Discrete Structures

- Sets
- Functions
- Automata
- Regular expressions
Formal definition
Relations between structures (e.g., =, ⊆, etc.)
Prove properties about structure
Logic! [Meta, I know]
Why study logic? (w/ Amazing clip-art)

I'm super excited to see the new Avengers movie tomorrow!

ME

Anonymous friend
Why study logic? (w/ Amazing clip-art)

Oh I saw that!
AQUAMAN DROWNS

ME

Anonymous friend
Why study logic? (w/ Amazing clip-art)
BUT WAIT! Aquaman isn't in the Marvel Universe and he can breathe under water. Therefore, he could not have drowned in the Avengers movie.
Why study logic? (w/ Amazing clip-art)
Our base is an **base proposition**

**Definition.**
An **base proposition** is a primitive statement that can be true or false. For example, “It is raining in Ithaca” is either true (most likely true) or false.

We represent propositions as variables such as $P, Q \in Prop$
Inductive definition

- **Inductive** definition! So we need
  - A base case
  - An inductive case(s)
Inductive definition

Connectives: $\land, \lor, \rightarrow, \neg$

Connectives: and, or, implies, not

Ex., it is raining in Ithaca and it is sunny in LA.
Inductive definition

(it is raining in Ithaca and sunny in LA)

or

(it is sunny in Ithaca and raining in LA).
**Definition**

A *formula* is one or more propositions connected by a connective.

More formally, \( \text{Formulae} ::= P | \phi \land \psi | \phi \lor \psi | \phi \rightarrow \psi | \neg \phi \) where \( P \in \text{Prop} \)
Inductive definition

Formulae ::= \( P \mid \phi \land \psi \mid \phi \lor \psi \mid \phi \rightarrow \psi \mid \neg \phi \), \( P \in \text{Prop} \)

(it is raining in Ithaca and sunny in LA) or (it is sunny in Ithaca and raining in LA).

\[(P \land Q) \lor (\neg P \land \neg Q)\]
Inductive definition

Formulae ::= $P|\phi \land \psi|\phi \lor \psi|\phi \rightarrow \psi|\neg\phi$, $P \in \text{Prop}$

(it is raining in Ithaca and sunny in LA) or (it is sunny in Ithaca and raining in LA).

Look familiar?
Inductive definition

**Formulae** ::= $P | \phi \land \psi | \phi \lor \psi | \phi \rightarrow \psi | \neg \phi$, $P \in \text{Prop}$

**Exp** ::= $\emptyset | \epsilon | a | r_1 r_2 | (r_1 + r_2) | r^*$, $a \in \Sigma$

**Math Exp** ::= $n | e_1 + e_2 | e_1 \times e_2 | - e | e_1 \div e_2$, $n \in \mathbb{N}$

**Str** ::= $\epsilon | xa$
Problem: How do we know if it’s raining in Ithaca? We need a mapping of propositions to truth values!
Definition.

An **interpretation** is any function \( I : \text{Prop} \to \{ T, F \} \)

For example:

\[
I_1(\text{“It is raining in Ithaca”}) = T \quad I_2(\text{“It is raining in Ithaca”}) = T
\]

\[
I_1(\text{“It is sunny in LA”}) = F \quad I_2(\text{“It is sunny in LA”}) = F
\]
$I : \text{Prop} \rightarrow \{T, F\}$

$I(\text{“It is raining in Ithaca” and “It is sunny in LA”}) = ???$

Problems?
Truth

Formulae ::= $P \mid \phi \land \psi \mid \phi \lor \psi \mid \phi \rightarrow \psi \mid \neg \phi$

eval : Interpretation × Formulae → \{T, F\}

eval(I, P) ::= I(P) \leftarrow base

eval(I, \phi \land \psi) ::= \begin{cases} T & \text{if } eval(I, \phi) = T \text{ and } eval(I, \psi) = T \\ F & \text{otherwise} \end{cases}
Truth

Formulæ ::= \( P \mid \phi \land \psi \mid \phi \lor \psi \mid \phi \rightarrow \psi \mid \neg \phi \)

eval : Interpretation \times \text{Formulæ} \rightarrow \{T, F\}

eval(I, P) ::= I(P)

\[
eval(I, \phi \land \psi) ::= \begin{cases} 
  T & \text{if eval}(\phi, I) = T \text{ and eval}(\psi, I) = T \\
  F & \text{otherwise}
\end{cases}
\]

\[
eval(I, \phi \lor \psi) ::= \begin{cases} 
  T & \text{if eval}(\phi, I) = T \text{ or eval}(\psi, I) = T \\
  F & \text{otherwise}
\end{cases}
\]

\[
eval(I, \phi \rightarrow \psi) ::= \begin{cases} 
  F & \text{if eval}(\phi, I) = T \text{ and eval}(\psi, I) = F \\
  T & \text{otherwise}
\end{cases}
\]

\[
eval(I, \neg \phi) ::= \begin{cases} 
  T & \text{if eval}(\phi, I) = F \\
  F & \text{otherwise}
\end{cases}
\]
The truth value of a formula is dependent on the truth value of its substructures

<table>
<thead>
<tr>
<th>eval($\phi, I$)</th>
<th>eval($\psi, I$)</th>
<th>eval($\phi \land \psi, I$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
</tbody>
</table>
### Truth

<table>
<thead>
<tr>
<th>$eval(\phi, I)$</th>
<th>$eval(\psi, I)$</th>
<th>$P$</th>
<th>$\phi \land \psi$</th>
<th>$\phi \lor \psi$</th>
<th>$\phi \rightarrow \psi$</th>
<th>$\neg \phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$I(P)$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$I(P)$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$I(P)$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$I(P)$</td>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
</tbody>
</table>
Truth

\[ I : \text{Prop} \rightarrow \{ T, F \} \]

\[ \text{eval} : \text{Formulae} \times \text{Interpretation} \rightarrow \{ T, F \} \]

Look familiar?
Truth

\[ I : \text{Prop} \rightarrow \{ T, F \} \]

\[ \text{eval} : \text{Interpretation} \times \text{Formulae} \rightarrow \{ T, F \} \]

\[
\text{eval}(I, P) ::= I(P)
\]

\[
\text{eval}(I, \phi \land \psi) ::= \begin{cases} T & \text{if eval}(\phi, I) = T \text{ and eval}(\psi, I) = T \\ F & \text{otherwise} \end{cases}
\]

\[
\text{eval}(I, \phi \lor \psi) ::= \begin{cases} T & \text{if eval}(\phi, I) = T \text{ or eval}(\psi, I) = T \\ F & \text{otherwise} \end{cases}
\]

\[
\text{eval}(I, \phi \rightarrow \psi) ::= \begin{cases} F & \text{if eval}(\phi, I) = T \text{ and eval}(\psi, I) = F \\ T & \text{otherwise} \end{cases}
\]

\[
\text{eval}(I, \neg \phi) ::= \begin{cases} T & \text{if eval}(\phi, I) = F \\ F & \text{otherwise} \end{cases}
\]

\[
\delta : Q \times \Sigma \rightarrow Q
\]

\[
\widehat{\delta}(q, \epsilon) = q
\]

\[
\widehat{\delta}(q, xa) = \delta(\widehat{\delta}(q, x), a)
\]
Two formulae $\phi$, $\psi$ are equivalent (i.e., $\phi \equiv \psi$) if they evaluate to the same truth values.

<table>
<thead>
<tr>
<th>$I(P)$</th>
<th>$I(Q)$</th>
<th>$P \rightarrow Q$</th>
<th>$\neg P \lor Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
</tbody>
</table>
Equality

But what does equality really tell us? “It is raining in Ithaca = It is bad weather in Ithaca”
Equality

ψ \implies \phi

“Because it is raining in Ithaca, we know that it is bad weather.”
We want to know if ψ entails φ.

\[ \text{LHS} \leq \text{RHS} \uparrow \text{must be true} \]
Definition.

If $I$ is an interpretation and $\phi$ a formula, we say $I$ satisfies $\phi$ if $I(\phi) = T$. We write $I \models \phi$.

Example.

“Leah loves 2800” is true in this world where $I_1(\text{“Leah loves 2800”}) = T$ therefore $I_1 \models \text{“Leah loves 2800”}$
Validity

Definition.

If $I \vDash \phi$ for all $I$, we say $\phi$ is valid or a tautology.
Entailment

Definition: If $l \vDash \phi$ for all $l$ that satisfy $\psi$, we say that $\psi$ entails $\phi$, written $\psi \models \phi$.

Entailment $\neq$ Implication!

$(\psi \models \phi) \neq (\psi \rightarrow \phi)$

\[\forall I. \text{ if } I \vDash \psi \text{ then } I \vDash \phi\]

if $\psi$ is true, and $\phi$ is true, then we can derive $\phi$ from $\psi$. 

(Minor correction: "derive $\phi$ from $\psi" \rightarrow "\phi" from $\psi""]
You have now crossed over into …

THE
LOGIC
ZONE
Entailment vs Implication

- Formulae are just strings of symbols!

\[ I \models \phi \rightarrow \psi \]

- Entailment, satisfaction and validity give us tools to relate formulae.

- SPOILER: These relations between different formulae allow us to build proofs.

"2abbc" is a string.

\[ \delta(b_0, x) \notin A \text{ is not a string} \]