# 1 Path Depth Limit Encoding

To encode: $x.\mathbf{cls} \equiv \texttt{Succ}, x.p.\mathbf{cls} \equiv \texttt{Zero}, x \equiv y \vdash y :: \texttt{Nat}$ with *depth–limit* $= 1$

$$\texttt{Variable} := \{\texttt{x}, \texttt{y}\} \tag{1}$$
$$\texttt{Class} := \{\texttt{Zero}, \texttt{Succ}, \texttt{Nat}\} \tag{2}$$
$$\texttt{Path} := \{\texttt{x}, \texttt{x.p}, \texttt{y}, \texttt{y.p}\} \tag{3}$$
$$p_{v \mapsto q} = s := \tag{4}$$
$$(p = \texttt{x} \land v = \texttt{x} \land q = \texttt{x} \land s = \texttt{x}) \lor \tag{5}$$
$$(p = \texttt{x} \land v = \texttt{x} \land q = \texttt{x.p} \land s = \texttt{x.p}) \lor \tag{6}$$
$$(p = \texttt{x} \land v = \texttt{x} \land q = \texttt{y} \land s = \texttt{y}) \lor \tag{7}$$
$$(p = \texttt{x} \land v = \texttt{x} \land q = \texttt{y.p} \land s = \texttt{y.p}) \lor \tag{8}$$
$$(p = \texttt{x.p} \land v = \texttt{x} \land q = \texttt{x} \land s = \texttt{x.p}) \lor \tag{9}$$
$$(p = \texttt{x.p} \land v = \texttt{x} \land q = \texttt{y} \land s = \texttt{y.p}) \lor \tag{10}$$
$$\ldots \tag{11}$$
$$\forall p.\ p \equiv p \qquad \text{(C-Refl)} \tag{12}$$
$$\forall p, c.\ p.\mathbf{cls} \equiv c \to p :: c \qquad \text{(C-Class)} \tag{13}$$
$$\forall p, q, v, r, s, a, b, c, d. \qquad \text{(C-Subst)} \tag{14}$$
$$s \equiv r \land p_{v \mapsto r} = a \land q_{v \mapsto r} = b \land \tag{15}$$
$$a \equiv b \land p_{v \mapsto s} = c \land q_{v \mapsto s} = d \tag{16}$$
$$\to c \equiv d \tag{17}$$
$$\forall p, c, v, r, s, a, b. \qquad \text{(C-Subst)} \tag{18}$$
$$s \equiv r \land p_{v \mapsto r} = a \land \tag{19}$$
$$a :: c \land p_{v \mapsto s} = b \tag{20}$$
$$\to b :: c \tag{21}$$
$$\forall p, c, v, r, s, a, b. \qquad \text{(C-Subst)} \tag{22}$$
$$s \equiv r \land p_{v \mapsto r} = a \land \tag{23}$$
$$a.\mathbf{cls} \equiv c \land p_{v \mapsto s} = b \tag{24}$$
$$\to b.\mathbf{cls} \equiv c \tag{25}$$
$$\texttt{x} :: \texttt{Zero} \to \texttt{x} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{26}$$
$$\texttt{x.p} :: \texttt{Zero} \to \texttt{x.p} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{27}$$
$$\texttt{x} :: \texttt{Succ} \land \texttt{x.p} :: \texttt{Nat} \to \texttt{x} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{28}$$
$$\texttt{y} :: \texttt{Zero} \to \texttt{y} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{29}$$
$$\texttt{y.p} :: \texttt{Zero} \to \texttt{y.p} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{30}$$
$$\texttt{y} :: \texttt{Succ} \land \texttt{y.p} :: \texttt{Nat} \to \texttt{y} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{31}$$
$$\neg(\texttt{x}.\mathbf{cls} \equiv \texttt{Succ} \land \texttt{x.p}.\mathbf{cls} \equiv \texttt{Zero} \land \texttt{x} \equiv \texttt{y} \to \texttt{y} :: \texttt{Nat}) \tag{32}$$

## 2 Ground Encoding

To encode: $x.\mathbf{cls} \equiv \texttt{Succ}, x.p.\mathbf{cls} \equiv \texttt{Zero}, x \equiv y \vdash y :: \texttt{Nat}$ with $\textit{depth–limit} = 1$

$$\texttt{Variable} := \{\texttt{x}, \texttt{y}\} \tag{1}$$

$$\texttt{Class} := \{\texttt{Zero}, \texttt{Succ}, \texttt{Nat}\} \tag{2}$$

$$\texttt{Path} := \{\texttt{x}, \texttt{x.p}, \texttt{y}, \texttt{y.p}\} \tag{3}$$

$$\texttt{x} \equiv \texttt{x} \wedge \texttt{x.p} \equiv \texttt{x.p} \wedge \texttt{y} \equiv \texttt{y} \wedge \texttt{y.p} \equiv \texttt{y.p} \qquad \text{(C-Refl)} \tag{4}$$

$$\texttt{x}.\mathbf{cls} \equiv \texttt{Zero} \rightarrow \texttt{x} :: \texttt{Zero} \qquad \text{(C-Class)} \tag{5}$$

$$\texttt{x.p}.\mathbf{cls} \equiv \texttt{Zero} \rightarrow \texttt{x.p} :: \texttt{Zero} \qquad \text{(C-Class)} \tag{6}$$

$$\ldots \qquad \text{(C-Class)} \tag{7}$$

$$\texttt{x} \equiv \texttt{y} \wedge \texttt{y} \equiv \texttt{y} \rightarrow \texttt{y} \equiv \texttt{x} \qquad \text{(C-Subst)} \tag{8}$$

$$\texttt{x} \equiv \texttt{y} \wedge \texttt{y} :: \textit{Nat} \rightarrow \texttt{x} :: \texttt{Nat} \qquad \text{(C-Subst)} \tag{9}$$

$$\texttt{x} \equiv \texttt{y} \wedge \texttt{y}.\mathbf{cls} \equiv \textit{Nat} \rightarrow \texttt{x}.\mathbf{cls} \equiv \texttt{Nat} \qquad \text{(C-Subst)} \tag{10}$$

$$\ldots \qquad \text{(C-Subst)} \tag{11}$$

$$\texttt{x} :: \texttt{Zero} \rightarrow \texttt{x} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{12}$$

$$\texttt{x.p} :: \texttt{Zero} \rightarrow \texttt{x.p} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{13}$$

$$\texttt{x} :: \texttt{Succ} \wedge \texttt{x.p} :: \texttt{Nat} \rightarrow \texttt{x} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{14}$$

$$\texttt{y} :: \texttt{Zero} \rightarrow \texttt{y} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{15}$$

$$\texttt{y.p} :: \texttt{Zero} \rightarrow \texttt{y.p} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{16}$$

$$\texttt{y} :: \texttt{Succ} \wedge \texttt{y.p} :: \texttt{Nat} \rightarrow \texttt{y} :: \texttt{Nat} \qquad \text{(C-Prog)} \tag{17}$$

$$\ldots \tag{18}$$

$$\neg(\texttt{x}.\mathbf{cls} \equiv \texttt{Succ} \wedge \texttt{x.p}.\mathbf{cls} \equiv \texttt{Zero} \wedge \texttt{x} \equiv \texttt{y} \rightarrow y :: \texttt{Nat}) \tag{19}$$

### 2.1 Substitution in the Ground Encoding

We want to finitely enumerate the quantified rule:

$$\begin{aligned}&\forall p, c, v, r, s, a, b.\\ &\quad s \equiv r \wedge p_{v \mapsto r} = a \wedge\\ &\quad a :: c \wedge p_{v \mapsto s} = b\\ &\qquad \rightarrow b :: c\end{aligned} \tag{C-Subst}$$

The naïve approach would be to take the cross product of all quantified variables. This would leave us with a lot of meaningless implications, e.g. if we instantiate the rule with $p = \mathtt{x}, v = \mathtt{y}, r = \mathtt{x}, a = \mathtt{y}, s = \mathtt{x}, b = \mathtt{x}, c = \mathtt{Nat}$

$$\begin{aligned}&\mathtt{x} \equiv \mathtt{x} \wedge \mathtt{x}_{\mathtt{y} \mapsto \mathtt{x}} = \mathtt{y} \ \wedge \mathtt{y} :: \mathtt{Nat} \wedge \mathtt{x}_{\mathtt{y} \mapsto \mathtt{x}} = \mathtt{x}\\ &\quad \rightarrow \mathtt{x} :: \mathtt{Nat}\end{aligned}$$

Since we know the substitution to be false, we do not have to include this instantiation into the encoding. Since we only need to include rule instantiations where the substitution predicate holds and we can calculate the substitution prior since all quantified variables are known, we can get rid of the substitution predicate in the encoding altogether.

E.g. the inatantiation with $p = \mathtt{x}, v = \mathtt{x}, r = \mathtt{y}, a = \mathtt{y}, s = \mathtt{x}, b = \mathtt{x}, c = \mathtt{Nat}$

$$\begin{aligned}&\mathtt{x} \equiv \mathtt{y} \wedge \mathtt{x}_{\mathtt{x} \mapsto \mathtt{y}} = \mathtt{y} \ \wedge \mathtt{y} :: \mathtt{Nat} \wedge \mathtt{x}_{\mathtt{x} \mapsto \mathtt{x}} = \mathtt{x}\\ &\quad \rightarrow \mathtt{x} :: \mathtt{Nat}\end{aligned}$$

turns into

$$\mathtt{x} \equiv \mathtt{y} \wedge \mathtt{y} :: Nat \rightarrow \mathtt{x} :: \mathtt{Nat}$$