Graphes et outils logiques
http://www.lri.fr/~blsk/GOL

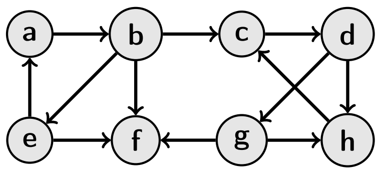
TD4 - Graphes, arbres, chemins, composantes

Exercice 1 *Parcours en profondeur*
Voici un algorithme de parcours en profondeur pour des graphes orientés. L'algorithme manipule une pile avec les notations suivantes : `[]` désigne la pile vide, `[p,s]` la pile formée en plaçant l'élément `s` au sommet de la pile `p`, et `[s]` la pile contenant le seul élément `s`.

```
exploration s:
    en_cours := []
    visiter s

visiter s:
    si s est déjà marqué:
        ne rien faire
    sinon:
        en_cours := [en_cours, s]
        marquer s
        pour chaque successeur s' de s:
            visiter s'
        [restants, _] := en_cours
        en_cours := restants
```

1. Exécuter cet algorithme sur le graphe suivant.



2. Proposer une spécification de l'effet de `visiter s` sur la pile `en_cours`. Justifier que cette spécification est correcte et dire quel est le sommet retiré du sommet de la pile par les instructions

```
[restants, _] := en_cours
en_cours := restants
```

3. Montrer qu'à tout moment de l'exécution de `exploration s`, les sommets contenus dans la pile `en_cours` décrivent un chemin dans le graphe exploré.

4. On propose la technique suivante pour tester la présence d'un cycle :
Si au cours de la visite d'un sommet `s` *le test* `si s est déjà marqué:` *s'évalue à vrai, alors on déclare que le graphe contient un cycle passant par* `s`*. Si en revanche le parcours se termine sans que ce cas se soit produit alors on déclare que la graphe de contient pas de cycle.*
Montrer que ce critère est invalide et le corriger.

Exercice 2 *Algorithme de Kosaraju-Sharir pour les composantes fortement connexes*
Étant donné un graphe orienté $G = (S, A)$ à n sommets, on cherche à identifier l'ensemble de ses composantes fortement connexes. On appellera *halo* d'une composante fortement connexe C de G et on notera $H(C)$ l'ensemble de sommets $s \in S \setminus C$ tels qu'il existe une arête d'un sommet de C vers s.

1. Supposons qu'on connaisse une numérotation de 1 à n des sommets de G (sommets qu'on pourra renommer en conséquence s_1 à s_n) telle que, pour toute composante fortement connexe C de G, tous les sommets de $H(C)$ ont des numéros strictement plus petits que tous les sommets de C.

   (a) Montrer que tous les sommets visités lors d'un parcours en profondeur effectué en partant du sommet s_1 appartiennent à la composante fortement connexe de s_1. (*Indice : raisonner par l'absurde*.)

   (b) En déduire que l'ensemble S_1 des sommets visités lors d'un parcours en profondeur effectué en partant du sommet s_1 est la composante fortement connexe de s_1.

   (c) En déduire un algorithme calculant l'ensemble des composantes fortement connexes de G (la numérotation étant supposée connue).

2. Pour établir une numérotation telle qu'utilisée à l'étape précédente, on propose l'algorithme suivant :

```
numéroter G:
    numéro courant := n
    pour chaque sommet s:
        visiter s
```

   où l'algorithme `visiter s` est le suivant :

```
visiter s:
    si s est déjà marqué:
        ne rien faire
    sinon:
        marquer s
        pour chaque prédécesseur p de s:
            visiter p
        affecter à s le numéro courant
        décrementer le numéro courant de 1
```

   (a) Cet algorithme *ressemble* à un parcours de graphe classique, mais quelle est la différence ?

   (b) Justifier que chaque numéro de n à 1 est affecté à un et un seul sommet.

   (c) Justifier que pour tous sommets s et s' tels qu'il existe une arête de s vers s', l'une des deux propriétés suivantes est vérifiée :
   — le numéro de s est supérieur au numéro de s', ou
   — il existe un chemin de s' à s.

   (d) En déduire que la numérotation produite par l'algorithme a bien la propriété supposée lors de la première question.

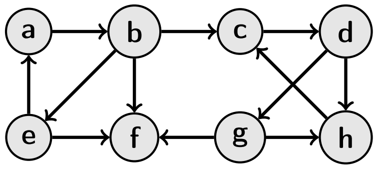
Exercice 3 *Graphe des composantes*
Étant donné un graphe $G = (S, A)$, on définit son *graphe des composantes* $C(G)$ comme suit :

— les sommets de $C(G)$ sont les composantes fortement connexes de G,

— il y a un arc de $C(G)$ du sommet C_1 vers le sommet C_2 si et seulement s'il existe dans G : un sommet s_1 dans la composante C_1, un sommet s_2 dans la composante C_2, un arc de s_1 à s_2.

On rappelle que la composante fortement connexe $C(s)$ d'un sommet s de G est l'ensemble des sommets s' de G tels qu'il existe dans G un chemin de s à s' et un chemin de s' à s.

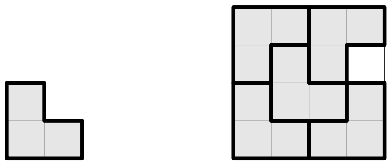
1. Identifier les composantes fortements connexes du graphe suivant, et donner son graphe des composantes.



2. Démontrer que si s_1 et s_2 appartiennent tous deux à la composante fortement connexe d'un même sommet s, alors il existe un chemin de s_1 vers s_2.
3. Soit un graphe G et son graphe des composantes $C(G)$. Démontrer par récurrence que pour tout k, s'il existe un chemin de longueur k dans $C(G)$ d'un sommet C_1 à un sommet C_2, alors pour tous sommets s_1, s_2 de G tels que $s_1 \in C_1$ et $s_2 \in C_2$, il existe dans G un chemin de s_1 à s_2 de longueur supérieure ou égale à k.
4. Démontrer qu'un graphe des composantes est toujours acyclique.
5. Peut-on en déduire que le graphe des composantes est un arbre ?

Exercice 4 *Récurrence sur les entiers*
Un triomino est une pièce formée de trois cases disposées en angle (voir figure 1(a)). On considère la propriété suivante : si on retire une case quelconque à un quadrillage de côté 2^n, $n \geq 1$, il est toujours possible de paver le reste du quadrillage avec des triominos. Montrer cette propriété par récurrence sur n.

FIGURE 1 – (a) Un triomino. (b) Pavage par des triominos d'un quadrillage de côté 4 privé d'une case.



(a) (b)