

## 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	11	
Exercise 2	14	
Exercise 3	10	
Exercise 4	10	
Exercise 5	10	
Exercise 6	5	
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**.

Name:

Immatriculation Number:

**Exercise 1 (Theoretical Foundations):**
**(3 + 4 + 4 = 11 points)**

Let  $\varphi = p(0, s(0), s(s(0))) \wedge \forall X, Y, Z (p(X, Y, Z) \rightarrow p(Y, Z, s(s(X))))$  and  $\psi = \exists A p(A, s(A), s(A))$  be formulas over the signature  $(\Sigma, \Delta)$  with  $\Sigma = \Sigma_0 \cup \Sigma_1$ ,  $\Sigma_0 = \{0\}$ ,  $\Sigma_1 = \{s\}$ , and  $\Delta = \Delta_3 = \{p\}$ .

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

*Hint: First transform the formula  $\varphi \wedge \neg\psi$  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.

c) Let  $\varphi_1, \dots, \varphi_k, \psi' \in \mathcal{F}(\Sigma', \Delta', \mathcal{V})$  for some signature  $(\Sigma', \Delta')$ . Prove that  $\{\varphi_1, \dots, \varphi_k\} \models \psi'$  holds if and only if  $\varphi_1 \wedge \dots \wedge \varphi_k \wedge \neg\psi'$  is unsatisfiable.

Name:

Immatriculation Number:

**Exercise 2 (Procedural Semantics, SLD tree):**
**(7 + 7 = 14 points)**

Consider the following Prolog program  $\mathcal{P}$  which can be used to replace the letter 'a' by the letter 'ä' in words:

```
transform([], []).
transform([a|XS], [ä|YS]) :- !, transform(XS, YS).
transform([X|XS], [X|YS]) :- transform(XS, YS).
```

As an example, the query `transform([b,a,n,a,n,a], Z)` would give the answer substitution  $Z = [b, ä, n, ä, n, ä]$ .

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

```
?- transform([a,n,a], [a|XS]).
```

For the logic program  $\mathcal{P}'$ , i.e. **without the cut**, please show a successful computation for the query above (i.e., a computation of the form  $(G, \emptyset) \vdash_{\mathcal{P}'}^+ (\square, \sigma)$  where  $G = \{\neg \text{transform}([a,n,a], [a|XS])\}$ ). It suffices to give substitutions only for those variables which are used to define the value of the variable XS in the query.

Name:

Immatriculation Number:

---

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

?- transform([a,n,a], XS).

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 marking the corresponding edge. For the cut-off parts only indicate the first cut-off goal, but do not evaluate further.

Name:

Immatriculation Number:

**Exercise 3 (Fixpoint Semantics):**
**(4 + 3 + 3 = 10 points)**

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

 $p(a, q(Y)).$ 
 $p(q(X), q(Y)) \text{ :- } p(X, Y).$ 

- a) For each  $n \in \mathbb{N}$  explicitly give  $\text{trans}_{\mathcal{P}}^n(\emptyset)$  in closed form, i.e., using a non-recursive definition.
- b) Compute the set  $\text{lfp}(\text{trans}_{\mathcal{P}})$ .
- c) Give  $F[\mathcal{P}, \{\neg p(q(q(a)), X)\}]$ .

Name:

Immatriculation Number:

**Exercise 4 (Universality):**
**(10 points)**

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

$$g(k_1, \dots, k_n) = k \text{ iff } f(k_1, \dots, k_n, k) = n \text{ and} \\ \text{for all } 0 \leq k' < k \text{ we have } f(k_1, \dots, k_n, k') \text{ is defined and } f(k_1, \dots, k_n, k') \neq n$$

As an example, consider the function  $\hat{f} : \mathbb{N}^3 \rightarrow \mathbb{N}$  with  $\hat{f}(x, y, z) = x + y - 2z$ . The function  $\hat{g} : \mathbb{N}^2 \rightarrow \mathbb{N}$ , constructed as described above, computes  $\hat{g}(3, 5) = 3$ . The reason is that for  $x = 3, y = 5$ , 3 is the smallest  $z$  so that  $\hat{f}(x, y, z) = 2$ . Indeed,  $\hat{f}(3, 5, \mathbf{0}) = \mathbf{8}$ ,  $\hat{f}(3, 5, \mathbf{1}) = \mathbf{6}$ ,  $\hat{f}(3, 5, \mathbf{2}) = \mathbf{4}$ ,  $\hat{f}(3, 5, \mathbf{3}) = \mathbf{2}$ .

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

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

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

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

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

Name:

Immatriculation Number:

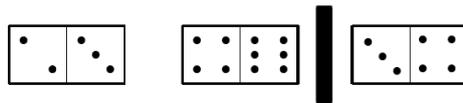
**Exercise 5 (Definite Logic Programming):**

**(10 points)**

Implement the predicate `addDomino/2` in Prolog. This predicate can be used in the setting of the domino game to check if a given piece can be inserted in an already existing series of domino pieces.

Every domino piece consists of two numbers, a left number and a right number. A list of pieces is called *valid* only if for each domino piece  $\alpha$  which is directly followed by another domino piece  $\beta$ , the right number of  $\alpha$  is identical to the left number of  $\beta$ .

In the following picture you can see that the first two pieces do not form a valid list because the left number of the second piece is 4 instead of 3. However, when adding the third piece after the first piece and before the second piece, the resulting list is valid: the left number 3 of the added piece matches the right number of the first piece, and its right number 4 matches the left number of the second piece.



In Prolog we represent a single domino piece using the predicate `p/2`. The list containing the first two pieces of the picture above is represented as `[p(2,3), p(4,6)]`. The predicate `addDomino/2` is true if and only if the piece in the second argument can be inserted between two pieces of the list in the first argument (or at the end) so that the resulting list of pieces is valid. Following the example above, `addDomino([p(2,3),p(4,6)], p(3,4))` should be `true` because adding the piece `p(3,4)` after `p(2,3)` is possible.

**Important:** You may not use predefined predicates!

**Hints:**

- Pieces cannot be rotated. Because of this, `addDomino([p(2,3),p(4,6)], p(4,3))` is `false`.
- You only need to check if the piece can be added. You do not have to compute the list that results from adding the piece.
- To check if a list `XS` of pieces is valid you can check if `addDomino(XS, p(Z,Z))` for a fresh variable `Z` is `true`.
- Every list containing at most one domino piece is valid.



**Name:**

**Immatriculation Number:**

---

Name: \_\_\_\_\_

Immatriculation Number: \_\_\_\_\_

**Exercise 6 (Arithmetic):****(5 points)**

Consider the following sequence of numbers:

$$5 \xrightarrow{-1} 4 \xrightarrow{+2} 6 \xrightarrow{-3} 3 \xrightarrow{+4} 7 \xrightarrow{-5} 2 \xrightarrow{+6} 8 \xrightarrow{-7} 1 \xrightarrow{+8} 9 \xrightarrow{-9} \dots$$

This sequence results from alternating the subtraction and addition of a number which is incremented in each step. When starting with 5 and first subtracting 1, the result of this is the sequence shown above. Implement the predicate `nth/2` in Prolog. For a number  $n > 0$  a call of `nth(n, X)` gives the answer substitution  $X = m$  where  $m$  is the  $n$ th number in the sequence described above. Here, we use 1 to describe the first element, so `nth(1, X)` gives the answer substitution  $X = 5$ .

As another example, `nth(5, X)` gives the answer substitution  $X = 7$ .