Correct-by-Construction Pretty-Printing

Nils Anders Danielsson

DTP 2013, Boston, 2013-09-24

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement n° 247219. This presentation does not necessarily reflect the views of the ERC or the EU. The EU is not liable for any use of the presented information. The European emblem is registered with WIPO under article 6ter of the Paris Convention in the name of the Council of Europe. The ERC logo is perhaps copyrighted by the EU or the ERC.

Pretty-printing

$$\textsf{add}\ (\textsf{mul}\ 1\ 2)\ (\textsf{mul}\ 3\ (\textsf{add}\ 4\ 5))$$

$$\Downarrow$$

```
1 * 2 + 3 * (4 + 5)
```

```
1 * 2
  +
3 * (4 + 5)
```

Classical pretty-printing combinators

Implemented by user:

$$pretty : A \to Doc$$

Combinator interface:

$$\begin{aligned}
&Doc &&: Set \\
&render &&: \mathbb{N} \to Doc \to String \\
&\mathsf{text} &&: String \to Doc \\
&_\Diamond_ &&: Doc \to Doc \to Doc \\
&\mathsf{line} &&: Doc \\
&\vdots
\end{aligned}$$

Correctness

Focus: Grammatical correctness, not “prettiness”.

Correctness

Grammars with semantic actions:

$$Grammar : Set \rightarrow Set$$

Relational semantics:

$$\_\in\_\cdot\_ : A \rightarrow Grammar\ A \rightarrow String \rightarrow Set$$

$x \in g \cdot s$ means that the string s and value x are generated by g.

Correctness

Round-tripping property:

$$\forall\; w\; x\; \rightarrow\; x\; \in\; g\; \cdot\; \mathit{render}\; w\; (\mathit{pretty}\; x)$$

Correctness

Round-tripping property:

$$\forall\ w\ x\ \rightarrow\ x\ \in\ g\ \cdot\ render\ w\ (pretty\ x)$$

This work

Strong types are used to ensure that the round-tripping property holds by construction.

Documents

Before:

$$Doc \;:\; Set$$

Now:

$$Doc \;:\; Grammar\; A \;\rightarrow\; A \;\rightarrow\; Set$$

Pretty-printers

Before:

$$pretty : A \to Doc$$

Now:

$$\begin{aligned} g &: Grammar\ A \\ pretty &: (x : A) \to Doc\ g\ x \end{aligned}$$

Renderers

Before:

$$render : \mathbb{N} \rightarrow Doc \rightarrow String$$

Now:

$$\begin{aligned} render &: \mathbb{N} \rightarrow Doc\; g\; x \rightarrow String \\ parsable &: \forall\; w\; (d : Doc\; g\; x) \rightarrow \\ &\qquad x \in g \cdot render\; w\; d \end{aligned}$$

Renderers

Now:

$$
\begin{aligned}
&render &&: \mathbb{N} \rightarrow Doc\ g\ x \rightarrow String \\
&parsable &&: \forall\ w\ (d : Doc\ g\ x) \rightarrow \\
& && \qquad x \in g \cdot render\ w\ d
\end{aligned}
$$

Correctness (by construction):

$$\forall\ w\ x \rightarrow x \in g \cdot render\ w\ (pretty\ x)$$

Overview

In talk:

- Grammars.
- Documents.
- Simple examples.

Not in talk:

- Renderer (based on Wadler's).

Grammars

For simplicity: regular expressions.

$$
\begin{array}{lll}
\multicolumn{3}{l}{\textbf{data}\ \mathit{Grammar} : \mathit{Set} \to \mathit{Set}_1\ \textbf{where}} \\
\quad \emptyset & : & \mathit{Grammar}\ A \\
\quad \varepsilon & : & A \to \mathit{Grammar}\ A \\
\quad \mathsf{string} & : & \mathit{String} \to \mathit{Grammar}\ \mathit{String} \\
\quad \_\circledast\_ & : & \mathit{Grammar}\ (A \to B) \to \\
 & & \mathit{Grammar}\ A \to \mathit{Grammar}\ B \\
\quad \_|\_ & : & \mathit{Grammar}\ A \to \mathit{Grammar}\ A \to \\
 & & \mathit{Grammar}\ A \\
\quad \_\star & : & \mathit{Grammar}\ A \to \mathit{Grammar}\ (\mathit{List}\ A)
\end{array}
$$

Semantics of grammars

$$\frac{}{x \in \varepsilon\, x \cdot [\,]} \qquad \frac{}{s \in \mathsf{string}\, s \cdot s}$$

$$\frac{f \in g_1 \cdot s_1 \qquad x \in g_2 \cdot s_2}{f\, x \in g_1 \circledast g_2 \cdot s_1 +\!\!+\, s_2}$$

$$\frac{x \in g_1 \cdot s}{x \in g_1 \mid g_2 \cdot s} \qquad \frac{x \in g_2 \cdot s}{x \in g_1 \mid g_2 \cdot s}$$

$$\frac{xs \in \varepsilon\, [\,] \mid (\varepsilon\, \_{::}\_ \circledast g) \circledast g \star \cdot s}{xs \in g \star \cdot s}$$

Some grammar combinators

$$\_\lessdot\!\circledast\_ : Grammar\ A \to Grammar\ B \to Grammar\ A$$

$$g_1 \lessdot\!\circledast\ g_2 = (\varepsilon\ (\lambda\ x\ \_ \to x) \circledast g_1) \circledast g_2$$

$$\_+ : Grammar\ A \to Grammar\ (List\ A)$$

$$g + = (\varepsilon\ \_{::}\_ \circledast g) \circledast g \star$$

$$whitespace : Grammar\ String$$

$$whitespace = \mathsf{string}\ \texttt{" "} \mid \mathsf{string}\ \texttt{"\textbackslash n"}$$

Documents

$$
\begin{array}{lll}
\textbf{data}\ \mathit{Doc} & : & \mathit{Grammar}\ A \to A \to \mathit{Set}_1\ \textbf{where} \\
\_\diamond\_ & : & \mathit{Doc}\ g_1\ f \to \mathit{Doc}\ g_2\ x \to \\
 & & \mathit{Doc}\ (g_1 \circledast g_2)\ (f\ x) \\
\mathsf{text} & : & \mathit{Doc}\ (\mathsf{string}\ s)\ s \\
\mathsf{line} & : & \mathit{Doc}\ (\varepsilon\ \mathsf{unit} \lessdot\!\circledast\ \mathit{whitespace}\ +)\ \mathsf{unit} \\
\mathsf{nest} & : & \mathbb{N} \to \mathit{Doc}\ g\ x \to \mathit{Doc}\ g\ x \\
\mathsf{group} & : & \mathit{Doc}\ g\ x \to \mathit{Doc}\ g\ x \\
\mathsf{embed} & : & (\forall\ s \to x_1 \in g_1 \cdot s \to x_2 \in g_2 \cdot s) \to \\
 & & \mathit{Doc}\ g_1\ x_1 \to \mathit{Doc}\ g_2\ x_2
\end{array}
$$

Defined document combinators

To handle $\_|\_$:

$$
\begin{aligned}
&\mathit{left} : \mathit{Doc}\ g_1\ x \to \mathit{Doc}\ (g_1 \mid g_2)\ x \\
&\mathit{left}\ d = \mathsf{embed}\ \ldots\ d \\
&\mathit{right} : \mathit{Doc}\ g_2\ x \to \mathit{Doc}\ (g_1 \mid g_2)\ x \\
&\mathit{right}\ d = \mathsf{embed}\ \ldots\ d
\end{aligned}
$$

Embedding proofs:

$$\forall\ s \to x \in g_1 \cdot s \to x \in g_1 \mid g_2 \cdot s$$

$$\forall\ s \to x \in g_2 \cdot s \to x \in g_1 \mid g_2 \cdot s$$

Defined document combinators

To handle ε:

$$
\begin{aligned}
\mathit{empty} &: \mathit{Doc}\ (\varepsilon\ x)\ x \\
\mathit{empty} &= \mathsf{embed}\ \ldots\ (\mathsf{text}\ \{s\ =\ ""\})
\end{aligned}
$$

To handle $\_\lessdot\!\circledast\_$:

$$
\begin{aligned}
\_\lessdot\!\diamond\_ &: \mathit{Doc}\ g_1\ x\ \rightarrow\ \mathit{Doc}\ g_2\ y\ \rightarrow \\
&\quad\ \mathit{Doc}\ (g_1 \lessdot\!\circledast\ g_2)\ x \\
d_1 \lessdot\!\diamond\ d_2 &= (\mathit{empty} \diamond d_1) \diamond d_2
\end{aligned}
$$

Recall that

$$
g_1 \lessdot\!\circledast\ g_2\ =\ (\varepsilon\ (\lambda\ x\ \_\ \rightarrow\ x) \circledast g_1) \circledast g_2.
$$

Simple example

$$bit : Grammar\ Bool$$
$$bit = \varepsilon\ \mathsf{true} \circledast\!\!\lessdot\ \mathsf{string}\ \texttt{"1"}$$
$$\quad | \ \varepsilon\ \mathsf{false} \circledast\!\!\lessdot\ \mathsf{string}\ \texttt{"0"}$$

$$bit_{\mathsf{P}} : (b : Bool) \to Doc\ bit\ b$$
$$bit_{\mathsf{P}}\ b = ? \quad \texttt{--}\ Doc\ bit\ b$$

Simple example

$$
\begin{aligned}
&bit \;:\; Grammar\ Bool \\
&bit \;=\; \varepsilon\ \mathsf{true} \;\circledast\; \mathsf{string}\ \texttt{"1"} \\
&\quad\;\;\mid\; \varepsilon\ \mathsf{false} \;\circledast\; \mathsf{string}\ \texttt{"0"}
\end{aligned}
$$

$$
\begin{aligned}
&bit_{\mathsf{P}} \;:\; (b \;:\; Bool) \;\to\; Doc\ bit\ b \\
&bit_{\mathsf{P}}\ \mathsf{true} \;=\; ? \quad \texttt{--}\ Doc\ bit\ \mathsf{true} \\
&bit_{\mathsf{P}}\ \mathsf{false} \;=\; ? \quad \texttt{--}\ Doc\ bit\ \mathsf{false}
\end{aligned}
$$

Simple example

$$
\begin{aligned}
&bit \;:\; Grammar\ Bool \\
&bit \;=\; \varepsilon\ \mathsf{true}\ \circledast\ \mathsf{string}\ \texttt{"1"} \\
&\quad\;\mid\; \varepsilon\ \mathsf{false}\ \circledast\ \mathsf{string}\ \texttt{"0"}
\end{aligned}
$$

$$
\begin{aligned}
&bit_{\mathsf{P}} \;:\; (b \;:\; Bool) \;\to\; Doc\ bit\ b \\
&bit_{\mathsf{P}}\ \mathsf{true} \;=\; left\ ? \quad \text{-- } Doc\ (\varepsilon\ \mathsf{true}\ \circledast\ \mathsf{string}\ \texttt{"1"}) \\
&\qquad\qquad\qquad\qquad\quad \text{-- } \qquad\quad \mathsf{true} \\
&bit_{\mathsf{P}}\ \mathsf{false} \;=\; ? \qquad\quad \text{-- } Doc\ bit\ \mathsf{false}
\end{aligned}
$$

Simple example

$$
\begin{aligned}
&bit : Grammar\ Bool \\
&bit = \varepsilon\ \mathsf{true} \circledast\!\!\lessdot\ \mathsf{string}\ \texttt{"1"} \\
&\quad\ \mid\ \varepsilon\ \mathsf{false} \circledast\!\!\lessdot\ \mathsf{string}\ \texttt{"0"}
\end{aligned}
$$

$$
\begin{aligned}
&bit_{\mathsf{P}} : (b : Bool) \rightarrow Doc\ bit\ b \\
&bit_{\mathsf{P}}\ \mathsf{true} = left\ (? \diamond ?) && \text{-- } Doc\ (\varepsilon\ \mathsf{true})\ \mathsf{true} \\
& && \text{-- } Doc\ (\mathsf{string}\ \texttt{"1"})\ s \\
&bit_{\mathsf{P}}\ \mathsf{false} = ? && \text{-- } Doc\ bit\ \mathsf{false}
\end{aligned}
$$

Simple example

$$
\begin{aligned}
&\mathit{bit} \;:\; \mathit{Grammar}\ \mathit{Bool}\\
&\mathit{bit} \;=\; \varepsilon\ \mathsf{true} \ \lessdot\!\circledast\ \mathsf{string}\ \texttt{"1"}\\
&\qquad\mid\ \varepsilon\ \mathsf{false} \ \lessdot\!\circledast\ \mathsf{string}\ \texttt{"0"}
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{bit}_{\mathsf{P}} \;:\; (b \;:\; \mathit{Bool}) \;\to\; \mathit{Doc}\ \mathit{bit}\ b\\
&\mathit{bit}_{\mathsf{P}}\ \mathsf{true} \;=\; \mathit{left}\ (\mathit{empty} \lessdot\!\diamond\ ?) && \text{-- } \mathit{Doc}\ (\mathsf{string}\ \texttt{"1"})\ s\\
&\mathit{bit}_{\mathsf{P}}\ \mathsf{false} \;=\; ? && \text{-- } \mathit{Doc}\ \mathit{bit}\ \mathsf{false}
\end{aligned}
$$

Simple example

$$
\begin{aligned}
&bit \;:\; Grammar\ Bool\\
&bit \;=\; \varepsilon\ \mathsf{true} \ \circledast\!\!< \ \mathsf{string}\ \texttt{"1"}\\
&\qquad \mid\ \varepsilon\ \mathsf{false} \ \circledast\!\!< \ \mathsf{string}\ \texttt{"0"}
\end{aligned}
$$

$$
\begin{aligned}
&bit_{\mathsf{P}} \;:\; (b \;:\; Bool) \;\to\; Doc\ bit\ b\\
&bit_{\mathsf{P}}\ \mathsf{true} \;=\; left\ (empty \diamond\!\!< \ \mathsf{text})\\
&bit_{\mathsf{P}}\ \mathsf{false} \;=\; ? \qquad\qquad \text{-- } Doc\ bit\ \mathsf{false}
\end{aligned}
$$

Simple example

$$\begin{aligned}
&bit : Grammar\ Bool \\
&bit = \varepsilon\ \mathsf{true} \ \circledast\!\!< \mathsf{string}\ \texttt{"1"} \\
&\quad\ \mid\ \varepsilon\ \mathsf{false} \ \circledast\!\!< \mathsf{string}\ \texttt{"0"}
\end{aligned}$$

$$\begin{aligned}
&bit_{\mathsf{P}} : (b : Bool) \to Doc\ bit\ b \\
&bit_{\mathsf{P}}\ \mathsf{true} = left\ (empty \lhd\!\!\diamond\ \mathsf{text}) \\
&bit_{\mathsf{P}}\ \mathsf{false} = right\ (empty \lhd\!\!\diamond\ \mathsf{text})
\end{aligned}$$

Simple example

$$
\begin{aligned}
&bit \;:\; Grammar\ Bool \\
&bit \;=\; \varepsilon\ \mathsf{true} \circledast\!\!\lessdot\ \mathsf{string}\ \texttt{"1"} \\
&\qquad \mid\ \varepsilon\ \mathsf{false} \circledast\!\!\lessdot\ \mathsf{string}\ \texttt{"0"}
\end{aligned}
$$

$$
\begin{aligned}
&bit_{\mathsf{P}} \;:\; (b : Bool) \to Doc\ bit\ b \\
&bit_{\mathsf{P}}\ \mathsf{true} \;=\; left\ \ (empty \lessdot\!\!\diamond\ \mathsf{text}) \\
&bit_{\mathsf{P}}\ \mathsf{false} \;=\; right\ (empty \lessdot\!\!\diamond\ \mathsf{text})
\end{aligned}
$$

$$render\ 10\ (bit_{\mathsf{P}}\ \mathsf{false}) \equiv \texttt{"0"} \qquad \mathsf{false} \in bit \cdot \texttt{"0"}$$

$$render\ 1\ (bit_{\mathsf{P}}\ \mathsf{true}) \equiv \texttt{"1"} \qquad \mathsf{true} \in bit \cdot \texttt{"1"}$$

More defined document combinators

To handle $\_\star$:

$$
\begin{aligned}
&nil \ : \ Doc\ (g\ \star)\ [\,] \\
&nil \ = \ \textsf{embed}\ \ldots\ empty
\end{aligned}
$$

$$
\begin{aligned}
&cons \ : \ Doc\ g\ x \ \rightarrow \ Doc\ (g\ \star)\ xs \ \rightarrow \\
&\qquad\quad Doc\ (g\ \star)\ (x :: xs) \\
&cons\ d_1\ d_2 \ = \ \textsf{embed}\ \ldots\ ((empty \diamond d_1) \diamond d_2)
\end{aligned}
$$

Swallowing trailing whitespace

$$symbol : String \to Grammar\ String$$
$$symbol\ s = \mathsf{string}\ s \lessdot\!\circledast\ whitespace\ \star$$

$$symbol\text{-}nil : Doc\ (symbol\ s)\ s$$
$$symbol\text{-}nil = \mathsf{text} \lessdot\!\diamond\ nil$$

$$symbol\text{-}line : Doc\ (symbol\ s)\ s$$
$$symbol\text{-}line = \mathsf{embed}\ \ldots\ (\mathsf{text} \lessdot\!\diamond\ \mathsf{line})$$

Another example

$$bit\text{-}list \;:\; Grammar\;(List\;Bool)$$
$$bit\text{-}list \;=\; (bit \lhd\!\!\circledast\; symbol\;\texttt{";"})\;\star$$

Another example

$$
\begin{aligned}
\mathit{bit\text{-}list} &: \mathit{Grammar}\ (\mathit{List}\ \mathit{Bool}) \\
\mathit{bit\text{-}list} &= (\mathit{bit} \mathbin{<\!\circledast} \mathit{symbol}\ \texttt{";"})\ \star
\end{aligned}
$$

```
"1; 0; 0;"
"1;     0;0;\n   "
"1;\n\n\n     \n  0;   0;"
```

Another example

$$
\begin{aligned}
&\mathit{bit\text{-}list} : \mathit{Grammar}\ (\mathit{List}\ \mathit{Bool}) \\
&\mathit{bit\text{-}list} = (\mathit{bit} \lessdot\!\!\circledast\ \mathit{symbol}\ \texttt{";"})\ \star
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{bit\text{-}list}_{\mathsf{P}} : (\mathit{bs} : \mathit{List}\ \mathit{Bool}) \to \mathit{Doc}\ \mathit{bit\text{-}list}\ \mathit{bs} \\
&\mathit{bit\text{-}list}_{\mathsf{P}}\ [\,] = \mathit{nil} \\
&\mathit{bit\text{-}list}_{\mathsf{P}}\ (b :: [\,]) = \mathit{cons}\ (\mathit{bit}_{\mathsf{P}}\ b \lessdot\!\!\diamond\ \mathit{symbol\text{-}nil})\ \mathit{nil} \\
&\mathit{bit\text{-}list}_{\mathsf{P}}\ (b :: \mathit{bs}) = \\
&\quad \mathit{cons}\ (\mathit{bit}_{\mathsf{P}}\ b \lessdot\!\!\diamond\ \mathsf{group}\ (\mathsf{nest}\ 1\ \mathit{symbol\text{-}line})) \\
&\qquad\quad (\mathit{bit\text{-}list}_{\mathsf{P}}\ \mathit{bs})
\end{aligned}
$$

Another example

$$
\begin{array}{lcl}
\mathit{bit\text{-}list}_{\mathsf{P}} & : & (\mathit{bs} : \mathit{List\ Bool}) \to \mathit{Doc\ bit\text{-}list\ bs} \\
\mathit{bit\text{-}list}_{\mathsf{P}}\ [\,] & = & \mathit{nil} \\
\mathit{bit\text{-}list}_{\mathsf{P}}\ (b :: [\,]) & = & \mathit{cons}\ (\mathit{bit}_{\mathsf{P}}\ b \diamond \mathit{symbol\text{-}nil})\ \mathit{nil} \\
\mathit{bit\text{-}list}_{\mathsf{P}}\ (b :: \mathit{bs}) & = & \\
\quad \mathit{cons}\ (\mathit{bit}_{\mathsf{P}}\ b \diamond \mathsf{group}\ (\mathsf{nest}\ 1\ \mathit{symbol\text{-}line})) & & \\
\qquad\qquad (\mathit{bit\text{-}list}_{\mathsf{P}}\ \mathit{bs}) & &
\end{array}
$$

$$\mathit{bs} = \mathsf{true} :: \mathsf{false} :: \mathsf{true} :: \mathsf{true} :: \mathsf{false} :: [\,]$$

$\mathit{render}\ 20\ (\mathit{bit\text{-}list}_{\mathsf{P}}\ \mathit{bs}) \equiv$ `"1; 0; 1; 1; 0;"`
$\mathit{render}\ 10\ (\mathit{bit\text{-}list}_{\mathsf{P}}\ \mathit{bs}) \equiv$ `"1; 0; 1;\n 1; 0;"`
$\mathit{render}\ 6\ \ (\mathit{bit\text{-}list}_{\mathsf{P}}\ \mathit{bs}) \equiv$ `"1; 0;\n 1; 1;\n 0;"`

Another example

$$
\begin{aligned}
&\mathit{bit\text{-}list}_{\mathsf{P}} \;:\; (\mathit{bs} \;:\; \mathit{List\ Bool}) \;\to\; \mathit{Doc}\ \mathit{bit\text{-}list}\ \mathit{bs} \\
&\mathit{bit\text{-}list}_{\mathsf{P}}\ [\,] \quad = \mathit{nil} \\
&\mathit{bit\text{-}list}_{\mathsf{P}}\ (b :: [\,]) = \mathit{cons}\ (\mathit{bit}_{\mathsf{P}}\ b \diamond \mathit{symbol\text{-}nil})\ \mathit{nil} \\
&\mathit{bit\text{-}list}_{\mathsf{P}}\ (b :: \mathit{bs}) = \\
&\quad \mathit{cons}\ (\mathit{bit}_{\mathsf{P}}\ b \diamond \mathsf{group}\ (\mathsf{nest}\ 1\ \mathit{symbol\text{-}line})) \\
&\qquad\quad (\mathit{bit\text{-}list}_{\mathsf{P}}\ \mathit{bs})
\end{aligned}
$$

$$\mathit{bs} \;=\; \mathsf{true} :: \mathsf{false} :: \mathsf{true} :: \mathsf{true} :: \mathsf{false} :: [\,]$$

$$
\begin{aligned}
\mathit{bs} &\in \mathit{bit\text{-}list} \cdot \texttt{"1; 0; 1; 1; 0;"} \\
\mathit{bs} &\in \mathit{bit\text{-}list} \cdot \texttt{"1; 0; 1;\textbackslash n 1; 0;"} \\
\mathit{bs} &\in \mathit{bit\text{-}list} \cdot \texttt{"1; 0;\textbackslash n 1; 1;\textbackslash n 0;"}
\end{aligned}
$$

More

- Can use much more general grammar formalism (recursively enumerable languages).
- More advanced examples available (operators with precedence, an XML-like language).
- One can prove that the document combinators satisfy certain algebraic properties.

Conclusions

- Light-weight approach to correct-by-construction pretty-printing.
- Based on classical pretty-printing, but precisely typed.
- Separates grammars and pretty-printers.