CS250: Discrete Math for Computer Science

L22: Inductive Definitions and Structural Induction

Inductive Definitions and Structural Induction

We define our data structures – or other objects of interest – inductively.

This is useful because we can:

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.

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**.

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

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 langugages allow recursive function definitions on recursively defined datatypes (Python – hw4)

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 langugages allow recursive function definitions on recursively defined datatypes (Python – hw4)
- Main examples today: logical formulas, truth

Terms in PredCalc

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

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

variables are terms

Terms in PredCalc

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

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

variables are terms

base 1. $k \in \Sigma \quad \rightarrow \quad k \in \mathbf{term}(\Sigma)$

constant symbols are terms

Terms in PredCalc

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

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

variables are terms

base 1. $k \in \Sigma \quad \rightarrow \quad k \in \textbf{term}(\Sigma)$

constant symbols are terms

ind. 2. $t_1, \ldots, t_r \in \textbf{term}(\Sigma), f^r \in \Sigma \quad \rightarrow \quad f(t_1, \ldots, t_r) \in \textbf{term}(\Sigma)$

terms are closed under function symbols

Terms in PredCalc

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

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

variables are terms

base 1. $k \in \Sigma \quad \rightarrow \quad k \in \textbf{term}(\Sigma)$

constant symbols are terms

ind. 2. $t_1, \ldots, t_r \in \textbf{term}(\Sigma), f^r \in \Sigma \quad \rightarrow \quad f(t_1, \ldots, t_r) \in \textbf{term}(\Sigma)$

terms are closed under function symbols

$$\textbf{term}(\Sigma_{\text{garst}}) \;=\; \mathrm{VAR} \cup \{s, t\} \qquad =$$

Terms in PredCalc

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

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

variables are terms

base 1. $k \in \Sigma \quad \rightarrow \quad k \in \textbf{term}(\Sigma)$

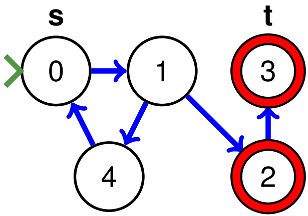
constant symbols are terms

ind. 2. $t_1, \ldots, t_r \in \textbf{term}(\Sigma), f^r \in \Sigma \quad \rightarrow \quad f(t_1, \ldots, t_r) \in \textbf{term}(\Sigma)$

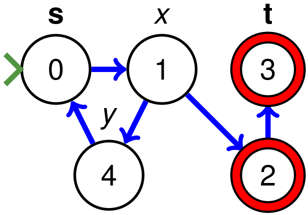
terms are closed under function symbols

$$\textbf{term}(\Sigma_{\text{garst}}) \;=\; \mathrm{VAR} \,\cup\, \{s, t\} \qquad =$$

$$\textbf{term}(\Sigma_{\text{\#thy}}) \;=\; \{0, 1, x, \ldots, \ldots, x \cdot y, \ldots (x+1) \cdot (y+0), \ldots\}$$

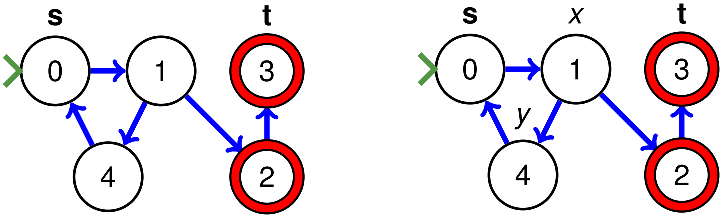


G_1



$G_1[1/x\ 4/y]$

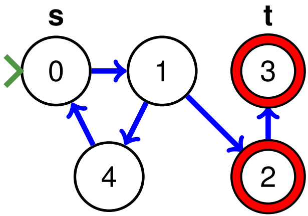
Default Interpretation of variables: Unless explicitly stated otherwise, $v^W = 0$, (or the min value in $|W|$ if $0 \notin |W|$).



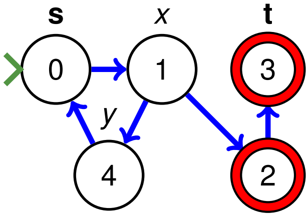
G_1 $\qquad\qquad$ $G_1[1/x\ 4/y]$

Default Interpretation of variables: Unless explicitly stated otherwise, $v^W = 0$, (or the min value in $|W|$ if $0 \notin |W|$).

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



G_1

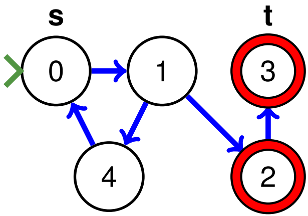


$G_1[1/x\ 4/y]$

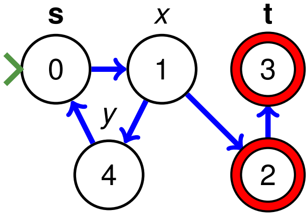
Default Interpretation of variables: Unless explicitly stated otherwise, $v^W = 0$, (or the min value in $|W|$ if $0 \notin |W|$).

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

$x^{G_1} = 0$



G_1

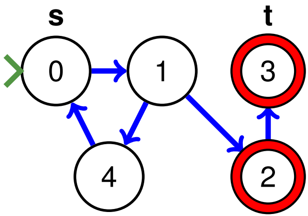


$G_1[1/x\ 4/y]$

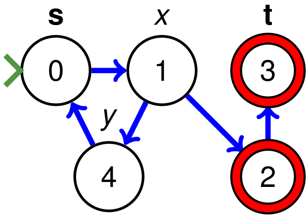
Default Interpretation of variables: Unless explicitly stated otherwise, $v^W = 0$, (or the min value in $|W|$ if $0 \notin |W|$).

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

$x^{G_1} = 0 \quad y^{G_1} = 0$



G_1

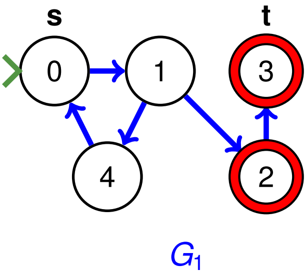


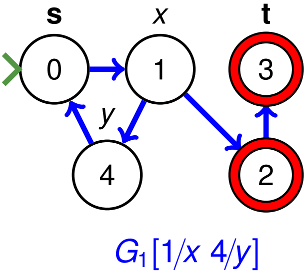
$G_1[1/x\ 4/y]$

Default Interpretation of variables: Unless explicitly stated otherwise, $v^W = 0$, (or the min value in $|W|$ if $0 \notin |W|$).

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

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





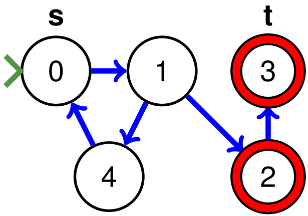
Default Interpretation of variables: Unless explicitly stated otherwise, $v^W = 0$, (or the min value in $|W|$ if $0 \notin |W|$).

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

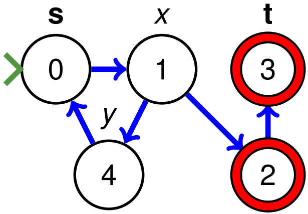
$x^{G_1} = 0 \quad y^{G_1} = 0 \quad x^{G_1[1/x\ 4/y]} = 1$

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

A: 0 B: 1 C: 3 D: 4



G_1



$G_1[1/x\ 4/y]$

Default Interpretation of variables: Unless explicitly stated otherwise, $v^W = 0$, (or the min value in $|W|$ if $0 \notin |W|$).

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

Worlds Recursively Interpret Terms

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

Worlds Recursively Interpret Terms

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

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

Worlds Recursively Interpret Terms

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

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

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

Worlds Recursively Interpret Terms

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

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

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

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

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

Worlds Recursively Interpret Terms

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

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

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

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

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

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

Worlds Recursively Interpret Terms

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

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

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

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

$$(\, f(t_1, \ldots, t_r)\,)^W \quad \stackrel{\mathrm{def}}{=} \quad f^W(t_1^W, \ldots, t_r^W)$$

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

Proof: By structural induction on t.

Worlds Recursively Interpret Terms

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

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

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

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

$$(\, f(t_1, \ldots, t_r)\,)^W \quad \stackrel{\mathrm{def}}{=} \quad f^W(t_1^W, \ldots, t_r^W)$$

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

Proof: By structural induction on t.

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

Worlds Recursively Interpret Terms

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

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

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

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

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

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

Proof: By structural induction on t.

base cases: For $v \in \mathrm{VAR}$, $v^W \in |W|$; 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 $t \in \textbf{term}(\Sigma)$; $W \in \mathrm{World}[\Sigma]$, **recursively define** t^W

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

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

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

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

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

Proof: By structural induction on t.

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

inductive case: indHyp: $t_1^W, \ldots, t_r^W \in |W|$.

$f^W : |W|^r \to |W|$, so $(f(t_1, \ldots, t_r))^W = f^W(t_1^W, \ldots, t_r^W) \in |W|$. $\square$

Tarski's Recursive Definition of Truth

For every $G \in \mathrm{World}[\Sigma]$ and $t \in \textbf{term}(\Sigma)$ $\qquad t^G \in |G|$

Tarski's Recursive Definition of Truth

For every $G \in \mathrm{World}[\Sigma]$ and $t \in \textbf{term}(\Sigma)$ $\qquad t^G \in |G|$

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

Tarski's Recursive Definition of Truth

For every $G \in \mathrm{World}[\Sigma]$ and $t \in \mathbf{term}(\Sigma)$ $\qquad t^G \in |G|$

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

$$G \models P(t_1, \ldots, t_a) \quad \text{iff} \quad (t_1^G, \ldots, t_a^G) \in P^G \qquad P^a \in \Sigma$$

Tarski's Recursive Definition of Truth

For every $G \in \mathrm{World}[\Sigma]$ and $t \in \textbf{term}(\Sigma)$ $\qquad t^G \in |G|$

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

$$G \models P(t_1, \ldots, t_a) \quad \text{iff} \quad (t_1^G, \ldots, t_a^G) \in P^G \qquad P^a \in \Sigma$$

$$G \models \sim\alpha \quad \text{iff} \quad G \not\models \alpha \qquad \text{PropCalc}$$

Tarski's Recursive Definition of Truth

For every $G \in \mathrm{World}[\Sigma]$ and $t \in \textbf{term}(\Sigma)$ $\quad t^G \in |G|$

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	$\models$	$\alpha \wedge \beta$	iff	$G \models \alpha$ **and** $G \models \beta$	PropCalc

Tarski's Recursive Definition of Truth

For every $G \in \mathrm{World}[\Sigma]$ and $t \in \mathbf{term}(\Sigma)$ $\qquad t^G \in |G|$

$$
\begin{array}{llllll}
G & \models & t_1 = t_2 & \text{iff} & t_1^G = t_2^G & \\
G & \models & P(t_1, \ldots, t_a) & \text{iff} & (t_1^G, \ldots, t_a^G) \in P^G & P^a \in \Sigma \\
G & \models & \sim\alpha & \text{iff} & G \not\models \alpha & \text{PropCalc} \\
G & \models & \alpha \wedge \beta & \text{iff} & G \models \alpha \quad \textbf{and} \quad G \models \beta & \text{PropCalc} \\
G & \models & \alpha \vee \beta & \text{iff} & G \models \alpha \quad \textbf{or} \quad G \models \beta & \text{PropCalc}
\end{array}
$$

Tarski's Recursive Definition of Truth

For every $G \in \mathrm{World}[\Sigma]$ and $t \in \textbf{term}(\Sigma)$ $\qquad t^G \in |G|$

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	$\models$	$\alpha \wedge \beta$	iff	$G \models \alpha$ **and** $G \models \beta$	PropCalc
G	$\models$	$\alpha \vee \beta$	iff	$G \models \alpha$ **or** $G \models \beta$	PropCalc
G	$\models$	$\forall x(\alpha)$	iff	**for all** $a \in |G|$ $\quad G[a/x] \models \alpha$	

Tarski's Recursive Definition of Truth

For every $G \in \mathrm{World}[\Sigma]$ and $t \in \textbf{term}(\Sigma)$ $\quad t^G \in |G|$

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	$\models$	$\alpha \wedge \beta$	iff	$G \models \alpha$ **and** $G \models \beta$	PropCalc
G	$\models$	$\alpha \vee \beta$	iff	$G \models \alpha$ **or** $G \models \beta$	PropCalc
G	$\models$	$\forall x(\alpha)$	iff	**for all** $a \in \lvert G \rvert \quad G[a/x] \models \alpha$	
G	$\models$	$\exists x(\alpha)$	iff	**exists** $a \in \lvert G \rvert \quad G[a/x] \models \alpha$	

Tarski's Recursive Definition of Truth

$$G(t_1 = t_2) \quad \stackrel{\text{def}}{=} \quad t_1^G == t_2^G$$

Tarski's Recursive Definition of Truth

$$G(t_1 = t_2) \stackrel{\text{def}}{=} t_1^G == t_2^G$$

$$G(P(t_1, \ldots, t_a)) \stackrel{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G$$

Tarski's Recursive Definition of Truth

$$
\begin{aligned}
G(t_1 = t_2) &\stackrel{\text{def}}{=} t_1^G == t_2^G \\
G(P(t_1, \ldots, t_a)) &\stackrel{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G \\
G(\sim \alpha) &\stackrel{\text{def}}{=} 1 - G(\alpha)
\end{aligned}
$$

Tarski's Recursive Definition of Truth

$$
\begin{aligned}
G(t_1 = t_2) &\stackrel{\text{def}}{=} t_1^G == t_2^G \\
G(P(t_1, \ldots, t_a)) &\stackrel{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G \\
G(\sim \alpha) &\stackrel{\text{def}}{=} 1 - G(\alpha) \\
G(\alpha \wedge \beta) &\stackrel{\text{def}}{=} \min(G(\alpha), G(\beta))
\end{aligned}
$$

Tarski's Recursive Definition of Truth

$$G(t_1 = t_2) \stackrel{\text{def}}{=} t_1^G == t_2^G$$

$$G(P(t_1, \ldots, t_a)) \stackrel{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G$$

$$G(\sim\alpha) \stackrel{\text{def}}{=} 1 - G(\alpha)$$

$$G(\alpha \wedge \beta) \stackrel{\text{def}}{=} \min(G(\alpha), G(\beta))$$

$$G(\alpha \vee \beta) \stackrel{\text{def}}{=} \max(G(\alpha), G(\beta))$$

Tarski's Recursive Definition of Truth

$$G(t_1 = t_2) \stackrel{\text{def}}{=} t_1^G == t_2^G$$

$$G(P(t_1, \ldots, t_a)) \stackrel{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G$$

$$G(\sim\alpha) \stackrel{\text{def}}{=} 1 - G(\alpha)$$

$$G(\alpha \wedge \beta) \stackrel{\text{def}}{=} \min(G(\alpha), G(\beta))$$

$$G(\alpha \vee \beta) \stackrel{\text{def}}{=} \max(G(\alpha), G(\beta))$$

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

Tarski's Recursive Definition of Truth

$$
\begin{aligned}
G(t_1 = t_2) &\stackrel{\text{def}}{=} t_1^G == t_2^G \\
G(P(t_1, \ldots, t_a)) &\stackrel{\text{def}}{=} (t_1^G, \ldots, t_a^G) \in P^G \\
G(\sim \alpha) &\stackrel{\text{def}}{=} 1 - G(\alpha) \\
G(\alpha \wedge \beta) &\stackrel{\text{def}}{=} \min(G(\alpha), G(\beta)) \\
G(\alpha \vee \beta) &\stackrel{\text{def}}{=} \max(G(\alpha), G(\beta)) \\
G(\forall x(\alpha)) &\stackrel{\text{def}}{=} \min_{a \in |G|} G[a/x](\alpha) \\
G(\exists x(\alpha)) &\stackrel{\text{def}}{=} \max_{a \in |G|} G[a/x](\alpha)
\end{aligned}
$$

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

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

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

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

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

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

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

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

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

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

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

base case: if φ 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 φ into NNF.

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

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

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

inductive cases:

$W \models \varphi \wedge \psi$ **G** 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 φ into NNF.

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

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

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

inductive cases:

$W \models \varphi \wedge \psi$ **G** chooses $\alpha \in \{\varphi, \psi\}$ continue on: $W \models \alpha$

$W \models \varphi \vee \psi$ **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 φ into NNF.

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

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

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

$W \models \varphi \wedge \psi$	**G** chooses $\alpha \in \{\varphi, \psi\}$ continue on:	$W \models \alpha$
$W \models \varphi \vee \psi$	**D** chooses $\alpha \in \{\varphi, \psi\}$ continue on:	$W \models \alpha$
$W \models \forall x\, \varphi$	**G** chooses $a \in \lvert W \rvert$ 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 φ into NNF.

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

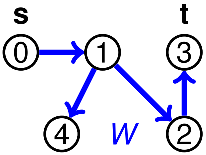
Gandalf wants to show that $W \not\models \varphi$.

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

inductive cases:

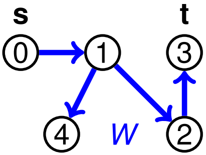
$W \models \varphi \wedge \psi$	**G** chooses	$\alpha \in \{\varphi, \psi\}$ continue on:	$W \models \alpha$
$W \models \varphi \vee \psi$	**D** chooses	$\alpha \in \{\varphi, \psi\}$ continue on:	$W \models \alpha$
$W \models \forall x\, \varphi$	**G** chooses	$a \in \lvert W \rvert$ continue on:	$W\, a/x \models \varphi$
$W \models \exists x\, \varphi$	**D** chooses	$a \in \lvert W \rvert$ continue on:	$W\, a/x \models \varphi$

Truth Game Example



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

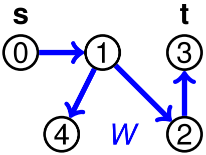
Truth Game Example



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

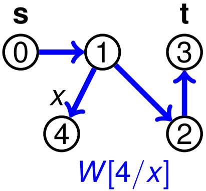
G moves, chooses $x = 4$

Truth Game Example



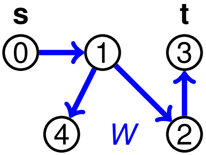
Does $W \models \forall x\ (x = s \vee \exists y\ E(y, x))$?

G moves, chooses $x = 4$



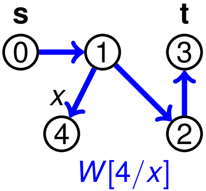
Does $W[4/x] \models x = s \vee \exists y\ E(y, x)$?

Truth Game Example



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

G moves, chooses $x = 4$



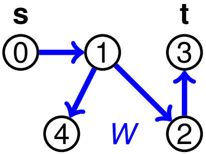
Does $W[4/x] \models x = s \vee \exists y \, E(y, x)$?

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

Truth Game Example

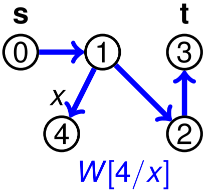
Does $W \models \forall x\ (x = s \vee \exists y\ E(y, x))$?

G moves, chooses $x = 4$



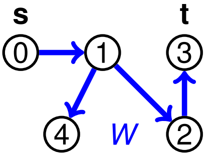
Does $W[4/x] \models x = s \vee \exists y\ E(y, x)$?

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

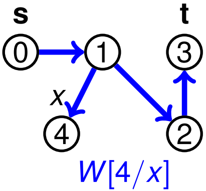


Does $W[4/x] \models \exists y\ E(y, x)$?

Truth Game Example



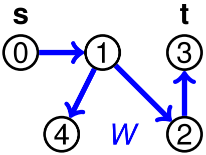
Does $W \models \forall x\ (x = s \vee \exists y\ E(y,x))$?
G moves, chooses $x = 4$



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

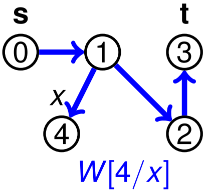
Does $W[4/x] \models \exists y\ E(y,x)$?
D moves, chooses $y = 1$

Truth Game Example



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

G moves, chooses $x = 4$

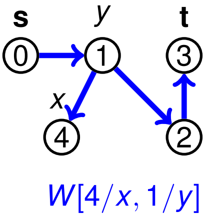


Does $W[4/x] \models x = s \vee \exists y\ E(y, x)$?

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

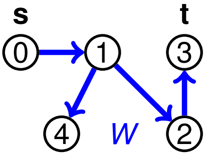
Does $W[4/x] \models \exists y\ E(y, x)$?

D moves, chooses $y = 1$



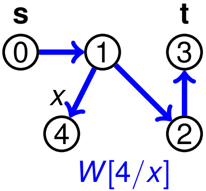
Does $W[4/x, 1/y] \models E(y, x)$?

Truth Game Example



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

G moves, chooses $x = 4$

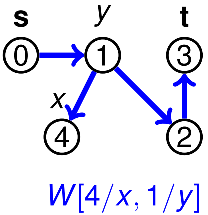


Does $W[4/x] \models x = s \vee \exists y\, E(y, x)$?

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

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

D moves, chooses $y = 1$



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

Yes, **D** wins!

Tarski's Def. of Truth and Truth Game are Equivalent

Thm. For any Σ, $\varphi \in \mathrm{PredCalc}\Sigma$, in NNF, $\mathcal{W} \in \mathrm{World}[\Sigma]$,

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

G wins the truth game on $\mathcal{W}, \varphi \quad \text{iff} \quad \mathcal{W} \models \sim\varphi$

Tarski's Def. of Truth and Truth Game are Equivalent

Thm. For any Σ, $\varphi \in \mathrm{PredCalc}\Sigma$, in NNF, $\mathcal{W} \in \mathrm{World}[\Sigma]$,

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

G wins the truth game on $\mathcal{W}, \varphi \quad \text{iff} \quad \mathcal{W} \models \sim\varphi$

Proof: By induction on the structure of φ.

Tarski's Def. of Truth and Truth Game are Equivalent

Thm. For any Σ, $\varphi \in \mathrm{PredCalc}\Sigma$, in NNF, $\mathcal{W} \in \mathrm{World}[\Sigma]$,

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

$$\textbf{G} \text{ wins the truth game on } \mathcal{W}, \varphi \quad \text{iff} \quad \mathcal{W} \models \sim\varphi$$

Proof: By induction on the structure of φ.

Details in hw4 $\square$