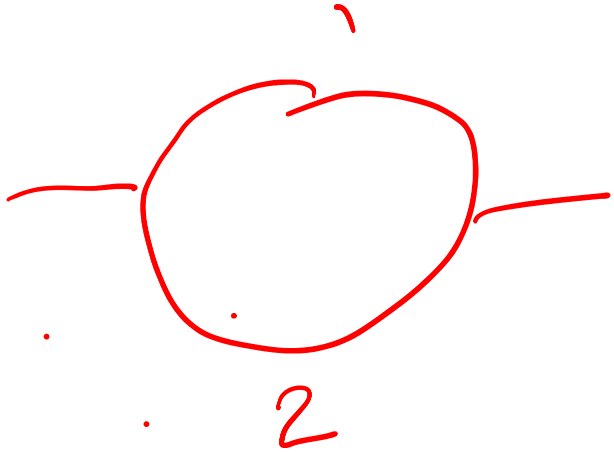
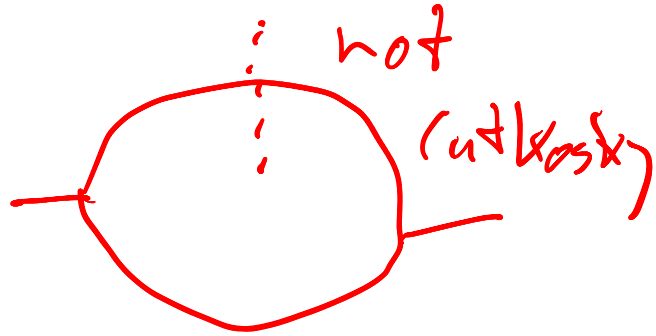
FEYNMAN DIAGRAMS AND THE S-MATRIX, AND OUTER SPACE (SUMMER 2020)

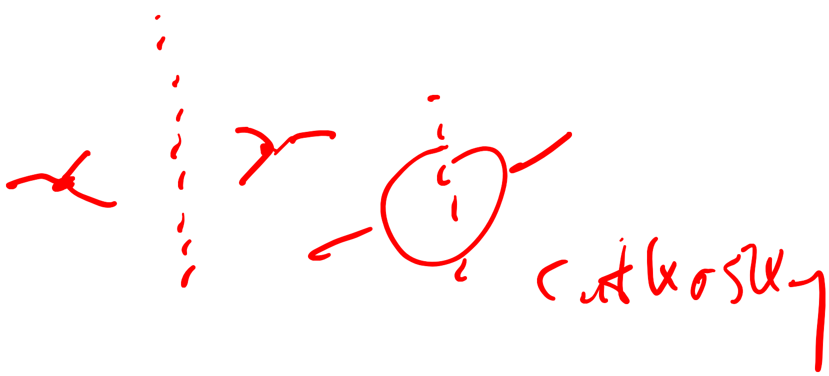
DIRK KREIMER (LECT. APRIL 27, 2020)

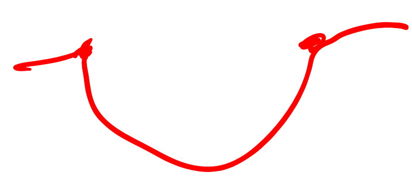
1. Feynman and Cutkosky Graphs

In this lecture we define Feynman graphs and Cutkosky graphs. The latter are graphs G where a subset E_{on} of their internal edges E_G is distinguished: 'removing' the edges $e \in E_{on}$ from G decomposes the graph into various bridge-free components.









1.1. **Feynman graphs.** We first settle the notion of a partition.

Definition 1.1. Given a set S a *partition* (or *set partition*) $\mathcal{P}$ of S is a decomposition of S into disjoint nonempty subsets whose union is S. The subsets forming this decomposition are the *parts* of $\mathcal{P}$. The parts of a partition are unordered, but it is often convenient to write a partition with k parts as $\dot{\cup}_{i=1}^{k} S_i = S$ with the understanding that permuting the S_i still gives the same partition. A partition $\mathcal{P}$ with k parts is called a k-partition and we write $k = |\mathcal{P}|$.

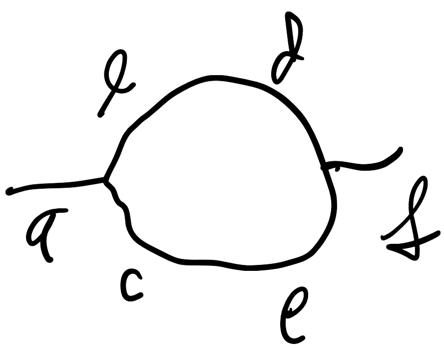
Now we can define a Feynman graph.

Definition 1.2. A *Feynman graph* G is a tuple $G = (H_G, \mathcal{V}_G, \mathcal{E}_G)$ consisting of

- H_G, the set of half-edges of G,
- $\mathcal{V}_G$, a partition of H_G with parts of cardinality at least 3 giving the vertices of G,
- $\mathcal{E}_G$, a partition of H_G with parts of cardinality at most 2 giving the edges of G.

From now on when we say graph we mean a Feynman graph.

a, b, c, d, e, f $\quad \mathcal{E}_G \quad \mathcal{V}_G$

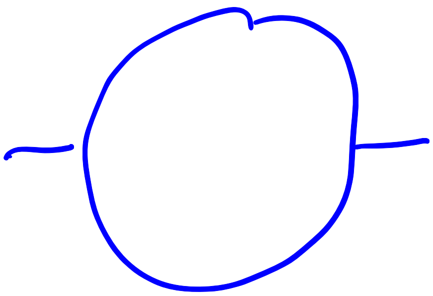


We do not require all parts of $\mathcal{E}_G$ to be of cardinality 2. We identify the parts of cardinality 2 with the set of edges E_G of the graph and set $e_G := |E_G|$. We identify the sets of cardinality 1 with the set of external edges L_G of the graph and set $l_G := |L_G|$. Also we set $v_G := |\mathcal{V}_G|$.

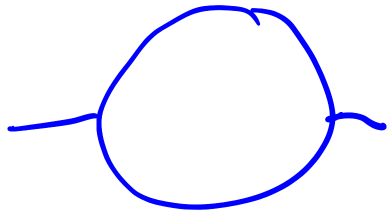
We say that a graph G is connected if there is no partition of H_G into two sets $H_G(1), H_G(2)$ such that the parts of cardinality two of $\mathcal{E}_G$ are either in $H_G(1)$ or $H_G(2)$. If it is not connected it has $|H^0(G)| > 1$ components.

$\mathcal{E}_G = \mathcal{E}_{G(1) \cup G(2)}$

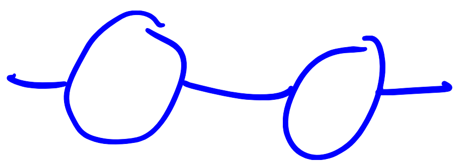
$G(1)$



$G(2)$



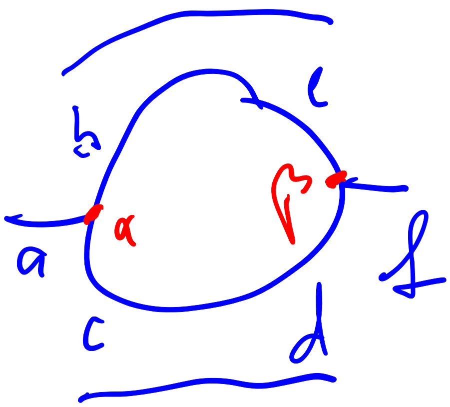
$H_G(1) \cap H_G(2) = \emptyset$



The partition $\mathcal{V}_G$ collects half-edges of G into vertices. This formulation of graphs does not distinguish between a vertex and the corolla of half-edges giving that vertex. However, it is sometime useful to have notation to distinguish when one should think of vertices as vertices and when one should think of them as corollas. Consequently let V_G, the set of vertices of G, be a set in bijection with the parts of $\mathcal{V}_G$, $|V_G| = v_G = |\mathcal{V}_G|$. This bijection can be extended to a map $\nu_G : H_G \to V_G$ by taking each half edge to the vertex corresponding to the part of $\mathcal{V}_G$ containing that vertex. For $v \in V_G$ define

$$C_v := \nu_G^{-1}(v) \subset H_G,$$

to be the corolla at v, that is the part of $\mathcal{V}_G$ corresponding to v.



$$\nu_G^{-1}(\alpha) = \{a, b, c\}$$

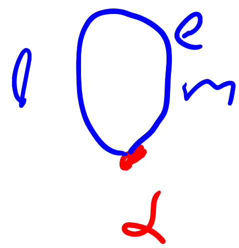
$$\nu_G^{-1}(\beta) = \{d, e, f\}$$

A graph G as above can be regarded as a set of corollas determined by $\mathcal{V}_G$ glued together according to $\mathcal{E}_G$.

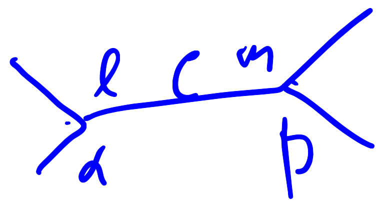
If $|\nu_G(e)| = 1$, we say e is a self-loop at v, with $\nu_G(e) = \{v\}$.

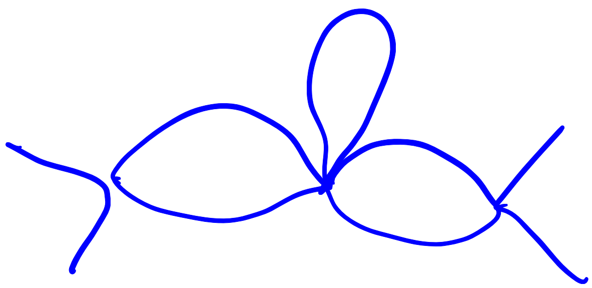
We frequently have cause to make an arbitrary choice of an orientation on the edges. If $|\nu_G(e)| = 2$, with $e = \{l, m\}$ and $\nu(l) = v, \nu(m) = w$ say, e is an edge e_{vw} from v to w or e_{wv} vice versa for the opposite orientation. This choice of an edge orientation corresponds to a choice of an order of e as a set of half-edges.

We emphasize that we allow multiple edges between vertices and allow self-loops as well.



$\nu_G(l) = \nu_G(m) = v$





legal!

We write $|G| := |H^1(G)| = e_G - v_G + |H^0(G)|$ for the number of independent loops, or the dimension of the cycle space of the graph G. Note that for disjoint unions of graphs h_1, h_2, we have $|h_1 \cup h_2| = |h_1| + |h_2|$.

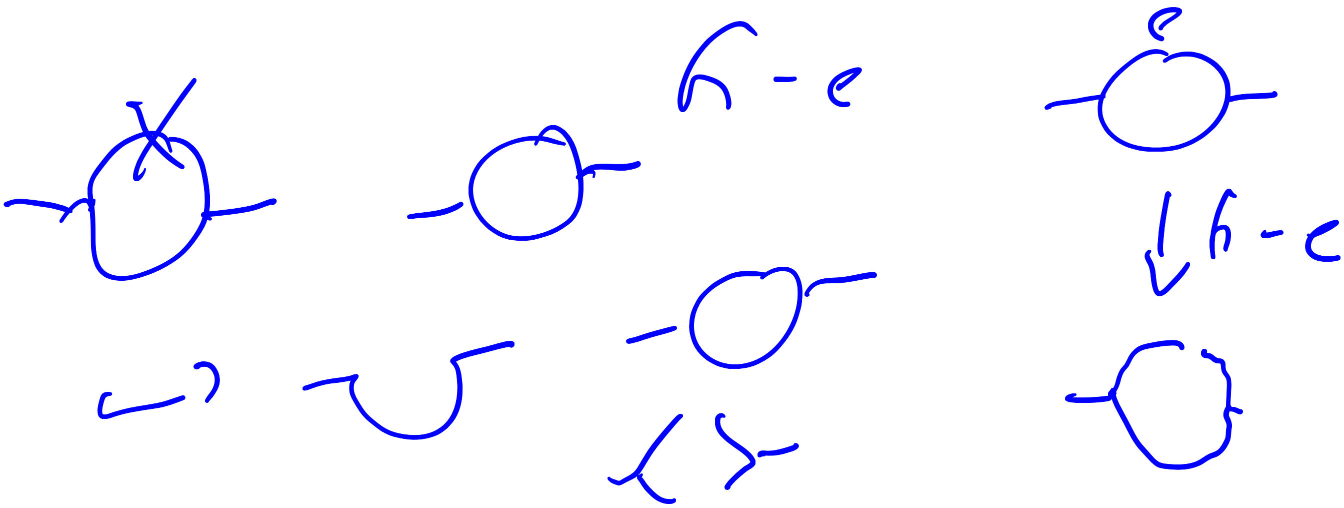
A graph is bridgeless if $(G - e)$ has the same number of connected components as G for any $e \in E_G$. A graph is 1PI or 2-edge-connected if it is both bridgeless and connected, equivalently if $(G - e)$ is connected for any $e \in E_G$. Here, for $G = (H_G, \mathcal{V}_G, \mathcal{E}_G)$, we define

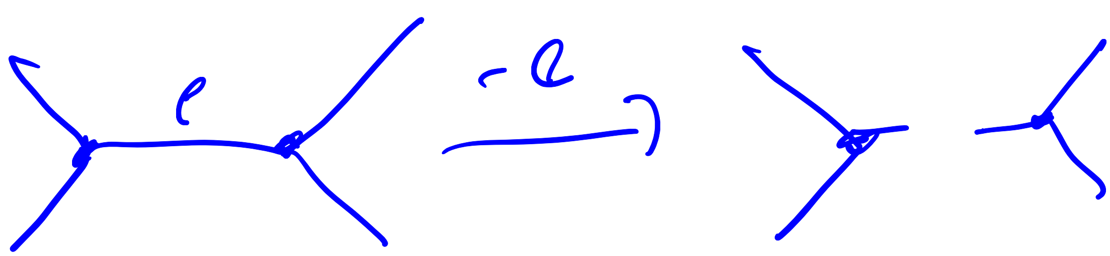
$$(G - e) := (H_G, \mathcal{V}_G, \mathcal{E}'_G)$$

where $\mathcal{E}'_G$ is the partition which is the same as $\mathcal{E}_G$ except that the part corresponding to e is split into two parts of size 1.

The removal $G - X$ of edges forming a subgraph $X \subset G$ is defined similarly by splitting the parts of $\mathcal{E}_G$ corresponding to edges of X. $G - X$ can contain isolated corollas.

Note that this definition is different from graph theoretic edge deletion as all the half-edges of the graph remain and the corollas are unchanged. We neither lose vertices nor half-edges when removing an internal edge. We just unglue the two corollas connected by that edge.





The graph resulting from the contraction of edge e, denoted G/e for $e \in E_G$, is defined to be

$$G/e = (H_G - e, \mathcal{V}'_G, \mathcal{E}_G - e) \tag{1.1}$$

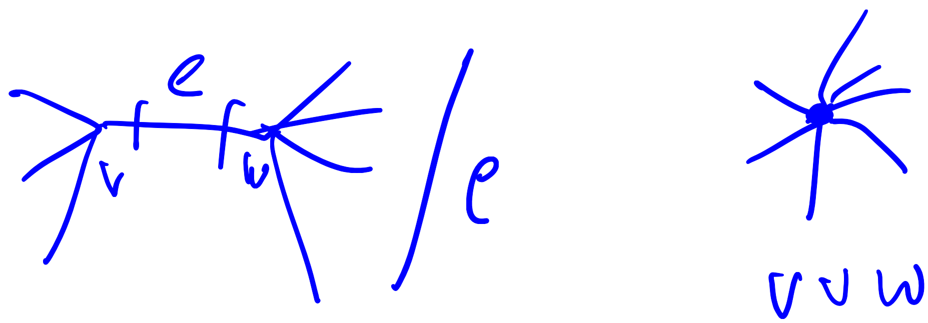
where $\mathcal{V}'_G$ is the partition which is the same as $\mathcal{V}_G$ except that in place of the parts C_v and C_w for $e = \{\nu^{-1}(v), \nu^{-1}(w)\}$, $\mathcal{V}'$ has a single part $(C_v \cup C_w) - e$.[1]

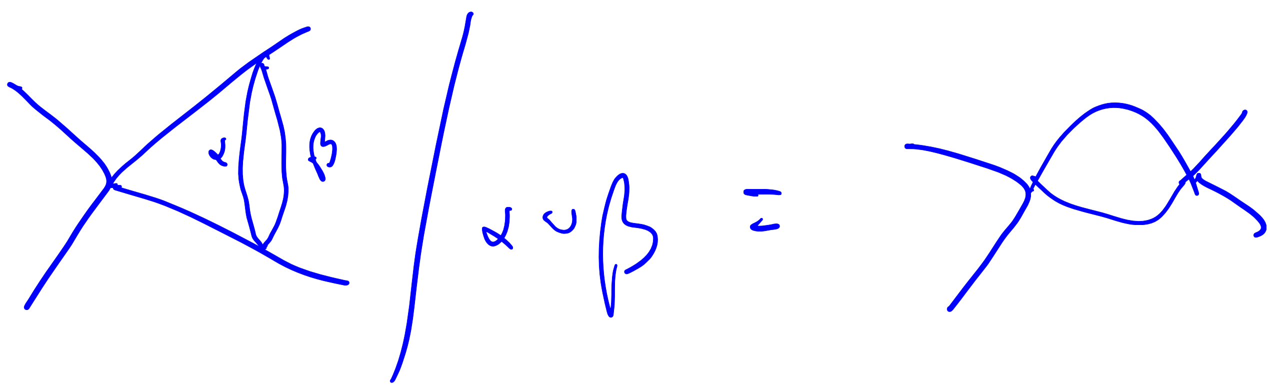
Likewise we define G/X, for $X \subseteq G$ a (not necessarily connected) graph, to be the graph obtained from G by contracting all internal edges of $X \subseteq G$.

Intuitively we can think of G/X as the graph resulting by shrinking all internal edges of X to zero length:

$$G/X = G|_{\text{length}(e)=0, e \in E_X}. \tag{1.2}$$

This intuitive definition can be made into a precise definition if we add the notion of edge lengths to our graphs, but doing so is not to the point at present.



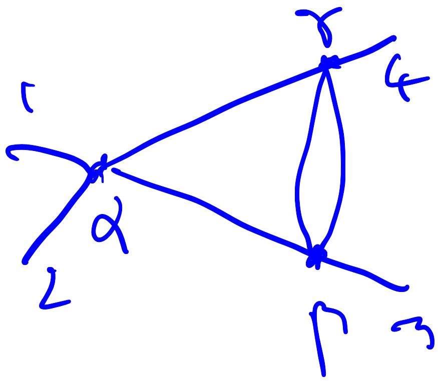


[1]We often use $-$ for the set difference, e.g. $H_G - e = H_G \setminus e$.

Note that restricting $\mathcal{V}_G$ to L_G we also obtain a partition of L_G into the sets $L_G \cap \nu_G^{-1}(v)$:[2]

$$L_G = \dot{\cup}_{v \in V_G} \underbrace{\left(L_G \cap \nu_G^{-1}(v)\right)}_{=:L_v}.$$

We let $\mathbf{val}(v) := |C_v|$ the degree or valence of v and $\mathbf{eval}(v) := |L_v|$ the number of external edges at v, and $\mathbf{ival}(v) := \mathbf{val}(v) - \mathbf{eval}(v)$ the number of internal edges at v.



$L_\alpha = \{h_1, h_2\}$

$L_\beta = \{h_3\}$

$L_\gamma = \{h_4\}$

[2]Techincally we must discard any subsets which are now empty in order to obtain a partition.

1.2. **Cutkosky graphs.** As we said above Cutkosky graphs are Feynman graphs with a distinguished set of edges. To be more precise we first have to define cuts.

1.3. **Cuts.** Consider a bridgeless connected graph G. We have

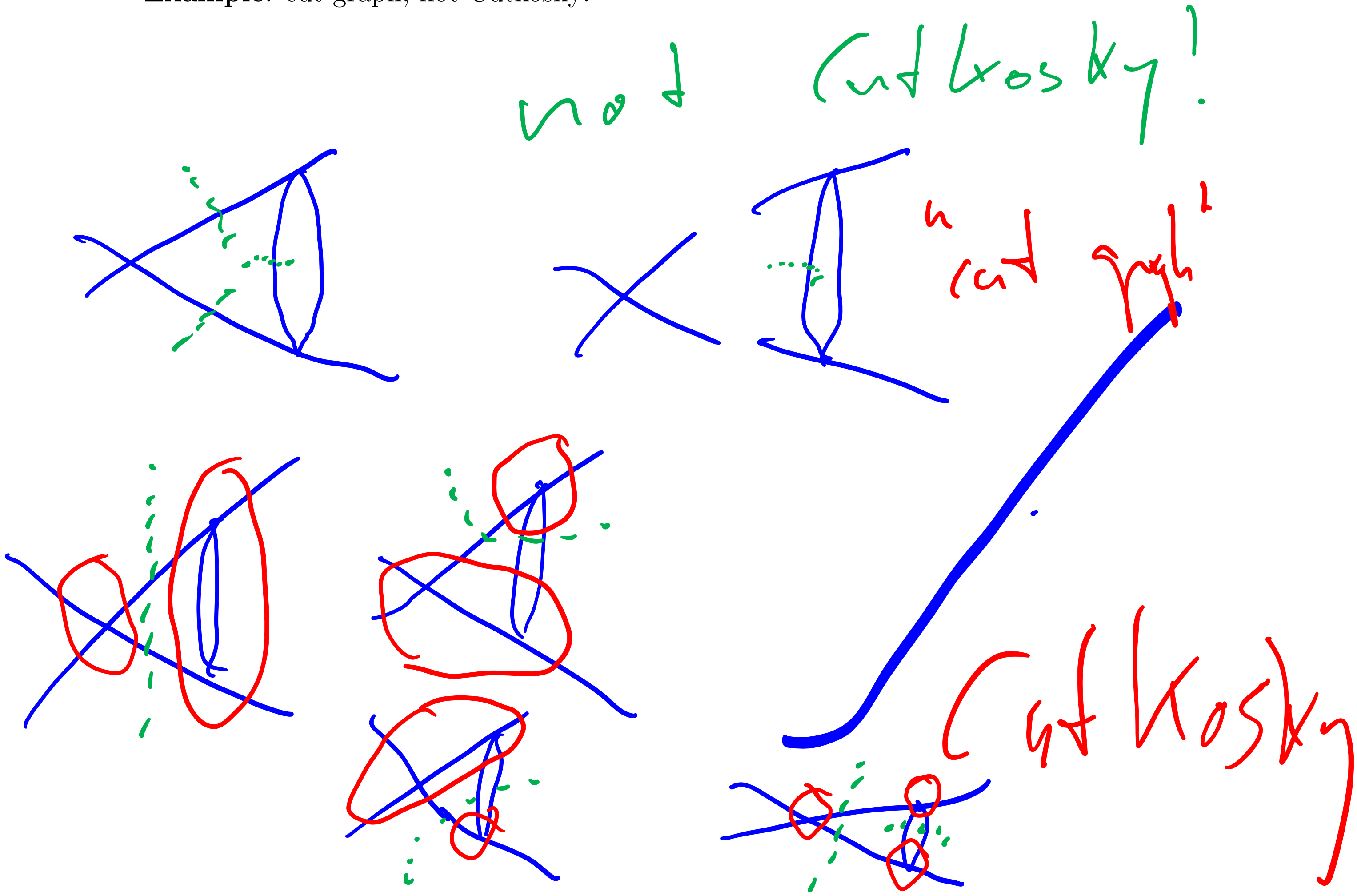
$$1 = h_0(G) = |G| - e_G + v_G.$$

If we want to cut G by removing edges, the Euler characteristic demands that we remove at least two edges.

From a physicist's viewpoint the cut edges can also be regarded as marked edges which are put on-shell when we apply Feynman rules.

We will introduce the vector space H_C generated by Cutkosky graphs, which are graphs which have cuts generated by a removal of edges. The base graph G is also allowed to vary.

Example: cut graph, not Cutkosky:



1.3.1. *Refinements.*

Definition 1.3. Given two partitions $\mathcal{P}$ and $\mathcal{P}'$ of a set S, we say $\mathcal{P}'$ is a *refinement* of $\mathcal{P}$ if every part of $\mathcal{P}'$ is a subset of a part of $\mathcal{P}$. Intuitively $\mathcal{P}'$ can be made from $\mathcal{P}$ by splitting some parts. The set of all partitions of S with the refinement relation gives a lattice called the *partition lattice*. The covering relation in this lattice is the special case of refinement where exactly one part of $\mathcal{P}$ is split into two parts to give $\mathcal{P}'$.

We will need more than just the refinements of partitions as defined above. Given a refinement $\mathcal{P}'$ of $\mathcal{P}$ it will often be useful that we additionally pick a maximal chain from $\mathcal{P}$ to $\mathcal{P}'$ in the partition lattice. Concretely this means we keep track of a way to build $\mathcal{P}'$ from $\mathcal{P}$ by a linear sequence of steps, each of which splits exactly one part into two. Unless otherwise specified our refinements always come with this sequence building them, and we will let a *j-refinement* be such a refinement where the sequence $\mathcal{P}(i), 0 \leq i \leq j$ of partitions has length j (including both ends). $\mathcal{P}(0) = S$ is the trivial partition.

We call a refinement maximal if it is a $|S| - 1$-refinement of a set S.

1 2, 3, 4, 5 $P(0)$

((1 2), (3 4 5))

((1 2), ((3 4), (5)))

(1 2), (3), (4), (5) $P(3)$

1.3.2. *Cuts.* Let us now consider cuts. Ultimately we will utilize cuts which decompose a graph G into a disjoint union

$$\dot{\cup}_{i=1}^{k} G_i,$$

of k graphs G_i which induce a k-partition of L_G.

Such a cut can be obtained by removing edges from the graph. This means that a cut can be obtained from refining $\mathcal{E}_G$.

Remark 1.4. *Following physics parlance when we refine $\mathcal{E}_G$, quite generally the first step giving two parts to L_G is called a normal cut.*

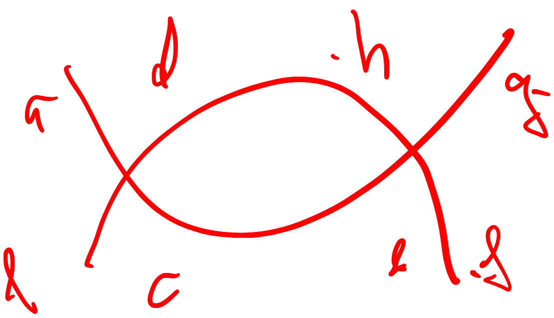
1.3.3. *cut graphs.*

G (G, H)

Definition 1.5. A cut graph G is a pair of graphs $((H_G, \mathcal{V}_G, \mathcal{E}_G), (H_G, \mathcal{V}_G, \mathcal{E}_H))$ on the same half-edges H_G such that $\mathcal{E}_H$ refines $\mathcal{E}_G$, along with a maximal chain giving the refinement for $\mathcal{E}_H$.

By abuse of notation the cut graph and the unrefined graph making it up have the same name (G in the above). This is because for physics applications we want to regard the cut graph as being the original G with the cut edges marked, so we view it as a decoration of G, or as G with extra structure added. Sometimes we write (G, H) as shorthand for the two graphs making up a cut graph.

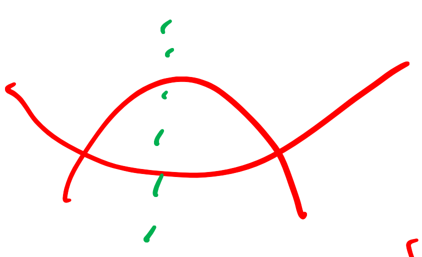
H_G

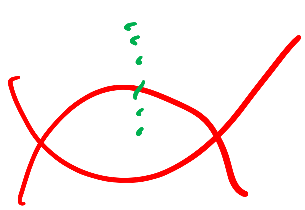


$\mathcal{E}_G = (a)(b)(dh)(ce)(fg)$

$\mathcal{E}_H = (a), \ldots (g)$

isolated half-edges





$\mathcal{E}_H$ . . . . $[c\cancel{h}]$ . . .

to be cut

In view of this, it will also be useful to have the notation $C_G \subset E_G$ for the edges which are cut, that is for those edges in E_G which are not edges in E_H.

Note that (G, G) is a cut graph as the trivial refinement is a refinement. Given a cut graph $G = (G, H)$, let $\hat{G} := (G, G)$ be the trivial cut graph built on G.

To a cut graph G we can assign more graphs:

Definition 1.6. To a cut graph $G = ((H_G, \mathcal{V}_G, \mathcal{E}_G), (H_G, \mathcal{V}_G, \mathcal{E}_H)$ we assign the amputated graph

$$\bar{G} = ((H_G - L_G, \bar{\mathcal{V}}_G, \bar{\mathcal{E}}_G), (H_G - L_G, \bar{\mathcal{V}}_G, \bar{\mathcal{E}}_H)$$

where for any partition $\mathcal{P}$ of H_G, $\bar{\mathcal{P}}$ is the partition whose parts are the parts of $\mathcal{P}$ intersected with $H_G - L_G$ (with empty parts removed).

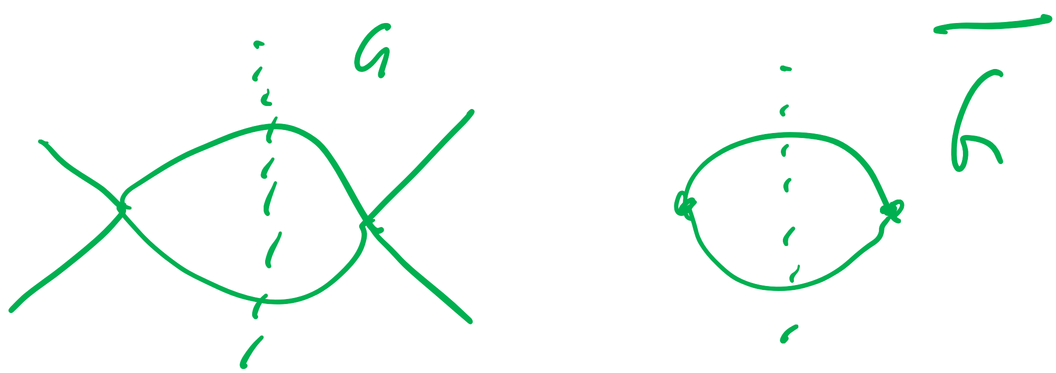
$\bar{G}$ is the pre-cut graph G with external edges removed.

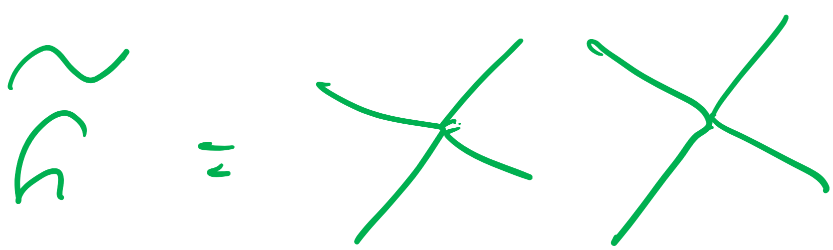
Furthermore:

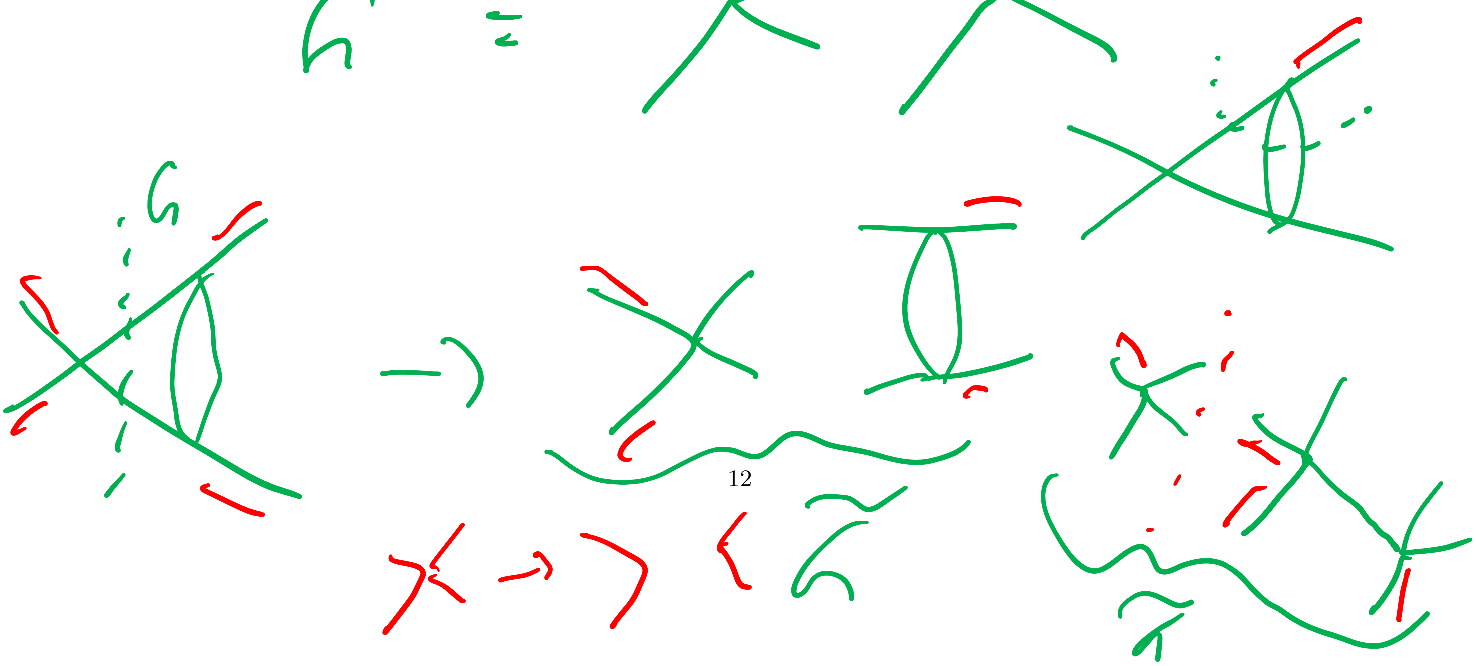
Definition 1.7. To a cut graph $G = ((H_G, \mathcal{V}_G, \mathcal{E}_G), (H_G, \mathcal{V}_G, \mathcal{E}_H)$ we also assign the associated graph

$$\tilde{G} = (H_G, \mathcal{V}_G, \mathcal{E}_H)$$

The graph $\tilde{G}$ associated to a cut graph is the graph with the cuts done; it is the more refined of the pair of graphs defining the pre-cut graph.







$h_0 = |H_0(\Gamma)|$

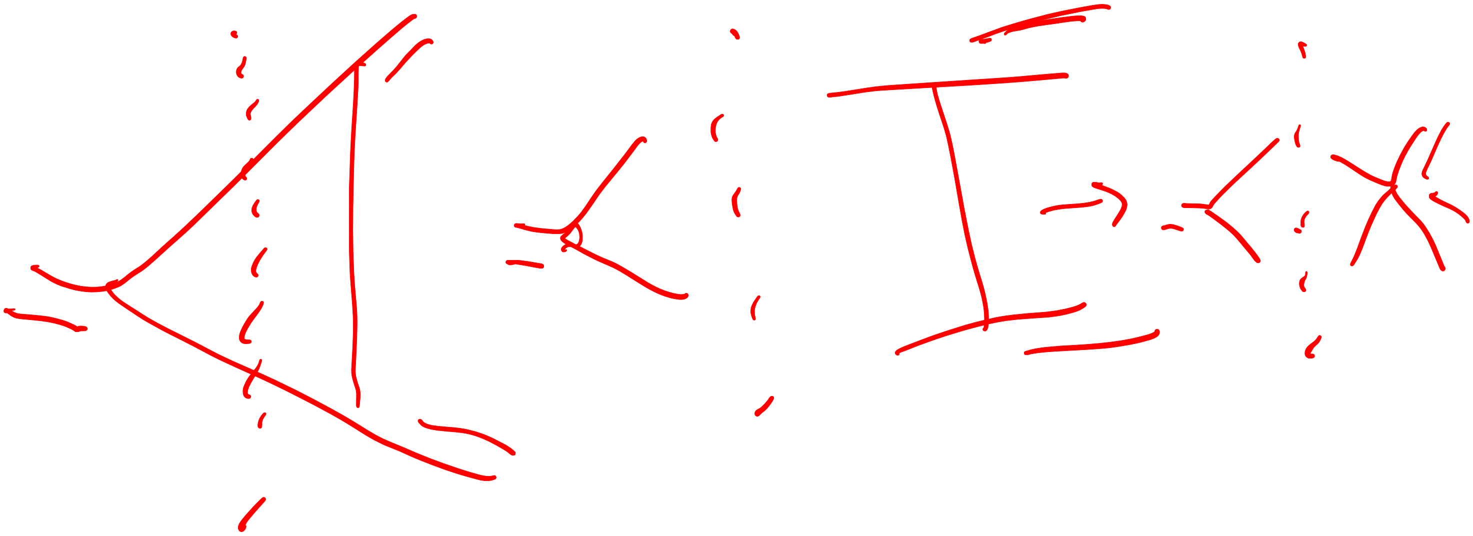
Definition 1.8. For a cut graph G we set $|G| := |\hat{G}|$ and $||G|| := |\tilde{G}|$.

By construction $|G| \equiv |\hat{G}|$.

There is a $h_0(\tilde{G})$-partition $L_G(h_0(\tilde{G}))$ of L_G. We have

$$L_G(h_0(\tilde{G})) = \tilde{G}/E_{\tilde{G}},$$

which is a $h_0(\tilde{G})$-partition of the corolla $\hat{G}/E_{\hat{G}}$.



Cutkosky!

We want to restrict the notion of cut graph to the notion of Cutkosky graph. For that we first need to discuss spanning forests.

1.4. **Spanning Forests and Cutkosky graphs.** To come to the notion of Cutkosky graphs we first have to discuss spanning trees and spanning forests. In particular a maximal chain of refinements of the set of external edges L_G of a graph G can then be identified with the removal of edges from a spanning tree in accordance with a chosen order of its edges.

1.4.1. *Spanning trees and forests.*

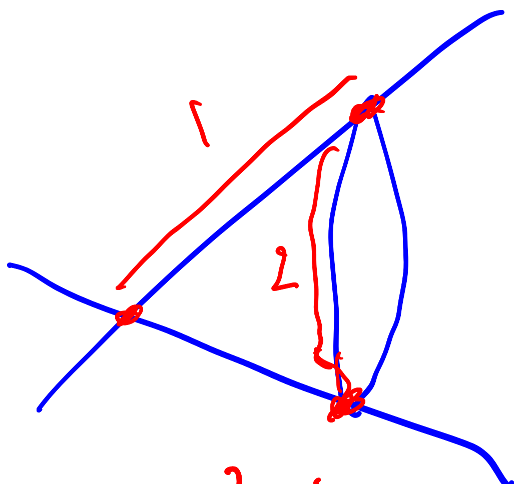
Definition 1.9. A spanning tree $T = (H_T, \mathcal{V}_T, \mathcal{E}_T)$ of a connected graph $G = (H_G, \mathcal{V}_G, \mathcal{E}_G)$ is a connected subgraph $T \subsetneq G$ such that $H_T \subsetneq H_G$, $H_T \cap L_G = \emptyset$, $V_T = V_G$, which has no cycles, i.e. is simply connected, $v_T - e_T = 1$.

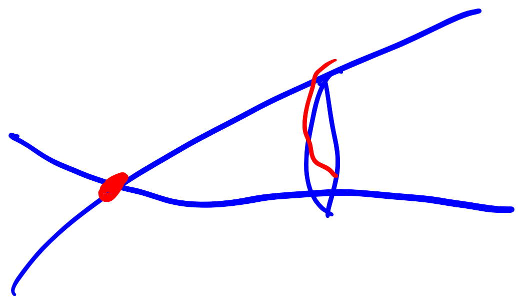
Definition 1.10. A spanning k-forest F is similarly a disjoint union $\dot\cup_{i=1}^{k} T_i$ of k trees $T_i \subsetneq G$, such that $\cup_i V_{T_i} = V_G$. Note $|G| = |G/F|$ for any spanning forest F of G.

E_F is the set of edges of F with cardinality $e_F = \sum_i e_{T_i}$. A spanning 1-forest is a spanning tree.

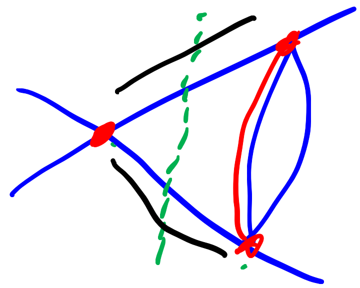
Equivalently, a spanning k-forest is a spanning tree from which $(k-1)$ edges are removed.

remove 1:

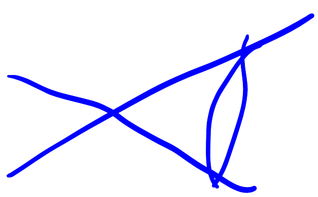




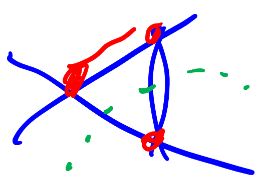
remove 2:

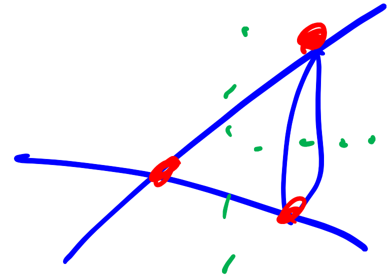


$\sim \cup E_G$



remove 1+2



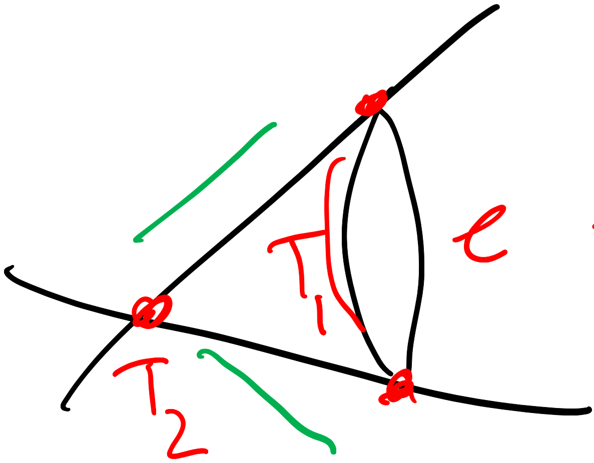


Given a spanning k-forest F of a cut graph G, there are a number of different sets of edges which will be important. First the edges $e \in E_F$ of the forest themselves are important.

Second are the edges $e \in C_G$ of G which are not in F but join distinct components of F. If we view F as a spanning tree T with some edges removed then all the edges of $T - F$ are in this second class, as well, typically, as others.

See above for C_G

Third are the edges $e \in (E_G - E_T)$ of G which are not in F but have both ends in the same tree of F.



$e \in F$

e ~~connect~~ edges in the same component T_1

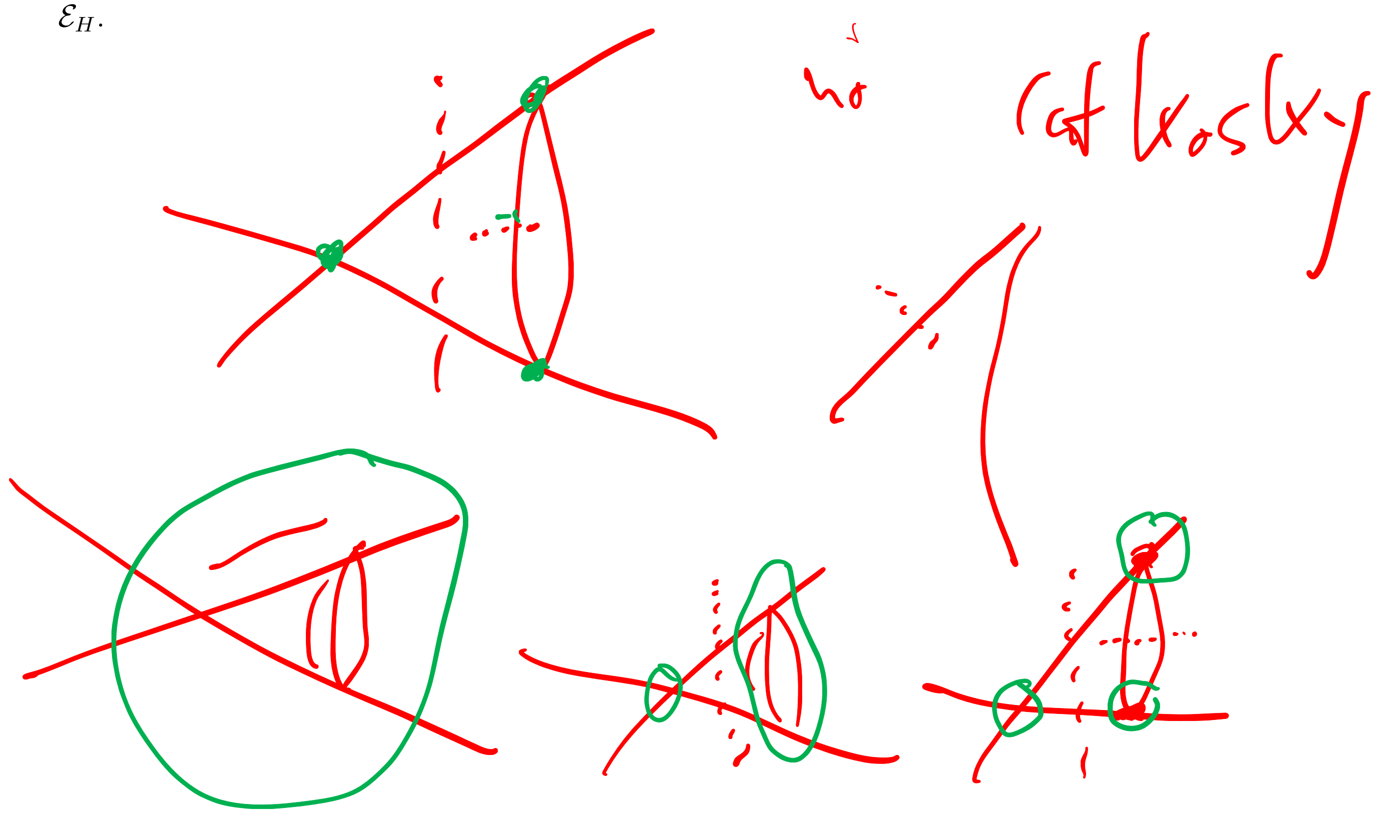
edges in E_F and edges not in E_F will be put on-shell and form the set E_{on}

Edges in E_F remain off-shell

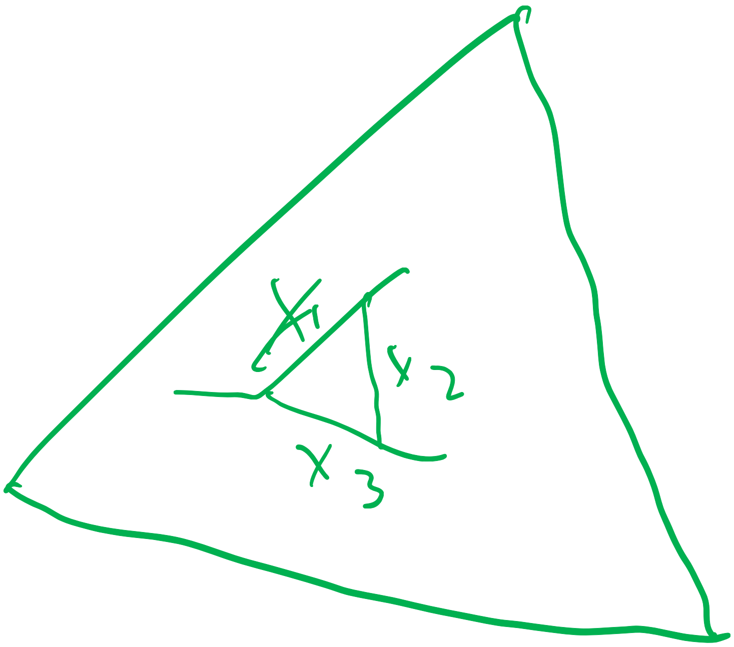
The second and third sets of edges above are those which will ultimately be put on-shell and define the set E_{on}, while those in the first set remain off-shell whilst we will use the notation $\breve{E}_F$ for the second of the above sets of edges.

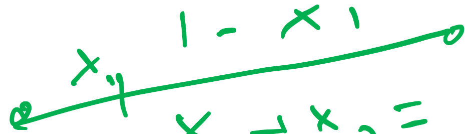
Definition 1.11. A Cutkosky graph G is a cut graph G for which a spanning forest F such that $C_G = \breve{E}_F$ exists.

Compatibility ensures that the spanning forest is in accordance with the chosen refinements $\mathcal{E}_H$.



Note $h_0(\tilde{G}) = h_0(F)$ for a compatible F and note that an ordering of edges in a spanning tree of a Cutkosky graph G induces a $h_0(v_G - 1)$-refinement of L_G.





$x_1 + x_2 = 1$

$x_1 + x_2 + x_3 = 1$

$\int \frac{1}{Q_1 Q_2 Q_3}$

$\int_\Delta \phi(x_1, x_2, x_3)$

We say that a spanning tree T of G with ordered edges is compatible with a given v_G-refinement R of L_G if and only if the forests $T - \amalg_{i=1}^{k} e_i$ induce the $k+1$-partition of R.

We let $\mathcal{F}_G^R$ be the set of ordered spanning trees of a graph G compatible with a v_G-refinement R of L_G . Note that if $R = L_G$ is the trivial partition, then $\mathcal{F}_G^R = \mathcal{T}_G$, the set of spanning trees of G.

We finish our lecture with:

Definition 1.12. We define H_C to be the $\mathbb{Q}$-vectorspace generated by Cutkosky graphs.

HUMBOLDT U. BERLIN