Representing Topological Structures Using Cell-Chains

David Cardoze **Gary Miller** Todd Phillips
Computer Science Department
Carnegie Mellon University

GMP 2006
July 27, 2006

Geometric Cells

A d-cell is a region **homeomorphic** to the open d-ball

$$\mathcal{B}^d = \{x \in \mathcal{R}^d : |x| < 1\}$$

Geometric Cells

A d-cell is a region **homeomorphic** to the open d-ball

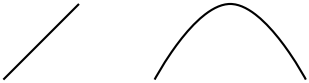
$$\mathcal{B}^d = \{x \in \mathcal{R}^d : |x| < 1\}$$

- 0-cell

Geometric Cells

A d-cell is a region **homeomorphic** to the open d-ball

$$\mathcal{B}^d = \{x \in \mathcal{R}^d : |x| < 1\}$$

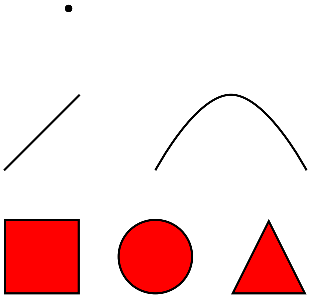
- 0-cell 
- 1-cells 

Geometric Cells

A d-cell is a region **homeomorphic** to the open d-ball

$$\mathcal{B}^d = \{x \in \mathcal{R}^d : |x| < 1\}$$

- 0-cell
- 1-cells
- 2-cells

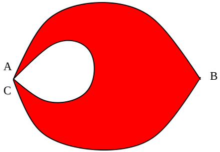


Geometric Cells

d-cell is a region **homeomorphic** to the open d-ball

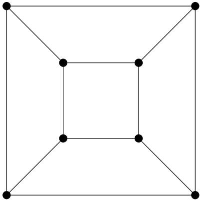
$$\mathcal{B}^d = \{x \in \mathcal{R}^d : |x| < 1\}$$

Pinched 2-cell:

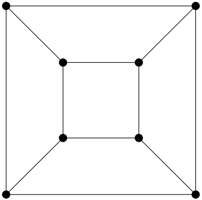


What is a Cellular Decomposition?

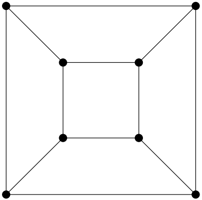
What is a Cellular Decomposition?

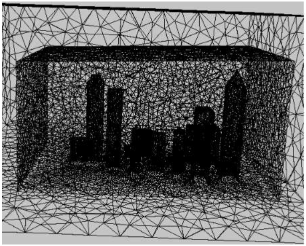


What is a Cellular Decomposition?



What is a Cellular Decomposition?



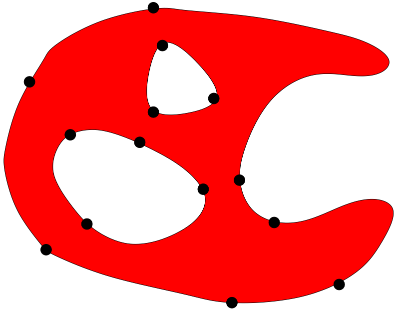


Basic Idea

- Represent a **geometric** domain **decomposed** into **basic** building blocks.

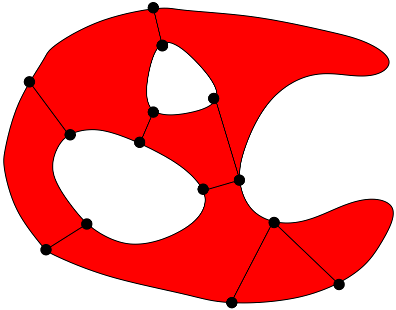
Basic Idea

- Represent a **geometric** domain **decomposed** into **basic** building blocks.



Basic Idea

- Represent a **geometric** domain **decomposed** into **basic** building blocks.



Our Data Structure

- Well defined Semantics.

Our Data Structure

- Well defined Semantics.
- Easy to determine local structure of the Cell Complex.

Our Data Structure

- Well defined Semantics.
- Easy to determine local structure of the Cell Complex.
- Easy to test if the Complex is well formed.

Our Data Structure

- Well defined Semantics.
- Easy to determine local structure of the Cell Complex.
- Easy to test if the Complex is well formed.
- Cells are first class objects.

Our Data Structure

- Well defined Semantics.
- Easy to determine local structure of the Cell Complex.
- Easy to test if the Complex is well formed.
- Cells are first class objects.
- Works in any fixed dimension.

Our Data Structure

- Well defined Semantics.
- Easy to determine local structure of the Cell Complex.
- Easy to test if the Complex is well formed.
- Cells are first class objects.
- Works in any fixed dimension.
- Space and Time efficient.

Data Structures Galore

Data Structure	Represents
Arcs *[Edmonds60]*	2D Surfaces
Crosses *[Tutte73]*	2D Unoriented Surfaces
Winged-Edge *[Bau75]*	3-d Polyhedra
Doubly-Linked Edge List *[MP78]*	Planar Subdivisions
Quad-Edge *[GS85]*	2D Surfaces
Facet-Edge *[DL87]*	Pseudo-Manifold Complexes
Radial-Edge *[Wei88]*	Non-Manifold B-Reps
Cell-Tuple-Complex *[Bri93]*	Regular Manifolds
nG-maps *[Lie94]*	Cellular Quasi-Manifolds
Cell-Chain-Complex *[CMP]*	Pseudo-Regular Manifolds

Data Structures Galore

Data Structure	Represents
Arcs *[Edmonds60]*	2D Surfaces
Crosses *[Tutte73]*	2D Unoriented Surfaces
Winged-Edge *[Bau75]*	3-d Polyhedra
Doubly-Linked Edge List *[MP78]*	Planar Subdivisions
Quad-Edge *[GS85]*	2D Surfaces
Facet-Edge *[DL87]*	Pseudo-Manifold Complexes
Radial-Edge *[Wei88]*	Non-Manifold B-Reps
Cell-Tuple-Complex *[Bri93]*	Regular Manifolds
*n*G-maps *[Lie94]*	Cellular Quasi-Manifolds
Cell-Chain-Complex *[CMP]*	Pseudo-Regular Manifolds

Combinatorial Data Structures

Data Structures Galore

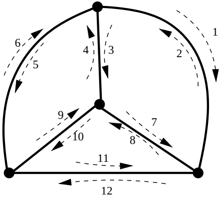
Data Structure	Represents
Arcs *[Edmonds60]*	2D Surfaces
Crosses *[Tutte73]*	2D Unoriented Surfaces
Winged-Edge *[Bau75]*	3-d Polyhedra
Doubly-Linked Edge List *[MP78]*	Planar Subdivisions
Quad-Edge *[GS85]*	2D Surfaces
Facet-Edge *[DL87]*	Pseudo-Manifold Complexes
Radial-Edge *[Wei88]*	Non-Manifold B-Reps
Cell-Tuple-Complex *[Bri93]*	Regular Manifolds
nG-maps *[Lie94]*	Cellular Quasi-Manifolds
Cell-Chain-Complex *[CMP]*	Pseudo-Regular Manifolds

Group Theoretic

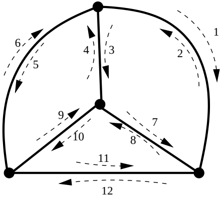
1960 Edmonds': Graphs on Surface

- An abstract set A of **arcs**(two for each edge) and two permutations ϕ and R of A:

1960 Edmonds': Graphs on Surface

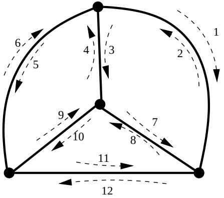
- An abstract set A of **arcs**(two for each edge) and two permutations ϕ and R of A:
- 

1960 Edmonds': Graphs on Surface

- An abstract set A of **arcs**(two for each edge) and two permutations ϕ and R of A:
- 

- **eg:** $\phi = (2,3,7)(4,5,9)(8,10,11)(1,12,6)$
  $R = (1,2)(3,4)(5,6)(7,8)(9,10)(11,12)$

1960 Edmonds': Graphs on Surface

- An abstract set A of **arcs**(two for each edge) and two permutations ϕ and R of A:
- 

- **eg:** $\phi = (2, 3, 7)(4, 5, 9)(8, 10, 11)(1, 12, 6)$
  $R = (1, 2)(3, 4)(5, 6)(7, 8)(9, 10)(11, 12)$
- $R^2 = id$ and fixed-point-free (fpf).

Edmonds' Graph on Surface

- ϕ traverses each face in CCW order

Edmonds' Graph on Surface

- ϕ traverses each face in CCW order
- R reverses each edge.

Edmonds' Graph on Surface

- ϕ traverses each face in CCW order
- R reverses each edge.
- The permutation $\phi^* = \phi R$ is the arcs with same tail in CW order.

Edmonds' Graph on Surface

- ϕ traverses each face in CCW order
- R reverses each edge.
- The permutation $\phi^* = \phi R$ is the arcs with same tail in CW order.
- Edges $=$ orbits(R)
  Faces $=$ orbits(ϕ)
  Vertices $=$ orbits(ϕ^*)
  Connected Components $=$ orbits($< \phi, R >$)

Edmonds' Graph on Surface

- ϕ traverses each face in CCW order
- R reverses each edge.
- The permutation $\phi^* = \phi R$ is the arcs with same tail in CW order.
- Edges = orbits(R)
  Faces = orbits(ϕ)
  Vertices = orbits(ϕ^*)
  Connected Components = orbits($< \phi, R >$)
- This is an Implicit representation!

Implicit Models

- There is only one basic class of objects.

Implicit Models

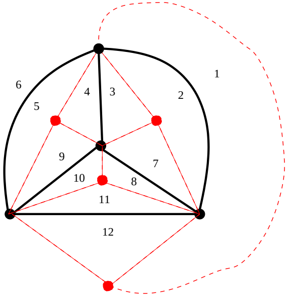
- There is only one basic class of objects.
- Faces are represented implicitly in terms of operations on these objects.

Modern View of Edmonds'
Holistic View

- We add a point to the "middle of each face and connect each face-vertex to vertices on the face.

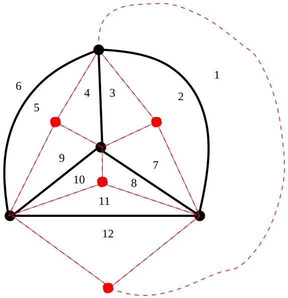
Modern View of Edmonds'
Holistic View

- We add a point to the "middle of each face and connect each face-vertex to vertices on the face.



Modern View of Edmonds'
Holistic View

- We add a point to the “middle of each face and connect each face-vertex to vertices on the face.



- The permutations now permute the new triangles.

Tutte's Representation of Surfaces

- An abstract set S of **crosses** and three permutations P, θ and ϕ of S satisfying:
- $\theta^2 = \phi^2 = I$ and $\theta\phi = \phi\theta$.
- X, θX, ϕX and $\theta\phi X$ are all distinct for all crosses X.
- $(P\theta)^2 = I$.
- The orbits of X and θX under P are distinct for all crosses X.

Tutte's Representation of Surfaces

- An abstract set S of **crosses** and three permutations P, θ and ϕ of S satisfying:
- $\theta^2 = \phi^2 = I$ and $\theta\phi = \phi\theta$.
- X, θX, ϕX and $\theta\phi X$ are all distinct for all crosses X.
- $(P\theta)^2 = I$.
- The orbits of X and θX under P are distinct for all crosses X.
- Modern View: Tutte had right group but wrong set of generators.

Beyond 1960, 1970

Beyond 1960, 1970

- Group Theoretic Approach is fine but:

Beyond 1960, 1970

- Group Theoretic Approach is fine but:
- Cells should be “first class” objects.

Beyond 1960, 1970

- Group Theoretic Approach is fine but:
- Cells should be "first class" objects.
- Represent more than two-dimensional decompositions.

Beyond 1960, 1970

- Group Theoretic Approach is fine but:
- Cells should be "first class" objects.
- Represent more than two-dimensional decompositions.
- Rely on the same general principle; operators acting on very primitive objects.

The Cell-Complex Data Structure
[Bri93]

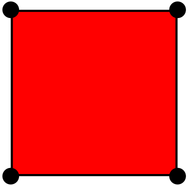
Brisson: Barycentric Subdivisions

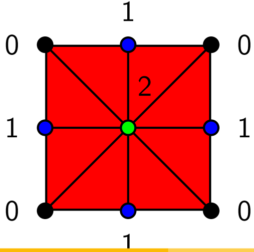
Creating a **triangulation** of the cell complex .

- Add a new vertex interior to each edge and face. Connect two vertices if they are visible to each other.

Brisson: Barycentric Subdivisions

Creating a **triangulation** of the cell complex .

- Add a new vertex interior to each edge and face. Connect two vertices if they are visible to each other.
- 



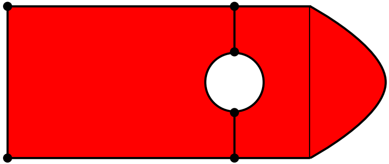
Why Barycentric Subdivisions?

- The permutation will act on the simplices of the barycentric subdivision.

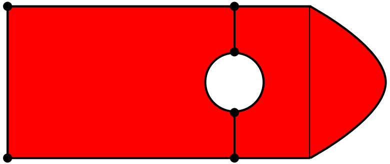
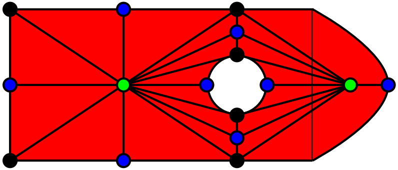
Why Barycentric Subdivisions?

- The permutation will act on the simplices of the barycentric subdivision.
- Giving simple higher dimensional representations, Brisson.

Why Barycentric Subdivisions?

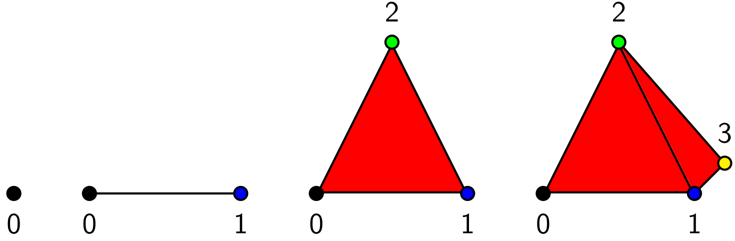
- The permutation will act on the simplices of the barycentric subdivision.
- Giving simple higher dimensional representations, Brisson.
- 

Why Barycentric Subdivisions?

- The permutation will act on the simplices of the barycentric subdivision.
- Giving simple higher dimensional representations, Brisson.
- 
- 

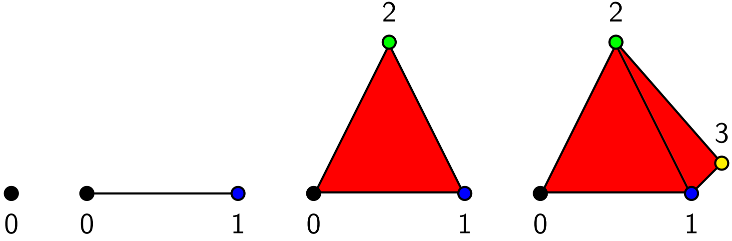
Numbered Simplices and Gluings

- A numbered d-simplex is a simplex whose vertices are uniquely labeled with numbers between 0 and d.



Numbered Simplices and Gluings

- A numbered d-simplex is a simplex whose vertices are uniquely labeled with numbers between 0 and d.

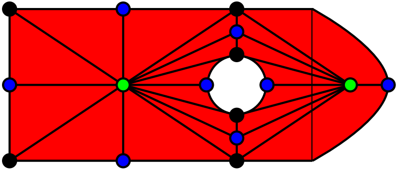


- Observe: The simplices in the Barycentric are numbered.

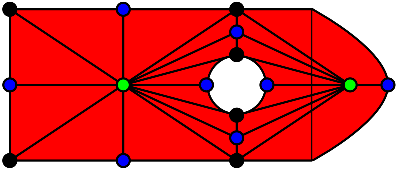
Numbered Simplicial Sets

- A numbered d-simplicial set is a collection numbered d-simplices glued along $(d-1)$-faces with compatible labels.

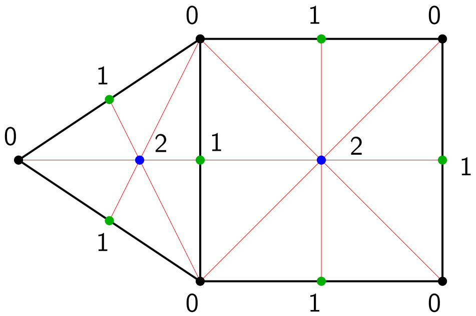
Numbered Simplicial Sets

- A numbered d-simplicial set is a collection numbered d-simplices glued along $(d-1)$-faces with compatible labels.
- 

Numbered Simplicial Sets

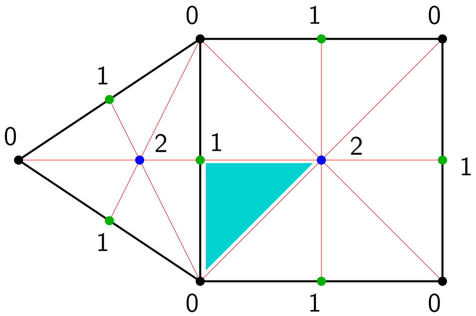
- A numbered d-simplicial set is a collection numbered d-simplices glued along $(d-1)$-faces with compatible labels.
- 
- The set of gluings with the same label/numbers can be viewed as matching or permutation(involution).

Moving Around



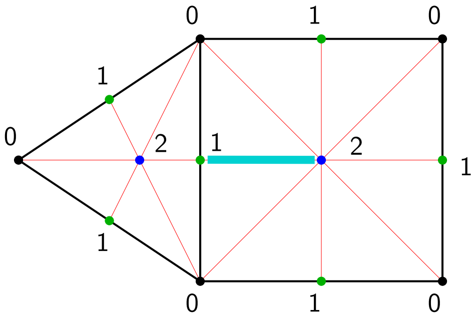
- Simplex Reflection. Remove a vertex, take the simplex on the other side.

Moving Around



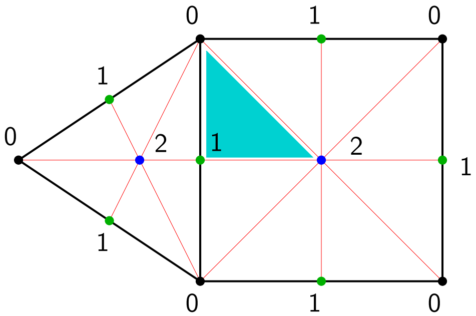
- Simplex Reflection. Remove a vertex, take the simplex on the other side.

Moving Around



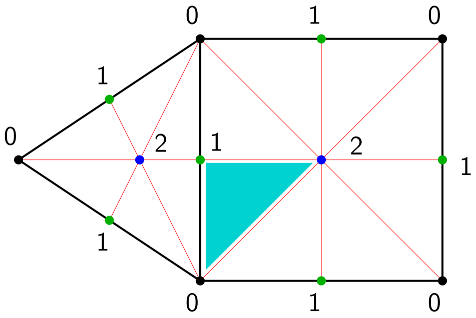
- Simplex Reflection. Remove a vertex, take the simplex on the other side.

Moving Around



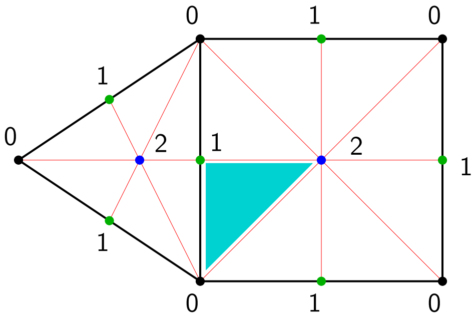
- Simplex Reflection. Remove a vertex, take the simplex on the other side.

Moving Around



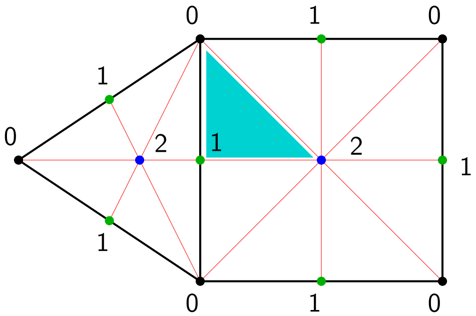
- Simplex Reflection. Remove a vertex, take the simplex on the other side.
- The reflection can be chosen based on vertex number.

Moving Around



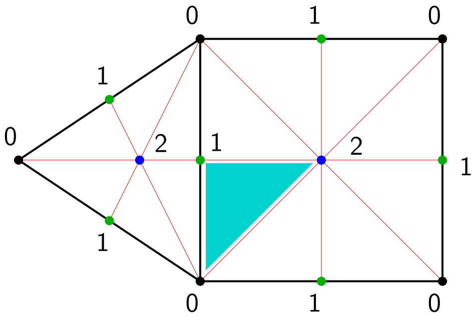
- Simplex Reflection. Remove a vertex, take the simplex on the other side.
- The reflection can be chosen based on vertex number.
- We will denote these permutations as α_i for $i \in \{0 \dots d\}$. (aka $\texttt{switch}_i$)

Moving Around



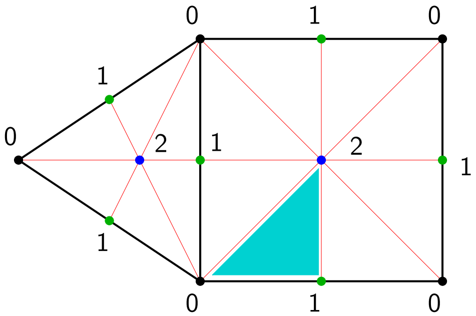
- Simplex Reflection. Remove a vertex, take the simplex on the other side.
- The reflection can be chosen based on vertex number.
- We will denote these permutations as α_i for $i \in \{0 \ldots d\}$. (aka $\texttt{switch}_i$)

Moving Around



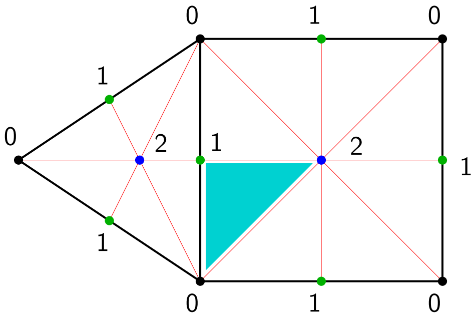
- Simplex Reflection. Remove a vertex, take the simplex on the other side.
- The reflection can be chosen based on vertex number.
- We will denote these permutations as α_i for $i \in \{0 \dots d\}$. (aka $\texttt{switch}_i$)

Moving Around



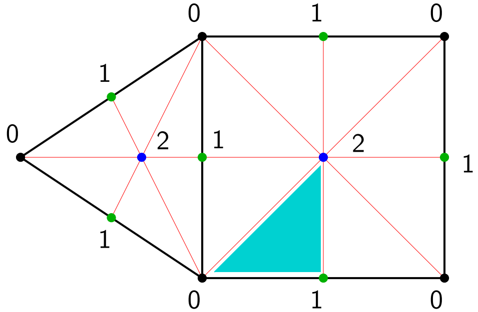
- Simplex Reflection. Remove a vertex, take the simplex on the other side.
- The reflection can be chosen based on vertex number.
- We will denote these permutations as α_i for $i \in \{0 \ldots d\}$. (aka $\texttt{switch}_i$)

Moving Around



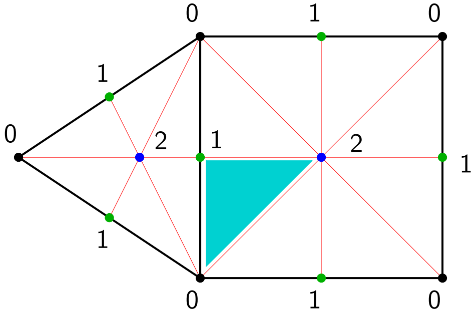
- Simplex Reflection. Remove a vertex, take the simplex on the other side.
- The reflection can be chosen based on vertex number.
- We will denote these permutations as α_i for $i \in \{0 \dots d\}$. (aka $\texttt{switch}_i$)

Moving Around



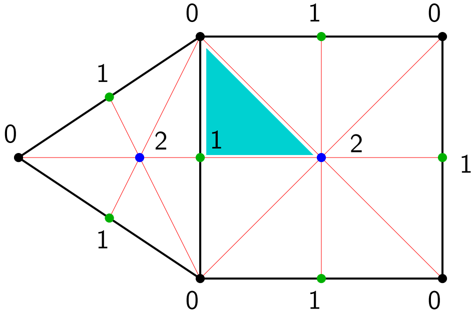
- Visit edges within a face? $\alpha_0 \alpha_1$

Moving Around



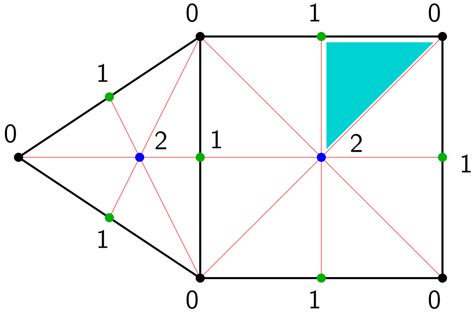
- Visit edges within a face? $\alpha_0\alpha_1$

Moving Around



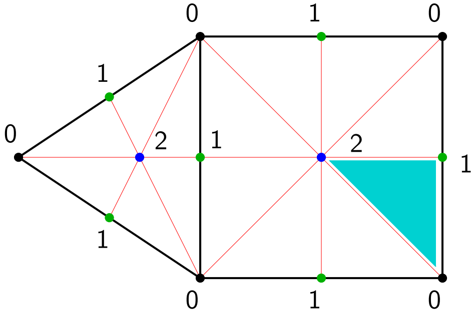
- Visit edges within a face? $\alpha_0\alpha_1$

Moving Around



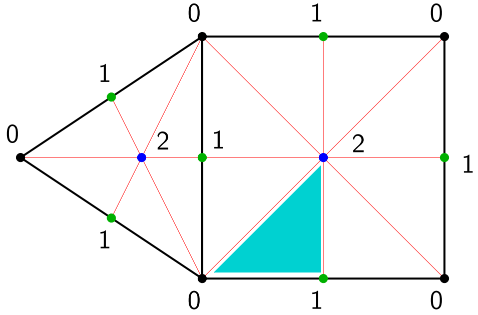
- Visit edges within a face? $\alpha_0\alpha_1$

Moving Around



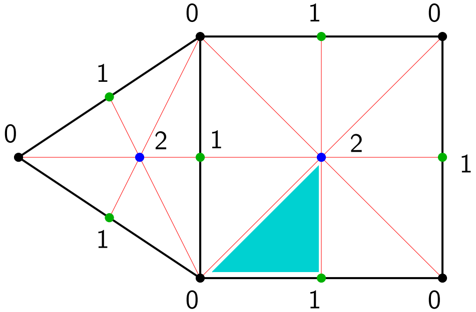
- Visit edges within a face? $\alpha_0\alpha_1$

Moving Around



- Visit edges within a face? $\alpha_0\alpha_1$

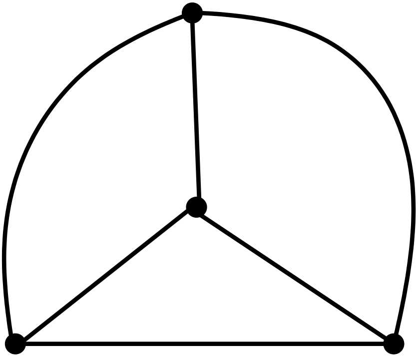
Moving Around



- Visit edges within a face? $\alpha_0\alpha_1$
- Edges around a vertex? $\alpha_2\alpha_1$
- Faces around a face? $\alpha_2\alpha_1\alpha_0\alpha_2$
- Lots more in 3+ dimensions ...

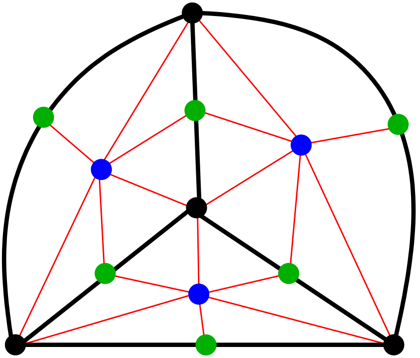
A Modern Viewpoint of Tutte

His group is just a group of reflections the Barycentric simplices

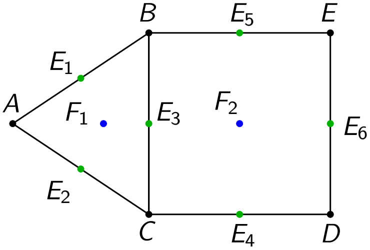


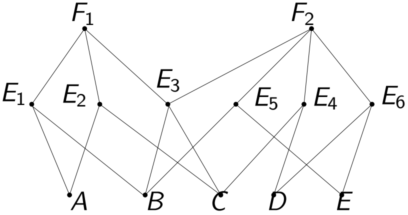
A Modern Viewpoint of Tutte

His group is just a group of reflections the Barycentric simplices



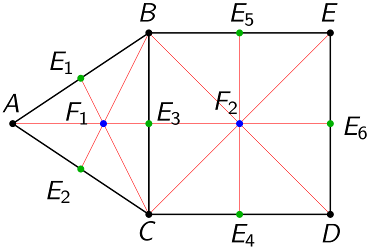
Incidence Graph and Cell-Tuples

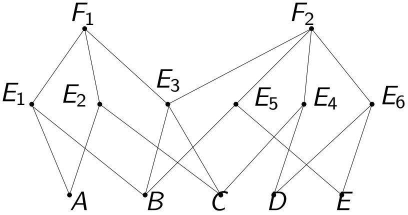




- The Incidence Graph *(Incidence Poset)*

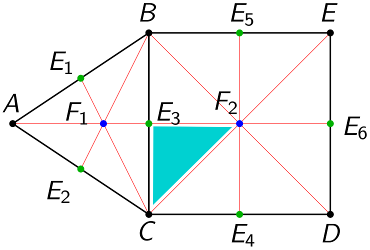
Incidence Graph and Cell-Tuples

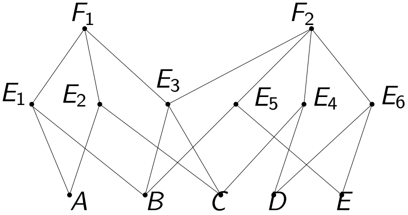




- The Incidence Graph *(Incidence Poset)*
- A Barycentric decomposition.

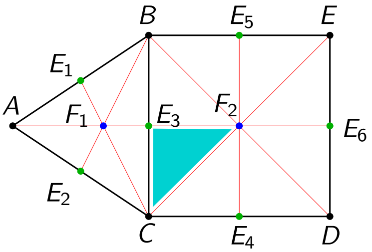
Incidence Graph and Cell-Tuples

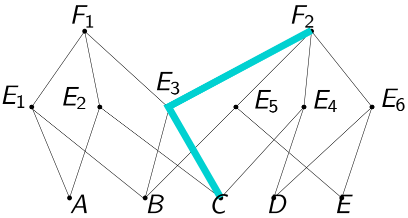




- The Incidence Graph *(Incidence Poset)*
- A Barycentric decomposition.
- A Barycentric Simplex S

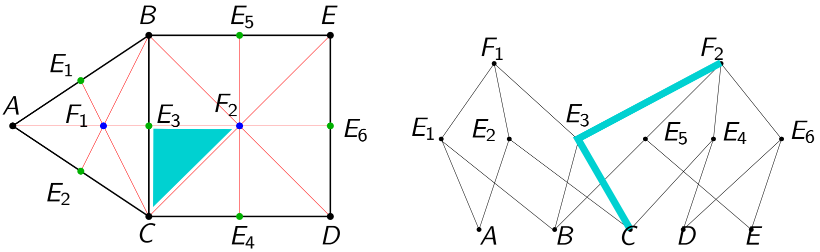
Incidence Graph and Cell-Tuples





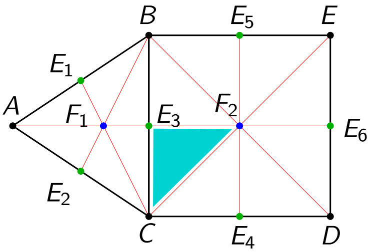
- The Incidence Graph *(Incidence Poset)*
- A Barycentric decomposition.
- A Barycentric Simplex S
- The labels on a S are a vertical **path** in the incidence graph.

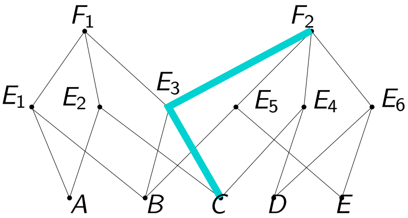
Incidence Graph and Cell-Tuples



- The Incidence Graph *(Incidence Poset)*
- A Barycentric decomposition.
- A Barycentric Simplex S
- The labels on a S are a vertical **path** in the incidence graph.
- **Cell-Tuple** given by $T = \langle C, E_3, F_2 \rangle$

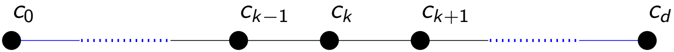
Incidence Graph and Cell-Tuples





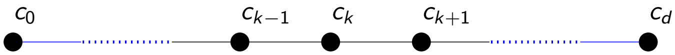
- The Incidence Graph *(Incidence Poset)*
- A Barycentric decomposition.
- A Barycentric Simplex S
- The labels on a S are a vertical **path** in the incidence graph.
- **Cell-Tuple** given by $T = \langle C, E_3, F_2 \rangle$
- **Thm:** Barycentric Simplices, Cell-Tuples, and Incidence Paths are all in a one-to-one correspondence.

Operating on Cell-Tuples with α_k



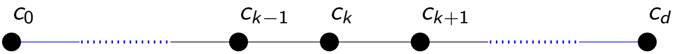
- Suppose we have a tuple $T = \langle c_1, \ldots, c_d \rangle$

Operating on Cell-Tuples with α_k



- Suppose we have a tuple $T = \langle c_1, \ldots, c_d \rangle$
- α_k only modifies the k^{th} entry in T (Simplex-Reflection).

Operating on Cell-Tuples with α_k

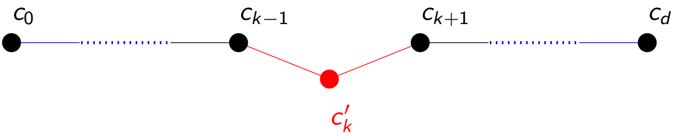


- Suppose we have a tuple $T = \langle c_1, \ldots, c_d \rangle$
- α_k only modifies the k^{th} entry in T (Simplex-Reflection).
- $\alpha_k[T] = \langle c_0, \ldots, c_{k-1}, c_k, c_{k+1}, \ldots, c_d \rangle$

Operating on Cell-Tuples with α_k

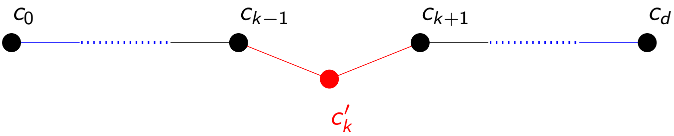
- Suppose we have a tuple $T = \langle c_1, \ldots, c_d \rangle$
- α_k only modifies the k^{th} entry in T (Simplex-Reflection).
- $\alpha_k[T] = \langle c_0, \ldots, c_{k-1}, *, c_{k+1}, \ldots, c_d \rangle$

Operating on Cell-Tuples with α_k



- Suppose we have a tuple $T = \langle c_1, \ldots, c_d \rangle$
- α_k only modifies the k^{th} entry in T (Simplex-Reflection).
- $\alpha_k[T] = \langle c_0, \ldots, c_{k-1}, c_k', c_{k+1}, \ldots, c_d \rangle$

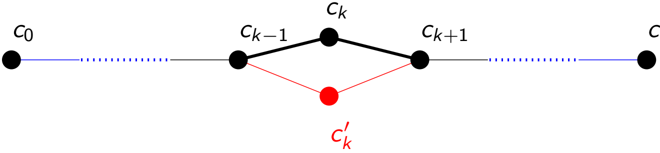
Operating on Cell-Tuples with α_k



- Suppose we have a tuple $T = \langle c_1, \ldots, c_d \rangle$
- α_k only modifies the k^{th} entry in T (Simplex-Reflection).
- $\alpha_k[T] = \langle c_0, \ldots, c_{k-1}, c'_k, c_{k+1}, \ldots, c_d \rangle$
- Furthermore, α_k is well-defined on the local portion of the Cell-Tuple:

$$\alpha_k \Big[\langle c_{k-1}, c_k, c_{k+1} \rangle \Big] = c'_k$$

Space savings using the switches α_k



One only stores all triples per switch.

For certain inputs . . .

- We haven't yet formally characterized the cellular decomposition that is supposedly our input.

For certain inputs . . .

- We haven't yet formally characterized the cellular decomposition that is supposedly our input.
- *[Bri93]* proves all of these things for "Regular Manifolds"

For certain inputs . . .

- We haven't yet formally characterized the cellular decomposition that is supposedly our input.
- *[Bri93]* proves all of these things for "Regular Manifolds"
- Determining whether an input is a Regular Manifold is hard:

For certain inputs ...

- We haven't yet formally characterized the cellular decomposition that is supposedly our input.
- *[Bri93]* proves all of these things for "Regular Manifolds"
- Determining whether an input is a Regular Manifold is hard:
  - d-Regular-Manifold reduces to $(d-1)$-Sphere-Recognition

For certain inputs . . .

- We haven't yet formally characterized the cellular decomposition that is supposedly our input.
- *[Bri93]* proves all of these things for "Regular Manifolds"
- Determining whether an input is a Regular Manifold is hard:
  - d-Regular-Manifold reduces to $(d-1)$-Sphere-Recognition
  - $d \leq 3$: Easy Polynomial Time (Euler's Formula)
  - $d = 4$: In $\mathcal{NP}$, not known to be in $\mathcal{P}$.

For certain inputs . . .

- We haven't yet formally characterized the cellular decomposition that is supposedly our input.
- *[Bri93]* proves all of these things for "Regular Manifolds"
- Determining whether an input is a Regular Manifold is hard:
  - d-Regular-Manifold reduces to $(d-1)$-Sphere-Recognition
  - $d \leq 3$: Easy Polynomial Time (Euler's Formula)
  - $d = 4$: In $\mathcal{NP}$, not known to be in $\mathcal{P}$.
  - $d = 5$: **Open**
  - $d \geq 6$: Undecidable *[Markov 58]*

For certain inputs ...

- We haven't yet formally characterized the cellular decomposition that is supposedly our input.
- *[Bri93]* proves all of these things for "Regular Manifolds"
- Determining whether an input is a Regular Manifold is hard:
  - d-Regular-Manifold reduces to $(d-1)$-Sphere-Recognition
  - $d \leq 3$: Easy Polynomial Time (Euler's Formula)
  - $d = 4$: In $\mathcal{NP}$, not known to be in $\mathcal{P}$.
  - $d = 5$: **Open**
  - $d \geq 6$: Undecidable *[Markov 58]*
- For decidable semantics either admit non-regular manifolds or reject some.

How can we make it work?

- **So Far:** Top-Down Approach

How can we make it work?

- **So Far:** Top-Down Approach
  - Begin with a whole object (cellular decomposition)

How can we make it work?

- **So Far:** Top-Down Approach
  - Begin with a whole object (cellular decomposition)
  - Subdivide into primitives

How can we make it work?

- **So Far:** Top-Down Approach
  - Begin with a whole object (cellular decomposition)
  - Subdivide into primitives
  - Characterize 'nice' properties of the subdivision

How can we make it work?

- **So Far:** Top-Down Approach
  - Begin with a whole object (cellular decomposition)
  - Subdivide into primitives
  - Characterize 'nice' properties of the subdivision
- **Problem:** No guarantees because we can't recognize proper input.

How can we make it work?

- **So Far:** Top-Down Approach
  - Begin with a whole object (cellular decomposition)
  - Subdivide into primitives
  - Characterize 'nice' properties of the subdivision
- **Problem:** No guarantees because we can't recognize proper input.
- **Solution:** Bottom-Up Approach

How can we make it work?

- **So Far:** Top-Down Approach
  - Begin with a whole object (cellular decomposition)
  - Subdivide into primitives
  - Characterize 'nice' properties of the subdivision
- **Problem:** No guarantees because we can't recognize proper input.
- **Solution:** Bottom-Up Approach
  - Begin with primitive pieces

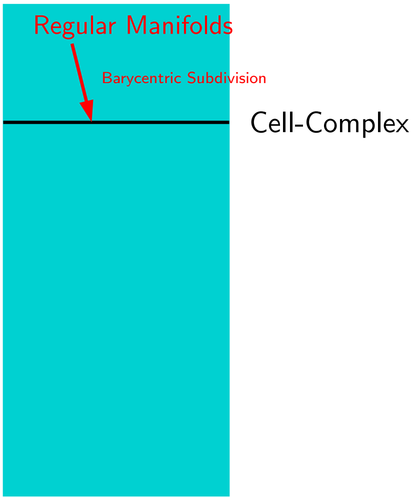
How can we make it work?

- **So Far:** Top-Down Approach
  - Begin with a whole object (cellular decomposition)
  - Subdivide into primitives
  - Characterize 'nice' properties of the subdivision
- **Problem:** No guarantees because we can't recognize proper input.
- **Solution:** Bottom-Up Approach
  - Begin with primitive pieces
  - Glue them together in ways that enforce nice properties

How can we make it work?

- **So Far:** Top-Down Approach
  - Begin with a whole object (cellular decomposition)
  - Subdivide into primitives
  - Characterize 'nice' properties of the subdivision
- **Problem:** No guarantees because we can't recognize proper input.
- **Solution:** Bottom-Up Approach
  - Begin with primitive pieces
  - Glue them together in ways that enforce nice properties
  - Characterize the class of cellular decompositions we can build.

Top-Down vs. Bottom-Up



Top-Down vs. Bottom-Up

Regular Manifolds

Barycentric Subdivision

Cell-Complex

Algebraic Construction

Recall: Permutation Group of a Set S

- Set of Permutation G of S.
- Closed: $\forall\ \alpha,\ \beta \in G,\ \alpha\beta \in G$
- Invertible: $\forall\ \alpha \in G,\ \exists\ \alpha^{-1} \in G$

Bottom Up Approach

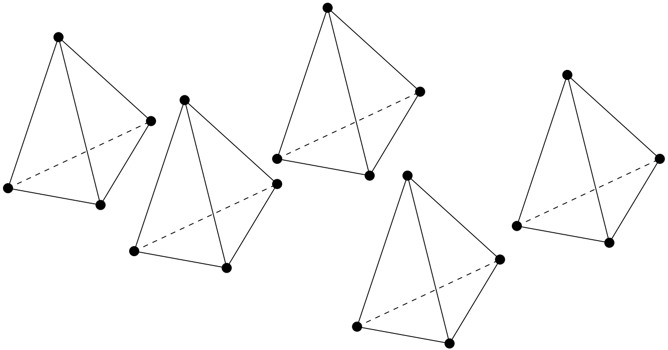
- Group G of Permuation operating on a Set S.

Bottom Up Approach

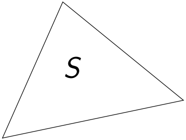
- Group G of Permuation operating on a Set S.
- G will be generated by $\langle \alpha_0, \ldots, \alpha_d \rangle$.

Bottom Up Approach

- Group G of Permuation operating on a Set S.
- G will be generated by $\langle \alpha_0, \ldots, \alpha_d \rangle$.
- S will be a set of numbered d-Simplices:



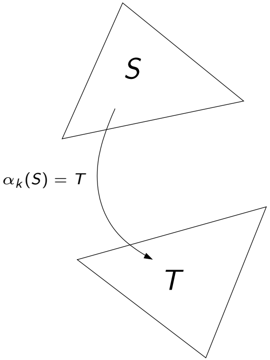
Basic Gluing Rules



- α_k must be **Fixed-Point-Free**

$$\alpha_k[s] \neq s$$

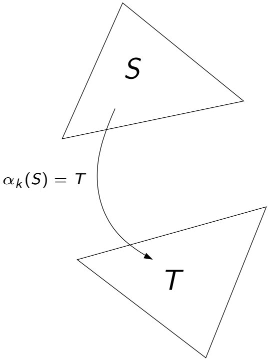
Basic Gluing Rules



- α_k must be **Fixed-Point-Free**

$$\alpha_k[s] \neq s$$

Basic Gluing Rules



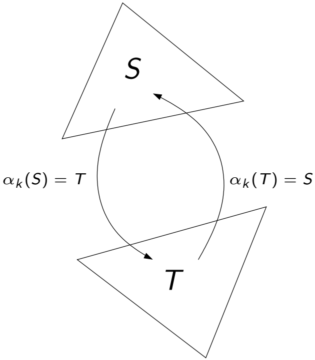
- α_k must be **Fixed-Point-Free**

$$\alpha_k[s] \neq s$$

- α_k must be an **Involution**

$$\alpha_k^2 = id$$

Basic Gluing Rules



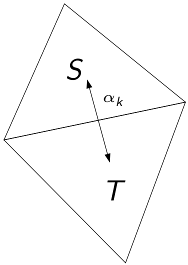
- α_k must be **Fixed-Point-Free**

$$\alpha_k[s] \neq s$$

- α_k must be an **Involution**

$$\alpha_k^2 = id$$

Basic Gluing Rules



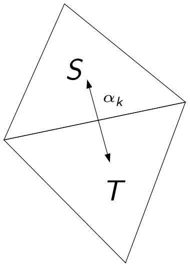
- α_k must be **Fixed-Point-Free**

$$\alpha_k[s] \neq s$$

- α_k must be an **Involution**

$$\alpha_k^2 = id$$

Basic Gluing Rules



- α_k must be **Fixed-Point-Free**

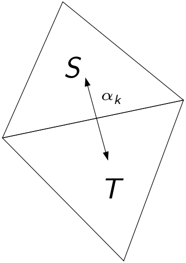
$$\alpha_k[s] \neq s$$

- α_k must be an **Involution**

$$\alpha_k^2 = id$$

- A d-simplex has $d + 1$ sides, we have $d + 1$ gluing rules $(\alpha_0 \ldots \alpha_d)$.

Basic Gluing Rules



- α_k must be **Fixed-Point-Free**

$$\alpha_k[s] \neq s$$

- α_k must be an **Involution**

$$\alpha_k^2 = id$$

- A d-simplex has $d+1$ sides, we have $d+1$ gluing rules $(\alpha_0 \ldots \alpha_d)$.
- These properties define a **Map**.

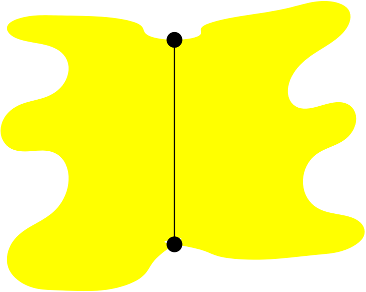
A Commuting Map

- $\forall\ i, j \in 0 \ldots d,\ j \geq i + 2$

$$\alpha_i \alpha_j = \alpha_j \alpha_i$$

- Higher dimensional switches commute with lower dimensional switches.

A Commuting Map

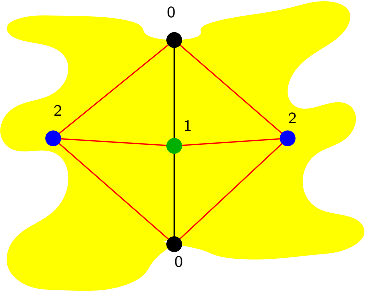


- $\forall\ i, j \in 0 \ldots d,\ j \geq i + 2$

$$\alpha_i \alpha_j = \alpha_j \alpha_i$$

- Higher dimensional switches commute with lower dimensional switches.

A Commuting Map

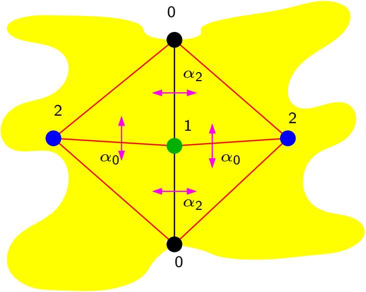


- $\forall\ i, j \in 0 \ldots d,\ j \geq i + 2$

$$\alpha_i \alpha_j = \alpha_j \alpha_i$$

- Higher dimensional switches commute with lower dimensional switches.

A Commuting Map

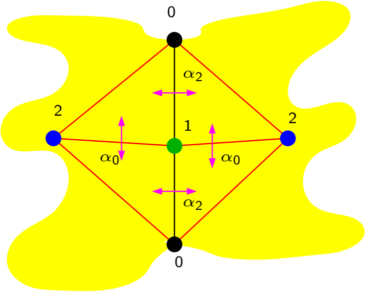


- $\forall\ i, j \in 0 \ldots d,\ j \geq i + 2$

$$\alpha_i \alpha_j = \alpha_j \alpha_i$$

- Higher dimensional switches commute with lower dimensional switches.
- e.g. $\alpha_0 \alpha_2 = \alpha_2 \alpha_0$.

A Commuting Map

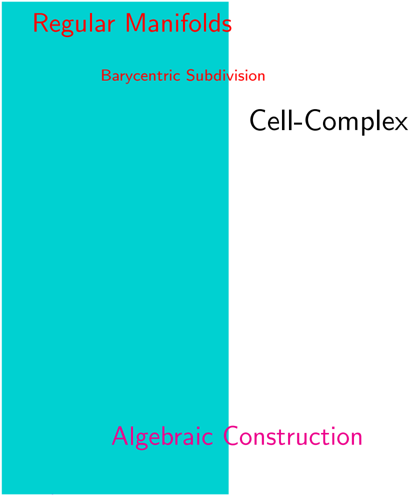


- $\forall\ i, j \in 0 \ldots d,\ j \geq i + 2$

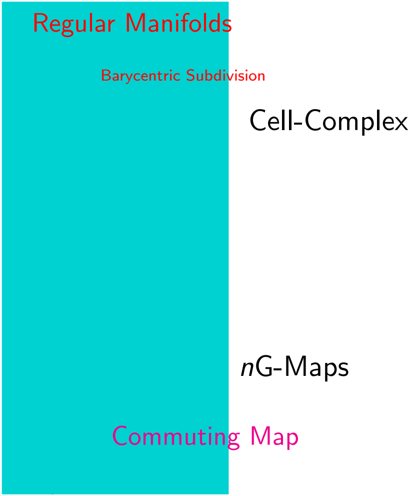
$$\alpha_i \alpha_j = \alpha_j \alpha_i$$

- Higher dimensional switches commute with lower dimensional switches.
- e.g. $\alpha_0 \alpha_2 = \alpha_2 \alpha_0$.
- Also known as an *n*G-Maps *[Lie94]*

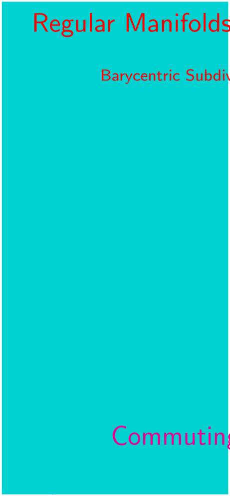
Commuting Maps (nG-Maps)



Commuting Maps (nG-Maps)



Commuting Maps (nG-Maps)



Commuting Maps (nG-Maps)

- nG-maps are often used in practice.

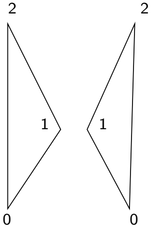
Commuting Maps (nG-Maps)

- nG-maps are often used in practice.
- Many lemmas can be proven at the level of commuting maps.

Commuting Maps (nG-Maps)

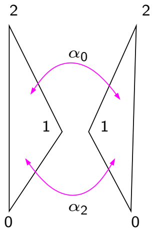
- nG-maps are often used in practice.
- Many lemmas can be proven at the level of commuting maps.
- A gap still exists: Barycentric Simplices, Incidence Paths, Cell-Tuples are not yet in correspondence

Commuting Maps (nG-Maps)



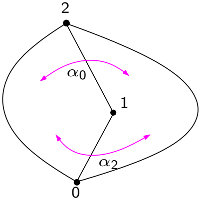
- nG-maps are often used in practice.
- Many lemmas can be proven at the level of commuting maps.
- A gap still exists: Barycentric Simplices, Incidence Paths, Cell-Tuples are not yet in correspondence

Commuting Maps (nG-Maps)



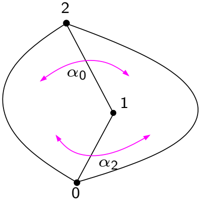
- nG-maps are often used in practice.
- Many lemmas can be proven at the level of commuting maps.
- A gap still exists: Barycentric Simplices, Incidence Paths, Cell-Tuples are not yet in correspondence

Commuting Maps (nG-Maps)



- nG-maps are often used in practice.
- Many lemmas can be proven at the level of commuting maps.
- A gap still exists: Barycentric Simplices, Incidence Paths, Cell-Tuples are not yet in correspondence

Commuting Maps (nG-Maps)



- nG-maps are often used in practice.
- Many lemmas can be proven at the level of commuting maps.
- A gap still exists: Barycentric Simplices, Incidence Paths, Cell-Tuples are not yet in correspondence
- We need more restrictions on the gluing rules.

Orthogonality Axiom *[CMP]*

- **Orthogonality Axiom**

$$\forall\ s \in S,\ k \in 1 \ldots (d-1)$$

$$\forall\ \alpha \in \langle \alpha_0, \ldots, \alpha_{k-1} \rangle,\ \beta \in \langle \alpha_{k+1}, \ldots, \alpha_d \rangle,$$

$$\alpha\beta(s) = s \longrightarrow \alpha(s) = \beta(s) = s$$

Orthogonality Axiom *[CMP]*

- **Orthogonality Axiom**

$$\forall\ s \in S,\ k \in 1 \ldots (d-1)$$

$$\forall\ \alpha \in \langle \alpha_0, \ldots, \alpha_{k-1} \rangle,\ \beta \in \langle \alpha_{k+1}, \ldots, \alpha_d \rangle,$$

$$\alpha\beta(s) = s \longrightarrow \alpha(s) = \beta(s) = s$$

- Abstractly, higher and lower dimensional operators must not only commute, they must not interfere at all.

Orthogonality Axiom *[CMP]*

- **Orthogonality Axiom**

$$\forall\ s \in S,\ k \in 1 \ldots (d-1)$$

$$\forall\ \alpha \in \langle \alpha_0, \ldots, \alpha_{k-1} \rangle,\ \beta \in \langle \alpha_{k+1}, \ldots, \alpha_d \rangle,$$

$$\alpha\beta(s) = s \longrightarrow \alpha(s) = \beta(s) = s$$

- Abstractly, higher and lower dimensional operators must not only commute, they must not interfere at all.
- Captures the idea that α_k can only affect the k^{th} entry.

Orthogonality Axiom *[CMP]*

- **Orthogonality Axiom**

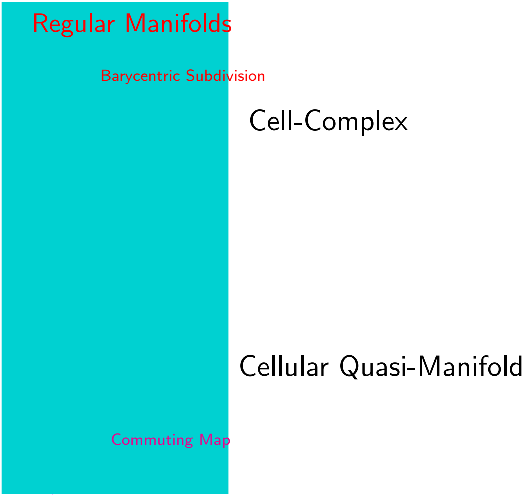
$$\forall\ s \in S,\ k \in 1 \ldots (d-1)$$

$$\forall\ \alpha \in \langle \alpha_0, \ldots, \alpha_{k-1} \rangle,\ \beta \in \langle \alpha_{k+1}, \ldots, \alpha_d \rangle,$$

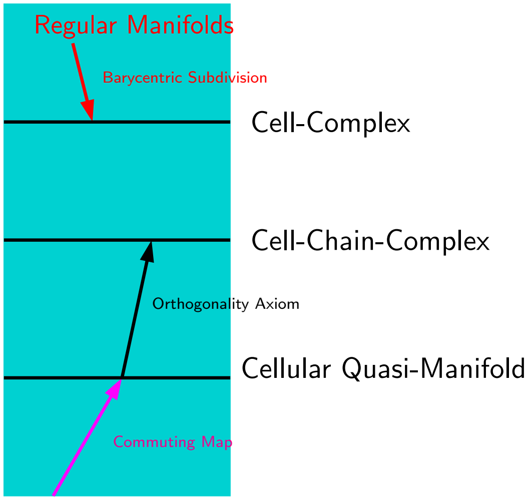
$$\alpha\beta(s) = s \longrightarrow \alpha(s) = \beta(s) = s$$

- Abstractly, higher and lower dimensional operators must not only commute, they must not interfere at all.
- Captures the idea that α_k can only affect the k^{th} entry.
- We call this a Cell-Chain-Complex.

Cell-Chain-Complex



Cell-Chain-Complex



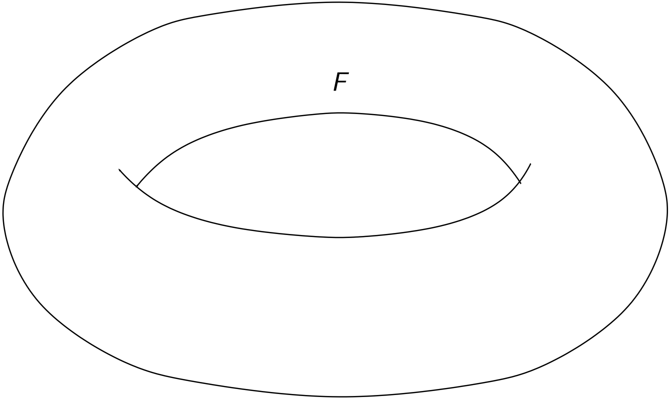
Back to Data Structures

Back to Data Structures

- Rather than operate on paths in the incidence graph, allow for an incidence *multi*-graph.

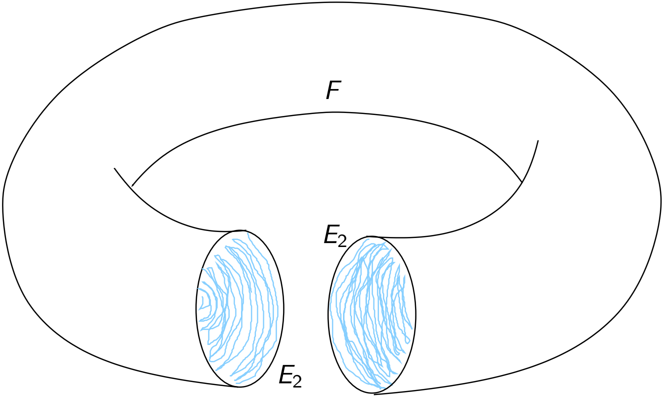
Back to Data Structures

- Rather than operate on paths in the incidence graph, allow for an incidence *multi*-graph.
- Example (Torus):



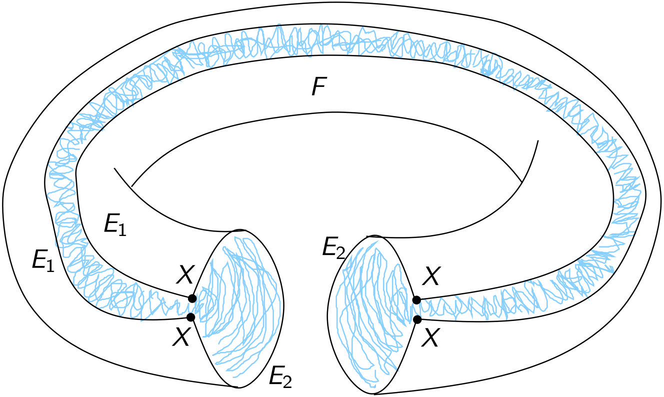
Back to Data Structures

- Rather than operate on paths in the incidence graph, allow for an incidence *multi*-graph.
- Example (Torus):



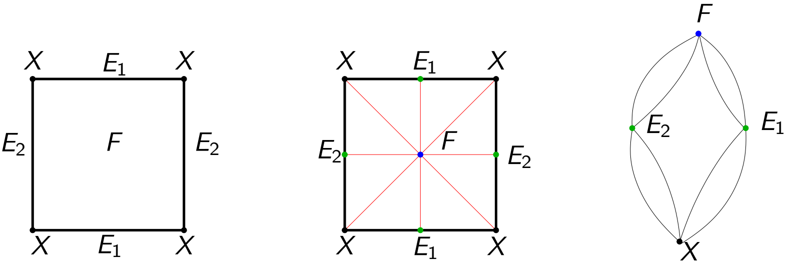
Back to Data Structures

- Rather than operate on paths in the incidence graph, allow for an incidence *multi*-graph.
- Example (Torus):

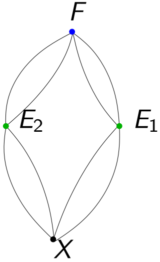


Back to Data Structures

- Rather than operate on paths in the incidence graph, allow for an incidence *multi*-graph.
- Example (Torus):

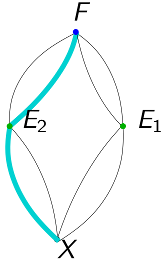


Edge Paths



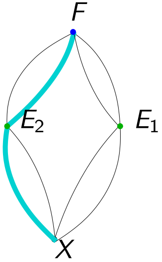
- Incidence Edge-Path

Edge Paths



- Incidence Edge-Path

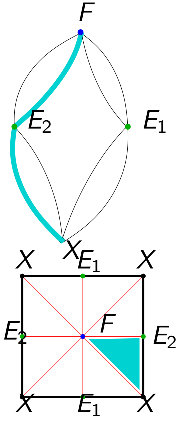
Edge Paths



- Incidence Edge-Path
- A Cell-Chain is a Cell-Tuple augmented with Edge Markers:

$$CC = \langle X, 0, E_2, 1, F \rangle$$

Edge Paths

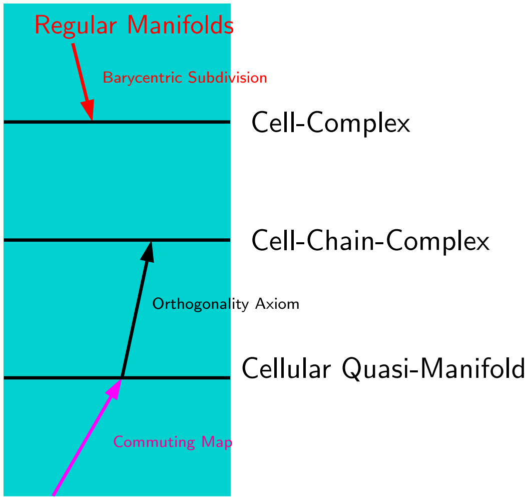


- Incidence Edge-Path
- A Cell-Chain is a Cell-Tuple augmented with Edge Markers:

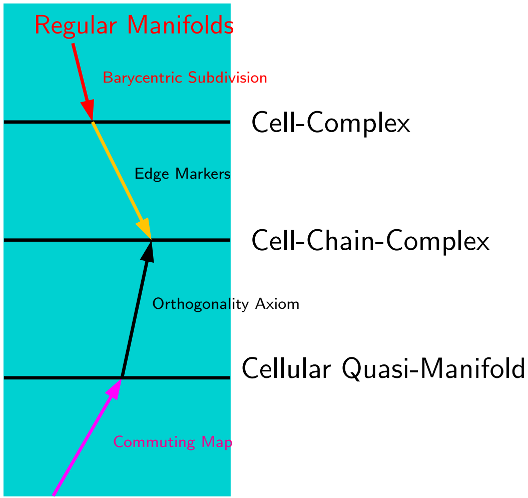
$$CC = \langle X, 0, E_2, 1, F \rangle$$

- **Thm(CMP):** In a Cell-Chain-Complex, Barycentric Simplices, Cell-Chains, and Incidence Edge-Paths are in one-to-one correspondence.

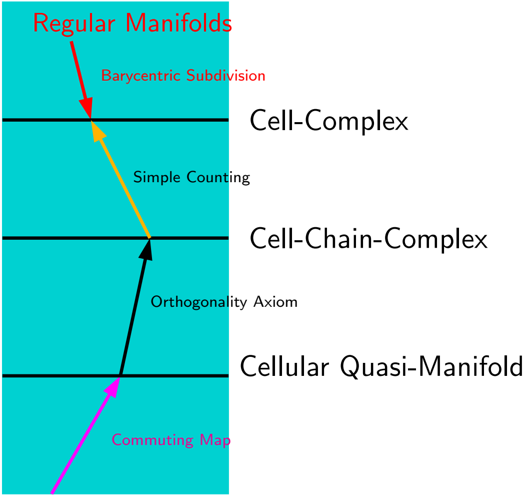
Bridging the Gap



Bridging the Gap



Bridging the Gap



Conclusions / Summary

Conclusions / Summary

- The Barycentric Subdivision is a powerful tool for designing and understanding data structures.

Conclusions / Summary

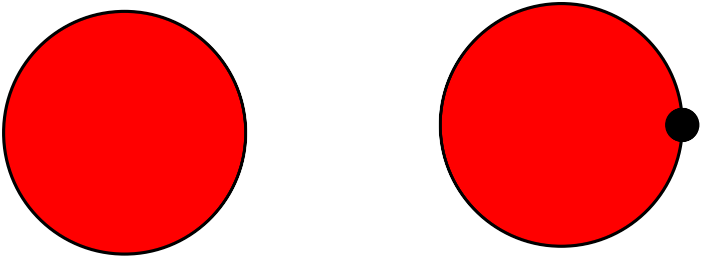
- The Barycentric Subdivision is a powerful tool for designing and understanding data structures.
- A Cell-Complex presents an elegant all-purpose data structure for cellular decompositions.

Conclusions / Summary

- The Barycentric Subdivision is a powerful tool for designing and understanding data structures.
- A Cell-Complex presents an elegant all-purpose data structure for cellular decompositions.
- A Cell-Chain-Complex successfully bridges the gap between rigorous algebraic structures and well-designed data structures.

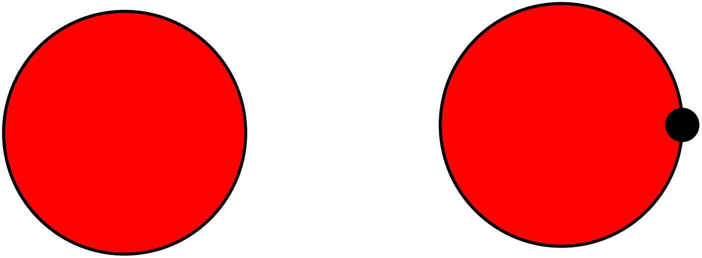
Question

- Can we have a precise semantics for more degenerate cells?



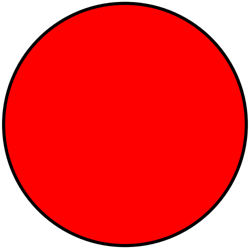
Question

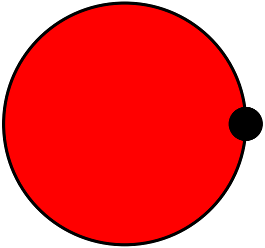
- Can we have a precise semantics for more degenerate cells?
- In particular: A simple cycle has to include at least one point.



Question

- Can we have a precise semantics for more degenerate cells?
- In particular: A simple cycle has to include at least one point.
- Here we are modeling a 2D red blood cell.





Why a cycle should be point free.

The partial simulation of a 2D red blood red cell moving through a restriction. The boundary of the cell was given a simple closed loop containing one vertex. The vertex is on the right side of the image.