- 4. Models and Truth
 - (a) The smallest possible domain that can make each formula true has 2 elements: $\mathcal{D} = \{one, two\}$. $A^{I} = one, B^{I} = two, C^{I} = two, P^{I} = \{one\}, Q^{I} = \{one\}$.
 - $\bullet \vDash_M P(A)$

iff $T_I(A) \in P^I$ iff $one \in \{one\}$, which is true.

 $\bullet \models_M \neg Q(B)$

iff
$$\nvDash_M Q(B)$$

iff $T_I(B) \notin Q^I$
iff $two \notin \{one\}$, which is true.

- $\bullet \models_M \neg Q(A) \lor \neg Q(C)$
 - iff $\nvDash_M Q(A)$ or $\nvDash_M Q(C)$ iff $T_I(A) \notin Q^I$ or $T_I(C) \notin Q^I$ iff $one \notin \{one\}$ or $two \notin \{one\}$, which is true.
- $\bullet \vDash_M \forall x. P(x) \Rightarrow Q(x)$

iff for all v.a.'s U and all $d \in \mathcal{D}$, $\vDash_M (P(x) \Rightarrow Q(x))[U_{x:d}]$ iff for all $d \in \mathcal{D}$, $\nvDash_M P(x)[U_{x:d}]$ or $\vDash_M Q(x)[U_{x:d}]$ iff for all $d \in \mathcal{D}$, $T_{IU_{x:d}}(x) \notin P^I$ or $T_{IU_{x:d}}(x) \in Q^I$ iff for all $d \in \mathcal{D}$, $d \notin \{one\}$ or $d \in \{one\}$, which is true.

- (b) The smallest possible domain that can make each formula false has 2 elements: $\mathcal{D} = \{one, two\}$. $A^I = one, B^I = one, C^I = one, P^I = \{two\}, Q^I = \{one\}$.
 - $\nvDash_{M'} P(A)$

iff $T_I(A) \notin P^I$ iff $one \notin \{two\}$, which is true.

 $\bullet \nvDash_{M'} \neg Q(B)$

iff $\vDash_{M'} Q(B)$ iff $T_I(B) \in Q^I$ iff $one \in \{one\}$, which is true.

• $\nvDash_{M'} \neg Q(A) \lor \neg Q(C)$

iff $\models_{M'} Q(A)$ and $\models_{M'} Q(C)$ iff $T_I(A) \in Q^I$ and $T_I(C) \in Q^I$ iff $one \in \{one\}$ and $one \in \{one\}$, which is true.

• $\nvDash_{M'} \forall x. P(x) \Rightarrow Q(x)$

iff for all v.a.'s U and all $d \in \mathcal{D}$, $\nvDash_{M'} (P(x) \Rightarrow Q(x))[U_{x:d}]$ iff for some $d \in \mathcal{D}$, $\vDash_{M'} P(x)[U_{x:d}]$ and $\nvDash_{M'} Q(x)[U_{x:d}]$ iff for some $d \in \mathcal{D}$, $T_{IU_{x:d}}(x) \in P^I$ and $T_{IU_{x:d}}(x) \notin Q^I$ iff for some $d \in \mathcal{D}$, $d \in \{two\}$ and $d \notin \{one\}$, which is true for d = two.

(c) One first-order language is defined by the vocabulary:

 $\Sigma = \{Circle, Triangle, Contained-Within, Left-Of, Right-Of, C1, C2, T1, T2\}$

A FOL description of the picture in this language might be:

 $\begin{aligned} Circle(C1) \wedge Circle(C2) \wedge Triangle(T1) \wedge Triangle(T2) \\ \wedge Contained-Within(T1,C1) \wedge Contained-Within(C2,T2) \\ \wedge Left-Of(C1,T2) \wedge Left-Of(T1,T2) \wedge Left-Of(C1,C2) \wedge Left-Of(T1,C2) \\ \wedge \forall x, y. \ Left-Of(x,y) \Rightarrow Right-Of(y,x) \end{aligned}$

- (d) $\vDash_M (\forall x \phi)[U]$
 - iff for all $d \in \mathcal{D}$, $\vDash_M \phi[U_{x:d}]$ (by satisfaction conds for ' \forall ') iff not for some $d \in \mathcal{D}$, $\nvDash_M \phi[U_{x:d}]$ iff not for some $d \in \mathcal{D}$, $\vDash_M \neg \phi[U_{x:d}]$ (by satisfaction conds for ' \neg ') iff not $\vDash_M (\exists x \neg \phi)[U]$ (by satisfaction conds for ' \exists ') iff $\vDash_M (\neg \exists x \neg \phi)[U]$ (by satisfaction conds for ' \neg ')

(e)
$$\vDash_M \neg (\phi \Rightarrow \psi)[U]$$

iff $\nvDash_M (\phi \Rightarrow \psi)[U]$ (by satisfaction conds for '¬') iff $\vDash_M \phi[U]$ and $\nvDash_M \psi[U]$ (by satisfaction conds for ' \Rightarrow ') iff $\vDash_M \phi[U]$ and $\vDash_M \neg \psi[U]$ (by satisfaction conds for '¬') iff $\vDash_M (\phi \land \neg \psi)[U]$ (by satisfaction conds for ' \land ')

(f)
$$\vDash_M (x = A \land P(x))[U]$$

iff $\vDash_M (x = A)[U]$ and $\vDash_M P(x)[U]$ (by satisfaction conds for ' \land ') iff $T_{IU}(x) = T_{IU}(A)$ and $T_{IU}(x) \in P^I$ (by satisfaction conds for equality/predicates) iff $T_{IU}(x) = T_{IU}(A)$ and $T_{IU}(A) \in P^I$ (substitution of equals) iff $\vDash_M (x = A)[U]$ and $\vDash_M P(A)[U]$ (by satisfaction conds for equality/predicates) iff $\vDash_M (x = A \land P(A))[U]$ (by satisfaction conds for ' \land ')

- 5. Validity and Entailment
 - (a) (i) Not valid. For a model M, $\vDash_M Thing(A)$ iff $A^I \in Thing^I$. Clearly, one can create a model where this is not true, such as one in which $Thing^I = \{\}$.
 - (ii) Valid. For a model M, $\vDash_M (Zod = Zod)$ iff $Zod^I = Zod^I$, which is clearly true regardless of which M is chosen.

- (iv) Valid. For a model $M, \vDash_M \forall x.Rose(x) \Rightarrow Rose(x)$ iff for all v.a.'s U and for all $d \in \mathcal{D}, \vDash_M (Rose(x) \Rightarrow Rose(x))[U_{x:d}]$, which is true iff for all $d \in \mathcal{D}, \nvDash_M Rose(x)[U_{x:d}]$ or $\vDash_M Rose(x)[U_{x:d}]$. Clearly this is true regardless of which M is chosen.
- (b) (i) This entailment holds. Suppose $\vDash_M \{P(A), A = B\}$. Then, by the truth conditions for each formula, we have $A^I \in P^I$ and $A^I = B^I$. Substituting equal terms, $B^I \in P^I$, therefore $\vDash_M P(B)$. Since every model of $\{P(A), A = B\}$ is a model of P(B), $\{P(A), A = B\} \vDash P(B)$.
 - (ii) This entailment holds. Suppose $\vDash_M \forall x P(x)$. Then, for all v.a.'s U and for all $d \in \mathcal{D}, \vDash_M P(x)[U_{x:d}]$. This is true iff for all $d \in \mathcal{D}, T_{IU_{x:d}}(x) \in P^I$, iff for all $d \in \mathcal{D}, d \in P^I$. Since $A^I \in \mathcal{D}$, this means that $A^I \in P^I$. Therefore, by the satisfaction condition for predicates, $\vDash_M P(A)$. Hence, $\forall x P(x) \vDash P(A)$.
 - (iii) This entailment holds. Suppose $\vDash_M P(A)$. Then, $A^I \in P^I$. It follows that for some $d \in \mathcal{D}$, $d \in P^I$. So for all v.a.'s U and for some $d \in \mathcal{D}$, $T_{IUx:d}(x) \in P^I$. Therefore, by the satisfaction condition for existential quantifiers, $\vDash_M \exists x P(x)$. Hence, $P(A) \vDash \exists x P(x)$.
 - (iv) This entailment holds. We have already proven in Problem 4, part (d) that $\vDash_M (\forall x \phi)[U]$ iff $\vDash_M (\neg \exists x \neg \phi)[U]$. Since this proof shows that any model M of the former must also be a model of the latter, it follows trivially that $\forall x \phi \vDash \neg \exists x \neg \phi$.

1. Semantics

- (a) (i) Valid.
 - For a model $M, \vDash_M P(A) \Rightarrow (Q(A) \Rightarrow P(A))$ iff $\nvDash_M P(A)$ or $\vDash_M Q(A) \Rightarrow P(A)$ [truth conditions of \Rightarrow]
 - This is true iff $\nvDash_M P(A)$ or $\nvDash_M Q(A)$ or $\vDash_M P(A)$ [truth conditions of \Rightarrow]
 - It's clearly true that either $\nvDash_M P(A)$ or $\vDash_M P(A)$ holds for M, and since M was arbitrarily chosen, this is true for any model M. Therefore, the formula is valid.
 - (ii) Contingent.
 - For a model M, $\vDash_M (\forall x P(x)) \lor (\forall x \neg P(x))$ iff $\vDash_M (\forall x P(x))$ or $\vDash_M (\forall x \neg P(x))$ [truth conditions of \lor]
 - This is true iff for all v.a.'s U and all $d \in \mathcal{D}$, $\vDash_M P(x)[U_{x:d}]$, or for all v.a.'s U and all $d \in \mathcal{D}$, $\nvDash_M P(x)[U_{x:d}]$ [truth conditions of \forall]
 - This is true iff for all $d \in \mathcal{D}$, $d \in P^I$, or for all $d \in \mathcal{D}$, $d \notin P^I$ [truth conditions of predicates]
 - One can construct a model M_T in which this is true, by setting $P^I = \mathcal{D}$. One can also construct a model M_F in which this is not true, for instance one in which $\mathcal{D} = \{one, two\}$ and $P^I = \{one\}$. Because there exists some models in which the formula is true, and some in which it is false, the formula is contingent.
 - (iii) Unsatisfiable.
 - For a model M, $\vDash_M \exists x \neg (x = x)$ iff for all v.a.'s U and for some $d \in \mathcal{D}$, $\vDash_M \neg (x = x)[U_{x:d}]$ [truth conditions of \exists]
 - This is true iff for all v.a.'s U and for some $d \in \mathcal{D}, \nvDash_M (x = x)[U_{x:d}]$ [truth conditions of \neg]
 - This is true iff for all v.a.'s U and for some $d \in \mathcal{D}$, $T_{IU_{x:d}}(x) \neq T_{IU_{x:d}}(x)$ [truth conditions of equality predicate]
 - This is true iff for all v.a.'s U and for some $d \in \mathcal{D}, d \neq d$ [application of T_{IU}]
 - Since \mathcal{D} is nonempty, let $a \in \mathcal{D}$ be some arbitrary individual from the domain. It is trivially true that a = a, so there exists

some $d \in \mathcal{D}$ such that d = d, hence $\vDash_M \exists x \neg (x = x)$ is false. Since M is arbitrary, the formula must be false in all models, therefore it is unsatisfiable.

(b) (i) • The BNF syntax of FOL for formulas would need to be extended with a rule for a predicate modifier being applied to a single predicate constant applied to a single term, as well as a rule for predicate modifiers. One possible syntax is the following:

 $\langle formula \rangle ::= (\langle predicate \ modifier \rangle \langle predicate \ constant \rangle (\langle term \rangle))$

 $\langle predicate \ modifier \rangle ::= Fake | White | Red | ...$

• The interpretation of the predicate modifier would be a function $f \in Pow(\mathcal{D}) \mapsto Pow(\mathcal{D})$. That is, a predicate modifier denotes some mapping from each possible subset of the domain (recall that the interpretation of predicates are domain subsets) to some new subset of the domain. For example, suppose we have:

 $\mathcal{D} = \{Snoopy, Tulip, Rose, SnoopyActionFigure, LegoTulip, PlasticRose\}$

with predicates $Dog^{I} = \{Snoopy\}$ and $Flower^{I} = \{Tulip, Rose\}$. Consider $Fake^{I}$. One possible interpretation of Fake would be a function f s.t. $f(\{Snoopy\}) = \{SnoopyActionFigure\}$ and $f(\{Tulip, Rose\}) = \{LegoTulip, PlasticRose\}$. Note that $Fake^{I}$ does have to be a total function according to the above definition, so all other elements of the powerset of \mathcal{D} could simply map to themselves.

- The satisfaction condition for predicate modifiers would be: for a model M and variable assignment $U, \vDash_M (\theta \pi(\tau))[U]$ iff $T_{IU}(\tau) \in \theta^I(\pi^I)$, where θ is some predicate modifier.
- (ii) The BNF syntax of FOL would need to be extended with the following rules. We omit the rule for <char*> in the following grammar, and simply assume that it can be any arbitrary sequence of characters, excluding ". Also <quote> is used in place of " due to bugs with the BNF Latex package I was using...

$$\langle term \rangle ::= \langle quote \rangle \langle char^* \rangle \langle quote \rangle$$

 $\langle term \rangle ::= \operatorname{concat}(\langle term \rangle, \langle term \rangle)$

 $\langle formula \rangle ::= \text{Substring}(\langle term \rangle, \langle term \rangle)$

• First, we need to extend \mathcal{D} to contain all possible strings. Let set \mathcal{S} represent the set of all possible strings (note that this

set necessarily includes all possible concatenations of strings within the set). Then, $S \subseteq D$.

The semantics of strings, and of functions/predicates applied to strings, are independent of I, so I does not need to be extended.

• We need to give an extension of T_{IU} for terms involving strings:

 $T_{IU}(\tau) = \tau$ if τ is a string

 $T_{IU}(concat(\tau_1, \tau_2)) = s$, where s is the concatenation of $T_{IU}(\tau_1)$ and $T_{IU}(\tau_2)$

We also need to extend satisfaction conditions for the Substring predicate:

For a model M and variable assignment U, $\vDash_M Substring(\sigma, \tau)[U]$ iff $T_{IU}(\sigma)$ is a substring of $T_{IU}(\tau)$

- 2. Soundness of inference rules
 - (a) Sound.
 - The inference rule is sound iff $\forall x.Thing(x) \vDash \exists x.Thing(x)$. That is, if for every model M such that $\vDash_M \forall x.Thing(x), \vDash_M \exists x.Thing(x)$.
 - Let M be some arbitrary model. $\vDash_M \forall x.Thing(x)$ iff for all v.a. U and all $d \in \mathcal{D}, \vDash_M Thing(x)[U_{x:d}]$ [truth conditions of \forall]
 - If this is true, however, then it follows that for all v.a. U and some $d \in \mathcal{D}, \vDash_M Thing(x)[U_{x:d}]$
 - This is true iff $\vDash_M \exists x. Thing(x)$ [truth conditions of \exists]
 - Therefore, for some arbitrary model M such that $\vDash_M \forall x.Thing(x)$, $\vDash_M \exists x.Thing(x)$. Hence, $\forall x.Thing(x) \vDash \exists x.Thing(x)$, so the inference rule is sound.
 - (b) Unsound.
 - The inference rule is sound iff $Dog(Snoopy) \vDash \exists x.Animal(x)$. To show that this is false, it suffices to find some model M such that $\vDash_M Dog(Snoopy)$ but $\nvDash_M \exists x.Animal(x)$.
 - One such model is $\mathcal{D} = \{one\}, Snoopy^I = one, Dog^I = \{one\}, Animal^I = \{\}.$
 - $\vDash_M Dog(Snoopy)$ iff $Snoopy^I \in Dog^I$, iff $one \in \{one\}$, [truth conditions of predicates], which is true.
 - $\nvDash_M \exists x.Animal(x)$ iff it is not the case that for all v.a.'s U and for some $d \in \mathcal{D}$, $Animal(x)[U_{x:d}]$ [truth conditions of \exists], iff it is not the case that for some $d \in \mathcal{D}$, $d \in Animal^I$, iff it is not the case that for some $d \in \mathcal{D}$, $d \in \{\}$, [truth conditions of predicates], which is true.

- Therefore, there exists a model M such that $\vDash_M Dog(Snoopy)$ but $\nvDash_M \exists x.Animal(x)$. Hence, $Dog(Snoopy) \nvDash \exists x.Animal(x)$, so the inference rule is unsound.
- 3. Deduction using several forward inference rules

1.	Loves(Juliet, Romeo)	Δ
2.	$(\forall x (\forall y (Loves(x, y) \lor Ignores(x, y))))$	Δ
3.	$(\forall x(Loves(x,Romeo) \Rightarrow \neg Ignores(Romeo,x)))$	Δ
4.	$(\forall y(Loves(Romeo, y) \lor Ignores(Romeo, y)))$	UI, 2
5.	$Loves(Juliet, Romeo) \Rightarrow \neg Ignores(Romeo, Juliet)$	UI, 3
6.	$Loves(Romeo, Juliet) \lor Ignores(Romeo, Juliet)$	UI, 4
7.	$\neg Ignores(Romeo, Juliet)$	MP, $1, 5$
8.	Loves(Romeo, Juliet)	MTP, 6, 7
9.	$(\exists x(Loves(Romeo, x)))$	EG, 8