In[1]:= `<< AMBREv1.2.m`

```
by K.Kajda   ver: 1.2
last modified 9 Apr 2008
last executed on 19.12.2013 at 13:51
```

In[2]:= `<< PlanarityTest.m`

```
by E. Dubovyk and K. Bielas ver: 1.0
created: December 2013
last executed: 19.12.2013 at 13:51

(*
   File available at:http://www.us.edu.pl/~gluza/planarity
  "Some Remarks on Non-planar Feynman Diagrams",
K.Bielas,I.Dubovyk,J.Gluza,T.Riemann, Acta Phys.Polon.B44 (2013) 11,2249
   arXiv:1312.5603 [hep-ph]
*)
```

■ DESCRIPTION:

function PlanarityTest returns True if diagram is planar and False otherwise

■ OPTIONS :

PrintResult -> True/False - turns on/off printing of the text message “The Diagram is planar/non-planar” (default value - True).

TestAlgorithm -> 1 or 2 - allows to change testing algorithm (default value - 1); 1 - for Method I, 2 - for Method II (see below) .

DrawGraph -> True/False - turns on/off drawing of the input diagram (default value - False). Drawing function is based on Method I.

VertexLegNum -> 4 or 5 - allows to change the maximal degree of vertices for Method I (default value - 4).

■ RESTRICTIONS:

All momenta incoming.

Vertices with maximally 5 legs. This restriction is valid only for Method I. Due to performance reasons maximal degree of vertices is set up by default to 4, if graph has 5-legs vertices, corresponding message will appear.

Basic scalar integrals, for tensor integrals tensor structer should be stripped away first.

AMBRE should be loaded first.

■ REMARK:

Drawing option works well with *Mathematica* 8.0, a bug with *Mathematica* 9.0 appears when two vertices are connected via more than two lines
(reported to Wolfram).

■ EXAMPLES:

Method I is based on Laplacian and adjacency matrices and PlanarQ function (from *Combinatorica* package).

Method II is based on dual variables.

■ Example 1: 4 loop tadpole a)

Method I

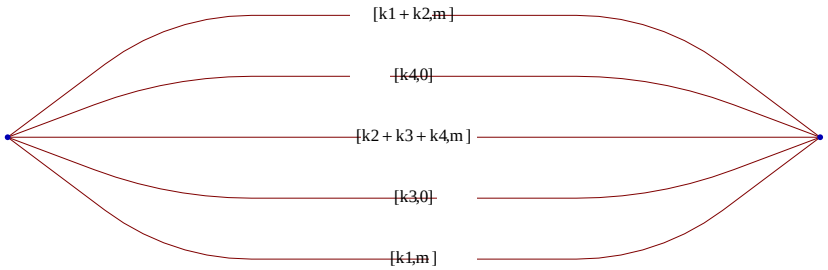
In[4]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2 + k1, m, n2] PR[k2 + k3 + k4, m, n3] PR[k4, 0, n4] PR[k3, 0, n5]},
  {k3, k4, k2, k1}, DrawGraph → True];
```

```
ERROR. input diagram cannot be analysed and/or drawn using Method I.
please try Method II without drawing option and/or with option VertexLegNum -> 5
```

In[5]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2 + k1, m, n2] PR[k2 + k3 + k4, m, n3] PR[k4, 0, n4] PR[k3, 0, n5]},
  {k3, k4, k2, k1}, DrawGraph → True, VertexLegNum → 5];
```

```
The Diagram

is planar.
```



Method II

In[6]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2 + k1, m, n2] PR[k2 + k3 + k4, m, n3] PR[k4, 0, n4] PR[k3, 0, n5]},
  {k3, k4, k2, k1}, TestAlgorithm → 2];
```

```
The Diagram

is planar.
```

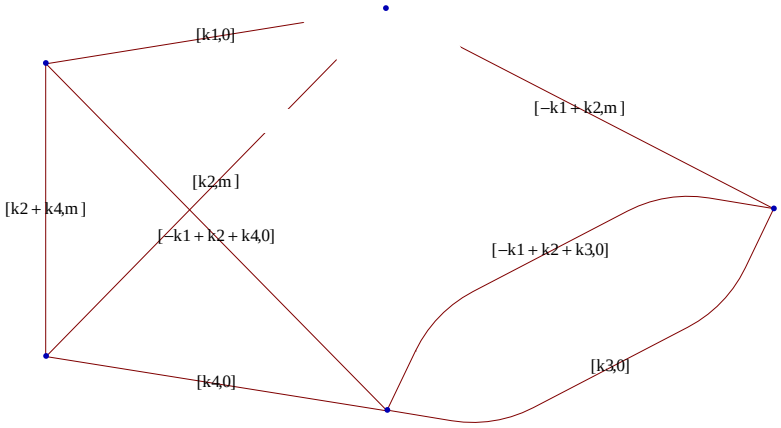
■ Example 2: 4 loop tadpole b)

Method I

In[7]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2 + k4, m, n2] PR[k2, m, n3] PR[k2 - k1, m, n4] PR[k3, 0, n5]
    PR[k2 - k1 + k3, 0, n6] PR[k4, 0, n7] PR[k2 - k1 + k4, 0, n8]}, {k1, k2, k3, k4}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[8]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2 + k4, m, n2] PR[k2, m, n3] PR[k2 - k1, m, n4] PR[k3, 0, n5]
    PR[k2 - k1 + k3, 0, n6] PR[k4, 0, n7] PR[k2 - k1 + k4, 0, n8]}, {k1, k2, k3, k4}, TestAlgorithm → 2];
```

The Diagram

is planar.

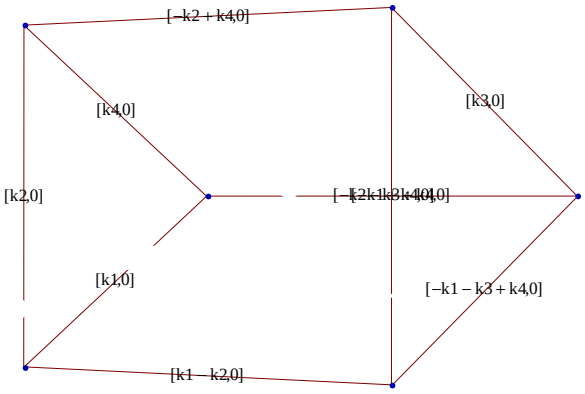
- **Example 3: 4 loop tadpole c)**

Method I

In[9]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k1 - k2, 0, n2] PR[k4 - k1, 0, n3] PR[k4 - k1 - k3, 0, n4] PR[k4 - k2 - k3, 0, n5]
    PR[k4 - k2, 0, n6] PR[k2, 0, n7] PR[k3, 0, n8] PR[k4, 0, n9]}, {k1, k2, k3, k4}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[10]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k1 - k2, 0, n2] PR[k4 - k1, 0, n3] PR[k4 - k1 - k3, 0, n4] PR[k4 - k2 - k3, 0, n5]
    PR[k4 - k2, 0, n6] PR[k2, 0, n7] PR[k3, 0, n8] PR[k4, 0, n9]}, {k1, k2, k3, k4}, TestAlgorithm → 2];
```

The Diagram

is planar.

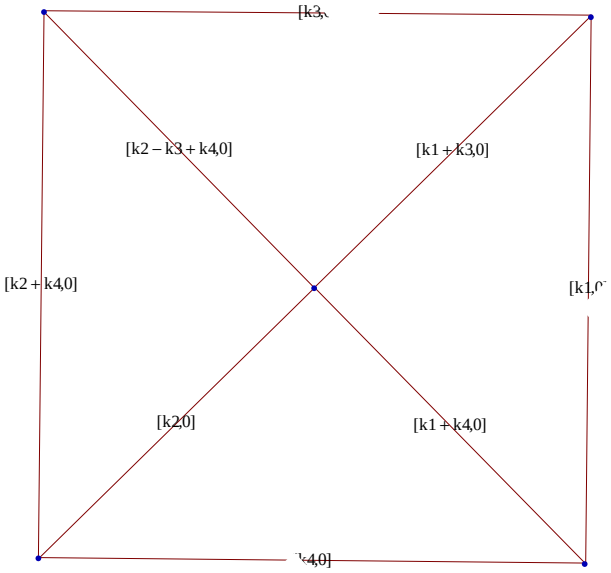
- **Example 4: 4 loop tadpole d)**

Method I

In[11]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k4, 0, n4] PR[k1 + k3, 0, n5]
    PR[k1 + k4, 0, n6] PR[k2 + k4, 0, n7] PR[k2 - k3 + k4, 0, n8]}, {k1, k2, k3, k4}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[12]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k4, 0, n4] PR[k1 + k3, 0, n5]
    PR[k1 + k4, 0, n6] PR[k2 + k4, 0, n7] PR[k2 - k3 + k4, 0, n8]}, {k1, k2, k3, k4}, TestAlgorithm → 2];
```

The Diagram

is planar.

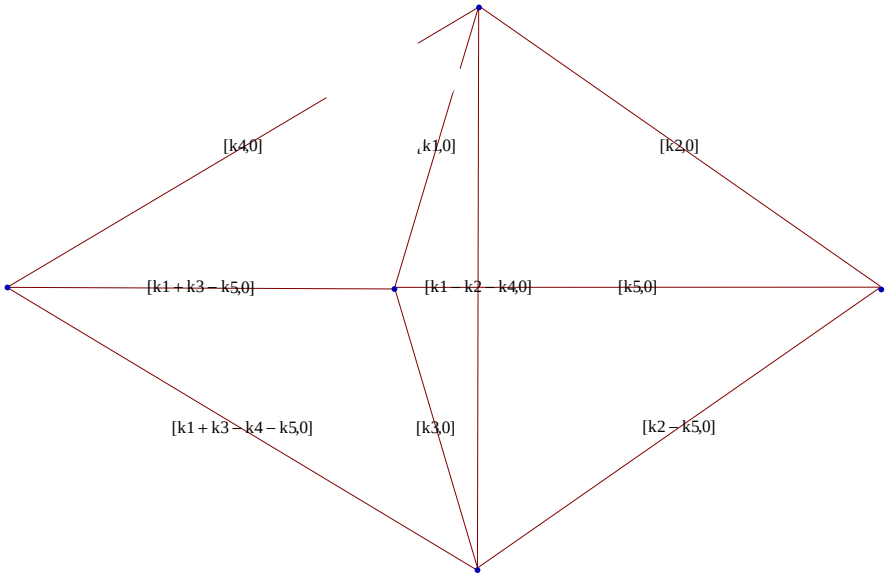
- **Example 5: 5 loop tadpole**

Method I

In[13]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k4, 0, n4] PR[k5, 0, n5] PR[k2 - k5, 0, n6] PR[k1 - k2 - k4, 0,
    n7] PR[k1 + k3 - k5, 0, n8] PR[k1 + k3 - k4 - k5, 0, n9]}, {k1, k2, k3, k4, k5}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[14]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k4, 0, n4] PR[k5, 0, n5] PR[k2 – k5, 0, n6] PR[k1 – k2 – k4, 0,
     n7] PR[k1 + k3 – k5, 0, n8] PR[k1 + k3 – k4 – k5, 0, n9]}, {k1, k2, k3, k4, k5}, TestAlgorithm → 2];
```

The Diagram

is planar.

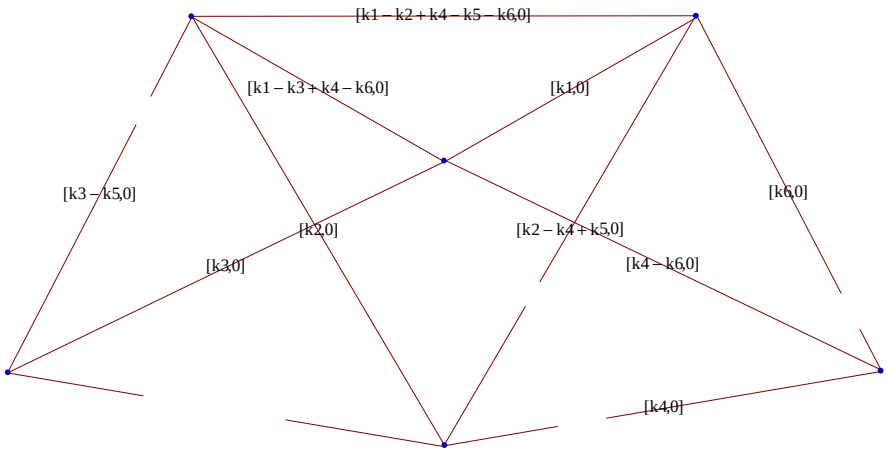
■ Example 6: 6 loop tadpole

Method I

In[15]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k4, 0, n4] PR[k5, 0, n5] PR[k6, 0, n6]
    PR[k3 – k5, 0, n7] PR[k4 – k6, 0, n8] PR[k2 + k5 – k4, 0, n9] PR[k1 – k3 + k4 – k6, 0, n10]
    PR[k1 – k2 + k4 – k5 – k6, 0, n11]}, {k1, k2, k3, k4, k5, k6}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[16]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k4, 0, n4] PR[k5, 0, n5] PR[k6, 0, n6]
    PR[k3 – k5, 0, n7] PR[k4 – k6, 0, n8] PR[k2 + k5 – k4, 0, n9] PR[k1 – k3 + k4 – k6, 0, n10]
    PR[k1 – k2 + k4 – k5 – k6, 0, n11]}, {k1, k2, k3, k4, k5, k6}, TestAlgorithm → 2];
```

The Diagram

is planar.

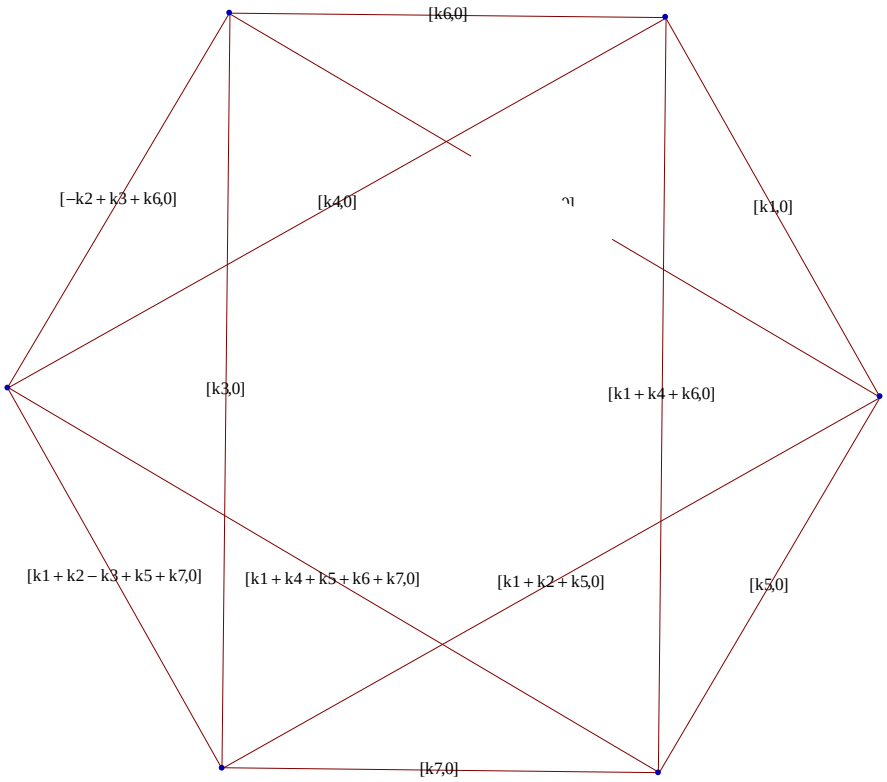
- **Example 7: 7 loop tadpole**

Method I

In[17]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k4, 0, n4]
    PR[k5, 0, n5] PR[k6, 0, n6] PR[k7, 0, n7] PR[k1 + k2 + k5, 0, n8] PR[k1 + k4 + k6, 0, n9]
    PR[k6 + k3 – k2, 0, n10] PR[k1 + k2 + k5 + k7 – k3, 0, n11] PR[k1 + k4 + k5 + k7 + k6, 0, n12]},
  {k1, k2, k3, k4, k5, k6, k7}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[17]= {0.228700, True}

Method II

In[18]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k4, 0, n4]
    PR[k5, 0, n5] PR[k6, 0, n6] PR[k7, 0, n7] PR[k1 + k2 + k5, 0, n8] PR[k1 + k4 + k6, 0, n9]
    PR[k6 + k3 − k2, 0, n10] PR[k1 + k2 + k5 + k7 − k3, 0, n11] PR[k1 + k4 + k5 + k7 + k6, 0, n12]},
  {k1, k2, k3, k4, k5, k6, k7}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is planar.

Out[18]= {5.484734, True}

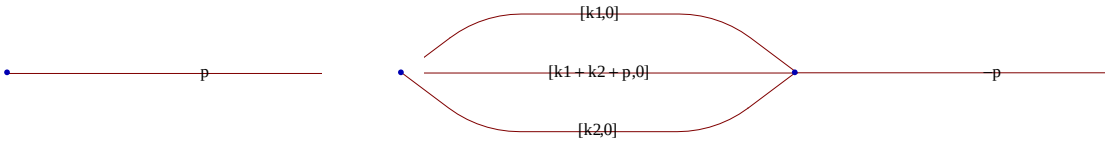
- **Example 8: 2 loop sunset diagram**

Method I

In[19]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + k2 + p, 0, n2] PR[k2, 0, n3]}, {k1, k2}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[20]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + k2 + p, 0, n2] PR[k2, 0, n3]}, {k1, k2}, TestAlgorithm → 2];
```

The Diagram

is planar.

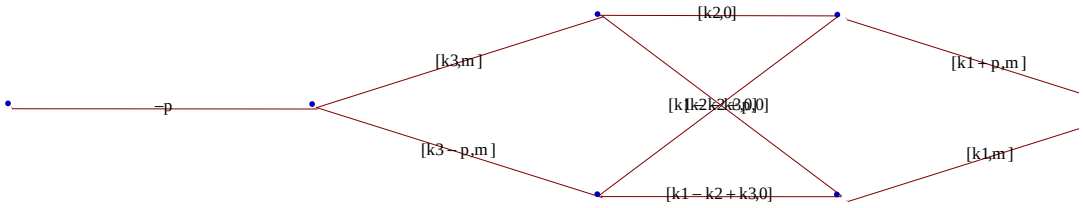
- **Example 9: 3 loop self self-energy diagram (non-planar)**

Method I

In[21]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2, 0, n2] PR[k3, m, n3] PR[k1 + p, m, n4] PR[k1 - k2 + p, 0, n5]
    PR[k3 - p, m, n6] PR[k1 - k2 + k3, 0, n7] PR[k2 - k3, 0, n8]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is non-planar.



Method II

In[22]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2, 0, n2] PR[k3, m, n3] PR[k1 + p, m, n4] PR[k1 - k2 + p, 0, n5]
    PR[k3 - p, m, n6] PR[k1 - k2 + k3, 0, n7] PR[k2 - k3, 0, n8]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is non-planar.

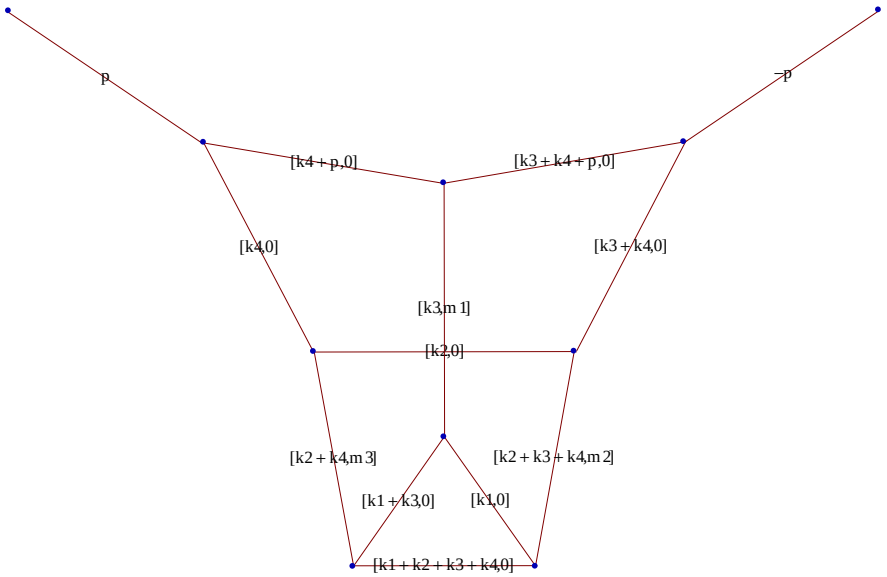
- **Example 10: four loop self self-energy diagram a)**

Method I

In[23]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + k3, 0, n2] PR[k1 + k2 + k3 + k4, 0, n3] PR[k3, m1, n4]
    PR[k2 + k4, m3, n5] PR[k2 + k3 + k4, m2, n6] PR[k4, 0, n7] PR[k4 + p, 0, n8] PR[k2, 0, n9]
    PR[k3 + k4, 0, n10] PR[k3 + k4 + p, 0, n11]}, {k1, k2, k3, k4}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[24]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + k3, 0, n2] PR[k1 + k2 + k3 + k4, 0, n3] PR[k3, m1, n4]
    PR[k2 + k4, m3, n5] PR[k2 + k3 + k4, m2, n6] PR[k4, 0, n7] PR[k4 + p, 0, n8] PR[k2, 0, n9]
    PR[k3 + k4, 0, n10] PR[k3 + k4 + p, 0, n11]}, {k1, k2, k3, k4}, TestAlgorithm → 2];
```

The Diagram

is planar.

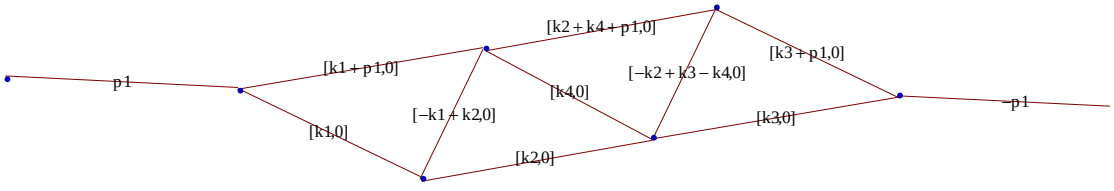
- **Example 11: four loop self self-energy diagram b)**

Method I

In[25]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k1 + p1, 0, n2] PR[k2 − k1, 0, n3] PR[k2, 0, n4] PR[k4, 0, n5] PR[k2 + k4 + p1, 0, n6]
    PR[k3, 0, n7] PR[k3 − k2 − k4, 0, n8] PR[k3 + p1, 0, n9]}, {k1, k2, k3, k4}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[26]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k1 + p1, 0, n2] PR[k2 - k1, 0, n3] PR[k2, 0, n4] PR[k4, 0, n5] PR[k2 + k4 + p1, 0, n6]
    PR[k3, 0, n7] PR[k3 - k2 - k4, 0, n8] PR[k3 + p1, 0, n9]}, {k1, k2, k3, k4}, TestAlgorithm → 2];
```

The Diagram

is planar.

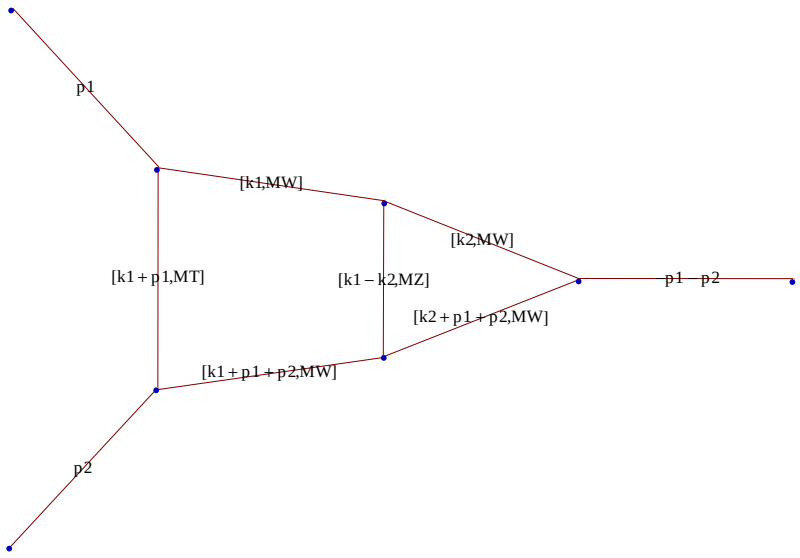
- **Example 12: 2 loop vertex + one loop tadpoles**

Method I

In[27]:=
```
PlanarityTest[{PR[k1, MW, n1] PR[k1 - k2, MZ, n2] PR[k2, MW, n3] PR[k1 + p1, MT, n4] PR[k1 + p1 + p2, MW, n5]
    PR[k2 + p1 + p2, MW, n6] PR[k3 + p1, 0, n7] PR[k4, 0, n8]}, {k1, k2, k3, k4}, DrawGraph → True];
```

The Diagram

is planar.



With tadpoles: PR[k4,0]PR[k3 + p1,0]

Method II

In[28]:=
```
PlanarityTest[{PR[k1, MW, n1] PR[k1 - k2, MZ, n2] PR[k2, MW, n3] PR[k1 + p1, MT, n4] PR[k1 + p1 + p2, MW, n5]
    PR[k2 + p1 + p2, MW, n6] PR[k3 + p1, 0, n7] PR[k4, 0, n8]}, {k1, k2, k3, k4}, TestAlgorithm → 2];
```

The Diagram

is planar.

In[29]:=
```
PlanarityTest[{PR[k1, MW, n1] PR[k1 - k2, MZ, n2] PR[k2, MW, n3] PR[k1 + p1, MT, n4] PR[k1 + p1 + p2, MW, n5]
    PR[k2 + p1 + p2, MW, n6] PR[k3 + p1, 0, n7] PR[k4, 0, n8]}, {k1, k2, k3, k4}, TestAlgorithm → 1];
```

The Diagram

is planar.

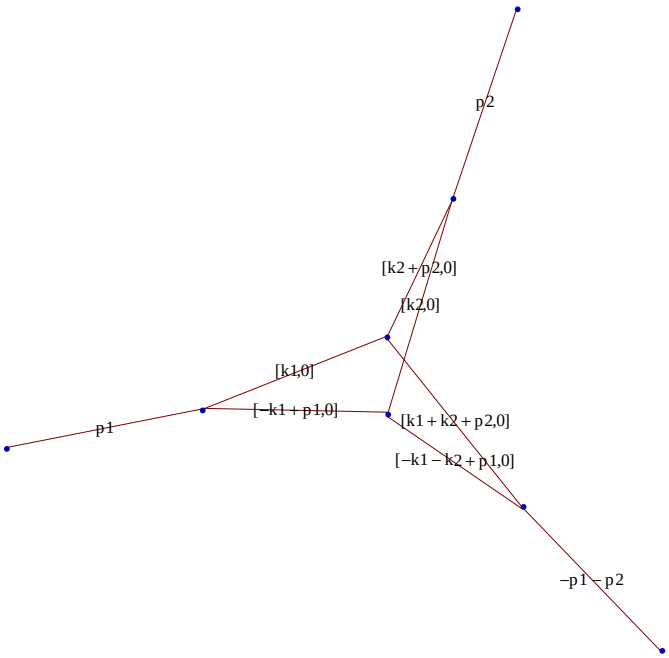
- **Example 13: 2 loop vertex (non-planar)**

Method I

In[30]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[p1 - k1, 0, n2] PR[p1 - k1 - k2, 0, n3]
    PR[p2 + k1 + k2, 0, n4] PR[p2 + k2, 0, n5] PR[k2, 0, n6]}, {k1, k2}, DrawGraph → True];
```

The Diagram

is non-planar.



Method II

In[31]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[p1 - k1, 0, n2] PR[p1 - k1 - k2, 0, n3]
    PR[p2 + k1 + k2, 0, n4] PR[p2 + k2, 0, n5] PR[k2, 0, n6]}, {k1, k2}, TestAlgorithm → 2];
```

The Diagram

is non-planar.

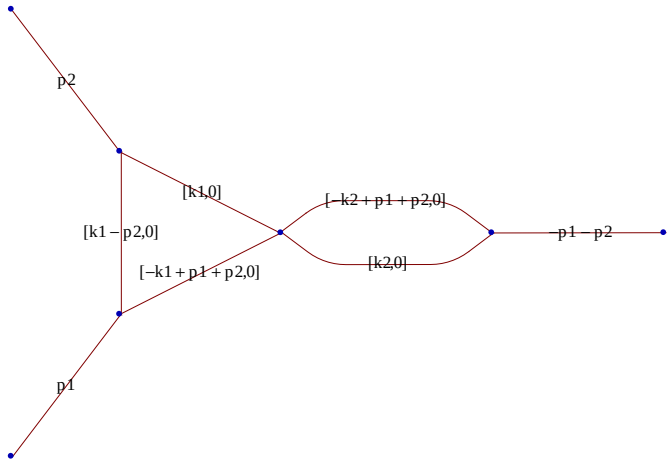
- **Example 14: 2 loop vertex**

Method I

In[32]:=
```
PlanarityTest[{PR[k1, 0 W, n1] PR[k2, 0, n2] PR[k1 - p2, 0, n3] PR[p1 + p2 - k1, 0, n4] PR[p1 + p2 - k2, 0, n5]},
  {k1, k2}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[33]:=
```
PlanarityTest[{PR[k1, 0 W, n1] PR[k2, 0, n2] PR[k1 – p2, 0, n3] PR[p1 + p2 – k1, 0, n4] PR[p1 + p2 – k2, 0, n5]},
  {k1, k2}, TestAlgorithm → 2];
```

The Diagram

is planar.

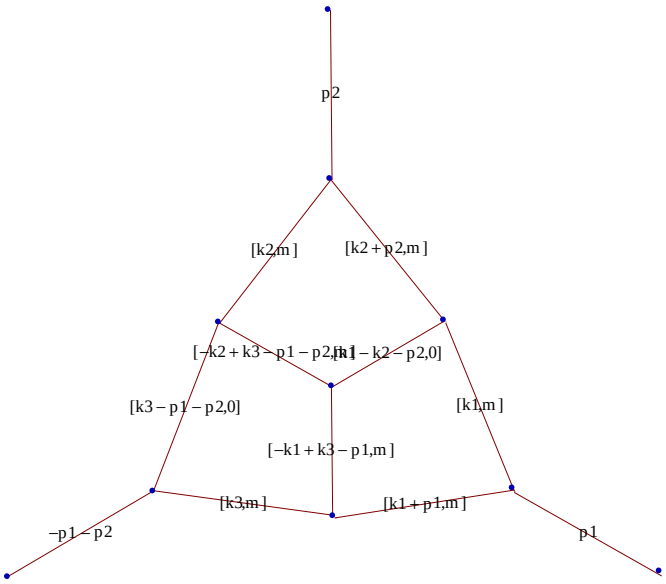
- **Example 15: 3 loop vertex a)**

Method I

In[34]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2, m, n2] PR[k3, m, n3]
    PR[k2 + p2, m, n4] PR[k1 – k2 – p2, 0, n5] PR[k1 + p1, m, n6] PR[k3 – k1 – p1, m, n7]
    PR[k3 – p1 – p2, 0, n8] PR[k3 – k2 – p1 – p2, m, n9]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[35]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2, m, n2] PR[k3, m, n3]
    PR[k2 + p2, m, n4] PR[k1 − k2 − p2, 0, n5] PR[k1 + p1, m, n6] PR[k3 − k1 − p1, m, n7]
    PR[k3 − p1 − p2, 0, n8] PR[k3 − k2 − p1 − p2, m, n9]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is planar.

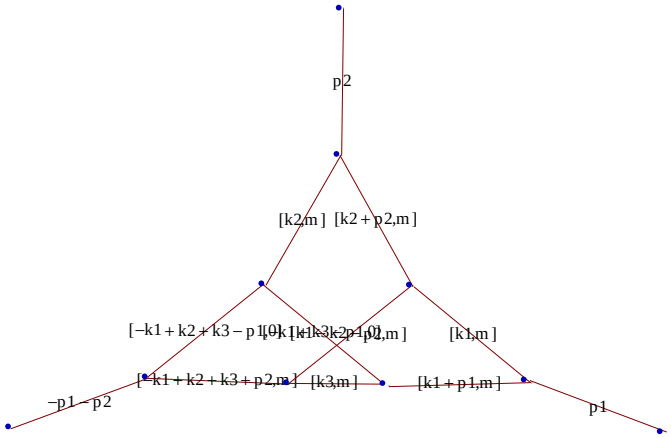
- **Example 16: 3 loop vertex b)**

Method I

In[36]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2, m, n2] PR[k3, m, n3]
    PR[k2 + p2, m, n4] PR[k1 + p1, m, n5] PR[k3 − k1 − p1, 0, n6] PR[k1 − k2 − p2, m, n7]
    PR[k3 + k2 − k1 − p1, 0, n8] PR[k3 + k2 − k1 + p2, m, n9]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is non-planar.



Method II

In[37]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2, m, n2] PR[k3, m, n3]
    PR[k2 + p2, m, n4] PR[k1 + p1, m, n5] PR[k3 − k1 − p1, 0, n6] PR[k1 − k2 − p2, m, n7]
    PR[k3 + k2 − k1 − p1, 0, n8] PR[k3 + k2 − k1 + p2, m, n9]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is non-planar.

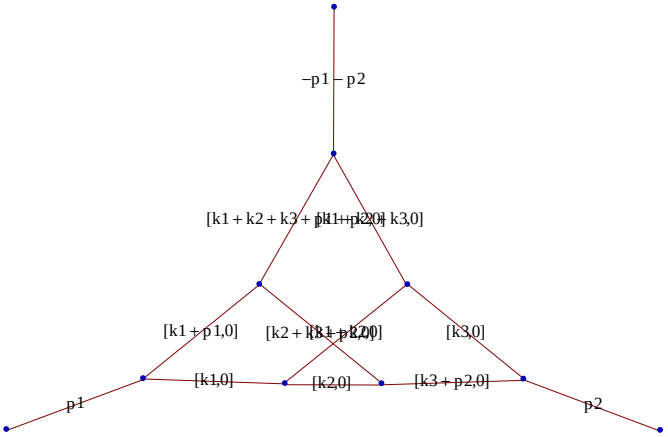
- **Example 17: 3 loop vertex c)**

Method I

In[38]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + p1, 0, n2] PR[k3 + p2, 0, n3]
    PR[k2 + k3 + p2, 0, n4] PR[k1 + k2, 0, n5] PR[k1 + k2 + k3 + p1 + p2, 0, n6]
    PR[k1 + k2 + k3, 0, n7] PR[k2, 0, n8] PR[k3, 0, n9]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is non-planar.



Method II

In[39]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + p1, 0, n2] PR[k3 + p2, 0, n3]
    PR[k2 + k3 + p2, 0, n4] PR[k1 + k2, 0, n5] PR[k1 + k2 + k3 + p1 + p2, 0, n6]
    PR[k1 + k2 + k3, 0, n7] PR[k2, 0, n8] PR[k3, 0, n9]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is non-planar.

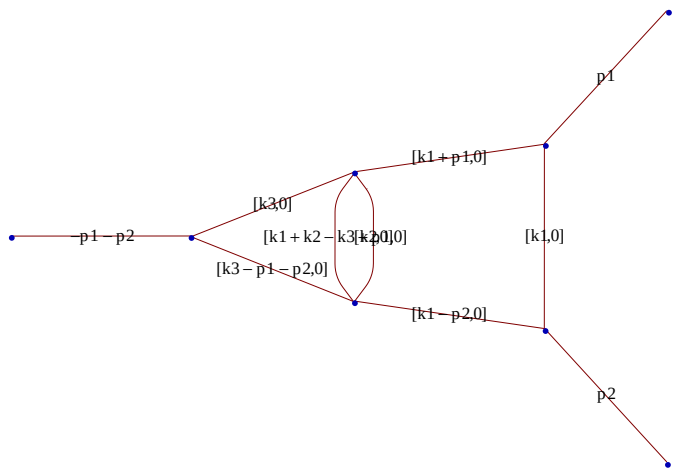
- **Example 18: 3 loop vertex d)**

Method I

In[40]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k1 - p2, 0, n4] PR[k1 + p1, 0, n5]
    PR[k1 + k2 - k3 + p1, 0, n6] PR[k3 - p1 - p2, 0, n7]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[41]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k1 – p2, 0, n4] PR[k1 + p1, 0, n5]
    PR[k1 + k2 – k3 + p1, 0, n6] PR[k3 – p1 – p2, 0, n7]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is planar.

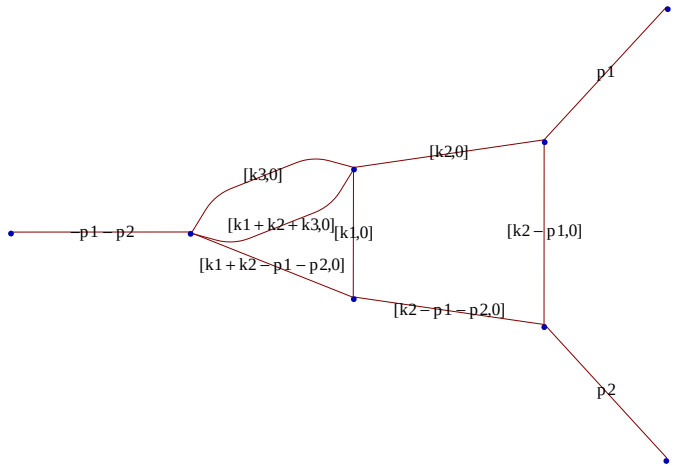
- **Example 19: 3 loop vertex e)**

Method I

In[42]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k2 – p1, 0, n4] PR[k2 – p1 – p2, 0, n5]
    PR[k1 + k2 + k3, 0, n6] PR[k1 + k2 – p1 – p2, 0, n7]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[43]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k2 – p1, 0, n4] PR[k2 – p1 – p2, 0, n5]
    PR[k1 + k2 + k3, 0, n6] PR[k1 + k2 – p1 – p2, 0, n7]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is planar.

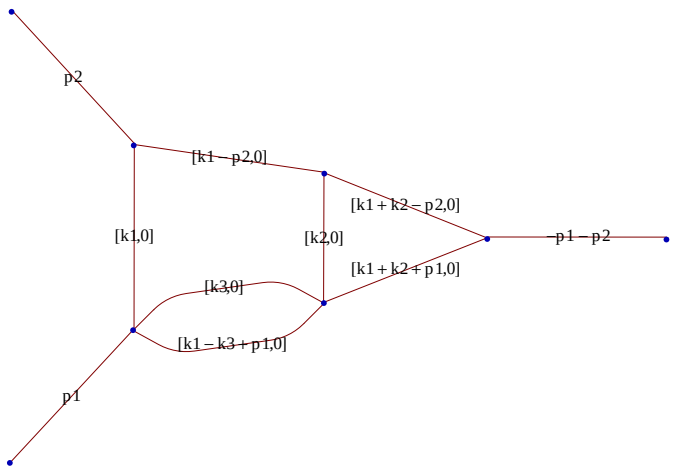
- **Example 20: 3 loop vertex f)**

Method I

In[44]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k1 – p2, 0, n4] PR[k1 + p1 – k3, 0, n5]
    PR[k1 + k2 + p1, 0, n6] PR[k1 + k2 – p2, 0, n7]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[45]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k1 − p2, 0, n4] PR[k1 + p1 − k3, 0, n5]
    PR[k1 + k2 + p1, 0, n6] PR[k1 + k2 − p2, 0, n7]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is planar.

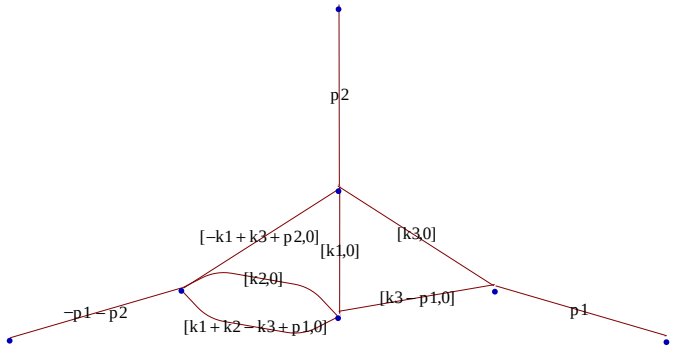
- **Example 21: 3 loop vertex g)**

Method I

In[46]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k3 − p1, 0, n4]
    PR[k3 − k1 + p2, 0, n5] PR[k1 + k2 − k3 + p1, 0, n6]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[47]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k3 − p1, 0, n4]
    PR[k3 − k1 + p2, 0, n5] PR[k1 + k2 − k3 + p1, 0, n6]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is planar.

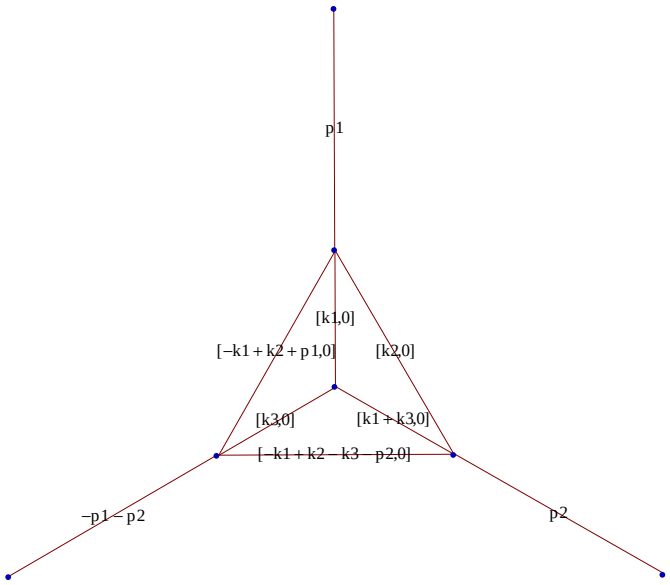
- **Example 22: 3 loop vertex h)**

Method I

In[48]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k1 + k3, 0, n4]
    PR[k2 - k1 + p1, 0, n5] PR[k2 - k1 - k3 - p2, 0, n6]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[49]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k1 + k3, 0, n4]
    PR[k2 - k1 + p1, 0, n5] PR[k2 - k1 - k3 - p2, 0, n6]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is planar.

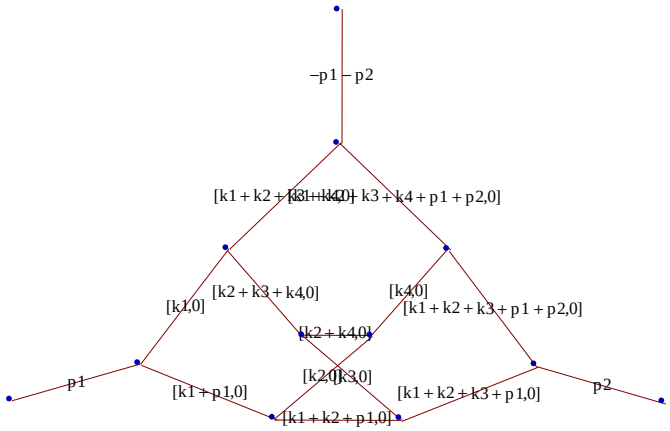
- **Example 23: 4 loop vertex**

Method I

In[50]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + p1, 0, n2] PR[k1 + k2 + p1, 0, n3]
    PR[k1 + k2 + k3 + p1, 0, n4] PR[k1 + k2 + k3 + p1 + p2, 0, n5] PR[k1 + k2 + k3 + k4 + p1 + p2, 0, n6]
    PR[k1 + k2 + k3 + k4, 0, n7] PR[k2 + k3 + k4, 0, n8] PR[k2 + k4, 0, n9] PR[k2, 0, n10]
    PR[k3, 0, n11] PR[k4, 0, n12]}, {k1, k2, k3, k4}, DrawGraph → True];
```

The Diagram

is non-planar.



Method II

In[51]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + p1, 0, n2] PR[k1 + k2 + p1, 0, n3]
  PR[k1 + k2 + k3 + p1, 0, n4] PR[k1 + k2 + k3 + p1 + p2, 0, n5] PR[k1 + k2 + k3 + k4 + p1 + p2, 0, n6]
  PR[k1 + k2 + k3 + k4, 0, n7] PR[k2 + k3 + k4, 0, n8] PR[k2 + k4, 0, n9] PR[k2, 0, n10]
  PR[k3, 0, n11] PR[k4, 0, n12]}, {k1, k2, k3, k4}, TestAlgorithm → 2];
```

The Diagram

is non-planar.

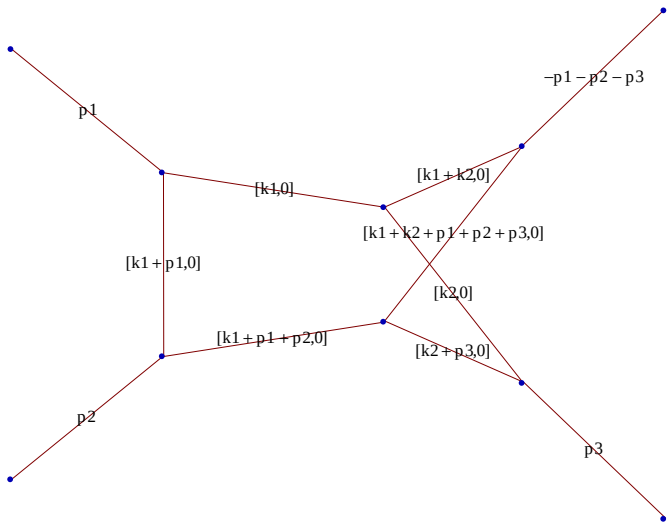
- **Example 24: double box (non-planar)**

Method I

In[52]:=
```
PlanarityTest[{PR[k1 + k2 + p1 + p2 + p3, 0, n1] PR[k1 + k2, 0, n2] PR[k1, 0, n3] PR[k1 + p1, 0, n4]
  PR[k1 + p1 + p2, 0, n5] PR[k2 + p3, 0, n6] PR[k2, 0, n7]}, {k1, k2}, DrawGraph → True];
```

The Diagram

is non-planar.



Method II

In[53]:=
```
PlanarityTest[{PR[k1 + k2 + p1 + p2 + p3, 0, n1] PR[k1 + k2, 0, n2] PR[k1, 0, n3] PR[k1 + p1, 0, n4]
    PR[k1 + p1 + p2, 0, n5] PR[k2 + p3, 0, n6] PR[k2, 0, n7]}, {k1, k2}, TestAlgorithm → 2];
```

The Diagram

is non-planar.

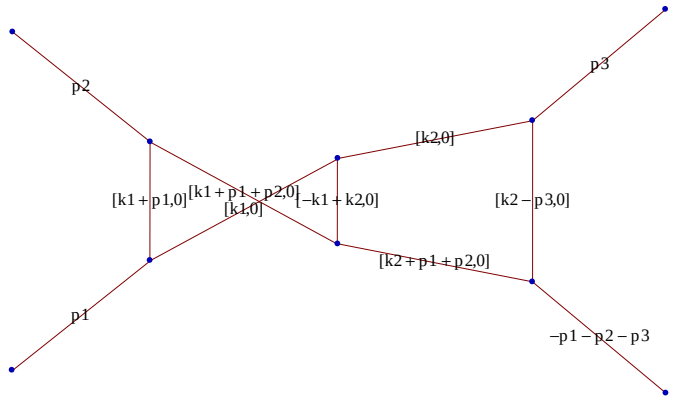
- **Example 25: double box**

Method I

In[54]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + p1, 0, n2] PR[k1 + p1 + p2, 0, n3] PR[-k1 + k2, 0, n4]
    PR[k2, 0, n5] PR[k2 + p1 + p2, 0, n6] PR[k2 - p3, 0, n7]}, {k1, k2}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[55]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 + p1, 0, n2] PR[k1 + p1 + p2, 0, n3] PR[-k1 + k2, 0, n4]
    PR[k2, 0, n5] PR[k2 + p1 + p2, 0, n6] PR[k2 - p3, 0, n7]}, {k1, k2}, TestAlgorithm → 2];
```

The Diagram

is planar.

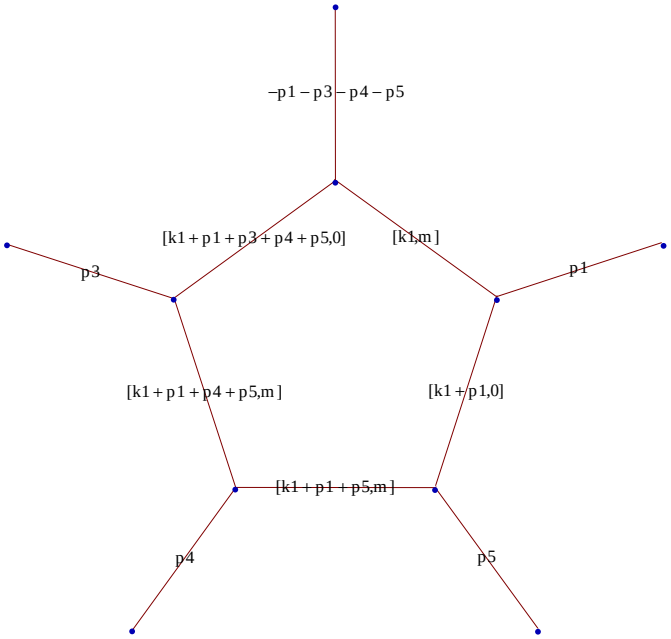
- **Example 26: pentagon**

Method I

In[56]:=
```
PlanarityTest[{PR[p1 + k1, 0, n1] PR[p1 + p5 + k1, m, n2] PR[p1 + p5 + p4 + k1, m, n3]
    PR[p1 + p5 + p4 + p3 + k1, 0, n4] PR[k1, m, n5]}, {k1}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[57]:=
```
PlanarityTest[{PR[p1 + k1, 0, n1] PR[p1 + p5 + k1, m, n2] PR[p1 + p5 + p4 + k1, m, n3]
    PR[p1 + p5 + p4 + p3 + k1, 0, n4] PR[k1, m, n5]}, {k1}, TestAlgorithm → 2];
```

The Diagram

is planar.

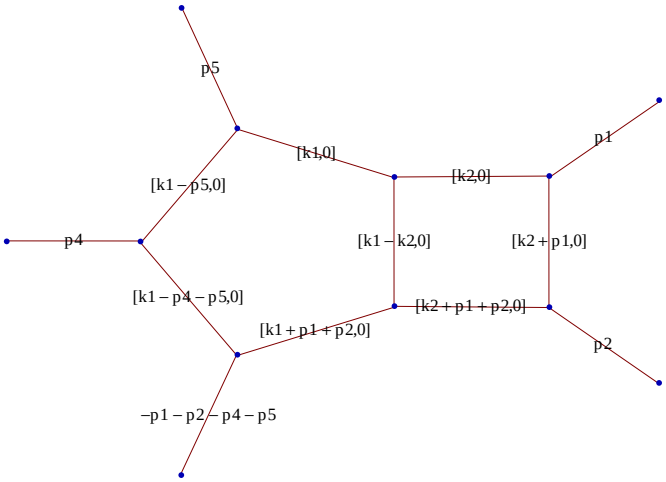
- **Example 27: 2 loop pentabox**

Method I

In[58]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 − k2, 0, n2] PR[k1 + p1 + p2, 0, n3] PR[k1 − p4 − p5, 0, n4]
    PR[k1 − p5, 0, n5] PR[k2, 0, n6] PR[k2 + p1, 0, n7] PR[k2 + p1 + p2, 0, n8]}, {k2, k1}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[59]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1 – k2, 0, n2] PR[k1 + p1 + p2, 0, n3] PR[k1 – p4 – p5, 0, n4]
    PR[k1 – p5, 0, n5] PR[k2, 0, n6] PR[k2 + p1, 0, n7] PR[k2 + p1 + p2, 0, n8]}, {k2, k1}, TestAlgorithm → 2];
```

The Diagram

is planar.

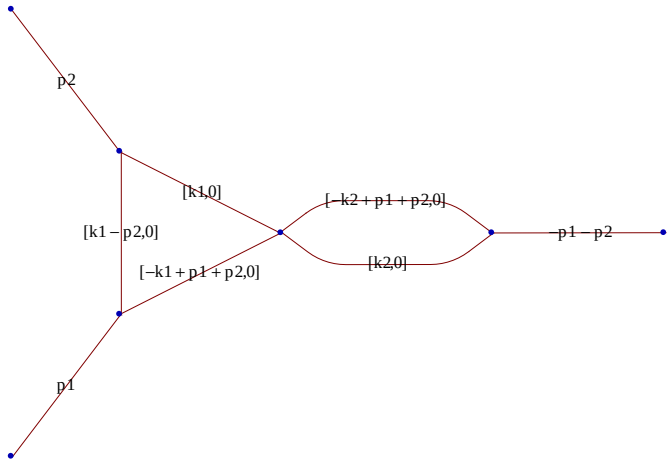
- **Example 28: product of diagrams**

Method I

In[60]:=
```
PlanarityTest[{PR[k1, 0 W, n1] PR[k2, 0, n2] PR[k1 – p2, 0, n3] PR[p1 + p2 – k1, 0, n4] PR[p1 + p2 – k2, 0, n5]},
  {k1, k2}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[61]:=
```
PlanarityTest[{PR[k1, 0 W, n1] PR[k2, 0, n2] PR[k1 – p2, 0, n3] PR[p1 + p2 – k1, 0, n4] PR[p1 + p2 – k2, 0, n5]},
  {k1, k2}, TestAlgorithm → 2];
```

The Diagram

is planar.

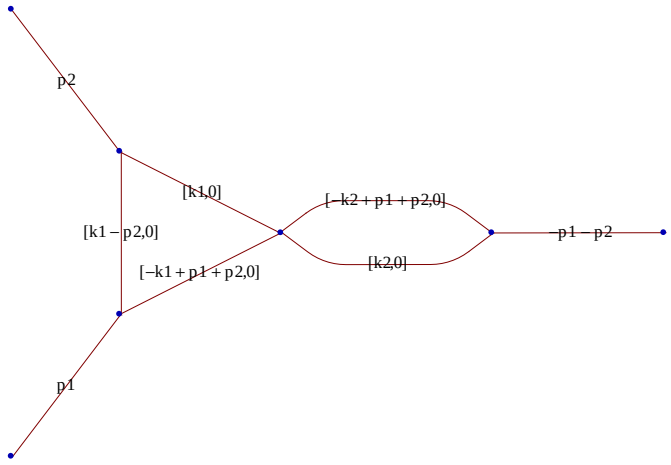
■ **Example 29: product of diagrams**

Method I

In[62]:=
```
PlanarityTest[{PR[k1, 0 W, n1] PR[k2, 0, n2] PR[k1 – p2, 0, n3] PR[p1 + p2 – k1, 0, n4] PR[p1 + p2 – k2, 0, n5]},
  {k1, k2}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[63]:=
```
PlanarityTest[{PR[k1, 0 W, n1] PR[k2, 0, n2] PR[k1 − p2, 0, n3] PR[p1 + p2 − k1, 0, n4] PR[p1 + p2 − k2, 0, n5]},
  {k1, k2}, TestAlgorithm → 2];
```

The Diagram

is planar.

- **Example 30: one loop vertex with 2 loop tadpole attached**

Method I

In[64]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k3 − p2, 0, n4] PR[k1 + k2, 0, n5] PR[k3 + p1, 0, n6]},
  {k1, k2, k3}, DrawGraph → True];
```

```
ERROR. input diagram cannot be analysed and/or drawn using Method I.
please try Method II without drawing option and/or with option VertexLegNum -> 5
```

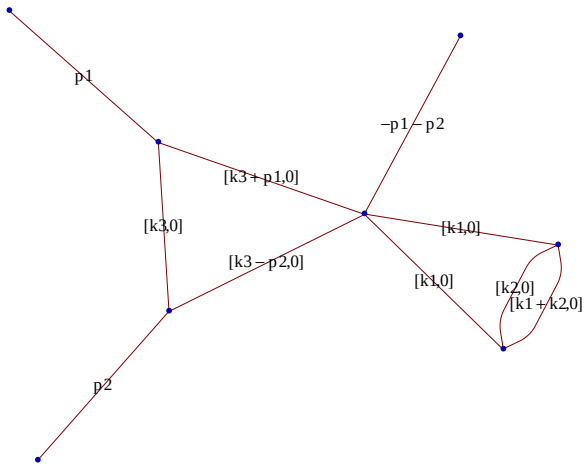
In[65]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k3 − p2, 0, n4] PR[k1 + k2, 0, n5] PR[k3 + p1, 0, n6]},
  {k1, k2, k3}, DrawGraph → True, VertexLegNum → 5];
```

```
ERROR. input diagram cannot be analysed and/or drawn using Method I.
please try Method II without drawing option and/or with option VertexLegNum -> 5
```

In[66]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1, 0, n11] PR[k2, 0, n2] PR[k3, 0, n3] PR[k3 − p2, 0, n4]
    PR[k1 + k2, 0, n5] PR[k3 + p1, 0, n6]}, {k1, k2, k3}, DrawGraph → True, VertexLegNum → 5];
```

The Diagram

is planar.



Method II

In[67]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k2, 0, n2] PR[k3, 0, n3] PR[k3 − p2, 0, n4] PR[k1 + k2, 0, n5] PR[k3 + p1, 0, n6]},
  {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is planar.

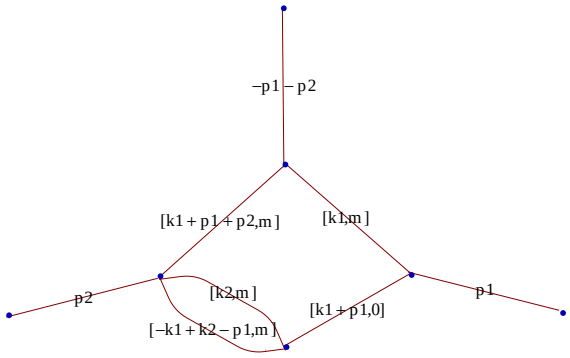
- **Example 31: 2 loop vertex with a bubble**

Method I

In[68]:=
```
PlanarityTest[{PR[k1, m, n1] PR[p1 + k1, 0, n2] PR[k2, m, n3] PR[k2 − k1 − p1, m, n4] PR[k1 + p1 + p2, m, n5]},
  {k1, k2}, DrawGraph → True];
```

The Diagram

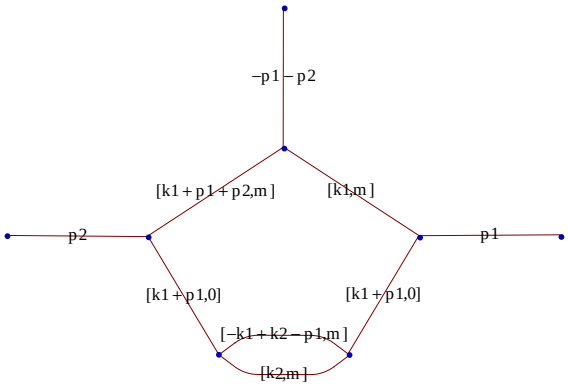
is planar.



In[69]:=
```
PlanarityTest[{PR[k1, m, n1] PR[p1 + k1, 0, n2] PR[p1 + k1, 0, n21]
    PR[k2, m, n3] PR[k2 − k1 − p1, m, n4] PR[k1 + p1 + p2, m, n5]}, {k1, k2}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[70]:=
```
PlanarityTest[{PR[k1, m, n1] PR[p1 + k1, 0, n2] PR[k2, m, n3] PR[k2 − k1 − p1, m, n4]
    PR[k1 + p1, 0, n5] PR[k1 + p1 + p2, m, n6]}, {k1, k2}, TestAlgorithm → 2];
```

The Diagram

is planar.

■ **Example 32: 3 loop vertex with 2 loop self-energy insertion**

Method I

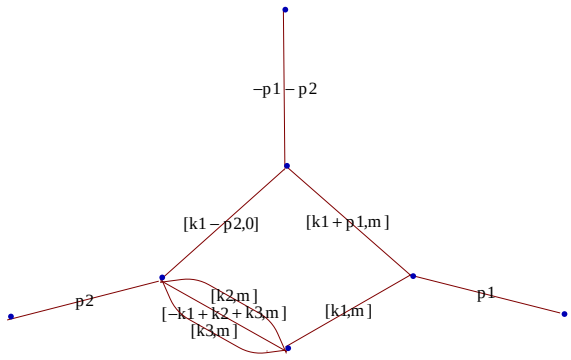
In[71]:=
```
PlanarityTest[
  {PR[k1, m, n1] PR[k2, m, n2] PR[k3, m, n3] PR[k2 + k3 − k1, m, n4] PR[k1 + p1, m, n5] PR[k1 − p2, 0, n6]},
  {k1, k2, k3}, DrawGraph → True];
```

```
ERROR. input diagram cannot be analysed and/or drawn using Method I.
please try Method II without drawing option and/or with option VertexLegNum -> 5
```

In[72]:=
```
PlanarityTest[
  {PR[k1, m, n1] PR[k2, m, n2] PR[k3, m, n3] PR[k2 + k3 − k1, m, n4] PR[k1 + p1, m, n5] PR[k1 − p2, 0, n6]},
  {k1, k2, k3}, DrawGraph → True, VertexLegNum → 5];
```

The Diagram

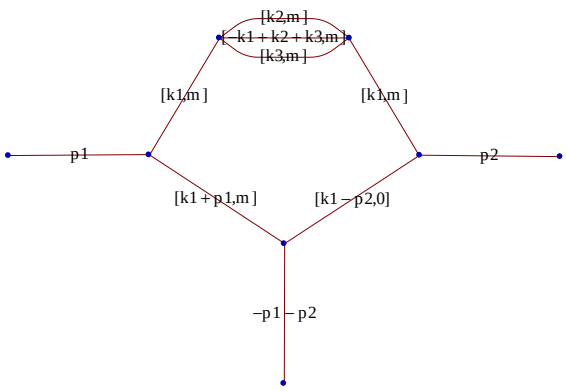
is planar.



In[73]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k1, m, n12] PR[k2, m, n2] PR[k3, m, n3]
    PR[k2 + k3 - k1, m, n4] PR[k1 + p1, m, n5] PR[k1 - p2, 0, n6]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[74]:=
```
PlanarityTest[
  {PR[k1, m, n1] PR[k2, m, n2] PR[k3, m, n3] PR[k2 + k3 - k1, m, n4] PR[k1 + p1, m, n5] PR[k1 - p2, 0, n6]},
  {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is planar.

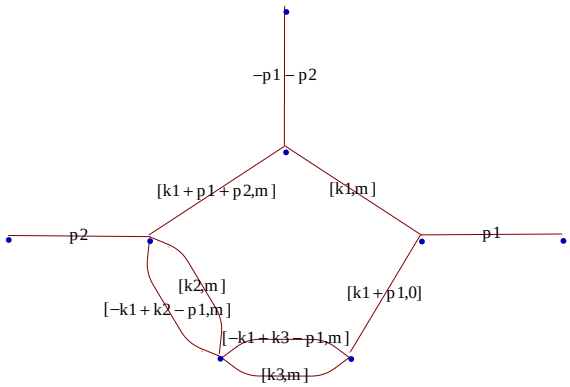
- **Example 33: vertex + bubbles a)**

Method I

In[75]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2, m, n2] PR[k2 - k1 - p1, m, n3] PR[k3, m, n4]
    PR[k3 - k1 - p1, m, n5] PR[k1 + p1, 0, n6] PR[k1 + p1 + p2, m, n7]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

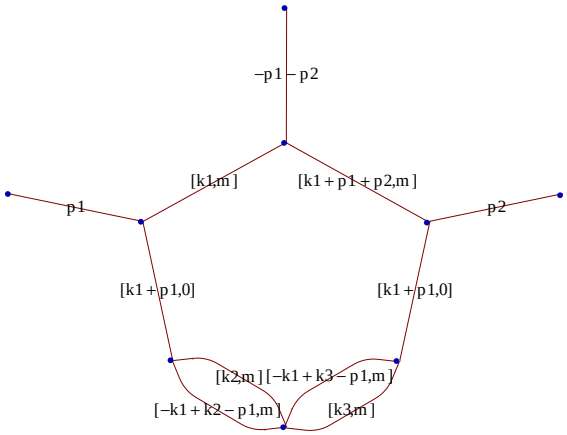
is planar.



In[76]:=
```
PlanarityTest[{PR[k1, m, n1] PR[k2, m, n2] PR[k2 - k1 - p1, m, n3] PR[k3, m, n4] PR[k3 - k1 - p1, m, n5]
    PR[k1 + p1, 0, n6] PR[k1 + p1, 0, n61] PR[k1 + p1 + p2, m, n7]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

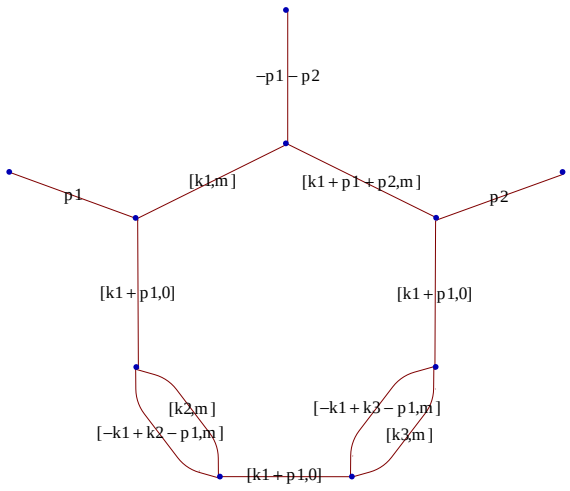
is planar.



In[77]:=
```
PlanarityTest[
  {PR[k1, m, n1] PR[k2, m, n2] PR[k2 - k1 - p1, m, n3] PR[k3, m, n4] PR[k3 - k1 - p1, m, n5] PR[k1 + p1, 0, n6]
    PR[k1 + p1, 0, n61] PR[k1 + p1, 0, n62] PR[k1 + p1 + p2, m, n7]}, {k1, k2, k3}, DrawGraph → True];
```

The Diagram

is planar.



Method II

In[78]:=
```
PlanarityTest[
  {PR[k1, m, n1] PR[k2, m, n2] PR[k2 - k1 - p1, m, n3] PR[k3, m, n4] PR[k3 - k1 - p1, m, n5] PR[k1 + p1, 0, n6]
    PR[k1 + p1, 0, n61] PR[k1 + p1, 0, n62] PR[k1 + p1 + p2, m, n7]}, {k1, k2, k3}, TestAlgorithm → 2];
```

The Diagram

is planar.

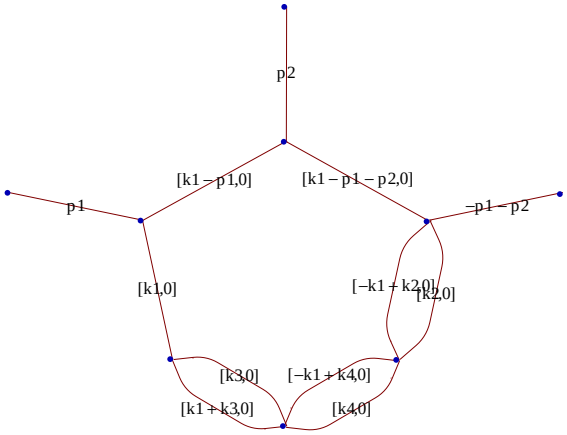
■ **Example 34: vertex + bubbles b)**

Method I

In[79]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 - k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k1, 0, n5] PR[k4, 0, n6] PR[k4 - k1, 0, n7] PR[k1 - p1, 0, n8] PR[k1 - p1 - p2, 0, n9]},
  {k1, k2, k3, k4}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.

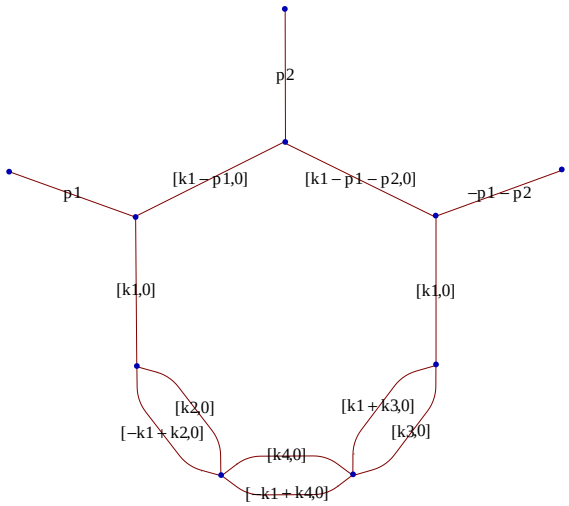


Out[79]= {0.234834, True}

In[80]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1, 0, n11] PR[k2, 0, n2] PR[k2 - k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k1, 0, n5] PR[k4, 0, n6] PR[k4 - k1, 0, n7] PR[k1 - p1, 0, n8] PR[k1 - p1 - p2, 0, n9]},
  {k1, k2, k3, k4}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.

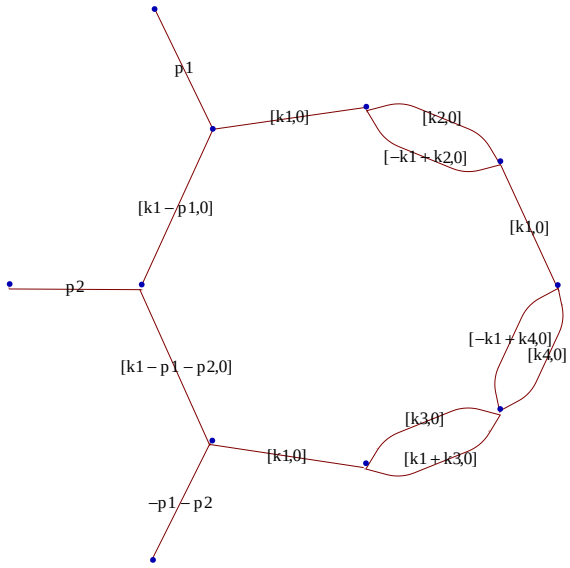


Out[80]= {0.336259, True}

In[81]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1, 0, n11] PR[k1, 0, n12] PR[k2, 0, n2] PR[k2 – k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k1, 0, n5] PR[k4, 0, n6] PR[k4 – k1, 0, n7] PR[k1 – p1, 0, n8] PR[k1 – p1 – p2, 0, n9]},
  {k1, k2, k3, k4}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.

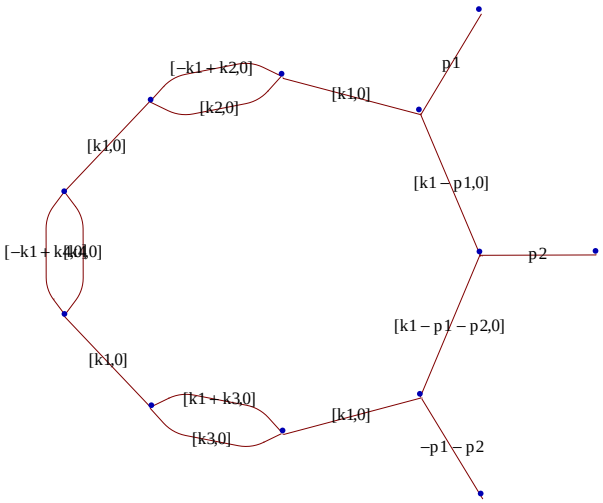


Out[81]= {0.938522, True}

In[82]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k1, 0, n11] PR[k1, 0, n12] PR[k1, 0, n13] PR[k2, 0, n2] PR[k2 - k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k1, 0, n5] PR[k4, 0, n6] PR[k4 - k1, 0, n7]  PR[k1 - p1, 0, n8] PR[k1 - p1 - p2, 0, n9]},
  {k1, k2, k3, k4}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[82]= {8.747735, True}

Method II

In[83]:=
```
PlanarityTest[
  {PR[k1, 0, n1] PR[k1, 0, n11] PR[k1, 0, n12] PR[k1, 0, n13] PR[k2, 0, n2] PR[k2 - k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k1, 0, n5] PR[k4, 0, n6] PR[k4 - k1, 0, n7]  PR[k1 - p1, 0, n8] PR[k1 - p1 - p2, 0, n9]},
  {k1, k2, k3, k4}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is planar.

Out[83]= {0.526891, True}

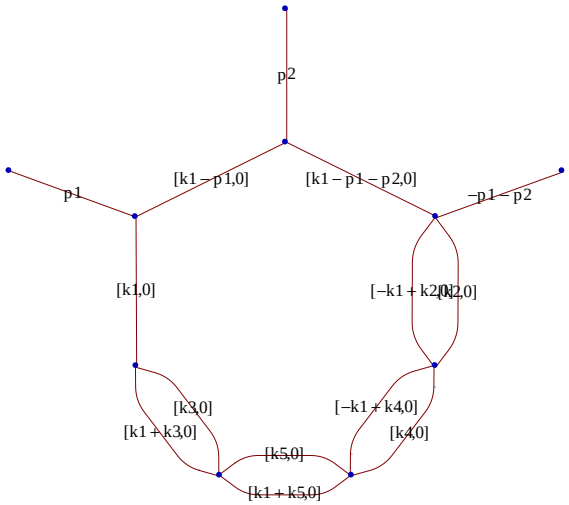
- **Example 35: vertex + bubbles c)**

Method I

In[84]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 - k1, 0, n3] PR[k3, 0, n4] PR[k3 + k1, 0, n5] PR[k4, 0, n6]
    PR[k4 - k1, 0, n7] PR[k5, 0, n8] PR[k5 + k1, 0, n9] PR[k1 - p1, 0, n10] PR[k1 - p1 - p2, 0, n11]},
  {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.

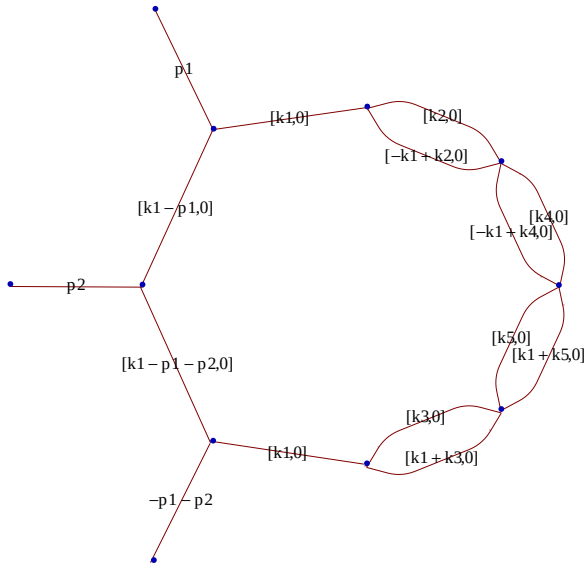


Out[84]= {0.475620, True}

In[85]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1, 0, n11] PR[k2, 0, n2] PR[k2 - k1, 0, n3] PR[k3, 0, n4] PR[k3 + k1, 0, n5]
    PR[k4, 0, n6] PR[k4 - k1, 0, n7] PR[k5, 0, n8] PR[k5 + k1, 0, n9] PR[k1 - p1, 0, n10]
    PR[k1 - p1 - p2, 0, n11]}, {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.

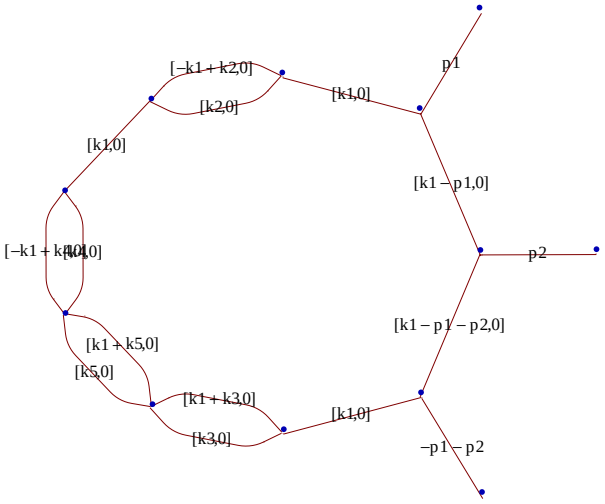


Out[85]= {1.942249, True}

In[86]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1, 0, n11] PR[k1, 0, n12] PR[k2, 0, n2] PR[k2 - k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k1, 0, n5] PR[k4, 0, n6] PR[k4 - k1, 0, n7] PR[k5, 0, n8] PR[k5 + k1, 0, n9] PR[k1 - p1, 0, n10]
    PR[k1 - p1 - p2, 0, n11]}, {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.

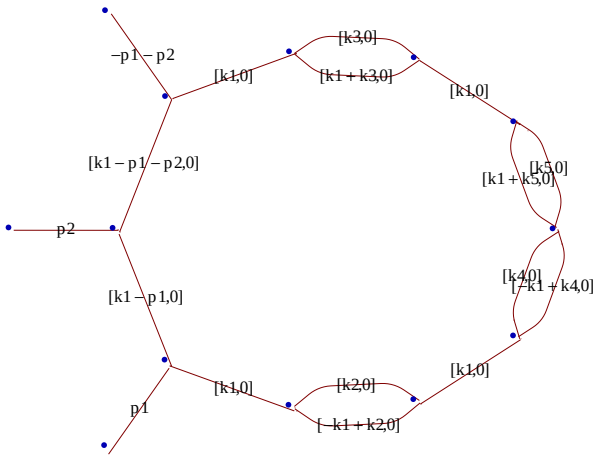


Out[86]= {18.253752, True}

In[87]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1, 0, n11] PR[k1, 0, n12] PR[k1, 0, n13] PR[k2, 0, n2]
    PR[k2 - k1, 0, n3] PR[k3, 0, n4] PR[k3 + k1, 0, n5] PR[k4, 0, n6] PR[k4 - k1, 0, n7]
    PR[k5, 0, n8] PR[k5 + k1, 0, n9] PR[k1 - p1, 0, n10] PR[k1 - p1 - p2, 0, n11]},
  {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[87]= {661.825879, True}

Method II

In[88]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k1, 0, n11] PR[k1, 0, n12] PR[k1, 0, n13] PR[k2, 0, n2]
    PR[k2 - k1, 0, n3] PR[k3, 0, n4] PR[k3 + k1, 0, n5] PR[k4, 0, n6] PR[k4 - k1, 0, n7]
    PR[k5, 0, n8] PR[k5 + k1, 0, n9] PR[k1 - p1, 0, n10] PR[k1 - p1 - p2, 0, n11]},
  {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is planar.

Out[88]= {2.437872, True}

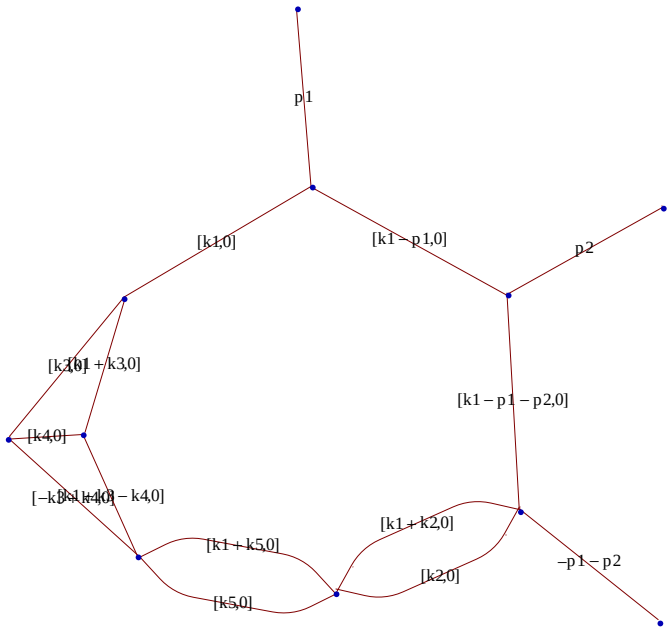
- **Example 36: vertex + a)**

Method I

In[89]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4] PR[k3 + k1, 0, n5] PR[k4, 0, n6]
    PR[k4 - k3, 0, n7] PR[k1 + k3 - k4, 0, n8] PR[k5, 0, n9] PR[k5 + k1, 0, n10] PR[k1 - p1, 0, n11]
    PR[k1 - p1 - p2, 0, n12]}, {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[89]= {1.226391, True}

Method II

In[90]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4] PR[k3 + k1, 0, n5] PR[k4, 0, n6]
    PR[k4 - k3, 0, n7] PR[k1 + k3 - k4, 0, n8] PR[k5, 0, n9] PR[k5 + k1, 0, n10] PR[k1 - p1, 0, n11]
    PR[k1 - p1 - p2, 0, n12]}, {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is planar.

Out[90]= {2.004222, True}

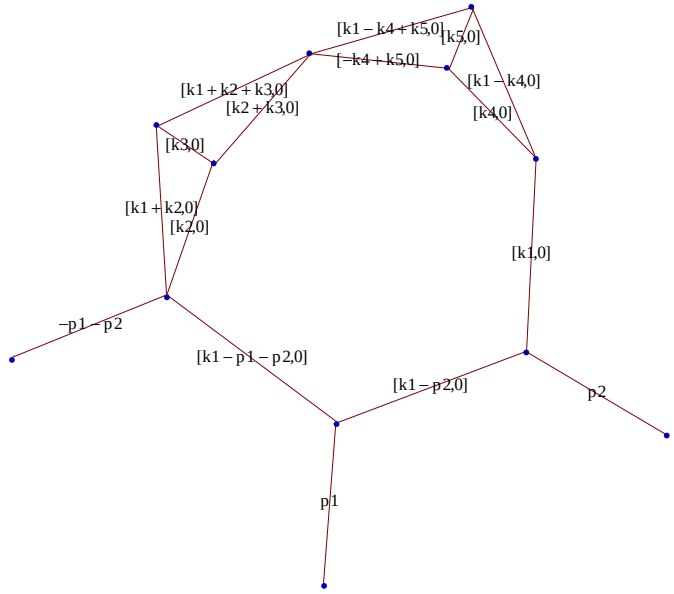
■ **Example 37: vertex + b)**

Method I

In[91]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k2, 0, n5] PR[k1 + k3 + k2, 0, n6] PR[k4, 0, n7] PR[k1 - k4, 0, n8] PR[k5, 0, n9]
    PR[k5 - k4, 0, n10] PR[k1 - k4 + k5, 0, n11] PR[k1 - p2, 0, n12] PR[k1 - p1 - p2, 0, n13]},
  {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[91]= {7.415171, True}

Method II

In[92]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k2, 0, n5] PR[k1 + k3 + k2, 0, n6] PR[k4, 0, n7] PR[k1 - k4, 0, n8] PR[k5, 0, n9]
    PR[k5 - k4, 0, n10] PR[k1 - k4 + k5, 0, n11] PR[k1 - p2, 0, n12] PR[k1 - p1 - p2, 0, n13]},
  {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is planar.

Out[92]= {2.233452, True}

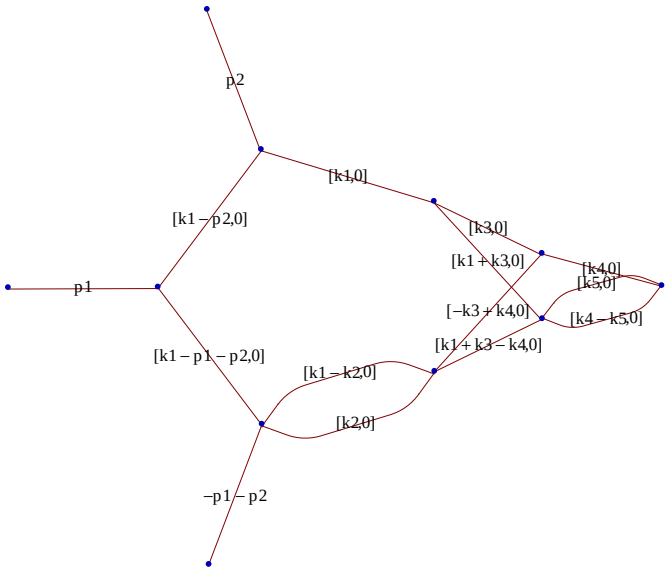
■ **Example 38: vertex + c)**

Method I

In[93]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[-k2 + k1, 0, n3] PR[k3, 0, n4] PR[k3 + k1, 0, n5]
    PR[k1 + k3 - k4, 0, n6] PR[k4, 0, n7] PR[k4 - k5, 0, n8] PR[k5, 0, n9] PR[k4 - k3, 0, n10]
    PR[k1 - p2, 0, n11] PR[k1 - p1 - p2, 0, n12]}, {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[93]= {0.866434, True}

Method II

In[94]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[-k2 + k1, 0, n3] PR[k3, 0, n4] PR[k3 + k1, 0, n5]
    PR[k1 + k3 - k4, 0, n6] PR[k4, 0, n7] PR[k4 - k5, 0, n8] PR[k5, 0, n9] PR[k4 - k3, 0, n10]
    PR[k1 - p2, 0, n11] PR[k1 - p1 - p2, 0, n12]}, {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is planar.

Out[94]= {2.054944, True}

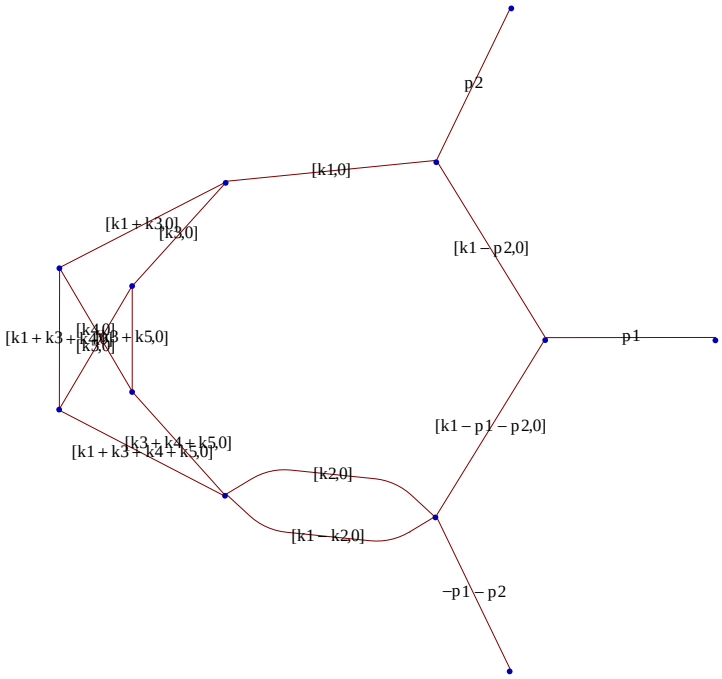
- **Example 39: vertex + d)**

Method I

In[95]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[-k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k1, 0, n5] PR[k1 + k3 + k4, 0, n6] PR[k4, 0, n7] PR[k3 + k5, 0, n8] PR[k5, 0, n9]
    PR[k3 + k4 + k5, 0, n10] PR[k1 + k3 + k4 + k5, 0, n11] PR[k1 - p2, 0, n12] PR[k1 - p1 - p2, 0, n13]},
  {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is non-planar.



Out[95]= {6.049647, False}

Method II

In[96]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[-k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k3 + k1, 0, n5] PR[k1 + k3 + k4, 0, n6] PR[k4, 0, n7] PR[k3 + k5, 0, n8] PR[k5, 0, n9]
    PR[k3 + k4 + k5, 0, n10] PR[k1 + k3 + k4 + k5, 0, n11] PR[k1 - p2, 0, n12] PR[k1 - p1 - p2, 0, n13]},
  {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is non-planar.

Out[96]= {2.244753, False}

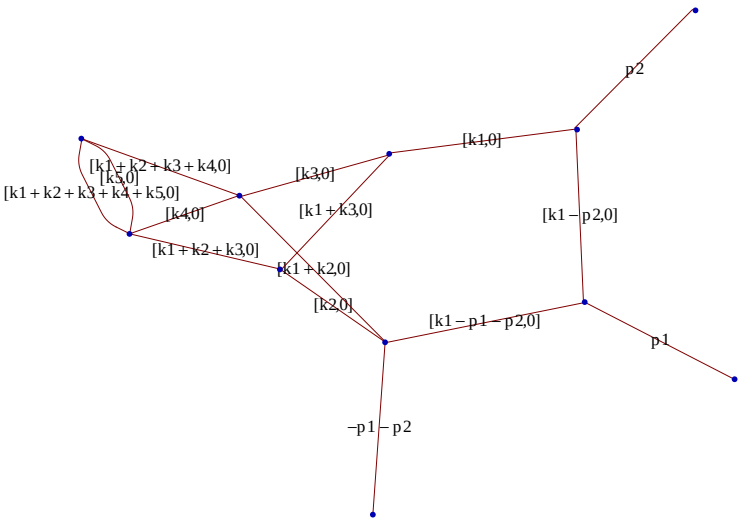
- **Example 40: vertex + e)**

Method I

In[97]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k2 + k3 + k1, 0, n5] PR[k1 + k2 + k3 + k4, 0, n6] PR[k4, 0, n7] PR[k1 + k2 + k3 + k4 + k5, 0, n8]
    PR[k5, 0, n9] PR[k1 + k3, 0, n10] PR[k1 - p2, 0, n11] PR[k1 - p1 - p2, 0, n12]},
  {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[97]= {0.683741, True}

Method II

In[98]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k2 + k3 + k1, 0, n5] PR[k1 + k2 + k3 + k4, 0, n6] PR[k4, 0, n7] PR[k1 + k2 + k3 + k4 + k5, 0, n8]
    PR[k5, 0, n9] PR[k1 + k3, 0, n10] PR[k1 − p2, 0, n11] PR[k1 − p1 − p2, 0, n12]},
  {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is planar.

Out[98]= {1.932294, True}

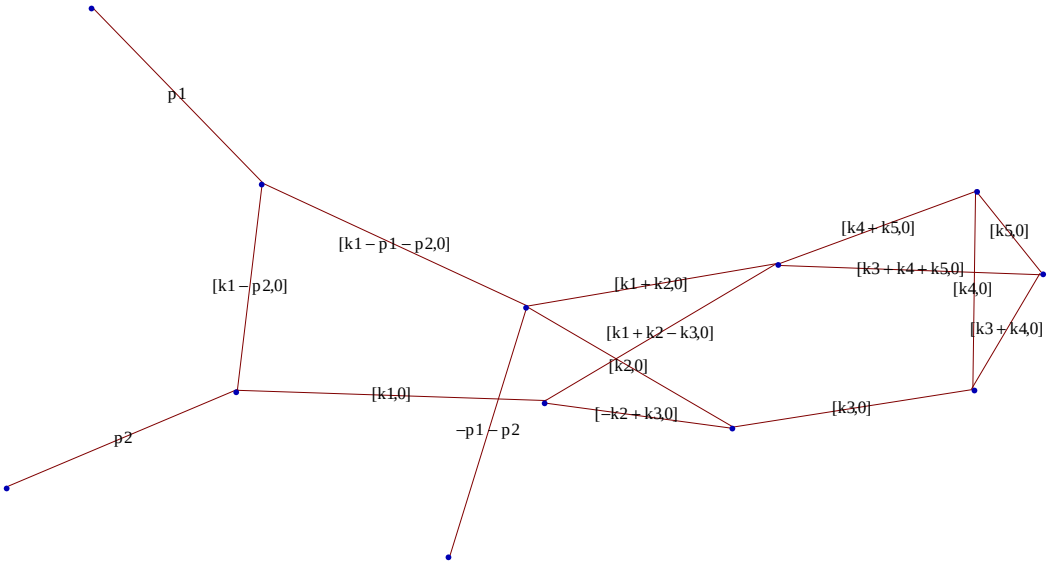
- **Example 41: vertex + f)**

Method I

In[99]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k2 − k3 + k1, 0, n5] PR[k3 + k4, 0, n6] PR[k4, 0, n7] PR[k3 + k4 + k5, 0, n8] PR[k5, 0, n9]
    PR[k4 + k5, 0, n10] PR[k3 − k2, 0, n11] PR[k1 − p2, 0, n12] PR[k1 − p1 − p2, 0, n13]},
  {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[99]= {4.806903, True}

Method II

In[100]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k2 – k3 + k1, 0, n5] PR[k3 + k4, 0, n6] PR[k4, 0, n7] PR[k3 + k4 + k5, 0, n8] PR[k5, 0, n9]
    PR[k4 + k5, 0, n10] PR[k3 – k2, 0, n11] PR[k1 – p2, 0, n12] PR[k1 – p1 – p2, 0, n13]},
  {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is planar.

Out[100]= {4.479948, True}

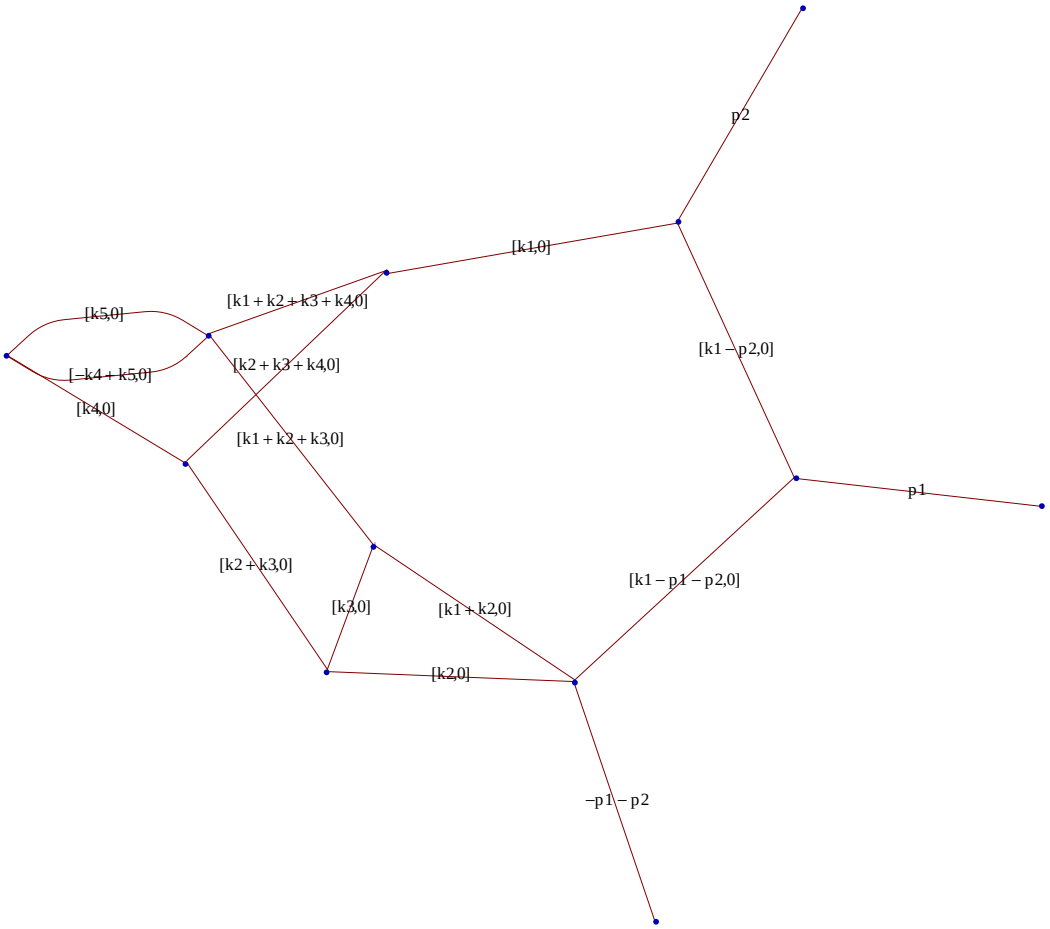
- **Example 42: vertex + g)**

Method I

In[101]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k2 + k3 + k1, 0, n5] PR[k3 + k2, 0, n6] PR[k4, 0, n7] PR[k3 + k4 + k2, 0, n8] PR[k5, 0, n9]
    PR[k1 + k2 + k4 + k3, 0, n10] PR[k5 – k4, 0, n11] PR[k1 – p2, 0, n12] PR[k1 – p1 – p2, 0, n13]},
  {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[101]= {8.588384, True}

Method II

In[102]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4]
    PR[k2 + k3 + k1, 0, n5] PR[k3 + k2, 0, n6] PR[k4, 0, n7] PR[k3 + k4 + k2, 0, n8] PR[k5, 0, n9]
    PR[k1 + k2 + k4 + k3, 0, n10] PR[k5 − k4, 0, n11] PR[k1 − p2, 0, n12] PR[k1 − p1 − p2, 0, n13]},
  {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is planar.

Out[102]= {2.184458, True}

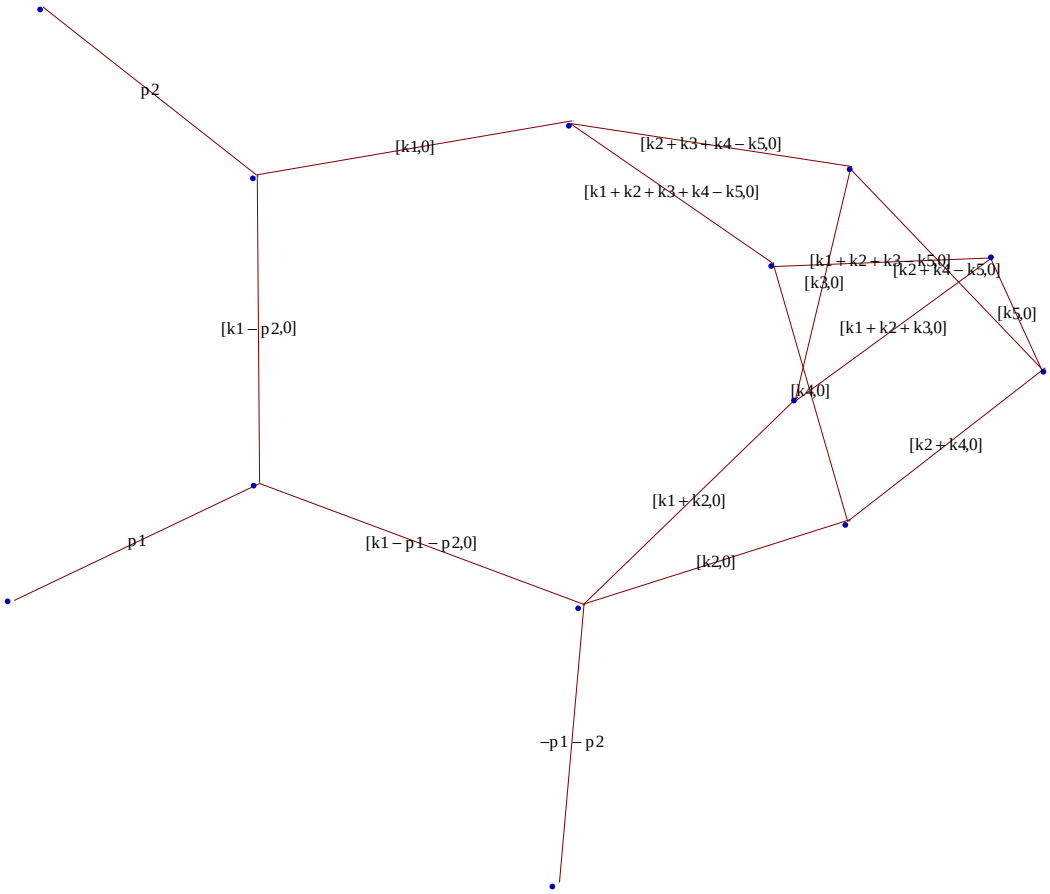
- **Example 43: vertex + h)**

Method I

In[103]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4] PR[k2 + k3 + k1, 0, n5]
    PR[k4 + k2, 0, n6] PR[k4, 0, n7] PR[-k5 + k4 + k2, 0, n8] PR[k5, 0, n9] PR[k1 + k2 - k5 + k3, 0, n10]
    PR[k1 + k2 + k3 + k4 - k5, 0, n11] PR[k2 + k3 + k4 - k5, 0, n12] PR[k1 - p2, 0, n13] PR[k1 - p1 - p2, 0, n14]},
  {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is non-planar.



Out[103]= {42.692192, False}

Method II

In[104]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4] PR[k2 + k3 + k1, 0, n5]
    PR[k4 + k2, 0, n6] PR[k4, 0, n7] PR[-k5 + k4 + k2, 0, n8] PR[k5, 0, n9] PR[k1 + k2 - k5 + k3, 0, n10]
    PR[k1 + k2 + k3 + k4 - k5, 0, n11] PR[k2 + k3 + k4 - k5, 0, n12] PR[k1 - p2, 0, n13] PR[k1 - p1 - p2, 0, n14]},
  {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

The Diagram

is non-planar.

Out[104]= {2.602694, False}

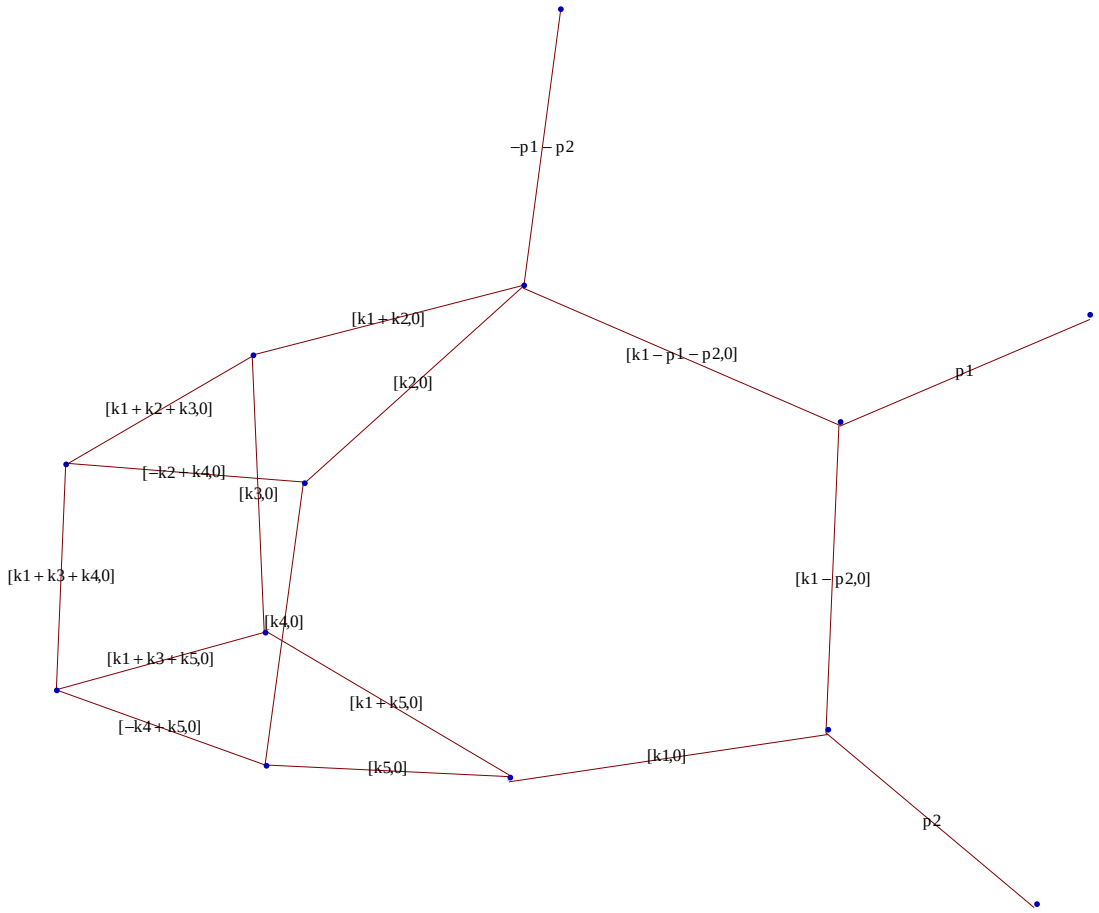
■ **Example 44: vertex + k)**

Method I

In[105]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4] PR[k2 + k3 + k1, 0, n5]
    PR[k4 - k2, 0, n6] PR[k4, 0, n7] PR[k1 + k4 + k3, 0, n8] PR[k5, 0, n9] PR[k1 + k5 + k3, 0, n10]
    PR[-k4 + k5, 0, n11] PR[k1 + k5, 0, n12] PR[k1 - p2, 0, n13] PR[k1 - p1 - p2, 0, n14]},
  {k1, k2, k3, k4, k5}, DrawGraph → True] // AbsoluteTiming
```

The Diagram

is planar.



Out[105]= {59.764877, True}

Method II

In[106]:=
```
PlanarityTest[{PR[k1, 0, n1] PR[k2, 0, n2] PR[k2 + k1, 0, n3] PR[k3, 0, n4] PR[k2 + k3 + k1, 0, n5]
    PR[k4 - k2, 0, n6] PR[k4, 0, n7] PR[k1 + k4 + k3, 0, n8] PR[k5, 0, n9] PR[k1 + k5 + k3, 0, n10]
    PR[-k4 + k5, 0, n11] PR[k1 + k5, 0, n12] PR[k1 - p2, 0, n13] PR[k1 - p1 - p2, 0, n14]},
  {k1, k2, k3, k4, k5}, TestAlgorithm → 2] // AbsoluteTiming
```

```
The Diagram
is planar.
```

Out[106]= {2.480502, True}