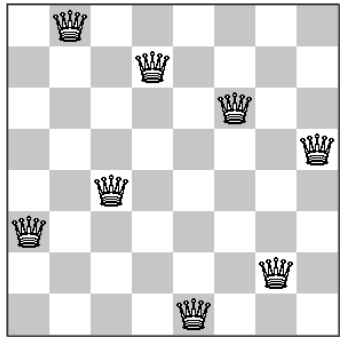
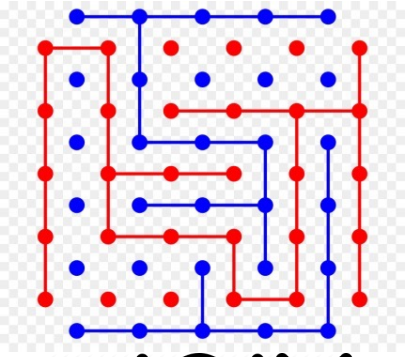
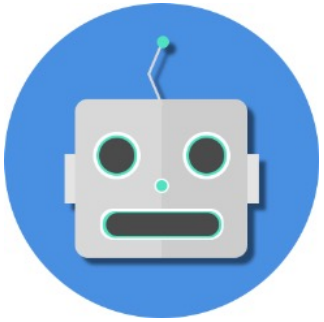
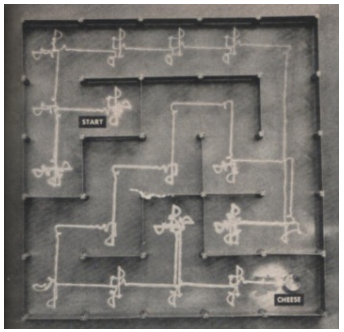
Artificial Intelligence:

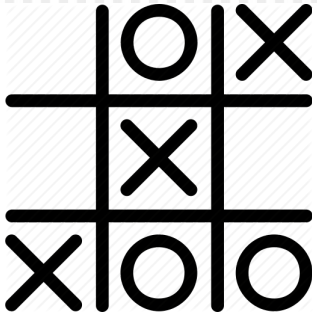
Past, Present and Future











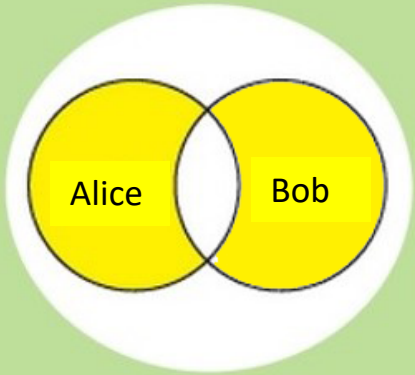
Chee Wei Tan

Recall: Boolean Logic and Boolean Algebra

- This is the calculus concerned with true or false statements connected by the following three binary relations:

- These three logic relations can build the entire universe of logic
- A Venn diagram is a convenient tool to understand logic

Recall: Boolean Logic and Boolean Algebra



EXCLUSIVE OR

- Data is true if and only if one operand is true and the other is false

Example of Exclusive–Or Statement that is valid

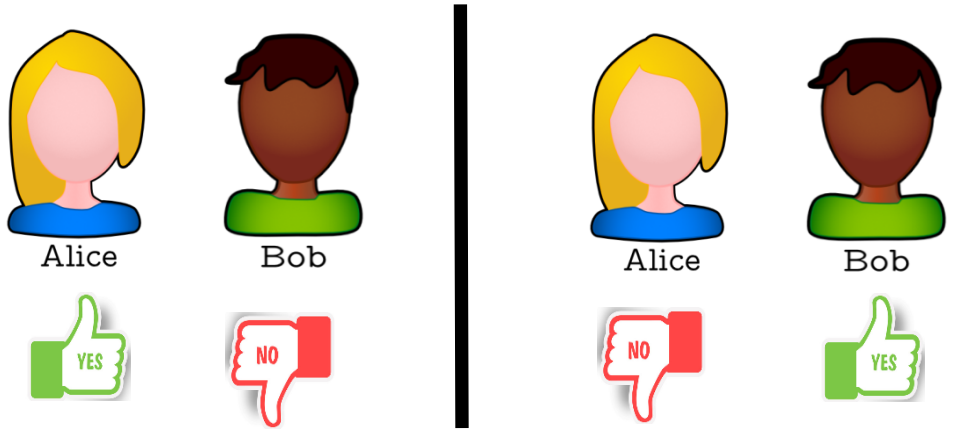
Either I vote for Alice or I vote for Bob in the election with a single vote.

I did not vote for Alice and I did not vote for Bob. (False)

I voted for Alice and I did not vote for Bob. (True)

I did not vote for Alice and I voted for Bob. (True)

I voted for Alice and I voted for Bob. (False)



Recall: Boolean Logic and Boolean Algebra

BOOLEAN SET ALGEBRA	PROPOSITIONAL CALCULUS
U (UNIVERSAL SET)	T (TRUE)
ϕ (NULL SET)	F (FALSE)
a, b, c, . . . (SETS, SUBSETS, ELEMENTS)	*p, q, r,* . . . (PROPOSITIONS)
$a \cup b$ (UNION: ALL OF *a* AND b)	$p \vee q$ (DISJUNCTION: EITHER *p* ALONE OR *q* ALONE, OR BOTH, ARE TRUE.)
$a \cap b$ (INTERSECTION: WHAT *a* AND *b* HAVE IN COMMON)	$p \bullet q$ (CONJUNCTION: BOTH *p* AND *q* ARE TRUE.)
$a = b$ (IDENTITY: *a* AND *b* ARE THE SAME SET.)	$p \equiv q$ (EQUIVALENCE: IF AND ONLY IF *p* IS TRUE, THEN *q* IS TRUE.)
a' (COMPLEMENT: ALL OF **U** THAT IS NOT *a*)	$\sim p$ (NEGATION: *p* IS FALSE.)
$a \in b$ (INCLUSION: *a* IS A MEMBER OF *b*.)	$p \supset q$ (IMPLICATION: IF *p* IS TRUE, *q* IS TRUE.)

The symbols of propositional calculus correspond to symbols for the Boolean set algebra

Tarski's Logic and Philosophy

TRUTH AND PROOF

The antinomy of the liar, a basic obstacle to an adequate definition of truth in natural languages, reappears in formalized languages as a constructive argument showing not all true sentences can be proved

by Alfred Tarski

The subject of this article is an old one. It has been frequently discussed in modern logical and philosophical literature, and it would not be easy to contribute anything original to the discussion. To many readers, I am afraid, none of the ideas put forward in the article will appear essentially novel; nonetheless, I hope they may find some interest in the way the material has been arranged and knitted together.

As the title indicates, I wish to discuss here two different though related notions: the notion of truth and the notion of proof. Actually the article is divided into three sections. The first section is concerned exclusively with the notion of truth, the second deals primarily with the notion of proof, and the third is a discussion of the relationship between these two notions.

The Notion of Truth

The task of explaining the meaning of the term "true" will be interpreted here in a restricted way. The notion of truth occurs in many different contexts, and there are several distinct categories of objects to which the term "true" is applied. In a psychological discussion one might speak of true emotions as well as true beliefs; in a discourse from the domain of esthetics the inner truth of an object of art might be analyzed. In this article, however, we are interested only in what might be called the logical notion of truth. More specifically, we concern ourselves exclusively with the meaning of the term "true" when this term is used to refer to sentences. Presumably this was the original use of the term "true" in human language. Sentences are treated here as linguistic objects, as certain strings of sounds or written signs. (Of course, not every such string is a sentence.) Moreover, when speaking of sentences, we shall always have in mind what are called in grammar declarative sentences, and not interrogative or imperative sentences.

Whenever one explains the meaning of any term drawn from everyday language, he should bear in mind that the goal and the logical status of such an explanation may vary from one case to another. For instance, the explanation may be intended as an account of the actual use of the term involved, and is thus subject to questioning whether the account is indeed correct. At some other time an explanation may be of a normative nature, that is, it may be offered as a suggestion that the term be used in some definite way, without claiming that the suggestion conforms to the way in which the term is actually used; such an explanation can be evaluated, for instance, from the point of view of its usefulness but not of its correctness. Some further alternatives could also be listed.

The explanation we wish to give in the present case is, to an extent, of mixed character. What will be offered can be treated in principle as a suggestion for a definite way of using the term "true", but the offering will be accompanied by the belief that it is in agreement with the prevailing usage of this term in everyday language.

Our understanding of the notion of truth seems to agree essentially with various explanations of this notion that have been given in philosophical literature. What may be the earliest explanation can be found in Aristotle's *Metaphysics:*

> **To say of what is that it is not, or of what is not that it is, is false, while to say of what is that it is, or of what is not that it is not, is true.**

Here and in the subsequent discussion the word "false" means the same as the expression "not true" and can be replaced by the latter.

The intuitive content of Aristotle's formulation appears to be rather clear. Nevertheless, the formulation leaves much to be desired from the point of view of precision and formal correctness. For one thing, it is not general enough; it refers only to sentences that "say" about something "that it is" or "that it is not"; in most cases it would hardly be possible to cast a sentence in this mold without slanting the sense of the sentence and forcing the spirit of the language. This is perhaps one of the reasons why in modern philosophy various substitutes for the Aristotelian formulation have been offered. As examples we quote the following:

> **A sentence is true if it denotes the existing state of affairs.**
>
> **The truth of a sentence consists in its conformity with (or correspondence to) the reality.**

Due to the use of technical philosophical terms these formulations have undoubtedly a very "scholarly" sound. Nonetheless, it is my feeling that the new formulations, when analyzed more closely, prove to be less clear and unequivocal than the one put forward by Aristotle.

The conception of truth that found its expression in the Aristotelian formulation (and in related formulations of more recent origin) is usually referred to as the *classical*, or *semantic conception of truth*. By semantics we mean the part of logic that, loosely speaking, discusses the relations between linguistic objects (such as sentences) and what is expressed by these objects. The semantic

63

Scientific American 220: 63-77, 1969

Alfred Tarski (1901 – 1983), was a Polish-American logician and mathematician who taught at the University of California, Berkeley. Educated in Poland at the University of Warsaw, and a member of the Lwów–Warsaw school of logic and the Warsaw school of mathematics, he is widely considered as one of the greatest logicians of the twentieth century (often regarded as second only to Gödel), and thus as one of the greatest logicians of all time.

https://en.wikipedia.org/wiki/Alfred_Tarski

Truth Table

A truth table is used to determine if a compound statement is true or false, usually used in mathematical logic specifically in connection with Boolean algebra and propositional calculus.

In a truth table, each statement is typically represented by a letter or variable, like p, q, or r, and each statement also has its own corresponding column in the truth table that enumerates all possible truth values.

A	B	A and B	A or B
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

T=true F=false

Truth Table of Contradiction

A contradiction is a compound statement that is always false.

P: She loves me and she loves me not.

- She loves me.(P)
- And(∧)
- she loves me not. (¬*P*)

P	1 P	3 ∧	2 ¬*P*
T	T	F	F
F	F	F	T

Anyway she will not love me.😢

Truth Table of Tautology

A tautology is a compound statement that is always true.

P: She loves me or she loves me not.

- She loves me.(P)
- Or(V)
- she loves me not. (¬*P*)

P	1 P	3 V	2 ¬*P*
T	T	T	F
F	F	T	T

Anyway she will love me.😆

Truth Table of Logical Conjunction

Logical conjunction is an operation on two logical values, typically the values of two propositions, producing a value of True if and only if both of its operands are true.

P	Q	P	∧	Q
T	T	T	T	T
T	F	T	F	F
F	T	F	F	T
F	F	F	F	F

The symbol for this is Λ (whenever you see Λ , just read 'and'). When two simple sentences, p and q, are joined in a conjunction statement, the conjunction is expressed symbolically as p Λ q.

Example 1:

Simple Sentences	Compound Sentence: Conjunction
p : Joe eats fries. q : Maria drinks soda.	p ∧ q : Joe eats fries, and Maria drinks soda.

P	Q	P	∧	Q
T	T	T	T	T
T	F	T	F	F
F	T	F	F	T
F	F	F	F	F

If “Joe eats fries” is true and “Maria drinks soda” is true, then “Joe eats fries, and Maria drinks soda” will be true.

Truth Table of Logical Disjunction

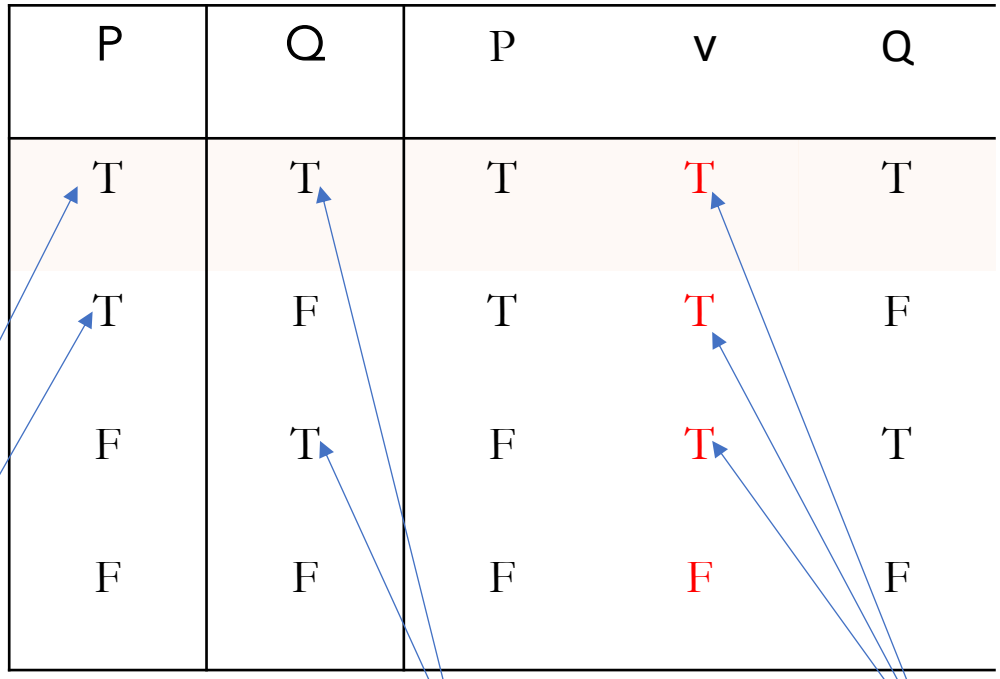
Logical disjunction is an operation on two logical values, typically the values of two propositions, producing a value of false if and only if both of its operands are false.

P	Q	P	v	Q
T	T	T	T	T
T	F	T	T	F
F	T	F	T	T
F	F	F	F	F

In logic, a disjunction is a compound sentence formed using the word `or' to join two simple sentences. The symbol for this is v (whenever you see v read 'or'). When two simple sentences, p and q, are joined in a disjunction statement, the disjunction is expressed symbolically as p v q.

Example 2:

Simple Sentences	Compound Sentence: Disjunction
p: The clock is slow. q: The time is correct.	p v q: The clock is slow, ***or*** the time is correct.



P	Q	P	v	Q
T	T	T	T	T
T	F	T	T	F
F	T	F	T	T
F	F	F	F	F

If “The clock is slow” is true or “The time is correct” is true, then “The clock is slow, ***or*** the time is correct.” will be true.

Truth Table of Negation

Classical negation is an operation on one logical value, typically the value of a proposition, that produces a value of true when its operand is false and a value of false when its operand is true.

P	¬
T	F
F	T

Indicates the opposite, usually employing the word `not'.

The symbol to indicate negation is : ~ or ¬

Example 3:

Original Statement	Negation of Statement
Today is Monday.	Today is not Monday.
That was fun.	That was not fun.

P	¬
T	F
F	T

If “Today is Monday” is true, the negation of “Today is Monday” statement is false.
If “That was fun” is true, the negation of “That was fun” statement is false.

Truth Table of Material Conditional

The compound p→q is false if and only if p is true and q is false. Otherwise, p→q is true if and only if either p is false or q is true (or both).

P	Q	P	→	Q
T	T	T	T	T
T	F	T	F	F
F	T	F	T	T
F	F	F	T	F

In logic, a conditional statement is compound sentence that is usually expressed with the key words 'If....then...'. Using the variables p and q to represent two simple sentences, the conditional "If p then q" is expressed symbolically as p → q or ¬p ∨ q

Example 4:

Simple Sentences	Compound Sentence: Conditional
p: win the next election q: reduce the price of petrol by 20%	p → q : ***If we win the next election, we will immediately reduce the price of petrol by 20%***

So according to their statements, there will be 2 situations.

Situation 1: If they won the elections
In the event that they really did reduce the petrol price, then the statement is true. But if they break their promise and the petrol price isn't reduced, then the statement is false.

Situation 2: If they lost in the elections
Since they didn't promise anything about what they will do if they lost, whatever that happens, be it that the price of petrol increases, remains the same or reduces, doesn't make the statement false, and so therefore, for either case, the statement is true.

P	Q	P	→	Q
T	T	T	T	T
T	F	T	F	F
F	T	F	T	T
F	F	F	T	F

Truth Table of Converse Implication

Notice that ‘p and q’ is true if both p and q are true. Also, ‘p or q’ is true if either one of p or q is true. The converse p ← q is a little tricky - it is false only if p is false, and q is true.

P	Q	P	→	Q	P	←	Q
T	T	T	T	T	T	T	T
T	F	T	F	F	T	T	F
F	T	F	T	T	F	F	T
F	F	F	T	F	F	T	F

Converse implication is logically equivalent to the disjunction of P ∨ ¬Q

Truth Table of Logical NOR

The NOR operation is a logical operation on two logical values, typically the values of two propositions, that produces a value of true if and only if both operands are false. In other words, it produces a value of false if and only if at least one operand is true.

P	Q	P	↓	Q
T	T	T	F	T
T	F	T	F	F
F	T	F	F	T
F	F	F	T	F

The NOR Function is sometimes known as the Peirce Function and is denoted by a downward-pointing arrow operator as shown, P NOR Q = P↓Q = ¬(P ∨ Q)

Example 5:

Simple Sentences	**Compound Sentence: Conditional**
p: Door p is open q: Door q is open	p ↓ q : p is closed and q is closed, nobody can leave here.

P	Q	P	↓	Q
T	T	T	F	T
T	F	T	F	F
F	T	F	F	T
F	F	F	T	F

Only p and q are false(closed) , then “p and q are closed, nobody can leave here” is true.

Truth Table of Logical NAND

The NAND operation is a logical operation on two logical values. It produces a value of true, if and only if at least one of the propositions is false.

P	Q	P	↑	Q
T	T	T	F	T
T	F	T	T	F
F	T	F	T	T
F	F	F	T	F

Logical NAND is logically equivalent to the disjunction of $\neg(P \land Q)$

Truth Table of Material Nonimplication

Material nonimplication is the negation of material implication. That is to say that for any two propositions P and Q, the material nonimplication from P to Q is true if and only if the negation of the material implication from P to Q is true.

P	Q	P	$\not\rightarrow$	Q
T	T	T	F	T
T	F	T	T	F
F	T	F	F	T
F	F	F	F	F

Material nonimplication may be defined as the negation of material implication, and is logically equivalent to ¬(P $\rightarrow$ Q).

Truth Table of Converse Nonimplication

The converse nonimplication is a logical connective which is the negation of converse implication (equivalently, the negation of the converse of implication).

P	Q	P	$\not\leftarrow$	Q
T	T	T	F	T
T	F	T	F	F
F	T	F	T	T
F	F	F	F	F

Converse nonimplication is denoted by $p \not\leftarrow q$, and is logically equivalent to $\neg(P \leftarrow Q)$.

Truth Table of Exclusive OR

Exclusive-Or or **Exclusive Disjunction** is a logical operation that outputs true only when inputs differ (one is true and the other is false)

P	Q	P	$\oplus$	Q
T	T	T	F	T
T	F	T	T	F
F	T	F	T	T
F	F	F	F	F

Exclusive disjunction essentially means 'either one, but not both nor none'. In other words, the statement is true if and only if one is true and the other is false.

Note that $p \oplus q$ is equivalent to $(p \vee q) \wedge (\neg p \vee \neg q)$ and $(\neg p \wedge q) \vee (p \wedge \neg q)$

Truth Table of Logical Biconditional

The logical biconditional (sometimes known as the material biconditional) is the logical connective of two statements asserting "P if and only if Q", where P is an antecedent and Q is a consequent. This is often abbreviated as "P iff Q" when used in mathematical proofs.

P	Q	P	↔	Q
T	T	T	T	T
T	F	T	F	F
F	T	F	F	T
F	F	F	T	F

What is the negation of the logical biconditional?

Challenge 1

Express each statement symbolically, and then state the truth value and truth table of each mathematical statement.

Statement: Christmas is a holiday and we do not work on Christmas.

Challenge 2

Express each statement symbolically, and then state the truth value and truth table of each mathematical statement.

Statement: If we go to school on Christmas, then we work on Christmas.

Challenge 3

Express each statement symbolically, and then state the truth value and truth table of each mathematical statement.

Statement: If we do not go to school on Christmas and Christmas is a public holiday, then we do not work on Christmas.

Answer 1

Let b represent ”Christmas is a holiday."
Let c represent "We work on Christmas.”

Statement in symbols	Truth value of parts	Truth value of entire statement
b Λ ~c	T Λ ~F = T Λ T	True statement

b	**c**	b	**Λ**	**~c**
T	T	T	F	F
T	F	T	T	T
F	T	F	F	F
F	F	F	F	T

Answer 2

Let a represent "We go to school on Christmas."
Let c represent "We work on Christmas."

Statement in symbols	**Truth value of parts**	**Truth value of entire statement**
$a \rightarrow c$	$F \rightarrow F$	True statement

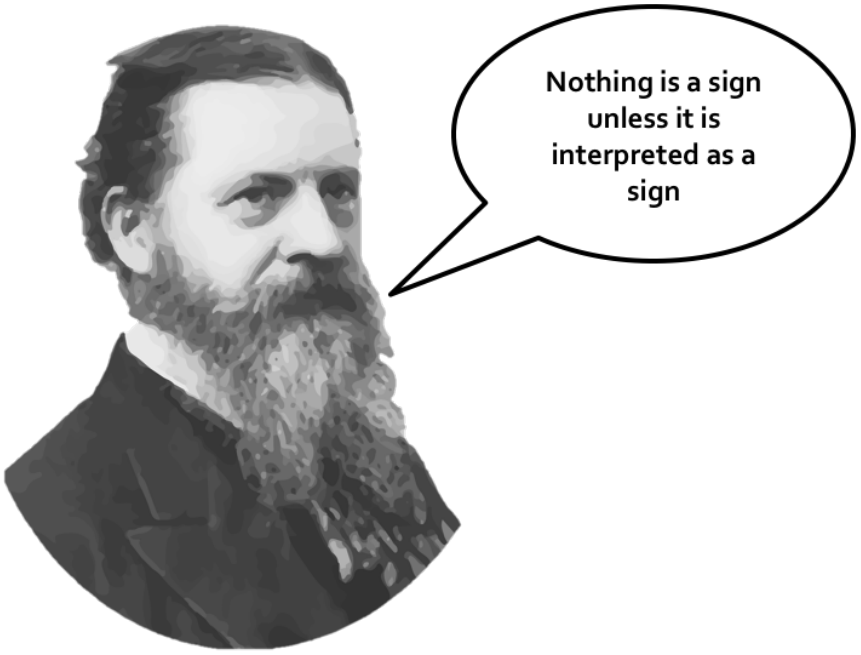
a	**c**	**a**	$\rightarrow$	**c**
T	T	T	T	T
T	F	T	F	F
F	T	F	T	T
F	F	F	T	F

Answer 3

Statement in symbols	Truth value of parts	Truth value of entire statement
$(\sim a \wedge b) \rightarrow \sim c$	$(\sim F \wedge T) \rightarrow \sim F$ $(T \wedge T) \rightarrow \sim F$ $T \rightarrow \sim F$ $T \rightarrow T$	True statement

a	b	c	~a	Λ	b	(~a Λ b)	→	~c
T	T	T	F	F	T	F	T	F
T	T	F	F	F	F	F	T	T
T	F	T	F	F	T	F	T	F
T	F	F	F	F	F	F	T	T
F	T	T	T	T	T	T	F	F
F	T	F	T	F	F	F	T	T
F	F	T	T	T	T	T	F	F
F	F	F	T	F	F	F	T	T

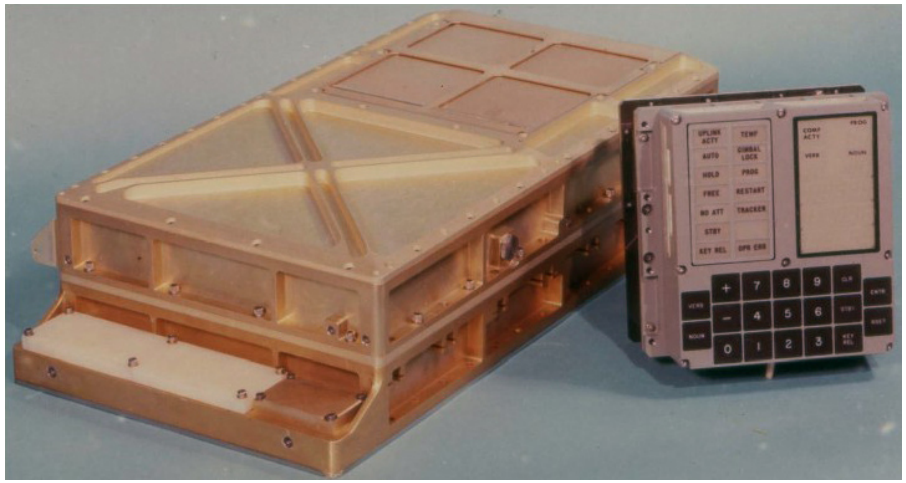
Charles Sanders Peirce

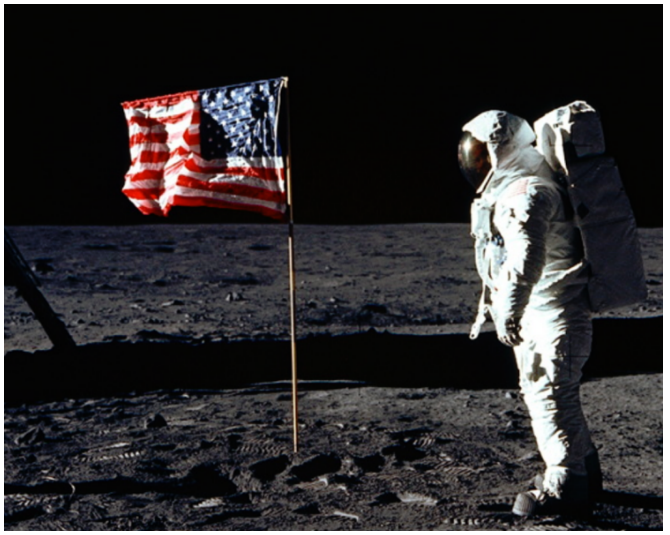


The Peirce function (Logical NOR) is due to Charles Sanders Peirce (1839-1914) who was an American mathematician, philosopher and logician. He is the founder of pragmatism (the attribute of accepting the facts of life and favouring practicality and literal truth). He was regarded as the most original thinker and greatest logician of his time.

Peirce defined the concept of abductive reasoning that is a logical inference which starts with a set of observations and then seeks to find the simplest and most likely explanation for the observations. The abductive reasoning is important in artificial intelligence.

Application: Apollo Guidance Computer





The Peirce function (Logical NOR) can be viewed as a building block for the three logical connectives (AND, OR, NOT). The computer in the spacecraft Apollo 11 for moon-landing mission was made up entirely of NOR circuits.

Watch Science Channel Video – Moon Machines Navigation Computer on the story of MIT's work on the Apollo Guidance Computer: https://www.youtube.com/watch?v=xQ1O0XR_cA0

Aristotle's Syllogism

Syllogism is a logical argument that applies deductive reasoning to arrive at a conclusion based on two or more propositions that are asserted, or assumed, to be true

Syllogism

Classical Syllogism Structure

P1: All the middle term(M) are the predicate(P).

P2:The subject(S) is the middle term(M).

P3:Therefore, the subject(S) is the predicate(P).

Classical Syllogism Example

P1: All cars are a form of transportation.

P2: All Honda Accords are cars.

P3: All Honda Accords are a form of transportation.

All M are P,

All S are M

All S are P

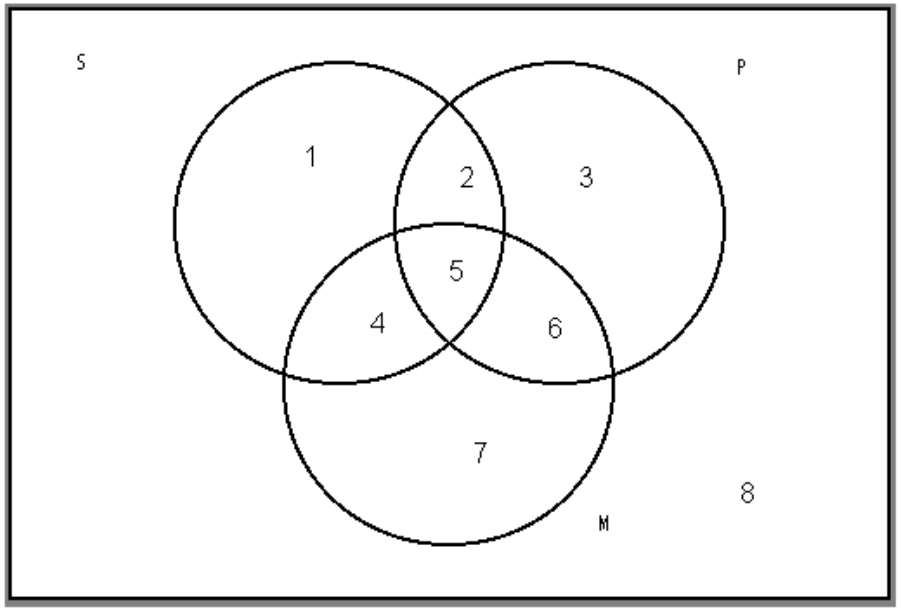
Venn Diagram of Syllogism

To test the validity of a categorical syllogism, one can use the method of Venn diagrams. Since a categorical syllogism has three terms, we need a Venn diagram using three intersecting circles, one representing each of the three terms in a categorical syllogism. This three-term Venn diagram has eight regions.

All M are P,

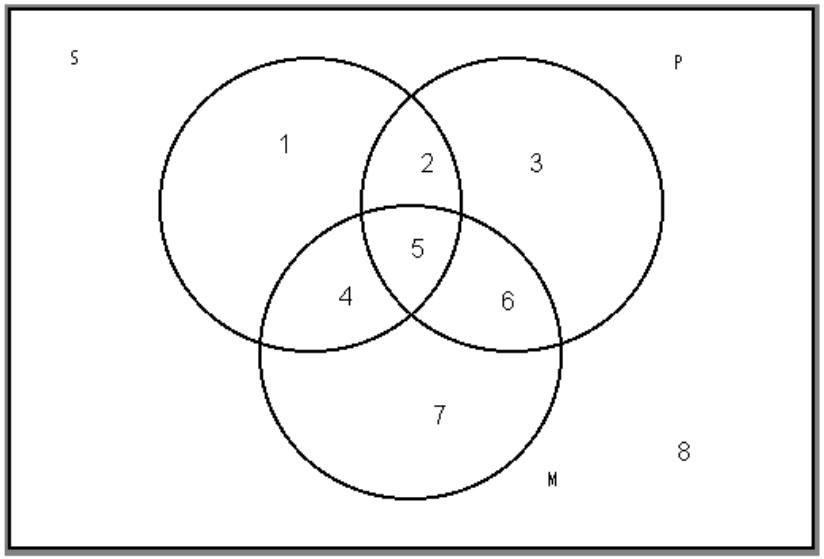
All S are M

All S are P



Validity Testing with Venn Diagram

The following truth table gives the extension of the predicates in the various regions of the diagram:

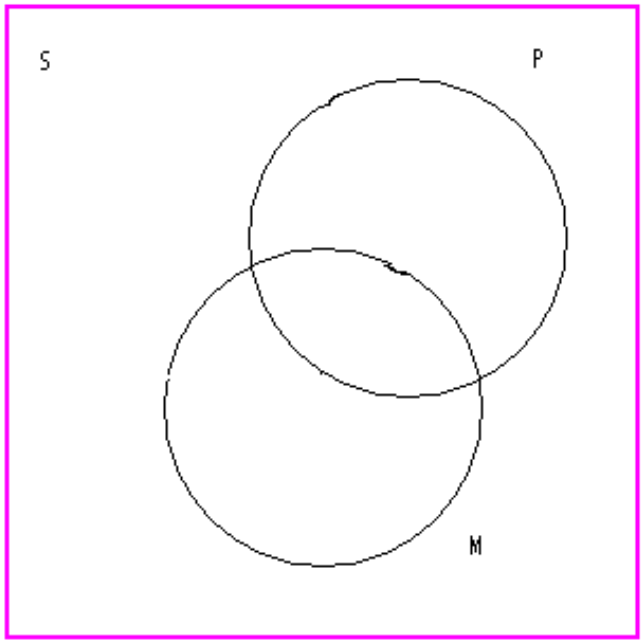


Region	S (Minor Term)	P (Major Term)	M (Middle Term)
1	yes	no	no
2	yes	yes	no
3	no	yes	no
4	yes	no	yes
5	yes	yes	yes
6	no	yes	yes
7	no	no	yes
8	no	no	no

Example 1

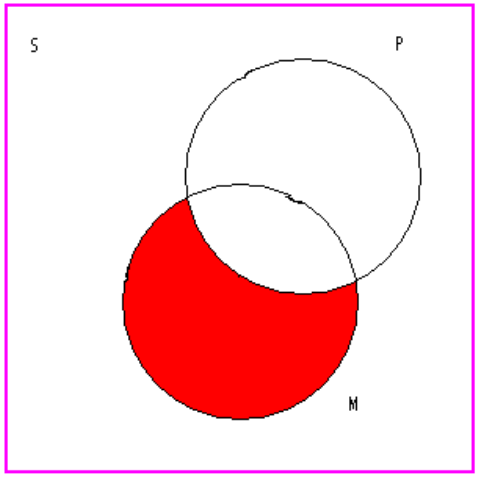
Consider the following argument:
All Greeks are mortal. (All M are P)
All Athenians are Greek. (All S are M)
So, all Athenians are mortal. (All S are P)

Next, diagram each of the premises. When doing this, act as if there are only 2 relevant circles. Begin with the first premise. In our example you need to diagram the proposition "All M are P". Ignoring for a moment the circle representing the minor term, your diagram should look like this:

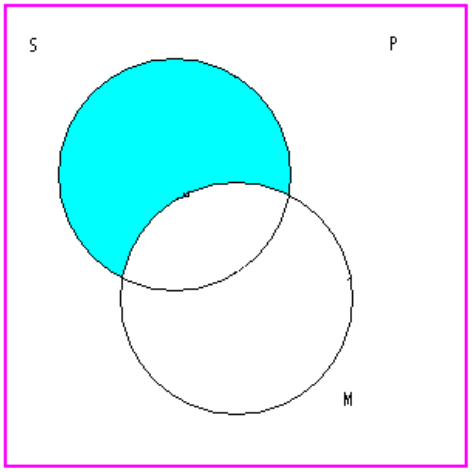


Example 1

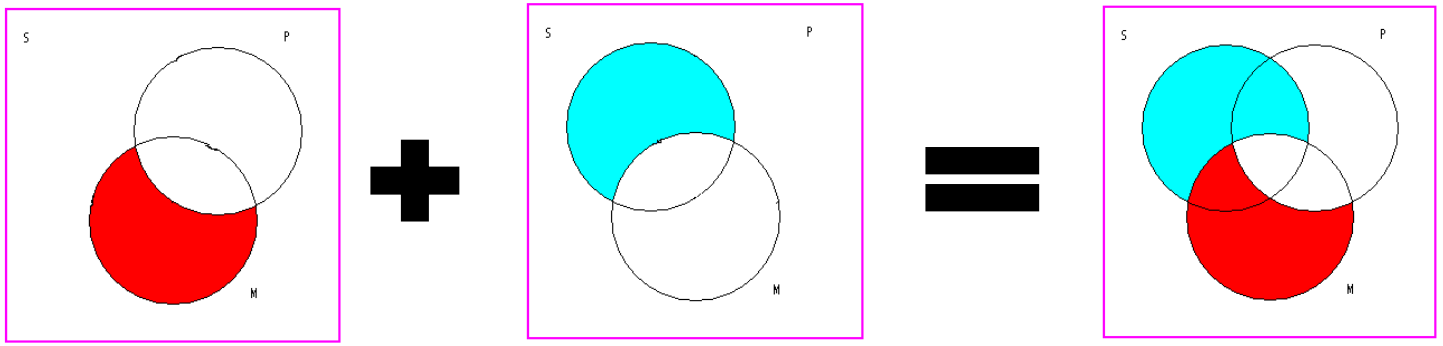
1. Following the standard conventions we get:



2. Diagram the second premise--"All S are M"-- to get:



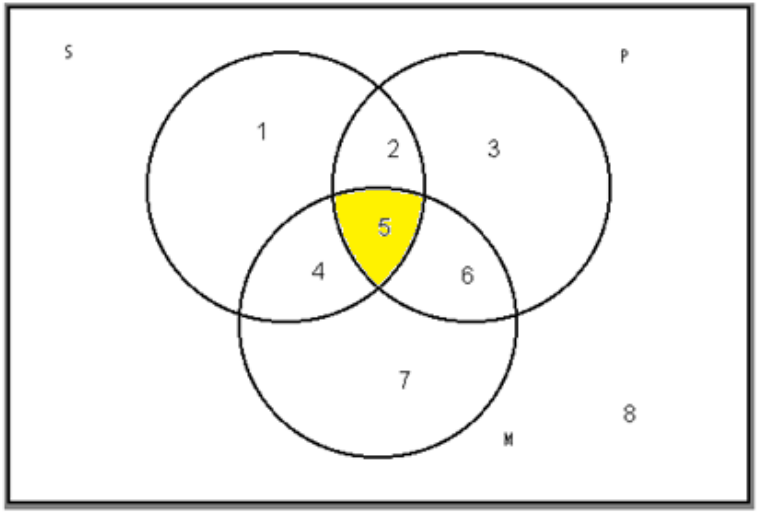
3. Overlap the diagrams to get:



Example 1

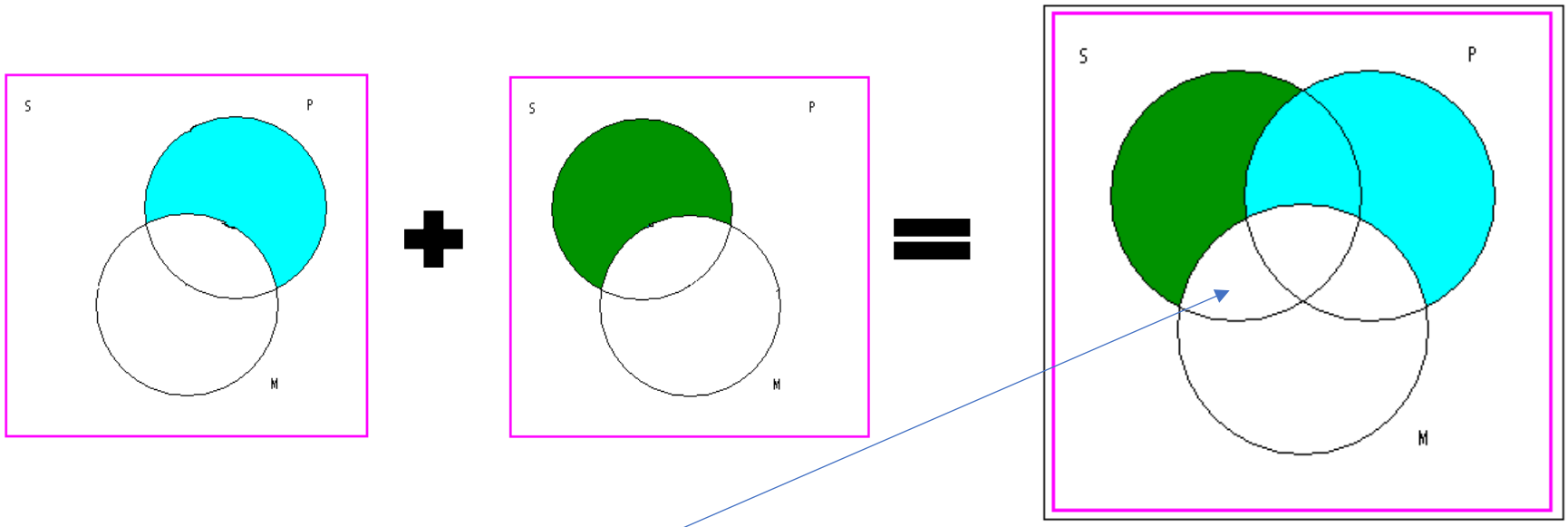
Question: Does this diagram express the informational content of the conclusion of the argument?

Answer: Yes, all of the S's that remain are in region 5 and everything in region 5 is an S, an M, and a P. Since all the S's are in region 5, all the S's are P's and the argument is VALID.



Example 2

Consider the following argument:
All mathematicians are rational. (All P are M)
All philosophers are rational. (All S are M)
SO, all philosophers are mathematicians. (All S are P)

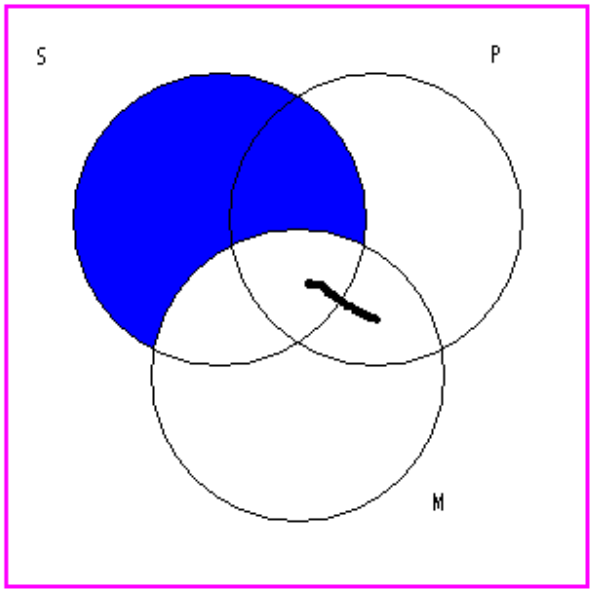


Does this diagram express the informational content of the conclusion "All S are P"?

NO. Region 4 of the diagram is not shaded (not empty) so it is possible that there is an S that is not a P. Accordingly, the argument is NOT VALID.

Example 3

Consider the following argument:
All philosophers are logical. (All P are M)
Some physicists are logical. (All S are M)
So, Some philosophers are physicists. (Some P are S)

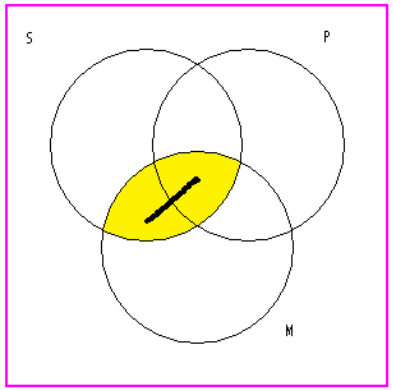


The bar that crosses from region 5 into region 6 indicates that the argument is **NOT VALID**.
All that we can be certain of is that there is either an SPM (region 5) or a PM non-S (region 6), but we don't know which. Since we don't know which, the conclusion does not follow logically from the premises.

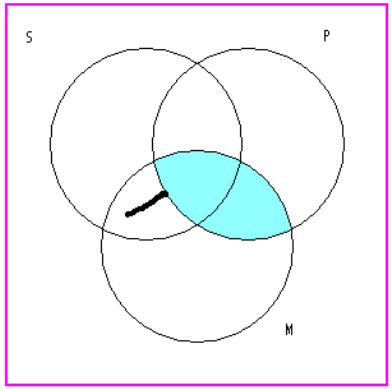
Example 4

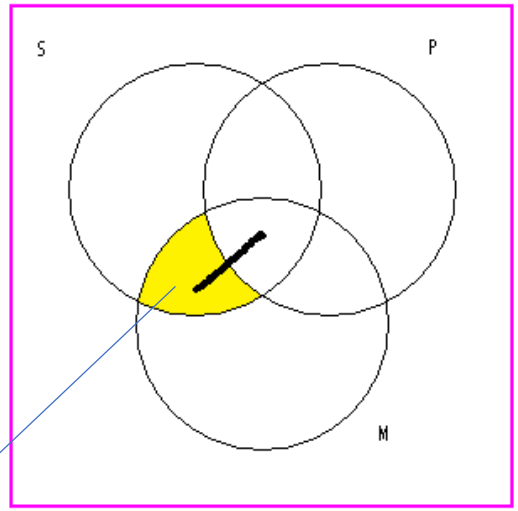
Consider the following argument:
Some physicists are logical. (Some S are M)
No philosophers are logical. (No P are M)
So, some physicists are not philosophers. (Some S are not P)

Some S are M



No P are M



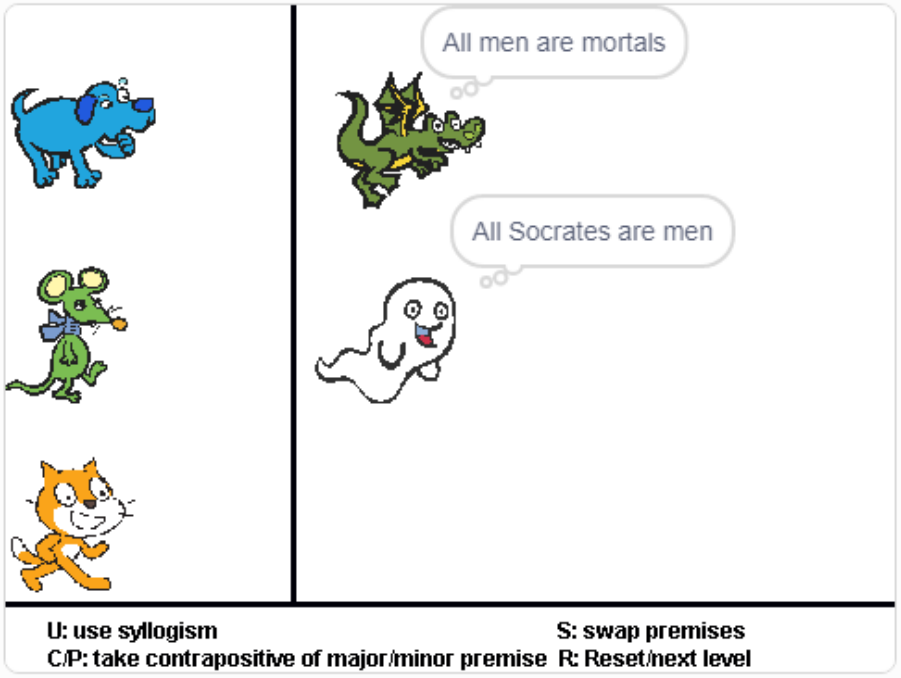


Does this diagram express the informational content of the conclusion "Some S are not P"? Yes, the S that is an M in region 4 is a non-P. So the argument is valid.

Terence Tao's Syllogism Game

Terence Tao created a logical puzzle game in 2012 to gamify classical syllogism and Lewis Carrol's logic puzzles. To win the game, players search for a syllogism by identifying which premise is logically linked to another. Venn diagram tools in the previous slides can be helpful. Play the game here:

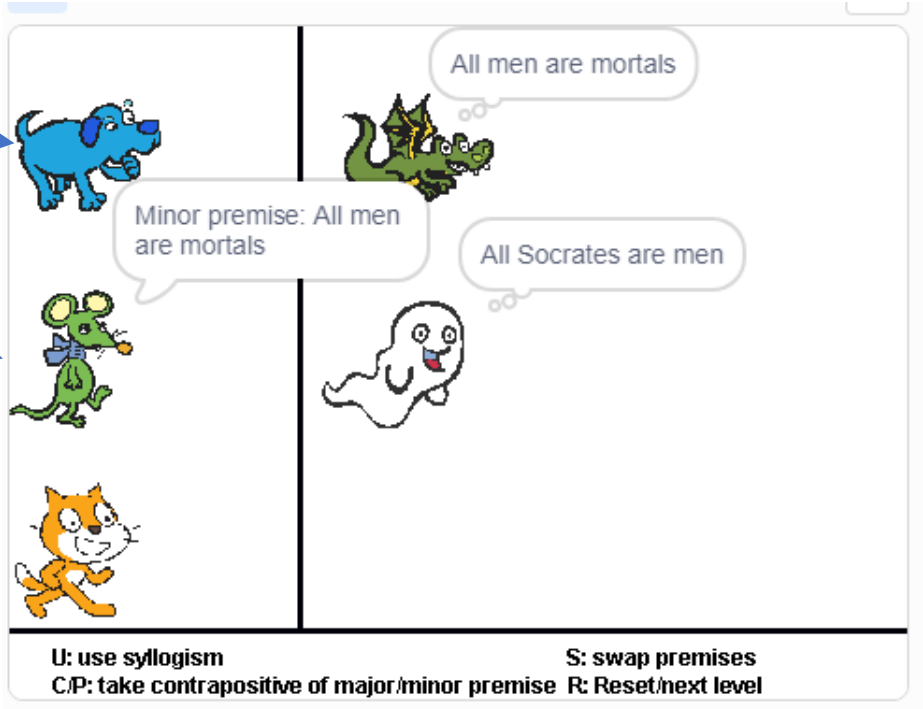
https://scratch.mit.edu/projects/2486639/



Terence Tao's Syllogism Game

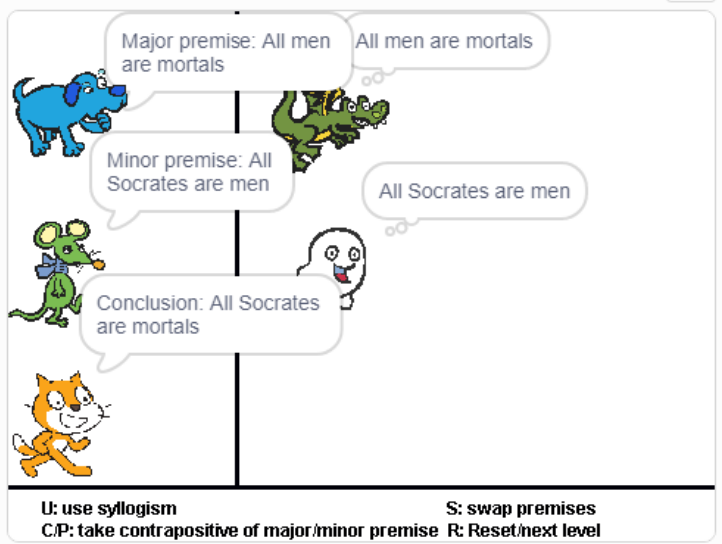
1. Select a major and a minor premise by cycling through the options using top two icons on the left.

2. Click the cat icon once the player is satisfied that a useful syllogism has been found.

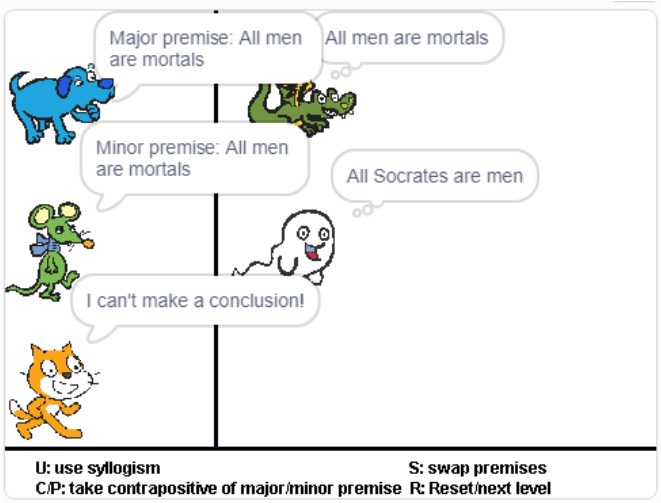


Terence Tao's Syllogism Game

If the selected major and minor premises are correctly chosen, the cat icon will give the conclusion of the valid syllogism.

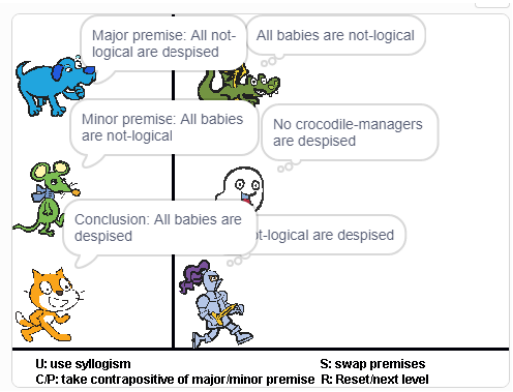


If the selected major premise and minor premise do not form a valid syllogism, the cat icon will answer that a conclusion cannot be made.

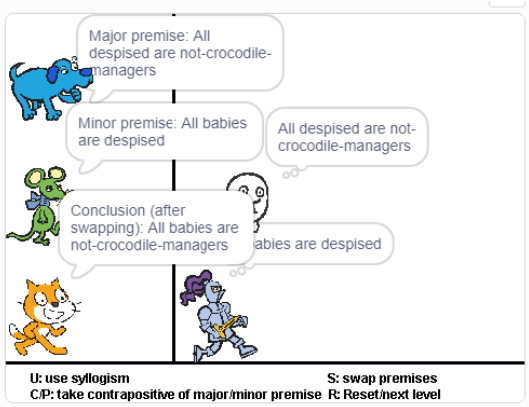


Terence Tao's Syllogism Game

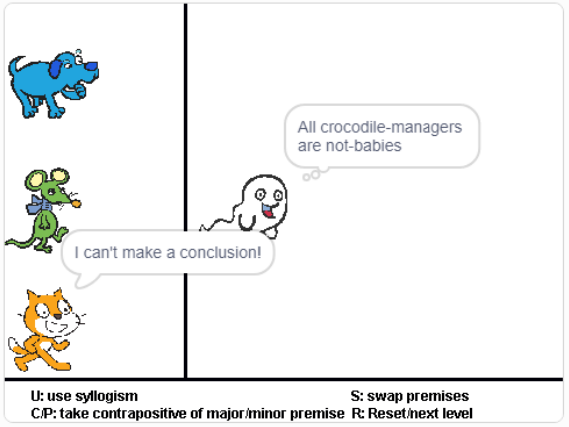
Once the conclusion is obtained, press ‘U’ to replace the icon on the right-hand side with the major premise’s statement and the one with the minor premise’s statement will disappear. When there is only one icon remaining on the right-hand side, player can press ‘R’ to move to the next level.



Press ‘U’



Press ‘U’



Press ‘R’

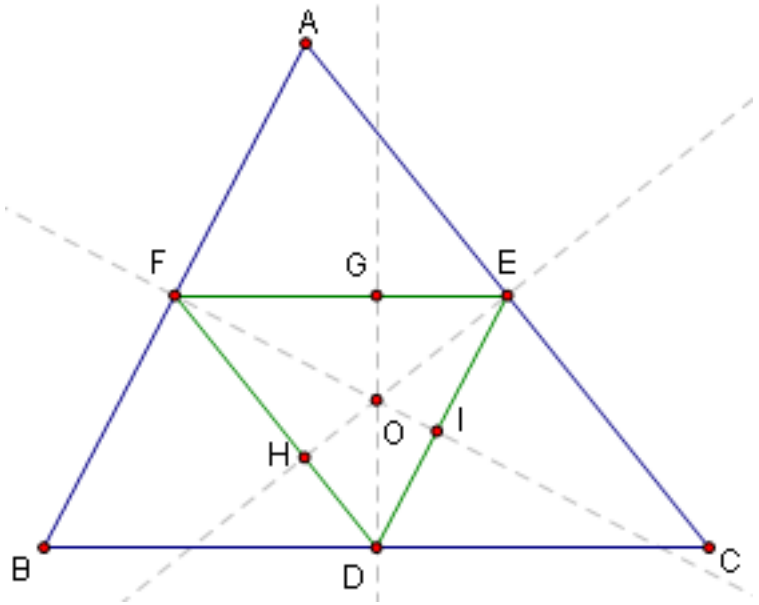
Mathematical Proofs

- A mathematical proof is an inferential argument for a *mathematical statement*, showing that the stated *assumptions* logically guarantee the *conclusion*.
- There are many proof techniques: Proof by contradiction, direct proof (axiomatic method) etc
- Euclid (300 BC) revolutionized mathematical proofs when he introduced the axiomatic method in his book *Elements*, where theorems are deduced starting with axioms (similarly to syllogistic reasoning)
- In mathematics, we often wish to establish the equivalence of two propositions P and Q, and we do it as follows using the biconditional:
  - P is equivalent to Q *if and only if* P ↔ Q is a tautology

Automated Theorem Proving

- Can a proof for a *true mathematical statement* always be found by using the axiomatic method?
  - Recall Godel's Incompleteness Theorem and what Tarski said in his "Truth and Proof" Scientific American article: *not all true statements can be proved.*
- The invention of computers enables *automated theorem proving* which is an important AI application
  - Geometry Theorem Prover (IBM)
  - Takski's High School Algebra Problem
  - Four Color Theorem
  - Wang's Algorithm
  - Robbin's Algebra Conjecture
  - Information Theoretic Inequality Prover

Geometry Theorem Prover



Can you prove that the *perpendicular bisectors* of all three sides of a triangle intersect in one point?

In early spring, 1959, Gelernter and Rochester (one of the ten at the 1956 Dartmouth Workshop) at IBM programmed an IBM 704 computer to prove its first theorem in elementary Euclidean plane geometry. Since that time, the geometry theorem-proving machine (a particular mode of the IBM 704 computer) has found solutions to a large number of problems taken from high school textbooks.

Gelernter, H. L., Rochester, N., "Intelligent Behavior in Problem-Solving Machines", IBM Journal of Research and Development, Oct. 1958;

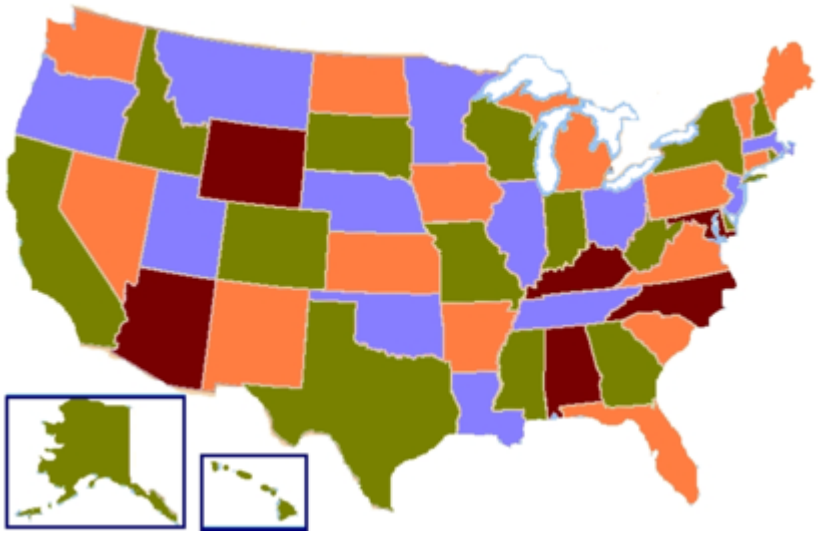
Tarski's High School Algebra Problem

1. $x + y = y + x$
2. $(x + y) + z = x + (y + z)$
3. $x \cdot 1 = x$
4. $x \cdot y = y \cdot x$
5. $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
6. $x \cdot (y + z) = x \cdot y + x \cdot z$
7. $1^x = 1$
8. $x^1 = x$
9. $x^{y+z} = x^y \cdot x^z$
10. $(x \cdot y)^z = x^z \cdot y^z$
11. $(x^y)^z = x^{y \cdot z}$.

Problem: Are there identities involving addition, multiplication, and exponentiation over the positive integers that *cannot be proved* using the eleven axioms about these operations that are taught in high-school-level mathematics. The question was solved in 1980 by Alex Wilkie, who showed that *such unprovable identities do exist.*

https://en.wikipedia.org/wiki/Tarski%27s_high_school_algebra_problem

Four Color Theorem



Problem (1852): given any separation of a plane into contiguous regions, producing a figure called a map, no more than four colors are required to color the regions of the map so that no two adjacent regions have the same color.
A computer-assisted proof was given in 1976 by Appel and Haken -- the first major theorem to be proved using a computer.

Formal Proof—The Four Color Theorem by Georges Gonthier, Notices of the American Mathematical Society, Vol. 55, No. 11:
https://www.ams.org/notices/200811/tx081101382p.pdf

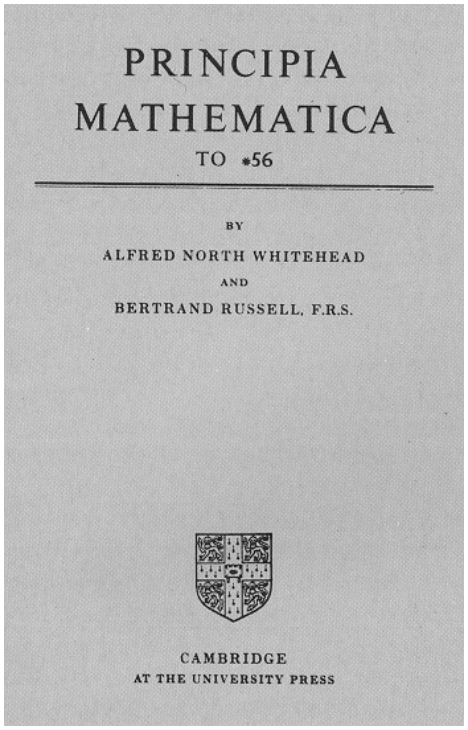
https://en.wikipedia.org/wiki/Four_color_theorem

Automated Theorem Proving

Hao Wang (Chinese: 王浩 1921 –1995) was a logician, philosopher, mathematician, and commentator on Kurt Gödel. In 1959, Wang wrote *on an IBM 704 computer a program that in only 9 minutes mechanically proved several hundred mathematical logic theorems in Whitehead and Russell's Principia Mathematica*. In 1983 he was presented with the first Milestone Prize for Automated Theorem-Proving, sponsored by the International Joint Conference on Artificial Intelligence.

https://en.wikipedia.org/wiki/Hao_Wang_(academic)

Automated Theorem Proving


PRINCIPIA
MATHEMATICA
TO *56

BY
ALFRED NORTH WHITEHEAD
AND
BERTRAND RUSSELL, F.R.S.

CAMBRIDGE
AT THE UNIVERSITY PRESS

$*54{\cdot}43.\quad \vdash :. \alpha, \beta \in 1 . \supset : \alpha \cap \beta = \Lambda . \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54{\cdot}26 . \supset \vdash :. \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

$[*51{\cdot}231] \qquad \equiv . \iota`x \cap \iota`y = \Lambda .$

$[*13{\cdot}12] \qquad \equiv . \alpha \cap \beta = \Lambda \qquad (1)$

$\vdash . (1) . *11{\cdot}11{\cdot}35 . \supset$

$\vdash :. (\exists x, y) . \alpha = \iota`x . \beta = \iota`y . \supset : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda \qquad (2)$

$\vdash . (2) . *11{\cdot}54 . *52{\cdot}1 . \supset \vdash .$ Prop

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

The Principia Mathematica (often abbreviated PM) is a three-volume work on the foundations of mathematics written by Alfred North Whitehead and Bertrand Russell and published in 1910, 1912, and 1913.
Somewhat infamously, several hundred pages of PM precede the proof of the validity of the proposition 1+1=2.

https://en.wikipedia.org/wiki/Principia_Mathematica

Wang's Algorithm

To determine whether or not a statement is valid:

Valid: All premises are TRUE and the conclusion is also TRUE.

Invalid: All Premises are TRUE but the conclusion is FALSE.

Wang's Algorithm: Search for a situation in which the statement is *invalid*.

To make an invalid statement, we assume all the premises are TRUE and the conclusion is FALSE.

e.g. Premises: A, B; Conclusion: A

Statement: A, B → A

This statement is *valid* as we cannot make it invalid.

Reason: we cannot make A to be true and false at the same time.

Wang's Algorithm

Wang's algorithm

Step I: Starting condition: Represent all sentences, involving only $\wedge, \vee$ and $\neg$ operators.

Step II: Recursive procedure: Repeat steps (a), (b) or (c) whichever is appropriate until the stopping condition, presented in **step III**, occurs.

a) **Negation Removal:** In case negated term is present at any side (separated by comma) bring it to the other side of implication symbol without its negation symbol.

e.g., $p, q, \neg r \Rightarrow s \vdash p, q \Rightarrow r, s$

b) **AND, OR Removal**: If the L.H.S. contains $\wedge$ operator, replace it by a comma. On the other hand if R.H.S. contains $\vee$ operator, also replace it by a comma. e.g., $p \wedge r, s \Rightarrow s \vee t \vdash p, r, s \Rightarrow s, t$

c) **Theorem splitting:** If the L.H.S. contains OR operator, then split the theorem into two sub-theorems by replacing the OR operator. Alternatively, if the R.H.S. contains AND operator, then also split the theorem into two sub-theorems.

e.g., $p \vee r \Rightarrow s, t \vdash p \Rightarrow s, t \;\&\; r \Rightarrow s, t$: **Sub-theorems**

e.g., $p, r \Rightarrow s \wedge t \vdash p, r \Rightarrow s \;\&\; p, r \Rightarrow t$: **Sub-theorems**

Step III: Stopping Condition: Stop theorem proving process if either of (a) or (b) occurs.

(a) If both L.H.S. and R.H.S. contain common atomic terms, then stop.

(b) If L.H.S. and R.H.S. have been represented as a collection of atomic terms, separated by commas only and there exist no common terms on both sides, then stop.

End of Algorithm.

Wang's Algorithm

Standard theorems in propositional logic:

Assuming **p, q** and **r** to be propositions, the list of the standard theorems is:

1. $p, q \Rightarrow p \wedge q$
2. $p, p \rightarrow q \Rightarrow q$ (Modus Ponens)
3. $\sim p, p \vee q \Rightarrow q$
4. $\sim q, p \rightarrow q \Rightarrow \sim p$ (Modus Tollens)
5. $p \vee q, p \rightarrow r, q \rightarrow r \Rightarrow r$
6. $p \rightarrow q, q \rightarrow r \Rightarrow p \rightarrow r$ (chaining)
7. $p, p \rightarrow q, q \rightarrow r \Rightarrow r$ (Modus Ponens & chaining)
8. $p \vee (q \wedge \sim q) \Leftrightarrow p$
9. $p \wedge (q \vee \sim q) \Leftrightarrow p$
10. $p \rightarrow q \Leftrightarrow \sim p \vee q$
11. $\sim(p \rightarrow q) \Leftrightarrow p \wedge \sim q$
12. $p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$
13. $p \leftrightarrow q \Leftrightarrow (p \wedge q) \vee (\sim p \wedge \sim q)$
14. $p \rightarrow (q \rightarrow r) \Leftrightarrow (p \wedge q) \rightarrow r$
15. $p \rightarrow q \Leftrightarrow \sim q \rightarrow \sim p$ (contraposition theorem)

Wang's Algorithm: Example

Illustration of Wang's Algorithm to prove a syllogism in Lewis Carroll's logic puzzle

Facts:

(a) All babies are illogical.

(b) Nobody is despised who can manage a crocodile.

(c) Illogical persons are despised.

Premises:

P1: If he is a baby, then he is not logical.

P2: If he can manage a crocodile, then he is not despised.

P3: If he is not logical, then he is despised.

Conclusion: If he is a baby, then he cannot manage a crocodile .

Wang's Algorithm: Example

The representation:

he is a baby

he is logical

he can manage a crocodile

he is despised

Rephrase the premises and conclusion with *p, q, r* and *s*.

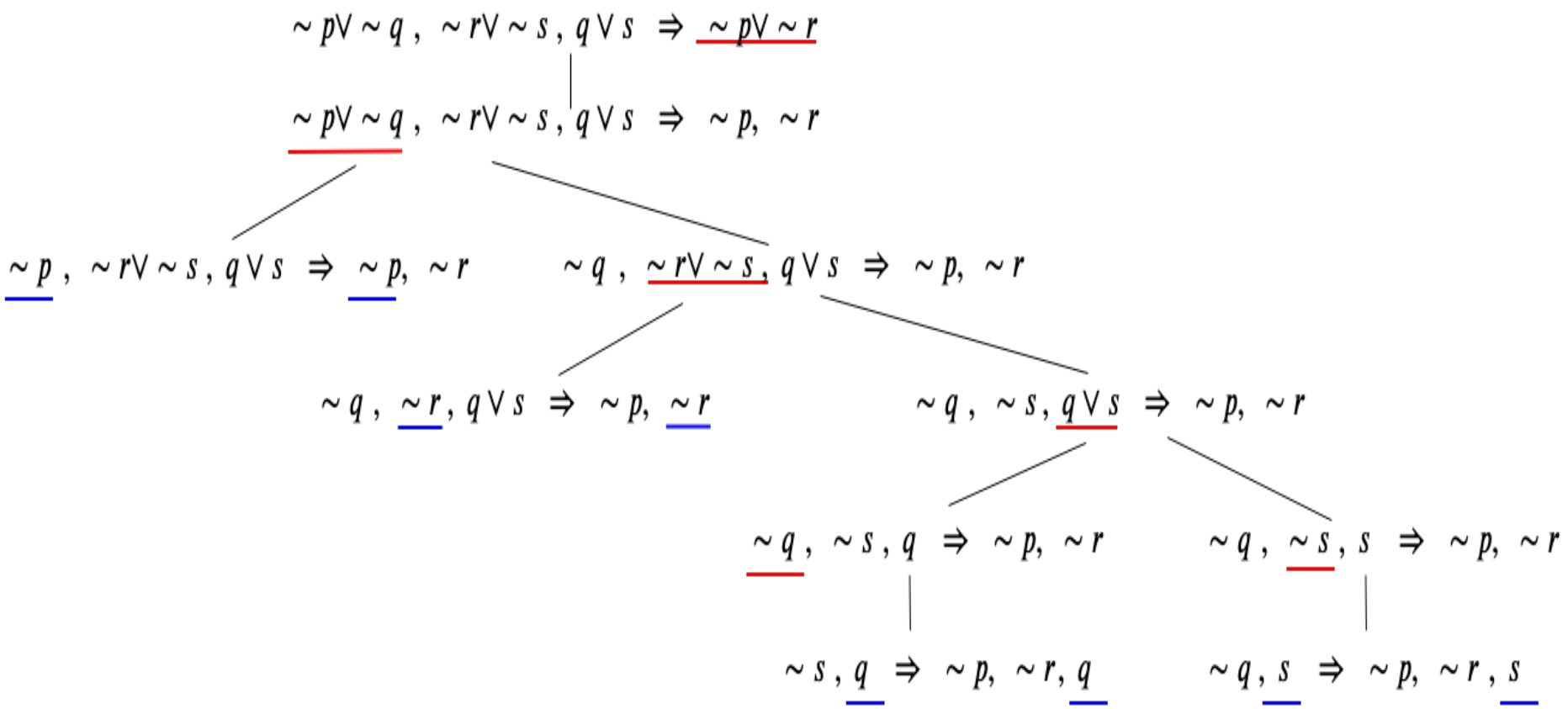
P1:

P2:

P3:

Conclusion:

Wang's Algorithm: Example



Robbin's Algebra Conjecture

For all elements *a*, *b*, and *c*:

1. Associativity: $a \vee (b \vee c) = (a \vee b) \vee c$
2. Commutativity: $a \vee b = b \vee a$
3. *Robbins equation*: $\neg(\neg(a \vee b) \vee \neg(a \vee \neg b)) = a$

In 1933, Edward Huntington proposed a new set of axioms for Boolean algebras. Very soon thereafter, Herbert Robbins posed the "Robbins conjecture", namely that the Huntington equation could be replaced with what came to be called the Robbins equation, and the result would still be Boolean algebra. Verifying the Robbins conjecture required proving Huntington's equation, or some other axiomatization of a Boolean algebra, as theorems of a Robbins algebra. Huntington, Robbins, Alfred Tarski, and others worked on the problem, but failed to find a proof or counterexample. William McCune proved the conjecture in 1996, using the automated theorem prover EQP.

https://en.wikipedia.org/wiki/Robbins_algebra

Automated Proving and Disproving

Proving and Disproving Information Inequalities: Theory and Scalable Algorithms

Siu-Wai Ho, *Senior Member, IEEE*, Lin Ling, Chee Wei Tan, *Senior Member, IEEE*, and Raymond W. Yeung, *Fellow, IEEE*

***Abstract*— Proving or disproving an information inequality is a crucial step in establishing the converse results in coding theorems. However, an information inequality involving more than a few random variables is difficult to be proved or disproved manually. In 1997, Yeung developed a framework that uses linear programming for verifying linear information inequalities. Under the framework, this paper considers a few other problems that can be solved by using Lagrange duality and convex approximation. We will demonstrate how linear programming can be used to find an analytic proof of an information inequality or an analytic counterexample to disprove it if the inequality is not true in general. The way to automatically find a shortest proof or a smallest counterexample is explored. When a given information inequality cannot be proved, the sufficient conditions for a counterexample to disprove the information inequality are found by linear programming. Lastly, we propose a scalable algorithmic framework based on the alternating direction method of multipliers to accelerate solving a multitude of user-specific problems whose overall computational cost can be amortized with the number of users, and present its publicly-available software implementation for large-scale problems.**

***Index Terms*— Entropy, mutual information, information inequality, automated reasoning by convex optimization.**

I. Introduction

THE importance of computers as a mathematical tool for automated reasoning cannot be overstated as is evident from computer-assisted proofs to derive theorems (e.g., proving all theorems in Principia Mathematica [4]) or to validate well-known mathematical conjectures such as the four color theorem in graph theory by Heesch–Appel–Haken and the Kepler conjecture in geometry by Hales [5]. It is thus imperative to understand how computers can play a similar role of automated reasoning in the realm of information theory. In this paper, we present a theoretical and algorithmic framework to generate computer-aided proofs for information inequalities. We also demonstrate its applicability to a variety of problems and its software implementation as a proof of concept. This paper presents a step toward a systematic understanding of "automated reasoning by convex optimization", where mathematical optimization techniques are used for proof search in large-scale problems.

In information theory, we may need to prove or disprove different kinds of information inequalities in different problems. For example, information inequalities often play a crucial role in establishing the converse of fundamental capacity theorems or Shannon's perfect secrecy theorem. As such, it is important to verify the correctness of a given information inequality by either providing a rigorous proof or otherwise disprove it by providing a counterexample. However, to manually prove or disprove an information inequality is non-trivial in general especially when it involves more than three random variables.

For example, we may want to check the correctness of the following two inequalities:

$$I(A;B|CD)+I(B;D|AC) \leq I(A;B|D)+I(B;D|A)+H(A)+I(B;D|C) \quad (1)$$

and

$$I(A;B|CD)+I(B;D|AC) \leq I(A;B|D)+I(B;D|A)+H(AB|D). \quad (2)$$

To prove information inequalities, the author in [6] developed a framework for linear information inequalities and provided a software package known as Information Theoretic Inequality Prover (ITIP) [7]. ITIP and XITIP (similar to ITIP but it uses a C-based linear programming solver instead [8]) are widely used software packages, and more recent implementations include minitip [9] and psitip [10]. These software packages can be used to verify an information inequality on a computer. For example, if we use ITIP to verify (1) and (2), we will get "Not provable by ITIP" and "True", respectively. However, after we know that (2) is true, we still need an analytic proof. An analytic proof is the formal way to verify an information inequality and, more importantly, it also provides us further insights about the inequality of interest. One important insight is about the necessary and sufficient conditions for the equality to hold

Manuscript received May 16, 2019; revised December 22, 2019; accepted March 4, 2020. Date of publication March 23, 2020; date of current version August 18, 2020. This work was supported by the Hong Kong University Grants Committee under Project 11207615 and Project AoE/E-02/08, and in part by the National Natural Science Foundation of China under Grant 61771018. This article was presented at the 2014 IEEE International Symposium on Information Theory, in part at the 2014 IEEE International Conference on Communication Systems, and in part at the 2019 IEEE International Symposium on Information Theory. *(Corresponding author: Lin Ling.)*

Siu-Wai Ho is with the Teletraffic Research Centre, The University of Adelaide, Adelaide, SA 5005, Australia.

Lin Ling and Chee Wei Tan are with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: linling2-c@my.cityu.edu.hk).

Raymond W. Yeung is with the Institute of Network Coding, The Chinese University of Hong Kong, Hong Kong, and also with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong.

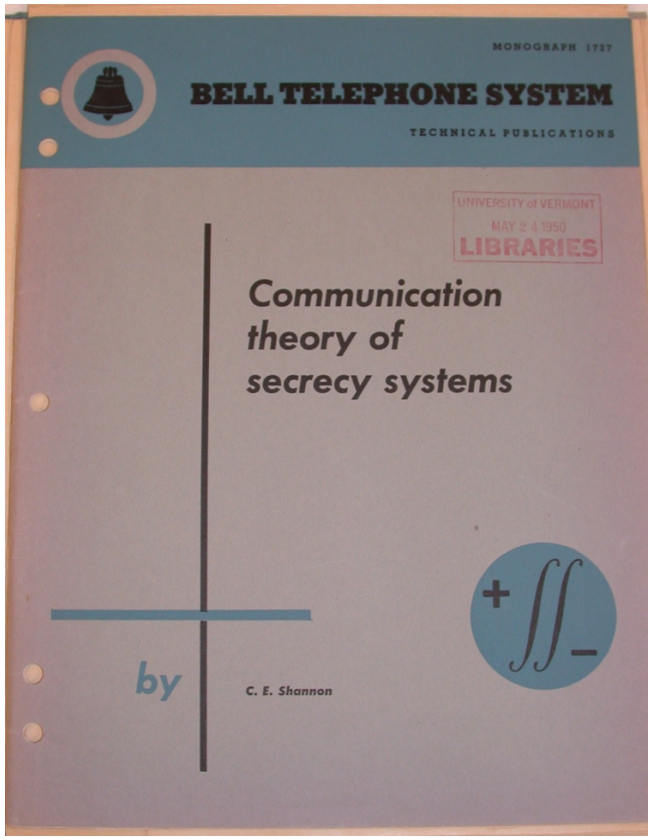
Communicated by F. Alajaji, Associate Editor for Shannon Theory.

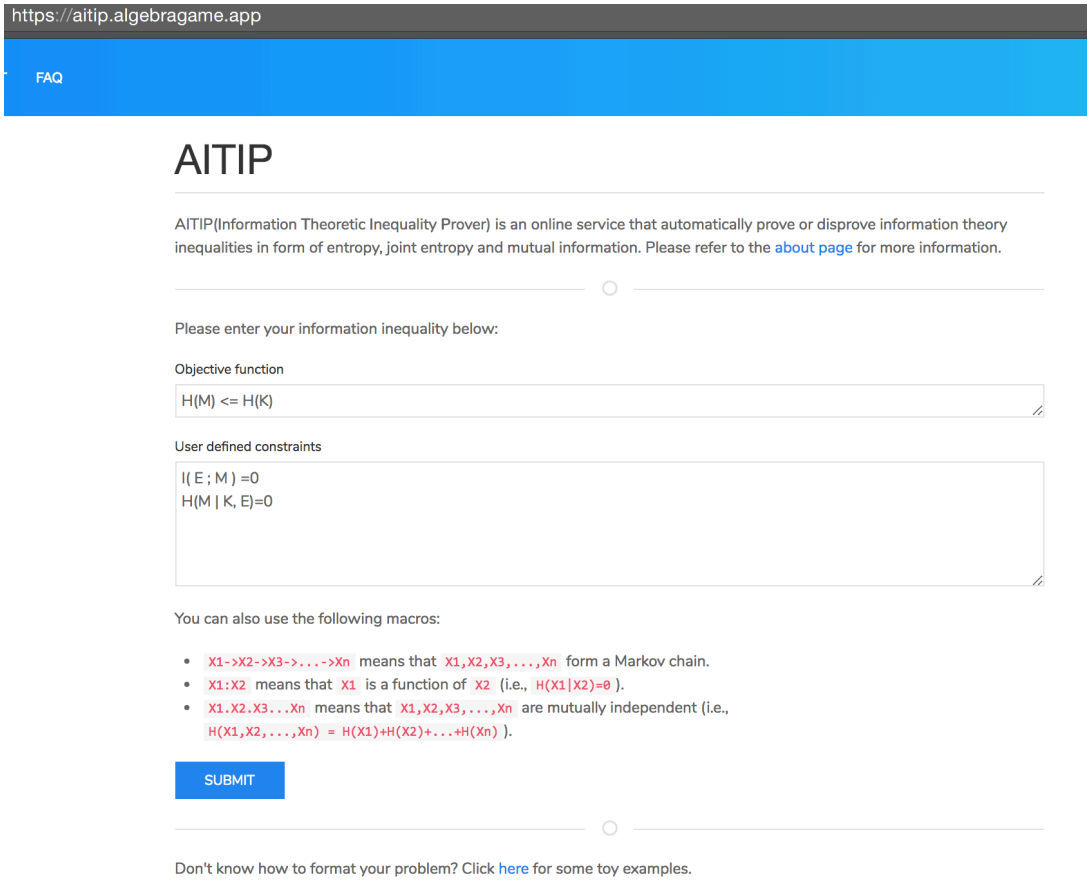
Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2020.2982642

Automated Reasoning by Convex Optimization: Proving or disproving Shannon-type Information Inequalities automatically.
There are non-Shannon type Information Inequalities that cannot be proved by this optimization framework.

Automated Theorem Proving: Example





Shannon's Perfect Secrecy Theorem states all theoretically unbreakable ciphers must have the same requirements as the *One-Time Pad*, which was invented in 1919 by Gilbert Vernam in using the XOR operation for encryption in conjunction with his Vernam cipher. The diagram on the righthand-side shows our *Software-as-a-Service* prototype to automatically prove this theorem.