

Modélisation et visualisation de graphes dynamiques pour la réécriture de graphes

Encadrants : Bruno Pinaud (bruno.pinaud@labri.fr) et Guy Melançon (guy.melancon@labri.fr)

Equipe d'accueil : MaBioVis, thèmes Visualisation, équipe projet INRIA GRAVITE

Lieu : LaBRI

Financement : possible en partie sur un projet ANR

Description du sujet

Le stage concerne la visualisation de graphes dynamiques [1] apparaissant dans le cadre de systèmes de réécriture de graphes. La réécriture de graphes apparait aujourd'hui dans des domaines d'applications divers tel que les réseaux d'interactions moléculaires où les sommets des graphes représentent des molécules qui interagissent les unes avec les autres au gré de réactions biochimiques décrites par des règles de réécriture. Ces règles permettent de tenir compte de la réalité biochimique et de contraindre les façons dont les molécules peuvent s'agencer. Pour plus d'informations voir le site du projet PORGY (http://gravite.labri.fr/?Projects_%2F_Grants:Porgy, Fig. 1) qui sert de contexte de travail pour ce stage.

Nous nous intéresserons plus particulièrement au graphe qui évolue au fur et à mesure de l'application des règles par l'utilisateur. Ce graphe est un graphe dynamique car l'application des règles entraînent des évolutions de sa topologie ou des propriétés associées à ses éléments. L'objectif est alors de proposer à l'utilisateur des dessins consécutifs du graphe qui restent lisibles et compréhensibles mais qui soient le plus stable possible. Le cas que nous considérons ici est à notre connaissance nouveau et à plusieurs titres :

- La dynamique des graphes n'est pas liée à une suite de modifications connues à l'avance. C'est l'application de règles qui régit les modifications possibles. De plus, ces modifications sont pilotées par l'utilisateur donc on ne peut donc pas connaître *a priori* comment va évoluer le graphe.
- La visualisation du système requiert non seulement de pouvoir visualiser le graphe à un moment donné et à suivre ses évolutions, mais elle exige aussi de pouvoir visualiser l'ensemble des réécritures possibles vers lesquelles peuvent évoluer ce graphe si on s'intéresse, par exemple, à la confluence des règles. L'environnement de visualisation doit alors offrir à l'utilisateur une vue sur l'historique des réécritures. Ces opérations débordent hors du cadre strict du dessin de graphes et exigent de mobiliser les techniques et algorithmes de recherche de motifs sur les graphes.
- Les graphes que nous manipulons sont le plus souvent issus de données bio-informatique. Ils ont la plupart du temps plusieurs composantes connexes. A notre connaissance, les algorithmes publiés prennent rarement correctement en charge ce type de graphes. Par souci de simplicité, les auteurs font l'hypothèse que les graphes ne possèdent qu'une seule composante connexe.

Pour dessiner correctement les graphes, les algorithmes de dessins de graphes dynamiques les plus connus ne sont pas utilisables en l'état. Par exemple GraphAEL [2] calcule tous les dessins de tous les états du graphe en même temps mais il nécessite en entrée de connaître la totalité des données. Les travaux de Frishman et Tal [4] ne traitent que le cas d'un graphe avec une seule composante connexe et de plus leur algorithme est difficile à reproduire. L'algorithme GEM est devenu un standard pour les graphes statiques [3] mais la gestion des composantes connexes n'est pas parfaite et son exécution reste assez lourde.

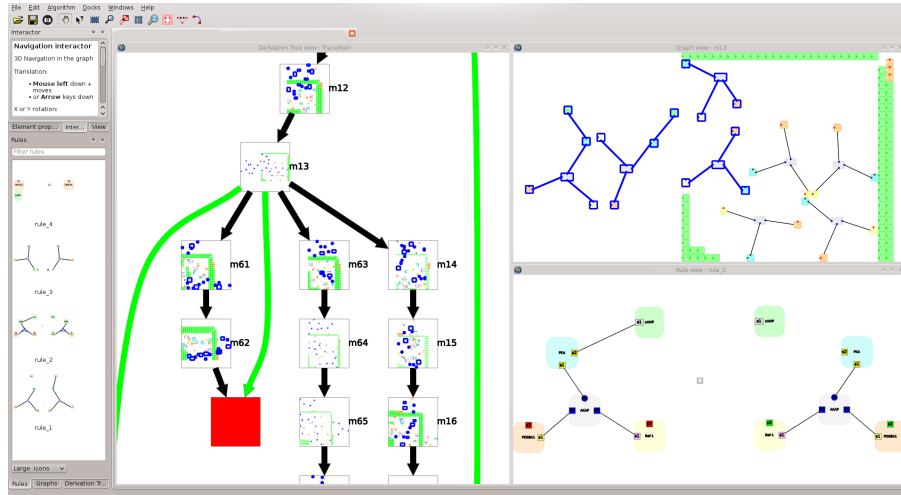


FIGURE 1 – Illustration de notre outil de réécriture de graphes. En bas à droite de la figure se trouve une règle de réécriture ; en haut à droite se trouve un état du graphe en cours de réécriture ; la partie centrale illustre l'historique des réécritures ; la partie gauche de la figure montre quelques contrôles de l'utilisateur sur le système.

En partant des travaux existants, l'objectif de ce stage est de développer un algorithme de visualisation adapté au dessins de graphes dynamiques liés aux contraintes de la réécriture de graphe. Dans la mesure du possible, l'algorithme pourra s'aider du dessin des règles (qui devront donc être "bien dessinées) pour positionner au mieux les sommets. *In fine*, l'objectif est de permettre à l'utilisateur de pouvoir raisonner visuellement [5] sur les dessins des graphes pour se concentrer au maximum sur sa tâche primaire qui est l'étude de son système de réécriture de graphe.

Références

- [1] J. Branke. Dynamic graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs : methods and models*, LNCS, pages 228–246. Springer, 2001.
- [2] C. Erten, P. J. Harding, S.G Kobourov, K. Wampler, and G. V. Yee. GraphAEL : Graph animations with evolving layouts. In G. Liotta, editor, *Proc. of Graph Drawing (GD '03)*, volume 2912 of LNCS, pages 98–110. Springer-Verlag, 2004.
- [3] Arne Frick, Andreas Ludwig, and Heiko Mehldau. A fast adaptive layout algorithm for undirected graphs. In Roberto Tamassia and Ioannis Tollis, editors, *Graph Drawing 94*, volume 894 of *Lecture Notes in Computer Science*, pages 388–403. Springer Berlin / Heidelberg, 1995.
- [4] Y. Frishman and A. Tal. Online dynamic graph drawing. *IEEE Trans. on Visualization and Computer Graphics*, 14(4) :727–740, 2008.
- [5] T. von Landesberger, A. Kujiper, T. Schreck, J. Kohlhammer, J.J. van Wijk, J.-D. Fekete, and D.W. Fellner. Visual analysis of large graphs : State-of-the-art and future research challenges. *Computer Graphics Forum*, 30(6) :1719–1749, September 2011.