CNS 188a
Computation Theory and Neural Systems

Monday and Wednesday 1:30-3:00
Moore 080

Lecturer: Shuki Bruck; 331 Moore
office hours: Mon, Wed, 3-4pm

TAs: Vincent Bohossian, Matt Cook; 311 Moore
office hours: Mon, Tue, 8-9pm

Secretary: Michelle Chen; 304 Moore
From Screws to Systems...
C. Elegans Lineage

A FAMILY TREE OF EVERY CELL
IN THE WORM

Scientists have learned where each of the 959 cells that make up an adult C. elegans worm
form, tracing it back to a single fertilized egg. As shown in the lineage map, the egg divides
twice, each daughter cell continues to divide. Each horizontal line represents one round of cell
division. The length of each vertical line represents the time between cell divisions, and the
end of each vertical line represents one fully
differentiated cell.

Some of these differentiated cells are "born"
after only 4 rounds of cell division—for example,
none of the cells that generate the muscle, the
toes or the tail. Other cell lines require as many
as 21 rounds. The cells that make up the worm's
pharynx, or feeding organs, are born after the 11
rounds of division. Cells in the hered require up
to 27 divisions.

Almost 600 nerve cells are destined for the
worm's nervous system. Exactly 131 cells are
programmed to die, mostly within minutes of
their birth. The fate of each cell is the same in
every C. elegans worm, except for the cells
maternal become egg and sperm. The major
organs of the worm are color-coded to match the
colors of the corresponding groups of cells on
the lineage map.

**total of 959 cells**
**302 nerve cells**
**131 cells are
destined to die**
C. Elegans Lineage – Simple Questions

Dealing with identity: How do cells remember what to do?

Dealing with time: How do cells know when? No clock...

Dealing with order: How do cells coordinate their actions?

Total of 959 cells
302 nerve cells
131 cells are destined to die
Control via
Stochastic Chemical Reactions

\[ A + B \xrightarrow{k_1} C \]
\[ B + C \xrightarrow{k_2} D \]
\[ D + E \xrightarrow{k_3} F \]
\[ F \xrightarrow{k_4} D + G \]
\[ E + G \xrightarrow{k_5} A \]
Chemical Reactions Circuits

\[ A + B \xrightarrow{k_1} C \]
\[ B + C \xrightarrow{k_2} D \]
\[ D + E \xrightarrow{k_3} F \]
\[ F \xrightarrow{k_4} D + G \]
\[ E + G \xrightarrow{k_5} A \]
Chemical Reactions Circuits

\[ \begin{align*}
A + B & \xrightarrow{k_1} C \\
B + C & \xrightarrow{k_2} D \\
D + E & \xrightarrow{k_3} F \\
F & \xrightarrow{k_4} D + G \\
E + G & \xrightarrow{k_5} A
\end{align*} \]
Chemical Reactions Circuits

\[ A + B \xrightarrow{k_1} C \]
\[ B + C \xrightarrow{k_2} D \]
\[ D + E \xrightarrow{k_3} F \]
\[ F \xrightarrow{k_4} D + G \]
\[ E + G \xrightarrow{k_5} A \]
Chemical Reactions Circuits

\[ A + B \xrightarrow{k_1} C \]
\[ B + C \xrightarrow{k_2} D \]
\[ D + E \xrightarrow{k_3} F \]
\[ F \xrightarrow{k_4} D + G \]
\[ E + G \xrightarrow{k_5} A \]
Chemical Reactions Circuits

\[ A + B \xrightarrow{k_1} C \]

\[ B + C \xrightarrow{k_2} D \]

\[ D + E \xrightarrow{k_3} F \]

\[ F \xrightarrow{k_4} D + G \]

\[ E + G \xrightarrow{k_5} A \]
Chemical Reactions Circuits

\[ A + B \overset{k_1}{\rightarrow} C \]
\[ B + C \overset{k_2}{\rightarrow} D \]
\[ D + E \overset{k_3}{\rightarrow} F \]
\[ F \overset{k_4}{\rightarrow} D + G \]
\[ E + G \overset{k_5}{\rightarrow} A \]
Chemical Reactions Circuits

\[ A + B \xrightleftharpoons[{k_1}]{} C \]
\[ B + C \xrightleftharpoons[{k_2}]{} D \]
\[ D + E \xrightleftharpoons[{k_3}]{} F \]
\[ F \xrightleftharpoons[{k_4}]{} D + G \]
\[ E + G \xrightleftharpoons[{k_5}]{} A \]
Descriptive Biology: Is It Enough?
A HUGE Gap between Ability to Design and Analyze

Design

\[ x \oplus y \to S \]
\[ z \to C \]

Analysis

\[ \text{Diagram of a complex circuit} \]
Key to the Progress in Design: Abstractions in Information Systems

Reasoning to Calculations to Physical Circuits

Logical reasoning

Boolean Calculus

Circuits
Key to the Progress in Design: Abstractions in Information Systems

Logic to **Boolean Calculus** to Physical Circuits

Boole
1815-1864

1847
Connected Logic with Algebra
Boolean Algebra
Logical Calculation

Shannon
1916-2001

1938
Boolean Algebra to Electrical Circuits
Logic Design
the interpretation of \( v \). But it has been thought better to write them separately, for greater ease and convenience. And it is further to be borne in mind, that although three different forms are given for the expression of each of the particular propositions, everything is really included in the first form.

\[
\begin{array}{|l|}
\hline
\text{The class } X & x \\
\text{The class not-}X & 1 - x \\
\text{All } Xs \text{ are } Ys & x = y \\
\text{All } Ys \text{ are } Xs & x = y \\
\text{All } Xs \text{ are } Ys & x(1 - y) = 0 \\
\text{No } Xs \text{ are } Ys & xy = 0 \\
\text{All } Ys \text{ are } Xs \quad y = vx & v(1 - x) = 0 \\
\text{Some } Xs \text{ are } Ys \quad y = vx & v(1 - x) = \text{ some } Xs \\
\text{No } Ys \text{ are } Xs \quad y = v(1 - x) & v = \text{ some not-}Xs \\
\text{Some not-}Xs \text{ are } Ys \quad y = v(1 - x) & v = \text{ some not-}Xs \\
\text{Some } Xs \text{ are } Ys \quad v = xy & v = \text{ some } Xs \text{ or some } Ys \\
\text{or } v = vy & v = \text{ some } Xs, vy = \text{ some } Ys \\
\text{or } v(1 - y) = 0 & v(1 - x) = 0, v(1 - y) = 0, \\
\text{Some } Xs \text{ are not } Ys \quad v = x(1 - y) & v = \text{ some } Xs, \text{ or some not-}Ys \\
\text{or } v = y(1 - y) & v = \text{ some } Xs, v(1 - y) = \text{ some not-}Ys \\
\text{or } v = xy = 0 & v(1 - x) = 0, ey = 0. \\
\hline
\end{array}
\]
The class $X$ $x$

The class not-$X$ $1 - x$

All $X$s are $Y$s \( x = y \)

All $Y$s are $X$s \( x(1-y) = 0 \)

No $X$s are $Y$s
\[ xy = 0 \]

All $Y$s are $X$s \( y = vx \)

Some $X$s are $Y$s
\[ vx = \text{some } Xs \]

\[ v(1-x) = 0 \]

No $Y$s are $X$s

Some not-$X$s are $Y$s
\[ y = v(1-x) \]

\[ v(1-x) = \text{some not-}Xs \]

\[ vx = 0 \]
Starting from the right, $s_0$ is one if $a_0$ is one and $b_0$ is zero or if $a_0$ is zero and $b_0$ one but not otherwise. Hence

$$s_0 = a_0b'_0 + a'_0b_0 = a_0 \oplus b_0.$$ 

$c_1$ is one if both $a_0$ and $b_0$ are one but not otherwise:

$$c_1 = a_0 \cdot b_0.$$ 

$s_j$ is one if just one of $a_j$, $b_j$, $c_j$ is one, or if all three are one:

$$s_j = S_{1,3}(a_j, b_j, c_j), \quad j = 1, 2, \ldots k.$$ 

$c_{j+1}$ is one if two or if three of these variables are one:

$$c_{j+1} = S_{2,3}(a_j, b_j, c_j), \quad j = 1, 2, \ldots k.$$ 

Using the method of symmetric functions, and shifting down for $s_j$ gives the circuits of Figure 35. Eliminating superfluous elements we arrive at Figure 36.
The Algebra (Boolean Calculus)
Boole, Jevons, Peirce, Schroder (18xx)
Axiomatic System: Huntington (1904)

**Algebraic system**: set of elements \( B \),

- two binary operations + and \( \cdot \).

\( B \) has at least two elements (0 and 1)

If the following axioms are true then it is a **Boolean Algebra**:
Algebraic system: set of elements $B$, two binary operations $+$ and $\cdot$. $B$ has at least two elements (0 and 1).

If the following axioms are true then it is a Boolean Algebra:

A1. identity

$$a + 0 = a; \quad a \cdot 1 = a$$

A2. complement

$$a + \overline{a} = 1; \quad a \cdot \overline{a} = 0$$

A3. commutative

$$a + b = b + a; \quad a \cdot b = b \cdot a$$

A4. distributive

$$a + b \cdot c = (a + b) \cdot (a + c); \quad a \cdot (b + c) = a \cdot b + a \cdot c$$
### Two-valued Boolean Algebra

Boolean Algebra: set of elements $B=\{0, 1\}$, two binary operations OR and AND

<table>
<thead>
<tr>
<th>$xy$</th>
<th>OR($x, y$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$xy$</th>
<th>AND($x, y$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>
Two-valued Boolean Algebra

Boolean Algebra: set of elements \( B = \{0, 1\} \), two binary operations OR and AND

The following axioms are obviously true:

\[
\begin{align*}
    a + 0 &= a; & a \cdot 1 &= a \\
    a + \overline{a} &= 1; & a \cdot \overline{a} &= 0 \\
    a + b &= b + a; & a \cdot b &= b \cdot a \\
    a + b \cdot c &= (a + b) \cdot (a + c); & a \cdot (b + c) &= a \cdot b + a \cdot c
\end{align*}
\]
Two-valued Boolean Algebra

Boolean Algebra: set of elements $B=\{0,1\}$, two binary operations OR and AND

A1. identity

\[
\begin{align*}
a + 0 &= a; & a \cdot 1 &= a \\
0 + 0 &= 0; & 0 \cdot 1 &= 0 \\
1 + 0 &= 1; & 1 \cdot 1 &= 1
\end{align*}
\]

<table>
<thead>
<tr>
<th>xy</th>
<th>OR(x,y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>xy</th>
<th>AND(x,y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>
Two-valued Boolean Algebra

Boolean Algebra: set of elements \( B = \{0, 1\} \), two binary operations OR and AND

A2. complement

\[
\begin{align*}
a + \bar{a} &= 1; \quad a \cdot \bar{a} = 0 \\
0 + 1 &= 1 \\
1 + 0 &= 1 \\
0 \cdot 1 &= 0 \\
1 \cdot 0 &= 0
\end{align*}
\]

<table>
<thead>
<tr>
<th>( xy )</th>
<th>( \text{OR}(x,y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( xy )</th>
<th>( \text{AND}(x,y) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
</tr>
</tbody>
</table>
Two-valued Boolean Algebra

Boolean Algebra: set of elements $B=\{0,1\}$, two binary operations OR and AND

A3. commutative

\[
\begin{align*}
0+0 &= 0+0 & 0 \cdot 0 &= 0 \cdot 0 \\
0+1 &= 1+0 & 0 \cdot 1 &= 1 \cdot 0 \\
1+0 &= 0+1 & 1 \cdot 0 &= 0 \cdot 1 \\
1+1 &= 1+1 & 1 \cdot 1 &= 1 \cdot 1
\end{align*}
\]
Two-valued Boolean Algebra

Boolean Algebra: set of elements $B=\{0,1\}$, two binary operations OR and AND

A4. distributive

$$a + b \cdot c = (a + b) \cdot (a + c); \quad a \cdot (b + c) = a \cdot b + a \cdot c$$
Two-valued Boolean Algebra

**Boolean Algebra**: set of elements \(B=\{0,1\}\), two binary operations OR and AND

**A4. distributive**

\[a + b \cdot c = (a + b) \cdot (a + c); \quad a \cdot (b + c) = a \cdot b + a \cdot c\]

\[
\begin{align*}
0+0 \cdot 0 &= (0+0) \cdot (0+0) = 0 \\
0+0 \cdot 1 &= (0+0) \cdot (0+1) = 0 \\
0+1 \cdot 0 &= (0+1) \cdot (0+0) = 0 \\
0+1 \cdot 1 &= (0+1) \cdot (0+1) = 1 \\
1+0 \cdot 0 &= (1+0) \cdot (1+0) = 1 \\
1+0 \cdot 1 &= (1+0) \cdot (1+1) = 1 \\
1+1 \cdot 0 &= (1+1) \cdot (1+0) = 1 \\
1+1 \cdot 1 &= (1+1) \cdot (1+1) = 1
\end{align*}
\]
Historical notes: Boolean Algebra Defined by Axioms (Postulates)

Pre-Boole (16xx): Leibniz; universal language for reasoning?

Beginning (18xx): Boole, Jevons, Peirce, Schroder, Whitehead

Next step: Huntington 1904; improved set of axioms

Sheffer 1913: Five axioms and one binary operation

Huntington/Robbins 1933 - three axioms, conjecture

Recent progress: McCune 1996; Robbins conjecture proved!
Historical notes: Three Axioms

- **A1. Commutativity:** \( a + b = b + a \)
- **A2. Associativity:** \( (a + b) + c = a + (b + c) \)
- **A3a. Huntington:** \( a = (\overline{a} + b) + (\overline{a} + \overline{b}) \)
- **A3b. Robbins:** \( a = (a + b) + (a + \overline{b}) \)
Back to the Axioms

- **A1. Identities:**
  \[ a + 0 = a \quad \text{and} \quad a \cdot 1 = a \]

- **A2. Complements:**
  \[ a + \overline{a} = 1 \quad \text{and} \quad a \cdot \overline{a} = 0 \]
  
  **Q1:** is the complement unique / well defined?

- **A3. Commutativity:**
  \[ a + b = b + a \quad \text{and} \quad a \cdot b = b \cdot a \]

- **A4. Distributivity:**
  \[ a + b \cdot c = (a + b) \cdot (a + c) \quad \text{and} \quad a \cdot (b + c) = a \cdot b + a \cdot c \]
One Way to Say No!

Theorem 1:
Each element of a Boolean Algebra has exactly one complement.

Proof:
First we will prove that an element is not self-complement

- A1. Identities:
  \[ a + 0 = a \quad \text{and} \quad a \cdot 1 = a \]

- A2. Complements:
  \[ a + \bar{a} = 1 \quad \text{and} \quad a \cdot \bar{a} = 0 \]

- A3. Commutativity:
  \[ a + b = b + a \quad \text{and} \quad a \cdot b = b \cdot a \]

- A4. Distributivity:
  \[ a + b \cdot c = (a + b) \cdot (a + c) \quad \text{and} \quad a \cdot (b + c) = a \cdot b + a \cdot c \]
Self Absorption

Lemma 1: \[ a + a = a \]

Proof:
\[
\begin{align*}
  a + a &= (a + a) \cdot 1 \\
        &= (a + a)(a + \bar{a}) \\
        &= a + a \cdot \bar{a} \\
        &= a + 0 \\
        &= a
\end{align*}
\]

\begin{itemize}
  \item A1. Identities: \( a + 0 = a \) and \( a \cdot 1 = a \)
  \item A2. Complements: \( a + \bar{a} = 1 \) and \( a \cdot \bar{a} = 0 \)
  \item A3. Commutativity: \( a + b = b + a \) and \( a \cdot b = b \cdot a \)
  \item A4. Distributivity: \( a + b \cdot c = (a + b) \cdot (a + c) \) and \( a \cdot (b + c) = a \cdot b + a \cdot c \)
\end{itemize}
SpinOut – Keister

William Keister (1907-1997) was a pioneer in switching theory and design at Bell Labs. When he retired in 1972, he was director of Bell Labs' Computing Technology Center at Holmdel, New Jersey.

Mr. Keister began working in his spare time to prove that puzzles could be solved using Boolean algebra.

U.S. Patent 3637216 (1972): The Hexadecimal Puzzle
Theorem 1:
Each element of a Boolean Algebra has exactly one complement.

Proof:
First we will prove that an element is not self-complement

Assume that: \( a = \bar{a} \)

By Lemma 1:
\[
\begin{align*}
    a + a &= a \\
    a + \bar{a} &= a
\end{align*}
\]

However by A2:
\[
\begin{align*}
    a + \bar{a} &= 1 \\
    a &= 1
\end{align*}
\]

Contradiction!

By A1: \( 1 \cdot 1 = 1 \)

By A2: \( a \cdot \bar{a} = 0 \)
One Way to Say No!

Theorem 1:
Each element of a Boolean Algebra has exactly one complement.

Proof: We proved that an element is not self-complement.

Next will prove that the complement is unique.
One Way to Say No!

Proof:

Need to prove that the complement is **unique**

By contradiction: Assume an element has two **distinct** complements

\[
\begin{align*}
  a + \bar{a} &= 1 \\
  a + x &= 1 \\
  a \cdot \bar{a} &= 0 \\
  a \cdot x &= 0
\end{align*}
\]

\[
\begin{align*}
  x &= x \cdot 1 \\
  &= x \cdot (a + \bar{a}) \\
  &= x \cdot a + x \cdot \bar{a} \\
  &= \bar{a} \cdot a + \bar{a} \cdot x \\
  &= a \cdot x + \bar{a} \cdot x \\
  &= 0 + \bar{a} \cdot x
\end{align*}
\]

Contradiction!
Back to the Axioms

- **A1. Identities:**
  \[ a + 0 = a \quad \text{and} \quad a \cdot 1 = a \]

- **A2. Complements:**
  \[ a + \bar{a} = 1 \quad \text{and} \quad a \cdot \bar{a} = 0 \]

- **A3. Commutativity:**
  \[ a + b = b + a \quad \text{and} \quad a \cdot b = b \cdot a \]

- **A4. Distributivity:**
  \[ a + b \cdot c = (a + b) \cdot (a + c) \quad \text{and} \quad a \cdot (b + c) = a \cdot b + a \cdot c \]
Duality

Theorem 0:
Any identity that is true in a Boolean algebra, is also true if + and \cdot are interchanged, and 0 and 1 are interchanged.

Proof:
It is true for the axioms!

- A1. Identities:
  \[ a + 0 = a \quad \text{and} \quad a \cdot 1 = a \]
- A2. Complements:
  \[ a + \bar{a} = 1 \quad \text{and} \quad a \cdot \bar{a} = 0 \]
- A3. Commutativity:
  \[ a + b = b + a \quad \text{and} \quad a \cdot b = b \cdot a \]
- A4. Distributivity:
  \[ a + b \cdot c = (a + b) \cdot (a + c) \quad \text{and} \quad a \cdot (b + c) = a \cdot b + a \cdot c \]
Lemma 1: \[ a + a = a \quad a \cdot a = a \]

Proof:
\[
\begin{align*}
a + a &= (a + a) \cdot 1 \\
&= (a + a)(a + \bar{a}) \\
&= a + a \cdot \bar{a} \\
&= a + 0 \\
&= a
\end{align*}
\]

- A1. Identities:
  \[ a + 0 = a \quad \text{and} \quad a \cdot 1 = a \]

- A2. Complements:
  \[ a + \bar{a} = 1 \quad \text{and} \quad a \cdot \bar{a} = 0 \]

- A3. Commutativity:
  \[ a + b = b + a \quad \text{and} \quad a \cdot b = b \cdot a \]

- A4. Distributivity:
  \[ a + b \cdot c = (a + b) \cdot (a + c) \quad \text{and} \quad a \cdot (b + c) = a \cdot b + a \cdot c \]
So far... True for any Boolean Algebra

- **A1. Identities:**
  \[ a + 0 = a \quad \text{and} \quad a \cdot 1 = a \]

- **A2. Complements:**
  \[ a + \bar{a} = 1 \quad \text{and} \quad a \cdot \bar{a} = 0 \]

- **A3. Commutativity:**
  \[ a + b = b + a \quad \text{and} \quad a \cdot b = b \cdot a \]

- **A4. Distributivity:**
  \[ a + b \cdot c = (a + b) \cdot (a + c) \quad \text{and} \quad a \cdot (b + c) = a \cdot b + a \cdot c \]
Non-Binary Boolean Algebras

**Boolean Integers** (Bunitskiy 1899)
\[ 2 \times 3 \times 5 = 30 \]
\[ 2 \times 3 \times 7 = 42 \]

Every prime in the prime factorization is a power of one

The set of divisors of a Boolean integer
\{1,2,3,5,6,10,15,30\}

The operations: \text{lcm} and \text{gcd}

The special elements: 1 and 30
Non-Binary Boolean Algebras

The set \{1, 2, 3, 5, 6, 10, 15, 30\}
The operations: \text{lcm} and \text{gcd}
The special elements: 1 and 30

6 + 15 = \text{lcm}(6, 15) = ?

6 \cdot 15 = \text{gcd}(6, 15) = ?
Non-Binary Boolean Algebras

The set \(\{1,2,3,5,6,10,15,30\}\)

The operations: \(\text{lcm}\) and \(\text{gcd}\)

The special elements: 1 and 30

\[6 + 15 = \text{lcm}(6,15) = 30\]

\[6 \cdot 15 = \text{gcd}(6,15) = 3\]
What is the Complement?

- **A1. Identities:**
  \[ a + 0 = a \quad \text{and} \quad a \cdot 1 = a \]

- **A2. Complements:**
  \[ a + \bar{a} = 1 \quad \text{and} \quad a \cdot \bar{a} = 0 \]

- **A3. Commutativity:**
  \[ a + b = b + a \quad \text{and} \quad a \cdot b = b \cdot a \]

- **A4. Distributivity:**
  \[ a + b \cdot c = (a + b) \cdot (a + c) \quad \text{and} \quad a \cdot (b + c) = a \cdot b + a \cdot c \]

The set \{1,2,3,5,6,10,15,30\}
The operations: \text{lcm} and \text{gcd}
The special elements: 1 and 30

\[ a + \bar{a} = lcm(a, \bar{a}) = 30 \]

\[ a \cdot \bar{a} = gcd(a, \bar{a}) = 1 \]
CNS 188a Overview

• **Boolean algebra** as an axiomatic system

• **Boolean functions** and their **representations** using Boolean formulas and spectral methods

• **Implementing** Boolean functions with circuits of AON (AND, OR, NOT) gates and LT (Linear Threshold) gates

• Analyzing the **complexity** (size and depth) of circuits

• **Relations** (as opposed to functions) and their implementation in circuits

• **Feedback** and convergence in LT circuits