Automated Bijections with Combinatorial Exploration

Jon Steinn Eliasson

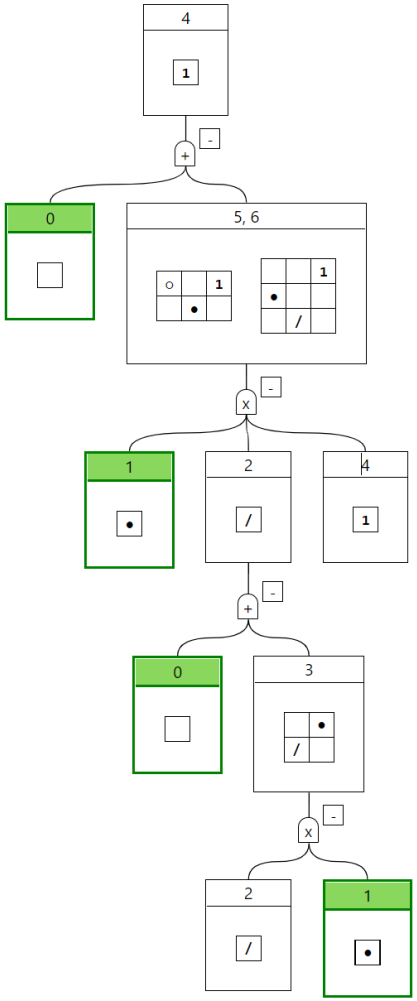
(Based on joint work with Christian Bean, Émile Nadeau, Jay Pantone and Henning Ulfarsson)

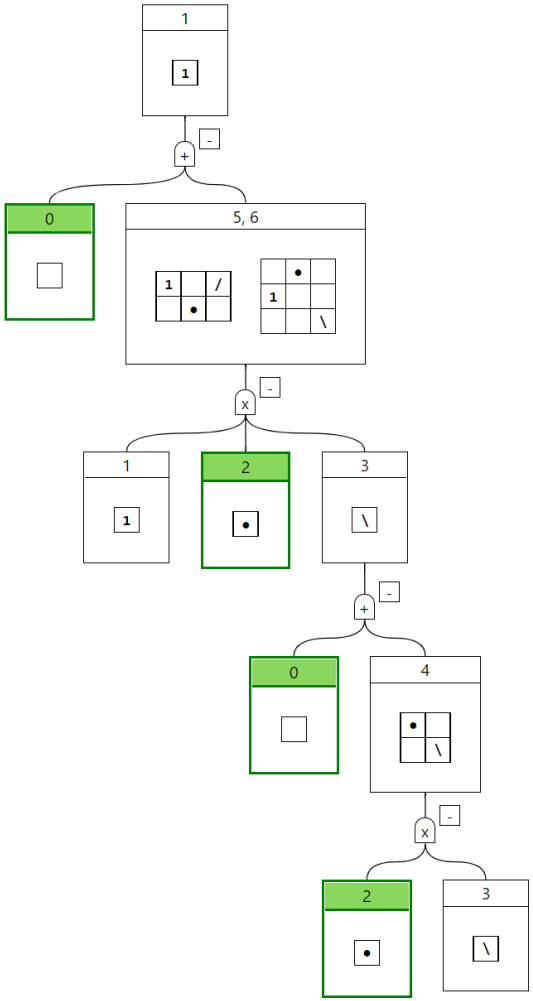
Reykjavik University

15/06/2021

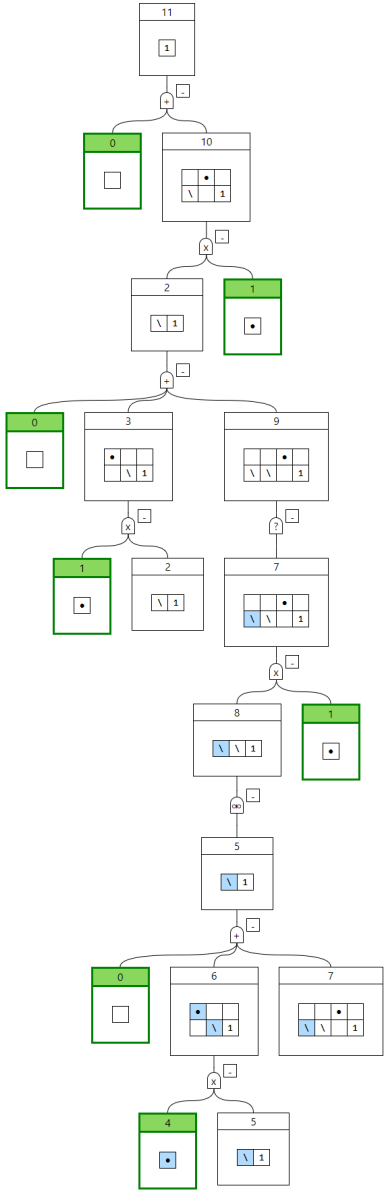
Parallel Specifications

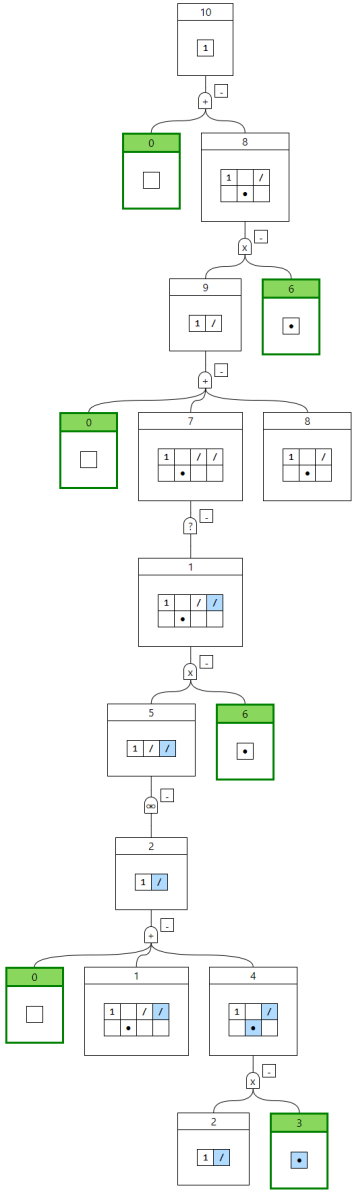
$\mathrm{Av}(231, 321)$ and $\mathrm{Av}(132, 312)$





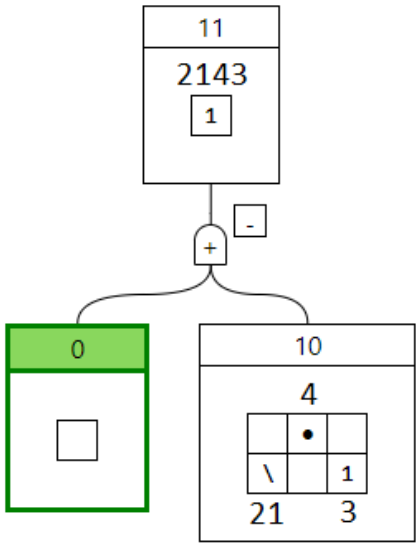
An example for $Av(123)$ and $Av(132)$

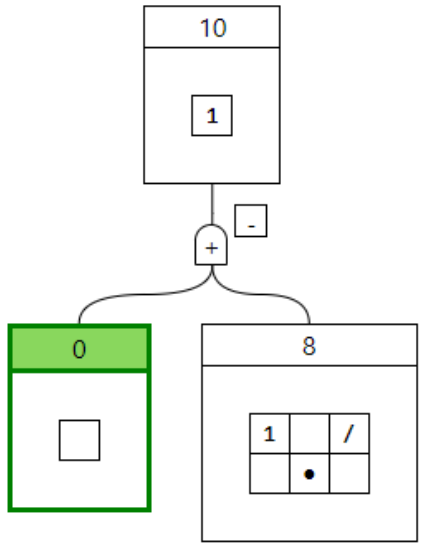




An example for $\mathbf{Av}(123)$ and $\mathbf{Av}(132)$

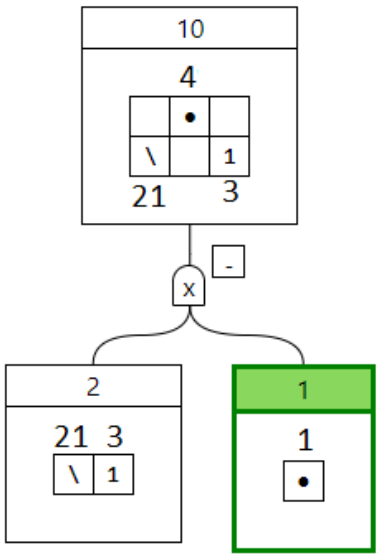
Start with $\pi = 2143 \in \mathrm{Av}(123)$ and place the topmost point.

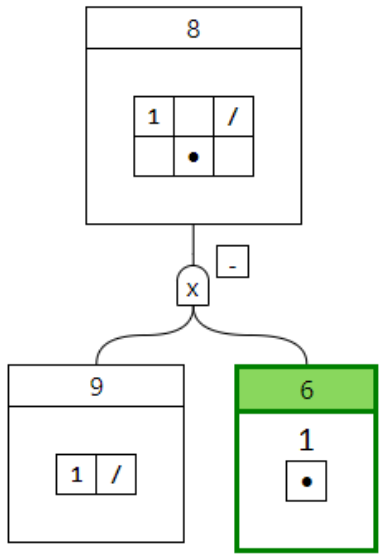




An example for $Av(123)$ and $Av(132)$

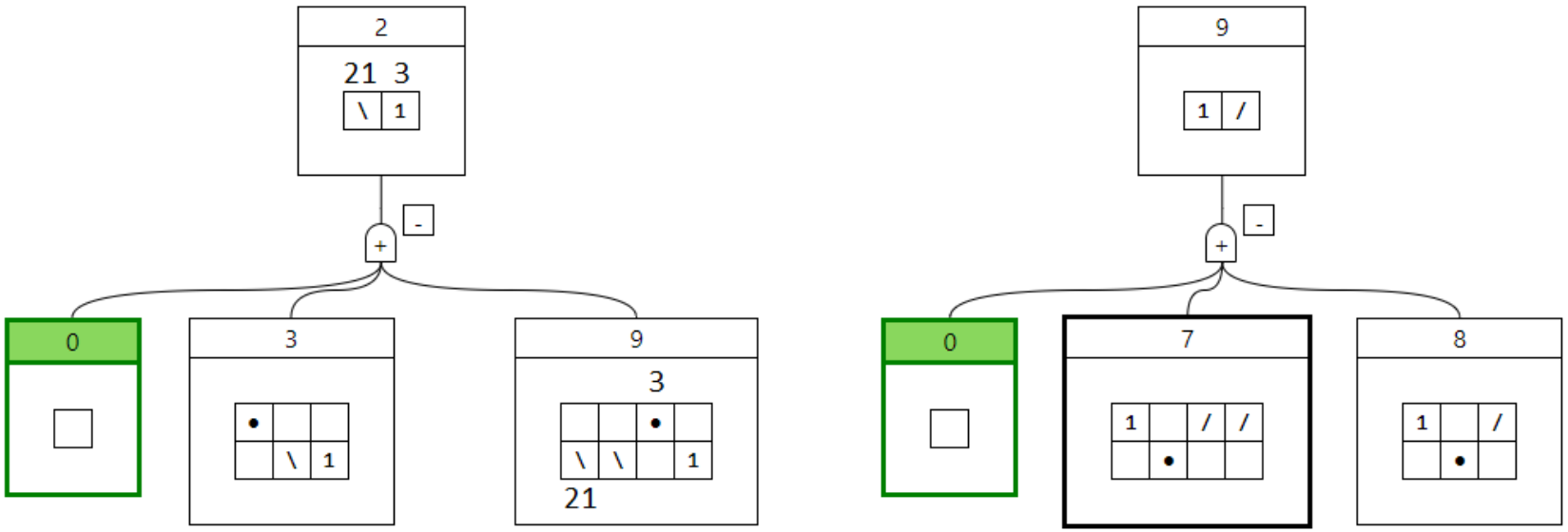
Factor





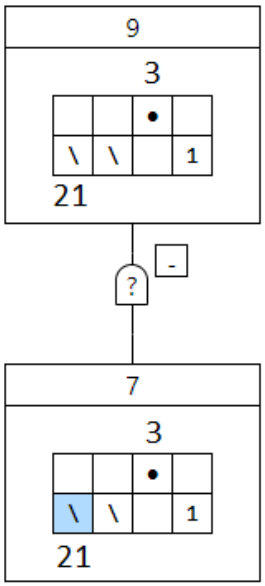
An example for $Av(123)$ and $Av(132)$

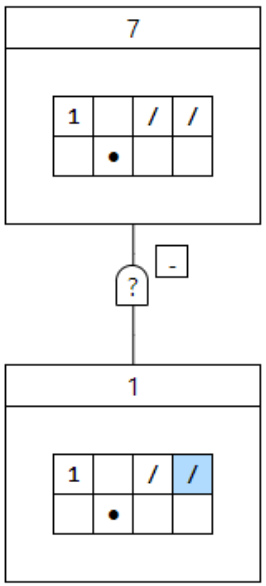
Place topmost in row



An example for $\mathbf{Av}(123)$ and $\mathbf{Av}(132)$

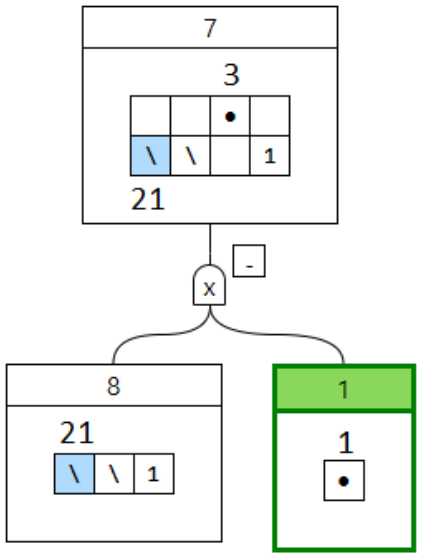
Add assumption in $(0, 0)$

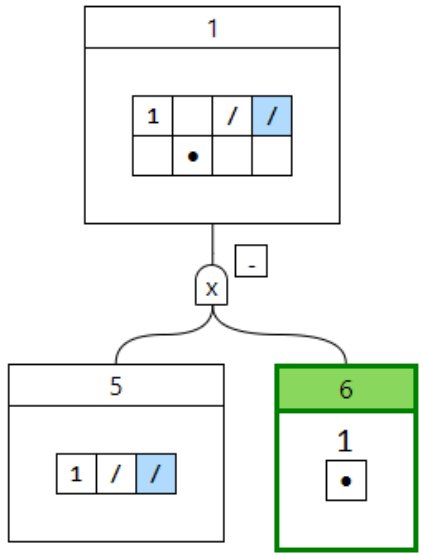




An example for $Av(123)$ and $Av(132)$

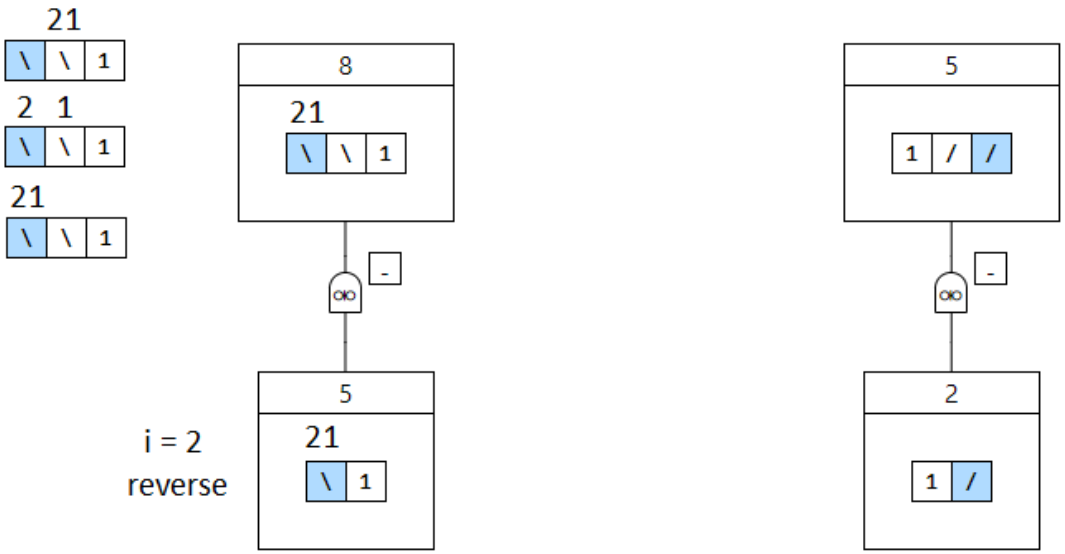
Factor





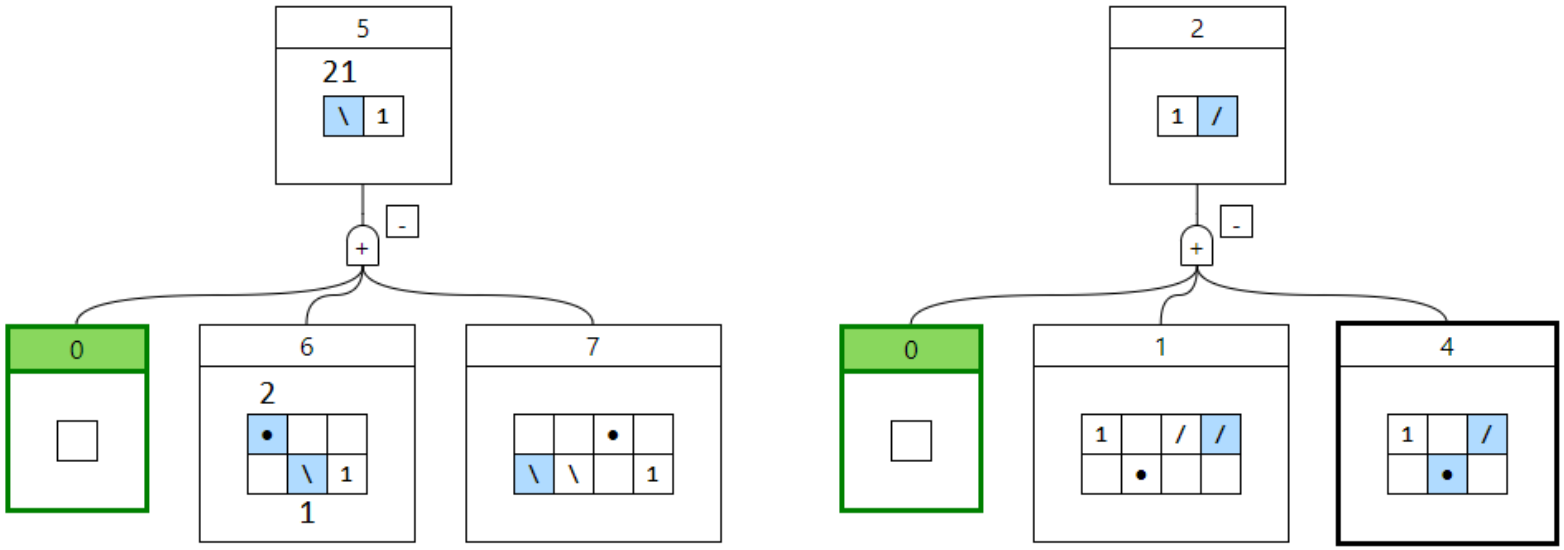
An example for $Av(123)$ and $Av(132)$

Fuse columns 0 and 1



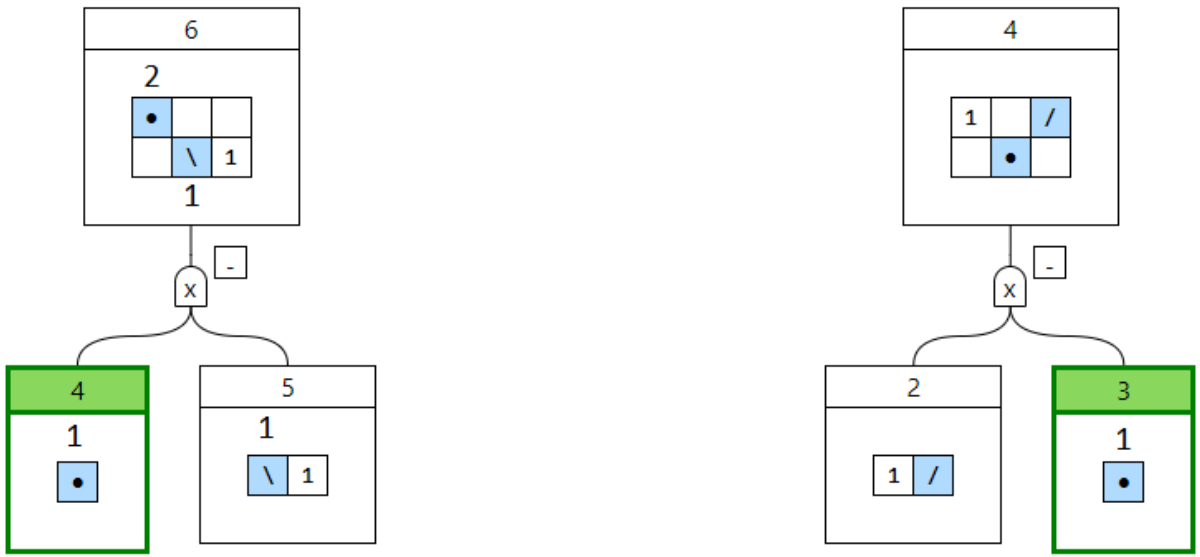
An example for $Av(123)$ and $Av(132)$

Place topmost in row



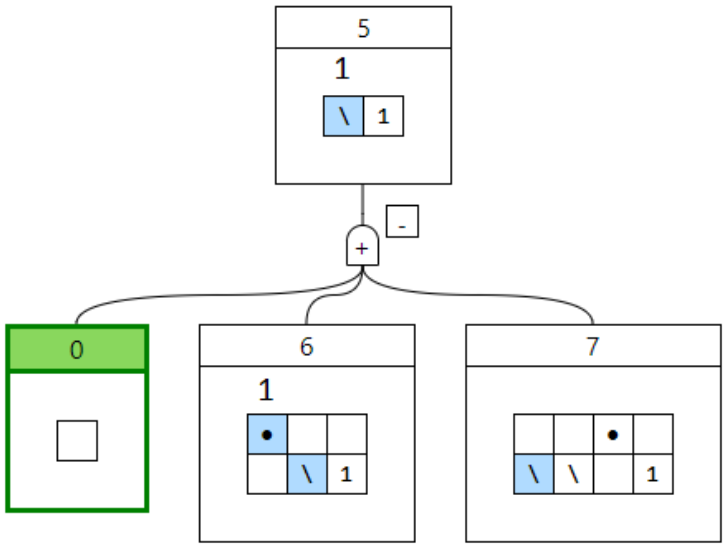
An example for $Av(123)$ and $Av(132)$

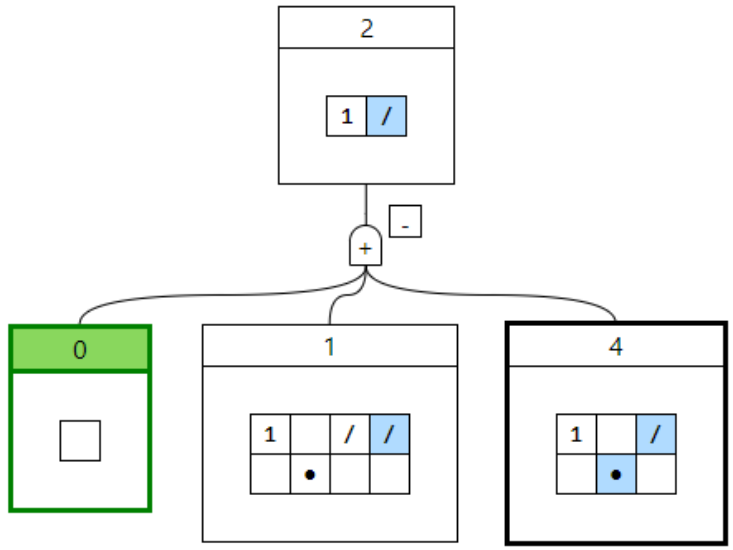
Factor



An example for $Av(123)$ and $Av(132)$

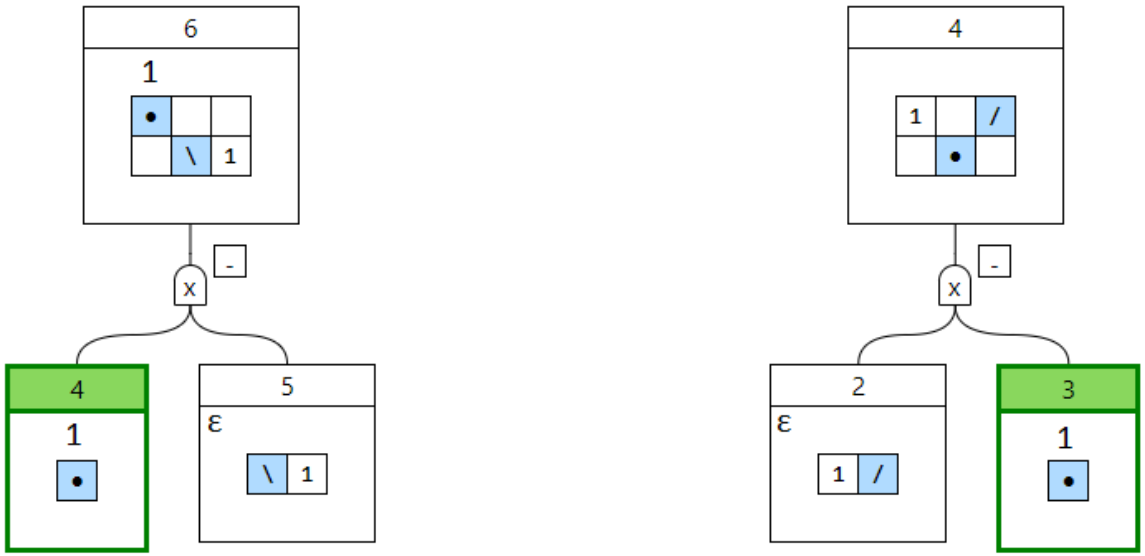
Place topmost in row





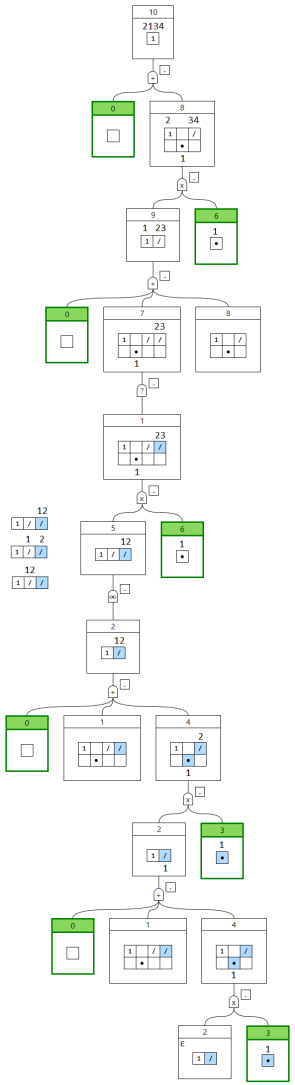
An example for $Av(123)$ and $Av(132)$

Factor



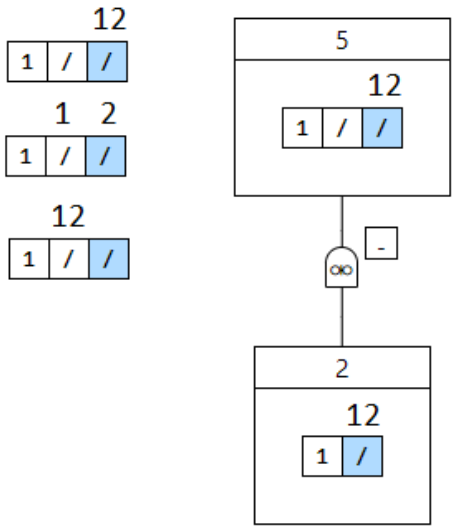
An example for $\mathrm{Av}(123)$ and $\mathrm{Av}(132)$

The parse tree we have created for $\mathrm{Av}(132)$.



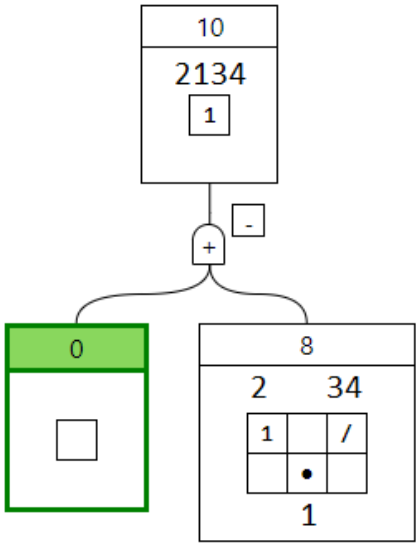
An example for $Av(123)$ and $Av(132)$

Backward for fusion



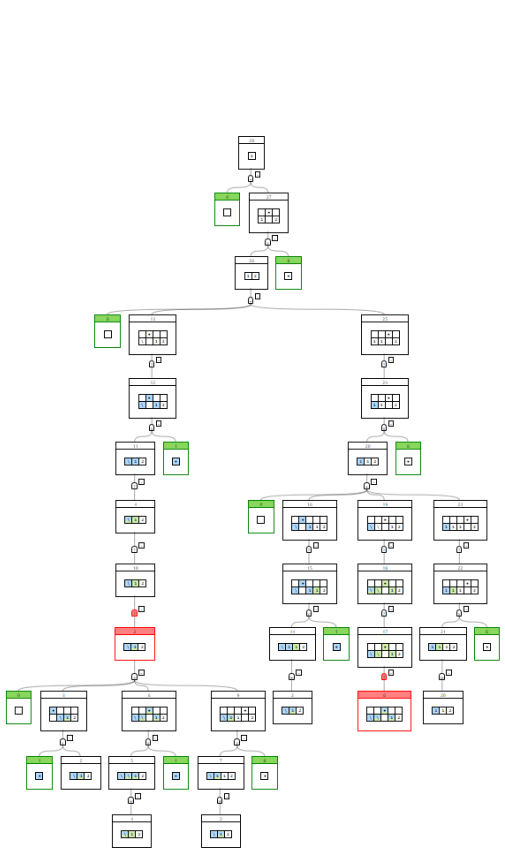
An example for $Av(123)$ and $Av(132)$

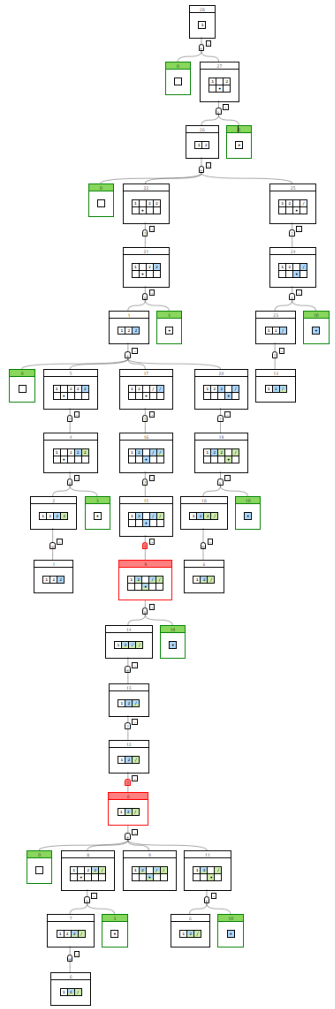
Final step



Successes

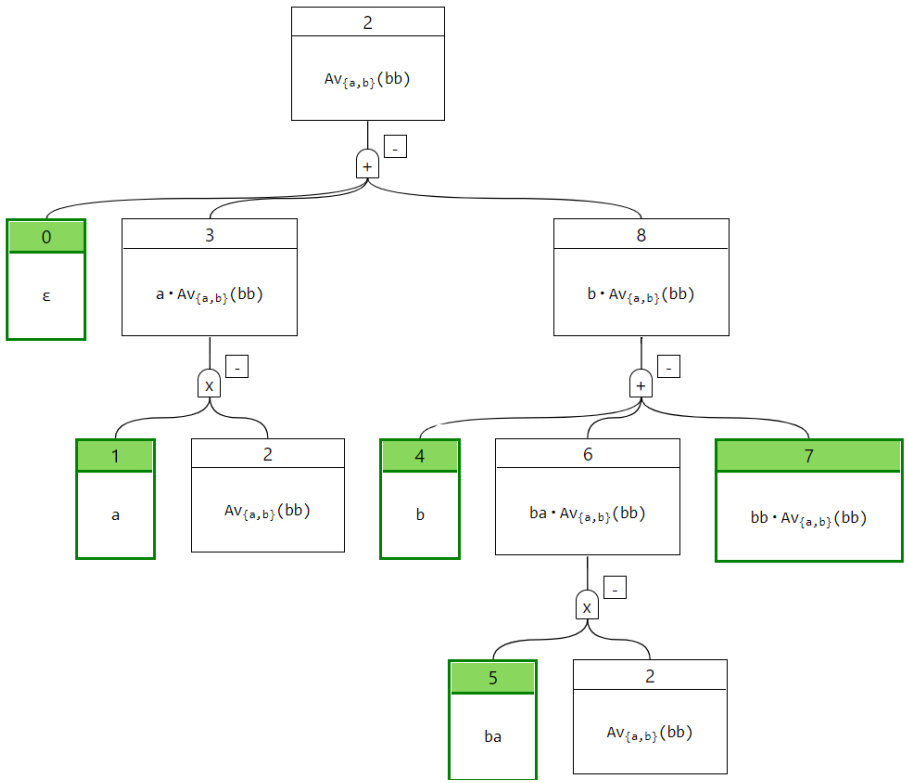
- $\mathrm{Av}(1234)$, $\mathrm{Av}(1243)$ and $\mathrm{Av}(1432)$.

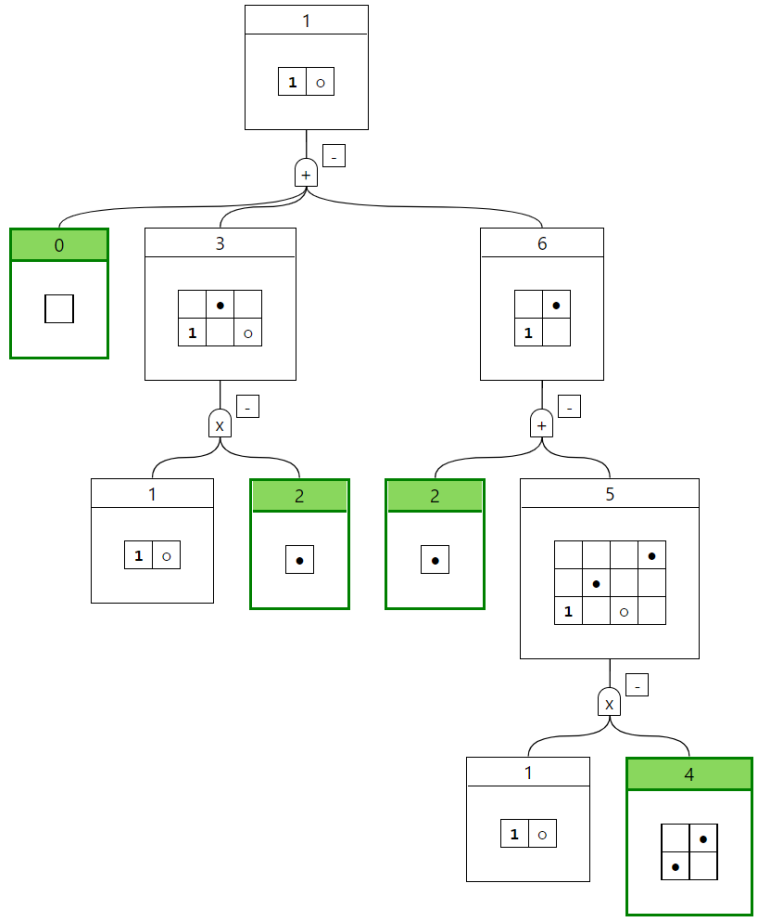




Successes

- $\mathrm{Av}(1234)$, $\mathrm{Av}(1243)$ and $\mathrm{Av}(1432)$.
- $\mathrm{Av}(231, 312, 321)$ and binary strings avoiding 11.





Successes

- $\mathrm{Av}(1234)$, $\mathrm{Av}(1243)$ and $\mathrm{Av}(1432)$.
- $\mathrm{Av}(231, 312, 321)$ and binary strings avoiding 11.
- $\mathrm{Av}(123)$ and $\mathrm{Av}(132)$.
- Classes avoiding one pattern of length 3 and one of length 4.
- Classes avoiding 11 patterns of length 4.
- Most classes avoiding 10 patterns of length 4. Still in progress.

Thanks for listening

Minimal code example

```
pack = TileScopePack.point_and_row_and_col_placements(row_only=True)
pack = pack.add_verification(BasicVerificationStrategy(), replace=True)
searcher1 = TileScope("0132_0213_0231_0321_1032_1320_2031_2301_3021_3120", pack)
searcher2 = TileScope("0132_0213_0231_0312_0321_1302_1320_2031_2301_3120", pack)
specs = ParallelSpecFinder(searcher1, searcher2).find()
bijection = Bijection.construct(*specs)
for p in bijection.domain.generate_objects_of_size(4):
    print(f"{p} -> {bijection.map(p)}")
```

- PermutaTriangle/Permuta

  General permutation library

- PermutaTriangle/comb_spec_searcher

  Domain independent specification searcher

- PermutaTriangle/Tilings

  Specification searcher for permutation classes