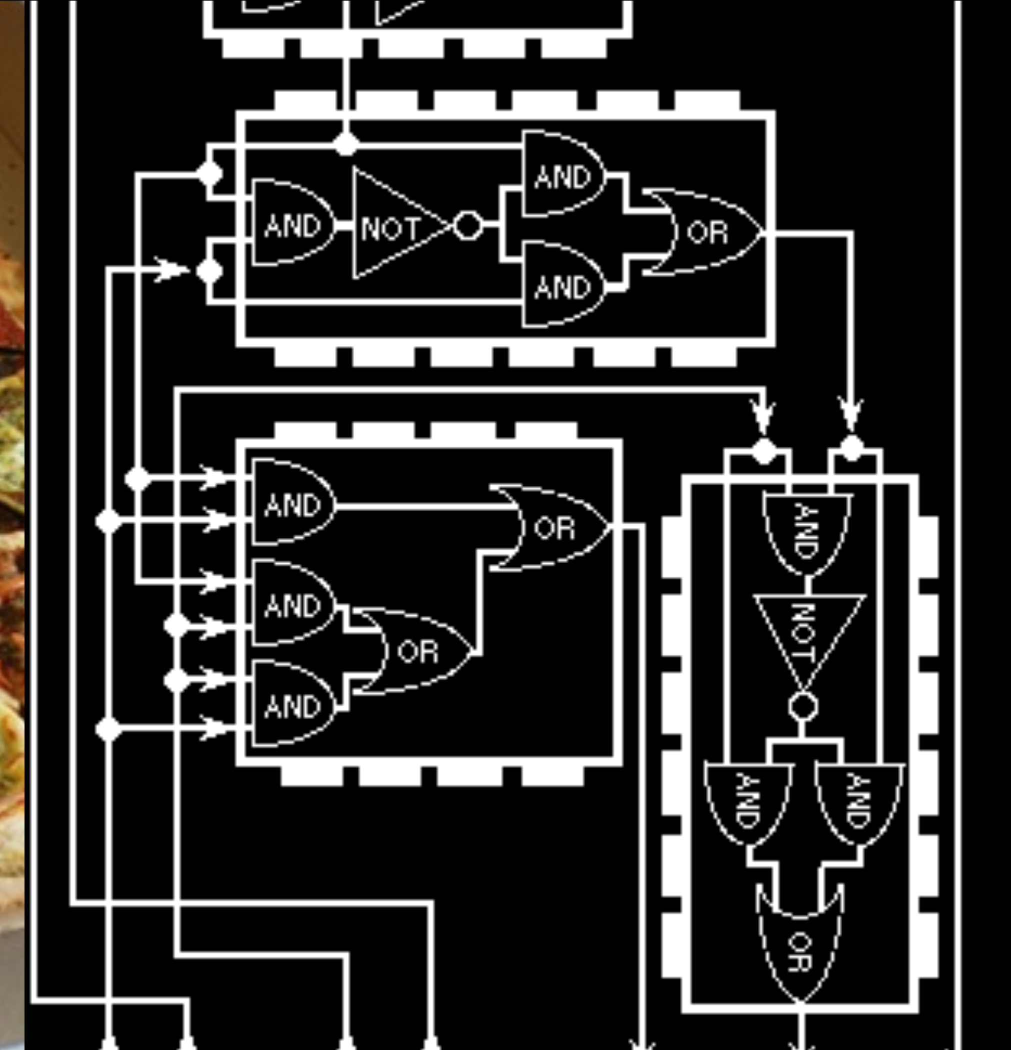
George Boole

Recursive Pizza



2+3

PHIL 50 - Introduction to Logic

Marcello Di Bello, Stanford University, Spring 2014

Week 2 — Monday Class

Today we Begin with the Simplest Logical System: **Propositional Logic**

Syntax: rules to build well-formed formulas

Semantics: rules to assign (truth) values to these formulas

SYNTAX of the Propositional Language

Ingredients of the Propositional Language

1. Basic (*atomic*) statements (**propositions**):

$$\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}, \ldots$$

2. Operators to build more statements:

"**not** ... "	becomes	$\neg \ldots$
"... **and** ..."	becomes	$\ldots \wedge \ldots$
"... **or** ..."	becomes	$\ldots \vee \ldots$
"**if** ... **then**"	becomes	$\ldots \rightarrow \ldots$
" ... **if and only if** ..."	becomes	$\ldots \leftrightarrow \ldots$

Well-Formed Formulas

The **language** $\mathcal{L}_{\mathtt{P}}$ is a set of formulas satisfying:

1. All the basic propositions are in $\mathcal{L}_{\mathtt{P}}$:

$$\boldsymbol{p} \in \mathcal{L}_{\mathtt{P}}, \quad \boldsymbol{q} \in \mathcal{L}_{\mathtt{P}}, \quad \boldsymbol{r} \in \mathcal{L}_{\mathtt{P}}, \quad \ldots$$

2. If $\varphi \in \mathcal{L}_{\mathtt{P}}$ and $\psi \in \mathcal{L}_{\mathtt{P}}$, then

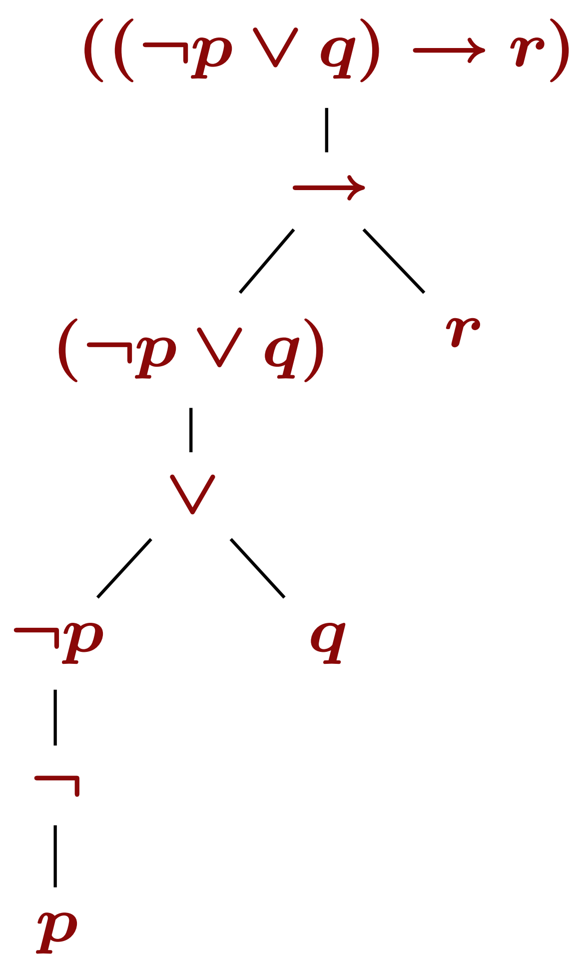
$$\begin{array}{lll} \neg\varphi \in \mathcal{L}_{\mathtt{P}}, & (\varphi \wedge \psi) \in \mathcal{L}_{\mathtt{P}}, & (\varphi \longrightarrow \psi) \in \mathcal{L}_{\mathtt{P}}, \\ & (\varphi \vee \psi) \in \mathcal{L}_{\mathtt{P}}, & (\varphi \leftrightarrow \psi) \in \mathcal{L}_{\mathtt{P}}. \end{array}$$

3. Nothing else is in $\mathcal{L}_{\mathtt{P}}$.

In practice, we will avoid parenthesis if they are not necessary.

Formulas as Trees

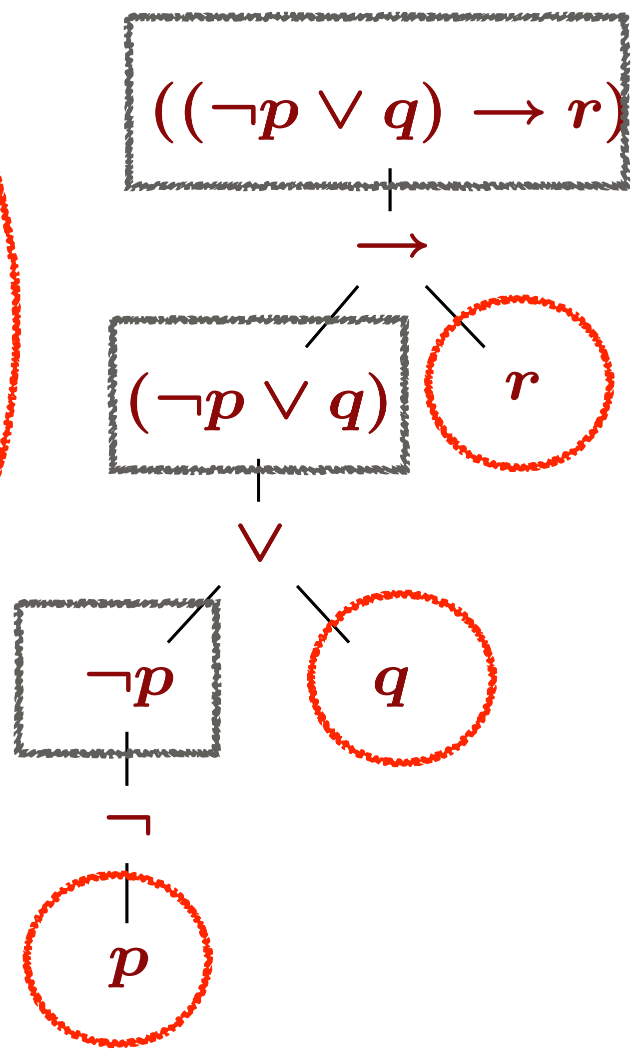
The construction of a formula can be seen as building a **tree**.

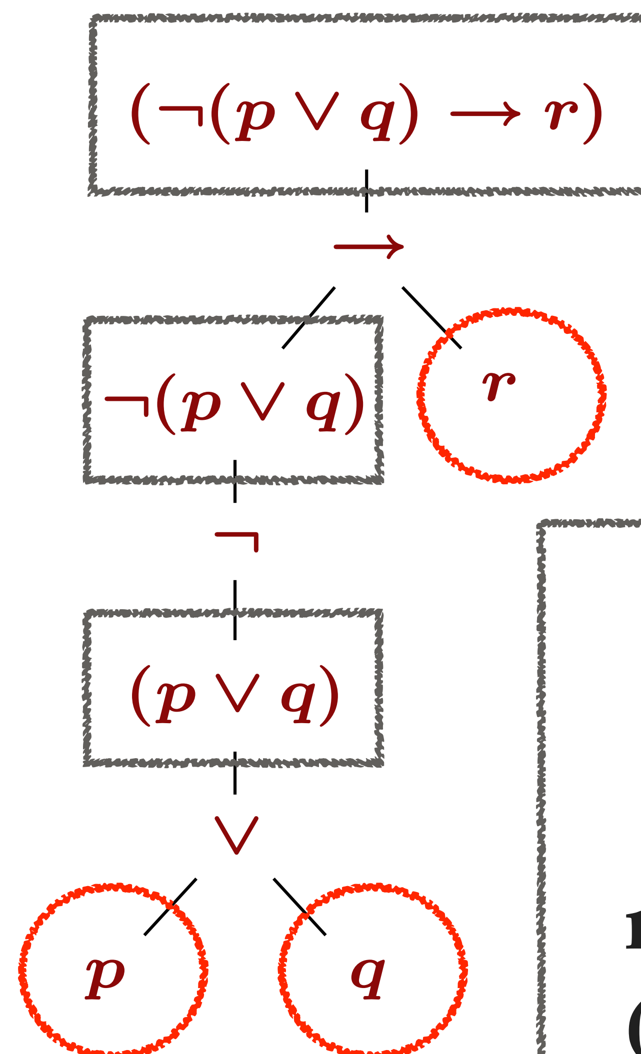


Formulas as Trees

The construction of a formula can be seen as building a **tree**.

The formulas that are circled in red are **basic (or atomic) formulas**





The formulas within a grey rectangle are **more complex (or molecular) formulas**

Formulas Are Defined Inductively or Recursively

What does
that mean?

Inductive (or Recursive) Definitions (1)

Inductive definition of the set of natural numbers

Base case:

1 is a natural number

Inductive case:

If **n** is a natural number, **n+1** is a natural number

Final clause:

Nothing else is a natural number

Inductive (or Recursive) Definitions (2)

`Inductive definition of the set of formulas of Lp`

Base case:

p, q, r … are formulas of **Lp**.

Inductive case(s):

If ϕ formula of **Lp**, then ¬ϕ is a formula of **Lp**

If ϕ and ψ are formulas of **Lp**, then ϕ ∧ ψ is a formula of **Lp**

…. *and so on for the other connectives*

Final clause:

Nothing else is a formula of **Lp**

ϕ and ψ are not formulas; they are **schemata for formulas**. This “trick” makes the definition possible.

Inductive (or Recursive) Definitions (3)

Inductive (or recursive) definitions are **somewhat circular** in the sense that **they define something in terms of itself.**

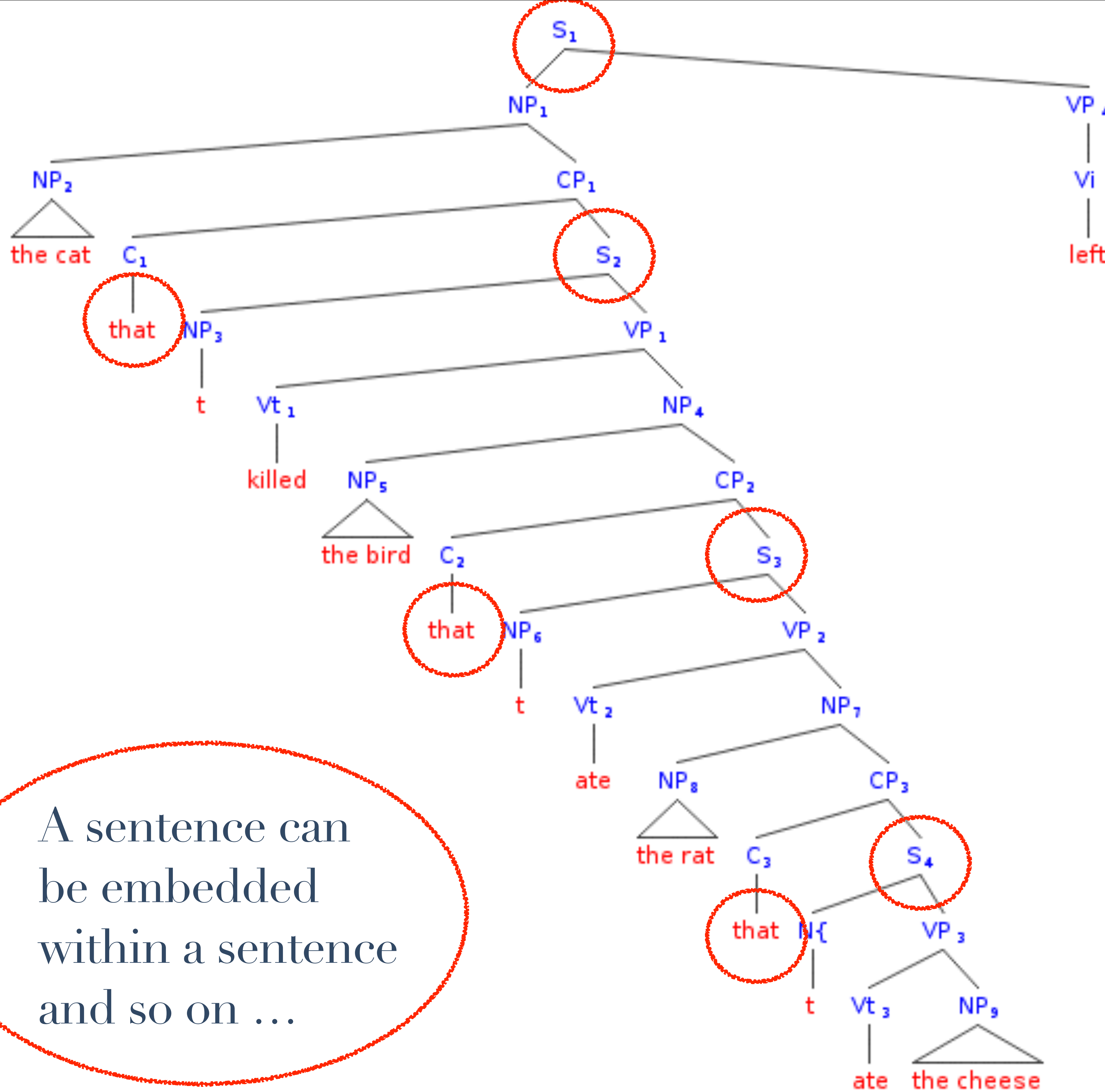
Look at the **inductive case(s)**:

A natural number is defined in terms of a natural number.

A formula is defined in terms of a formula.

But there are **no vicious circles** because of the **base case**.

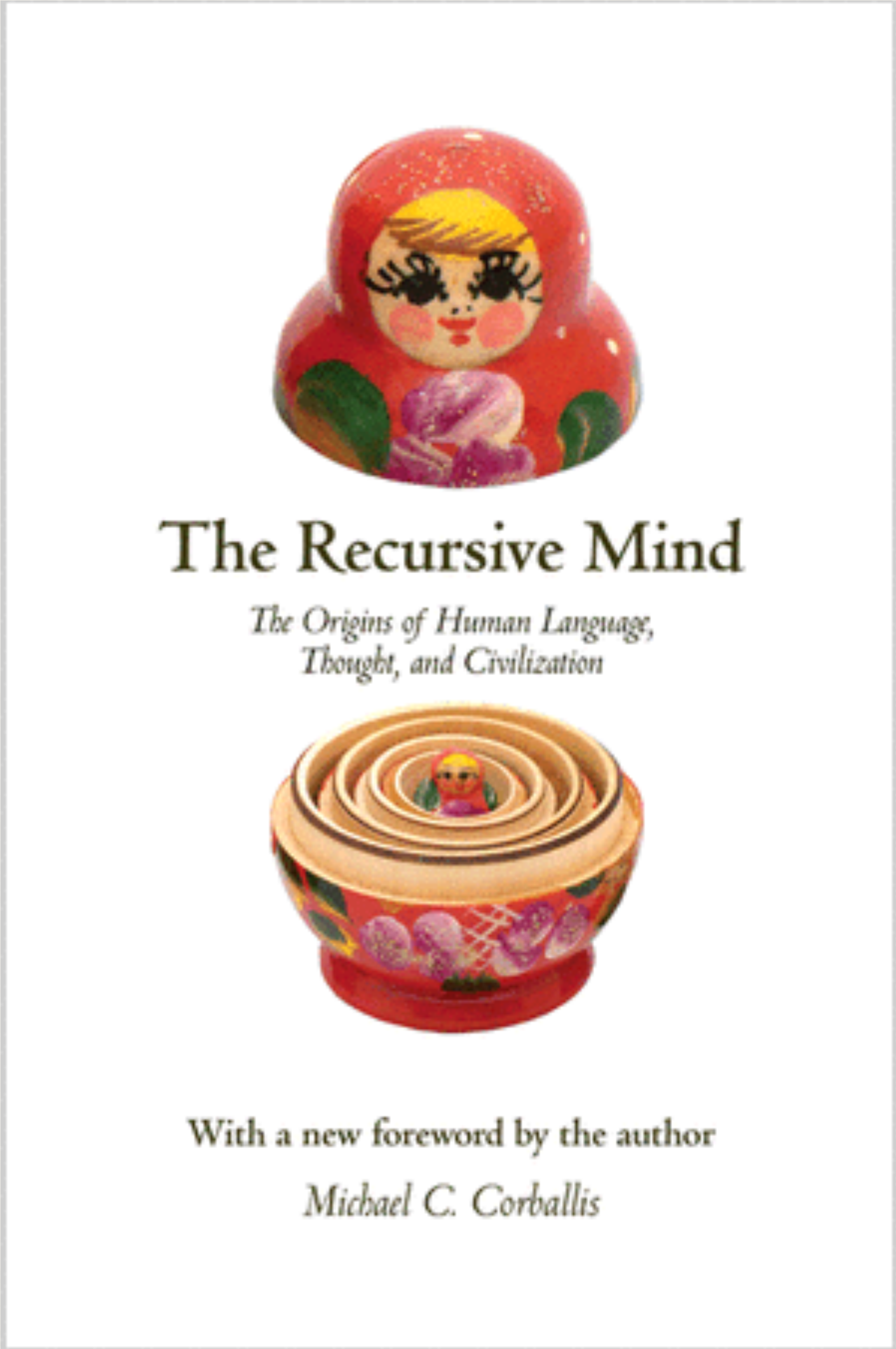
Recursion in the Grammar of Natural Language Sentences



A sentence can be embedded within a sentence and so on …

The Recursive Pizza

…and The Recursive Mind



SEMANTICS of the Propositional Language

Evaluating Formulas

How do we know if a given formula φ is **true** or **false**?

- We need the **truth-values** of the basic propositions $\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{r}$, ... that appear in φ.
- We need to know the **meaning** of $\neg$, $\wedge$, $\vee$, $\longrightarrow$ and $\leftrightarrow$.

Valuation Functions

This encodes the **principe of bivalence**. For every atomic propositions is assigned value 1 or 0.

Valuation. Let $\mathtt{P} = \{p, q, r, \ldots\}$ be a set of atomic propositions. A **valuation** V from $\mathtt{P}$ to $\{0, 1\}$ assigns to each element of $\mathtt{P}$ a unique truth-value.

Example: assume $\mathtt{P} = \{p, q\}$.
There are **four** different valuations (**four** different situations):

$V_1(p) = 1$	$V_1(q) = 1$
$V_2(p) = 1$	$V_2(q) = 0$
$V_3(p) = 0$	$V_3(q) = 1$
$V_4(p) = 0$	$V_4(q) = 0$

How MANY Valuations Functions?

With **one** atomic proposition, there are **two** possible valuations.

With **two** atomic propositions, there are **four** possible valuations.

With **three** atomic propositions, there are **2^3=8** possible valuations.

With **n** atomic propositions, there are **2^n** possible valuations.

So Far We Have Only Assigned Truth Values to Atomic Formulas

How can we assign truth values to more complex formulas?

Extending V for Negation

Use 1 for **true**, and 0 for **false**.

For **negation** $\neg$

φ	$\neg\varphi$
1	**0**
0	**1**

or, in a shorter format:

$\neg$	φ
0	1
1	0

Negation behaves like the 1-place function **1-x=y**.

Extending V for Conjunction and Disjunction

For **conjunction** $\wedge$

φ	$\wedge$	ψ
1	**1**	1
1	**0**	0
0	**0**	1
0	**0**	0

Conjunction behaves like the 2-place functions ***(x_1 · x_2)=y*** and ***min(x_1, x_2)=y.***

For **disjunction** $\vee$

φ	$\vee$	ψ
1	**1**	1
1	**1**	0
0	**1**	1
0	**0**	0

Disjunction behaves like the 2-place functions ***(x_1+x_2)-(x_1 · x_2)=y*** and ***max(x_1, x_2)=y.***

George Boole's **Algebra of Logic** (mid 19th century)

- Statements have value 0 or 1
- "**and**" is understood as *multiplication*
- "**not**" is understood as *subtraction*
- "**or**" is understood as *Boolean addition* (define Boolean addition as 1+1=1; 1+0=1; 0+1=1; and 0=0+0)

Evaluating One Formula Relative to One Valuation

	$($ $\neg$	$p)$	$\wedge$	q
V:	1	0	1	1

$$V \models (\neg p) \wedge q$$

The order matters:

First, assign a truth value to **p** and **q**; *then* to **(¬p)**; and *finally* to **(¬p) ∧ q**.

Go from the simplest to the more complex.

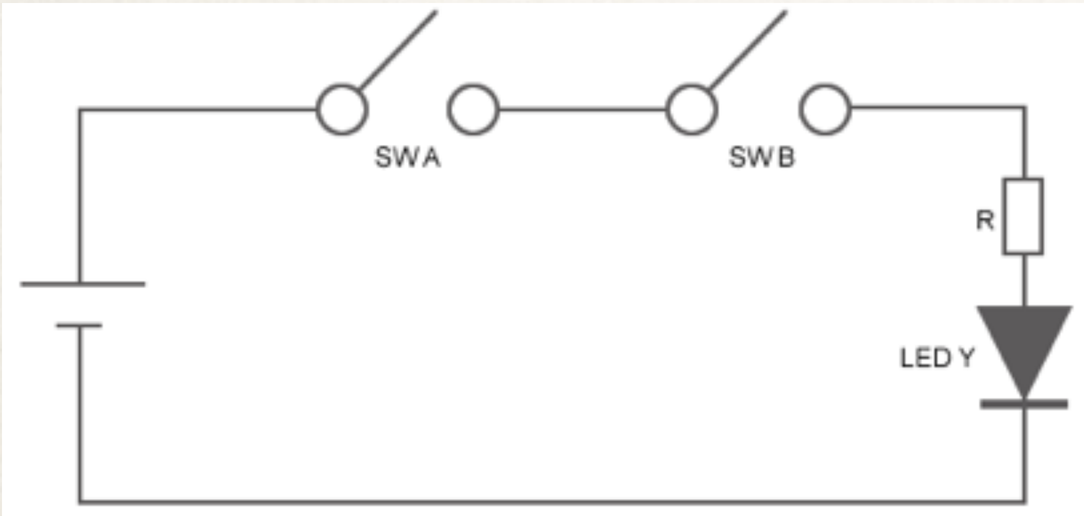
The expression

$V \models (\neg p) \wedge q$

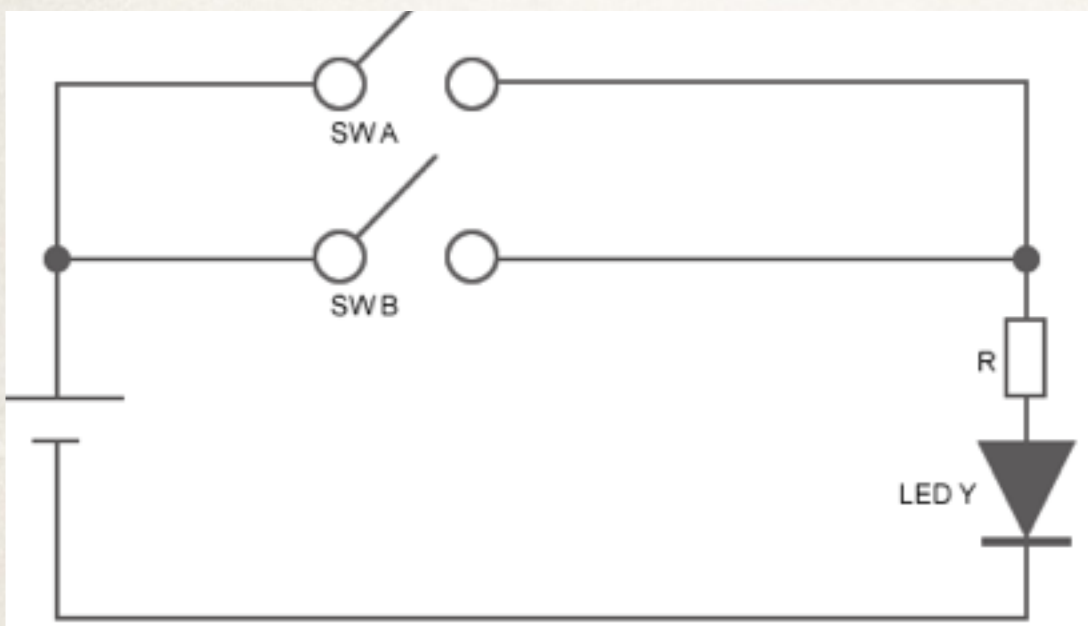
should be understood as saying that ***V* makes true** the formula **(¬p) ∧ q**

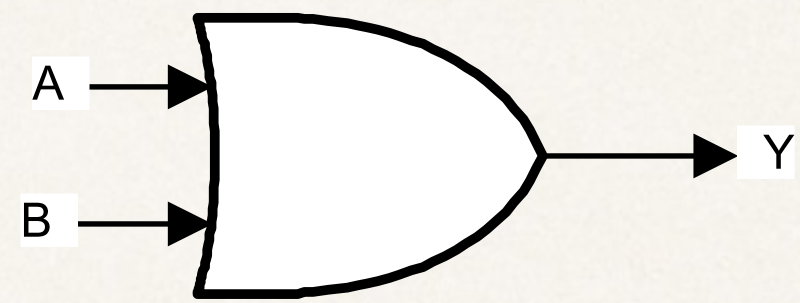
Importantly, $V \models (\neg p) \wedge q$ is ***not*** a formula.

Logic Gates

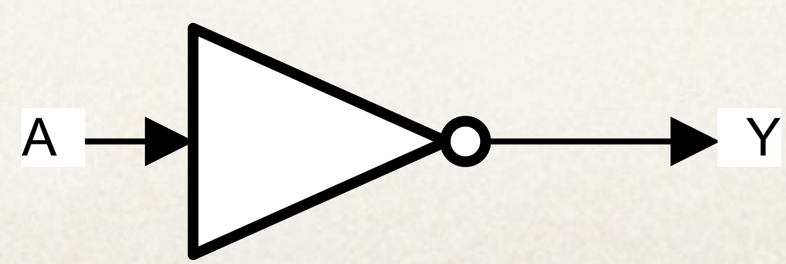


AND gate



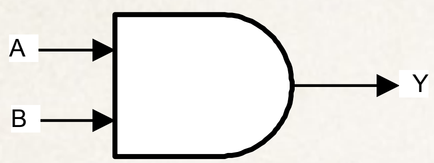


OR gate

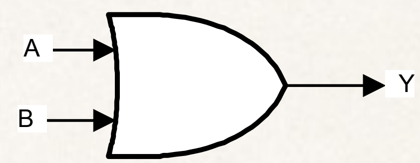


NOT gate

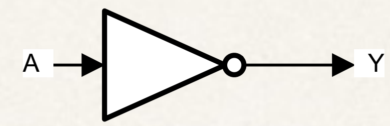
Logic Circuits and Formulas



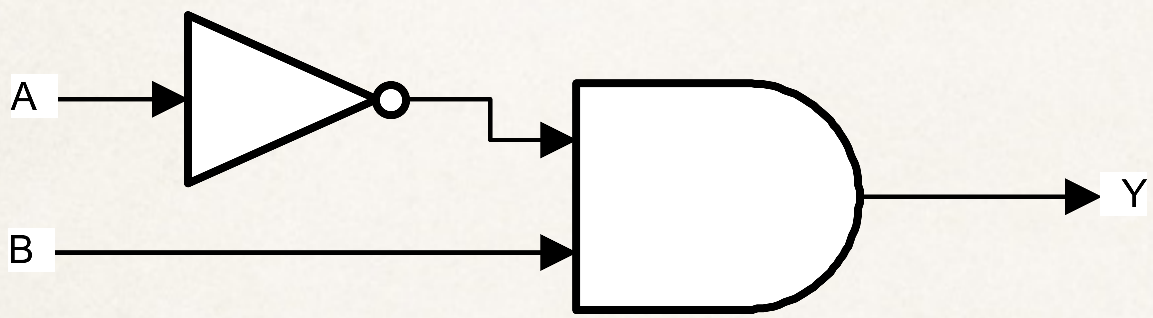
AND



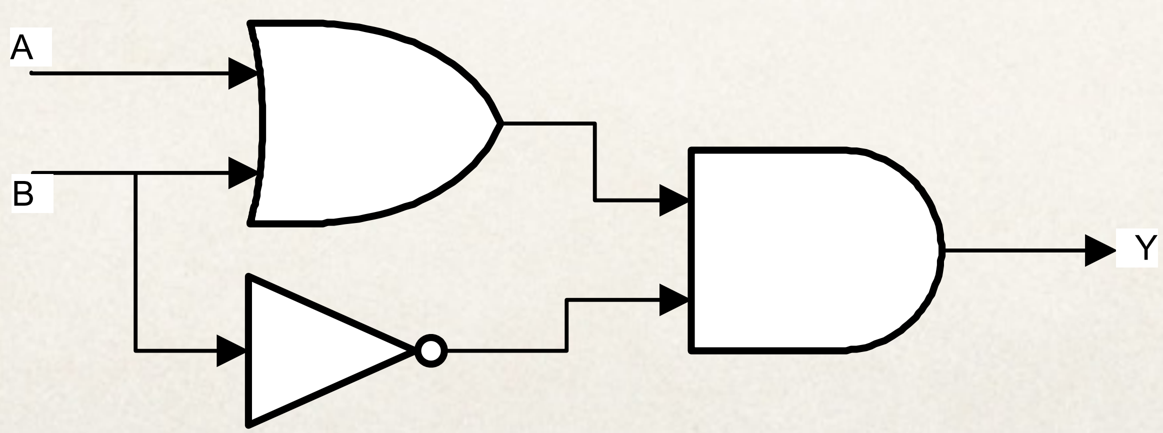
OR



NOT

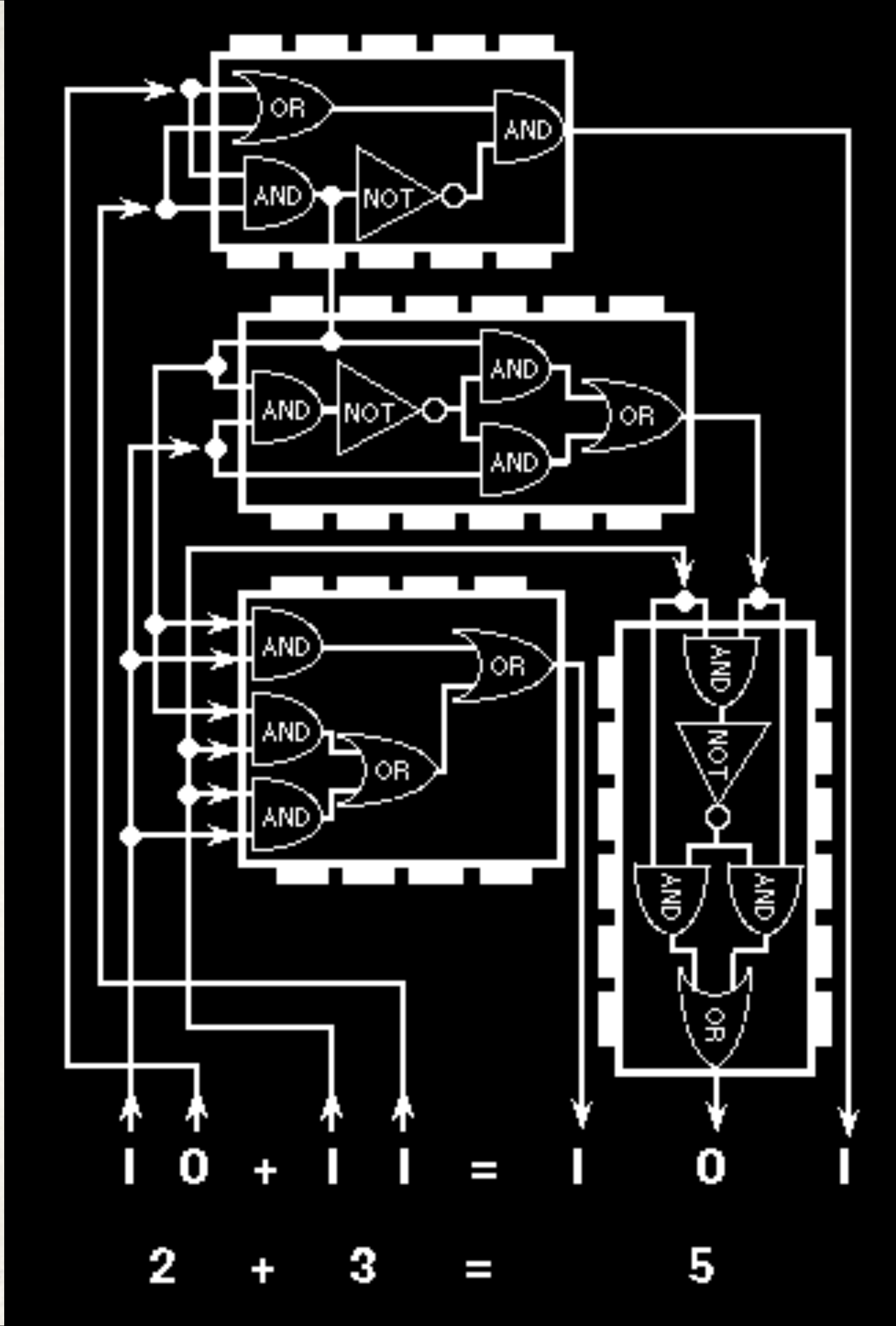


(¬A) ∧ B



(A ∨ B) ∧ (¬B)

Adding 2+3


OR
AND
AND
NOT
AND
AND
NOT
OR
AND
AND
OR
AND
OR
AND
AND
NOT
AND
AND
OR
1 0 + 1 1 = 1 0 1
2 + 3 = 5

Two Standpoints: Language and Circuits

You can regard formulas as **statements capable of being true or false**, e.g. statements about how things are, who is guilty or innocent, etc.

You can also regard formulas as representing **circuits with inputs and outputs.** The inputs are the values (0 or 1) of the atomic formulas and the output is the value (0 or 1) of the complex formula.

Conjunction, Disjunction, and Negation…What About Implication?

For
next
class….