PhD Seminars 2003/2004: Dr. C. D. Wensley Semester 2

Gareth Evans

February 13, 2004

Contents

1 Seminar 1: 27th January 2004

1.1 Ellis' 'Computing Group Resolutions'

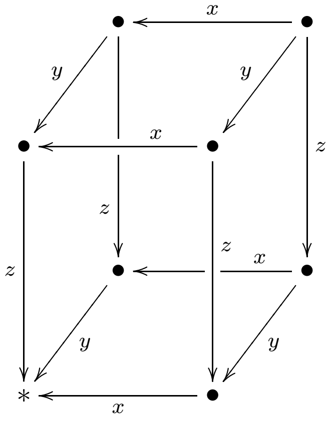
Given an input group G, we want a classifying space for G which is a CW-space X with one 0-cell, $\pi_1(X) \cong G$, $\pi_i(X) \cong 0$ for $i > 1$, and information on how to attach $(n+1)$-dimensional cells to $X(n)$ so as to form $X(n+1)$.

Example 1.1 Let $G = \mathbb{Z}^2 = \langle x, y \mid r = [x, y]\rangle$, where $r = x^{-1}y^{-1}xy$. $X(0)$ is a single vertex whilst $X(1)$ is made up of two loops x and y at this vertex. For $X(2)$, the disc shown below is attached as determined by labels on the boundary to give a torus isomorphic to $S^1 \times S^1$.

$$\begin{tikzcd}
r = \bullet \arrow[d, "y"'] & \bullet \arrow[l, "x"'] \arrow[d, "y"] \\
* & \bullet \arrow[l, "x"]
\end{tikzcd}$$

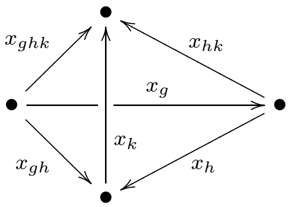
Because $\pi_i(S^1) = 0$ for $i > 1$ and $\pi_i(X \times Y) \cong \pi_i(X) \times \pi_i(Y)$, it follows that $\pi_i(X(i)) = 0$ for $i > 1$.

Example 1.2 Let $G = \mathbb{Z}^3 = \langle x, y, z \mid r = [x, y],\ s = [y, z],\ t = [z, x]\rangle$. In this example, $X(0)$ is again a single vertex, $X(1)$ has 3 1-cells, $X(2)$ has 3 squares attached and $X(3)$ is the following cube (isomorphic to $S^1 \times S^1 \times S^1$ and whose labels give $f_\sigma : S^2 \to X(2)$):



The word $w = [x, y]^{z^y}[y, x].[z, x]^{y^x}[x, z].[y, z]^{x^z}[z, y]$ is a Jacobi-Hall-Wilt identity which in this form is a product of conjugates of relators which reduces to the empty word.

Example 1.3 (Regular Presentation) Let G be arbitrary with generating set $\mathbf{x} = \{x_g \mid g \neq 1\}$ and relators $\mathbf{r} = \{r_{g,h} = x_{gh}x_h^{-1}x_g^{-1}\}$, where $x_{gg^{-1}}$ is the empty word. For each triple of elements we get a tetrahedron



For each 4-tuple of elements we get a 4-simplex with 5 tetrahedral faces and $T_{g,h,k,\ell} = {}^{g}S_{h,k,\ell} - S_{gh,k,\ell} + S_{g,hk,\ell} - S_{g,h,k\ell} + S_{g,h,k}$.

2 Seminar 2: 30th January 2004

2.1 Ellis' 'Computing Group Resolutions' Part 2

2.1.1 The Algorithm

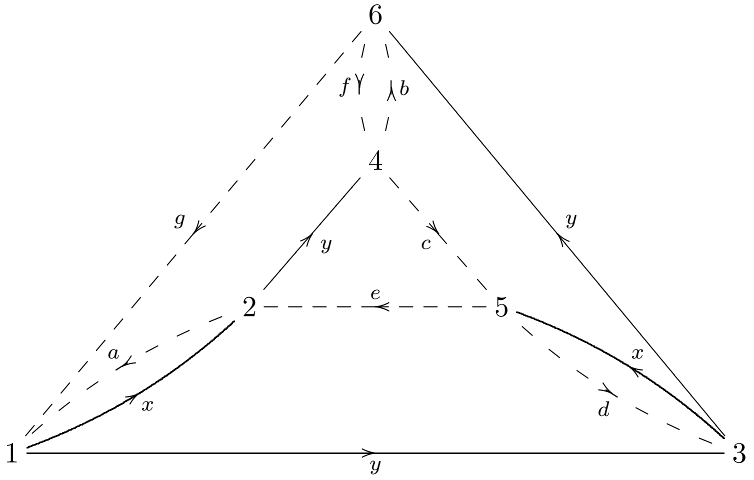
Given a group G with normal forms and a presentation, the goal of the algorithm is to compute $X(n)$ inductively (given $X(n)$, $\tilde{X}(n)$ and the action, the algorithm computes the $(n+1)$-versions).

The 0-skeleton is always a single vertex whilst the 1-skeleton is always a wedge of circles. Further, $\tilde{X}(1)$ is the topological space underlying the Cayley Graph $\Gamma(\mathbf{x})$ of G and the action of G on $\tilde{X}(1)$ is ${}^hg = hg$, ${}^h(g \xrightarrow{z} gz)$, $(hg) \xrightarrow{z} (hgz)$.

The first step of the algorithm involves constructing a maximal contractible CW subspace $Y(n)$ in $\tilde{X}(n)$ (for the Cayley Graph this is a spanning tree). Consider the following presentation for the group S_3: $S_3 = \langle x, y \mid r = x^2, s = y^{-2}x^{-1}yx \rangle$. There are two ways of obtaining the subspace $Y(1)$ of $\tilde{X}(1)$:

(i) Use Todd-Coxeter with selection strategy $1x, 1y, 2x, 2y, \ldots, nx, ny, \ldots$ to obtain the following definitions and deductions (which in turn imply the spanning tree of the Cayley Graph shown below (where the dotted edges are labelled with respect to the order of discovery in method (ii) on the next page)).

Definitions	Deductions
$1x = 2$	$2x = 1$
$1y = 3$	
$2y = 4$	
$3x = 5$	$5x = 5$, $4y = 5$
$3y = 6$	$6x = 4$, $4x = 6$, $5y = 2$, $6y = 1$



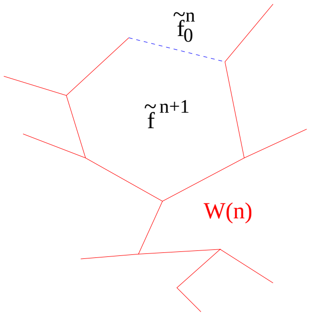
(ii) Use a permutation (or other) representation and create a list of elements. In our example, taking $x = (12)$ and $y = (123)$ we have the following list of elements:

	$1. = (\,)$
$1x$	$2. = (12) = x$
$1y$	$3. = (123) = y$
$2x = (\,) = 1.$	
$2y$	$4. = (13) = xy$
$3x$	$5. = (23) = yx$
$3y$	$6. = (132) = y^2$
$4x = (132) = 6.$	
$4y = (23) = 5.$	
$5x = (123) = 3.$	
$5y = (12) = 2.$	
$6x = (13) = 4.$	
$6y = (\,) = 1.$	

Now for each $g \in G$ the permutation $G \to G$, $w \mapsto gw$ extends to a map $\alpha_g : \tilde{X}(1) \to \tilde{X}(1)$ giving a fixed-point-free cellular action. Setting $X(1) := \tilde{X}(1)/G$, we see that we have a quotient map $p : \tilde{X}(1) \to X(1)$ which is a $|G|$-fold covering map.

2.1.2 The Elimination Procedure

- Set $U(n) := \tilde{X}(n)$ and $W(n) := Y(n)$.
- While $(W(n) \neq \tilde{X}(n))$ Do
  - Choose an n-cell $\tilde{e}^n \in \tilde{X}(n) \setminus W(n)$ ('not in the tree') with corresponding n-syzygy μ.
  - The G-action yields $g \cdot \mu$ for all $g \in G$. For each g, attach the $(n+1)$-cell $g \cdot \tilde{e}_\mu^{(n+1)}$ to $U(n)$ via $g \cdot \mu$, i.e. $U(n) := U(n) \cup \{g \cdot \tilde{e}_\mu^{(n+1)} : g \in G\}$ and $W(n) := W(n) \cup \{\tilde{e}^n\}$.
  - An n-cell $\tilde{f}_0^n \in \tilde{X}(n) \setminus W(n)$ is *contractible into* $W(n)$ iff there exists an $(n+1)$-cell $\tilde{f}^{(n+1)} \in U(n)$ such that the boundary of $\tilde{f}^{(n+1)}$ in $\tilde{X}(n)$ involves only the n-cell $\tilde{f}^n$ and the n-cells in $W(n)$ (as illustrated by the following diagram):



- While possible, repeat the following:
    - Choose $\tilde{f}_0^n \in \tilde{X}(n) \setminus W(n)$ which is contractible into $W(n)$;
    - Set $W' = W(n) \cup \{\tilde{f}_0^n\}$;
    - Record *why* $\tilde{f}_0^n$ is contractible into $W(n)$;
    - Set $W(n) := W'$.

In the example, $\tilde{e}^1$ is λ, an edge not in the tree. Choosing $\lambda = a$ so that $R_a = x^2$, we now have to attach the 2-cells $g \cdot \tilde{e}^2_\lambda$ to $\tilde{X}(1)$. The 2-cell $\tilde{e}^2_\lambda = 1 \cdot \tilde{e}^2_\lambda$ is attached so that its boundary $\partial \tilde{e}^2_\lambda$ involves only λ and edges in the tree, and is oriented by choosing (say) the direction of λ. From this we deduce that the boundary spells out a word R_λ (starting from the identity). The 2-cell $g \cdot \tilde{e}^2_\lambda$ is attached so that its boundary is the image of $1 \cdot \tilde{e}^2_\lambda$ under g. In total, 42 2-cells are added in, originating from the following list:

$$\begin{array}{ll} R_a = x^2 & R_e = yxyx^{-1} \\ R_b = xyxy^{-2} & R_f = y^2xy^{-1}x^{-1} \\ R_c = xy^2x^{-1}y^{-1} & R_g = y^3 \\ R_d = yx^2y^{-1} & \end{array}$$

Remark 2.1 If we execute the algorithm, we see that we only need the 2-cells $\{\tilde{e}^2_a, \tilde{e}^2_b\}$.

3 Seminar 3: 13th February 2004

3.1 Ellis' 'Computing Group Resolutions' Part 3

In this seminar a series of handouts were given out summarising the ideas considered so far and analysing an example involving a presentation of the group S_3. In addition the Todd-Coxeter algorithm was briefly discussed.