UNIVERSIDADE ESTADUAL DE
CAMPINAS

Instituto de Matemática, Estatística e
Computação Científica

DIANA MILENA OTALORA MUÑOZ

ALGORITMOS E PROTOCOLOS DE CRIPTOGRAFIA BASEADA EM GRUPOS

Campinas

2018

DIANA MILENA OTALORA MUÑOZ

ALGORITMOS E PROTOCOLOS DE CRIPTOGRAFIA BASEADA EM GRUPOS

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestra em Matemática.

Orientador: Francesco Matucci

Este exemplar corresponde à versão final da Dissertação defendida pela aluna DIANA MILENA OTALORA MUÑOZ e orientada pelo Prof. Dr. Francesco Matucci.

Campinas

2018

Agência(s) de fomento e nº(s) de processo(s): CAPES

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Ot1a

Otálora Muñoz, Diana Milena, 1989-
Algoritmos e protocolos de criptografia baseada em grupos / Diana Milena Otálora Muñoz. – Campinas, SP : [s.n.], 2018.

Orientador: Francesco Matucci.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Grupos livres. 2. Thompson, Grupo F de. 3. Protocolos criptográficos. 4. Algoritmos. I. Matucci, Francesco, 1977-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Algorithms and protocols in group-based cryptography
Palavras-chave em inglês:
Free groups
Thompson group F
Cryptographic protocols
Algorithms
Área de concentração: Matemática
Titulação: Mestra em Matemática
Banca examinadora:
Francesco Matucci [Orientador]
Plamen Emilov Kochloukov
Slobodan Tanushevski
Data de defesa: 02-03-2018
Programa de Pós-Graduação: Matemática

Dissertação de Mestrado defendida em 02 de março de 2018 e aprovada pela banca examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). FRANCESCO MATUCCI

Prof(a). Dr(a). PLAMEN EMILOV KOCHLOUKOV

Prof(a). Dr(a). SLOBODAN TANUSHEVSKI

As respectivas assinaturas dos membros encontram-se na Ata de defesa

Aos meus pais Pedro Alfredo Otálora e María del Carmen Muñoz, as minhas irmãs Elizabeth, Rosalba e Lucía.

Agradecimentos

Agradeço a Deus por guiar meu caminho.

Aos meus pais e irmãs, dos quais recebo um imenso apoio e carinho apesar da distância.

Agradeço ao Prof. Dr. Francesco Matucci, pela orientação em temas tão interessantes da matemática, por seu tempo e sua colaboração dedicados ao desenvolvimento deste trabalho, pela paciência e motivação, por me lembrar o motivo pelo qual estudei matemática, por compartilhar seu conhecimento com uma profunda humildade e com a melhor atitude, tornando fácil o difícil. Foi um privilégio para mim ter sua orientação e sua ajuda.

Agradeço aos professores da banca Plamen Emilov Kochloukov e Slobodan Tanushevski, por me ajudar com as dúvidas e pelas valiosas observações que contribuíram no meu trabalho.

Agradeço aos professores Marcelo dos Santos e Lucio Centrone por fazer parte deste processo, deles eu recebi, além de conhecimentos matemáticos, muita motivação quando tive que enfrentar dificuldades no caminho.

Agradeço ao meu amigo e colega Ever Ticona, por acompanhar os dois anos deste processo, pela sua valiosa ajuda e as suas palavras de ânimo nos momentos difíceis, pelas muitas horas de estudo e pela sua amizade.

Agradeço aos meus caros amigos Daniela, Carlos, Heraclio, Davidson, Murilo, Miguel, Carlos Arturo, Alejandro, Javier, Juan Carlos, Matheus e Sebastián pelos momentos compartilhados e pelas alegrias vividas no transcurso destes anos, eles fizeram mais agradável minha passagem pelo IMECC, deles eu levo maravilhosas lembranças e muitas coisas aprendidas. Agradeço de maneira especial aos meus amigos Carlos Augusto Bassani e Davidson Freitas pelo tempo dedicado e pela paciência para me ajudar com o idioma no processo de digitação e correção deste trabalho.

Agradeço a Mayerlin Castro pela sua amizade sincera, obrigada por me mostrar o início do caminho que me levou até o Brasil.

Também estou profundamente agradecida ao Instituto de Matemática, Estatística e Computação Científica em geral, pela contribuição na minha formação matemática.

Por fim, agradeço a CAPES por ter financiado este projeto.

Resumo

Neste trabalho estudaremos um protocolo criptográfico no grupo F de Thompson segundo os textos de Myasnikov, Shpilrain e Ushakov [9], Shpilrain e Ushakov [12] e Matucci [8]. O objetivo deste estudo é mostrar algumas técnicas aplicadas no desenvolvimento de chaves criptográficas no grupo acima mencionado e dar uma alternativa para não permitir possíveis ataques à segurança das chaves.

Na primeira parte estudaremos os conceitos básicos da teoria combinatória de grupos, como o grupo livre e algumas das suas propriedades. Em seguida, apresentaremos algumas propriedades do grupo F de Thompson que permitam entender a estrutura algébrica na que vai ser desenvolvido o protocolo criptográfico. Mostraremos alguns algoritmos que complementam o marco teórico feito sobre dito grupo. Também apresentaremos os conceitos básicos de criptografia e mostraremos alguns dos protocolos mais importantes que têm sido desenvolvidos através da historia.

Na última parte estudaremos o protocolo criptográfico que Shpilrain e Ushakov desenvolveram no grupo F. Apresentaremos alguns aportes ao protocolo feitos por Matucci e, por fim, mostraremos que a segurança do protocolo de Shpilrain e Ushakov pode ser quebrada e mostraremos alguns resultados experimentais do protocolo, feitos por Ruinskiy, Shamir e Tsaban.

Palavras chave: Grupo livre; grupo F de Thompson; protocolos criptográficos, chaves criptográficas.

Abstract

In this work we will study a cryptographic protocol based on Thompson's group F following the texts of Myasnikov, Shpilrain and Ushakov [9], Shpilrain and Ushakov [13] and Matucci [8]. The aim of this work is to show some techniques developed to find the shared secret key for the protocol based on the aforementioned group and give some possible alternatives to improve security of the keys against possible attacks.

In the first part of this thesis we will study the basic concepts of combinatorial group theory, such as the free group and its properties. Then we will present some known properties of Thompson's group F to better understand the algebraic structure on which the cryptographic protocol will be based. We will show algorithms providing the theoretical basis of the protocol. Moreover, we will show basic concepts in cryptography and some of the most important protocols that have been developed.

In the last part, we will study the cryptographic protocol introduced by Shpilrain and Ushakov for the group F. We will present some attacks made by Matucci and, finally, we will show that the security of the Shpilrain and Ushakov protocol can be broken using some experimental attacks proposed by Ruinskiy, Shamir and Tsaban.

Keywords: Free groups; Thompson's group F; cryptographic protocols, cryptographic keys.

Lista de algoritmos

Algoritmo 1 – Forma seminormal do produto entre uma forma seminormal negativa e uma positiva. 47

Algoritmo 2 – Forma seminormal do produto de formas seminormais. . 49

Algoritmo 3 – Forma seminormal. 50

Algoritmo 4 – Eliminação de pares que não cumprem a condição *(NF2)* da forma seminormal. 53

Algoritmo 4 – 54

Algoritmo 5 – Ataque baseado no comprimento. 65

Algoritmo 6 – Ataque baseado no comprimento no grupo F. 66

Algoritmo 7 – Ataque baseado na função distancia. 68

Sumário

Introdução 12

1 PRELIMINARES 14
1.1 Grupos livres 14
1.2 Apresentação de grupos 17

2 O GRUPO F DE THOMPSON 20
2.1 Definições e exemplos 20
2.2 Representações do grupo F 22
2.3 Geradores do grupo F 26
2.4 Formas normais e seminormais no grupo F 29
2.5 Apresentações do grupo F 33

3 INTRODUÇÃO À CRIPTOGRAFIA BASEADA EM GRUPOS . . . 36
3.1 Problemas algorítmicos de teoria de grupos 36
3.2 O protocolo de Diffie-Hellman 37
3.3 O criptosistema RSA 38
3.4 O protocolo de I. Anshel, M. Anshel e Goldfeld 40
3.5 Protocolo de conjugação no grupo de tranças 42

4 CRIPTOGRAFIA NO GRUPO F DE THOMPSON 44
4.1 Protocolo do problema de decomposição 44
4.1.1 Parâmetros e geração de chaves 46
4.2 O problema da palavra no grupo de Thompson F 46
4.2.1 Alguns algoritmos para calcular formas seminormais 46
4.2.2 Calculando a forma normal 52
4.3 Descobrindo as chaves secretas do protocolo 55
4.3.1 Descobrindo a chave privada de Bob 55
4.3.2 Descobrindo a chave privada de Alice 56
4.4 Ataque das chaves secretas usando a transitividade de As e Bs . . . 56

5 ALGUNS RESULTADOS EXPERIMENTAIS NO GRUPO F 64
5.1 A criptoanálise baseada no comprimento 64
5.2 Funções distância a um subgrupo 66
5.3 Resultados experimentais 73

REFERÊNCIAS 75

Introdução

O termo criptografia surgiu da fusão das palavras gregas "Kryptós" e "gráphein", que significam "oculto" e "escrever", respectivamente. Trata-se de um conjunto de regras para codificar a informação de forma que só o emissor e o receptor consigam decifrá-la. Na computação, a técnica usada é chamada de *chaves criptográficas*. A *chave* é um conjunto de bit's baseado em um algoritmo capaz de codificar e de decodificar informações. Se o receptor da mensagem usar uma chave diferente e incompatível com a do emissor, ele não conseguirá obter a informação correta. Existem dois tipos de chaves criptográficas: *simétrica* e *assimétrica*. A criptografia simétrica permite que duas pessoas enviem mensagens baseadas em texto. As mensagens são criptografadas com uma chave compartilhada exclusiva. A maneira mais frequente de enviar essas mensagens é através de e-mail. Para trabalhar com criptografia simétrica, as duas pessoas devem conhecer a palavra ou chave secreta. Isto pode ser um número aleatório, uma palavra ou uma série de letras. Um problema dos esquemas simétricos de encriptação é que cada par de usuários que deseje se comunicar em sigilo precisa compartilhar uma chave secreta. Este tipo de criptografia é usada em compras digitais e sistemas bancários. Já na criptografia assimétrica são usadas duas chaves diferentes, uma chave pública e outra privada. A chave pública está disponível para qualquer pessoa que deseje enviar uma mensagem, enquanto a privada é aquela que o emissor conhece. O receptor pode descriptografar todas as mensagens recebidas com a chave privada. Um exemplo deste tipo de criptografia foi o sistema de comunicação *Enigma* usado na Segunda Guerra Mundial.

Em 1976 W. Diffie e M. Hellman introduziram uma maneira engenhosa de transmissão de informações, o que levou a o que agora é conhecido como criptografia de chave pública ou criptografia assimétrica, antes disso, todos os algoritmos usados, eram algoritmos de cifrado simétrico. Em 1977 Rivest, Shamir and Adleman desenvolveram o algoritmo RSA que é o criptosistema de chave pública mais comum em uso hoje, que é empregado, por exemplo, nos navegadores Firefox e Google Chrome.

Os mais recentes estudos em criptografia de chave pública têm como objetivo procurar alternativas ao criptosistema RSA. O objetivo desta dissertação é mostrar alguns desses estudos mostrados por Matucci em [8] e por Shpilrain e Ushakov em [12], os quais trabalham com um grupo de homeomorfismos lineares por partes no intervalo $[0, 1]$ chamado de grupo F de Thompson.

O trabalho será dividido em cinco capítulos: iniciamos o capítulo 1 apresentando conceitos referentes à teoria combinatória de grupos, definiremos o grupo livre e mostraremos sua importância no estudo dos grupos finitamente apresentáveis.

No capítulo 2 estudaremos o grupo F de Thompson, dando algumas caracterizações e propriedades importantes deste grupo. As referências para este capítulo serão [2] e [4].

No capítulo 3 mostraremos alguns problemas algorítmicos de teoria de grupos aplicados à criptografia, tais como o problema de Diffie-Hellman, o protocolo RSA e o protocolo teórico proposto por I. Anshel, M. Anshel e Goldfeld que depois foi aplicado ao grupo das trancas.

No capítulo 4 estudaremos o protocolo proposto por Shpilrain e Ushakov, no artigo [12]. Trata-se de um protocolo de criptografia que usa como plataforma especificamente o grupo F de Thompson, e veremos como são geradas as chaves de criptografia, pública e privada, tendo como base algumas características próprias do grupo F. Também mostraremos o pseudocódigo de alguns algoritmos para calcular as formas seminormal e normal das palavras do grupo F mostrando assim, que o problema da palavra é solúvel em F. Na segunda parte deste capítulo mostraremos o estudo feito por Matucci em [8], onde é mostrado que a seguridade de protocolo proposto por Shpilrain e Ushakov pode ser quebrada.

Finalmente, no capítulo 5 mostraremos alguns resultados experimentais do artigo [11] no qual Ruinskiy, Shamir e Tsaban apresentam o conceito de funções distância e mostram algumas melhoras ao algoritmo usado por Shpilrain e Ushakov.

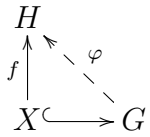
1 Preliminares

Nesse primeiro capítulo vamos apresentar os conceitos e resultados introdutórios que serão necessários ao desenvolvimento de todo o trabalho. A referência para o material apresentado neste capitulo foi o livro de Bogopolski [3]. Aqui, como em todo o texto, serão assumidos sem menção explícita os conhecimentos mais elementares sobre teoria de grupos.

1.1 Grupos livres

Nesta seção vamos denotar por G um grupo arbitrário e X um subconjunto de G.

Definição 1.1.1. Seja $X \subseteq G$, dizemos que G é um *grupo livre* com base X se: para cada grupo H e cada aplicação $f : X \to H$, existe um único homomorfismo de grupos $\varphi : G \to H$ tal que $\varphi(x) = f(x)$ para todo $x \in X$, ou seja, o seguinte diagrama é comutativo:



A comutatividade do diagrama acima é conhecida como *propriedade universal de grupos livres.*

Exemplo 1.1.2. O grupo $(\mathbb{Z}, +)$ é um grupo livre com base $X = \{1\}$.
Mais para frente estudaremos um grupo que não é livre.

Definição 1.1.3. Denotamos por $X^{-1} = \{x^{-1} \mid x \in X\}$. Os elementos de $X \cup X^{-1}$ são chamados de *letras* e o conjunto $X \cup X^{-1}$ é chamado *alfabeto*, que vai ser denotado por A, ou seja, o alfabeto é o conjunto de todas as letras. Definimos uma *palavra sobre* A como uma lista finita de letras do alfabeto A. Mais formalmente uma *palavra* w é uma função

$$w : \{1, 2, \cdots, n\} \to X \cup X^{-1}$$

definida por $w(i) = x_i^{\varepsilon_i}$, onde $x_i \in X$ e $\varepsilon_i = \pm 1$.
Se $w = x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}$ definimos a *palavra inversa* $w^{-1} = x_n^{-\varepsilon_n} \cdots x_1^{-\varepsilon_1}$. Neste caso o comprimento de w é n, e será denotado por $|w| = n$. A palavra vazia é denotada por 1 e $|1| = 0$.

Definição 1.1.4. Uma *subpalavra* da palavra $w = x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}$ ou é a palavra vazia ou é uma palavra da forma $x_r^{\varepsilon_r} \cdots x_s^{\varepsilon_s}$, $1 \leq r \leq s \leq n$.

Definição 1.1.5. Uma palavra w em X é dita *reduzida* se $w = 1$ ou w não tem subpalavras da forma xx^{-1} ou $x^{-1}x$.

Definição 1.1.6. Sejam $u = x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}$ e $v = y_1^{\delta_1} \cdots y_m^{\delta_m}$ palavras em X, a *multiplicação de palavras* está definida pela justaposição, ou seja,

$$uv = x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n} y_1^{\delta_1} \cdots y_m^{\delta_m}.$$

Se denotamos $w := uv$, a *inserção* consiste em mudar w por $uxx^{-1}v$ ou $ux^{-1}xv$. A *eliminação* significa mudar $uxx^{-1}v$ ou $ux^{-1}xv$ por w.
A inserção e a eliminação são as *operações elementares* em A. Temos também que $1w = w1 = w$.

Definição 1.1.7. Seja $X^* := \{$Conjunto de todas as palavras sobre X$\}$. Definimos a seguinte relação de equivalência sobre X^*:
$u \sim v \Leftrightarrow u$ pode ser obtido aplicando finitas operações elementares (inserção ou eliminação) à palavra v.
A classe de equivalência da palavra w é denotada $[w]$. Denotamos por $F(X) = \{[w] \mid w \in X^*\}$.

Proposição 1.1.8. *O conjunto $F(X)$ munido da operação $[u][v] := [uv]$ tem estrutura de grupo.*

Demonstração. Note que se $u \sim v$ e $u' \sim v'$ então $uu' \sim vv'$, pois se denotamos por m e k ao número de operações aplicadas em $u \sim v$ e $u' \sim v'$ respectivamente, então o número de operações aplicadas à palavra vv' até obter uu' corresponde a $m + k$. Logo a operação em $F(X)$ está bem definida.
O elemento identidade é [1] e o inverso multiplicativo de $[w]$ é $[w^{-1}]$. □

Proposição 1.1.9. *Seja $i : X \to F(X)$ definida por $i(x) = [x]$. Para cada grupo G e cada aplicação $f : X \to G$ existe um único homomorfismo $\varphi : F(X) \to G$ tal que $\varphi(i(x)) = f(x)$.*

Demonstração. Basta considerar $\varphi([x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}]) := f(x_1)^{\varepsilon_1} \cdots f(x_n)^{\varepsilon_n}$.

Dadas u, $v \in X$ temos que $\varphi([uv]) = \varphi([u])\varphi([v])$ pela definição de φ. A unicidade do homomorfismo é consecuência da propriedade universal de grupos livres. □

O fato seguinte implica que dado $X \subseteq G$, sempre é possível construir o grupo livre $F(X)$ sobre X, ou seja, $F(X)$ é o grupo livre com base X.

Corolário 1.1.10. *$F(X)$ é o grupo livre sobre X.*

Demonstração. É consequência das proposições 1.1.8 e 1.1.9. Para cada $x \in X$, $[x]$ é uma palavra reduzida, para provar que a função i é um mergulho em $F(X)$ basta considerar a seguinte proposição. □

Proposição 1.1.11. *Cada classe de equivalência no grupo livre contém uma única palavra reduzida.*

Demonstração. A prova deste resultado será estudada mais na frente, no capítulo 2. □

Teorema 1.1.12. *Se F_1 e F_2 são grupos livres sobre X, então existe um único isomorfismo φ de F_1 em F_2 tal que $\varphi(i_1(x)) = i_2(x)$ para cada $x \in X$.*

Demonstração. Sejam $i_1 : X \hookrightarrow F_1$ e $i_2 : X \hookrightarrow F_2$ os mapas inclusão em F_1 e F_2, respectivamente. Como F_1 é livre sobre X então existe único homomorfismo $\varphi : F_1 \to F_2$ tal que $\varphi(i_1(x)) = i_2(x)$. Analogamente existe um homomorfismo $\psi : F_2 \to F_1$ tal que $\psi(i_2(x)) = i_1(x)$.

Dessa forma temos que $\varphi \circ \psi(i_2(x)) = i_2(x)$ e $\psi \circ \varphi(i_1(x)) = i_1(x)$, logo $\varphi \circ \psi = Id_{G_2}$ e $\psi \circ \varphi = Id_{G_1}$.
Assim φ é isomorfismo. A unicidade é consequência da propriedade universal. □

O próximo corolário permite identificar um grupo livre gerado por X com o grupo $F(X)$.

Corolário 1.1.13. *Seja G um grupo livre sobre X. Então a aplicação identidade $Id : X \to X$ induz um isomorfismo $\varphi : G \to F(X)$.*

Demonstração. Considere o seguinte diagrama:

$$\begin{array}{ccc} X & \xrightarrow{Id} & X \\ \downarrow{\scriptstyle i_G} & & \downarrow{\scriptstyle i_{F(X)}} \\ G & \overset{\varphi}{\dashrightarrow} & F(X) \end{array}$$

onde i_G e $i_{F(X)}$ representam as aplicações inclusão de X em G e $F(X)$ respectivamente. O resultado é consequência do teorema 1.1.12 tomando $F_1 = G$ e $F_2 = F(X)$. □

Proposição 1.1.14. *Todo grupo é um quociente de um grupo livre.*

Demonstração. Seja G um grupo e tome X um conjunto de geradores de G, que sempre existe pois basta considerar $X = G$. Pela proposição 1.1.9 temos que dada a aplicação inclusão de X em G, existe um único homomorfismo $\varphi : F(X) \longrightarrow G$ tal que $\varphi([x]) = x$. Como X gera G, qualquer elemento de G pode se escrever como $g = x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}$, onde $x_i \in X$. Daí, para todo $g \in G$ existe $\widehat{x} = [x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}]$ tal que $\varphi(\widehat{x}) = g$, ou seja, φ é sobrejetiva. Pelo primeiro teorema de isomorfismo segue que $F(X)/ker\varphi \cong G$. □

1.2 Apresentação de grupos

Nesta seção vamos denotar por G e H grupos, $\langle X\rangle$ será o grupo gerado por $X \subseteq G$ e também vamos usar a notação $S \triangleleft G$ para dizer que S é um subgrupo normal de G.

Definição 1.2.1. Considere $G = \langle X\rangle$ e a aplicação natural sobrejetiva

$$\begin{aligned} \pi : F(X) &\to G \\ [x] &\to x. \end{aligned}$$

Os elementos do conjunto *ker* π são chamados de *relações da apresentação* π .

De maneira mais geral, dados $S \triangleleft F(X)$ e R um subconjunto de S, tal que S é o subgrupo normal minimal de $F(X)$ que contém R, dizemos que G está definido pelo subconjunto de geradores X e o conjunto de relações R, e escrevemos

$$G = \langle X \mid R\rangle,$$

se $G \cong F(X)/S$.

No caso que R seja finito podemos escrever

$$G = \langle X \mid r_1, \cdots, r_l\rangle,$$

onde $R = \{[r_1], \cdots, [r_l]\}$.

Definição 1.2.2. Um grupo G é dito *finitamente gerado* se $G = \langle X\rangle$ sendo X um conjunto finito e G é dito *finitamente apresentado* se X e R são finitos.

Aqui vale ressaltar que encontrar uma apresentação de um grupo não é um processo fácil. O próximo teorema é uma ferramenta para verificar uma apresentação de um grupo:

Teorema 1.2.3. *(Von Dyck)*

Sejam $G = \langle X \mid R\rangle$, $i : X \to G$ a função inclusão e $f : X \to H$.

(i) Suponha que para cada relação $r = [x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}] \in R$ temos que

$$f(x_1^{\varepsilon_1}) \cdots f(x_n^{\varepsilon_n}) = 1_H.$$

Então existe um único homomorfismo de grupos $\psi : G \to H$ tal que $\psi(i(x)) = f(x)$ para todo $x \in X$.

(ii) Por outro lado, se existe um homomorfismo de grupos $\varphi : G \to H$ então $\varphi(x_1^{\varepsilon_1}) \cdots \varphi(x_n^{\varepsilon_n}) = 1_H$ para cada relação $r = [x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}] \in R$.

Demonstração. Considere a aplicação $\varphi : G \to H$, onde $\varphi(x) = f(x)$, para todo $x \in X$, e $f(x)$ é a aplicação da proposição 1.1.9, tal que

$$\varphi(x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n}) = \varphi(x_1)^{\varepsilon_1} \cdots \varphi(x_n)^{\varepsilon_n} = f(x_1)^{\varepsilon_1} \cdots f(x_n)^{\varepsilon_n}.$$

Pela construção de φ pode se verificar que é homomorfismo.
Pela propriedade universal de grupos livres existe um único homomorfismo $\alpha : F(X) \to H$ tal que $\alpha(x) = f(x)$ para cada $x \in X$. Se denotamos por $K = ker(\alpha)$ e S o subgrupo normal minimal gerado por R então $S \subseteq K$. Sendo assim, podemos construir $\psi : F(X)/S \to F(X)/K = Im(\alpha) \subseteq H$.
Como $i(X)$ gera G então ψ é único.

A segunda parte do teorema é imediata da definição de homomorfismo de grupos, pois se $x_1^{\varepsilon_1} \cdots x_n^{\varepsilon_n} = 1_G$ então $\varphi(x_1^{\varepsilon_1}) \cdots \varphi(x_n^{\varepsilon_n}) = 1_H$. □

Exemplo 1.2.4. 1. *O grupo diedral D_{2n} é finitamente apresentado por:*

$$D_{2n} = \langle a, b \mid a^n = 1,\, b^2 = 1,\, bab^{-1} = a^{-1} \rangle.$$

Seja P um polígono regular de n lados e seja $\boldsymbol{x}$ a rotação de $\dfrac{2\pi}{n}$ em sentido anti horário ao redor do centro de P e $\boldsymbol{y}$ uma reflexão ao redor do eixo através do centro e um vértice do polígono P. Seja D_{2n} o grupo de isometrias do plano $\mathbb{R}^2$ gerado por $\boldsymbol{x}$ e $\boldsymbol{y}$.

Note que $x^n = 1$, $yxy^{-1} = x^{-1}$. Pelo teorema de Von Dyck existe um homomorfismo sobrejetivo f de $G := \langle a, b \mid a^n = 1,\, b^2 = 1,\, aba^{-1} = a^{-1} \rangle$ sobre D_{2n}, tal que $f(a) = x$ e $f(b) = y$. Resta provar que $|G| = |D_{2n}| = 2n$, mas das relações definidas acima, cada elemento de G pode ser escrito como $a^r b^s$ com $0 \leq r < n$, $0 \leq s < 2$. Os elementos $a^r b^s$ são todos elementos distintos, pois f é sobrejetivo.

2. *O grupo cíclico finito $\mathbb{Z}_6$ tem as seguintes apresentações:*

$$\mathbb{Z}_6 = \langle x | x^6 = 1 \rangle,$$
$$\mathbb{Z}_6 = \langle x, y \mid x^3 = 1,\, y^2 = 1,\, xyx^{-1}y^{-1} = 1 \rangle.$$

3. *O grupo de quatérnios pode ser apresentado como segue:*

$$Q = \langle a, b \mid a^4 = 1,\, b^2 = a^2,\, bab^{-1} = a^{-1} \rangle,$$
$$Q = \{x, y \mid xyx = y,\, x^2 = y^2\}.$$

4. *O grupo F de Thompson é um grupo que tem apresentações infinita e finita:*

$$\langle x_0, x_1, \cdots \mid x_n x_k = x_k x_{n+1},\, \forall k < n \rangle,$$
$$\langle x_0, x_1 \mid x_0^{-2} x_1 x_0^2 = x_1^{-1} x_0^{-1} x_1 x_0 x_1,\, x_0^{-3} x_1 x_0^3 = x_1^{-1} x_0^{-2} x_1 x_0^2 x_1 \rangle.$$

No capítulo 2 vamos estudar com mais profundidade este grupo.

5. *Todo os grupos abelianos finitamente gerados G são finitamente apresentados,*

$$G = \langle x_1, \cdots, x_n \mid x_i x_j = x_j x_i, \forall\ i, j \rangle.$$

6. *Um grupo livre sobre X tem apresentação $\langle X \mid R \rangle$, com $R = \emptyset$.*

2 O grupo F de Thompson

Neste capítulo vamos definir o grupo F de Thompson, apresentar algumas de suas propriedades e exibir alguns dos seus elementos. As principais referências usadas nesta parte foram o capítulo 1 da tese de Belk [2] e o artigo de Cannon, Floyd e Parry [4]. Algumas das figuras mostradas neste capítulo foram tomadas de [2].
Daqui em diante vamos usar F para representar o grupo de Thompson. O intervalo $[0, 1]$ vai ser denotado por I.

2.1 Definições e exemplos

Definição 2.1.1. O *grupo F de Thompson* é o conjunto de todos os homeomorfismos lineares por partes do intervalo fechado I nele mesmo, munidos com a operação composição de funções, tais que:

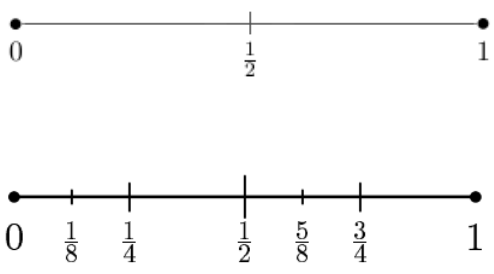
(i) as derivadas em cada intervalo são da forma 2^m com $m \in \mathbb{Z}$;

(ii) só existe uma quantidade finita de pontos onde ela não é derivável, e estes pontos correspondem a números racionais diádicos, ou seja, pontos da forma $\frac{p}{2^q}$, com $p, q \in \mathbb{N}$.

Na literatura o grupo F também é denotado por $PL_2(I)$.

Definição 2.1.2. Uma *partição diádica* do intervalo I é uma partição onde os subintervalos são da forma $\left[\frac{p}{2^q}, \frac{p+1}{2^q}\right]$ com $p, q \in \mathbb{N}$ e $p \leq 2^q - 1$. Os intervalos $\left[\frac{p}{2^q}, \frac{p+1}{2^q}\right]$ se chamam *intervalos diádicos padrão.*

Exemplo 2.1.3. *São exemplos de partições diádicas as seguintes partições de I:*



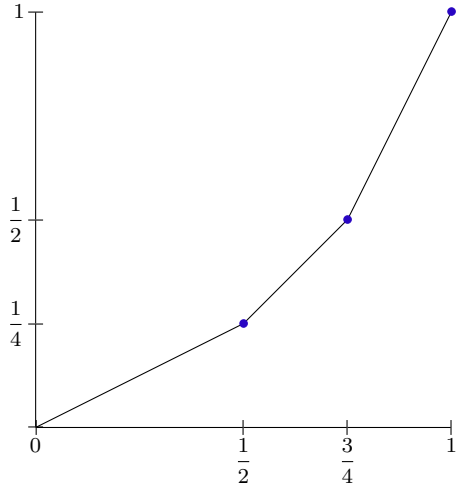
,

onde na primeira partição o primeiro e segundo intervalo são obtidos, respectivamente, tomando-se fixo $q = 1$ e $p = 0$ e $p = 1$ e, na segunda partição, os intervalos são obtidos

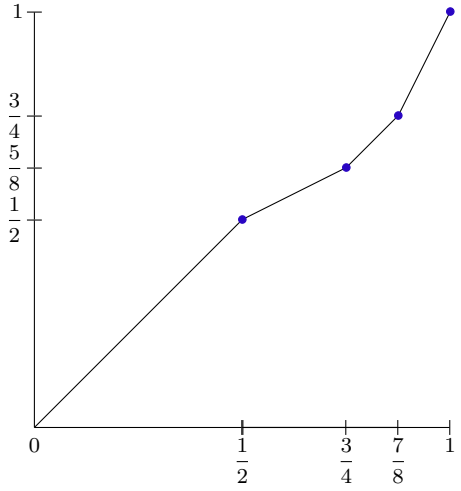
para os valores $p = 0$ *e* $q = 3$, $p = 1$ *e* $q = 3$, $p = 1$ *e* $q = 2$, $p = 0$ *e* $q = 3$, $p = 4$ *e* $q = 3$, $p = 5$ *e* $q = 3$ *e finalmente* $p = 3$ *e* $q = 2$, *respectivamente.*

Observação 2.1.4. Uma característica importante do grupo F é que dadas duas partições diádicas P e Q de I com o mesmo número de subintervalos, existe uma aplicação $f \in F$ que terá como imagem do i-ésimo subintervalo de P o i-ésimo subintervalo de Q.

Exemplo 2.1.5. Segue como exemplos, dois elementos do grupo F de Thompson.



x_0



x_1

Onde

$$x_0 = \begin{cases} \dfrac{x}{2} & \text{se} \quad 0 \leq x \leq \dfrac{1}{2} \\ x - \dfrac{1}{4} & \text{se} \quad \dfrac{1}{2} \leq x \leq \dfrac{3}{4} \\ 2x - 1 & \text{se} \quad \dfrac{3}{4} \leq x \leq 1 \end{cases}$$

e

$$x_1 = \begin{cases} x & \text{se} \quad 0 \leq x \leq \dfrac{1}{2} \\ \dfrac{x}{2} + \dfrac{1}{4} & \text{se} \quad \dfrac{1}{2} \leq x \leq \dfrac{3}{4} \\ x - \dfrac{1}{8} & \text{se} \quad \dfrac{3}{4} \leq x \leq \dfrac{7}{8} \\ 2x - 1 & \text{se} \quad \dfrac{7}{8} \leq x \leq 1 \end{cases}.$$

Veremos mais na frente que os elementos x_0 e x_1 geram o grupo F.

Daqui em diante, quando conveniente, os elementos x_0 e x_1 serão denotados por A e B, respectivamente. Dado $n \geq 2$, denotaremos também

$$x_n \;:=\; A^{-(n-1)}BA^{n-1}. \tag{2.1}$$

Observação 2.1.6. Os elementos x_n podem ser representados como homeomorfismos lineares por partes, reduzindo o gráfico da função x_0 sobre o intervalo $\left[1-\frac{1}{2^n},1\right]$ e extendendo a função identidade no intervalo $\left[0,1-\frac{1}{2^n}\right]$.

Definição 2.1.7. O grupo G é dito:

a) *grupo de torção* se $|g| < \infty$, para todo $g \in G$.

b) *livre de torção* se $|g| = \infty$, para todo $g \in G$, $g \neq 1$.

Proposição 2.1.8. *O grupo F é livre de torção.*

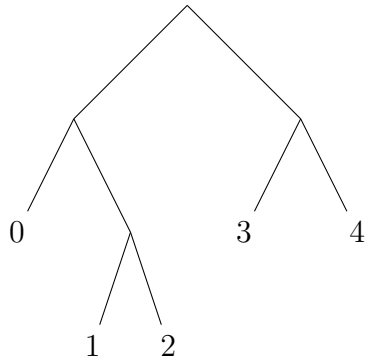
Demonstração. Sejam $f \in F$, $f \neq e$, onde e é o elemento identidade de F, e $t_0 = \inf\{t \in [0,1] : f(t) \neq t\}$. Então $f(t_0) = t_0$ e f tem derivada à direita igual a 2^m no ponto t_0, para algum $m \neq 0$.
Pela regra da cadeia, a derivada de f^n em t_0 é 2^{mn}, para todo $n \in \mathbb{N}$, assim todas as potências positivas de f são distintas. □

2.2 Representações do grupo F

Uma maneira possível de representar os elementos de F é utilizando diagramas de árvore e, como o número de intervalos de uma partição diádica é finito, tais diagramas serão árvores finitas.

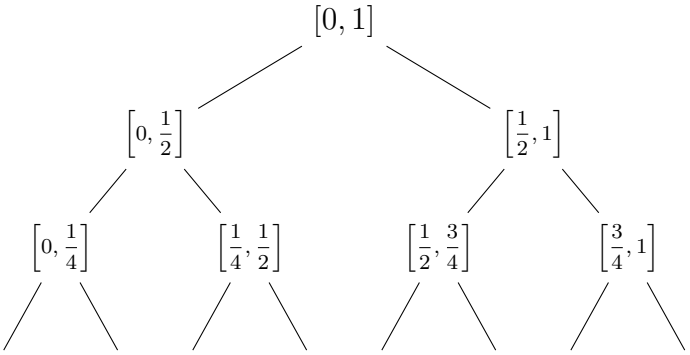
Definição 2.2.1. Uma *árvore binária* S é feita de nós, onde cada nó contém uma referência esquerda e uma referência direita. O nó mais alto da árvore é chamado de *raiz*. Cada nó (excluindo a raiz) em uma árvore é conectado por uma aresta direcionada exatamente de um outro nó. Este nó é chamado de pai. Por outro lado, cada nó pode ser conectado a dois nós, chamados de filhos. Os nós sem filhos são chamados de *folhas*.

Exemplo 2.2.2. *O diagrama abaixo representa uma árvore de cinco folhas.*



Definição 2.2.3. A *árvore $\mathcal{T}$ da partição diádica padrão* está definida como segue: os vértices são os subintervalos diádicos de $[0,1]$. Uma aresta é um par (I,J) onde J é um intervalo diádico padrão e I é a metade esquerda de J (e neste caso (I,J) é uma aresta esquerda) ou I é a metade direita de J (e neste caso (I,J) é uma aresta direita). Não é difícil ver que $\mathcal{T}$ é uma árvore ordenada binaria infinita.
Uma $\mathcal{T}$-árvore é uma subárvore binária finita de $\mathcal{T}$ com raiz $[0,1]$.

Exemplo 2.2.4. *A árvore de um intervalo diádico padrão $\mathcal{T}$ é mostrada a seguir:*

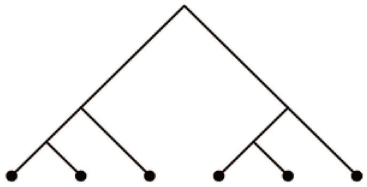


Definição 2.2.5. Um *diagrama de árvore* é um par $[R,S]$ de $\mathcal{T}$-árvores tal que R e S têm o mesmo número de folhas. R e S são as componentes binárias do diagrama. Isto é processado esquematicamente como segue:

$$R \to S$$

A árvore R é dita *árvore domínio* do diagrama e S é dita *árvore imagem* do diagrama.

A seguir temos o diagrama de árvore correspondente à partição dada pela sequência $S = \left\{0 < \frac{1}{8} < \frac{1}{4} < \frac{5}{8} < \frac{3}{4} < 1\right\}$.

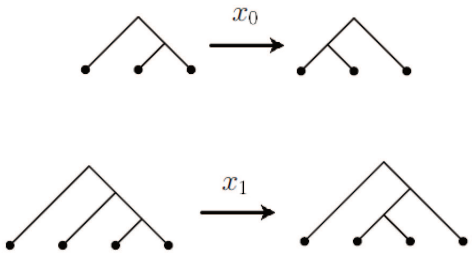


A sequência $S = \{a_1 < \cdots < a_n\} \subset [0,1]$ é chamada de *sequência de árvore* de comprimento n.

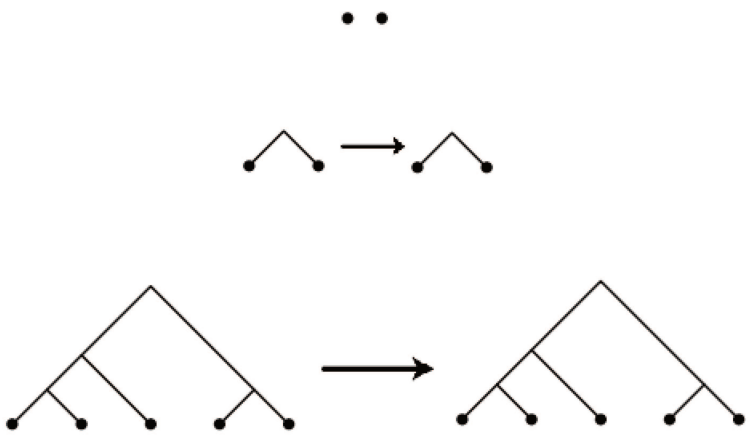
Uma sequência $S(1) = \{a_1(1) < \cdots < a_n(1)\}$ é transformada em uma segunda sequência $S(2) = \{a_1(2) < \cdots < a_n(2)\}$ pelo homeomorfismo $f : [0,1] \to [0,1]$ tal que $f(0) = 0$, $f(1) = 1$ e $f(a_i(1)) = a_i(2)$ para $i = 1, \cdots, n$.

Os elementos do grupo F podem ser representados por pares de árvores binárias finitas. A árvore à esquerda representa a partição de $[0,1]$ no domínio do homeomorfismo e a árvore à direita representa a partição feita no intervalo $[0,1]$ na imagem do homeomorfismo.

Exemplo 2.2.6. *Os elementos x_0 e x_1 do grupo F definidos em 2.1 são representados pelos seguintes diagramas de árvore:*



Observação 2.2.7. O diagrama de árvore para algum elemento do grupo F não é único. Por exemplo, os três diagramas seguintes representam o homeomorfismo identidade.

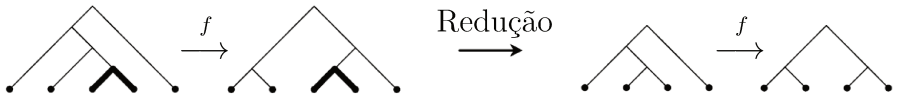


Observação 2.2.8. Sejam $f, g \in F$ com diagramas $[T, U]$ e $[U, V]$ respectivamente, então $[T, V]$ é o diagrama de gf. A notação gf corresponde ao homeomorfismo $f \circ g$, onde $\circ$ é a composição usual de funções.

Observação 2.2.9. Um circunflexo está constituído por duas arestas: uma esquerda e outra direita partindo do mesmo vértice. Se na i-ésima folha da árvore domínio e também na i-ésima folha da árvore imagem existem circunflexos, então dizemos que eses circunflexos serão opostos.

Definição 2.2.10. Um diagrama de árvore é dito *reduzido* se não tem pares de circunflexos opostos.

Em geral, a redução consiste em remover pares de circunflexos opostos, isto pode ser visto no seguinte diagrama, onde f é uma função em F:



Definição 2.2.11. Um intervalo diádico padrão J é chamado de *regular* se dado $f \in F$, f mapea o intervalo J linearmente sobre um intervalo diádico padrão.

Lema 2.2.12. *Seja $f \in F$. Então existe uma partição diádica padrão $0 = x_0 < x_1 < \cdots < x_n = 1$ tal que f é linear sobre cada intervalo da partição e $0 = f(x_0) < f(x_1) < \cdots < f(x_n) = 1$ é também uma partição diádica padrão.*

Demonstração. Seja P uma partição de $[0, 1]$ tal que os pontos de P são racionais diádicos e f é linear sobre cada intervalo P.

Seja $[a, b]$ um intervalo de P e suponha que a derivada de f sobre $[a, b]$ é 2^{-k}. Seja $m \in \mathbb{Z}$, tal que $m \geq 0$, $m + k \geq 0$, $2^m a \in \mathbb{Z}$, $2^m b \in \mathbb{Z}$, $2^{m+k} f(a) \in \mathbb{Z}$ e $2^{m+k} f(b) \in \mathbb{Z}$. Como $m \in \mathbb{Z}$ então $a < a + \dfrac{1}{2^m} < a + \dfrac{2}{2^m} < \cdots < b$ é uma partição de $[a, b]$. Como $m+k \geq 0$ então $f(a) < f(a)+\dfrac{1}{2^{m+k}} < f(a)+\dfrac{2}{2^{m+k}} <...< f(b)$ é uma partição do intervalo $[f(a), f(b)]$. □

Teorema 2.2.13. *Cada elemento de F tem um único diagrama reduzido.*

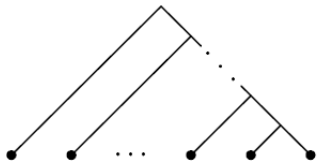
Demonstração. Uma árvore do domínio de um elemento $f \in F$ junto com o homeomorfismo f determinam a árvore da imagem. Suponha que T e T' são possíveis árvores do domínio de f tal que $T \subset T'$, então o diagrama com domínio T é uma redução do diagrama com domínio T'.
Basta provar que o conjunto de todas as possíveis árvores do domínio de f tem um elemento mínimo.

Note que a árvore T é uma possível árvore domínio se, e somente se, todas as folhas são regulares. Daí, o conjunto das árvores domínio de f é fechado sobre interseções, logo existe elemento mínimo. □

2.3 Geradores do grupo F

Definição 2.3.1. Uma árvore que em cada nó só tem uma folha à esquerda se chama de *right vine*.

Para cada inteiro não negativo n, denotamos $\mathcal{T}_n$ ao right vine com $n+1$ folhas.



Definição 2.3.2. As funções em F da forma $x_0^{b_0}x_1^{b_1}\cdots x_n^{b_n}$, com $b_k \geq 0$, são chamadas de *positivas*.

Observação 2.3.3. A árvore do domínio no diagrama reduzido de um elemento positivo é right vine. Um elemento positivo de $f \in F$ com diagrama $[\mathcal{T}_n, R]$ vai ser denotado por $[R]$.

Observação 2.3.4. Uma subárvore de uma árvore right vine, é também right vine.

Proposição 2.3.5. *Cada elemento de F pode ser escrito da forma $p^{-1}q$, onde p e q são positivos.*

Demonstração. Seja $f \in F$ com diagrama $[R, S]$. Note que $[R, S] = [R, \mathcal{T}_n]\ [\mathcal{T}_n, S]$, onde $\mathcal{T}_n$ é uma árvore como na definição 2.3.1.
Pela observação 2.2.8 temos o seguinte:

$$\begin{aligned}[R, S] &= [R, \mathcal{T}_n][\mathcal{T}_n, S] \\ &= [\mathcal{T}_n, R]^{-1}[\mathcal{T}_n, S] \\ &= [R]^{-1}[S]\end{aligned}$$

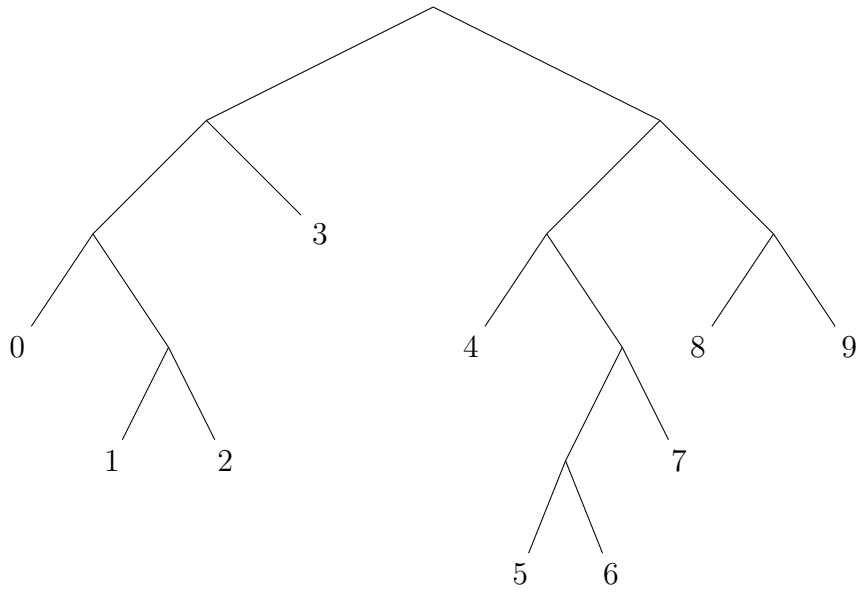
logo $f = [R]^{-1}[S]$.

□

Definição 2.3.6. A *largura* de uma árvore binária é a quantidade de folhas menos um. As folhas de uma árvore binária de largura w são numeradas com $0, 1, \ldots w$, da esquerda para a direita.

Definição 2.3.7. Sejam $I_0, \cdots, I_n$ folhas de uma $\mathcal{T}$-árvore S. Para cada inteiro k, $0 \leq k \leq n$, definimos o k-ésimo *expoente* de S e escrevemos a_k ao comprimento do arco máximo das arestas esquerdas em S que iniciam em I_k e que não chegam até o lado direito de S.

Exemplo 2.3.8. *Considere a árvore*



As relações entre as folhas e os expoentes seguem na tabela abaixo:

Folha	0	1	2	3	4	5	6	7	8	9
Expoente	2	1	0	0	1	2	0	0	0	0

Teorema 2.3.9. *Sejam R e S árvores com $n+1$ folhas, $n \geq 0$. Sejam a_0, ..., a_n os expoentes de R e sejam $b_0, \cdots, b_n$ os expoentes de S. Então a função f correspondente ao diagrama $[R, S]$ é*

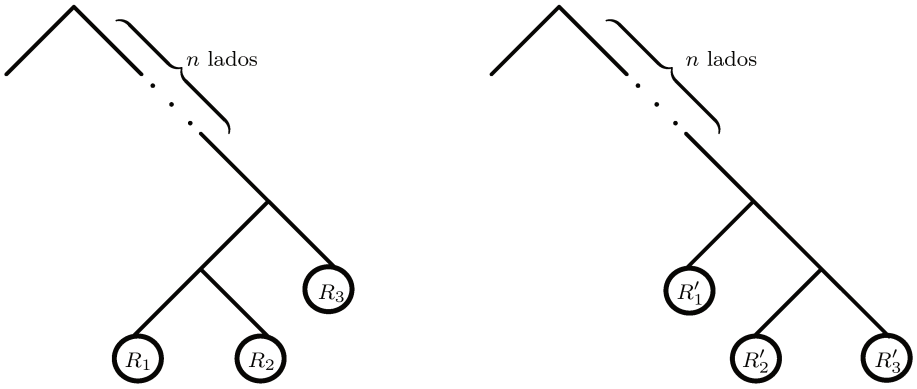
$$x_0^{b_0}\ x_1^{b_1} \cdots x_n^{b_n}\ x_n^{-a_n} \cdots x_1^{-a_1}\ x_0^{-a_0}.$$

Demonstração. Pela proposição 2.3.5 temos que um elemento $f \in F$ do diagrama $[R, S]$ pode ser reescrito como $[R, \mathcal{T}_n][\mathcal{T}_n, S] = [R, \mathcal{T}_n][S, \mathcal{T}_n]^{-1}$.
Denotemos $g = [R, \mathcal{T}_n]$ e $h = [S, \mathcal{T}_n]^{-1}$.

Vamos provar que a função de $[R, \mathcal{T}_n]$ é $x_n^{-a_n} \cdots x_1^{-a_1}\ x_0^{-a_0}$. Faremos isso por indução sobre $a := \sum_{i=1}^{n} a_i$.

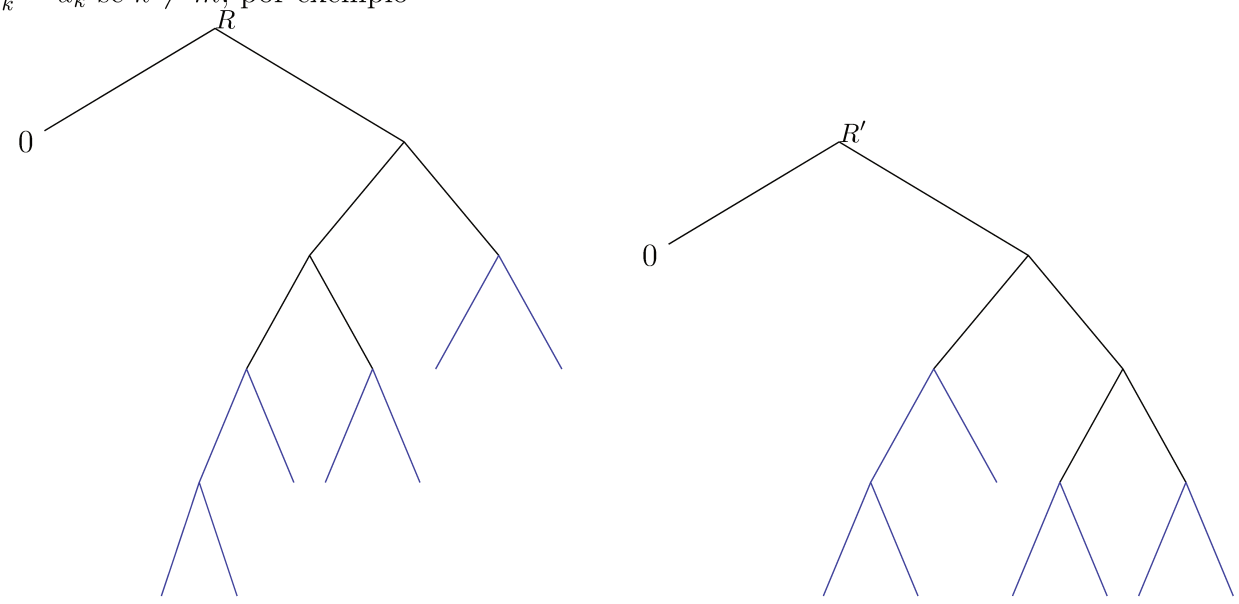
Se $a = 0$, ou seja, $a_i = 0$ para todo i, então $R = \mathcal{T}_n$, pois se $a_i = 0$, R é positivo e daí $[R, \mathcal{T}_n]$ é o mapa identidade, ou seja, o elemento $x_n^0 \cdots x_1^0\ x_0^0$. Suponha que a afirmação é verdadeira para todo $k < a$, onde $a > 0$. Seja m o menor índice tal que $a_m > 0$ então

existem subárvores binárias R_1, R_2, R_3 de R que têm a seguinte forma (a imagem mostra a forma das árvores R à esquerda e R' à direita):



Seja $R^{'}$ a árvore tal que $R_1^{'}$, $R_2^{'}$, $R_3^{'}$ sejam isomorfos com R_1, R_2, R_3. Pelos diagramas do exemplo 2.2.6, temos que $[R, R^{'}]$ corresponde a x_m^{-1}.

Se $a_0^{'}$, ..., $a_n^{'}$ são os expoentes de $R^{'}$, pela construção de R e $R^{'}$ temos que $a_m^{'} = a_m - 1$ e $a_k^{'} = a_k$ se $k \neq m$, por exemplo



Pela hipótese de indução aplicada a $R^{'}$ temos que $[R^{'}, \mathcal{T}_n] = x_n^{-a_n'} \cdots x_2^{-a_1'} x_1^{-a_1'} x_0^{-a_0'}$, então temos que o diagrama $[R, \mathcal{T}_n] = R \to R' \to \mathcal{T}_n$ corresponde à função $x_n^{-a_n} \cdots x_2^{-a_2}\ x_1^{-a_1}\ x_0^{-a_0}$, pois

$$\begin{aligned}
[R, \mathcal{T}_n] &= [R, R'][R', \mathcal{T}_n] \\
&= x_n^{-a_n'} \cdots x_2^{-a_1'} x_1^{-a_1'} x_0^{-a_0'} x_m^{-1} \\
&= x_n^{-a_n} x_{n-1}^{-a_{n-1}} \cdots x_{m+1}^{-a_{m+1}} x_m^{-(a_m-1)} x_{m-1}^{-a_{m-1}} \cdots x_2^{-a_2} x_1^{-a_1} x_0^{-a_0} x_m^{-1} \\
&= x_n^{-a_n} x_{n-1}^{-a_{n-1}} \cdots x_{m+1}^{-a_{m+1}} x_m^{-a_m} x_m^{1} x_{m-1}^{-a_{m-1}} \cdots x_2^{-a_2} x_1^{-a_1} x_0^{-a_0} x_m^{-1} \\
&= x_n^{-a_n} \cdots x_2^{-a_2} x_1^{-a_1} x_0^{-a_0}.
\end{aligned}$$

Assim temos que $g = x_n^{-a_n} \cdots x_1^{-a_1}\ x_0^{-a_0}$ e também temos que $h = (x_n^{-b_n} \cdots x_1^{-b_1}\ x_0^{-b_0})^{-1}$,

então pela observação 2.2.8 temos que $[R, S]$ é o diagrama da função hg, tendo assim provado o resultado.

□

Corolário 2.3.10. *O grupo F de Thompson é gerado pelos elementos x_0 e x_1.*

Demonstração. Pelo teorema 2.3.9 sabemos que para todo $f \in F$, o diagrama $[R, S]$ associado a esse elemento é $x_0^{b_0}\ x_1^{b_1} \cdots x_n^{b_n}\ x_n^{-a_n} \cdots x_1^{-a_1}\ x_0^{-a_0}$, mas $x_0 = A$ e $x_n = A^{-(n-1)}BA^{n-1}$ quando $n \geq 1$, então o diagrama $[R, S]$ representa o seguinte elemento

$$A^{b_0}B^{b_1}(A^{-2}BA^2)^{b_2} \cdots A^{-(n-1)a_n}BA^{(n-1)a_n} \cdots B^{-a_1}A^{-a_0}.$$

Daí os elementos A e B geram o grupo F. □

2.4 Formas normais e seminormais no grupo F

Definição 2.4.1. A *forma normal* para um elemento do grupo F de Thompson é uma palavra da forma

$$x_0^{b_0} \cdots x_s^{b_s} \cdots x_n^{b_n} x_n^{-a_n} \cdots x_t^{-a_t} \cdots x_0^{-a_0} \tag{2.2}$$

tal que as seguintes condições são satisfeitas:

(*NF1*) Exatamente um dos expoentes a_n ou b_n é distinto de zero.

(*NF2*) Se ambos x_i e x_i^{-1} ocorrem (ou seja $a_i \neq 0$ e $b_i \neq 0$), então x_{i+1} ou x_{i+1}^{-1} ocorre (ou seja $a_{i+1} \neq 0$ ou $b_{i+1} \neq 0$).

Uma palavra w é dita estar na forma *seminormal* se está na forma 2.2 e satisfaz (*NF1*).

Nesta seção explicamos a maneira para demonstrar a existência e unicidade da forma normal. Precisamos definir reduções no grupo F de Thompson e explicar porque estas reduções formam um sistema confluente.

Observação 2.4.2. Qualquer elemento de F pode ser escrito na forma seminormal usando as seguintes regras ou *reduções*, que vão ser denotadas por $\mathcal{R}$:

$$
\begin{array}{lccc}
(\mathcal{R}1) & x_n^{-1}x_k & \to & x_k x_{n+1}^{-1} \\
(\mathcal{R}2) & x_k^{-1}x_n & \to & x_{n+1}x_k^{-1} \\
(\mathcal{R}3) & x_n x_k & \to & x_k x_{n+1} \\
(\mathcal{R}4) & x_k^{-1}x_n^{-1} & \to & x_{n+1}^{-1}x_k^{-1} \\
(\mathcal{R}5) & x_i^{-1}x_i & \to & 1
\end{array}
$$

para $k < n$.

É importante dizer que independentemente da ordem em que sejam aplicadas as reduções $\mathcal{R}$, sempre vamos obter o mesmo resultado. O anterior fato é uma consequência do Lema do Diamante de Newman, mas primeiro precisamos de algumas definições:

Definição 2.4.3. Um *grafo dirigido* $\mathcal{G}$ é uma coleção de um conjunto não vazio $\mathcal{G}^0$ de vértices e um conjunto $\mathcal{G}^1$ de arestas dirigidas. Cada aresta dirigida está associada a dois vértices: o primeiro é a ponta inicial da aresta e o segundo é a ponta final.
Um *caminho dirigido* é uma sequência de arestas dirigidas tal que dado um conjunto de vértices $a_0 \cdots, a_m$, para cada i $a_i \to a_{i+1}$ é aresta dirigida.
Um subconjunto $C \subseteq \mathcal{G}^0$ é uma *componente conexa* de $\mathcal{G}$ se para cada $a, b \in C$, existe *caminho não dirigido* entre a e b, ou seja uma sequência de vértices $a = a_1, a_2, \cdots, a_m = b$, tal que para cada i ou $a_i \to a_{i+1}$ ou $a_{i+1} \to a_i$.

Lema 2.4.4. *(Lema do Diamante de Newman) Seja $\mathcal{G}$ um grafo dirigido que cumpre as seguintes condições:*

(1) Condição de finalização: para cada $u \in \mathcal{G}$, existe um inteiro $r = r(u)$ tal que, para qualquer caminho dirigido $u = u_1 \to u_2 \to u_3 \to \cdots$, que inicia em u e tal que $u_i \neq u_{i+1}$ para todo i, tem no máximo r arestas.

(2) Confluência local: Se existem arestas dirigidas $u \to x$ e $u \to y$ então existem dois caminhos dirigidos $x = a_1 \to a_2 \to \cdots \to a_k = v$ e $y = b_1 \to b_2 \to \cdots \to b_s = v$, para algum vértice $v \in \mathcal{G}$.

Então $\mathcal{G}$ é confluente, ou seja, cada componente conexa tem um único vértice reduzido.

Demonstração. Seja C uma componente conexa do grafo $\mathcal{G}$.

Afirmação 1: Para cada $u \in C$ existe um único vértice $O(u) \in C$ tal que para cada caminho maximal (um caminho de comprimento máximo tal que cada dois vértices consecutivos sejam distintos) que inicia em u, $u = u_1 \to \cdots \to u_k$, com $u_i \neq u_{i+1}$ para todo i , então $u_k = O(u)$. Pela construção, $O(u)$ é um vértice reduzido (ou seja, não existe uma aresta que inicie em $O(u)$ e termine em outro vértice $v \neq O(u)$).

Demonstração. Se o vértice u é reduzido, então $r(u) = 0$ e definimos $O(u) = u$. Por indução sobre $r(u)$, pela condição de finalização e assumindo que $O(v)$ existe para quaisquer vértice v tal que $r(v) < r(u)$.
Sejam $u = a_1 \to \cdots \to a_r$ e $u = b_1 \to \cdots \to b_s$ dois caminhos maximais, precisamos provar que $a_r = b_s$. Pela condição (2) aplicada as arestas $u \to a_2$ e $u \to b_2$ é possível achar dois caminhos $p_1 = a_2 \to a'_3 \to \cdots \to a'_m \to c$ e $p_2 = b_2 \to b'_3 \to \cdots \to b'_n \to c$. Se c não é reduzido, podemos encontrar um caminho maximal p_3 iniciando em c. Pela condição de terminação temos que p_3 é finito e o último vértice deste caminho é reduzido.
Agora observe que $r(a_2) < r(u)$, assim, por indução temos que $a_r = O(a_2) = c$. Analogamente temos que $b_s = O(b_2) = c$, então neste caso definimos $O(u) := a_r$. □

Afirmação 2: Se $a \to b$, $a, b \in C$, então $O(a) = O(b)$.

Demonstração. Seja $b = u_1 \to \cdots \to u_k = O(b)$ um caminho maximal com $u_i \neq u_{i+1}$, então $a = u_0 \to \cdots \to u_k$ também é um caminho maximal com $u_i \neq u_{i+1}$, e pela afirmação 1 temos que $O(a) = O(b)$. □

Afirmação 3: Se $a_1, a_m \in C$ e $a_1, a_2, \cdots, a_m$ um caminho não dirigido entre a_1 e a_m, então $O(a_1) = O(a_m)$.

Demonstração. Pela afirmação 2 temos que $O(a_i) = O(a_{i+1})$ e isto implica que $O(a_1) = O(a_m)$. □

Pela combinação das afirmações 1, 2 e 3 temos provado o lema. □

O lema anterior significa que iniciando num vértice de um grafo e pegando diferentes caminhos dirigidos, sempre é possível chegar a um mesmo vértice depois de um número finito de passos.

Agora podemos provar a proposição 1.1.11 do capítulo 1, que diz que cada classe de equivalência tem uma única palavra reduzida:

Demonstração da proposição 1.1.11. Seja w uma palavra. Vamos construir um grafo cujos vértices sejam as palavras sobre X e a aresta $w_1 \longrightarrow w_2$ existe se w_2 é obtida fazendo uma redução a w_1, ou seja, eliminando xx^{-1} ou $x^{-1}x$ da palavra w_1. Note que esse grafo satisfaz a condição de finalização do Lema 2.4.4 pois cada aresta reduz o comprimento da

palavra. Vamos provar agora a segunda condição do Lema 2.4.4: Sejam $u, a, b \in [w]$ tal que $u \longrightarrow a$ e $u \longrightarrow b$, e consideremos os seguintes casos:

(1) Se $u = v_1x^{-1}xpx^{-1}xv_2$, $a = v_1px^{-1}xv_2$ e $b = v_1x^{-1}xpv_2$ então tomamos $c = v_1pv_2$.

(2) Se $u = v_1xx^{-1}xv_2$, $a = v_1xv_2 = b$, mesmo se as duas reduções $u \to a$ e $u \to b$ têm subpalavras diferentes de u.

(3) Se $u = v_1xx^{-1}v_2$, $a = v_1v_2 = b$.

Daí, pelo Lema 2.4.4 podemos concluir que o grafo tem exatamente um vértice reduzido e a palavra deste vértice é a única palavra reduzida na classe de $[w]$. □

No grupo F criamos o grafo cujos vértices são todas as palavras geradas por $x_0, x_1, x_2, \cdots$, e podemos construir uma aresta entre dois vértices a, b, ou seja $a \to b$, sempre que seja satisfeita alguma das reduções $\mathcal{R}$ da observação 2.4.2, ou seja que o vértice final b é obtido aplicando uma redução ao vértice inicial a. O grafo do grupo F de Thompson cumpre a primeira condição do lema do diamante pois cada redução ou não muda a quantidade de letras de uma palavra ou diminui o número de letras em duas unidades; as reduções também ordenam as letras positivas no início e as letras negativas no final da palavra. Para verificar a confluência local no grafo de F usamos a mesma ideia da prova da proposição 1.1.11 acima. O número de casos para verificar é maior, mas neste caso também se pode verificar as duas condições do Lema 2.4.4 e obter que, sem importar a ordem com que sejam feitas as reduções, sempre vamos obter um mesmo resultado e então temos uma formal seminormal para cada elemento de F. Omitimos os detalhes neste caso.

O lema 4.2.10 completa o procedimento da seção 2.4, ou seja, obtemos uma prova alternativa do teorema 2.3.9.

Exemplo 2.4.5. *Considere a seguinte palavra* $w = x_0x_3x_6x_3^{-1}x_1x_4^{-1}x_0x_3^{-1}x_0^{-1}$. *Aplicando as regras* $\mathcal{R}$ *vamos obter a forma seminormal de* w*:*

$$\begin{aligned}
x_0x_3x_6x_3^{-1}x_1x_4^{-1}x_0x_3^{-1}x_0^{-1} &= x_0x_3x_6(x_3^{-1}x_1)(x_4^{-1}x_0)x_3^{-1}x_0^{-1}\\
&= x_0x_3x_6x_1(x_4^{-1}x_0)x_5^{-1}x_3^{-1}x_0^{-1}\\
&= x_0x_3x_6(x_1x_0)x_5^{-1}x_5^{-1}x_3^{-1}x_0^{-1}\\
&= x_0x_3(x_6x_0)x_2x_5^{-1}x_5^{-1}x_3^{-1}x_0^{-1}\\
&= x_0(x_3x_0x_2)x_8x_5^{-1}x_5^{-1}x_3^{-1}x_0^{-1}\\
&= x_0(x_0x_2x_5x_8)x_5^{-1}x_5^{-1}x_3^{-1}x_0^{-1}\\
&= x_0x_1x_4x_7x_0(x_5^{-1}x_5^{-1}x_3^{-1}x_0^{-1})\\
&= x_0x_1x_4x_7x_0x_0^{-1}x_4^{-1}x_4^{-1}x_2^{-1}\\
&= x_0x_1(x_4x_7)x_4^{-1}x_4^{-2}x_2^{-1}\\
&= x_0x_1x_6x_4x_4^{-1}x_4^{-1}x_2^{-1}\\
&= x_0x_1x_6x_4^{-1}x_2^{-1}.
\end{aligned}$$

2.5 Apresentações do grupo F

Nesta seção vamos mostrar duas apresentações do grupo F.

Considere

$$F_1 = \langle y_0, y_1, \cdots \mid y_k^{-1}y_ny_k = y_{n+1}, \text{ para } k < n\rangle,$$

e o homomorfismo

$$\Theta : F_1 \longrightarrow F$$

definido pela imagem dos geradores de F_1, tal que $\Theta(y_n) := x_n$, onde x_n é exatamente um dos geradores do grupo F definidos na seção 2.1. Não é difícil verificar que x_n representa o homeomorfismo em F onde a derivada muda nos seguintes pontos:

$$\left(1-\frac{1}{2^n}, 1-\frac{1}{2^n}\right), \left(1-\frac{1}{2^{n+1}}, 1-\frac{3}{2^{n+2}}\right) \text{ e } \left(1-\frac{1}{2^{n+2}}, 1-\frac{1}{2^{n+1}}\right).$$

Observemos que Θ é homomorfismo pois as relações de F_1 são satisfeitas em F e o teorema de Von Dyck 1.2.3.

Lema 2.5.1. *A aplicação Θ é sobrejetiva.*

Demonstração. Um elemento $f \in F$ pode ser representado por um diagrama de árvore $[T, T']$. Pelo teorema 2.3.9 sabemos que o diagrama $[T, T']$ corresponde ao elemento $x_0^{b_0}x_1^{b_1}\cdots x_n^{b_n}x_m^{-a_m}\cdots x_1^{-a_1}x_0^{-a_0}$. Definimos o elemento $y := y_0^{b_0}y_1^{b_1}\cdots y_n^{b_n}y_m^{-a_m}\cdots y_1^{-a_1}y_0^{-a_0}$ então $\Theta(y) = f$. □

Lema 2.5.2. *A aplicação Θ é injetiva.*

Demonstração. Sejam $u, v \in F_1$, $u \neq v$, tal que $\Theta(u) = \Theta(v)$. A imagem desses elementos em F tem uma única forma normal $x_0^{b_0}\ x_1^{b_1} \cdots x_n^{b_n}\ x_m^{-a_m} \cdots x_1^{-a_1}\ x_0^{-a_0}$. Usando os geradores de F_1 correspondentes a cada $x_i \in F$ é possível obter a mesma forma normal a partir de u e de v, ou seja, as duas palavras representam um mesmo elemento. □

Corolário 2.5.3. *O grupo $F \simeq F_1$, ou seja, o grupo F admite a apresentação infinita dada por F_1.*

Demonstração. Segue da bijetividade do homomorfismo Θ definido anteriormente. □

Definição 2.5.4. Sejam x, y elementos de um grupo G, definimos $[x, y] := xyx^{-1}y^{-1}$.

Teorema 2.5.5. *O grupo F admite a seguinte apresentação finita:*

$$F_2 = \langle A, B \mid [AB^{-1}, A^{-1}BA], [AB^{-1}, A^{-2}BA^2] \rangle.$$

Demonstração. A prova deste teorema consiste em definir duas aplicações Φ e Ψ, que sejam uma a inversa da outra, e que estejam bem definidas, ou seja, as aplicações preservam as relações de F_2 e de F (visto através as relações de F_1, pois $F \simeq F_1$ pelo corolário 2.5.3). Agora verificaremos as hipóteses do Teorema de Von Dyck 1.2.3.
Definamos a aplicação:

$$\Psi : F_2 \longrightarrow F,$$

tal que $\Psi(A) = x_0$ e $\Psi(B) = x_1$. Note que

$$x_1^{-1}x_2x_1 = x_0^{-1}x_2x_0 \qquad \text{e} \qquad x_1^{-1}x_3x_1 = x_0^{-1}x_3x_0,$$

então

$$\begin{aligned}
\Psi([AB^{-1}, A^{-1}BA]) &= \Psi(AB^{-1}A^{-1}BABA^{-1}A^{-1}B^{-1}A) \\
&= x_0x_1^{-1}x_0^{-1}x_1x_0x_1x_0^{-1}x_0^{-1}x_1^{-1}x_0 \\
&= x_0x_1^{-1}x_2x_1x_0^{-1}x_2^{-1} \\
&= x_0x_0^{-1}x_2x_0x_0^{-1}x_2^{-1} \\
&= e.
\end{aligned}$$

Analogamente podemos verificar que $\Psi([AB^{-1}, A^{-2}BA^2]) = e$.

Para o homomorfismo inverso definimos

$$\Phi : F \longrightarrow F_2,$$

tal que $\Phi(x_0) = A$ e $\Phi(x_1) = B$. Vejamos que as relações são satisfeitas, para isso precisamos provar que

$$\Phi(x_n) = A^{-(n-1)}BA^{n-1} \text{ para todo } n \geq 2.$$

Denotemos $Y_0 = A$ e $Y_n = A^{-(n-1)}BA^{n-1}$ para $n \geq 1$. Vamos provar que

$$Y_k^{-1}Y_nY_k = Y_{n+1} \tag{2.3}$$

para $k < n$, e que

$$[A^{-1}B, Y_m] = 1 \tag{2.4}$$

para $m \geq 3$. Seja $m = 3$

$$\begin{aligned}
[AB^{-1}, A^{-1}BA] = 1 &\Rightarrow A^{-1}[AB^{-1}, A^{-1}BA]A = 1 \\
&\Rightarrow [B^{-1}A, A^{-2}BA^2] = 1 \\
&\Rightarrow [A^{-1}B, A^{-2}BA^2] = 1 \\
&\Rightarrow [A^{-1}B, Y_3] = 1.
\end{aligned}$$

Analogamente podemos verificar que a afirmação é verdadeira para $m = 4$.
O seguinte processo mostra que 2.3 é verdadeira se 2.4 é verdadeira para $m = n - k + 2$.

$$\begin{aligned}
Y_nY_k &= A^{-n+1}BA^{n-1}A^{-k+1}BA^{k-1} \\
&= A^{-k+2}A^{-(n-k+1)}BA^{n-k+1}A^{-1}BA^{k-1} \\
&= A^{-k+2}Y_{n-k+2}A^{-1}BA^{k-1} \\
&= A^{-k+2}A^{-1}BY_{n-k+2}A^{k-1} \\
&= A^{-k+1}BA^{k-1}A^{-k+1}Y_{n-k+2}A^{k-1} \\
&= Y_kY_{n+1}.
\end{aligned}$$

Daí a afirmação 2.3 é verdadeira para cada inteiro positivo n e $k = n - 1$. Fazendo indução sobre m provamos que $Y_k^{-1}Y_nY_k = Y_{n+1}$ e $[A^{-1}B, Y_m] = 1$. □

Observação 2.5.6. Lembremos que $x_n = A^{-(n-1)}BA^{n-1}$, para $n \geq 1$, $x_0 = A$ e $x_1 = B$. Denotaremos $r_1 = [AB^{-1}, A^{-1}BA]$ e $r_2 = [AB^{-1}, A^{-2}BA^2]$. Pela definição de r_1 temos o seguinte: $x_0x_1^{-1}x_2x_1x_0^{-2}x_1^{-1}x_0 = 1$ e aplicando as regras da forma normal $\mathcal{R}$ obtemos que

$$x_0x_1^{-1}x_2x_1x_0^{-2}x_1^{-1}x_0 = x_0x_1^{-1}x_1x_3x_0^{-2}x_1^{-1}x_0,$$

logo $x_2x_1 = x_1x_3$.

Analogamente, da relação r_2 obtemos que $x_3x_1 = x_1x_4$. Daí podemos reescrever o grupo F_1 da seguinte forma: $F_2 = \langle x_0, x_1 \mid x_2x_1 = x_1x_3, x_3x_1 = x_1x_4\rangle$.

3 Introdução à Criptografia baseada em grupos

O objetivo deste capítulo é mostrar algumas aplicações da teoria de grupos na criptografia. Vamos mostrar alguns problemas algorítmicos da teoria combinatória de grupos e algumas das características desses problemas. As referências usadas para o desenvolvimento desse capítulo foram [7] e [9].

3.1 Problemas algorítmicos de teoria de grupos

Nos últimos 10 anos foram considerados vários tipos de problemas criptográficos em grupos (por exemplo, ver o livro de Myasnikov, Shpilrain e Ushakov [9]).

Existem dois tipos diferentes de problemas algorítmicos de teoria de grupos: O primeiro é o *problema de decisão*, que consiste em determinar um algoritmo que permita saber se um determinado objeto O possui uma propriedade p. Ao passo que o *problema de pesquisa*, já sabendo que existem objetos com a propriedade p, consiste em encontrar pelo menos um objeto particular que cumpra p.

Nos problemas de decisão vamos assumir que existe um algoritmo que permite saber que existem objetos com a propriedade a estudar. Alguns exemplos de problemas algorítmicos de teoria de grupos são os seguintes:

- O *problema da palavra* que consiste em: dado um grupo G finitamente apresentado, se existe um algoritmo que decide, dado $g \in G$, se $g = 1$ ou $g \neq 1$, onde 1 representa o elemento identidade de G. O anterior é um problema de decisão. É importante observar que o algoritmo (se existir) depende do grupo e não do elemento $g \in G$, além disso, é claro que um grupo tem elementos iguais à identidade, o problema é, dada uma sequência de letras, saber se dita sequência corresponde à identidade.

- O *problema de pesquisa da palavra*: dado um grupo finitamente apresentado G se existe um algoritmo que decide, dado um elemento $g \in G$ tal que $g \equiv_G 1$, como escrever g em um produto de conjugados das relações e dos inversos deles.

- O *problema do logaritmo discreto* que consiste em: dado um grupo G, se existe um algoritmo que decida se dados $a, b \in G$, podemos encontrar um número natural x tal que $a^x = b$, ou seja, $a * \cdots * a = b$, x vezes.

- O *problema da conjugação* consiste em: dado um grupo G, se existe um algoritmo que decida se, dados $a, b \in G$, existe um $g \in G$ tal que $a = g^{-1}bg$.

- O *problema da k-conjugação simultânea* consiste em: dado um grupo G finitamente apresentado, se existe um algoritmo que decida se, dados $a_1, \ldots, a_k, b_1, \ldots, b_k \in G$, existe um $g \in G$ tal que $a_i = g^{-1} b_i g$, para todos $i = 1, \ldots, k$.

Existem também os problemas de decomposição e de fatorização, os quais são base de alguns algoritmos criptográficos de geração de chaves e que vão ser estudados mais na frente.

3.2 O protocolo de Diffie-Hellman

Um *protocolo* é um algoritmo definido por uma lista de passos específicos que precisam de duas ou mais partes para alcançar o objetivo. A troca de chaves de Diffie-Hellman é um método de criptografia específico desenvolvido por Whitfield Diffie e Martin Hellman e publicado em 1976. Foi um dos primeiros exemplos práticos de métodos de troca de chaves implementado dentro do campo da criptografia. O método da troca de chaves de Diffie-Hellman permite que duas partes que não possuem conhecimento a priori de cada uma, compartilhem uma chave secreta sob um canal de comunicação.

A seguir descrevemos o protocolo de Diffie-Hellman:

(1) Duas pessoas, digamos Alice e Bob, escolhem um grupo cíclico finito G e um elemento gerador g em G. (Isto é usualmente feito muito antes do resto do protocolo; o elemento g é público.)

(2) Alice escolhe um número natural aleatório a e envia g^a para Bob.

(3) Bob também escolhe um número natural aleatório b e envia g^b para Alice.

(4) Alice calcula $(g^b)^a$.

(5) Bob calcula $(g^a)^b$.

(6) A chave secreta será $K = (g^b)^a = (g^a)^b$.

Uma terceira pessoa intercepta g^a e g^b, e precisa encontrar um valor c tal que $g^c = g^a$, então calcula $(g^b)^c = (g^c)^b = (g^a)^b$, logo se determina a chave secreta K. Para resolver o protocolo de Diffie-Hellman precisamos de resolver o problema do logaritmo discreto, mas não está provado se o problema de Diffie-Hellman e o problema do logaritmo discreto sejam equivalentes.

Existem ataques de força ao protocolo de Diffie-Hellman que consistem em fazer potencias até obter o número natural c, ou seja, calcular $g^1, g^2, \cdots$ até que $g^c = g^a$, este ataque é muito devagar, mas é eficiente. De fato, para calcular g^n de maneira eficiente

têm que ser feitas normalmente $O(|g|)$ multiplicações, e para calcular g^a para um valor particular de a são necessarios $O(\log_2 a)$ cálculos.

A chave secreta é usada por Bob e Alice para enviar mensagens cifrados. Neste trabalho não nos concentramos sobre cómo codificar mensagens através da chave secreta, mas sobre como trocar a chave e os ataques para achar esta chave.

3.3 O criptosistema RSA

O protocolo RSA é um algoritmo de criptografia de dados, que deve o seu nome a três professores do Instituto de Tecnologia de Massachusetts, Ronald Rivest, Adi Shamir e Leonard Adleman. Atualmente o RSA é considerado um dos sistemas de criptografia mais seguros que existe.

No protocolo RSA as chaves são geradas da seguinte maneira:

(1) Escolha de forma aleatória dois números primos grandes p e q, da ordem de 10^{100} no mínimo.

(2) Calcule $n = pq$.

(3) Calcule a função de Euler: $\phi(n) = (p-1)(q-1)$.

(4) Escolha um inteiro e tal que $1 < e < \phi(n)$, de forma que e e $\phi(n)$ sejam primos entre si.

(5) Calcule d, de forma que $de \equiv 1 \pmod{\phi(n)}$, ou seja, d seja o inverso multiplicativo de $e \pmod{\phi(n)}$.

Por fim obtemos a chave pública (n, e) e a chave privada que é a tripla (p, q, d). Note que calcular $\phi(n)$ não é fácil para o inimigo, isso implica a dificuldade para calcular d.

RSA baseia-se no fato de que, embora seja fácil encontrar dois números primos de grandes dimensões (100 dígitos), conseguir factorizar o produto de tais dois números é considerado computacionalmente complexo, em outras palavras, o tempo estimado para o conseguir ronda os milhares de anos.

Para compreender melhor alguns detalhes da aplicação do protocolo RSA vamos mencionar o seguinte resultado (a prova deste teorema não faz parte dos objetivos deste trabalho, por isso não vai ser apresentada):

Teorema 3.3.1. *(Teorema de Euler)*
Se a e $n \in \mathbb{Z}$ tal que $MDC(n, a) = 1$ (máximo divisor comum entre n e a) então

$$a^{\Phi(n)} \equiv 1 \ mod \ n,$$

onde $\Phi(n)$ *representa a* função de Euler *(quantidade de números primos relativos a* n*, que sejam menores que* n*).*

Para melhor compreensão do protocolo RSA, daremos um exemplo tomado de [13]. Primeiramente devemos ter uma mensagem que deva ser escondida do público, e vamos criptografa-la. Usaremos a seguinte frase hipotética:

"Achamos Osama Bin Laden no Tibet".

O primeiro passo a ser dado é dar valores numéricos para cada letra do alfabeto, para que possamos trabalhar todas as letras usando estes valores que lhes foram atribuídos. A escolha dos valores numéricos tem que ser cuidadosa para se evitar ambiguidades, por exemplo se $a = 1, b = 2, c = 3, \ldots, m = 13, \ldots$, vamos ter que m=ac.

Neste exemplo usaremos os seguintes valores numéricos: $a = 11, b = 12, c = 13, \ldots, z = 36$ e o espaço entre letras será representado pelo número 37. Aplicando isto na frase acima obtemos:

11131811232529372529112311371219243722111415243724253730191215 30

Neste momento vamos gerar dois primos distintos para desenvolver as chaves. Considere $p = 5$ e $q = 11$. Logo $n = 55$, $\phi(n) = 40$ e o menor número primo que não divide $\phi(n)$ será 3. Neste caso a chave de codificação é (n, e).
Depois de conhecer os valores de $n, \phi(n)$ e e, fragmentaremos aquele número em partes ou blocos B_i sempre menores que n, pois vamos usar a congruência $mod\ n$. Como exemplo:

$1-11-31-8-1-12-32-52-9-37-25-2-9-1-12-31-13-7-12-19-$
$2-43-7-2-21-11-41-52-43-7-2-42-53-7-30-19-1-21-5-30.$

A seguir pegamos cada bloco B_i e aplicamos a congruência

$$B_i^e \equiv C_i \pmod{(n)}.$$

Por exemplo

$$31^3 \equiv 29791 \equiv 36 \ (mod\ 55).$$

Após fazermos este processo com todos os blocos, teremos a mensagem codificada. Ela seria a seguinte:

$1-11-36-17-1-23-43-28-14-53-5-8-14-1-23-36-52-13-23-39-$
$8-32-13-8-21-11-6-28-32-13-8-3-47-13-50-39-1-21-15-50.$

Desse modo a pessoa que deseje decodificar esta mensagem terá de encontrar a chave de decodificação. Dita chave precisa de dois números: o primeiro é n e o segundo vai ser um número inteiro d tal que $MDC(n, d) = 1$ e que também cumpra o seguinte:

$$de + \beta\Phi(n) = 1.$$

Essa pessoa terá de conhecer $\phi(n)$ para aplicar o algoritmo euclidiano estendido em $\phi(n)$ e em e para encontrar d. Assim teremos que $d = 27$, pois

$$\phi(n) = 3(13) + 1, \text{ logo } 1 = 40 + 3(-13),$$

e como $-13 \equiv 27 \ \ mod \ \ 40$ então 27 é o inverso de 3 módulo 40. Assim vamos usar $d = 27$ e a congruência adequada para decodificar a mensagem codificada.

Vamos usar a seguinte congruência para decodificar a mensagem:

$$C_i^d \equiv D_i \ \ mod \ \ n.$$

Precisamos provar que $D_i = B_i$. No processo fizemos o seguinte:

$$(B_i^e)^d) = C_i^d \equiv D_i \ \ mod \ \ n,$$

e sabemos que $(B_i^e)^d) \equiv B_i^{1-\beta\Phi(n)} \equiv D_i \ \ mod \ \ n$ então $B_i(B_i^{\Phi(n)})^{-\beta} \equiv D_i \ \ mod \ \ n$.

Pelo Teorema de Euler 3.3.1 temos que $B_i^{\Phi(n)} \equiv 1 \ \ mod \ \ n$ então

$$B_i(1)^{-\beta} \equiv B_i \equiv D_i \ \ mod \ \ n.$$

Conforme nosso caso numérico temos que

$$31^{ed} = 31^{3.27} \equiv 31 \ \ mod \ \ 55.$$

3.4 O protocolo de I. Anshel, M. Anshel e Goldfeld

Em 1999, I. Anshel, M. Anshel e Goldfeld em [1] propuseram o seguinte protocolo sobre o problema de conjugação simultânea:
Considere a 5-upla $(U, V, \beta, \gamma_1, \gamma_2)$ onde U, V são monóides e

$$\beta : U \times U \longrightarrow V, \qquad \gamma_i : U \times V \longrightarrow V,$$

tais que:

(i) Para todos $x, y_1, y_2 \in U$,

$$\beta(x, y_1y_2) = \beta(x, y_1)\beta(x, y_2).$$

(ii) Para todos $x, y \in U$,

$$\gamma_1(x, \beta(y, x)) = \gamma_2(y, \beta(x, y)).$$

(iii) Suponha que $y_1, y_2, \cdots, y_k \in U$ e $\beta(x, y_1), \cdots, \beta(x, y_k)$ são elementos públicos para algum elemento privado $x \in U$, e que em geral é impossível encontrar x.

Sejam S_A, $S_B \subseteq U$ submonóides associados a determinados usuários A e B, respectivamente. Suponha que

$$S_A = \langle s_1, \cdots, s_m \rangle \quad \text{e} \quad S_B = \langle t_1, \cdots, t_n \rangle.$$

O usuário A escolhe $a \in S_A$ e envia os elementos $\beta(a, t_i)$, $i = 1, \cdots, n$. Analogamente o usuário B escolhe $b \in S_B$ e envia os elementos $\beta(b, s_i)$, $i = 1, \cdots, m$. Pela propriedade iii) em cada transmissão num canal público, os elementos a e b estão protegidos. Pela propriedade i) o usuário A pode calcular

$$\beta(b, a) \quad \text{e} \quad \gamma_1(a, \beta(b, a)),$$

e o usuário B calcula

$$\beta(a, b) \quad \text{e} \quad \gamma_2(b, \beta(a, b)).$$

Pela propriedade ii) podemos afirmar que a chave pode ser:

$$k = \gamma_1(a, \beta(b, a)) = \gamma_2(b, \beta(a, b)).$$

Se o monóide U é um grupo e $U = V$, então são definidos os seguintes subgrupos em U, denotado agora por G:

$$S_A = \langle s_1, s_2, \cdots, s_m \rangle \quad \text{e} \quad S_B = \langle t_1, t_2, \cdots, t_n \rangle \ .$$

Neste caso a função $\beta : G \times G \longrightarrow G$ é a conjugação de y por x:

$$\beta(x, y) = x^{-1}yx,$$

e as funções γ_i são definidas como segue:

$$\gamma_1(u, v) = u^{-1}v \quad \text{e} \quad \gamma_2(u, v) = v^{-1}u \ .$$

Resumindo o escrito acima, os usuários A e B escolhem os elementos secretos $a \in S_A$ e $b \in S_B$, respectivamente. O usuário A inicia o protocolo calculando e enviando os elementos $a^{-1}t_1a$, $a^{-1}t_2a, \cdots a^{-1}t_na$.
Da mesma forma o usuário B calcula e envia os elementos $b^{-1}s_1b$, $b^{-1}s_2b, \cdots b^{-1}s_mb$.
O inimigo precisa resolver o conjunto de equações de conjugação simultaneamente para encontrar os elementos secretos a e b.
Para obter a chave, o usuário A calcula

$$K = \gamma_1(a, \beta(b, a)) = a^{-1}b^{-1}ab := [a, b]$$

e o usuário B calcula

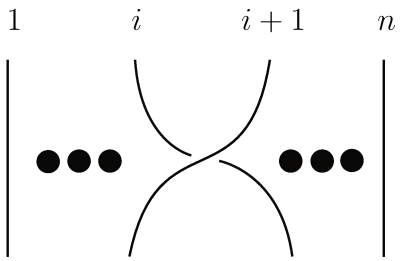
$$K = \gamma_2(b, \beta(a, b)) = [a, b].$$

3.5 Protocolo de conjugação no grupo de tranças

I. Anshel, M. Anshel e D. Goldfeld [1] sugeriram usar o grupo das tranças em protocolos criptográficos e que foi depois desenvolvido por K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang e C. Park [6] é estudado o problema de conjugação tendo como plataforma o grupo de tranças denotado por B_n. Este grupo é infinito e não abeliano e tem a seguinte apresentação:

$$B_n = \langle \sigma_1, \cdots, \sigma_{n-1} \mid \sigma_i\sigma_j = \sigma_j\sigma_i, \qquad \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}, \qquad \text{para } |i-j| \geq 2 \rangle.$$

Este grupo tem uma interpretação geométrica particular, onde os geradores σ_i são da seguinte forma:



Estes objetos podem se imaginar como n fios que se cruzam ao longo do caminho de acima a abaixo, e são chamados de *tranças*.

Em B_n são definidos os seguintes subgrupos:

$$LB_n := \langle \sigma_1, \cdots, \sigma_{\frac{n}{2}-1} \rangle$$
$$UB_n := \langle \sigma_{\frac{n}{2}+1}, \cdots, \sigma_{n-1} \rangle.$$

Esses subgrupos têm a propriedade de que $ab = ba$ para todo $a \in LB_n$ e todo $b \in UB_n$. O problema de pesquisa da conjugação aplicado no grupo B_n consiste em que dados $a, b \in B_n$ tal que $a = x^{-1}bx$ para algum $x \in LB_n$, encontrar $y \in LB_n$ tal que $a = y^{-1}by$. Este problema também pode ser estudado tomando os elementos x e y no subgrupo UB_n.

Se escrevemos o protocolo proposto por I. Anshel, M. Anshel e Goldfeld usando o grupo de tranças como plataforma, obtemos o seguinte:

1) Informação pública:

- O índice n do grupo B_n.
- Alice publica o subgrupo $G_A = \langle x_1, \cdots, x_s \rangle \subseteq B_n$, especificando os seus geradores.
- Bob publica o subgrupo $G_B = \langle y_1, \cdots, y_t \rangle \subseteq B_n$, especificando os seus geradores.

2) Geração de chaves

- Alice escolhe a palavra secreta $a = W(x_1, \cdots, x_s) \in G_A$ e envia $a^{-1}y_i a$, para cada $i = 1, \cdots, t$, ao Bob.
- Bob escolhe a palavra secreta $b = W(y_1, \cdots, y_t) \in G_B$ e envia $b^{-1}x_i b$ para cada $i = 1, \cdots, s$ a Alice.
- Alice calcula $K = a^{-1}(b^{-1}ab)$ que corresponde ao comutador de a e b.
- Bob calcula $K = (a^{-1}b^{-1}a)b$.

Assim, podemos ver que diferentes grupos estão sendo estudados como plataformas dos protocolos de criptografia, por exemplo o grupo das tranças (estudado anteriormente) e o grupo F de Thompson (que será estudado no próximo capitulo).

4 Criptografia no grupo F de Thompson

Os resultados mostrados neste capítulo foram estudados por Matucci, Shpilrain e Ushakov em [8] e [12].

4.1 Protocolo do problema de decomposição

Em [12] Shpilrain e Ushakov estudaram o problema de *decomposição*, que é uma generalização do problema de conjugação e que é semelhante ao problema de fatorização, este por sua vez, é o coração do criptosistema RSA. O problema de decomposição consiste em: Dados $w \in G$ (semigrupo), $A \subseteq G$ e um elemento $x \cdot w \cdot y$, encontrar $x^{'}$, $y^{'} \in A$ tais que $x^{'} \cdot w \cdot y^{'} = x \cdot w \cdot y$.

O protocolo de troca de chaves no problema de decomposição é descrito a seguir:

Considere $A, B \subseteq G$ tais que $ab = ba$, para todo $a \in A$ e para todo $b \in B$, e $w \in G$, onde A, B e w são elementos públicos.

(1) Alice: Escolhe $a_1, a_2 \in A$ privados e envia $a_1 w a_2$.

(2) Bob: Escolhe $b_1, b_2 \in B$ privados e envia $b_1 w b_2$.

(3) Alice: Calcula $K_a = a_1 b_1 w b_2 a_2$.

(4) Bob: Calcula $K_b = b_1 a_1 w a_2 b_2$.

Note que $K_a = K_b = K$, pois $a_i b_i = b_i a_i$ para todo i. K é um elemento de G, que recebe o nome de chave secreta compartilhada entre Alice e Bob.

Neste trabalho Shpilrain e Ushakov usam o grupo F de Thompson para aplicar o protocolo criptográfico, e eles propõem a seguinte modificação:

(1) Alice: Escolhe $a_1 \in A, b_1 \in B$ privados e envia $a_1 w b_1$.

(2) Bob: Escolhe $b_2 \in B, a_2 \in A$ privados e envia $b_2 w a_2$.

(3) Alice: Calcula $a_1 b_2 w a_2 b_1$.

(4) Bob: Calcula $b_2 a_1 w b_1 a_2$.

Definição 4.1.1. Sejam $S_{A_s} = \{x_0 x_1^{-1}, \cdots, x_0 x_s^{-1}\}$ e $S_{B_s} = \{x_{s+1}, \cdots, x_{2s}\}$, definimos os seguintes subgrupos de F:

$$A_s = \langle S_{A_s} \rangle$$
$$B_s = \langle S_{B_s} \rangle.$$

Lema 4.1.2. *Para cada inteiro positivo k, seja $\varphi_k = 1 - \frac{1}{2^{k+1}}$. Cada gerador $x_0 x_k^{-1}$ do subgrupo A_s corresponde ao homeomorfismo identidade no intervalo $[\varphi_k, 1]$.*

Demonstração. Lembrando a definição de $x_k = x_0^{-(k-1)} x_1 x_0^{k-1}$ para $k \geq 2$ temos que

$$x_k^{-1}([\varphi_k, 1]) = [\varphi_{k+1}, 1] \subseteq \left[\frac{3}{4}, 1\right]$$

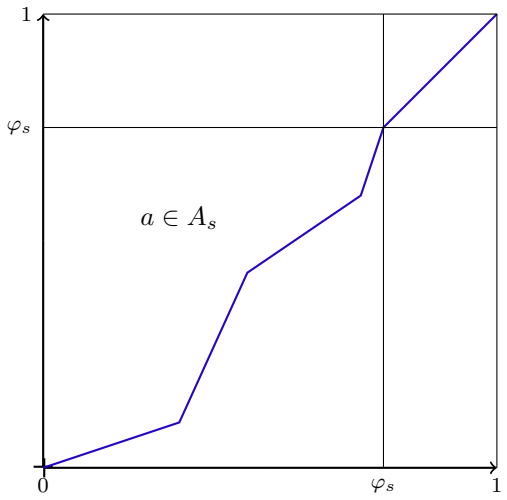
isto implica que, para $t \in [\varphi_k, 1]$,

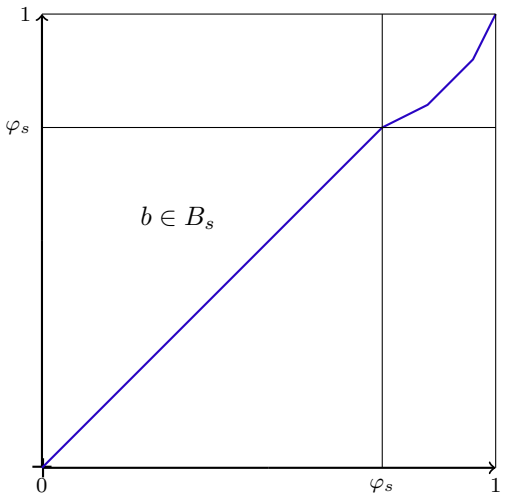
$$\frac{d}{dt} x_0 x_k^{-1}(t) = x_0'(x_k^{-1}(t))(x_k^{-1})'(t) = 2 \cdot \frac{1}{2} = 1.$$

Logo $x_0 x_k^{-1}$ é a função identidade no intervalo $[\varphi_k, 1]$. □

Lema 4.1.3. *Para cada $s \in \mathbb{N}$ fixo, $ab = ba$ para cada elemento $a \in A_s$ e $b \in B_s$.*

Demonstração. Pela observação 2.1.6 temos que o gerador x_n do grupo F é a identidade no intervalo $\left[0, 1 - \frac{1}{2^n}\right]$, logo, um gerador do subgrupo B_s que tem a forma x_{s+m} corresponde ao homeomorfismo identidade no intervalo $\left[0, 1 - \frac{1}{2^{s+m}}\right]$, em particular x_{s+1} é igual ao homeomorfismo identidade no intervalo $[0, \varphi_s]$, daí concluímos que todos os elementos do subgrupo B_s são a identidade no intervalo $[0, \varphi_s]$. Usando este fato e pelo lema anterior, temos que os elementos de A_s e B_s podem ser representados como segue:





Daí segue que $ab = ba$ para todo $a \in A_s$ e todo $b \in B_s$. □

Para cada número diádico $d \in [0, 1]$, denotaremos por $PL_2([0, d])$ o conjunto de funções em $PL_2(I)$ que são a identidade no intervalo $[d, 1]$.

4.1.1 Parâmetros e geração de chaves

Em [12] os autores escolhem os seguintes parâmetros:

(i) Escolher aleatoriamente $s \in [3, 8]$ e $M \in \{256, 258, \cdots, 318, 320\}$.

(ii) Escolha a palavra base w como um produto dos geradores $S_W = \{x_0, x_1, \cdots, x_{s+2}\}$ e os inversos dos elementos de S_W, tal que a forma normal de w tenha comprimento M, sendo w obtida com uma palavra vazia $u_0 = 1$ e multiplicando à direita por algum gerador x_{s+i} de $S_W^{\pm 1}$, calculamos a forma normal do produto obtendo u_1, continuando o processo até obter uma palavra em forma normal de comprimento M.

(iii) Selecionar a_1 e a_2 no subgrupo A_s e repetimos o processo do passo anterior, até construir palavras de comprimento M.

(iv) Selecionar b_1 e b_2 elementos em B_s e seus inversos até construir palavras de comprimento M.

Finalmente é gerada a chave secreta $K = a_1 b_1 w b_2 a_2$.

4.2 O problema da palavra no grupo de Thompson F

Em [14] é definido o *time complexity* de um algoritmo como a quantidade de tempo que um algoritmo demora para executar em função do comprimento da sequência que representa a entrada. A ideia desta seção é mostrar que o *time complexity* para reduzir uma palavra no grupo F, tal que sua forma normal tenha comprimento n, é calculado pela expressão $O(|n| \log |n|)$. O algoritmo para calcular a forma normal é obtido usando outros algoritmos que vamos estudar na continuação.

4.2.1 Alguns algoritmos para calcular formas seminormais

Usando as relações $\mathcal{R}$ é possível encontrar as formas seminormal e normal de uma palavra qualquer, nesta seção vamos mostrar alguns algoritmos para fazer este processo, partindo de alguns casos particulares até obter um algoritmo geral para calcular a forma seminormal de uma palavra.

Primeiro vamos considerar o caso onde a palavra $w = w_1 w_2$, sendo w_1 e w_2 formas seminormais.

Sejam $w_1 = p_1 n_1$ e $w_2 = p_2 n_2$ tal que p_i e n_i são as partes positiva e negativa de w_i, respectivamente. Daí podemos escrever $w = p_1 n_1 p_2 n_2$.

Usando as regras $\mathcal{R}$ é possível calcular a forma seminormal de $n_1 p_2$ denotada por $p_2' n_1'$, e denotamos o resultado obtido por: $w' = p_1 p_2' n_1' n_2$.

Depois reescrevemos a subpalavra positiva p_1p_2' em forma seminormal denotada por p e também obtemos a forma seminormal n correspondente à subpalavra negativa $n_1'n_2$ obtendo $w'' = pn$.

Definição 4.2.1. Seja $\epsilon \in \mathbb{Z}$ definimos a função δ_ϵ definida sobre o conjunto de todas as palavras no alfabeto $\{x_0^{\pm 1}, x_1^{\pm 1}, \cdots\}$ tal que

$$\delta_\epsilon(x_{i_1}^{\pm 1} \cdots x_{i_n}^{\pm 1}) = x_{i_1+\epsilon}^{\pm 1} \cdots x_{i_n+\epsilon}^{\pm 1}.$$

A função δ_ϵ pode não estar definida para alguns ϵ negativos, porém, quando seja usada vamos supor que a função está definida para o valor de ϵ escolhido.

O seguinte algoritmo executa os passos descritos acima:

Algoritmo 1: Forma seminormal do produto entre uma forma seminormal negativa e uma positiva.

Entrada: Formas seminormais n e p (onde $n = x_{j_t}^{-1} \cdots x_{j_1}^{-1}$ e $p = x_{i_1} \cdots x_{i_s}$), e números $\epsilon_1, \epsilon_2 \in \mathbb{Z}$

Signatura: $w = Merge_{-,+}(n, p, \epsilon_1, \epsilon_2)$

Saída: Forma seminormal w tal que $w =_F \delta_{\epsilon_1}(n)\delta_{\epsilon_2}(p)$.

início

  para $s = 0$ *ou* $t = 0$ **faça**
    $w = np$.
  fim
  retorna w
  para $j_1 + \epsilon_1 = i_1 + \epsilon_2$ **faça**
    $w = Merge_{-,+}(x_{j_t}^{-1} \cdots x_{j_2}^{-1}, x_{i_2} \cdots x_{i_s}, \epsilon_1, \epsilon_2)$
  fim
  retorna w
  para $j_1 + \epsilon_1 < i_1 + \epsilon_2$ **faça**
    $w = Merge_{-,+}(x_{j_t}^{-1} \cdots x_{j_2}^{-1}, x_{i_1} \cdots x_{i_s}, \epsilon_1, \epsilon_2 + 1)$
  fim
  retorna $wx_{j_1+\epsilon_1}^{-1}$
  para $j_1 + \epsilon_1 > i_1 + \epsilon_2$ **faça**
    $w = Merge_{-,+}(x_{j_t}^{-1} \cdots x_{j_1}^{-1}, x_{i_2} \cdots x_{i_s}, \epsilon_1 + 1, \epsilon_2)$
  fim
  retorna $x_{i_1+\epsilon_2} w$

fim

Usando as ideias do algoritmo anterior é possível implementar algoritmos para palavras positivas e negativas, esses algoritmos vão ser denotados por $Merge_{+,+}(p_1, p_2, \epsilon_1, \epsilon_2)$ e $Merge_{-,-}(n_1, n_2, \epsilon_1, \epsilon_2)$, respectivamente.

Lema 4.2.2. *Sejam $n = x_{j_t}^{-1} \cdots x_{j_1}^{-1}$ e $p = x_{i_1} \cdots x_{i_s}$ formas seminormais e $\epsilon_1, \epsilon_2 \in \mathbb{Z}$, a saída w do algoritmo 1 é a forma seminormal da palavra $\delta_{\epsilon_1}(n)\delta_{\epsilon_2}(p)$. Além disso, o time complexity necessário para calcular w está limitado por $C(|n| + |p|)$ para alguma constante C.*

Demonstração. Por indução sobre $|n| + |p|$:
Se $|n| + |p| = 0$ então $w = np$ é a palavra vazia, logo o lema é verdadeiro. Suponha verdadeira a afirmação para $|n| + |p| = N$. Se $|n| + |p| = N + 1$ considere os seguintes casos:

(i) Se $|n| = 0$ ou $|p| = 0$ então uma das palavras é a palavra trivial, logo np é a forma seminormal.

(ii) Se $j_1 + \epsilon_1 = i_1 + \epsilon_2$ então $x_{j_1+\epsilon_1}^{-1} x_{i_1+\epsilon_2} = 1$, logo $|\delta_{\epsilon_1(n)}\delta_{\epsilon_2(p)}| < |n| + |p|$, logo a afirmação é verdadeira.

(iii) Se $j_1 + \epsilon_1 < i_1 + \epsilon_2$ temos que $\delta_{\epsilon_1(n)}\delta_{\epsilon_2(p)} = x_{j_t+\epsilon_1}^{-1} \cdots x_{j_2+\epsilon_1}^{-1} x_{j_1+\epsilon_1}^{-1} x_{i_1+\epsilon_2} \cdots x_{i_s+\epsilon_2}$, aplicando $\mathcal{R}$ obtemos $x_{j_t+\epsilon_1}^{-1} \cdots x_{j_2+\epsilon_1}^{-1} x_{i_1+\epsilon_2+1} \cdots x_{i_s+\epsilon_2+1} x_{j_1+\epsilon_1}^{-1}$.

Como n e p são formas seminormais, então $j_1 + \epsilon_1$ é o menor índice em $\delta_{\epsilon_1(n)}\delta_{\epsilon_2(p)}$.

Agora o menor índice de $w = Merge_{-,+}(x_{j_t}^{-1} \cdots x_{j_2}^{-1}, x_{i_2} \cdots x_{i_s}, \epsilon_1, \epsilon_2)$ não é menor que $j_1 + \epsilon_1$ e

$$\delta_{\epsilon_1}(x_{j_t}^{-1}, \cdots, x_{j_2}^{-1})\delta_{\epsilon_2}(x_{i_2}, \cdots, x_{i_s}) = x_{j_t+\epsilon_1}^{-1} \cdots x_{j_2+\epsilon_1}^{-1} x_{i_2+\epsilon_2} \cdots x_{i_2+\epsilon_s}.$$

Ou seja que w é a forma seminormal de $\delta_{\epsilon_1}(x_{j_t}^{-1} \cdots x_{j_2}^{-1})\delta_{\epsilon_2}(x_{i_2} \cdots x_{i_s})$. O caso quando $j_1 + \epsilon_1 > i_1 + \epsilon_2$ é análogo. Finalmente obtemos que $w x_{j_1+\epsilon_2+1}^{-1}$ é a forma normal de $\delta_{\epsilon_1}(n)\delta_{\epsilon_2}(p)$.
Para provar que o *time complexity* é $C(|n| + |p|)$, note que é possível mover o elemento $x_{j_1+\epsilon_1^{-1}}$ percorrendo as letras positivas em apenas um passo, pois esse passo implica aumentar em 1 cada subíndice das letras positivas. Fazendo esse processo para cada $x_{j_k+\epsilon_1}$ com $1 \leq k \leq n$ levamos n passos; de forma análoga $x_{i_1+\epsilon_2}$ percorre as letras negativas em apenas um passo, e fazendo esse processo para cada $x_{i_t+\epsilon_2}$, com $1 \leq t \leq p$, somamos mais p passos, obtendo o resultado desejado. Na expressão $C(|n| + |p|)$, C representa uma constante que depende das características de cada computador. □

Exemplo 4.2.3. *Considere $n = x_4^{-1} x_2^{-1} x_1^{-1}$, $p = x_2 x_3$ formas seminormais e sejam $\epsilon_1 = 1$ e $\epsilon_2 = 2$.*

Calcule $w = Merge_{-,+}(n, p, \epsilon_1, \epsilon_2)$, ou seja, $w = x_5^{-1} x_3^{-1} x_2^{-1} x_4 x_5$. Note que $j_1 + \epsilon_1 < i_1 + \epsilon_2$, então calcule $w = Merge_{-,+}(n', p, \epsilon_1, \epsilon_2 + 1)$ onde $n' = x_4^{-1} x_2^{-1}$, ou seja, temos que calcular a forma seminormal de $w = x_5^{-1} x_3^{-1} x_5 x_6$.

Neste primeiro ciclo do algoritmo 1 obtemos $x_5^{-1}x_3^{-1}x_5x_6x_2^{-1}$ *(se aplicamos as regras* $\mathcal{R}$ *obtemos também o resultado anterior).*

Continuando com o processo, agora para a palavra $w_2 = x_5^{-1}x_3^{-1}x_5x_6x_2^{-1}$ *(obtida no passo anterior), novamente temos que* $j_2+\epsilon_1 < i_1+\epsilon_2+1$ *logo obtemos* $w = x_5^{-1}x_6x_7x_3^{-1}x_2^{-1}$*, e finalmente aplicando de novo o algoritmo obtemos a palavra* $w_3 = x_7x_8x_5^{-1}x_3^{-1}x_2^{-1}$*, que é a forma seminormal desejada.*

Exemplo 4.2.4. *Vejamos agora o caso quando* $j_1 + \epsilon_1 < i_2 + \epsilon_2$*: considere* $\epsilon_1 = 3$ *e* $\epsilon_1 = 2$*, ou seja,* $w = x_7^{-1}x_5^{-1}x_2^{-1}x_3x_4$*.*

Calcule $w = Merge_{-,+}(n, p', \epsilon_1 + 1, \epsilon_2)$ *onde* $p' = x_4$*, daí temos que calcular a forma seminormal de* $w = x_8^{-1}x_6^{-1}x_5^{-1}x_4$*.*

Saída: $x_{i_1+\epsilon_2}w$, ou seja, $x_3x_8^{-1}x_6^{-1}x_5^{-1}x_4$.

Calcule a forma seminormal de $x_9^{-1}x_7^{-1}x_6^{-1}$.

Saída: $x_{i_2+\epsilon_2}w = x_4x_9^{-1}x_7^{-1}x_6^{-1}$.

Obtendo o resultado final: $x_3x_4x_9^{-1}x_7^{-1}x_6^{-1}$.

Na continuação é mostrado o algoritmo para calcular a forma seminormal de duas formas seminormais quaisquer w_1, w_2, usando as ideias do algoritmo 1.

Algoritmo 2: Forma seminormal do produto de formas seminormais.

Entrada: Formas seminormais w_1 e w_2

Signatura: $w = Merge(w_1, w_2)$

Saída: Forma seminormal w tal que $w =_F w_1w_2$.

início

A) Represente w_i como um produto de uma palavra positiva e outra negativa $w_i = p_in_i$.

B) Calcule $w' = Merge_{-,+}(n_1, p_2, 0, 0)$ e represente isto como um produto de palavras positiva e negativa $w' = p_2'n_1'$.

C) Calcule $w'' = Merge_{+,+}(p_1, p_2', 0, 0)$.

D) Calcule $w''' = Merge_{-,-}(n_1', n_2, 0, 0)$.

retorna $w''w'''$

fim

Exemplo 4.2.5. *A) Considere* $w_1 = p_1n_1 = x_2x_4x_5x_6^{-1}x_3^{-1}$ *e* $w_2 = p_2n_2 = x_1x_2x_5^{-1}x_0^{-1}$*. B) Considere* $n_1p_2 = x_6^{-1}x_3^{-1}x_1x_2$ *e aplicamos o algoritmo 1. O resultado é* $x_1x_2x_8^{-1}x_5^{-1} = p_2'n_1'$*.*

C) Agora calculamos a forma seminormal de p_1p_2':

$$\begin{aligned} x_2x_4x_5x_1x_2 &= x_2x_4x_1x_2x_7 \\ &= x_2x_1x_2x_6x_7 \\ &= x_1x_2x_4x_6x_7 \\ &= w''. \end{aligned}$$

D) Calculamos a forma seminormal de $n_1'n_2$, *daí* $w''' = x_8^{-1}x_5^{-1}x_5^{-1}x_0^{-1}$.
E) O resultado final é $w = w''w'''$.

Lema 4.2.6. *Sejam* w_1, w_2 *formas seminormais arbitrárias, então a saída ou resultado do algoritmo anterior é a forma seminormal do produto* w_1w_2.

Demonstração. É consequência do lema 4.2.2, pois na parte B aplicando o algoritmo 1 obtemos a forma seminormal de w'. □

Finalmente vamos analisar um algoritmo geral para calcular formas seminormais.

Algoritmo 3: FORMA SEMINORMAL.

Entrada: A palavra w em geradores do grupo F
Signatura: $u =$ FormaSemiNormal (w).
Saída: Forma seminormal u de w tal que $u = w$ em F.
início
  para $|w| \leq 1$ **faça**
    w
  fim
  para $|w| > 1$ **faça**
    A) Represente w como um produto w_1w_2 tal que $||w_1| - |w_2|| \leq 1$.
    B) Calcule recursivamente:
    $u_1 = FormaSemiNormal(w_1)$;
    $u_2 = FormaSemiNormal(w_2)$.
    C) Seja $u = Merge(u_1, u_2)$.
  fim
  retorna u
fim

Exemplo 4.2.7. *Seja* $w = x_0x_3^{-1}x_4^{-1}x_5x_1x_2x_1^{-1}$ *e escolhemos* $w_1 = x_0x_3^{-1}x_4^{-1}x_5$ *e* $w_2 = x_1x_2x_1^{-1}$.

Como consequência dos algoritmos 1 e 2 obtemos as formas seminormais:

$$\begin{aligned} w_1 &= x_0 x_3^{-1} x_4^{-1} x_5 \\ &= x_0 x_7 x_3^{-1} x_4^{-1} \\ &= x_0 x_7 x_5^{-1} x_3^{-1} \\ &= u_1. \end{aligned}$$

Neste caso $w_2 = x_1 x_2 x_1^{-1} = u_2$ *já está em forma seminormal.*

Aplicamos agora o algoritmo 2 para calcular a forma seminormal de $u_1 u_2 = x_0 x_7 x_5^{-1} x_3^{-1} x_1 x_2 x_1^{-1} = p_1 n_1 p_2 n_2$.
Depois aplicamos o algoritmo 1 para calcular a forma seminormal de $n_1 p_2$ *e obtemos* $x_1 x_2 x_7^{-1} x_5^{-1} = p_2' n_1' = w'$.

Agora tomando $p_1 p_2'$ *e calculando a forma seminormal obtemos* $x_0 x_1 x_2 x_9 = w''$.
Tome $n_1' n_2 = x_7^{-1} x_5^{-1} x_1^{-1} = w'''$.

Finalmente, o resultado é $w'' w''' = u$, *ou seja* $u = x_0 x_1 x_2 x_9 x_7^{-1} x_5^{-1} x_1^{-1}$.

Lema 4.2.8. *Seja w uma palavra em F. O resultado do algoritmo 3 é uma forma seminormal de w. A quantidade de operações necessárias neste algoritmo é $\mathcal{O}(|w| \log |w|)$.*

Demonstração. Seja $T(n)$ a quantidade de passos requeridos no algoritmo 3 aplicado numa palavra de comprimento n.

Note que $T(n) = 2T\left(\frac{n}{2}\right) + Cn$, pois o algoritmo divide uma palavra w em duas subpalavras w_1 e w_2, ou seja, o tempo para achar u_1 é $T\left(\frac{n}{2}\right)$ e este é também o tempo para achar u_2. O time complexity de $Merge(w_1, w_2)$ está dado por Cn onde $n = |w_1| + |w_2|$.

Agora, escrevendo $T(n)$ de forma recursiva temos o seguinte:

$$\begin{aligned} 2T\left(\frac{n}{2}\right) + Cn &= 2\left(2T\left(\frac{n}{4}\right) + \frac{Cn}{2}\right) + Cn \\ &= 4T\left(\frac{n}{4}\right) + 2Cn \\ &= 2^2 T\left(\frac{n}{2^2}\right) + 2Cn \\ &= 2^r T\left(\frac{n}{2^r}\right) + rCn. \end{aligned}$$

Se $r \in \mathbb{R}^+$ tal que $\frac{n}{2^r} = 1$, então $r = \log_2 n$, ou seja $T(n) = nT(1) + \log_2(n)Cn \sim n \log_2(n)$. Logo $T(n) = \mathcal{O}(n \log(n))$. □

Corolário 4.2.9. *No grupo F de Thompson, o tempo para calcular a forma normal de uma palavra w é $\mathcal{O}(|w| \log |w|)$.*

4.2.2 Calculando a forma normal

Já temos estudado alguns algoritmos para calcular a forma seminormal de uma palavra no grupo F. Na presente seção vamos estudar um método para que a forma seminormal achada, usando os métodos acima estudados, também cumpra a propriedade *(NF2)*, e assim obter a forma normal de uma palavra.

O seguinte lema sugere um método para remover pares de geradores numa palavra, que contradizem *(NF2)*, i.e., x_i e x_i^{-1} pertencem a w, mas nem x_{i+1} nem x_{i+1}^{-1} pertencem a w.

Lema 4.2.10. *Sejam $w = x_{i_1} \cdots x_{i_s} x_{j_t}^{-1} \cdots x_{j_1}^{-1}$ uma forma seminormal e $(x_{i_a}, x_{j_b}^{-1})$ um par de geradores em w que contradiz (NF2), sendo a e b maximais com esta propriedade. Seja*

$$w' = x_{i_1} \cdots x_{i_{a-1}} \delta_{-1}(x_{i_{a+1}} \cdots x_{i_s} x_{j_t}^{-1} \cdots x_{j_{b+1}}) x_{j_{b-1}} x_{j_1}^{-1}.$$

Então w' é uma forma seminormal de w. Além disso, se $(x_{i_c}, x_{j_d}^{-1})$ é um par de geradores em w' que contradiz (NF2) então $c < a$ e $d < b$.

Demonstração. Sejam $w = x_{i_1} \cdots x_{i_s} x_{j_t}^{-1} \cdots x_{j_1}^{-1}$ uma forma seminormal e $(x_{i_a}, x_{j_b}^{-1})$ o par que contradiz *(NF2)*.

Pela definição de forma seminormal temos que os índices em $x_{i_{a+1}} \cdots x_{i_s} x_{j_t}^{-1} \cdots x_{j_{b+1}}^{-1}$ são maiores do que $i_a + 1$ pois pela hipótese x_{i_a+1} não aparece em w, isto implica que os índices em $\delta_{-1}(x_{i_{a+1}} \cdots x_{i_s} x_{j_t}^{-1} \cdots x_{j_{b+1}}^{-1})$ são maiores do que i_a. Agora, note que por construção temos que w' é uma forma seminormal.

Usando a primeira relação em $\mathcal{R}$ na ordem inversa, ou seja, $x_k x_{n+1}^{-1} \to x_n^{-1} x_k$ para $k < n$ temos que $w =_F w'$.

Para provar a segunda parte do lema seja $(x_{i_c}, x_{j_d}^{-1})$ o par de geradores em w' que contradiz *(NF2)* e suponha que $c > a$ ou $d > b$. Como $(x_{i_a}, x_{j_b}^{-1})$ são geradores que não cumprem *(NF2)* em w, então $(x_{i_a+\epsilon}, x_{j_b+\epsilon}^{-1})$ contradiz *(NF2)* em $\delta_\epsilon(w)$. As desigualdades $c > a$ e $d > b$ implicam que a e b não são maximais, então $c < a$ e $d < b$. □

Exemplo 4.2.11. *Considere a palavra em F*

$$w = x_3 x_5 x_6 x_8 x_{12} x_{16} x_{20}^{-1} x_{10}^{-1} x_8^{-1} x_6^{-1}$$

Neste caso temos que $a = 4$ e $b = 2$, pois o par $(x_{i_4}, x_{j_2}^{-1})$ contradiz (NF2).

Pelo lema 4.2.10 temos que

$$\begin{aligned} w' &= x_3 x_5 x_6 \delta_{-1}(x_{12} x_{16} x_{20}^{-1} x_{10}^{-1}) x_6^{-1} \\ &= x_3 x_5 x_6 x_{11} x_{15} x_{19}^{-1} x_9^{-1} x_6^{-1}. \end{aligned}$$

Na última palavra os geradores x_6 *e* x_6^{-1} *correpondem ao par* $(x_{i_3}, x_{j_1}^{-1})$ *e contradizem (NF2). Neste caso* $c = 3$ *e* $d = 1$.

Repetindo o processo obtemos:

$$\begin{aligned} w'' &= x_3x_5\delta_{-1}(x_{11}x_{15}x_{19}^{-1}x_9^{-1}) \\ &= x_3x_5x_{10}x_{14}x_{18}^{-1}x_8^{-1}. \end{aligned}$$

O resultado é a forma normal da palavra w.

O seguinte algoritmo está baseado no lema acima. A primeira parte do algoritmo encontra os pares que contradizem *(NF2)* em uma palavra w, a segunda parte aplica a função δ_ϵ ao segmento da palavra onde é preciso.

A característica do algoritmo 4 é que este não aplica a função δ_{-1} imediatamente quando o par $(x_{i_a}, x_{j_b}^{-1})$ é encontrado, o algoritmo guarda a informação dos índices que têm que ser trocados. Esta informação é acumulada em duas sequências (stacks): para as subpalavras positivas e negativas de w respectivamente.

Os tamanhos das sequências S_1 e S_2 são iguais aos comprimentos das palavras auxiliares w_1 e w_2, respectivamente. Além disso, x_a e x_b estão definidos se, e somente se, ϵ_1 e ϵ_2 estão definidos.

Algoritmo 4: Eliminação de pares que não cumprem a condição *(NF2)* da forma seminormal.

Entrada: A forma seminormal $u = x_{i_s} \cdots x_{i_1} x_{j_t}^{-1} \cdots x_{j_1}^{-1}$.
Signatura: $w = EraseBadPairs(u)$
Saída: Forma normal de u denotadas por w.
início
  $\delta = 0$, $\delta_1 = 0$, $\delta_2 = 0$, $w_1 = 0$, $u_1 = x_{i_s} \cdots x_{i_1}$, $u_2 = x_{j_t}^{-1} \cdots x_{j_1}^{-1}$ e S_1 e S_2 conjuntos vazios.
  A) Sejam $u_1 = x_{i_s} \cdots x_{i_1}$ e $u_2 = x_{j_t}^{-1} \cdots x_{j_1}^{-1}$.
  B) Seja x_a a menor letra de w_1, x_b a menor letra de w_2 e ϵ_i o último elemento que foi posto no conjunto S_i. Se algum desses elementos não existir, então a variável correspondente não está definida.
  1)**para** $s > 0$ *e* $t = 0$ *ou* $i_s > j_t$ **faça**
    a) Multiplique w_2 à esquerda por x_{i_s};
    b) Apague x_{i_s} de u_1 ;
    c) Coloque 0 em S_1;
    d) Ir a 5).
  fim
  continúa ...
fim

Algoritmo 4: Eliminação de pares que não cumprem a condição *(NF2)* da forma seminormal.

2)**para** $t > 0$ *e* $s = 0$ *ou* $j_t > i_s$ **faça**
- a) Multiplique w_2 à direita por $x_{j_t}^{-1}$;
- b) Apague $x_{j_t}^{-1}$ de u_2 ;
- c) Coloque 0 em S_2;
- d) Ir a 5).

fim

3)**para** $j_t = i_s$ *e os números* $a - \epsilon_1$ *e* $a - \epsilon_2$ *são distintos de* i_s *ou* $i_s + 1$ **faça**
- a) Apague x_{j_s} de u_1;
- b) Apague $x_{j_t}^{-1}$ de u_2 ;
- c) Se S_1 é não vazio, aumente o último elemento de S_1;
- d) Se S_2 é não vazio, aumente o último elemento de S_2;
- e) Ir a 5).

fim

4)**para** $i_s = j_t$ *e um dos dois elementos* $a - \epsilon_1$ *e* $a - \epsilon_2$ *são iguais a* i_s *ou* $i_s + 1$ **faça**
- a) Multiplique w_1 à esquerda por x_{i_s};
- b) Multiplique w_2 à direita por $x_{i_t}^{-1}$;
- c) Apague x_{i_s} de u_1 e $x_{i_t}^{-1}$ de u_2;
- e) Coloque 0 em S_1 e 0 em S_2;
- g) Ir a 5).

fim

5)**para** w_1 *ou* w_2 *distintas de vazio* **faça**
- ir a 1).

fim

para $w_1 \neq \emptyset$ **faça**
- 1) Coloque x_{i_1} como o primeiro elemento de w_1, ou seja, $w_1 = x_{i_1} w_1'$;
- 2) Tome o elemento maior $c \in S_1$ e faça $\delta_1 + c$;
- 3) Multiplique u_1 à direita por $x_{i_1 - \delta_1}$;
- 4) Apague x_{i_1} de w_1;

fim

para $w_2 \neq \emptyset$ **faça**
- 1) Coloque $x_{j_1}^{-1}$ como o último elemento de w_2, ou seja, $w_2 = w_2' x_{j_1}^{-1}$;
- 2) Tome o elemento maior $c \in S_2$ e faça $\delta_2 + c$;
- 3) Multiplique u_2 à esquerda por $x_{j_1 - \delta_2}^{-1}$;
- 4) Apague $x_{j_1}^{-1}$ de w_2.

fim

retorna $u_1 u_2$

fim

O resultado do algoritmo anterior é a forma normal w da forma seminormal u.

4.3 Descobrindo as chaves secretas do protocolo

Já temos estudado o processo de geração de chaves secretas que propõem Shpilrain e Ushakov em [12] usando como plataforma o grupo F de Thompson, nesta seção vamos estudar o ataque proposto em [8]; neste artigo é provado que um espião pode obter a chave privada de uma das duas partes do protocolo de criptografia baseado no grupo F.

Vamos descrever a forma em que o espião Eve pode descobrir uma das duas chaves privadas: Eve conhece w, u_1 e u_2, onde $u_1 = a_1wb_1$ e $u_2 = b_2wa_2$, $a_1, a_2 \in A_s$, $b_1, b_2 \in B_s$ $w \in F$.

$$\text{Alice } \overrightarrow{u_2} \text{ Bob.}$$
$$\text{Alice } \overleftarrow{u_1} \text{ Bob.}$$

Eve escolhe a chave a ser descoberta dependendo se o gráfico de w está acima ou abaixo do ponto (φ_s, φ_s), lembrando que $\varphi_s = 1 - \dfrac{1}{2^{s+1}}$.

4.3.1 Descobrindo a chave privada de Bob

Suponha que $w(\varphi_s) \leq \varphi_s$. Então $w(t) \leq \varphi_s$ para todo $t \in [0, \varphi_s]$.

Temos também a seguinte igualdade:

$$u_2(t) = b_2wa_2(t).$$

Note que $a_2(t) < \varphi_s$ para todo $t \in [0, \varphi_s]$, então $wa_2(t) \leq \varphi_s$. Logo $b_2(wa_2(t)) = wa_2(t)$ para todo $t \in [0, \varphi_s]$, pois $b_2 \in B_s$.

Logo, Eve pode aplicar w^{-1} à esquerda:

$$w^{-1}(u_2(t)) = a_2(t) \text{ para todo } t \in [0, \varphi_s].$$

Como $w^{-1}u_2$ fixa o ponto φ_s e, restrito a $[0, \varphi_s]$ está em A_s. Sabemos que B_s é identificado com $PL_2[\varphi_s, 1]$, significa que $w^{-1}u_2$, restrito a $[\varphi_s, 1]$ é um elemento de B_s, daí temos que

$$w^{-1}u_2 \in A_sB_s \tag{4.1}$$

e

$$a_2(t) = \begin{cases} w^{-1}u_2(t) & \text{se} \quad t \in [0, \varphi_s] \\ \\ t & \text{se} \quad t \in [\varphi_s, 1] \end{cases}$$

Agora Eve conhece w, a_2 e u_2, assim é possível achar $b_2 = u_2a_2^{-1}w^{-1}$, logo Eve agora conhece a chave secreta de Bob.

4.3.2 Descobrindo a chave privada de Alice

Se $w(\varphi_s) > \varphi_s$, então $w^{-1}(t) < \varphi_s$ para todo $t \in [0, \varphi_s]$. Como $a_1(t) = t$ e $wb_1(t) > \varphi_s$ para todo $t \in [\varphi_s, 1]$, temos que:

$$w^{-1}u_1(t) = w^{-1}a_1wb_1(t) = b_1(t), \text{ para todo } t \in [\varphi_s, 1].$$

Logo

$$b_1(t) = \begin{cases} t & \text{se} \quad t \in [0, \varphi_s] \\ \\ w^{-1}u_1(t) & \text{se} \quad t \in [\varphi_s, 1] \end{cases}$$

Agora Eve conhece w, b_1 e u_1, assim é possível achar $a_1 = u_1b_1^{-1}w^{-1}$, logo Eve agora conhece a chave secreta de Alice.

4.4 Ataque das chaves secretas usando a transitividade de As e Bs

Pela seção anterior sabemos que se $w(\varphi_s) \leq \varphi_s$ é possível achar a chave secreta do Bob, e se $w(\varphi_s) \geq \varphi_s$ é descoberta a chave secreta de Alice. Nesta seção vamos ver que quando mudamos as condições para $w(\varphi_s)$ também é possível achar as chaves secretas.

Para isto vamos primeiro provar alguns fatos:

Definição 4.4.1. Seja $[c, d] \subseteq \mathbb{R}$, definimos $Homeo([c, d])$ ao conjunto dos homeomorfismos f definidos em $\mathbb{R}$ tais que $f(x) = x$, para todo $x \notin [c, d]$.

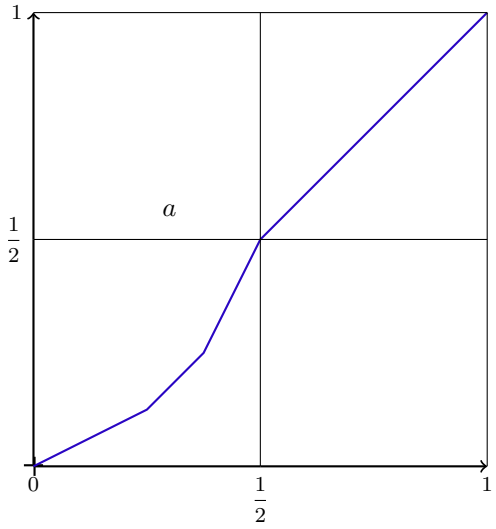
Lema 4.4.2. *Sejam $[a, b] \subseteq [c, d]$ intervalos em $\mathbb{R}$ e $f \in Homeo([c, d])$, então $f^{-1}Homeo([a, b])f = Homeo([f^{-1}(a), f^{-1}(b)])$.*

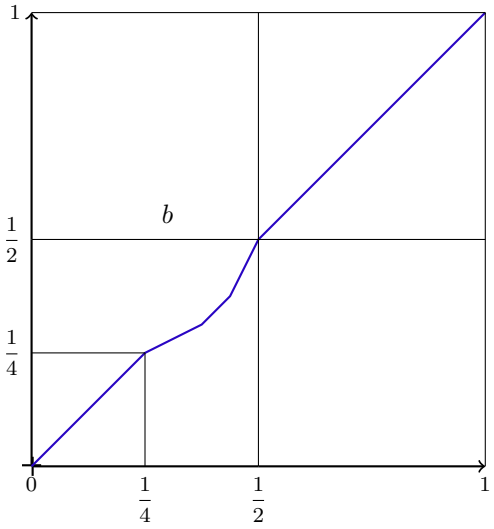
Demonstração. Seja $g \in Homeo([a, b])$. Se $t \notin [f^{-1}(a), f^{-1}(b)]$, então $f(t) \notin [a, b]$, logo $g^{-1}f(t) = f(t)$ para todo $t \notin [f^{-1}(a), f^{-1}(b)]$. Isto quer dizer que $f^{-1}gf(t) = t$ para todo $t \notin [f^{-1}(a), f^{-1}(b)]$, então $f^{-1}gf \in Homeo([f^{-1}(a), f^{-1}(b)])$, ou seja $f^{-1}Homeo([a, b])f \subseteq Homeo([f^{-1}(a), f^{-1}(b)])$.

Agora, suponha que $g \in Homeo([f^{-1}(a), f^{-1}(b)])$. Se $t \notin [a, b]$, então $f^{-1}(t) \notin [f^{-1}(a), f^{-1}(b)]$, logo $g^{-1}f^{-1}(t) = f^{-1}(t)$ para todo $t \notin [a, b]$. Isto quer dizer que $fgf^{-1}(t) = t$ para todo $t \notin [a, b]$, então $fgf^{-1} \in Homeo([a, b])$, ou seja $fHomeo([f^{-1}(a), f^{-1}(b)])f^{-1} \subseteq Homeo([a, b])$, que significa que $Homeo([f^{-1}(a), f^{-1}(b)]) \subseteq f^{-1}Homeo([a, b])f$, como desejamos. □

Lema 4.4.3. $A_2 = PL_2\left(\left[0, \frac{7}{8}\right]\right)$.

Demonstração. Sejam $a = x_0^2 x_1^{-1} x_0^{-1}$ e $b = x_0 x_1^2 x_2^{-1} x_1^{-1} x_0^{-1}$ geradores de $PL_2\left(\left[0, \frac{1}{2}\right]\right)$ mostrados na seguinte figura.





Fazendo conjugação de $PL_2\left(\left[0, \frac{1}{2}\right]\right)$ com x_0^{-2}, ou seja,

$$x_0^{-2} PL_2\left(\left[0, \frac{1}{2}\right]\right) x_0^2 = \langle x_0^{-2} a x_0^2, x_0^{-2} b x_0^2 \rangle.$$

Como $\left[0, \frac{1}{2}\right] \subset \left[0, \frac{7}{8}\right]$ segue se do lema 4.4.2 que $PL_2\left(\left[0, \frac{7}{8}\right]\right) = x_0^{-2} PL_2\left(\left[0, \frac{1}{2}\right]\right) x_0^2$.

Além disso, aplicando as reduções da forma normal e lembrando a definição de x_2, temos o seguinte:

$$x_0^{-2} a x_0^2 = x_0 x_2^{-1} \in A_2 \text{ e}$$

$$\begin{aligned} x_0^{-2} b x_0^2 &= x_0^{-1} x_1^2 x_2^{-1} x_1^{-1} x_0 \\ &= x_0^{-1} x_1^2 x_2^{-1} x_1^{-1} x_0 \\ &= x_0^{-1} x_1 x_1 x_0^{-1} x_1^{-1} x_0 x_1^{-1} x_0 \\ &= (x_1^{-1} x_0)^{-1} (x_0 x_1^{-1})^{-1} (x_0 x_2^{-1})(x_0 x_2^{-1}) \\ &= (x_0 x_2^{-1})^{-1} (x_0 x_1^{-1})^{-1} (x_0 x_2^{-1})(x_0 x_2^{-1}) \in A_2. \end{aligned}$$

Assim, $PL_2\left(\left[0, \frac{7}{8}\right]\right) \subseteq A_2$. A outra inclusão é imediata. □

Teorema 4.4.4. *$A_s = PL_2([0, \varphi_s])$, para cada $s \geq 0$.*

Demonstração. Afirmação 1: $x_0^{-1}(\varphi_s) = \varphi_{s+1}$: De fato, do gráfico do homeomorfismo x_0^{-1} podemos ver que a derivada no intervalo $\left[\frac{1}{2}, 1\right]$ é $\frac{1}{2}$, então a equação da reta nesse intervalo, que passa pelo ponto $(1, 1)$, fica da seguinte maneira:

$$x_0^{-1}(t) = \frac{1}{2}(t-1) + 1,$$

então $x_0^{-1}(\varphi_s) = 1 - \dfrac{1}{2^{s+2}} = \varphi_{s+1}$.
Pelo lema 4.4.2 temos que

$$x_0^{-1} PL_2\left([0,\varphi_s]\right) x_0 = PL_2([x_0^{-1}(0), x_0^{-1}(\varphi_s)]) = PL_2([0,\varphi_{s+1}])$$

para todo $s \geq 0$.

Agora observemos que multiplicando $2-s$ vezes à direita e à esquerda por x_0 e x_0^{-1}, respectivamente, obtemos o seguinte:

$$\begin{aligned} PL_2([0,\varphi_s]) &= x_0^{2-s} PL_2\left([0,\varphi_2]\right) x_0^{s-2} \\ &= x_0^{2-s} A_2 x_0^{s-2} \subseteq A_s. \end{aligned}$$

A última relação de inclusão da fórmula anterior segue do seguinte raciocínio:lembrando que os geradores de A_s, $x_0 x_k^{-1}$, têm derivada 1 no intervalo $[0,\varphi_s]$, e além disso

$$\begin{aligned} x_0^{2-s} x_0 x_1^{-1} x_0^{s-2} &= x_0 x_0^{2-s} x_1^{-1} x_0^{s-2} \\ &= x_0 x_0^{2-s} x_0^{s-2} x_{s-1}^{-1} \\ &= x_0 x_{s-1}^{-1} \in A_s. \end{aligned}$$

Analogamente, também $x_0^{2-s} x_0 x_2^{-1} x_0^{s-2} = x_0 x_s^{-1} \in A_s$. Dos fatos anteriores, temos que

$$PL_2([0,\varphi_s]) = x_0^{2-s} A_2 x_0^{s-2} \subseteq A_s \subseteq PL_2([0,\varphi_s]).$$

A última inclusão é verdadeira porque pelo lema 4.1.2 todos os geradores de A_s pertencem a $PL_2([0,\varphi_s])$. □

Corolário 4.4.5. $x_0^{-1} A_s x_0 = A_{s+1}$

Demonstração. Segue se imediatamente do teorema anterior. □

Corolário 4.4.6. $A_s \cong B_s \cong F$.

Demonstração. O fato de que $A_s \cong F$ segue imediatamente do Teorema 4.4.4 e o fato de que $PL_2([0,\varphi_s])$ sempre é isomorfo a F. O fato de que $B_s \cong F$ segue do fato que já sabemos que B_s é gerado pelos $x_{s+1}, \ldots, x_{2s}$ que geram uma copia de F em $[\varphi_s, 1]$. □

Lema 4.4.7. *Seja H_s o conjunto dos elementos de F cuja forma normal é do tipo:*

$$x_{i_1} \cdots x_{i_m} x_{j_m}^{-1} \cdots x_{j_1}^{-1},$$

onde $i_k - k < s$ e $j_k - k < s$ para todo $k = 1, \cdots, m$. Então $A_s = H_s$.

Demonstração. O conjunto H_s contém o elemento identidade expressado como $x_{i_k} x_{i_k}^{-1}$, com $i_k - k < s$. Também, H_s é fechado com a multiplicação pois dadas duas palavras:

$$u = x_{i_1} \cdots x_{i_m} x_{j_m}^{-1} \cdots x_{j_1}^{-1}$$
$$v = x_{p_1} \cdots x_{p_s} x_{q_s}^{-1} \cdots x_{q_1}^{-1},$$

a forma normal do produto terá igual número de letras positivas e negativas, pois aplicando as relações de F para calcular a forma seminormal de uv obtemos o seguinte:

$$uv = x_{i_1} \cdots x_{i_m} x_{p_1+m} \cdots x_{p_s+m} x_{j_m}^{-1} \cdots x_{j_1}^{-1} x_{q_s}^{-1} \cdots x_{q_1}^{-1}.$$

Sabemos que $p_r - r < s$, então $p_r + m - (m + r) < s$; aplicando indução sobre $m + s$ podemos ver que a palavra obtida do produto uv satisfaz a condiçaõ dos índices: $i_k - k < s$ e $j_k - k < s$ para todo $k = 1, \cdots, m + s$. Até agora temos provado que H_s é um subgrupo. Falta provar que $A_s = H_s$: é claro que $A_s \subseteq H_s$ porque os geradores de A_s têm a forma dos elementos de H_s, e também sabemos que $A_s \subseteq A_t$ sempre que $k \leq t$.

Note que pela definição de A_s temos que $A_s x_{i_k} = A_s x_0$ para todo $i_k = 1, \cdots, s$, denotemos $H = A_2$ e note que $Hx_{i_1}x_{i_2} = Hx_0x_0$ se, e somente se, $x_{i_2}x_0^{-1} \in x_0^{-1}Hx_0 = A_3$, mas $x_{i_2}x_0^{-1} = (x_0x_{i_2}^{-1})^{-1} \in A_3$ pois $i_2 - 2 < 2$, logo $i_2 < 4$.

Analogamente, temos que $Hx_{i_1}x_{i_2}x_{i_3} = Hx_0x_0x_0$ se, e somente se, $x_{i_3}x_0^{-1} \in x_0^{-2}Hx_0^2 = A_4$, isto, de fato, é verdadeiro pois $x_{i_3}x_0^{-1} = (x_0x_{i_3}^{-1})^{-1} \in A_4$. Por indução obtemos que $Hx_{i_1}x_{i_2} \cdots x_{i_m} = Hx_0^m$ e também que $Hx_{j_1}x_{j_2} \cdots x_{j_m} = Hx_0^m$, então
$Hx_{i_1}x_{i_2} \cdots x_{i_m}x_{j_m}^{-1} \cdots x_{j_1}^{-1} = H$, isto implica que $x_{i_1}x_{i_2} \cdots x_{i_m}x_{j_m}^{-1} \cdots x_{j_1}^{-1} \in H$.

O cálculo que fizemos prova que $H_2 = A_2$, por causa dos índices. Adaptando esta prova e tomando $H = A_s$ pode se demonstrar que $H_s = A_s$. □

Observação 4.4.8. No artigo original a restrição dos índices do lema anterior é dada para $k = 1, \cdots, s$, mas é possível ver que com esta restrição a afirmação do lema é falsa, ou seja, $A_s \neq H_s$. Para mostrar o fato anterior apresentaremos um exemplo, mas primeiro precisamos provar a seguinte afirmação:

Afirmação: $A_2x_2x_2 = A_2x_0x_3 = A_2x_0x_0$.
De fato:

$$\begin{aligned} A_2x_2x_2 = A_2x_0x_3 = A_2x_0x_0 &\iff A_2x_0x_2 = A_2x_0x_3 = A_2x_0x_0 \\ &\iff A_2x_0x_2x_3^{-1}x_0^{-1} = A_2 = A_2x_0x_3x_0^{-1}x_0^{-1} \\ &\iff x_0x_2x_3^{-1}x_0^{-1}, x_0x_3x_0^{-1}x_0^{-1} \in A_2, \\ &\iff x_2x_3^{-1} \in x_0^{-1}A_2x_0 \quad e \quad x_3x_0^{-1} \in x_0^{-1}A_2x_0 = A_3. \end{aligned}$$

mas $x_2x_3^{-1} = x_2x_0^{-1}x_0x_3^{-1} \in A_3$ e $x_3x_0^{-1} \in A_3$.

Considere a palavra $w = x_2x_2x_7^{-1}x_3^{-1}x_0^{-1} \in H_2$. Suponha que $w \in A_2$.

$$\begin{aligned} w \in A_2 &\iff A_2x_2x_2x_3 = A_2x_0x_3x_7 \\ &\iff A_2x_0x_0x_3 = A_2x_0x_0x_7, pois \\ &\iff x_0^2x_3x_7^{-1}x_0^{-2} \in A_2 \\ &\iff x_3x_7^{-1} \in x_0^{-2}A_2x_0^2 = A_4, \end{aligned}$$

onde $x_3x_7^{-1} = x_3x_0^{-1}x_0^1x_7^{-1} = x_0^{-1}x_1x_0^{-4}x_1^{-1}x_0^6$, a última igualdade é consequência da igualdade mostrada em 2.1. Pelo teorema 4.4.4 temos que $A_4 = PL([0, \varphi_4])$, onde $\varphi_4 = \dfrac{31}{32}$. Considere $p = \dfrac{63}{64} > \varphi_4$ e fazendo os cálculos pode se verificar que $x_3x_7^{-1}(p) \neq p$, então $x_3x_7^{-1} \notin A_4$.

Na continuação é mostrada a forma na que Eve descobre as chaves secretas, desde a teoría combinatorial:

Suponha que Eve conhece os elementos w, u_1 e u_2. Denotemos $z_1 = u_1w^{-1}$ e $z_2 = w^{-1}u_2$.

(1) Eve escreve a forma normal de z_1 e z_2.

(2) Pelo exposto acima em 4.1 temos que ou $z_1 \in A_sB_s$ ou $z_2 \in A_sB_s$.
Eve pode saber qual elemento pertence ao conjunto A_sB_s, pois os elementos em A_s tem forma normal da forma $x_{i_1} \cdots x_{i_m}x_{j_m}^{-1} \cdots x_{j_1}^{-1}$.

(3) Sabendo que $z_i \in A_sB_s$, Eve calcula a parte de z_i que pertence ao subgrupo A_s, denotado por a_{z_i} ($u_i = a_{z_i}wb_{z_i}$, $a_{z_i} \in A_s, b_{z_i} \in B_s$).

(4) Se $i = 1$, ela calcula $b_{z_1} := w^{-1}a_{z_1}^{-1}u_1$.
Se $i = 2$, ela calcula $b_{z_2} := u_2a_{z_2}^{-1}w^{-1}$.

(5) Finalmente, Eve calcula a chave secreta compartilhada, $K = a_1b_2wa_2b_1$, usando u_1, u_2, a_{z_i} e b_{z_i}.

Observação 4.4.9. Observamos que cada elemento no subgrupo A_sB_s pode-se escrever de maneira única como um elemento da forma a_1b_1, onde $a_1 \in A_s$ e $b_1 \in B_s$. Como os elementos do subgrupo A_s correspondem ao homeomorfismo identidade no intervalo $[\varphi_s, 1]$, enquanto os elementos do subgrupo B_s correspondem ao homeomorfismo identidade em $[0, \varphi_s]$, então temos que $A_s \cap B_s = 1$, assim, se $a_1b_1 = a_2b_2$ então $a_2^{-1}a_1 = b_2b_1^{-1}$. Note que no lado esquerdo da igualdade anterior temos um elemento de A_s e no lado direito há um elemento de B_s, ou seja, esses elementos estão na interseção de A_s e B_s, daí temos que $a_1 = a_2$ e $b_1 = b_2$.

Exemplo 4.4.10. A) *Considere* $s = 3$ e $a = x_1x_3x_4x_5^{-1}x_2^{-1}x_1^{-1} \in A_3$.
Então $B_s = \langle x_4, x_5, x_6 \rangle$. *Seja* $b = x_5x_6^{-1}x_5x_4$.
Seja $z = x_1x_2x_8x_5^{-1}x_4^{-1}x_3^{-1}x_1^{-1}$, *queremos saber se* $z \in A_3B_3$.

(1) *Achar o menor índice r em z tal que $i_{r+1} - (r+1) > s$ ou $j_{r+1} - (r+1) > s$. Neste caso $r = 2$.*

(2) *Remover as primeiras e últimas duas letras de z e diminuir em 2 os índices restantes, ou seja,*

$$x_1x_2x_8x_5^{-1}x_4^{-1}x_3^{-1}x_1^{-1} \longrightarrow x_6x_3^{-1}x_2^{-1}.$$

Note que a palavra obtida não pertence ao subgrupo B_3, logo $z \notin A_3B_3$.

B) *Seja $z = x_2x_4x_6x_7^{-1}x_4^{-1}x_3^{-1}$ Repetindo o processo do exemplo anterior obtemos a palavra $x_4x_5^{-1}$, e note que os índices desta palavra pertencem ao conjunto $\{s{+}1, \cdots 2s\}$, ou seja, $x_4x_5^{-1} \in B_s$.*

Note que $z = x_2x_4\delta_2(x_4x_5^{-1})x_4^{-1}x_3^{-1}$, e considere $a = x_2x_4x_4^{-1}x_3^{-1} = x_2x_3^{-1}$ e *$b = x_4x_5^{-1}$.*

Agora, usando as relações $\mathcal{R}$ da forma normal temos o seguinte:

$$ab = x_2x_3^{-1}x_4x_5^{-1} = x_2x_5x_3^{-1}x_5^{-1} = x_2x_5x_6^{-1}x_3^{-1}.$$

Além disso, é possível verificar que a forma normal da palavra z coincide com o resultado imediatamente anterior, usando a relação

$$x_i^{-1}x_k \to x_{k+1}x_i^{-1}$$

equivalente a uma das relações de $\mathcal{R}$, só que no sentido inverso.

De fato,

$$\begin{aligned} z &= x_2x_4x_6x_7^{-1}x_4^{-1}x_3^{-1} \\ &= x_2x_4x_6x_4^{-1}x_6^{-1}x_3^{-1} \\ &= x_2x_4x_4^{-1}x_5x_6^{-1}x_3^{-1} \\ &= x_2x_5x_6^{-1}x_3^{-1}. \end{aligned}$$

Observação 4.4.11. O processo feito acima é equivalente ao método proposto no lema 4.2.10, onde vemos que (x_4, x_4^{-1}) é um par que contradiz a propriedade da forma normal *(NF2)*, sendo $a = b = 2$, logo aplicando o lema a forma normal é dada por

$$x_2\delta_{-1}(x_6x_7^{-1})x_3^{-1}.$$

Na continuação temos o lema 2.5 de [5] que ajuda a provar alguns resultados que precisamos:

Lema 4.4.12. *Suponha que $I_1, \cdots, I_k \subseteq [0,1]$ é uma família de intervalos compactos, $I_i = [a_i, b_i]$, com $b_i \leq a_{i+1}$ para todo $i = 1, \cdots k$, e $a_i, b_i \in \mathbb{Z}\left[\frac{1}{2}\right]$.*

Sejam $J_1, \cdots J_k \subseteq [0,1]$ com $J_i = [c_i, d_i]$, outra família de intervalos com a mesma propriedade da família acima. Suponha que

$$g_i : I_i \longrightarrow J_i$$

é uma função linear por partes com um número finito de pontos onde ela é não derivável, e satisfaz a definição dos elementos do grupo F, então existe $\widehat{g} \in PL_2(I)$ tal que $\widehat{g}|_{I_i} = g_i$.

Demonstração. Considere $0 < a_1 < b_1 < \cdots < a_k < b_k < 1$ e $0 < c_1 < d_1 < \cdots < c_k < d_k < 1$ duas partições de $[0,1]$ com a mesma quantidade de pontos, pelo lema 2.2.12 existe $f \in F$ tal que $f(a_i) = c_i$ e $f(b_i) = d_i$. Defina

$$\widehat{g} = \begin{cases} f(t) & \text{se} \quad t \notin \bigcup_{i=1}^{n} I_i \\ \\ g_i(t) & \text{se} \quad I_i \end{cases}$$

□

Corolário 4.4.13. *(Transitividade de A_s)*
Sejam t_1, $t_2 \in \mathbb{Z}\left[\frac{1}{2}\right] \cap [0, \varphi_s]$, existe $a \in A_s$ tal que $a(t_1) = t_2$.

Demonstração. Como $A_s = PL_2([0, \varphi_s])$ pelo lema acima temos que existe $a \in A_s$ tal que $a(t_1) = t_2$, pois basta considerar as seguintes partições:

$$0 < t_1 < \varphi_s$$
$$0 < t_2 < \varphi_s$$

□

Corolário 4.4.14. *(Extendibilidade de A_s)*
Sejam $t_0 \in \mathbb{Z}\left[\frac{1}{2}\right] \cap [0, \varphi_s]$ e $\tilde{a}(t) = a|_{[0,t_0]}$ para algum elemento $a \in A_s$, suponha que a função $\tilde{a}$ é conhecida, então existe $a_\sigma \in A_s$ tal que $a_\sigma(t) = \tilde{a}(t)$ para todo $t \in [0, \varphi_s]$.

Demonstração. É consequência do lema 4.4.12.

□

Observação 4.4.15. Os resultados estudados acima também são verdadeiros no intervalo $[\varphi_s, 1]$ e elementos de B_s, pois $B_s \cong F$ e os elementos neste subgrupo são a função identidade no intervalo $[0, \varphi_s]$.

Agora vamos mostrar o processo para descobrir as chaves secretas de Alice e Bob, trocando as condições da palavra w sobre φ_s estudadas na seção 3.3:

(i) Se $w(\varphi_s) \leq \varphi_s$, lembrando que $u_1 = a_1 w b_1$, temos que $u_1(t) = a_1 w(t)$ para todo $t \in [0, \varphi_s]$.
Então $a_1(t) = u_1 w^{-1}(t)$ para todo $t \in [0, \varphi_s]$ e como $w(\varphi_s) \leq \varphi_s$, em particular temos que $a_1(t) = u_1 w^{-1}(t)$ para todo $t \in [0, w(\varphi_s)]$.
Considere $a_\sigma \in A_s$ tal que $a_\sigma = a_1$ no intervalo $[0, w(\varphi_s)]$ como no corolário 4.4.14.

Definamos $b_\sigma = w^{-1} a_\sigma^{-1} u_1$. Então

$$b_\sigma(t) = w^{-1} a_\sigma^{-1} a_1 w(t) = w^{-1} w(t) = t, \text{ para todo } t \in [0, \varphi_s].$$

Daí $b_\sigma \in B_s$ e $a_\sigma w b_\sigma = u_1$.
Agora é possível achar a chave secreta pois como $u_2 = b_2 w a_2$, então

$$\begin{aligned} a_\sigma u_2 b_\sigma &= a_\sigma b_2 w a_2 b_\sigma \\ &= b_2 a_\sigma w b_\sigma a_2 \\ &= b_2 u_1 a_2 \\ &= K \end{aligned}$$

(ii) Se $w(\varphi_s) > \varphi_s$ a espião Eve considera $u_2^{-1} = a_2^{-1} w^{-1} b_2^{-1}$ onde $a_2^{-1} \in A_s$, logo

$$u_2^{-1}(t) = w^{-1} b_2^{-1}(t) \text{ para todo } t \in [\varphi_s, 1].$$

Daí temos que $b_2^{-1}(t) = w u_2^{-1}(t)$ para todo $t \in [w(\varphi_s), 1]$. Pelo corolário 4.4.14 existe $b_\sigma \in B_s$ tal que $b_\sigma(t) = b_2(t)$ para todo $t \in [\varphi_s, 1]$.
Definamos $a_\sigma^{-1} = b_\sigma w u_2^{-1}$. Então

$$a_\sigma^{-1}(t) = b_\sigma w(w^{-1} b_2^{-1}(t)) = t \text{ para todo } t \in [\varphi_s, 1].$$

Daí $a_\sigma^{-1} \in A_s$ e além disso $a_\sigma^{-1} w^{-1} b_\sigma^{-1} = u_2^{-1}$ logo $b_\sigma w a_\sigma = b_2 w a_2$.
Neste caso também é possível achar a chave secreta compartilhada entre Alice e Bob.

5 Alguns resultados experimentais no grupo F

Neste capítulo queremos mostrar alguns resultados experimentais feitos com o grupo F de Thompson trabalhados nos artigos [10] e [11], nos quais é estudada a criptoanálise baseada no comprimento (length-based cryptanalysis) e a criptoanálise usando funções distância entre um elemento e um subgrupo.

5.1 A criptoanálise baseada no comprimento

A criptoanálise baseada no comprimento é um método usado para atacar protocolos, que usualmente tem menor probabilidade de sucesso comparado com ataques mais especializados, mas tem a vantagem de ser mais genérico no sentido que se um grupo tem uma boa funcão comprimento, então o ataque ao protocolo baseado nesse grupo tem uma probabilidade de succeso não nula.

O principal problema do algoritmo baseado no comprimento é que precisa que o grupo usado no criptosistema tenha poucas relações, ou seja, esteja próximo de ser um grupo livre. O anterior não é o caso do grupo F de Thompson. Em [12] Shpilrain e Ushakov mostram que este ataque é uma falha absoluta no protocolo proposto por eles.

Em [11] Ruinskiy, Shamir e Tsaban mostram algumas melhoras ao algoritmo usado por Shpilrain e Ushakov.

Definição 5.1.1. Dado G um grupo, definimos a *função comprimento* como sendo uma aplicação

$$L : G \longrightarrow \mathbb{R}^+ \cup \{0\}$$

tal que

(i) $L(e) = 0$.

(ii) $L(g^{-1}) = L(g)$ para todo $g \in G$.

(iii) $L(g_1 g_2) \leq L(g_1) + L(g_2)$ para todos $g_1, g_2 \in G$.

Exemplo 5.1.2. *Considere um grupo arbitrário apresentado por $G = \langle X \mid R \rangle$. Neste caso, um exemplo de função comprimento é $L(g) = |w|$, onde w é a palavra mais curta no grupo livre $F(X)$ que representa o elemento g em G.*

Na continuação vamos mostrar o ataque baseado no comprimento:

Seja G um grupo finitamente gerado e seja $S_G = \{g_1^{\pm 1}, g_2^{\pm 1}, \cdots, g_k^{\pm 1}\}$ um conjunto de geradores de G. Seja $x \in G$ representado como produto de geradores de G, $x = x_1 x_2 \cdots x_n$, com $x_i \in S_G$.

Considere $w \in G$ escolhido de forma independente do elemento x, sejam x e w desconhecidos e $z = xw$ um elemento conhecido que pertence a G, ou seja, conhecemos o elemento z escrito como produto de geradores, mas não sabemos quantos dos primeiros x_i formam x e quantos deles formam w.

Suponha que existe uma função comprimento L sobre os elementos de G tal que

$$L(x_1^{-1} z) < L(z) < L(x_j z),$$

para cada $x_j \neq x_1^{-1}$. A ideia é tentar recuperar x procurando os geradores que compõem ele. Para isto, em [11] é proposto o seguinte algoritmo, que é um ataque feito pelo inimigo para descobrir a chave secreta da Alice. Usaremos a notação $a \longrightarrow b$ para dizer que b substitui a no passo seguinte do algoritmo (no caso $1 \longrightarrow j$, j é um índice, e significa que o primeiro passo do algoritmo inicia com ele).

Algoritmo 5: Ataque baseado no comprimento.

Entrada: $S_G = \{g_1^{\pm 1}, g_2^{\pm 1}, \cdots, g_k^{\pm 1}\}$

início

Seja $1 \longrightarrow j$ e $z \longrightarrow y$

para $g \in S_G$ **faça**

$g^{-1} y$.

fim

2) Considere o elemento $h_1 \in S_G$ que minimize $L(h_1^{-1} y)$.

O elemento h_1 não precisa ser único, nesse caso escolha o elemento h_1 aleatoriamente. O anterior quer dizer que fazendo $h_1^{-1} y$ eliminamos um gerador no produto xz, ou seja, é provável que $h_1 \in \{x_1, \cdots, x_n\}$.

para $j = n$ **faça**

Terminou.

fim

para $j \neq n$ **faça**

$h_1 \longrightarrow h_j$, $j \longrightarrow j + 1$ e $y \longrightarrow h_1^{-1} y$.

fim

Agora y tem comprimento menor com respeito ao início do algoritmo. Fazendo o produto dos elementos de S_G pela nova palavra y, achamos outro elemento $h_2 \in S_G$ que reduz $h_1^{-1} y$, e então enviamos $h_2 \longrightarrow h_j$.

fim

Se L é uma boa função comprimento é provável que no passo 3 tenhamos $h_1 = x_1$, daí quando $j = n$ o algoritmo termina pois como $x = x_1 x_2 \cdots x_n$ e h_j contém todos os geradores

que eliminam letras de y, então existe uma probabilidade não nula de que $x = h_1 h_2 \cdots h_n$.

Em [12] é usado o ataque baseado no comprimento aplicado ao grupo F de Thompson, usando o seguinte algoritmo:

Algoritmo 6: Ataque baseado no comprimento no grupo F.

Entrada: A palavra pública inicial w e a palavra w' enviada por Alice

Saída: Um par de palavras $x_1 \in S_A$, $x_2 \in S_B$ tal que $w' = x_1 w x_2$.

início

$S_w = \{w\}$, $S_{w'} = \{w'\}$, $M_w = \emptyset$, $M_{w'} = \emptyset$

A) Encontre a menor palavra $u \in S_w$ tal que $u \notin M_w$.

B) Faça o produto de u pelos elementos de $S_A^{\pm 1}$ à esquerda e pelos elementos de $S_B^{\pm 1}$ à direita. Adicione o resultado ao conjunto S_w.

C) Adicione u ao conjunto M_w.

D) Realize os passos A-C trocando S_w por $S_{w'}$ e M_w por $M_{w'}$.

E) Se $S_w \cap S_{w'} = \emptyset$ então volte até o passo A .

F) Se existe $\overline{w} \in S_w \cap S_{w'}$ então encontre o caminho, ou seja, uma cadeia de letras em S_w, que leva da palavra w até a palavra $\overline{w}$ e, analogamente, o caminho em $S_{w'}$ desde $\overline{w}$ até w'.

Concatene os caminhos $\cdot$

retorna *A justaposição dos caminhos.*

fim

O algoritmo anterior foi implementado por Shpilrain e Ushakov, e em [12] eles concluem que depois de fazer experimentos nenhum programa deu resultados favoráveis.

5.2 Funções distância a um subgrupo

Nesta seção serão mostradas algumas funções distância dos subgrupos A_s e B_s no grupo F de Thompson definidos em 4.1.1. Basicamente, quando avaliamos uma função distância f em um elemento g de um grupo G, queremos deduzir se o elemento g pertence ou não a um subgrupo dado de G. A motivação para estabelecer um ataque baseado em funções distância é porque vários protocolos de criptografia de chave pública trabalham sobre pares de subgrupos, tal que a segurança do protocolo depende da habilidade que tenha o inimigo para encontrar um gerador nesses subgrupos. As distâncias definidas nesta seção serão usadas na implementação do algoritmo estudado na seção anterior.

Lembremos que o problema de decomposição aplicado ao protocolo proposto por Shpilrain e Ushakov consiste em achar elementos $a' \in A_s$ e $b' \in B_s$ tais que $a' z b' = a_1 z b_1$,

onde $u_1 = a_1 z b_1$ é um elemento público enviado por Alice ao Bob. A ideia do espião é encontrar a chave privada compartilhada.

Parte 1 do ataque: Para um elemento $a' \in A_s$ é possível calcular seu complemento

$$b' = z^{-1}(a')^{-1}u_1 = z^{-1}(a')^{-1}a_1 z b_1.$$

Então temos que $a'zb' = a_1 z b_1$. O par a' e b' é uma solução do problema se, e somente se, $b' \in B_s$.

Parte 2 do ataque: A chave pública enviada por Bob é $u_2 = b_2 z a_2$. Suponha que foram encontrados os elementos a' e b', então

$$\begin{aligned} a'u_2b' &= a'b_2za_2b' \\ &= b_2a'zb'a_2 \\ &= b_2u_1a_2 \\ &= K, \end{aligned}$$

onde K é a chave secreta compartilhada entre Alice e Bob. O espião pode quebrar o protocolo encontrando a decomposição certa de $u_2 = b_2 z a_2$ de forma análoga.

Pelo fato anterior vamos introduzir o seguinte conceito:

Definição 5.2.1. Seja G um grupo e $H \leq G$ um subgrupo. Uma função

$$d_H : G \longrightarrow \mathbb{R}^+ \cup \{0\}$$

é dita *função distância* se cumpre as seguintes duas condições:

(1) $d_H(h) = 0$ para todo $h \in H$.

(2) $d_H(g) > 0$ para todo $g \notin H$.

Se além das propriedades acima, a função satisfaz que $d_H(gh) = d_H(hg) = d_H(g)$ para todo $g \in G$ e todo $h \in H$, a função distância é dita *invariante*.

Segue um algoritmo que precisa de uma função distancia e que no passo 2 oferece uma solução ao problema de decomposição. O algoritmo a mostrar é semelhante ao algoritmo baseado na função comprimento (Algoritmo 6), a diferença é que usa uma função distância no lugar de uma função comprimento para determinar a qualidade dos candidatos a estudar:

Algoritmo 7: Ataque baseado na função distancia.

Entrada: As palavras z, $xzy \in F$, onde $x \in A_s$ e $y \in B_s$. O algoritmo itera no máximo N vezes, e depois disso, ele para. N é um número definido no inicio do algoritmo.

Saída: $\overline{x} \in A_s$ e $\overline{y} \in B_s$ tais que $xzy = \overline{x}z\overline{y}$.

início

1) $1 \longrightarrow \overline{x}$

2) **para** $g_i \in S_{A_s}$ **faça**

$x_i = \overline{x}g_i$, $y_i = z^{-1}x_i^{-1}xzy$ e calcule $d_{B_s}(y_i)$.

para $d_{B_s}(y_i) = 0$ **faça**

$\overline{x} = x_i$ e $\overline{y} = y_i$

fim

fim

3) Seja j é o índice no qual a função $d_{B_s}(y_i)$ alcança seu valor mínimo, se foi realizado o número máximo de iterações, acabou. Em outro caso, $x_j \longrightarrow \overline{x}$, voltar ao passo 2.

fim

Definição 5.2.2. Seja $w \in F$, definimos $P_i(w)$ e $N_i(w)$ o número de ocorrências de x_i e x_i^{-1} na forma normal de w.
Dado $s \geq 2$ um número inteiro. Para cada $w \in F$ definimos a *função distância ao subgrupo* B_s da seguinte maneira:

$$d_{B_s}(w) = \sum_{i=0}^{s}(P_i(w) + N_i(w)).$$

Exemplo 5.2.3. *Considere a palavra* $w = x_3x_8^2x_4x_6^{-1}x_5^{-1}$ *e seja* $s = 4$.

$$\begin{aligned} d_{B_s}(w) &= \sum_{i=0}^{4}(P_i(w) + N_i(w)) \\ &= 0 + 0 + 0 + 1 + 1 \\ &= 2. \end{aligned}$$

Notação: Vamos escrever $\ell_{NF}(w)$ para o comprimento da forma normal da palavra w.

Lema 5.2.4. *Seja* $w \in F$ *e* $x = x_i^{\pm 1}$ *um gerador do grupo* F. *Então* $\ell_{NF}(xw) = \ell_{NF}(w) \pm 1$ *(e dualmente* $\ell_{NF}(wx) = \ell_{NF}(w) \pm 1$*).*

Demonstração. Sem perda de generalidade, suponha que $w = x_{i_1} \cdots x_{i_k}x_{j_l}^{-1} \cdots x_{j_1}^{-1}$ é uma forma normal. Denotemos w_p e w_n as subpalavras positiva e negativa de w, respectivamente. Assumindo que $x = x_t$ é uma letra positiva, então a forma normal da palavra xw é obtida mudando de lugar a letra x aplicando as regras da forma normal um número adequado de vezes até obter $x_{i_1} \cdots x_{i_m}\ x_{t+m} \cdots x_{i_k}x_{j_l}^{-1} \cdots x_{j_1}^{-1}$, onde $i_m < t + m - 1$ e $i_{m+1} \geq t + m$. □

Lema 5.2.5. *A função distância d_{B_s} é invariante. No caso que x apague uma das letras na forma normal, o comprimento diminui de 1, se não aumenta de 1 porque a nova letra não desaparece da forma normal.*

Demonstração. É suficiente considerar os geradores de B_s. Sejam $w \in F$ e $b = x_{s+\alpha}^{\pm 1}$, onde $\alpha > 0$. Pelo lema 5.2.4 $\ell_{NF}(bw) = \ell_{NF}(w) \pm 1$, ou seja, que b ou muda de posição ou é eliminado com o elemento inverso em w. Além disso, o índice de b é maior que s e pode aumentar usando as relações da forma normal; assim, quando reescrevemos as letras de uma palavra usando as regras, temos que as letras que têm índice menor continuam tendo o mesmo índice, daí que todas as letras com índice menor ou igual a s não são afetadas quando movemos a letra b. Isto quer dizer que, em todos os casos, os geradores de índice menor ou igual a s continuam iguais, logo $d_{B_s}(bw) = d_{B_s}(w)$. □

Definição 5.2.6. Definimos a *função distância pesada de* B_s como segue:

$$\overline{d_{B_s}}(w) = \sum_{i=0}^{s}(s+1-i)(P_i(\widehat{w}) + N_i(\widehat{w})),$$

onde $\widehat{w}$ representa a forma normal da palavra w.

Observação 5.2.7. $d_{B_s}(w) \leq \overline{d_{B_s}}(w)$ para todo $w \in F$.

Exemplo 5.2.8. *Continuando o exemplo 5.2.3, dada $w = x_3x_8^2x_4x_6^{-1}x_5^{-1}$, calculamos agora a função distância pesada de w sabendo que $\widehat{w} = x_3x_4x_9^2x_6^{-1}x_5^{-1}$.*

$$\begin{aligned}\overline{d_{B_s}}(w) &= \sum_{i=0}^{4}(4+1-i)(P_i(\widehat{w}) + N_i(\widehat{w})) \\ &= 4(0) + 3(0) + 2(1) + 1(1) \\ &= 3.\end{aligned}$$

Lema 5.2.9. *$\overline{d_{B_s}}(w)$ é uma função distância invariante.*

Demonstração. A mesma demonstração do Lema 5.2.5 mostra que a multiplicação por b não altera nenhuma letra abaixo de $s+1$ na palavra w, assim $\overline{d_{B_s}}(w)$ também é invariante com a multiplicação. □

Agora vamos definir algumas funções distância para o subgrupo A_s.

Definição 5.2.10. Sejam $s \geq 2$ um número inteiro e $w \in F$ com forma normal $\widehat{w} = x_{i_1} \cdots x_{i_p}x_{j_n}^{-1} \cdots x_{j_1}^{-1}$. Definimos a *função distância ao subgrupo* A_s assim:

$$d_{A_s}(w) = |\{k : i_k - k \geq s\}| + |\{l : j_l - l \geq s\}| + |p - n|$$

Note que $d_{A_s}(w)$ é o número de letras que não cumprem a propriedade de A_s ($i_k - k < s$ e $j_k - k < s$ para todo k) mais a diferença entre o comprimento das partes positiva e negativa de w.

Exemplo 5.2.11. *Sejam $g = x_3x_7^{-1}$ e $h = x_0x_5^{-1} \in A_5$. Note que $d_{A_5}(g) = 1$ e além disso, aplicando as relações $\mathcal{R}$ para obter a forma normal do produto hg, temos o seguinte:*

$$hg = x_0x_5^{-1}x_3x_7^{-1} = x_0x_3x_6^{-1}x_7^{-1} = x_0x_3x_8^{-1}x_6^{-1},$$

daí, $d_{A_5}(hg) = 2$. O fato anterior significa que a função d_{A_s} não é invariante.

Observação 5.2.12. Considere a palavra $w = x_{i_1} \cdots x_{i_p}x_{j_n}^{-1} \cdots x_{j_1}^{-1}$ e seja

$$m_p = \max_{k\in\mathbb{Z}}\{i_k - k - s + 1\}.$$

Se multiplicamos w por $x_0^{m_p}$, ou seja, fazemos $x_0^{m_p}w$, as letras positivas mudam de posição m_p lugares, assim todas as letras positivas agora cumprem a propriedade de que $i_k - k < s$, pois nas primeiras m_p letras temos que $i_1, \cdots, i_{m_p}$ são todas zero, logo $i_k - k < 0 < s$. Nas letras restantes i_k vira $i_k + m_p$ e pela hipótese $m_p = i_k - k - s + 1$, logo $i_k + m_p - k - s + 1 = m_p$ e $i_k - k - s + 1 = 0$ implica que $i_k - k < s$.
Por semelhança, se multiplicamos à direita por $x_0^{-m_n}$, onde

$$m_n = \max_{k\in\mathbb{Z}}\{j_k - k - s + 1\}.$$

Os fatos anteriores não implicam que as palavras $x_0^{m_p}w$ e $wx_0^{-m_n}$ estavam em A_s, pois os comprimentos das subpalavras positiva e negativa de w são diferentes.
Sejam w' a forma normal da palavra $x_0^{m_p}wx_0^{-m_n}$, l_p e l_n o comprimento das subpalavras positiva e negativa de w', respectivamente. Se $l_p - l_n > 0$ então $w'x_0^{l_n-l_p} \in A_s$. Se $l_p - l_n < 0$ então $x_0^{l_n-l_p}w' \in A_s$, assim conseguimos que o comprimento da subpalavras positiva e da subpalavra negativa sejam iguais.
O anterior motiva a seguinte definição:

Definição 5.2.13. Definimos a *função distância pesada de* A_s como segue:

$$\overline{d_{A_s}}(w) = \sum_{\substack{k=1 \\ i_k-k\geq s}}^{p} (i_k - k - s + 1) + \sum_{\substack{k=1 \\ j_k-k\geq s}}^{n} (j_k - k - s + 1) + |p - n|.$$

Exemplo 5.2.14. *Lembrando o exemplo 5.2.11, onde $g = x_3x_7^{-1}$ e $h = x_0x_5^{-1} \in A_5$ observamos que na primeira palavra os índices tais que $i_k - k \geq 5$ ou $j_k - k \geq 5$ são $j_1 = 7$, enquanto que na palavra h temos que $j_1 - 1 \geq 5$ e $j_2 - 2 \geq 5$. Calculando $\overline{d_{A_s}}$ nos dois casos temos o seguinte:*

$$\overline{d_{A_s}}(g) = (7-1-4) + |1-1| = 2, \text{ enquanto}$$
$$\overline{d_{A_s}}(hg) = (6-1-4) + (8-2-4) + |2-2| = 3.$$

Isto permite afirmar que a função $\overline{d_{A_s}}$ não é invariante.

A função $\overline{d_{A_s}}$ vai ser usada numa aplicação experimental que será apresentada na próxima seção.

Definição 5.2.15. Seja $s \geq 2$ um número inteiro. Seja $w \in F$ com forma normal $\widehat{w} = x_{i_1} \cdots x_{i_p} x_{j_n}^{-1} \cdots x_{j_1}^{-1}$. Seja

$$m_p = max(\{0\} \cup \{i_k - k - s + 1 : k = 1, \ldots, p\}).$$

e

$$m_n = max(\{0\} \cup \{j_k - k - s + 1 : k = 1, \ldots, n\}).$$

A *função distância baseada no máximo* é definida como segue:

$$d_{A_s}^m(w) = m_p + m_n + |(p + m_p) - (n + m_n)|.$$

Lema 5.2.16. *$d_{A_s}^m$ é uma função distância invariante.*

Demonstração. É suficiente provar que dado um gerador b de A_s e $w \in F$, $d_{A_s}^m(bw) = d_{A_s}^m(wb) = d_{A_s}^m(w)$. Vamos considerar a multiplicação à esquerda pelos geradores e seus inversos. Seja $w = x_{i_1} \cdots x_{i_p} x_{j_n}^{-1} \cdots x_{j_1}^{-1}$ uma forma normal e suponha que $|w| = n$. Considere o gerador $x_0 x_t^{-1}$ com $1 \leq t \leq s$, denotemos w' a forma normal de $x_0 x_t^{-1} w$, e para os parâmetros p, n, m_p, m_n denotamos por p', n', m_p', m_n' os parâmetros correspondentes em w'.

Cada uma das letras x_0 ou x_t^{-1} pode ser eliminada: nesse caso $|w'| = n - 1$, ou pode ocupar uma posição adequada: agora teremos que $|w'| = n + 1$. Do raciocínio anterior temos 4 possíveis casos:

(1) x_0 não é eliminado e x_t^{-1} é eliminado:
Neste caso $w' = x_0 x_{i_1} \cdots x_{i_m} x_{i_m+2} \cdots x_{i_p} x_{j_n}^{-1} \cdots x_{j_1}^{-1}$; aqui aplicamos $(\mathcal{R}2)$ m vezes até que x_{t+m}^{-1} é eliminado com x_{i_m+1}. Note que $p' = p$, $n' = n$ e como as letras negativas não são afetadas então $m_n' = m_n$. Observe que as primeiras m letras não podem ser letras ruins pois quando usamos $(\mathcal{R}2)$ para cada $k = 1, \ldots, m$ e reescrevemos o seguinte $x_{t+k-1}^{-1} x_{i_k} \longrightarrow x_{i_k} x_{t+k}^{-1}$, necessariamente $i_k < t + k - 1$ se, e somente se, $i_k - k < t - 1 < s$. A multiplicação por x_0 à esquerda só incrementa as posições, logo $i_k - k$ diminui.
Agora, as possíveis letras ruins acima de x_{i_m} não são alteradas, logo $m_p' = m_p$ e em consequência temos que $d_{A_s}^m(w') = d_{A_s}^m(w)$.

(2) x_t^{-1} e x_0 são eliminados:
Aplicando $(\mathcal{R}1)$ a x_0 percorrendo as letras positivas os índices destas letras vão diminuir em um, depois aplicamos $(\mathcal{R}2)$, agora x_0 percorrendo as letras negativas até que seja eliminado com alguma letra denotada x_{j_q}, daí temos que $w' = x_{i_1-1}\cdots x_{i_m-1}x_{i_{m+2}-1}\cdots x_{i_p-1}x_{j_n-1}^{-1}\cdots x_{j_{q+1}-1}^{-1}x_0^{-q+1}$.
Aqui $p' = p-1$, $n' = n-1$ e $m'_n = m_n$ porque todas as letras negativas $x_{j_k}^{-1}$ com $j_k > 0$ diminuem os índices e as posições em um, a mesma idéia acontece no caso das letras positivas acima de i_m, então $m'_p = m_p$, logo $d_{A_s}^m(w') = d_{A_s}^m(w)$.

(3) Nem x_t^{-1} nem x_0 são eliminados:
Se $i_1, \cdots, i_m < t < s$ aplicamos $(\mathcal{R}3)$, então

$$w' = x_0x_{i_1}\cdots x_{i_m}x_{t+m}^{-1}x_{i_{m+1}}\cdots x_{i_p}x_{j_n}^{-1}\cdots x_{j_1}^{-1}.$$

Daí, se $i_{m+1} > t+m$, então $w' = x_0x_{i_1}\cdots x_{i_m}x_{i_{m+1}+1}x_{i_{m+2}+1}\cdots x_{i_p+1}x_{t+m}^{-1}x_{j_n}^{-1}\cdots x_{j_1}^{-1}$.
E finalmente, se $j_1, \cdots, j_n > t+m$ aplicamos $(\mathcal{R}4)$ e obtemos que
$w' = x_0x_{i_1}\cdots x_{i_m}x_{i_{m+1}+1}x_{i_{m+2}+1}\cdots x_{i_p+1}x_{j_n+1}^{-1}\cdots x_{j_{q+1}+1}$.
Aqui $p' = p+1$, $n' = n+1$. Como $i_1, \cdots, i_m < t < s$ então as letras ruins só podem estar depois de x_{i_m}, depois desta letra tanto i_k como k aumentam em um, logo $m'_p = m_p$. No caso das letras negativas cujos índices aumentam, também as posições aumentam, daí $j_k - k$ é preservado para todas as letra originais em w. Assim, $m'_n \geq m_n$ e o único caso quando pode aumentar é quando o novo máximo é alcançado na nova letra, ou seja, $m'_n = (t+m)-(q+1)-s+1 > m_n$. Como $t \leq s$, $m \leq p$ e $q \leq n$, temos que $m'_n \leq p-q$, do anterior se segue que:

$$\begin{aligned}(p'+m'_p)-(n'+m'_n) = (p'-n')+(m'_p-m'_n) = (p+1)-(n+1)+m_p-m'_n \geq \\ \geq m_p+(p-n)-(p-q) = m_p+q-n \geq 0.\end{aligned}$$

Assumindo que $m'_n > m_n$, temos que

$$(p-n)+(m_p-m_n) > (p'-n')+(m'_p-m'_n) \geq 0,$$

e se m_n aumenta, $|(p+m_p)-(n+m_n)|$ decresce a mesma quantidade, logo $d_{A_s}^m(w') = d_{A_s}^m(w)$.

(4) Se x_t^{-1} não é eliminada e x_0 é eliminada: Neste caso temos o seguinte:

$$w' = x_{i_1-1}\cdots x_{i_m-1}x_{i_m+1}\cdots x_{i_p}x_{j_n}^{-1}\cdots x_{j_q+1}^{-1}x_{j_q-1}^{-1}\cdots x_{j_{r+1}-1}^{-1}x_0^{1-r},$$

onde $p' = p$, $n' = n$, $m'_p = m_p$ (pois as primeiras m letras positivas, cujos índices mudaram não contem letras ruins) e m'_n pode aumentar se m'_n é alcançado na letra x_{t+m-1}^{-1} . Repetindo os cálculos provamos que $d_{A_s}^m(w') = d_{A_s}^m(w)$.

Agora considerando o elemento inverso $x_t x_0^{-1}$ e denotando $w' = x_t x_0^{-1} w$ temos quatro casos:

(1) Se x_0^{-1} é eliminado mas x_t não: isto só pode acontecer quando $i_1 = 0$, então $w' = x_{i_2} \cdots x_{i_m} x_{t+m-1} x_{i_m+1} \cdots x_{i_p} x_{j_n}^{-1} \cdots x_{j_1}^{-1}$. Aqui $p' = p$, $n' = n$, $m'_n = m_n$ e $m'_p = m_p$ porque nas letras x_{i_2} até x_{i_m} não existem letras ruins e as posições das outras letras não mudam.

(2) Se x_t e x_0^{-1} são eliminados: Suponha que x_t é eliminado com $x_{j_q}^{-1}$, então $w' = x_{i_2} \cdots x_{i_m} x_{i_{m+1}-1} \cdots x_{i_p-1} x_{j_n-1}^{-1} \cdots x_{j_{q+1}-1}^{-1} \cdots x_{j_{q-1}}^{-1} \cdots x_{j_1}^{-1}$. Aqui $p' = p-1$, $n' = n-1$, $m'_p = m_p$ e $m'_n = m_n$.

(3) Nem x_t nem x_0^{-1} são eliminados: Neste caso temos que

$$w' = x_{i_1+2} \cdots x_{i_m+2} x_{t+m} x_{i_{m+1}+1} \cdots x_{i_p+1} x_{j_n+1}^{-1} \cdots x_{j_q+1}^{-1} x_0^{-q}.$$

Aqui $p' = p+1$, $n' = n+1$, $m'_p = m_p$ e $m'_n = m_n$.

(4) Se x_0^{-1} não é eliminado e x_t é eliminado: Neste caso temos o seguinte: $w' = x_{i_1+2} \cdots x_{i_m+2} x_{i_{m+1}} \cdots x_{i_p} x_{j_n}^{-1} \cdots x_{j_{q+1}}^{-1} x_{j_{q-1}+1}^{-1} \cdots x_{j_r+1}^{-1} x_0^{-r}$, onde $j_q = t + m$. Neste caso, quaisquer letra positiva que seja ruim preserva os índices e as posições, as letras negativas $j_r+1, \cdots, j_q-1$ mudaram os índices e as posições, enquanto as letras $j_q + 1, \cdots, j_n$ preservam índices e posições. Assim, $m'_p = m_p$, $m'_n = m_n$, $p' = p$ e $n' = n$. Daí temos que $d_{A_s}^m(w') = d_{A_s}^m(w)$.

□

5.3 Resultados experimentais

Nesta seção vamos mostrar uma aplicação feita por Ruinskiy, Shamir e Tsaban em [10], onde fazem um teste das funções distância definidas anteriormente na implementação do algoritmo 6 proposto por Shpilrain e Ushakov.

Primeiro foi gerado o elemento público azb, e o objetivo é descobrir um dos dois elementos privados a ou b. Para descobrir a são usadas as funções d_{B_s} e $\overline{d_{B_s}}$ para analisar a qualidade do complemento de a. Analogamente, as funções d_{A_s}, $\overline{d_{A_s}}$ e $d_{A_s}^m$ são usadas para achar o elemento b.

Para cada função, o experimento foi desenvolvido pelo menos 1000 vezes, cada vez com novas chaves geradas aleatoriamente, usando os parâmetros $s = 3$, $L = 256$, onde L representa o comprimento da forma normal da palavra escolhida no grupo F. Os resultados obtidos são mostrados nas seguintes tabelas, denotando P_a a probabilidade de encontrar a e P_b a probabilidade de encontrar b :

	d_{B_s}	$\overline{d_{B_s}}$
P_a	11.7%	3.4%

É possível pensar que a caracterização de invariância das funções distância poderia ser usada para avaliar a adequação das funções. Neste caso lembremos que as duas funções, d_{B_s} e $\overline{d_{B_s}}$, são invariantes.

	d_{A_s}	$\overline{d_{A_s}}$	$d^m_{A_s}$
P_b	3.7%	3.4%	23.3%

Neste caso, as funções d_{A_s} e $\overline{d_{A_s}}$ são não invariantes, enquanto a função $d^m_{A_s}$ é invariante. Experimentos anteriores têm mostrado que dados $a_1 z b_1$ e $a_2^{-1} z^{-1} b_2^{-1}$ elementos públicos, a probabilidade de achar a_1 ou a_2^{-1} são similares, e acontece o mesmo com os elementos b_1 ou b_2^{-1}. Daí, para calcular a probabilidade geral, é suficiente calcular a probabilidade de um dos quatro elementos; assumindo que todas as probabilidades são independentes, então a probabilidade total de sucesso é aproximadamente $1-(1-P_a)^2(1-P_b)^2$ pois o protocolo tem dois elementos do tipo de a e dois do tipo de b. Isto significa que a probabilidade de encontrar as chaves é não nula, daí, a segurança do protocolo pode ser quebrada.

Referências

[1] I. Anshel, M. Anshel, and D. Goldfeld. An algebraic method for public-key cryptography. *Math. Res. Lett.*, 6(3-4):287–291, 1999.

[2] J. M. Belk. *Thompsons' group F*. ProQuest LLC, Ann Arbor, MI, 2004. Thesis (Ph.D.)–Cornell University.

[3] O. Bogopolski. *Introduction to group theory*. EMS Textbooks in Mathematics. European Mathematical Society (EMS), Zürich, 2008. Translated, revised and expanded from the 2002 Russian original.

[4] J. W. Cannon, W. J. Floyd, and W. R. Parry. Introductory notes on Richard Thompson's groups. *Enseign. Math. (2)*, 42(3-4):215–256, 1996.

[5] M. Kassabov and F. Matucci. The simultaneous conjugacy problem in groups of piecewise linear functions. *Groups Geom. Dyn.*, 6(2):279–315, 2012.

[6] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J.-s. Kang, and C. Park. New public-key cryptosystem using braid groups. In *Advances in cryptology—CRYPTO 2000 (Santa Barbara, CA)*, volume 1880 of *Lecture Notes in Comput. Sci.*, pages 166–183. Springer, Berlin, 2000.

[7] N. Koblitz. *A course in number theory and cryptography*, volume 114 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1994.

[8] F. Matucci. Cryptanalysis of the Shpilrain-Ushakov protocol for Thompson's group. *J. Cryptology*, 21(3):458–468, 2008.

[9] A. Myasnikov, V. Shpilrain, and A. Ushakov. *Group-based cryptography*. Advanced Courses in Mathematics. CRM Barcelona. Birkhäuser Verlag, Basel, 2008.

[10] D. Ruinskiy, A. Shamir, and B. Tsaban. Cryptanalysis of group-based key agreement protocols using subgroup distance functions. In *Public key cryptography—PKC 2007*, volume 4450 of *Lecture Notes in Comput. Sci.*, pages 61–75. Springer, Berlin, 2007.

[11] D. Ruinskiy, A. Shamir, and B. Tsaban. Length-based cryptanalysis: the case of Thompson's group. *J. Math. Cryptol.*, 1(4):359–372, 2007.

[12] V. Shpilrain and A. Ushakov. Thompson's group and public key cryptography. *Lecture Notes Comp. Sc 3531, Springer, Verlag*, pages 151–164, 2005.

[13] E. A. M. Silva. Um estudo do sistema criptográfico RSA. 2005. Trabalho de Conclusão do Curso, Universidade Católica de Brasilia, Departamento de Matemática.

[14] M. Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston, 2006.