Teoría de Grupos: Desarrollo de una librería en Python

Alberto Jesús Durán López

Universidad de Granada

Doble grado en Ingeniería Informática y Matemáticas

25 de noviembre de 2020

Contenidos

Introducción

Introducción

Contexto: Estudio y optimización de la librería de grupos.

Objetivos:

- Construcción de grupos libres.
- Acciones de grupo.
- Algoritmo de Todd Coxeter.

Grupos libres

Motivación

Existen dos formas para describir un grupo abstracto:

- Definición axiomatizada (asociatividad, identidad e inversos).

$$S_4 = \{id, (12), (23), \dots, (13)(24), (1234)\}.$$

Motivación

Existen dos formas para describir un grupo abstracto:

- Definición axiomatizada (asociatividad, identidad e inversos).

$$S_4 = \{id, (12), (23), \ldots, (13)(24), (1234)\}.$$

- Definición usando generadores y relatores.

$$S_4 = \langle a, b \mid a^3, b^2, (ab)^4 \rangle.$$

Construcción

Sea X un conjunto arbitrario. Consideramos el conjunto $X^{\pm 1} = X^{+1} \cup X^{-1}$. Cada $x \in X$ tendrá un elemento $x^{+1} \in X^{+}$ y otro asociado $x^{-1} \in X^{-}$.

Construcción

Sea X un conjunto arbitrario. Consideramos el conjunto $X^{\pm 1} = X^{+1} \cup X^{-1}$. Cada $x \in X$ tendrá un elemento $x^{+1} \in X^{+}$ y otro asociado $x^{-1} \in X^{-}$.

Definición

Una palabra en X es una secuencia finita de elementos:

$$w = x_{a_1}^{\epsilon 1} x_{a_2}^{\epsilon 2} \cdots x_{a_n}^{\epsilon n} \quad \text{con } x_{a_i} \in X, \;\; \epsilon_i \in \{+1, -1\}, \; n \in \mathbb{N}.$$

Construcción

Sea X un conjunto arbitrario. Consideramos el conjunto $X^{\pm 1} = X^{+1} \cup X^{-1}$. Cada $x \in X$ tendrá un elemento $x^{+1} \in X^{+}$ y otro asociado $x^{-1} \in X^{-}$.

Definición

Una palabra en X es una secuencia finita de elementos:

$$w = x_{a_1}^{\epsilon_1} x_{a_2}^{\epsilon_2} \cdots x_{a_n}^{\epsilon_n} \quad \text{con } x_{a_i} \in X, \ \ \epsilon_i \in \{+1, -1\}, \ n \in \mathbb{N}.$$

Una palabra w será **reducida** si no contiene subpalabras del tipo xx^{-1} o $x^{-1}x$, para todo $x \in X$.

Ejemplo

Sea $X = \{a, b, c\}$. La palabra $ab^{-1}b^{-1}ca$ está reducida, mientras que $aa^{-1}bca$ no lo está.

Construcción

Definición

El producto de dos palabras reducidas se define como la única palabra reducida en la clase de la palabra que se obtiene por contatenación de ambas.

Construcción

Definición

El producto de dos palabras reducidas se define como la única palabra reducida en la clase de la palabra que se obtiene por contatenación de ambas.

Teorema (Existencia)

El conjunto $\mathrm{F}(X)$ de palabras reducidas en X dotadas con el producto anterior forman un grupo libre con base el conjunto X.

Teorema (Unicidad)

Si G es un grupo libre y X es una base de G, entonces G es isomorfo a $\mathrm{F}(X)$.

Construcción

Teorema

Todo grupo es isomorfo a un cociente de un grupo libre.

Teorema (Nielsen-Schreier)

Todo subgrupo de un grupo libre es libre.

Construcción

- Por el teorema anterior, todo grupo G es isomorfo a un cociente de un grupo libre:

$$G \cong \mathrm{F}(X)/N,$$

donde $\mathrm{F}(X)$ es el grupo libre generado por X, $\varphi : \mathrm{F}(X) \twoheadrightarrow G$ un epimorfismo y $N = \ker(\varphi) \trianglelefteq G$ un subgrupo normal.

Construcción

- Por el teorema anterior, todo grupo G es isomorfo a un cociente de un grupo libre:

$$G \cong \mathrm{F}(X)/N,$$

  donde $\mathrm{F}(X)$ es el grupo libre generado por X, $\varphi : \mathrm{F}(X) \twoheadrightarrow G$ un epimorfismo y $N = \ker(\varphi) \trianglelefteq G$ un subgrupo normal.
- Por el Teorema de Nielsen-Schreier, N es libre y podemos tomar $R \subseteq N$ una base de N.

Construcción

- Por el teorema anterior, todo grupo G es isomorfo a un cociente de un grupo libre:

$$G \cong \mathrm{F}(X)/N,$$

  donde $\mathrm{F}(X)$ es el grupo libre generado por X, $\varphi : \mathrm{F}(X) \twoheadrightarrow G$ un epimorfismo y $N = \ker(\varphi) \trianglelefteq G$ un subgrupo normal.
- Por el Teorema de Nielsen-Schreier, N es libre y podemos tomar $R \subseteq N$ una base de N.

El par $\langle X \mid R \rangle$ se conoce como presentación. Llamaremos generadores de G a los elementos de X y relaciones a los elementos de R, que serán dados como palabras en el alfabeto $X^{\pm 1}$.

Ejemplos

Ejemplos

- $\langle x \mid x^n = 1 \rangle \cong C_n$.
- $\langle x, y \mid x^n,\ y^2,\ (xy)^2 \rangle \cong D_n$.
- $\langle x, y \mid x^{2n},\ x^n = y^2,\ yxy^{-1} = x^{-1} \rangle \cong Q_n$.

Ejemplos

Ejemplos

- $\langle x \mid x^n = 1 \rangle \cong C_n$.
- $\langle x, y \mid x^n,\ y^2,\ (xy)^2 \rangle \cong D_n$.
- $\langle x, y \mid x^{2n},\ x^n = y^2,\ yxy^{-1} = x^{-1} \rangle \cong Q_n$.

Problema de Palabras

- $\langle a, b \mid aba^{-1}b^{-1}b^{-1}, bab^{-1}a^{-1}a^{-1} \rangle$

Ejemplos

Ejemplos

- $\langle x \mid x^n = 1 \rangle \cong C_n$.
- $\langle x, y \mid x^n,\ y^2,\ (xy)^2 \rangle \cong D_n$.
- $\langle x, y \mid x^{2n},\ x^n = y^2,\ yxy^{-1} = x^{-1} \rangle \cong Q_n$.

Problema de Palabras

- $\langle a, b \mid aba^{-1}b^{-1}b^{-1}, bab^{-1}a^{-1}a^{-1} \rangle \cong \{1\}$.

Librería en Python

Principales cambios

- Clases para definir grupos.
- Métodos que definen cada una de sus operaciones binarias.
- Actualización de métodos y ampliación con nuevos grupos.
- Posibilidad de dar grupos definidos por generadores y relatores.

Acciones de grupo

Definición

Sea X un conjunto y G un grupo. Una acción (izquierda) de G sobre X es una aplicación $G \times X \to X$; $(g, x) \mapsto {}^{g}x$ que cumple las propiedades:

1. ${}^{1}x = x, \quad \forall x \in X.$
2. ${}^{g}({}^{g'}x) = {}^{gg'}x, \quad \forall x \in X, \forall g, g' \in G.$

Algoritmo de Todd Coxeter: descripción

Sea $G = \langle X \mid R \rangle$ y $H \leq G$.
Consideramos la acción de G sobre las clases de $S := G/H$.

Algoritmo de Todd Coxeter: descripción

Sea $G = \langle X \mid R \rangle$ y $H \leq G$.
Consideramos la acción de G sobre las clases de $S := G/H$.

Sea $g \in X$ y $s \in S$.

$$\begin{tikzcd} s \arrow[r, bend left, "g"] & s^g \arrow[l, dotted, bend left, "g^{-1}"] \end{tikzcd}$$

Algoritmo de Todd Coxeter: pseudocódigo

Entrada: $G = \langle X \mid R \rangle$ y $H = \langle Y \rangle \leq G$.

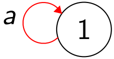
Inicializar la tabla de clases del grupo G.
for $w \in Y$ **do**
    $\mathit{ScanAndFill}(1, w)$
end
for $\alpha \in \Omega$ **do**
    for $w \in R$ **do**
        if $\mathit{isAlive}(\alpha)$ **then**
            $\mathit{ScanAndFill}(\alpha, w)$
        end
    end
end

Salida: $[G : H]$ y tabla de clases laterales/Grafo de Schreier.

Algoritmo de Todd Coxeter: ejemplo

Sea $G = \langle a, b \mid a^2, b^2, (ab)^3 \rangle$ y $H = \langle a \rangle \leq G$.

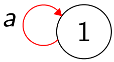
1. Comenzamos representando la clase trivial H por el 1, que debe satisfacer la relación a^2.



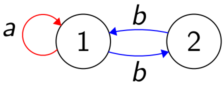
Algoritmo de Todd Coxeter: ejemplo

Sea $G = \langle a, b \mid a^2, b^2, (ab)^3 \rangle$ y $H = \langle a \rangle \leq G$.

1. Comenzamos representando la clase trivial H por el 1, que debe satisfacer la relación a^2.



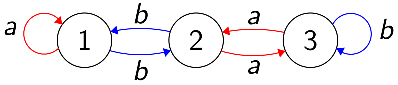
2. Cada clase se debe satisfacer cada una de las relaciones.



Algoritmo de Todd Coxeter: ejemplo

Sea $G = \langle a, b \mid a^2, b^2, (ab)^3 \rangle$ y $H = \langle a \rangle \leq G$.

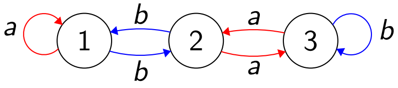
3. Completamos el resto de relaciones para que sean satisfechas por todos los vértices.



Algoritmo de Todd Coxeter: ejemplo

Sea $G = \langle a, b \mid a^2, b^2, (ab)^3 \rangle$ y $H = \langle a \rangle \leq G$.

3. Completamos el resto de relaciones para que sean satisfechas por todos los vértices.



Representación por permutaciones:

$$\begin{aligned} \varphi\colon G &\longrightarrow S(G/H) \\ a &\mapsto \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} = (23) \\ b &\mapsto \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = (12) \end{aligned}$$

Algoritmo de Todd Coxeter: Jupyter

Ejemplo en Jupyter.

Software

- Python.
- Jupyter & Rise.
- https://github.com/lmd-ugr/Grupos

Conclusiones

Conclusiones

- La libreria admite definir grupos dados por generadores y relatores.
- Nuevos grupos implementados.
- Algoritmos de Reidemeister-Schreier y Todd-Coxeter Schreier-Sims.

Bibliografía fundamental

Dummit, David S. and Foote, Richard M. (2003)
Abstract Algebra
John Wiley

Judson, Thomas W. (2017)
Abstract Algebra: Theory and Applications
Orthogonal Publishing L3c

Derek, F. , Bettina, Eick. and Eamonn O'brien (2006)
Handbook of Computational Group Theory
Math. Comput.

Miller, Kyle
The Todd-Coxeter Algorithm
UC Berkeley

Gracias por su atención.