M4104C - Algorithmique distribuée

TD3

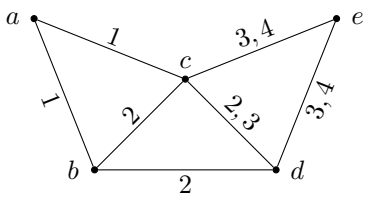
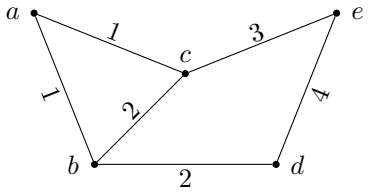
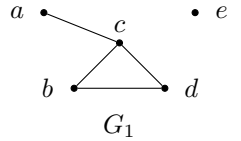
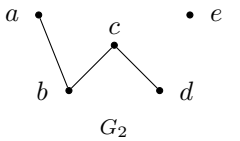
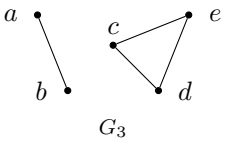
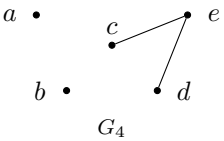
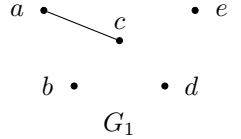
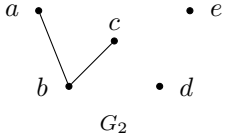
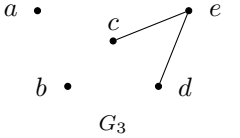
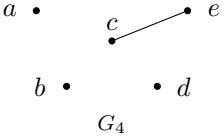
Réseaux dynamiques

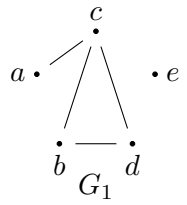
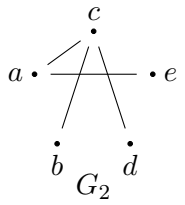
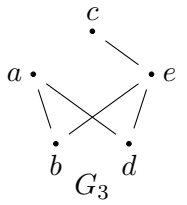
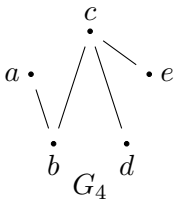
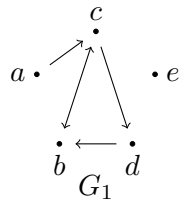
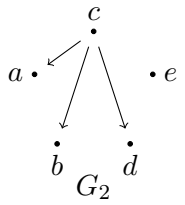
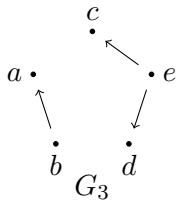
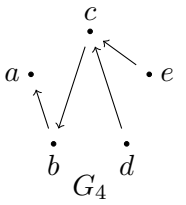
1 Quelques familles de graphes

Voici pour rappel quelques familles de graphes basées sur des propriétés temporelles basiques. Elles ont ici des noms plus explicites que précédemment ($\mathcal{C}_1 \dots \mathcal{C}_7$), en se basant sur $\mathcal{J}$ pour journey (trajet), $\mathcal{TC}$ pour temporal connectivity (connexité temporelle) et $\mathcal{E}$ pour edge (arête).

- $\mathcal{J}^{1\forall}$: Au moins un sommet peut joindre tous les autres par un trajet
- $\mathcal{J}^{1\forall>}$: Au moins un sommet peut joindre tous les autres par un trajet strict
- $\mathcal{J}^{\forall 1}$: Au moins un sommet peut être joint depuis tous les autres par un trajet
- $\mathcal{TC}$: Le graphe est temporellement connexe ($= \mathcal{J}^{\forall\forall}$)
- $\mathcal{TC}^{>}$: Le graphe est temporellement connexe par des trajets stricts ($= \mathcal{J}^{\forall\forall>}$)
- $\mathcal{E}^{1\forall}$: Au moins un sommet a une arête (non-orientée) avec chaque autre au moins une fois
- $\mathcal{K}$: Les sommets partagent deux à deux une arête au moins une fois ($= \mathcal{E}^{\forall\forall}$)

2 Quelques graphes

$\mathcal{G}_1$				
$\mathcal{G}_2$				
$\mathcal{G}_3$	 G_1	 G_2	 G_3	 G_4
$\mathcal{G}_4$	 G_1	 G_2	 G_3	 G_4

$\mathcal{G}_5$	 G_1	 G_2	 G_3	 G_4
$\mathcal{G}_6$	 G_1	 G_2	 G_3	 G_4

2.1 Exercice 1 : trouver les inclusions

Cochez les cases correspondant à des inclusions valides d'un graphe dans une famille de graphe.

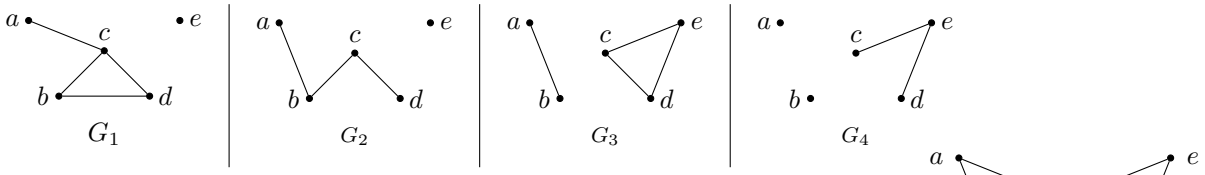
	$\mathcal{J}^{1\forall}$	$\mathcal{J}^{1\forall>}$	$\mathcal{J}^{\forall 1}$	$\mathcal{TC}$	$\mathcal{TC}^{>}$	$\mathcal{E}^{1\forall}$	$\mathcal{K}$
$\mathcal{G}_1$							
$\mathcal{G}_2$							
$\mathcal{G}_3$							
$\mathcal{G}_4$							
$\mathcal{G}_5$							
$\mathcal{G}_6$							

3 Quelques constructions utiles

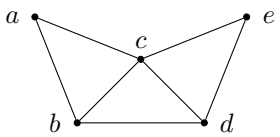
3.1 Empreinte du graphe

Soit un graphe dynamique $\mathcal{G} = \{G_1, ..., G_k\}$, on appelle *empreinte* de $\mathcal{G}$ le graphe (statique) $G = (V, E)$ tel que $V = \cup_i V_i$ et $E = \cup_i E_i$. Dans notre cas, seul l'ensemble des arêtes change, mais la définition est générale.

Exemple : le graphe dynamique suivant...

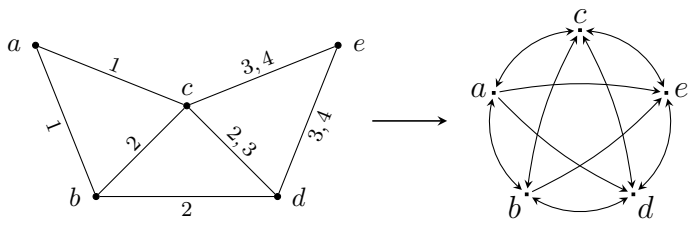


... a pour empreinte le graphe (statique) suivant



3.2 Fermeture transitive des trajets (stricts et non-stricts)

Soit un graphe dynamique $\mathcal{G} = \{G_1, ..., G_k\}$, orienté ou non, on appelle *fermeture transitive [des trajets]* de $\mathcal{G}$ le graphe orienté $\vec{G} = (V, \vec{E})$ tel que $V = \cup_i V_i$ et $(u, v) \in \vec{E}$ si et seulement si $u \rightsquigarrow v$ (*c.à.d.*, u peut joindre v par un trajet). On distinguera ici deux versions de la fermeture transitive, selon que les trajets considérés sont contraints d'être stricts ou non. La figure ci-dessous donne un exemple de fermeture transitive *stricte* pour un graphe donné.



3.3 Exercice : trouver l'empreinte et la fermeture transitive

Dessinez ci-dessous l'empreinte et les fermetures transitives des graphes précédents.

	Empreinte	Fermeture transitive stricte	Fermeture transitive non-stricte
$\mathcal{G}_1$			
$\mathcal{G}_2$			
$\mathcal{G}_3$			
$\mathcal{G}_4$			
$\mathcal{G}_5$			
$\mathcal{G}_6$			

4 Algorithmes de tests

On souhaite tester automatiquement si un graphe dynamique donné $\mathcal{G}$ appartient à telle ou telle famille, par exemple parmi $\mathcal{J}^{1\forall}$, $\mathcal{J}^{1\forall>}$, $\mathcal{J}^{\forall 1}$, $\mathcal{TC}$, $\mathcal{TC}^{>}$, $\mathcal{E}^{1\forall}$, et $\mathcal{K}$.

4.1 Observations préliminaires

On observe les équivalences suivantes :

— $\mathcal{G} \in \mathcal{E}^{1\forall} \Leftrightarrow$ L'empreinte de $\mathcal{G}$ contient un sommet universel (sommet relié à tous les autres par une arête non-orientée)
— $\mathcal{G} \in \mathcal{K} \Leftrightarrow$ L'empreinte de $\mathcal{G}$ est un graphe complet
— $\mathcal{G} \in \mathcal{J}^{1\forall} \Leftrightarrow$ La fermeture transitive de $\mathcal{G}$ contient un sommet universel sortant (sommet relié à tous les autres par une arête sortante)
— $\mathcal{G} \in \mathcal{J}^{\forall 1} \Leftrightarrow$ La fermeture transitive de $\mathcal{G}$ contient un sommet universel entrant (l'inverse)
— $\mathcal{G} \in \mathcal{TC} \Leftrightarrow$ La fermeture transitive de $\mathcal{G}$ est un graphe complet
— $\mathcal{G} \in \mathcal{J}^{1\forall >} \Leftrightarrow$ La fermeture transitive stricte de $\mathcal{G}$ contient un sommet universel sortant
— $\mathcal{G} \in \mathcal{TC}^{>} \Leftrightarrow$ La fermeture transitive stricte est un graphe complet

Ces observations nous motivent à concevoir des algorithmes pour construire automatiquement l'empreinte et les deux fermetures transitives d'un graphe dynamique.

4.2 Exercice : algorithmes de construction

Nous nous intéresserons en travaux pratiques à la construction automatique des trois objets présentés ci-dessus. Concevez (dans les grandes lignes) les algorithmes que vous coderez alors pour construire chacun des trois objets, dans l'ordre de difficulté croissant suivant :

1. L'empreinte
2. La fermeture transitive des trajets stricts
3. La fermeture transitive des trajets non-stricts

Vous pouvez supposer que votre algorithme est notifié lors des apparitions et disparitions de liens.

5 Pour aller plus loin...

La notion de fermeture transitive des trajets a été introduite dans [1]. Les familles de graphes ci-dessus ont été introduites dans [3]. Il en existe de nombreuses autres identifiées à travers des travaux variés. Une synthèse de cette problématique est proposée dans la Section 3.3 de [2].

Références :

[1] S. Bhadra and A. Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *Proceedings of the 2nd International Conference on Ad Hoc, Mobile and Wireless Networks (AdHoc-Now)*, 2003.

[2] A. Casteigts. *A Journey Through Dynamic Networks (with Excursions)*. Habilitation à diriger des recherches, University of Bordeaux, June 2018.

[3] A. Casteigts, S. Chaumette, and A. Ferreira. Characterizing topological assumptions of distributed algorithms in dynamic networks. In *Proc. of 16th Intl. Conference on Structural Information and Communication Complexity (SIROCCO)*, 2009.