MA5219 LOGIC AND FOUNDATION OF MATHEMATICS I

Lecture 10: Programs for the verification of proofs

Tin Lok Wong

13 September, 2018

The aim of this lecture is to describe how one can Σ_1-define (or equivalently, program) provability explicitly in the standard model of arithmetic $\mathbb{N}$, without invoking the Church–Turing Thesis.

The formulas we define here are by no means canonical. In fact, many choices of the formulas below are non-standard. Therefore, the point is not to learn the formulas themselves, but to learn how one can arrive at such formulas. We will start by setting up some tools for dealing with sequences. Then we use these tools to make recursive definitions. Several standard tricks will come in handy in keeping the formulas $\Delta_0(\exp)$.

Remark 10.1. We will define a number of partial functions in $\mathcal{L}_{\mathrm{A}}(\exp)$. When there is no risk of ambiguity, we will use these new partial functions as if they were primitive in $\mathcal{L}_{\mathrm{A}}(\exp)$. Whenever these new partial functions appear in a formula, it is assumed implicitly that the values exist. All formulas we are going to define in this lecture, except the provability predicate $\Box(y)$, are $\Delta_0(\exp)$. This includes the graphs of the new partial functions. The appearance of these new partial functions in formulas do not introduce new unbounded quantifiers because all the new partial functions in this lecture can be bounded by an $\mathcal{L}_{\mathrm{A}}(\exp)$ term in the standard model of arithmetic. More precisely, every partial function $f(\bar{x})$ we introduce below comes with a bounding $\mathcal{L}_{\mathrm{A}}(\exp)$ term $t(\bar{x})$ such that $\mathbb{N} \models \forall \bar{x}, y\ \big(y = f(\bar{x}) \to y \leqslant t(\bar{x})\big)$. In this case we consider a formula $\theta(\bar{x}, f(\bar{x}))$ as a shorthand for

$$\exists y{\leqslant}t(\bar{x})\ \big(y = f(\bar{x}) \wedge \theta(\bar{x}, y)\big),$$

which is $\Delta_0(\exp)$ whenever $\theta(\bar{x}, y)$ is a $\Delta_0(\exp)$ formula, because the graph $y = f(\bar{x})$ is $\Delta_0(\exp)$.

Recall that we read strings of symbols in $\mathcal{L}_{\mathrm{A}}(\exp)$ as numbers written in hexadecimal representation via Table 8.1. We use commas and double-commas as separators in lists, and $\ulcorner, \urcorner = 0$. To make our definitions below more readable, we omit the underlines for numerals.

Notation. If $\theta(\bar{x}, y)$ is a formula, then $\exists! y\ \theta(\bar{x}, y)$ is a shorthand for

$$\exists y\ \theta(\bar{x}, y) \wedge \forall y, y'\ \big(\theta(\bar{x}, y) \wedge \theta(\bar{x}, y') \to y = y'\big).$$

$\boxed{x - z = y}$ $\quad x = y + z$

Functionality: $\forall x, z\ \big(x \geqslant z \to \exists! y\ (x - z = y)\big)$.

Bound: $\forall x, y, z\ \big(x - z = y \to y \leqslant x\big)$.

$\boxed{\mathrm{len}(s) = \ell}$ $\quad 2^{4\ell} > s \wedge \forall i{<}\ell\ (2^{4i} \leqslant s)$

Picture:

$$s = \begin{array}{|c|c|c|c|} \multicolumn{1}{c}{\ell-1} & & \multicolumn{1}{c}{1} & \multicolumn{1}{c}{0} \\ \hline a_{\ell-1} & \cdots & a_1 & a_0 \\ \hline \end{array} = \sum_{j<\ell} a_j 2^{4j}.$$

Explanation: $\mathrm{len}(s)$ is the length of s when written in hexadecimal representation, i.e.,

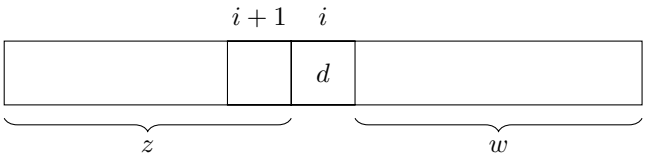
$$\mathrm{len}(s) = \lceil \log_{16}(s+1) \rceil.$$

Functionality: $\forall s\ \exists!\ell\ (\mathrm{len}(s) = \ell)$.

Bound: $\forall s, \ell\ \big(\mathrm{len}(s) = \ell \to \ell \leqslant s\big)$.

$\boxed{\langle s\rangle_i = d}$ $\quad \exists z, w{\leqslant}s\ \big(s = 2^{4(i+1)}z + 2^{4i}d + w \wedge w < 2^{4i} \wedge d < 2^4\big)$

Picture:



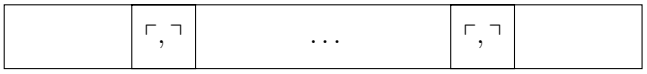
Explanation: $\langle s\rangle_i$ is the ith digit (when counted from the right, starting from 0) in the hexadecimal representation of s

Functionality: $\forall s, i\ \exists!d\ \langle s\rangle_i = d$.

Bound: $\forall s, i, d\ (\langle s\rangle_i = d \to d < 2^4)$.

$\boxed{s \text{ is a comma-separated list}}$ $\quad s = 0 \vee \big(\langle s\rangle_0 \neq 0 \wedge \forall i{<}\mathrm{len}(s)\ \neg\big(\langle s\rangle_i = 0 \wedge \langle s\rangle_{i+1} = 0\big)\big)$

Picture:

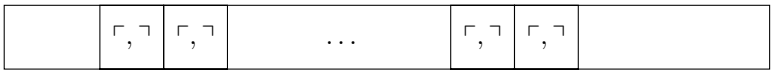


Explanation: A comma-separated list is a (possibly empty) list of strings of symbols separated by commas in which no item contains a comma and no item is empty.

$\boxed{s \text{ is a comma}^2\text{-separated list}}$

$$s = 0 \vee \big(\langle s\rangle_0 \neq 0 \wedge \forall i{<}\mathrm{len}(s)\ \neg\big(\langle s\rangle_i = 0 \wedge \langle s\rangle_{i+1} = 0 \wedge \langle s\rangle_{i+2} = 0\big)\big)$$

Picture:

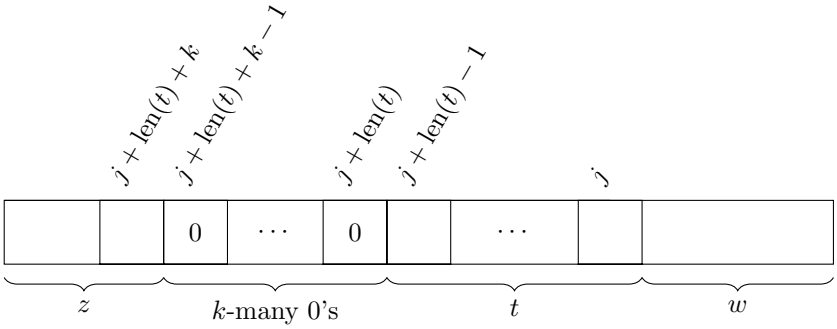


Explanation: A comma^2-separated list is a (possibly empty) list of strings of symbols separated by double-commas in which no item contains a double-comma and no item is empty.

$\boxed{t \text{ is a } k\text{-substring of } s \text{ at position } j}$

$$t \neq 0 \wedge j < \mathrm{len}(s) \wedge \exists z, w{\leqslant}s\ \big(s = 2^{4(j+\mathrm{len}(t)+k)}z + 2^{4j}t + w \wedge w < 2^{4j}\big)$$

Picture:

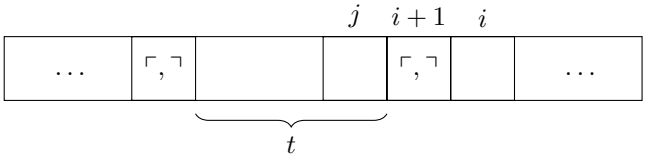


$\boxed{(s)_j^{\text{,}} = t}$ t is a 1-substring of s at position j

$\wedge \big(j = 0 \vee \exists i{<}j \ \big(j = i + 2 \wedge \langle s\rangle_{i+1} = 0 \wedge \langle s\rangle_i \neq 0\big)\big)$

$\wedge \forall k{<}\mathrm{len}(t) \ \langle t\rangle_k \neq 0$

Picture:



Explanation: $(s)_j^{\text{,}}$ is the element in the comma-separated list s whose rightmost symbol is at position j (when counted from the right, starting from 0) in the hexadecimal representation of s.

Functionality: $\forall s, j, t, t' \ \big((s)_j^{\text{,}} = t \wedge (s)_j^{\text{,}} = t' \to t = t'\big)$.

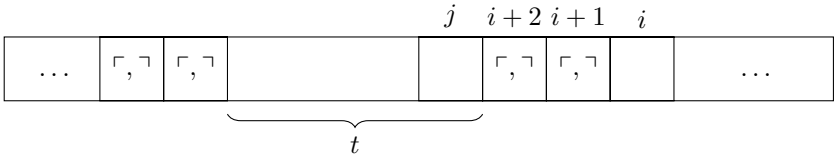
Bound: $\forall s, j, t \ \big((s)_j^{\text{,}} = t \to t \leqslant s\big)$.

$\boxed{(s)_j^{\text{,,}} = t}$ t is a 2-substring of s at position j

$\wedge \big(j = 0 \vee \exists i{<}j \ \big(j = i + 3 \wedge \langle s\rangle_{i+2} = 0 \wedge \langle s\rangle_{i+1} = 0 \wedge \langle s\rangle_i \neq 0\big)\big)$

$\wedge \forall k{<}\mathrm{len}(t) \ \neg\big(\langle t\rangle_k = 0 \wedge \langle t\rangle_{k+1} = 0\big)$

Picture:



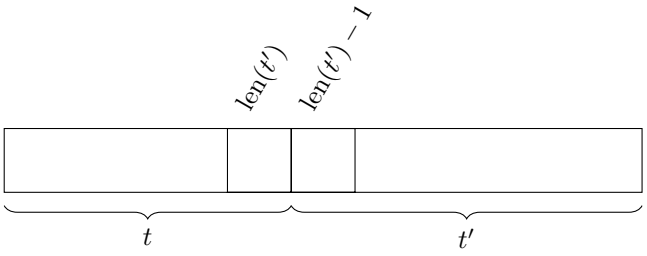
Explanation: $(s)_j^{\text{,,}}$ is the element in the comma^2-separated list s whose rightmost symbol is at position j (when counted from the right, starting from 0) in the hexadecimal representation of s.

Functionality: $\forall s, j, t, t' \ \big((s)_j^{\text{,,}} = t \wedge (s)_j^{\text{,,}} = t' \to t = t'\big)$.

Bound: $\forall s, j, t \ \big((s)_j^{\text{,,}} = t \to t \leqslant s\big)$.

$\boxed{t \mathbin{\hat{}} t' = s}$ $s = 2^{4\,\mathrm{len}(t')}t + t'$

Picture:



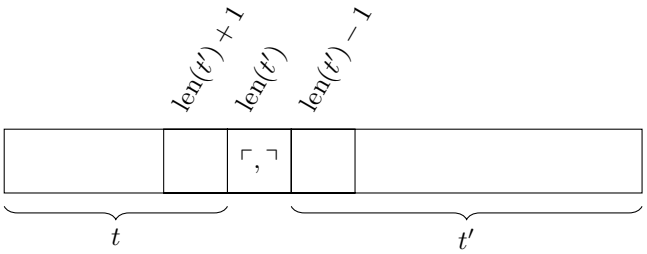
Explanation: $t \mathbin{\hat{}} t'$ is the concatenation of t and t'.

Functionality: $\forall t, t' \ \exists! s \ (t \mathbin{\hat{}} t' = s)$.

Bound: $\forall t, t', s \ (t \mathbin{\hat{}} t' = s \to s \leqslant 2^{4t'}t + t')$.

$\boxed{t \mathbin{\widehat{,}} t' = s}$ $\quad s = 2^{4(\mathrm{len}(t')+1)}t + t'$

Picture:



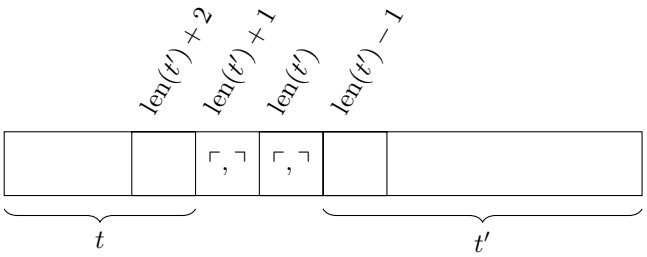
Explanation: $t \mathbin{\widehat{,}} t'$ is the comma-separated list obtained from the comma-separated list t by appending the comma-separated list t' on the right.

Functionality: $\forall t, t'\ \exists! s\ (t \mathbin{\widehat{,}} t' = s)$.

Bound: $\forall t, t', s\ (t \mathbin{\widehat{,}} t' = s \to s \leqslant 2^{4(t'+1)}t + t')$.

$\boxed{t \mathbin{\widehat{,,}} t' = s}$ $\quad s = 2^{4(\mathrm{len}(t')+2)}t + t'$

Picture:



Explanation: $t \mathbin{\widehat{,,}} t'$ is the comma^2-separated list obtained from the comma^2-separated list t by appending the comma^2-separated list t' on the right.

Functionality: $\forall t, t'\ \exists! s\ (t \mathbin{\widehat{,,}} t' = s)$.

Bound: $\forall t, t', s\ (t \mathbin{\widehat{,,}} t' = s \to s \leqslant 2^{4(t'+2)}t + t')$.

$\boxed{s \subseteq_, t}$ $\quad \forall s' \leqslant s\ \forall i < \mathrm{len}(s)\ \exists j < \mathrm{len}(t)\ \left((s)^,_i = s' \to (t)^,_j = s'\right)$

Meaning: Every element in the comma-separated list s is in the comma-separated list t.

$\boxed{s \doteq^, t}$ $\quad s \subseteq_, t \wedge t \subseteq_, s$

Meaning: The comma-separated lists s and t have the same elements.

$\boxed{\mathrm{term}(t)}$ $\quad \exists s \leqslant 2^{4(t+1)^2}$

$$\left(\begin{array}{l} s \text{ is a comma-separated list} \wedge (s)^,_0 = t \\ \wedge\ \forall x \leqslant s\ \forall j < \mathrm{len}(s) \\ \quad \left((s)^,_j = x \to x = \underline{\ulcorner 0 \urcorner} \vee x = \underline{\ulcorner 1 \urcorner} \vee \mathrm{var}(x) \right. \\ \qquad\qquad \left. \vee\ \mathrm{sum}(s,x,j) \vee \mathrm{prod}(s,x,j) \vee \mathrm{power}(s,x,j)\right) \end{array}\right)$$

where

- $\mathrm{var}(x)$ is $\mathrm{len}(x) \geqslant 1 \wedge \forall i < \mathrm{len}(x)\ \langle x \rangle_i = \underline{\mathrm{v}}$;

- $\mathrm{sum}(s,x,j)$ is $\exists i_1,i_2{<}\mathrm{len}(s)\ \big(i_1>j\wedge i_2>j\wedge x=(\underline{\ }\hat{}\,(s)'_{i_1}\,\hat{}\,\underline{+}\,\hat{}\,(s)'_{i_2}\,\hat{}\,\underline{\ })\big)$;
- $\mathrm{prod}(s,x,j)$ and $\mathrm{power}(s,x,j)$ are defined similarly.

Meaning: t is an $\mathcal{L}_{\mathrm{A}}(\mathrm{exp})$ term.

Example: If $t=\ulcorner(2^{\mathsf{v}_0+1}\mathsf{v}_1+2^{\mathsf{v}_0})+\mathsf{v}_2\urcorner$, then we can take

$$s=\ulcorner\mathsf{v}_0,1,\mathsf{v}_0+1,2^{\mathsf{v}_0+1},\mathsf{v}_1,2^{\mathsf{v}_0+1}\mathsf{v}_1,2^{\mathsf{v}_0},2^{\mathsf{v}_0+1}\mathsf{v}_1+2^{\mathsf{v}_0},\mathsf{v}_2,(2^{\mathsf{v}_0+1}\mathsf{v}_1+2^{\mathsf{v}_0})+\mathsf{v}_2\urcorner.$$

Bound for s: A construction sequence for a term t can be assumed to have the form

$$t_0,t_1,\ldots,t_k$$

where all the t_i's are subterms of t. So we may also assume $k\leqslant\mathrm{len}(t)$ because at least one symbol is added in a construction step. The number of symbols in such a sequence is at most $\mathrm{len}(t)\times(\mathrm{len}(t)+1)\leqslant(\mathrm{len}(t)+1)^2$. Thus the code s of such a sequence is at most $2^{4(\mathrm{len}(t)+1)^2}\leqslant 2^{4(t+1)^2}$ by our bound on $\mathrm{len}(t)$.

$\boxed{\mathrm{fma}(t)}$ $\quad\exists s{\leqslant}2^{4(t+1)^2}$

$$\left(\begin{array}{l} s\text{ is a comma-separated list}\wedge(s)'_0=t\\ \wedge\,\forall x{\leqslant}s\ \forall j{<}\mathrm{len}(s)\ \left(\begin{array}{r}(s)'_j=x\to x=\underline{\top}\vee\mathrm{eq}(x)\vee\mathrm{ineq}(x)\\ \vee\,\mathrm{neg}(s,x,j)\vee\mathrm{disj}(s,x,j)\vee\mathrm{exqn}(s,x,j)\end{array}\right)\end{array}\right)$$

where $\mathrm{eq}(x),\mathrm{ineq}(x),\mathrm{neg}(s,x,j),\mathrm{disj}(s,x,j),\mathrm{exqn}(s,x,j)$ correspond to the construction rules for $\mathcal{L}_{\mathrm{A}}(\mathrm{exp})$ formulas.

Meaning: t is an $\mathcal{L}_{\mathrm{A}}(\mathrm{exp})$ formula.

Bound for s: This was found in a way similar to that for $\mathrm{term}(t)$.

$\boxed{\mathrm{seqt}(s)}$ $\quad\exists s_1,s_2{\leqslant}s$

$$\left(\begin{array}{l} s=s_1\,\hat{}\,\underline{\vdash}\,\hat{}\,s_2\wedge\mathrm{fma}(s_2)\\ \wedge\,s_1\text{ is a comma-separated list}\wedge\forall j{<}\mathrm{len}(s_1)\ \mathrm{fma}((s_1)'_j)\end{array}\right)$$

Meaning: s is a sequent.

$\boxed{\mathrm{antet}(s)=t}$ $\quad\exists s_2{\leqslant}s\ (s=t\,\hat{}\,\underline{\vdash}\,\hat{}\,s_2)$

Terminology: $\mathrm{antet}(s)$ is the *antecedent* of the sequent s.

Functionality: $\forall s\ \big(\mathrm{seqt}(s)\to\exists!t\ \mathrm{antet}(s)=t\big)$

Bound: $\forall s,t\ \big(\mathrm{antet}(s)=t\to t\leqslant s\big)$.

$\boxed{\mathrm{succt}(s)=t}$ $\quad\exists s_1{\leqslant}s\ (s=s_1\,\hat{}\,\underline{\vdash}\,\hat{}\,t)$

Terminology: $\mathrm{succt}(s)$ is the *succedent* of the sequent s.

Functionality: $\forall s\ \big(\mathrm{seqt}(s)\to\exists!t\ \mathrm{succt}(s)=t\big)$

Bound: $\forall s,t\ \big(\mathrm{succt}(s)=t\to t\leqslant s\big)$.

$\boxed{\mathrm{pf}(s)}$ $\quad s$ is a comma^2-separated list $\wedge\ s \neq 0$

$$\wedge\, \forall x{\leqslant}s\ \forall j{<}\mathrm{len}(s)$$

$$\left((s)^{''}_j = x \quad \rightarrow \mathrm{seqt}(x) \wedge \left(\begin{array}{l} \mathrm{asn}(x) \vee \mathrm{top}(x) \vee \mathrm{cut}(s,x,j) \vee \mathrm{RAA}(s,x,j) \vee \mathrm{bot}(s,x,j) \\ \vee\, \mathrm{or}(s,x,j) \vee \mathrm{nor}(s,x,j) \vee \mathrm{exL}(s,x,j) \vee \mathrm{exR}(s,x,j) \\ \vee\, \mathrm{refl}(x) \vee \mathrm{sym}(s,x,j) \vee \mathrm{tran}(s,x,j) \vee \mathrm{Leibniz}(s,x,j) \\ \vee\, \mathrm{weak}(s,x,j) \vee \mathrm{MP}(s,x,j) \end{array} \right) \right)$$

where

- $\mathrm{asn}(x), \mathrm{top}(x), \ldots, \mathrm{Leibniz}(s,x,j)$ correspond to the deduction rules listed in Figure 6.1; for example, we can let $\mathrm{cut}(s,x,j)$ be

$$\exists y_1, y_2{\leqslant}s\ \exists i_1, i_2{<}\mathrm{len}(s)$$

$$\left(\begin{array}{l} i_1 > j \wedge i_2 > j \wedge (s)^{''}_{i_1} = y_1 \wedge (s)^{''}_{i_2} = y_2 \\ \wedge\, \exists a{\leqslant}y_1\ \left(\mathrm{fma}(a) \wedge \mathrm{antet}(y_1) \stackrel{,}{=} \mathrm{antet}(x)\,\widehat{,}\,a \wedge \mathrm{antet}(y_2) \stackrel{,}{=} \mathrm{antet}(x)\,\widehat{,}\,\sqsupset\,\widehat{}\,a \right) \\ \wedge\, \mathrm{succt}(y_1) = \bot \wedge \mathrm{succt}(y_2) = \bot \wedge \mathrm{succt}(x) = \bot \end{array} \right);$$

- $\mathrm{weak}(s,x,j)$ and $\mathrm{MP}(s,x,j)$ correspond respectively to the following weakening rule and *modus ponens* rule:

$$\frac{\Phi \vdash \theta}{\Phi \cup \{\psi_1, \psi_2, \ldots, \psi_\ell\} \vdash \theta}\ (\mathrm{w}) \qquad\qquad \frac{\Phi \vdash \neg\theta \vee \eta \qquad \Phi \vdash \theta}{\Phi \vdash \eta}\ (\mathrm{MP})$$

Meaning: s is a proof.

Assignment 10.2. Construct a $\Delta_0(\mathrm{exp})$ formula $\mathrm{MP}(s,x,j)$ which expresses

> the sequent x at position j in the comma^2-separated list s is obtained by an application of (MP) to sequents at positions bigger than j

in $\mathbb{N}$. [5 points]

$\boxed{\Box(y)}$ $\quad \exists s, w$

$$\left(\mathrm{pf}(s) \wedge \exists a{\leqslant}s \left(\begin{array}{l} \mathrm{antet}((s)^{''}_0) = a \wedge \mathrm{succt}((s)^{''}_0) = y \\ \wedge\, \forall x{\leqslant}a\ \forall i{<}\mathrm{len}(a)\ \left((a)^{,}_i = x \rightarrow \exists \bar{z}{\leqslant}w\ \varepsilon(x,\bar{z}) \right) \end{array} \right) \right)$$

Meaning: $T \vdash y$, where T is a recursive $\mathcal{L}_\mathrm{A}(\mathrm{exp})$ theory given beforehand, and $\varepsilon(x,\bar{z})$ is a $\Delta_0(\mathrm{exp})$ formula such that

$$\{x \in \mathbb{N} : \mathbb{N} \models \exists \bar{z}\ \varepsilon(x,\bar{z})\} = \{\ulcorner \sigma \urcorner : \sigma \in T\}.$$

The bound w: To make the formula Σ_1, we add the bound w for $\bar{z}$. A proof is needed to show that the addition of this bound does not alter the truth value of this formula in $\mathbb{N}$. So let $\Box'(y)$ denote the formula obtained from $\Box(y)$ by removing the bound w. Trivially, we have $\mathbb{N} \models \forall y\ \big(\Box(y) \rightarrow \Box'(y)\big)$. Conversely, suppose $y_0 \in \mathbb{N} \models \Box'(y_0)$. Let $s_0, a_0 \in \mathbb{N}$ witnessing this fact. For every $x \leqslant a_0$ and every $i < \mathrm{len}(a_0)$, the definition of $\Box'(y)$ gives $\bar{z}_{x,i} \in \mathbb{N}$ such that $\mathbb{N} \models (a_0)^{,}_i = x \rightarrow \varepsilon(x, \bar{z}_{x,i})$. Since $a_0, \mathrm{len}(a_0) \in \mathbb{N}$, there are only finitely many such pairs (x,i), and hence we only need finitely many tuples $\bar{z}_{x,i}$. These finitely many $\bar{z}_{x,i}$'s have an upper bound $w \in \mathbb{N}$. This witnesses $\mathbb{N} \models \Box(y)$.

Bound for s: By Theorem 9.2, if $T \vdash \mathrm{R}(\mathrm{exp})$ and $T \nvdash \bot$, then the variable s cannot be bounded.

In the next lecture, we will introduce a natural $\mathcal{L}_\mathrm{A}(\mathrm{exp})$ theory extending R(exp) which can prove the key properties of the formulas defined above.