

Visualisation de graphes avec Graphviz

Graphviz est un logiciel de visualisation de graphes. Il permet de représenter des données structurées sous la forme de diagrammes de graphes. La visualisation automatique de graphes a de nombreuses applications, aussi bien pour l'ingénierie, les bases de données, les réseaux que pour des interfaces visuelles dans beaucoup d'autres domaines.

Graphviz est un logiciel de diffusion libre disponible sur le site <http://graphviz.org/> pour différentes distributions Linux ainsi que pour Windows ou MacOS X. Graphviz est un ensemble de programmes correspondant à l'implémentation de différents algorithmes.

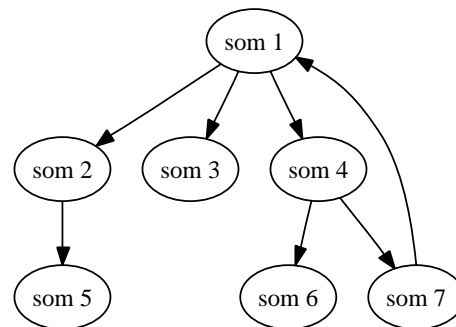
Graphviz utilise des fichiers textes dans un format simple (DOT) pour décrire les graphes. Les diagrammes sont exportés sous différents formats d'images (FIG, postscript, GIF, JPEG, PIC,...). Ce document présente brièvement l'utilisation de Graphviz. Une documentation détaillée sur le logiciel Graphviz ainsi que sur son utilisation en tant que bibliothèque sont données en détails sur le site <http://graphviz.org/>.

Le format DOT

Le format DOT décrit trois types d'objets : graphes, sommets et arêtes. Un sous-graphe définit un sous-ensemble de sommets et d'arêtes. Les graphes peuvent être orientés (digraph) ou non-orientés (graph). La figure suivante donne un exemple de graphe au format DOT. La première ligne donne le type de graphe, ainsi que son nom. Les lignes suivantes représentent les sommets, les arêtes et les sous-graphes ainsi que les ensembles d'attributs associés. Un sommet est créé quand son nom apparaît pour la première fois. Une arête est créée lorsque les sommets sont reliés par l'opérateur d'arête `->` (ou l'opérateur `--` en non-orienté).

```

1: digraph G {
2:     "som 1" -> "som 2" -> "som 5";
3:     "som 1" -> "som 3";
4:     "som 1" -> "som 4";
5:     "som 4" -> {"som 6";"som 7"};
6:     "som 7" -> "som 1";
7: }
```

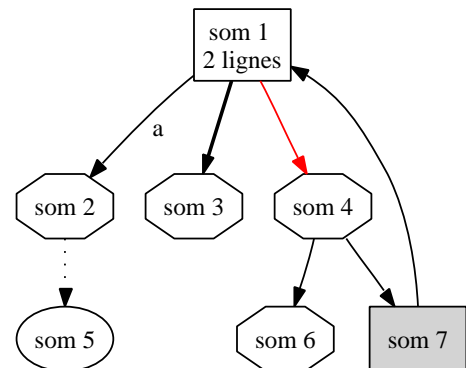


Il est souvent intéressant de pouvoir modifier la représentation et la disposition des sommets et des arêtes. Ceci est rendu possible par l'ajout de différents attributs liés aux sommets, aux arêtes et aux sous-graphes dans le fichier d'entrée. La figure suivante présente quelque-uns de ces attributs. La ligne 2 impose la taille du graphe (en pouces). Les attributs sur les sommets et les arêtes sont donnés entre crochets `[]`. La ligne 3 indique que tous les sommets du graphe sont représentés par des polygones à 8 côtés. On peut également définir des attributs associés à toutes les arêtes en utilisant le mot réservé `edge`. On peut remarquer que plusieurs attributs peuvent être conjugués entre les crochets `[]`. La ligne 4

```

1: digraph G {
2:   size ="5,5";
3:   node [shape=polygon, sides=8];
4:   som1 [label="som 1 \n2 lignes"];
5:   som1 [shape=box];
6:   "som 5" [shape=ellipse];
7:   "som 7" [shape=box,style=filled];
8:   som1 -> "som 2" [label="a"];
9:   "som 2" -> "som 5" [style=dotted];
10:  som1 -> "som 3" [style=bold];
11:  som1 -> "som 4" [color=red];
12:  "som 4" -> {"som 6";"som 7"};
13:  "som 7" -> som1;
14: }

```



permet d'associer une étiquette au sommet `som 1`. La ligne 5 indique que le sommet `som 1` sera représenté par un rectangle et non par un octogone. Le sommet `som 5` sera représenté par une ellipse (ligne 6). Le sommet `som 7` sera représenté par un rectangle plein (ligne 7). L'arête entre `som 1` et `som 2` portera l'étiquette `a` (ligne 8). L'arête entre `som 2` et `som 5` sera en pointillés (ligne 10). L'arête entre `som 1` et `som 3` sera en gras (ligne 11). L'arête entre `som 1` et `som 4` sera rouge (ligne 12).

Attributs sur les sommets

Attribut	Par défaut	Rôle
<code>color</code>	noir	couleur du contour
<code>fillcolor</code>	gris clair	couleur de remplissage
<code>fixedsize</code>	false	relation entre le texte et la taille du noeud
<code>fontname</code>	Times-Roman	police du texte
<code>fontsize</code>	14	taille de la police
<code>height</code>	0.5	hauteur en pouces
<code>label</code>		attribut une étiquette à un sommet
<code>orientation</code>	0.0	rotation du nœud
<code>pos</code>		fixe les coordonnées d'un sommet
<code>regular</code>	false	force un polygone à être régulier
<code>shape</code>	ellipse	style de boîte (<code>box,polygon,ellipse,circle,point,triangle,plaintext,diamond,trapezium,parallelogram,house,hexagon,octagon,doublecircle, doubleoctagon,invtriangle,invtrapezium,invhouse,Mdiamond,Msquare, Mcircle,record,Mrecord</code>)
<code>sides</code>	4	nombre de côtés pour <code>shape=polygon</code>
<code>style</code>		options graphiques (<code>solid,dashed,dotted,bold,invis, filled,diagonals,rounded</code>)
<code>width</code>	0.75	largeur en pouces

Le logiciel dot ne prend pas en compte l'attribut `pos` et l'ignore. Pour représenter un graphe dont les coordonnées des sommets sont fixées, on utilisera l'utilitaire `neato`. Les coordonnées sont données entre double-cotes. La ligne

`“sommets 1” [pos = “1,2 !”] ;`

signifie que le sommet 1 aura pour coordonnées 1 et 2.

Attributs sur les arêtes

Attribut	Par défaut	Rôle
<code>arrowhead</code>	<code>normal</code>	flèches (<code>normal, dot, odot, inv, invdot, invodot, none</code>)
<code>color</code>	<code>black</code>	couleur du trait
<code>fontcolor</code>	<code>black</code>	couleur de l'étiquette
<code>fontsize</code>	<code>14</code>	taille de l'étiquette
<code>id</code>		distinguer des arêtes multiples
<code>label</code>		étiquette
<code>style</code>		options graphiques (<code>solid, dashed, dotted, bold, invis</code>)
<code>weight</code>	<code>1.0</code>	épaisseur

Différentes représentations

GraphViz contient plusieurs programmes utilisant des algorithmes différents de représentation de graphes.

<code>dot</code>	Algorithme de Sugiyama	représentation en couches
<code>neato</code>	Algorithme de Kamada-Kawai	
<code>fdp</code>	Algorithme de Fruchterman-Reingold	similaire à <code>neato</code>
<code>twopi</code>	Algorithme de Willis	représentation radiale
<code>circo</code>	Algorithme de Six et Tollis et de Kaufmann et Wiese	représentation circulaire

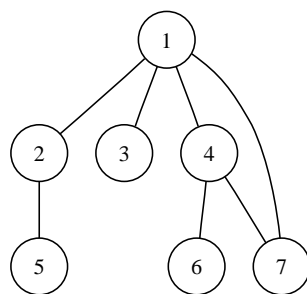
Ces différents utilitaires donnent des représentations différentes des graphes. Les figures suivantes donnent les représentations possibles d'un même graphe non orienté.

```

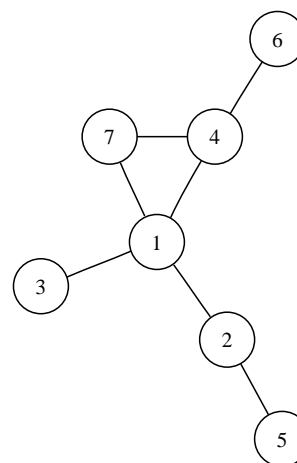
1: graph G {
2:   1 -- 2 -- 5;
3:   1 -- {3;4};
4:   4 -- {6;7};
5:   7 -- 1;
6: }

```

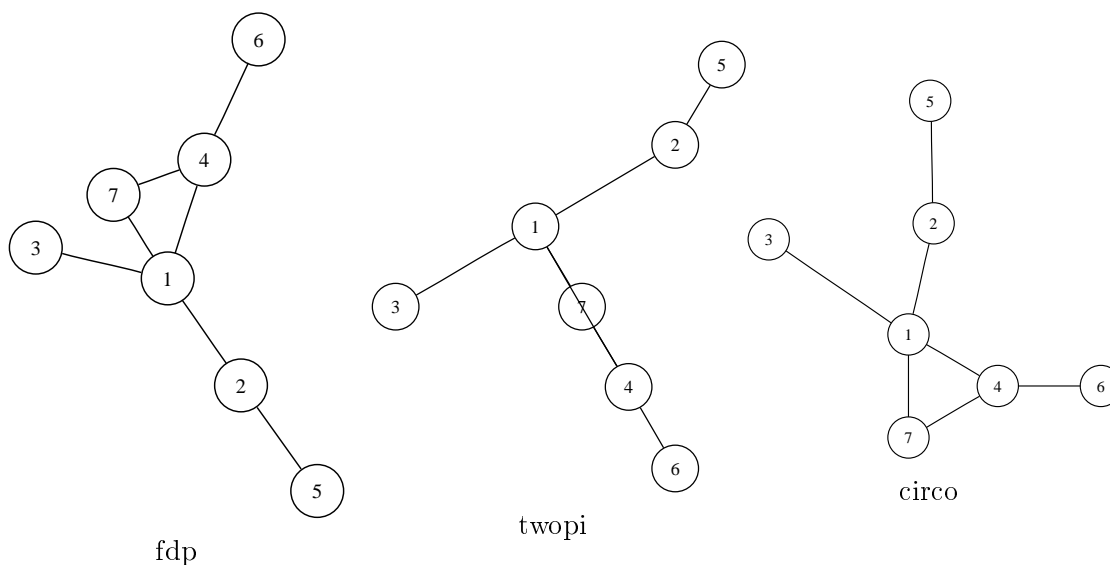
G.dot



dot



neato



Format de sortie

Par défaut, le format de sortie est le format `.dot`. Graphviz supporte de nombreux formats de sorties qui sont définis dans la ligne de commande par l'option `-Tformat` où *format* peut prendre, entre autres, les valeurs `dot`, `fig`, `gd`, `gif`, `imap`, `jpg`, `plain`, `png`, `ps`, `ps2` et `svg`.

Lancement de Graphviz

Pour lancer `dot` sur le fichier `graphe.dot` et obtenir par exemple une représentation du graphe en fichier `.ps` qui pourra être lu par un visualiseur postscript, il faut utiliser la ligne de commande suivante :

```
dot -Tps -o graphe.ps graphe.dot
```

`dot` peut être remplacé par un des autres utilitaires : `neato`, `fdp`, `twopi`, `circo`.

Graphviz à partir d'un programme C/C++

Pour utiliser Graphviz à partir d'un programme C/C++, il suffit de créer le fichier `graphe.dot` contenant le graphe. Puis, il faut lancer la commande système suivante :

```
system("dot -Tps -o graphe.ps graphe.dot");
```

pour obtenir par exemple un fichier postscript.

L'instruction `system` est déclarée dans le fichier d'en-tête standard `stdlib.h`.

Visualisation

Graphviz contient également deux utilitaires de visualisation de graphes appelés `dotty` et `lefty` qui permettent de modifier la représentation graphique.