



**HAL**  
open science

## Spécifications formelles du chanfreinage

Franck Ledoux, Jean-Marc Mota, Agnès Arnould, Catherine Dubois, Pascale Le Gall, Yves Bertrand

► **To cite this version:**

Franck Ledoux, Jean-Marc Mota, Agnès Arnould, Catherine Dubois, Pascale Le Gall, et al.. Spécifications formelles du chanfreinage. *Revue des Sciences et Technologies de l'Information - Série TSI: Technique et Science Informatiques*, 2002, 21 (8), pp.1-26. hal-00352083

**HAL Id: hal-00352083**

**<https://hal.science/hal-00352083>**

Submitted on 12 Jan 2009

**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.

# Spécifications formelles du chanfreinage

**Franck Ledoux\*** – **Jean-Marc Mota\*** – **Agnès Arnould\*\*** – **Catherine Dubois\*\*\*** – **Pascale Le Gall\*** – **Yves Bertrand\*\***

\* *LaMI, UMR 8042, Université d'Évry val d'Essonne  
Evry2, 523 place des Terrasses, 91000 Évry, France.  
{jmmota, fledoux, legall}@lami.univ-evry.fr*

\*\* *IRCOM-SIC, UMR 6615, Université de Poitiers  
SP2MI, 86962 Futuroscope Cedex, France.  
{arnould, bertrand}@sic.sp2mi.univ-poitiers.fr*

\*\*\* *CEDRIC, Institut d'Informatique d'Entreprise  
18 allée J. Rostand, 91025 Évry Cedex, France  
dubois@iie.cnam.fr*

*RÉSUMÉ. La représentation mathématique des objets géométriques, la complexité et le nombre d'algorithmes nécessaires à leur manipulation, sont des indices forts pour une utilisation aisée et bénéfique des méthodes formelles. Nous présentons dans cet article une étude de la spécification formelle d'une opération complexe et importante en modélisation géométrique. Il s'agit de l'opération de chanfreinage consistant à aplanir les angles vifs des objets 3D. Cette étude est menée dans le cadre de deux méthodes formelles, B (orientée modèles) et CASL (orientée propriétés) avec pour objectif de privilégier la lisibilité par les experts du domaine. De plus, la formalisation et la rétro-ingénierie que nous avons réalisées, nous permettent de jeter les bases d'une méthodologie dédiée.*

*ABSTRACT. The mathematical representation of geometric objects, the complexity and the number of algorithms necessary to handle them, make us think that formal methods will be useful and designed to this field. In this article, we study the formal specification of one of the most complex and important process in the field of geometric modelling: the chamfering one. To chamfer is to flatten 3D objects' angles. For this study, we have used two formal methods, the B method (model oriented) and CASL (property oriented) in order to make it readable by the think-tank of the concerned field. Moreover, as we have realized formalisation and reengineering we are able to lay foundations of a dedicated methodology.*

*MOTS-CLÉS : spécifications formelles, méthode B, spécifications algébriques, CASL<sup>1</sup>, modélisation géométrique, cartes généralisées de dimension n, chanfreinage.*

*KEYWORDS: formal specifications, B method, algebraic specifications, CASL, geometric modelling, n-dimensional generalized map, chamfering*

<sup>1</sup>. Le travail qui suit est en partie soutenu par le le groupe de travail ESPRIT 29432 (CoFI WG).

## 1. Introduction

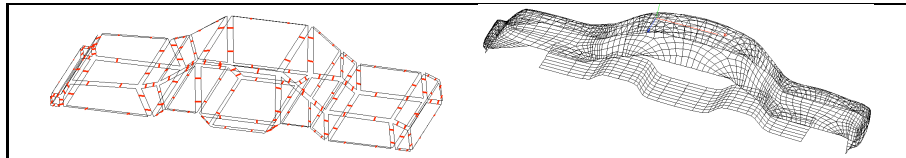
Les modeleurs géométriques sont des logiciels qui permettent de créer et de manipuler interactivement des objets géométriques complexes (en 3 dimensions généralement). Ces objets sont construits à partir de bibliothèques d'objets simples en appliquant diverses opérations de composition et/ou de modification. On distingue deux catégories d'opérations dans un modeleur :

- les opérations globales qui permettent de définir les formes principales des objets, et agissent sur l'ensemble des composantes de l'objet;
- les opérations locales qui permettent de préciser les détails de leurs formes en n'opérant que sur des parties restreintes de l'objet.

L'opération d'arrondi est une opération locale qui permet de remplacer un sommet ou une arête vive (ou une suite de sommets et d'arêtes) par une surface arrondie (ou un groupe de surfaces). Cette opération est essentielle pour construire de nombreux objets en les rendant plus réalistes (voir fig. 1). En outre, elle est complexe et les algorithmes proposés par les modeleurs commerciaux manquent de généralité : ils procurent un aspect de lissage insuffisant et sont même inapplicables sur certaines configurations d'objets. Une des raisons à cela est qu'une grande partie des modeleurs représentent les objets par leur géométrie, c'est-à-dire par les courbes et les surfaces qui composent la forme des objets. Or de telles représentations obligent à considérer une grande variété de cas pour l'arrondi en fonction de la géométrie locale de l'objet et à gérer des problèmes d'approximations numériques importants. On obtient ainsi fréquemment des définitions par cas qui deviennent obsolètes dès qu'un nouveau cas particulier est rencontré.

Une manière de représenter les objets consiste à mettre en évidence la topologie de l'objet, c'est-à-dire sa subdivision en sommets, arêtes, faces, etc. La géométrie (courbes et surfaces) d'un objet est alors obtenue en associant une géométrie à chaque élément de la subdivision. On associe par exemple une courbe à une arête et une surface à une face. Cette association est appelée un plongement. Cette représentation des objets permet de découper les opérations en deux parties, l'une topologique qui modifie la structure des objets, et l'autre géométrique qui modifie son plongement. En adoptant cette approche, les opérations de transformation des objets ont été décrites plus rigoureusement en définissant mathématiquement les résultats attendus des opérations, avant de proposer les algorithmes proprement dit. Les premières généralisations de l'opération d'arrondi sur une large classe d'objets [VÁR 94] utilisent une représentation des objets à base topologique. Mais elles ne sont pas encore satisfaisantes. L'arrondi n'a par exemple été traité que pour les objets 2D, ou plus exactement pour la surface extérieure d'objets 3D. Son extension à des objets dont la structure topologique est de dimension 3, ie qui est composé de plusieurs volumes, n'est pas immédiate car de nombreux problèmes de continuité apparaissent.

L'utilisation des méthodes formelles permet aux infographistes de poursuivre leur effort de formalisation tout en limitant son coût grâce aux différents outils. Par exemple, le prototypage rapide permet de visualiser, et donc de valider, le résultat des définitions



**FIG. 1.** Une carrosserie de voiture obtenue en arrondissant les angles et arêtes d'un objet brut.

sur de petits objets ; les raffinements permettent d'assurer la correction des algorithmes par rapport aux définitions, etc. Pour qu'une telle approche soit réaliste dans un domaine non sécuritaire mais concurrentiel comme l'infographie, elle doit s'appuyer sur une méthode et un atelier performant<sup>1</sup>. Ces derniers doivent comprendre : un langage adapté au domaine, des bibliothèques riches des différents objets et opérations classiques, des outils adaptés et interfacés avec les visualisateurs d'objets. Notons que la modélisation géométrique représente un domaine atypique pour l'utilisation de spécifications formelles. En effet, on manipule des structures de données et des algorithmes complexes alors qu'habituellement les méthodes formelles sont utilisées pour résoudre des problèmes de conception et de modélisation aboutissant à des structures simples.

Cet article traite du premier point : la recherche d'un langage de spécification formelle adapté au domaine de la modélisation géométrique à base topologique. Pour ce faire, nous proposons un travail de rétro-ingénierie des opérations locales de couture et de chanfreinage à l'aide des langages B [ABR 91, ABR 96] et CASL [LAN 00, MOS 99]. L'opération de couture consiste à coller deux objets entre eux. On collera par exemple deux cubes entre eux le long d'une face carrée. Cette opération est présentée dans un but pédagogique de présentation de notre approche tandis que l'étude du chanfreinage, qui est la partie topologique de l'opération d'arrondi, nous fournit un cas d'étude complexe. Cette expérience nous permettra de mettre en œuvre deux approches de spécification distinctes, l'une orientée modèles (la méthode B), l'autre orientée propriétés (CASL) en nous attaquant à une opération difficile de modélisation. Le choix de la méthode B est motivé par sa disposition à manipuler intrinsèquement les notions d'ensembles et de relations, ce qui nous permettra de spécifier aisément les modèles mathématiques rencontrés en modélisation géométrique. Quant à CASL, il permet de poursuivre différentes approches algébriques prometteuses [DUF 89, BER 94, FUC 96, DUF 97] en fournissant désormais des outils plus nombreux associés à une sémantique complètement définie.

La section 2 de cet article introduit le modèle topologique utilisé et les opérations de couture et de chanfreinage. La section 3 est consacrée à leur formalisation en B et la section 4 à leur formalisation en CASL. Chacune de ces sections se termine par

1. Le développement d'un tel outil s'inscrit dans le cadre d'un projet universitaire français (Plan Pluri Formation) entre deux équipes de recherche spécialisées en modélisation géométrique (Poitiers, Strasbourg) et une équipe de recherche spécialisée en méthodes formelles (Évry).

la présentation des avantages et des inconvénients de chacun des langages dans le cadre de la modélisation géométrique. Enfin, nous revenons sur cette expérience de rétro-ingénierie en section 5 avant de conclure en section 6.

## 2. Éléments de modélisation à base topologique

L'objet d'étude de cet article est l'opération de chanfreinage qui est une opération topologique locale. Nous présentons dans un premier temps le modèle topologique des cartes généralisées de dimension  $n$  [LIE 94], ou  $n$ -G-cartes. Il permet de représenter une large classe d'objets, pour être précis les variétés<sup>2</sup> de taille quelconque, ouvertes ou fermées, orientables ou non. Notons dès à présent que nous ne nous intéresserons dans la suite de cet article qu'aux objets de dimension 3, donc aux 3-G-cartes. Cependant pour des raisons de lisibilité, certaines figures sont réalisées en 2D.

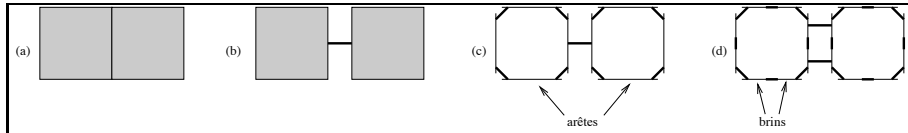
### 2.1. Qu'est qu'une $n$ -G-carte ?

L'idée sous-jacente à ce modèle est de décomposer un objet complexe de dimension  $n$  en un ensemble d'objets basiques de dimension 0, nommés communément des brins, reliés entre eux par diverses relations. Les brins sont représentés par des segments. Les liens entre brins sont modélisés mathématiquement par des involutions<sup>3</sup>. Considérons l'objet géométrique de la figure 2(a) qui représente un rectangle composé de deux carrés. À la figure 2(b), sa décomposition fait apparaître explicitement les deux surfaces carrées et leur relation d'adjacence. À la figure 2(c), on poursuit la décomposition de l'objet initial en décomposant chacun des carrés en quatre arêtes. Enfin, à la figure 2(d), les arêtes sont dissociées en brins (dimension 0). Nous avons ainsi structuré l'objet initial en le décomposant en objets de dimension 0, les brins. Toutes les relations entre faces ou arêtes ont été reportées sur les brins. Chacune de ces relations est modélisée par une involution  $\alpha_i$  dont l'index  $i$  correspond à la dimension de la relation d'adjacence. Par exemple, deux brins reliés par  $\alpha_2$  appartiennent à deux objets de dimension 2 distincts mais adjacents. Mathématiquement, les  $n$ -G-cartes sont des structures combinatoires dont la définition a été introduite dans [LIE 94].

**Définition 1** Une carte généralisée de dimension  $n$  (ou  $n$ -G-carte) est une algèbre  $G = (D, \alpha_0, \dots, \alpha_n)$  telle que :

2. Un objet de dimension  $n$  est une variété de dimension  $n$  si et seulement si tout point de cet objet admet un voisinage ouvert inclus dans cet objet qui soit isomorphe à une boule ouverte de  $\mathbb{R}^n$ .

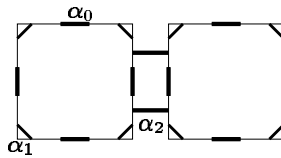
3. Une application  $f : D \rightarrow D$  est une involution si et seulement si pour tout élément  $x$  appartenant à  $D$ , on a  $xf^2 = x$  avec  $f^2$  la composition de  $f$  et  $f$  et  $xf^2$  l'application de  $f^2$  sur  $x$ .



**FIG. 2.** Décomposition d'un objet.

- $D$  est un ensemble fini de brins,
- $\alpha_i$  est une involution sur  $D$  ( $0 \leq i \leq n$ ), (c<sub>1</sub>)
- $\alpha_i \alpha_j$  est une involution ( $0 \leq i < i + 1 < j \leq n$ ). (c<sub>2</sub>)

Intéressons nous à la figure 3 sur laquelle figurent en trait épais toutes les relations d'adjacence. L'involution  $\alpha_0$  représente le lien entre deux sommets,  $\alpha_1$  le lien entre deux arêtes et  $\alpha_2$  le lien entre deux surfaces. On dit qu'un brin  $d$  est **i-cousu** à un brin  $d'$  de  $D$  si  $d\alpha_i = d'$ , **i-libre** si  $d\alpha_i = d$  et **libre** si pour tout  $i \in [0, n]$ ,  $d\alpha_i = d$ .



**FIG. 3.** Une 2-G-carte représentant deux carrés collés.

Lorsqu'on manipule un objet géométrique, on désigne naturellement les sommets, les arêtes, les faces et les volumes le composant. Ceci est bien sûr possible avec le modèle des  $n$ -G-cartes. On introduit à cet effet la notion d'orbite.

**Définition 2** Soit une  $n$ -G-carte  $G = (D, \alpha_0, \dots, \alpha_n)$ , un brin  $d \in D$  et  $P$  un ensemble de  $m$  permutations  $\{p_1, \dots, p_m\}$  sur  $D$ . On appelle **orbite** de  $d$  par rapport à  $P$  l'ensemble  $\{d' | \exists p'_1 \in P, \dots, \exists p'_k \in P, \text{ avec } d' = dp'_1 \dots p'_k\}$  (on peut avoir  $p'_i = p'_j$  pour  $i \neq j$ ) et on note cette orbite  $\langle p_1, \dots, p_m \rangle (d)$ .

L'orbite  $\langle p_1, \dots, p_k \rangle (d)$ , on peut être vue comme l'ensemble des brins atteignables à partir de  $d$  par la composition des fonctions  $p_1, \dots, p_k$ . Les sommets, arêtes, faces et volumes en sont des cas particuliers. On parle alors de *cellules*.

**Définition 3** Soient une  $n$ -G-carte  $G = (D, \alpha_0, \dots, \alpha_n)$ , un brin  $d \in D$ , et un entier  $i$  inférieur à  $n$ . La **i-cellule** ( $i \leq n$ ) contenant  $d$ , est l'orbite  $\langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle (d)$ .

Considérons la 2-G-carte de la figure 4(a). La 0-cellule  $\langle \alpha_1, \alpha_2 \rangle (d)$  qui définit un sommet est représentée en trait plein à la figure 4(b). A la figure 4(c) la 1-cellule  $\langle \alpha_0, \alpha_2 \rangle (d)$  définit une arête et à la figure 4(d) la 2-cellule  $\langle \alpha_0, \alpha_1 \rangle (d)$  définit une face.

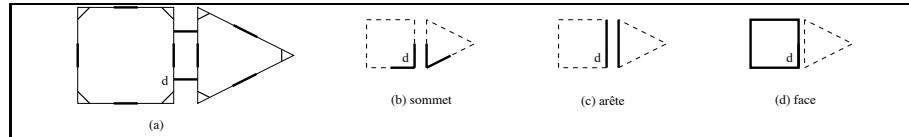


FIG. 4. Différentes cellules pour un brin  $d$ .

## 2.2. L'opération de couture

Intuitivement, la couture permet de coller des objets de même dimension. Par exemple, les deux cubes de la figure 5. L'opération de couture entre deux objets 3D nécessite de repérer chacune des surfaces de collage (grisées sur la figure 5(a)) par un brin. Ces deux brins sont notés  $d$  et  $d'$  à la figure 5(b). L'opération de couture consiste alors à les coudre ensemble. Cette couture engendre d'autres coutures de manière à conserver les propriétés liées aux composées d'involution dans la nouvelle 3-G-carte. Les liaisons induites apparaissent à la figure 5(c). Tout ceci n'est possible que si les deux surfaces de collage sont compatibles c'est à dire isomorphes. Seules les liaisons entre brins des deux surfaces de collage sont modifiées, les autres restent inchangées. Plus précisément tout brin de l'une des surfaces est maintenant relié au brin qui lui est isomorphe dans l'autre surface. De manière générale, la définition de la couture de deux brins est la suivante :

**Définition 4** Soient une  $n$ -G-carte  $G = (D, \alpha_0, \dots, \alpha_n)$ , un entier  $i$  inférieur à  $n$  et deux brins  $d, d' \in D$   $i$ -libres. Soit  $\varphi$  un isomorphisme de  $\langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (d)$  dans  $\langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (d')$  tel que  $d\varphi = d'$ . On définit  $G' = (D', \alpha'_0, \dots, \alpha'_n)$  par :

- $D' = D$
- $\forall j, 0 \leq j \leq n, j \neq i, \alpha'_j = \alpha_j$
- $e\alpha'_i = \begin{cases} e\varphi & \text{si } e \in \langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (d) \\ e\varphi^{-1} & \text{si } e \in \langle \alpha_0, \dots, \alpha_{i-2}, \alpha_{i+2}, \dots, \alpha_n \rangle (d') \\ e\alpha_i & \text{sinon.} \end{cases}$

$G'$  est le résultat de la **couture** de  $d$  et  $d'$  à la dimension  $i$  à partir de  $G$ .

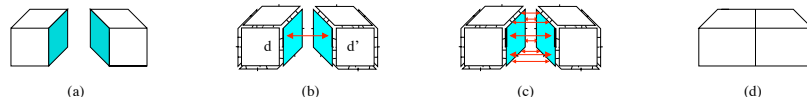


FIG. 5. Couture de deux brins à la dimension 2.

### 2.3. L'opération de chanfreinage

L'opération de chanfreinage est une opération purement topologique qui correspond à la partie topologique de l'arrondi<sup>4</sup>. Ainsi, si l'on considère la 2-G-carte de la figure 6 (a)1) qui représente la surface d'un cube et que nous chanfreinons l'un de ses sommets agrandi à la figure 6(a)2), nous obtenons la 2-G-carte de la figure 6(b)2), soit la 2-G-carte complète de la figure 6(b)1).

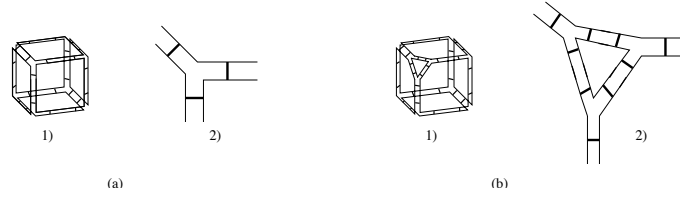


FIG. 6. Un cube en (a) dont on chanfreine un sommet

L'approche que nous retenons est celle présentée dans [ELT 94]. Nous avons en figure 6 chanfreiné un sommet. Ce sommet qui possède trois faces incidentes a été remplacé par une face à trois côtés. De manière générale, chanfreiner une  $i$ -cellule  $I$  dans un objet de dimension  $n$  consiste à la remplacer par une  $n$ -cellule dont le nombre de  $(n - 1)$ -cellules la bordant est égal au nombre de  $n$ -cellules incidentes à  $I$ . La définition donnée dans [ELT 94] énumère tous les brins et toutes les involutions de la  $n$ -G-carte obtenue par chanfreinage. Nous la reproduisons ci-dessous.

**Définition 5** Soient une  $n$ -G-carte  $G = (D, \alpha_0, \dots, \alpha_n)$ ,  $d$  un brin de  $D$  et  $C$  la  $i$ -cellule

$\langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle (d)$ . Soit  $(D_j)_{j \in [i, n]}$  une famille d'ensembles de brins telle que  $D_i = C$  et pour tout  $j \in [i + 1, n]$ ,

- $D \cap D_j = \emptyset$ ,
- pour tout  $k \in [i + 1, n]$ ,  $k \neq j$ ,  $D_j \cap D_k = \emptyset$ ,
- il existe une bijection  $\varphi^j : D_i \rightarrow D_j$ . On posera  $\varphi^i = id_{D_i}$ .

En chanfreinant  $C$  dans  $G$ , on obtient la  $n$ -G-carte  $E = (D^E, \alpha_0^E, \dots, \alpha_n^E)$  définie par :

- 1)  $D^E = D \cup \bigcup_{j \in [i+1, n]} D_j$
- 2)  $\forall d \in D, \forall k, (0 \leq k \leq n), d\alpha_k^E = \begin{cases} d\varphi^{i+1} & \text{si } k = i + 1 \text{ et } d \in D_i \\ d\alpha_k & \text{sinon} \end{cases} \quad (2.1)$
- 3)  $\forall j, (i < j \leq n), \forall d \in D_i, \forall k, (0 \leq k \leq n), d\varphi^j \alpha_k^E =$

4. Nous simplifions ici la relation entre chanfreinage et arrondi afin de faciliter le dialogue. Par exemple, dans le cas d'une arête, la topologie associée à l'arrondi ne sera pas celle d'un simple chanfreinage mais d'une composition de plusieurs chanfreins.



$$\left\{ \begin{array}{ll} d\alpha_k \varphi^j & \text{si } 0 \leq k < i \quad (3.1) \\ d\alpha_{k+1} \varphi^j & \text{si } i \leq k < j \quad (3.2) \\ d\varphi^{j-1} & \text{si } k = j \quad (3.3) \\ d\varphi^{j+1} & \text{si } k = j + 1 \quad (3.4) \\ d\alpha_k \varphi^j & \text{si } j + 1 < k \leq n \quad (3.5) \end{array} \right.$$

Il n'est pas nécessaire de maîtriser parfaitement cette définition pour comprendre la suite de cet article. Néanmoins, certains points peuvent être éclaircis. Les cellules introduites par le chanfreinage peuvent être déduites de la cellule à chanfreiner par simple translation des dimensions le long des fonctions d'introduction de nouveaux brins  $\varphi^j$ ,  $j \in [i + 1, n]$ . Ainsi, chaque ensemble  $D_j$  ( $j \in [i + 1, n]$ ) de brins est la copie par la bijection  $\varphi^j$  des brins de  $D_i$  (cellule de départ) et les coutures entre les brins de  $D_j$  sont déduits de celles entre les brins de la cellule à chanfreiner (point 3 de la définition 5). Le point 2.1 nous indique comment sont rattachés les nouveaux brins introduits à l'objet initial. Le point 2.2 nous montre que le reste de l'objet est inchangé.

Cette définition du chanfreinage est constructive. Notre démarche de rétro-ingénierie et notamment l'utilisation de spécifications formelles nous a amené à proposer une définition observationnelle plus abstraite. Pour cela nous introduisons préalablement une notion d'inclusion entre  $n$ -G-cartes qui permet de définir toutes les opérations locales. Par exemple, celles qui enrichissent l'objet (ajout de brins, nouvelle couture, chanfreinage) se définissent comme la plus petite  $n$ -G-carte contenant celle d'origine et les enrichissements apportés.

**Définition 6** Soient deux  $n$ -G-carte  $G = (D, \alpha_0, \dots, \alpha_n)$  et  $G' = (D', \alpha'_0, \dots, \alpha'_n)$ .  $G$  est **include** dans  $G'$ , noté  $G \subseteq G'$ , ssi :

- $D \subseteq D'$ ,
- $\forall i \in \{0, \dots, n\}, \forall d, d' \in D$ , si  $d\alpha_i = d'$  avec  $d' \neq d$  dans  $G$  alors  $d\alpha'_i = d'$  dans  $G'$ ,

**Théorème 1** Soient une  $n$ -G-carte  $G = (D, \alpha_0, \dots, \alpha_n)$ ,  $d$  un brin de  $D$  et  $C$  la  $i$ -cellule  $\langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle (d)$ . Soit  $(D_j)_{j \in [i, n]}$  une famille d'ensembles de brins telle que  $D_i = C$  et pour tout  $j \in [i + 1, n]$ ,

- $D \cap D_j = \emptyset$ ,
- pour tout  $k \in [i + 1, n]$ ,  $k \neq j$ ,  $D_j \cap D_k = \emptyset$ ,
- il existe une bijection  $\psi^j : D_i \rightarrow D_j$ .

Soit  $L = (D^L, \alpha_0^L, \dots, \alpha_n^L)$  la plus grande  $n$ -G-carte contenue dans  $G$  telle que :  $\forall d \in D_i$ ,  $d\alpha_{i+1}^L = d$ . Alors la  $n$ -G-carte  $E = (D^E, \alpha_0^E, \dots, \alpha_n^E)$  définie à la définition 5 est la plus petite  $n$ -G-carte contenant  $L$  et telle que :

- 1)  $D^E = D \cup \bigcup_{j \in [i+1, n]} D_j$
- 2) pour tout  $d \in D_i$ ,  $d\alpha_{i+1}^E = d\psi^{i+1}$
- 3) pour tout  $d \in D_i$ , pour tout  $k \in [i + 1, n - 1]$ ,  $d\psi^k \alpha_{k+1}^E = d\psi^{k+1}$

4) pour tout  $d_1, d_2 \in D_i$ ,

- pour tout  $k \in [0, i - 1]$ ,  $d_1 \alpha_k = d_2 \Rightarrow d_1 \psi^n \alpha_k^E = d_2 \psi^n$

- pour tout  $k \in [i + 1, n]$ ,  $d_1 \alpha_k = d_2 \Rightarrow d_1 \psi^n \alpha_{k-1}^E = d_2 \psi^n$

*Idée de la preuve :*

Nous considérons une  $n$ -G-carte vérifiant la définition 5 et nous prouvons par l'absurde qu'il s'agit de la plus petite  $n$ -G-carte vérifiant les propriétés du théorème 1. La preuve complète est relativement technique et se trouve dans [LED 00].

Comme pour la définition constructive 5, les fonctions  $\psi^j$ ,  $j \in [i + 1, n]$  nous permettent de manipuler les nouveaux brins introduits. Les points (2) et (3) indiquent que pour chaque brin de  $D^i$ , nous ajoutons  $n - i$  brins cousus d'une certaine façon et le point (4) indique comment sont cousus certains brins introduits (ceux atteignables par  $\psi^n$ ). Nous ne donnons explicitement que le minimum de coutures. Les autres coutures sont imposées par le respect des invariants du modèle des  $n$ -G-cartes. Ce théorème introduit une nouvelle façon de caractériser le chanfreinage et par la même une nouvelle définition du chanfreinage plus abstraite que la définition 5. Cette caractérisation du chanfreinage est formalisée par la suite en utilisant l'approche axiomatique du langage CASL. La définition 5 très opérationnelle est pour sa part formalisée à l'aide de la méthode B de la manière présentée ci-dessous.

### 3. L'approche B

Dans cette partie, nous présentons un modèle abstrait d'une 3-G-carte en langage B sous la forme d'une machine abstraite. Nous commentons ce modèle ci-dessous. Le but de notre modèle formel est de donner une représentation simple de l'opération de chanfreinage. Nous proposons une machine abstraite qui se compose d'un état évolutif encapsulant la 3-G-carte, d'une opération de couture et d'une opération permettant de chanfreiner une cellule de cette 3-G-carte.

#### MACHINE

*BGmap*

#### SETS

*DARTS*

#### ABSTRACT\_VARIABLES

*darts, alpha*

#### INVARIANT

$darts \subseteq DARTS \wedge$

$alpha \in 0 .. 3 \rightarrow (darts \rightarrow darts) \wedge$

$\forall (i).(i \in 0 .. 3 \Rightarrow is\_an\_involution(alpha(i))) \wedge$

$\forall (i, j).(i \in 0 .. 1 \wedge j \in i+2 .. 3 \Rightarrow is\_an\_involution(alpha(i); alpha(j)))$

#### DEFINITIONS

$is\_an\_involution(g) == (g = g^{-1}); (..)$

### 3.1. L'état de la machine abstraite BGmap : le modèle des 3-G-cartes

Les clauses ABSTRACT\_VARIABLES et INVARIANT contiennent la traduction de la définition mathématique d'une 3-G-carte (définition 1 dans le cas  $n = 3$ ). Nous appellerons cette définition mathématique, la définition informelle. Nous introduisons deux variables *darts* et *alpha*. L'ensemble *darts* ( $D$  dans la définition informelle) est défini comme un sous-ensemble de l'ensemble  $DARTS$  qui représente l'univers de tous les brins possibles. La fonction *alpha* regroupe toutes les involutions  $\alpha_0, \alpha_1, \alpha_2$  et  $\alpha_3$  : l'objet  $B$ ,  $alpha(i)$ , représente l'objet mathématique  $\alpha_i$ . Dans la première partie de l'invariant, nous exprimons que  $alpha(i)$  est une fonction totale de *darts* dans *darts*. De plus nous ajoutons qu'il s'agit d'une involution en utilisant l'opérateur inverse fourni par  $B$ . La dernière partie de l'invariant exprime la condition ( $c_2$ ) de la définition informelle sur les composées d'involution.

### 3.2. L'opération de couture

La méthode  $B$  possède un pouvoir d'expression qui nous permet de formaliser quasi-directement la définition 4. Ainsi, cette opération pose en précondition, outre les contraintes usuelles sur les brins à coudre, un prédicat d'existence d'isomorphisme. Il a été exprimé grâce à une macro qui définit l'ensemble de tous les isomorphismes possibles entre les deux orbites respectives des deux brins à coudre. L'orbite a quant à elle été décrite en utilisant une constante définie axiomatiquement à ce stade de la spécification.

La couture consiste alors à choisir dans l'ensemble des isomorphismes, celui qui coud effectivement les deux brins  $d1$  et  $d2$  passés en paramètre de l'opération, il est appelé *Iso*. La couture est une opération locale ainsi les brins n'appartenant pas aux orbites définies à partir des deux brins  $d1$  et  $d2$  ne seront pas modifiés par la couture. L'opérateur de surcharge noté  $\triangleleft$  possède intrinsèquement cette propriété. En surchargeant  $alpha(dim)$  par *Iso*, nous modifions uniquement les coutures correspondant aux brins des deux orbites définies par  $d1$  et  $d2$ . L'isomorphisme *Iso* de la spécification correspond à  $\varphi$  union  $\varphi^{-1}$  dans la définition 4.

#### ABSTRACT\_CONSTANTS

*Orbit*

#### PROPERTIES

$$Orbit \in \mathbb{P}(DARTS) * DARTS * \mathbb{P}(DARTS \leftrightarrow DARTS) \rightarrow \mathbb{P}(DARTS) \wedge$$

$$\forall (Set, dart, AlphaSet). (Set \cup DARTS \wedge dart \in Set \wedge AlphaSet \in \mathbb{P}(Set \rightarrow Set) \Rightarrow dart \in Orbit(Set, dart, AlphaSet)) \wedge$$

$$\forall (Set, dart, dart2, AlphaSet, alphai).$$

$$(Set \subseteq DARTS \wedge dart \in Set \wedge dart2 \in Set \wedge AlphaSet \subseteq Set \rightarrow Set \wedge alphai \in AlphaSet \Rightarrow dart2 \in Orbit(Set, dart, AlphaSet)) \Leftrightarrow$$

$$Set \subseteq DARTS \wedge dart \in Set \wedge dart2 \in Set \wedge AlphaSet \subseteq Set \rightarrow Set \wedge alphai \in AlphaSet \wedge alphai(dart2) \in Orbit(Set, dart, AlphaSet)) \wedge$$

$$\forall (Set, dart, AlphaSet).(Set \cup DARTS \wedge dart \in Set \wedge AlphaSet \in \mathbb{P}(Set \rightarrow Set) \\ \Rightarrow Orbit(Set, dart, AlphaSet) \cup Set)$$
**DEFINITIONS**

$$(\dots) O(d) == Orbit(darts, d, \mathbf{ran}(\{dim-1, dim, dim+1\} \triangleleft \alpha));$$

$$isomorphous\_set(set1, set2, di) == \{f \mid f \in set1 \cup set2 \rightsquigarrow set1 \cup set2 \wedge f = f^{-1} \wedge \\ \forall (k).(k \in (0..di-2) \cup (di+2..3) \Rightarrow (f; \alpha(k)) = (\alpha(k); f))\}; \quad (\dots)$$
**OPERATIONS**

$$Sew(dim, d1, d2) =$$

**PRE**  $d1 \in darts \wedge d2 \in darts \wedge dim \in 0..3 \wedge isomorphous\_set(O(d1), O(d2), dim) \neq \emptyset \wedge$   
 $O(d1) \triangleleft \alpha(dim) = \mathbf{id}(O(d1)) \wedge O(d2) \triangleleft \alpha(dim) = \mathbf{id}(O(d2))$

**THEN**

**ANY**  $Iso$

**WHERE**  $Iso \in isomorphous\_set(O(d1), O(d2), dim) \wedge Iso(d1) = (d2)$

**THEN**

$\alpha(dim) := \alpha(dim) \triangleleft Iso$

**END**

**END;**

**3.3. L'opération de chanfreinage**

L'opération de chanfreinage d'un brin  $d$  (d'une 3-G-carte) en dimension 0, 1 ou 2 est décrite comme une opération modifiant la 3-G-carte sur laquelle elle agit : elle ajoute des nouveaux brins, les coud et modifie quelques anciennes coutures. Les seuls anciens brins touchés par l'opération (par opposition aux nouveaux créés par l'opération) sont les brins appartenant à la  $i$ -cellule du brin  $d$ , calculés par l'expression :  $Orbit(darts, dart, \{dim\} \triangleleft \alpha)$  déduite de la définition 3.

Pour spécifier l'opération de chanfreinage, nous sommes partis de la définition 5. Nous avons pris un à un les éléments qui la composent et nous les avons traduits en B. Comme pour la définition des involutions  $\alpha_i$ , nous avons rassemblé les  $D_j$  au sein d'une même fonction  $newD$ , les  $\varphi^j$  au sein d'une même fonction  $phi$ . L'ensemble  $DiDj$  contient l'ensemble des nouveaux et anciens brins concernés par le chanfreinage.

La structure globale de la substitution généralisée qui décrit l'opération de chanfreinage est présentée ci-dessous :

$$darts := darts \cup \mathbf{union}(\mathbf{ran}(newD)) - newD(i) \parallel$$

$$\alpha := \lambda(k).(k \in 0..3 \mid \alpha(k) \triangleleft \{d1, d2 \mid C_1(k) \vee C_2(k) \vee C_3(k) \vee C_4(k) \vee C_5(k) \vee C_6(k)\})$$

Intuitivement cette opération ajoute les nouveaux brins à l'ensemble des brins et modifie les différentes involutions en surchargeant chacune d'elles par un ensemble de

couples  $(d_1 \mapsto d_2)$  où les brins  $d_1, d_2$  sont des brins arbitraires cousus par  $alpha(k)$  à l'aide d'une des conditions  $C_i(k)$ <sup>5</sup> pour  $i$  de 1 à 6. Ces contraintes  $C_i(k)$  correspondent respectivement aux cas 2.1, 3.1, 3.2, 3.3, 3.4, 3.5 de la définition 5.

Notons que le cas 2.2 de la définition 5 n'apparaît dans aucune des conditions ci-dessus. En effet, cette propriété caractérise la partie non-chanfreinée de l'objet qui reste donc identique. Cette proposition est directement prise en compte par la sémantique même de l'opération de surcharge.

**DEFINITIONS**

$(..) Cell(dart, dim) == Orbit(darts, dart, \{dim\} \triangleleft alpha);$

$DiDj == \mathbf{union}(\mathbf{ran}(newD))$

**OPERATIONS** (...)

**Chamfering**  $(d, i) =$

**PRE**  $d \in darts \wedge i \in 0 .. 2$

**THEN**

**ANY**  $phi, newD$

**WHERE**

$newD \in i .. 3 \rightarrow \mathbb{P}(DARTS) \wedge$

$newD(i) = Cell(d, i) \wedge$

$\forall (j). (j \in i+1 .. 3 \Rightarrow newD(j) \in \mathbb{F}(DARTS - darts)) \wedge$

$\forall (x, y). (x \in i+1 .. 3 \wedge y \in i+1 .. 3 \Rightarrow newD(x) \cap newD(y) = \emptyset) \wedge$

$phi \in i .. 3 \rightarrow (newD(i) \rightarrow DiDj) \wedge$

$\forall (j). (j \in i .. 3 \Rightarrow phi(j) \in (newD(i) \rightarrow newD(j))) \wedge$

$phi(i) = \mathbf{id}(newD(i))$

**THEN**

$darts := darts \cup \mathbf{union}(\mathbf{ran}(newD)) - newD(i) \parallel$

$alpha := \lambda (k) . (k \in 0 .. 3 \mid$

$alpha(k) \triangleleft \{ d1, d2 \mid d1 \in DiDj \wedge d2 \in DiDj \wedge ($

$/ * c_1(k) * / (k = i+1 \wedge ((d1 \in newD(i) \wedge d2 = phi(i+1)(d1))))$

$/ * c_2(k) * / \vee (k \in 0 .. i-1 \wedge \exists (j, d3). (j \in i+1 .. 3 \wedge d3 \in newD(i) \wedge d1 = phi(j)(d3) \wedge d2 = phi(j)(alpha(k)(d3))))$

$/ * c_3(k) * / \vee (\exists (j, d3). (j \in i+1 .. 3 \wedge k \in i .. j-1 \wedge d3 \in newD(i) \wedge d1 = phi(j)(d3) \wedge d2 = phi(j)(alpha(k+1)(d3))))$

$/ * c_4(k) * / \vee (k \in i+1 .. 3 \wedge \exists (d3). (d3 \in newD(i) \wedge (d1 = phi(k)(d3) \wedge d2 = phi(k-1)(d3))))$

$/ * c_5(k) * / \vee (k \in i+2 .. 3 \wedge \exists (d3). (d3 \in newD(i) \wedge (d1 = phi(k-1)(d3) \wedge d2 = phi(k)(d3))))$

5. Si  $(d1 \mapsto d2)$  vérifie  $C_i(k)$  et  $(d1' \mapsto d2')$  vérifie  $C_j(k)$  pour  $i \neq j$ , par construction les conditions  $C_i(k)$  assurent que  $d1 \neq d1'$ . Ainsi la surcharge définit bien une application

```

/ * c6(k) * / ∨ ( ∃ (j,d3).(j ∈ i+1 .. 3 ∧ k ∈ j+2 .. 3 ∧ d3 ∈ newD(i)
  ∧ d1 = phi(j)(d3) ∧ d2 = phi(j) (alpha(k)(d3))))))}

```

**END**

**END**

### **3.4. Avantages de l'approche B**

La méthode B nous a permis d'effectuer facilement le passage de l'informel au formel, c'est-à-dire de la définition mathématique à la machine abstraite B. Il s'agit là d'un avantage très fort car le passage de l'informel au formel est souvent délicat dans d'autres domaines d'application ou avec d'autres méthodes formelles. Cette relative aisance vient du fait que B nous fournit gratuitement les ensembles, éléments centraux dans la définition des  $n$ -G-cartes, ainsi que les fonctions, les relations et des opérateurs mathématiques de haut niveau comme la surcharge ou la soustraction sur le domaine. Enfin le concept de machine abstraite, nous permet d'encapsuler les types de données et de faire correspondre directement les opérations avec les primitives d'un modèleur. Tous ces éléments rendent l'activité de modélisation aisée, naturelle et rapide.

En outre, un des avantages reconnus de la méthode B est la vérification formelle de la préservation de l'invariant. Ce travail de preuve a été effectué sur l'opération de couture mais aussi sur d'autres opérations de difficultés similaires ou moindres (ajouter, retirer et découdre un brin). Cette expérience a été menée à bien pour l'ensemble de ces opérations. Elle comporte des lemmes sur les involutions et des propriétés sur les orbites. Toutefois pour l'opération de chanfreinage, nous n'avons pas extrait interactivement de notre spécification une telle preuve, nous avons néanmoins tous les éléments pour le faire. En effet, il existe une preuve mathématique papier-crayon [ELT 94] qui nous permet d'avoir une idée précise de la forme de la future preuve formelle. Néanmoins, nous devons adapter cette preuve à l'opération de chanfreinage car elle a été réalisée dans un contexte plus général. En l'occurrence, il s'agit ici d'une preuve par cas. Ce travail est en cours et 55 % des obligations de preuves ont été prouvées mécaniquement.

### **3.5. Inconvénients de l'approche B**

Lorsqu'on a compris intuitivement ce qu'est l'opération de chanfreinage, on peut être surpris, notamment lors de la lecture des paragraphes antérieurs, par le manque de lisibilité la spécification B. C'est un inconvénient qu'il faut pondérer, car il n'est pas dû à la méthode B directement, mais à la complexité de l'opération de chanfreinage et de sa définition. Comme nous l'avons vu dans la partie précédente, la méthode B favorise le passage de l'informel au formel. Néanmoins, si l'informel est complexe et peu lisible, le formel a de grandes chances de l'être tout autant. Dans notre cas, nous avons, par rapport à la définition mathématique, fait un pas vers une meilleure lisibilité en supprimant un cas grâce au caractère très haut niveau de l'opération de surcharge.

Même si notre machine B peut au prime abord apparaître trop complexe à saisir, elle correspond néanmoins aux attentes des infographistes. Ces derniers y reconnaissent dans l'invariant la définition des  $n$ -G-cartes et dans l'opération de chanfreinage la définition mathématique telle qu'ils la connaissent. En ce sens, notre machine B constitue en l'état un document de référence à la fois pour le développement rigoureux d'un modèleur et pour les experts du domaine.

Par ailleurs, même si nous avons tous les éléments nécessaires pour effectuer la preuve du chanfreinage, cette dernière ne sera pas triviale. En effet, la preuve mathématique de référence est conséquente. De plus, elle est fondée sur de nombreux lemmes et de nombreuses connaissances implicites que nous devons formuler. À notre connaissance, il existe peu de travaux qui se sont attaqués à des objets aussi complexes, nous ne pouvons donc pas compter sur l'expérience d'autres études antérieures pour nourrir ou évaluer notre travail.

Même si dans le cas du chanfreinage nous avons tous les éléments pour prouver interactivement la préservation de l'invariant, cette étape est fastidieuse et coûteuse en temps : nous avons pu nous en rendre compte lors de la preuve des obligations de preuve engendrées par l'opération de couture. Afin de factoriser les efforts de preuve, nous aimerions disposer d'un prouveur spécialisé dédié au domaine de la modélisation géométrique. Mais pour cela il faudra dégager les lemmes fondamentaux et récurrents en modélisation géométrique, notamment les propriétés des involutions, en étudiant les preuves des obligations de preuve engendrées à partir de la spécification, mais aussi d'autres spécifications sur d'autres opérations, puis concevoir et implanter, autour de cette bibliothèque de lemmes, une tactique spécialisée pour ce type de preuves au sein de l'Atelier B.

#### 4. Une spécification en CASL

Le langage de spécifications algébriques CASL (*Common Algebraic Specification Language*) a été développé par CoFI (*Common Framework Initiative*) [LAN 00] dont l'objectif est de fournir un cadre de développement outillé unifiant la plupart des approches algébriques. De ce fait, CASL dispose de nombreuses caractéristiques telles que la gestion des fonctions partielles et totales, des mécanismes de sous-typage, différents constructeurs de type, une structuration importante des spécifications, des bibliothèques de spécifications, etc. Le lecteur voulant plus d'informations peut consulter [MOS 99, BID 00] pour avoir une introduction à CASL, et [LAN 00] pour une présentation plus complète.

##### 4.1. Le type abstrait $3Gmap$

Afin de manipuler les objets auxquels on s'intéresse, il nous faut avoir des types de données les représentant. En l'occurrence, nous manipulons une  $3$ -G-carte, nous introduisons donc le type de données  $3Gmap$ . Pour cela, nous définissons au préalable

différents types de base utiles pour formaliser les 3-G-cartes : les types de données permettant de manipuler les brins et les fonctions  $\alpha_0$ ,  $\alpha_1$ ,  $\alpha_2$ , et  $\alpha_3$ . Un brin est un élément abstrait sans propriété particulière. Les brins sont donc spécifiés par la simple déclaration d'une sorte : `sort Dart`. Les ensembles de brins sont spécifiés par instantiation de la spécification paramétrique FINITESSET de la bibliothèque STRUCTURED DATATYPES [ROG 00]. Nous avons ainsi à disposition la sorte `FinSet[Dart]`. Comme nous nous restreignons à l'étude des 3-G-cartes, les seules valeurs entières nous intéressant sont 0, 1, 2 et 3. Nous définissons donc le type `Dim` comme une sous-sorte de la sorte `Nat`<sup>6</sup>.

```
sort Dim = {n : Nat • n ≤ 3}
```

Les involutions  $\alpha_0$ ,  $\alpha_1$ ,  $\alpha_2$ , et  $\alpha_3$ , qui représentent les liens d'une 3-G-carte sont spécifiés par instantiation de la spécification des applications à domaine et codomaine finis FINITEMAP (bibliothèque STRUCTURED DATATYPES [ROG 00]) enrichie de prédicats sur les fonctions tels que l'injection, la surjection, la bijection et l'involution et des opérations telles que la composition. Une opération telle que la composition nécessite que le domaine et le codomaine soit de même type. On a donc spécifié le type `SFiniteMap[Dart]`. Comme dans la spécification B, les quatre involutions sont représentées à l'aide d'une application unique de sorte `FiniteMap[Dim, SFiniteMap[Dart]]` obtenue par uinstantiation de la spécification FINITEMAP. Nous avons maintenant tous les éléments pour formaliser les 3-G-cartes qui rappelons le, sont des structures combinatoires formées d'un ensemble de brins et de quatre involutions vérifiant les conditions ( $c_1$ ) et ( $c_2$ ) de la définition 1.

```
spec 3GMAP =
  ...
then
  free
  { type 3Gmap ::= gmap(__.D : FinSet[Dart];
    __.α : FiniteMap[Dim, SFiniteMap[Dart]]?);
    var G : 3Gmap
    • def gmap(G.D, G.α) ⇔
      ∀i, j : Dim • (G.α(i) :: G.D → G.D ∧ involution(G.α(i), G.D) ∧
        (j > i + 1 ⇒ involution(G.α(i) ◦ G.α(j), G.D)))
  }
end
```

Pour spécifier les 3-G-cartes nous avons utilisé le mot clef `type` qui permet de déclarer simultanément la sorte `3Gmap`, son constructeur `gmap` et ses accesseurs `D`

<sup>6</sup>. `Nat` est une spécification des entiers naturels (`Nat`) donnée dans la librairie NUMBERS [ROG 00]



et  $\alpha$ . Le mot clef **free** qui indique qu'on ne considère que la sémantique initiale<sup>7</sup>. Notons que le constructeur *gmap* est déclaré comme partiel (mot clef "?"), car il n'est défini que si  $\alpha$  fournit bien quatre involutions définies sur  $D$  vérifiant les conditions de composition d'involutions. D'autre part, nous avons utilisé la mise en forme des opérations qui nous permet par exemple de mettre la dimension des liens en indice dans  $\alpha_i$  ou d'accéder aux différents éléments d'une  $\beta$ -G-carte comme dans  $G.D$  et  $G.\alpha_j$ . Cette dernière notation est particulièrement agréable pour les opérations de modification d'un objet, qui peuvent ainsi être "mises en séquence".

**Remarque :** L'ensemble des spécifications CASL présentées dans ce papier sont extraites de bibliothèques de spécifications [A.A 01] que nous avons développées. Ces bibliothèques ont été parsées et statiquement analysées avec la version 0.73 de cats [MOS 01]. Cependant la version actuelle de cats ne permet pas de générer du LaTeX comme le faisaient les versions antérieures. De ce fait la présentation des spécifications CASL de ce papier a été faite de manière ad-hoc en accord avec les résultats obtenus sur des versions antérieures de cats.

#### 4.2. L'opération de couture

Quelque soit l'opération géométrique traitée, nous la formalisons observationnellement en utilisant la notion d'inclusion entre  $n$ -G-cartes. que nous formalisons à l'aide du prédicat suivant :

```

pred  $\_ \subseteq \_ : \mathcal{G}map \times \mathcal{G}map$ 
vars  $G, G' : \mathcal{G}map$ 
•  $G \subseteq G' \Leftrightarrow$ 
   $\forall d : Dart \bullet ((d \in G.D \Rightarrow d \in G'.D) \wedge$ 
   $\forall i : Dim \bullet (d \in G.D \wedge \neg(iFreeDart(G, d, i))) \Rightarrow G.\alpha(i)(d) = G'.\alpha(i)(d))$ 

```

Le prédicat  $iFreeDart(G, d, i)$  est vérifié si le brin  $d$  est  $i$ -libre dans la  $\beta$ -G-carte  $G$ . La couture se spécifie alors comme étant la plus petite  $n$ -G-carte contenant la  $n$ -carte initiale et telle que les deux brins à coudre soient effectivement cousus à la dimension souhaitée. Les conditions à vérifier sont regroupées à l'aide du prédicat suivant.

```

pred  $predSew : \mathcal{G}map \times \mathcal{G}map \times Dart \times Dart \times Dim;$ 
vars  $G, Gafter : \mathcal{G}map; d, d' : Dart; i : Dim$ 
•  $predSew(Gafter, G, d, d', i) \Leftrightarrow (G \subseteq Gafter \wedge Gafter.\alpha(i)(d) = d')$ 

```

On spécifie dès lors le résultat de la couture comme étant la plus petite 3-G-carte vérifiant ce prédicat, d'où la spécification :

**local**

<sup>7</sup>. Parmi l'ensemble des modèles validant la spécification, on ne considère que la classe des modèles isomorphes au modèle initial. Intuitivement tout élément du domaine associé à la sorte  $\mathcal{G}map$  est l'interprétation d'un et un seul terme construit à l'aide du constructeur *gmap*.

```

pred predSew :  $3Gmap \times 3Gmap \times Dart \times Dart \times Dim$ ;
    ...
within
    op sew :  $3Gmap \times Dart \times Dart \times Dim \rightarrow? 3Gmap$ ;
    vars G :  $3Gmap$ ; d, d' :  $Dart$ ; i :  $Dim$ 
    • def G.sew(d, d', i)  $\Leftrightarrow$  ...
    • def G.sew(d, d', i)  $\Rightarrow$  predSew(G.sew(d, d', i), G, d, d', i)
    • def G.sew(d, d', i)  $\Rightarrow$ 
      ( $\forall G' : 3Gmap \bullet$  predSew(G', G, d, d', i)  $\Rightarrow$  G.sew(d, d', i)  $\subseteq$  G')

```

Le prédicat *predSew* étant introduit localement, le résultat de la couture est spécifiée comme prévue c'est-à-dire comme la plus petite *n*-G-carte vérifiant *predSew* (deux derniers axiomes). L'opération de couture est partielle (" $\rightarrow?$ ") puisque les brins qu'on veut coudre doivent être dans la *n*-G-carte, la dimension doit être inférieure à *n* et que les deux cellules à coudre doivent être isomorphes.

### 4.3. L'opération de chanfreinage

La formalisation du chanfreinage est une traduction fidèle du théorème 1. Nous introduisons la bijection  $\psi$  qui permet lors du chanfreinage d'introduire les nouveaux brins en fonction de la cellule à chanfreiner; le prédicat *isChamfered* qui spécifie si une  $\beta$ -G-carte vérifie toutes les propriétés des résultats du chanfreinage; et enfin l'opération de chanfreinage elle-même qui définit son résultat comme la plus petite  $\beta$ -G-carte validant le prédicat *isChamfered*.

```

spec CHAMFERED3GMAP =
    3GMAP
then
local
    op  $\_.\psi$  :  $3Gmap \times Dim \times Dart \rightarrow? Dart$ ;
    vars G :  $3Gmap$ ; d, d' :  $Dart$ ; i, j :  $Dim$ 
    • def G. $\psi^i$ (d)  $\Leftrightarrow$  d  $\in$  G.D
    •  $\neg$ (G. $\psi^i$ (d)  $\in$  G.D)
    • G. $\psi^i$ (d) = G. $\psi^j$ (d')  $\Rightarrow$  (d = d'  $\wedge$  i = j)
    pred isChamfered :  $3Gmap \times 3Gmap \times Dart \times Dim$ ;
    vars G, Gafter :  $3Gmap$ ; d :  $Dart$ ; i :  $Dim$ 

```

- $isChamfered(Gafter, G, d, i) \Leftrightarrow$   
 $(G.unsew(G.cell(d, i).D, (i + 1) as Dim) \subseteq Gafter \quad \%A_1$   
 $\wedge$   
 $\forall d' : Dart \bullet d' \in G.cell(d, i).D \Rightarrow$   
 $(Gafter.\alpha(i + 1 as Dim)(d') = G.\psi^{(i+1) as Dim}(d') \quad \%A_2$   
 $\wedge$   
 $\forall k : Dim \bullet ((i + 1 \leq k \wedge k \leq 2) \Rightarrow$   
 $Gafter.\alpha(k + 1 as Dim)(G.\psi^k(d')) = G.\psi^{(k+1) as Dim}(d')) \quad \%A_3$   
 $\wedge$   
 $\forall d_1, d_2 : Dart \bullet \forall k : Dim \bullet$   
 $(d_1 \in G.cell(d, i).D \wedge d_2 \in G.cell(d, i).D \wedge k \leq 2) \Rightarrow$   
 $((k < i \wedge G.\alpha(k)(d_1) = d_2) \Rightarrow Gafter.\alpha(k)(G.\psi^3(d_1)) = G.\psi^3(d_2)) \wedge$   
 $((k > i \wedge G.\alpha(k)(d_1) = d_2) \Rightarrow$   
 $Gafter.\alpha(k-1 as Dim)(G.\psi^3(d_1)) = G.\psi^3(d_2)) \quad \%A_4$

**within**

- op**  $_.chamfering : 3Gmap \times Dart \times Dim \rightarrow ? 3Gmap;$
- vars**  $G : 3Gmap; i, j : Dim; d : dart$
- $def G.chamfering(d, i) \Leftrightarrow (d \in D(G) \wedge i < 3)$
- $isChamfered(G.chamfering(d, i), G, d, i)$  if  $def G.chamfering(d, i)$
- $(\forall G' : 3Gmap \bullet isChamfered(G', G, d, i) \Rightarrow G.chamfering(d, i) \subseteq G')$   
if  $def G.chamfering(d, i)$

**end**

La spécification du prédicat *isChamfered* est un simple plagiat des conditions énoncées dans le théorème 1. La condition  $A_1$  correspond au point 1. Notons que l'opération *unsewset* permet de découdre les brins de la cellule à chanfreiner (avant de les coudre aux nouveaux brins). Les conditions  $A_2$  et  $A_3$  traduisent respectivement les points 2 et 3. La formalisation est évidente, nous avons juste à préciser que nous considérons seulement les brins de la cellule à chanfreiner. Enfin, la condition  $A_4$  est l'immédiate formalisation du point 4 du théorème.

#### 4.4. Avantages de l'approche CASL

La richesse syntaxique et sémantique de CASL et de ses bibliothèques nous a permis de spécifier la couture et le chanfreinage de manière très concise. Conformément à nos attentes, nos travaux en CASL permettent de poursuivre les précédentes utilisations des spécifications algébriques en modélisation géométrique en offrant une spécification complète dans un langage formel unique et donc doté d'une sémantique cohérente.

Nous avons été particulièrement séduit par la très grande souplesse de la syntaxe de CASL qui nous a permis d'obtenir une spécification concise, structurée et finalement très proche des définitions mathématiques. Cet avantage peut paraître à première vue

futile, mais il prend une grande importance du point de vue méthodologique. En effet, il rend possible la lecture des spécifications par les infographistes, et garantit ainsi leur validation.

Enfin CASL se prête particulièrement bien à l'abstraction. C'est son utilisation qui nous a permis de proposer un nouveau type de définitions plus abstraites des opérations topologiques.

#### **4.5. Inconvénients de l'approche CASL**

Contrairement à la spécification faite en B, notre spécification a été longue à établir. Ceci est dû au coût du travail de rétro-ingénierie pour obtenir une description observationnelle de l'opération de chanfreinage. Ce travail nous a permis de définir une réelle méthodologie de spécification des besoins des opérations topologiques locales. Bien sûr pour conforter totalement notre méthode, nous devons l'utiliser sur d'autres modèles topologiques que les  $n$ -G-cartes, et sur d'autres opérations complexes. Concernant la définition constructive du chanfreinage, nous aurions aussi pu la formaliser en CASL en introduisant toutes les opérations nécessaires pour la manipulation des ensembles et des fonctions. Nous aurions alors obtenu une spécification proche de la spécification B présentée ici.

Outre cet inconvénient lié à notre souci d'abstraction, nous sommes aussi tributaires d'un langage jeune encore susceptible d'évoluer. Par exemple, nous avons déjà dû modifier à plusieurs reprises nos spécifications pour prendre en compte des changements syntaxiques et de nouvelles bibliothèques. De plus les outils de preuve ne sont pas encore stabilisés et ne supportent pas des tactiques suffisantes pour élaborer des preuves importantes. Étant donnée la richesse du langage CASL, leur utilisation effective prochaine nous poussera sans doute à modifier le style de nos spécifications pour en tirer pleinement partie. Par exemple, les fonctions partielles et les sous-sortes coexistent dans CASL, alors qu'on peut généralement les substituer les unes aux autres. Il est donc probable qu'un outil soit plus adapté pour manipuler un concept plutôt que l'autre.

Enfin, d'un point de vue purement technique, nous aurions souhaité l'introduction d'une notion de prédicat faible, un prédicat faible étant un prédicat vérifié dès qu'un de ses argument est indéfini. Dans CASL, tous les prédicats définis par le spécifieur sont existentiels, c'est à dire non vérifiés sur des arguments indéfinis. C'est souvent ce qui convient, car un prédicat existentiel utilisé en précondition d'un axiome évite d'ajouter la précondition de définition de ses arguments. Par contre, avec notre schéma de spécification des opérations topologiques, ce n'est pas suffisant. Nous sommes fréquemment amenés à écrire des axiomes se réduisant à un prédicat dont l'un des arguments est un opération partielle, et à une précondition assurant que cette opération est définie. Avec des prédicats faibles, le second axiome de la spécification du chanfreinage se réduirait à *isChamfered*( $G.chamfering(d, i), G, d, i$ ). Étant donnée la

répétition de ce type de préconditions, une telle possibilité allégerait réellement nos spécifications.

### 5. Rétro-ingénierie et bilan méthodologique

Nous avons réalisé dans cet article un travail abouti de rétro-ingénierie de l'opération de chanfreinage. Cette opération centrale en modélisation géométrique, est représentative des problèmes rencontrés en modélisation géométrique. À la fois très utilisée et complexe, elle est en pratique souvent réalisée de façon approximative : elle est implantée par énumération de cas, chaque cas correspondant à une configuration d'objet particulière. Les différentes configurations se distinguent généralement par le nombre d'arêtes adjacentes à traiter et par l'alternance d'arêtes entrantes et sortantes au sommet à chanfreiner. Par exemple, arrondir quelques arêtes adjacentes au sommet S de l'objet de la figure 7 n'est pas faisable avec n'importe quel modèleur.



**FIG. 7.** *Un objet difficile à chanfreiner avec alternance d'arêtes entrantes et sortantes au sommet S et le résultat correct de son chanfreinage.*

En fait, dans la plupart des modèleurs commerciaux, certaines configurations sont répertoriées comme non traitées. Ce problème est inhérent à la définition par cas qui fait que tout nouveau cas doit être traité indépendamment. L'intérêt de la mise au point de la définition 5 et du théorème 1 est de proposer un cadre général, systématique et abstrait susceptible d'être une référence de correction pour toutes les configurations d'objets possibles. Cette expérience de rétro-ingénierie, menée de concert avec les experts du domaine, nous a permis de cerner les besoins du domaine. En particulier le résultat du théorème 1 est une caractérisation habituelle des besoins en modélisation géométrique. La caractérisation du chanfreinage donnée dans le théorème 1 n'est ni plus ni moins que l'analogue de la démarche suivie par la couture (voir le sous-chapitre 2.2) : on indique quelle cellule chanfreiner, ce qu'est le chanfreinage (insertion d'une face), comment cette face est reliée à l'objet initial, le reste de l'objet restant inchangé.

Avoir spécifié ainsi le chanfreinage nous apporte un retour significatif sur la caractérisation intrinsèque de l'opération de chanfreinage. Nous avons deux définitions mathématiques équivalentes à notre disposition, chacune d'elles étant très proche respectivement de la spécification B et de la spécification CASL. Nous avons ainsi un niveau de confiance élevé dans nos spécifications des besoins. L'intérêt principal des spécifications formelles est la capacité de raisonnement sur les besoins. En particulier, l'approche B permet de vérifier que le résultat du chanfreinage vérifie bien les invariants du modèle des 3-G-cartes. En outre, l'introduction d'une nouvelle manière d'appréhender le chanfreinage nous a permis d'obtenir une implantation directement

issue de la démarche préconisée dans la spécification. Celle-ci s'avère simple, structurée, locale à la cellule et même à chaque brin de la cellule, ce qui nous assure de son efficacité et de sa fiabilité. Il reste néanmoins à combler le fossé entre la spécification initiale et le programme final de manière formelle en exhibant par exemple une série de raffinements successifs.

## 6. Conclusion

Au delà de l'activité de rétro-ingénierie proprement dite sur l'opération de chanfreinage, l'autre motivation de notre travail est la recherche d'un langage de spécification adapté à la modélisation géométrique. Nous pourrions dès lors mener un développement formel des opérations des modélisateurs géométriques : partant d'un noyau de base déjà existant, on assurera fiabilité et efficacité des différentes opérations de haut niveau à ajouter au-dessus d'un tel noyau. Notre étude comparative a porté sur les langages B et CASL, qui sont de nature différente. De fait et à notre étonnement malgré tout, nous ne pouvons pas trancher en faveur de l'un ou l'autre, tant leurs avantages et inconvénients sont distincts (voir paragraphes 3.4, 3.5, 4.4 et 4.5). Un emploi plus avancé des outils permettrait d'argumenter plus finement dans l'optique de privilégier une approche au détriment de l'autre. Nous comptons donc poursuivre l'expérience, en étudiant complètement l'opération d'arrondi via CASL, et les opérations booléennes (intersection, union d'objets, etc) via la méthode B. Ces dernières sont aux opérations globales ce que l'arrondi est aux opérations locales, c'est-à-dire une opération cruciale pour le moment mal appréhendée. Une autre voie que nous envisageons est celle d'une approche formelle hétérogène, soit en faisant côtoyer des spécifications CASL et des machines B au sein d'une même spécification (par exemple CASL pour spécifier les opérations locales et B pour les opérations globales), ou soit, en raffinant la spécification CASL plus abstraite (cf. définition 5 et théorème 1) vers une machine B, cette dernière permettant à terme d'aboutir à du code fiable.

## 7. Bibliographie

- [A.A 01] A. ARNOULD F. L., « GeoSpec : specification libraries for geometric modelling », sept. 2001, <http://www.sic.sp2mi.univ-poitiers.fr/GL/GeoSpec>.
- [ABR 91] ABRIAL J. R., « The B reference manual », Edinburgh Portable Compilers, 17 Alva Street, Edinburgh EH2 4PH, UK, 1991.
- [ABR 96] ABRIAL J.-R., *The B-Book : Assigning Programs to Meanings*, Cambridge University Press, 1996.
- [BER 94] BERTRAND Y., DUFOURD J.-F., « Algebraic Specification of a 3D-Modeller Based on Hypermaps », *Computer vision, graphical model, and image processing*, vol. 56, n° 1, 1994, p. 29-60.
- [BID 00] BIDOIT M., MOSSES P. D., « A Gentle Introduction to CASL », Tutorial, CoFI Workshop at the 3rd European Joint Conferences on Theory and Practice of Software (ETAPS'2000), Berlin, Germany, avril 2000.

- [DUF 89] DUFOURD J.-F., « Algebraic Map-Based Topological Kernel for Polyhedron Models », *Conf. Eurographics*, Elsevier, 1989, p. 301-312.
- [DUF 97] DUFOURD J.-F., « Algebras and formal specifications in geometric modeling », *The Visual Computer*, vol. 13, 1997, p. 131-154, Springer-Verlag.
- [ELT 94] ELTER H., « Etude de structures combinatoires pour la représentation de complexes cellulaires », PhD thesis, Université de Strasbourg, 1994.
- [FUC 96] FUCHS L., BECHMANN D., BERTRAND Y., DUFOURD J.-F., « Formal specification for free-form curves and surfaces », *Spring Conference on Computer Graphics*, Bratislava, juin 1996.
- [LAN 00] ON LANGUAGE DESIGN C. C. F. I. T. G., « CASL The Common Algebraic Specification Language Summary », 2000, ftp://ftp.brics.dk/Projects/CoFI.
- [LED 00] LEDOUX F., MOTA J.-M., ARNOULD A., DUBOIS C., GALL P. L., BERTRAND Y., « Formal specification for a mathematics-based application domain : geometric modelling », rapport, 2000, Évry, ftp://ftp.lami.univ-evry.fr/pub/publications/reports/index.html.
- [LIE 94] LIENHARDT P., « N-dimensional generalized combinatorial maps and cellular quasi-manifolds », *International Journal of Computational Geometry and Applications*, , 1994.
- [MOS 99] MOSSES P. D., « CASL : A Guided Tour of Its Design », *LNCS*, vol. 1589, 1999, p. 216–240.
- [MOS 01] MOSSAKOWSKI T., ROGGENBACH M., SCHRÖDER L., SCHMIDT P., « CASL tool set (CATS) version 0.73 », sept. 2001, <http://www.tzi.de/cofi/CASL/CATS/download.html>.
- [ROG 00] ROGGENBACH M., MOSSAKOWSKI T., SCHRODER L., « Basic Datatypes in CASL », CoFI Note n° L-12-Version 0.4.1, May 2000, <http://www.brics.dk/Projects/CoFI/Notes/L-12/index.html>.
- [VÁR 94] VÁRADY T., MARTIN R. R., VIDA J., « A survey of blending methods that use parametric surfaces », *Computer-aided Design*, vol. 26, n° 5, 1994, p. 341–365, Elsevier Science.