Extraction from classical proofs with uniformities (a case study)

Trifon Trifonov

Ludwig Maximilian Universität, München

Russell'08 Workshop
Swansea,16 March 2008

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \texttt{nat} \mid \texttt{bool} \mid \texttt{L}(\rho) \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathtt{nat} \mid \mathtt{bool} \mid \mathtt{L}(\rho) \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

$$t, s \quad ::= \quad x \mid ts \mid \lambda_x t \mid \langle t, s \rangle \mid t_{\llcorner} \mid t_{\lrcorner} \mid$$

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathtt{nat} \mid \mathtt{bool} \mid \mathtt{L}(\rho) \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

$$\begin{aligned} t, s \quad ::= \quad & x \mid ts \mid \lambda_x t \mid \langle t, s \rangle \mid t_{\llcorner} \mid t_{\lrcorner} \mid \\ & \mathrm{tt}^{\mathtt{bool}} \mid \mathrm{ff}^{\mathtt{bool}} \mid 0^{\mathtt{nat}} \mid \mathrm{S}^{\mathtt{nat} \Rightarrow \mathtt{nat}} \mid \mathrm{nil}^{\mathtt{L}(\rho)} \mid x^{\rho} :: l^{\mathtt{L}(\rho)} \end{aligned}$$

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathtt{nat} \mid \mathtt{bool} \mid \mathtt{L}(\rho) \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

$$\begin{aligned} t, s \quad ::= \quad & x \mid ts \mid \lambda_x t \mid \langle t, s \rangle \mid t_{\llcorner} \mid t_{\lrcorner} \mid \\ & \mathsf{tt}^{\mathtt{bool}} \mid \mathsf{ff}^{\mathtt{bool}} \mid 0^{\mathtt{nat}} \mid \mathsf{S}^{\mathtt{nat} \Rightarrow \mathtt{nat}} \mid \mathsf{nil}^{\mathtt{L}(\rho)} \mid x^{\rho} :: l^{\mathtt{L}(\rho)} \\ & \textbf{if } b^{\mathtt{bool}} \textbf{ then } s^{\tau} \textbf{ else } t^{\tau} \mid \end{aligned}$$

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathtt{nat} \mid \mathtt{bool} \mid \mathtt{L}(\rho) \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

$$\begin{aligned} t, s \quad ::= \quad & x \mid ts \mid \lambda_x t \mid \langle t, s\rangle \mid t_{\llcorner} \mid t_{\lrcorner} \mid \\ & \mathrm{tt}^{\mathtt{bool}} \mid \mathrm{ff}^{\mathtt{bool}} \mid 0^{\mathtt{nat}} \mid \mathrm{S}^{\mathtt{nat}\Rightarrow\mathtt{nat}} \mid \mathrm{nil}^{\mathtt{L}(\rho)} \mid x^{\rho} :: l^{\mathtt{L}(\rho)} \\ & \textbf{if } b^{\mathtt{bool}} \textbf{ then } s^{\tau} \textbf{ else } t^{\tau} \mid \\ & \mathcal{R}^{\tau}_{\mathtt{nat}} : \mathtt{nat} \Rightarrow \tau \Rightarrow (\mathtt{nat} \Rightarrow \tau \Rightarrow \tau) \Rightarrow \tau \mid \end{aligned}$$

Gödel's T

Types

$$\sigma, \rho \quad ::= \quad \mathtt{nat} \mid \mathtt{bool} \mid \mathtt{L}(\rho) \mid \sigma \Rightarrow \rho \mid \sigma \times \rho$$

Terms

$$\begin{aligned} t, s \quad ::= \quad & x \mid ts \mid \lambda_x t \mid \langle t, s \rangle \mid t_{\llcorner} \mid t_{\lrcorner} \mid \\ & \mathrm{tt}^{\mathtt{bool}} \mid \mathrm{ff}^{\mathtt{bool}} \mid 0^{\mathtt{nat}} \mid \mathrm{S}^{\mathtt{nat} \Rightarrow \mathtt{nat}} \mid \mathrm{nil}^{\mathtt{L}(\rho)} \mid x^{\rho} :: l^{\mathtt{L}(\rho)} \mid \\ & \textbf{if } b^{\mathtt{bool}} \textbf{ then } s^{\tau} \textbf{ else } t^{\tau} \mid \\ & \mathcal{R}^{\tau}_{\mathtt{nat}} : \mathtt{nat} \Rightarrow \tau \Rightarrow (\mathtt{nat} \Rightarrow \tau \Rightarrow \tau) \Rightarrow \tau \mid \\ & \mathcal{R}^{\tau}_{\mathtt{L}(\rho)} : \mathtt{L}(\rho) \Rightarrow \tau \Rightarrow (\rho \Rightarrow \mathtt{L}(\rho) \Rightarrow \tau \Rightarrow \tau) \Rightarrow \tau \end{aligned}$$

Negative Arithmetic (NA^ω) — formulas

We consider the negative fragment of Heyting Arithmetic.

$$A, B \quad ::= \quad P(\vec{t}) \mid \mathtt{at}(b^{\mathtt{bool}}) \mid A \to B \mid A \wedge B \mid \forall_x A$$

Here $\perp$ is just a nullary predicate variable.

Negative Arithmetic (NA$^\omega$) — formulas

We consider the negative fragment of Heyting Arithmetic.

$$
\begin{aligned}
A, B &::= P(\vec{t}) \mid \mathtt{at}(b^{\mathtt{bool}}) \mid A \to B \mid A \wedge B \mid \forall_x A \\
\neg A &::= A \to \bot
\end{aligned}
$$

Here $\bot$ is just a nullary predicate variable.

Negative Arithmetic (NA^{ω}) — formulas

We consider the negative fragment of Heyting Arithmetic.

$$
\begin{array}{rcl}
A, B & ::= & P(\vec{t}) \mid \mathtt{at}(b^{\mathtt{bool}}) \mid A \to B \mid A \wedge B \mid \forall_x A \\
\neg A & ::= & A \to \bot \\
\tilde{\exists}_x A & ::= & \neg \forall_x \neg A
\end{array}
$$

Here $\bot$ is just a nullary predicate variable.

Negative Arithmetic (NA^{ω}) — formulas

We consider the negative fragment of Heyting Arithmetic.

$$\begin{array}{rcl} A, B & ::= & P(\vec{t}) \mid \mathtt{at}(b^{\mathtt{bool}}) \mid A \to B \mid A \wedge B \mid \forall_x A \\ \neg A & ::= & A \to \bot \\ \tilde{\exists}_x A & ::= & \neg \forall_x \neg A \end{array}$$

Here $\bot$ is just a nullary predicate variable.

Negative Arithmetic (NA$^\omega$) — proofs

$$M, N \quad ::= \quad u^A \mid (\lambda_{u^A} M^B)^{A \to B} \mid (M^{A \to B} N^A)^B \mid$$

Negative Arithmetic (NA^ω) — proofs

$$\begin{aligned} M, N \quad ::= \quad & u^A \mid (\lambda_{u^A} M^B)^{A \to B} \mid (M^{A \to B} N^A)^B \mid \\ & (\lambda_x M^A)^{\forall_x A} \ \text{(v.c)} \mid (M^{\forall_x A} r)^{A[x:=r]} \mid \end{aligned}$$

Negative Arithmetic (NA$^\omega$) — proofs

$$
\begin{aligned}
M, N \quad ::= \quad & u^A \mid (\lambda_{u^A} M^B)^{A \to B} \mid (M^{A \to B} N^A)^B \mid \\
& (\lambda_x M^A)^{\forall_x A} \ \text{(v.c)} \mid (M^{\forall_x A} r)^{A[x:=r]} \mid \\
& \left\langle M^A, N^B \right\rangle^{A \wedge B} \mid (M^{A \wedge B} \llcorner)^A \mid (M^{A \wedge B} \lrcorner)^B \mid
\end{aligned}
$$

Negative Arithmetic (NA$^\omega$) — proofs

$$
\begin{aligned}
M, N \quad ::= \quad & u^A \mid (\lambda_{u^A} M^B)^{A \to B} \mid (M^{A \to B} N^A)^B \mid \\
& (\lambda_x M^A)^{\forall_x A} \ \textcolor{red}{\text{(v.c)}} \mid (M^{\forall_x A} r)^{A[x:=r]} \mid \\
& \left\langle M^A, N^B \right\rangle^{A \wedge B} \mid (M^{A \wedge B} \llcorner)^A \mid (M^{A \wedge B} \lrcorner)^B \mid \\
& \mathrm{Cases}_{b,A} : \forall_b \big(A(\mathrm{ff}) \to A(\mathrm{tt}) \to A(b)\big) \mid
\end{aligned}
$$

Negative Arithmetic (NA^{ω}) — proofs

$$
\begin{aligned}
M, N \quad ::= \quad & u^A \mid (\lambda_{u^A} M^B)^{A \to B} \mid (M^{A \to B} N^A)^B \mid \\
& (\lambda_x M^A)^{\forall_x A} \ \text{(v.c)} \mid (M^{\forall_x A} r)^{A[x:=r]} \mid \\
& \left\langle M^A, N^B \right\rangle^{A \wedge B} \mid (M^{A \wedge B} \llcorner)^A \mid (M^{A \wedge B} \lrcorner)^B \mid \\
& \mathrm{Cases}_{b,A} : \forall_b \big(A(\mathrm{ff}) \to A(\mathrm{tt}) \to A(b)\big) \mid \\
& \mathrm{Ind}_{n,A} : \forall_n \big(A(0) \to \forall_n \big(A(n) \to A(Sn)\big) \to A(n)\big) \mid
\end{aligned}
$$

Negative Arithmetic (NA$^\omega$) — proofs

$$
\begin{aligned}
M, N \quad ::= \quad & u^A \mid (\lambda_{u^A} M^B)^{A \to B} \mid (M^{A \to B} N^A)^B \mid \\
& (\lambda_x M^A)^{\forall_x A} \ \textcolor{red}{\text{(v.c)}} \mid (M^{\forall_x A} r)^{A[x:=r]} \mid \\
& \left\langle M^A, N^B \right\rangle^{A \wedge B} \mid (M^{A \wedge B} \llcorner)^A \mid (M^{A \wedge B} \lrcorner)^B \mid \\
& \mathrm{Cases}_{b,A} : \forall_b \big(A(\mathrm{ff}) \to A(\mathrm{tt}) \to A(b)\big) \mid \\
& \mathrm{Ind}_{n,A} : \forall_n \big(A(0) \to \forall_n \big(A(n) \to A(Sn)\big) \to A(n)\big) \mid \\
& \mathrm{Ind}_{l,A} : \forall_l \big(A(\mathrm{nil}) \to \forall_{x,l} \big(A(l) \to A(x :: l)\big) \to A(l)\big) \mid
\end{aligned}
$$

Negative Arithmetic (NA^ω) — proofs

$$
\begin{aligned}
M, N \quad ::= \quad & u^A \mid (\lambda_{u^A} M^B)^{A \to B} \mid (M^{A \to B} N^A)^B \mid \\
& (\lambda_x M^A)^{\forall_x A} \ \textcolor{red}{\text{(v.c)}} \mid (M^{\forall_x A} r)^{A[x:=r]} \mid \\
& \left\langle M^A, N^B \right\rangle^{A \wedge B} \mid (M^{A \wedge B} \llcorner)^A \mid (M^{A \wedge B} \lrcorner)^B \mid \\
& \mathrm{Cases}_{b,A} : \forall_b (A(\mathrm{ff}) \to A(\mathrm{tt}) \to A(b)) \mid \\
& \mathrm{Ind}_{n,A} : \forall_n (A(0) \to \forall_n (A(n) \to A(Sn)) \to A(n)) \mid \\
& \mathrm{Ind}_{l,A} : \forall_l (A(\mathrm{nil}) \to \forall_{x,l} (A(l) \to A(x :: l)) \to A(l)) \mid \\
& \mathrm{Truth} : \mathtt{at}(\mathrm{tt})
\end{aligned}
$$

Notation shortcuts

We will use a special nulltype symbol ε and stipulate:

$$
\begin{array}{rcr}
\varepsilon \times \rho \rightsquigarrow \rho, & t_{\llcorner} \rightsquigarrow t, & \langle t, \varepsilon \rangle \rightsquigarrow t \\
\rho \times \varepsilon \rightsquigarrow \rho, & t_{\lrcorner} \rightsquigarrow t, & \langle \varepsilon, t \rangle \rightsquigarrow t \\
\rho \Rightarrow \varepsilon \rightsquigarrow \varepsilon, & \lambda_x \varepsilon \rightsquigarrow \varepsilon, & \varepsilon t \rightsquigarrow \varepsilon \\
\varepsilon \Rightarrow \rho \rightsquigarrow \rho, & \lambda_{x^\varepsilon} t \rightsquigarrow t, & t\varepsilon \rightsquigarrow t \\
 & \forall_{x^\varepsilon} A \rightsquigarrow A, & M\varepsilon \rightsquigarrow M
\end{array}
$$

We could have used a `unit` type, but then the equalities above become explicit isomorphisms.

Boolean falsity

Using a general predicate variable $\bot$ we work in a minimal logic setting.

However, if we use *decidable falsity* $F := \mathtt{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$F \to A$

Lemma (stability)

$((A \to F) \to F) \to A$

if A contains no predicate variables.

Boolean falsity

Using a general predicate variable $\perp$ we work in a minimal logic setting.
However, if we use *decidable falsity* $F := \mathtt{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$F \to A$

Lemma (stability)

$((A \to F) \to F) \to A$

if A contains no predicate variables.

Boolean falsity

Using a general predicate variable $\bot$ we work in a minimal logic setting.
However, if we use *decidable falsity* $F := \mathtt{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$F \to A$

Lemma (stability)

$((A \to F) \to F) \to A$

if A contains no predicate variables.

Boolean falsity

Using a general predicate variable $\bot$ we work in a minimal logic setting.
However, if we use *decidable falsity* $F := \mathtt{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$F \to A$

Lemma (stability)

$((A \to F) \to F) \to A$

if A contains no predicate variables.

Boolean falsity

Using a general predicate variable $\bot$ we work in a minimal logic setting.
However, if we use *decidable falsity* $F := \texttt{at}(\mathrm{ff})$, we are able to prove by induction on the definition of formulas

Lemma (ex falso quodlibet)

$F \to A$

Lemma (stability)

$((A \to F) \to F) \to A$

if A contains no predicate variables.

Realisability and Dialectica

Realisability

- Formulas A are problems
- They ask for solutions of type A°
- Translations $|A|^r$ verify if r is a solution to A

Dialectica

- Formulas A are problems
- They ask for solutions of type A^+
- that are challenged by terms of type A^-
- Translations $|A|^r_s$ verify if r solves A for a challenge s

Realisability and Dialectica

Realisability

- Formulas A are problems
- They ask for solutions of type A°
- Translations $|A|^r$ verify if r is a solution to A

Dialectica

- Formulas A are problems
- They ask for solutions of type A^+
- that are challenged by terms of type A^-
- Translations $|A|^r_s$ verify if r solves A for a challenge s

Computational type

C	C°	C^+	C^-
$P(\vec{t})$	τ	τ^+	τ^-
$\mathtt{at}(b)$	ε	ε	ε
$A \wedge B$	$A^\circ \times B^\circ$	$A^+ \times B^+$	$A^- \times B^-$
$A \to B$	$A^\circ \Rightarrow B^\circ$	$(A^+ \Rightarrow B^+) \times$ $(A^+ \Rightarrow B^- \Rightarrow A^-)$	$A^+ \times B^-$
$\forall_{x^\rho} A$	$\rho \Rightarrow A^\circ$	$\rho \Rightarrow A^+$	$\rho \times A^-$

Realisability needs predicates with computational content.

Computational type

C	C°	C^+	C^-
$P(\vec{t})$	τ	τ^+	τ^-
$\mathtt{at}(b)$	ε	ε	ε
$A \wedge B$	$A^\circ \times B^\circ$	$A^+ \times B^+$	$A^- \times B^-$
$A \to B$	$A^\circ \Rightarrow B^\circ$	$(A^+ \Rightarrow B^+) \times$ $(A^+ \Rightarrow B^- \Rightarrow A^-)$	$A^+ \times B^-$
$\forall_{x^\rho} A$	$\rho \Rightarrow A^\circ$	$\rho \Rightarrow A^+$	$\rho \times A^-$

Realisability needs predicates with computational content.

Translation

C	$\lvert C\rvert^r$	$\lvert C\rvert^r_s$
$P(\vec{t})$	$P^\circ(r^{\tau^\circ}, \vec{t})$	$P^\pm(r^{\tau^+}, s^{\tau^-}, \vec{t})$
$\mathtt{at}(b)$	$\mathtt{at}(b)$	$\mathtt{at}(b)$
$A \wedge B$	$\lvert A\rvert^{r_\llcorner} \wedge \lvert B\rvert^{r_\lrcorner}$	$\lvert A\rvert^{r_\llcorner}_{s_\llcorner} \wedge \lvert B\rvert^{r_\lrcorner}_{s_\lrcorner}$
$A \to B$	$\forall_x(\lvert A\rvert^x \to \lvert B\rvert^{rx})$	$\lvert A\rvert^{s_\llcorner}_{(r_\lrcorner)(s_\llcorner)(s_\lrcorner)} \to \lvert B\rvert^{(r_\llcorner)(s_\llcorner)}_{s_\lrcorner}$
$\forall_x A(x)$	$\forall_x \lvert A(x)\rvert^{rx}$	$\lvert A(s_\llcorner)\rvert^{r(s_\llcorner)}_{s_\lrcorner}$

The Dialectica translation is always a quantifier-free formula.

Translation

C	$\lvert C\rvert^r$	$\lvert C\rvert^r_s$
$P(\vec{t})$	$P^\circ(r^{\tau^\circ}, \vec{t})$	$P^\pm(r^{\tau^+}, s^{\tau^-}, \vec{t})$
$\mathtt{at}(b)$	$\mathtt{at}(b)$	$\mathtt{at}(b)$
$A \wedge B$	$\lvert A\rvert^{r_\llcorner} \wedge \lvert B\rvert^{r_\lrcorner}$	$\lvert A\rvert^{r_\llcorner}_{s_\llcorner} \wedge \lvert B\rvert^{r_\lrcorner}_{s_\lrcorner}$
$A \to B$	$\forall_x(\lvert A\rvert^x \to \lvert B\rvert^{rx})$	$\lvert A\rvert^{s_\llcorner}_{(r_\lrcorner)(s_\llcorner)(s_\lrcorner)} \to \lvert B\rvert^{(r_\llcorner)(s_\llcorner)}_{s_\lrcorner}$
$\forall_x A(x)$	$\forall_x \lvert A(x)\rvert^{rx}$	$\lvert A(s_\llcorner)\rvert^{r(s_\llcorner)}_{s_\lrcorner}$

The Dialectica translation is always a quantifier-free formula.

Soundness

Theorem (Realisability)

Let $\mathcal{P}$ be a proof of A from assumptions $u_i : C_i$. Then we can prove $|A|^{[\![\mathcal{P}]\!]^\circ}$ from assumptions $|C_i|^{x_{u_i}}$.

Theorem (Dialectica)

Let $\mathcal{P}$ be a proof of A from assumptions $u_i : C_i$. Then we can prove $|A|^{[\![\mathcal{P}]\!]^+}_{y_A}$ from assumptions $|C_i|^{x_{u_i}}_{[\![\mathcal{P}]\!]^-_i}$ and $y_A \notin FV([\![\mathcal{P}]\!]^+)$.

Soundness

Theorem (Realisability)

Let $\mathcal{P}$ be a proof of A from assumptions $u_i : C_i$. Then we can prove $|A|^{[\![\mathcal{P}]\!]^\circ}$ from assumptions $|C_i|^{x_{u_i}}$.

Theorem (Dialectica)

Let $\mathcal{P}$ be a proof of A from assumptions $u_i : C_i$. Then we can prove $|A|_{y_A}^{[\![\mathcal{P}]\!]^+}$ from assumptions $|C_i|_{[\![\mathcal{P}]\!]_i^-}^{x_{u_i}}$ and $y_A \notin FV([\![\mathcal{P}]\!]^+)$.

Extracted terms

$\mathcal{P}$	$[\![\mathcal{P}]\!]^\circ$	$[\![\mathcal{P}]\!]^+$	$[\![\mathcal{P}]\!]_i^-$
$u : A$	x_u	x_u	y_A
$M\llcorner$	$[\![M]\!]^\circ\llcorner$	$[\![M]\!]^+\llcorner$	$[\![M]\!]_i^-\,[y_{A\wedge B} := \langle y_A, \sqcup\rangle]$
$M\lrcorner$	$[\![M]\!]^\circ\lrcorner$	$[\![M]\!]^+\lrcorner$	$[\![M]\!]_i^-\,[y_{A\wedge B} := \langle \sqcup, y_B\rangle]$
$\langle M, N\rangle$	$\langle [\![M]\!]^\circ, [\![N]\!]^\circ\rangle$	$\langle [\![M]\!]^+, [\![N]\!]^+\rangle$	$[\![M]\!]_i^-\,[y_A := y_{A\wedge B}\llcorner] \overset{i}{\bowtie} [\![N]\!]_i^-\,[y_B := y_{A\wedge B}\lrcorner]$

Contraction: $t_1 \overset{A}{\bowtie} t_2 :=$ **if** $|A|^x_{t_1}$ **then** t_2 **else** t_1. Only works if A has no predicate variables!

Extracted terms

$\mathcal{P}$	$[\![\mathcal{P}]\!]^\circ$	$[\![\mathcal{P}]\!]^+$	$[\![\mathcal{P}]\!]^-_i$
$u : A$	x_u	x_u	y_A
$M\llcorner$ $M\lrcorner$	$[\![M]\!]^\circ\llcorner$ $[\![M]\!]^\circ\lrcorner$	$[\![M]\!]^+\llcorner$ $[\![M]\!]^+\lrcorner$	$[\![M]\!]^-_i\ [y_{A\wedge B} := \langle y_A, \sqcup\rangle]$ $[\![M]\!]^-_i\ [y_{A\wedge B} := \langle \sqcup, y_B\rangle]$
$\langle M, N\rangle$	$\langle [\![M]\!]^\circ, [\![N]\!]^\circ\rangle$	$\langle [\![M]\!]^+, [\![N]\!]^+\rangle$	$[\![M]\!]^-_i\ [y_A := y_{A\wedge B}\llcorner]$ $\overset{i}{\bowtie}$ $[\![N]\!]^-_i\ [y_B := y_{A\wedge B}\lrcorner]$

Contraction: $t_1 \overset{A}{\bowtie} t_2 :=$ **if** $|A|^x_{t_1}$ **then** t_2 **else** t_1. Only works if A has no predicate variables!

Extracted terms

$\mathcal{P}$	$[\![\mathcal{P}]\!]^\circ$	$[\![\mathcal{P}]\!]^+$	$[\![\mathcal{P}]\!]^-_i$
$u : A$	x_u	x_u	y_A
$M\llcorner$ $M\lrcorner$	$[\![M]\!]^\circ\llcorner$ $[\![M]\!]^\circ\lrcorner$	$[\![M]\!]^+\llcorner$ $[\![M]\!]^+\lrcorner$	$[\![M]\!]^-_i\ [y_{A\wedge B} := \langle y_A, \sqcup\rangle]$ $[\![M]\!]^-_i\ [y_{A\wedge B} := \langle \sqcup, y_B\rangle]$
$\langle M, N\rangle$	$\langle [\![M]\!]^\circ, [\![N]\!]^\circ\rangle$	$\langle [\![M]\!]^+, [\![N]\!]^+\rangle$	$[\![M]\!]^-_i\ [y_A := y_{A\wedge B}\llcorner]$ $\overset{i}{\bowtie}$ $[\![N]\!]^-_i\ [y_B := y_{A\wedge B}\lrcorner]$

Contraction: $t_1 \overset{A}{\bowtie} t_2 :=$ **if** $|A|^x_{t_1}$ **then** t_2 **else** t_1. Only works if A has no predicate variables!

Extracted terms (ctd.)

$\mathcal{P}$	$\llbracket\mathcal{P}\rrbracket^\circ$	$\llbracket\mathcal{P}\rrbracket^+$	$\llbracket\mathcal{P}\rrbracket^-_i$
$\lambda_u M$	$\lambda_{x_u}\llbracket M\rrbracket^\circ$	$\langle\lambda_{x_u}\llbracket M\rrbracket^+,$ $\lambda_{x_u,y_B}\llbracket M\rrbracket^-_u\rangle$	$\llbracket M\rrbracket^-_i\,[x_u := y_{A\to B}\llcorner]\,[y_B := y_{A\to B}\lrcorner]$
MN	$\llbracket M\rrbracket^\circ\llbracket N\rrbracket^\circ$	$(\llbracket M\rrbracket^+\llcorner)\llbracket N\rrbracket^+$	$\llbracket M\rrbracket^-_i\,[y_{A\to B} := \langle\llbracket N\rrbracket^+, y_B\rangle]$ $\overset{i}{\bowtie}$ $\llbracket N\rrbracket^-_i\,[y_A := (\llbracket M\rrbracket^+\lrcorner)\llbracket N\rrbracket^+ y_B]$

Extracted terms (ctd.)

$\mathcal{P}$	$[\![\mathcal{P}]\!]^\circ$	$[\![\mathcal{P}]\!]^+$	$[\![\mathcal{P}]\!]^-_i$
$\lambda_x M$	$\lambda_x [\![M]\!]^\circ$	$\lambda_x [\![M]\!]^+$	$[\![M]\!]^-_i \, [x := y_{\forall_x A \llcorner}] \, [y_A := y_{\forall_x A \lrcorner}]$
Mt	$[\![M]\!]^\circ t$	$[\![M]\!]^+ t$	$[\![M]\!]^-_i \, [y_{\forall_x A} := \langle t, y_A \rangle]$

Realising boolean induction

$\mathcal{P}$	$\mathrm{Cases}_{b,A}\, b\, M\, N$
$[\![\mathcal{P}]\!]^\circ$	**if** b **then** $[\![M]\!]^\circ$ **else** $[\![N]\!]^\circ$
$[\![\mathcal{P}]\!]^+$	**if** b **then** $[\![M]\!]^+$ **else** $[\![N]\!]^+$
$[\![\mathcal{P}]\!]_i^-$	$[\![M]\!]_i^- \overset{i}{\bowtie} [\![N]\!]_i^-$

Realising natural induction

$\mathcal{P}$	$\mathrm{Ind}_{n,A}\, n\, M\, N$
$[\![\mathcal{P}]\!]^\circ$	$\mathcal{R}_{\mathtt{nat}}\, n\, [\![M]\!]^\circ\, [\![N]\!]^\circ$
$[\![\mathcal{P}]\!]^+$	$\mathcal{R}_{\mathtt{nat}}\, n\, [\![M]\!]^+\, (\lambda_n [\![N]\!]^+\, n\, \llcorner)$
$[\![\mathcal{P}]\!]^-_i$	$\mathcal{R}_{\mathtt{nat}}\, n\, (\lambda_{y_A} [\![M]\!]^-_i)$
	$\left(\lambda_{n,p,y} \begin{array}{c} [\![N]\!]^-_i\, [y_S := \langle n, [\![\mathcal{P}]\!]^+\, n, y\rangle] \\ \overset{i}{\bowtie} \\ p([\![N]\!]^+\, n\, \lrcorner([\![\mathcal{P}]\!]^+\, n)y) \end{array}\right)$

Realising list induction

$\mathcal{P}$	$\mathrm{Ind}_{l,A}\, n\, M\, N$
$[\![\mathcal{P}]\!]^\circ$	$\mathcal{R}_{\mathbb{L}(\rho)}\, l\, [\![M]\!]^\circ\, [\![N]\!]^\circ$
$[\![\mathcal{P}]\!]^+$	$\mathcal{R}_{\mathbb{L}(\rho)}\, l\, [\![M]\!]^+\, (\lambda_{x,l}[\![N]\!]^+\, x\, l\, \llcorner)$
$[\![\mathcal{P}]\!]^-_i$	$\mathcal{R}_{\mathbb{L}(\rho)}\, l\, (\lambda_{y_A}[\![M]\!]^-_i) \left(\lambda_{x,l,p,y} \begin{array}{c} [\![N]\!]^-_i\, [y_S := \langle x, l, [\![\mathcal{P}]\!]^+\, l, y\rangle] \\ \overset{i}{\bowtie} \\ p([\![N]\!]^+\, x\, l\, \lrcorner([\![\mathcal{P}]\!]^+\, l) y) \end{array} \right)$

A-translation

Theorem (Extraction via A-translation)

Let $\mathcal{P}$ be a proof of

$$\mathrm{NA}^{\omega} \vdash \vec{D} \to \tilde{\exists}_{y^{\rho}} G$$

with $\vec{D}$ and G not containing $\perp$. Then

$$\mathrm{NA}^{\omega} \vdash \vec{D} \to G(\llbracket \mathcal{P} \rrbracket^{\circ}(\lambda_y y)).$$

Proof.

Let $\tau^{\circ}(\perp) = \rho$ and $\perp$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. $\square$

A-translation

Theorem (Extraction via A-translation)

Let $\mathcal{P}$ be a proof of

$$\mathsf{NA}^\omega \vdash \vec{D} \to (\forall_{y^\rho}.G \to \bot) \to \bot$$

with $\vec{D}$ and G not containing $\bot$. Then

$$\mathsf{NA}^\omega \vdash \vec{D} \to G([\![\mathcal{P}]\!]^\circ(\lambda_y y)).$$

Proof.

Let $\tau^\circ(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. $\square$

A-translation

Theorem (Extraction via A-translation)

Let $\mathcal{P}$ be a proof of

$$\mathrm{NA}^{\omega} \vdash \vec{D} \to (\forall_{y^{\rho}}.G \to \bot) \to \bot$$

with $\vec{D}$ and G not containing $\bot$. Then

$$\mathrm{NA}^{\omega} \vdash \vec{D} \to G([\![\mathcal{P}]\!]^{\circ}(\lambda_y y)).$$

Proof.

Let $\tau^{\circ}(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. $\square$

A-translation

Theorem (Extraction via A-translation)

Let $\mathcal{P}$ be a proof of

$$\mathrm{NA}^\omega \vdash \vec{D} \to (\forall_{y^\rho}.G \to \bot) \to \bot$$

with $\vec{D}$ and G not containing $\bot$. Then

$$\mathrm{NA}^\omega \vdash \vec{D} \to G([\![\mathcal{P}]\!]^\circ(\lambda_y y)).$$

Proof.

Let $\tau^\circ(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. $\square$

A-translation

Theorem (Extraction via A-translation)

Let $\mathcal{P}$ be a proof of

$$\mathsf{NA}^\omega \vdash \vec{D} \to (\forall_{y^\rho}.G \to \bot) \to \bot$$

with $\vec{D}$ and G not containing $\bot$. Then

$$\mathsf{NA}^\omega \vdash \vec{D} \to G([\![\mathcal{P}]\!]^\circ(\lambda_y y)).$$

Proof.
Let $\tau^\circ(\bot) = \rho$ and $\bot$ be translated to a unary predicate variable $\mathcal{A}$. Substitute $\mathcal{A}$ with $G(y)$. $\square$

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m (n = 2m \to \tilde{\exists}_k\, n^2 = 4k)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall^{\mathrm{U}}_x A$, i.e., the verifying proof is *uniform* in this variable.

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m (n = 2m \to \tilde{\exists}_k \, n^2 = 4k)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall^{\mathrm{U}}_x A$, i.e., the verifying proof is *uniform* in this variable.

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m (n = 2m \to \tilde{\exists}_k \, n^2 = 4k)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall_x^{\mathsf{U}} A$, i.e., the verifying proof is *uniform* in this variable.

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m (n = 2m \to \tilde{\exists}_k\, n^2 = 4k)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall^{\mathrm{U}}_x A$, i.e., the verifying proof is *uniform* in this variable.

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m (n = 2m \to \tilde{\exists}_k\, n^2 = 4k)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall_x^{\mathrm{U}} A$, i.e., the verifying proof is *uniform* in this variable.

Uniform quantifiers

Consider the formula

$$\forall_n \forall_m (n = 2m \to \tilde{\exists}_k \, n^2 = 4k)$$

A realiser could be $k = m^2$ or $k = \mathrm{Quot}(n^2, 4)$. Hence, a quantified variable is not necessarily computationally relevant. If not, we quantify it *uniformly*: $\forall_x^{\mathrm{U}} A$, i.e., the verifying proof is *uniform* in this variable.

Negative Arithmetic (NA$^\omega$) — formulas

We consider the negative fragment of Heyting Arithmetic.

$$\begin{aligned} A, B &::= P(\vec{t}) \mid \mathtt{at}(b^{\mathtt{bool}}) \mid A \to B \mid A \wedge B \mid \forall_x A \mid \forall^{\mathsf{U}}_x A \\ \neg A &::= A \to \bot \\ \tilde{\exists}_x A &::= \neg \forall_x \neg A \end{aligned}$$

Negative Arithmetic (NA^{ω}) — proofs

$$
\begin{aligned}
M, N \quad ::= \quad & u^A \mid (\lambda_{u^A} M^B)^{A \to B} \mid (M^{A \to B} N^A)^B \mid \\
& (\lambda_x M^A)^{\forall_x A} \text{ (v.c)} \mid (M^{\forall_x A} r)^{A[x:=r]} \mid \\
& (\lambda_x^{\mathsf{U}} M^A)^{\forall_x^{\mathsf{U}} A} \text{ (v.c)} \mid (M^{\forall_x^{\mathsf{U}} A} r)^{A[x:=r]} \mid \\
& \left\langle M^A, N^B \right\rangle^{A \wedge B} \mid (M^{A \wedge B}{}_{\llcorner})^A \mid (M^{A \wedge B}{}_{\lrcorner})^B \mid \\
& \mathrm{Cases}_{b,A} : \forall_b \big(A(\mathrm{ff}) \to A(\mathrm{tt}) \to A(b)\big) \mid \\
& \mathrm{Ind}_{n,A} : \forall_n \big(A(0) \to \forall_n \big(A(n) \to A(Sn)\big) \to A(n)\big) \mid \\
& \mathrm{Ind}_{l,A} : \forall_l \big(A(\mathrm{nil}) \to \forall_{x,l} \big(A(l) \to A(x :: l)\big) \to A(l)\big) \mid \\
& \mathrm{Truth} : \mathtt{at}(\mathrm{tt})
\end{aligned}
$$

Computational type

C	C°	C^+	C^-
$P(\vec{t})$	τ	τ^+	τ^-
$\mathtt{at}(b)$	ε	ε	ε
$A \wedge B$	$A^\circ \times B^\circ$	$A^+ \times B^+$	$A^- \times B^-$
$A \rightarrow B$	$A^\circ \Rightarrow B^\circ$	$(A^+ \Rightarrow B^+) \times$ $(A^+ \Rightarrow B^- \Rightarrow A^-)$	$A^+ \times B^-$
$\forall_{x^\rho} A$	$\rho \Rightarrow A^\circ$	$\rho \Rightarrow A^+$	$\rho \times A^-$
$\forall^{\mathsf{U}}_{x^\rho} A$	A°	A^+	A^-

Translation

C	$\lvert C\rvert^r$	$\lvert C\rvert^r_s$
$P(\vec{t})$	$P^\circ(r^{\tau^\circ}, \vec{t})$	$P^\pm(r^{\tau^+}, s^{\tau^-}, \vec{t})$
$\mathtt{at}(b)$	$\mathtt{at}(b)$	$\mathtt{at}(b)$
$A \wedge B$	$\lvert A\rvert^{r_\llcorner} \wedge \lvert B\rvert^{r_\lrcorner}$	$\lvert A\rvert^{r_\llcorner}_{s_\llcorner} \wedge \lvert B\rvert^{r_\lrcorner}_{s_\lrcorner}$
$A \to B$	$\forall_x(\lvert A\rvert^x \to \lvert B\rvert^{rx})$	$\lvert A\rvert^{s_\llcorner}_{(r_\lrcorner)(s_\llcorner)(s_\lrcorner)} \to \lvert B\rvert^{(r_\llcorner)(s_\llcorner)}_{s_\lrcorner}$
$\forall_x A(x)$	$\forall_x \lvert A(x)\rvert^{rx}$	$\lvert A(s_\llcorner)\rvert^{r(s_\llcorner)}_{s_\lrcorner}$
$\forall^{\mathsf{U}}_x A(x)$	$\forall_x \lvert A(x)\rvert^r$	$\forall_x \lvert A(x)\rvert^r_s$

The Dialectica translation is not always quantifier-free!

Translation

C	$\lvert C \rvert^r$	$\lvert C \rvert^r_s$
$P(\vec{t})$	$P^\circ(r^{\tau^\circ}, \vec{t})$	$P^\pm(r^{\tau^+}, s^{\tau^-}, \vec{t})$
$\mathtt{at}(b)$	$\mathtt{at}(b)$	$\mathtt{at}(b)$
$A \wedge B$	$\lvert A \rvert^{r_\llcorner} \wedge \lvert B \rvert^{r_\lrcorner}$	$\lvert A \rvert^{r_\llcorner}_{s_\llcorner} \wedge \lvert B \rvert^{r_\lrcorner}_{s_\lrcorner}$
$A \to B$	$\forall_x(\lvert A \rvert^x \to \lvert B \rvert^{rx})$	$\lvert A \rvert^{s_\llcorner}_{(r_\lrcorner)(s_\llcorner)(s_\lrcorner)} \to \lvert B \rvert^{(r_\llcorner)(s_\llcorner)}_{s_\lrcorner}$
$\forall_x A(x)$	$\forall_x \lvert A(x) \rvert^{rx}$	$\lvert A(s_\llcorner) \rvert^{r(s_\llcorner)}_{s_\lrcorner}$
$\forall^{\mathrm{U}}_x A(x)$	$\forall_x \lvert A(x) \rvert^r$	$\forall_x \lvert A(x) \rvert^r_s$

The Dialectica translation is not always quantifier-free!

Extracted terms

$\mathcal{P}$	$[\![\mathcal{P}]\!]^\circ$	$[\![\mathcal{P}]\!]^+$	$[\![\mathcal{P}]\!]_i^-$
$\lambda_x M$	$\lambda_x [\![M]\!]^\circ$	$\lambda_x [\![M]\!]^+$	$[\![M]\!]_i^- \, [x := y_{\forall_x A \llcorner}] \, [y_A := y_{\forall_x A \lrcorner}]$
Mt	$[\![M]\!]^\circ t$	$[\![M]\!]^+ t$	$[\![M]\!]_i^- \, [y_{\forall_x A} := \langle t, y_A \rangle]$
$\lambda_x^{\mathsf{U}} M$	$[\![M]\!]^\circ$	$[\![M]\!]^+$	$[\![M]\!]_i^- \left[y_A := y_{\forall_x^{\mathsf{U}} A} \right]$
$M^{\forall_x^{\mathsf{U}} A} t$	$[\![M]\!]^\circ$	$[\![M]\!]^+$	$[\![M]\!]_i^- \left[y_{\forall_x^{\mathsf{U}} A} := y_A \right]$

Uniformisation correctness

Definition (Realisability)
$\lambda_x^{\mathsf{U}} M$ is correct if $x \notin FV(\llbracket M \rrbracket^\circ)$

Definition (Dialectica)
$\lambda_x^{\mathsf{U}} M$ is correct if $x \notin FV(\llbracket M \rrbracket^+)$ and $x \notin FV(\llbracket M \rrbracket_i^-)$
$\mathcal{P}$ is correct if also for every $t_1 \overset{A}{\bowtie} t_2$, $\forall^{\mathsf{U}}$ does not occur in A

Uniformisation correctness

Definition (Realisability)

$\lambda^{\mathsf{U}}_x M$ is correct if $x \notin FV(\llbracket M \rrbracket^\circ)$

Definition (Dialectica)

$\lambda^{\mathsf{U}}_x M$ is correct if $x \notin FV(\llbracket M \rrbracket^+)$ and $x \notin FV(\llbracket M \rrbracket^-_i)$

$\mathcal{P}$ is correct if also for every $t_1 \overset{A}{\bowtie} t_2$, $\forall^{\mathsf{U}}$ does not occur in A

Uniformisation correctness

Definition (Realisability)

$\lambda^{\mathsf{U}}_x M$ is correct if $x \notin FV(\llbracket M \rrbracket^\circ)$

Definition (Dialectica)

$\lambda^{\mathsf{U}}_x M$ is correct if $x \notin FV(\llbracket M \rrbracket^+)$ and $x \notin FV(\llbracket M \rrbracket^-_i)$

$\mathcal{P}$ is correct if also for every $t_1 \overset{A}{\bowtie} t_2$, $\forall^{\mathsf{U}}$ does not occur in A

List reversal

Let Rev be a binary predicate without computational content for the graph of the list reversal function. Assumptions:

$$R_0 : \quad \mathrm{Rev}(\mathrm{nil}, \mathrm{nil})$$

$$R_1 : \quad \forall_{x, l_1, l_2} \left(\mathrm{Rev}(l_1, l_2) \to \mathrm{Rev}(l_1 {:}{+}{:} x{:}, x :: l_2)\right)$$

Here $l_1 {:}{+}{:} l_2$ denotes appending l_2 to l_1. We assume that Rev is undecidable.

List reversal

Let Rev be a binary predicate without computational content for the graph of the list reversal function. Assumptions:

$$\begin{aligned} R_0 &: \quad \mathrm{Rev}(\mathrm{nil}, \mathrm{nil}) \\ R_1 &: \quad \forall_{x,l_1,l_2} \left(\mathrm{Rev}(l_1, l_2) \to \mathrm{Rev}(l_1{:}{+}{:}x{:}, x :: l_2)\right) \end{aligned}$$

Here $l_1{:}{+}{:}l_2$ denotes appending l_2 to l_1. We assume that Rev is undecidable.

List reversal

Let Rev be a binary predicate without computational content for the graph of the list reversal function. Assumptions:

$$R_0 : \quad \mathrm{Rev}(\mathrm{nil}, \mathrm{nil})$$
$$R_1 : \quad \forall_{x,l_1,l_2} \left(\mathrm{Rev}(l_1, l_2) \to \mathrm{Rev}(l_1{:+:}x{:}, x :: l_2)\right)$$

Here $l_1{:+:}l_2$ denotes appending l_2 to l_1. We assume that Rev is undecidable.

List reversal

Let Rev be a binary predicate without computational content for the graph of the list reversal function. Assumptions:

$$\begin{aligned} R_0: &\quad \mathrm{Rev}(\mathsf{nil}, \mathsf{nil}) \\ R_1: &\quad \forall_{x,l_1,l_2} \left(\mathrm{Rev}(l_1, l_2) \to \mathrm{Rev}(l_1{:+:}x{:}, x :: l_2)\right) \end{aligned}$$

Here $l_1{:+:}l_2$ denotes appending l_2 to l_1. We assume that Rev is undecidable.

We will prove that it is not possible that a list is not reversible.

$$M : \forall_{l_0} \tilde{\exists}_l \mathrm{Rev}(l_0, l)$$

To this end, we prove by induction that if a list is not reversible then none of its initial segments is:

$$\forall_l (\mathrm{Rev}(l_0, l) \to \bot) \to \forall_{l_2} \forall_{l_1} \left(l_1{:}{+}{:}l_2 = l_0 \to \forall_l (\mathrm{Rev}(l_1, l) \to \bot)\right)$$

We will prove that it is not possible that a list is not reversible.

$$M : \forall_{l_0} \tilde{\exists}_l \mathrm{Rev}(l_0, l)$$

To this end, we prove by induction that if a list is not reversible then none of its initial segments is:

$$\forall_l (\mathrm{Rev}(l_0, l) \to \bot) \to \forall_{l_2} \forall_{l_1} \left(l_1 {:+:} l_2 = l_0 \to \forall_l (\mathrm{Rev}(l_1, l) \to \bot) \right)$$

Proof

Induction on l_2 for

$$L : \forall_{l_2} \forall_{l_1} \left(l_1 {:+:} l_2 = l_0 \to \forall_l (\mathrm{Rev}(l_1, l) \to \bot) \right)$$

Proof

Induction on l_2 for

$$L : \forall_{l_2}\forall_{l_1}\big(l_1{:}{+}{:}l_2 = l_0 \to \forall_l(\mathrm{Rev}(l_1, l) \to \bot)\big)$$

$$M := \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0$$

Proof

Induction on l_2 for

$$L : \forall_{l_2} \forall_{l_1} \big(l_1 {:+:} l_2 = l_0 \to \forall_l (\mathrm{Rev}(l_1, l) \to \bot) \big)$$

$$\begin{aligned} M &:= \lambda_{R_0, R_1} \lambda_{l_0} \lambda_u^{\forall_l (\mathrm{Rev}(l_0, l) \to \bot)} \, L \, l_0 \, \mathrm{nil} \, M_E^{\mathrm{nil}{:+:}l_0 = l_0} \, \mathrm{nil} \, R_0 \\ L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)} \, l_2 \, L_B \, L_S \end{aligned}$$

Proof

Induction on l_2 for

$$L : \forall_{l_2}\forall_{l_1}\left(l_1{:}{+}{:}l_2 = l_0 \to \forall_l(\mathrm{Rev}(l_1, l) \to \bot)\right)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u
\end{aligned}$$

Proof

Induction on l_2 for

$$L : \forall_{l_2}\forall_{l_1}\left(l_1 {:+:} l_2 = l_0 \to \forall_l(\mathrm{Rev}(l_1, l) \to \bot)\right)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1:+:\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1:+:(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)}
\end{aligned}$$

Proof

Induction on l_2 for

$$L : \forall_{l_2}\forall_{l_1}\left(l_1{:}{+}{:}l_2 = l_0 \to \forall_l(\mathrm{Rev}(l_1, l) \to \bot)\right)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}{+}{:}(x\,::\,l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x\,::\,l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1:+:\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1:+:(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}
$$

Extraction by A-translation

Extracted program:

$$[\![L_B]\!]^\circ := \lambda_u u$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_B]\!]^\circ := \lambda_u u$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_B]\!]^\circ := \lambda_u u$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0\\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S\\
L_B &:= \lambda_{l_1}\lambda_v^{l_1:+:\mathrm{nil}=l_0}\mathrm{Compat}\; l_1\; l_0\; u\\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1:+:(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)}\\
&\qquad p\,(l_1:+:x:)\,L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\,L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)}\\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_B]\!]^\circ := (\lambda_u u) x_u$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)} \, L \, l_0 \, \mathrm{nil} \, M_E^{\mathrm{nil}:+:l_0=l_0} \, \mathrm{nil} \, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)} \, l_2 \, L_B \, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat} \, l_1 \, l_0 \, u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1:+:(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p \, (l_1:+:x:) \, L_E^{(l_1:+:x:):+:l_2=l_0} \, (x::l) \, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1 \, x \, l_1 \, l \, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_B]\!]^\circ := x_u$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1:+:\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1:+:(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_B]\!]^\circ := \lambda_{l_1} x_u$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1:+:(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad x_p$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1:+:(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad x_p(l_1{:}{+}{:}x{:})$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0\\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S\\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\; l_1\; l_0\; u\\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)}\\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)}\\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad x_p(l_1 {:+:} x{:})$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0 \,\mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1:+:(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad x_p(l_1{:}{+}{:}x{:})(x :: l)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}+{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}+{:}\mathrm{nil}=l_0}\mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}+{:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:}+{:}x{:})\, L_E^{(l_1{:}+{:}x{:}){:}+{:}l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:}+{:}x{:},x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad x_p(l_1 {:+:} x{:})(x :: l)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:+:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:+:}\mathrm{nil}=l_0}\mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:+:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1{:+:}x{:}){:+:}l_2=l_0}\,(x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:+:}x{:},x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad x_p(l_1{:}{+}{:}x{:})(x :: l)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0} \mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad \lambda_l x_p (l_1 {:+:} x{:})(x :: l)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l (\mathrm{Rev}(l_0,l) \to \bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:+:}l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:+:}\mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:+:}(x :: l_2) = l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1{:+:}x{:})\, L_E^{(l_1{:+:}x{:}){:+:}l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:+:}x{:},x :: l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad \lambda_l x_p (l_1 {:}{+}{:} x{:})(x :: l)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad \lambda_{l_1} \lambda_l x_p (l_1 {:}{+}{:} x{:})(x :: l)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0} \mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \qquad \lambda_{x_p}\lambda_{l_1}\lambda_l x_p(l_1{:+:}x{:})(x :: l)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:+:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:+:}\mathrm{nil}=l_0}\mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:+:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1{:+:}x{:}){:+:}l_2=l_0}\,(x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:+:}x{:},x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L_S]\!]^\circ := \lambda_{x,l_2}\lambda_{x_p}\lambda_{l_1}\lambda_l x_p(l_1{:+:}x{:})(x :: l)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0 \,\mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\,\mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1:+:\mathrm{nil}=l_0}\mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1:+:(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L]\!]^\circ := \quad \mathcal{R}\, l_0 (\lambda_{l_1} x_u)\, (\lambda_{x,l_2,x_p,l_1,l} x_p (l_1{:}{+}{:}x{:}) (x :: l))$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l (\mathrm{Rev}(l_0,l) \to \bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}(x :: l_2) = l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:}, x :: l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![L]\!]^\circ := \quad \lambda_{l_2}\mathcal{R}\, l_2(\lambda_{l_1} x_u)\,(\lambda_{x,l_2,x_p,l_1,l} x_p(l_1{:+:}x{:})(x :: l))$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:+:}l_0=l_0}\, \mathrm{nil}\, R_0\\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S\\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:+:}\mathrm{nil}=l_0}\mathrm{Compat}\; l_1\; l_0\; u\\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:+:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)}\\
&\qquad\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1{:+:}x{:}){:+:}l_2=l_0}\,(x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:+:}x{:},x::l)}\\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \quad \mathcal{R}\, l_0 (\lambda_{l_1} x_u)\, (\lambda_{x,l_2,x_p,l_1,l} x_p (l_1{:}{+}{:}x{:})(x :: l))$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0 \,\text{nil}\, M_E^{\text{nil}{:}{+}{:}l_0=l_0}\, \text{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\text{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \quad \mathcal{R}\, l_0 (\lambda_{l_1} x_u)\,(\lambda_{x,l_2,x_p,l_1,l} x_p (l_1 {:+:} x{:})(x :: l))\mathrm{nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\; L\, l_0\, \mathrm{nil}\; M_E^{\mathrm{nil}{:+:}l_0=l_0}\; \mathrm{nil}\; R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\; l_2\; L_B\; L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:+:}\mathrm{nil}=l_0} \mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:+:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:+:}x{:})\; L_E^{(l_1{:+:}x{:}){:+:}l_2=l_0}\; (x :: l)\; L_{S1}^{\mathrm{Rev}(l_1{:+:}x{:},x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \quad \mathcal{R}\, l_0 (\lambda_{l_1} x_u)\, (\lambda_{x,l_2,x_p,l_1,l} x_p (l_1 {:+:} x{:})(x :: l))\mathrm{nil}$$

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:+:}l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:+:}\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:+:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1{:+:}x{:}){:+:}l_2=l_0}\, (x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:+:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}
$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^{\circ} := \quad \mathcal{R}\, l_0 (\lambda_{l_1} x_u)\, (\lambda_{x,l_2,x_p,l_1,l} x_p (l_1 {:+:} x{:})(x :: l))\,\mathrm{nil}\,\mathrm{nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l (\mathrm{Rev}(l_0,l) \to \bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1 :+: \mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1 :+: (x :: l_2) = l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1 {:+:} x{:})\, L_E^{(l_1 :+: x:) :+: l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1 :+: x:, x :: l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \quad \mathcal{R}\, l_0 (\lambda_{l_1} x_u)\, (\lambda_{x,l_2,x_p,l_1,l} x_p (l_1{:}{+}{:}x{:})(x :: l))\mathrm{nil}\,\mathrm{nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \quad \lambda_{x_u}\mathcal{R}\, l_0(\lambda_{l_1} x_u)\, (\lambda_{x,l_2,x_p,l_1,l} x_p(l_1{:}{+}{:}x{:})(x :: l))\mathrm{nil}\,\mathrm{nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \lambda_{l_0}\lambda_{x_u}\mathcal{R}\, l_0(\lambda_{l_1} x_u)\,(\lambda_{x,l_2,x_p,l_1,l} x_p(l_1{:}{+}{:}x{:})(x :: l))\mathrm{nil\,nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\; L\, l_0\, \mathrm{nil}\; M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\; \mathrm{nil}\; R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\; l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\,\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:}{+}{:}x{:})\; L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x :: l)\; L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \lambda_{l_0}\big(\lambda_{x_u}\mathcal{R}\, l_0 (\lambda_{l_1} x_u)\,(\lambda_{x,l_2,x_p,l_1,l} x_p (l_1 {:+:} x{:})(x :: l))\,\mathrm{nil}\,\mathrm{nil}\big)(\lambda_l l)$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \lambda_{l_0} \mathcal{R}\, l_0 (\lambda_{l_1,l} l)\, (\lambda_{x,l_2,x_p,l_1,l} x_p (l_1 {:+:} x{:}) (x :: l)) \mathrm{nil}\, \mathrm{nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:+:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:+:}\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:+:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1{:+:}x{:}){:+:}l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:+:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^{\circ} := \lambda_{l_0} \mathcal{R}\, l_0 (\lambda_{l_1, l} l)\, (\lambda_{x, l_2, x_p, l_1, l} x_p (l_1 {:}{+}{:} x{:})(x :: l))\mathrm{nil}\,\mathrm{nil}$$

$$
\begin{aligned}
M &:= \lambda_{R_0, R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0, l) \to \bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x, l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x :: l_2) = l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1, l)} \\
&\qquad p\, (l_1 {:}{+}{:} x{:})\, L_E^{(l_1:+:x:):+:l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:, x :: l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}
$$

Extraction by A-translation

Extracted program:

$$\forall_{l_2} \forall^{\mathsf{U}}_{l_1} (l_1 {:+:} l_2 = l_0 \to \forall_l (\mathrm{Rev}(l_1, l) \to \bot))$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)} \, L \, l_0 \, \mathsf{nil} \, M_E^{\mathsf{nil}{:+:}l_0 = l_0} \, \mathsf{nil} \, R_0 \\
L &:= \lambda_{l_2} \mathsf{Ind}_{L(N)} \, l_2 \, L_B \, L_S \\
L_B &:= \lambda^{\mathsf{U}}_{l_1} \lambda_v^{l_1{:+:}\mathsf{nil} = l_0} \mathrm{Compat} \, l_1 \, l_0 \, u \\
L_S &:= \lambda_{x,l_2} \, \lambda_p \, \lambda^{\mathsf{U}}_{l_1} \, \lambda_v^{l_1{:+:}(x :: l_2) = l_0} \, \lambda_l \, \lambda_w^{\mathrm{Rev}(l_1, l)} \\
&\qquad\qquad p \, (l_1 {:+:} x{:}) \, L_E^{(l_1{:+:}x{:}){:+:}l_2 = l_0} \, (x :: l) \, L_{S1}^{\mathrm{Rev}(l_1{:+:}x{:}, x :: l)} \\
L_{S1} &:= R_1 \, x \, l_1 \, l \, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \lambda_{l_0} \mathcal{R}\, l_0 (\lambda_{l_1,l} l)\, (\lambda_{x,l_2,x_p,l_1,l} x_p (l_1{:}{+}{:}x{:})(x :: l))\,\mathrm{nil}\,\mathrm{nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}^{\mathrm{U}} \lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}^{\mathrm{U}}\, \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \lambda_{l_0} \mathcal{R}\, l_0 (\lambda_l l)\, (\lambda_{x,l_2,x_p,l} x_p (x :: l))\, \mathrm{nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l (\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}^{\mathrm{U}} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}^{\mathrm{U}}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by A-translation

Extracted program:

$$[\![M]\!]^\circ := \lambda_{l_0} \mathcal{R}\, l_0 (\lambda_l l)\, (\lambda_{x,l_2,x_p,l} x_p (x :: l))\, \mathrm{nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to\bot)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}^{\mathrm{U}} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}^{\mathrm{U}}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1:+:(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}
$$

Extraction by Dialectica

Extracted program:

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}
$$

Extraction by Dialectica

Extracted program:

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1}\,\lambda_{l_0}\,\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\; L\; l_0\; \mathrm{nil}\; M_E^{\mathrm{nil}:+:l_0=l_0}\; \mathrm{nil}\; R_0 \\
L &:= \lambda_{l_2}\,\mathrm{Ind}_{L(N)}\; l_2\; L_B\; L_S \\
L_B &:= \lambda_{l_1}\,\lambda_v^{l_1:+:\mathrm{nil}=l_0}\,\mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\;\lambda_v^{l_1:+:(x::l_2)=l_0}\;\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1:+:x:)\; L_E^{(l_1:+:x:):+:l_2=l_0}\;(x::l)\; L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}
$$

Extraction by Dialectica

Extracted program:

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}
$$

Extraction by Dialectica

Extracted program:

$$[\![L_B]\!]_u^- := y$$

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)} \; L \; l_0 \; \text{nil} \; M_E^{\text{nil}:+:l_0=l_0} \; \text{nil} \; R_0 \\
L &:= \lambda_{l_2} \text{Ind}_{L(N)} \; l_2 \; L_B \; L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\text{nil}=l_0} \mathrm{Compat} \; l_1 \; l_0 \; u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1:+:(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:}+{:}x{:}) \; L_E^{(l_1:+:x:):+:l_2=l_0} \; (x::l) \; L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1 \; x \; l_1 \; l \; w
\end{aligned}
$$

Extraction by Dialectica

Extracted program:

$$[\![L_B]\!]_u^- := y$$

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1:+:(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}
$$

Extraction by Dialectica

Extracted program:

$$[\![L_B]\!]_u^- := y\lrcorner$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1:+:\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1:+:(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![L_B]\!]^-_u := y\lrcorner \qquad [\![L_S]\!]^-_{R_1} := y$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1:+:(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![L_B]\!]^-_u := y\lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle x, l_1, l\rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:}{+}{:}x{:})\,L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x::l)\,L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]_p^- := y$$

$$[\![L_B]\!]_u^- := y\lrcorner \qquad [\![L_S]\!]_{R_1}^- := \langle x, l_1, l \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x::l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]^-_p := \langle l_1 :+: x :, y \rangle$$

$$[\![L_B]\!]^-_u := y \lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle x, l_1, l \rangle$$

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l (\mathrm{Rev}(l_0,l) \to F)} \, L \, l_0 \, \mathrm{nil} \, M_E^{\mathrm{nil} :+: l_0 = l_0} \, \mathrm{nil} \, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)} \, l_2 \, L_B \, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1 :+: \mathrm{nil} = l_0} \mathrm{Compat} \, l_1 \, l_0 \, u \\
L_S &:= \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1 :+: (x :: l_2) = l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p \, (l_1 :+: x :) \, L_E^{(l_1 :+: x :) :+: l_2 = l_0} \, (x :: l) \, L_{S1}^{\mathrm{Rev}(l_1 :+: x :, x :: l)} \\
L_{S1} &:= R_1 \, x \, l_1 \, l \, w
\end{aligned}
$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]^-_p := \langle l_1{:}{+}{:}x{:}, y\rangle$$

$$[\![L_B]\!]^-_u := y\lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle x, l_1, l\rangle$$

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}
$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]_p^- := \langle l_1{:}{+}{:}x{:}, x :: l\rangle$$

$$[\![L_B]\!]_u^- := y_\lrcorner \qquad [\![L_S]\!]_{R_1}^- := \langle x, l_1, l\rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0\\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S\\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u\\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)}\\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)}\\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]^-_p := \langle l_1{:}{+}{:}x{:},\, x :: l\rangle$$

$$[\![L_B]\!]^-_u := y_\lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle x, l_1, l\rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0\\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S\\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u\\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)}\\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)}\\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]_p^- := \langle l_1{:}{+}{:}x{:}, x :: l \rangle$$

$$[\![L_B]\!]_u^- := y\lrcorner \qquad [\![L_S]\!]_{R_1}^- := \langle x, l_1, l \rangle$$

$$M := \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)} \, L \, l_0 \, \mathrm{nil} \, M_E^{\mathrm{nil}{:}+{:}l_0=l_0} \, \mathrm{nil} \, R_0$$

$$L := \lambda_{l_2} \mathrm{Ind}_{L(N)} \, l_2 \, L_B \, L_S$$

$$L_B := \lambda_{l_1} \lambda_v^{l_1{:}+{:}\mathrm{nil}=l_0} \mathrm{Compat} \, l_1 \, l_0 \, u$$

$$L_S := \lambda_{x,l_2} \, \lambda_p \, \lambda_{l_1} \, \lambda_v^{l_1{:}+{:}(x :: l_2)=l_0} \, \lambda_l \, \lambda_w^{\mathrm{Rev}(l_1,l)}$$

$$p \, (l_1{:}{+}{:}x{:}) \, L_E^{(l_1{:}+{:}x{:}){:}+{:}l_2=l_0} \, (x :: l) \, L_{S1}^{\mathrm{Rev}(l_1{:}+{:}x{:}, x :: l)}$$

$$L_{S1} := R_1 \, x \, l_1 \, l \, w$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]^-_p := \langle l_1{:}{+}{:}x{:}, x :: y\rangle$$

$$[\![L_B]\!]^-_u := y\lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle x, l_1, y\rangle$$

$$M := \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0$$

$$L := \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S$$

$$L_B := \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u$$

$$L_S := \lambda_{x,l_2}\lambda_p\lambda_{l_1}\lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\lambda_l\lambda_w^{\mathrm{Rev}(l_1,l)}$$

$$p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)}$$

$$L_{S1} := R_1\, x\, l_1\, l\, w$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]^-_p := \langle l_1{:}{+}{:}x{:}, x :: y\rangle$$

$$[\![L_B]\!]^-_u := y_\lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle x, l_1, y\rangle$$

$$
\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}+{:}l_0=l_0}\, \mathrm{nil}\, R_0\\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S\\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}+{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u\\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}+{:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)}\\
&\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}+{:}x{:}){:}+{:}l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:}+{:}x{:},x::l)}\\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}
$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]_p^- := \langle y_\llcorner{:}{+}{:}x{:}, x :: y_\lrcorner \rangle$$

$$[\![L_B]\!]_u^- := y_\lrcorner \qquad [\![L_S]\!]_{R_1}^- := \langle x, y_\llcorner, y_\lrcorner \rangle$$

$$M := \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0$$

$$L := \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S$$

$$L_B := \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u$$

$$L_S := \lambda_{x,l_2} \lambda_p \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0} \lambda_l \lambda_w^{\mathrm{Rev}(l_1,l)}$$

$$p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)}$$

$$L_{S1} := R_1\, x\, l_1\, l\, w$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]^+ := \lambda_y \langle y_\llcorner{:}{+}{:}x{:}, x :: y_\lrcorner \rangle$$

$$[\![L_B]\!]^-_u := y_\lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle x, y_\llcorner, y_\lrcorner \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)} \; L\, l_0 \,\mathrm{nil}\, M_E^{\mathrm{nil}{:}+{:}l_0=l_0} \,\mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}+{:}\mathrm{nil}=l_0} \mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}+{:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:}+{:}x{:})\, L_E^{(l_1{:}+{:}x{:}){:}+{:}l_2=l_0}\, (x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:}+{:}x{:},x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]^+ := \lambda_{x,l_2}\lambda_y \, \langle y_{\llcorner}{:}{+}{:}x{:}, x :: y_{\lrcorner}\rangle$$

$$[\![L_B]\!]^-_u := y_{\lrcorner} \qquad [\![L_S]\!]^-_{R_1} := \langle y_{\llcorner}, y_{\lrcorner\lrcorner\llcorner}, y_{\lrcorner\lrcorner\lrcorner}\rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1}\lambda_{l_0}\lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2}\mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}\lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0}\mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\,\lambda_p\,\lambda_{l_1}\,\lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\,\lambda_l\,\lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\,(x::l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]^+ := \lambda_{x,l_2}\lambda_y \, \langle y_{\llcorner} {:}{+}{:} x{:},\, x :: y_{\lrcorner} \rangle$$

$$[\![L_B]\!]^-_u := y_{\lrcorner} \qquad [\![L_S]\!]^-_{R_1} := \langle y_{\llcorner}, y_{\lrcorner\lrcorner\llcorner}, y_{\lrcorner\lrcorner\lrcorner} \rangle$$

$$[\![L]\!]^-_i := \mathcal{R}_{\mathbb{L}(\rho)}\, l\, (\lambda_{y_A} [\![L_B]\!]^-_i) \left(\lambda_{x,l,p,y} \begin{array}{c} [\![L_S]\!]^-_i\,[y_S := \langle x, l, y \rangle] \\ \overset{i}{\bowtie} \\ p\big([\![L_S]\!]^+\, x\, l\, y\big) \end{array} \right)$$

Extraction by Dialectica

Extracted program:

$$[\![L_S]\!]^+ := \lambda_{x,l_2}\lambda_y \, \langle y_{\llcorner}{:}{+}{:}x{:}, x :: y_{\lrcorner} \rangle$$

$$[\![L_B]\!]^-_u := y_{\lrcorner} \qquad [\![L_S]\!]^-_{R_1} := \langle y_{\llcorner}, y_{\lrcorner\lrcorner\llcorner}, y_{\lrcorner\lrcorner\lrcorner} \rangle$$

$$[\![L]\!]^-_i := \mathcal{R}_{\mathbb{L}(\rho)}\, l\, (\lambda_{y_A}[\![L_B]\!]^-_i) \left(\lambda_{x,l,p,y} \begin{array}{c} [\![L_S]\!]^-_i \, [y_S := \langle x, l, y \rangle] \\ \overset{i}{\bowtie} \\ p([\![L_S]\!]^+ \, x\, l\, y) \end{array} \right)$$

Cannot contract neither on u, nor on R_1!

Extraction by Dialectica

Extracted program:

$$[\![L]\!]^-_u := \mathcal{R}\, l_2\, (\lambda_y y_\lrcorner)\, \left(\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:}{+}{:}x{:}, x :: y_\lrcorner \rangle\right)$$

$$[\![L_B]\!]^-_u := y_\lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle y_\llcorner, y_{\lrcorner\lrcorner\llcorner}, y_{\lrcorner\lrcorner\lrcorner} \rangle$$

$$[\![L]\!]^-_i := \mathcal{R}_{\mathbb{L}(\rho)}\, l\, (\lambda_{y_A} [\![L_B]\!]^-_i) \left(\lambda_{x,l,p,y} \begin{array}{c} [\![L_S]\!]^-_i\, [y_S := \langle x, l, y \rangle] \\ \overset{i}{\bowtie} \\ p([\![L_S]\!]^+\, x\, l\, y) \end{array} \right)$$

Cannot contract neither on u, nor on R_1!

For u we don't need to contract

Extraction by Dialectica

Extracted program:

$$R_1 : \quad \forall^{\mathsf{U}}_{x,l_1,l_2} \left(\mathrm{Rev}(l_1, l_2) \to \mathrm{Rev}(l_1{:}{+}{:}x{:}, x :: l_2)\right)$$

$$[\![L_B]\!]^-_u := y_\lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle y_\llcorner, y_{\lrcorner\lrcorner\llcorner}, y_{\lrcorner\lrcorner\lrcorner} \rangle$$

$$[\![L]\!]^-_i := \mathcal{R}_{\mathbb{L}(\rho)}\, l\, (\lambda_{y_A} [\![L_B]\!]^-_i) \left(\lambda_{x,l,p,y} \begin{array}{c} [\![L_S]\!]^-_i \, [y_S := \langle x, l, y \rangle] \\ \overset{i}{\bowtie} \\ p([\![L_S]\!]^+ \, x\, l\, y) \end{array} \right)$$

Cannot contract neither on u, nor on R_1!
For u we don't need to contract , for R_1 we need uniform quantifiers

Extraction by Dialectica

Extracted program:

$$R_1 : \quad \forall^{\mathrm{U}}_{x,l_1,l_2} \left(\mathrm{Rev}(l_1, l_2) \rightarrow \mathrm{Rev}(l_1{:+:}x{:}, x :: l_2)\right)$$

$$[\![L_B]\!]^-_u := y_\lrcorner \qquad [\![L_S]\!]^-_{R_1} := \langle y_\llcorner, y_{\lrcorner\lrcorner\llcorner}, y_{\lrcorner\lrcorner\lrcorner} \rangle$$

$$[\![L]\!]^-_i := \mathcal{R}_{\mathbb{L}(\rho)}\, l\, (\lambda_{y_A} [\![L_B]\!]^-_i) \left(\lambda_{x,l,p,y} \begin{array}{c} [\![L_S]\!]^-_i\, [y_S := \langle x, l, y \rangle] \\ \overset{i}{\bowtie} \\ p([\![L_S]\!]^+\, x\, l\, y) \end{array} \right)$$

Cannot contract neither on u, nor on R_1!
For u we don't need to contract , for R_1 we need uniform quantifiers

Extraction by Dialectica

Extracted program:

$$[\![L]\!]^{-}_{u} := \mathcal{R}\, l_2\, (\lambda_y y_\lrcorner)\, \big(\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner \rangle\, \big)\, y$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \text{nil}\, M_E^{\text{nil}:+:l_0=l_0}\, \text{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\text{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]_u^- := \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, \big(\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner\rangle\,\big)\, y$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^{-}_{u} := \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, (\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner \rangle\,)\, \langle \mathrm{nil}, y \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]_u^- := \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, (\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:+:}x{:},\, x :: y_\lrcorner \rangle\,)\, \langle \mathrm{nil}, y \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\, (l_1{:+:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^{-}_{u} := \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, (\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner\rangle\,)\, \langle \mathrm{nil}, \mathrm{nil}\rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]_u^- := \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, \left(\lambda_{x,l_2,p,y} p \,\langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner\rangle\,\right) \langle \mathrm{nil}, \mathrm{nil}\rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\; L\; l_0\; \mathrm{nil}\; M_E^{\mathrm{nil}:+:l_0=l_0}\; \mathrm{nil}\; R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\; l_2\; L_B\; L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\; l_1\; l_0\; u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\,(l_1{:}{+}{:}x{:})\; L_E^{(l_1:+:x:):+:l_2=l_0}\; (x :: l)\; L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\; x\; l_1\; l\; w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^+ := \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, \big(\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner\rangle\big)\, \langle \mathrm{nil}, \mathrm{nil}\rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,\, x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^+ := \lambda_{l_0} \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, \big(\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner \rangle\, \big)\, \langle \mathrm{nil}, \mathrm{nil} \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}{+}{:}(x :: l_2) = l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},\, x :: l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^+ := \lambda_{l_0} \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, (\lambda_{x,l_2,p,y} p \langle y_\llcorner {:}{+}{:} x{:}, x :: y_\lrcorner \rangle)\, \langle \mathrm{nil}, \mathrm{nil} \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}{+}{:}(x :: l_2) = l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:}, x :: l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^+ := \lambda_{l_0} \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, \big(\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner \rangle\big)\, \langle \mathrm{nil}, \mathrm{nil} \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1:+:\mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1:+:x:):+:l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,\, x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^+ := \lambda_{l_0} \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, (\lambda_{x,l_2,p,y} p \langle y_\llcorner{:}{+}{:}x{:}, x :: y_\lrcorner \rangle)\, \langle \mathrm{nil}, \mathrm{nil} \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1} \lambda_v^{l_1{:}{+}{:}\mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}\, \lambda_v^{l_1{:}{+}{:}(x :: l_2) = l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:}, x :: l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^+ := \lambda_{l_0} \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, (\lambda_{x,l_2,p,y} p\, \langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner \rangle\,)\, \langle \mathrm{nil}, \mathrm{nil} \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0=l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}^{\mathrm{U}} \lambda_v^{l_1:+:\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}^{\mathrm{U}}\, \lambda_v^{l_1:+:(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1:+:x:):+:l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:,x::l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

▸ Check #1 ▸ Check #2

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^+ := \lambda_{l_0} \mathcal{R}\, l_0\, (\lambda_y y_\lrcorner)\, (\lambda_{x,l_2,p,y} p \langle y_\llcorner{:}{+}{:}x{:},\, x :: y_\lrcorner \rangle)\, \langle \mathrm{nil}, \mathrm{nil} \rangle$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}^{\mathsf{U}} \lambda_v^{l_1{:}{+}{:}\mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}^{\mathsf{U}}\, \lambda_v^{l_1{:}{+}{:}(x :: l_2) = l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},\, x :: l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^+ := \lambda_{l_0} \mathcal{R}\, l_0\, (\lambda_y y)\, \big(\lambda_{x,l_2,p,y} p(x :: y)\big) \mathrm{nil}$$

$$\begin{aligned}
M &:= \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l (\mathrm{Rev}(l_0,l) \to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}:+:l_0 = l_0}\, \mathrm{nil}\, R_0 \\
L &:= \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S \\
L_B &:= \lambda_{l_1}^{\mathrm{U}} \lambda_v^{l_1:+:\mathrm{nil} = l_0} \mathrm{Compat}\, l_1\, l_0\, u \\
L_S &:= \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}^{\mathrm{U}}\, \lambda_v^{l_1:+:(x::l_2) = l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)} \\
&\qquad p\, (l_1:+:x:)\, L_E^{(l_1:+:x:):+:l_2 = l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1:+:x:, x :: l)} \\
L_{S1} &:= R_1\, x\, l_1\, l\, w
\end{aligned}$$

Extraction by Dialectica

Extracted program:

$$[\![M]\!]^+ := \lambda_{l_0} \mathcal{R}\, l_0\, (\lambda_y y)\, \big(\lambda_{x,l_2,p,y} p(x :: y)\big) \mathrm{nil}$$

$$[\![M]\!]^\circ := \lambda_{l_0} \mathcal{R}\, l_0\, (\lambda_l l)\, (\lambda_{x,l_2,x_p,l} x_p(x :: l))\, \mathrm{nil}$$

$$M := \lambda_{R_0,R_1} \lambda_{l_0} \lambda_u^{\forall_l(\mathrm{Rev}(l_0,l)\to F)}\, L\, l_0\, \mathrm{nil}\, M_E^{\mathrm{nil}{:}{+}{:}l_0 = l_0}\, \mathrm{nil}\, R_0$$

$$L := \lambda_{l_2} \mathrm{Ind}_{L(N)}\, l_2\, L_B\, L_S$$

$$L_B := \lambda_{l_1}^{\mathrm{U}} \lambda_v^{l_1{:}{+}{:}\mathrm{nil}=l_0} \mathrm{Compat}\, l_1\, l_0\, u$$

$$L_S := \lambda_{x,l_2}\, \lambda_p\, \lambda_{l_1}^{\mathrm{U}}\, \lambda_v^{l_1{:}{+}{:}(x::l_2)=l_0}\, \lambda_l\, \lambda_w^{\mathrm{Rev}(l_1,l)}$$

$$p\,(l_1{:}{+}{:}x{:})\, L_E^{(l_1{:}{+}{:}x{:}){:}{+}{:}l_2=l_0}\, (x :: l)\, L_{S1}^{\mathrm{Rev}(l_1{:}{+}{:}x{:},x::l)}$$

$$L_{S1} := R_1\, x\, l_1\, l\, w$$

Conclusions and future work

Uniformities

- are quantifier annotations
- express lack of computational dependence
- can remove unnecessary parameters
- can improve complexity
- can remove unnecessary contractions in Dialectica
- could be refined for Dialectica
- could be inferred automatically for a given proof of a fixed formula

Conclusions and future work

Uniformities

- are quantifier annotations
- express lack of computational dependence
- can remove unnecessary parameters
- can improve complexity
- can remove unnecessary contractions in Dialectica
- could be refined for Dialectica
- could be inferred automatically for a given proof of a fixed formula

Conclusions and future work

Uniformities

- are quantifier annotations
- express lack of computational dependence
- can remove unnecessary parameters
- can improve complexity
- can remove unnecessary contractions in Dialectica
- could be refined for Dialectica
- could be inferred automatically for a given proof of a fixed formula

Conclusions and future work

Uniformities

- are quantifier annotations
- express lack of computational dependence
- can remove unnecessary parameters
- can improve complexity
- can remove unnecessary contractions in Dialectica
- could be refined for Dialectica
- could be inferred automatically for a given proof of a fixed formula

Conclusions and future work

Uniformities

- are quantifier annotations
- express lack of computational dependence
- can remove unnecessary parameters
- can improve complexity
- can remove unnecessary contractions in Dialectica
- could be refined for Dialectica
- could be inferred automatically for a given proof of a fixed formula

Conclusions and future work

Uniformities

- are quantifier annotations
- express lack of computational dependence
- can remove unnecessary parameters
- can improve complexity
- can remove unnecessary contractions in Dialectica
- could be refined for Dialectica
- could be inferred automatically for a given proof of a fixed formula

Conclusions and future work

Uniformities

- are quantifier annotations
- express lack of computational dependence
- can remove unnecessary parameters
- can improve complexity
- can remove unnecessary contractions in Dialectica
- could be refined for Dialectica
- could be inferred automatically for a given proof of a fixed formula

Thank you

Thank you for your attention