

Logic

Exercise Sheet 11

Exercise 11.1

- Show that $\Sigma \subseteq \text{Th}(\text{Mod}(\Sigma))$ holds for any set of axioms Σ . When does equality hold?
- Consider the language consisting of just the single predicate P . Show that $\text{Cn}(\forall x.P(x))$ is complete.
- Give an example for a complete theory T and a formula p (which may contain free variables), such that $T \not\models p$ and $T \not\models \neg p$.
- We have shown that a complete theory is always decidable. Give a counterexample for the reverse implication.

Solution

- This follows directly from the definitions: Let $p \in \Sigma$, and pick an arbitrary model $M \in \text{Mod}(\Sigma)$. By definition, we have that $\models_M p$, and since M was arbitrary, we know that $p \in \text{Th}(\text{Mod}(\Sigma))$. The inclusion becomes equality when Σ is a theory, i.e., closed under consequence.
- We show that $T = \text{Cn}(\forall x.P(x))$ is decidable, which implies its completeness. In all models of $\forall x.P(x)$, the predicate P is true for every domain element. Hence, we can decide if $p \in T$ for any sentence p by replacing any occurrence of $P(t)$ by \top . The quantifiers can then be dropped, because there are no terms in which variables occur. The resulting formula p' consists only of propositional connectives and the two atoms \top and \perp , and $p \in T$ iff p' is valid, which can be easily checked.
- Take for example the theory of dense linear orders and the formula $x < y$.
- In a language of monadic first-order logic, the theory $\text{Cn}(\emptyset)$, which contains all valid formulas, is decidable. However, it is not complete. Consider, for example, the sentence $\forall x.P(x) \vee Q(x)$. It is not valid, but satisfiable; moreover, its negation is also not valid, but again satisfiable.

Exercise 11.2 Quantifier elimination for dense linear orders

Apply quantifier elimination to the following formulas:

- $\forall x. x < a \Rightarrow x < b$
- $\exists x y. \forall z. (z < x \Rightarrow z \leq y) \wedge (z > y \Rightarrow z \geq x)$
- $\exists x. \neg(\forall y. x = y \Rightarrow (y \geq a \vee y \geq b \vee y \leq c))$

Solution

a) By double-negation and applying de Morgan's law, we obtain

$$\neg(\exists x. \neg(x < a \Rightarrow x < b))$$

and further

$$\neg(\exists x. x < a \wedge \neg(x < b))$$

and thus the following formula

$$\neg(\exists x. x < a \wedge (x = b \vee b < x))$$

We need to bring the inner formula into disjunctive normal form

$$\neg(\exists x. (x < a \wedge x = b) \vee (x < a \wedge b < x))$$

and push the quantifier in front of each disjunct

$$\neg((\exists x. x < a \wedge x = b) \vee (\exists x. x < a \wedge b < x))$$

The left disjunct is equivalent to $b < a$ and the right disjunct is equivalent to $b < a$. Thus, the result of quantifier elimination is

$$\neg(b < a)$$

b)

$$\begin{aligned} & \exists x y. \forall z. (z < x \Rightarrow z \leq y) \wedge (z > y \Rightarrow z \geq x) \\ \equiv & \exists x y. \neg(\exists z. (z < x \wedge z > y) \vee (z > y \wedge z < x)) \\ \equiv & \exists x y. \neg(y < x) \\ \equiv & \exists x y. y = x \vee y > x \\ \equiv & \top \end{aligned}$$

c)

$$\begin{aligned} & \exists x. \neg(\forall y. x = y \Rightarrow (y \geq a \vee y \geq b \vee y \leq c)) \\ \equiv & \exists x y. x = y \wedge y < a \wedge y < b \wedge y > c \\ \equiv & \exists x. x < a \wedge x < b \wedge x > c \\ \equiv & c < a \wedge c < b \end{aligned}$$

Exercise 11.3 Dense linear orders with endpoints

Dense linear orders with endpoints ∞ and $-\infty$ are axiomatizable by:

$$\begin{array}{ll} \forall x y. x = y \vee x < y \vee y < x & \forall x y. x < y \Rightarrow \exists z. x < z \wedge z < y \\ \forall x y z. x < y \wedge y < z \Rightarrow x < z & \forall x. x < \infty \vee x = \infty \\ \forall x. \neg(x < x) & \forall x. -\infty < x \vee x = -\infty \end{array}$$

Show that this theory admits quantifier elimination by describing an algorithm.

Solution

The quantifier elimination algorithm for dense linear orders without endpoints needs only slight modification to be usable as quantifier elimination algorithm for dense linear orders with endpoints. Consider a clause $C = \exists x.P[x]$ where P is a conjunction of literals $x < t$, $t < x$, $x = t$ or $t = x$ for arbitrary t . As usual, we treat this conjunction as set M . The elimination of x then yields:

- a) if $(x < x) \in M$, then \perp ,
- b) if $(x < -\infty) \in M$ or $(\infty < x) \in M$, then \perp ,
- c) if $(x = t) \in M$ or $(t = x) \in M$, then $P[t] \setminus (t = t)$,
- d) if $M = (x < t)$, then $-\infty < t$,
- e) if $M = (t < x)$, then $t < \infty$,
- f) in any other case, group the inequalities and construct a new conjunction as in the quantifier elimination for DLO without endpoints.

Step b) follows directly from the last two axioms, while step d) and e) are correct due to the following equivalences:

$$\begin{aligned} & \exists x. x < y \\ \equiv & \exists x. x < y \wedge (-\infty < x \vee x = -\infty) \\ \equiv & \exists x. (x < y \wedge -\infty < x) \vee (x < y \wedge x = -\infty) \\ \equiv & -\infty < y \end{aligned}$$

Exercise 11.4

The Compactness Theorem for first-order logic states that a countable set of formulas has a model iff all its finite subsets have a model.

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 $\text{Cn}(\Gamma) = \text{Cn}(\Delta)$ with Δ finite, then there is a finite $\Gamma' \subseteq \Gamma$ with $\text{Cn}(\Gamma') = \text{Cn}(\Gamma)$.

Solution

Clearly, $\Delta \subseteq \text{Cn}(\Delta) = \text{Cn}(\Gamma)$, and hence $\Gamma \cup \neg\Delta$ is unsatisfiable. Then, by the Compactness Theorem, there is a finite subset Γ' , such that $\Gamma' \cup \neg\Delta$ is unsatisfiable. Hence $\Gamma' \models \Delta$ and $\text{Cn}(\Gamma) = \text{Cn}(\Delta) \subseteq \text{Cn}(\Gamma')$. The reverse inclusion is obvious.

Note that the notation $\neg\Delta$ is a little sloppy, since Δ is a set of formulas. However, the set is finite, so we can also view it as a big conjunction, which gives a meaning to $\neg\Delta$.