

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)

Équipe d'accueil : MaBioVis, thèmes Visualisation

Lieu : LaBRI

Description du sujet

Ce sujet a pour base les systèmes de réécriture de graphes appliqués à la modélisation de systèmes complexes. Imaginez un jeu dans lequel des règles de transformation sont successivement appliquées sur un graphe (résultant de la modélisation d'un système) jusqu'à atteindre une condition d'arrêt. Une règle décrit un motif local qui doit être identifié (un sous-graphe) dans le graphe et comment transformer ce motif. La réécriture de graphes est utilisée dans de nombreux domaines notamment la simulation de l'évolution de réseaux d'interactions moléculaires ou la modélisation/vérification de logiciels [1].

Le formalisme de la réécriture de graphes est à la fois très riche et complexe rendant l'étude d'un système utilisant ce formalisme difficile. C'est ici que l'utilisation de la visualisation prend du sens. Elle rend intuitif et a fortiori plus facile le raisonnement et la manipulation du système étudié. Ces opérations débordent néanmoins hors du cadre strict de la visualisation et exigent notamment de mobiliser les techniques et algorithmes de recherche de motifs (isomorphisme) sur les graphes. Le travail s'appuiera sur les premiers résultats obtenus par l'équipe du thème visualisation implémentés dans le système visuel et interactif de réécriture de graphe PORGY¹ (Figure 1).

Nous nous intéresserons pour ce stage 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 [2] 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.
- 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 [3] 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 [5] ne traite que le cas d'un graphe avec une seule composante connexe et

1. <http://tulip.labri.fr/TulipDrupal/?q=porgy>

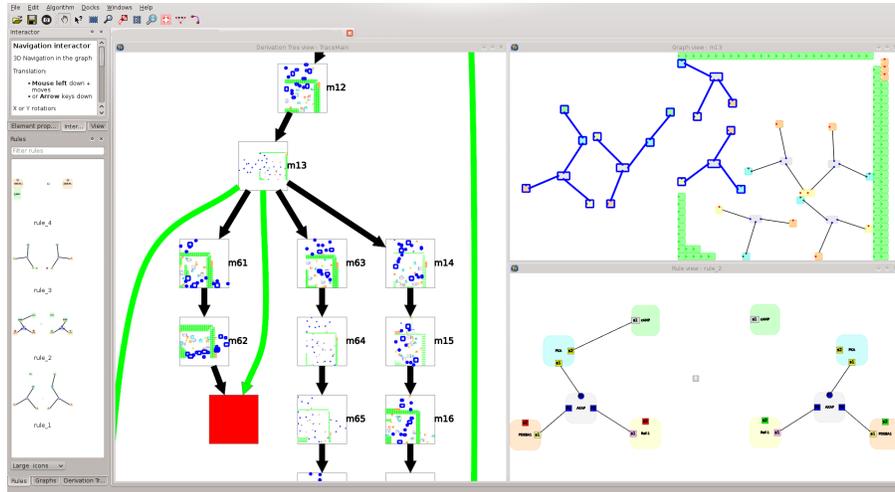


FIGURE 1 – Illustration de notre outil de réécriture de graphes. En bas à droite de la figure se trouve une règle de réécriture ; en haut à droite se trouve un état du graphe en cours de réécriture ; la partie centrale illustre l'historique des réécritures ; la partie gauche de la figure montre quelques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

de plus leur algorithme est difficile à reproduire. L'algorithme GEM est devenu un standard pour les graphes statiques [4] mais la gestion des composantes connexes n'est pas parfaite et son exécution reste assez lourde.

En partant des travaux existants, l'objectif de ce stage est de développer un algorithme de visualisation adapté au dessin 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 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] Oana Andrei, Maribel Fernández, Hélène Kirchner, Guy Melançon, Olivier Namet, and Bruno Pinnaud. PORGY : Strategy-Driven Interactive Transformation of Graphs. In Rachid Echahed, editor, *TERMGRAPH*, volume 48, pages 54–68, 2011.
- [2] J. Branke. Dynamic graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs : methods and models*, LNCS, pages 228–246. Springer, 2001.
- [3] 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.
- [4] Arne Frick, Andreas Ludwig, and Heiko Mehltau. 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.
- [5] Y. Frishman and A. Tal. Online dynamic graph drawing. *IEEE Trans. on Visualization and Computer Graphics*, 14(4) :727–740, 2008.