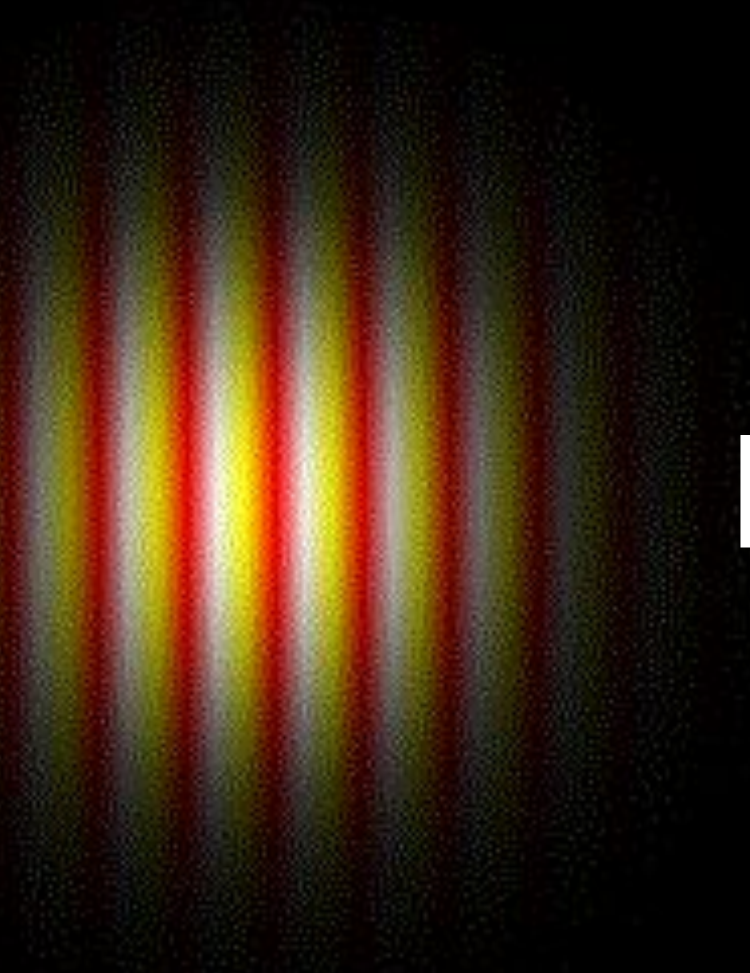
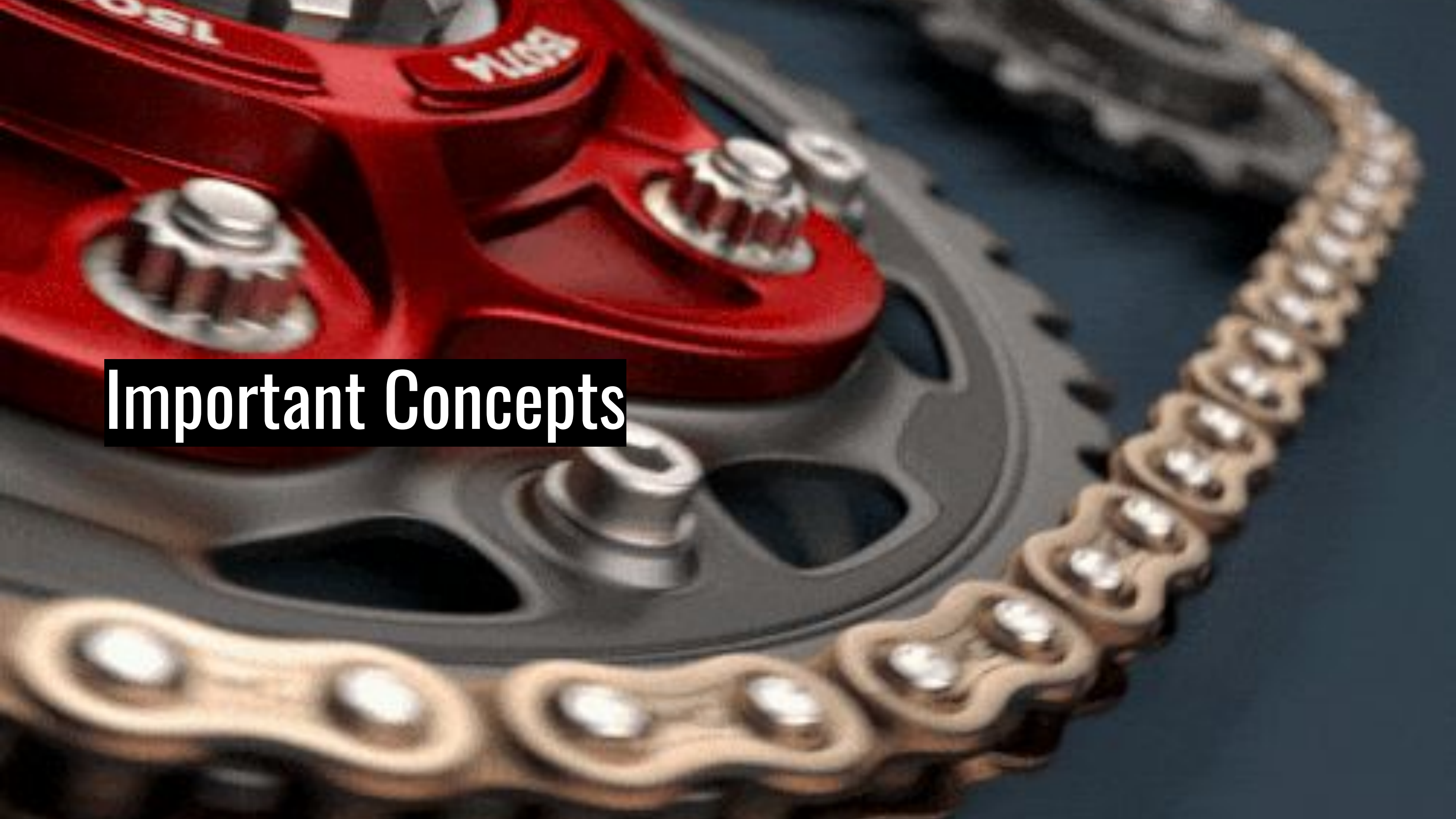


Inference Rules (Pt. II)

Important Concepts



Subderivations

A subderivation is a **procedure** through which we make a new assumption to derive a sentence of TL we couldn't have otherwise derived.

All subderivations are initiated with an **assumption** for the application of a subderivational rule of inference, i.e., $\supset$I, ~I, ~E, $\vee$E, $\equiv$I.

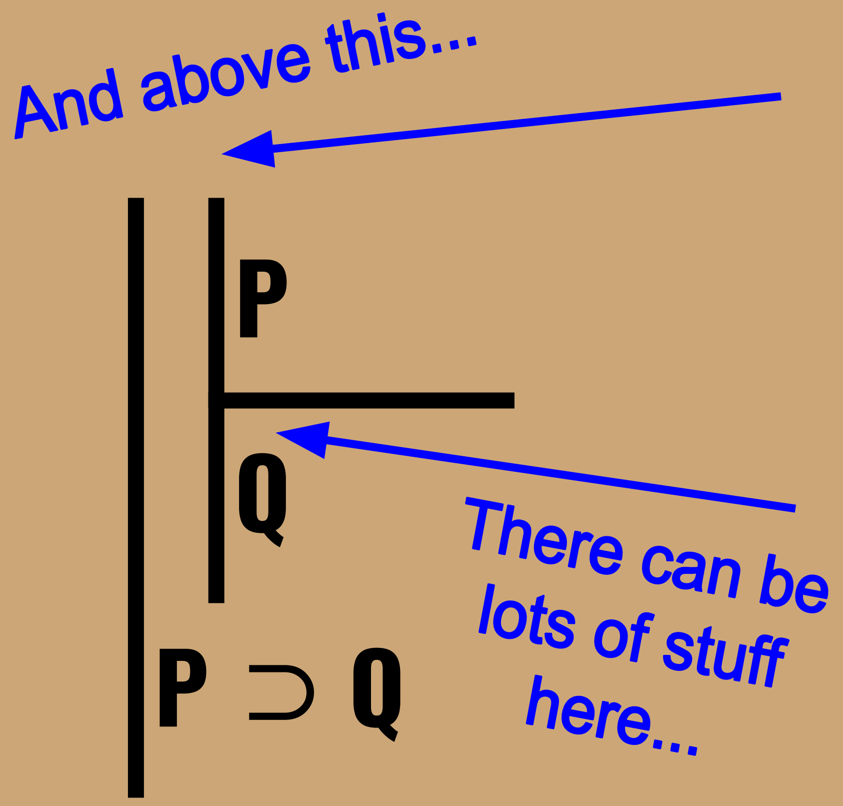
All subderivation rules, however, **must eventually close** the assumption with which it was initated.

Rules of Inference

Conditional Introduction

(⊃I)

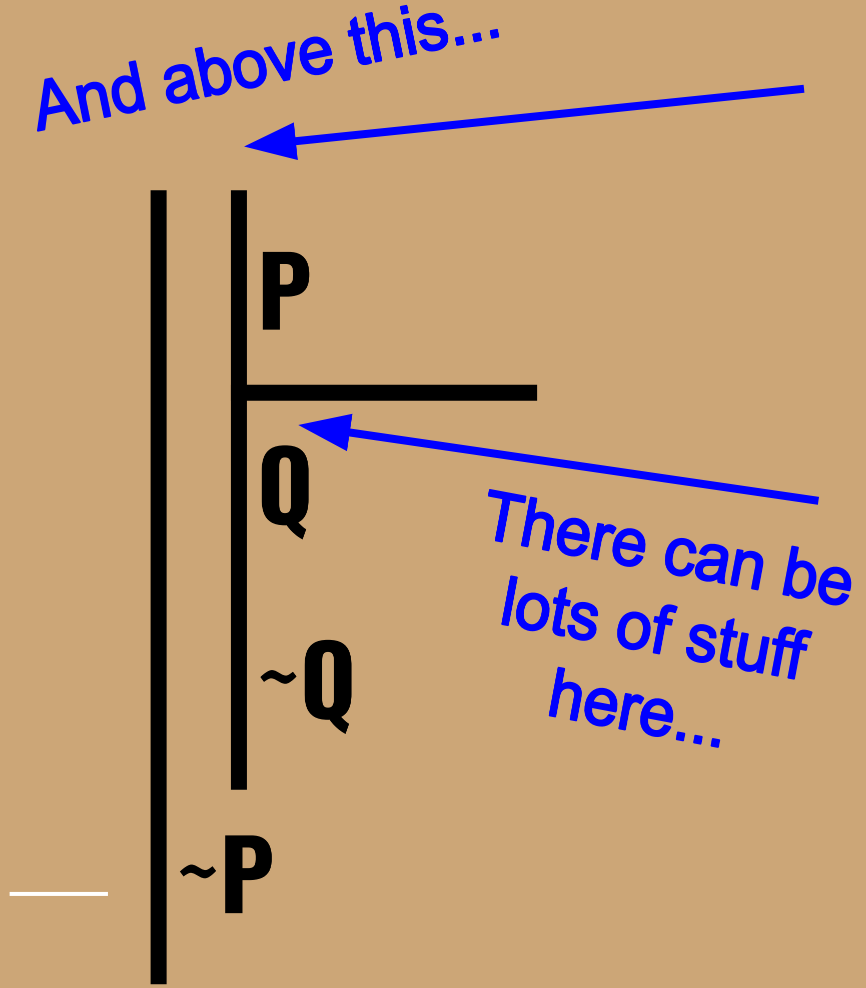
Where **P** and **Q** are meta-variables ranging over declarative sentences...



Negation Introduction

(~I)

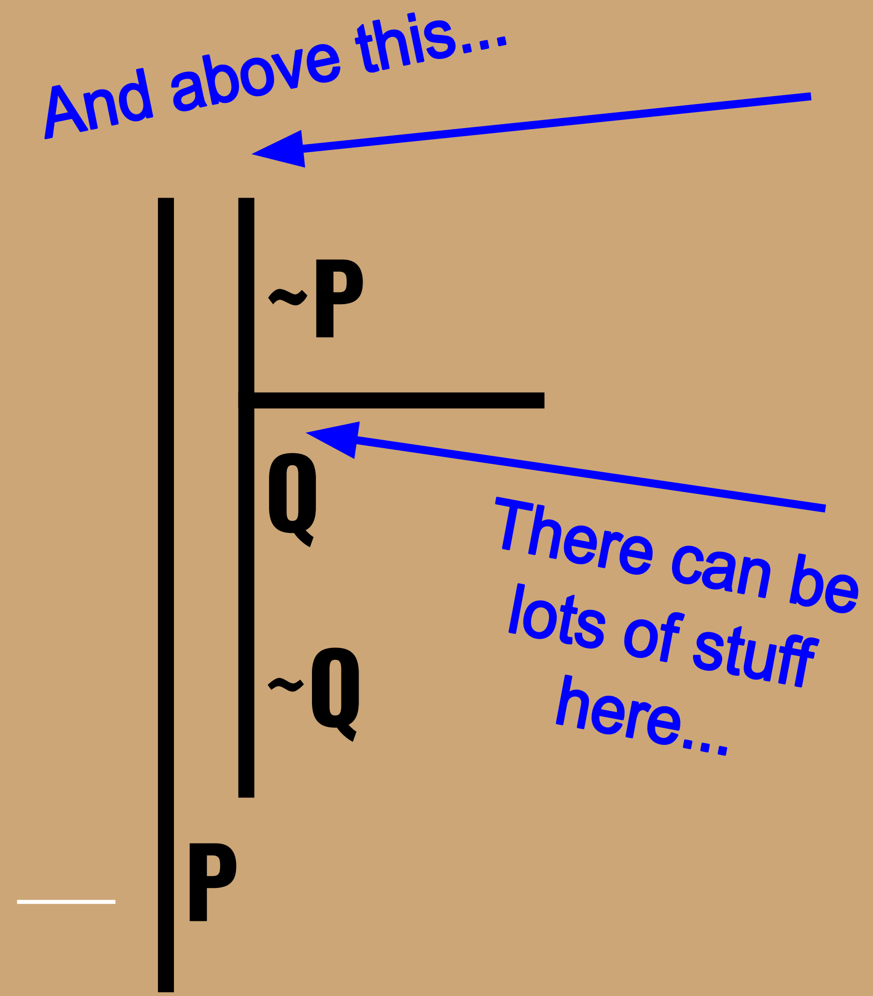
Where **P** and **Q** are meta-variables ranging over declarative sentences...



Negation Elimination

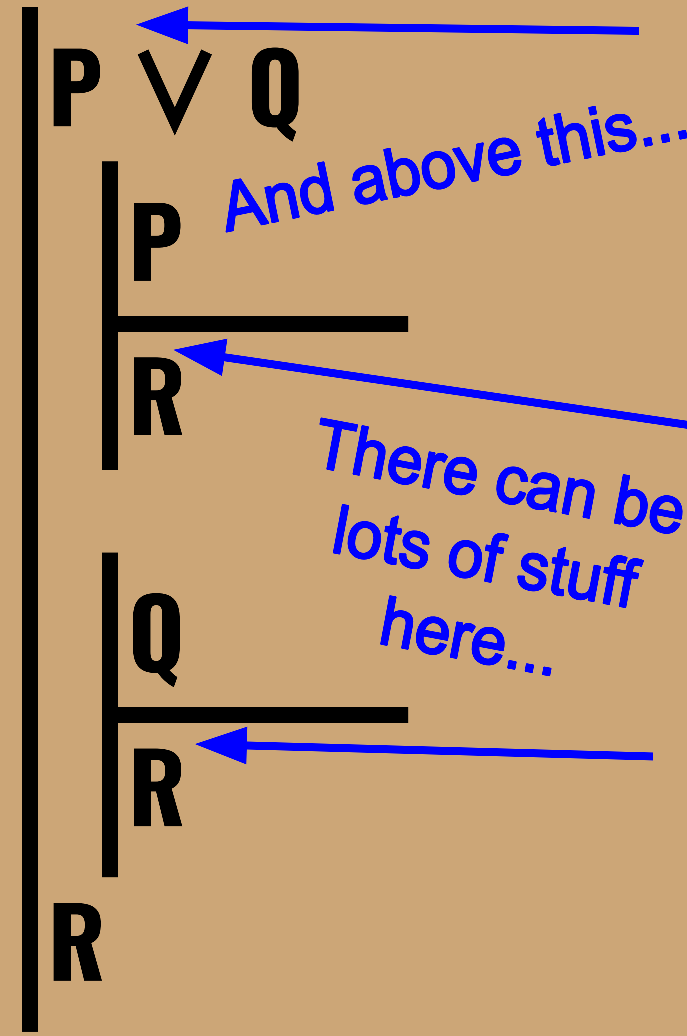
(~E)

Where **P** and **Q** are meta-variables ranging over declarative sentences...



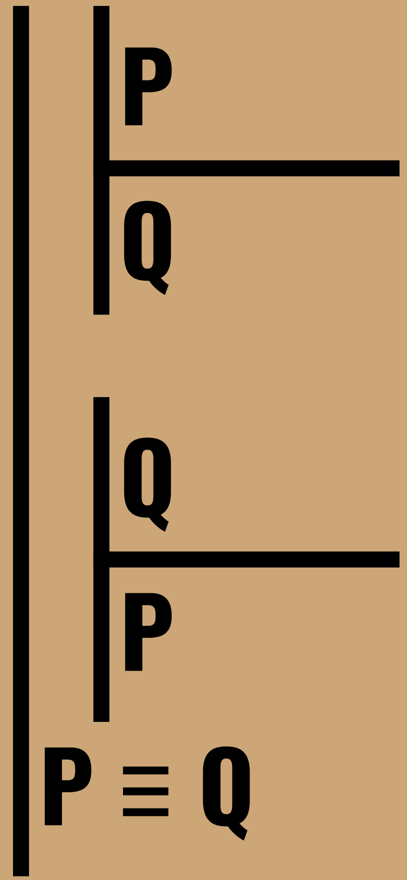
Disjunction Elimination
(⋁ E)

Where **P** and **Q** are meta-variables ranging over declarative sentences…



Biconditional Introduction (≡I)

Where **P** and **Q** are meta-variables ranging over declarative sentences…



Derivations in SD

A derivation in SD is a series of sentences of TL, each of which is either an assumption or is obtained from previous sentences by one of the rules of SD.

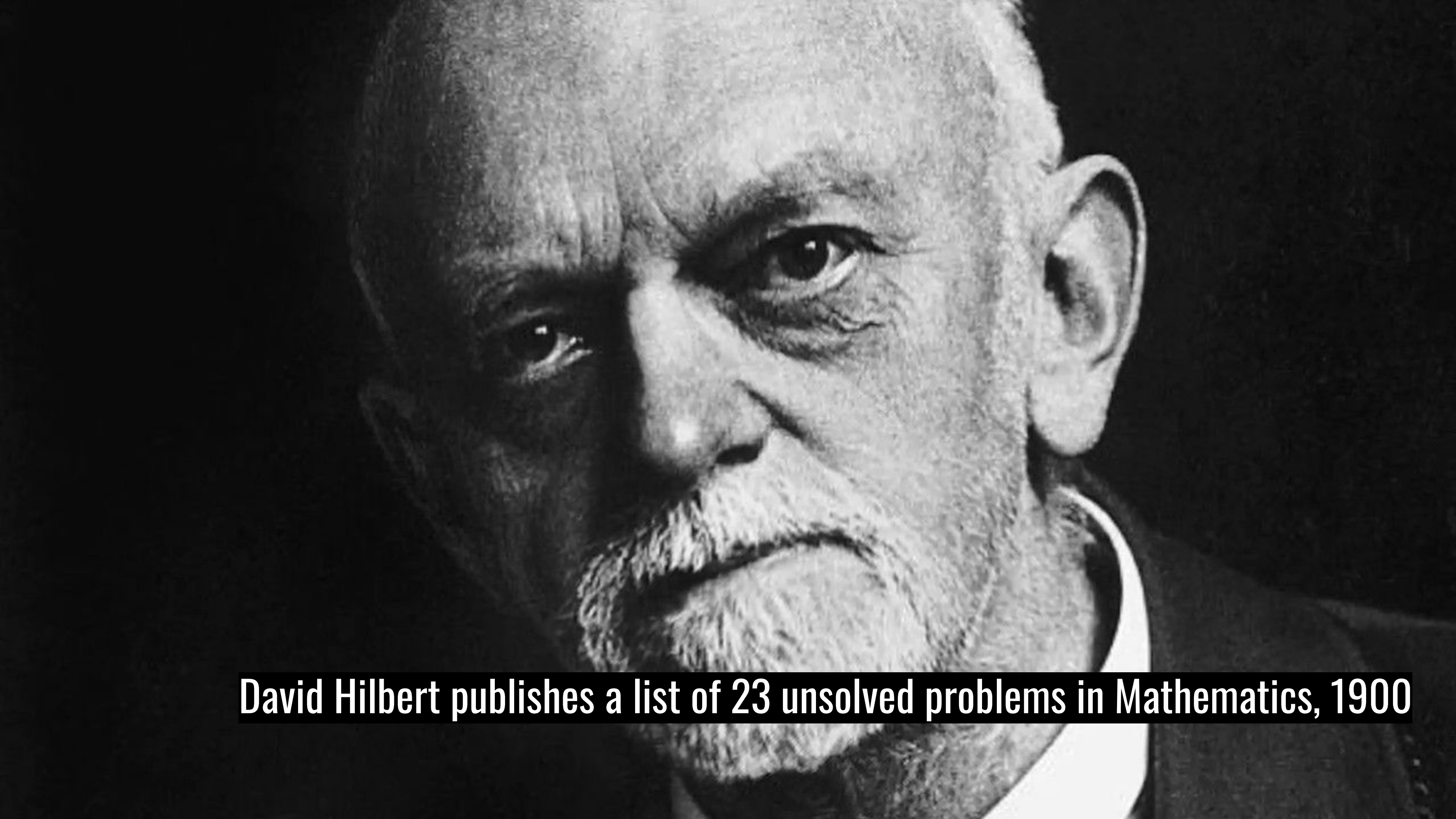
Fitch Notation

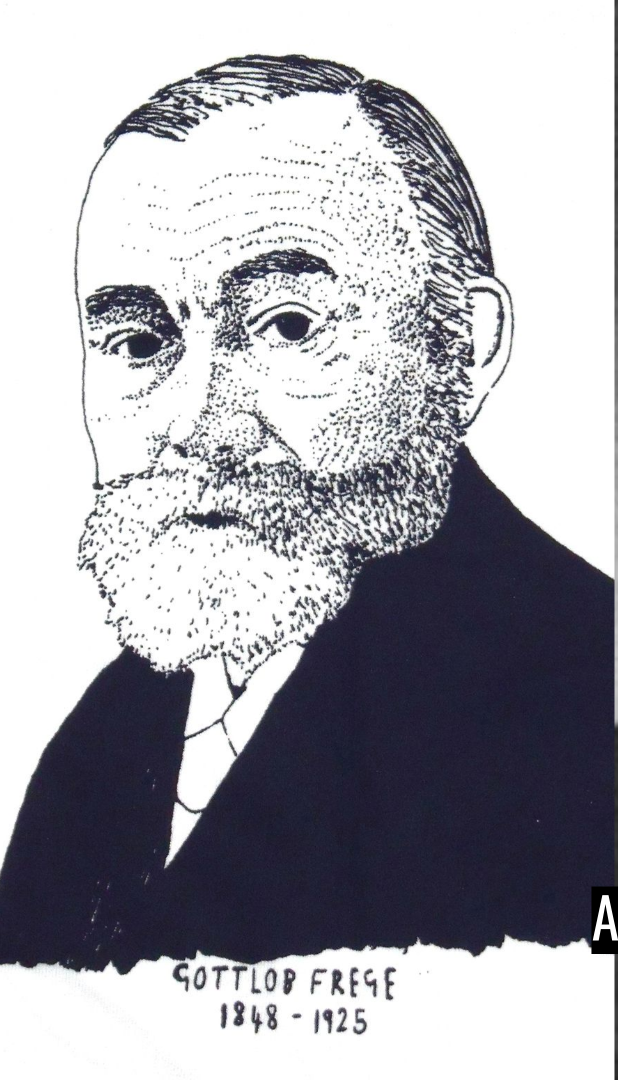
Fitch Notation is the **notational system** we will use for constructing formal proofs.

Fitch-style proofs arrange the sequence of sentences that make up the proof into **rows** and use varying degrees of **indentation** for the assumptions made throughout the proof.

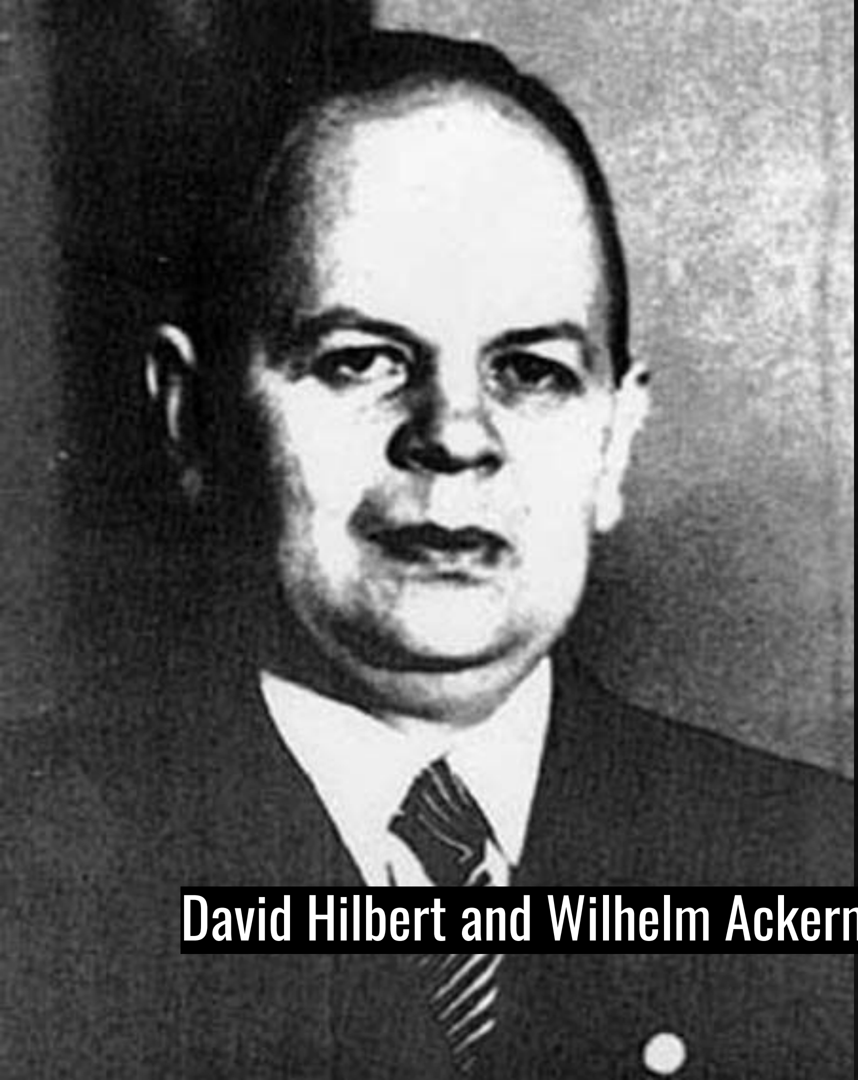
Premise(s)	Assumption
…	…
Inference(s)	Justification
…	…
Conclusion	Justification

Storytime!

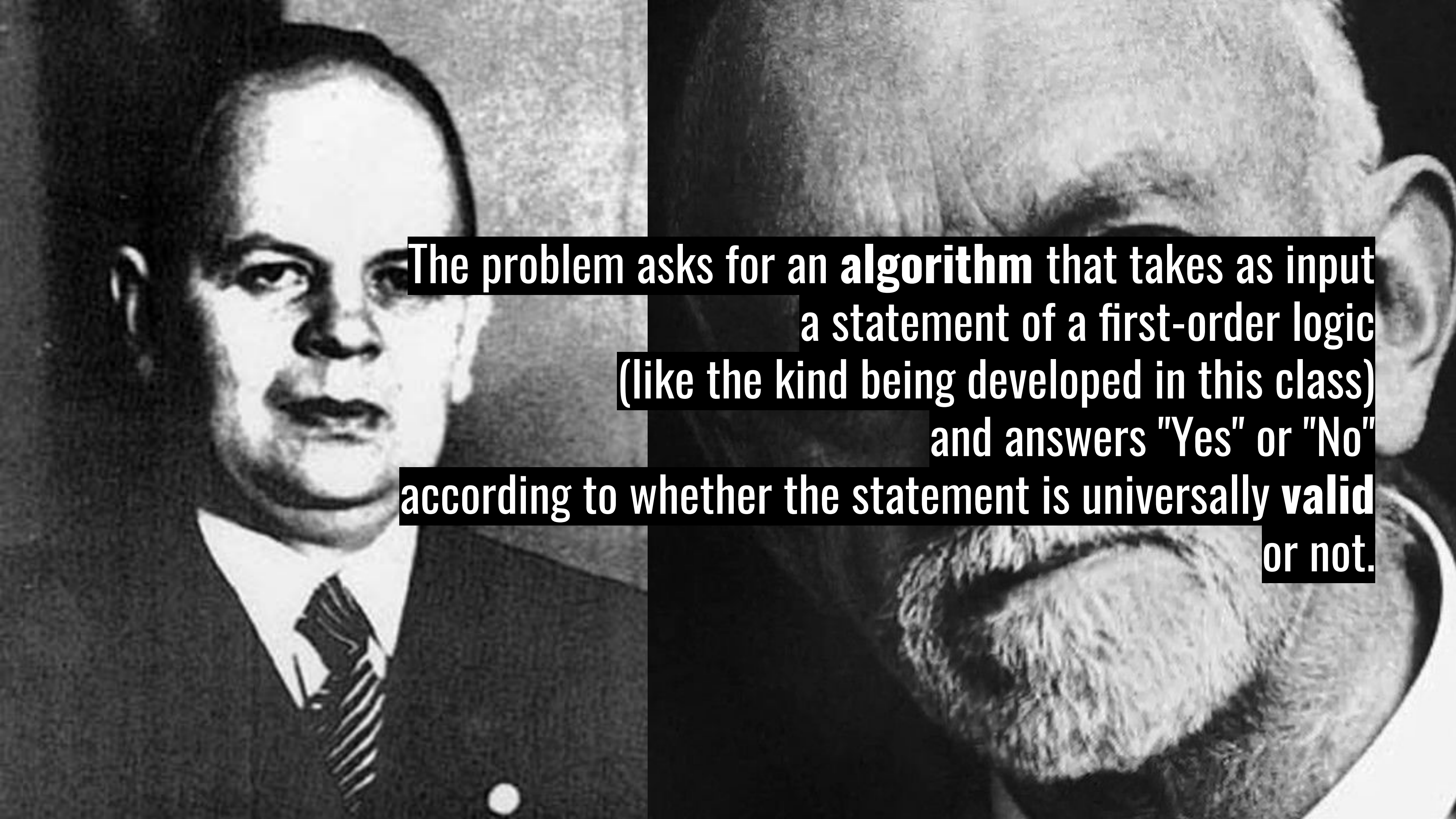

David Hilbert publishes a list of 23 unsolved problems in Mathematics, 1900




Among the problems was the continuing puzzle over Logicism...




David Hilbert and Wilhelm Ackermann propose the [Entscheidungsproblem], 1928


The problem asks for an **algorithm** that takes as input
a statement of a first-order logic
(like the kind being developed in this class)
and answers "Yes" or "No"
according to whether the statement is universally **valid**
or not.

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

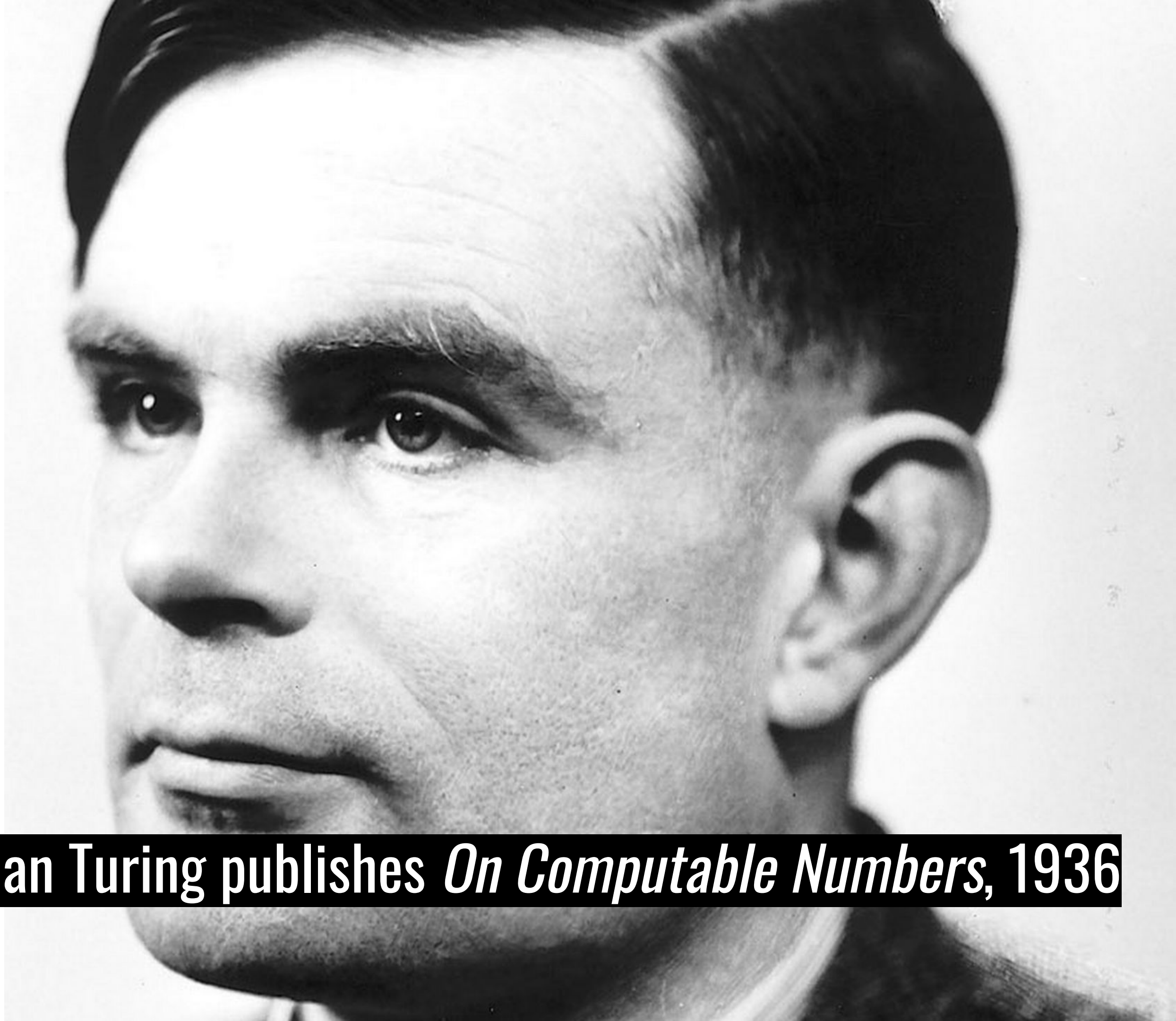
[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§ 9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions, the numbers π, e, etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerab In § 8 I examine certain arguments which would seem to prove the contrar By the correct application of one of these arguments, conclusions a reached which are superficially similar to those of Gödel†. These results

† Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I", *Monatshefte Math. Phys.*, 38 (1931), 173–198.



Alan Turing publishes *On Computable Numbers*, 1936

Person of Interest: Alan Turing

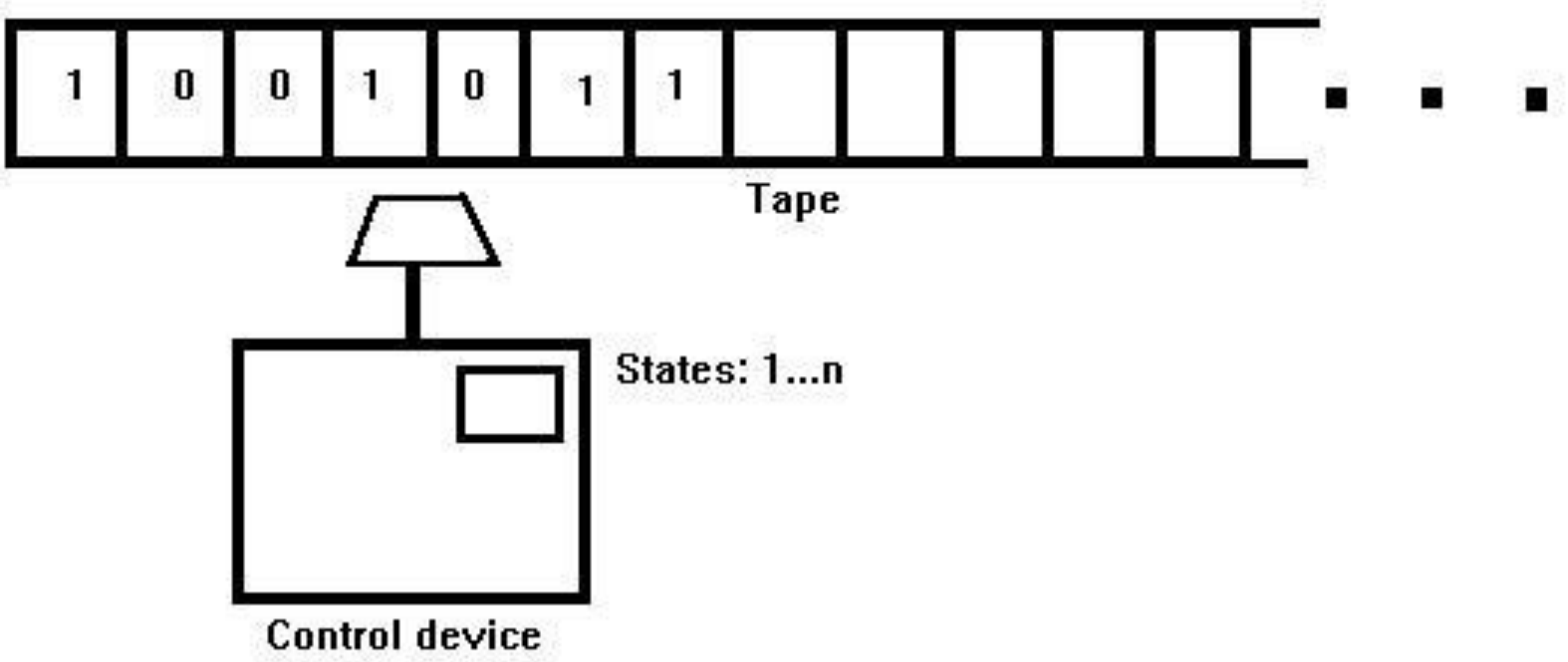
Occupation:
Mathematician
Logician
Philosopher

Notable Accomplishments:
Solving the Entscheidungsproblem
Cryptanalysis of Enigma
Church-Turing Thesis
Turing Machines
Turing Tests

Turing Machines

A Turing machine is an abstract computational device intended to help investigate the extent and limitations of what can be computed.

In other words, it's an imaginary device that can do computations.


1
0
0
1
0
1
1
Tape
States: 1...n
Control device

Question: Can machines think?

Turing: 'Think' is an ambiguous term.

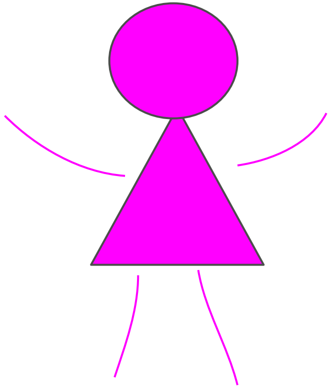
Consider the following question instead-

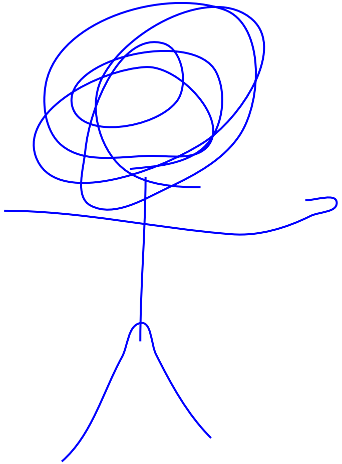
"Are there imaginable digital computers which would do well in *The Imitation Game*?"

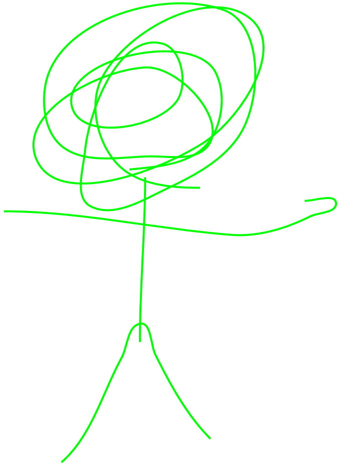
We now call this a Turing test...

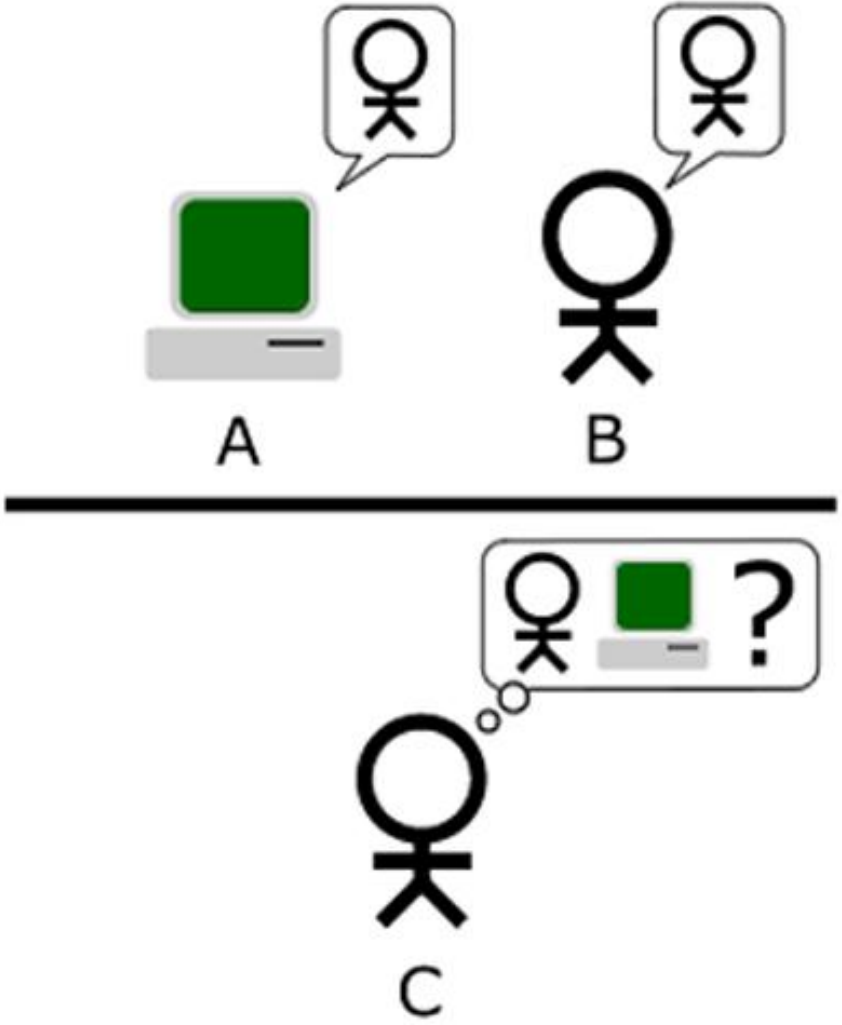
Turing Test

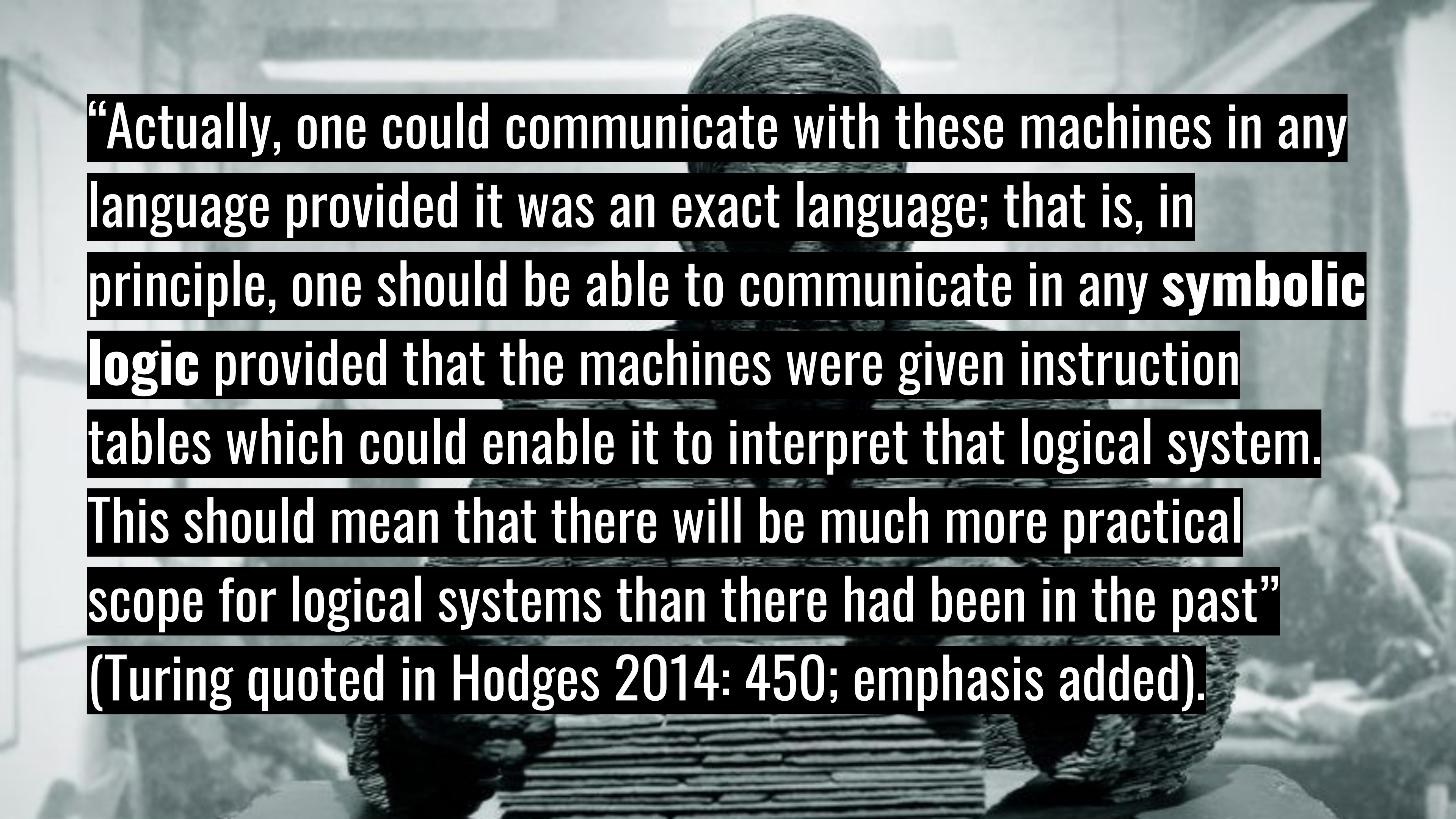
A **Turing test** is a **test** of a machine's ability to exhibit intelligent behavior equivalent to (or indistinguishable from) **that** of a human.





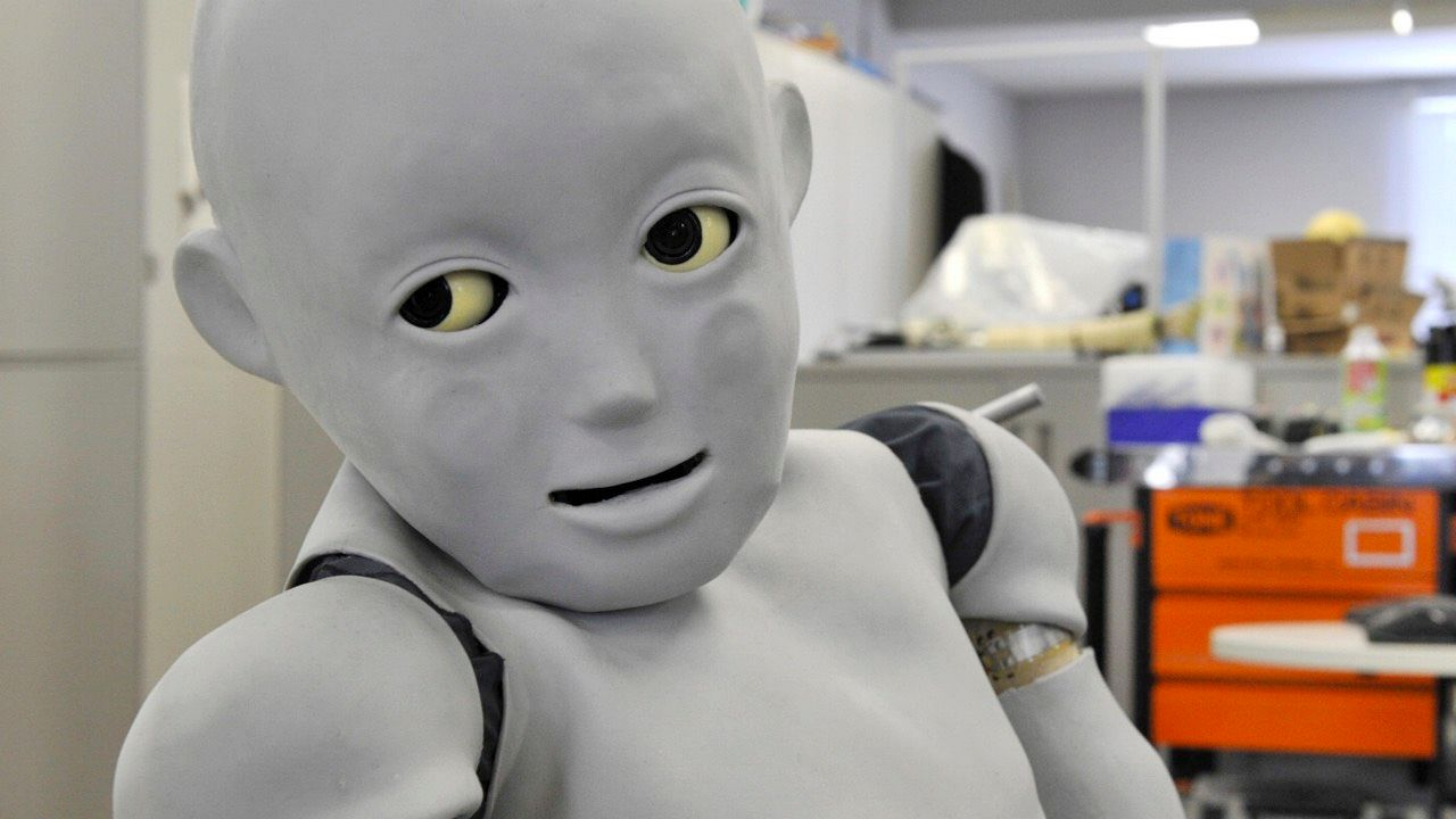


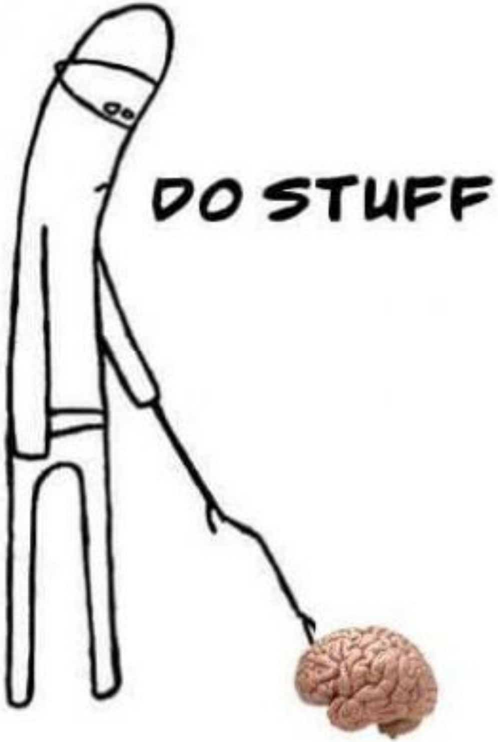

A
B
?
C



"Actually, one could communicate with these machines in any language provided it was an exact language; that is, in principle, one should be able to communicate in any **symbolic logic** provided that the machines were given instruction tables which could enable it to interpret that logical system. This should mean that there will be much more practical scope for logical systems than there had been in the past" (Turing quoted in Hodges 2014: 450; emphasis added).

Public unveiling of the ENIAC, 1948




DO STUFF

This!