Master Exam Version V3M

First Name: ________________________________________________

Last Name: ________________________________________________

Immatriculation Number: ____________________________________

Course of Studies (please mark exactly one):

- Informatik Bachelor
- Informatik Master
- SSE Master
- Other: ________________________________

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Maximal Points</th>
<th>Achieved Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercise 1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Exercise 2</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Exercise 3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Exercise 4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Exercise 5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Exercise 6</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>Grade</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Instructions:

- On every sheet please give your first name, last name, and immatriculation number.
- You must solve the exam without consulting any extra documents (e.g., course notes).
- Make sure your answers are readable. Do not use red or green pens or pencils.
- Please answer the exercises on the exercise sheets. If needed, also use the back sides of the exercise sheets.
- Answers on extra sheets can only be accepted if they are clearly marked with your name, your immatriculation number, and the exercise number.
- Cross out text that should not be considered in the evaluation.
- Students that try to cheat do not pass the exam.
- At the end of the exam, please return all sheets together with the exercise sheets.
Exercise 1 (Theoretical Foundations): (5 + 3 = 8 points)

Let $\varphi = p(s^2(0), 0) \land \forall X(p(s^2(X), X) \rightarrow p(s^4(X), s^2(X))) \land \neg p(s^3(0), s(0))$ and $\psi = \exists Y p(s^6(0), Y)$ be formulas over the signature $(\Sigma, \Delta)$ with $\Sigma = \Sigma_0 \cup \Sigma_1$, $\Sigma_0 = \{0\}$, $\Sigma_1 = \{s\}$, and $\Delta = \Delta_2 = \{p\}$. Here, $s^2(0)$ stands for $s(s(0))$, etc.

a) Prove that $\{\varphi\} \models \psi$ by means of SLD resolution.

Hints: First transform the formula $\varphi \land \neg \psi$ into an equivalent clause set.

b) Explicitly give a Herbrand model of the formula $\varphi$ (i.e., specify a carrier and a meaning for all function and predicate symbols). You do not have to provide a proof for your answer.
Exercise 2 (Procedural Semantics, SLD tree):  
(7 + 7 + 2 = 16 points)

Consider the following Prolog program \( \mathcal{P} \) which can be used to replace the letter sequence 'ba' by 'zz':

\[
\begin{align*}
\text{replace}([], []) & . \\
\text{replace}([b,a|XS], [z,z|YS]) & : - \text{replace}(XS, YS) . \\
\text{replace}([X|XS], [X|YS]) & : - \text{replace}(XS, YS) .
\end{align*}
\]

For example, the query \(?- \text{replace}([b,a,b,a], Z)\) would give the answer substitution \(Z = [z,z,z,z]\). Due to backtracking it is also possible to leave (parts of) the word unchanged. Because of that the answer substitutions \(Z = [b,a,z,z]\), \(Z = [z,z,b,a]\), and \(Z = [b,a,b,a]\) are also possible.

a) Consider the following query:

\(?- \text{replace}([a,b,b,a], Res)\).

For the logic program \( \mathcal{P} \) please show a successful computation for the query above (i.e., a computation of the form \((G, \emptyset) \vdash_\mathcal{P} (\emptyset, \sigma)\) where \( G = \{ - \text{replace}([a,b,b,a], Res) \})\). It suffices to give substitutions only for those variables which are used to define the value of the variable \( \text{Res} \) in the query.
b) Please give a graphical representation of the SLD tree for the query

?- replace([a,b,b,a], Res).

in the program \( \mathcal{P} \).

c) Modify the program \( \mathcal{P} \) by inserting a single cut. No other modification is allowed. Your modified program must replace all occurrences of 'ba' by 'zz'.

For example, now the query ?- replace([b,a,b,a], Z) must have the only answer substitution \( Z = [z,z,z,z] \).
Exercise 3 (Fixpoint Semantics):  
(4 + 3 + 3 = 10 points)

Consider the following logic program $\mathcal{P}$ over the signature $(\Sigma, \Delta)$ with $\Sigma = \{a, q\}$ and $\Delta = \{p\}$.

\[
\begin{align*}
p(a, a, Z) &. \\
p(q(Y), q(X), Z) &:- p(X, Y, Z).
\end{align*}
\]

a) For each $n \in \mathbb{N}$ explicitly give $\text{trans}_n^\mathcal{P}(\emptyset)$ in closed form, i.e., using a non-recursive definition.

b) Compute the set $\text{lfp}(\text{trans}_n^\mathcal{P})$.

c) Give $F[\mathcal{P}, \{\neg p(X, Y, Z)\}]$. 
Exercise 4 (Universality): (10 points)

Consider a function $f : \mathbb{N}^{n+1} \to \mathbb{N}$. The function $g : \mathbb{N}^n \to \mathbb{N}$ is defined as:

$$g(k_1, \ldots, k_n) = k \text{ iff } f(k_1 + k, \ldots, k_n + k, k) = 0 \text{ and}$$

for all $0 \leq k' < k$ we have $f(k_1 + k', \ldots, k_n + k', k')$ is defined and $f(k_1 + k', \ldots, k_n + k', k') > 0$

As an example, consider the function $\hat{f} : \mathbb{N}^2 \to \mathbb{N}$ with $\hat{f}(x, y) = \max\{x - 4y, 0\}$. The function $\hat{g} : \mathbb{N} \to \mathbb{N}$, constructed as described above, computes $\hat{g}(6) = 2$. The reason is that for $x = 6$, 2 is the smallest $y$ such that $\hat{f}(x + y, y) = 0$. Indeed, $\hat{f}(6+0, 0) = \hat{f}(6, 0) = 6, \hat{f}(6+1, 1) = \hat{f}(7, 1) = 3, \hat{f}(6+2, 2) = \hat{f}(8, 2) = 0$.

Consider a definite logic program $\mathcal{P}$ which computes the function $f$ using a predicate symbol $\overline{f} \in \Delta^{n+2}$:

$$f(k_1, \ldots, k_{n+1}) = k \text{ iff } \mathcal{P} \models \overline{f}(k_1, \ldots, k_{n+1}, k').$$

Here, numbers are represented by terms built from $0 \in \Sigma_0, s \in \Sigma_1$ (i.e., $0 = 0, 1 = s(0), 2 = s(s(0)), \ldots$).

Please extend the definite logic program $\mathcal{P}$ such that it also computes the function $g$ using the predicate symbol $\overline{g} \in \Delta^{n+1}$ (but without the cut or any other built-in predicate):

$$g(k_1, \ldots, k_n) = k \text{ iff } \mathcal{P} \models \overline{g}(k_1, \ldots, k_n, k).$$
Exercise 5 (Definite Logic Programming): (12 points)

Implement the predicate `noDupl/2` in Prolog. This predicate can be used to identify numbers in a list that appear exactly once, i.e., numbers which are no duplicates. The first argument of `noDupl` is the list to analyze. The second argument is the list of numbers which are no duplicates, as described below.

As an example, for the list `[2, 0, 3, 2, 1]` the result `[0, 3, 1]` is computed (because 2 is a duplicate). In Prolog the corresponding call `noDupl([s(s(0)), 0, s(s(s(0))), s(s(0)), s(0)], Res)` gives the answer substitution `Res = [0, s(s(s(0))), s(0)]`.

In your implementation you may (only) use the following two predefined predicates:

- `contained(X, XS)` is true if and only if the list `XS` contains `X`.
- `notContained(X, XS)` is true if and only if the list `XS` does not contain `X`.

**Important:** You may not use the cut or any other predefined predicates in your implementation! However, you may implement auxiliary predicates.
Exercise 6 (Arithmetic): (4 points)

Tetration is the logical extension of multiplication and exponentiation:

- **multiplication** \( a \cdot n := \underbrace{a + a + \cdots + a}_n \)
- **exponentiation** \( a^n := \underbrace{a \cdot a \cdot \cdots \cdot a}_n \)
- **tetration** \( a \uparrow^n n := \underbrace{a \cdot (a \cdot (a \cdots a))}_n \)

Examples:

- \( 4 \uparrow 2 = 4^4 = 256 \)
- \( 1 \uparrow 3 = 1^{(1^1)} = 1^1 = 1 \)
- \( 2 \uparrow 4 = 2^{(2^{2^2})} = 2^{(2^4)} = 2^{16} = 65,536 \)

Implement the predicate `tetration/3` in Prolog. For numbers \( x > 0, y > 0 \) the call `tetration(x, y, Z)` gives the answer substitution \( Z = m \) where \( m \) is \( x \uparrow y \).

As an example, `tetration(2, 4, Z)` gives the answer substitution \( Z = 65536 \).

Your predicate only needs to work on input values \( x > 0, y > 0 \), i.e., for other input values the result of the computation is irrelevant.

**Hint:** To compute \( x^y \) in Prolog you can use \( x**y \).