Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Lehrstuhl für Programmiersprachen und Übersetzerkonstruktion

Institut für Informatik

Sandra Dylus, Priv.-Doz. Dr. Frank Huch, Niels Bunkenburg

10. Klausur zur Vorlesung „Advanced Programming (pre-masters)"

WS 18/19

You can obtain 28 points within two assignments. To pass the test, you have to reach at least 14 points.

You have 90 minutes to complete the test. It is not allowed to use any material other than a pen. Electronic devices have to be turned off.

Hold your student card nearby, we will check it during the examination.

You will be informed about the results on Wednesday, March 27th (1:00 to 2:00 PM) in room number 715 in CAP 4.

Aufgabe 1 - Questions and Multiple Choice 12 Punkte

Answer the following questions directly on this sheet of paper.

Questions

1. (1P) What is the result for the query `?- rem(X,[1,2,3]).`?

```
rem(X, [X|_]) :- X\=1, !.
rem(X, [_|L]) :- rem(X,L).
```

2. (2P) Give the most general unifier (if it exists) for the terms $t_1 =$ `append(p(A),[A|[p(A)|[]]])` and $t_2 =$ `append(X,[f(1)|Xs])`. Otherwise, explain why no MGU exists.

3. (1.5P) Give all solutions of the goal `?- append(Xs,[2,3|Ys],[2,3,2,3,2,1]).`.

For the next five questions, simply mark all correct answers with an X. **If you want to change your answer after marking a statement, fill the original square and draw a new one, as shown in the example.** Note that any number of answers can be correct. Each question is worth 1.5 points. For each incorrect answer 0.5 points are deducted; negative scores are not possible.

Example

1. Which names are associated with Advanced Programming?

   a) ☒ Sandra

   b) ☐ Peter

   c) ☒ Frank

   d) ■ ☐ Rebecca

   e) ☒ Niels

Multiple Choice

1. Let σ be an MGU for the terms t_1 and t_2. Which of the following statements hold?

   a) ☐ $\sigma(t_2) = (\sigma \circ \sigma)(t_1)$

   b) ☐ $\sigma(t_2) = t_1$

   c) ☐ $\sigma(t_2) = \sigma(t_1)$

   d) ☐ $t_2 \neq t_1$

   e) ☐ $\sigma(t_2) = \sigma(\sigma(t_1))$

2. Which of the following statements about the unification algorithm are true?

   a) ☐ The occurs check avoids non-termination.

   b) ☐ The algorithm never terminates.

   c) ☐ The algorithm does only terminate if no MGU exists.

   d) ☐ The algorithm returns an MGU if and only if the terms are unifiable.

   e) ☐ Unification is used in Haskell's type inference algorithm.

3. Which of the following queries are answered with `true` or a binding for the occuring variables?

   a) ☐ `?- 7 * 6 is 42.`

   b) ☐ `?- 1 + 7 is 1 + 7.`

   c) ☐ `?- 3 + 1 = 5 - 1.`

   d) ☐ `?- 3 * 4 = Y.`

   e) ☐ `?- 7 is X.`

4. Which of the following sentences is correct?

   a) ☐ `?- findall(Xs,append([2],Ys,[1,2]),L).` has no solution.

   b) ☐ σ is a most general unifier if there exists a unifier σ' and a substitution ϕ such that $\sigma' = \phi \circ \sigma$.

   c) ☐ Prolog uses SLD resolution to compute the results of a query.

   d) ☐ $ds(f(Y, g(Y)), f(A, true)) = \{(Y, A)\}$

   e) ☐ `?- X = f(X).` has no solution in swi-Prolog.

5. Which of the following (list) patterns unify with the expression `[true,false]`?

a) ☐ `X`

b) ☐ `[true | X]`

c) ☐ `[true | [X]]`

d) ☐ `[X | [false]]`

e) ☐ `[true, false | []]`

Aufgabe 2 - Programming in Prolog 16 Punkte

In this excercise we use Peano numbers. The structure should be clear from the following predicate, which can be used to check whether a given value is a Peano number.

```
isPeano(o).
isPeano(s(N)) :- isPeano(N).
```

1. (1P) Define a predicate `even(P)`, which checks whether a given Peano number is even.
2. (3P) Define a predicate `maxList(Ps,Max)`, which computes the maximum `Max` for a list of Peano numbers `Ps`. You can use the predicate `max(X,Y,Max)`, which you do not need to implement.
3. (4P) Define a predicate `drop(P,Xs,Ys)` that checks whether the `Xs` without the first `P` elements is identical with the list `Ys`. Note that `P` is a Peano number.
4. (8P) We consider the following Prolog program

   ```
   app([],Ys,Ys).
   app([X|Xs],Ys,[X|Zs]) :- app(Xs,Ys,Zs).
   ```

   and the goal `?- app(Xs,Ys,[1,2]), Ys = [1|Zs].`

   Construct the SLD-Resolution-Tree for this goal.