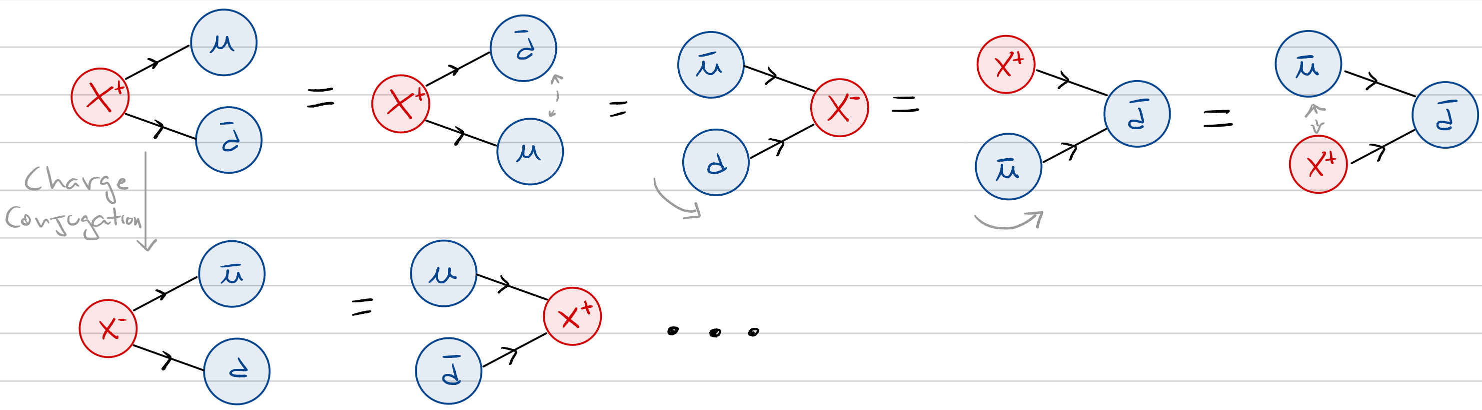
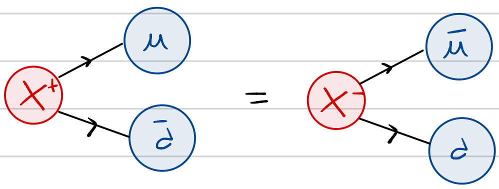
— Vertex: Canonical Representation

- We need to be able to match vertices irrespective of their ordering:

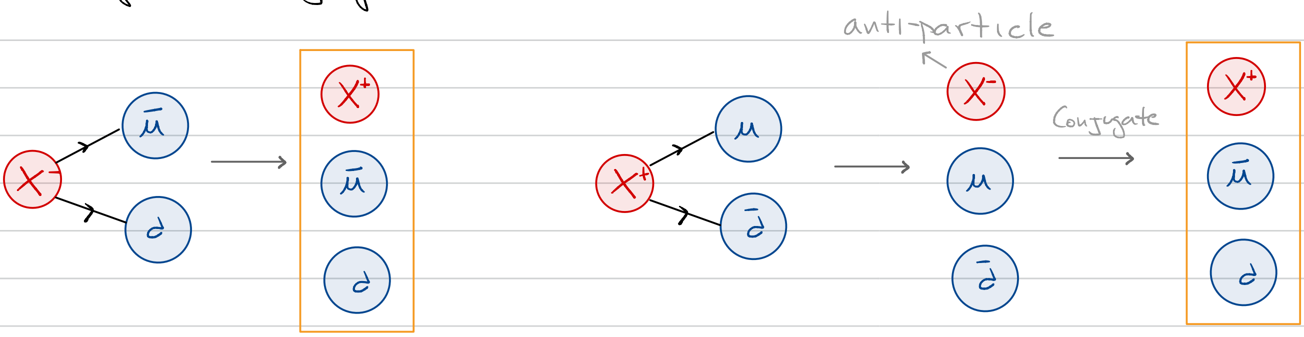


- In addition vertices differing only by charge conjugation should also be considered as equivalent (assumes C-conservation):

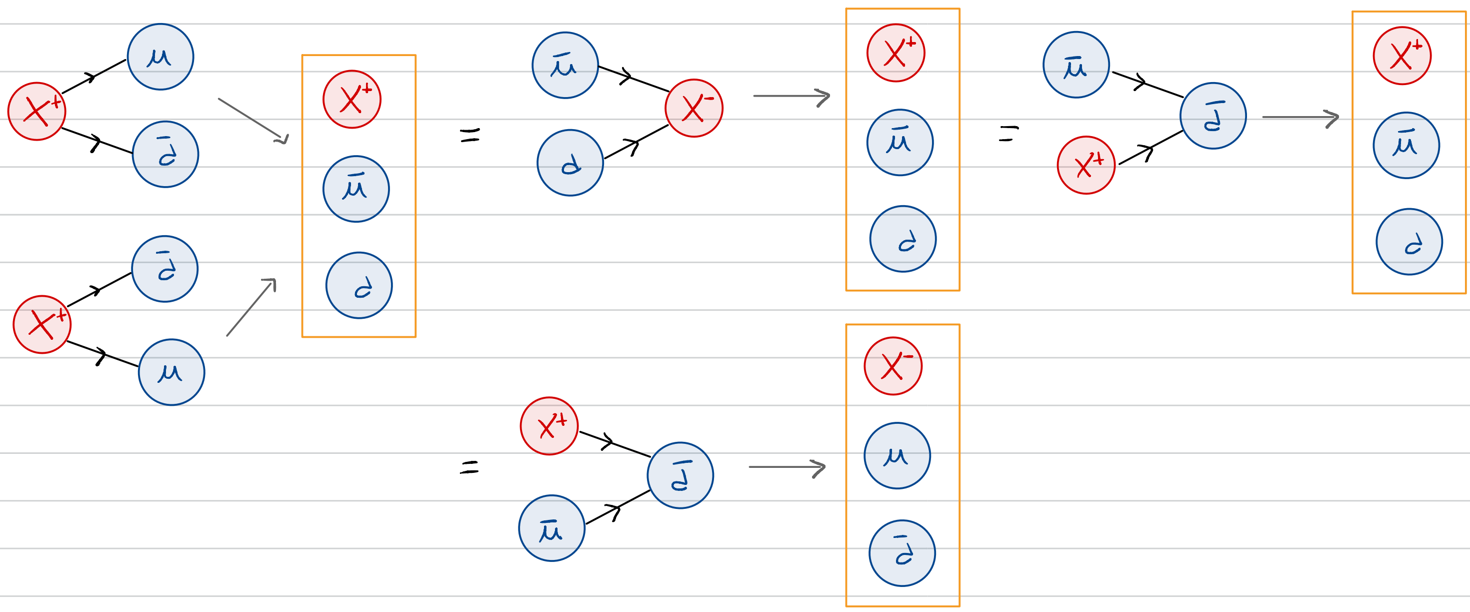


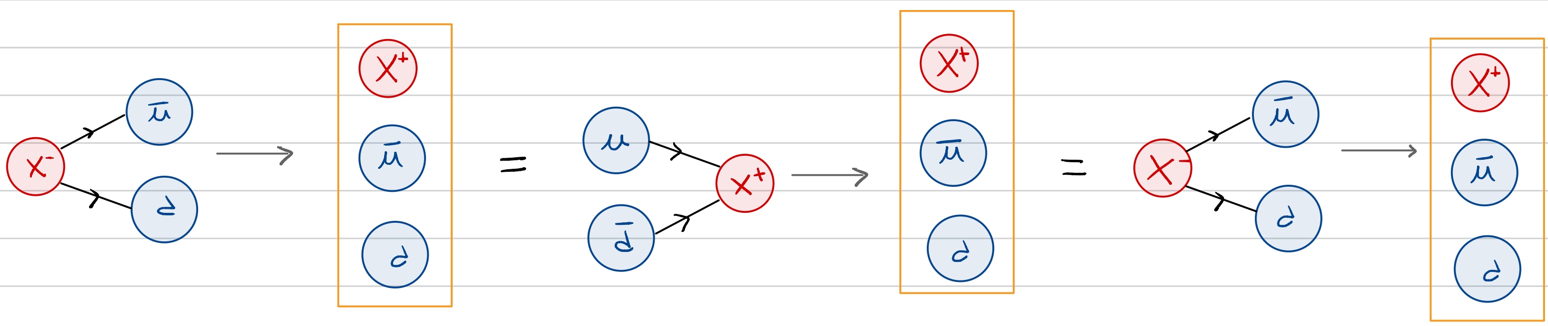
With this definition it is not necessary to keep track of all the vertices and their charge conjugation.

- Hence to simplify the matching all vertices are represented with all particles as outgoing. In addition if the particle with largest |PDG| and which is not its own anti-particle is an anti-particle (PDG<0), all the particles are charge conjugated:



• Applying this definition of a "canonical" representation of a vertex we have:

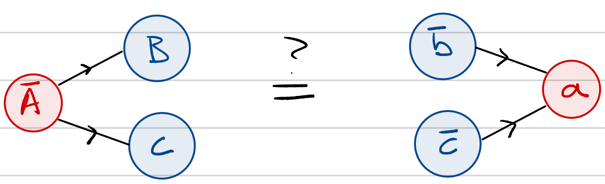


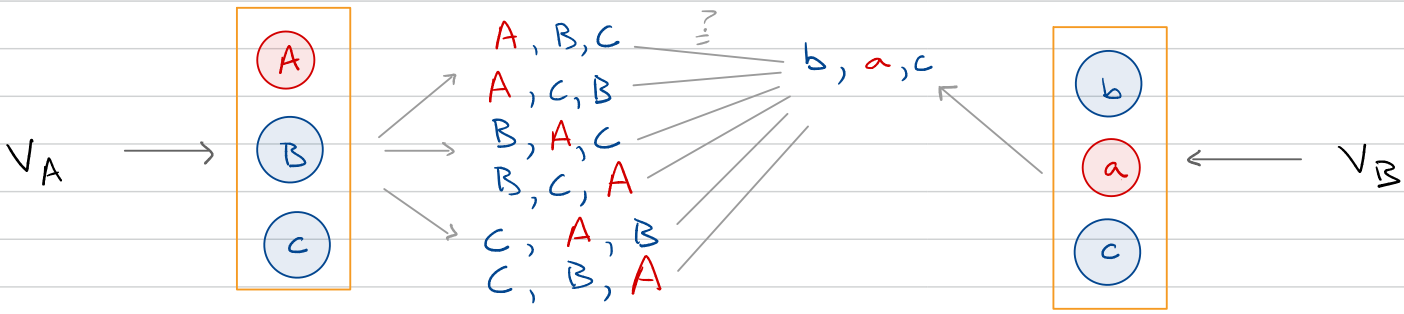


— Vertex: Matching

• Using this "canonical representation" of a vertex, it is only need to compare two lists of particles irrespective of their ordering.

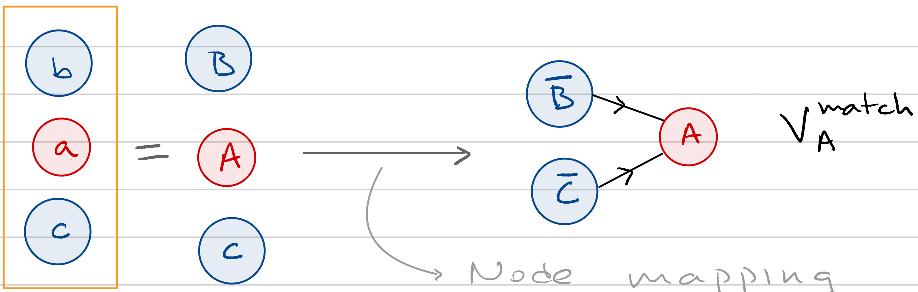
$$V_A \stackrel{?}{=} V_B$$





$\longrightarrow$ Output : B, A, C = b, a, c

- Once the vertices match, V_A can be brought to V_B format:



Node mapping + charge conjugation
(if the canonical rep. applied charge conjugation of the whole vertex)

— Vertex: String Representation

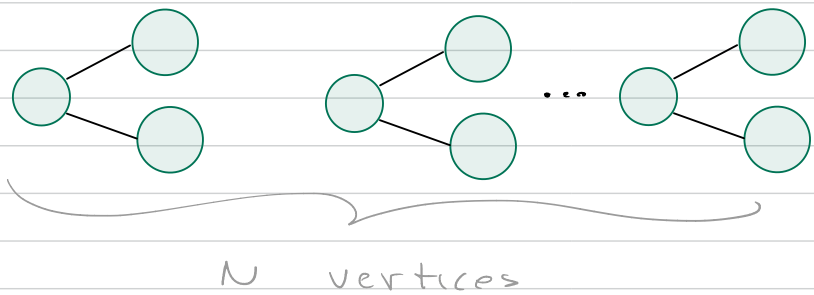
- The vertex $\longleftrightarrow$ string mapping is given by:

$$\underbrace{A, B, C}_{\text{Incoming Particles}} > \underbrace{a, b, \ldots}_{\text{Outgoing Particles}}$$

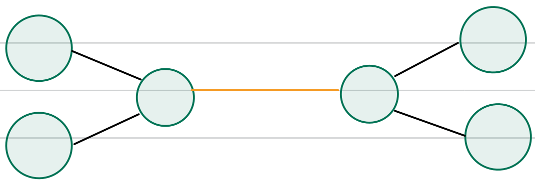
— Building $2 \to N$ Topologies

- Assuming we only have 3-point vertices, the $2 \to N$ topologies can be built (for a given N) with the following steps:

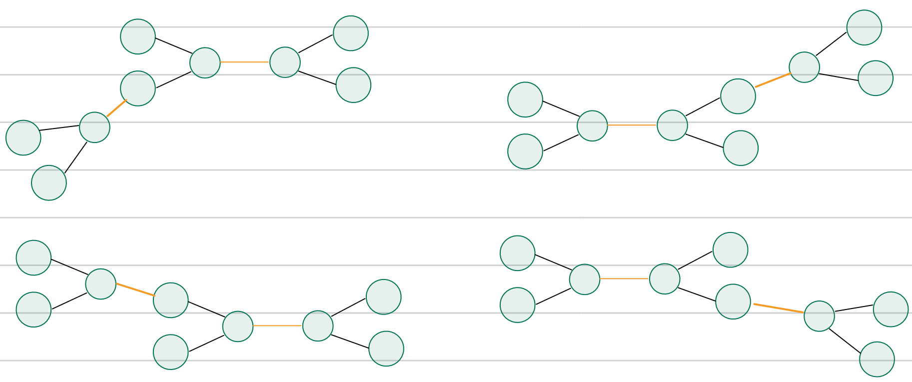
1) Create N vertices with empty or generic particles



2) Connect two vertices with an edge ("propagator") to obtain a base topology



3) Connect the 3rd vertex adding an edge to each possible node, generating multiple topologies:

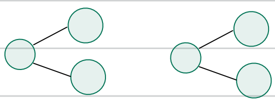
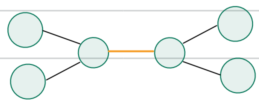


4) For each new topology repeat step 3 until all vertices have been attached

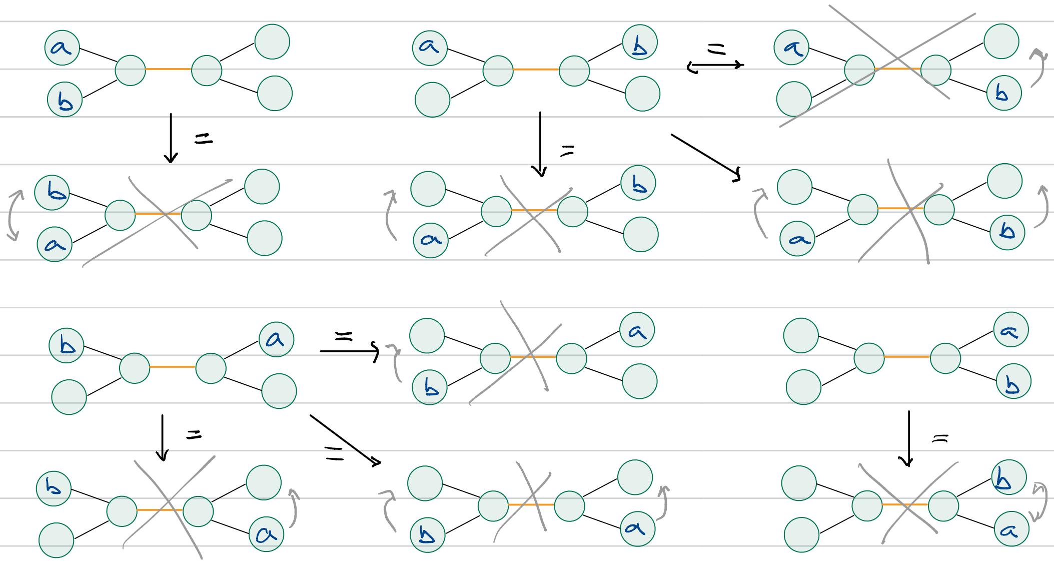
5) For each final topology choose all combinations of pairs of external nodes as the initial states. The remaining N nodes will be final states

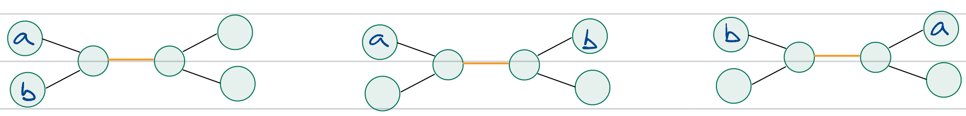
- The procedure above is general, but it can lead to duplicate diagrams.

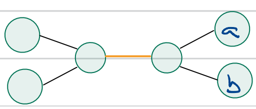
— Exemplo: $2 \to 2$

1)  → 2)  → 3), ~~4)~~

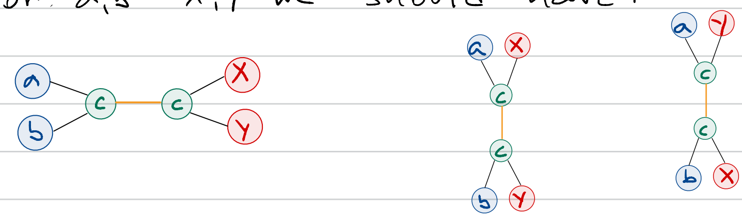
→ 5)



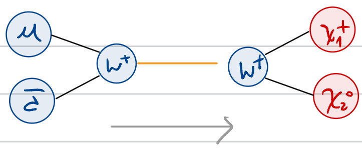




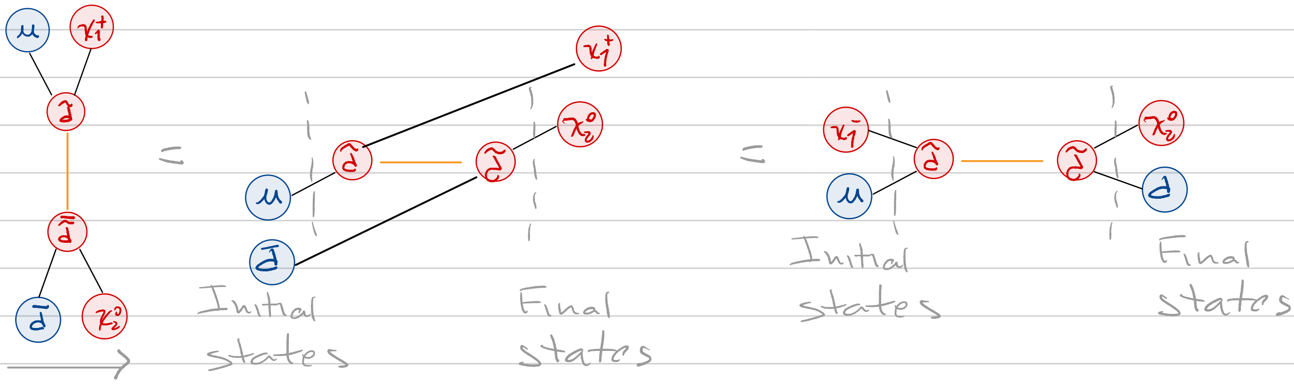
For $a, b \to X, Y$ we should have:



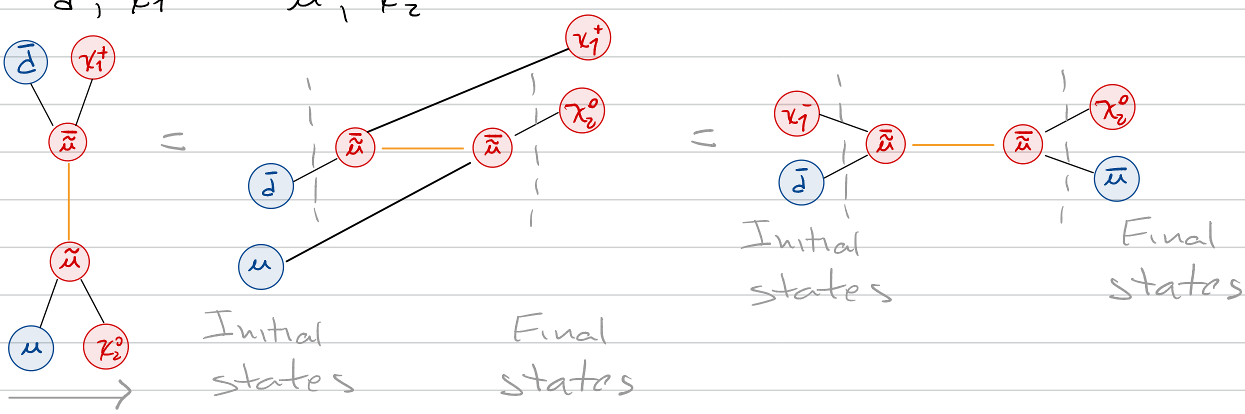
$u, \bar{d} \to \tilde{\chi}_1^+ \, \tilde{\chi}_2^0$



$\longrightarrow \quad u, \chi_1^- \to d, \chi_2^0$



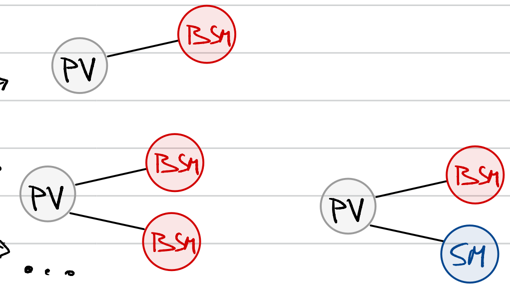
$\longrightarrow \quad \bar{d}, \chi_1^- \to \bar{u}, \chi_2^0$



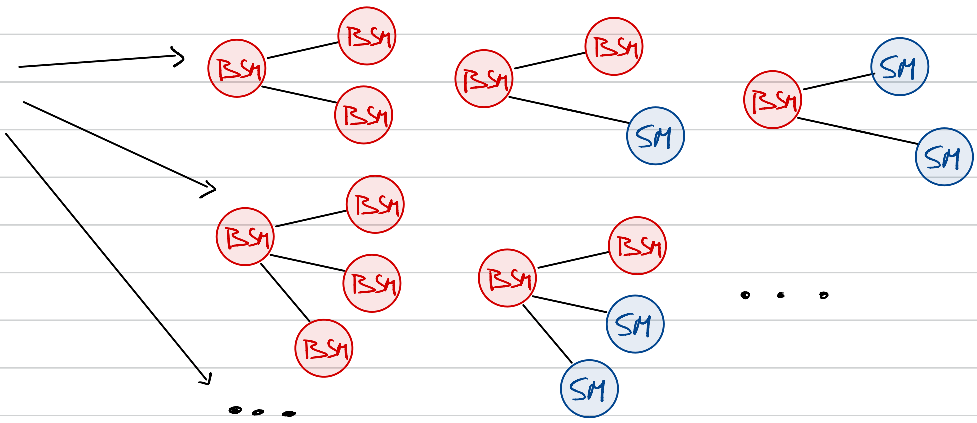
— Creating Production Diagrams - SModelS

- Input:

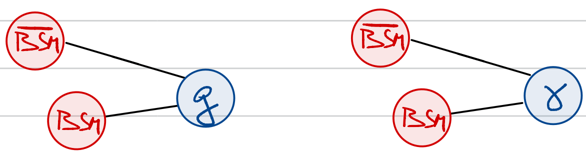
Cross - Sections



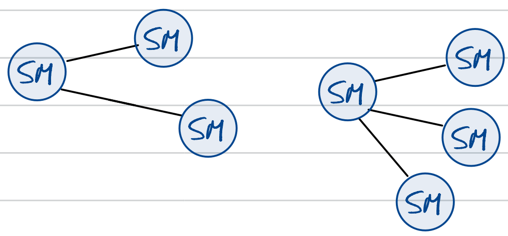
Decays



Quantum Numbers →

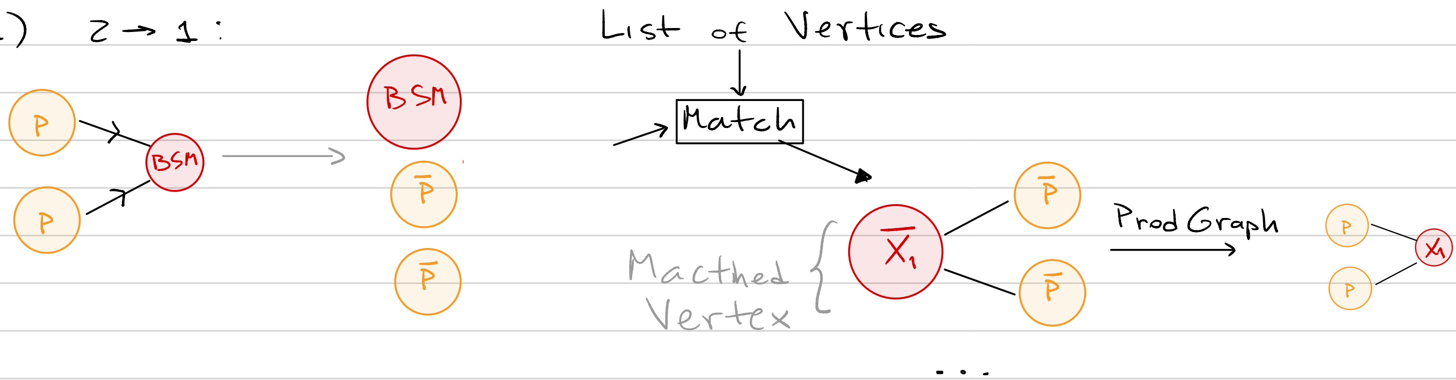


SM vertices →

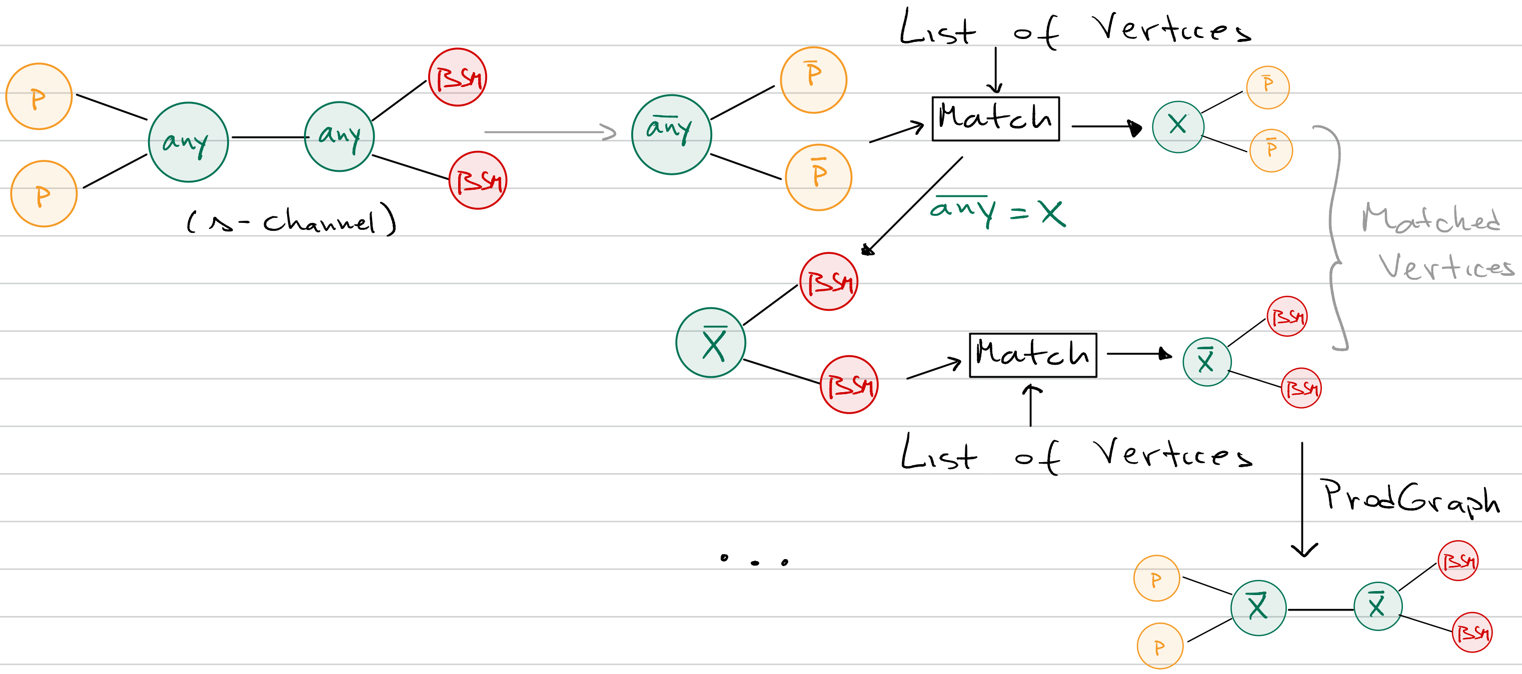


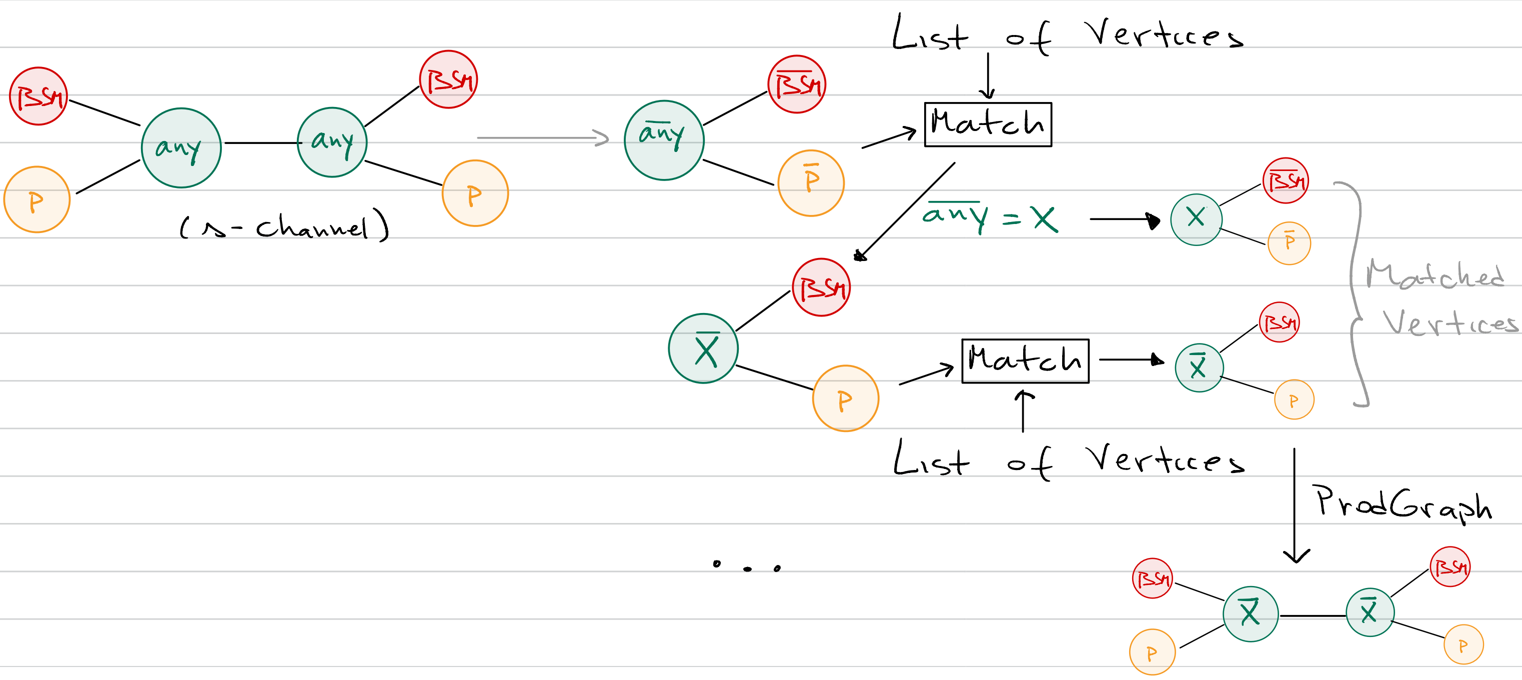
• <u>Diagrams</u> :

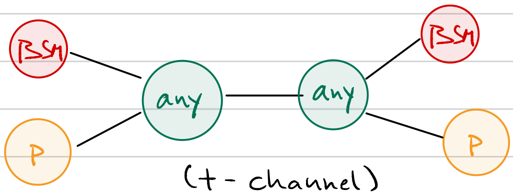
1) $2 \to 1$:



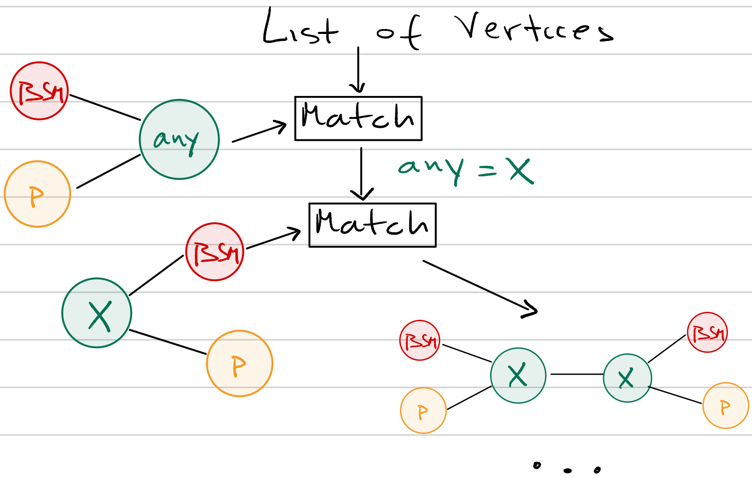
2) $2 \to 2$:







(t-channel)



+ 3-body Decay ??