Prof. Dr. Jürgen Giesl Carsten Otto

Master Exam Version V3M

First Name: ____________________

Last Name: ____________________

Immatriculation Number: ____________________

Course of Studies (please mark exactly one):

- ○ **Informatik Bachelor**
- ○ **Informatik Master**
- ○ **SSE Master**
- ○ **Other:** ____________________

	Maximal Points	Achieved Points
Exercise 1	8	
Exercise 2	16	
Exercise 3	10	
Exercise 4	10	
Exercise 5	12	
Exercise 6	4	
Total	60	
Grade	-	

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

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

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

b) Explicitly give a Herbrand model of the formula φ (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':

```
replace([],        []).
replace([b,a|XS], [z,z|YS]) :- replace(XS, YS).
replace([X|XS],   [X|YS])   :- replace(XS, YS).
```

For example, the query `?- 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:

```
?- 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 $(G, \varnothing) \vdash_{\mathcal{P}}^{+} (\Box, \sigma)$ where $G = \{\neg$`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 `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 ocurrences 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 (Σ, Δ) with $\Sigma = \{\mathtt{a}, \mathtt{q}\}$ and $\Delta = \{\mathtt{p}\}$.

```
p(a, a, Z).
p(q(Y), q(X), Z) :- p(X, Y, Z).
```

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

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

c) Give $F[\![\mathcal{P}, \{\neg \mathtt{p(X,\ Y,\ Z)}\}]\!]$.

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:

$$\begin{aligned} g(k_1, \ldots, k_n) = k \text{ iff } & f(k_1 + k, \ldots, k_n + k, k) = 0 \text{ and} \\ & \text{for all } 0 \le k' < k \text{ we have } f(k_1 + k', \ldots, k_n + k', k') \text{ is defined and} \\ & f(k_1 + k', \ldots, k_n + k', k') > 0 \end{aligned}$$

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, $\hat{f}(6{+}\mathbf{0}, \mathbf{0}) = \hat{f}(6, 0) = \mathbf{6}$, $\hat{f}(6{+}\mathbf{1}, \mathbf{1}) = \hat{f}(7, 1) = \mathbf{3}$, $\hat{f}(6{+}\mathbf{2}, \mathbf{2}) = \hat{f}(8, 2) = \mathbf{0}$.

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

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

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

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

$$g(k_1, \ldots, k_n) = k \text{ iff } \mathcal{P} \models \underline{\texttt{g}}(\underline{k_1}, \ldots, \underline{k_n}, \underline{k}).$$

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	$a \cdot n$	$:=$	$\underbrace{a + a + \cdots + a}_{n}$
exponentation	a^n	$:=$	$\underbrace{a \cdot a \cdot \ldots \cdot a}_{n}$
tetration	$a \upuparrows n$	$:=$	$\underbrace{a^{\left(a^{\cdot^{\cdot^{(a^a)}}}\right)}}_{n}$

Examples:

- $4 \upuparrows 2 = 4^4 = 256$
- $1 \upuparrows 3 = 1^{(1^1)} = 1^1 = 1$
- $2 \upuparrows 4 = 2^{\left(2^{(2^2)}\right)} = 2^{(2^4)} = 2^{16} = 65.536$

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 $x \upuparrows y$.
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`.