Modelling Parsing and Unparsing

Vadim Zaytsev

University of Amsterdam

Anya Helene Bagge

University of Bergen

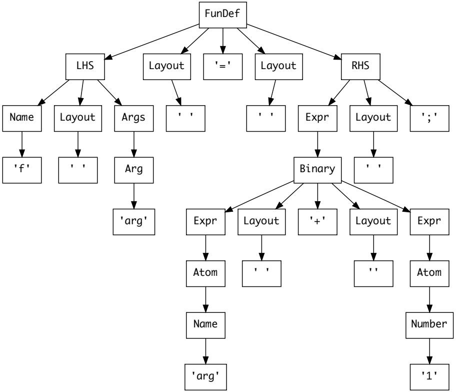
Parsing@SLE
2014-09-14

SLE Artifacts

String

```
f arg = arg +1;
```

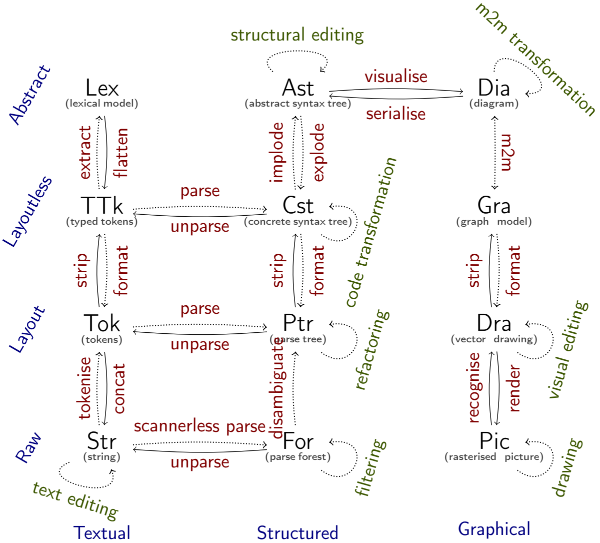
Parse Tree



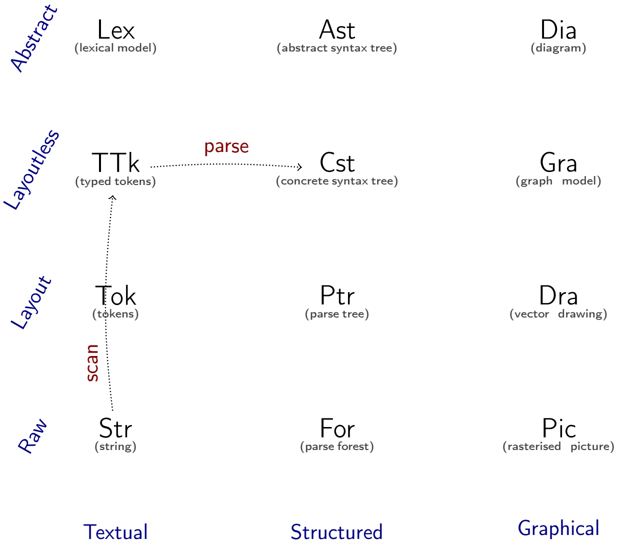
Token List

f	arg	=	arg	+	1	;

The Megamodel

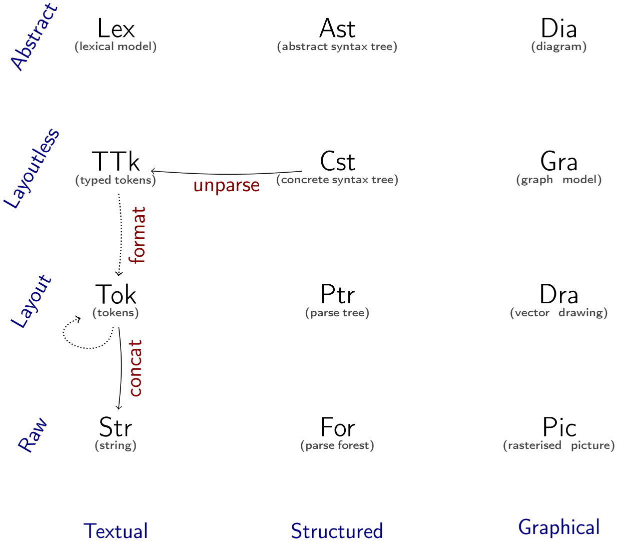


Classic (Lex+Yacc) Parsing



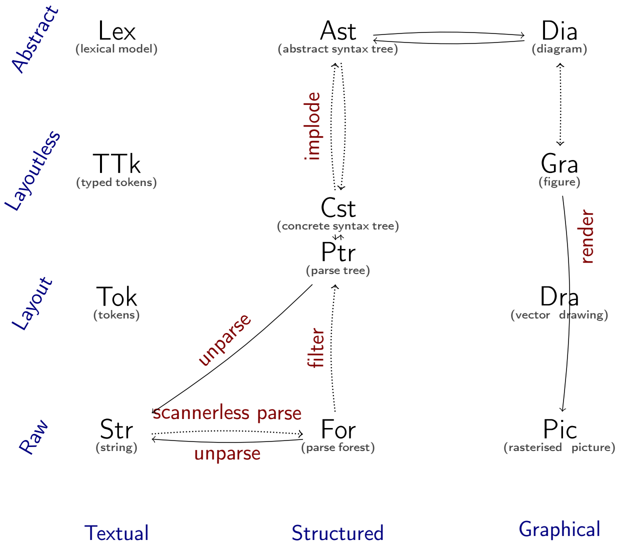
	Textual	Structured	Graphical
Abstract	Lex (lexical model)	Ast (abstract syntax tree)	Dia (diagram)
Layoutless	TTk (typed tokens)	Cst (concrete syntax tree)	Gra (graph model)
Layout	Tok (tokens)	Ptr (parse tree)	Dra (vector drawing)
Raw	Str (string)	For (parse forest)	Pic (rasterised picture)

Code Formatting



	Textual	Structured	Graphical
Abstract	Lex (lexical model)	Ast (abstract syntax tree)	Dia (diagram)
Layoutless	TTk (typed tokens)	Cst (concrete syntax tree)	Gra (graph model)
Layout	Tok (tokens)	Ptr (parse tree)	Dra (vector drawing)
Raw	Str (string)	For (parse forest)	Pic (rasterised picture)

Rascal



Abstract

Lex
(lexical model)

Ast
(abstract syntax tree)

Dia
(diagram)

implode

Layoutless

TTk
(typed tokens)

Gra
(figure)

Cst
(concrete syntax tree)

Ptr
(parse tree)

render

Layout

Tok
(tokens)

Dra
(vector drawing)

unparse

filter

Raw

Str
(string)

scannerless parse

unparse

For
(parse forest)

Pic
(rasterised picture)

Textual

Structured

Graphical

Lowered Transformations

$$\begin{tikzcd}
\mathsf{Ast} \arrow[r, bend left=5, "\mathsf{transform}"'] & \mathsf{Ast}
\end{tikzcd}$$

Lowered Transformations

$$\begin{tikzcd}
\text{Ast} \arrow[r, bend left=5, "\text{transform}"'] & \text{Ast} \arrow[d, "\text{explode}"] \\
\text{Cst} \arrow[u, "\text{implode}"] \arrow[r, dotted, bend left=5, "\text{transform}"'] & \text{Cst}
\end{tikzcd}$$

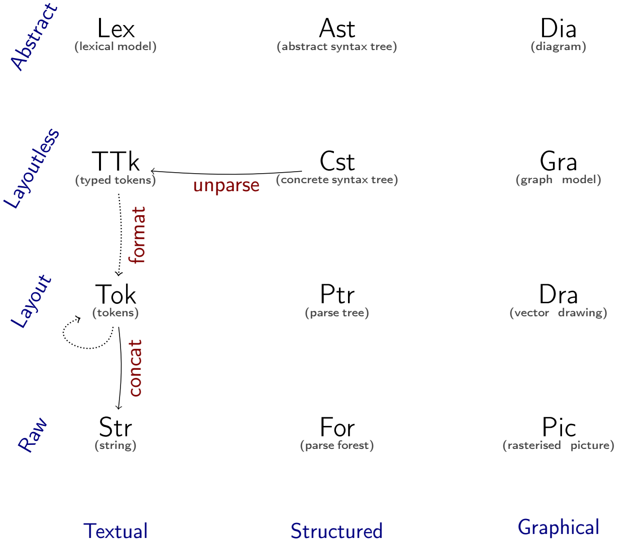
Lowered Transformations

$$\begin{tikzcd}
\text{Ast} \arrow[r, bend left=5, "\text{transform}"'] & \text{Ast} \arrow[d, bend left=5, "\text{explode}"] \\
\text{Cst} \arrow[u, bend left=5, "\text{implode}"] \arrow[r, dotted, bend left=5, "\text{transform}"'] & \text{Cst} \arrow[d, bend left=5, "\text{format}"] \\
\text{Ptr} \arrow[u, bend left=5, "\text{strip}"] \arrow[r, dotted, bend left=5, "\text{transform}"'] & \text{Ptr}
\end{tikzcd}$$

Lowered Transformations

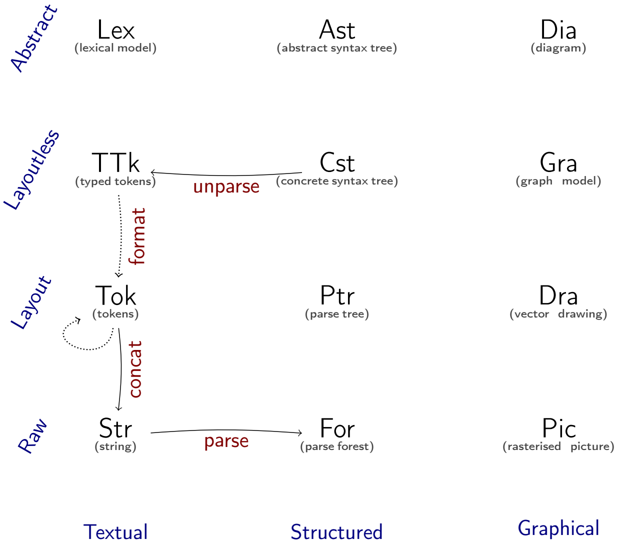
$$\begin{tikzcd}
 & \mathsf{Ast} \arrow[r, bend left, "\text{transform}"'] & \mathsf{Ast} \arrow[d, "\text{explode}"] & \\
 & \mathsf{Cst} \arrow[u, "\text{implode}"] \arrow[r, dotted, bend left, "\text{transform}"'] & \mathsf{Cst} \arrow[d, "\text{format}"] & \\
 & \mathsf{Ptr} \arrow[u, "\text{strip}"] \arrow[r, dotted, bend left, "\text{transform}"'] & \mathsf{Ptr} \arrow[dr, "\text{unparse}"] & \\
\mathsf{Str} \arrow[ur, "\text{parse}"] \arrow[rrr, dotted, bend left, "\text{transform}"'] & & & \mathsf{Str}
\end{tikzcd}$$

Composing Transformations



	Textual	Structured	Graphical
Abstract	Lex (lexical model)	Ast (abstract syntax tree)	Dia (diagram)
Layoutless	TTk (typed tokens)	Cst (concrete syntax tree)	Gra (graph model)
Layout	Tok (tokens)	Ptr (parse tree)	Dra (vector drawing)
Raw	Str (string)	For (parse forest)	Pic (rasterised picture)

Composing Transformations



Abstract

Lex (lexical model)

Ast (abstract syntax tree)

Dia (diagram)

Layoutless

TTk (typed tokens)

unparse

Cst (concrete syntax tree)

Gra (graph model)

format

Layout

Tok (tokens)

Ptr (parse tree)

Dra (vector drawing)

concat

Raw

Str (string)

parse

For (parse forest)

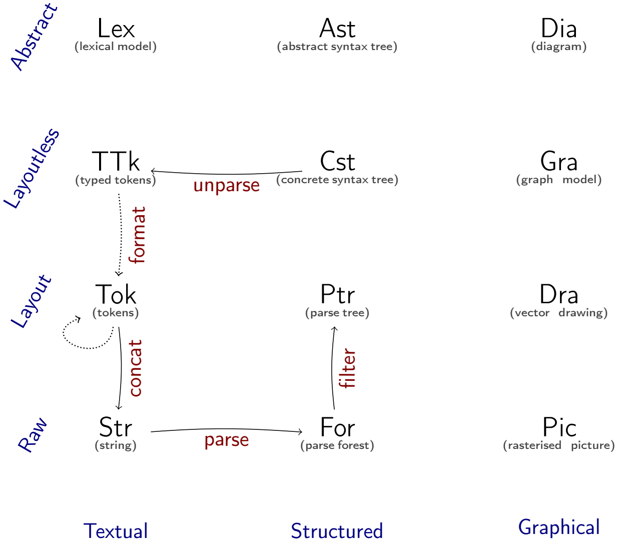
Pic (rasterised picture)

Textual

Structured

Graphical

Composing Transformations



Abstract

Lex (lexical model)

Ast (abstract syntax tree)

Dia (diagram)

Layoutless

TTk (typed tokens)

unparse

Cst (concrete syntax tree)

Gra (graph model)

format

Layout

Tok (tokens)

Ptr (parse tree)

Dra (vector drawing)

concat

filter

Raw

Str (string)

parse

For (parse forest)

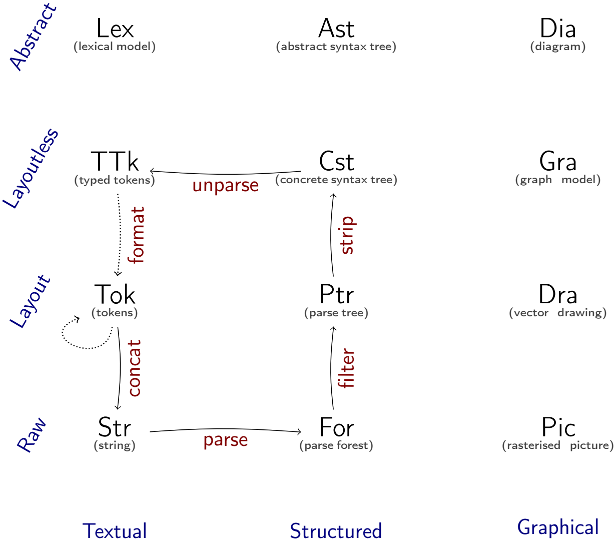
Pic (rasterised picture)

Textual

Structured

Graphical

Composing Transformations



Abstract

Lex
(lexical model)

Ast
(abstract syntax tree)

Dia
(diagram)

Layoutless

TTk
(typed tokens)

unparse

Cst
(concrete syntax tree)

Gra
(graph model)

format

strip

Layout

Tok
(tokens)

Ptr
(parse tree)

Dra
(vector drawing)

concat

filter

Raw

Str
(string)

parse

For
(parse forest)

Pic
(rasterised picture)

Textual

Structured

Graphical

Is it Useful?

- Explain representations and transformations to students
- Explain what you mean with “parsing” in your paper
- Simplify dealing with bidirectional transformations
- Amaze your friends with fancy graphic models in your papers

Conclusion

- We have identified SLE artifacts and translations between them

Prototype w/transformations:

https://github.com/grammarware/bx-parsing

The full paper:

Vadim Zaytsev and Anya Helene Bagge. Parsing in a Broad Sense. In Jurgen Dingel and Wolfram Schulte, editors, Proceedings of the 17th International Conference on Model Driven Engineering Languages and Systems (MODELS'14), volume 8767 of LNCS, pages 50–67. Springer, 2014.