Combinatorics

Combinatorics are worked out in the routines `genflavreglist`, `find_regions`, `reorder_regions`, and other minore ancillary routines. The driving routine `genflavreglist` is called, after the `flst_nreal`, `flst_real(nlegreal,flst_nreal)`, `flst_nborn`, `flst_born(nlegborn,flst_nborn)` are filled with the flavour structure of the real and born processes. The format is the following:

`flst_real(k=1...nlegreal,j)`: flavours of the legs of the jth real graph

`flst_born(k=1,...,nlegborn,j)`: flavours pf the legs of the jth born graph

Flavour format: integer number corresponding to the PDG id particle code, except for gluons, that are assinged 0 instead of 21.

Call sequence:

`call genflavreglist`

Effects:

- `flst_alr(1...nlegreal,1...flst_nalr)` is filled with the flavour structures of all the regions α_r (the "alr" in the name stands for alpha-r) as defined in the FNO paper. These are computed in the following way:

  – To each gluon in the final state, one associates an ISR region.

  – To each quark in the final state, if it is equal to a quark in the initial state one associates an ISR region, defined by the incoming and outgoing quark.

  – To each quark-gluon (or antiquark-gluon) pair in the final state one associates a FSR region.

  – To each gluon-gluon pair in the final state we associate two regions, corresponding to each of the gluons being soft.

  – To each quark-antiquark pair of the same flavour in the final state we associate a FSR region.

  We then reorder the flavours in a region in the following way

  – In ISR region, the radiated parton is moved to the last position in the flavour list (i.e. to the `nlegreal` positin)

  – In a FSR region, the two particles involved are moved to the endo of the list, making sure that the particle that can give soft singularities is moved to the last position, and if we have a $q\bar{q}$ pair, the q is moved to the last position.

  After the reordering, we assign to the `flst_emitter(j=1...flst_nalr)` array the value

  – 0 if the `flst_alr(...,j)` region is an ISR region with an emitted gluon

  – 1(2) if it is an ISR region collinear to 1(2), and the emitted parton is not a gluon

  – The value `nlegreal-1` for all other (FSR) cases

This definition correspond to the emitting particles, except in the firsts case, where the value 0 is used to mean 1 and 2 at the same time.

- As a consequence of the reordering, some regions may turn out to obe identical. We thus goe through the list of regions and lump together identical region. The value of `flst_nalr` is reduce to account for this lumping procedure, and the array `flst_mult(1...flst_nalr)` is filled with the multiplicity of each region. The array `flst_emitter` is updated accordingly.

- The underlying Born graph fllavour associated to each region is computed, and stored in the array `flst_uborn(1...nlegborn,1...flst_nalr)`.

- At this stage we run through the list, and look for the underlying Born flavour sturture in the list of born graphs given in `flst_born(1...nlegborn,1...flst_nborn)`. We may find the underlying Born structure in this list, in which case we set the pointer `flst_alr2born(j)` (where `j` is the index of the region we are considering) to the index `k` such that `flst_born(1...nlegborn,k)` is equal to `flst_uborn(1...nlegborn,j)`. It may happen, however, that we find a flavour structure that is equivalent only up to the permutation of some final state coloured partons. Also in this case we set the `flst_alr2born(j)` index as before, but we also perform a permutation of final state particles upon the arrays `flst_alr(1...nlegreal,j)`, `flst_uborn(1...nlegborn,j)`, and `flst_emitter(j)` if the permutation changes the position of the `nlegborn` element of the list. At the end of this procedure, `flst_emitter` may end up being different from `nlegborn`. This fact can take place if the Born graphs have more than one coloured massless parton in the final state, as, for example, in dijet production.

- At this stage, an array of arrays of pointers

  `flst_born2alr(0...flst_nalr,1...flst_nborn)`

  is setup, that associated to each Born flavour structure, a list of the regions that share this same underlying born. The integer `flst_born2alr(0,k)` stores the number of regions that have `flst_born(1...nlegborn,k)` as underlying Born, and `flst_born2alr(1...flst_born(1...nlegborn,k),k)` are the indices in the region list.

- As a final step, for each region in the list, we also need a list of all the singular regions associated to that flavour structures. This is needed, because the contribution of each region is given by the real graph multiplied by weight factors S_i, S_{ij}. These in turn have the structure described in section 2.4.1 of FNO. In order to compute them, we need a list of all singular regions associated to the given real graph flavour structure. We thus setup the array

  `flst_allreg(1:2,0:maxregions,1:flst_nalr)`

  the first two elements corresponding to the emitter and emitted parton, the integer `flst_allreg(1,0,j)=m` is the number of singular regions associated to the flavour structure of the `jth` region, and `flst_allreg(1:2,1:m,j)` is the list of singular regions associated to the `jth` region.