Instructions
1. Please fill in your answers on the exam sheets (5 pages).
2. Exam duration: 2.5 hours
3. You may use any pre-prepared material.
4. Please write your student number here: ____________.

Good luck!

Question 1  (5+5+3+7+5=25 points)
Consider the following sentences.
(1.1) John is mayor (of Utrecht).
(1.2) John was mayor (of Utrecht).

Remark: the addition “in Utrecht” does not matter for our analysis, and is only for the sake of clarification.

Obviously, there is no entailment between (1.1) and (1.2). In order to capture this, we treat grammatical tense (is/was) as indicating time in possible world semantics.
To do that, we assume a function time that maps every index \( i \in D_s \) to a real number. The times of the indices in \( D_s \) introduce an order between them. Thus, if \( \text{time}(i_1) < \text{time}(i_2) \) then the time of the index \( i_1 \) is earlier than that of the index \( i_2 \).

For the words John, mayor is and was in (1.1) and (1.2) we assume the following types and denotations:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>e</td>
</tr>
<tr>
<td>mayor</td>
<td>e(st)</td>
</tr>
<tr>
<td>is</td>
<td>((e(st))(e(st)))</td>
</tr>
<tr>
<td>was</td>
<td>((e(st))(e(st)))</td>
</tr>
</tbody>
</table>

\( IS = \lambda P_e(st).P \)
\( WAS = \lambda P_e(st).\lambda x.\lambda i.\exists j[\text{time}(j) < \text{time}(i) \land P(x)(j)] \)

a. Simplify as much as possible the following formulas for (1.1) and (1.2), respectively:

\[ IS(\text{mayor})(\text{john}) \]
\[ WAS(\text{mayor})(\text{john}) \]

b. Assume a model \( M \) where \( D_s = \{i_1, i_2\} \), \( \text{time}(i_1) = 1 \), \( \text{time}(i_2) = 2 \) and \( \text{mayor(}\text{john}) = \{i_1\} \). Write down the denotations of sentences (1.1) and (1.2) in \( M \):

\[ [(1.1)]^M = \]
\[ [(1.2)]^M = \]
c. Explain briefly how the denotations that you showed in your answer to b account for the lack of entailment (in both directions) between sentences (1.1) and (1.2):


d. Consider now the following sentences.
(1.3) John is former mayor (of Utrecht).
(1.4) John was mayor (of Utrecht) and John is not mayor (of Utrecht).
Assuming that sentence (1.3) is equivalent to (1.4), suggest a type and a meaning for the adjective former in (1.3).

<table>
<thead>
<tr>
<th>type former:</th>
</tr>
</thead>
<tbody>
<tr>
<td>denotation former: FORMER =</td>
</tr>
</tbody>
</table>

e. Using your answer to d, simplify as much as possible the following formula for (1.3):

\[ \text{IS(FORMER(mayor))(john)} \]

Make sure that the result is equivalent to our treatment of (1.4).

**Question 2** (4+6+5+5+6+6=32 points)
Consider the following sentences, where \( V_1 \) and \( V_2 \) stand for verbs.

(2.1) All students who \( V_1 V_2 \).
(2.2) All students who \( V_2 V_1 \).

For instance, when \( V_1=\text{dance} \) and \( V_2=\text{smile} \) we get:
sentence (2.1) = all students who dance smile;
sentence (2.2) = all students who smile dance.

a. Write down two verbs for \( V_1 \) and \( V_2 \), for which sentence (2.1) is a tautology, but (2.2) is not.

\( V_1 = \) _______ \( V_2 = \) _______

b. Write down two other examples for pairs like \( V_1 \) and \( V_2 \).

pair 1: _______ _______

pair 2: _______ _______

c. Using only the words students, who, entities, only, are and the two verbs \( V_1 \) and \( V_2 \) from your answer to a, form a sentence (2.3) that is equivalent to (2.1). Assume that the noun entities denotes the function characterizing the whole domain \( D_e \) of entities.

(2.3) _______

d. Reconsider the two verbs \( V_1 \) and \( V_2 \) from your answer to a. Using the set denotations \([V_1]\) (for the verb \( V_1 \)), \([V_2]\) (for the verb \( V_2 \)) and \( S \) (for the noun students), and the set theoretical operations \( \cap \) (intersection) and \( \subseteq \) (set inclusion), write down the (identical)
truth-value of sentences (2.1) and (2.3).

e. Reconsider the two verbs $V_1$ and $V_2$ from your answer to a, as appearing in the following (non-)entailments, where $D_1$, $D_2$, and $D_3$ are determiner expressions.

(2.4) $D_1$ student(s) who $V_1$ smiled $\Rightarrow$ $D_1$ student(s) who $V_2$ smiled

(2.5) $D_2$ student(s) who $V_2$ smiled $\Rightarrow$ $D_2$ student(s) who $V_1$ smiled

(2.6) $D_3$ student(s) who $V_1$ smiled $\nRightarrow$ $D_3$ student(s) who $V_2$ smiled;

$D_3$ student(s) who $V_2$ smiled $\nRightarrow$ $D_3$ student(s) who $V_1$ smiled

Write down examples for the determiner expressions in (2.4)-(2.6) that satisfy these non-entailments:

$D_1 =$ ________________  $D_2 =$ ________________  $D_3 =$ ________________

f. Answer the following questions:

Which property of $D_1$ does entailment (2.4) illustrate?

Which property of $D_2$ does entailment (2.5) illustrate?

Which property of $D_3$ does entailment (2.6) illustrate?

**Question 3** (3+3+13=19 points)

Consider the following sentences, with the assumed binary structures:

(3.0) John [[showed Mary] Fido].
(3.1) John [[showed [every student]] his dog].

We analyze the verb *show* as being of type $e(e(et))$, denoting a (Curried char. function) of a trinary relation between entities.

a. Write down the (most simplified) truth-value denotation of sentence (3.0), using the denotations *show* of type $e(e(et))$ and *john*, *mary* and *fido* (all three of type $e$). You must use the assumed structure in (3.0).

For the analysis of (3.1), we define the following $Z$ operator:

$$ Z = \lambda R_{e(e(et))} \cdot \lambda Q_{e(et)} \cdot \lambda f_{e^e} \cdot \lambda x_{e^e} \cdot Q(\lambda y_{e^e} . R(y)(f(y))(x)) $$

Using this operator we analyze sentence (3.1) as follows:

(3.2) $Z(\text{showed}_{e(e(et))})(\lambda A_{et} \cdot \text{student}_{et} \subseteq A)(\text{his}_{e^e} \cdot \text{dog}_{e^e})(\text{john}_{e^e})$

b. Consider the following four statements in (i)-(iv). Formula (3.2) represents the following paraphrase of sentence (3.1) –
(i) There is some masculine entity $x$, and John showed every student the dog that belongs to $x$.

(ii) John showed every student the dog that belongs to John.

(iii) For every student $x$, John showed $x$ the dog that belongs to $x$.

(iv) No one of the statements above.

Mark the most appropriate statement among (i)-(iv).

c. To support your answer to $a$, write down the most simplified form of formula (3.2).

\[
\text{Remark: You are requested not to write your simplification steps on the exam sheet.}
\]

**Question 4** (6+6+6+6=24 points)

For this question, refer to the lexicon on page 5. Suppose we have a predefined function

\[
\text{height :: E -> Int}
\]

that takes an entity and returns an integer which represents the entity's length in centimeters.

1. Use height to define a function taller :: E -> E -> T that takes two entities and returns True if the second has a larger or equal height than the first and False otherwise.

\[
taller :: E \rightarrow E \rightarrow T
\]

2. Add a lexicon entry for taller such that the following sentences can be parsed with the given lexicon.

1) "Everyone is taller than Yoda"
2) "Chewbacca is taller than everyone"
3) "No_one is taller than Chewbacca"  
   ( @ is short for the SAT combinator )

, entry "taller" (______________) taller

3. Give a denotation for the adjective tall in terms of taller, such that the following entailments hold for all $x$ of type $e$ and all $F$ of type $et$:

1) $x$ is a tall $F \Rightarrow x$ is a $F$
2) $x$ is a tall $F \Rightarrow x$ is taller than most $F$

You may use any of the functions that are present in the lexicon in your definition.

\[
tall_adj ::
\]

\[
tall_adj _________________
\]

4. Add a lexicon entry for taller such that the following sentences can be parsed with the given lexicon.

4) "Vader is_a tall boy"
5) "Leia is_a tall girl"

, entry "tall" (______________) tall_adj
-- charf takes a set of entities and returns its characteristic function
charf :: (Eq a) => [a] -> a -> T
charf = \set \rightarrow \ \x \rightarrow \ x 'elem' set

-- toList take a char. funtion and return the set it characterizes
toList :: (E->T) \rightarrow \ [E]
toList f = \ filter f (domain f)

{-- card : takes a function f and returns the

cardinality of the set that f characterizes --}
card :: (E -> Bool) \rightarrow \ Int
card f = \ length (toList f)

sat :: (E->E->T) \rightarrow \ ((E->T)->T) \rightarrow \ (E \rightarrow \ T)
sat r q y = q (\x \rightarrow \ r x y)

{---- Denotations of GQ and DET’s---------}
every f g = f .<. g
some f g = \ exists (f /\ g)
most f g = \ card (f /\ g) > card (f /\ (compl g))
everyone f = \forall f

-- some abbreviations for common syntactic categories
n = N \hspace{1cm} -- nouns
s = S \hspace{1cm} -- sentences
np = NP \hspace{1cm} -- noun phrases
iv = np :\ s \hspace{1cm} -- intransitive verbs
tv = iv :/ np \hspace{1cm} -- transitive verbs
det = s :/ iv :/ n \hspace{1cm} -- determiners
gq = s:/ (np:s) \hspace{1cm} -- generalized quantifiers

lexicon = Lexicon
{=== Entities ===-}
 , entry "Leia" np Leia
 , entry "Chewbacca" np Chewbacca
 , entry "Yoda" np Yoda
 , entry "Vader" np Vader
 , entry "@" (iv:/gq:/tv) sat
 , entry "is_a" ((np:\s):/n) ( (\x \rightarrow x) :: (E->T) \rightarrow (E->T))
 , entry "is" ((np:\s):/n) ( (\x \rightarrow x) :: (E->T) \rightarrow (E->T))
 , entry "is" (iv:/iv) ( (\x \rightarrow x) :: (E->T) \rightarrow (E->T))
 , entry "than" (tv:/tv) ( (\x \rightarrow x) :: (E->E->T) \rightarrow (E->E->T) )
 , entry "boy" n (charf [Luke,Yoda,Chewbacca,Vader])
 , entry "girl" n (charf [Leia])
 , entry "alien" n (charf [Chewbacca,Yoda])
 , entry "every" det everyone
 , entry "most" det most
 , entry "everyone" gq everyone
 , entry "no_one" gq (\f \rightarrow \ card f == 0) ]