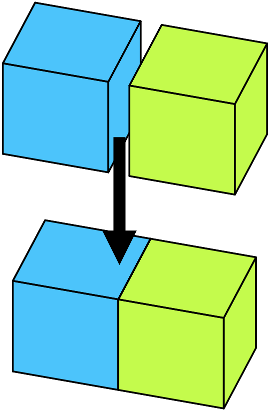
Transformations de graphes décorés

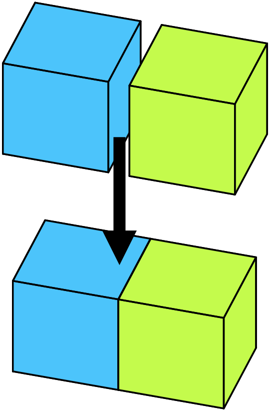
Application aux opérations de modélisation géométrique

Pascale Le Gall[1] Romain Pascual[1]
Hakim Belhaouari[2] Agnès Arnould[2]

[1] Laboratoire MICS, CentraleSupélec, Université Paris-Saclay
[2] Laboratoire XLIM, Université de Poitiers

22 juin 2023

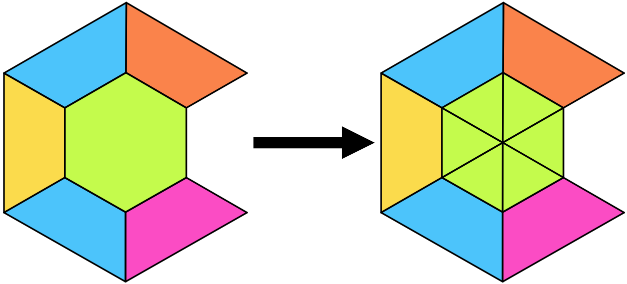




CGAL's sew operation

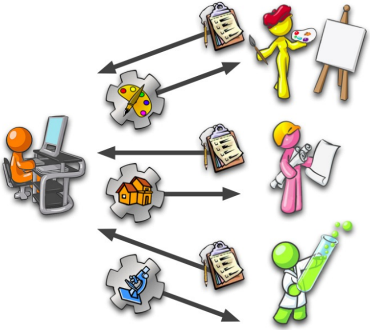
```
template<unsigned int i>
void sew(Dart_descriptor adart1, Dart_descriptor adart2)

  CGAL_assertion( i<=dimension );
  CGAL_assertion( (is_sewable<i>(adart1,adart2)) );
  size_type amark=get_new_mark();
  CGAL::GMap_dart_iterator_basic_of_involution<Self, i>
    I1(*this, adart1, amark);
  CGAL::GMap_dart_iterator_basic_of_involution<Self, i>
    I2(*this, adart2, amark);
  for ( ; I1.cont(); ++I1, ++I2 )
  {
    Helper::template Foreach_enabled_attributes_except
      <CGAL::internal::GMap_group_attribute_functor<Self, i>, i>::
      run(*this, I1, I2);
  }
  negate_mark( amark );
  for ( I1.rewind(), I2.rewind(); I1.cont(); ++I1, ++I2 )
  {
    basic_link_alpha<i>(I1, I2);
  }
  negate_mark( amark );
  CGAL_assertion( is_whole_map_unmarked(amark) );
  free_mark(amark);
}
```

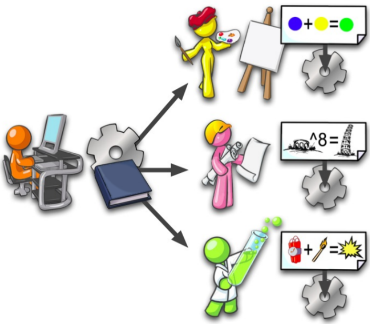


Ambition : définir un *domain-specific language* (DSL) pour la modélisation géométrique

Motivations : abstraction, performance, concision, correction, simplicité



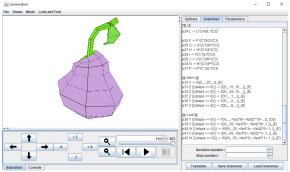
Approche Classique



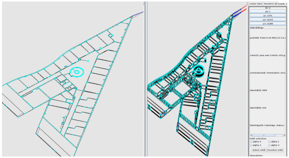
Jerboa [1]

1. Belhaouari et al. 2014

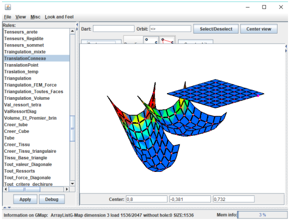
Croissance de plantes



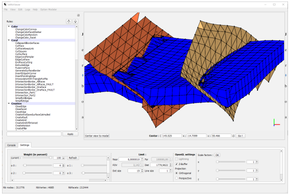
Architecture



Simulation physique



Géologie



Topologie vs géométrie

Objet

Topologie vs géométrie

Objet Topologie différente

Topologie vs géométrie

Objet Topologie différente Géométrie différente

Topologie

- Cartes généralisées
- Opérations topologiques

Géométrie

- Plongements
- Calculs

Jerboa

Topologie

- Représentation et modi cation de la structure des objets

Cellules topologiques

Catégorie(s) des graphes

Graphe

Morphisme de graphes

Forme une catégorie adhésive

Cartes généralisées [1]

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées [1]

Représentation par graphes

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées [1]

Représentation par graphes

Brin d identi e
sommet orange
arête violette
face verte

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées [1]

Représentation par graphes

0-arc

- sommets distincts
- mêmes arêtes et faces

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées [1]

Représentation par graphes

1-arc

- arêtes distinctes
- mêmes sommets et faces

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées [1]

Représentation par graphes

2-arc

- faces distinctes
- mêmes sommets et arêtes

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Cartes généralisées [1]

Superposition des graphes

Légende : 0, 1, 2

1. LIENHARDT 1989 ; DAMIAND et al. 2014

Exercice

Donner la carte généralisée associée à l'objet suivant

Exercice

Donner la carte généralisée associée à l'objet suivant

Question : Comment décrire formellement une carte généralisée (sous forme de graphe) ?

Question : Comment décrire formellement une carte généralisée (sous forme de graphe) ?

Pour rappel, voici la dé nition combinatoire d'une carte généralisée (cours de Samuel Peltier)

Une n-Gcarte (n 0), est un $(n + 2)$ -tuple $(B; \;_0; \; ; \;_n)$ tel que :

- B est un ensemble ni de brins
- 8i 2 0::n; $\;_i$ est une involution sur B
- 8i 2 0::(n 2); 8j 2 (i + 2) ::n; $\;_i \;_j$ est une involution sur B

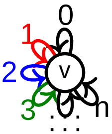
Graphes topologiques

Graphes topologiques : arcs typés par une dimension de 0::n

Graphes topologiques : arcs typés par une dimension de 0::n

n-Graph := Graph =TG

avec TG =



Contraintes topologiques

Gcartes : n-Graph avec contraintes topologiques

Contraintes topologiques

Gcartes : n-Graph avec contraintes topologiques

non-orientation chaque arc admet un unique inverse

Contraintes topologiques

Gcartes : n-Graph avec contraintes topologiques

non-orientation chaque arc admet un unique inverse

arcs incidents chaque nœud est incident à un unique arc par dimension i 2 0::n

Contraintes topologiques

Gcartes : n-Graph avec contraintes topologiques

non-orientation chaque arc admet un unique inverse

arcs incidents chaque nœud est incident à un unique arc par dimension i 2 0::n

cycles pour i , j tels que i + 2 j , tout chemin ijij est un cycle

Contrainte d'arcs incidents

chaque nœud est incident à un unique arc par dimension i 2 0::n

Traduction de ¡ permutation
! cours de Samuel Peltier

involution car graphe non-orienté

Légende : 0, 1, 2

Contrainte de cycle

pour i , j tels que i + 2 j , tout chemin ijij est un cycle

Traduction de i j
permutation
! cours de Samuel Peltier

involution car graphe non-orienté

Légende : 0, 1, 2

Digression sur les contraintes

Contraintes exprimées dans le `langage courant'

Reformulation possible en logique du premier ordre, à la Courcelle [1]

Représentation graphique des contraintes par `nested conditions' [2]

1. Courcelle 1997
2. Habel et al. 2009 ; Ehrig et al. 2014 ; Behr et al. 2021

Orbites et cellules topologiques

Orbite : Sous-graphe induit par un sous-ensemble $\langle o \rangle$ de dimensions

- Sommets : orbites $\langle 1; 2 \rangle$

Légende : 0, 1, 2

Orbites et cellules topologiques

Orbite : Sous-graphe induit par un sous-ensemble $\langle o \rangle$ de dimensions

- Sommets : orbites $\langle 1; 2 \rangle$
- Arêtes : orbites $\langle 0; 2 \rangle$

Légende : 0, 1, 2

Orbites et cellules topologiques

Orbite : Sous-graphe induit par un sous-ensemble $\langle o \rangle$ de dimensions

- Sommets : orbites $\langle 1; 2 \rangle$
- Arêtes : orbites $\langle 0; 2 \rangle$
- Faces : orbites $\langle 0; 1 \rangle$

Légende : 0, 1, 2

Rappel transformation de graphes [1]

1. Rozenberg 1997 ; Ehrig et al. 2006 ; Heckel et al. 2020

Rappel transformation de graphes [1]

1. Rozenberg 1997 ; Ehrig et al. 2006 ; Heckel et al. 2020

Rappel transformation de graphes [1]

1. ROZENBERG 1997 ; EHRIG et al. 2006 ; HECKEL et al. 2020

Rappel transformation de graphes[1]

1. Rozenberg 1997 ; Ehrig et al. 2006 ; Heckel et al. 2020

Rappel sur la réécriture par DPO

Règle $p = L \overset{l}{\leftarrowtail} K \overset{r}{\rightarrowtail} R$, match $m : L \hookrightarrow G$

Alors $G \overset{r,m}{\Longrightarrow} H$ si

$$\begin{tikzcd}
L \arrow[d, tail, "m"'] \arrow[dr, phantom, "PO"] & K \arrow[l, tail, "l"'] \arrow[r, tail, "r"] \arrow[d, tail] \arrow[dr, phantom, "PO"] & R \arrow[d, tail, "m'"] \\
G & D \arrow[l, tail] \arrow[r, tail] & H
\end{tikzcd}$$

existe

Opération de modélisation géométrique

Opération de modélisation géométrique

$$G \Rightarrow^{r,m} G'$$

avec

- r règle de réécriture de graphes
- m match

où G et G' sont des Gcartes bien formées

Réécriture d'orbites

Réécriture d'orbites

Réécriture d'orbites

Réécriture d'orbites

Réécriture d'orbites

Réécriture d'orbites

Réécriture d'orbites

Réécriture d'orbites

Produits de graphes [1]

1. inspiré de Bauderon 1995

Produits de graphes [1]

1. inspiré de BAUDERON 1995

Produits de graphes [1]

1. inspiré de BAUDERON 1995

Produits de graphes [1]

- $(\ \ ; P)$: instanciation

1. inspiré de Bauderon 1995

Produits de graphes [1]

- $(\ ;P)$: instanciation
- E : foncteur de plongement

1. inspiré de BAUDERON 1995

Produits de graphes [1]

- (;P) : instanciation
- E : foncteur de plongement

1. inspiré de BAUDERON 1995

Produits de graphes [1]

- (;P) : instanciation
- E : foncteur de plongement
- : foncteur de projection

1. inspiré de BAUDERON 1995

Construction complète

Construction complète

Construction complète

Préservation de la consistance

Modi cation d'un objet bien formé vers un objet bien formé

correction de typage d'un programme

Préservation de la consistance

Modification d'un objet bien formé vers un objet bien formé

correction de typage d'un programme

Soit une contrainte c, une règle $r = L \hookleftarrow K \hookrightarrow R$ et un morphisme $m : L \hookrightarrow G$.

Si $G \models c$, comment assurer que $H \models c$ avec $H := G \Rightarrow^{r,m} H$?

Préservation de la consistance

Modification d'un objet bien formé vers un objet bien formé

correction de typage d'un programme

Soit une contrainte c, une règle $r = L \hookleftarrow K \hookrightarrow R$ et un morphisme $m : L \hookrightarrow G$.

Si $G \models c$, comment assurer que $H \models c$ avec $H := G \Rightarrow^{r,m} H$?

$\rightarrow$ Retour utilisateur

Inconsistances topologiques

Contrainte de cycle : les chemins 0202 sont des cycles

Inconsistances topologiques

Contrainte de cycle : les chemins 0202 sont des cycles

Inconsistances topologiques

Contrainte de cycle : les chemins 0202 sont des cycles

Inconsistances topologiques

Contrainte de cycle : les chemins 0202 sont des cycles

Inconsistances topologiques

Contrainte de cycle : les chemins 0202 sont des cycles

Intuition

Intuition

Intuition

Véri cation statique et syntaxique

Non-orientation

Par convention on manipule toujours un arc et son inverse

Véri cation statique et syntaxique

Non-orientation
Par convention on manipule toujours un arc et son inverse

Arcs incidents

nœuds créés contrainte imposée

nœuds préservés préservation des incidences

Véri cation statique et syntaxique

Non-orientation
Par convention on manipule toujours un arc et son inverse

Arcs incidents

nœuds créés contrainte imposée

nœuds préservés préservation des incidences

Cycles

nœuds créés contrainte imposée

nœuds préservés préservation des chemins partiels

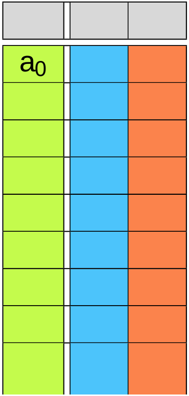
On peut maintenant considérer un schéma de règle S et qu'on applique le foncteur (; P) à S.

Question :

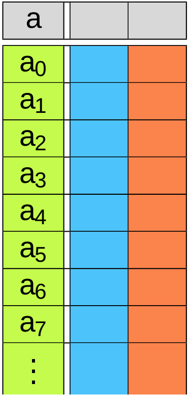
Comment récupérer P ?
Comment résoudre l'unicité à isomorphisme près ?

Instanciation : approche constructive

Instanciation : approche constructive



Instanciation : approche constructive



Instanciation : approche constructive

a		
a_0		
a_1		
a_2		
a_3		
a_4		
a_5		
a_6		
a_7		
$\vdots$		

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Instanciation : approche constructive

a	b	c
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4
a_5	b_5	c_5
a_6	b_6	c_6
a_7	b_7	c_7
$\vdots$	$\vdots$	$\vdots$

Algorithme d'application des règles

1. Récupérer l'orbite de support du hook principal
2. Par parcours de la structure, constuire le tableau décrivant le membre gauche
3. Créer les brins correspondant aux nouveaux nœuds
4. Construire les arcs de renommage de l'orbite (liens verticaux), en terminant par le hook principal
5. Eliminer les brins des nœuds supprimés
6. Construire les arcs entre copies de nœuds distincts (liens horizontaux)

Géométrie

- Représentation et modi cation des valeurs associées aux objets

Géométrie

Géométrie

Géométrie

Géométrie

Plongements

Légende : 0, 1, 2

Plongements

Plongement : fonction
: ho i ! avec un type de
données abstrait

Légende : 0, 1, 2

Plongements

Plongement : fonction
: h o i ! avec un type de données abstrait

- position : h 1; 2 i ! Point3

Légende : 0, 1, 2

Plongements

Plongement : fonction

$: \langle o \rangle \to$ avec un type de données abstrait

- position $: \langle 1, 2 \rangle \to$ Point3
- color $: \langle 0, 1 \rangle \to$ ColorRGB

Légende : 0, 1, 2

Question : Quelle représentation (abstraite) pour les plongements ?

Rappel sur les types de données (Syntaxe)

Signature $= (S; F)$ d'un type de données

- ensemble S de noms de types
- ensemble F de noms de fonctions munis d'un pro l
  $: F \to S^+$

Pour $f \in F$ tel que $(f) = (s_1; \ldots; s_m; s)$, on note

$$f : s_1 \quad \ldots \quad s_m \to s$$

Forment une catégorie (les morphismes sont les fonctions préservant les pro ls)

$\Sigma(\mathit{position}; \mathit{color})$

$$S(\mathit{position}; \mathit{color}) = \{\mathrm{Point3}; \mathrm{Vector3}; \mathrm{ColorRGB}\}$$

$$F(\mathit{position}; \mathit{color}) = \{\mathrm{plus}; \mathrm{midpoint}; \mathrm{mix}\}$$

- $\mathrm{plus} : \mathrm{Point3} \times \mathrm{Vector3} \to \mathrm{Point3}$
- $\mathrm{midpoint} : \mathrm{Point3} \times \mathrm{Point3} \to \mathrm{Point3}$
- $\mathrm{mix} : \mathrm{ColorRGB} \times \mathrm{ColorRGB} \to \mathrm{ColorRGB}$

Rappel sur les types de données (Sémantique)

-algèbre pour une signature $= (S; F)$

- famille d'ensembles de valeurs $(A_s)_{s \in S}$
- fonction $f^A : A_{s_1} \ldots A_{s_m} \to A_s$ pour tout nom de fonction $f : s_1 \ldots s_m \to s$ de F

Forment une catégorie (les morphismes sont les fonctions préservant la structure)

Algèbre $A(\text{position};\text{color})$

- $A(\text{position};\text{color})_{\text{Point3}} = \mathbb{R}^3$
- $A(\text{position};\text{color})_{\text{Vector3}} = \mathbb{R}^3$
- $A(\text{position};\text{color})_{\text{ColorRGB}} = [0;1]^3$

- $\text{plus}^{A(\text{position};\text{color})}$: translation d'un point par un vecteur
- $\text{midpoint}^{A(\text{position};\text{color})}$: milieu du segment
- $\text{mix}^{A(\text{position};\text{color})}$: mélange de deux couleurs

Rappel sur les graphes attribués

Graphe -attribué $AG = (G; A)$ est dé ni par

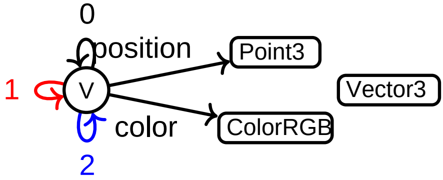
- E-graphe $G = (V_G; D_G; E_G; A_G; s_G; t_G; sa_G; ta_G)$
- -algèbre A telle que $D_G = F_{s2S} A_s$

Gartes plongées sur position et color

On considère la catégorie

$$(\text{position} ; \text{color})\text{-AGraph} = (\text{ETG} ; 1_{\text{Alg} ((\text{position} ; \text{color}))})$$

avec ETG =

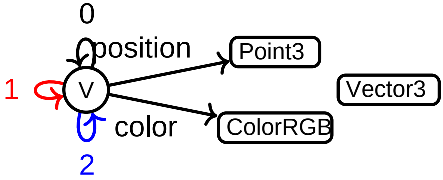


Gartes plongées sur position et color

On considère la catégorie

$$(\ \text{position}\ ;\text{color}\ )\text{-AGraph} = (\text{ETG}\ ;1_{\text{Alg}\ ((\ \text{position}\ ;\text{color}\ ))})$$

avec ETG =

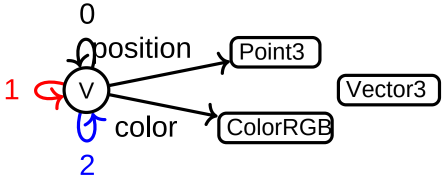


Contraintes pour l'existence et l'unicité des valeurs

Gartes plongées sur position et color

On considère la catégorie

$$(\ \text{position} \ ; \text{color}\ )\text{-AGraph} = (\text{ETG} ; 1_{\text{Alg}\ ((\ \text{position}\ ;\text{color}\ ))})$$

avec ETG = 

Contraintes pour l'existence et l'unicité des valeurs

Contrainte de cohérence au type d'orbite

Question : Quel type de données pour la modélisation géométrique ?

Question : Quel type de données pour la modélisation géométrique ?

Signature et algèbre étendues `à la volée'

Question : Comment modi er le type de données ?

Question : Quel type de données pour la modélisation géométrique ?

Signature et algèbre étendues `à la volée'

Question : Comment modi er le type de données ?

Ajout d'un nom de type (resp. nom de fonction) dans la signature et d'un support (resp. d'une fonction) dans l'algèbre

! implémentation de la fonction

Modi cation d'un attribut

Type de données algébrique

- Point3 , Vector3 , ColorRGB, ...
- plus , midpoint , mix, ...

Modi cation d'un attribut

Etendu avec des opérateurs de structure

- Accesseurs
  a: position = A
  a: color =

Modi cation d'un attribut

Etendu avec des opérateurs de structure

- Accesseurs
- Opérateurs de voisinage

  a@0@1@0: position =

  f: position = C

  a@1@0: color = c: color =

Modi cation d'un attribut

Etendu avec des opérateurs de structure

- Accesseurs
- Opérateurs de voisinage
- Opérateurs de collecte
  $\text{position}_{\langle 0;1\rangle}(a) = [\![A;B;C;D]\!]$
  $\text{color}_{\langle 0;1\rangle}(a) = [\![\quad]\!]$

Extension aux schémas

Extension aux schémas

Extension aux schémas

Extension aux schémas

$$\frac{1}{4}(A + B + C + D)$$

Extension aux schémas

$$\frac{1}{4}(A + B + C + D)$$
$$= \mathrm{middle}\,([\![A; B; C; D]\!])$$

Extension aux schémas

$$
\begin{aligned}
&\tfrac{1}{4}(A + B + C + D) \\
= \;& \mathrm{middle}\,(\llbracket A; B; C; D \rrbracket) \\
= \;& \mathrm{middle}\,(\mathrm{position}\;\; \langle 0; 1 \rangle\,(a))
\end{aligned}
$$

Extension aux schémas

$$
\begin{aligned}
&\tfrac{1}{4}(A + B + C + D) \\
&= \mathrm{middle}\,(\llbracket A; B; C; D \rrbracket) \\
&= \mathrm{middle}\,(\mathrm{position}\quad \langle 0; 1 \rangle\,(a)) \\
&= \mathrm{middle}\,(\mathrm{position}\quad \langle 0; 1 \rangle\,(n0))
\end{aligned}
$$

Inconsistance géométrique

Constrainte brins d'une $\langle 0; 1 \rangle$-orbite ont la même couleur

Inconsistance géométrique

Constrainte brins d'une h0; 1i -orbite ont la même couleur

Inconsistance géométrique

Constrainte brins d'une $\langle 0, 1 \rangle$-orbite ont la même couleur

D'un langage abstrait vers un langage concret

Conclusion

Dans cette séance nous avons évoqué une mise en application les transformations de graphes dans un domaine spéci que

- la modélisation géométrique
  - Adapter la théorie générale à nos besoins
  - Construction abstraite dédiées
  - Implémentation par des algorithmes ef caces

Suite : mise en application

Jerboa

- Une plateforme de conception de modeleurs géométriques

References I

BAUDERON, Michel (1er jan. 1995). « Parallel Rewriting of Graphs through the Pullback Approach ». In : Electronic Notes in Theoretical Computer Science. SEGRAGRA 1995 2, p. 19-26. ISSN : 1571-0661. DOI : 10.1016/S1571-0661(05)80176-8.

BEHR, Nicolas et al. (22 avr. 2021). « Compositionality of Rewriting Rules with Conditions ». In : Compositionality 3.2, p. 51. ISSN : 2631-4444. DOI : 10.32408/compositionality-3-2. URL : https://compositionality-journal.org/.

BELHAOUARI, Hakim et al. (2014). « Jerboa: A Graph Transformation Library for Topology-Based Geometric Modeling ». In : Graph Transformation. ICGT 2014. Sous la dir. d'Holger GIESE et al. Lecture Notes in Computer Science. Cham : Springer International Publishing, p. 269-284. ISBN : 978-3-319-09108-2. DOI : 10.1007/978-3-319-09108-2_18.

References II

- Courcelle, B. (fév. 1997). « The expression of graph properties and graph transformations in monadic second-order logic ». In : *Handbook of Graph Grammars and Computing by Graph Transformation: Volume I. Foundations*. USA : WORLD SCIENTIFIC, p. 313-400. ISBN : 978-981-02-2884-2. DOI : 10.1142/9789812384720_0005.
- Damiand, Guillaume et al. (19 sept. 2014). *Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing*. CRC Press. 407 p. ISBN : 978-1-4822-0652-4.
- Ehrig, Hartmut et al. (2006). *Fundamentals of Algebraic Graph Transformation*. Monographs in Theoretical Computer Science. An EATCS Series. Berlin Heidelberg : Springer-Verlag. ISBN : 978-3-540-31187-4. DOI : 10.1007/3-540-31188-2.

References III

- EHRIG, Hartmut et al. (août 2014). « M-adhesive transformation systems with nested application conditions. Part 1: parallelism, concurrency and amalgamation ». In : *Mathematical Structures in Computer Science* 24.4. ISSN : 0960-1295, 1469-8072. DOI : 10.1017/S0960129512000357.
- HABEL, Annegret et al. (avr. 2009). « Correctness of high-level transformation systems relative to nested conditions ». In : *Mathematical Structures in Computer Science* 19.2. Publisher: Cambridge University Press, p. 245-296. ISSN : 1469-8072, 0960-1295. DOI : 10.1017/S0960129508007202.

References IV

- HECKEL, Reiko et al. (2020). *Graph Transformation for Software Engineers: With Applications to Model-Based Development and Domain-Specific Language Engineering*. Cham : Springer International Publishing. ISBN : 978-3-030-43915-6. DOI : 10.1007/978-3-030-43916-3. URL : http://link.springer.com/10.1007/978-3-030-43916-3 (visité le 25/06/2020).
- LIENHARDT, Pascal (5 juin 1989). « Subdivisions of N-dimensional Spaces and N-dimensional Generalized Maps ». In : *Proceedings of the Fifth Annual Symposium on Computational Geometry*. SCG '89. New York, NY, USA : Association for Computing Machinery, p. 228-236. ISBN : 978-0-89791-318-8. DOI : 10.1145/73833.73859.

References V

ROZENBERG, Grzegorz, éd. (fév. 1997). *Handbook of Graph Grammars and Computing by Graph Transformation: Volume 1: Foundations*. T. Foundations. 1 t. USA : World Scientific. 572 p. ISBN : 978-981-02-2884-2. DOI : 10.1142/3303.