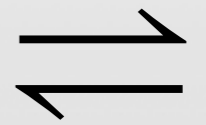


15TH INTL. SYMPOSIUM ON TRENDS IN FUNCTIONAL PROGRAMMING

CASE STUDIES IN

BIDIRECTIONALISATION



VADIM ZAYTSEV A.K.A. @GRAMMARWARE

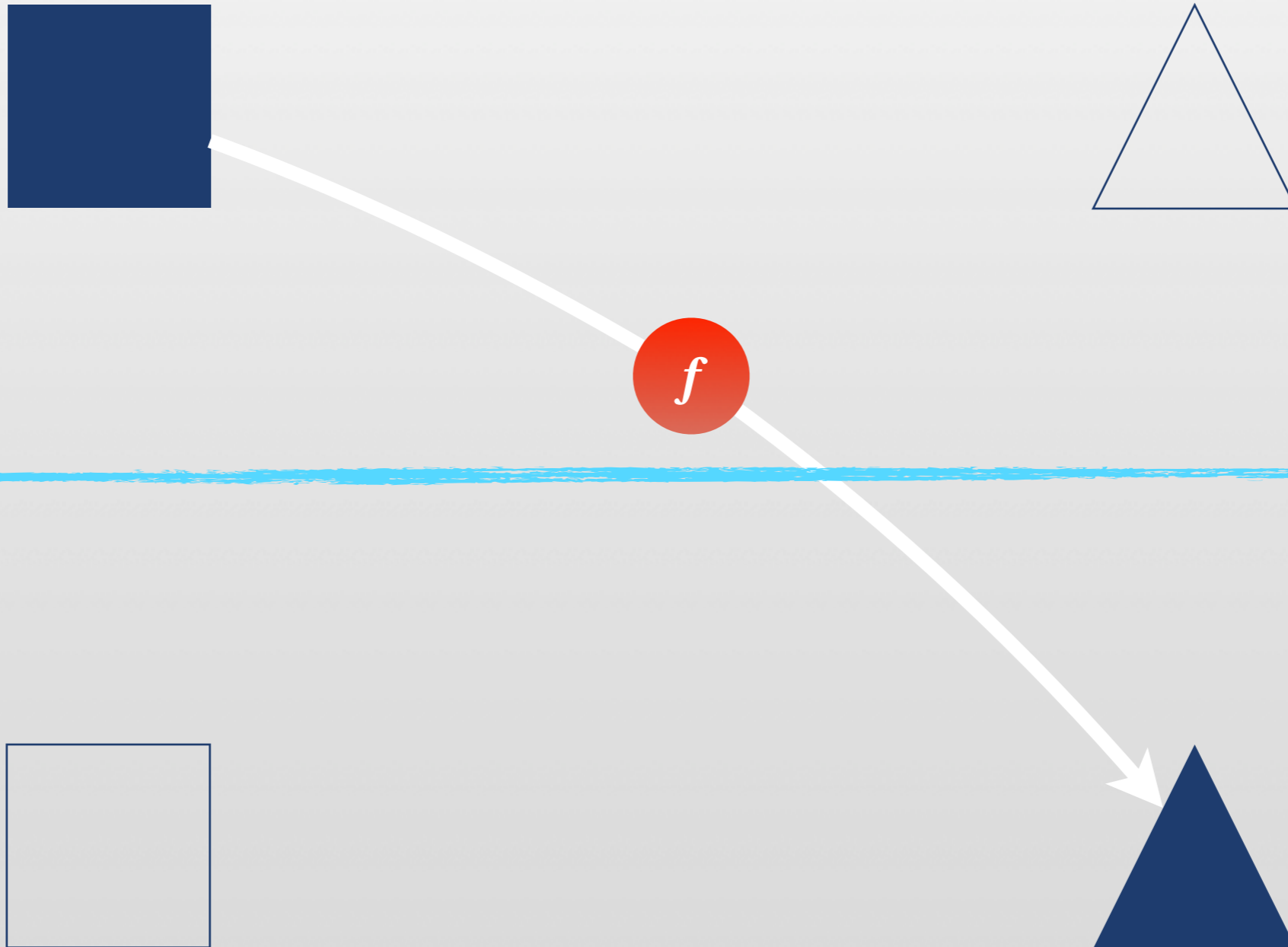
PART 1 OF 3

BIDIRECTIONAL TRANSFORMATIONS

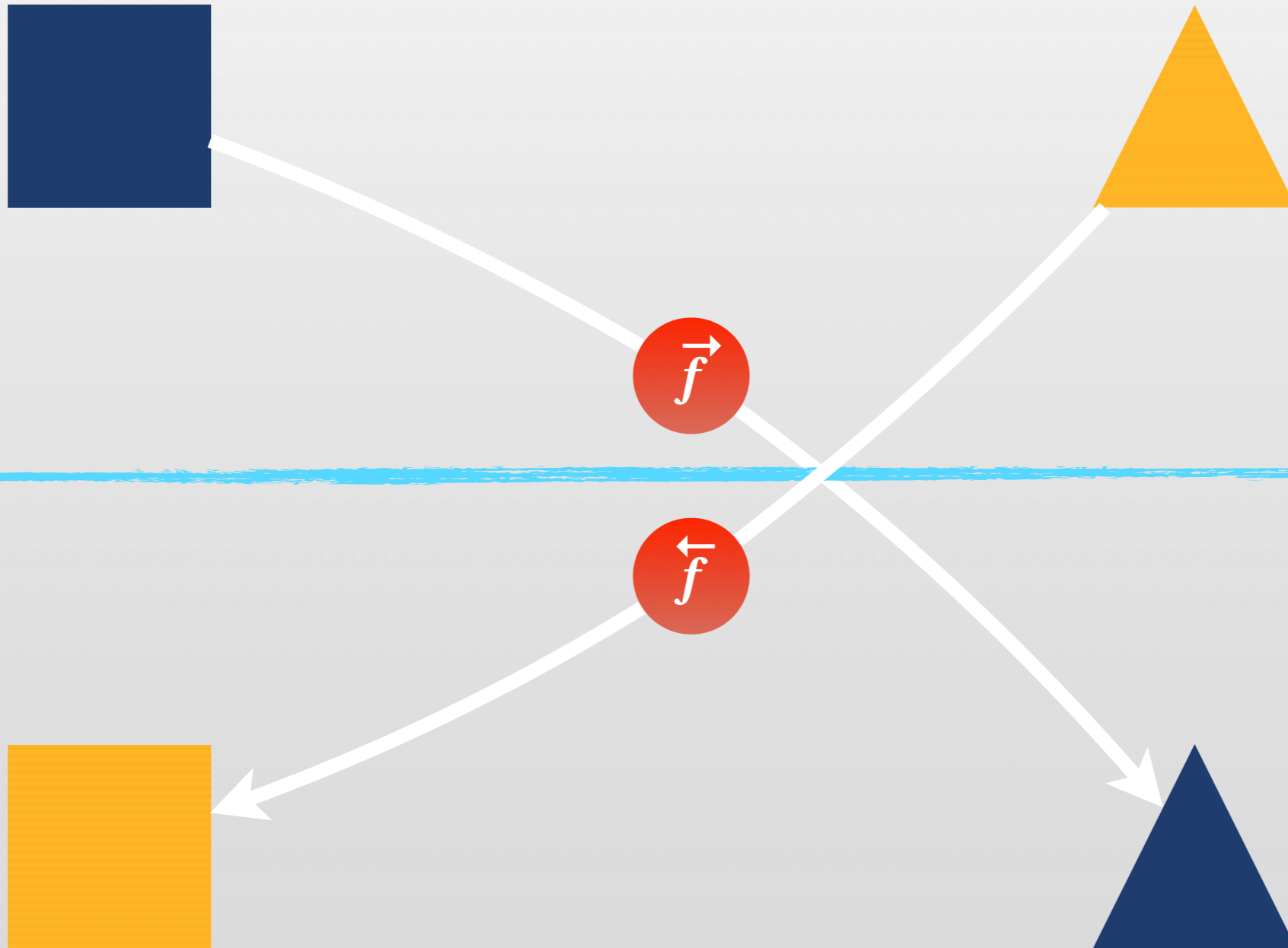
FROM FP TO BX: UNIDIRECTIONAL



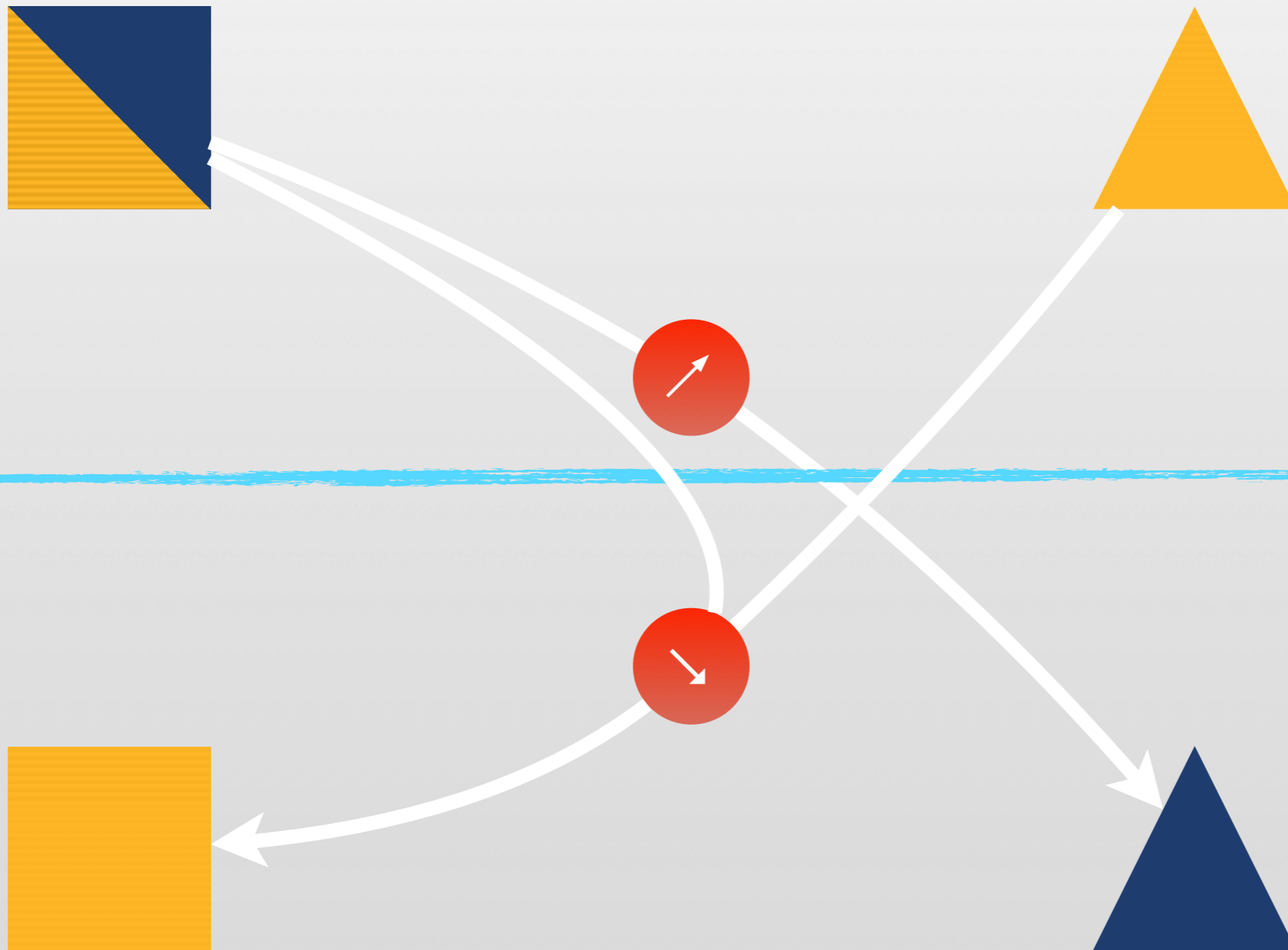
FROM FP TO BX: UNIDIRECTIONAL



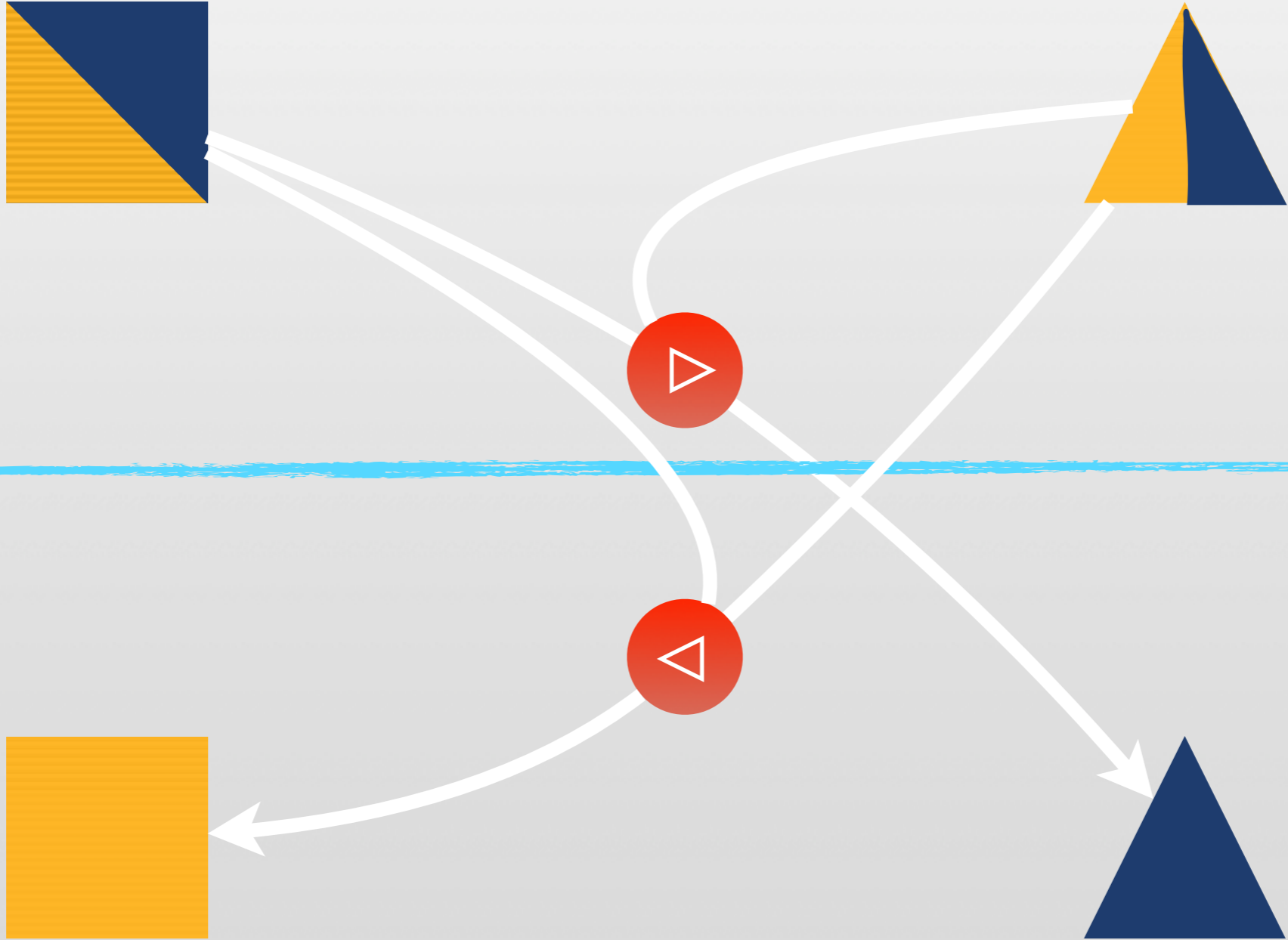
FROM FP TO BX: UNIDIR.PAIR



FROM FP TO BX: LENS



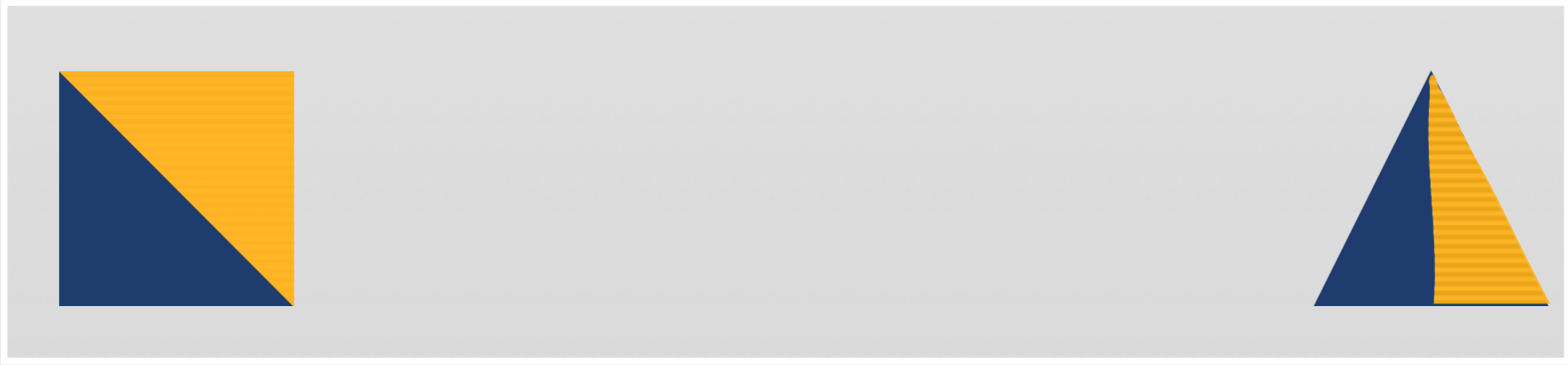
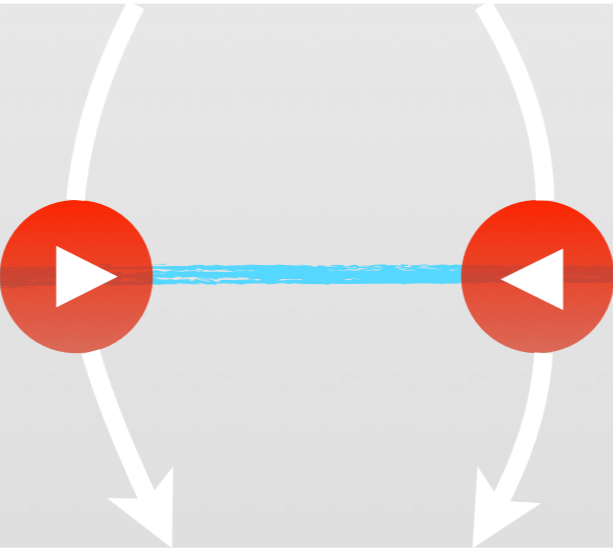
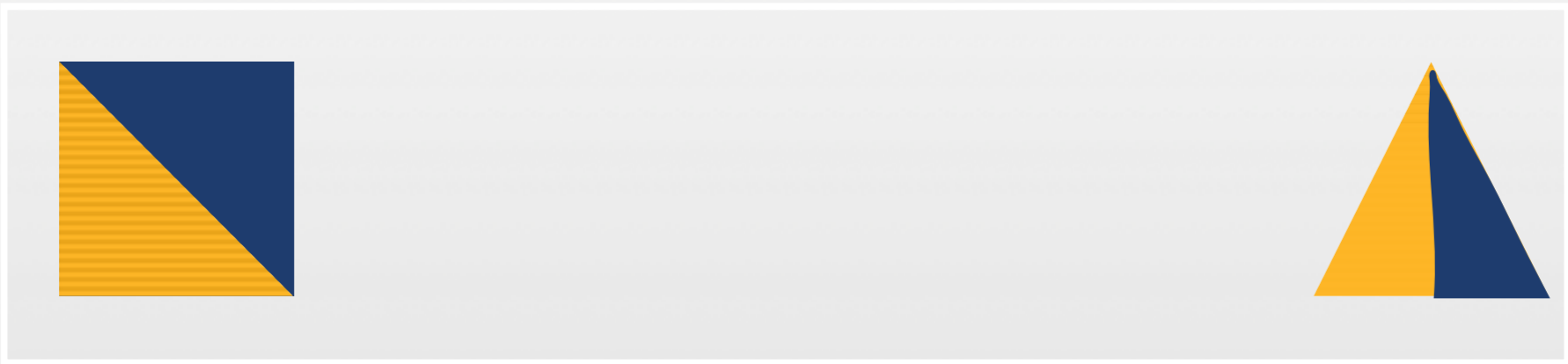
FROM FP TO BX: SYNCHRONISER



FROM FP TO BX: SUSTAINER



FROM FP TO BX: SUSTAINER



BIDIRECTIONALISATION

MAKE OR INFER BX FROM FP

PART 2 OF 3

GRAMMAR TRANSFORMATIONS

GRAMMAR TRANSFORMATIONS

expr : ...;
atom : ID | INT | '(' expr '');

 **abstractize**

expr : ...;
atom : ID | INT | expr;

vertical 

expr : ...;
atom : ID;
atom : INT;
atom : expr;

unite 

expr : ...;
expr : ID;
expr : INT;
expr : expr;

expr : ...;
expr : ID;
expr : INT;

 **abridge**

IN GRAMMARLAB:

```
importG
```

```
  expr ::= atom "+" expr;
```

```
  atom ::= ID | INT | "(" expr ")";
```

```
  .  
  abstractize atom ::= ID | INT | <>:"(" expr <>:")"; .  
  vertical in atom.  
  unite atom with expr.  
  abridge expr ::= expr; .
```



GRAMMAR TRANSFORMATIONS

expr : ...;
atom : ID | INT | '(' expr '');

abstractize
concretize

expr : ...;
atom : ID | INT | expr;

vertical
horizontal

expr : ...;
atom : ID;
atom : INT;
atom : expr;

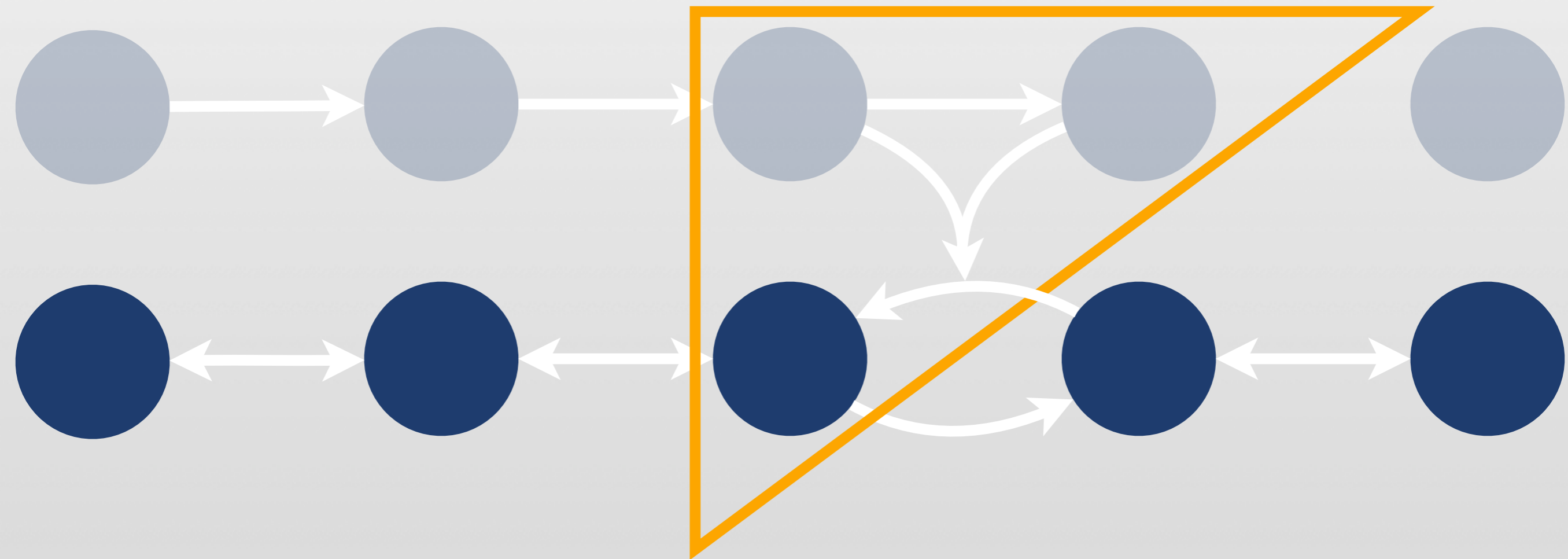
unite

expr : ...;
expr : ID;
expr : INT;

abridge
detour

expr : ...;
expr : ID;
expr : INT;
expr : expr;

MINI-LENS IN ACTION



SAME PROBLEMS WITH...

- eliminate
 - introduce what?
- inline
 - extract what?
- unlabel
 - designate what?
- etc

PART 3 OF 3

PARSING & UNPARSING

(UN)PARSING

- Parsing: recognising structure
 - text → tree
 - parse tree → AST
 - disambiguation of trees/forests
 - tokeniser vs. scannerless
- Unparsing: representing structure
 - model → picture
 - tree → text
 - (re)formatting
 - serialisation

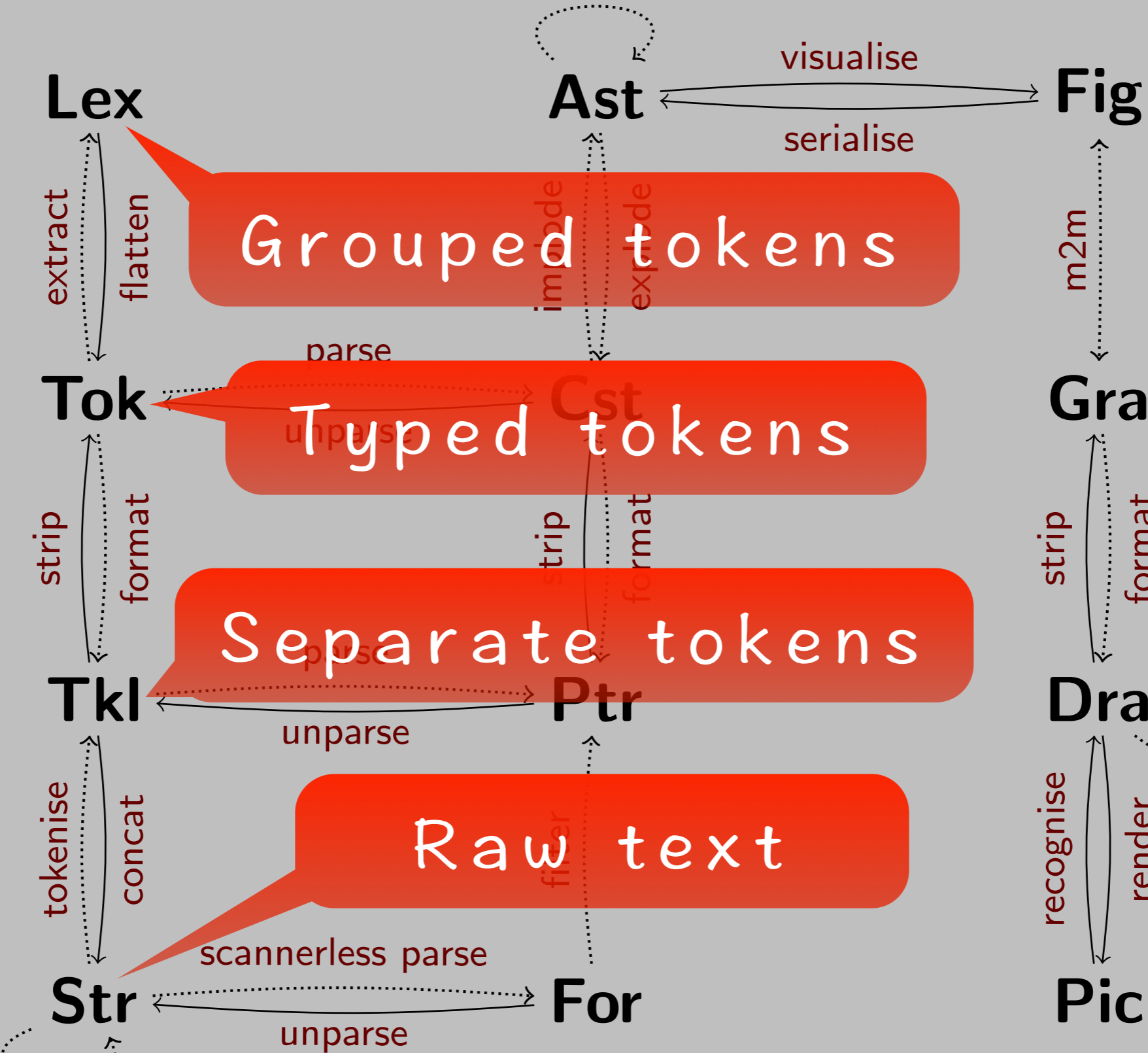
program restructuring

ABSTRACT

LAYOUTLESS

LAYOUT

RAW



TEXTUAL

STRUCTURED

GRAPHICAL

program restructuring

Abstract syntax tree

Parse tree without layout

Parse tree with layout

Parse forest

TEXTUAL

STRUCTURED

GRAPHICAL



ABSTRACT

LAYOUTLESS

LAYOUT

RAW

text editing

Lex

extract
flatten

strip
format

tokenise
concat

Ast

implode
explode

Cst

strip
format

Ptr

filter

For

visualise
serialise

Fig

m2m

Gra

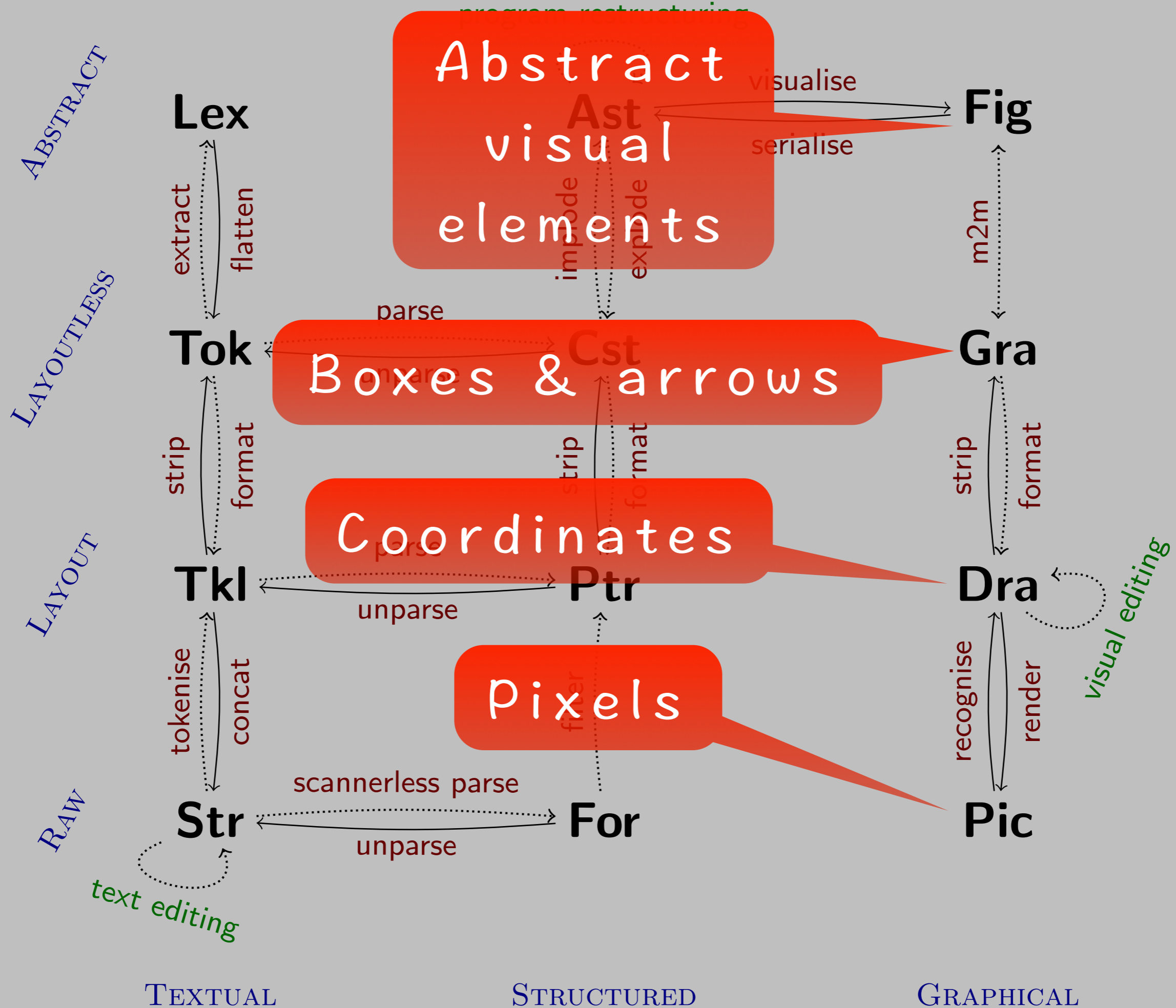
strip
format

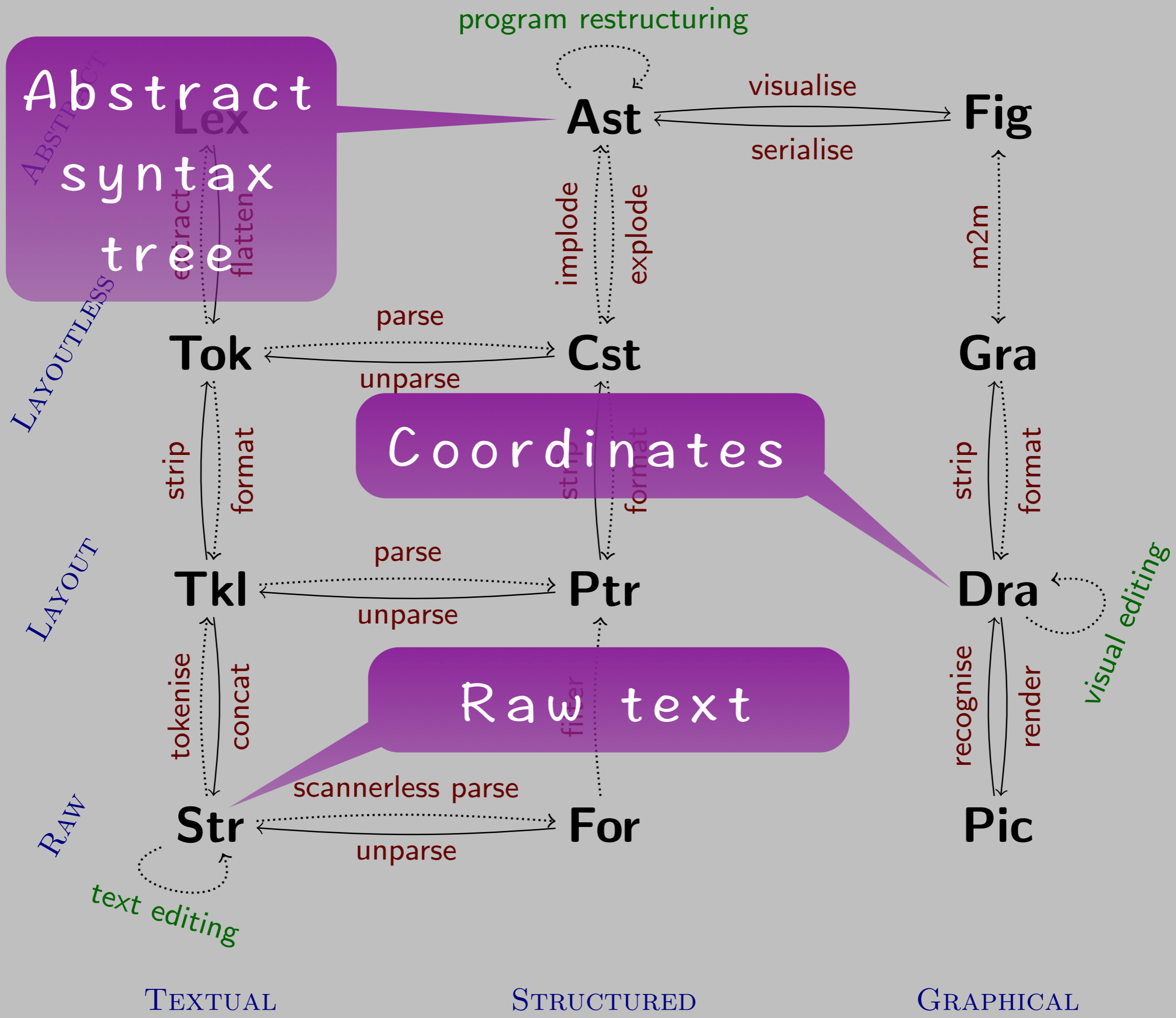
Dra

recognise
render

Pic

visual editing





program restructuring

ABSTRACT

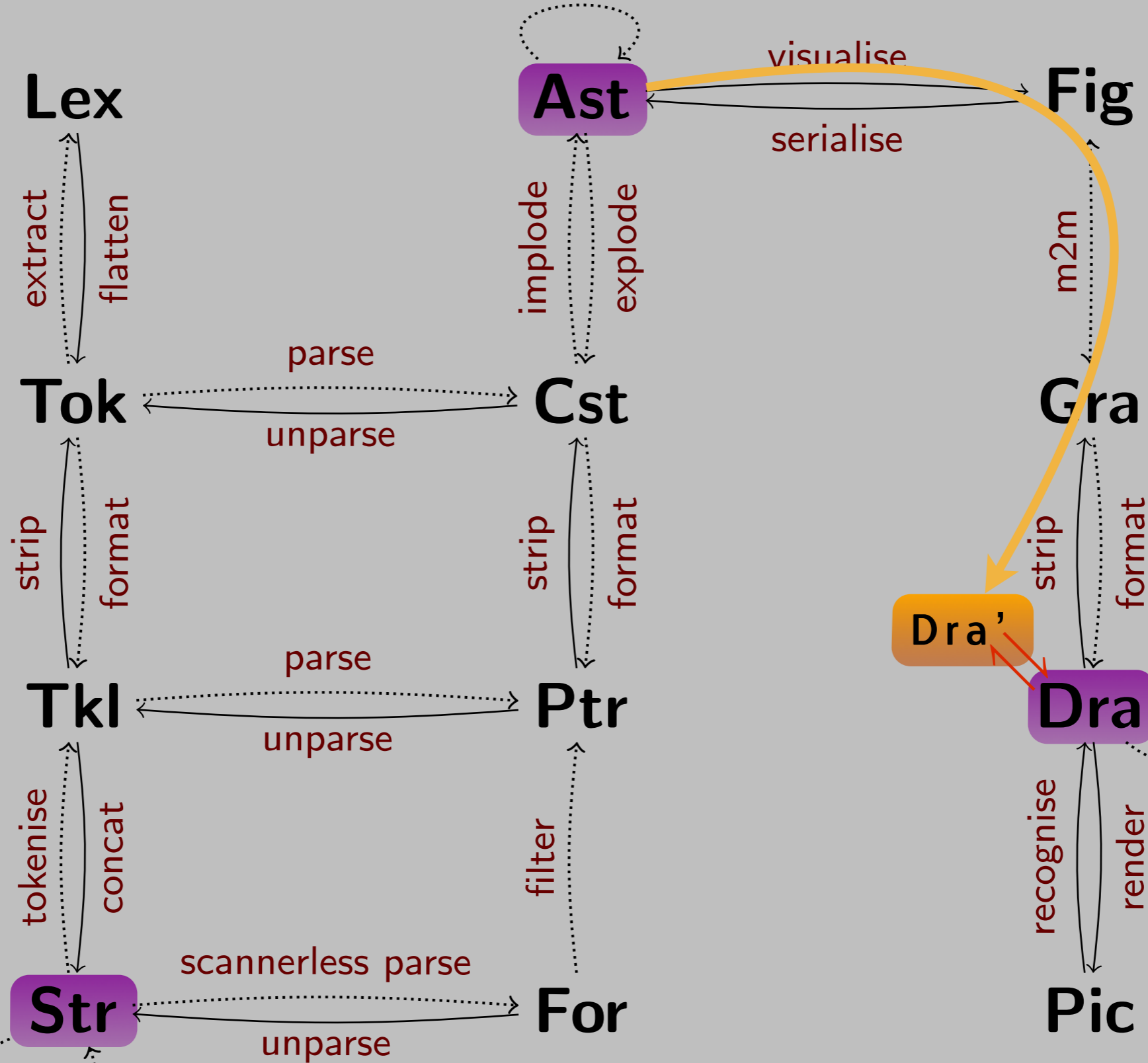
LAYOUTLESS

LAYOUT

RAW

text editing

visual editing



Ast ▷ Dra

program restructuring

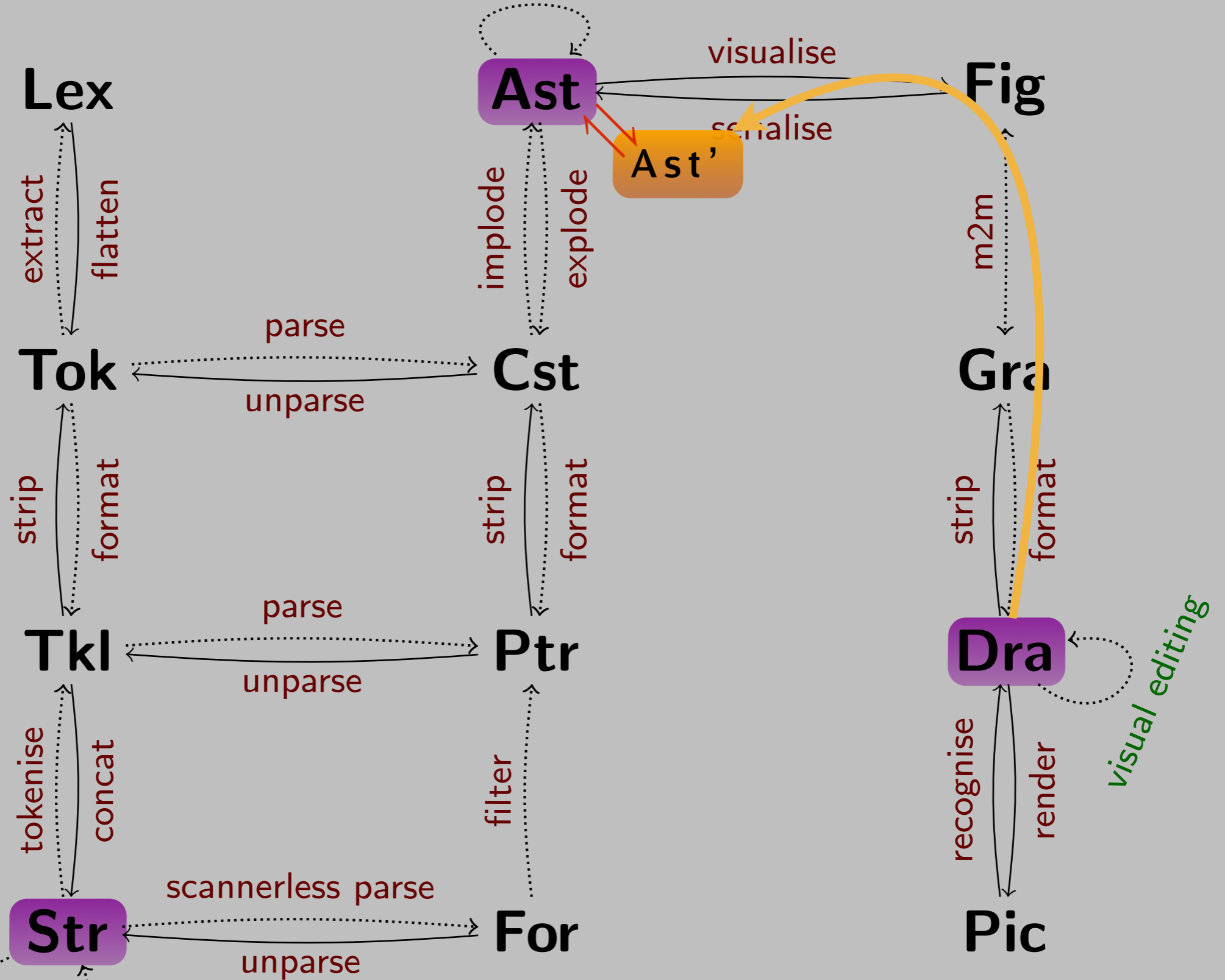
ABSTRACT

LAYOUTLESS

LAYOUT

RAW

text editing



Dra ▷ Ast

program restructuring

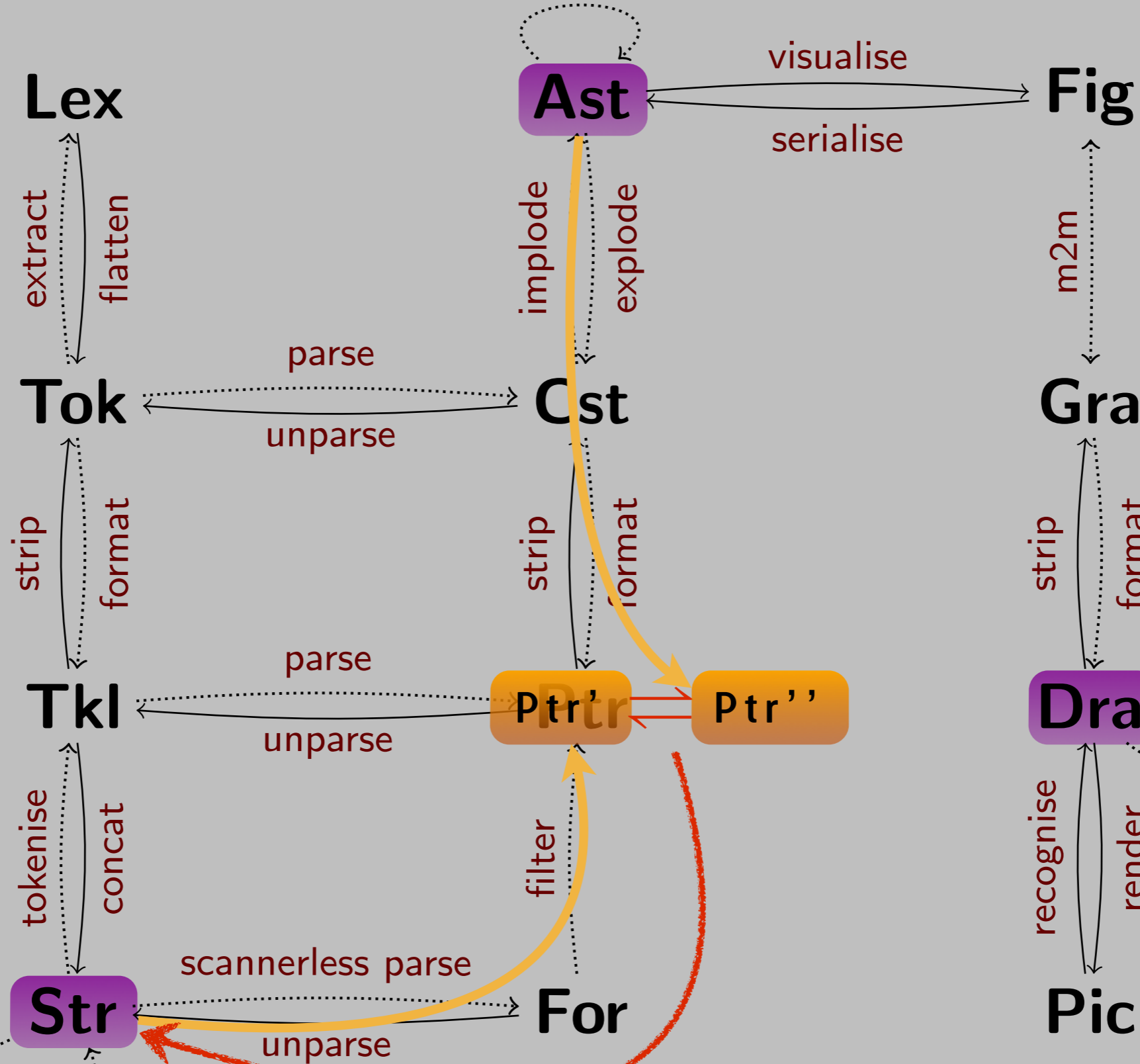
ABSTRACT

LAYOUTLESS

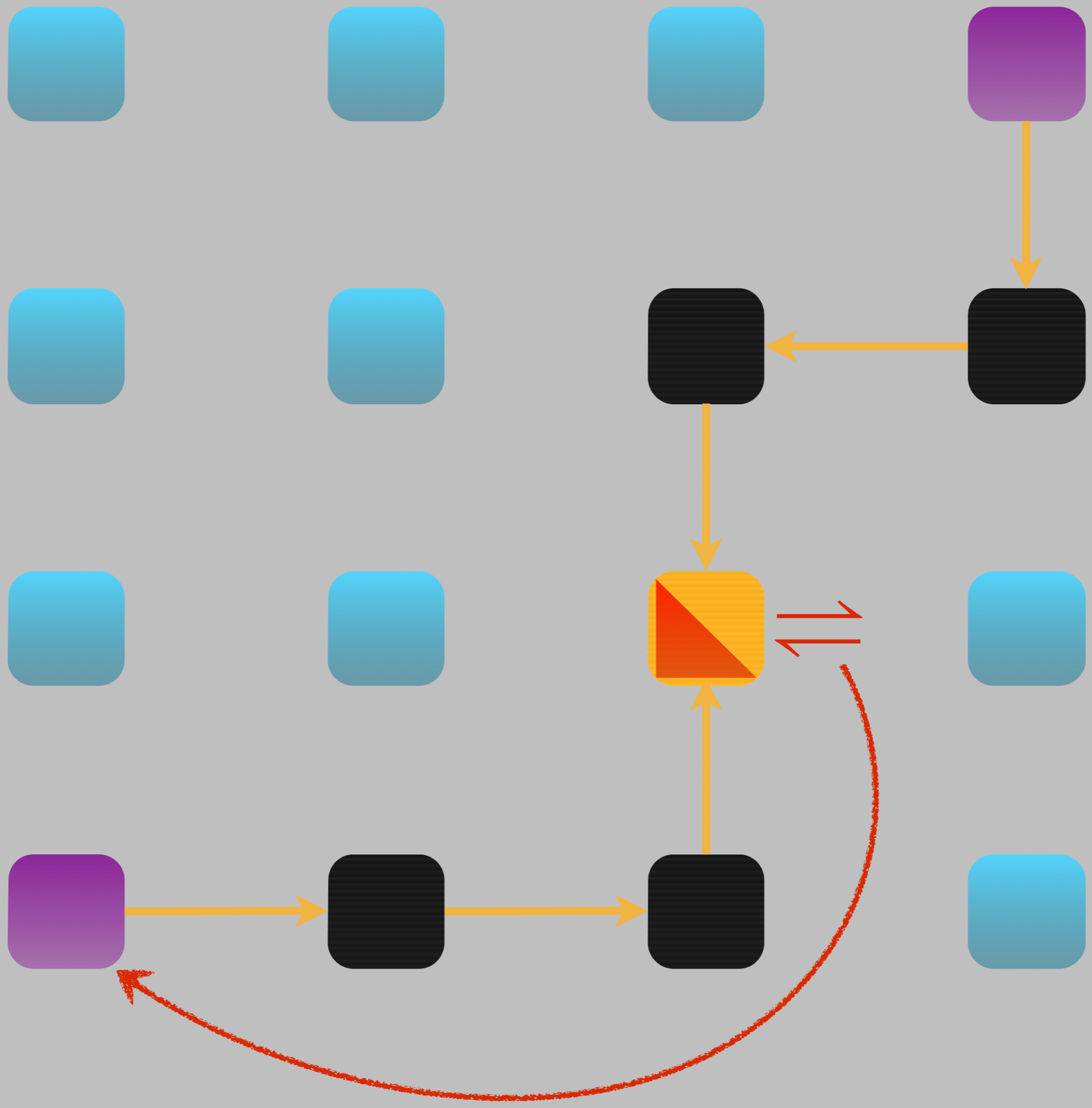
LAYOUT

RAW

text editing



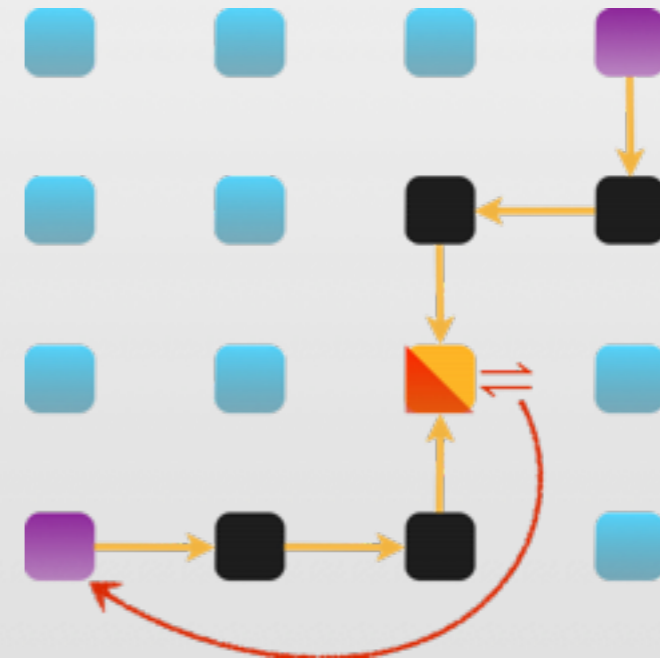
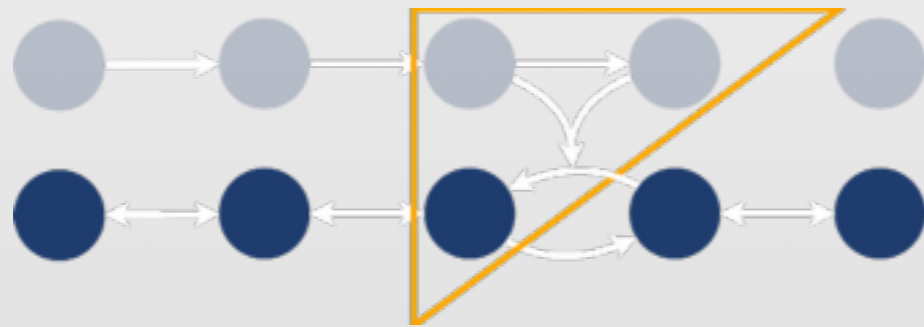
Ast > Str



THANKS FOR YOUR ATTENTION!



- Bidirectionalisation: pushing FP to BX



- This was Dr. Vadim Zaytsev a.k.a. **grammarware**
 - grammarware.net, twitter.com/grammarware, [grammarware.github.com](https://github.com/grammarware), ...
- Slides are CC-BY-SA: <http://grammarware.net/talks/#TFP2014>