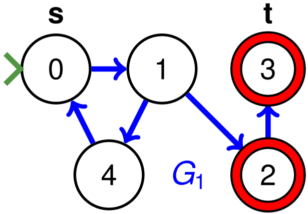
CS250: Discrete Math for Computer Science

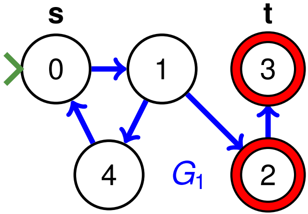
L8: More PredCalc

vocabulary of graphs $\Sigma_{\text{garst}} = (A^1, R^1, E^2; s, t)$



$G_1 \in \text{World}[\Sigma_{\text{garst}}]$ = **set of graphs** + pred. A, R; const. s, t

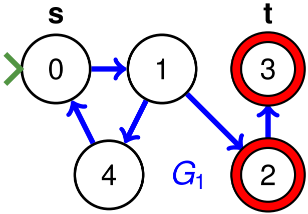
vocabulary of graphs $\Sigma_{\text{garst}} = (A^1, R^1, E^2; s, t)$



$G_1 \in \mathrm{World}[\Sigma_{\text{garst}}] =$ **set of graphs** + pred. A, R; const. s, t

$$|G_1| \;=\; V^{G_1} \;=\; \{0, 1, \ldots, 4\}$$ the **vertices**

vocabulary of graphs $\Sigma_{\text{garst}} = (A^1, R^1, E^2; s, t)$

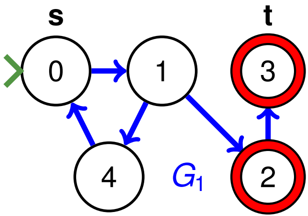


$G_1 \in \text{World}[\Sigma_{\text{garst}}]$ = **set of graphs** + pred. A, R; const. s, t

$$|G_1| = V^{G_1} = \{0, 1, \ldots, 4\} \quad \text{the } \textbf{vertices}$$

$$E^{G_1} = \{(0,1), (1,2), (1,4), (2,3), (4,0)\}$$

vocabulary of graphs $\Sigma_{\text{garst}} = (A^1, R^1, E^2; s, t)$



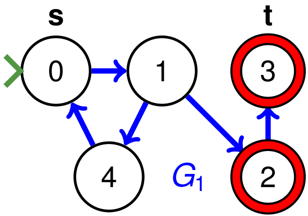
$G_1 \in \mathrm{World}[\Sigma_{\text{garst}}] =$ **set of graphs** + pred. A, R; const. s, t

$$|G_1| = V^{G_1} = \{0, 1, \ldots, 4\} \quad \text{the \textbf{vertices}}$$

$$E^{G_1} = \{(0,1), (1,2), (1,4), (2,3), (4,0)\}$$

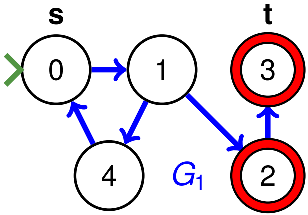
$$A^{G_1} = \{0\} \qquad R^{G_1} = \{2, 3\}$$

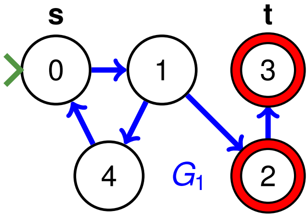
vocabulary of graphs $\Sigma_{\text{garst}} = (A^1, R^1, E^2; s, t)$



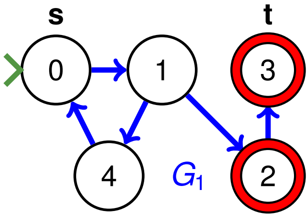
$G_1 \in \text{World}[\Sigma_{\text{garst}}]$ = **set of graphs** + pred. A, R; const. s, t

$$|G_1| = V^{G_1} = \{0, 1, \ldots, 4\} \quad \text{the } \textbf{vertices}$$
$$E^{G_1} = \{(0,1), (1,2), (1,4), (2,3), (4,0)\}$$
$$A^{G_1} = \{0\} \qquad R^{G_1} = \{2, 3\}$$
$$s^{G_1} = 0 \qquad t^{G_1} = 3$$

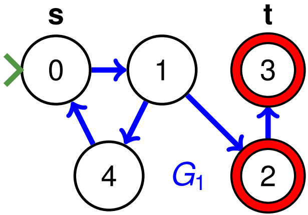

s
t
0
1
3
4
G_1
2



T/F: $G_1 \models \forall x \exists y\, E(x, y)$

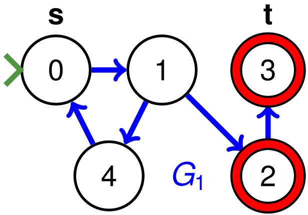


T/F: $G_1 \models \forall x \exists y \, E(x, y)$ **False**



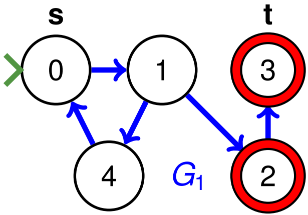
T/F: $G_1 \models \forall x \exists y \, E(x, y)$ **False**

T/F: $G_1 \models \exists x \forall y \sim E(x, y)$



T/F: $G_1 \models \forall x \exists y\ E(x, y)$ **False**

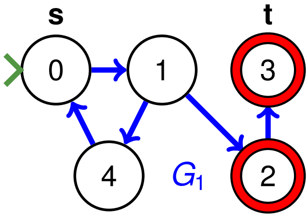
T/F: $G_1 \models \exists x \forall y\ \sim E(x, y)$ **True**



T/F: $G_1 \models \forall x \exists y \; E(x, y)$ **False**

T/F: $G_1 \models \exists x \forall y \; \sim E(x, y)$ **True**

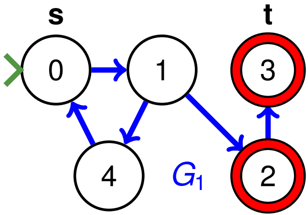
T/F: $G_1 \models \exists x \; E(x, x)$



T/F: $G_1 \models \forall x \exists y\ E(x, y)$ **False**

T/F: $G_1 \models \exists x \forall y\ \sim E(x, y)$ **True**

T/F: $G_1 \models \exists x\ E(x, x)$ **False**

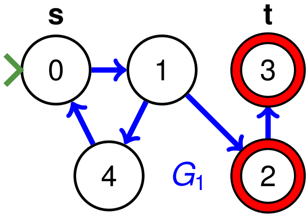


T/F: $G_1 \models \forall x \exists y\ E(x, y)$ **False**

T/F: $G_1 \models \exists x \forall y\ \sim E(x, y)$ **True**

T/F: $G_1 \models \exists x\ E(x, x)$ **False**

T/F: $G_1 \models \forall x \sim E(x, x)$

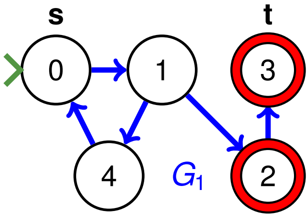


T/F: $G_1 \models \forall x \exists y\, E(x,y)$ **False**

T/F: $G_1 \models \exists x \forall y\, \sim E(x,y)$ **True**

T/F: $G_1 \models \exists x\, E(x,x)$ **False**

T/F: $G_1 \models \forall x \sim E(x,x)$ **True**



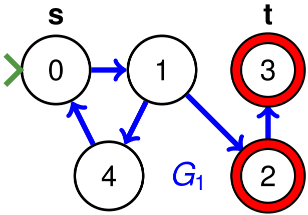
T/F: $G_1 \models \forall x \exists y\ E(x,y)$ **False**

T/F: $G_1 \models \exists x \forall y\ \sim E(x,y)$ **True**

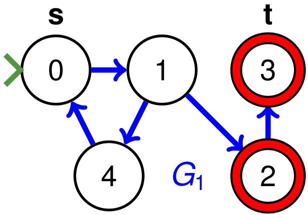
T/F: $G_1 \models \exists x\ E(x,x)$ **False**

T/F: $G_1 \models \forall x \sim E(x,x)$ **True**

T/F: $G_1 \models A(s) \wedge R(t)$



T/F: $G_1 \models \forall x \exists y\ E(x, y)$		**False**
T/F: $G_1 \models \exists x \forall y\ \sim E(x, y)$		**True**
T/F: $G_1 \models \exists x\ E(x, x)$		**False**
T/F: $G_1 \models \forall x \sim E(x, x)$		**True**
T/F: $G_1 \models A(s) \wedge R(t)$		**True**



iClicker 8.1 T/F: $G_1 \models \forall x \exists y\ E(y, x)$

A: True **B: False**

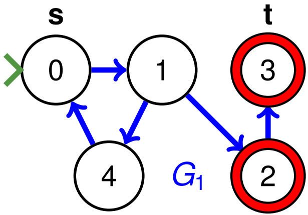
T/F: $G_1 \models \forall x \exists y\ E(x, y)$ **False**

T/F: $G_1 \models \exists x \forall y\ \sim E(x, y)$ **True**

T/F: $G_1 \models \exists x\ E(x, x)$ **False**

T/F: $G_1 \models \forall x \sim E(x, x)$ **True**

T/F: $G_1 \models A(s) \wedge R(t)$ **True**



iClicker 8.2 T/F: $G_1 \models \forall x \exists y\, (\sim R(x) \to E(x,y))$

A: True **B: False**

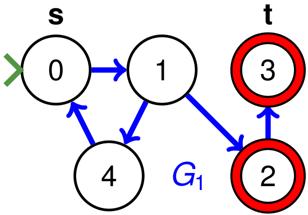
T/F: $G_1 \models \forall x \exists y\; E(x,y)$ **False**

T/F: $G_1 \models \exists x \forall y\; \sim E(x,y)$ **True**

T/F: $G_1 \models \exists x\; E(x,x)$ **False**

T/F: $G_1 \models \forall x \sim E(x,x)$ **True**

T/F: $G_1 \models A(s) \land R(t)$ **True**



iClicker 8.3 T/F: $G_1 \models \exists x\,(x = s \wedge A(x))$

A: True **B: False**

T/F: $G_1 \models \forall x \exists y\, E(x,y)$ **False**

T/F: $G_1 \models \exists x \forall y\, \sim E(x,y)$ **True**

T/F: $G_1 \models \exists x\, E(x,x)$ **False**

T/F: $G_1 \models \forall x \sim E(x,x)$ **True**

T/F: $G_1 \models A(s) \wedge R(t)$ **True**

Easy to identify symbols, from now on

vocabularies = $\Sigma, \Sigma_{\text{Tarski}}, \Sigma_{\text{garst}}, \ldots$

Easy to identify symbols, from now on

$$
\begin{aligned}
\textbf{vocabularies} &= \Sigma, \Sigma_{\text{Tarski}}, \Sigma_{\text{garst}}, \ldots \\
\textbf{PredCalc variables} &= x, y, z, u, v, w, x_1, y_1, z_1, \ldots
\end{aligned}
$$

Easy to identify symbols, from now on

vocabularies	=	$\Sigma, \Sigma_{\mathsf{Tarski}}, \Sigma_{\mathsf{garst}}, \ldots$
PredCalc variables	=	$x, y, z, u, v, w, x_1, y_1, z_1, \ldots$
PropCalc variables	=	$p, q, r, p_1, q_1, r_1, \ldots$

Easy to identify symbols, from now on

vocabularies	=	$\Sigma, \Sigma_{\text{Tarski}}, \Sigma_{\text{garst}}, \ldots$
PredCalc variables	=	$x, y, z, u, v, w, x_1, y_1, z_1, \ldots$
PropCalc variables	=	$p, q, r, p_1, q_1, r_1, \ldots$
constant symbols	=	$s, t, k, k_1, \ldots$

Easy to identify symbols, from now on

vocabularies	=	$\Sigma, \Sigma_{\mathrm{Tarski}}, \Sigma_{\mathrm{garst}}, \ldots$
PredCalc variables	=	$x, y, z, u, v, w, x_1, y_1, z_1, \ldots$
PropCalc variables	=	$p, q, r, p_1, q_1, r_1, \ldots$
constant symbols	=	$s, t, k, k_1, \ldots$
function symbols	=	$f, g, h, f_1, g_1, h_1, \ldots$

Easy to identify symbols, from now on

vocabularies	=	$\Sigma, \Sigma_{\text{Tarski}}, \Sigma_{\text{garst}}, \ldots$
PredCalc variables	=	$x, y, z, u, v, w, x_1, y_1, z_1, \ldots$
PropCalc variables	=	$p, q, r, p_1, q_1, r_1, \ldots$
constant symbols	=	$s, t, k, k_1, \ldots$
function symbols	=	$f, g, h, f_1, g_1, h_1, \ldots$
predicate symbols	=	$A, R, E, P, P_1, P_2, \ldots$

Easy to identify symbols, from now on

vocabularies	=	$\Sigma, \Sigma_{\text{Tarski}}, \Sigma_{\text{garst}}, \ldots$
PredCalc variables	=	$x, y, z, u, v, w, x_1, y_1, z_1, \ldots$
PropCalc variables	=	$p, q, r, p_1, q_1, r_1, \ldots$
constant symbols	=	$s, t, k, k_1, \ldots$
function symbols	=	$f, g, h, f_1, g_1, h_1, \ldots$
predicate symbols	=	$A, R, E, P, P_1, P_2, \ldots$
worlds	=	$W, S, T, G, H, W_1, S_1, T_1, G_1, H_1, \ldots$

Easy to identify symbols, from now on

vocabularies	=	$\Sigma, \Sigma_{\text{Tarski}}, \Sigma_{\text{garst}}, \ldots$
PredCalc variables	=	$x, y, z, u, v, w, x_1, y_1, z_1, \ldots$
PropCalc variables	=	$p, q, r, p_1, q_1, r_1, \ldots$
constant symbols	=	$s, t, k, k_1, \ldots$
function symbols	=	$f, g, h, f_1, g_1, h_1, \ldots$
predicate symbols	=	$A, R, E, P, P_1, P_2, \ldots$
worlds	=	$W, S, T, G, H, W_1, S_1, T_1, G_1, H_1, \ldots$
elements of $\vert\mathbf{W}\vert$	=	$a, b, c, d, e, a_1, b_1, \ldots$

Easy to identify symbols, from now on

vocabularies	=	$\Sigma, \Sigma_{\text{Tarski}}, \Sigma_{\text{garst}}, \ldots$
PredCalc variables	=	$x, y, z, u, v, w, x_1, y_1, z_1, \ldots$
PropCalc variables	=	$p, q, r, p_1, q_1, r_1, \ldots$
constant symbols	=	$s, t, k, k_1, \ldots$
function symbols	=	$f, g, h, f_1, g_1, h_1, \ldots$
predicate symbols	=	$A, R, E, P, P_1, P_2, \ldots$
worlds	=	$W, S, T, G, H, W_1, S_1, T_1, G_1, H_1, \ldots$
elements of |W|	=	$a, b, c, d, e, a_1, b_1, \ldots$
logical formulas	=	$\alpha, \quad \beta, \quad \gamma, \quad \varphi, \quad \psi, \ \ldots$

Easy to identify symbols, from now on

vocabularies	=	$\Sigma, \Sigma_{\text{Tarski}}, \Sigma_{\text{garst}}, \ldots$
PredCalc variables	=	$x, y, z, u, v, w, x_1, y_1, z_1, \ldots$
PropCalc variables	=	$p, q, r, p_1, q_1, r_1, \ldots$
constant symbols	=	$s, t, k, k_1, \ldots$
function symbols	=	$f, g, h, f_1, g_1, h_1, \ldots$
predicate symbols	=	$A, R, E, P, P_1, P_2, \ldots$
worlds	=	$W, S, T, G, H, W_1, S_1, T_1, G_1, H_1, \ldots$
elements of $\|W\|$	=	$a, b, c, d, e, a_1, b_1, \ldots$
logical formulas	=	$\alpha, \quad \beta, \quad \gamma, \quad \varphi, \quad \psi, \ \ldots$
	=	**alpha,beta,gamma,phi,psi,** ...

Abbreviations

$\hookrightarrow$ is an abbreviation for "is an abbreviation for"

Abbreviations

$\hookrightarrow$	is an abbreviation for	"is an abbreviation for"
$t_1 \neq t_2$	$\hookrightarrow$	$\sim(t_1 = t_2)$

Abbreviations

$\hookrightarrow$	is an abbreviation for	"is an abbreviation for"
$t_1 \neq t_2$	$\hookrightarrow$	$\sim(t_1 = t_2)$
$(\forall x \,.\, \alpha)\beta$	$\hookrightarrow$	$\forall x(\alpha \to \beta)$

Abbreviations

$\hookrightarrow$	is an abbreviation for	"is an abbreviation for"
$t_1 \neq t_2$	$\hookrightarrow$	$\sim(t_1 = t_2)$
$(\forall x \,.\, \alpha)\beta$	$\hookrightarrow$	$\forall x(\alpha \rightarrow \beta)$
$(\exists x \,.\, \alpha)\beta$	$\hookrightarrow$	$\exists x(\alpha \wedge \beta)$

Abbreviations

$\hookrightarrow$	is an abbreviation for	"is an abbreviation for"
$t_1 \neq t_2$	$\hookrightarrow$	$\sim(t_1 = t_2)$
$(\forall x \,.\, \alpha)\beta$	$\hookrightarrow$	$\forall x(\alpha \rightarrow \beta)$
$(\exists x \,.\, \alpha)\beta$	$\hookrightarrow$	$\exists x(\alpha \wedge \beta)$
$\exists ! x(\alpha(x))$	$\hookrightarrow$	$\exists x \forall y(\alpha(x) \wedge (\alpha(y) \rightarrow y = x))$

Precedence of of Logical Operators

1. $\forall, \exists, \sim$ bind the **tightest**

Precedence of of Logical Operators

1. $\forall, \exists, \sim$ bind the **tightest**
2. $\wedge, \vee, \oplus$

Precedence of of Logical Operators

1. $\forall, \exists, \sim$ bind the **tightest**
2. $\wedge, \vee, \oplus$
3. $\rightarrow, \leftrightarrow$

Precedence of of Logical Operators

1. $\forall, \exists, \sim$ bind the **tightest**
2. $\wedge, \vee, \oplus$
3. $\rightarrow, \leftrightarrow$

$\rightarrow$ binds **from right**: $\alpha \rightarrow \beta \rightarrow \gamma \equiv \alpha \rightarrow (\beta \rightarrow \gamma)$

Precedence of of Logical Operators

1. $\forall, \exists, \sim$ bind the **tightest**
2. $\wedge, \vee, \oplus$
3. $\rightarrow, \leftrightarrow$

$\rightarrow$ binds **from right**: $\alpha \rightarrow \beta \rightarrow \gamma \equiv \alpha \rightarrow (\beta \rightarrow \gamma)$

$$\alpha_1 \rightarrow \alpha_2 \rightarrow \alpha_3 \rightarrow \alpha_4 \quad \equiv \quad \alpha_1 \rightarrow (\alpha_2 \rightarrow (\alpha_3 \rightarrow \alpha_4))$$

if α_1 then (if α_2 then (if α_3 then α_4))

Precedence of of Logical Operators

1. $\forall, \exists, \sim$ bind the **tightest**
2. $\wedge, \vee, \oplus$
3. $\rightarrow, \leftrightarrow$

$\rightarrow$ binds **from right**: $\alpha \rightarrow \beta \rightarrow \gamma \equiv \alpha \rightarrow (\beta \rightarrow \gamma)$

$$\alpha_1 \rightarrow \alpha_2 \rightarrow \alpha_3 \rightarrow \alpha_4 \quad \equiv \quad \alpha_1 \rightarrow (\alpha_2 \rightarrow (\alpha_3 \rightarrow \alpha_4))$$

if α_1 then (if α_2 then (if α_3 then α_4))

$$\varphi \stackrel{\text{def}}{=} \forall x\, E(x, y) \rightarrow y = t$$

Precedence of of Logical Operators

1. $\forall, \exists, \sim$ bind the **tightest**
2. $\wedge, \vee, \oplus$
3. $\rightarrow, \leftrightarrow$

$\rightarrow$ binds **from right**: $\alpha \rightarrow \beta \rightarrow \gamma \;\equiv\; \alpha \rightarrow (\beta \rightarrow \gamma)$

$$\alpha_1 \rightarrow \alpha_2 \rightarrow \alpha_3 \rightarrow \alpha_4 \quad \equiv \quad \alpha_1 \rightarrow (\alpha_2 \rightarrow (\alpha_3 \rightarrow \alpha_4))$$

if α_1 then (if α_2 then (if α_3 then α_4))

$$\begin{aligned} \varphi &\stackrel{\text{def}}{=} \forall x\, E(x,y) \rightarrow y = t \\ &\equiv (\forall x\, E(x,y)) \rightarrow y = t \end{aligned}$$

Precedence of of Logical Operators

1. $\forall, \exists, \sim$ bind the **tightest**
2. $\wedge, \vee, \oplus$
3. $\rightarrow, \leftrightarrow$

$\rightarrow$ binds **from right**: $\alpha \rightarrow \beta \rightarrow \gamma \;\equiv\; \alpha \rightarrow (\beta \rightarrow \gamma)$

$$\alpha_1 \rightarrow \alpha_2 \rightarrow \alpha_3 \rightarrow \alpha_4 \quad \equiv \quad \alpha_1 \rightarrow (\alpha_2 \rightarrow (\alpha_3 \rightarrow \alpha_4))$$

if α_1 then (if α_2 then (if α_3 then α_4))

$$\begin{aligned} \varphi &\stackrel{\text{def}}{=} \forall x\, E(x,y) \rightarrow y = t \\ &\equiv (\forall x\, E(x,y)) \rightarrow y = t \end{aligned}$$

$$\psi \stackrel{\text{def}}{=} \forall x\, (E(x,y) \rightarrow y = t)$$

Precedence of of Logical Operators

1. $\forall, \exists, \sim$ bind the **tightest**
2. $\wedge, \vee, \oplus$
3. $\rightarrow, \leftrightarrow$

$\rightarrow$ binds **from right**: $\alpha \rightarrow \beta \rightarrow \gamma \;\equiv\; \alpha \rightarrow (\beta \rightarrow \gamma)$

$$\alpha_1 \rightarrow \alpha_2 \rightarrow \alpha_3 \rightarrow \alpha_4 \quad \equiv \quad \alpha_1 \rightarrow (\alpha_2 \rightarrow (\alpha_3 \rightarrow \alpha_4))$$

if α_1 then (if α_2 then (if α_3 then α_4))

$$\varphi \stackrel{\text{def}}{=} \forall x\, E(x, y) \rightarrow y = t$$
$$\equiv (\forall x\, E(x, y)) \rightarrow y = t$$

$$\psi \stackrel{\text{def}}{=} \forall x\, (E(x, y) \rightarrow y = t)$$

more about $\varphi \not\equiv \psi$ **in L9 when we define truth**

Negation of PredCalc Formulas

$\sim\forall x(f) \quad \equiv \quad \exists x(\sim f)$ **deMorgan** $\sim(p \wedge q) \equiv (\sim p \vee \sim q)$

Negation of PredCalc Formulas

$\sim\forall x(f)$	$\equiv$	$\exists x(\sim f)$	**deMorgan**	$\sim(p \wedge q) \equiv (\sim p \vee \sim q)$
$\sim\exists x(f)$	$\equiv$	$\forall x(\sim f)$	**deMorgan**	$\sim(p \vee q) \equiv (\sim p \wedge \sim q)$

Negation of PredCalc Formulas

$$\sim\forall x(f) \quad \equiv \quad \exists x(\sim f) \qquad \textbf{deMorgan} \qquad \sim(p \land q) \equiv (\sim p \lor \sim q)$$

$$\sim\exists x(f) \quad \equiv \quad \forall x(\sim f) \qquad \textbf{deMorgan} \qquad \sim(p \lor q) \equiv (\sim p \land \sim q)$$

$$\alpha \quad \stackrel{\text{def}}{=} \quad \forall x \exists y\ E(y, x)$$

Negation of PredCalc Formulas

$$\sim\forall x(f) \equiv \exists x(\sim f)$$ **deMorgan** $\sim(p \wedge q) \equiv (\sim p \vee \sim q)$

$$\sim\exists x(f) \equiv \forall x(\sim f)$$ **deMorgan** $\sim(p \vee q) \equiv (\sim p \wedge \sim q)$

$$\alpha \overset{\text{def}}{=} \forall x \exists y\, E(y, x)$$

$\alpha \equiv$ **"All vertices have an incoming edge."**

Negation of PredCalc Formulas

$\sim\forall x(f) \quad \equiv \quad \exists x(\sim f)$ **deMorgan** $\sim(p \wedge q) \equiv (\sim p \vee \sim q)$

$\sim\exists x(f) \quad \equiv \quad \forall x(\sim f)$ **deMorgan** $\sim(p \vee q) \equiv (\sim p \wedge \sim q)$

$$\alpha \stackrel{\text{def}}{=} \forall x \exists y\ E(y, x)$$

$\alpha \equiv$ **"All vertices have an incoming edge."**

$$\sim\alpha \equiv \exists x \forall y\ \sim E(y, x)$$

Negation of PredCalc Formulas

$$\sim\forall x(f) \equiv \exists x(\sim f) \qquad \textbf{deMorgan} \qquad \sim(p \wedge q) \equiv (\sim p \vee \sim q)$$

$$\sim\exists x(f) \equiv \forall x(\sim f) \qquad \textbf{deMorgan} \qquad \sim(p \vee q) \equiv (\sim p \wedge \sim q)$$

$$\alpha \stackrel{\text{def}}{=} \forall x \exists y\ E(y, x)$$

$\alpha \equiv$ **"All vertices have an incoming edge."**

$$\sim\alpha \equiv \exists x \forall y\ \sim E(y, x)$$

$\equiv$ **"Some vertex has no incoming edge."**

Negation of PredCalc Formulas

$$\sim\forall x(f) \quad\equiv\quad \exists x(\sim f)$$ **deMorgan** $\sim(p \wedge q) \equiv (\sim p \vee \sim q)$

$$\sim\exists x(f) \quad\equiv\quad \forall x(\sim f)$$ **deMorgan** $\sim(p \vee q) \equiv (\sim p \wedge \sim q)$

$$\alpha \stackrel{\text{def}}{=} \forall x \exists y\, E(y,x)$$

$\alpha \equiv$ **"All vertices have an incoming edge."**

$$\sim\alpha \equiv \exists x \forall y \;\sim E(y,x)$$

$\equiv$ **"Some vertex has no incoming edge."**

$$\beta \stackrel{\text{def}}{=} \forall xyz\, (E(x,y) \wedge E(x,z) \rightarrow y = z)$$

Negation of PredCalc Formulas

$$\sim\forall x(f) \equiv \exists x(\sim f) \qquad \textbf{deMorgan} \qquad \sim(p \wedge q) \equiv (\sim p \vee \sim q)$$

$$\sim\exists x(f) \equiv \forall x(\sim f) \qquad \textbf{deMorgan} \qquad \sim(p \vee q) \equiv (\sim p \wedge \sim q)$$

$$\alpha \stackrel{\text{def}}{=} \forall x \exists y\, E(y,x)$$

$\alpha \equiv$ **"All vertices have an incoming edge."**

$$\sim\alpha \equiv \exists x \forall y\ \sim E(y,x)$$

$\equiv$ **"Some vertex has no incoming edge."**

$$\beta \stackrel{\text{def}}{=} \forall xyz\, (E(x,y) \wedge E(x,z) \rightarrow y = z)$$

$\equiv$ **"Every vertex has at most one outgoing edge."**

Negation of PredCalc Formulas

$$\sim\forall x(f) \equiv \exists x(\sim f) \qquad \textbf{deMorgan} \qquad \sim(p \wedge q) \equiv (\sim p \vee \sim q)$$

$$\sim\exists x(f) \equiv \forall x(\sim f) \qquad \textbf{deMorgan} \qquad \sim(p \vee q) \equiv (\sim p \wedge \sim q)$$

$$\alpha \stackrel{\text{def}}{=} \forall x \exists y\ E(y, x)$$

$\alpha \equiv$ **"All vertices have an incoming edge."**

$$\sim\alpha \equiv \exists x \forall y\ \sim E(y, x)$$

$\equiv$ **"Some vertex has no incoming edge."**

$$\beta \stackrel{\text{def}}{=} \forall xyz\ (E(x, y) \wedge E(x, z) \rightarrow y = z)$$

$\equiv$ **"Every vertex has at most one outgoing edge."**

$$\sim\beta \equiv \exists xyz\ (E(x, y) \wedge E(x, z) \wedge y \neq z)$$

Negation of PredCalc Formulas

$$\sim\forall x(f) \equiv \exists x(\sim f)$$ **deMorgan** $\sim(p \wedge q) \equiv (\sim p \vee \sim q)$

$$\sim\exists x(f) \equiv \forall x(\sim f)$$ **deMorgan** $\sim(p \vee q) \equiv (\sim p \wedge \sim q)$

$$\alpha \stackrel{\text{def}}{=} \forall x \exists y\ E(y, x)$$

$\alpha \equiv$ **"All vertices have an incoming edge."**

$$\sim\alpha \equiv \exists x \forall y\ \sim E(y, x)$$

$\equiv$ **"Some vertex has no incoming edge."**

$$\beta \stackrel{\text{def}}{=} \forall xyz\ (E(x, y) \wedge E(x, z) \rightarrow y = z)$$

$\equiv$ **"Every vertex has at most one outgoing edge."**

$$\sim\beta \equiv \exists xyz\ (E(x, y) \wedge E(x, z) \wedge y \neq z)$$

$\equiv$ **"Some vertex has at least two outgoing edges."**

Negation Normal Form (NNF)

Usually **best** to write φ in **NNF**

Negation Normal Form (NNF)

Usually **best** to write φ in **NNF**

$$\varphi \stackrel{\text{def}}{=} \sim\forall x \exists y \, (\sim R(x) \rightarrow E(x, y))$$

Negation Normal Form (NNF)

Usually **best** to write φ in **NNF**

$$\varphi \stackrel{\text{def}}{=} \sim\forall x \exists y\ (\sim R(x) \rightarrow E(x, y))$$

As in PropCalc, first **remove all "$\rightarrow$"s**

Negation Normal Form (NNF)

Usually **best** to write φ in **NNF**

$$\varphi \stackrel{\text{def}}{=} \sim\forall x \exists y\ (\sim R(x) \to E(x, y))$$

As in PropCalc, first **remove all "$\to$"s**

$$\varphi \equiv \sim\forall x \exists y\ (R(x) \vee E(x, y))$$

Negation Normal Form (NNF)

Usually **best** to write φ in **NNF**

$$\varphi \stackrel{\text{def}}{=} \sim\forall x \exists y\, (\sim R(x) \rightarrow E(x,y))$$

As in PropCalc, first **remove all "$\rightarrow$"s**

$$\varphi \equiv \sim\forall x \exists y\, (R(x) \vee E(x,y))$$

Next, **push** the **"$\sim$"s** all the way **inside** using **de Morgan**

Negation Normal Form (NNF)

Usually **best** to write φ in **NNF**

$$\varphi \stackrel{\text{def}}{=} \sim\forall x \exists y\ (\sim R(x) \rightarrow E(x,y))$$

As in PropCalc, first **remove all "$\rightarrow$"s**

$$\varphi \equiv \sim\forall x \exists y\ (R(x) \vee E(x,y))$$

Next, **push** the **"$\sim$"s** all the way **inside** using **de Morgan**

$$\varphi \equiv \exists x \sim\exists y\ (R(x) \vee E(x,y))$$

Negation Normal Form (NNF)

Usually **best** to write φ in **NNF**

$$\varphi \stackrel{\text{def}}{=} \sim\forall x\exists y\,(\sim R(x) \to E(x,y))$$

As in PropCalc, first **remove all "$\to$"s**

$$\varphi \equiv \sim\forall x\exists y\,(R(x) \vee E(x,y))$$

Next, **push** the **"$\sim$"s** all the way **inside** using **de Morgan**

$$\begin{aligned}\varphi &\equiv \exists x \sim\exists y\,(R(x) \vee E(x,y)) \\ &\equiv \exists x\forall y\ \sim(R(x) \vee E(x,y))\end{aligned}$$

Negation Normal Form (NNF)

Usually **best** to write φ in **NNF**

$$\varphi \stackrel{\text{def}}{=} \sim\forall x \exists y\ (\sim R(x) \to E(x,y))$$

As in PropCalc, first **remove all "$\to$"s**

$$\varphi\ \equiv \sim\forall x \exists y\ (R(x) \vee E(x,y))$$

Next, **push** the **"$\sim$"s** all the way **inside** using **de Morgan**

$$\begin{aligned} \varphi &\equiv \exists x \sim\exists y\ (R(x) \vee E(x,y)) \\ &\equiv \exists x \forall y\ \sim(R(x) \vee E(x,y)) \\ &\equiv \exists x \forall y\ (\sim R(x) \wedge \sim E(x,y)) \end{aligned}$$

Answers to R8 Quiz

1. Negate “All dogs are loyal”

Answers to R8 Quiz

1. Negate "All dogs are loyal"

$$\alpha \stackrel{\text{def}}{=} \forall x \, (D(x) \rightarrow L(x))$$

Answers to R8 Quiz

1. Negate "All dogs are loyal"

$$\alpha \stackrel{\text{def}}{=} \forall x\,(D(x) \to L(x))$$

$$\sim\alpha \equiv \sim\forall x\,(D(x) \to L(x))$$

Answers to R8 Quiz

1. Negate "All dogs are loyal"

$$\begin{aligned} \alpha &\stackrel{\text{def}}{=} \forall x\ (D(x) \to L(x)) \\ \sim\alpha &\equiv \sim\forall x\ (D(x) \to L(x)) \\ &\equiv \sim\forall x\ (\sim D(x) \vee L(x)) \end{aligned}$$

Answers to R8 Quiz

1. Negate "All dogs are loyal"

$$\begin{aligned} \alpha &\stackrel{\text{def}}{=} \forall x\ (D(x) \rightarrow L(x)) \\ \sim\alpha &\equiv \sim\forall x\ (D(x) \rightarrow L(x)) \\ &\equiv \sim\forall x\ (\sim D(x) \vee L(x)) \\ &\equiv \exists x\ \sim(\sim D(x) \vee L(x)) \end{aligned}$$

Answers to R8 Quiz

1. Negate "All dogs are loyal"

$$\begin{aligned}
\alpha &\stackrel{\text{def}}{=} \forall x\ (D(x) \rightarrow L(x)) \\
\sim\alpha &\equiv \sim\forall x\ (D(x) \rightarrow L(x)) \\
&\equiv \sim\forall x\ (\sim D(x) \vee L(x)) \\
&\equiv \exists x\ \sim(\sim D(x) \vee L(x)) \\
&\equiv \exists x\ (D(x) \wedge \sim L(x))
\end{aligned}$$

Answers to R8 Quiz

1. Negate “All dogs are loyal”

$$\begin{aligned}
\alpha &\stackrel{\text{def}}{=} \forall x\,(D(x) \rightarrow L(x)) \\
\sim\alpha &\equiv \sim\forall x\,(D(x) \rightarrow L(x)) \\
&\equiv \sim\forall x\,(\sim D(x) \vee L(x)) \\
&\equiv \exists x\ \sim(\sim D(x) \vee L(x)) \\
&\equiv \exists x\,(D(x) \wedge \sim L(x))
\end{aligned}$$

Ans = “Some dogs are disloyal”

“There is a dog that is disloyal”

Answers to R8 Quiz

2. Negate "For all computer programs P
   (if P compiles without errors then P is correct)."

Answers to R8 Quiz

2. Negate "For all computer programs P
(if P compiles without errors then P is correct)."

$$\beta \stackrel{\text{def}}{=} \forall x\, (\sim E(x) \rightarrow C(x))$$

Answers to R8 Quiz

2. Negate "For all computer programs P
(if P compiles without errors then P is correct)."

$$\beta \stackrel{\text{def}}{=} \forall x\,(\sim E(x) \rightarrow C(x))$$

$$\sim\beta \equiv \sim\forall x\,(\sim E(x) \rightarrow C(x))$$

Answers to R8 Quiz

2. Negate "For all computer programs P (if P compiles without errors then P is correct)."

$$\beta \stackrel{\text{def}}{=} \forall x\ (\sim E(x) \rightarrow C(x))$$

$$\sim\beta \equiv \sim\forall x\ (\sim E(x) \rightarrow C(x))$$

$$\equiv \sim\forall x\ (E(x) \vee C(x))$$

Answers to R8 Quiz

2. Negate "For all computer programs P
(if P compiles without errors then P is correct)."

$$\begin{aligned}
\beta &\stackrel{\text{def}}{=} \forall x\ (\sim E(x) \rightarrow C(x)) \\
\sim\beta &\equiv \sim\forall x\ (\sim E(x) \rightarrow C(x)) \\
&\equiv \sim\forall x\ (E(x) \vee C(x)) \\
&\equiv \exists x\ \sim(E(x) \vee C(x))
\end{aligned}$$

Answers to R8 Quiz

2. Negate "For all computer programs P
(if P compiles without errors then P is correct)."

$$\begin{aligned}
\beta &\overset{\text{def}}{=} \forall x\ (\sim E(x) \to C(x)) \\
\sim\beta &\equiv \sim\forall x\ (\sim E(x) \to C(x)) \\
&\equiv \sim\forall x\ (E(x) \vee C(x)) \\
&\equiv \exists x\ \sim(E(x) \vee C(x)) \\
&\equiv \exists x\ (\sim E(x) \wedge \sim C(x))
\end{aligned}$$

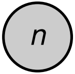
Answers to R8 Quiz

2. Negate "For all computer programs P
(if P compiles without errors then P is correct)."

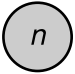
$$\begin{aligned} \beta &\stackrel{\text{def}}{=} \forall x\, (\sim E(x) \to C(x)) \\ \sim\beta &\equiv \sim\forall x\, (\sim E(x) \to C(x)) \\ &\equiv \sim\forall x\, (E(x) \vee C(x)) \\ &\equiv \exists x\ \sim(E(x) \vee C(x)) \\ &\equiv \exists x\, (\sim E(x) \wedge \sim C(x)) \end{aligned}$$

Ans = $\exists$ computer program *P*
(*P* compiles without errors but *P* is not correct)

"There is at least one incorrect computer program
which compiles without errors."

N_1 = 

1. $N_1 \models \forall x(\text{Triangle}(x) \rightarrow \text{Blue}(x))$ **True:** vacuously.
2. $N_1 \models \forall x(\text{Blue}(x) \rightarrow \text{Triangle}(x))$ **True:** vacuously.
3. $N_1 \models \forall x(\text{Square}(x) \rightarrow \exists y \text{LeftOf}(y, x))$ **True:** vacuously.
4. $N_1 \models \forall x(\text{Gray}(x) \rightarrow \text{Circle}(x))$ **True:** All elts. are Circles.
5. $N_1 \models \forall x(\text{Circle}(x) \rightarrow \text{Gray}(x))$ **True:** All Circles are Gray.
6. $N_1 \models \exists x \forall y(\text{LeftOf}(y, x) \vee y = x)$ **True:** there is only 1 elt.

$N_1 \quad = $ 

7. $N_1 \models \exists x \forall y (\text{Above}(y, x) \vee y = x)$
True: there is only 1 elt.

8. $N_1 \models \forall x \exists y (\text{Triangle}(x) \rightarrow \text{LeftOf}(x, y) \wedge \text{Circle}(y))$
True: vacuously.

9. $N_1 \models \forall x \exists y (\text{Triangle}(x) \rightarrow \sim\text{Above}(x, y) \wedge \sim\text{Above}(y, x) \wedge \text{Circle}(y))$
True: vacuously.

10. $N_1 \models \exists x \exists y (\text{Square}(x) \wedge \text{Circle}(y) \wedge \mathit{Above}(y, x) \wedge \text{Leftof}(x, y))$
False: there is no square.

Disc. 2 Answers

$R_1 =$ If I don't drink beer with dinner, then I always have fish.

$R_2 =$ Any time I have both beer and fish for dinner,
I do without ice cream.

$R_3 =$ If I have ice cream or don't have beer, then I never eat fish.

$b \equiv$ "I drink beer" $f \equiv$ "I have fish" $i \equiv$ "I have ice cream"

Translate each R_i into PropCalc:

$R_1 \equiv \sim b \rightarrow f$ $R_2 \equiv b \wedge f \rightarrow \sim i$ $R_3 \equiv i \vee \sim b \rightarrow \sim f$

Next, draw a truth table for these three rules.

W	b	f	i	$\sim b \;\rightarrow\; f$	$b \wedge f \;\rightarrow \sim i$	$i \vee \sim b \;\rightarrow \sim f$	S
W_7	1	1	1	1	0	0	0
W_6	1	1	0	1	1	1	1
W_5	1	0	1	1	1	1	1
W_4	1	0	0	1	1	1	1
W_3	0	1	1	1	1	0	0
W_2	0	1	0	1	1	0	0
W_1	0	0	1	0	1	1	0
W_0	0	0	0	0	1	1	0

Finally, using your truth table, try to come up with a simpler PropCalc formula equvalent to $S \stackrel{\text{def}}{=} R_1 \wedge R_2 \wedge R_3$.

$S \;\equiv\; b \wedge (\sim i \vee \sim f)$ "I always have beer, and I never have both fish and ice cream at the same meal."