Assignment 6

1. Prove Lemma 5.5: Let S be a set of ground clauses.
   - The TRS R_S is convergent.
   - If $R_C^* \models C$, then $R_S^* \models C$
2. Prove Lemma 5.6: If a clause C, where $C \in S$ is productive, i.e. it has the form $\Gamma \rightarrow l \approx r, \Delta$ and it generates the rule $l \rightarrow r$, then $R_S \models \Gamma$ and $R_S \not\models \Delta$.
3. Complete the proof of Theorem 5.3. Consider the following cases:
   - $C\gamma = (\Gamma\gamma \rightarrow s\gamma \approx t\gamma, \Delta\gamma)$ where $s\gamma \approx t\gamma$ is maximal, but not strictly maximal.
   - $C\gamma = (\Gamma\gamma \rightarrow s\gamma \approx t\gamma, \Delta\gamma)$ where $s\gamma$ is reducible in $R_{C\gamma}$.
   - $C\gamma = (\Gamma\gamma, s\gamma \approx t\gamma \rightarrow \Delta\gamma)$, where $s\gamma = t\gamma$, $s\gamma \approx t\gamma$ is selected or nothing is selected, but $s\gamma \approx t\gamma$ is maximal.
   - $C\gamma = (\Gamma\gamma, s\gamma \approx t\gamma \rightarrow \Delta\gamma)$ where $s\gamma \neq t\gamma$, $s\gamma \approx t\gamma$ is selected or nothing is selected, but $s\gamma \approx t\gamma$ is maximal.
4. Using superposition with deletion of redundant clauses (standard redundancy criterion), prove that the satisfiability of ground clauses is decidable.

   Hint:

   - Show that a set of ground clauses S can be transformed into a set of sets of Horn ground clauses $S_1, \ldots, S_n$. (Splitting) S is satisfiable iff there is i, $1 \leq i \leq n$, such that S_i is satisfiable.
   - Show that satisfiability of a set of ground equational (all atoms are equations) Horn clauses is decidable: by showing that one of the premises of superposition right or left may be removed due to standard redundancy criterion. What measure is decreased with each inference then?
5. Let $a \succ b \succ c \succ d$. Show that the following set of clauses is unsatisfiable. Use Superposition with Merging Paramodulation and Positive Factoring instead of Equality Factoring:

   $S := \{a \approx d \rightarrow a \approx c, \quad \rightarrow b \approx d, \quad \rightarrow a \approx b, a \approx d, \quad c \approx d \rightarrow\}$

6. Pigeon-hole problem

   Suppose there are 2 holes and 3 pigeons to put in the holes. Every pigeon is in a hole and no hole contains more than one pigeon. Prove that this is impossible.

   Use resolution and superposition inferences with OTTER or SPASS.

   To obtain superposition in OTTER use the following settings:

```
set(para_from). %from a given clause
set(para_into). %into a given clause
clear(para_from_right).  %ordered paramodulation
clear(para_into_right). %superposition
```

   To obtain superposition in SPASS use the options:

   -ISpR Enables/disables the inference rule Superposition Right. Default is 0.

   -ISpL Enables/disables the inference rule Superposition Left. Default is 0.

   -IEqR Enables/disables the inference rule Equality Resolution. Default is 0.

   -IEqF Enables/disables the inference rule Equality Factoring. Default is 0.

   -IMPm Enables/disables the inference rule Merging Paramodulation. Default is 0.

   More options in SPASS can be found here:

   `http://www.spass-prover.org/webspass/help/options.html`

   You have to be able to explain the information in the output file.

   (Try it also for a bigger number of holes and pigeons.)