

# LOGICS EXERCISE

TU MÜNCHEN  
INSTITUT FÜR INFORMATIK

PROF. TOBIAS NIPKOW  
DR. PETER LAMMICH  
SIMON WIMMER

SS 2016

EXERCISE SHEET 7

25.05.2016

**Submission of Homework:** Before tutorial on June 1

## Exercise 7.1. [QE for DLO]

Use the quantifier-elimination procedure for DLOs to check whether the following formula is a member of  $Th(DLO)$ :

$$\exists x \forall y \exists z ((x < y \vee z < x) \wedge y < z)$$

## Exercise 7.2. [Compactness for FOL]

Prove the Compactness Theorem for first-order logic:

*If every finite subset of a (countable) set  $M$  of formulas has a model, then  $M$  as a whole has a model.*

Hint: You may use the following definitions and theorems:

Let  $M = \{F_1, F_2, \dots\}$ . Then  $skolem(M) = \{F'_1, F'_2, \dots\}$ , where  $F'_i$  is a Skolem form of  $F_i$ , and the Skolem-functions in the  $F'_i$  are pairwise disjoint.

Note: Wlog, we assume that  $M$  is of a form that allows us to find enough fresh function symbols for Skolem functions.

Theorem:  $M$  is satisfiable iff  $skolem(M)$  is satisfiable.

## Exercise 7.3. [Axiomatizations and Compactness]

Using compactness, show that if a theory is finitely axiomatizable, any countable axiomatization of it has a finite subset that axiomatizes the same theory. In other words, if  $Cn(\Gamma) = Cn(\Delta)$  with  $\Gamma$  countable and  $\Delta$  finite, then there is a finite  $\Gamma' \subseteq \Gamma$  with  $Cn(\Gamma') = Cn(\Gamma)$ .

**Homework 7.1.** [Theories]

(5 points)

1. Show  $Cn(S) = Th(Mod(S))$ , i.e. show  $Th(Mod(S)) = \{F \mid F \Sigma\text{-sentence and } S \models F\}$
2. Show that  $Cn$  is a closure operator, i.e.  $Cn$  fulfills the following properties:
  - $S \subseteq Cn(S)$
  - if  $S \subseteq S'$  then  $Cn(S) \subseteq Cn(S')$
  - $Cn(Cn(S)) = Cn(S)$

**Solution:** In the following, suppose  $S, S'$  are sets of  $\Sigma$ -sentences and  $F$  is  $\Sigma$ -sentence.

$$\begin{aligned}
 & F \in Cn(S) \\
 & \text{iff } S \models F \\
 & \text{iff for all suitable } \mathcal{A}, \text{ if } \mathcal{A} \models S \text{ then } \mathcal{A} \models F \\
 & \text{iff for all } \Sigma\text{-structures } \mathcal{A}, \text{ if } \mathcal{A} \models S \text{ then } \mathcal{A} \models F \\
 & \text{iff for all } \Sigma\text{-structures, for all } F \in S, \text{ if } \mathcal{A} \models S \text{ then } \mathcal{A} \models F \\
 & \text{iff for all } \mathcal{A} \in Mod(S), \mathcal{A} \models F \\
 & \text{iff } F \in Th(Mod(S))
 \end{aligned}$$

$$F \in S \implies S \models F \implies Cn(S) \models F$$

$$F \in Cn(S) \implies S \models F \implies S' \models F \implies F \in Cn(S')$$

From the above two:  $Cn(S) \subseteq Cn(Cn(S))$

$$F \in Cn(Cn(S)) \implies Cn(S) \models F \implies^{(*)} S \models F \implies F \in Cn(S)$$

We have (\*) because  $\mathcal{A} \models Cn(S)$  iff  $\mathcal{A} \models S$  by definition of  $Cn$ .

**Homework 7.2.** [Quantifier Elimination for  $Th(\mathbb{N}, 0, S, =)$ ]

(5 points)

Give a quantifier-elimination procedure for  $Th(\mathbb{N}, 0, S, =)$  where  $S$  is the successor operation on natural numbers, i.e.  $S(n) = n + 1$ .

*Hint:*  $a = b$  iff  $S^k(a) = S^k(b)$  for any  $a, b, k \in \mathbb{N}$ .

**Solution:** We assume  $F = \exists x(A_1 \wedge \dots \wedge A_n)$  where  $x$  occurs in all  $A_i$  and each  $A_i$  is of the form

$$S^k(x) = S^m(t) \text{ or } S^k(x) \neq S^m(t)$$

where  $t$  is 0 or a variable (using symmetry of  $=$ ). By observing that  $Th(\mathbb{N}, 0, S) \models (S^k(x) = S^k(x)) \leftrightarrow \top$  and  $Th(\mathbb{N}, 0, S) \models (S^k(x) \neq S^k(x)) \leftrightarrow \perp$ , we may further assume  $x \neq t$ .

We have to distinguish two cases: (1) all  $A_i$  only use  $\neq$  but not  $=$ . Then

$$Th(\mathbb{N}, 0, S) \models (\exists x(A_1 \wedge \dots \wedge A_n)) \leftrightarrow \top$$

(2) There is at least one  $A_i$  of the form  $S^k(x) = t$  where  $x \neq t$ . We replace  $A_i$  using the following procedure:

- If  $k > 0$ , we add the constraints  $t \neq 0 \wedge \dots \wedge t \neq S^{k-1}(0)$  to ensure that the solution for  $x$  is non-negative.
- In each other  $A_j$ , we replace  $S^m(x) = u$  by  $S^m(t) = S^k(u)$  using the hint.

**Homework 7.3.** [Quantifier Elimination for DLOs with endpoints] (5 points)

Let  $\Sigma = \{a, b, <, =\}$  and replace the last two axioms for DLOs with:

- $\forall x(x = a \vee a < x)$
- $\forall x(x = b \vee x < b)$

Modify the quantifier-elimination procedure for dense linear orders to obtain a quantifier-elimination procedure for this theory.

What happens if there is only one endpoint?

**Solution:** Let us denote the new theory by  $Th(DLOP)$ . We extend step three of the quantifier-elimination procedure for DLOs from the lecture to cater for cases where there are only bounds of the form  $t < x$  (of the form  $x < t$ ). For this case we need ensure that none of them may be the respective endpoint ( $a$  or  $b$ ) for those constraints to be satisfied. Thus, we use

$$Th(DLOP) \models \exists x \left( \bigwedge_{i=1}^m l_i < x \right) \leftrightarrow l_i < b$$

and

$$Th(DLOP) \models \exists x \left( \bigwedge_{i=1}^m x < u_i \right) \leftrightarrow a < u_i$$

.

If there is only one endpoint, adding only the one type of corresponding constraints from above is sufficient.

**Homework 7.4. [Decidable Axiomatizations]** (5 points)

Show that any set of sentences that is axiomatized by a recursively enumerable set is also axiomatized by a decidable set.

*Hint:* For each  $n \in \mathbb{N}$  a possible encoding of  $n$  as a formula could be of the form

$$\underbrace{F \wedge \dots \wedge F}_{n \text{ times } F}$$

for some formula  $F$ .

**Solution:** Suppose the recursively enumerable set  $M = \{F_1, F_2, F_3, \dots\}$  axiomatizes  $S$ . We define  $M' = \{F_1 \wedge F_1, F_2 \wedge F_2 \wedge F_2, F_3 \wedge F_3 \wedge F_3 \wedge F_3, \dots\}$ . Clearly,  $M'$  axiomatizes  $S$ .

To decide whether  $F \in M'$  for any sentence  $F$ , we check syntactically if  $F = \bigwedge_{i=1}^n F'$  for some sentence  $F'$  and  $n > 1$ . If this is the case, we compute  $F_{n-1}$  (a terminating computation) and check whether  $F' = F_{n-1}$ . In this case we answer with 'yes', and we answer with 'no' in all other cases.