L22: Inductive Definitions and Structural Induction
We define our data structures – or other objects of interest – inductively.

This is useful because we can:

- Prove things about these objects inductively.
- Define operations on these objects inductively, i.e., recursively.
- Examples: lists, trees, XML.
- Modern programming languages allow recursive function definitions on recursively defined datatypes (Python – hw4).
- Main examples today: logical formulas, truth
We define our data structures – or other objects of interest – inductively.

This is useful because we can:

- Prove things about these objects inductively.
We define our data structures – or other objects of interest – inductively.

This is useful because we can:

- Prove things about these objects inductively.
- Define operations on these objects inductively, i.e., \textit{recursively}. 
We define our data structures – or other objects of interest – inductively.

This is useful because we can:

- Prove things about these objects inductively.
- Define operations on these objects inductively, i.e., recursively.
- Examples: lists, trees, xml
We define our data structures – or other objects of interest – inductively.

This is useful because we can:

- Prove things about these objects inductively.
- Define operations on these objects inductively, i.e., recursively.
- Examples: lists, trees, xml
- Modern programming languages allow recursive function definitions on recursively defined datatypes (Python – hw4)
We define our data structures – or other objects of interest – inductively.

This is useful because we can:

- Prove things about these objects inductively.
- Define operations on these objects inductively, i.e., recursively.
- Examples: lists, trees, xml
- Modern programming languages allow recursive function definitions on recursively defined datatypes (Python – hw4)
- Main examples today: logical formulas, truth
**Def:** Let $\Sigma$ be a PredCalc vocabulary. A **term** $t \in \text{term}(\Sigma)$ is a string of symbols that every world $W \in \text{World}[\Sigma]$ must interpret as an element $t^W \in |W|$. Terms are defined recursively as follows:

**base 0.**  
$v \in \text{VAR} \quad \rightarrow \quad v \in \text{term}(\Sigma)$

variables are terms
**Def:** Let $\Sigma$ be a PredCalc vocabulary. A **term** $t \in \text{term}(\Sigma)$ is a string of symbols that every world $W \in \text{World}[\Sigma]$ must interpret as an element $t^W \in |W|$. Terms are defined recursively as follows:

**base 0.** \hspace{1cm} $v \in \text{VAR} \hspace{3cm} \rightarrow \hspace{1cm} v \in \text{term}(\Sigma)$

variables are terms

**base 1.** \hspace{1cm} $k \in \Sigma \hspace{3cm} \rightarrow \hspace{1cm} k \in \text{term}(\Sigma)$

constant symbols are terms
Terms in PredCalc

**Def:** Let $\Sigma$ be a PredCalc vocabulary. A **term** $t \in \text{term}(\Sigma)$ is a string of symbols that every world $W \in \text{World}[\Sigma]$ must interpret as an element $t^W \in |W|$. Terms are defined recursively as follows:

**base 0.** $\nu \in \text{VAR}$  
$\rightarrow \nu \in \text{term}(\Sigma)$  
variables are terms

**base 1.** $k \in \Sigma$  
$\rightarrow k \in \text{term}(\Sigma)$  
constant symbols are terms

**ind. 2.** $t_1, \ldots, t_r \in \text{term}(\Sigma), f^r \in \Sigma$  
$\rightarrow f(t_1, \ldots, t_r) \in \text{term}(\Sigma)$  
terms are closed under function symbols
**Def:** Let $\Sigma$ be a PredCalc vocabulary. A term $t \in \text{term}(\Sigma)$ is a string of symbols that every world $W \in \text{World}[\Sigma]$ must interpret as an element $t^W \in |W|$. Terms are defined recursively as follows:

**base 0.** $v \in \text{VAR}$ $\rightarrow$ $v \in \text{term}(\Sigma)$  
variables are terms

**base 1.** $k \in \Sigma$ $\rightarrow$ $k \in \text{term}(\Sigma)$  
constant symbols are terms

**ind. 2.** $t_1, \ldots, t_r \in \text{term}(\Sigma), f^r \in \Sigma$ $\rightarrow$ $f(t_1, \ldots, t_r) \in \text{term}(\Sigma)$  
terms are closed under function symbols

\[ \text{term}(\Sigma_{\text{garst}}) = \text{VAR} \cup \{s, t\} \]
**Def:** Let $\Sigma$ be a PredCalc vocabulary. A **term** $t \in \text{term}(\Sigma)$ is a string of symbols that every world $W \in \text{World}[\Sigma]$ must interpret as an element $t^W \in |W|$. Terms are defined recursively as follows:

**base 0.** \( \forall \in \text{VAR} \quad \rightarrow \quad \forall \in \text{term}(\Sigma) \)
variables are terms

**base 1.** \( k \in \Sigma \quad \rightarrow \quad k \in \text{term}(\Sigma) \)
constant symbols are terms

**ind. 2.** \( t_1, \ldots, t_r \in \text{term}(\Sigma), f^r \in \Sigma \quad \rightarrow \quad f(t_1, \ldots, t_r) \in \text{term}(\Sigma) \)
terms are closed under function symbols

\[
\text{term}(\Sigma_{\text{garst}}) = \text{VAR} \cup \{s, t\} = \{s, t, x, y, z, u, v, w, x_1, \ldots\}
\]

\[
\text{term}(\Sigma_{\text{#thy}}) = \{0, 1, x, \ldots, \ldots, x \cdot y, \ldots (x + 1) \cdot (y + 0), \ldots\}
\]
**Default Interpretation of variables:** Unless explicitly stated otherwise, \( v^W = 0 \), (or the min value in \(|W|\) if \(0 \notin |W|\)).
Default Interpretation of variables: Unless explicitly stated otherwise, $v^W = 0$, (or the min value in $|W|$ if $0 \not\in |W|$).

Notation: $W[e/v]$ is same as $W$, except $v^{W[e/v]} = e$. 
Default Interpretation of variables: Unless explicitly stated otherwise, \( v^W = 0 \), (or the min value in \(|W|\) if \(0 \not\in |W|\)).

Notation: \( W[e/v] \) is same as \( W \), except \( v^{W[e/v]} = e \).

\( x^{G_1} = 0 \)
Default Interpretation of variables: Unless explicitly stated otherwise, \( v^W = 0 \), (or the min value in \( |W| \) if \( 0 \not\in |W| \)).

Notation: \( W[e/v] \) is same as \( W \), except \( v^{W[e/v]} = e \).

\( x^{G_1} = 0 \quad y^{G_1} = 0 \)
**Default Interpretation of variables:** Unless explicitly stated otherwise, $\nu^W = 0$, (or the min value in $|W|$ if $0 \not\in |W|$).

**Notation:** $W[e/v]$ is same as $W$, except $\nu^{W[e/v]} = e$.

$x^{G_1} = 0 \quad y^{G_1} = 0 \quad x^{G_1[1/x \ 4/y]} = 1$
Default Interpretation of variables: Unless explicitly stated otherwise, \( v^W = 0 \), (or the min value in \( |W| \) if \( 0 \not\in |W| \)).

Notation: \( W[e/v] \) is same as \( W \), except \( v^{W[e/v]} = e \).

\[
\begin{align*}
\chi^{G_1} & = 0 \\
y^{G_1} & = 0 \\
x^{G_1[1/x \ 4/y]} & = 1
\end{align*}
\]

iClicker 22.1  What is \( y^{G_1[1/x \ 4/y]} \)?

A: 0  B: 1  C: 3  D: 4
Default Interpretation of variables: Unless explicitly stated otherwise, $v^W = 0$, (or the min value in $|W|$ if $0 \not\in |W|$).

Notation: $W[e/v]$ is same as $W$, except $v^{W[e/v]} = e$.

$x^{G_1} = 0 \quad y^{G_1} = 0 \quad x^{G_1[1/x \ 4/y]} = 1$

iClicker 22.2  What is $t^{G_1[1/x \ 4/y]}$

A: 0  B: 1  C: 3  D: 4
For $t \in \text{term}(\Sigma)$; $W \in \text{World}[\Sigma]$, recursively define $t^W$. 

Proposition 1

For $t \in \text{term}(\Sigma)$; $W \in \text{World}[\Sigma]$, $t^W \in |W|$.

Proof: By structural induction on $t$.

Base cases:
- For $v \in \text{VAR}$, $v^W \in |W|$;
- For $k \in \Sigma$, $k^W \in |W|$.

Inductive case: indHyp: $t^W_1, \ldots, t^W_r \in |W|$.

$f^W: |W|^r \rightarrow |W|$, so $(f(t^W_1, \ldots, t^W_r))^W \in |W|$.

□
For $t \in \text{term}(\Sigma)$; $W \in \text{World}[\Sigma]$, recursively define $t^W$.

**base case 0:** For $v \in \text{VAR}$, $v^W$ already has default value.
Worlds Recursively Interpret Terms

For \( t \in \text{term}(\Sigma); \ W \in \text{World}[\Sigma] \), **recursively define** \( t^W \)

**base case 0:** For \( v \in \text{VAR} \), \( v^W \) already has default value.

**base case 1:** For constant symbol, \( k \in \Sigma \), \( k^W \) already defined.
For $t \in \text{term}(\Sigma); W \in \text{World}[\Sigma]$, recursively define $t^W$

**base case 0:** For $v \in \text{VAR}$, $v^W$ already has default value.

**base case 1:** For constant symbol, $k \in \Sigma$, $k^W$ already defined.

**inductive case:** For $t_1, \ldots, t_r \in \text{term}(\Sigma), f^r \in \Sigma$

\[
(f(t_1, \ldots, t_r))^W \overset{\text{def}}{=} f^W(t_1^W, \ldots, t_r^W)
\]
For \( t \in \text{term}(\Sigma) \); \( W \in \text{World}[\Sigma] \), **recursively define** \( t^W \)

**base case 0:** For \( v \in \text{VAR} \), \( v^W \) already has default value.

**base case 1:** For constant symbol, \( k \in \Sigma \), \( k^W \) already defined.

**inductive case:** For \( t_1, \ldots, t_r \in \text{term}(\Sigma) \), \( f^r \in \Sigma \)

\[
(f(t_1, \ldots, t_r))^W \overset{\text{def}}{=} f^W(t_1^W, \ldots, t_r^W)
\]

**Prop.** For \( t \in \text{term}(\Sigma) \); \( W \in \text{World}[\Sigma] \), \( t^W \in |W| \)
Worlds Recursively Interpret Terms

For \( t \in \text{term}(\Sigma); \ W \in \text{World}[\Sigma], \) **recursively define** \( t^W \)

**base case 0:** For \( v \in \text{VAR}, \ v^W \) already has default value.

**base case 1:** For constant symbol, \( k \in \Sigma, \ k^W \) already defined.

**inductive case:** For \( t_1, \ldots, t_r \in \text{term}(\Sigma), \ f^r \in \Sigma \)

\[
\left( f(t_1, \ldots, t_r) \right)^W \overset{\text{def}}{=} f^W(t_1^W, \ldots, t_r^W)
\]

**Prop.** For \( t \in \text{term}(\Sigma); \ W \in \text{World}[\Sigma], \ t^W \in |W| \)

**Proof:** By structural induction on \( t \).
For \( t \in \text{term}(\Sigma) ; W \in \text{World}[\Sigma] \), \textbf{recursively define} \( t^W \)

**base case 0:** For \( \nu \in \text{VAR}, \nu^W \) already has default value.

**base case 1:** For constant symbol, \( k \in \Sigma, k^W \) already defined.

**inductive case:** For \( t_1, \ldots, t_r \in \text{term}(\Sigma), f^r \in \Sigma \)

\[
(f(t_1, \ldots, t_r))^W \overset{\text{def}}{=} f^W(t_1^W, \ldots, t_r^W)
\]

**Prop.** For \( t \in \text{term}(\Sigma) ; W \in \text{World}[\Sigma], t^W \in \mid W \mid \)

**Proof:** By structural induction on \( t \).

**base cases:** For \( \nu \in \text{VAR}, \nu^W \in \mid W \mid ; \) for \( k \in \Sigma, k^W \in \mid W \mid \)
Worlds Recursively Interpret Terms

For \( t \in \text{term}(\Sigma) \); \( W \in \text{World}[\Sigma] \), recursively define \( t^W \)

**base case 0:** For \( v \in \text{VAR} \), \( v^W \) already has default value.

**base case 1:** For constant symbol, \( k \in \Sigma \), \( k^W \) already defined.

**inductive case:** For \( t_1, \ldots, t_r \in \text{term}(\Sigma) \), \( f^r \in \Sigma \)

\[
( f(t_1, \ldots, t_r) )^W \overset{\text{def}}{=} f^W(t_1^W, \ldots, t_r^W)
\]

**Prop.** For \( t \in \text{term}(\Sigma) \); \( W \in \text{World}[\Sigma] \), \( t^W \in |W| \)

**Proof:** By structural induction on \( t \).

**base cases:** For \( v \in \text{VAR} \), \( v^W \in |W| \); for \( k \in \Sigma \), \( k^W \in |W| \)

**inductive case: indHyp:** \( t_1^W, \ldots, t_r^W \in |W| \).
Worlds Recursively Interpret Terms

For $t \in \text{term}(\Sigma); \ W \in \text{World}[\Sigma]$, **recursively define** $t^W$

**base case 0:** For $v \in \text{VAR}, \ v^W$ already has default value.

**base case 1:** For constant symbol, $k \in \Sigma, \ k^W$ already defined.

**inductive case:** For $t_1, \ldots, t_r \in \text{term}(\Sigma), \ f^r \in \Sigma$

$$ (f(t_1, \ldots, t_r))^W \overset{\text{def}}{=} f^W(t_1^W, \ldots, t_r^W) $$

**Prop.** For $t \in \text{term}(\Sigma); \ W \in \text{World}[\Sigma]$, $t^W \in |W|$

**Proof:** By structural induction on $t$.

**base cases:** For $v \in \text{VAR}, \ v^W \in |W|; \text{ for } k \in \Sigma, \ k^W \in |W|$

**inductive case: indHyp:** $t_1^W, \ldots, t_r^W \in |W|$. \n
$f^W : |W|^r \rightarrow |W|$, so $(f(t_1, \ldots, t_r))^W = f^W(t_1^W, \ldots, t_r^W) \in |W|$. □
Tarski’s Recursive Definition of Truth

For every $G \in \text{World}[\Sigma]$ and $t \in \text{term}(\Sigma)$

$t^G \in |G|$
Tarski’s Recursive Definition of Truth

For every $G \in \text{World}[\Sigma]$ and $t \in \text{term}(\Sigma)$, $t^G \in |G|$

$G \models t_1 = t_2$ iff $t_1^G = t_2^G$
Tarski’s Recursive Definition of Truth

For every $G \in \text{World}[\Sigma]$ and $t \in \text{term}(\Sigma)$, $t^G \in |G|$.

$G \models t_1 = t_2$ iff $t_1^G = t_2^G$

$G \models P(t_1, \ldots, t_a)$ iff $(t_1^G, \ldots, t_a^G) \in P^G$

$P^a \in \Sigma$
Tarski’s Recursive Definition of Truth

For every \( G \in \text{World}[\Sigma] \) and \( t \in \text{term}(\Sigma) \)

\[
G \models t_1 = t_2 \quad \text{iff} \quad t_1^G = t_2^G
\]

\[
G \models P(t_1, \ldots, t_a) \quad \text{iff} \quad (t_1^G, \ldots, t_a^G) \in P^G
\]

\[
G \models \sim \alpha \quad \text{iff} \quad G \not\models \alpha
\]
Tarski’s Recursive Definition of Truth

For every $G \in \text{World}[\Sigma]$ and $t \in \text{term}(\Sigma)$, $t^G \in |G|$

$G \models t_1 = t_2 \iff t_1^G = t_2^G$

$G \models P(t_1, \ldots, t_a) \iff (t_1^G, \ldots, t_a^G) \in P^G$  $P^a \in \Sigma$

$G \models \sim \alpha \iff G \not\models \alpha$  PropCalc

$G \models \alpha \land \beta \iff G \models \alpha \text{ and } G \models \beta$  PropCalc
Tarski’s Recursive Definition of Truth

For every $G \in \text{World}[\Sigma]$ and $t \in \text{term}(\Sigma)$

- $G \models t_1 = t_2$ iff $t_1^G = t_2^G$
- $G \models P(t_1, \ldots, t_a)$ iff $(t_1^G, \ldots, t_a^G) \in P^G$
- $G \models \sim \alpha$ iff $G \not\models \alpha$
- $G \models \alpha \land \beta$ iff $G \models \alpha$ and $G \models \beta$
- $G \models \alpha \lor \beta$ iff $G \models \alpha$ or $G \models \beta$

$t^G \in |G|$
Tarski’s Recursive Definition of Truth

For every $G \in \text{World}[\Sigma]$ and $t \in \text{term}(\Sigma)$, $t^G \in |G|$

- $G \models t_1 = t_2$ iff $t_1^G = t_2^G$
- $G \models P(t_1, \ldots, t_a)$ iff $(t_1^G, \ldots, t_a^G) \in P^G$
- $G \models \sim \alpha$ iff $G \not\models \alpha$
- $G \models \alpha \land \beta$ iff $G \models \alpha$ and $G \models \beta$
- $G \models \alpha \lor \beta$ iff $G \models \alpha$ or $G \models \beta$
- $G \models \forall x(\alpha)$ iff for all $a \in |G|$ $G[a/x] \models \alpha$
Tarski’s Recursive Definition of Truth

For every $G \in \text{World}[\Sigma]$ and $t \in \text{term}(\Sigma)$

$G \models t_1 = t_2$ iff $t_1^G = t_2^G$

$G \models P(t_1, \ldots, t_a)$ iff $(t_1^G, \ldots, t_a^G) \in P^G$

$G \models \sim \alpha$ iff $G \not\models \alpha$

$G \models \alpha \land \beta$ iff $G \models \alpha$ and $G \models \beta$

$G \models \alpha \lor \beta$ iff $G \models \alpha$ or $G \models \beta$

$G \models \forall x(\alpha)$ iff for all $a \in |G|$ $G[a/x] \models \alpha$

$G \models \exists x(\alpha)$ iff exists $a \in |G|$ $G[a/x] \models \alpha$
Tarski’s Recursive Definition of Truth

\[ G(t_1 = t_2) \overset{\text{def}}{=} t_1^G = t_2^G \]
Tarski’s Recursive Definition of Truth

\[ G(t_1 = t_2) \overset{\text{def}}{=} t_1^G == t_2^G \]

\[ G(P(t_1, \ldots, t_a)) \overset{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G \]
Tarski’s Recursive Definition of Truth

\[ G(t_1 = t_2) \overset{\text{def}}{=} t_1^G = t_2^G \]

\[ G(P(t_1, \ldots, t_a)) \overset{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G \]

\[ G(\sim \alpha) \overset{\text{def}}{=} 1 - G(\alpha) \]
Tarski’s Recursive Definition of Truth

\[
G(t_1 = t_2) \overset{\text{def}}{=} t_1^G = t_2^G \\
G(P(t_1, \ldots, t_a)) \overset{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G \\
G(\sim \alpha) \overset{\text{def}}{=} 1 - G(\alpha) \\
G(\alpha \land \beta) \overset{\text{def}}{=} \min(G(\alpha), G(\beta))
\]
Tarski’s Recursive Definition of Truth

\[ G(t_1 = t_2) \overset{\text{def}}{=} t_1^G \iff t_2^G \]

\[ G(P(t_1, \ldots, t_a)) \overset{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G \]

\[ G(\sim \alpha) \overset{\text{def}}{=} 1 - G(\alpha) \]

\[ G(\alpha \land \beta) \overset{\text{def}}{=} \min(G(\alpha), G(\beta)) \]

\[ G(\alpha \lor \beta) \overset{\text{def}}{=} \max(G(\alpha), G(\beta)) \]
Tarski’s Recursive Definition of Truth

\[
G(t_1 = t_2) \overset{\text{def}}{=} t_1^G = t_2^G
\]

\[
G(P(t_1, \ldots, t_a)) \overset{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G
\]

\[
G(\sim \alpha) \overset{\text{def}}{=} 1 - G(\alpha)
\]

\[
G(\alpha \land \beta) \overset{\text{def}}{=} \min(G(\alpha), G(\beta))
\]

\[
G(\alpha \lor \beta) \overset{\text{def}}{=} \max(G(\alpha), G(\beta))
\]

\[
G(\forall x(\alpha)) \overset{\text{def}}{=} \min_{a \in |G|} G[a/x](\alpha)
\]
Tarski’s Recursive Definition of Truth

\[ G(t_1 = t_2) \overset{\text{def}}{=} t_1^G = t_2^G \]

\[ G(P(t_1, \ldots, t_a)) \overset{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G \]

\[ G(\sim \alpha) \overset{\text{def}}{=} 1 - G(\alpha) \]

\[ G(\alpha \land \beta) \overset{\text{def}}{=} \min(G(\alpha), G(\beta)) \]

\[ G(\alpha \lor \beta) \overset{\text{def}}{=} \max(G(\alpha), G(\beta)) \]

\[ G(\forall x(\alpha)) \overset{\text{def}}{=} \min_{a \in |G|} G[a/x](\alpha) \]

\[ G(\exists x(\alpha)) \overset{\text{def}}{=} \max_{a \in |G|} G[a/x](\alpha) \]
**Truth Game**: a two player game that is an equivalent but more fun way to tell whether $\mathcal{W} \models \varphi$. First put $\varphi$ into NNF.
Truth Game: a two player game that is an equivalent but more fun way to tell whether \( W \models \varphi \). First put \( \varphi \) into NNF.

Dumbledore wants to show that \( W \models \varphi \)
Truth Game: a two player game that is an equivalent but more fun way to tell whether $\mathcal{W} \models \varphi$. First put $\varphi$ into NNF.

Dumbledore wants to show that $\mathcal{W} \models \varphi$

Gandalf wants to show that $\mathcal{W} \not\models \varphi$. 
Truth Game: a two player game that is an equivalent but more fun way to tell whether $\mathcal{W} \models \varphi$. First put $\varphi$ into NNF.

Dumbledore wants to show that $\mathcal{W} \models \varphi$.

Gandalf wants to show that $\mathcal{W} \not\models \varphi$.

base case: if $\varphi$ is a literal, then D wins iff $\mathcal{W} \models \varphi$. 
Truth Game: a two player game that is an equivalent but more fun way to tell whether $W \models \varphi$. First put $\varphi$ into NNF.

**Dumbledore** wants to show that $W \models \varphi$.

**Gandalf** wants to show that $W \not\models \varphi$.

**base case:** if $\varphi$ is a literal, then $D$ wins iff $W \models \varphi$.

**inductive cases:**
**Truth Game**: a two player game that is an equivalent but more fun way to tell whether $W \models \varphi$. First put $\varphi$ into NNF.

Dumbledore wants to show that $W \models \varphi$

Gandalf wants to show that $W \not\models \varphi$.

**base case**: if $\varphi$ is a literal, then D wins iff $W \models \varphi$.

**inductive cases**:

$W \models \varphi \land \psi$  \hspace{1cm} G chooses $\alpha \in \{\varphi, \psi\}$ continue on:  \hspace{1cm} $W \models \alpha$
**Truth Game**: a two player game that is an equivalent but more fun way to tell whether \( W \models \varphi \). First put \( \varphi \) into NNF.

**Dumbledore** wants to show that \( W \models \varphi \).

**Gandalf** wants to show that \( W \not\models \varphi \).

**base case**: if \( \varphi \) is a literal, then \( D \) wins iff \( W \models \varphi \).

**inductive cases**:

- \( W \models \varphi \land \psi \)  
  \( G \) chooses \( \alpha \in \{ \varphi, \psi \} \) continue on: \( W \models \alpha \)

- \( W \models \varphi \lor \psi \)  
  \( D \) chooses \( \alpha \in \{ \varphi, \psi \} \) continue on: \( W \models \alpha \)
**Truth Game**: a two player game that is an equivalent but more fun way to tell whether $W \models \varphi$. First put $\varphi$ into NNF.

**Dumbledore** wants to show that $W \models \varphi$

**Gandalf** wants to show that $W \not\models \varphi$.

**base case**: if $\varphi$ is a literal, then $D$ wins iff $W \models \varphi$.

**inductive cases:**

- $W \models \varphi \land \psi\quad G$ chooses $\alpha \in \{\varphi, \psi\}$ continue on: $W \models \alpha$
- $W \models \varphi \lor \psi\quad D$ chooses $\alpha \in \{\varphi, \psi\}$ continue on: $W \models \alpha$
- $W \models \forall x \varphi\quad G$ chooses $a \in |W|$ continue on: $W a/x \models \varphi$
**Truth Game**: a two player game that is an equivalent but more fun way to tell whether $W \models \varphi$. First put $\varphi$ into NNF.

**Dumbledore** wants to show that $W \models \varphi$

**Gandalf** wants to show that $W \not\models \varphi$.

**base case**: if $\varphi$ is a literal, then $D$ wins iff $W \models \varphi$.

**inductive cases**:

- $W \models \varphi \land \psi$  \quad G chooses $\alpha \in \{\varphi, \psi\}$ continue on: $W \models \alpha$
- $W \models \varphi \lor \psi$  \quad D chooses $\alpha \in \{\varphi, \psi\}$ continue on: $W \models \alpha$
- $W \models \forall x \ \varphi$  \quad G chooses $a \in |W|$ continue on: $W a/x \models \varphi$
- $W \models \exists x \ \varphi$  \quad D chooses $a \in |W|$ continue on: $W a/x \models \varphi$
Truth Game Example

Does $W \models \forall x (x = s \lor \exists y \ E(y, x))$?
Truth Game Example

Does $W \models \forall x (x = s \lor \exists y E(y, x))$?

G moves, chooses $x = 4$
Truth Game Example

Does $W \models \forall x \ (x = s \lor \exists y \ E(y, x))$ ?

$G$ moves, chooses $x = 4$

Does $W[4/x] \models x = s \lor \exists y \ E(y, x)$ ?
Truth Game Example

Does $\mathcal{W} \models \forall x (x = s \lor \exists y E(y, x))$ ?
\begin{itemize}
    \item \textbf{G} moves, chooses $x = 4$
\end{itemize}

Does $\mathcal{W}[4/x] \models x = s \lor \exists y E(y, x)$ ?
\begin{itemize}
    \item \textbf{D} moves, chooses $\exists y E(y, x)$
\end{itemize}
Truth Game Example

Does $W \models \forall x (x = s \lor \exists y E(y, x))$?

G moves, chooses $x = 4$

Does $W[4/x] \models x = s \lor \exists y E(y, x)$?

D moves, chooses $\exists y E(y, x)$

Does $W[4/x] \models \exists y E(y, x)$?
Truth Game Example

Does $W \models \forall x \ (x = s \lor \exists y \ E(y, x))$ ?
G moves, chooses $x = 4$

Does $W[4/x] \models x = s \lor \exists y \ E(y, x)$ ?
D moves, chooses $\exists y \ E(y, x)$

Does $W[4/x] \models \exists y \ E(y, x)$ ?
D moves, chooses $y = 1$
Truth Game Example

Does $W \models \forall x (x = s \lor \exists y E(y, x))$?
- $G$ moves, chooses $x = 4$

Does $W[4/x] \models x = s \lor \exists y E(y, x)$?
- $D$ moves, chooses $\exists y E(y, x)$

Does $W[4/x] \models \exists y E(y, x)$?
- $D$ moves, chooses $y = 1$

Does $W[4/x, 1/y] \models E(y, x)$?
Truth Game Example

Does $\mathcal{W} \models \forall x (x = s \lor \exists y E(y, x))$?  
**G** moves, chooses $x = 4$

Does $\mathcal{W}[4/x] \models x = s \lor \exists y E(y, x)$?  
**D** moves, chooses $\exists y E(y, x)$

Does $\mathcal{W}[4/x] \models \exists y E(y, x)$?  
**D** moves, chooses $y = 1$

Does $\mathcal{W}[4/x, 1/y] \models E(y, x)$?  
Yes, **D** wins!
Thm. For any $\Sigma$, $\varphi \in \text{PredCalc}\Sigma$, in NNF, $\mathcal{W} \in \text{World}[\Sigma]$,

- D wins the truth game on $\mathcal{W}, \varphi$ iff $\mathcal{W} \models \varphi$
- G wins the truth game on $\mathcal{W}, \varphi$ iff $\mathcal{W} \models \neg \varphi$

Proof: By induction on the structure of $\varphi$. Details in hw4 □
Thm. For any $\Sigma, \varphi \in \text{PredCalc}\Sigma$, in NNF, $\mathcal{W} \in \text{World}[\Sigma]$,

- $D$ wins the truth game on $\mathcal{W}, \varphi$  iff  $\mathcal{W} \models \varphi$

- $G$ wins the truth game on $\mathcal{W}, \varphi$  iff  $\mathcal{W} \models \neg \varphi$

Proof: By induction on the structure of $\varphi$.  

Details in hw4
Thm. For any $\Sigma$, $\varphi \in \text{PredCalc}\Sigma$, in NNF, $\mathcal{W} \in \text{World}[\Sigma]$, 

- $D$ wins the truth game on $\mathcal{W}, \varphi$ iff $\mathcal{W} \models \varphi$
- $G$ wins the truth game on $\mathcal{W}, \varphi$ iff $\mathcal{W} \models \neg \varphi$

Proof: By induction on the structure of $\varphi$.

Details in hw4