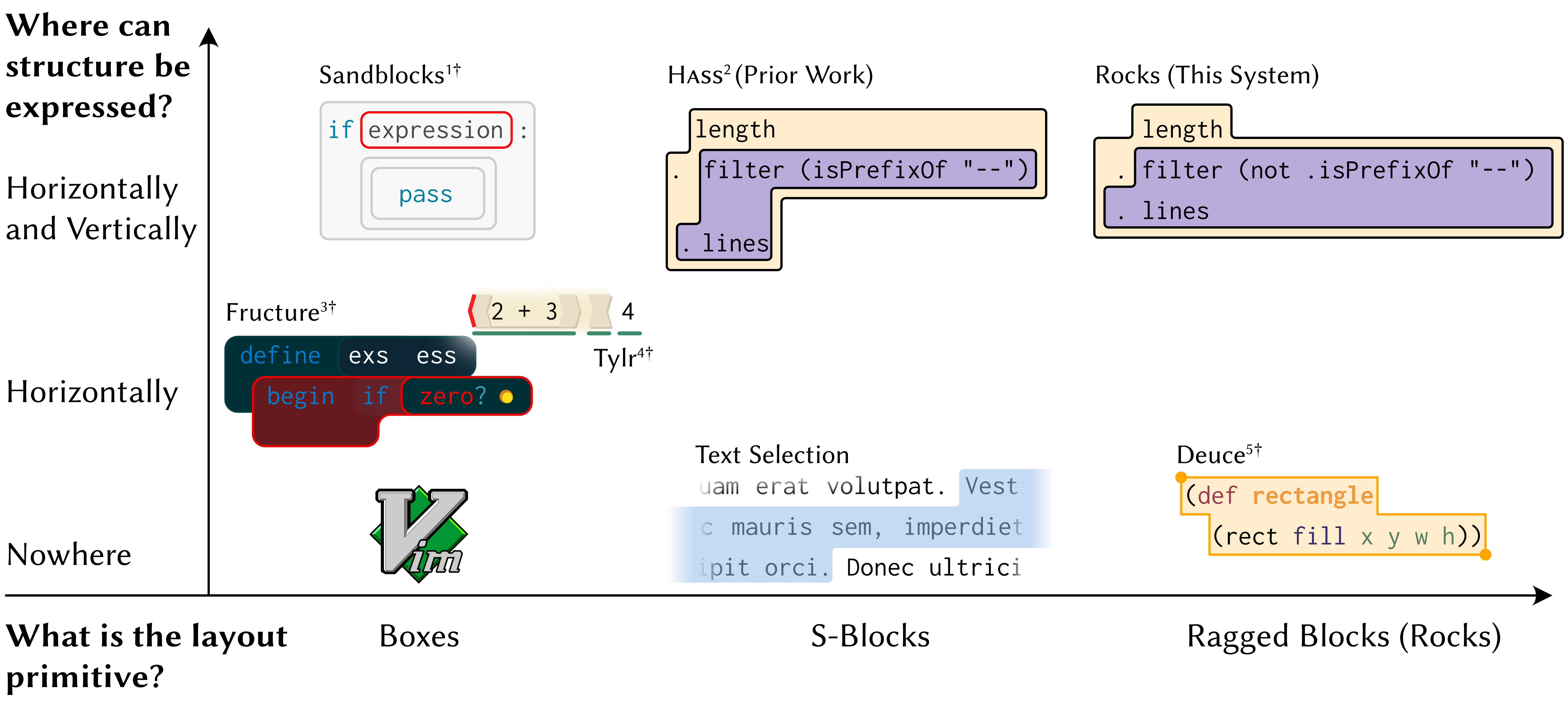
Ragged Blocks: Rendering Structured Text With Style

Sam Cohen and Ravi Chugh

Design Space

What other systems visualize structured text?



Goals

What makes for a good structured text visualization?

Text layout looks "natural"

With styles removed, the algorithm should produce layouts that are similar to those produced by an ordinary text editor.

Structure is visually clear

The layout of the text should obviously convey its underlying tree structure.

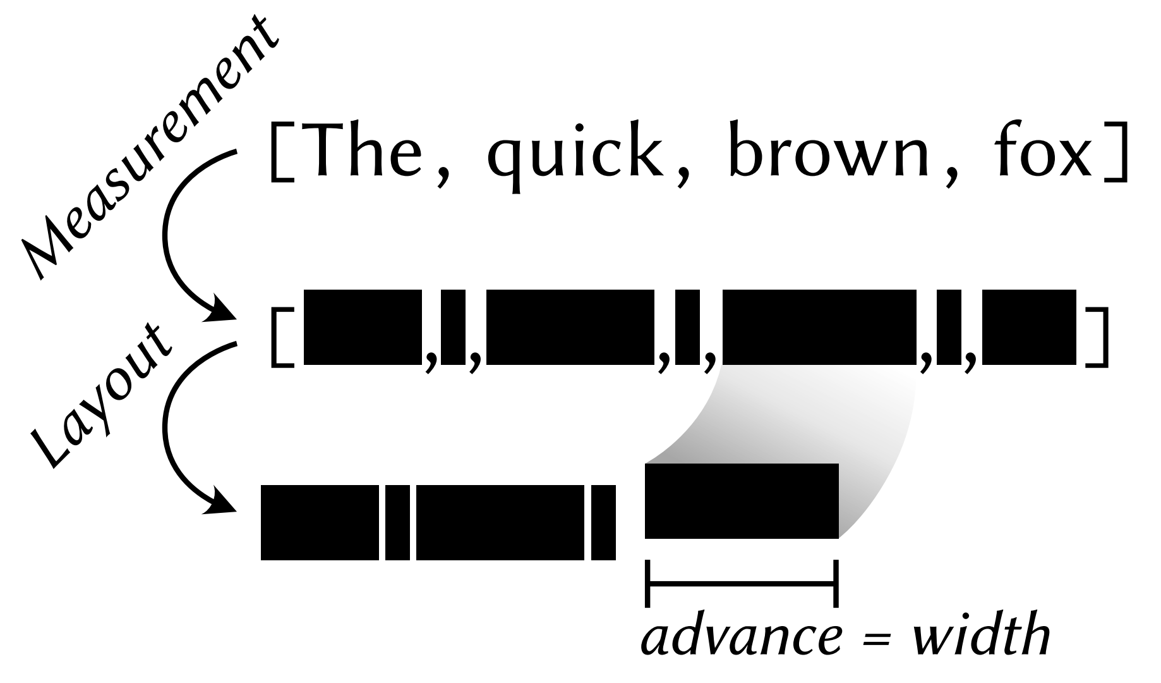
Outlines are simple

The visual elements produced by the algorithm ought to be visually simple in order to minimize noise.

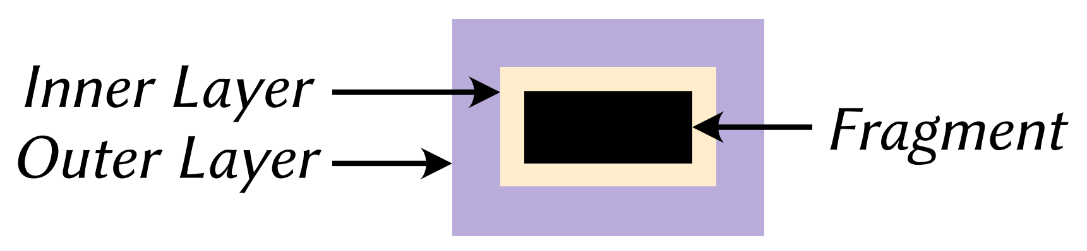
Regions

A data structure for structured text layout

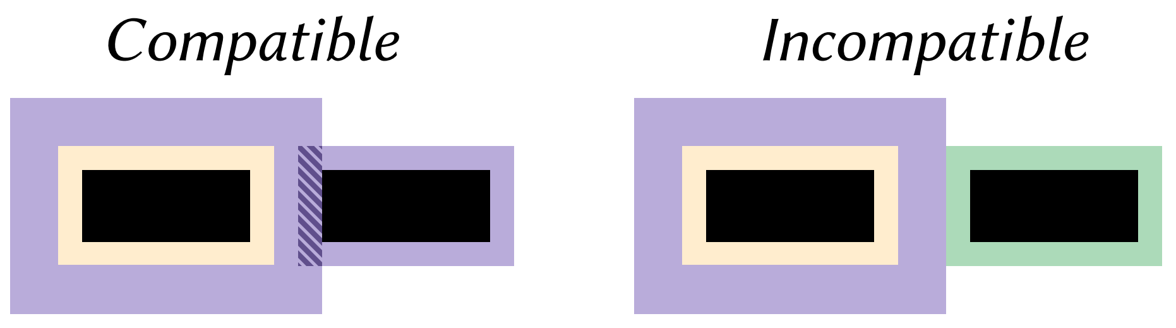
All text layout algorithms (structural or otherwise) start by breaking the input into a list of *fragments*, then *measuring* each fragment.



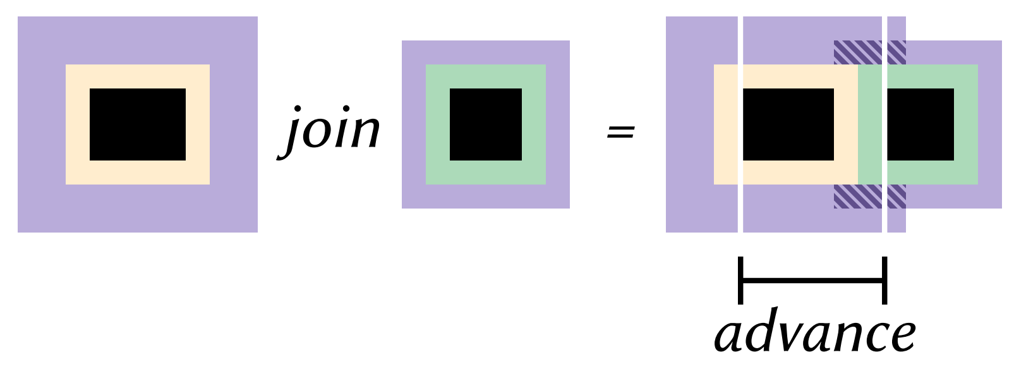
We generalize the notion of *advance* to handle fragments which might have *many layers of padding*.



We introduce a notion of compatibility between layers. Layers can be overlapped if they derive from the same document subtree.



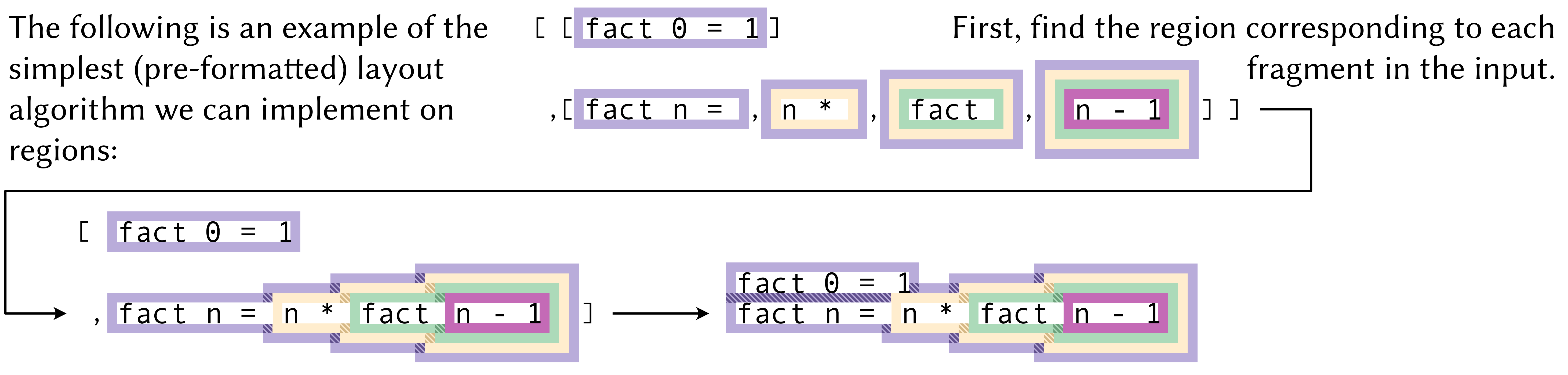
Finally, we define an operator, *join*, which merges compatible layers. Advance, in this new setting, works on *pairs* of fragments.



Layout

Creating text layouts from Regions

The following is an example of the simplest (pre-formatted) layout algorithm we can implement on regions:


[[fact 0 = 1]
, [fact n = , n * , fact , n - 1]]
[fact 0 = 1
, fact n = n * fact n - 1]
fact 0 = 1
fact n = n * fact n - 1

First, find the region corresponding to each fragment in the input.

Then, *join* the fragments on each line.

Finally, *join* the lines together.

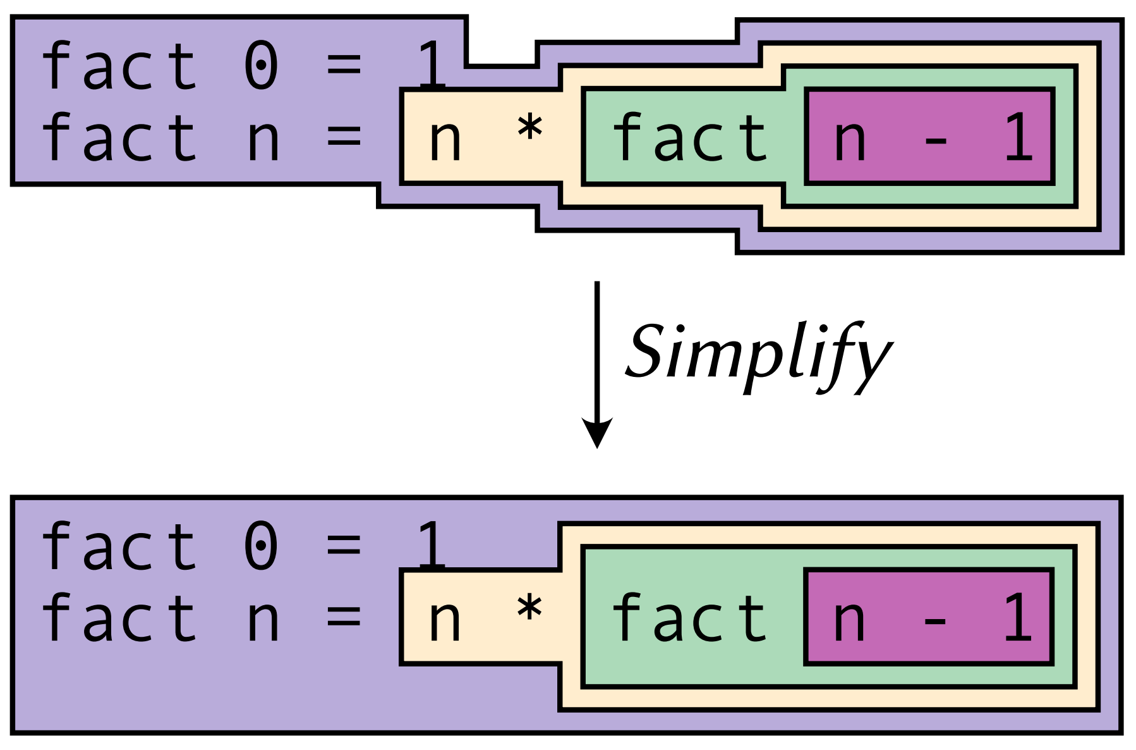
Simplification

Minimizing the number of corners in each Rock

The text layout is minimally disturbed by the padding, and the structure is clear, but we would also like simple outlines. The simplification process reduces the number of corners in the generated rocks without moving the underlying fragments.

The simplification algorithm used to produce the picture on the right finds the maximum (rectangular) union of regions s.t. the union doesn't intersect its parent.

There are further involved methods which could yield even better outlines.



[1] Tom Beckmann, Patrick Rein, Stefan Ramson, Joana Bergsiek, and Robert Hirschfeld. 2023. Structured Editing for All: Deriving Usable Structured Editors from Grammars.

[2] Cohen and Chugh

[3] Andrew Blinn, 2019. Fructure: A Structured Editing Engine in Racket

[4] D. Moon, A. Blinn and C. Omar. Gradual Structure Editing with Obligations.

[5] Brian Hempel, Justin Lubin, Grace Lu, and Ravi Chugh. 2018. Deuce: a Lightweight User Interface for Structured Editing.

[†] These images were modified from their original publication to use a consistent font and line width.

THE UNIVERSITY OF CHICAGO