SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Extracting Computational Content From Proofs

av

Alice Brolin

2021 - No K49

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET, 106 91 STOCKHOLM

Extracting Computational Content From Proofs

Alice Brolin

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Peter LeFanu Lumsdaine

2021

Abstract

Kurt Gödel developed a translation, called the Dialectica translation, from Heyting Arithmetic into a type system called T. This type system can be described by a lambda calculus. I present Gödel's translation and then, to get a sense of how it functions in practice, I apply it to two theorems in Heyting Arithmetic. The first theorem is about $\leq$ being a total order on the natural numbers. When I apply the Dialectica translation I get lambda terms for subtraction and for the characteristic function of $\leq$. The second theorem is about there being arbitrarily large prime numbers. This gives me under the translation a lambda term for a function that from any natural number can produce a prime number bigger than that number. I also implement these lambda terms as functions in the programming language Haskell.

Contents

1 Introduction

Intuitionistic logic is a form of logic that uses the same language as classical logic. However, in intuitionistic logic we do not have the law of the excluded middle $\phi \vee \neg\phi$. This means that our normal interpretation used for classical logic, where each formula is given a truth value of true or false and formulas can be checked with truth tables, is not sufficient for intuitionistic logic. Instead, we can use the BHK-interpretation where formulas are interpreted by their construction so e.g. $\phi \wedge \psi$ is a construction of ϕ and a construction of ψ whereas $\phi \to \psi$ is a method of turning a construction of ϕ into a construction of ψ. We see that this gives a more algorithmic interpretation of proofs. [SU06, p.28-29]

We will look at an intuitionistic theory of the natural numbers called Heyting Arithmetic (HA). HA is a first order theory; that is we are allowed to quantify over numbers but not over anything else (like sets or functions). It is an analogue of Peano Arithmetic (PA) in intuitionistic logic. PA and HA use the same axioms but the difference is that for PA the underlying logic is classical first order logic whereas for HA the underlying logic is intuitionistic first order logic, so the law of the excluded middle can be used in PA but not in HA.[AF95, p.5]

Gödel developed a system called T and a translation from HA to T called the Dialectica translation. This allowed him to prove the consistency of HA using T. [AF95, p.2] The main benefit of formulas in T compared to HA is that they are essentially quantifier free. To do this he used a many-sorted system, that is whereas HA only allows talking about one type of object (numbers), T allows talking about different types of objects. [AF95, p.6] These can be numbers, functions between numbers and also functions between functions. These functions can be described by a version of typed lambda calculus[AF95, p.8-9]. This lambda calculus can be seen as a sort of limited programming language.

In this paper I will present the translation and translate some theorems in HA to T using the Dialectica translation. Using the algorithmic content in the proofs I will find the related functions arising from the translation. I will also implement these functions in the programming language Haskell.

2 Gödel's Dialectica Translation

2.1 Heyting Arithmetic

Heyting Arithmetic can be described as a theory with one constant 0, one unitary function S (the successor function) and two binary functions $+, \cdot$. To describe HA we can use the following axioms (from [SU06, p.233-234]):

1. $\forall a, b : S(a) = S(b) \to a = b$
2. $\forall a : \neg S(a) = 0$
3. $\forall a : a + 0 = a$
4. $\forall a, b : a + S(b) = S(a + b)$
5. $\forall a : a \cdot 0 = 0$
6. $\forall a, b : a \cdot S(b) = a \cdot b + a$
7. For any proposition ϕ the axiom $\phi(0) \to \forall n : (\phi(n) \to \phi(S(n))) \to \forall n : \phi(n)$

Here the last item in the list is the induction scheme.

As shown in appendix A one can easily derive the basic rules of arithmetic, e.g. commutativity and associativity, in HA.

2.2 System T

The Dialectica translation uses a type system called system T. This description of T as a typed lambda calculus is from [AF95, p.6-7]. T has one primitive type Nat representing natural numbers and further types that are defined inductively by saying that if σ,τ are types then we have a function type $\sigma \to \tau$ (representing functions from σ to τ) .

All terms in system T have a unique type. If τ is the type of a term t we write $t : \tau$. The terms are:

1. Infinite sets of variables for each type.
2. $0 :$ Nat representing the natural number 0.
3. $S :$ Nat $\to$ Nat representing the successor function.
4. For any term t of type τ and type σ we can form the term $\lambda x^\sigma.t : \sigma \to \tau$ representing abstraction.
5. If t is a term of type $\sigma \to \tau$ and s is a term of type σ then ts (or $t(s)$) is a term of type τ representing the value of t applied to s.

6. For every type σ we get a term $R_\sigma : \sigma \to (\text{Nat} \to \sigma \to \sigma) \to \text{Nat} \to \sigma$ representing recursion.

I will write $t(s_1, s_2, ..., s_n)$ for $(((ts_1)s_2)...)s_n$, and $\lambda x_1^{\sigma_1}...x_n^{\sigma_n}.t$ for $\lambda x_1^{\sigma_1}.(\lambda x_2^{\sigma_2}.(...\lambda x_n^{\sigma_n}.t))$. Also I will write $t[s/x]$ for the term t with all occurrences of x swapped for s (and if necessary a renaming of bound variables in t to avoid capture).

Computation in system T is carried out by substituting occurrences of $(\lambda x^\sigma.t)s$ with $t[s/x]$, $R_\sigma(s, t, 0)$ with s and $R_\sigma(s, t, S(n))$ with $t(n, R_\sigma(s, t, n))$ [AF95, p.9]. The strong normalization theorem for system T says that for any term t this process will always result in a unique normal term (i.e. a term where no more substitutions can be done)[SU06, p.256].

Basic operators in T

We see from the axioms that addition and multiplication are defined recursively. Thus we can in system T define:

$$a + b := R_{\text{Nat}}(a, \lambda n^{\text{Nat}} x^{\text{Nat}}.S(x), b)$$

$$ab := R_{\text{Nat}}(0, \lambda n^{\text{Nat}} x^{\text{Nat}}.x + a, b)$$

It will also be useful to define some Boolean operators. For that I identify 0 with “True” and 1 with “False”. Then I can define the operators “and”, “not” as well as “or” by $a\&b = ab$, $!a = R_{\text{Nat}}(1, \lambda n^{\text{Nat}} x^{\text{Nat}}.0, a)$ and $a|b =!(!a\&!b)$ Finally I can define a “if then else” operator as $a?s : t = R_{\text{Nat}}(t, \lambda n^{\text{Nat}} x^{\text{Nat}}.s, a)$.

2.3 The Translation

The Dialectica translation (as given in [G94, p.249]) works as follows. For any formula $\phi = \phi(z)$ in HA we get a formula $\phi' = \exists x \forall y\ \alpha(x, y, z)$ in system T where x, y and z are lists of variables and α does not have any quantifiers and only contains the variables x, y, z. The translation is defined inductively by first taking $\phi' = \phi$ if ϕ contains no logical operators. Then if we have formulas ϕ, ψ such that we have already defined $\phi' = \exists x \forall y\ \alpha(x, y, z), \psi' = \exists u \forall v\ \beta(u, v, w)$, we define:

1. $(\phi \wedge \psi)' = \exists x, u \forall y, v\ (\alpha(x, y, z) \wedge \beta(u, v, w))$

2. $(\phi \vee \psi)' = \exists c, x, u \forall y, v\ ((c = 1 \wedge \alpha(x, y, z)) \vee (c = 0 \wedge \beta(u, v, w)))$

3. $(\forall n : \phi(z))' = \exists X \forall y, n\ \alpha(X(n), y, z)$

4. $(\exists n : \phi(z))' = \exists n, x \forall y\ \alpha(x, y, z)$

5. $(\phi \to \psi)' = \exists U, Y \forall x, v\ (\alpha(x, Y(x, v), z) \to \beta(U(x), v, w))$

6. $(\neg\phi)' = \exists Y \forall x\ \neg\alpha(x, Y(x), z)$

In the base case we need to translate the numbers and $S, +, \times$ in to their counterparts in system T.[AF95, p.10] We see that in the case of disjunction we add a constant that describes which case we are in. For universal quantification we need to swap the place of a "for all" and an "exists". If for all n there exists some x satisfying some proposition then there should be a function $X(n)$ that maps an n to such a x, which is the idea behind that rule.

In the case of implication we use that $\exists x \phi(x) \to \exists u \psi(u)$ means that if there exists a witness of ϕ we should be able to find a witness of ψ and $\forall y \phi(y) \to \forall v \psi(v)$ means that if there is a v such that ψ is false then we need a y such that ϕ is false[G94, p.249]. This is because $\forall y \phi(y) \to \forall v \psi(v)$ implies it's contrapositive $\neg \forall v \psi(v) \to \neg \forall y \phi(y)$. Combining these we get that with ϕ', ψ' as above we can transform $\phi \to \psi$ into $\exists U \forall x \exists Y \forall v (\alpha(x, Y(v), z) \to \beta(U(x), v, w))$ and then use the translation rules for quantification to get 5. The rule for negation comes from the rule for implication and the fact that $\neg\phi$ is equivalent to $\phi \to \perp$[G94, p.249].

2.4 Proofs in T

Proofs in T use the inference rules of propositional logic and for equality. The axioms in HA are replaced with quantifier free axiom schemes. For example the axiom $\forall a : \neg S(a) = 0$ is replaced with the axiom scheme, that for any a : Nat we have $\neg S(a) = 0$. Instead of the the normal induction scheme we have the rule that for any formula $\alpha(x^{\mathrm{Nat}})$ and term t : Nat that we from $\alpha(0)$ and $\alpha(x) \to \alpha(S(x))$ can deduce $\alpha(t)$.[AF95, p.9]

Gödel's main result about the Dialectica translation was that for any formula ϕ, where $\phi' = \exists x \forall y \alpha(x, y, z)$, if ϕ is provable in HA then there is a term t such that $\alpha(t, y, z)$ is provable in T.[AF95, p.13]

To prove that result we can show by induction on the size of a proof, that if HA proves $\sigma_1, ..., \sigma_n \vdash \phi$, where $\sigma'_k = \exists x_k, \forall y_k \gamma_k(x_k, y_k, w_k)$ then there are terms $t_1, ..., t_n, s$ such that T proves

$$(\gamma_1(x_1, t_1(x_1, ..., x_n, y), w_1) \wedge ... \wedge \gamma_n(x_n, t_n(x_1, ..., x_n, y), w_n)) \to \alpha(s(x_1, ..., x_n), y, z),$$

for arbitrary $x_1, ..., x_n, y$ and for choice of $w_1, ..., w_k, z$ (being mindful of the fact that these variables may overlap).

Most of the axioms follow immediately from their counterpart in HA. For the induction scheme one uses the fact that T has a term for recursion[AF95, p.13].

For the inductive step we assume that the translation works for all shorter proofs and then verify that it still works when we apply each inference rule.

For $\to$-introduction we have a proof of $\sigma_1, ..., \sigma_n, \phi \vdash \psi$ and get a proof of $\phi \vdash \phi \to \psi$. Here we say $\psi' = \exists x \forall y : \beta(x, y, z_2)$. Thus we need to show that if there are $t'_1, ..., t'_{n+1}, s'$, such that for all $x'_1, ..., x'_{n+1}, y'$ T proves

$$\begin{aligned}(\gamma_1(x'_1, t'_1(x'_1, ..., x'_{n+1}, y'), w_1) \wedge ... \wedge \gamma_n(x'_n, t'_n(x'_1, ..., x'_{n+1}, y'), w_n) \\ \wedge\, \alpha(x'_{n+1}, s(x'_1, ..., x'_{n+1}, y'), z_1)) \to \beta(s'(x'_1, ..., x'_{n+1}, y'_1), y', z_2),\end{aligned}$$

then there are $t_1, ..., t_n, t, s_1, s_2$ such that for all $x_1, ..., x_n, y_1, y_2$ one can prove

$$(\gamma_1(x_1, t_1(x_1, ..., x_n, y_1, y_2), w_1) \wedge ... \wedge \gamma_n(x_n, t_n(x_1, ..., x_n, y_1, y_2), w_n)) \\ \to (\alpha(y_1, s_1(x_1, ..., x_n, y_1, y_2), z_1) \to \beta(s_2(x_1, ..., x_n, y_1), y_2, z_2)).$$

Clearly one can simply take $x'_1 = x_1, ..., x'_n = x_n, x'_{n+1} = y_1, y' = y_2$ and define $t_1 = t'_1, ..., t_n = t'_n, s_1 = t'_{n+1}, s_2 = s'$.

For inference rules with multiple premises we use the following lemmas:

Lemma 2.1. *Assume I have formulas $\gamma_1, ..., \gamma_n$ and terms $t'_1, ..., t'_n, t''_1, ..., t''_n$. Define $\overline{x} = x_1, ..., x_n$,*

$$\gamma'(\overline{x}, y) = \gamma_1(x_1, t'_1(\overline{x}, y)) \wedge ... \wedge \gamma_n(x_n, t'_n(\overline{x}, y'))$$

and

$$\gamma''(\overline{x}, y) = \gamma_1(x_1, t''_1(\overline{x}, y'')) \wedge ... \wedge \gamma_n(x_n, t''_n(\overline{x}, y'')).$$

Then there are terms $t_1, ..., t_n$ such that T proves $\gamma(\overline{x}, y) \to \gamma'(\overline{x}, y) \wedge \gamma''(\overline{x}, y)$, where $\gamma(\overline{x}, y) = \gamma_1(x_1, t_1(\overline{x}, y)) \wedge ... \wedge \gamma_n(x_n, t_n(\overline{x}, y))$.

Proof. One can show that for any formula $\gamma^*(x, y)$ there is a term $t_{\gamma^*}(x, y)$ such that T proves $t_{\gamma^*}(x, y) = 0 \leftrightarrow \gamma^*(x, y)$ [AF95, p.12]. Let

$$t_k(\overline{x}, y) = t_{\gamma'}(\overline{x}, y) ? t'_k(\overline{x}, y) : t''_k(\overline{x}, y).$$

Now assume $\gamma(\overline{x}, y)$. Next either $t_{\gamma'}(\overline{x}, y) = 0$ or not.

- If $t_{\gamma'}(\overline{x}, y) = 0$ then since $t_{\gamma'}(\overline{x}) \to \gamma'(\overline{x})$ we get $\gamma'(\overline{x}, y)$. Also if $t_{\gamma'}(\overline{x}, y) = 0$ then $t_k(\overline{x}, y) = t''_k(\overline{x}, y)$ so $\gamma(\overline{x}, y)$ becomes $\gamma''(\overline{x}, y)$. Hence we get $\gamma'(\overline{x}, y) \wedge \gamma''(\overline{x}, y)$
- If $t_{\gamma'}(\overline{x}, y) \neq 0$ then since $\gamma'(\overline{x}) \to t_{\gamma'}(\overline{x}) = 0$ we get $\neg\gamma'(\overline{x})$. However since $t_{\gamma'}(\overline{x}, y) \neq 0$ we also get that $t_k(\overline{x}, y) = t'_k(\overline{x}, y)$ so $\gamma(\overline{x}, y)$ becomes $\gamma'(\overline{x}, y)$ and we get a contradiction.

Thus T proves $\gamma(\overline{x}, y) \to \gamma'(\overline{x}, y) \wedge \gamma''(\overline{x}, y)$. □

Lemma 2.2. *Assume there are terms $t'_1, ..., t'_n, s', t''_1, ..., t''_n, s''$ and $\tilde{t}_1, \tilde{t}_2, \tilde{s}$ such that for any $x'_1, ..., x'_n, y', x''_1, ..., x''_n, y''$ and $\tilde{x}_1, \tilde{x}_2, \tilde{y}$, T proves*

$$\gamma_1(x'_1, t'_1(x'_1, ..., x'_n, y'), w_1) \wedge ... \wedge \gamma_n(x'_n, t'_n(x'_1, ..., x'_n, y'), w_n) \to \alpha_1(s'(x'_1, ..., x'_n), y', \overline{w}, z_1),$$

$$\gamma_1(x''_1, t''_1(x''_1, ..., x''_n, y''), w_1) \wedge ... \wedge \gamma_n(x''_n, t''_n(x''_1, ..., x''_n, y''), w_n) \to \alpha_2(s''(x''_1, ..., x''_n), y'', \overline{w}, z_2)$$

and

$$\alpha_1(\tilde{x}_1, \tilde{t}_1(\tilde{x}_1, \tilde{x}_2, \tilde{y}), z_1) \wedge \alpha_2(\tilde{x}_2, \tilde{t}_2(\tilde{x}_1, \tilde{x}_2, \tilde{y}), z_2) \to \beta(\tilde{s}(\tilde{x}_1, \tilde{x}_2), \tilde{y}, z_3).$$

Then there are terms $t_1, ..., t_n, s$ such that for any $\overline{x}, y$

$$\gamma(x_1, t_1(\overline{x}, y), w_1) \wedge ... \wedge \gamma(x_n, t_n(\overline{x}, y), w_n) \to \beta(s(\overline{x}), y, z_3).$$

Proof. Let $x_1' = x_1'' = x_1, ..., x_n' = x_n'' = x_n, \tilde{y} = y$, let $\tilde{x_1} = s'(\overline{x}), \tilde{x_2} = s''(\overline{x})$ and let $y' = \tilde{t}_1(s'(\overline{x}), s''(\overline{x}), y), y'' = \tilde{t}_2(s'(\overline{x}), s''(\overline{x}), y)$. Define $s(\overline{x}) = \tilde{s}(s'(\overline{x}), s''(\overline{x}))$ and use the previous lemma to get $t_1, ..., t_k$ such that T proves

$$\begin{aligned}&\gamma_1(x_1, t_1(\overline{x}, y), w_1) \wedge ... \wedge \gamma_n(x_n, t_n(\overline{x}, y), w_n)\\ &\quad\to \gamma_1(x_1, t_1'(\overline{x}, \tilde{t}_1(s'(\overline{x}), s''(\overline{x}), y)), w_1) \wedge ... \wedge \gamma_n(x_n, t_n'(\overline{x}, \tilde{t}_1(s'(\overline{x}), s''(\overline{x}), y)), w_n)\\ &\quad\ \wedge \gamma_1(x_1, t_1''(\overline{x}, \tilde{t}_2(s'(\overline{x}), s''(\overline{x}), y)), w_1) \wedge ... \wedge \gamma_n(x_n, t_n''(\overline{x}, \tilde{t}_2(s'(\overline{x}), s''(\overline{x}), y)), w_n).\end{aligned}$$

□

Using lemma 2.2 we can easily verify the induction step for most inference rules that do not discharge assumptions. For example to verify it for $\wedge$-introduction we only need to show that there are t_1, t_2, s_1, s_2 such that

$$\begin{aligned}&\alpha(x_1, t_1(x_1, x_2, y_1, y_2), z_1) \wedge \beta(x_1, t_1(x_1, x_2, y_1, y_2), z_2)\\ &\qquad\qquad\to \alpha(s_1(x_1, x_2), y_1, z_1) \wedge \beta(s_2(x_1, x_2), y_2, z_2)\end{aligned}$$

so we just take $t_1(x_1, x_2, y_1, y_2) = y_1, t_2(x_1, x_2, y_1, y_2) = y_2, s_1(x_1, x_2) = x_1$ and $s_2(x_1, x_2) = x_2$.

For $\to$-elimination the condition $\phi \to \psi$ corresponds with

$$\exists X_1, X_2 \forall y_1, y_2 : \alpha(y_1, X_1(y_1, y_2), z_1) \to \beta(X_2(y_1), y_2, z_2)$$

so we need to show that there are t_1, t_2, t_3, s such that

$$\begin{aligned}&(\alpha(t_1(X_1, X_2, x_3, y), X_1(t_1, t_2), z_1) \to \beta(X_2(t_1), t_2(X_1, X_2, x_3, y), z_2))\\ &\qquad\qquad \wedge \alpha(x_3, t_3(X_1, X_2, x_3, y), z_1) \to \beta(s(X_1, X_2, x_3), y, z_2).\end{aligned}$$

Thus we take $t_1(X_1, X_2, x_3, y) = x_3, t_2(X_1, X_2, x_3, y) = y, t_3(X_1, X_2, x_3, y) = X_1(t_1, t_2) = X_1(x_3, y)$ and $s(X_1, X_2, x_3) = X_2(t_1) = X_2(x_3)$.

Finally for the rules $\vee$-elimination and $\exists$-elimination which do discharge assumptions one needs to directly apply lemma 2.1. So e.g. if we use $\exists$-elimination to prove ψ from $\exists z : \phi$ we need to show that there are $t_1, ..., t_n, s$ such that T proves

$$(\gamma_1(x_1, t_1(x_1, ..., x_n, y), w_1) \wedge ... \wedge \gamma_n(x_n, t_n(x_1, ..., x_n, y), w_n)) \to \beta(s(x_1, ..., x_n), y, z_2).$$

Say $\phi' = \exists x : \forall y : \alpha(x, y, z, z_1)$ and $(\exists z : \phi(z))' = \exists z, x \forall y : \alpha(x, y, z, z_1)$ then the induction hypothesis gives that we have $t', s_1', s_2', t_2'', t_2'', s$such that

$$\begin{aligned}&(\gamma_1(x_1', t_1'(x_1', ..., x_n', y'), w_1) \wedge ... \wedge \gamma_n(x_n', t_n'(x_1', ..., x_n', y'), w_n)\\ &\qquad\qquad \to \alpha(s_1'(x_1', ..., x_n'), y', s_2'(x_1', ..., x_n'), z_1),\end{aligned}$$

and

$$\begin{aligned}&(\gamma_1(x_1'', t_1''(x_1'', ..., x_{n+1}'', y''), w_1) \wedge ... \wedge \gamma_n(x_n'', t_n''(x_1'', ..., x_{n+1}'', y''), w_n)\\ &\qquad \wedge \alpha(x_{n+1}'', t_{n+1}''(x_1'', ..., x_{n+1}'', y''), z, z_1) \to \beta(s''(x_1'', ..., x_{n+1}''), y'', z_2).\end{aligned}$$

By assumption for the $\exists$-elimination rule z is not a free variable in $\sigma_1, ..., \sigma_n$ or ψ so it is independent of $w_1, ..., w_n$ and z_2. Hence we can freely pick z. Thus we take $x'_1 = x''_1 = x_1, ..., x'_n = x''_n$ and $x''_{n+1} = s'(x_1, ..., x_n)$. Then we take $y'' = y, y' = t''_{n+1}(x_1, ..., x_n, s'(x_1, ..., x_n), y)$ and $z = s_2(x_1, ..., x_n)$. Finally we define $s(x_1, ..., x_n) = s''(x_1, ..., x_n, s'_1(x_1, ..., x_n))$ and use lemma 2.1 to get $t_1, ..., t_n$.

This ends the proof, so we can conclude the presentation of Gödel's Dialectica translation with:

Theorem 2.3. *If HA proves ϕ where $\phi' = \exists x \forall y : \alpha(x, y, z)$ then there is a term t such that T proves $\alpha(t, y, z)$.*

3 Example 1: Inequality

In HA we can define the normal ordering of natural numbers by saying that $n \leq m$ means that $\exists d : n + d = m$. To test the Dialectica translation I want to start by looking at a useful proposition relating to this ordering. To be able to use the translation on the proposition I first need to formulate it within HA. Then I need to have the constructive proof of the proposition, where the proof is directly from the axioms of HA.

The proposition I look at is that either $n \leq m$ or not, in which case $m < n$, i.e. $\forall n, m : n \leq m \vee (\neg n \leq m \wedge m < n)$. Here $m < n$ is defined as $m \leq n \wedge \neg m = n$. Thus the statement is formally given by

$$\forall n, m : (\exists a : n + a = m) \vee (\neg(\exists b : n + b = m) \wedge (\exists c : m + c = n \wedge \neg m = n)).$$

3.1 Proof of example 1

3.1.1 Lemma: Equality

For all n, m either $n = m$ or not. Formally $\forall n, m : n = m \vee \neg n = m$. The proof is by induction on a.

First I need to show that $\forall b : 0 = b \vee \neg 0 = b$ which I do by induction on b. We immediately have $0 = 0$ and $\neg S(b) = 0$.

Now I assume $\forall b : a = b \vee \neg a = b$ and I need to show $\forall b : S(a) = b \vee \neg S(a) = b$. For this we again use induction on b. We have immediately that $\neg S(a) = 0$. In the case of $S(b)$ we have that $S(a) = S(b)$ if and only if $a = b$, which is either true or false by the induction hypothesis.

3.1.2 Proof of proposition

First note that $a < b$ if and only if $a + S(d) = b$ for some d. This is since if $a < b$ then $a + d = b$ and if $d = 0$ then $a = b$. This gives a contradiction so $d \neq 0$ which means $d = S(e)$ for some e. On the other hand if $a + S(d) = b$ then $a \leq b$. Also if $a = b$ then we get $b + S(d) = b$ so $S(d) = 0$. This gives a contradiction so $\neg a = b$.

Next note that if $m < n$ then $\neg n \leq m$, because if $m < n$ and $n \leq m$ then there is d, e such that $m + S(d) = n$ and $n + e = m$. This give us that $n + e + S(d) = m + S(d) = n$ so $S(e + d) = e + S(d) = 0$ which is impossible.

Now to prove the proposition I use induction on n. First $0 + m = m$ so $\forall m : 0 \leq m$.

Next assume that for all m either $n \leq m$ or the negation holds and $m < n$. If $m < n$ then $m + d = n$ for some d and $m + S(d) = S(m + d) = S(n)$ so $m < S(n)$. Otherwise if $n \leq m$ then by the lemma $n = m$ or $\neg n = m$ i.e. $n < m$. If $n < m$ then $n + S(d) = m$ for some d. Thus $S(n) + d = n + S(d) = m$ so $S(n) \leq m$. If $n = m$ let $d = 0$ so $m + S(d) = S(m) = S(n)$ and $m < S(n)$.

3.2 Translation

3.2.1 Lemma: Equality

The translation of $\forall n, m : n = m \vee \neg n = m$ becomes

$$\exists Eq : \mathrm{Nat}^2 \to \mathrm{Nat}\ \forall n, m : \mathrm{Nat}\ (Eq(n,m) = 1 \wedge n = m) \vee (Eq(n,m) = 0 \wedge \neg n = m).$$

We see that Eq is the characteristic function for equality. From the proof we see that Eq is defined by multiple recursions. In the first case we get a recursion that starts with "True" and is "False" in all other steps. For the recursion step we have a recursive function that starts with "False" and for later steps it applies the previous result to b. So

$$Eq = R_{\mathrm{Nat}\to\mathrm{Nat}}(R_{\mathrm{Nat}}(1, \lambda b^{\mathrm{Nat}} y^{\mathrm{Nat}}.0,), \lambda a^{\mathrm{Nat}} X^{\mathrm{Nat}\to\mathrm{Nat}}.R_{\mathrm{Nat}}(0, \lambda b^{\mathrm{Nat}} y^{\mathrm{Nat}}.X(b),),).$$

3.2.2 Proposition

The translation of

$$\forall n, m : (\exists a : n + a = m) \vee (\neg(\exists b : n + b = m) \wedge (\exists c : m + c = n \wedge \neg m = n))$$

is

$$\begin{aligned}\exists Dif, Dif_2, Leq : \mathrm{Nat}^2 \to \mathrm{Nat} \forall n, m, b : \mathrm{Nat} \\ (Leq(n,m) = 1 \wedge n + Dif(n,m) = m) \\ \vee (Leq(n,m) = 0 \wedge \neg n + b = m \wedge (m + Dif_2(n,m) = n \wedge \neg m = n)).\end{aligned}$$

Here we see that Leq is the characteristic function of $\leq$. Dif is the difference of n and m if $n \leq m$ and Dif_2 is the difference of m and n if $m < n$.

In the proof $n \leq m$ held in the base case and $S(n) \leq m$ held in the induction case when it held for n and $n \neq m$. Thus we can write Leq as

$$Leq = \lambda n^{\mathrm{Nat}} m^{\mathrm{Nat}}.R_{\mathrm{Nat}}(1, \lambda k^{\mathrm{Nat}} x^{\mathrm{Nat}}.x \& ! Eq(k, m), n).$$

The difference between n and m was m at first and then increased by one each step so

$$Dif = \lambda n^{\mathrm{Nat}} m^{\mathrm{Nat}}.R_{\mathrm{Nat}}(m, \lambda k^{\mathrm{Nat}} x^{\mathrm{Nat}}.S(k), n).$$

The difference between m and $S(n)$ was 1 when $m = n$ and then increased by one when $m < n$ so

$$Dif_2 = \lambda n^{\mathrm{Nat}} m^{\mathrm{Nat}}.R_{\mathrm{Nat}}(0, \lambda k^{\mathrm{Nat}} x^{\mathrm{Nat}}.Eq(m,k)?1 : [Eq(x,0)?0 : S(x)], n).$$

4 Example 2: Existence of infinite primes

For a look at how the Dialectica translation works on a more complex statement, I look at the famous theorem stating that there are infinite primes. As in example 1 I need to formulate and prove this in HA.

To formulate the theorem I start with the definition that n divides m ($n|m$) if $\exists b : nb = m$. Then we can define the statement "p is a prime number" ($Prime(p)$) as:

$$(\forall a : a|p \to (a = 1 \vee a = p)) \wedge \neg p = 1$$

Now our theorem becomes

$$\forall n : \exists p : Prime(p) \wedge n \leq p,$$

which is the same as

$$\forall n \exists p : [\forall a : (\exists b : ab = p) \to (a = 1 \vee a = p) \wedge \neg p = 1] \wedge [\exists d : n + d = p].$$

4.1 Proof of example 2

4.1.1 Lemma 1: Bounded quantification

We need to prove that if a formula $\phi(\cdot)$ either holds or does not hold for any given number, then the same is true for the statement that for a given n there is a number $m \leq n$ satisfying ϕ. This we can write as

$$[\forall n : \phi(n) \vee \neg\phi(n)] \to \forall n : [(\exists m : m \leq n \wedge \phi(m)) \vee \neg(\exists m : m \leq n \wedge \phi(m))].$$

For this proof first note that if $m \leq 0$ then $m = 0$. This follows since from Example 1 we have that if $m \leq 0$ then either $m = 0$ or $m < 0$. If $m < 0$ then $m + S(d) = 0$ for some d. However if $m + S(d) = 0$ then $S(m + d) = m + S(d) = 0$ which gives a contradiction.

Now we assume that we have a formula ϕ that is either true or false for any given number, and then work by induction on n. If $n = 0$ we know that if $m \leq n$ then $m = 0$ so we only need to check if $\phi(0)$ is true or false.

Next we assume that for a specific n we have that there either is or is not an $m \leq n$ such that $\phi(m)$ is true. If there is such an m then we also have $m \leq S(n)$. If there is no $m \leq n$ such that $\phi(m)$ is satisfied we use our fist assumption to get that $\phi(S(n))$ is true or false. If $\phi(S(n))$ is true then $m = S(n)$ satisfies $m \leq S(n)$ and $\phi(m)$. If $\phi(S(n))$ is false then there is no $m \leq n$ satisfying $S(m)$ and $m = S(n)$ does not satisfy $\phi(S(m))$ so there is no $m \leq S(n)$ satisfying $\phi(m)$.

4.1.2 Lemma 2: Divisors

Next I show that it is either true or not that m divides n, i.e. $\forall m, n : m|n \vee \neg m|n$. We start by noting that for any b either $mb = n$ is true or not. By Lemma 1 we get that either there is $b \leq n$ such that $mb = n$ or not. If there is b such that $mb = n$ then $m|n$. If there is no such $b \leq n$ then assume there is b such that $mb = n$. Either $b \leq n$ or $n < b$, but we have assumed that we do not have $b \leq n$ so $n < b$ i.e. there is d such that $n + S(d) = b$. Now I show that $mb \neq n$ by induction on m. If $m = 0$ we get $n = mb = 0$ but then $0 \leq n$ and $m0 = n$ which gives a contradiction. For $S(m)$ we get $S(m)b = mb + b = mb + n + S(d) = n + S(mb + d) \neq n$.

4.1.3 Lemma 3: Prime divisors

Finally I need to show that every number (except 1) has a prime divisor i.e. $\forall n : \exists p : \neg n = 1 \rightarrow Prime(p) \wedge p|n$.

For this I use strong induction on n, that is I want to prove by induction on n that for all n and for all $m \leq n$, if $m \neq 1$ then $Prime(p) \wedge p|m$. If I can show this then the lemma follows since $n + 0 = n$ so $n \leq n$.

For the base case we have $m \leq 0$ so $m = 0$. If there exists any prime number p then $p0 = 0$ so p is a prime divisor of 0. Thus I only need to prove the existence of a prime number. Assume $a|2$ so $ab = 2$ for some b. Since $0b = 0$, we get that $a \neq 0$. Now either $a = 1$ or not and either $a = 2$ or not, so start by assuming that $a \neq 1$ and $a \neq 2$. Since $a \neq 0$ we get $a = S(\alpha)$ for some α. If $\alpha = 0$ we would get that $a = 1$ so $\alpha = S(\beta)$ for some β. Similarly $\beta \neq 0$ because $a \neq 2$ so $\beta = S(\gamma)$. Next if $b = 0$ I get $ab = 0$ so $b \neq 0$ and I can write it as $S(c)$ for some c. Then

$$\begin{aligned} S(S(0)) = ab = S(S(S(\gamma)))S(c) &= S(S(S(\gamma)))c + S(S(S(\gamma))) \\ &= S(S(S(S(S(S(\gamma)))c + \gamma))), \end{aligned}$$

so $0 = S(S(S(S(\gamma)))c + \gamma)$ which is impossible. Hence $a = 1$ or $a = 2$ so 2 is prime.

For the induction step, assume that all $m \leq n$ where $m \neq 1$ have prime divisors. We use Lemma 1 and Lemma 2 to get that either there is $m \leq n$ such that $m|S(n)$ and $m \neq 1$ or there is not. If there is such an m then there is b such that $mb = S(n)$ and by our induction hypothesis there is a prime number p such that $p|m$ i.e. there is c such that $pc = m$. Then $p(cb) = S(n)$ so p is a prime divisor of $S(n)$.

Now assume that there is no such m. If $S(n) = 1$ then clearly $S(n)$ satisfies that if $S(n) \neq 1$ then it has a prime divisor. Otherwise assume $a|S(n)$ so $ab = S(n)$ for some b. Then $a \leq n$ or $n < a$. If $a \leq n$ then $a = 1$ or $a \neq 1$, but if $a \neq 1$ we get a contradiction with our assumption so $a = 1$. If $n < a$ then $n + S(d) = a$ for some d so $S(n) + d = n + S(d) = a$ i.e. $S(n) \leq a$. Now $S(n) = a$ or $S(n) < a$. If $S(n) < a$ then $S(n) + S(d) = a$ for some a. Next if $b = 0$ then $ab = 0 \neq S(n)$ and if $b \neq 0$ then $b = S(c)$ for some c so

$$\begin{aligned}ab &= (S(n)+S(d))S(c) = S(n)S(c)+S(d)S(c)\\ &= S(n)c+S(n)+S(d)c+S(d) = S(n)+S(S(n)c+S(dc)+d) \neq S(n).\end{aligned}$$

This means that $a = S(n)$. Thus $a = 1$ or $a = S(n)$ so $S(n)$ is prime. Then since $S(n)1 = S(n)$ we have that $S(n)$ divides $S(n)$. If $m \leq S(n)$ then $m = S(n)$ or $m < S(n)$. If $m = S(n)$ we have shown that we can find a prime divisor of p. Otherwise if $m < S(n)$ then $m + S(d) = S(n)$ for some n so $S(m + d) = S(n)$ and $m + d = n$, which means that $m \leq n$ so we can find a prime divisor of m by the induction hypothesis.

4.1.4 Proof of theorem

I start by showing by induction that for any n there is a number $k \neq 0$ such that for all $m \leq n$ if $m \neq 0$ then $m|k$. Since there are no numbers $m \leq 0$ such that $m \neq 0$ this case is satisfied by showing that there is some $k \neq 0$. $k = 1$ satisfies this. Now assume that there is $k \neq 0$ such that for all $m \leq n$ if $m \neq 0$ then $m|k$ i.e. there is b_m such that $mb_m = k$. Then $m(b_m S(n)) = kS(n)$ so any $m \leq n$ such that $m \neq 0$ divides $kS(n)$. Also $S(n)k = kS(n)$ so $S(n)|kS(n)$. This completes the induction.

Now take any n and take $k \neq 0$ such that for all $m \leq S(n)$ if $m \neq 0$ then $m|k$. Since $k \neq 0$ we have that $S(k) \neq 1$ so by Lemma 3 there exists a prime number p such that $p|S(k)$. Now either $p \leq n$ or $n < p$ but since $2|0$, 0 is not prime so if $p \leq n$ then $p|k$. Thus $pa = S(k)$ for some a and $pb = k$ for some b. Also since $p \neq 1$ we get $p = S(S(q))$ for some q. Now $a \leq b$ or $b < a$. If $a \leq b$ then $a + d = b$ for some d so

$$S(k) = S(pb) = S(p(a+d)) = S(pa+pd) = pa + S(pd) = S(k) + S(pd)$$

this is impossible. Thus $b < a$ so $b + S(d) = a$ for some d. Then

$$\begin{aligned}S(k) = pa = p(b+S(d)) = pb + pS(d) &= k + S(S(q))S(d)\\ &= k + S(S(q))d + S(S(q)) = S(k) + S(S(S(q)d+q)\end{aligned}$$

This is again impossible so $n < p$.

4.2 Translation

4.2.1 Lemma 1: Bounded quantification

The statement of the theorem was

$$[\forall n : \phi(n) \vee \neg\phi(n)] \to \forall n : [(\exists m : m \leq n \wedge \phi(m)) \vee \neg(\exists m : m \leq n \wedge \phi(m))],$$

which is the same as

$$[\forall n' : \phi(n') \vee \neg\phi(n')] \to \forall n : [(\exists m : (\exists d : m+d = n) \wedge \phi(m)) \vee \neg(\exists k : (\exists e : k+e = n) \wedge \phi(k))].$$

The first part $\forall n : \phi(n) \vee \neg\phi(n)$ translates into

$$\exists Z : \text{Nat} \to \text{Nat}, x : \text{Nat}\ \forall n', y : \text{Nat}\ (Z(n') = 1 \wedge \phi(n')) \vee (Z(n') = 0 \wedge \neg\phi(n'))$$

whereas

$$\forall n : [(\exists m : (\exists d : m + d = n) \wedge \phi(m)) \vee \neg(\exists k : (\exists e : k + e = n) \wedge \phi(k))]$$

translates into

$$\begin{aligned}&\exists M, D, W : \text{Nat} \to \text{Nat}\ \forall n, k, e : \text{Nat}\\ &\quad (W(n) = 1 \wedge M(n) + D(n) = n \wedge \phi(m)) \vee (W(n) = 0 \wedge \neg(k + e = n \wedge \phi(k))),\end{aligned}$$

so the statement translates into

$$\begin{gathered}\exists N : (\text{Nat} \to \text{Nat}) \to \text{Nat}^3 \to \text{Nat}, M, D, W : (\text{Nat} \to \text{Nat}) \to \text{Nat} \to \text{Nat}\\ \forall Z : \text{Nat} \to \text{Nat}, n, k, e : \text{Nat}\\ [(Z(N(Z, n, k, e)) = 1 \wedge \phi(N(Z, n, k, e))) \vee (Z(N(Z, n, k, e)) = 0 \wedge \neg\phi(N(Z, n, k, e)))]\\ \to [(W(Z, n) = 1 \wedge M(Z, n) + D(Z, n) = n \wedge \phi(M(Z, n)))\\ \vee (W(Z, n) = 0 \wedge \neg(k + e = n \wedge \phi(k)))].\end{gathered}$$

Here Z is supposed to be characteristic of $\phi(n)$, and W characteristic for $\exists m : m \leq n \wedge \phi(m)$. In the proof $\exists m : m \leq n \wedge \phi(m)$ was decided by induction. For $n = 0$ this was the same as $\phi(n)$ and for $S(n)$ it was true if $\exists m : m \leq n \wedge \phi(m)$ held for n or if $\phi(S(n))$ was true. Thus W can be given by the term

$$W_1 = \lambda Z^{\text{Nat}\to\text{Nat}}.R_{\text{Nat}}(Z(0), \lambda x^{\text{Nat}} n^{\text{Nat}}.x?1 : Z(S(n)),).$$

M is the witness for $\exists m : m \leq n \wedge \phi(m)$ (if such exists) and so has to be zero in the first step of the induction. If there has already been a witness for $\exists m : m \leq n \wedge \phi(m)$ then that was used as a witness for $\exists m : m \leq S(n) \wedge \phi(m)$, and otherwise the witness would have to be $S(n)$, so M can be given by

$$M_1 = \lambda Z^{\text{Nat}\to\text{Nat}}.R_{\text{Nat}}(0, \lambda x^{\text{Nat}} n^{\text{Nat}}.W_1(n)?x : S(n),).$$

Similarly D, the difference between M and n becomes

$$D_1 = \lambda Z^{\text{Nat}\to\text{Nat}}.R_{\text{Nat}}(0, \lambda x^{\text{Nat}} n^{\text{Nat}}.W_1(n)?S(x) : 0,).$$

Finally N needs to come in to make sure that the statement still holds if Z is not characteristic for $\phi(n)$ and the conclusion fails. Then N need to be a witness to the fact that Z is not characteristic for ϕ. The conclusion can fail if:

- $W_1(Z, n) \notin \{0, 1\}$ so $Z(n) \notin \{0, 1\}$ and Z is not a characteristic function at all.

- $W_1(Z,n) = 1$ so $Z(M_1(Z,n)) \neq 0$ but $\phi(M_1(Z,n))$ is false i.e. Z produces a false positive.
- $W_1(Z,n) = 0$ and $k+e=n$ so $k \leq n$ and $Z(n) = 0$ but $\phi(k)$ is true. That is Z produces a false negative.

Thus we can take N to be

$$N_1 = \lambda Z^{\text{Nat}\to\text{Nat}}, e^{\text{Nat}} k^{\text{Nat}}, n^{\text{Nat}}.Eq(W_1(Z,n),0)?k : (Eq(W_1(Z,n),1)?M_1(Z,n) : n).$$

4.2.2 Lemma 2: Divisors

Here the statement was simply $\forall m,n : m|n \vee \neg m|n$ i.e. $\forall m,n(\exists a : ma = n) \vee \neg(\exists b : mb = n)$. This translates into

$$\exists Z, A : \text{Nat}^2 \to \text{Nat}\ \forall m,n,b : \text{Nat}$$
$$(Z(m,n) = 1 \wedge mA(m,n) = n) \vee (Z(m,n) = 0 \wedge \neg mb = n).$$

Here Z is the characteristic function for $m|n$ and A is the witness i.e. A gives the quotient of two numbers if it exists. The proof relied on the fact that $mb = n \vee \neg mb = n$ for given values of m,b,n. Then I used Lemma 1 and checked number b up to n. Thus using W_1, M_1 from Lemma 1 I get

$$Z_2 = \lambda m^{\text{Nat}} n^{\text{Nat}}.W_1(\lambda b^{\text{Nat}}.Eq(mb,n),n),$$
$$A_2 = \lambda m^{\text{Nat}} n^{\text{Nat}}.M_1(\lambda b^{\text{Nat}}.Eq(mb,n),n).$$

4.2.3 Lemma 3: Prime divisors

The statement of lemma 3 was $\forall n : \exists p : \neg n = 1 \to Prime(p) \wedge p|n$ which is the same as

$$\forall n : \exists p : \neg n = 1 \to [\forall a : (\exists b : ab = p) \to (a = 1 \vee a = p) \wedge \neg p = 1] \wedge \exists c : pc = n.$$

The condition of being prime $\forall a : (\exists b : ab = p) \to (a = 1 \vee a = p) \wedge \neg p = 1$ translates into

$$\exists Z : \text{Nat}^2 \to \text{Nat}\ \forall a,b : \text{Nat}$$
$$ab = p \to ((Z(a,b) = 1 \wedge a = 1) \vee (Z(a,b) = 0 \wedge a = p)) \wedge \neg p = 1,$$

so the entire statement becomes

$$\exists P, C : \text{Nat} \to \text{Nat}, Z : \text{Nat}^3 \to \text{Nat}\ \forall a,b,n : \text{Nat}\ \neg n = 1 \to$$
$$[ab = P(n) \to ((Z(a,b,n) = 1 \wedge a = 1) \vee (Z(a,b,n) = 0 \wedge a = P(n))) \wedge \neg P(n) = 1]$$
$$\wedge P(n)C(n) = n.$$

Here $Z(a,b,n)$ should be 1 if $a = 1$ and 0 otherwise.

$P(n)$ should be the function that gives a prime divisor of n. In the proof it was given by inductively showing that a prime exists for all numbers below a certain number and then applying n. Thus we want to recursively define a function that for a given n gives a function that gives a function that for $m \leq n$ returns a prime that divides m. We saw that in the first step this could be the constant function 2. For later numbers if $m = S(n)$ we took $p = m$ if m had no nontrivial divisor and otherwise we applied the induction hypothesis to the nontrivial divisor whereas if $m \neq S(n)$ we simply applied the induction hypothesis to m. To get the nontrivial divisor we used Lemma 1 and 2 so the characteristic function for $S(n)$ having nontrivial divisors is $w(n) = W_1(\lambda k^{\text{Nat}}.Z_2(k,S(n))\&!Eq(k,1),n)$ whereas the divisor is given by $a(n) = M_1(\lambda k^{\text{Nat}}.Z_2(k,S(n))\&!Eq(k,1),n)$. This gives

$$\begin{aligned}P_3 = R_{\text{Nat}\to\text{Nat}}(\lambda m^{\text{Nat}}.2, \lambda n^{\text{Nat}} X^{\text{Nat}\to\text{Nat}} m^{\text{Nat}}.&Eq(m,S(n))?\\ &[w(n)?X(a(n)):S(n)]:X(m),n)(n).\end{aligned}$$

$C(n)$ is the quotient between the prime divisor and n. This was found in a similar way in the proof as the prime. But in the base case it is 0. If $S(n)$ was prime it was 1 and if $S(n)$ was not prime then after we used the induction hypothesis to get a prime divisor of $m|S(n)$ I needed to multiply the quotient of the prime number dividing m and the quotient of m and $S(n)$.

$$\begin{aligned}C_3 = R_{\text{Nat}\to\text{Nat}}(\lambda m^{\text{Nat}}.2, \lambda n^{\text{Nat}} X^{\text{Nat}\to\text{Nat}} m^{\text{Nat}}.&Eq(m,S(n))?\\ &[w(n)?(X(a(n))A_2(a(n),S(n))):S(n)]:X(m),n)(n)\end{aligned}$$

4.2.4 Theorem: Infinite primes

The statement of the theorem was

$$\forall n \exists p : [\forall a : (\exists b : ab = p) \to (a = 1 \vee a = p) \wedge \neg p = 1] \wedge [\exists d : n + d = p].$$

This translates into

$$\begin{aligned}&\exists Z,P,D : \text{Nat}\to\text{Nat}\ \forall a,b,n:\text{Nat}\\ &[ab = P(n) \to ((Z(a,b,n)=1 \wedge a=1) \vee (Z(a,b,n)=0 \wedge a = P(n))) \wedge \neg P(n)=1]\\ &\qquad \wedge [n + D(n) = P(n)].\end{aligned}$$

Here $Z(a,b,n)$ should be 1 if $a = 1$ and 0 otherwise.

$P(n)$ should be the function that gives a prime number larger than n. In the proof such a prime was found by first finding a number k and then applying Lemma 3 to $S(k)$. k was found by induction on n. It was one in the first step and then multiplied by $S(n)$ in each step (so $k = n!$). Thus

$$P = \lambda n^{\text{Nat}}.P_3(S(R_{\text{Nat}}(1, \lambda m^{\text{Nat}} x^{\text{Nat}} : xS(m), n))).$$

$D(n)$ is the witness of $n < P(n)$. In the proof that inequality was shown by proving that $P(n)$ was not less than or equal to n. Thus D is simply

$$D = \lambda n^{\mathrm{Nat}}.\mathit{Dif}_2(P(n), n).$$

A Appendix: Some basic rules in HA

These are some useful results for proving things in HA. However the statements are of the form $\forall x : \phi(x)$ where ϕ does not contain any quantifiers or disjunctions. Therefore the Dialectica translation does not affect these statements and so there is no computational content we can extract that way.

A.1 Addition

Associativity: (a+b)+0=a+b=a+(b+0). Assume $(a+b)+c = a+(b+c)$ then $(a+b)+S(c) = S((a+b)+c) = S(a+(b+c)) = a+S(b+c) = a+(b+S(c))$.

Commutativity: Since $0+0 = 0+0$ and if $a+0 = 0+a$ we get $S(a)+0 = S(a) = S(a+0) = S(0+a) = 0+S(a)$. So $\forall a : a+0 = 0+a$.

Assume $\forall a : a+b = b+a$. Then $0+S(b) = S(b)+0$ and if $a+S(b) = S(b)+a$ we get $S(a)+S(b) = S(S(a)+b) = S(b+S(a)) = S(S(b+a)) = S(S(a+b)) = S(a+S(b)) = S(S(b)+a) = S(b)+S(a)$ so $\forall a : a+S(b) = S(b)+a$.

Cancellation: $a+0 = b+0 \to a = b$. Assume $a+c = b+c \to a = c$ then if $a+S(c) = b+S(c)$ I get $S(a+c) = a+S(c) = b+S(c) = S(b+c)$ so $a+c = b+c$ and $a = b$. Thus $a+S(c) = b+S(c) \to a = b$.

A.2 Multiplication

Distributivity $(a+b)0 = 0 = 0+0 = a0+b0$ so assume $(a+b)c = ac+bc$. Then $(a+b)S(c) = (a+b)c+a+b = ac+bc+a+b = ac+a+bc+b = aS(c)+bS(c)$.

Identity We have $S(0)0 = 0$ and if $S(0)a = a$ we get $S(0)S(a) = S(0)a+S(0) = a+S(0) = S(a)$.

Commutativity We have $0 \cdot 0 = 0 \cdot 0$. Assume $a0 = 0a$ then $S(a)0 = 0 = a0 = 0a = 0a+0 = 0S(a)$. Thus $\forall a : a0 = 0a$.

Assume $\forall a : ab = ba$, then $aS(b) = ab+a = ba+a = ba+S(0)a = (b+S(0))a = S(b)a$.

Associativity We have $(ab)0 = 0 = a0 = a(b0)$. Assume $(ab)c = a(bc)$, then $(ab)S(c) = (ab)c+ab = a(bc)+ab = a(bc+b) = a(bc+bS(0)) = a(b(c+S(0))) = a(bS(c))$.

B Appendix: Haskell implementations

The code is also available at https://pastebin.com/8dB1JGtW

```
s::Int->Int
s x=x+1

r :: a -> (Int -> a -> a) -> Int -> a
r x f n = if n==0 then x else (f (n-1) (r x f (n-1)))

----------EXAMPLE 1----------
--Lemma
eq::Int->Int->Bool
eq a b = r (r True (\n x->False) ) (\n x -> r False (\m y-> x m)) a b


--Proposition
leq::Int->Int->Bool
leq n m=r True (\k x ->x && (not (eq k m))) n

dif::Int->Int->Int
dif n m = r m (\k x->(s k)) n

dif2::Int->Int->Int
dif2 n m = r 0 (\k x -> if (eq m k)
  then 1
  else (if (eq x 0) then 0 else (s x))) n



----------EXAMPLE 2----------
--Lemma 1
--W1
boundedZ :: (Int->Bool)->Int->Bool
boundedZ f n = r (f 0) (\n x-> if x then True else f (s n)) n

--M1
boundedX :: (Int->Bool)->Int->Int
boundedX f n = r 0 (\n x-> if (boundedZ f n) then x else (s n)) n

--D1
```

```
boundedD :: (Int->Bool)->Int->Int
boundedD f n = r 0 (\n x-> if (boundedZ f n) then (s x) else 0) n


--Lemma 2
--Z2
divides:: Int->Int->Bool
divides a b= boundedZ (\d -> a*d==b) b

--A2
quotient:: Int->Int->Int
quotient a b= boundedX (\d -> a*d==b) b


--Lemma 3
--P3
primedivisor :: Int->Int
primedivisor n = (r (\m-> 2) (\n x m-> if m==s n
  then (if (boundedZ (\k-> (divides k (s n)) && (not (k==1))) n)
    then x (boundedX (\k-> (divides k (s n)) && (not (k==1))) n)
    else s n)
  else (x m))) n n

--C3
seconddivisor :: Int->Int
seconddivisor n = (r (\m-> 2) (\n x m-> if m==s n
  then (if (boundedZ (\k-> (divides k (s n)) && (not (k==1))) n)
    then ((\m-> (x m)*(quotient m (s n)) )
      (boundedX (\k-> (divides k (s n)) && (not (k==1))) n))
    else 1)
  else (x m))) n n


--Theorem
largerprime :: Int->Int
largerprime n=primedivisor (s (r 1 (\m x-> x*(s m)) n))
```

References

[AF95] J. Avigad, S.Feferman *Gödel's Functional ("Dialectica") Translation.* Elsevier Science B.V, 1995.

[SU06] H. Sørenson, P.Urzyczyn *Lectures on the Curry-Howard isomorphism.* Elsevier Science B.V, 2006.

[G94] K. Gödel *On an extension of finitary methods which has not yet been used* In: Collected Works, vol. III, Oxford University Press, 1994. Solomon Feferman et al.eds. pp. 271–280