WORK LOG OF JUNE 13 2025

DANIEL R. BARRERO R.

1. General

• In splitForest, the typed value (lrdr,krdr) is matched against the typed value plusAssoc j1 j2', j2''.

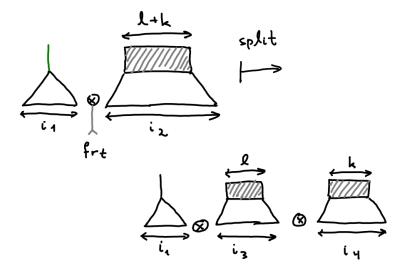
)

2. Type signatures and function evaluations

2.1. The type signature of the splitForest function is

```
splitForest :: forall f a b z q. SNat a -> SNat b ->
  Forest f q (a+b) \rightarrow
  forall j j'. (j+j') ~ q =>
  (Forest f j a, Forest f j' b) -> z
  ) -> z
  And the evaluation of its recursive case is given by
splitForest (SS (sl :: SNat 1))
            (sk :: SNat k)
            (Cons (t :: f j1) (frt :: Forest f j2 (1+k)))
  splitForest sl sk frt $
    (
     \((lrdr :: Forest f j2' 1),(krdr :: Forest f j2'' k)) ->
        case plusAssoc (j1 :: Proxy j1)
                        (j2' :: Proxy j2')
                        (j2'' :: Proxy j2'') of
        Dict -> c (Cons t lrdr , krdr)
```

this can be visualized in the following picture:



2.2. The signature of the Operad typeclass is

```
class (Graded f) Operad where
  ident :: f One
  compose :: f n -> Forest f m n -> fm
```

And our instance of interest is the type MoveTree, which is parametrized by the Nat kind. Its signature is coupled with that of Trees, also parametrized by Nat. These signatures are

```
data MoveTree n where
  Leaf :: MoveTree One
  Fan :: Trees n -> MoveTree n

data Trees n where
  NilT :: Trees Z
  (:+) :: (Move, MoveTree a) -> Trees b -> Trees (a+b)
```