid \neg F \mid (F \vee F) \mid (F \wedge F) \mid (F \rightarrow F) \mid (F \leftrightarrow F) \mid (F \oplus F)$$

where $p \in \textbf{AP}$. Any such formula is also called well-formed.

A bit of history

Named after John Backus and Peter Naur, who used it to formally specify the grammar of ALGOL 60 in 1963 – one of the earliest programming languages, and ancestor of Pascal, C, and more!



Reprinted from the Communications of the Association for Computing Machinery
Vol. 3, No. 5, May 1960
Made in U.S.A.
With typographical corrections as of June 28, 1960

Report on the Algorithmic Language ALGOL 60

Peter Naur (Editor)

J. W. Backus, C. Katz, H. Rutishauser, J. H. Wegstein, F. L. Bauer, J. McCarthy, K. Samelson, A. van Wijngaarden, J. Green, A. J. Perlis, B. Vauquois, M. Woodger

Dedicated to the Memory of William Turanski

INTRODUCTION

Background

After the publication of a preliminary report on the algorithmic language Algol,[1,2] as prepared at a conference in Zürich in 1958, much interest in the Algol language developed.

As a result of an informal meeting held at Mainz in November 1958, about forty interested persons from several European countries held an Algol implementation conference in Copenhagen in February 1959. A "hardware group" was formed for working cooperatively right down to the level of the paper tape code. This conference also led to the publication by Regnecentralen, Copenhagen, of an ALGOL *Bulletin*, edited by Peter Naur, which served as a forum for further discussion. During the June 1959 ICIP Conference in Paris several meetings, both formal and informal ones, were held. These meetings revealed some misunderstandings as to the intent of the group which was primarily responsible for the formulation of the language, but at the same time made it clear that there exists a wide appreciation of the effort involved. As a result of the discussions it was decided to hold an international meeting in January 1960 for improving the Algol language and preparing a final report. At a European Algol Conference in Paris in November 1959 which was attended by about fifty people, seven European representatives were selected to attend the January 1960 Conference, and they represent the following organizations: Association Francaise de Calcul, British Computer Society, Gesellschaft fur Angewandte Mathematik und Mechanik, and Nederlands Rekenmachine Genootschap. The seven representatives held a final preparatory meeting at Mainz in December 1959.

Meanwhile, in the United States, anyone who wished to suggest changes or corrections to Algol was requested to send his comments to the ACM *Communications* where they were published. These comments then became the basis of consideration for changes in the Algol language. Both the Share and USE organizations established Algol working groups, and both organizations were represented on the ACM Committee on Programming Languages. The ACM Committee met in Washington in November 1959 and considered all comments on Algol that had been sent to the ACM *Communications*. Also, seven representatives were selected to attend the January 1960 international conference. These seven representatives held a final preparatory meeting in Boston in December 1959.

January 1960 Conference

The thirteen representatives,[3] from Denmark, England, France, Germany, Holland, Switzerland, and the United States, conferred in Paris from January 11 to 16, 1960.

Prior to this meeting a completely new draft report was worked out from the preliminary report and the recommendations of the preparatory meetings by Peter Naur and the conference adopted this new form as the basis for its report. The Conference then proceeded to work for agreement on each item of the report. The present report represents the union of the Committee's concepts and the intersection of its agreements.

As with the preliminary Algol report, three different levels of language are recognized, namely a Reference Language, a Publication Language and several Hardware Representations.

Reference Language

1. It is the working language of the committee.
2. It is the defining language.

[1] Preliminary report—International Algebraic Language, *Comm. Assoc. Comp. Mach.* 1, No. 12 (1958), 8.

[2] Report on the Algorithmic Language ALGOL by the ACM Committee on Programming Languages and the GAMM Committee on Programming, edited by A. J. Perlis and K. Samelson, *Numerische Mathematik* Bd. 1, S. 41-60 (1959).

[3] William Turanski of the American group was killed by an automobile just prior to the January 1960 Conference.

Communications of the ACM 299

Examples of propositional formulas

Exercise 1.3 Which of the following belong to **Prop** (i.e., are well-formed)?

1. $\top \to \bot$
2. $(\top \to \bot)$
3. $(p_1 \to \neg p_2)$
4. (p_1)
5. $\neg\neg\neg\neg\neg\neg\neg p_1$

Examples of propositional formulas

Exercise 1.3 Which of the following belong to **Prop** (i.e., are well-formed)?

1. $\top \rightarrow \bot$
2. $(\top \rightarrow \bot)$
3. $(p_1 \rightarrow \neg p_2)$
4. (p_1)
5. $\neg\neg\neg\neg\neg\neg\neg\neg p_1$

Obs: Not all strings over **AP** and logical connectives are in **Prop**.

Examples of propositional formulas

Exercise 1.3 Which of the following belong to **Prop** (i.e., are well-formed)?

1. $\top \to \bot$
2. $(\top \to \bot)$
3. $(p_1 \to \neg p_2)$
4. (p_1)
5. $\neg\neg\neg\neg\neg\neg\neg\neg\neg p_1$

Obs: Not all strings over **AP** and logical connectives are in **Prop**.

How can we argue that a string does or does not belong to **Prop**?

The Lady Or the Tiger

A prisoner must choose between three rooms. One room has a Lady and other two have tigers. If the prisoner chooses the room with the Lady, he wins! If he chooses a wrong room, he (probably) gets eaten by the tiger!
Each room has a sign in front of it. BUT Only one of the signs is true; the other two are false.

(**Room1**) *The Lady is not in this room*
(**Room2**) *The Lady is not in this room*
(**Room3**) *The Lady is in Room 2*

Which room has the Lady?!

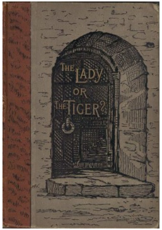
The Lady Or the Tiger

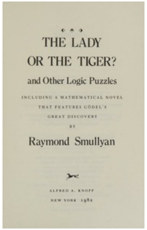
A prisoner must choose between three rooms. One room has a Lady and other two have tigers. If the prisoner chooses the room with the Lady, he wins! If he chooses a wrong room, he (probably) gets eaten by the tiger!
Each room has a sign in front of it. BUT Only one of the signs is true; the other two are false.

(**Room1**)	*The Lady is not in this room*
(**Room2**)	*The Lady is not in this room*
(**Room3**)	*The Lady is in Room 2*

Which room has the Lady?!

- Story from book by Frank Stockton.
- Puzzle from book by Raymond M. Smullyan





The Lady Or the Tiger

A prisoner must choose between three rooms. One room has a Lady and other two have tigers. If the prisoner chooses the room with the Lady, he wins! If he chooses a wrong room, he (probably) gets eaten by the tiger!
Each room has a sign in front of it. BUT Only one of the signs is true; the other two are false.

(**Room1**)	*The Lady is not in this room*
(**Room2**)	*The Lady is not in this room*
(**Room3**)	*The Lady is in Room 2*

Which room has the Lady?!

Exercise 1.4

1. Solve the above puzzle!
2. Write each of the above statements in propositional logic.
3. Reduce the problem to a question in this logic!

Formalizing Exercise 1.4

- Propositions $S1, S2, S3$ represent whether the sign of room 1, 2, 3 (resp.) is true.
- Propositions $L1, L2, L3$ represent Lady in room 1, 2, 3 resp.

Formalizing Exercise 1.4

- Propositions $S1, S2, S3$ represent whether the sign of room 1, 2, 3 (resp.) is true.
- Propositions $L1, L2, L3$ represent Lady in room 1, 2, 3 resp.
- The statements given:
  - $S1 \leftrightarrow \neg L1$, $S2 \leftrightarrow \neg L2$, $S3 \leftrightarrow L2$.

Formalizing Exercise 1.4

- Propositions $S1, S2, S3$ represent whether the sign of room 1, 2, 3 (resp.) is true.
- Propositions $L1, L2, L3$ represent Lady in room 1, 2, 3 resp.
- The statements given:
  - $S1 \leftrightarrow \neg L1$, $S2 \leftrightarrow \neg L2$, $S3 \leftrightarrow L2$.
  - $(L1 \wedge \neg L2 \wedge \neg L3) \vee (L2 \wedge \neg L1 \wedge \neg L3) \vee (L3 \wedge \neg L1 \wedge \neg L2)$.
  - $(S1 \wedge \neg S2 \wedge \neg S3) \vee (S2 \wedge \neg S1 \wedge \neg S3) \vee (S3 \wedge \neg S1 \wedge \neg S2)$.

Formalizing Exercise 1.4

- Propositions $S1, S2, S3$ represent whether the sign of room 1, 2, 3 (resp.) is true.
- Propositions $L1, L2, L3$ represent Lady in room 1, 2, 3 resp.
- The statements given:
  - $S1 \leftrightarrow \neg L1$, $S2 \leftrightarrow \neg L2$, $S3 \leftrightarrow L2$.
  - $(L1 \wedge \neg L2 \wedge \neg L3) \vee (L2 \wedge \neg L1 \wedge \neg L3) \vee (L3 \wedge \neg L1 \wedge \neg L2)$.
  - $(S1 \wedge \neg S2 \wedge \neg S3) \vee (S2 \wedge \neg S1 \wedge \neg S3) \vee (S3 \wedge \neg S1 \wedge \neg S2)$.
- Is there a way to give truth values to propositions such that all these constraints are satisfied?

Formalizing Exercise 1.4

- Propositions $S1, S2, S3$ represent whether the sign of room 1, 2, 3 (resp.) is true.
- Propositions $L1, L2, L3$ represent Lady in room 1, 2, 3 resp.
- The statements given:
  - $S1 \leftrightarrow \neg L1$, $S2 \leftrightarrow \neg L2$, $S3 \leftrightarrow L2$.
  - $(L1 \wedge \neg L2 \wedge \neg L3) \vee (L2 \wedge \neg L1 \wedge \neg L3) \vee (L3 \wedge \neg L1 \wedge \neg L2)$.
  - $(S1 \wedge \neg S2 \wedge \neg S3) \vee (S2 \wedge \neg S1 \wedge \neg S3) \vee (S3 \wedge \neg S1 \wedge \neg S2)$.
- Is there a way to give truth values to propositions such that all these constraints are satisfied?
- **Bonus:** Can you encode this using only 3 propositions instead of 6?

Bonus: Encoding with only 3 propositions

Intuition

$S1, S2, S3$ were never really needed – they're just names for what the signs say.

Bonus: Encoding with only 3 propositions

Intuition

$S1, S2, S3$ were never really needed – they're just names for what the signs say. We can substitute their definitions directly wherever they're used.

Bonus: Encoding with only 3 propositions

Intuition

$S1, S2, S3$ were never really needed – they're just names for what the signs say. We can substitute their definitions directly wherever they're used.

- exactly one of $\neg L1,\ \neg L2,\ L2$ holds (exactly one sign is true)

$$(\neg L1 \land L2 \land \neg L2) \lor (L1 \land \neg L2 \land \neg L2) \lor (L1 \land L2 \land L2)$$

Bonus: Encoding with only 3 propositions

Intuition

$S1, S2, S3$ were never really needed – they're just names for what the signs say. We can substitute their definitions directly wherever they're used.

- exactly one of $\neg L1,\ \neg L2,\ L2$ holds (exactly one sign is true)

$$(\neg L1 \wedge L2 \wedge \neg L2) \vee (L1 \wedge \neg L2 \wedge \neg L2) \vee (L1 \wedge L2 \wedge L2)$$

- exactly one of $L1, L2, L3$ holds (Lady is in exactly one room)

$$(L1 \wedge \neg L2 \wedge \neg L3) \vee (L2 \wedge \neg L1 \wedge \neg L3) \vee (L3 \wedge \neg L1 \wedge \neg L2)$$

Bonus: Encoding with only 3 propositions

Intuition

$S1, S2, S3$ were never really needed – they're just names for what the signs say. We can substitute their definitions directly wherever they're used.

- exactly one of $\neg L1,\ \neg L2,\ L2$ holds (exactly one sign is true)

$$(\neg L1 \wedge L2 \wedge \neg L2) \vee (L1 \wedge \neg L2 \wedge \neg L2) \vee (L1 \wedge L2 \wedge L2)$$

- exactly one of $L1, L2, L3$ holds (Lady is in exactly one room)

$$(L1 \wedge \neg L2 \wedge \neg L3) \vee (L2 \wedge \neg L1 \wedge \neg L3) \vee (L3 \wedge \neg L1 \wedge \neg L2)$$

Another Bonus: Can we encode it with just two propositions?