

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 **Wednesday, June 29th, 2011**, in lecture hall **AH 2**. 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!

Exercise 1 (Undecidability of Local Confluence):
(3 + 3 = 6 points)

In the solution of Exercise 4 a) from the fourth exercise sheet, you find the following construction of a TRS for a given arbitrary Turing machine $(Q, \Gamma, \varepsilon, q_s, q_e, \delta)$:

"We first consider the representation of the tape. We use the binary concatenation operator $:$ and interpret $a : b : \text{nil}$ as the sequence of symbols ab . Now we insert rules for the state transitions. For this we need binary function symbols f_q for each state $q \in Q$. The first argument represents the tape contents left from and at the current position while the second argument represents the tape contents right from the current position. Although the tape is infinite, it suffices to use finite terms, since only finitely many positions may contain symbols different from the blank symbol: we do not represent the infinite sequences of blank symbols left and right from the other tape contents explicitly. The finite alphabet Γ is represented by a finite set of constants, so we use $\Gamma \subseteq \Sigma_0$.

Now we insert two rules to add blank symbols to our representation of the tape:

$$f_q(xs, \text{nil}) \rightarrow f_q(xs, \varepsilon : \text{nil}) \quad f_q(\text{nil}, ys) \rightarrow f_q(\varepsilon : \text{nil}, ys)$$

Depending on the movement of the Turing machine, we insert corresponding rules for the step function.

left movement: Let $\delta(q, a) = (q', \text{left}, b)$. Then we insert the following rule into the TRS:

$$f_q(a : l, x : r) \rightarrow f_{q'}(l, b : x : r)$$

right movement: Let $\delta(q, a) = (q', \text{right}, b)$. Then we insert the following rule into the TRS:

$$f_q(a : l, x : r) \rightarrow f_{q'}(x : b : l, r)$$

To easily detect termination of the Turing machine, we additionally insert the rule $f_{q_e}(x, y) \rightarrow \text{term}$. Moreover, we add the 'initial' rule $\text{init} \rightarrow f_{q_s}(\text{nil}, \text{nil})$."

- a) Please prove that the TRS constructed that way for a given arbitrary Turing machine is locally confluent.

Hints:

- Prove that all critical pairs have a common reduct.

- b) Please prove that the question whether a given arbitrary TRS is locally confluent is undecidable.

Hints:

- For an arbitrary Turing machine $(Q, \Gamma, \varepsilon, q_s, q_e, \delta)$, the constructed TRS \mathcal{R} according to the description above has the following property:
 $f_{q_s}(\text{nil}, \text{nil}) \rightarrow_{\mathcal{R}}^* \text{term}$ iff the Turing machine is terminating.
 Now modify the above construction principle such that for every Turing machine one constructs a TRS which is locally confluent iff the Turing machine is terminating.

Exercise 2 (Convergence):

(1 + 1 + 2 + 2 = 6 points)

In this exercise we investigate *convergence* (i.e., termination and confluence) for several given term rewrite systems. For each of the following term rewrite systems \mathcal{R}_i , please state whether \mathcal{R}_i is convergent and give an explanation for your answer.

If \mathcal{R}_i is not convergent, it suffices to sketch an infinite rewrite sequence from a term t or rewrite sequences from a term t to two terms which are not joinable (e.g., because they have no common normal forms).

If \mathcal{R}_i is convergent, please both give a proof of termination and a proof of confluence. For each required termination proof in this exercise, it will suffice to use an RPOS (or a weaker ordering from the lecture). Here you should also state explicitly which status and which precedence you are using.

Hints:

- For the confluence proofs, recall that a *terminating* term rewrite system is confluent if and only if it is locally confluent.

\mathcal{R}_a :

$$\begin{aligned} f(f(x, y), z) &\rightarrow f(x, f(y, z)) \\ f(x, y) &\rightarrow f(y, x) \end{aligned}$$

\mathcal{R}_b :

$$\begin{aligned} g(\mathcal{O}) &\rightarrow \mathcal{O} \\ g(s(x)) &\rightarrow x \\ g(s(s(x))) &\rightarrow s(g(x)) \end{aligned}$$

\mathcal{R}_c :

$$\begin{aligned} \text{plus}(\text{plus}(x, y), z) &\rightarrow \text{plus}(x, \text{plus}(y, z)) \\ \text{plus}(x, \mathcal{O}) &\rightarrow x \\ \text{plus}(x, s(y)) &\rightarrow s(\text{plus}(x, y)) \end{aligned}$$

\mathcal{R}_d :

$$\begin{aligned} \text{minus}(x, \mathcal{O}) &\rightarrow x \\ \text{minus}(\mathcal{O}, y) &\rightarrow \mathcal{O} \\ \text{minus}(s(x), s(y)) &\rightarrow \text{minus}(x, y) \end{aligned}$$

Exercise 3 (Completion):

(1 + 1 + 4 = 6 points)

Try to use the algorithm BASIC_COMPLETION from the lecture to complete the following systems $\mathcal{R}_1, \dots, \mathcal{R}_3$. Please give all critical pairs examined by the algorithm (please note from which rules they were created), the respective normal forms and if applicable, the constructed rewrite rule. If the algorithm fails, give the reason. In this exercise you do not need to give a proof for $s \succ t$ if you generate a new rule $s \rightarrow t$ (but this statement should be true, of course).

\mathcal{R}_1 :

$$\begin{aligned} \text{element}(\text{Cons}(x, xs)) &\rightarrow x && (1) \\ \text{element}(\text{Cons}(x, xs)) &\rightarrow \text{element}(xs) && (2) \end{aligned}$$

As reduction order \succ , use the LPO with precedence $\text{element} \sqsupset \text{Cons}$.

\mathcal{R}_2 :

$$f(x) \rightarrow s(p(x)) \quad (1)$$

$$f(x) \rightarrow p(s(x)) \quad (2)$$

$$p(s(x)) \rightarrow x \quad (3)$$

As reduction order \succ , use the LPO with precedence $f \sqsupset s \sqsupset p$.

\mathcal{R}_3 :

$$f(f(x)) \rightarrow h(x) \quad (1)$$

$$f(g(x)) \rightarrow f(x) \quad (2)$$

$$f(x) \rightarrow g(x) \quad (3)$$

As reduction order \succ , use the LPO with precedence $f \sqsupset h \sqsupset g$.