Rational sequences in dependently typed programming

Niccolò Veltri
joint work with James Chapman, Tarmo Uustalu

Representing Streams II, Leiden, 24 January 2014
Finiteness and rational datatypes

- Rational datatypes are types of trees with finitely many distinct (in the sense of bisimilarity) subtrees.
- In constructive logic or type theory, there is no single "right" notion of finiteness of a set (or a subset of a set), there are several inequivalent definitions.
- Which notions of finiteness should we use??
- For sets of subtrees, the important finiteness notions coincide.
- Rational datatypes lie between inductive and coinductive datatypes and have traits of both.
Listability

- Listability says we can fit all elements of $A$ into a list.

- \[
\begin{array}{c}
\hline
[] : \text{List } A \\
x : A \quad xs : \text{List } A \\
\hline
\end{array}
\]

\[
x :: xs : \text{List } A
\]

- \[
\begin{array}{c}
y \equiv x \\
y \in x :: xs
\end{array}
\]

- \[
\begin{array}{c}
y \in xs \\
y \in x :: xs
\end{array}
\]

(single rule lines—inductive definitions)

- Listable $A = \Sigma xs : \text{List } A. \Pi x : A. x \in xs$

- (Note that we allow duplicates in the listing.)

- From a listability proof, we can learn an upper bound on the size of $A$. 

Noetherianness

- Noetherianness says, if we are shown elements from $A$ one after another, sooner or later we will have seen some element twice.

\[
\begin{align*}
\frac{x \in xs}{\text{Dup} (x :: xs)} & \quad \frac{\text{Dup} xs}{\text{Dup} (y :: xs)} \\
\frac{\text{Dup} xs}{\Pi x : A. \, \text{Noeth}'_{x :: xs} A} & \quad \frac{}{\text{Noeth}'_{xs} A}
\end{align*}
\]

- From a proof of Noetherianness we can generally infer no bound on the size of $A$. 

$$\text{Noeth} A = \text{Noeth}' [] A$$
Listability vs Noetherianness

- Listability is stronger than Noetherianness:
  
  \[ \text{Listable } A \rightarrow \text{Noeth } A \]

- The converse is generally not true.

- Listability is not closed under (general) subsets, Noetherianness is.
  
  So Noetherianness is more robust.
Sequences

- Sequences are like lists, but they may be infinitely long.

\[
\begin{align*}
[] & : \text{Seq } A \\
A & : A \quad A & : \text{Seq } A \\
x & :: A \quad x & :: A \quad xs & : \text{Seq } A \\
\end{align*}
\]

(coinductive definition—double rule lines)

- Bisimilarity:

\[
\begin{align*}
[] & \sim [] \\
x & :: xs \sim x & :: ys \\
xs & \sim ys \\
\end{align*}
\]

- Extensionality principle for sequences:

\[xs \sim ys \rightarrow xs \equiv ys.\]

(Cf. extensionality principle for functions.)
Corecursion for sequences

Corecursion for sequences says that state machines define sequences:

\[
\text{unfold} : (C \to 1 + A \times C) \to C \to \text{Seq} A
\]

\[
\text{unfold } f \ z = \text{case } f \ z \text{ of}
\]

\[
\text{inl } * = []
\]

\[
\text{inr } (x, z') = x :: \text{unfold } f \ (f \ z')
\]

(by corecursion)
Finiteness of state-machine generated subsets

- Given \( f : C \to 1 + C, z : C \), define

\[
f^\Delta : C \to 1 + C \times C
\]

\[
f^\Delta z = \text{case } f z \text{ of}
\]

\[
\begin{align*}
\text{inl } \ast &= \text{inl } \ast \\
\text{inr } z' &= \text{inr } (z', z')
\end{align*}
\]

\[
\text{Gen } f z = \{ x : C . x \in \text{unfold } f^\Delta z \}
\]

- For such state-machine generated subsets of \( C \), Noetherianness implies listability:

\[
\text{Noeth } (\text{Gen } f z) \to \text{Listable } (\text{Gen } f z)
\]

\((C \text{ need not be finite for this})\)
Rational sequences

- Rational sequences are sequences with a finite number of distinct subsequences.

\[
\begin{align*}
y s & \equiv x s \\
\Rightarrow & \\
ys & \trianglelefteq xs
\end{align*}
\]

\[
\begin{align*}
ys & \trianglelefteq xs \\
\Rightarrow & \\
y s & \trianglelefteq (x :: xs)
\end{align*}
\]

\[
\text{Subs } xs = \{ ys : \text{Seq A. } ys \trianglelefteq xs \}
\]

\[
\text{SeqRN A} = \{ xs : \text{Seq A. Noeth (Subs xs)} \}
\]

- The set of subsequence of a sequences is state-machine generated, so its Noetherianness is equivalent to listability, hence

\[
\text{SeqRN A} \cong \text{SeqRL A} = \{ xs : \text{Seq A. Listable (Subs xs)} \}
\]
Corecursion for rational sequences

- Rational sequences support corecursion like sequences, but for finite state spaces only.
- It suffices with a Noetherian state space, listability is not necessary.
- The “rational corecursor”

\[
\text{unfoldR : Noeth } C \to (C \to 1 + A \times C) \to C \to \text{SeqRN } A
\]

is defined by recursion on the given proof of Noetherianness.

- Example: Conversion of fractions to decimal representations.
Formalization in Agda

- Ongoing formalization in Agda, not completed
- Technicalities: programming with coinductive types, subsets, quotients is tricky in Agda...