Prof. Dr. Jürgen Giesl Florian Frohn

Notes:

- Please solve these exercises in **groups of two**!
- The solutions must be handed in **directly before (very latest: at the beginning of)** the exercise course on **Friday, December 11th, 2015**, in lecture hall **AH 1**. Alternatively you can drop your solutions into a box which is located right next to Prof. Giesl's office (until the exercise course starts).
- Please write the **names** and **immatriculation numbers** of all (two) students on your solution. Please staple the individual sheets!
- Exercises or exercise parts marked with a star are voluntary **challenge** exercises with advanced difficulty. However, they do not contribute to the overall number of points that will be required for the exam qualification or for the Übungsschein, respectively (i.e., you can obtain **bonus points** for such exercises).

Exercise 1 (Termination Proofs with Advanced Simplification Orders): (3 + 3 = 6 points)

Please prove termination of the following TRSs using RPO or RPOS. In this exercise, x, xs, y, ys, n, and m denote variables while all other identifiers are function symbols. For each TRS, explicitly state your chosen simplification order (i.e., give its name, the used precedence, and the used status where appropriate). Use the proof tree notation introduced on the last exercise sheet to prove $t_1 \succ t_2$ for two terms t_1 and t_2 and $\succ \in \{\succ_{rpo}, \succ_{rpos}\}$. You can again refer to the different cases in the definition of the respective order by their numbers. However, in case $t_1 \succ_{emb} t_2$, you may omit the proof tree for this pair of terms. To illustrate these notations (especially for multiset comparisons), we give the full solution for the following example from the lecture:

$$\begin{aligned} \mathsf{plus}(\mathcal{O}, y) &\to y \\ \mathsf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathsf{plus}(y, x)) \end{aligned}$$

We choose the RPO with precedence $\mathsf{plus} \sqsupset \mathsf{s}$.

$$\dfrac{}{\mathsf{plus}(\mathcal{O},y) \succ_{rpo} y}\,\text{emb} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\mathsf{s}(x)\succ_{rpo} x}\,\text{emb}}{\{\mathsf{s}(x),y\}\ (\succ_{rpo})_{mul}\ \{y,x\}}\,\text{mul}}{\mathsf{plus}(\mathsf{s}(x),y)\succ_{rpo}\mathsf{plus}(y,x)}\,3}{\mathsf{plus}(\mathsf{s}(x),y)\succ_{rpo}\mathsf{s}(\mathsf{plus}(y,x))}\,2$$

a)

$$\begin{aligned} \mathsf{u}(\mathsf{Cons}(x, xs), ys) &\to \mathsf{Cons}(x, \mathsf{u}(ys, xs)) \\ \mathsf{Cons}(x, \mathsf{i}(ys, xs)) &\to \mathsf{i}(\mathsf{Cons}(x, xs), \mathsf{Cons}(x, ys)) \end{aligned}$$

Here, "u" and "i" stand for "union" and "intersection" of sets.

b)

$$\mathsf{f}(\mathsf{ack}(\mathsf{s}(n), \mathsf{s}(m)), \mathsf{s}(y)) \to \mathsf{f}(y, \mathsf{ack}(n, \mathsf{ack}(\mathsf{s}(n), m)))$$

Exercise 2 (Reduction Orders): (3 + 3 = 6 points)

In this exercise, we will prove termination using so-called *polynomial orders*. In a (linear) polynomial order, one uses a polynomial interpretation $\mathcal{P}$ that maps each function symbol f of arity n to a polynomial $f_{\mathcal{P}}(x_1, \dots, x_n) = c_0 + c_1 x_1 + \dots + c_n x_n$ using the variables $x_1, \dots, x_n$ and coefficients $c_0, \dots, c_n$ from $\mathbb{N}$. Such an interpretation for function symbols can then be extended to terms using the following rules:

- $\mathcal{P}(x) := x$ for all variables x.
- $\mathcal{P}(f(t_1, \dots, t_n)) := f_{\mathcal{P}}(\mathcal{P}(t_1), \dots, \mathcal{P}(t_n))$ for all terms $f(t_1, \dots, t_n)$.

As example, consider the term $t = \mathsf{minus}(\mathsf{s}(x), \mathsf{s}(y))$. We choose $\mathsf{s}_{\mathcal{P}}(x_1) = 1 + x_1$ and $\mathsf{minus}_{\mathcal{P}}(x_1, x_2) = 1 + x_1 + x_2$. Then we have $\mathcal{P}(\mathsf{s}(x)) = 1 + x$ and thus $\mathcal{P}(t) = 1 + (1 + x) + (1 + y) = 3 + x + y$.
Using $\mathcal{P}$, we can then define the polynomial order over $\mathcal{T}(\Sigma, \mathcal{V})$ such that $s \succ_{\mathcal{P}} t$ holds if and only if $\mathcal{P}(s) > \mathcal{P}(t)$ holds for all variable assignments with values from $\mathbb{N}$. As example, consider the rule

$$\mathsf{minus}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{minus}(x, y)$$

and our interpretation from above. We have $\mathsf{minus}(\mathsf{s}(x), \mathsf{s}(y)) \succ_{\mathcal{P}} \mathsf{minus}(x, y)$ as $\mathcal{P}(\mathsf{minus}(\mathsf{s}(x), \mathsf{s}(y))) = 3 + x + y > 1 + x + y = \mathcal{P}(\mathsf{minus}(x, y))$ holds for all natural numbers x, y.
Of course, to use $\succ_{\mathcal{P}}$ as a reduction order, we need to ensure that it is well founded, stable, and monotonic. To this end, one requires that in $f_{\mathcal{P}}(x_1, \dots, x_n) = c_0 + c_1 x_1 + \dots + c_n x_n$, $c_i > 0$ holds for all $1 \le i \le n$. However, $c_0 = 0$ is allowed.

a) Show termination of the following TRS $\mathcal{R}_1$ using a polynomial interpretation $\mathcal{P}_1$. In this exercise part, x and y denote variables while all other identifiers are function symbols:

$$\begin{aligned} \mathsf{plus}(\mathcal{O}, y) &\to y \\ \mathsf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathsf{plus}(x, y)) \\ \mathsf{plus}(\mathsf{s}(x), y) &\to \mathsf{plus}(x, \mathsf{s}(y)) \end{aligned}$$

Give a polynomial $f_{\mathcal{P}_1}$ for each symbol f from Σ and show that $l \succ_{\mathcal{P}_1} r$ holds for all $l \to r \in \mathcal{R}_1$.

Hints:

- You do not need coefficients that are greater than 2.

b) Show termination of the following TRS $\mathcal{R}_2$ using a polynomial interpretation $\mathcal{P}_2$. In this exercise part, x denotes a variables while all other identifiers are function symbols:

$$\begin{aligned} \mathsf{f}(\mathsf{s}(\mathsf{s}(x)), \mathsf{b}) &\to \mathsf{f}(x, \mathsf{a}) \\ \mathsf{f}(x, \mathsf{a}) &\to \mathsf{f}(\mathsf{s}(x), \mathsf{b}) \end{aligned}$$

Give a polynomial $f_{\mathcal{P}_2}$ for each symbol f from Σ and show that $l \succ_{\mathcal{P}_2} r$ holds for all $l \to r \in \mathcal{R}_2$.

Hints:

- You do not need coefficients that are greater than 3.

Exercise 3 (Unification): (2 + 4 = 6 points)

Apply the algorithm UNIFY from the lecture to compute a most general unifier for the following unification problems:

i) $\{\mathsf{f}(\mathsf{f}(x, y), \mathsf{f}(x, \mathsf{f}(y, x))) =^? \mathsf{f}(\mathsf{f}(\mathsf{g}(x, y, z), y), \mathsf{f}(\mathsf{f}(x, y), z))\}$

ii) $\{\mathsf{f}(\mathsf{h}(x_1), \mathsf{f}(x_3, x_4)) =^? \mathsf{f}(x_5, \mathsf{f}(x_4, x_2)), \mathsf{f}(\mathsf{h}(x_2), \mathsf{f}(x_3, x_4)) =^? \mathsf{f}(x_1, \mathsf{f}(x_2, x_2))\}$

Include all intermediate unification problems that are created in the computation and note which transformation rule is used in each step.
If the computation fails (i.e., if the problem is not unifiable), note the type of the error.

Challenge Exercise 4 (Unification): (4* points)

Implement the algorithm UNIFY from the lecture in your language of choice and use your implementation to analyze the following unification problems:

i) $\{\mathsf{f}(\mathsf{h}(x_1), \mathsf{f}(x_3, x_4)) =^? \mathsf{f}(x_2, \mathsf{f}(x_4, x_2)), \mathsf{f}(\mathsf{h}(x_1), \mathsf{f}(x_3, x_4)) =^? \mathsf{f}(x_3, \mathsf{f}(x_2, x_2))\}$

ii) $\{\mathsf{f}(\mathsf{h}(x_1), \mathsf{f}(x_3, x_4)) =^? \mathsf{f}(x_2, \mathsf{f}(x_4, x_2)), \mathsf{f}(\mathsf{h}(x_1), \mathsf{f}(x_3, x_4)) =^? \mathsf{f}(x_1, \mathsf{f}(x_2, x_2))\}$

iii) $\{\mathsf{g}(x_1, x_2, \mathsf{f}(y_0, y_0), \mathsf{f}(y_1, y_1), \mathsf{f}(y_2, y_2)) =^?$
$\mathsf{g}(\mathsf{f}(x_0, x_0), \mathsf{f}(x_1, x_1), y_1, y_2, x_2)\}$

iv) $\{\mathsf{g}(\mathsf{g}(x_1, \mathsf{f}(x_1, \mathsf{a}), x_2, x_2, x_3), \mathsf{f}(\mathsf{a}, x_2), x_1, x_2, \mathsf{f}(x_2, \mathsf{a})) =^?$
$\mathsf{g}(\mathsf{g}(x_2, x_4, x_1, x_2, \mathsf{f}(x_4, x_1)), \mathsf{f}(x_1, \mathsf{a}), x_1, x_2, \mathsf{f}(\mathsf{a}, x_1))\}$

v) $\{\mathsf{g}(x_2, x_1, \mathsf{f}(\mathsf{a}, y_3), \mathsf{f}(y_1, y_1), \mathsf{f}(y_2, y_2)) =^?$
$\mathsf{g}(\mathsf{f}(x_0, x_0), y_1, \mathsf{f}(x_1, x_1), x_2, y_3)\}$

Hand in the results as well as a detailed log of the steps performed by your program, i.e., whenever one of the rules "Delete", "Term reduction", "Swap", or "Variable reduction" is applied, your program has to log the name of the rule, the pair of terms processed by the rule, and the resulting unification problem.
Write an E-Mail with the source code to `florian.frohn@cs.rwth-aachen.de`. Do not use any non-standard libraries (e.g., just the Java Standard Library if you implement the algorithm in Java). Of course, you are *not* allowed to use predefined unification procedures.