Advanced Mathematical Logic - Exercises

Janos Tapolczai

May 22, 2014

1 Exercise 1

$$
\begin{aligned}
&L = \{e/0, \circ/2\} \\
\\
&F = F_1 \wedge F_2 \wedge F_3 \\
&F_1 = [\forall x, y, z] \circ (\circ(x, y), z) = \circ(x, \circ(y, z)) \\
&F_2 = [\forall x] \circ (e, x) = x \\
&F_3 = [\forall x][\exists y] \circ (x, y) = e \\
\\
&S = \{\mathbb{Z}, \{\hat{e}, \hat{\circ}\}\}
\end{aligned}
$$

Prove: $S, \phi \models F$ (for any ϕ).

Solution.

$$S, \phi \models F \equiv S, \phi \models (F_1 \wedge F_2) \wedge F_3 \equiv S, \phi \models F_1 \text{ and } S, \phi \models F_2 \text{ and } S, \phi \models F_3$$

- F_1:
  $S, \phi \models F_1 \equiv S, \phi^{x\ y\ z}_{a\ b\ c} \models ((x \circ y) \circ z) = (x \circ (y \circ z))$ for all a, b, c in $\mathbb{Z}$

  (We will refer to $\phi^{x\ y\ z}_{a\ b\ c}$ by the marco φ and "for all a, b, c in $\mathbb{Z}$" will henceforth be ommitted for readability.)

  $S, \phi \models F_1 \equiv (\varphi((x \circ y) \circ z),\ \varphi(x \circ (y \circ z))) \in \hat{=}$

  $S, \phi \models F_1 \equiv (\varphi(x \circ y)\ \hat{\circ}\ \varphi(z),\ \varphi(x)\ \hat{\circ}\ \varphi(y \circ z)) \in \hat{=}$

  $S, \phi \models F_1 \equiv ((a\ \hat{\circ}\ b),\ c \mathrel{\hat{=}} a\ \hat{\circ}\ (b\ \hat{\circ}\ c)) \in \hat{=}$

  $S, \phi \models F_1 \equiv ((a + b)\ c,\ a + (b + c)) \in \hat{=}$
  Associativity; provable by induction on $+$.

- F_2:
  $S, \phi \models F_2 \equiv S, \phi^x_a \models e \circ x = x$ for all a in $\mathbb{Z}$

  $S, \phi \models F_2 \equiv (\phi^x_a(e \circ x),\ \phi^x_a(x)) \in \hat{=}$

  $S, \phi \models F_2 \equiv (\phi^x_a(e)\ \hat{\circ}\ \phi^x_a(x),\ a) \in \hat{=}$

  $S, \phi \models F_2 \equiv\ (\hat{e}\ + a,\ a) \in \hat{=}$

  $S, \phi \models F_2 \equiv\ (0 + a,\ a) \in \hat{=}$
  Unit element; provable by induction on/definition of $+$.

- F_3:
  $S, \phi \models F_3 \equiv S, \phi^x_a \models [\exists y] x \circ y = e$ for all a in $\mathbb{Z}$

  $S, \phi \models F_3 \equiv S, \phi^{x\ y}_{a\ (-a)} \models x \circ y = e$ for all a in $\mathbb{Z}$

  $S, \phi \models F_3 \equiv S, \phi^{x\ y}_{a\ (-a)} \models x \circ y = e$

  $S, \phi \models F_3 \equiv (\phi^{x\ y}_{a\ (-a)}(x \circ y),\ \phi^{x\ y}_{a\ (-a)}(e)) \in \hat{=}$

  $S, \phi \models F_3 \equiv (a + (-a),\ 0) \in \hat{=}$
  Provable by induction on $+$.

2 Exercise 2

Show that $S \models \exists x F$ iff $S \models \neg\forall x \neg F$ and $\exists x F$ is unsatisfiable iff $\forall x \neg F$ is valid.

Solution. Let S be any structure. Case distinction:

1. $S \models \exists x F$. Then there exists a term t s.t. $S, \phi_t^x \models F$ (for any environment ϕ). Now we evaluate $S \models \neg\forall x \neg F$ (again, for any environment ϕ):

$$\begin{aligned} S, \phi \models \neg\forall x \neg F \quad &\equiv S, \phi \not\models \forall x \neg F \\ &\equiv S, \phi_d^x \not\models \neg F \text{ for all } d \text{ in } D \\ &\text{We choose } d \rightarrow t. \\ &\equiv S, \phi_t^x \not\models \neg F \\ &\equiv S, \phi_t^x \models F \end{aligned}$$

2. $S \not\models \exists x F$. Then there exists no term t s.t. $S, \phi_t^x \models F$. We now evaluate $S \not\models \neg\forall x \neg F$:

$$\begin{aligned} S, \phi \not\models \neg\forall x \neg F \quad &\equiv S, \phi \models \forall x \neg F \\ &\equiv S, \phi_d^x \models \neg F \text{ for all } d \text{ in } D \\ &\equiv S, \phi_d^x \not\models F \text{ for all } d \text{ in } D \end{aligned}$$

Therefore, if we substitute any term t for d, $S, \phi_t^x \models F$ will be false.

3 Exercise 3

Show that $\forall x(P(x) \to Q(x) \to \forall xP(x) \to \forall xQ(x)$ is valid but hat $(\forall xP(x) \to \forall xQ(x)) \to \forall x(P(x) \to Q(x)$ is not.

Solution.
The first formula can be proven in **LK**:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(\alpha) \vdash P(\alpha) \qquad Q(\alpha) \vdash Q(\alpha)}{P(\alpha) \to Q(\alpha), P(\alpha) \vdash Q(\alpha)} \to l}{P(\alpha) \to Q(\alpha), \forall xP(x) \vdash Q(\alpha)} \forall l(x := \alpha)}{\forall x(P(x) \to Q(x), \forall xP(x) \vdash Q(\alpha)} \forall l(x := \alpha)}{\forall x(P(x) \to Q(x), \forall xP(x) \vdash \forall xQ(x)} \forall r(x := \alpha)}{\forall x(P(x) \to Q(x) \vdash \forall xP(x) \to \forall xQ(x)} \to r}{\vdash \forall x(P(x) \to Q(x) \to (\forall xP(x) \to \forall xQ(x))} \to r
$$

For the second, we can give a counterexample:

$$S = \{\mathbb{N}, \{(P(x) \mapsto x \text{ is even}),\ (Q(x) \mapsto x \text{ is odd})\}\}$$

Informally, we can see that $\forall xP(x)$ is false, and hence, $\forall xP(x) \to \forall xQ(x)$ is true. Its consequent, $\forall x(P(x) \to Q(x)$, however, is false because $Q(x)$ is false precisely when $P(x)$ is true. Thereby the formula is falsified.

4 Exercise 4

Prove that the definition of $\models$ is sound for sentences F, i.e. $S, \phi \models F$ iff $S, \phi' \models F$ for all environments ϕ, ϕ'.

Solution. We proceed inductively from the root to the leaves of the formula tree represented by F — that is, we show that, if we take any two ϕ and ϕ' and expand the definition of $\models$, then the predicates of F will evaluate to the same truth value under ϕ and ϕ'.

The proof has the following structure: when we proceed downward through F's formula tree, the same changes to ϕ and ϕ' are made. Taking that as an assumption, the formula will evaluate to the same value under ϕ and ϕ' when we go back up.

Let ϕ and ϕ' be two arbitrary environments and F be a sentence. For simplicity, we number the variables in the formula $x_1, \ldots$ and identify these environments by their function graphs over V, that is:

$$\begin{aligned} \phi &= \{(x_1, v_1), (x_2, v_2), \ldots\} \\ \phi' &= \{(x_1, v_1'), (x_2, v_2'), \ldots\} \end{aligned}$$

Now we make a case distinction on F and its subformulas:

- $F = A \wedge B$, $F = A \vee B$, or $F = \neg A$. We proceed by evaluating the subformulas A, B. The environment is neither consulted nor changed.
- $F = \forall x_i A$. $S, \phi|\phi' \models F$ iff $S, \phi_d^x|\phi_d'^x \models F$ for all $d \in D$. We overwrite x_i with some d in both environments. Per definition, their function graphs (in A) are now:

$$\begin{aligned} \phi_d^x &= \{(x_1, v_1), \ldots, (x_i, d), (x_{i+1}, v_{i+1}), \ldots\} \\ \phi_d'^x &= \{(x_1, v_1'), \ldots, (x_i, d), (x_{i+1}, v_{i+1}'), \ldots\} \end{aligned}$$

- $F = \exists x_i A$. $S, \phi|\phi' \models F$ iff $S, \phi_d^x|\phi_d'^x \models F$ for some $d \in D$.

  We again change the function graphs of ϕ, ϕ' (but only in A):

$$\begin{aligned} \phi_d^x &= \{(x_1, v_1), \ldots, (x_i, d), (x_{i+1}, v_{i+1}), \ldots\} \\ \phi_d'^x &= \{(x_1, v_1'), \ldots, (x_i, d), (x_{i+1}, v_{i+1}'), \ldots\} \end{aligned}$$

  This is analogous to the $\forall$-case, but we now choose a concrete element $d \in D$ instead of a variable one.

- $F = P(t_1, \ldots, t_n)$. Per definition of $S, \phi|\phi' \models F$:

$$\begin{aligned} S, \phi \models F &\equiv (\phi(t_1), \ldots, \phi(t_n)) \in \hat{P} \\ S, \phi' \models F &\equiv (\phi'(t_1), \ldots, \phi'(t_n)) \in \hat{P} \end{aligned}$$

$\hat{P}$ only depends on S, not on the environment. To show that $(\phi(t_1), \ldots, \phi(t_n)) \in \hat{P} \Leftrightarrow (\phi'(t_1), \ldots, \phi'(t_n)) \in \hat{P}$, we prove that $\phi(t_i) = \phi'(t_i)$ for all $1 \leq i \leq n$. Induction on the depth of t_i's formula tree:

Base case. $t_i = x_j \in V$: Since F was a sentence, x_j was bound (most recently) by a quantifier to some d. Per the $\forall/\exists$-cases, the function-graphs of ϕ and ϕ' contain the tuple (x_j, d). Therefore, $\phi(x_j) = \phi'(x_j) = d$.

Step case. $t_i = f(s_1, \ldots, s_m)$. Per definition,

$$\begin{aligned}\phi(t_i) &= \hat{f}(\phi(s_1), \ldots, \phi(s_m))\\ \phi'(t_i) &= \hat{f}(\phi'(s_1), \ldots, \phi'(s_m))\end{aligned}$$

Per the induction hypothesis, $\phi(s_i) = \phi'(s_i)$ $(1 \leq i \leq m)$. That is, the arguments of $\hat{f}$ evaluate to the same values under both environments. Since $\hat{f}$ itself depends only on S, it follows that $\phi(t_i) = \phi'(t_i)$.

5 Exercise 5

Prove that $\mathrm{nnf}(\neg A) = \overline{\mathrm{nnf}(A)}$.

Solution. Structural induction on the cases of the definitions of $\mathrm{nnf}(A)$ and $\overline{A}$:

- Base case: A is an atom. By definition, $\overline{\mathrm{nnf}(A)} = \neg A = \mathrm{nnf}(\neg A)$. Note: the base case of nnf is undefined, but $\mathrm{nnf}(\neg A) = \neg A$ for atomic A is assumed.
- Step cases (IH: the equivalence holds for subformulas B, C of A):
  1. $A = B \vee C$. By definition:

     $$\mathrm{nnf}(\neg(B \vee C)) = \mathrm{nnf}(\neg B) \wedge \mathrm{nnf}(\neg C) \tag{1}$$

     By definition of nnf and the complement:

     $$\overline{\mathrm{nnf}(B \vee C)} = \overline{\mathrm{nnf}(B) \vee \mathrm{nnf}(C)} = \overline{\mathrm{nnf}(B)} \wedge \overline{\mathrm{nnf}(C)} \tag{2}$$

     By the IH, $\mathrm{nnf}(\neg B) \wedge \mathrm{nnf}(\neg C) = \overline{\mathrm{nnf}(B)} \wedge \overline{\mathrm{nnf}(C)}$.
  2. $A = B \wedge C$. Analogous to the previous case.
  3. $A = \neg(B \vee C)$. By definition:

     $$\mathrm{nnf}(\neg\neg(B \vee C)) = \mathrm{nnf}(B \vee C) = \mathrm{nnf}(B) \vee \mathrm{nnf}(C) \tag{3}$$

     By definition of nnf and the complement:

     $$\overline{\mathrm{nnf}(\neg(B \vee C))} = \overline{\mathrm{nnf}(\neg B) \wedge \mathrm{nnf}(\neg C)} = \overline{\mathrm{nnf}(\neg B)} \vee \overline{\mathrm{nnf}(\neg C)} \tag{4}$$

     By the IH, $\mathrm{nnf}(B) \vee \mathrm{nnf}(C) = \overline{\mathrm{nnf}(\neg B)} \vee \overline{\mathrm{nnf}(\neg C)}$[1].
  4. $A = \neg(B \wedge C)$. Analogous to the previous case.
  5. $A = \exists x B$. By definition:

     $$\mathrm{nnf}(\neg\exists x B) = \forall x\ \mathrm{nnf}(\neg B) \tag{5}$$

     By definition of nnf and the complement:

     $$\overline{\mathrm{nnf}(\exists x B)} = \overline{\exists x\ \mathrm{nnf}(B)} = \forall x\ \overline{\mathrm{nnf}(B)} \tag{6}$$

     By the IH, $\forall x\ \mathrm{nnf}(\neg B) = \forall x\ \overline{\mathrm{nnf}(B)}$.

[1] The equivalence can be derived from the IH as follows: $\overline{\mathrm{nnf}(\neg F)} \underset{IH}{=} \overline{\overline{\mathrm{nnf}(F)}} = \mathrm{nnf}(F)$. This detail will be omitted in the subsequent cases where it applies.

6. $A = \forall x B$. Analogous to the previous case (with $\exists$ and $\forall$ exchanged).
7. $A = \neg\exists x B$. By definition:

$$\mathrm{nnf}(\neg\neg\exists x B) = \mathrm{nnf}(\exists x B) = \exists x\ \mathrm{nnf}(B) \tag{7}$$

   By definition of nns and the complement:

$$\overline{\mathrm{nnf}(\neg\exists x B)} = \overline{\forall x\ \mathrm{nnf}(\neg B)} = \exists x\ \overline{\mathrm{nnf}(\neg B)} \tag{8}$$

   By the IH, $\exists x\ \mathrm{nnf}(B) = \exists x\ \overline{\mathrm{nnf}(\neg B)}$.
8. $A = \neg\forall x B$. Analogous to the previous case.

6 Exercise 6

Prove that $\vdash \text{nnf}(A \to B)$ implies $\vdash \{\overline{\text{nnf}(A)}, \text{nnf}(B)\}$.

Solution. $\text{nnf}(A \to B) \underset{\to\text{-def}}{=} \text{nnf}(\neg A \vee B) \underset{\text{nnf-def}}{=} \text{nnf}(\neg A) \vee \text{nnf}(B)$. Therefore,

$$\vdash \text{nnf}(A \to B) \Leftrightarrow \vdash \text{nnf}(\neg A) \vee \text{nnf}(B)$$

Case distinction:

- $\vdash \text{nnf}(\neg A)$. In the previous example, we proved that $\text{nnf}(\neg A) = \overline{\text{nnf}(A)}$. Therefore, $\vdash \text{nnf}(\neg A) \Leftrightarrow \vdash \overline{\text{nnf}(A)}$ and $\vdash \text{nnf}(\neg A) \vee \text{nnf}(B) \Leftrightarrow \vdash \overline{\text{nnf}(A)} \vee \text{nnf}(B)$. For the latter statement, it suffices to provide an LK-proof (since LK is complete):

$$\cfrac{\cfrac{P_A}{\vdash \overline{\text{nnf}(A)}}}{\vdash \overline{\text{nnf}(A)} \vee \text{nnf}(B)} \vee r$$

  By the assumption $\vdash \text{nnf}(\neg A)$, the proof P_A exists.

- $\vdash \text{nnf}(B)$. We again provide an LK-proof:

$$\cfrac{\cfrac{P_B}{\vdash \text{nnf}(B)}}{\vdash \overline{\text{nnf}(A)} \vee \text{nnf}(B)} \vee r$$

  By the assumption $\vdash \text{nnf}(B)$, the proof P_B exists.

- The third case is that $A = B$. In that case, neither $\vdash \text{nnf}(\neg A)$ nor $\vdash \text{nnf}A$ has to hold in general. The statement $\vdash \text{nnf}(\neg A) \vee \text{nnf}(A)$ is still true, however. First, we "unapply" the $\vee$-case of nnf: $\text{nnf}(\neg A) \vee \text{nnf}(A) \Rightarrow \text{nnf}(\neg A \vee A)$. Now let us observe that nnf preserves provability for any formula P, i.e. $\vdash P \Leftrightarrow \vdash \text{nnf}(P)$[2]. Applying this to our example, we get: $\vdash \text{nnf}(\neg A \vee A) \Rightarrow \vdash \neg A \vee A$. Now it suffices to give a proof of $\neg A \vee A$[3]:

$$\cfrac{A \vdash A}{\vdash \neg A \vee A} \neg r$$

Since the semantics of $\vdash \{\overline{\text{nnf}(A)}, \text{nnf}(B)\}$ are "$\vdash \overline{\text{nnf}(A)}$ or $\vdash \text{nnf}(B)$" and we have showed that either $A = B$, or at least one of P_A, P_B exists, the proof is complete.

---

[2]This can be easily proven by induction on the definition of nnf, but will be omitted here.

[3]This technique would have been powerful enough to cover the other two cases too, but the third case was a later addition.

7 Exercise 7

Prove that $\vdash A \to B$ and $\vdash A$ imply that $\vdash B$.

Solution. Instead of LK, we now use the sequent calculus from Definition 6 in the script.

We show the following[4]: $\vdash A \wedge (A \to B) \to B$. By the definition of $\to$, this is equivalent to: $\vdash \neg A \vee (A \vee \neg B) \vee B$.

Sequent proof:

1.	$\{A, \neg A, B\}$	Axiom
2.	$\{\neg A, \neg B, B\}$	Axiom
3.	$\{A \wedge \neg B, \neg A, B\}$	$\wedge$-rule
4.	$\{\neg A \vee (A \wedge \neg B), B\}$	$\vee$-rule
5.	$\{\neg A \vee (A \wedge \neg B) \vee B\}$	$\vee$-rule

Which was to be proven.

[4] $A, A \to B \vdash B$ is transformed into $\vdash A \wedge (A \to B) \to B$ by the deduction theorem.

8 Exercise 8

Complete the proof of Proposition 2 (soundness: $A \vdash \Gamma$ implies $A \models I(\Gamma)$).

Solution. By induction on the length of the sequent proof π. The base cases have already been described; the step cases remain. As induction hypothesis, we take the soundness up to the sequent Γ_{n-1}. For simplicity, we will denote the contents of Γ' as $\{F_1, \dots, F_k\}$. Case distinction on the derivation of the sequent Γ_n:

1. $\Gamma_n = \Gamma' \cup \{A \vee B\}$ and there exists a $j < n$ s.t. $\Gamma_j = \Gamma' \cup \{A, B\}$.

   Then, by definition, $I(\Gamma_j) = I(\Gamma') \vee I(\{A, B\}) = F_1 \vee F_2 \vee \cdots \vee F_k \vee A \vee B$. The interpretation I of Γ_n is $I(\Gamma_n) = I(\Gamma') \vee I(\{A \vee B\}) = I(\Gamma_j)$[5].

2. $\Gamma_n = \Gamma' \cup \{A \wedge B\}$ and there exist $j, l < n$ s.t. $\Gamma_j = \Gamma' \cup \{A\}$ and $\Gamma_l = \Gamma' \cup \{B\}$.

   Let M be a model of Γ_j[6]. There are two sub-cases:

   a) $M \models I(\Gamma')$. Then, since I interprets its argument as a disjunction, $M \models I(\Gamma' \cup X)$ for any X — especially for the case $X = \{A \wedge B\}$.

   b) $M \not\models I(\Gamma')$. By the IH (applied to Γ_j and Γ_l), it must hold that $M \models I(\{A\})$ and $M \models I(\{B\})$. Consequently, $M \models I(\{A \wedge B\})$ and $M \models I(\Gamma_n)$.

3. $\Gamma_n = \Gamma' \cup \{\exists x F(x)\}$ and there exists a $j < n$ s.t. $\Gamma_j = \Gamma' \cup \{\exists x F(x), F(t)\}$ for some term t.

   Take again a model M of Γ_j. Again, there are two sub-cases:

   a) $M \models I(\Gamma')$. See above.

   b) $M \not\models I(\Gamma')$. Then $M \models I(\{\exists x F(x), F(t)\})$. If $M \models I(\{\exists x F(x)\})$, then $M \models I(\Gamma_n)$ and we are done. If, instead, $M \models I(\{F(t)\})$, then $M, \phi_t^x \models I(\{F(x)\})$. By the semantics of $\exists x F(x)$, this implies that $M \models I(\{\exists x F(x)\})$ and we are, again, done.

4. $\Gamma_n = \Gamma' \cup \{\forall x F(x)\}$. This case is shown in the script.

5. There exist $j, l < n$ and a formula C s.t. $\Gamma_j = \Gamma_n \cup \{C\}$ and $\Gamma_l = \Gamma_n \cup \{\overline{C}\}$. By the IH, there exists a model M_j of Γ_j and a model M_l of Γ_l. If either is also a model Γ_n, we are done. If we cannot find such M_l, M_j s.t. at least one of them is a model of Γ_n, then there are models for both C and $\overline{C}$. This means that the system is inconsistent and therefore, no models of Γ_j, Γ_l exist, contradicting the IH!

[5] This is so because the set of first-order formulas $\mathcal{F}$, together with $\vee$, forms a monoid.

[6] The existence of a model is guaranteed by the IH.

9 Exercise 9

Let $L = \{P/1\}$ and $F = \forall x(P(x) \vee \neg P(x)$. Let $T = CL(\{F\})$. Show that T is incomplete.

Solution. Per Definition of "completene", T being incomplete means:

$$\exists S(F \nvdash S \wedge F \nvdash \neg S).$$

Since **LK** (without cut) is complete, it suffices to find a formula S s.t. not proof of either S or $\neg S$ exists. Let S be $\forall x P(x)$.

$$
\begin{prooftree}
\AxiomC{$\lightning$}
\UnaryInfC{$\vdash P(y), P(y)$}
\RightLabel{$t \leftarrow y$}
\UnaryInfC{$\vdash P(y), P(t)$}
\RightLabel{$\neg l$}
\UnaryInfC{$\neg P(t) \vdash P(y)$}
\AxiomC{$P(y) \vdash P(y)$}
\RightLabel{$t \leftarrow y$}
\UnaryInfC{$P(t) \vdash P(y)$}
\RightLabel{$\vee l$}
\BinaryInfC{$P(t) \vee \neg P(t) \vdash P(y)$}
\RightLabel{$\forall l$}
\UnaryInfC{$\forall x(P(x) \vee \neg P(x)) \vdash P(y)$}
\RightLabel{$\forall r$}
\UnaryInfC{$\forall x(P(x) \vee \neg P(x)) \vdash \forall x P(x)$}
\end{prooftree}
$$

We can easily see that the $\lightning$-branch cannot possibly be closed, whatever term we assign to t. The only way in which we could permute the rule applications is by applying $\forall l$ first. This, however, would be fruitless, as the subsequent application of $\forall r$ would then require the introduction of an eigenvariable which did not occur on the left side. Structural rules.

Since LK is also sound, we can provide a model both F and S and thereby prove that F cannot prove $\neg S$. Let D be some non-empty domain and I and interpretation with $I(P) = \emptyset$. Clearly, both F and S evaluate to true under I. Therefore, there exists a model of F and S and no sound calculus can prove $F \vdash \neg S$ (assuming consistency).

10 Exercise 10

Prove Proposition 3: Let A be a set of sentences [over $\mathcal{L}$]. Then $T = CL(A)$ is a theory, and A is an axiomatisation of T. Let S be a structure for $\mathcal{L}$. Then $\{F \mid S \models F, F \text{ sentence over } \mathcal{L}\}$ is a theory.

Solution. The first claim is that the deductive closure $T = CL(A)$ of A is a theory. A theory is defined as fulfilling the condition $CL(T) = T$. If we replace T with $CL(A)$, this statement becomes $CL(CL(A)) = CL(A)$, i.e. that CL is idempotent. $CL(A)$ is the deductive closure , defined as $\{F \mid S \models F, F \text{ sentence over } \mathcal{L}\}$ (we will henceforth assume "F sentence over $\mathcal{L}$" implicitly). Therefore:

$$\begin{gathered} CL(CL(A)) = CL(A) \\ CL(\{F \mid A \vdash F\}) = \{F \mid A \vdash F\} \\ \{G \mid \{F \mid A \vdash F\} \vdash G\} = \{F \mid A \vdash F\} \end{gathered}$$

Since $A \vdash A$, $CL(CL(A))$ can be expressed as

$$\{G \mid A \cup CL(A) \vdash G\} = \{F \mid A \vdash F\}.$$

Now we show the inclusion in both directions. First, let F be a formula in $CL(A)$, i.e. a proof $A \vdash F$ exists. To that proof, we can introduce $CL(A)$ via left weakening:

$$\cfrac{\cfrac{A \vdash F}{A, CL(A) - A \vdash F}\ \text{WL}}{A \cup CL(A) \vdash F}$$

Therefore, $CL(CL(A)) \supseteq CL(A)$. For the converse case, let G be a formula in $CL(CL(A))$, i.e. a proof $CL(A) \vdash G$ exists. Since such a proof can only make use of a finite number of assumptions, it is implied that a proof $F_1, \ldots, F_n \vdash G$ also exists (for some $\{F_1, \ldots, F_n\} \subseteq CL(A)$). We can now construct a proof $A \vdash G$ via iterated application of the cut-rule:

$$\cfrac{\cfrac{\cfrac{\cfrac{A \vdash F_2 \qquad \cfrac{A \vdash F_1 \qquad F_1, F_2, \ldots, F_n \vdash G}{A, F_2, \ldots, F_n \vdash G}\ \text{cut}}{A, A, F_3, \ldots, F_n \vdash G}\ \text{cut}}{\vdots}}{A, \ldots, A \vdash G}}{A \vdash G}\ CL^*$$

A proof of $A \vdash G$ implies that $G \in CL(A)$. Therefore $CL(CL(A)) \subseteq CL(A)$. The two inclusions thus shown imply $CL(CL(A)) = CL(A)$.

The second claim is that A is an axiomatisation of T. This is simply true by definition of "axiomatisation": $CL(A) = T$.

The third claim is that $\{F \mid S \models F, F \text{ sentence over } \mathcal{L}\}$ is a theory. This is simply the set of sentences which, under S, evaluate to true — let us call this set T and form its deductive closure:

$$\begin{aligned} CL(T) &= CL(\{F \mid S \models F\}) \\ &= \{G \mid \{F \mid S \models F\} \vdash G\} \end{aligned}$$

Let, as previously, $F_1, \ldots, F_n \vdash G$ be a proof in the end sequent of which only the finite subset $\{F_1, \ldots, F_n\} \subseteq T$ of assumptions occurs. Per the correctness of **LK**,

$$F_1, \ldots, F_n \vdash G \Leftrightarrow F_1, \ldots, F_n \models G.$$

$F_1, \ldots, F_n \models G$, in turn, means that any model of $F_1, \ldots, F_n$ is also a model of G. Since S is, by assumption, a model of $F_1, \ldots, F_n$, it must also be a model of G. Therefore, $\forall G. G \in CL(T) \Rightarrow G \in T$, i.e. $CL(T) \subseteq T$. The converse — $T \subseteq CL(T)$ — holds trivially, since $\ldots, F, \cdots \vdash F$ is an axiom of **LK**.

11 Exercise 11

A theory T is called maximally consistent if all theories $T' \supset T$ are inconsistent.

1. Prove that a theory is maximally consistent iff it is complete and consistent.

   Solution. Per the definition of consistency for a theory T, T is consistent iff there is no sentenced F s.t. both $T \vdash F$ and $T \vdash \neg F$. It is complete iff, for every sentence F, $T \vdash F$ or $T \vdash \neg F$ hold.

   The $\Rightarrow$-direction. Let T be a complete and consistent theory and let T' be its superset. T' being a superset implies that $F \in (T' - T)$ for some sentence F. Since F is not in T and T is complete, $T \vdash \neg F$ must hold. That, in turn, implies that $\neg F \in T'$, making T inconsistent.

   The $\Leftarrow$-direction. Let T be a maximally consistent theory. We show its completeness:

   Completeness Suppose there is a sentence F s.t. $F \notin T$ and $\neg F \notin T$. Then we could add either to T, obtaining a consistent superset. This contradicts the definition of maximal consistency.

   Consistency, in general, doesn't hold for maximally consistent theories as defined here. Suppose that $T = \{F | F \text{ is a sentence over } \mathcal{L}\}$. Then no theory T' which is a strict superset of T exists and therefore, "all" superset-theories of T are inconsistent, but T itself is not consistent. Consistency therefore has to be added to the definition of "maximally consistent".

2. Assume that T is a theory over a language $\mathcal{L}$ containing $=$. Prove that the following are equivalent: (a) T is inconsistent, (b) $T = \{F | F \text{ sentence over } \mathcal{L}\}$, (c) $(\exists x x \neq x) \in T$.

   Solution. I shall prove this via a cycle of implications.

   (a) $\Rightarrow$ (b) . If T is inconsistent, there exists a sentence G s.t. $G \in T$ and $\neg G \in T$. From the assumptions $G, \neg G$, any sentence F follows, as this LK-proof shows:

   $$\dfrac{G \vdash G, F}{G, \neg G \vdash F}\ \neg\text{-l}$$

   Since theories are, by definition, deductively closed, F must also be in T. That, in turn, is exactly the definition (b).

   (b) $\Rightarrow$ (c) The implication trivially holds, since $(\exists x . x \neq x)$ is a sentence of any language that includes $=$.

(c) $\Rightarrow$ (a) . $\exists x.x \neq x$ is defined as $\exists x.\neg(x = x)$, which can be further reduced to $\neg(\forall x.x = x)$. This is a direct negation of the reflexivity axiom: $\forall x.x = x$). Thereby, T is inconsistent.

12 Exercise 12

Prove the following properties of the deductive closure, where $(\Gamma_i)_{i\in\mathbb{N}}$ are sets of sentences:

1. $\Gamma_1 \subseteq \Gamma_2$ implies $CL(\Gamma_1) \subseteq CL(\Gamma_2)$,

   Solution. Let F be any sentence in $CL(\Gamma_1)$ and let P_F be a proof of F using the assumptions $\Delta \subseteq \Gamma_1$. Since $\Gamma_1 \subseteq \Gamma_2$, $\Delta \subseteq \Gamma_2$, and therefore, F can be proven in Γ_2 too.

   More generally (and to contrast the classical $\vdash$-relation to that of non-monotonic logics), we can add any set of assumptions to a proof P_F without diminishing the provability of sentences. It is because of that that the addition of new sentences to a theory cannot take away any element from its deductive closure.

2. $CL(CL(\Gamma_1)) = CL(\Gamma_1)$, **Solution.** See the proof in the solution to Exercise 10.

3. $\bigcup_{i\in\mathbb{N}} CL(\Gamma_i) \subseteq CL\left(\bigcup_{i\in\mathbb{N}} \Gamma_i\right)$,

   Solution. $\bigcup_{i\in\mathbb{N}} CL(\Gamma_i)$ is the union of the deductive closure of every $CL(\Gamma_i)$ ($i \in \mathbb{N}$). Therefore, it suffices to show that every sentence F which occurs in $CL(\Gamma_k)$ (for some $k \in \mathbb{N}$) is also in $CL\left(\bigcup_{i\in\mathbb{N}} \Gamma_i\right)$.

   Let k thus be an index and let F be a sentence in $CL(\Gamma_k)$. This implies that $\Gamma_k \vdash F$. Per the monotonicity discussed in 1., $\Gamma_k \vdash F$ implies that $\Delta \cup \Gamma_k \vdash F$ for any Δ and, specifically, $\left(\bigcup_{i\in\mathbb{N}-\{k\}} \Gamma_i\right) \cup \Gamma_k \vdash F \Leftrightarrow \left(\bigcup_{i\in\mathbb{N}} \Gamma_i\right) \vdash F$. Therefore, $F \in CL(\bigcup_{i\in\mathbb{N}} \Gamma_i)$.

13 Exercise 13

Show that for all $n, k \in \mathbb{N}$ the function $c_{k,n} : \mathbb{N}^n \to \mathbb{N} : (x_1, \ldots, x_n) \mapsto k$ is primitively recursive. Further show that the functions `neg`, `or`, `IfThenElse` are primitively recursive.

Solution. $c_{n,k}$ is the n-ary constant function which discards its arguments and returns k (note that k is a constant, not an argument). It can be realized primitively recursively thus:

$$c_{n,k} \equiv \mathrm{Cn}[\underbrace{s \circ \cdots \circ s}_{k \text{ times}}, z_n] \quad \text{where } f \circ g \equiv \mathrm{Cn}[f, g]$$

Equational reasoning proves the correctness of the definition:

$$\begin{aligned} c_{n,k}(x_1, \ldots, x_n) &= \mathrm{Cn}[\underbrace{s \circ \cdots \circ s}_{k \text{ times}}, z_n](x_1, \ldots, x_n) \\ &= \underbrace{s \circ \cdots \circ s}_{k \text{ times}}(z_n(x_1, \ldots, x_n)) \\ &= \underbrace{s \circ \cdots \circ s}_{k \text{ times}}(0) \\ &= k \end{aligned}$$

- $\mathtt{sgn} : \mathbb{N} \to \mathbb{N}$. `sgn` returns 0 if its argument is 0 and 1 otherwise. Its p.r. definition is:

  $$\mathtt{sgn} \equiv \mathrm{Pr}[z_0, c_{1,1} \circ id_2^1]$$

  For $x_1 = 0$, $\mathtt{sgn}(x_1) = z_0() = 0)$. For $x_1 \neq 0$, $\mathtt{sgn}(x_1)$ first projects out x_1 and then calls the constant function $c_{1,1}$, returning 1.

- $\mathtt{neg} : \mathbb{N} \to \mathbb{N}$. `neg` returns 1 if its argument is 0 and 1 otherwise. Its definition:

  $$\mathtt{neg} \equiv \mathrm{Pr}[c_{0,1}, z_2]$$

  For the base case of x_1, the constant function $c_{0,1}$ is returned. Otherwise, z_2 returns 0.

- $\mathtt{or} : \mathbb{N}^2 \to \mathbb{N}$. `or` returns 0 if both its arguments are 0 and 1 otherwise.

  $$\mathtt{or} \equiv \mathtt{sgn} \circ \mathtt{plus}$$

  `plus` is defined in the script and simply performs addition. It is easy to see that it, composed with `sgn`, delivers the correct result.

- $\texttt{IfThenElse} : \mathbb{N}^3 \to \mathbb{N}$. If $x_1 \neq 0$, x_2 is returned, otherwise x_3.

$$\begin{array}{l} \texttt{IfThenElse} \equiv \mathrm{Cn}[\texttt{if}, \mathrm{id}_3^3, \mathrm{id}_2^3, \mathrm{id}_1^3] \\ \text{where} \\ \quad \texttt{if} \equiv \mathrm{Pr}[\mathrm{id}_1^2, \mathrm{id}_2^4] \end{array}$$

  We first rearrange (x_1, x_2, x_3) into (x_3, x_2, x_1). Then we use Pr to perform a case distinction: if $x_1 = 0$, we select x_3 (the else-branch). Otherwise, we select x_2 (the if-branch).

14 Exercise 14

1. Show that if $S \subseteq \mathbb{N}^n$ is p.r., then $\mathbb{N}^n - S$ is p.r.

   Solution. Let χ_S be the p.r. characteristic function of S, i.e. $\chi_S(x_1, \ldots, x_n) = 1$ if $(x_1, \ldots, x_n) \in S$ and $\chi_S(x_1, \ldots, x_n) = 0$ otherwise. We define the characteristic function $\chi_{\mathbb{N}^n - S}$ for $\mathbb{N}^n - S$ thus:

   $$\chi_{\mathbb{N}^n - S} \equiv \texttt{neg} \circ \chi_S$$

   The correctness of this function is trivial: we simply execute χ_S and then flip the result with `neg`. Thereby, $\chi_{\mathbb{N}^n - S}$ will return 1 exactly for those tuples which are not in S and 0 for those which are.

2. Show that if $S, T \subseteq \mathbb{N}^n$ are p.r., then $S \cap T$ and $S \cup T$ are p.r.

   Solution. We again define the characteristic functions of these sets:

   $$\chi_{S \cup T} \equiv \mathrm{Cn}[\texttt{or}, \chi_S, \chi_T]$$

   $$\chi_{S \cap T} \equiv \mathrm{Cn}[\texttt{and}, \chi_S, \chi_T]$$
   where
   $$\texttt{and} \equiv \mathrm{Cn}[(\texttt{neg} \circ \texttt{or}), (\texttt{neg} \circ \chi_S), (\texttt{neg} \circ \chi_T)]$$

   Again, the correctness of these characteristic functions is trivial: we simply execute both of them. If χ_S or χ_T returns 1, the corresponding tuple is in $S \cup T$. If both χ_S and χ_T returns 1, the tuple is in $S \cap T$. `and` in the second case is just a translation of De Morgan's law $(A \wedge B) \Leftrightarrow \neg(\neg A \vee \neg B)$ into p.r. parlance[7].

3. Do these statement still hold if we replace "primitive recursive" by "total recursive"?

   Solution. Yes. For t.r. functions $f : D \to \mathbb{N}$ $dom(f) = D$ and therefore, the characteristic functions of S and T are defined for every tuple. The characteristic functions we constructed from these only perform p.r. transformations[8] on these and therefore still result in total functions.

[7] We can also define `and` more easily as `sgn` $\circ$ `mult`.

[8] The transformations can be seen as p.r. if we take the characteristic functions χ_S and χ_T as primitives exempt from the requirements of primitive recursiveness.

15 Exercise 15

1. For $x, y \in \mathbb{N}$, write the relation $| : \mathbb{N}^2 \to \mathbb{N}$ s.t. $|(x,y) = 1$ if there exists a $k \in \mathbb{N}$ with $x * k = y$ and $|(x,y) = 0$ otherwise.

   Solution.

   We first define the binary relation $=$, making use of **and** as defined above and of m as defined in the script:

   $$\begin{aligned} &= \equiv \mathrm{Cn}[(\texttt{neg} \circ \texttt{or}), m', m] \\ &\text{where} \\ &\quad m' = \mathrm{Cn}[m, \mathrm{id}_2^2, \mathrm{id}_1^2] \end{aligned}$$

   Through m and m', we compute $x - y$ and $y - x$ and, through $\texttt{neg} \circ \texttt{or}$, check that both result in 0. If so, $x = y$. We then move on to $|$:

   $$\begin{aligned} &| \equiv \mathrm{Cn}[\texttt{trymult}, \mathrm{id}_1^2, s \circ \mathrm{id}_2^2, \mathrm{id}_2^2] \\ &\text{where} \\ &\quad \texttt{trymult} \equiv \mathrm{Pr}[c_{2,0}, \texttt{rec}] \\ &\quad \texttt{rec} \equiv \mathrm{Cn}[\texttt{IfThenElse}, \texttt{check}, c_{4,1}, \mathrm{id}_4^4] \\ &\quad \texttt{check} \equiv \mathrm{Cn}[=, \mathrm{id}_2^4, \mathrm{Cn}[\texttt{mult}, \mathrm{id}_1^4, \mathrm{id}_3^4]] \end{aligned}$$

   In functional notation, the algorithm is written thus:

   ```
   |(x,y) = trymult(x,y,s(y))
      where trymult(x,y,0) = 0
            trymult(x,y,i+1) = if x*i = y then 1
                                    else trymult(x,y,i)
   ```

   First, we duplicate the larger number +1 ($s \circ \mathrm{id}_2^2$) and then use it as a counter[9]. At each step, `rec` checks whether $x * i = y$. If so, it returns 1 ($c_{4,1}$). Otherwise, it decrements the counter and recurses (id_4^4 in `rec`). If the counter reaches 0, `trymult` returns 0 ($c_{2,0}$).

2. Show that the sets $E = \{(x,x) | x \in \mathbb{N}\}$ and $D = \{(x,y) | x, y \in \mathbb{N}, |(x,y)\}$ are p.r.

   Solution.

   $$\chi_E \equiv =$$

   $$\chi_D \equiv |$$

   The previously defined p.r. functions $=$ and $|$ serve as the characteristic functions of E and D and therefore, E and D are p.r.

[9] The successor function s is used to cover the edge case $x_1 = 1$.

16 Exercise 16

- Show that if a set $S \in \mathbb{N}^2$ is p.r., then the set

$$\pi(S) = \{n \mid \forall m < n : (n, m) \in S\}$$

is p.r.

Solution. We give a p.r. characteristic function for $\pi(S)$. First, we define the template `forall`, which is instantiated with a predicate $P : \mathbb{N}^2 \to \mathbb{N}$. It takes a number n and returns 1 if $P(m, n) = 1$ for all $m < n$ and 0 otherwise[10]:

$$\begin{array}{l} \texttt{forall}_P \equiv \mathrm{Cn}[\mathrm{Pr}[c_{0,1}, \texttt{rec}], \mathrm{id}^1_1, \mathrm{id}^1_1] \\ \text{where} \\ \quad \texttt{rec} \equiv \mathrm{Cn}[\texttt{IfThenElse}, \mathrm{Cn}[P, \mathrm{id}^3_1, \mathrm{id}^3_1], \mathrm{id}^3_3, c_{3,0}] \end{array}$$

In functional notation, `forall` reads:

```
forall{P}(n) = forall'(n,n)
where forall'(n,0) = 1
      forall'(n,m+1) = if P(n,m) then forall'(n,m) else 0
```

We copy n (say, into m) and begin counting that copy m down to 0, checking at each stage whether $P(n, m)$ holds. If so, we recurse; if not, we halt and return 0. When m reaches 0, 1 is returned.

The characteristic function of $\pi(S)$ is now easily defined and its correctness follows from that of `forall`:

$$\chi_{\pi(S)} \equiv \texttt{forall}_{\chi_S}$$

- Show that the set of primes

$$\mathbb{P} = \{p \mid p > 1 \wedge \forall n \in \mathbb{N} : n|p \Rightarrow (n = 1 \vee n = p)\}$$

is p.r.

[10] If we want to be a bit more in line with recursion theory, we can define forall and `exists` in terms of `mult` and `plus`, just as we did with and and or:

$$\begin{array}{ll} \texttt{product}_P \equiv \mathrm{Cn}[\mathrm{Pr}[c_{2,1}, \mathrm{Cn}[\texttt{mult}, \mathrm{Cn}[P, \mathrm{id}^3_1, \mathrm{id}^3_2], \mathrm{id}^3_3]], \mathrm{id}^1_1, \mathrm{id}^1_1] & \texttt{product}_P = \prod_{1<m<n} P(m, n) \\ \texttt{sum}_P \equiv \mathrm{Cn}[\mathrm{Pr}[c_{2,0}, \mathrm{Cn}[\texttt{plus}, \mathrm{Cn}[P, \mathrm{id}^3_1, \mathrm{id}^3_2], \mathrm{id}^3_3]], \mathrm{id}^1_1, \mathrm{id}^1_1] & \texttt{sum}_P = \sum_{1<m<n} P(m, n) \\ \texttt{forall}_P \equiv \texttt{sgn} \circ \texttt{product}_P & \\ \texttt{exists}_P \equiv \texttt{sgn} \circ \texttt{sum}_P & \end{array}$$

Solution. Again, we can make good use of the `forall` template in giving a p.r. characteristic function for $\mathbb{P}$.

$$
\begin{aligned}
&\chi_{\mathbb{P}} \equiv \mathrm{Cn}[\texttt{and}, \texttt{gt1}, \texttt{forall}_{\texttt{factor}}] \\
&\text{where} \\
&\quad \texttt{gt1} \equiv \mathrm{Cn}[(\texttt{sgn} \circ m), \mathrm{id}_1^1, c_{1,1}] \\
&\quad \texttt{factor} \equiv \mathrm{Cn}[\texttt{IfThenElse}, |', \texttt{cond}, c_{2,1}] \\
&\quad |' \equiv \mathrm{Cn}[|, \mathrm{id}_2^2, \mathrm{id}_1^2] \\
&\quad \texttt{cond} \equiv \mathrm{Cn}[\texttt{or}, =, \mathrm{Cn}[=, c_{2,1}, \mathrm{id}_2^2]]
\end{aligned}
$$

$\chi_{\mathbb{P}}$ is a rather straightforward encoding of the definition of $\mathbb{P}$. `and` was defined above; `gt1` stands for "greater than 1", `factor` encodes the condition $n|p \Rightarrow (n = 1 \vee n = p)$, $|'$ is $|$ with its arguments flipped and `cond` encodes $(n = 1 \vee n = p)$.

The only thing of note is that, in the definition of $\mathbb{P}$, an unbounded universal quantification was used, whereas `forall` is bounded from above. This is not a problem: the quantified variable n is only used in the test $n|p$ and, of course, all factors of p are $\leq p$. Here, the bounded quantification of `forall` is sufficient.

We can quite easily see that the construction is correct and that thereby, $\mathbb{P}$ is p.r.

17 Exercise 17

Prove the Theorem 10 formally: let $S, \overline{S} \subseteq \mathbb{N}^n$ be r.e. sets. Then, $S, \overline{S}$ are recursive sets.

Solution. Per the definition of "recursive set" in the script, there exist recursive functions $\varphi_S, \varphi_{\overline{S}} : \mathbb{N} \to \mathbb{N}^n$ which, given an index y, will produce the yth tuple/element of the corresponding set.

I will provide the recursive characteristic function χ_S for S. The one for $\overline{S}$ is fully analogous.

$$\begin{aligned}
&\chi_S \equiv \mathrm{Cn}[\texttt{IfThenElse}, \mathrm{Cn}[=, \mathrm{id}_1^1, (\varphi_S \circ \texttt{firstHit})], c_{1,1}, c_{1,0}] \\
&\text{where} \\
&\quad \texttt{firstHit} \equiv \mathrm{Mn}[\mathrm{Cn}[\mathrm{neg} \circ \mathit{or}, \texttt{inS}, \texttt{inScomp}]] \\
&\quad \texttt{inS} \equiv \mathrm{Cn}[=, \mathrm{id}_1^2, \varphi_S \circ \mathrm{id}_2^2] \\
&\quad \texttt{inScomp} \equiv \mathrm{Cn}[=, \mathrm{id}_1^2, \varphi_{\overline{S}} \circ \mathrm{id}_2^2]
\end{aligned}$$

Partial correctness Using Mn, we find the first index y for which $x_1 = \varphi_S(y)$ or $x_1 = \varphi_{\overline{S}}(y)$ (`firstHit`). Having obtained y, we check $x_1 = \varphi_S(y)$. If that check returns 1, we know that x_1 is in S and return 1. Otherwise, Mn must have halted because of $x_1 = \varphi_{\overline{S}}(y)$ and, correspondingly, we return 0.

Termination Because x_1 must either be in S or $\overline{S}$ and because both of these sets are recursively enumerable, the call to Mn will always terminate: we enumerate all elements of both sets in parallel and are bound to encounter x_1 after finite time.

18 Exercise 18

Show that if t is a ground term, then there is a $k \in \mathbb{N}$ such that $\mathbf{Q} \vdash\ t = k$.

Solution. The theory $\mathbf{Q}$ and the here used language $L = (\mathbb{N}, \{0\backslash 0, s\backslash 1, +\backslash 2, \cdot\backslash 2, = \backslash 2, < \backslash 2\})$ are defined in section 7 "Formal arithmetic". We prove the proposition via structural induction. To avoid confusion, we'll denote $\mathbf{Q}$'s language-level equality as $=$ and our syntactic equality as $\equiv$. Note that, in addition to $\mathbf{Q}$'s axioms, we also need the equality axioms Refl, Symm, Trans and Ext. Ext is the axiom schema of extensionality and allows us to replace a subterm x_i of $f(x_1, \ldots, x_n)$ with a subterm y_i if $x_i = y_i$ (for all $f \in L$ and all $1 \leq i \leq n$).

$$\begin{aligned}
&\text{Refl} \equiv [\forall x]\, x = x,\\
&\text{Symm} \equiv [\forall x, y]\, x = y \Rightarrow y = x,\\
&\text{Trans} \equiv [\forall x, y, z]\, x = y \wedge y = z \Rightarrow x = z,\\
&\text{Ext}_{f,i} \equiv [\forall x_1, \ldots, x_n, y_i]\, x_i = y_i \Rightarrow f(x_1, \ldots, x_i, \ldots, x_n) = f(x_1, \ldots, y_i, \ldots, x_n).
\end{aligned}$$

Base case. $t \equiv s(\cdots s(0) \cdots) \equiv s^n(0)$. This follows from Refl: $s^n(0) = s^n(0)$.

Step case "$+_1$". $t \equiv t' + 0$. IH: $t' = s^n(0)$.
Per (3), $t' + 0 = t'$.
Per Trans, $(t' + 0 = t') \wedge (t' = s^n(0)) \Rightarrow (t' + 0 = s^n(0))$.
Therefore, $t' + 0 = s^n(0)$.

Step case "$+_2$". $t \equiv t' + s(r)$. IH: $t' = s^n(0)$ and $r = s^m(0)$ and $s^n(0) + s^m(0) = s^{n+m}(0)$.
Per (4), $t' + s(r) = s(t' + r)$.
Now we apply Symm two times, followed by $\text{Ext}_{+,1}$ and $\text{Ext}_{+,2}$, instantiating x_i,y_i with the parts of the IH:

$$\begin{array}{lcl}
t' = s^n(0) & \Rightarrow & s^n(0) = t'\\
r = s^m(0) & \Rightarrow & s^m(0) = r\\
s^n(0) = t' & \Rightarrow & s^n(0) + s^m(0) = t' + s^m(0)\\
s^m(0) = r & \Rightarrow & s^n(0) + s^m(0) = t' + r
\end{array}$$

We now know that $s^n(0) + s^m(0) = t' + r$. Applying Symm, we get $t' + r = s^n(0) + s^m(0)$. We again apply $\text{Ext}_s, 1$ to the term $s(t' + r)$:

$$t' + r = s^n(0) + s^m(0) \quad \Rightarrow \quad s(t' + r) = s(s^n(0) + s^m(0))$$

We apply $\text{Ext}_s, 1$ again to this, using the third part of the IH:

$$s^n(0) + s^m(0) = s^{n+m}(0) \quad \Rightarrow \quad s(s^n(0) + s^m(0)) = s(s^{n+m}(0))$$

Through repeated application of Trans, we get

$$t \equiv t' + s(r) = s(t' + r) = s(s^n(0) + s^m(0)) = s(s^{n+m}(0)) \equiv s^{n+m+1}(0)$$

Step case "$\cdot_1$". $t \equiv t' \cdot 0$. IH: $t' = s^n(0)$.
Per (5), $t' + 0 = 0$.
Per Trans, $(t' \cdot 0 = 0) \wedge (t' = s^n(0)) \Rightarrow (t' \cdot 0 = 0)$.
Therefore, $t' \cdot 0 = 0$.

Step case "$\cdot_2$". $t \equiv t' \cdot s(r)$. IH: $t' = s^n(0)$ and $r = s^m(0)$ and $s^n(0) \cdot s^m(0) = s^{n \cdot m}(0)$ and $s^{n \cdot m}(0) + s^n(0) = s^{n \cdot (m+1)}(0)$.
Per (6), $t' \cdot s(r) = (t' \cdot r) + t$.

This case is basically analogous to $+_2$. We again apply Sym and $\text{Ext}_{\cdot,1}$, $\text{Ext}_{\cdot,2}$:

$$\begin{array}{lll}
t' = s^n(0) & \Rightarrow & s^n(0) = t' \\
r = s^m(0) & \Rightarrow & s^m(0) = r \\
s^n(0) = t'(0) & \Rightarrow & s^n(0) \cdot s^m(0) = t' \cdot s^m(0) \\
s^m(0) = r & \Rightarrow & s^n(0) \cdot s^m(0) = t' \cdot r
\end{array}$$

Through Sym, we get $t' \cdot r = s^n(0) \cdot s^m(0)$ and, through the IH and Trans, $t' \cdot r = s^{n \cdot m}(0)$. We now apply $\text{Ext}_{+,1}$, $\text{Ext}_{+,2}$ to $(t' \cdot r) + t'$:

$$\begin{array}{lll}
t' \cdot r = s^{n \cdot m}(0) & \Rightarrow & (t' \cdot r) + t' = s^{n \cdot m}(0) + t' \\
t' = s^n(0) & \Rightarrow & (t' \cdot r) + t' = s^{n \cdot m}(0) + s^n(0)
\end{array}$$

We apply the last part of the IH and Trans to get

$$(t' \cdot r) + t' = s^{n \cdot m}(0) + s^n(0) = s^{n \cdot (m+1)}(0)$$

The induction hypotheses (especially $s^n(0) + s^m(0) = s^{n+m}(0)$ and $s^n(0) \cdot s^m(0) = s^{n \cdot m}(0)$) might seem problematic, but these are always indeed always proven in the last lines of "$+_2$" and "$\cdot_2$". The hypothesis $s^{n \cdot m}(0) + s^n(0) = s^{n \cdot (m+1)}(0)$ can be derived from these two if we substitute suitable values for n and m.

The induction in this proof is not part of **Q**, but works on a metalinguistical level. Since t is a concrete (but arbitrary) term, this is not a problem, however: for any given t, we can unfold the definitions of **Q**'s formulas and obtain a finitely long proof which, *is constructed by induction*, but isn't inductive itself.

19 Exercise 19

Show that if

1. If s, t are ground terms, then either $\mathbf{Q} \vdash\ s = t$ or $\mathbf{Q} \vdash\ s \neq t$.

   Solution. In the previous example, we showed that all ground terms s, t are equal $(=)$ to terms of the form $s^n(0), s^m(0)$. If $n = m$, then, through Refl, $\mathbf{Q} \vdash\ s = t$. Suppose, on the other hand, that $n \neq m$ and, w.l.o.g., $n < m$. We can give an indirect inductive proof:

Step case. If $s(s^{n-1}(0)) = s(s^{m-1}(0))$, then, per (2), $s^{n-1}(0) = s^{m-1}(0)$.

Base case. Since we assumed $n < m$, we must at some point come to the assertion that $0 = s^{m-k}(0)$ (for some k). However, this contradicts (1). Consequently, $s^n(0) = s^m(0)$ cannot hold if $n \neq m$ and thus, $s^n(0) \neq s^m(0)$.

   We can encode this proof in $\mathbf{Q}$ through the following formula:

$$s^n(0) = s^m(0) \Rightarrow s^{n-1}(0) = s^{m-1}(0) \Rightarrow \cdots \Rightarrow 0 = s^{m-k}(0)$$

   By using $0 \neq s(x)$, we show $\neg(0 = s^{m-1}(0))$ and therefrom "roll up" the chain of implications until we get $\neg(s^n(0) = s^m(0))$.

2. If s, t are ground terms, then either $\mathbf{Q} \vdash\ s > t$ or $\mathbf{Q} \vdash\ s = t$ or $\mathbf{Q} \vdash\ s < t$.

   Solution. This follows immediately from (9). The semantics of $Q \vdash\ s < t \vee s = t \vee s > t$ are precisely "$\mathbf{Q} \vdash\ s > t$ or $\mathbf{Q} \vdash\ s = t$ or $\mathbf{Q} \vdash\ s < t$".

20 Exercise 20

Prove Proposition 10: for all $k \in \mathbb{N}$ we have $\mathbf{Q} \vdash\ [\forall x]\, x < k \Leftrightarrow (x = 0 \vee x = 1 \vee \cdots \vee x = (k-1))$.

Solution. We can unroll (8) by repeatedly instantiating y to attain this formula. We can construct the proof inductively, in a sense: we construct a proof for $k = 1$ and, having a proof of $k = n$, we can construct a proof for $k = n+1$. Merely the *construction* of the proof is inductive, the proof itself won't be.

Optional base case. It's not clear whether the formula is defined for $k = 0$, but we can do so if we assume the empty disjunction to be $\bot$ (the neural element of $\vee$).

$$
\begin{prooftree}
\AxiomC{$\mathbf{Q}', x_0 < 0 \vdash\ \bot, x_0 < 0$}
\RightLabel{$\neg l$}
\UnaryInfC{$\mathbf{Q}', \neg(x_0 < 0), x_0 < 0 \vdash\ \bot$}
\RightLabel{$\forall l$}
\UnaryInfC{$\mathbf{Q}, x_0 < 0 \vdash\ \bot$}
\RightLabel{$\Rightarrow r$}
\UnaryInfC{$\mathbf{Q} \vdash\ x_0 < 0 \Rightarrow \bot$}
\AxiomC{(Note: $\vdash\ A \equiv \top \vdash\ A$)}
\UnaryInfC{$\vdash\ \top, x_0 < 0$}
\RightLabel{$\neg l$}
\UnaryInfC{$\bot \vdash\ x_0 < 0$}
\RightLabel{$\Rightarrow r$}
\UnaryInfC{$\vdash\ \bot \Rightarrow x_0 < 0$}
\RightLabel{$\wedge r$}
\BinaryInfC{$\mathbf{Q} \vdash\ x_0 < 0 \Rightarrow \bot \wedge \bot \Rightarrow x_0 < 0$}
\RightLabel{def. $\Leftrightarrow$}
\UnaryInfC{$\mathbf{Q} \vdash\ x_0 < 0 \Leftrightarrow \bot$}
\RightLabel{$\forall r$}
\UnaryInfC{$\mathbf{Q} \vdash\ [\forall x]\, x < 0 \Leftrightarrow \bot$}
\end{prooftree}
$$

As we can see, even this case is quite cumbersome; I will therefore sketch the other two somewhat more informally.

Base case. Let $k = 1 = s(0)$. We have to construct a proof s.t.

$$\mathbf{Q} \vdash\ [\forall x]\, x < s(0) \Leftrightarrow x = 0$$

We can instantiate (8) with $y \to s(0)$. It becomes:

$$[\forall x]\, x < s(0) \Leftrightarrow x < 0 \vee x = 0$$

From (7), we know that $x < 0$ is false and thus, if we appropriately unpack and re-pack the formula above, we get $[\forall x]\, x < s(0) \Leftrightarrow x = 0$, which is what we wanted.

1. Let $k = n + 1 = s^{n+1}(0)$ and let us assume the existence of a proof P_n for $k = n$ as the IH — that is:

$$\frac{P_n}{\mathbf{Q} \vdash\ [\forall x]\, x < s^n(0) \Leftrightarrow (x = 0 \vee \cdots \vee x = s^{n-1}(0))}$$

From this, we construct a proof P_{n+1} by instantiating (8) with $y \to s^{n+1}(0)$, getting

$$[\forall x]\, x < s^{n+1}(0) \Leftrightarrow x < s^n(0) \vee x = s^n(0)$$

Now we use the IH and replace $s^n(0)$ with $(x = 0 \vee \cdots \vee x = s^{n-1}(0))$, again by unpacking and re-packing the formula according to the rules of $\Leftarrow$ and $\forall r$. We get:

$$[\forall x]\, x < s^{n+1}(0) \Leftrightarrow x = 0 \vee \cdots \vee x = s^{n-1}(0) \vee x = s^n(0)$$

If we write this procedure down as an LK proof, we get P_{n+1} s.t.

$$\frac{P_{n+1} \text{ (containing } P_n)}{[\forall x]\, x < s^{n+1}(0) \Leftrightarrow x = 0 \vee \cdots \vee x = s^{n-1}(0) \vee x = s^n(0)}$$

21 Exercise 21

Prove that if F is a ground formula, then either $\mathbf{Q} \vdash F$ or $\mathbf{Q} \vdash \neg F$.

Solution. We can proceed via structural induction. The base cases consists of atoms of the form $s = t$ or $s < t$, since $=$ and $<$ are the only two predicates in L. The step cases are formed via logical connectives.

Base case "=". Let F be an atom of the form $s = t$. In Exercise 19, showed that, if s, t are ground terms, then $\mathbf{Q} \vdash s = t$ or $\mathbf{Q} \vdash s \neq t$.

Base case "<". Let F be an atom of the form $s < t$. Also in Exercise 19, we showed that $\mathbf{Q} \vdash s < t$ or $\mathbf{Q} \vdash s = t$ or $\mathbf{Q} \vdash t < s$. Two sub-cases:

- If $\mathbf{Q} \vdash s < t$, then $\mathbf{Q} \vdash F$.
- If $\mathbf{Q} \vdash s = t$ or $\mathbf{Q} \vdash t < s$, then, $s \neq 0 \wedge \cdots \wedge s \neq t - 1$[11]. Per Exercise 20, this is a direct negation of $s < t$. Thereby, we can prove $s \not< t$.

Step case. Let F be $\neg F_1$, $F_1 \vee F_2$ or $F_1 \wedge F_2$. Without quantifiers, any complete, propositional calculus (like LK) suffices to show $\mathbf{Q} \vdash F$ or $\mathbf{Q} \vdash \neg F$.

Prove Proposition 11: If $F(x)$ is a formula with x being the only free variable, then $\mathbb{N} \models [\exists x]\, F(x)$ iff $\mathbf{Q} \vdash [\exists x]\, F(x)$.

Solution.

$\Rightarrow$-direction. Suppose that $\mathbb{N} \models [\exists x]\, F(x)$. Then there exists a witness n s.t. $F(n)$ is true. Since $F(n)$ is ground, there exists a proof P_F for $F(n)$ with the theory $\mathbf{Q}$, as we showed above. That proof can be transformed into one of $[\exists x]\, F(x)$ thus:

$$\dfrac{\dfrac{P_F}{\mathbf{Q} \vdash F(n)}}{\mathbf{Q} \vdash F(x)}\ \exists r$$

$\Leftarrow$-direction. Suppose that $\mathbf{Q}$ is consistent. Since we know that LK is sound and complete, it follows that LK with theory $\mathbf{Q}$ is also sound — that is, if $\mathbf{Q} \vdash [\exists x]\, F(x)$, then $\mathbb{N} \models [\exists x]\, F(x)$. $\mathbf{Q}$ is consistent if it has a model; we assume $\mathbb{N}$ to be such a model, although no proof of that exists in $\mathbf{Q}$ itself.

[11] This is so because otherwise, there would be two distinct numbers n_1, n_2 s.t. $n_1 \neq n_2$ and $s = n_1$ and $s = n_2$. Applying Trans would then lead to a contradiction.