There are about 75 minutes available, one per point. Keep that in mind as you choose the order in which to do the problems. The number of in brackets is the point value.

For the questions 1–10, we’ll use our standard universe of individuals where relevant:

\[ U = \{ \text{Bond, Loren, Pavarotti} \} \]

Suppose that there is a function \( f_3 \), which produces the following result:

\[
\begin{align*}
f_3(\text{true}) &= \text{false} \\
f_3(\text{false}) &= \text{true}
\end{align*}
\]

1. [2] What is the type of \( f_1 \)?
2. [2] What is the set that \( f_1 \) characterizes?
3. [4] Write a definition for \( f_1 \) in lambda notation:

\[
f_1 = \lambda p \ [ \ldots ]
\]

(fill in the “…” part)

Suppose that there is a function \( f_2 \), which produces the following result:

\[
\begin{align*}
f_2(\text{Bond}) &= \text{Bond} \\
f_2(\text{Loren}) &= \text{Loren} \\
f_2(\text{Pavarotti}) &= \text{Pavarotti}
\end{align*}
\]

4. [2] What is the type of \( f_2 \)?
5. [4] Write in prose what \( f_2 \) seems to return.
6. [4] Suppose our universe of individuals was instead \( U_2 = \{ \text{Bond, Loren, Pavarotti, Waterston} \} \). Evaluated in the universe \( U_2 \),

\[
f_2(\text{Waterston}) = \text{Waterston}
\]

Given this, write a definition for \( f_2 \) in lambda notation, one that will work in any universe we pick.

\[
f_2 = \lambda x \ [ \ldots ]
\]

(fill in the “…” part)
Suppose that we have a model $M=\langle U,V \rangle$ where $U$ is \{Bond, Loren, Pavarotti\} and $V$ is the following:

$V(Pavarotti) = \text{Pavarotti}$ \hspace{1cm} $V(\text{is hungry}) = \{\text{Pavarotti}\}$

$V(Bond) = \text{Bond}$ \hspace{1cm} $V(\text{is boring}) = \{\text{Loren, Pavarotti}\}$

$V(Loren) = \text{Loren}$ \hspace{1cm} $V(\text{is happy}) = \{\text{Loren, Bond}\}$

Suppose we have a function $f_3$ that defined as follows:

$$f_3 = \boxempty x \left[ \left[ P \left[ P(x) \right] \right] \right]$$

(fill in the “…” part)

where $P$ is a predicate—a function of type $\langle e,t \rangle$. Further, suppose that the only predicates we have are those functions that characterize $V(\text{is boring})$, $V(\text{is hungry})$, and $V(\text{is happy})$; call them $f_{\text{boring}}$, $f_{\text{hungry}}$, and $f_{\text{happy}}$. Thus:

$$f_{\text{happy}} = \boxempty x \left[ x \in V(\text{happy}) \right]$$

7. [2] What is the type of $f_{\text{happy}}$?

8. [2] What is the type of $f_3$?

9. [3] What is the extension (that is, the set characterized by) $f_3(\text{Pavarotti})$?

One of these things is not like the others. One of these things doesn’t belong. Which of a–d is not equivalent to the others?

It may help to think of $P$ and $Q$ as real predicates, such as \textit{is tall} or \textit{is intelligent}, so you could read $\boxempty x \left[ \left( P(x) \boxempty Q(x) \right) \right]$ as “there is an individual $x$ such that it is not the case that $x$ is tall and $x$ is intelligent” (or “someone is not both tall and intelligent”).

10. [6]

a. $\boxempty x \left[ \left( P(x) \boxempty Q(x) \right) \right]$

b. $\boxempty \boxempty x \left[ \left( \boxempty y[\boxempty P(y)](x) \boxempty Q(x) \right) \right]$

c. $\boxempty \boxempty x \left[ \left( P(x) \boxempty \boxempty z[ Q(z)](x) \right) \right]$

d. $\boxempty x \left[ \left( P(x) \boxempty \boxempty z[ Q(z)](x) \right) \right]$
Use the phrase structure rules, lexicon, transformation, semantics, and model given on the next page for questions 11–16. Note that we have added a new determiner not every (considered a single lexical item) and the interpretation rule for not every is not given.

11. [6] On the model of the interpretation rule for structures that have an NP every $N_C$ in them, write a quantifier rule for structures with not every $N_C$.
\[
\llbracket \text{S [not every } \square_i \text{ S}_1 \rrbracket \rrbracket^{M,g} = \ldots \]  
(fill in the “…” part)

12. [8] Draw the DS structure for this sentence using the syntax from the next page.
Not every monster is hungry.


\[
\llbracket \text{VP} \rrbracket^{M,g} = \ldots \]  
(fill in the “…” part)

\[
\llbracket \text{S}_2 \rrbracket^{M,g} = \ldots \]  
(fill in the “…” part)

\[
\llbracket \text{S}_1 \rrbracket^{M,g} = \ldots \]  
(fill in the “…” part)
The grammar for 10–15. This is not significantly different from the grammar on the handout, but I have omitted things that are irrelevant. And there’s no semantics defined for the determiner *not every*.

**Phrase structure rules**

\[ S \rightarrow NP \ VP \]
\[ VP \rightarrow V_i \]
\[ NP \rightarrow Det \ N \]

**Lexicon**

<table>
<thead>
<tr>
<th>Det</th>
<th>not every, every</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>monster</td>
</tr>
<tr>
<td>V_i</td>
<td>is hungry</td>
</tr>
</tbody>
</table>

**Transformation**

\[ [S \ X \ NP \ Y ] \rightarrow [S \ NP_i \ [S \ X \ e_i \ Y ] ] \]

*QR (Quantifier Raising)*

**Semantics** (all same as on the summary handout)

**Functional application:**

\[ \left[ [A \ B \ C] \right]^{M,g} = \left[ B \right]^{M,g} \left( \left[ C \right]^{M,g} \right) \quad \text{or} \quad \left[ C \right]^{M,g} \left( \left[ B \right]^{M,g} \right) \]

**Pass-up:**

\[ \left[ [A \ B] \right]^{M,g} = \left[ B \right]^{M,g} \text{ for } A, B \text{ of any category.} \]

**Trace/Pronoun Interpretation:**

\[ \left[ \Box_i \right]^{M,g} = g(i) \]

where \( \Box \) is a trace or a pronoun

**Lexical Interpretation:** Where \( A \) is a lexical category

\[ \left[ [A \ Box] \right]^{M,g} = V(\Box) \]

\[ \text{or} \quad = \Box x[ x \Box V(\Box) ] \]

where \( V(\Box) \) is an individual

\[ \text{where } V(\Box) \text{ is a set of individuals} \]

**Quantifier Interpretation:**

**Every:**

\[ \left[ [S \ \text{every} \ Box_i \ S_1] \right]^{M,g} = \Box u \Box U \quad \left[ \Box \right]^{M,g(u)} \left[ S_1 \right]^{M,g[u/i]} \]

The model \( M = \langle U_4, V_4 \rangle \), where \( U_4 = \{Elmo, Oscar, Grover\} \), and \( V_4 \) is as below:

\[ V_4(Elmo) = Elmo \]
\[ V_4(Elmo) = Elmo \]
\[ V_4(Oscar) = Oscar \]
\[ V_4(Grover) = Grover \]

\[ V_4(is \ hungry) = \{Oscar\} \]
\[ V_4(monster) = \{Elmo, Oscar, Grover\} \]

The assignment function \( g \) assigns all indices to Elmo.