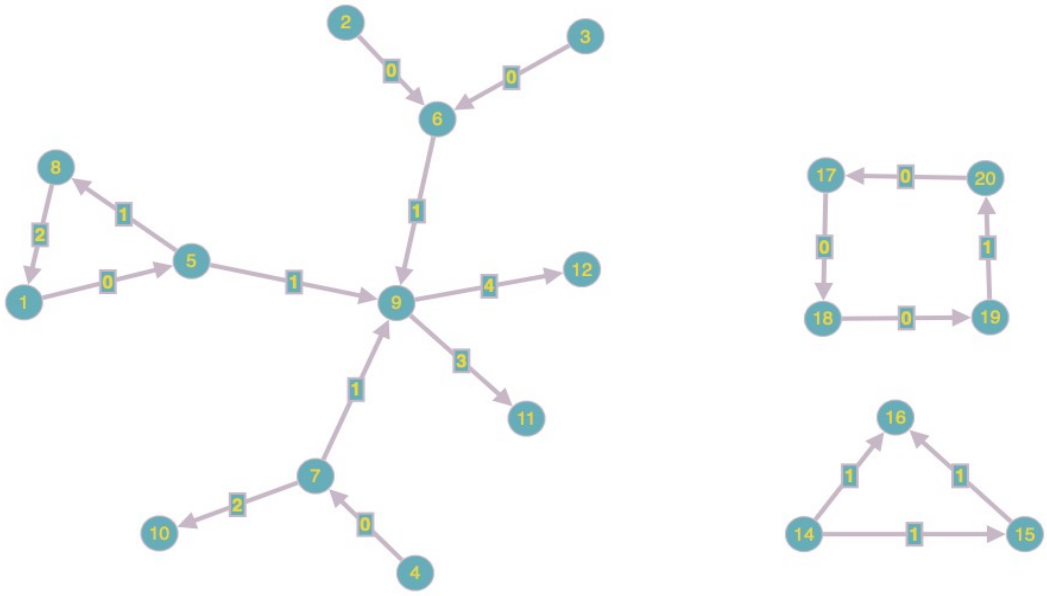
TP Scala

Exercice 1

Vous travaillez sur un document structuré, un graphe orienté labellisé.

On considère l'ensemble de triplets suivants :
((1,0,5),(5,1,8),(8,2,1),(2,0,6),(3,0,6),(6,1,9),(5,1,9),(9,3,11),(9,4,12),(4,0,7),(7,1,9),(7,2,10),
(14,1,15),(15,1,16),(14,1,16),(17,0,18),(18,0,19),(19,1,20),(20,0,17))

La structure de chaque triplet correspond à l'identifiant d'un nœud du graphe (le noeud émettant un arc – on parle de sujet du triplet), le label de l'arc et l'identifiant d'un nœud du graphe (recevant l'arc, on parle d'objet du triplet).
Cet ensemble de triplets correpsond donc au graph suivant :



1.1. Chargement des données

Vous devez charger le graphe dans une liste de tuples et structurer ce dernier sous la former (Int,Int,Int). Ensuite, vous créez une nouvelle liste qui ne comportera que les paires (sujet, objet), par exemple les deux premiers triplets deviendront (1,5),(5,8) .

1.2. Extraire les racines du graphe

Vous créez une liste contenant uniquement des nœuds racines du graphe, c'est-à- dire les nœuds qui n'ont pas d'arcs entrants. Exemple, les nœuds 2,3.

1.3. Obtenir les feuilles du graphe

Vous créez une liste contenant uniquement des nœuds feuilles du graphe, c'est-à- dire les nœuds qui n'ont pas d'arcs sortants. Exemple, les nœuds 10,11.

1.4. Fermeture transitive

Vous créez une nouvelle liste qui contiendra l'ensemble des nœuds accessibles à partir de chaque nœud du graphe. Cela correspond à la fermeture transitive du graphe. Exemple, étant donné a → b → c, nous nous retrouverons avec (a,b),(b,c), que nous avions déjà, et (a,c).

1.4.a Pour calculer la fermeture transitive, vous aurez besoin d'un algorithme de jointure. Rédiger un algorithme de jointure, du type sort merge join, en respectant la signature suivante :
def join(pair1 : List[(Int,Int)], pair2: List[(Int,Int)]) : List[(Int,Int)] = ???
NB : Attention aux duplicats à la position de clé.

1.4.b Rédiger une implantation itérative du calcul de la fermeture transitive

1.5. Présentation des paires inférées

Vous affichez, dans un ordre trié sur la valeur du sujet, uniquement les paires (sujet,objet) qui ont été ajoutées lors du calcul de la fermeture transitive (c'est-à-dire les paires qui n'étaient pas à l'origine dans le graphe).

1.6. rootedGraph

Créer une nouvelle liste, nommée rootedGraph, qui contient l'ensemble des nœuds accessibles à partir d'une racine. La liste résultante doit ressembler à un tuple avec la racine en première position et une liste triée de tous les nœuds accessibles en deuxième position.

1.7. Version récursive du calcul de la fermeture transitive