Prof. Dr. Jürgen Giesl
Marc Brockschmidt, Carsten Fuhs, Thomas Ströder

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 $(\mathcal{Q}, \Gamma, \varepsilon, q_s, q_e, \delta)$:

> *"We first consider the representation of the tape. We use the binary concatenation operator : and interpret* a : b : 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* $q \in \mathcal{Q}$*. 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:*
>
> $$\mathsf{f}_q(xs, \mathsf{nil}) \to \mathsf{f}_q(xs, \varepsilon : \mathsf{nil}) \qquad \mathsf{f}_q(\mathsf{nil}, ys) \to \mathsf{f}_q(\varepsilon : \mathsf{nil}, ys)$$
>
> *Depending on the movement of the Turing machine, we insert corresponding rules for the step function.*
>
> **left movement:** *Let* $\delta(q, \mathsf{a}) = (q', \mathsf{left}, \mathsf{b})$*. Then we insert the following rule into the TRS:*
>
> $$\mathsf{f}_q(\mathsf{a} : l, x : r) \to \mathsf{f}_{q'}(l, \mathsf{b} : x : r)$$
>
> **right movement:** *Let* $\delta(q, \mathsf{a}) = (q', \mathsf{right}, \mathsf{b})$*. Then we insert the following rule into the TRS:*
>
> $$\mathsf{f}_q(\mathsf{a} : l, x : r) \to \mathsf{f}_{q'}(x : \mathsf{b} : l, r)$$
>
> *To easily detect termination of the Turing machine, we additionally insert the rule* $\mathsf{f}_{q_e}(x, y) \to \mathsf{term}$*.*
> *Moreover, we add the 'initial' rule* $\mathsf{init} \to \mathsf{f}_{q_s}(\mathsf{nil}, \mathsf{nil})$*."*

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 $(\mathcal{Q}, \Gamma, \varepsilon, q_s, q_e, \delta)$, the constructed TRS $\mathcal{R}$ according to the description above has the following property:
  $\mathsf{f}_{q_s}(\mathsf{nil}, \mathsf{nil}) \to_{\mathcal{R}}^* \mathsf{term}$ 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.

Solution:

a) All critical pairs are of one of the following forms:

- $\langle \mathsf{f}_q(\mathsf{nil}, \varepsilon : \mathsf{nil}), \mathsf{f}_q(\varepsilon : \mathsf{nil}, \mathsf{nil}) \rangle$:
  Such pairs can obviously be reduced to $\mathsf{f}_q(\varepsilon : \mathsf{nil}, \varepsilon : \mathsf{nil})$.
- $\langle \mathsf{f}_{q_e}(xs, \varepsilon : \mathsf{nil}), \mathsf{term} \rangle$:
  Such pairs can obviously be reduced to term.
- $\langle \mathsf{f}_{q_e}(\varepsilon : \mathsf{nil}, ys), \mathsf{term} \rangle$:
  Such pairs can obviously be reduced to term.

There are no other critical pairs for the following reasons:

- Since no constructed rule contains a function symbol corresponding to a state except at the root position, the only possibility to obtain critical pairs is that two rules have unifying left-hand sides.
- Thus, rules for different states cannot lead to critical pairs.
- As δ is a function, there is only one rule constructed for each combination of a state q and a symbol a which has no critical pairs with itself.
- Different rules for the same state q must hence be constructed for different symbols a and b. But this means that there are different constants at the same position in the left-hand sides and, thus, the left-hand sides of the rules cannot unify. Therefore, we do not obtain critical pairs from such rules, either.

Since all critical pairs can be reduced to a common reduct, the constructed TRS is locally confluent. □

b) We know that the halting problem for Turing machines is undecidable. For an arbitrary Turing machine $(\mathcal{Q}, \Gamma, \epsilon, q_s, q_e, \delta)$, the constructed TRS $\mathcal{R}$ according to the given description is shown to be locally confluent in the previous exercise part. Now we modify this TRS such that it is locally confluent iff the Turing machine is terminating. Then undecidability of local confluence follows from undecidability of the halting problem.

We introduce one additional rule $\mathsf{init} \to \mathsf{term}$. The only critical pair arising from this modification is $\langle \mathsf{f}_{q_s}(\mathsf{nil}, \mathsf{nil}), \mathsf{term} \rangle$. By the hint, we know that $\mathsf{f}_{q_s}(\mathsf{nil}, \mathsf{nil}) \to_{\mathcal{R}}^* \mathsf{term}$ iff the Turing machine is terminating. Moreover, term is a normal form w.r.t. the constructed TRS. Thus, the critical pair is reducible to a common reduct iff the Turing machine is terminating. Hence, a decision procedure for local confluence would also decide the halting problem for Turing machines. Contradiction. □

______________________.

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.

$\mathcal{R}_a$:

$$\mathsf{f}(\mathsf{f}(x, y), z) \to \mathsf{f}(x, \mathsf{f}(y, z))$$
$$\mathsf{f}(x, y) \to \mathsf{f}(y, x)$$

$\mathcal{R}_b$:

$$\mathsf{g}(\mathcal{O}) \to \mathcal{O}$$
$$\mathsf{g}(\mathsf{s}(x)) \to x$$
$$\mathsf{g}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{g}(x))$$

$\mathcal{R}_c$:

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

$\mathcal{R}_d$:

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

Solution:

$\mathcal{R}_a$: The TRS $\mathcal{R}_a$ is not terminating, as witnessed by the following infinite rewrite sequence:

$$\mathsf{f}(x,y) \to_{\mathcal{R}_a} \mathsf{f}(y,x) \to_{\mathcal{R}_a} \mathsf{f}(x,y) \to_{\mathcal{R}_a} \dots$$

Therefore, it is not convergent either.

$\mathcal{R}_b$: The TRS $\mathcal{R}_b$ is not confluent. One can rewrite the term $\mathsf{g}(\mathsf{s}(\mathsf{s}(x)))$ to two different normal forms:

$$\mathsf{g}(\mathsf{s}(\mathsf{s}(x))) \to_{\mathcal{R}_b} \mathsf{s}(x) \not\to_{\mathcal{R}_b}$$

and

$$\mathsf{g}(\mathsf{s}(\mathsf{s}(x))) \to_{\mathcal{R}_b} \mathsf{s}(\mathsf{g}(x)) \not\to_{\mathcal{R}_b}$$

$\mathcal{R}_c$: The TRS $\mathcal{R}_c$ is not confluent. One can rewrite the term $\mathsf{plus}(\mathsf{plus}(x,\mathcal{O}),z)$ to two different normal forms:

$$\mathsf{plus}(\mathsf{plus}(x,\mathcal{O}),z) \to_{\mathcal{R}_c} \mathsf{plus}(x,\mathsf{plus}(\mathcal{O},z)) \not\to_{\mathcal{R}_c}$$

and

$$\mathsf{plus}(\mathsf{plus}(x,\mathcal{O}),z) \to_{\mathcal{R}_c} \mathsf{plus}(x,z) \not\to_{\mathcal{R}_c}$$

$\mathcal{R}_d$: The TRS $\mathcal{R}_d$ is indeed convergent. Termination is proved already by the embedding order. Since $\mathcal{R}_d$ is terminating, confluence and local confluence are equivalent for $\mathcal{R}_d$. To investigate local confluence, we check whether the critical pairs of $\mathcal{R}_d$ are joinable. For $\mathcal{R}_d$ we get the single critical pair $\langle \mathcal{O}, \mathcal{O} \rangle$, which is trivially joinable. Hence, we have also proved confluence.

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 $s \to t$ (but this statement should be true, of course).

$\mathcal{R}_1$:

$$\mathsf{element}(\mathsf{Cons}(x, xs)) \to x \quad (1)$$
$$\mathsf{element}(\mathsf{Cons}(x, xs)) \to \mathsf{element}(xs) \quad (2)$$

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

$\mathcal{R}_2$:

$$\mathsf{f}(x) \to \mathsf{s}(\mathsf{p}(x)) \quad (1)$$
$$\mathsf{f}(x) \to \mathsf{p}(\mathsf{s}(x)) \quad (2)$$
$$\mathsf{p}(\mathsf{s}(x)) \to x \quad (3)$$

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

$\mathcal{R}_3$:

$$\mathsf{f}(\mathsf{f}(x)) \to \mathsf{h}(x) \quad (1)$$
$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(x) \quad (2)$$
$$\mathsf{f}(x) \to \mathsf{g}(x) \quad (3)$$

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

Solution:

$\mathcal{R}_1$:

Rules	Critical pair	normal forms	new rule
1,2	$\langle x, \mathsf{element}(xs)\rangle$	$x, \mathsf{element}(xs)$	

As x and element(xs) are in normal form and we cannot orient the two using $\succ$, the algorithm returns FAIL.

$\mathcal{R}_2$:

Rules	Critical pair	normal forms	new rule
1,2	$\langle \mathsf{s}(\mathsf{p}(x)), \mathsf{p}(\mathsf{s}(x))\rangle$	$\mathsf{s}(\mathsf{p}(x)), x$	$\mathsf{s}(\mathsf{p}(x)) \to x$ (4)

No other critical pairs exist, so the algorithm returns $\mathcal{R}_2'$:

$$\mathsf{f}(x) \to \mathsf{s}(\mathsf{p}(x)) \quad (1)$$
$$\mathsf{f}(x) \to \mathsf{p}(\mathsf{s}(x)) \quad (2)$$
$$\mathsf{p}(\mathsf{s}(x)) \to x \quad (3)$$
$$\mathsf{s}(\mathsf{p}(x)) \to x \quad (4)$$

$\mathcal{R}_3$:

Rules	Critical pair	normal forms	new rule
1,1	$\langle \mathsf{f}(\mathsf{h}(x)), \mathsf{h}(\mathsf{f}(x))\rangle$	$\mathsf{h}(\mathsf{g}(x)), \mathsf{g}(\mathsf{h}(x))$	$\mathsf{h}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{h}(x))$ (4)
1,2	$\langle \mathsf{h}(\mathsf{g}(x)), \mathsf{f}(\mathsf{f}(x))\rangle$	$\mathsf{h}(\mathsf{g}(x)), \mathsf{h}(x)$	$\mathsf{h}(\mathsf{g}(x)) \to \mathsf{h}(x)$ (5)
1,3	$\langle \mathsf{h}(x), \mathsf{f}(\mathsf{g}(x))\rangle$	$\mathsf{h}(x), \mathsf{g}(\mathsf{g}(x))$	$\mathsf{h}(x) \to \mathsf{g}(\mathsf{g}(x))$ (6)
1,3	$\langle \mathsf{h}(x), \mathsf{g}(\mathsf{f}(x))\rangle$	$\mathsf{h}(x), \mathsf{g}(\mathsf{g}(x))$	$\mathsf{h}(x) \to \mathsf{g}(\mathsf{g}(x))$ (6)
2,3	$\langle \mathsf{f}(x), \mathsf{g}(\mathsf{g}(x))\rangle$	$\mathsf{g}(x), \mathsf{g}(\mathsf{g}(x))$	$\mathsf{g}(\mathsf{g}(x)) \to \mathsf{g}(x)$ (7)
2,7	$\langle \mathsf{f}(\mathsf{g}(x)), \mathsf{f}(\mathsf{g}(x))\rangle$	$\mathsf{g}(x), \mathsf{g}(x)$	
4,5	$\langle \mathsf{g}(\mathsf{h}(x)), \mathsf{h}(x)\rangle$	$\mathsf{g}(x), \mathsf{g}(x)$	
4,6	$\langle \mathsf{g}(\mathsf{h}(x)), \mathsf{g}(\mathsf{g}(\mathsf{g}(x)))\rangle$	$\mathsf{g}(x), \mathsf{g}(x)$	
4,7	$\langle \mathsf{g}(\mathsf{h}(\mathsf{g}(x))), \mathsf{h}(\mathsf{g}(x))\rangle$	$\mathsf{g}(x), \mathsf{g}(x)$	
5,6	$\langle \mathsf{h}(x), \mathsf{g}(\mathsf{g}(\mathsf{g}(x)))\rangle$	$\mathsf{g}(x), \mathsf{g}(x)$	
5,7	$\langle \mathsf{h}(\mathsf{g}(x)), \mathsf{h}(\mathsf{g}(x))\rangle$	$\mathsf{g}(x), \mathsf{g}(x)$	
7,7	$\langle \mathsf{g}(\mathsf{g}(x)), \mathsf{g}(\mathsf{g}(x))\rangle$	$\mathsf{g}(x), \mathsf{g}(x)$	

No other critical pairs exist, so the algorithm returns $\mathcal{R}'_3$:

$$\mathsf{f}(\mathsf{f}(x)) \to \mathsf{h}(x) \tag{1}$$
$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(x) \tag{2}$$
$$\mathsf{f}(x) \to \mathsf{g}(x) \tag{3}$$
$$\mathsf{h}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{h}(x)) \tag{4}$$
$$\mathsf{h}(\mathsf{g}(x)) \to \mathsf{h}(x) \tag{5}$$
$$\mathsf{h}(x) \to \mathsf{g}(\mathsf{g}(x)) \tag{6}$$
$$\mathsf{g}(\mathsf{g}(x)) \to \mathsf{g}(x) \tag{7}$$