Prof. Dr. Jürgen Giesl

Jera Hensel

Bachelor/Master Exam

First Name: ______________________________

Last Name: ______________________________

Immatriculation Number: ______________________________

Course of Studies (please mark exactly one):

- ○ **Informatik Bachelor**
- ○ **Mathematik Master**
- ○ **Other:** ______________________
- ○ **Informatik Master**
- ○ **SSE Master**

	Maximal Points	Achieved Points
Exercise 1	8	
Exercise 2	10	
Exercise 3	11	
Exercise 4	9	
Exercise 5	9	
Exercise 6	7	
Exercise 7	6	
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**.
- You can solve the exercises in **English** or **German**.
- 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 $\varphi = \exists X\, \mathtt{p}(X) \wedge \forall Y \Big(\mathtt{p}(Y) \to \big(\, \mathtt{p}(\mathtt{s}(\mathtt{s}(Y))) \wedge \neg \mathtt{p}(\mathtt{s}(Y))\,\big)\Big)$ and $\psi = \exists Z\, \mathtt{p}(\mathtt{s}(Z))$ be formulas over the signature (Σ, Δ) with $\Sigma = \Sigma_0 \cup \Sigma_1, \Sigma_0 = \{\mathtt{0}\}, \Sigma_1 = \{\mathtt{s}\}$, and $\Delta = \Delta_1 = \{\mathtt{p}\}$.

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

Hint: First transform the formula $\varphi \wedge \neg\psi$ into a clause set that is satisfiable iff $\varphi \wedge \neg\psi$ is satisfiable.

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): (4 + 6 = 10 points)

Consider the following Prolog program $\mathcal{P}$ which deletes all numbers greater than 4 from a list.

```
filter([],[]).
filter([X|XS],[X|YS]) :- X =< 4, !, filter(XS,YS).
filter([X|XS],YS) :- filter(XS,YS).
```

a) The program $\mathcal{P}'$ results from $\mathcal{P}$ by **removing the cut**. Consider the following query:

```
?- filter([2,5,2],[X]).
```

For the logic program $\mathcal{P}'$ (i.e., **without the cut**), show a successful computation for the query above (i.e., a computation of the form $(G, \varnothing) \vdash^{+}_{\mathcal{P}'} (\Box, \sigma)$ where $G = \{\neg\texttt{filter([2,5,2],[X])}\}$). You may leave out the negations in the queries and you do not have to show how the substitutions operate on variables that only occur in (renamed) program clauses.

b) Please give a graphical representation of the SLD tree for the query

```
?- filter([4,5],Res).
```

in the program $\mathcal{P}$ (i.e., **with the cut**). For every part of a tree that is cut off by evaluating `!`, please indicate the cut by crossing out the edge that leads to the cut-off part. For the cut-off parts only indicate the first cut-off goal, but do not evaluate further. Please also indicate all answer substitutions.

Exercise 3 (Definite Logic Programming): (11 points)

We consider the following problem: An object is sold by auction. The auctioneer has a list of the incoming bids and wants to sell the object to the bidder with the highest offer. If there is more than one highest bid, then the first of these bidders gets the object at the price he or she offers. However, if the second highest bid is strictly lower than the highest bid, the winner of the auction only has to pay the price of the second highest bid plus one.

Implement a predicate `auction/3` in Prolog. If the first argument is a list of bids constructed using the function `bid`(x, n) for a name x and a price $n \in \mathbb{N}$, the predicate should return the name of the winner of the auction as its second argument and the price at which the object is sold as its third argument.

For example, the query

```
?- auction([bid(paul,1),bid(lea,7),bid(paul,4)], X, N)
```

should succeed with the only answer `X = lea, N = 5`. The query

```
?- auction([bid(paul,1),bid(lea,7),bid(paul,7)], X, N)
```

should succeed with the only answer `X = lea, N = 7`.

Hints:

- First implement a predicate `findAndDeleteMax/3` that finds the highest bid or, if there is no unique highest bid, the first of the highest bids, and returns this bid as its second argument and the bid list where this bid is removed as its third argument.
- Afterwards, use the predicate `findAndDeleteMax/3` to find the two highest bids.
- You may use the usual arithmetic operators like `+`, `*`, `-`, `/`, and `min/2`, which computes the minimum of two numbers. So for example, `?- N is min(2,1)` succeeds with the answer `N = 1`.
- Moreover, you may use the built-in predicates `is/2`, `</2`, `=</2`, `>/2`, `>=/2`, and the cut.
- You may assume that there are at least two bids in the list.

Exercise 4 (Fixpoint Semantics): **(5 + 2 + 2 = 9 points)**

Consider the following logic program $\mathcal{P}$ over the signature (Σ, Δ) with $\Sigma = \{\mathtt{0}, \mathtt{s}\}$ and $\Delta = \{\mathtt{p}\}$.

```
p(X, 0, s(0)).
p(X, s(Y), s(Z)) :- p(s(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 $F[\![\mathcal{P}, \{\neg\mathtt{p(s(X),\ s(s(0)),\ Z)}\}]\!]$.

Exercise 5 (Meta-Programming): (9 points)

Implement a predicate `vars/2` which computes a list of all variables occurring in the term given as its first argument. If a variable occurs more than once in the given term, the list may contain this variable also more than once. For example, the query `?- vars(p(X,q(a,X,Z)),R)` should succeed with the answer `R = [Z,X,X]` (or any permutation of this list).
You **may only use** the built-in predicates `var/1`, `atomic/1`, `compound/1`, `=../2`, and `append/3`, which appends the lists given as its first and its second argument. So `?- append([1,2],[3],R)` succeeds with the answer `R = [1,2,3]`.

Exercise 6 (Universality): **(7 points)**

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

$$\begin{aligned} g(k_1, \ldots, k_n) = k \text{ iff } & f(k_1, \ldots, k_n, k) = 0 \text{ and} \\ & \text{for all } 0 \leq i < k, \text{ we have that } f(k_1, \ldots, k_n, i) \text{ is defined and} \\ & \text{there exists } 0 \leq k' < k \text{ such that } f(k_1, \ldots, k_n, k') = 0 \text{ and} \\ & \text{for all } 0 \leq k'' < k, k'' \neq k', \text{ we have } f(k_1, \ldots, k_n, k'') > 0 \end{aligned}$$

As an example, consider the function $\hat{f} : \mathbb{N}^2 \to \mathbb{N}$ with $\hat{f}(x, k) = x + k^2 - 3k$. The function $\hat{g} : \mathbb{N} \to \mathbb{N}$, constructed as described above, computes $\hat{g}(2) = 2$. The reason is that for $x = 2$, 2 is the smallest k such that $\hat{f}(x, k) = 0$ and such that there exists $0 \leq k' < k$ with $\hat{f}(x, k') = 0$ (i.e., 2 is the second smallest k such that $\hat{f}(x, k) = 0$). Indeed, $\hat{f}(2, \mathbf{0}) = \mathbf{2}$, $\hat{f}(2, \mathbf{1}) = \mathbf{0}$, $\hat{f}(2, \mathbf{2}) = \mathbf{0}$.

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

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

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

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

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

Exercise 7 (Programming with CLP): (6 points)

In this task, we use Prolog to find sequences of numbers that are either strictly descending or “gradually ascending”. Here, a sequence is represented by a list in Prolog.
Implement a Prolog predicate `seq/1` such that `seq(`s`)` is true if and only if there is an $m \geq 1$ such that $s = [n_1, \dots n_m]$ is a list of pairwise distinct integers between 1 and m where for two adjacent numbers n_i and n_{i+1}, $1 \leq i < m$, we have that $n_i - 1 \leq n_{i+1} \leq n_i + 2$. For example, the query `?- seq([2,X,Y,4,Z])` should succeed with the unique substitution `X = 1, Y = 3, Z = 5`.

The following line is already given:

```
:- use_module(library(clpfd)).
```

Hint:

- You may use the built-in predicates `length/2` and `all_distinct/1`.