Adigraph, V1.4

Luca Cappelletti

March 2018

Contents

Chapter 1

Introduction

1.1 What is Adigraph?

Adigraph is a latex library for drawing directed graphs and augmenting directed graphs, and to draw cuts over them.

It handles automatically the positioning of labels, with the exception of the horizontal position, and the inclinations of cuts.

The latest version is available on Github.

1.2 License

Chapter 2

Setup

2.1 Installing the dependencies

Clearly you need to have texlive installed. Then, make sure you have the following packages:

fp Used for floating point calculations.

xparse Used for elaborating parameters.

xstring Used for elaborating strings.

etoolbox Used for operations on lists.

tikz Used for drawing the actual graphs.

tikz calc library Used for some internal calculations in tikz.

To be sure you can run the following, that will install the packages only if they are not already present:

```
sudo tlmgr install etoolbox fp xstring
```

2.2 Installing Adigraph

You can install Adigraph, if it isn't already present in your setup, by running the following on Unix systems:

```
sudo tlmgr install adigraph
```

On windows you should check on your package manager of choice (some latex distribution have a tlmgr implementation on windows too.)

Chapter 3

Usage

3.1 Creating a new graph

Here we create a new Adigraph object, called *myAdigraph*.

```
\NewAdigraph{myAdigraph}{
    <nodes here, separated by semicolon>
}{
    <edges here, separated by semicolon>
}{
    <cuts here, separated by semicolon>
}
```

3.2 Changing an existing graph

You can renovate an older graph by calling \RenewAdigraph

```
\RenewAdigraph{myAdigraph}{
    <nodes here, separated by semicolon>
}{
    <edges here, separated by semicolon>
}{
    <cuts here, separated by semicolon>
}
```

3.3 Adding nodes

We set its nodes with the following syntax: *<node name, color: x coordinate, y coordinate: label>*.

```
\NewAdigraph{myAdigraph}{
      s:0,0;
      t:4,0;
}
\myAdigraph{}
```

3.3.1 Custom node colors

To color a node you can use the following syntax: *<node name, textual color: x coordinate, y coordinate>*. For example, to draw s in red and t in blue we would write:

```
\NewAdigraph{myAdigraph}{
      s,red:0,0;
      t,blue:4,0;
}
\myAdigraph{}
```

Tested available colors are: red, blue, black, green. You may extend the possible colors with LaTex libraries such as xcolor.

3.3.2 Custom node labels

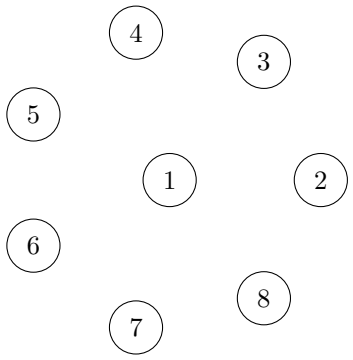
To add a custom label you can use the following syntax: either *<node name: x coordinate, y coordinate: node label>* or *<node name,textual color: x coordinate, y coordinate: node label>* will work:

```
\NewAdigraph{myAdigraph}{
      s,red:0,0:start;
      t:4,0:end;
}
\myAdigraph{}
```

3.4 Automatically position nodes

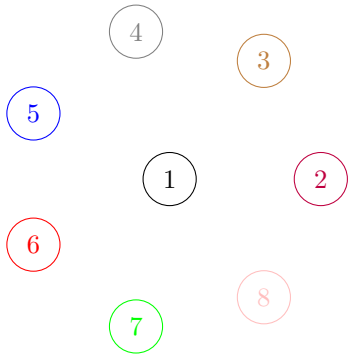
When no coordinates are given or you just don't have time to think abount where to put those nodes, just choose a radius and Adigraph will position them on the circle of that radius.

```
\NewAdigraph{myAdigraph}{
    1:0,0;
    2:2;
    3:2;
    4:2;
    5:2;
    6:2;
    7:2;
    8:2;
}
\myAdigraph{}
```



3.4.1 Colored automatically positioned nodes

```
\NewAdigraph{myAdigraph}{
    1:0,0;
    2,purple:2;
    3,brown:2;
    4,gray:2;
    5,blue:2;
    6,red:2;
    7,green:2;
    8,pink:2;
}
\myAdigraph{}
```

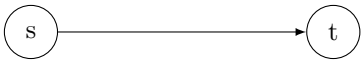


3.5 Adding edges

We set its nodes with the following syntax: *<node name: x coordinate, y coordinate, color : label>*.

3.5.1 A simple edge

```
\NewAdigraph{myAdigraph}{
      s:0,0;
      t:4,0;
}{
     s,t;
}
\myAdigraph{}
```

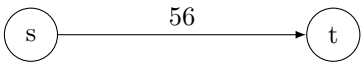


3.5.2 A colored simple edge

```
\NewAdigraph{myAdigraph}{
      s:0,0;
      t:4,0;
}{
     s,t,red;
}
\myAdigraph{}
```

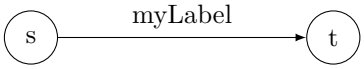
3.5.3 A weighted edge

```
\NewAdigraph{myAdigraph}{
      s:0,0;
      t:4,0;
}{
     s,t:56;
}
\myAdigraph{}
```



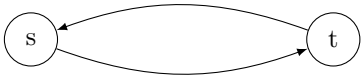
3.5.4 A weighted edge with label

```
\NewAdigraph{myAdigraph}{
     s:0,0;
     t:4,0;
}{
    s,t:56:myLabel;
}
\myAdigraph{}
```



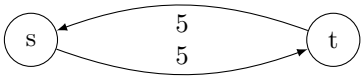
3.5.5 Edge in both directions

```
\NewAdigraph{myAdigraph}{
     s:0,0;
     t:4,0;
}{
    s,t;
    t,s;
}
\myAdigraph{}
```



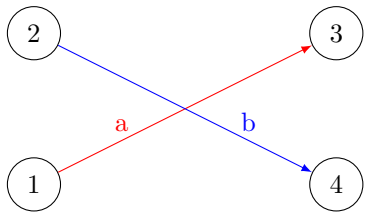
3.5.6 Edge with weights in both directions

```
\NewAdigraph{myAdigraph}{
     s:0,0;
     t:4,0;
}{
    s,t:5;
    t,s:5;
}
\myAdigraph{}
```



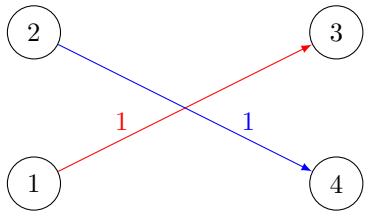
3.5.7 Positioning labels

```
\NewAdigraph{myAdigraph}{
    1:0,0;
    2:0,2;
    3:4,2;
    4:4,0;
}{
    1,3,red:1:a:near start;
    2,4,blue:1:b:near end;
}
\myAdigraph{}
```



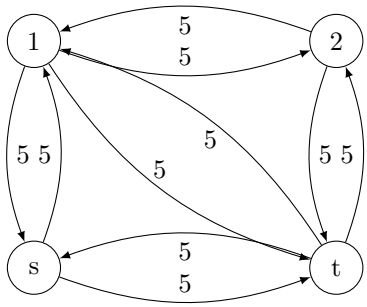
3.5.8 Positioning weights

```
\NewAdigraph{myAdigraph}{
    1:0,0;
    2:0,2;
    3:4,2;
    4:4,0;
}{
    1,3,red:1::near start;
    2,4,blue:1::near end;
}
\myAdigraph{}
```



3.5.9 Multiple edges with weights

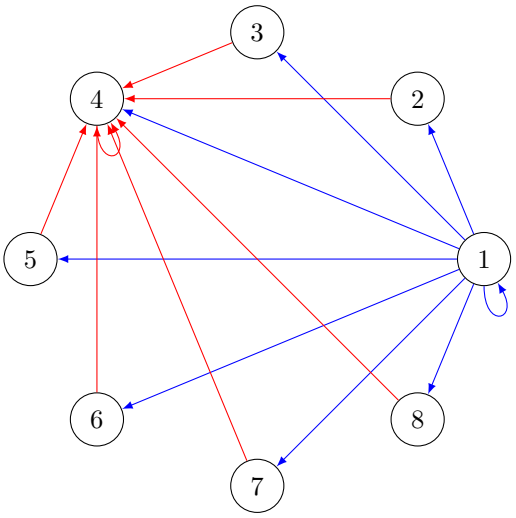
```
\NewAdigraph{myAdigraph}{
    s:0,0;
    t:4,0;
    1:0,3;
    2:4,3;
}{
    s,t:5;
    t,s:5;
    s,1:5;
    1,s:5;
    1,2:5;
    2,1:5;
    2,t:5;
    t,2:5;
    t,1:5;
    1,t:5;
}
\myAdigraph{}
```



3.5.10 Kleene star

This works only when you don't have a node called <*>. When this happens, the behaviour of a tuple <*a*,*> becomes the normal one.

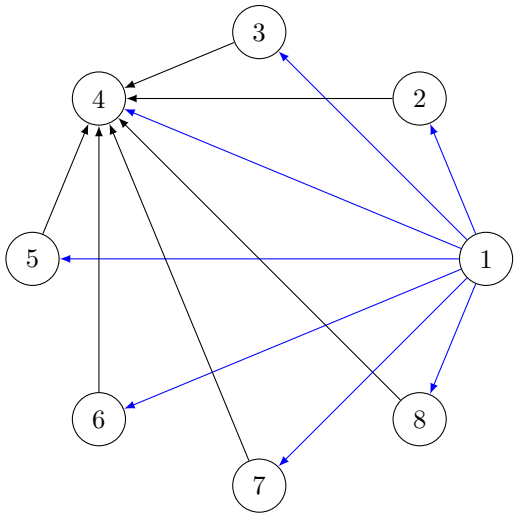
```
\NewAdigraph{myAdigraph}{
    1:3;
    2:3;
    3:3;
    4:3;
    5:3;
    6:3;
    7:3;
    8:3;
}{
    1,*,blue;
    *,4,red;
}
\myAdigraph{}
```



3.5.11 Kleene star minus the element

This works only when you don't have a node called <+>. When this happens, the behaviour of a tuple <*a*,+> becomes the normal one.

```
\NewAdigraph{myAdigraph}{
    1:3;
    2:3;
    3:3;
    4:3;
    5:3;
    6:3;
    7:3;
    8:3;
}{
    1,+,blue;
    +,4;
}
\myAdigraph{}
```



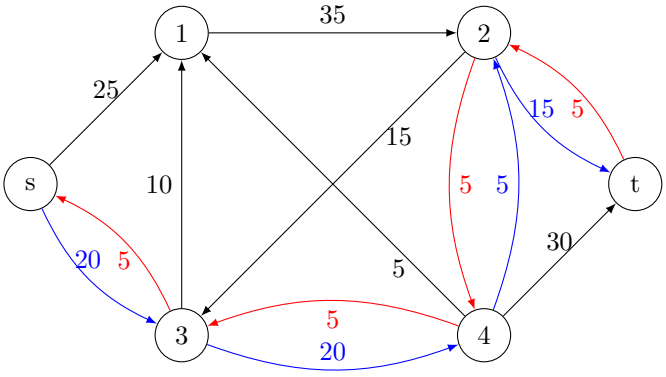
3.5.12 Combining Kleene operations

Sadly, operations such as <*,+> or <+,+> are not currently supported and not for lack of trying. I'll try implementing them again in the future when I'll have more time.

3.6 Augmenting paths

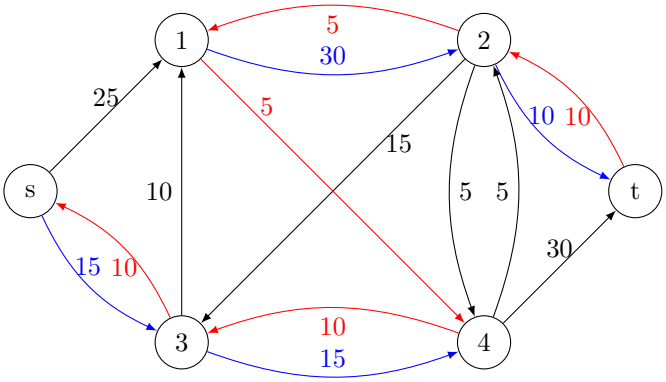
An augmenting path is specified by the following syntax: <*comma separated list of nodes:units*>. It is **very important** to note that incremental paths called upon the same object are memorized by default.

```
\NewAdigraph{myAdigraph}{
    s:0,0;
    1:2,2;
    3:2,-2;
    2:6,2;
    4:6,-2;
    t:8,0;
}{
    s,1:25;
    s,3:25;
    3,4:25;
    1,2:35;
    2,t:20;
    4,t:30;
    3,1:10;
    4,2:10;
    2,3:15::near start;
    4,1:5::near start;
}
\myAdigraph{
    s,3,4,2,t:5;
}
```

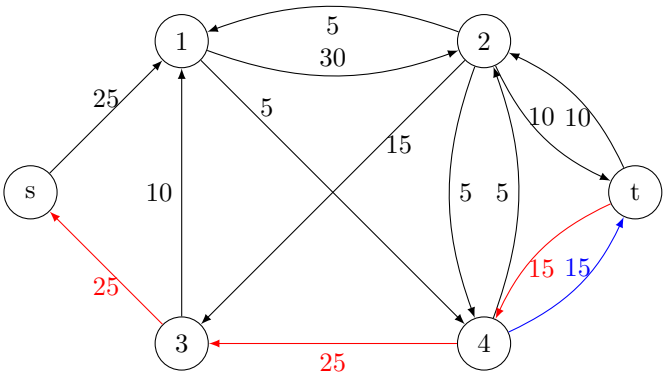


For example, suppose now we'd like to send another 5 units on the graph edited by the previous incremental path, we'll have just to write the following:

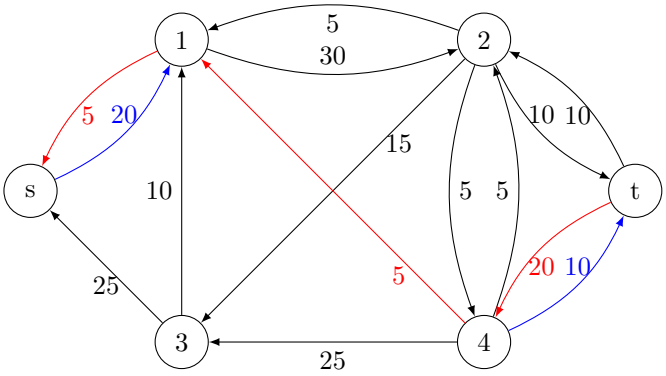
```
\myAdigraph{
    s,3,4,1,2,t:5;
}
```



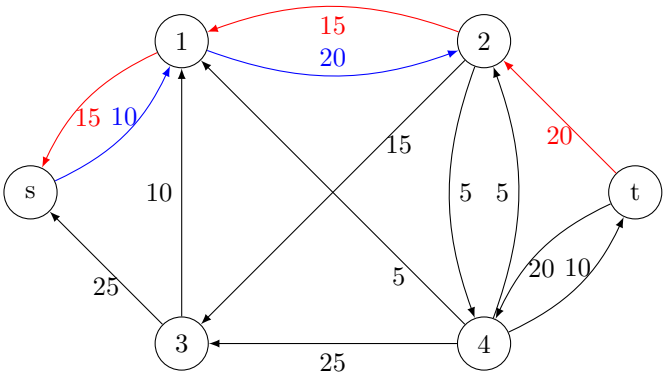
```
\myAdigraph{
    s,3,4,t:15;
}
```



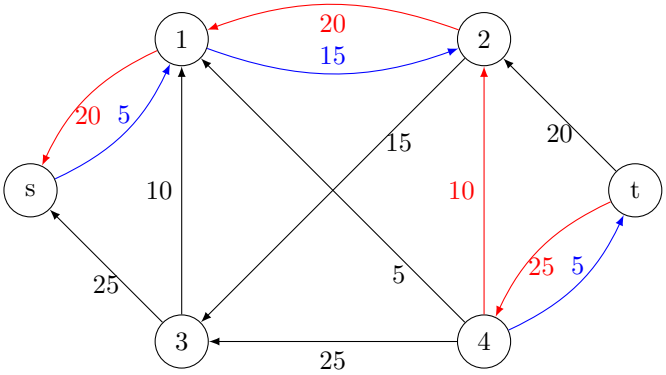

```
\myAdigraph{
    s,1,4,t:5;
}
```




```
\myAdigraph{
    s,1,2,t:10;
}
```



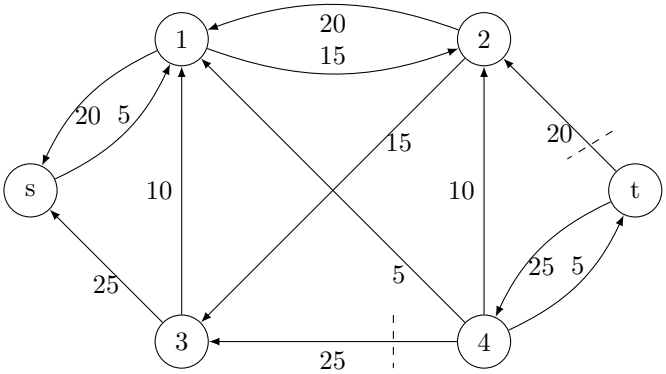
```
\myAdigraph{
    s,1,2,4,t:5;
}
```



3.7 Cuts

The following is to add cuts to show minimum cuts for example, the syntax is: *<first node, second node;>*

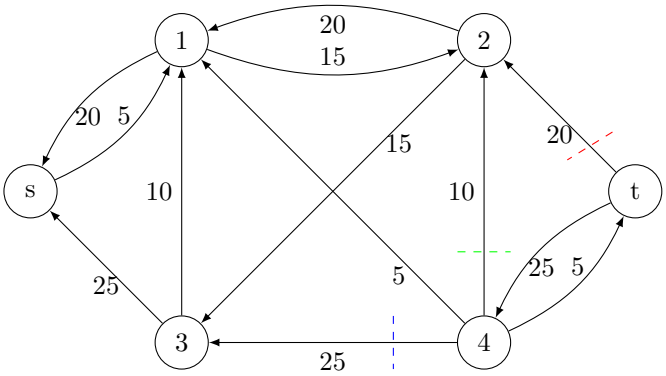
```
\myAdigraph{}{
    3,4;
    2,t;
}
```



3.7.1 Colored cuts

If you'd like to color the cuts you just have to add the color as follows: *<first node, second node, color;>*

```
\myAdigraph{}{
    3,4,red;
    2,t,blue;
}
```



Chapter 4

Corner case testing

4.1 Full features on automatically positioned nodes

Chapter 5

Warnings

5.1 Reserved words

I reserve to use for the package the following tokens:

1. \Adigraph
2. \AdigraphBuildEdge
3. \AdigraphBuildEdgeWrapper
4. \AdigraphBuildNode
5. \AdigraphBuildNodeWrapper
6. \AdigraphBuildPath
7. \AdigraphCalculateOrientation
8. \AdigraphCountPaths
9. \AdigraphCutBuilder
10. \AdigraphDrawEdge
11. \AdigraphDrawNode
12. \AdigraphEdgeBuilder
13. \AdigraphEdgeDrawer
14. \AdigraphElaboratePath
15. \AdigraphExecuteCutBuilder
16. \AdigraphGenerateNodeName
17. \AdigraphMemorizeEdge
18. \AdigraphMemorizeNode
19. \AdigraphNodeBuilder
20. \AdigraphNodeCounter
21. \AdigraphNodeCounterSecond-Wrapper
22. \AdigraphNodeCounterWrapper
23. \AdigraphNodesCounter
24. \AdigraphPathBuilder
25. \AdigraphProcessAugmenting-Paths
26. \AdigraphProcessAugmenting-PathsList
27. \AdigraphProcessCuts
28. \AdigraphProcessEdges
29. \AdigraphProcessNodes
30. \AdigraphProcessPaths
31. \AdigraphSimpleSum
32. \NewAdigraph
33. \RenewAdigraph