Permutation Patterns

Delyan Kirov

May 2023

1 Bivincular Pattern

Bivincular patterns (as written in the github model) are vincular patterns which additionally require some values to be adjacent. More formally, an occurrence of the bivincular pattern (π, A) in σ is a subsequence $\sigma(i_1)...\sigma(i_k)$ of σ such that the following all hold:

1. $\pi \leq \sigma$
2. $\forall a \in A. i_{a+1} = i_a + 1$
3. $\forall a \in A. \sigma(i_{\pi^{-1}(a+1)}) = \sigma(i_{\pi^{-1}(a)}) + 1$

2 Composed Model

```
    language ESSENCE 2.4.0

given length : int
# For a mesh (permutation, Mesh) we have a decomposition Mesh =
Bivincular + Consecutive + Mesh or something similar
given general_avoidance : set of (sequence (injective) of int, "composition of a mesh")

find perm : sequence (size length, injective) of int(1..length)

$ creating a padded version of perm, where position 0 contains the value 0 and
position length+1 contains the value length+1

find permPadded : matrix indexed by [int(0..length+1)] of int(0..length+1)
such that permPadded[0] = 0, permPadded[length+1] = length+1
such that forAll i : int(1..length) . permPadded[i] = perm(i)

# Why do we need the padded version for bivincular?

such that
    forAll (pattern, ind_bars, val_bars) in bivincular_avoidance .


        $ Look for all places where the pattern can occur
        # Is the matrix ix simply a subsequence of {0,1,2,3...|pattern|+1}
        forAll ix : matrix indexed by [int(0..|pattern|+1)] of int(0..length+1),
            and([ ix[0]=0
                , ix[|pattern|+1]=length+1
                # the matrix is increasing (as a sequence)?
                , forAll i : int(0..|pattern|) . ix[i] < ix[i+1]
                # the matrix avoids the patterns?
                , forAll n1, n2 : int(1..|pattern|) , n1 < n2 .
                    pattern(n1) < pattern(n2) <-> permPadded[ix[n1]] < permPadded[ix[n2]]
                ])

        # Is the matrix ix simply a subsequence of {0,1,2,3...|pattern|+1}

            .
            mesh_constraint /\ bivincular_constraint /\ consecutive_constraint + ...?
```

3 Questions

1. Why does the bivincular pattern use the padded matrix?
2. After you decompose the mesh, do you need to change anything?
3. SAT solve the decomposition vs python?