



HAL
open science

Préservation de la cohérence des transformations topologiques et géométriques

Valentin Gauthier, Thomas Bellet, Hakim Belhaouari, Agnès Arnould

► **To cite this version:**

Valentin Gauthier, Thomas Bellet, Hakim Belhaouari, Agnès Arnould. Préservation de la cohérence des transformations topologiques et géométriques. 15èmes journées Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL 2016), Jun 2016, Besançon, France. hal-01817884

HAL Id: hal-01817884

<https://hal.science/hal-01817884v1>

Submitted on 18 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Préservation de la cohérence des transformations topologiques et géométriques

Valentin Gauthier¹, Thomas Bellet², Hakim Belhaouari¹ et Agnès Arnould¹

¹ Université de Poitiers, XLim , ² École Centrale Paris, MICS

Résumé

Dans des travaux antérieurs, nous avons proposé un formalisme basé sur les transformations de graphes pour définir des opérations géométriques à base topologique. Une règle de transformation définit une opération sur une orbite topologique, telle que la triangulation d'une face, ou la translation d'une composante connexe. Elle définit les nouveaux plongements (dont la géométrie) à l'aide d'expressions de plongement. Des conditions syntaxiques garantissent la préservation de la cohérence topologique des objets d'une part et de leurs plongements d'autre part. Cet article présente l'utilisation conjointe des orbites topologiques et des expressions de plongement. Il étend les conditions syntaxiques pour garantir simultanément la cohérence de la topologie et des plongements.

Mots clés : transformation de graphes, modélisation géométrique à base topologique, conditions syntaxiques, préservation de la cohérence.

1 Introduction

Les modeleurs géométriques permettent de construire des objets complexes pour des applications variées (résistance des matériaux, écoulement de fluides, simulation chirurgicale, reconstruction du sous-sol...). Pour cela, ils s'appuient sur des modèles à base topologique décrivant les objets modélisés par leur structure topologique (leur décomposition en sommets, arêtes, faces, volumes, etc.) et leurs plongements (valeurs géométriques et physiques - par exemple les coordonnées des sommets, les couleurs des faces, la densité des volumes). Les opérations que subissent les objets lors d'une construction ou d'une simulation doivent préserver la cohérence des objets.

Des travaux antérieurs ont proposé l'utilisation des transformations de graphes pour définir les opérations sur les objets (*cf.* [Pou09] et [Bel12]). Ce formalisme offre la possibilité de définir de façon graphique et concise des opérations parfois complexes, rendant possible le prototypage rapide de nouvelles opérations. Il permet également de définir des conditions syntaxiques garantissant par construction la préservation de la cohérence des objets modélisés. Cet article présente les conditions à vérifier lorsque les opérations modifient à la fois la structure topologique et les plongements des objets.

Après une présentation du modèle des objets en section 2, nous présentons en section 3 les transformations topologiques et les conditions pour préserver la cohérence topologique. Symétriquement, la section 4 présente les transformations des plongements et les conditions associées. Nous combinons les deux aspects (topologie et plongement) dans la section 5 et donnons les conditions nécessaires à la préservation de la cohérence. Enfin, nous concluons dans la section 6.

2 Objets géométriques à base topologique

Le modèle topologique utilisé est celui des cartes généralisées [Lie94]. Cette structure topologique est issue de la décomposition des objets en cellules topologiques de dimensions décroissantes (sommets, arêtes, faces, volumes, etc.). Par exemple l'objet 2D en Figure 1a se décompose en faces (cellules de dimension 2) reliées entre elles par des arcs de dimension 2 (*cf.* Figure 1b). Puis les faces se décomposent en arêtes reliées par des arcs de dimension 1 (*cf.* Figure 1c). Enfin, la décomposition des arêtes en

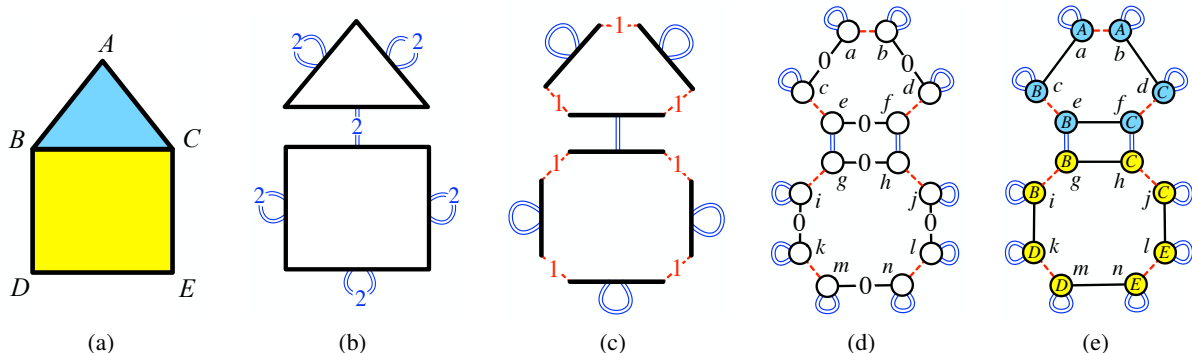


FIGURE 1 – Décomposition d'un objet géométrique 2D

sommets reliés par des arcs de dimension 0 (cf. Figure 1d) donne les nœuds du graphe. Ainsi les graphes manipulés ont leurs arcs étiquetés sur $[0, n]$ où n est la dimension topologique des objets représentés. Les objets pris pour exemple dans cet article étant de dimension 2, leurs arcs seront étiquetés sur $[0, 2]$ et seront représentés par le code couleur de la Figure 1.

Les cellules topologiques des objets sont définis par des sous-graphes appelés orbites :

Orbite : pour o un sous-ensemble de $[0, n]$ et v un nœud d'un graphe G de dimension n , l'orbite $\langle o \rangle(v)$ est le sous-graphe de G contenant v , les nœuds de G atteignables par des arcs étiquetés sur o , ainsi que ces arcs. On dit que $\langle o \rangle(v)$ est de type $\langle o \rangle$, ou est une $\langle o \rangle$ -orbite.

Par exemple sur la Figure 2, le sommet B est défini par le sous-graphe atteignable à partir du nœud e et des arcs de dimensions 1 et 2, c'est-à-dire l'orbite $\langle 1, 2 \rangle(e)$ de la Figure 2a. De même, l'arête BC est l'orbite $\langle 0, 2 \rangle(e)$ de la Figure 2b, et la face ABC est l'orbite $\langle 0, 1 \rangle(e)$ de la Figure 2c. La notion d'orbite n'est pas limitée aux cellules. Par exemples les Figures 2d et 2e présentent respectivement une demi-arête et une composante connexe.

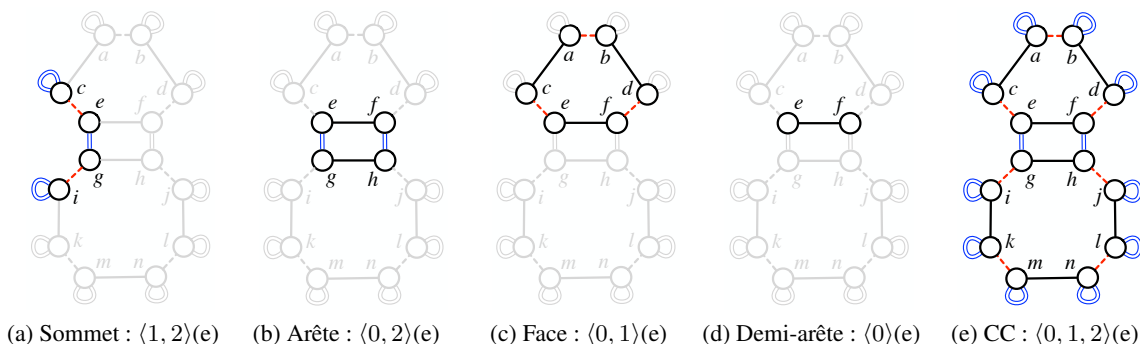


FIGURE 2 – Orbites incidentes à e

Les plongements sont pour leur part représentés par les étiquettes des nœuds du graphe. Par exemple, Figure 1e, les nœuds sont étiquetés par des points géométriques (A, B, C, D et E) et des couleurs (bleu et jaune). Chaque plongement se caractérise par le type de l'information (ici un point 2D et une couleur) et le type des orbites support du plongement (ici les sommets de type orbite $\langle 1, 2 \rangle$ ou $\langle 1, 2 \rangle$ -orbites et les faces de type orbite $\langle 0, 1 \rangle$ ou $\langle 0, 1 \rangle$ -orbites).

La cohérence d'un objet de dimension n est définie par les contraintes suivantes :

- **Non-orientation** : un objet est un graphe non orienté.
- **Arcs adjacents** : tout nœud est la source de $n + 1$ arcs étiquetés respectivement de 0 à n .
- **Cycle** : pour toutes dimensions i et j telles que $0 \leq i \leq i + 2 \leq j \leq n$, tout nœud est la source d'un cycle¹ étiqueté par iji .

1. Un arc ω a un nœud source $s(\omega)$, un nœud cible $t(\omega)$, et une étiquette $\alpha(\omega)$. Un chemin $\omega_1\omega_2\omega_3\omega_4$ est une suite d'arcs tel que $t(\omega_i) = s(\omega_{i+1})$ et est étiqueté par $\alpha(\omega_1)\alpha(\omega_2)\alpha(\omega_3)\alpha(\omega_4)$. Un cycle $\omega_1\omega_2\omega_3\omega_4$ est un chemin tel que $s(\omega_1) = t(\omega_4)$.

— **Plongements** : pour chaque plongement de support $\langle o \rangle$, tous les nœuds d'une même $\langle o \rangle$ -orbite sont étiquetés par la même valeur de plongement.

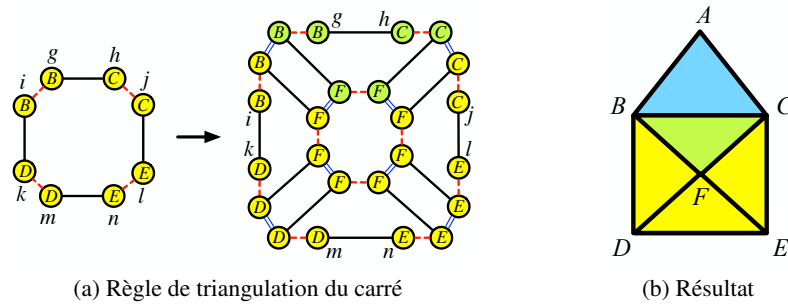


FIGURE 3 – Opération de triangulation

Les règles de transformation de graphe [EEPT06] permettent de définir des opérations sur les objets. Une règle de transformation de graphe $L \rightarrow R$ se compose de deux parties : le membre gauche L spécifie le motif filtré et le membre droit R spécifie le motif réécrit. Par exemple, la règle de la Figure 3a définit la triangulation du carré de l'objet de la Figure 1 dont le résultat est représenté en Figure 3b. Intuitivement, cette opération subdivise le carré en quatre triangles en introduisant un sommet F au barycentre de la face d'origine, et en coloriant les triangles créés en fonction des faces voisines (le triangle BCF est colorié en vert par mélange avec le bleu du triangle ABC alors que les autres, sans voisins, restent jaunes).

Cependant, une telle règle de transformation ne définit qu'une application donnée d'une opération de modélisation (dans le cas présent, la triangulation d'une face carrée jaune aux points B, C, D, E). Pour permettre une définition générique des opérations, les transformations de graphes permettent l'utilisation de variables [Hof05]. Dans les deux sections suivantes, nous présentons deux types de variables respectivement dédiées aux modifications de la topologie et des plongements.

3 Transformation topologique

Les variables topologiques permettent de définir génériquement les transformations topologiques. Elles sont basées sur la notion d'orbite. Par exemple, la triangulation d'une face quel que soit son nombre d'arêtes peut être définie par la règle avec variable topologique de la Figure 4a. Le membre gauche de la règle, effectuant le filtrage, possède désormais un seul nœud nommé v étiqueté par le type orbite face $\langle 0, 1 \rangle$. Ce nœud sera instancié par une face de l'objet à transformer, quelle qu'elle soit (par exemple un triangle ou un carré, cf. Figure 4b).

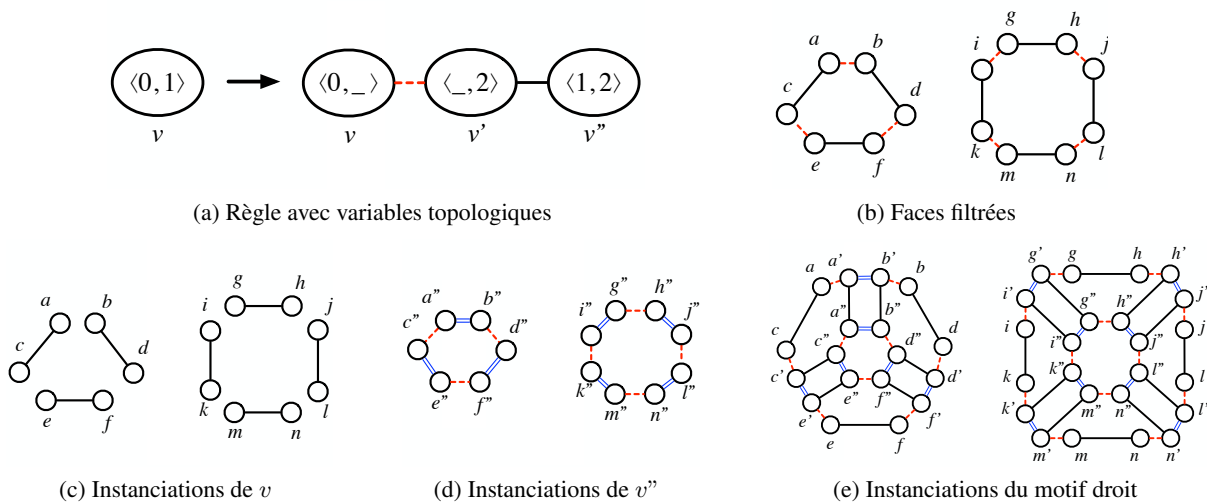


FIGURE 4 – Règle de triangulation

Dans le membre droit, chaque nœud abstrait une copie de l'orbite filtrée à renommage/suppression

des arcs près. Par exemple, le nœud v étiqueté par $\langle 0, _ \rangle$ signifie que les arcs de dimension 1 sont supprimés (notés ' $_$ ') dans l'instanciation (cf. Figure 4c) et donc que les arêtes seront séparées. De façon similaire, le nœud v'' est étiqueté par $\langle 1, 2 \rangle$. Par conséquent, tous les arcs de dimension 0 sont renommés en arcs de dimension 1, et tous les arcs de dimension 1 sont renommés en arcs de dimension 2, ce qui construit le sommet dual de la face triangulée, cf. Figure 4d. Enfin, notons que les arcs du motif droit permettent de relier deux à deux les copies instances des nœuds. Ainsi, la Figure 4e contient bien la face triangulée quelle que soit sa forme.

Afin de garantir la préservation de la cohérence topologique des objets lors de l'application d'une règle, des conditions syntaxiques sont définies dans [Pou09]. Par exemple, pour garantir la contrainte d'arcs adjacents, chaque nœud ajouté doit posséder toutes les dimensions. En particulier, le nœud v'' ajouté par la règle de la Figure 4a possède bien les trois dimensions 0, 1 et 2. La dimension 0 est portée par l'arc entre v' et v'' et relie dans l'objet les instances de v' à celles de v'' . Les dimensions 1 et 2 sont portées par le nœud lui-même et relient les instances de v'' entre elles par renommage des arcs filtrés.

4 Transformation des plongements

De façon similaire, une méthode est introduite dans [Bel12] pour calculer les nouveaux plongements dans une règle pour tous les objets transformés. Elle consiste à utiliser les nœuds de la partie gauche de la règle comme variables pour construire les expressions des nouveaux plongements. Ces expressions utilisent conjointement des opérateurs sur la structure topologique et sur les plongements.

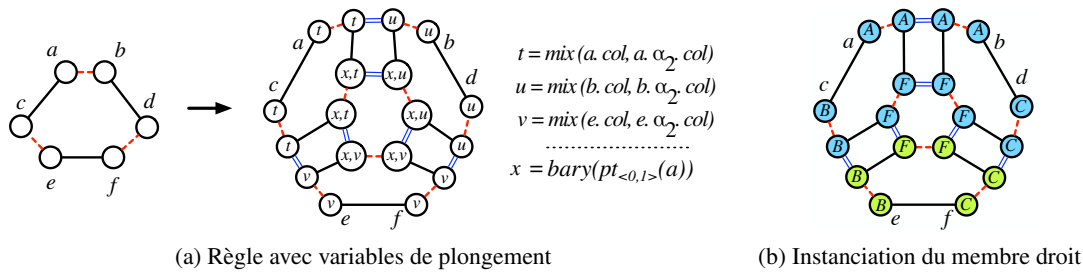


FIGURE 5 – Triangulation avec variables de plongement

Par exemple, dans la règle de la Figure 5a, la couleur du nœud a est réécrite à l'aide de l'expression de plongement $t = mix(a.col, a.\alpha_2.col)$. Outre l'accès aux plongements (dont $a.col$ qui permet d'accéder à la couleur du nœud a), les expressions contiennent différents opérateurs propres à la topologie, comme les opérateurs de voisinage ou les collectes. Par exemple, $a.\alpha_2$ permet l'accès au nœud voisin de a par l'arc de dimension 2 (unique grâce à la contrainte d'arc incident). Ou encore, $pt_{\langle 0,1 \rangle}(a)$ permet de collecter le multi-ensemble des plongements point de l'orbite face $\langle 0, 1 \rangle(a)$.

Les expressions de plongement permettent aussi de manipuler des opérateurs définis sur les types des plongements de l'utilisateur. Par exemple, mix représente un opérateur du type couleur qui permet de mélanger deux couleurs. Cet opérateur est ici utilisé par exemple sur a pour calculer sa nouvelle couleur comme le mélange de la sienne et celle de sa face voisine. De même, l'opérateur $bary$ permet de calculer le barycentre d'un ensemble de points, ici celui des points de la face incidente à a .

Les différentes expressions de plongements sont évaluées lors de l'instanciation de la règle avec variables de plongement (cf. Figure 5a), sur le triangle ABC de l'objet de la Figure 1. La Figure 5b présente l'instance du membre droit de la règle. Ainsi l'expression v sur la face incidente à e , a été évaluée en vert qui est le mélange de la couleur jaune de e et de la couleur bleu de son voisin de dimension 2, g . De même, l'expression t est évaluée en jaune car a est son propre voisin de dimension 2, et donc le jaune est mélangé avec lui-même.

Des conditions syntaxiques de préservation de la cohérence des plongements sont également définies dans [Bel12]. Par exemple, tous les nœuds d'une même orbite support de plongement doivent porter la même expression de plongement. Sur la Figure 5a par exemple, tous les nœuds de l'orbite face $\langle 0, 1 \rangle(e)$

portent bien la même expression de plongement couleur v . De même, ces conditions interdisent la redéfinition d'une valeur de plongement si toute son orbite support n'est pas filtrée. Dans le cas présent, la règle respecte cette condition pour la redéfinition de la couleur car toute la face est filtrée.

Montrons maintenant comment ces variables s'utilisent avec les variables topologiques.

5 Combiner les transformations de topologies et de plongements

Les variables topologiques et les variables de plongement peuvent être utilisées simultanément. Pour cela une instanciation en deux étapes est introduite dans [Bel12]. La première consiste à instancier les variables topologiques, et la deuxième à instancier les variables de plongement.

Par exemple, la Figure 6a illustre la règle de triangulation avec les deux types de variables. La première étape d'instanciation topologique, se déroule comme nous l'avons illustré en Figure 4, en ajoutant les expressions de plongement dans les nœuds (cf. Figure 6b). Ainsi par exemple, l'expression du plongement point $x = \text{bary}(pt_{\langle 0,1 \rangle}(v))$, portée par le nœud v'' de la règle de la Figure 6a, s'instancie en six expressions sur les six nœuds instances de v sur la Figure 6b. De même, l'expression du plongement couleur $u = \text{mix}(v.col, v.\alpha_2.col)$, portée par les trois nœuds v, v', v'' de la règle de la Figure 6a, s'instancie en six expressions pour les six nœuds instances de v et leurs copies instances de v' et v'' .

L'instanciation des variables de plongement décrite en section 4, appliquée à la règle de la Figure 6b, produit exactement la règle instanciée en Figure 5b. Ainsi son application sur le triangle bleu de l'objet de la Figure 1a produit l'objet de la Figure 6c.

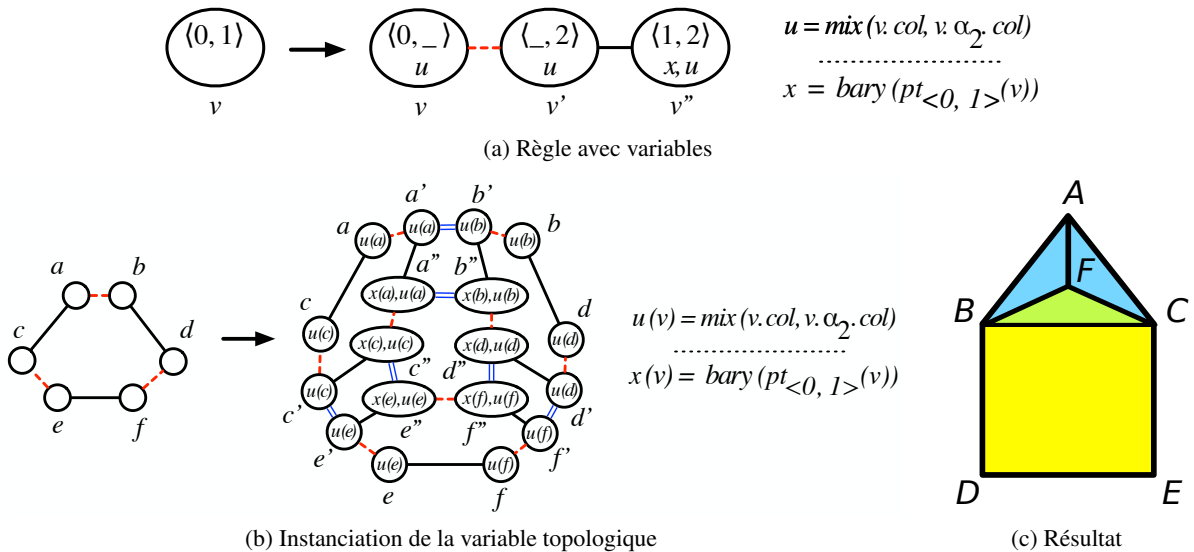


FIGURE 6 – Triangulation avec variables topologiques et variables de plongement

Les conditions syntaxiques de préservation de la cohérence topologique présentées en section 3, peuvent être utilisées telles quelles en présence simultanée de variables topologiques et de plongement. En effet, l'instanciation des expressions de plongement ne modifie pas la structure topologique.

Au contraire, les conditions syntaxiques de préservation des plongements présentées en section 4, ne peuvent être utilisées. En particulier une même expression de plongement dans la règle avec des variables topologiques, correspond à plusieurs expressions de plongement dans la règle instanciée. Par exemple, Figure 6b, les nœuds a et c portent deux expressions de couleurs différentes $u(a)$ et $u(c)$, alors qu'ils appartiennent à la même orbite face support du plongement couleur.

L'objectif de cet article est d'introduire des conditions de préservation des plongements sur les règles qui combinent variables topologiques et les variables de plongement. Ces conditions assurent la vérification des conditions syntaxiques introduites section 4 sur les règles avec variables de plongement, et donc la préservation de la cohérence des plongements lors de l'application des règles. Une équivalence à orbite près a été introduite dans [Bel12]. En particulier, dans notre exemple en Figure 6b, $u(a) \equiv_{\langle 0 \rangle} u(c)$, et

donc a et c portent la même couleur dans l’objet de la Figure 6c. Dans le présent article nous poursuivons cette approche. Nous définissons la notion de stabilité d’une expression le long d’une orbite sur un nœud, qui définit les conditions syntaxiques recherchées.

Reprenons l’exemple de la Figure 6, l’expression v est stable le long de $\langle 0, 1 \rangle$ sur le nœud v , car à gauche ce dernier est étiqueté par $\langle 0, 1 \rangle$ (tous les nœuds filtrés par v appartiendront ainsi à la même orbite $\langle 0, 1 \rangle$). Donc, $v.\alpha_2$ est stable le long de $\langle 0 \rangle$ sur le nœud v , car le 0202 vérifie la condition de cycle (deux instances de v , voisins de dimension 0, ont leurs voisins de dimension 2 respectifs liés par un arc de dimension 0). Donc, $v.\alpha_2.col$ est stable le long de $\langle 0 \rangle$ sur le nœud v , car $\langle 0 \rangle$ est une sous-orbite du support $\langle 0, 1 \rangle$ du plongement col (deux instances de v , voisins de dimension 0, ont leurs voisins de dimension 2 respectifs de la même couleur).

Si on poursuit le raisonnement, on peut déduire que $mix(v.col, v.\alpha_2.col)$ est stable le long de $\langle 0 \rangle$ sur le nœud v . Ce qui est l’une des conditions syntaxiques de préservation de la cohérence du plongement pour la règle de la Figure 6a. Elle garantit que a et c portent deux expressions $\langle 0 \rangle$ -équivalente dans la règle de la Figure 6b et que ces deux nœuds portent la même couleur dans l’objet transformé de la Figure 6c.

6 Conclusion

Dans cet article, nous avons présenté l’extension des conditions de préservation de la cohérence des plongements en cas de règles qui combinent variables topologiques et variables de plongement. Cette extension permettra la définition plus sûre et plus rapide d’opérations géométriques qui modifient simultanément la topologie et les plongements des objets (dont la géométrie).

Actuellement la bibliothèque Jerboa [BALGB14] permet déjà d’utiliser les variables topologiques et de plongement simultanément. Cependant, les expressions de plongement sont programmées directement dans le langage de programmation sous-jacent (Java ou C++), et seules la préservation de la cohérence topologique est vérifiée. Plus précisément, actuellement Jerboa ne vérifient pas les conditions syntaxiques de préservation des plongements présentés en section 5. Dans ce cas la règle instanciée porte différentes valeurs de plongement pour une même orbite de plongement. Lors de l’application, Jerboa calcule aléatoirement l’une de ces valeurs pour mettre à jour le plongement partagé de l’objet transformé. L’objet construit est donc cohérent, mais il ne correspond généralement pas au résultat attendu de l’opération.

La future version de Jerboa disposera d’un véritable langage d’expressions qui simplifiera l’écriture des règles et de leurs plongements. Surtout, elle vérifiera l’ensemble des conditions syntaxiques de préservation de la cohérence topologique et de plongement, ce qui facilitera la mise au point des règles.

Références

- [BALGB14] H. Belhaouari, A. Arnould, P. Le Gall, and T. Bellet. JERBOA : A graph transformation library for topology-based geometric modeling. In *7th International Conference on Graph Transformation (ICGT 2014)*, volume 8571 of *LNCS*, York, UK, July 2014. Springer.
- [Bel12] Thomas Bellet. *Transformations de graphes pour la modélisation géométrique à base topologique*. PhD thesis, 2012. 2012POIT2261.
- [EEPT06] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation (Monographs in Theoretical Computer Science. An EATCS Series)*. Springer, Secaucus, NJ, USA, 2006.
- [Hof05] Berthold Hoffmann. Graph transformation with variables. In *Formal Methods in Software and System Modeling (Festschrift for Hartmut Ehrig on the Occasion of his 60th Birthday)*, volume 3393 of *Lecture Notes in Computer Science*, pages 101–115. Springer-Verlag Heidelberg, 2005.
- [Lie94] P. Lienhardt. N-dimensional generalised combinatorial maps and cellular quasimanifolds. *International Journal of Computational Geometry and Applications*, 1994.
- [Pou09] Mathieu Poudret. *Graphs transformations for the topological operations of the geometric modeling - Application to the study of the Golgi apparatus dynamics*. PhD thesis, University of Evry-Val d’Essonne, October 2009.