# First-Order Logic = FOL = $\mathscr{L}_1$ and the First (Logic) Programmer

**Selmer Bringsjord**

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic-based AI
9/30/2024

# First-Order Logic = FOL = $\mathscr{L}_1$ and the First (Logic) Programmer

Who is it?

**Selmer Bringsjord**

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic-based AI
9/30/2024

# First-Order Logic = FOL = $\mathscr{L}_1$ and the First (Logic) Programmer

Who is it?

Leibniz
Aristotle
Gödel
…


**Selmer Bringsjord**

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA


Intro to Logic-based AI
9/30/2024

# MiniMaxularity-Related Logic-&-AI In The News

## *Beneath the Potential Strike at U.S. Ports: Tensions Over Innovation*

Port operators have long embraced automation, while dockworkers view it as a threat to their livelihoods.

Listen to this article · 8:20 min Learn more

The Port of Savannah in Georgia. Dockworkers on the East and Gulf Coasts of the United States are threatening to go on strike beginning Tuesday. Erin Schaff/The New York Times

By Peter S. Goodman
Peter S. Goodman has reported on shipping and ports around the world for more than two decades.

Throughout the centuries, as ships have navigated oceans bearing all manner of freight, the companies that operate ports have pressed to limit what they spend on the people who load and unload cargo.

Dockworkers, for their part, have mobilized to pursue a greater share of the bounty through a familiar tactic: They have threatened to disrupt international commerce by going on strike.

Confronted by the militancy of longshore unions, port operators have deployed automation, in part to limit their vulnerability to labor troubles. Not coincidentally, dockworkers tend to look suspiciously at robots and other forms of innovation, divining threats to their livelihoods.

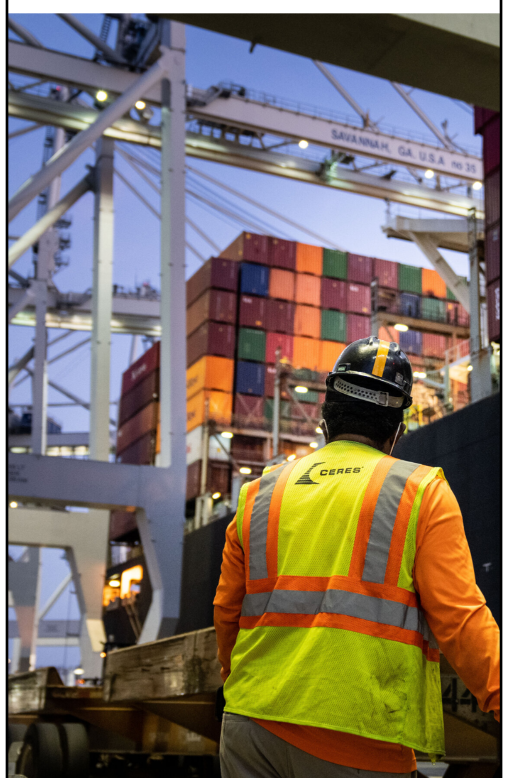
That, in a nutshell, is the history of labor relations on docks from Australia to Britain. And that dynamic is at the center of a contractual impasse now threatening to produce a debilitating strike starting Tuesday at ports on the East and Gulf Coasts of the United States.

# MiniMaxularity-Related Logic-&-AI In The News

## *Beneath the Potential Strike at U.S. Ports: Tensions Over Innovation*

Port operators have long embr
automation, while dockworker
threat to their livelihoods.

▶ Listen to this article · 8:20 min Learn m




The Port of Savannah in Georgia. Dockworkers on the East and Gulf Coasts of the United States are threatening to go on strike beginning Tuesday. Erin Schaff/The New York Times


By Peter S. Goodman
Peter S. Goodman has reported on shipping and ports around the world for more than two decades.


THE NATIONAL ARCHIVES

Visit
What's on
Explore the collection

Home > Education > Classroom resources > Why did the Luddites protest?

View lesson as PDF | View full image

### Why did the Luddites protest?

#### Political reform in 19th century Britain

- Tasks
- Background
- Teachers' notes
- External links
- Connections to curriculum

The machine-breaking disturbances that rocked the wool and cotton industries were known as the 'Luddite riots'. The Luddites were named after 'General Ned Ludd' or 'King Ludd', a mythical figure who lived in Sherwood Forest and supposedly led the movement.

They began in Nottinghamshire in 1811 and quickly spread throughout the country, especially to the West Riding of Yorkshire and Lancashire in 1812, and also to Leicestershire and Derbyshire. In Yorkshire, they wanted to get rid of the new machinery that was causing unemployment among workers. Hand loom weavers did not want the introduction of power looms. In Nottinghamshire, they protested against wage reductions.

Workers sent threatening letters to employers and broke into factories to destroy the new machines, such as the new wide weaving frames. They also attacked employers, magistrates and food merchants. There were fights between Luddites and government soldiers.

Using the original documents in this lesson, find out how the Luddites protested against changes affecting their working conditions. How did the government respond?

Throughout the centuries, as ships have
navigated oceans bearing all manner of
npanies that operate ports have
t what they spend on the people
nload cargo.

for their part, have mobilized to
er share of the bounty through
c: They have threatened to
tional commerce by going on

the militancy of longshore
erators have deployed
part to limit their vulnerability
es. Not coincidentally,
end to look suspiciously at
er forms of innovation, divining
livelihoods.

nell, is the history of labor
cks from Australia to Britain.
mic is at the center of a
contractual impasse now threatening to
produce a debilitating strike starting Tuesday
at ports on the East and Gulf Coasts of the
United States.


# Why did the Luddites protest?

## Political reform in 19th century Britain

- Tasks
- Background
- Teachers' notes
- External links
- Connections to curriculum

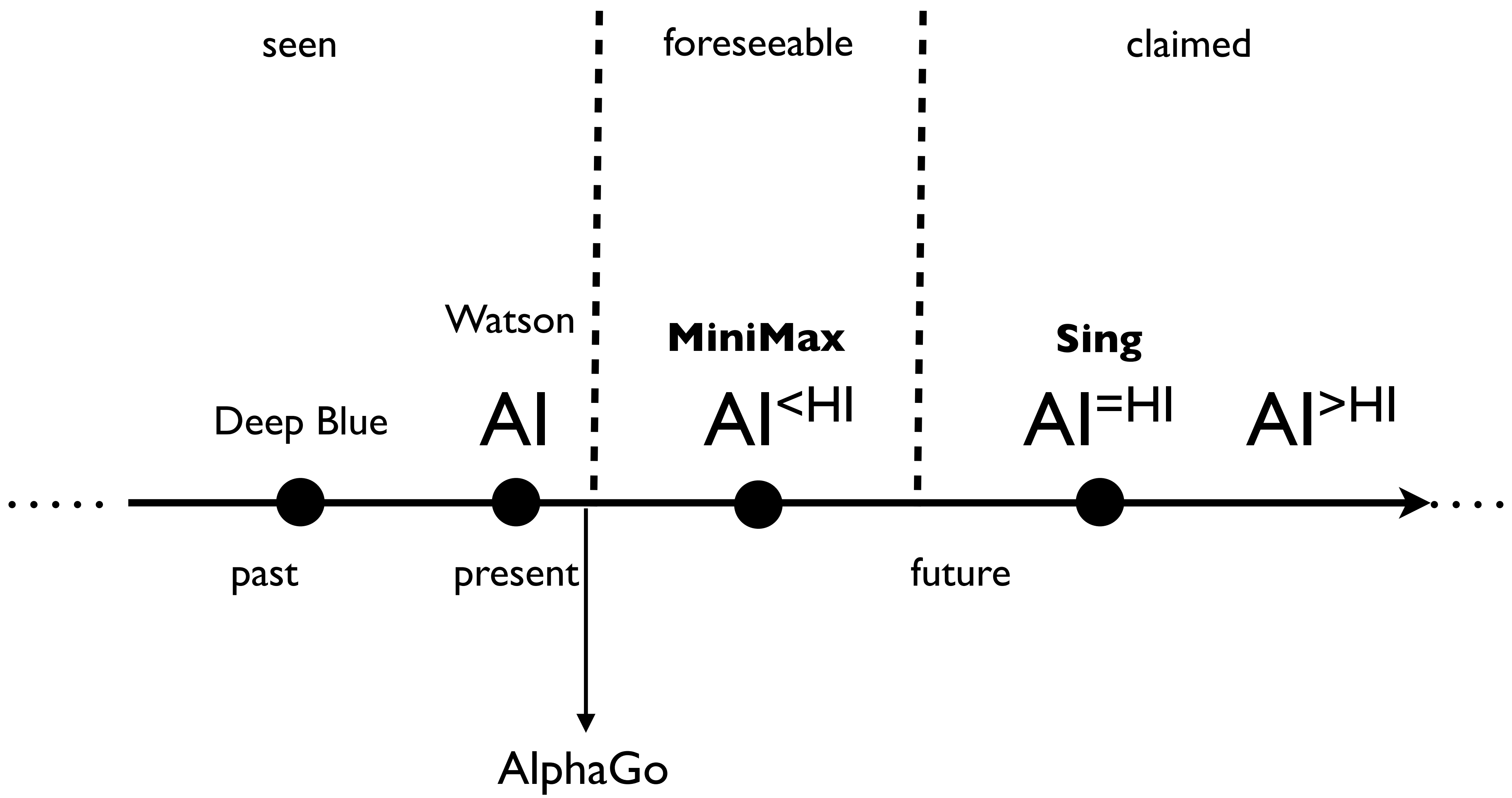
The machine-breaking disturbances that rocked the wool and cotton industries were known as the ‘Luddite riots’. The Luddites were named after ‘General Ned Ludd’ or ‘King Ludd’, a mythical figure who lived in Sherwood Forest and supposedly led the movement.

They began in Nottinghamshire in 1811 and quickly spread throughout the country, especially to the West Riding of Yorkshire and Lancashire in 1812, and also to Leicestershire and Derbyshire. In Yorkshire, they wanted to get rid of the new machinery that was causing unemployment among workers. Hand loom weavers did not want the introduction of power looms. In Nottinghamshire, they protested against wage reductions.

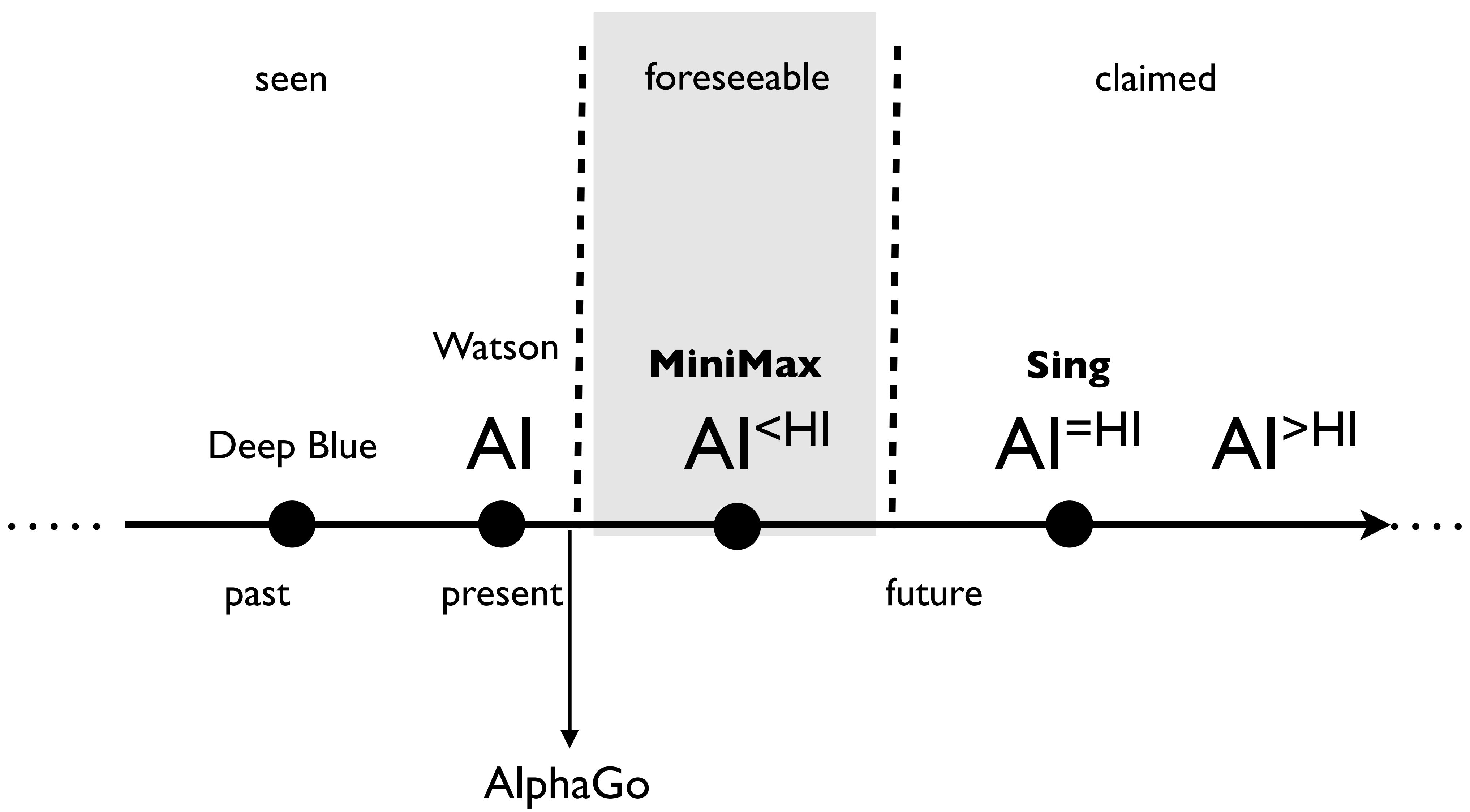
Workers sent threatening letters to employers and broke into factories to destroy the new machines, such as the new wide weaving frames. They also attacked employers, magistrates and food merchants. There were fights between Luddites and government soldiers.

Using the original documents in this lesson, find out how the Luddites protested against changes affecting their working conditions. How did the government respond?

# A Realistic Timeline (from last time)



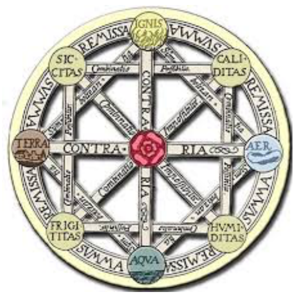
# A Realistic Timeline (from last time)



?

***Entscheidungsproblem***

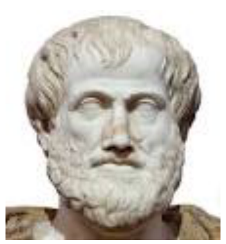
"Universal Computational Logic"



Logic Theorist
(birth of modern logicist AI)

350 BC | 300 BC | 1666 | 1854 | 1956 | 2023 | 2024

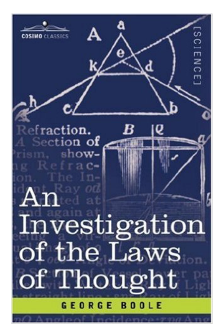
Euclid



*Organon*

Leibniz

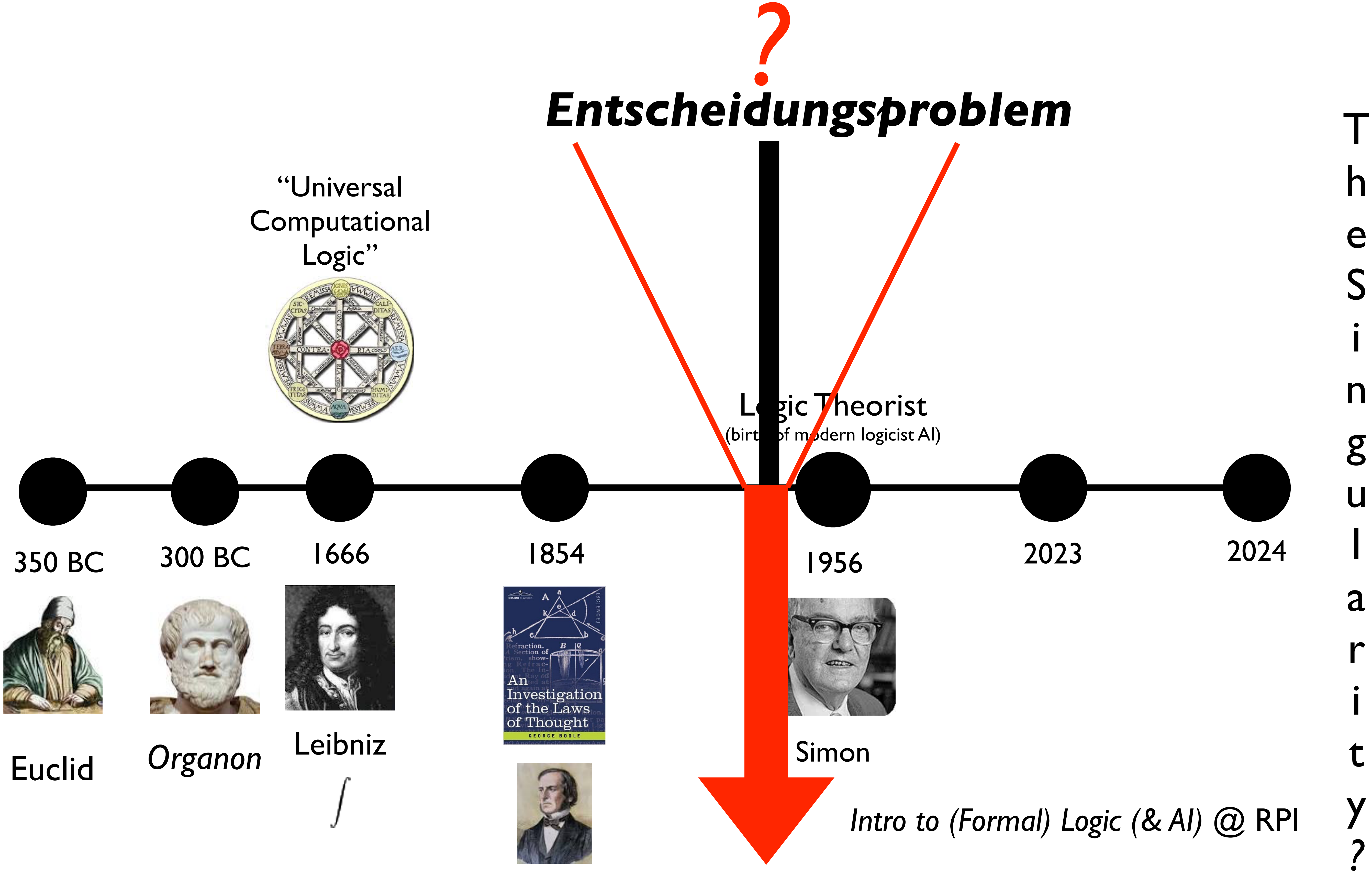
$\int$




Simon

T h e S i n g u l a r i t y ?

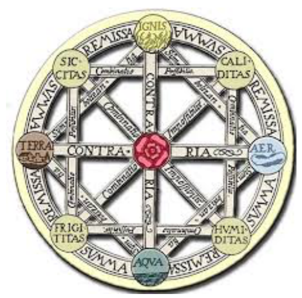
*Intro to (Formal) Logic (& AI)* @ RPI


?
*Entscheidungsproblem*
"Universal Computational Logic"
Logic Theorist
(birth of modern logicist AI)
350 BC
300 BC
1666
1854
1956
2023
2024
Euclid
*Organon*
Leibniz
∫
An Investigation of the Laws of Thought
GEORGE BOOLE
Simon
*Intro to (Formal) Logic (& AI) @* RPI
The Singularity?

?

***Entscheidungsproblem***

"Universal Computational Logic"

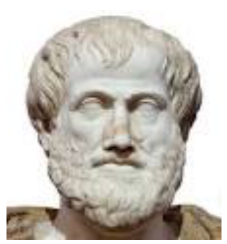


Logic Theorist

(birth of modern logicist AI)

350 BC | 300 BC | 1666 | 1854 | 1956 | 2023 | 2024

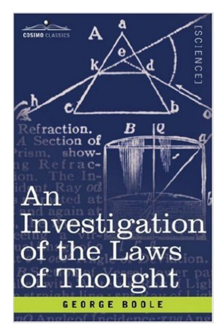
Euclid



*Organon*

Leibniz

$\int$




Simon

*Intro to (Formal) Logic (& AI)* @ RPI

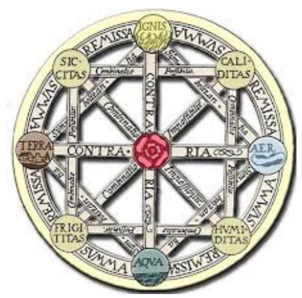
Frege

The Singularity?

?

***Entscheidungsproblem***

"Universal Computational Logic"



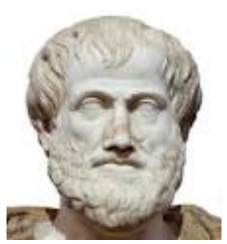
Logic Theorist

(birth of modern logicist AI)

350 BC

Euclid

300 BC



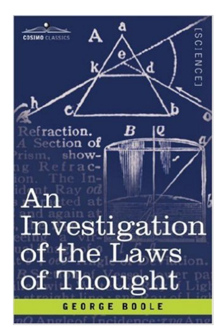
*Organon*

1666

Leibniz

$\int$

1854




1956

Simon

2023

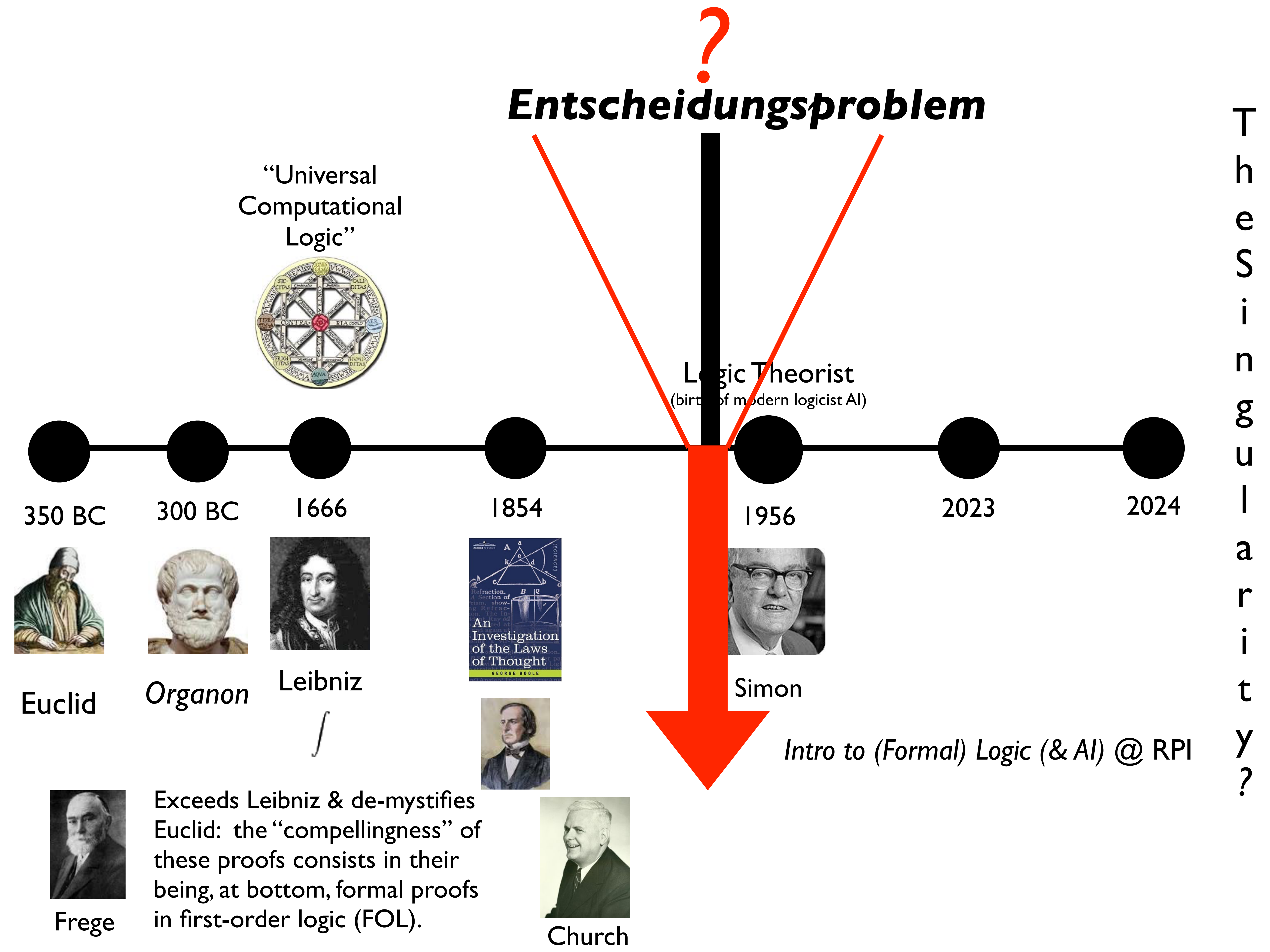
2024

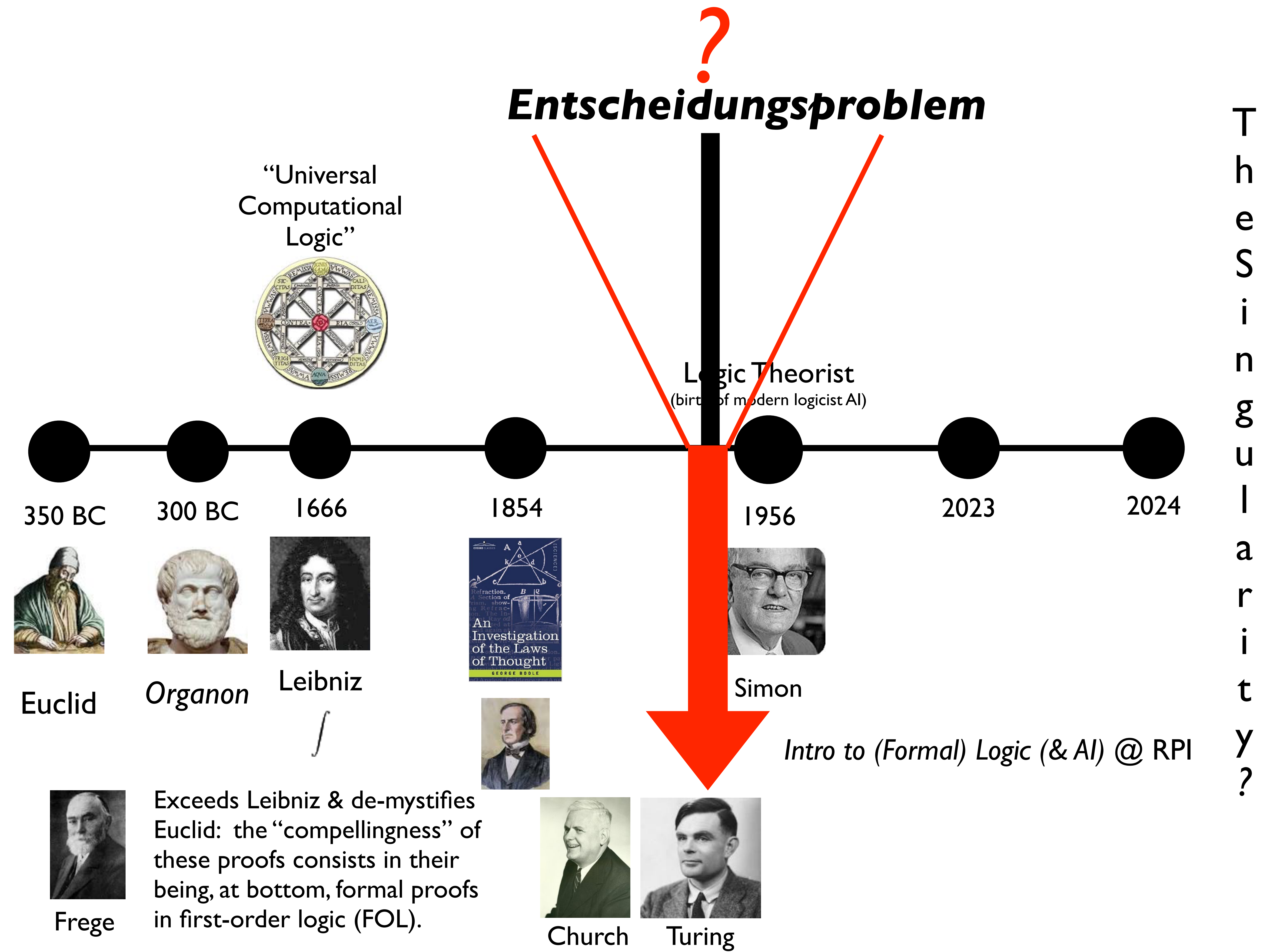
*Intro to (Formal) Logic (& AI)* @ RPI

The Singularity?

Frege

Exceeds Leibniz & de-mystifies Euclid: the "compellingness" of these proofs consists in their being, at bottom, formal proofs in first-order logic (FOL).

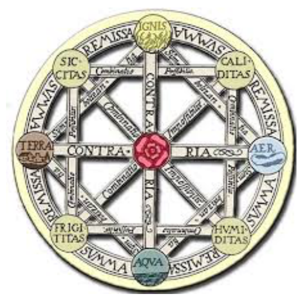

?
Entscheidungsproblem
"Universal Computational Logic"
Logic Theorist
(birth of modern logicist AI)
350 BC
300 BC
1666
1854
1956
2023
2024
Euclid
Organon
Leibniz
∫
An Investigation of the Laws of Thought
GEORGE BOOLE
Simon
Intro to (Formal) Logic (& AI) @ RPI
The Singularity?
Frege
Exceeds Leibniz & de-mystifies Euclid: the "compellingness" of these proofs consists in their being, at bottom, formal proofs in first-order logic (FOL).
Church


?
Entscheidungsproblem
"Universal Computational Logic"
Logic Theorist
(birth of modern logicist AI)
350 BC
300 BC
1666
1854
1956
2023
2024
Euclid
Organon
Leibniz
∫
An Investigation of the Laws of Thought
George Boole
Simon
Intro to (Formal) Logic (& AI) @ RPI
Frege
Exceeds Leibniz & de-mystifies Euclid: the "compellingness" of these proofs consists in their being, at bottom, formal proofs in first-order logic (FOL).
Church
Turing
The Singularity?

?

**Entscheidungsproblem**

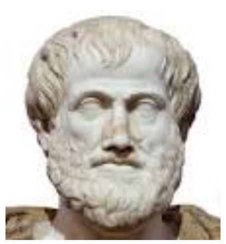
"Universal Computational Logic"



Logic Theorist
(birth of modern logicist AI)

| 350 BC | 300 BC | 1666 | 1854 | 1956 | 2023 | 2024 |
|---|---|---|---|---|---|---|

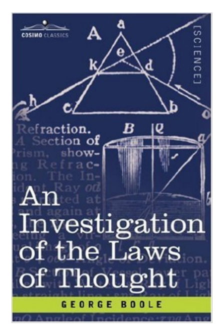
Euclid



*Organon*

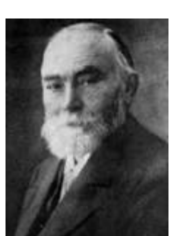
Leibniz

$\int$




Simon

*Intro to (Formal) Logic (& AI)* @ RPI



Frege

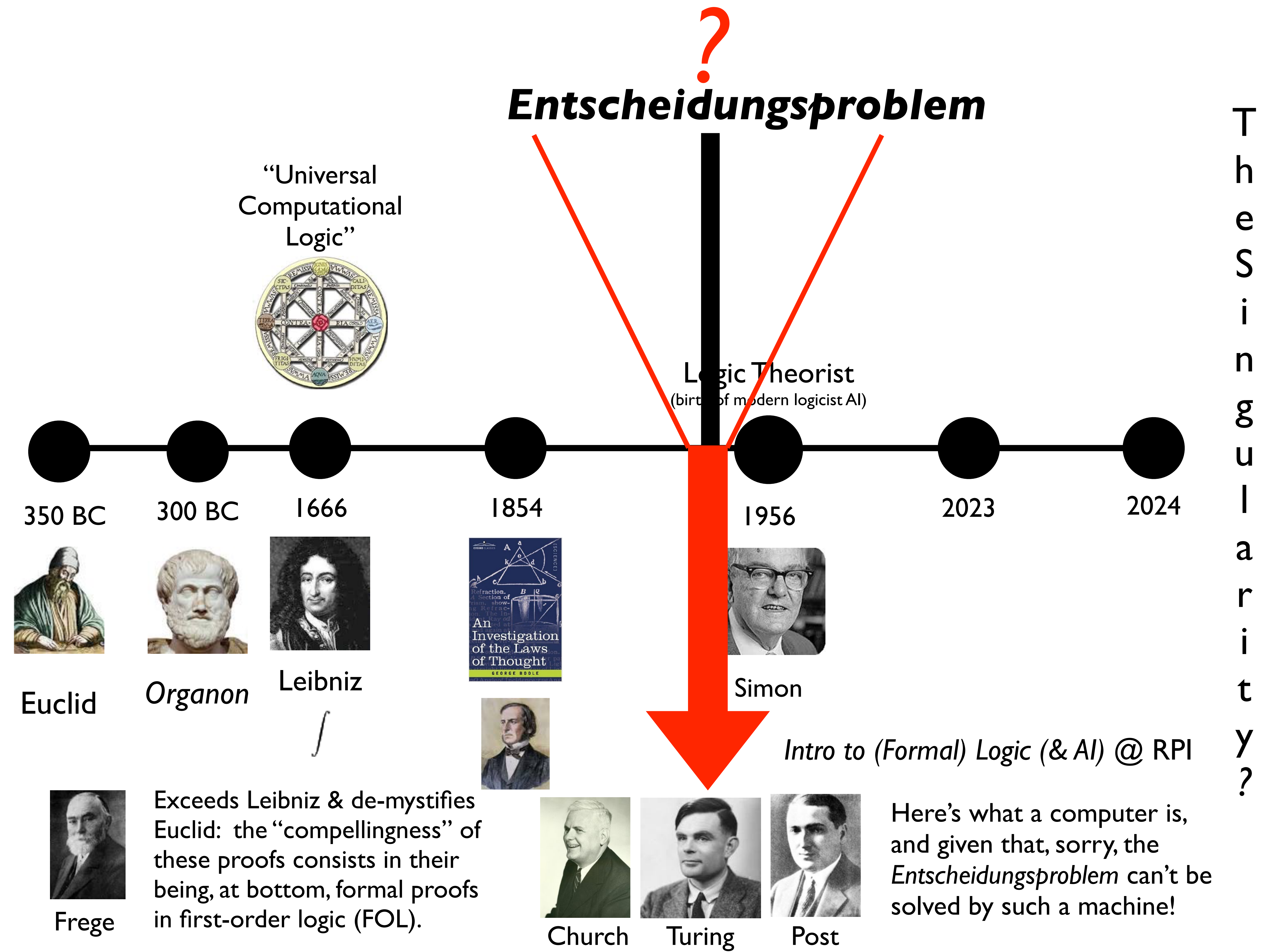
Exceeds Leibniz & de-mystifies Euclid: the "compellingness" of these proofs consists in their being, at bottom, formal proofs in first-order logic (FOL).

Church

Turing

Post

The Singularity?


?
*Entscheidungsproblem*
"Universal Computational Logic"
Logic Theorist
(birth of modern logicist AI)
350 BC
300 BC
1666
1854
1956
2023
2024
Euclid
*Organon*
Leibniz
∫
An Investigation of the Laws of Thought
Simon
*Intro to (Formal) Logic (& AI)* @ RPI
Frege
Exceeds Leibniz & de-mystifies Euclid: the "compellingness" of these proofs consists in their being, at bottom, formal proofs in first-order logic (FOL).
Church
Turing
Post
Here's what a computer is, and given that, sorry, the *Entscheidungsproblem* can't be solved by such a machine!
The Singularity ?

Two HyperGrader®
Required Prop-Calc Problems for **A**
(+ one coming AI-generated
personalized one) …

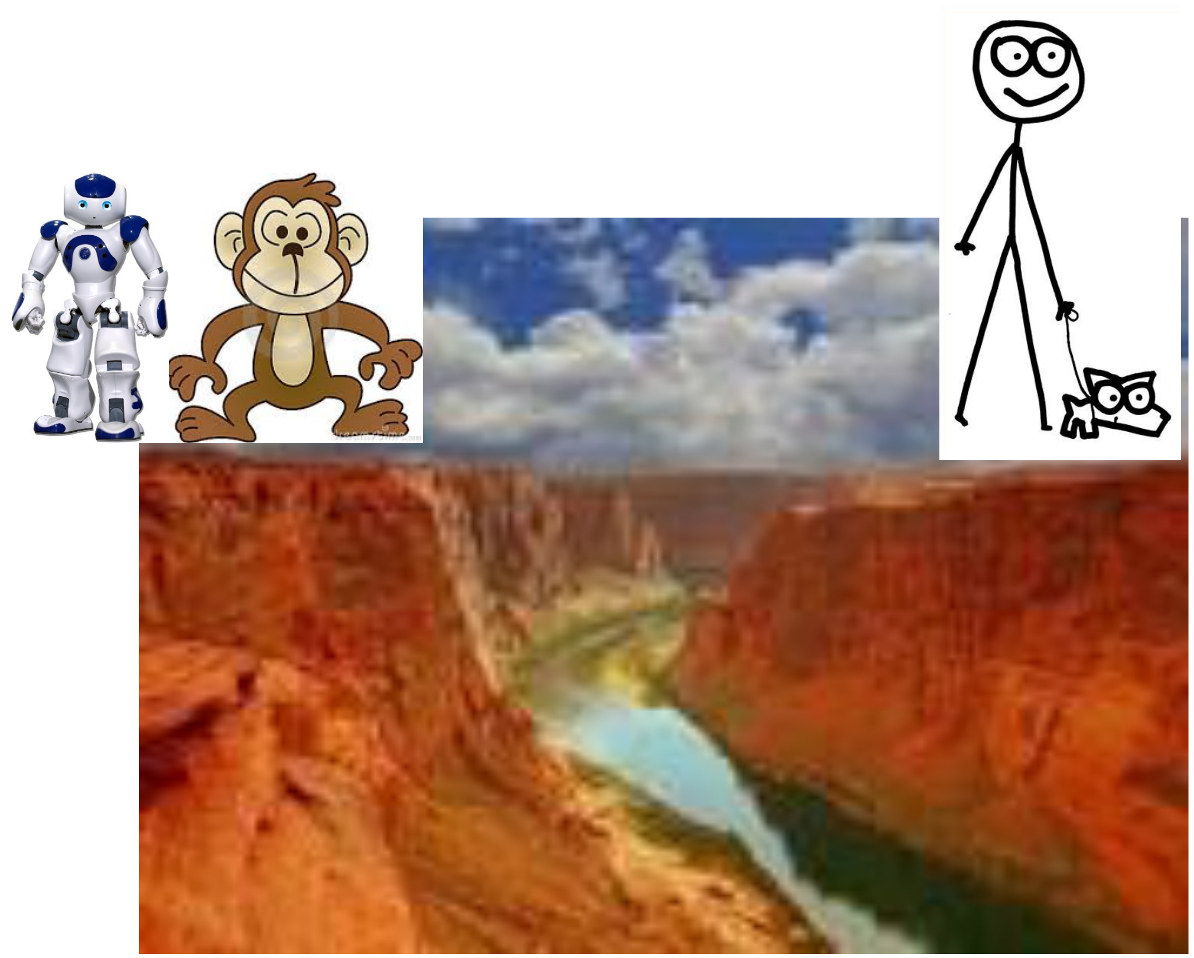
TertiumNonDaturILBAI24

TertiumNonDaturILBAI24

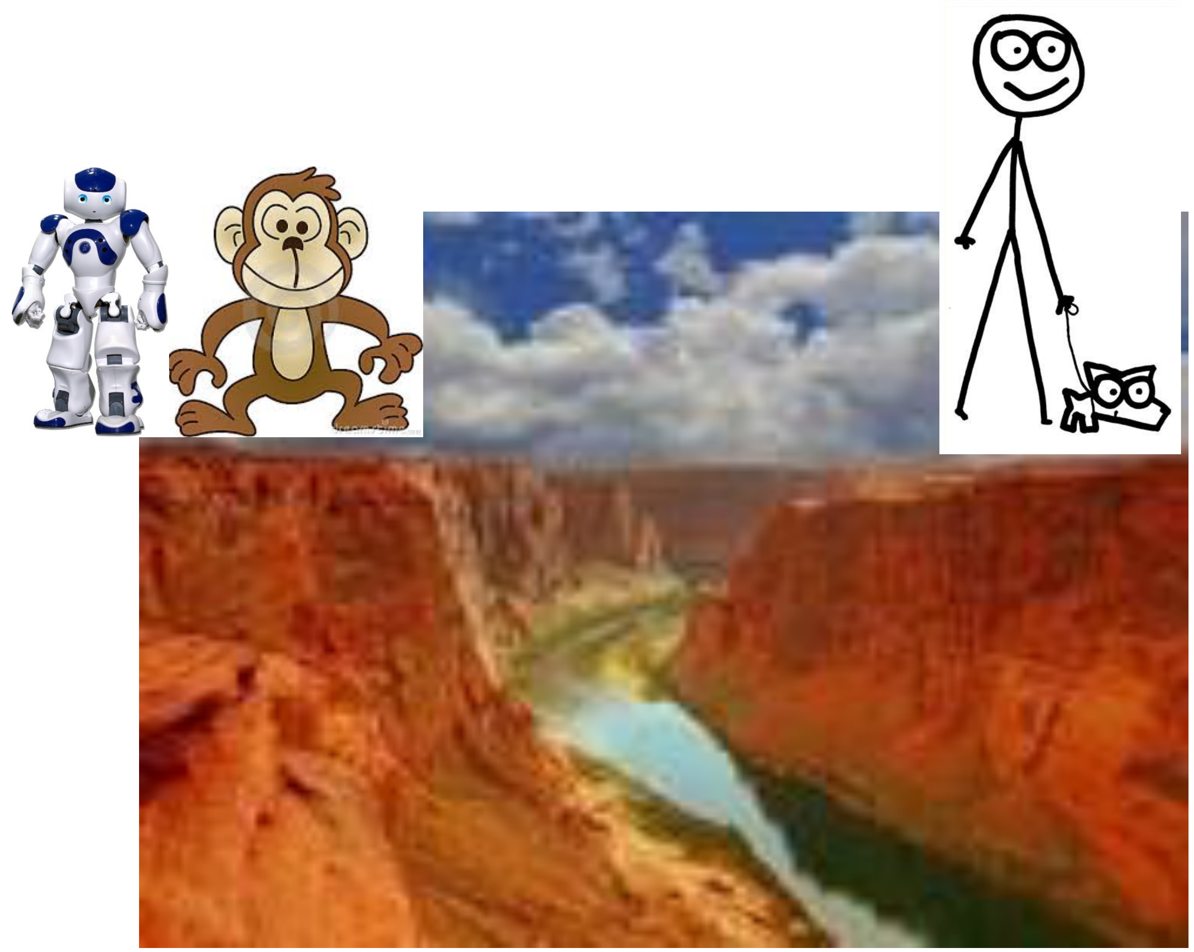
DisjunctiveSyllogismILBAIF24

# Quantifiers (etc) …

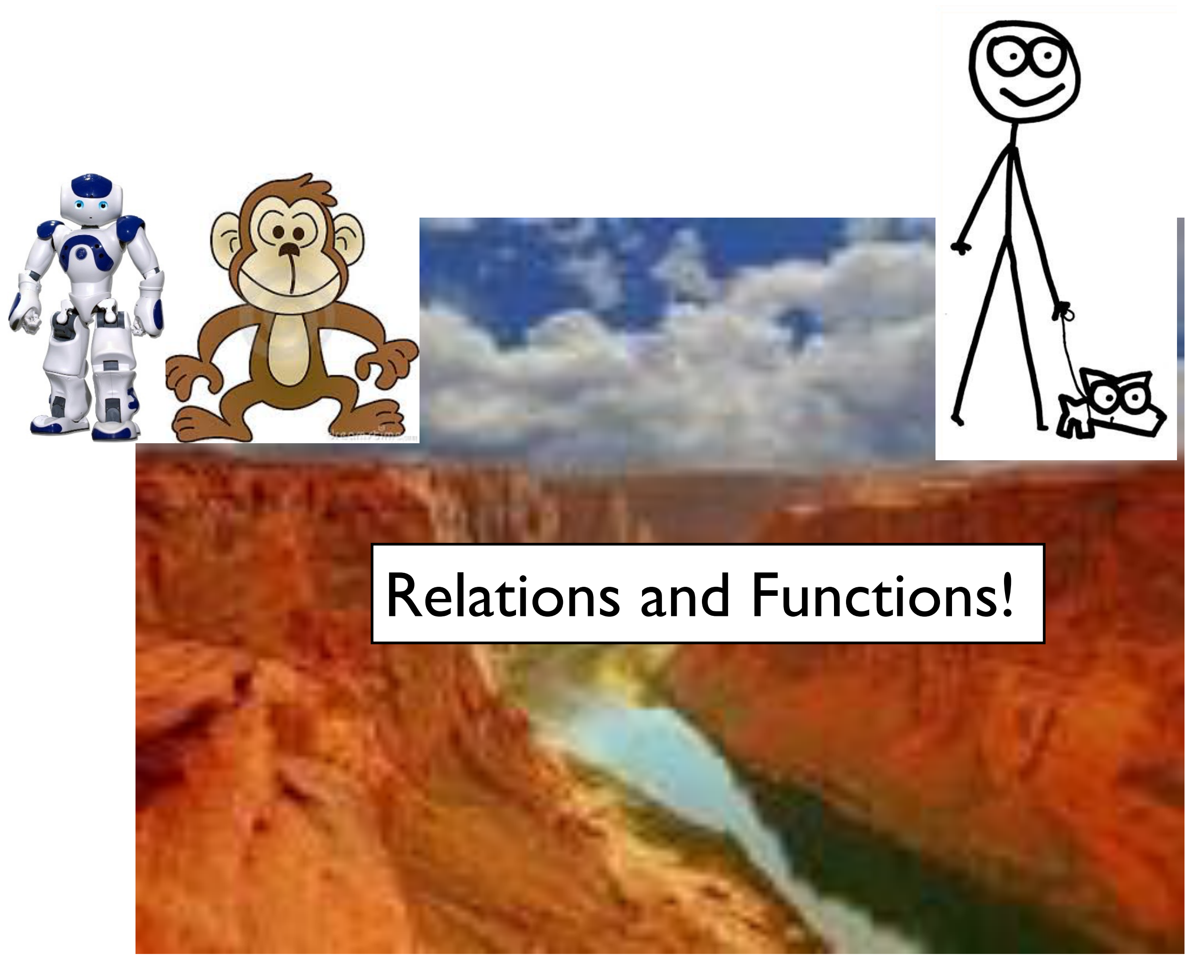
# The Canyon of Discontinuity (or Darwin's Dread)



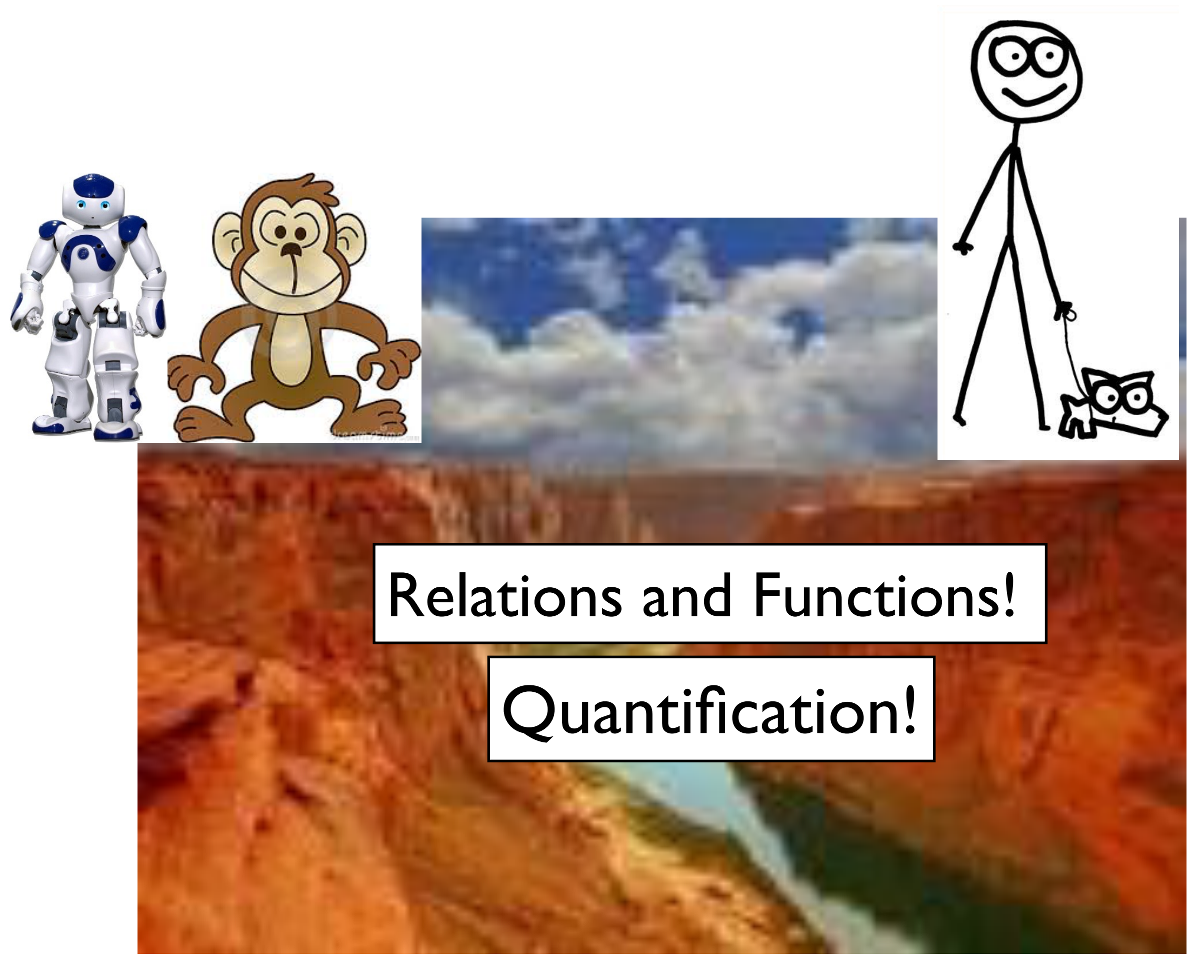
# The Canyon of Discontinuity (or Darwin's Dread)



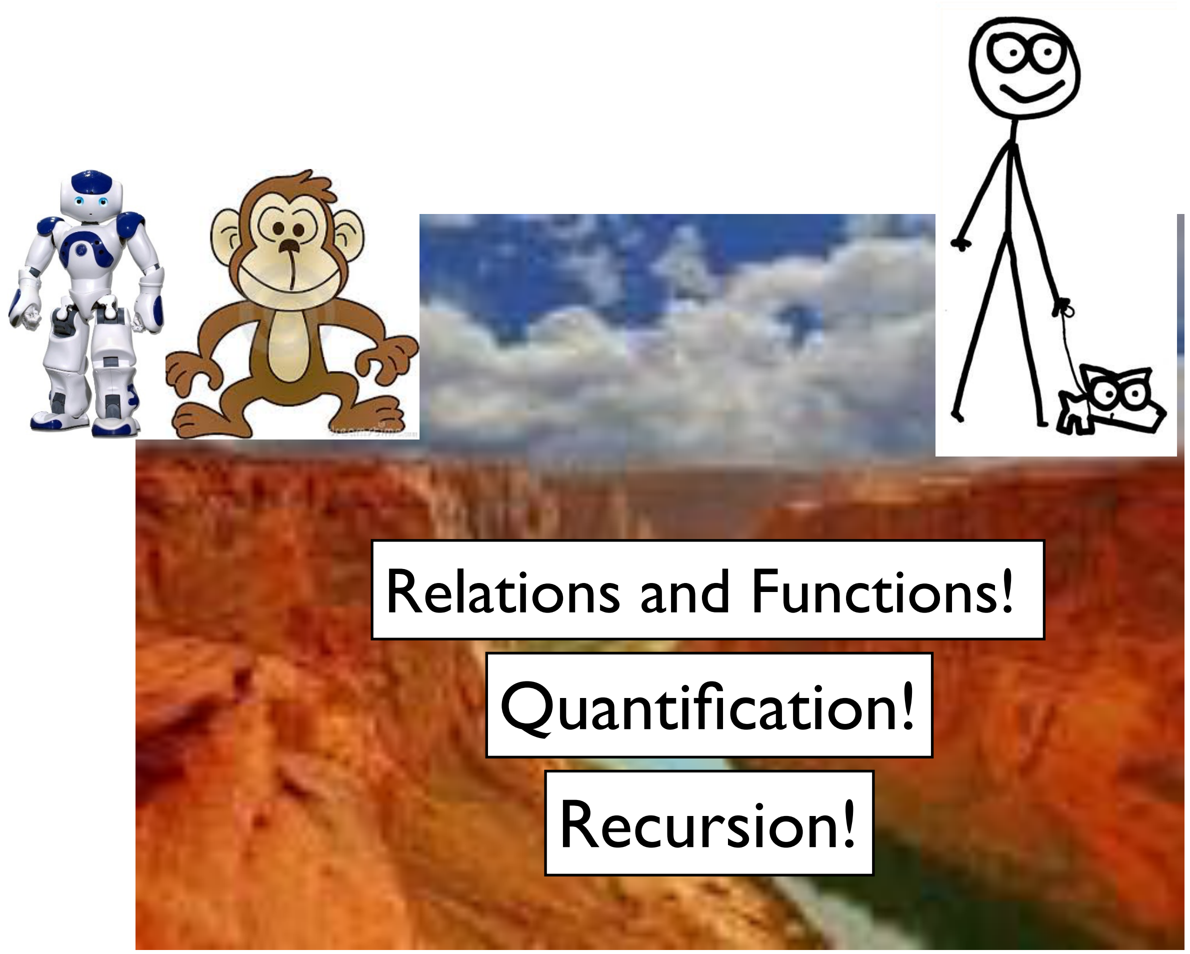
# The Canyon of Discontinuity (or Darwin's Dread)



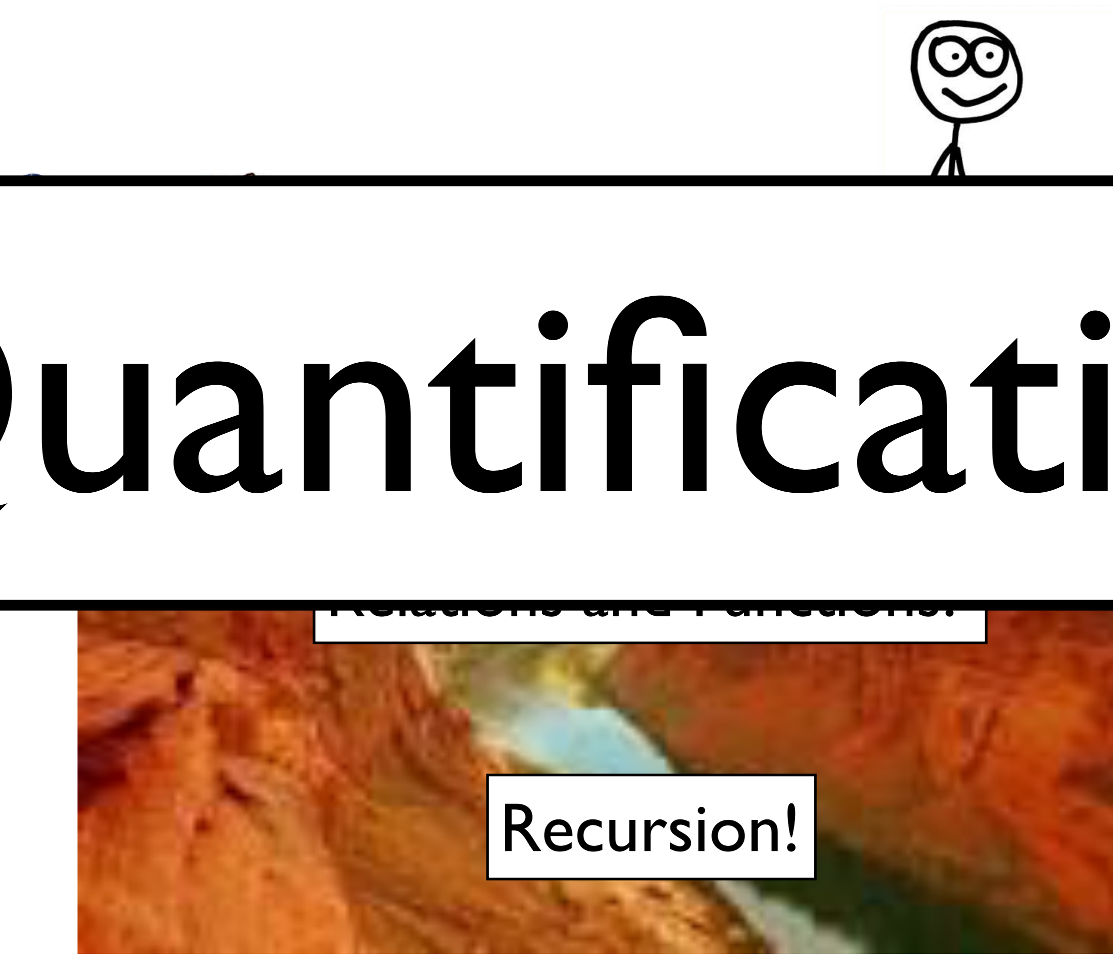
# The Canyon of Discontinuity (or Darwin's Dread)



# The Canyon of Discontinuity (or Darwin's Dread)



# The Canyon of Discontinuity (or Darwin’s Dread)




Quantification!

# Karkooking Problem …

Everyone karkooks anyone who karkooks someone.

Alvin karkooks Bill.

Can you infer that everyone karkooks Bill?

ANSWER:

JUSTIFICATION:

# Karkooking Problem …

Everyone karkooks anyone who karkooks someone.

Alvin karkooks Bill.

Can you infer that everyone karkooks Bill?

ANSWER:

JUSTIFICATION:

# Karkooking Problem …

Everyone karkooks anyone who karkooks someone.

Alvin karkooks Bill.

Can you infer that everyone karkooks Bill?

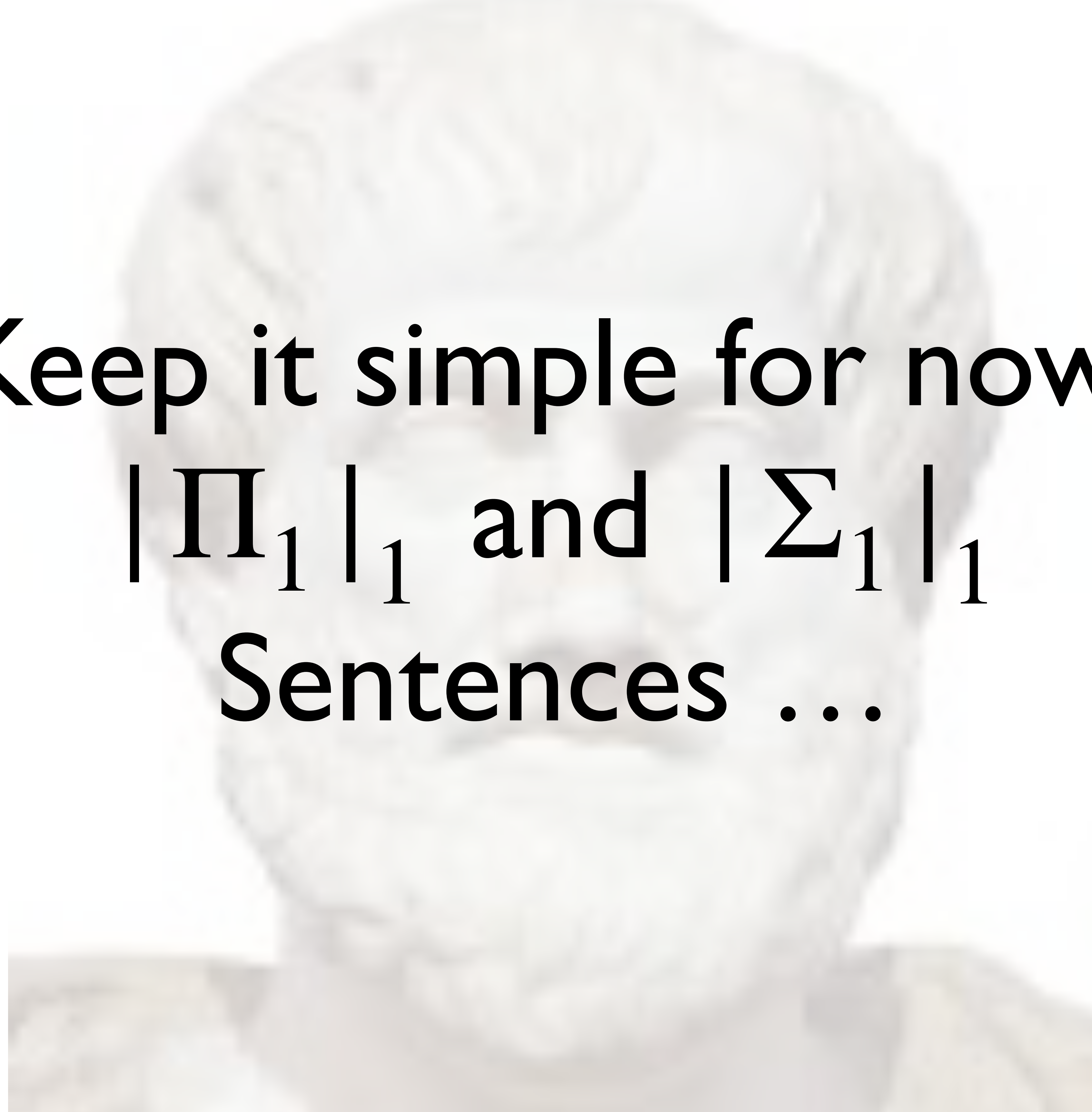
ANSWER:

JUSTIFICATION:

Relations and Functions!

Quantification!

Recursion!

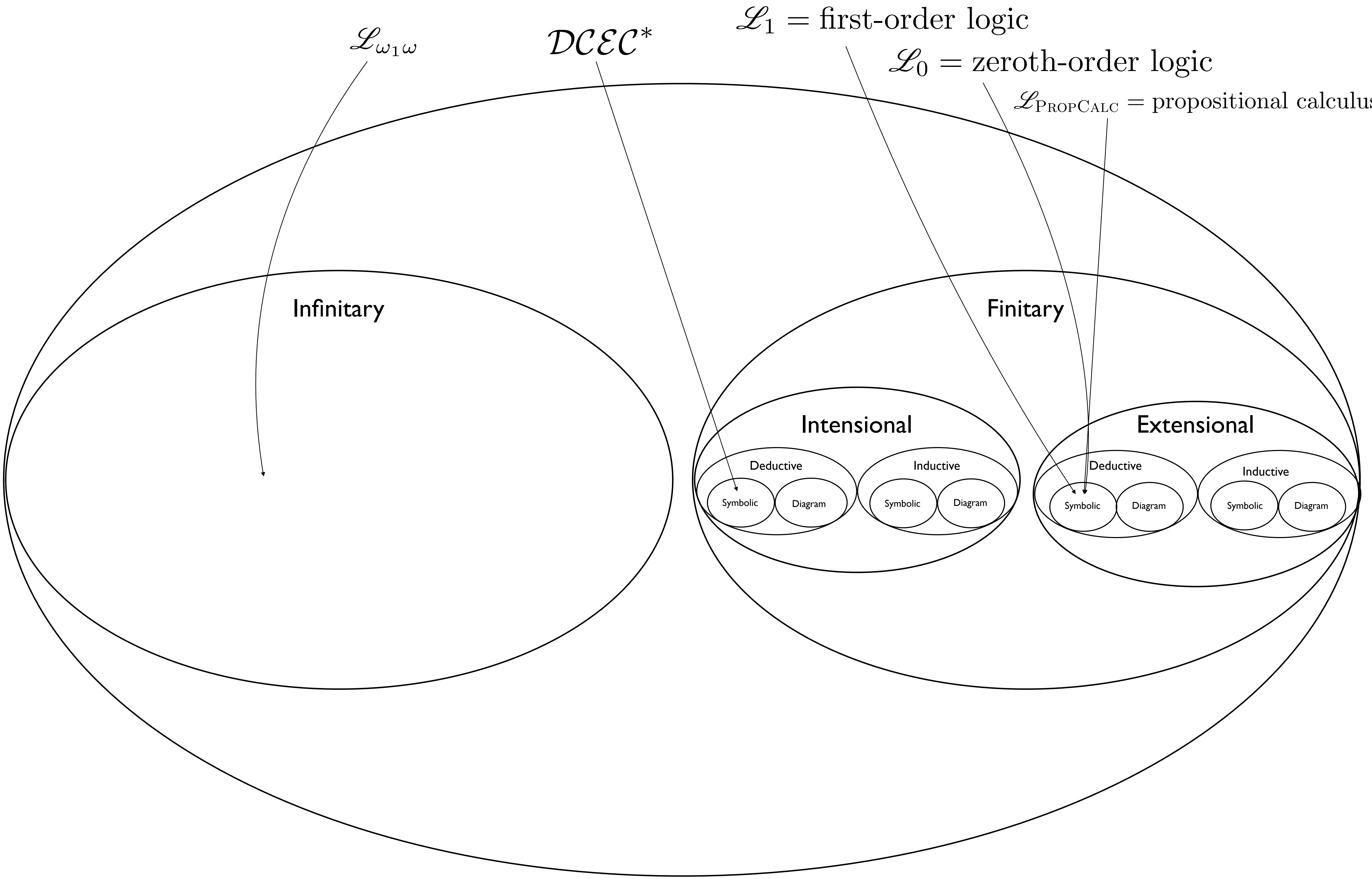


Keep it simple for now!

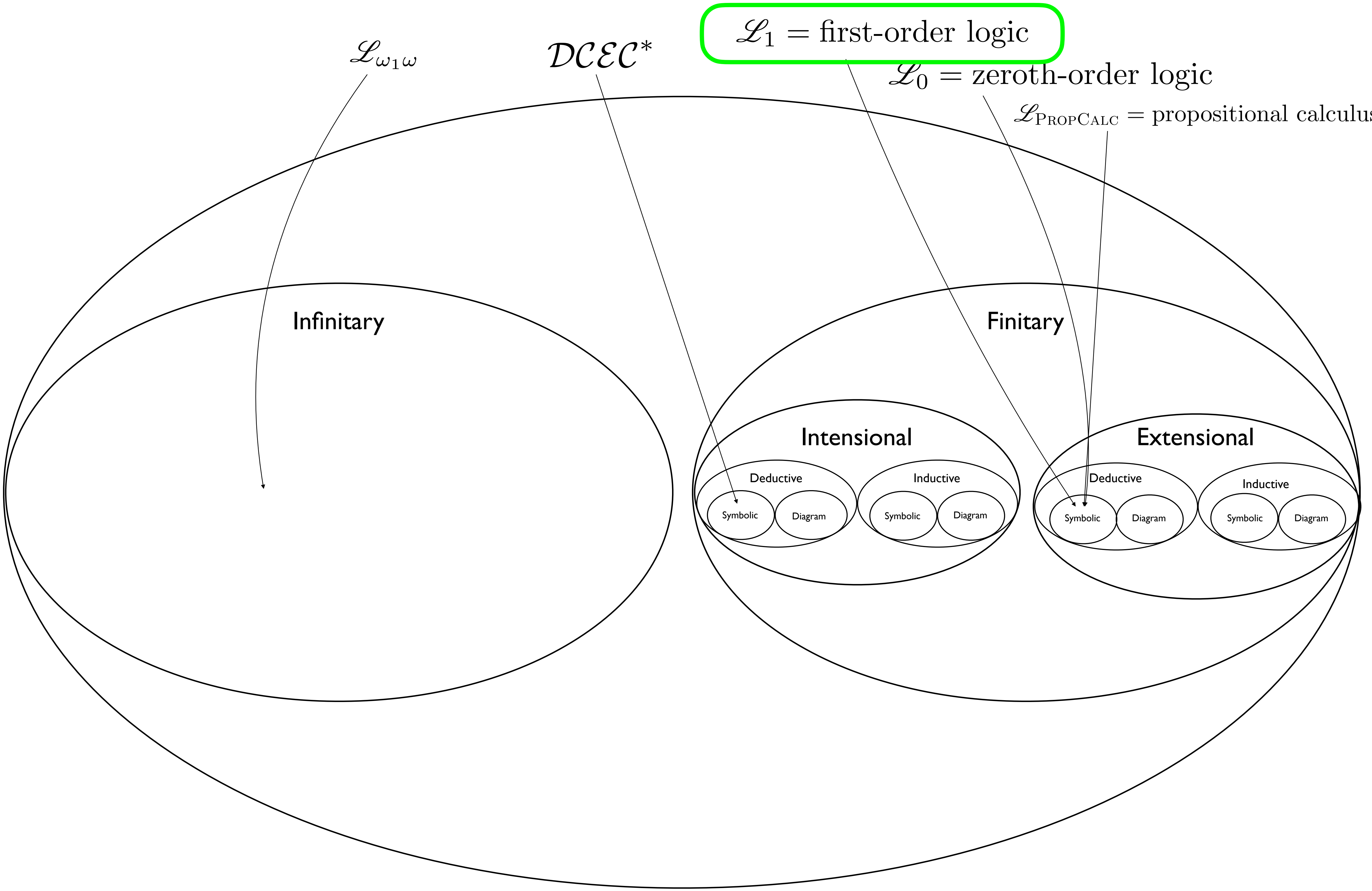
$|\Pi_1|_1$ and $|\Sigma_1|_1$

Sentences …

# The Universe of Logics



# The Universe of Logics



# Two Proposed Arguments; Valid?

- All mammals walk.
- Whales are mammals.
- Therefore:
- Whales walk.

- All of the Frenchmen in the room are wine-drinkers.
- Some of the wine-drinkers in the room are gourmets.
- Therefore:
- Some of the Frenchmen in the room are gourmets.

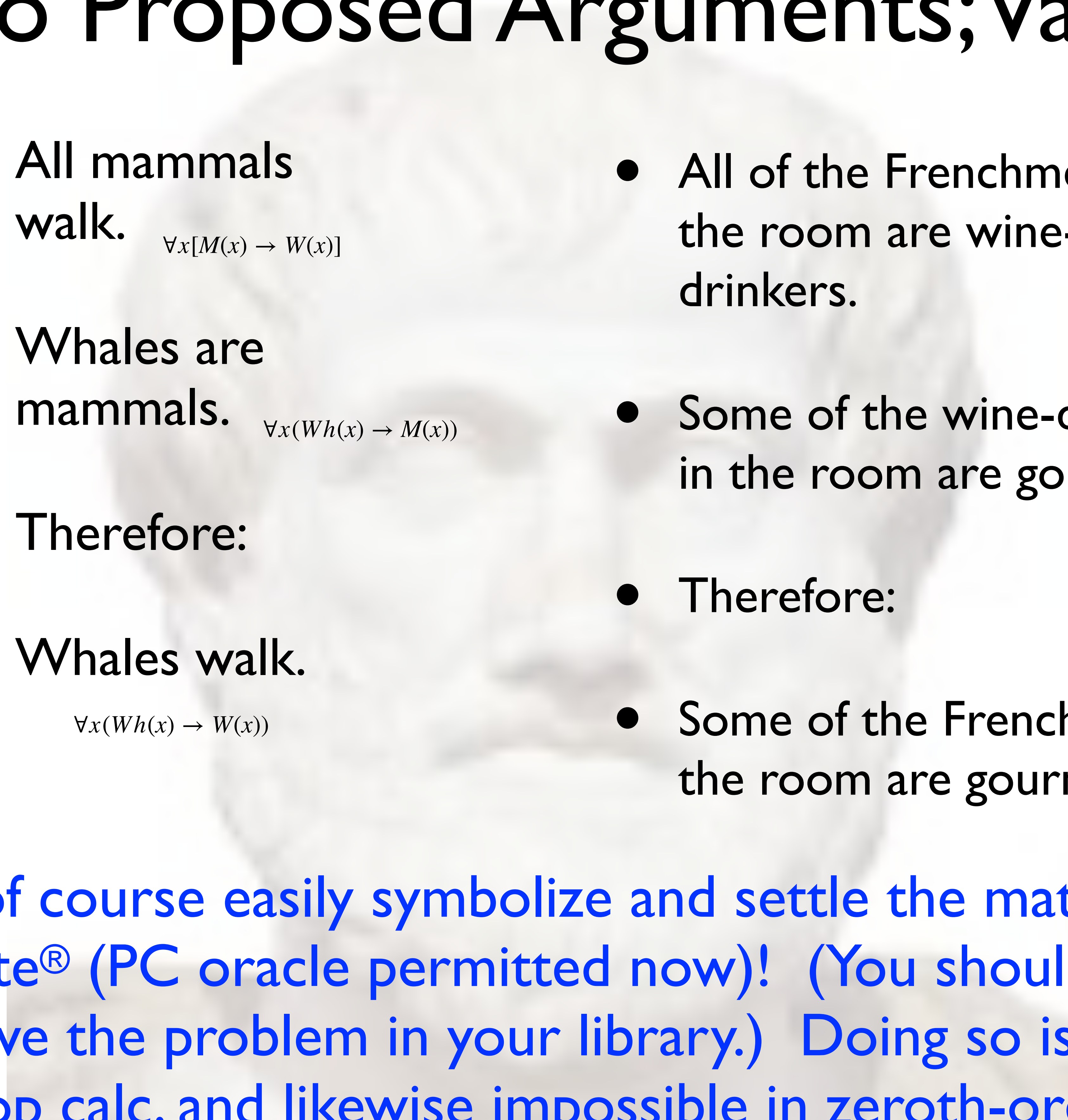
# Two Proposed Arguments; Valid?

- All mammals walk.
- Whales are mammals.
- Therefore:
- Whales walk.

- All of the Frenchmen in the room are wine-drinkers.
- Some of the wine-drinkers in the room are gourmets.
- Therefore:
- Some of the Frenchmen in the room are gourmets.

We can of course easily symbolize and settle the matter in HyperSlate® (PC oracle permitted now)! (You should do this now & save the problem in your library.) Doing so is *impossible* in the prop calc, and likewise impossible in zeroth-order logic!

# Two Proposed Arguments; Valid?



- All mammals walk. $\forall x[M(x) \rightarrow W(x)]$
- Whales are mammals. $\forall x(Wh(x) \rightarrow M(x))$
- Therefore:
- Whales walk.

$\forall x(Wh(x) \rightarrow W(x))$

- All of the Frenchmen in the room are wine-drinkers.
- Some of the wine-drinkers in the room are gourmets.
- Therefore:
- Some of the Frenchmen in the room are gourmets.

We can of course easily symbolize and settle the matter in HyperSlate® (PC oracle permitted now)! (You should do this now & save the problem in your library.) Doing so is *impossible* in the prop calc, and likewise impossible in zeroth-order logic!

# Two Proposed Arguments; Valid?

- All mammals walk. $\forall x[M(x) \rightarrow W(x)]$
- Whales are mammals. $\forall x(Wh(x) \rightarrow M(x))$
- Therefore:
- Whales walk. $\forall x(Wh(x) \rightarrow W(x))$

- All of the Frenchmen in the room are wine-drinkers. $\forall x(F(x) \rightarrow W(x))$
- Some of the wine-drinkers in the room are gourmets. $\exists x(W(x) \wedge G(x))$
- Therefore:
- Some of the Frenchmen in the room are gourmets. $\exists x(F(x) \wedge G(x))$

We can of course easily symbolize and settle the matter in HyperSlate® (PC oracle permitted now)! (You should do this now & save the problem in your library.) Doing so is *impossible* in the prop calc, and likewise impossible in zeroth-order logic!

s-expressions

# Two Proposed Arguments; Valid?

- All mammals walk. $\forall x[M(x) \rightarrow W(x)]$
- Whales are mammals. $\forall x(Wh(x) \rightarrow M(x))$
- Therefore:
- Whales walk. $\forall x(Wh(x) \rightarrow W(x))$

- All of the Frenchmen in the room are wine-drinkers. $\forall x(F(x) \rightarrow W(x))$
- Some of the wine-drinkers in the room are gourmets. $\exists x(W(x) \land G(x))$
- Therefore:
- Some of the Frenchmen in the room are gourmets. $\exists x(F(x) \land G(x))$

We can of course easily symbolize and settle the matter in HyperSlate® (PC oracle permitted now)! (You should do this now & save the problem in your library.) Doing so is *impossible* in the prop calc, and likewise impossible in zeroth-order logic!

s-expressions

# Two Proposed Arguments; Valid?

- All mammals walk. $\forall x[M(x) \rightarrow W(x)]$
- Whales are mammals. $\forall x(Wh(x) \rightarrow M(x))$
- Therefore:
- Whales walk.

$\forall x(Wh(x) \rightarrow W(x))$

- All of the Frenchmen in the room are wine-drinkers. $\forall x(F(x) \rightarrow W(x))$

$\forall x(F(x) \rightarrow W(x))$ • `(forall (x) (if (F x) (W x)))`

- Some of the wine-drinkers in the room are gourmets.

$\exists x(W(x) \land G(x))$

- Therefore:
- Some of the Frenchmen in the room are gourmets.

$\exists x(F(x) \land G(x))$

We can of course easily symbolize and settle the matter in HyperSlate® (PC oracle permitted now)! (You should do this now & save the problem in your library.) Doing so is *impossible* in the prop calc, and likewise impossible in zeroth-order logic!

s-expressions

# Two Proposed Arguments; Valid?

- All mammals walk. $\forall x[M(x) \rightarrow W(x)]$
- Whales are mammals. $\forall x(Wh(x) \rightarrow M(x))$
- Therefore:
- Whales walk.

$\forall x(Wh(x) \rightarrow W(x))$

- All of the Frenchmen in the room are wine-drinkers. $\forall x(F(x) \rightarrow W(x))$

$\forall x(F(x) \rightarrow W(x))$ • `(forall (x) (if (F x) (W x)))`

- Some of the wine-drinkers in the room are gourmets.

$\exists x(W(x) \land G(x))$

$\exists x(W(x) \land G(x))$ • `(exists (x) (and (W x) (G x)))`

- Therefore:
- Some of the Frenchmen in the room are gourmets.

$\exists x(F(x) \land G(x))$

We can of course easily symbolize and settle the matter in HyperSlate® (PC oracle permitted now)! (You should do this now & save the problem in your library.) Doing so is *impossible* in the prop calc, and likewise impossible in zeroth-order logic!

s-expressions

# Two Proposed Arguments; Valid?

- All mammals walk. $\forall x[M(x) \rightarrow W(x)]$
- Whales are mammals. $\forall x(Wh(x) \rightarrow M(x))$
- Therefore:
- Whales walk.

$\forall x(Wh(x) \rightarrow W(x))$

- All of the Frenchmen in the room are wine-drinkers. $\forall x(F(x) \rightarrow W(x))$

$\forall x(F(x) \rightarrow W(x))$ • `(forall (x) (if (F x) (W x)))`

- Some of the wine-drinkers in the room are gourmets.

$\exists x(W(x) \land G(x))$

$\exists x(W(x) \land G(x))$ • `(exists (x) (and (W x) (G x)))`

- Therefore:
- Some of the Frenchmen in the room are gourmets.

$\exists x(F(x) \land G(x))$

$\exists x(F(x) \land G(x))$ • `(exists (x) (and (F x) (G x)))`

We can of course easily symbolize and settle the matter in HyperSlate® (PC oracle permitted now)! (You should do this now & save the problem in your library.) Doing so is *impossible* in the prop calc, and likewise impossible in zeroth-order logic!

# (Pure General) Logic Programming …

$$\mathcal{L} := \langle L, \mathcal{I} \rangle$$

$$\frac{\begin{array}{cc} \mathbb{P} & L \\ \mathfrak{q} & L \end{array}}{\begin{array}{c} \mathbb{R} \;:\; \langle \mathbb{P}, \mathfrak{q} \rangle \longrightarrow \langle \mathtt{Y} | \mathtt{N} | \mathtt{U}, \delta, \pi_{(s)} | \alpha_{(s)} \rangle \\ \mathbb{C} \;:\; \pi_{(s)} | \alpha_{(s)} \longrightarrow \langle \mathtt{Y} | \mathtt{N} | \mathtt{U}, \delta \rangle \end{array}}$$

$$\mathcal{L} := \langle L, \mathcal{I} \rangle$$

$$\frac{\begin{array}{cc} \mathbb{P} & L \\ \mathfrak{q} & L \end{array}}{\begin{array}{l} \mathbb{R} \;:\; \langle \mathbb{P}, \mathfrak{q} \rangle \longrightarrow \langle \mathtt{Y}|\mathtt{N}|\mathtt{U}, \delta, \pi_{(s)}|\alpha_{(s)} \rangle \\ \mathbb{C} \;:\; \pi_{(s)}|\alpha_{(s)} \longrightarrow \langle \mathtt{Y}|\mathtt{N}|\mathtt{U}, \delta \rangle \end{array}}$$

$$\mathcal{L} := \langle L, \mathcal{I} \rangle$$

$$\frac{\mathbb{P} \quad L \qquad \mathfrak{q} \quad L}{\begin{array}{l} \mathbb{R} \; : \; \langle \mathbb{P}, \mathfrak{q} \rangle \longrightarrow \langle \mathtt{Y}|\mathtt{N}|\mathtt{U}, \delta, \pi_{(s)}|\alpha_{(s)} \rangle \\ \mathbb{C} \; : \; \pi_{(s)}|\alpha_{(s)} \longrightarrow \langle \mathtt{Y}|\mathtt{N}|\mathtt{U}, \delta \rangle \end{array}}$$

program

query

degree of "confidence"

proof(s)

argument(s)

reasoner

checker

For just "logic programming," and a vintage approach that goes back to circa 1970, restrict this to FOL or a fragment thereof, and use resolution as the only inference schema.

$$\mathcal{L} := \langle L, \mathcal{I} \rangle$$

$$\frac{\mathbb{P} \quad L \qquad \mathfrak{q} \quad L}{\begin{array}{l}\mathbb{R} \; : \; \langle \mathbb{P}, \mathfrak{q} \rangle \longrightarrow \langle \mathtt{Y}|\mathtt{N}|\mathtt{U}, \delta, \pi_{(s)}|\alpha_{(s)} \rangle \\ \mathbb{C} \; : \; \pi_{(s)}|\alpha_{(s)} \longrightarrow \langle \mathtt{Y}|\mathtt{N}|\mathtt{U}, \delta \rangle\end{array}}$$

program

query

degree of "confidence"

proof(s)

argument(s)

reasoner

checker

For just "logic programming," and a vintage approach that goes back to circa 1970, restrict this to FOL or a fragment thereof, and use resolution as the only inference schema.

$$\mathcal{L} := \langle L, \mathcal{I} \rangle$$

program

query

degree of "confidence"

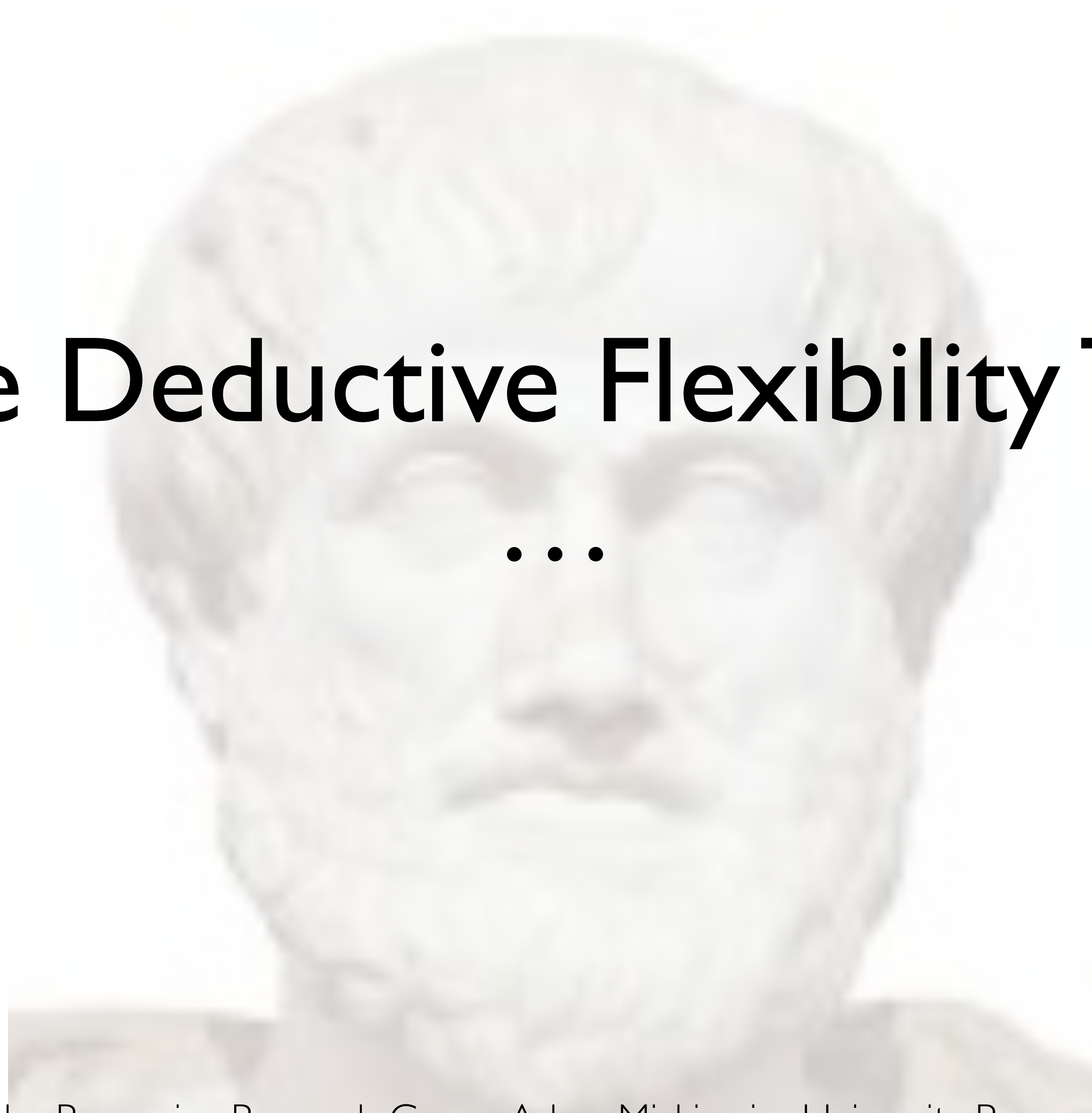
proof(s)

argument(s)

$$\frac{\begin{array}{cc} \mathbb{P} & L \\ \mathfrak{q} & L \end{array}}{\begin{array}{l} \mathbb{R} \;:\; \langle \mathbb{P}, \mathfrak{q} \rangle \longrightarrow \langle \mathtt{Y}|\mathtt{N}|\mathtt{U}, \delta, \pi_{(s)}|\alpha_{(s)} \rangle \\ \quad \mathbb{C} \;:\; \pi_{(s)}|\alpha_{(s)} \longrightarrow \langle \mathtt{Y}|\mathtt{N}|\mathtt{U}, \delta \rangle \end{array}}$$

reasoner

checker



# The Deductive Flexibility Test

. . .

From the Reasoning Research Group, Adam Mickiewicz University, Poznan, Poland
http://murbansk-rrg.home.amu.edu.pl/research-projects/deductive-flexibility-test

From the Reasoning Research Group, Adam Mickiewicz University, Poznan, Poland
http://murbansk-rrg.home.amu.edu.pl/research-projects/deductive-flexibility-test

**Example with correct answer:**

(1) Some nedges are dwens.
(2) Some strimes are dwens.
(3) Every nedge is a strime.
(4) Every nedge is a dwen.
(5) This is not the case, that no dwen is a strime.

Some dwens are strimes.

**Answer**: (2); (5); (3 and 4); (1 and 4)

# Correct? To HyperSlate® …

I (1) Every faub is a non-veece.
(2) No phodd is a veece.
(3) No veece is a phodd.
(4) Every veece is a non-faub.
(5) Every faub is a phodd.

---

No faub is a veece.

**Answer**:________________________

*Den rasjonelle delen av menneskesinnet er basert på logikk.*