



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

LUCAS ISMAILY BEZERRA FREITAS

**MÚSICA E AUTÔMATO: UMA RECEITA DE NOMEAÇÃO
DE ACORDES**

**QUIXADÁ
2012**

LUCAS ISMAILY BEZERRA FREITAS

**MÚSICA E AUTÔMATO: UMA RECEITA DE NOMEAÇÃO
DE ACORDES**

Trabalho de Conclusão de Curso submetido à Coordenação do
Curso de Graduação em Sistemas de Informação da
Universidade Federal do Ceará como requisito parcial para
obtenção do grau de Bacharel.

Área de concentração: computação

Orientador: Prof. Dr. Davi Romero de Vasconcelos
Co-Orientador: Prof. Me. Francisco E. F. de Aragão

**QUIXADÁ
2012**

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca do Campus de Quixadá

F936m Freitas, Lucas Ismaily Bezerra.

Música e Autômato: uma receita de nomeação de acordes / Lucas Ismaily Bezerra Freitas. –
2012.
115 f. : il. p &b; enc.; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de
Sistemas de Informação, Quixadá, 2012.

Orientação: Prof. Dr. Davi Romero de Vasconcelos.

Coorientação: Prof. Me. Francisco Erivelton Fernandes de Aragão.

1. Teoria dos Autômatos 2. Teoria Musical 3. Padrão MIDI I. Título.

CDD 005.131

LUCAS ISMAILY BEZERRA FREITAS

**MÚSICA E AUTÔMATO: UMA RECEITA DE NOMEAÇÃO DE
ACORDES**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso de Graduação em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: computação

Aprovado em: _____ / junho / 2012.

BANCA EXAMINADORA

Prof. Dr. Davi Romero de Vasconcelos (Orientador)
Universidade Federal do Ceará-UFC

Prof. Me. Francisco Erivelton Fernandes de Aragão
Universidade Federal do Ceará-UFC

Prof. Me. Samy Soares Passos de Sá
Universidade Federal do Ceará-UFC

A minha mãe

AGRADECIMENTOS

Agradeço primeiramente a Deus, não só pela monografia, mas por tudo. Depois, a minha família, em especial minha mãe. Mãe, muito obrigado! A você, sempre você, irão todos os créditos por todos os trabalhos que um dia eu possa realizar!

A Priscila, minha namorada, por todo apoio e compreensão nas horas de *stress*, Priscila sem você este trabalho não sairia da introdução. Mesmo sendo uma chata, eu ainda consigo te amar... Como se explica isso?

Ao Davi, muito mais que orientador, é complicado expressar em palavras o tamanho de minha gratidão, não pela orientação nesse trabalho, mas por você ter a capacidade ímpar de sempre usar a palavra certa no momento certo. Davi, muito obrigado! Como te falei, serás um ótimo pai, tomara que logo ☺.

Ao Aragão. Aragão, você é um exemplo a ser seguido por todos. Sua simplicidade, humor e inteligência serão sempre por mim copiados, sem nenhum pudor! Aragão muito obrigado por todas as conversas que mudavam totalmente do tema. Nunca me esquecerei delas!

Por fim, agradeço a todos que de alguma forma ajudou neste trabalho, sintam-se todos cumprimentados! Em especial aos meus amigos: Antônio Wellington, Tobias Saraiva, Laurertan Tavares e Ricelly Jáder, este último um primo ingrato.

"É a ciência feita de suor. Temos sobressalto de lógica, temos anos de indagações não respondidas, temos frustrações, temos horas de arrancar os cabelos, mas o verdadeiro poder do gênio é a força de vontade para fazer todos os erros necessários para chegar à resposta"

(Michio Kaku)

RESUMO

O problema de nomeação de acordes consiste em determinar os acordes de uma sequência de notas musicais. Há regras musicais que definem quais notas formam ou não acordes. Uma complicação surge devido às notas que possuem a mesma frequência, porém nomes distintos. Este trabalho se propõe a abordar esta problemática com Teoria dos Autômatos, portanto nos propomos a desenvolver um Autômato que reconhece os acordes (tríades e tétrade) de uma música retirada de um arquivo no formato MIDI (Music Instrument Digital Interface), a partir do Autômato, implementar um sistema Java Desktop que o utiliza como rotina para exibir ao usuário a sequência de Acordes. O Autômato proposto recebe como entrada uma sequência de notas retiradas de um arquivo MIDI e retorna como saída uma sequência de acordes, considerando o problema dos nomes das notas. O sistema, chamaremos de *QMusic*, utiliza as APIs Java *Sound* e *jMusic* para facilitar a manipulação de arquivos MIDI e permite várias formas de exibição, desde a notação ABC até a notação completa do acorde, exibindo toda sua estrutura, a saber: Tônica, Terça, Quinta e, caso Tétrade, Sétima.

Palavras chave: Teoria dos Autômatos, Teoria Musical, Padrão MIDI.

ABSTRACT

The naming chords problem consists in determine the chords sequence of musical notes. There are rules that define what are the notes that form or not chords. A complication arises due the notes that have the same frequency, but different names. This paper aims approach this problem with Automata Theory. Therefore, we propose develop an automata who recognizes the chords (triads and sevenths) of a song taken from a file in format MIDI (Music Instrument Digital Interface), from the automata, implement a Java Desktop system that uses him as routine to display to the user the sequence of chords. The proposed automata, receives as input a sequence of notes taken from a MIDI file and returns as output a sequence of chords, considering the problem of the names of the notes. The system will call *QMusic*, and it uses the Java Sound API and JMusic to facilitate manipulation of MIDI files. It also allows, multiple forms of visualizations, ranging from the ABC notation to the complete notation of the chord, showing its structure, namely: Tonic or Root, Third, Fifth and if Tetrad, Seventh.

Keywords: Automata Theory, Musical Theory, Standard MIDI.

LISTA DE FIGURAS

Figura 1 – Os instrumentos de fabricantes diferentes não se comunicavam. Fonte: Machado (2001).....	14
Figura 2 – As notas Dó sustenido e Ré bemol são <i>Enharmonic</i>	15
Figura 3 – Fluxo geral do trabalho.....	16
Figura 4 – Diferenças de altura entre as notas na escala musical ocidental. Fonte: Mello (2010).....	19
Figura 5 – “Escala de <i>dó</i> Maior”. Fonte: VILAR (2005).....	20
Figura 6 – Tríade de Sol maior (G). Fonte: ARAGÃO (2011).....	22
Figura 7 – Tríade de Dó menor (Cm).....	22
Figura 8 – Tríade de Lá diminuta (A°).....	23
Figura 9 – Tríade de Dó aumentado (C+).....	23
Figura 10 – Sequência de passos para gravar as notas em um SMF. Fonte: Machado (2005).....	30
Figura 11 – Sistema de Estados Finitos que modela um interruptor liga/desliga.....	33
Figura 12 – Representação Autômato Finito. Fonte: Rangel (2001).....	36
Figura 13 – AFD que reconhece uma linguagem que possui somente a palavra ‘ <i>priscila</i> ’	38
Figura 14 – AFN que reconhece uma Linguagem com a palavra ‘ <i>amor</i> ’ como sufixo.....	42
Figura 15 – AFN ϵ que reconhece uma Linguagem com <i>b</i> ’s seguido por <i>a</i> ’s.....	45
Figura 16 – Ilustração do AFN ϵ ACORDE que Reconhece Tríades e Tétrade.....	49
Figura 17 – AFN ϵ TRIADE que Reconhece Tríade.....	49
Figura 18 – AFN ϵ T_MA que Reconhece Tríade Maior.....	50
Figura 19 – AFN ϵ T_ME que Reconhece Tríade Menor.....	51
Figura 20 – AFN ϵ T_AUM que Reconhece Tríade Aumentada.....	53
Figura 21 – AFN ϵ T_DIM que Reconhece Tríade Diminuta.....	54
Figura 22 – AFN ϵ TRIADE que Reconhece uma Tríade.....	55
Figura 23 – AFN ϵ TE_MA que Reconhece uma Tétrade a partir de uma Tríade Maior.....	57
Figura 24 – AFN ϵ TE_ME que Reconhece uma Tétrade a partir de uma Tríade Menor.....	58
Figura 25 – AFN ϵ TE_DIM que Reconhece uma Tétrade a partir de uma Tríade Diminuta.....	60
Figura 26 – AFN ϵ TE_AUM que Reconhece uma Tétrade a partir de uma Tríade Aumentada.....	61
Figura 27 – AFN ϵ TETRADE que Reconhece uma Tétrade.....	62
Figura 28 – AFN ϵ ACORDE que Reconhece Tríades e Tétrade.....	64
Figura 29 – Tela inicial do <i>Qmusic</i>	66
Figura 30 – Tela depois do carregamento do arquivo teste.mid.....	67
Figura 31 – Tela em modo <i>Standard</i>	68
Figura 32 – Tela em modo <i>Advanced</i>	69
Figura 33 – Tela em modo <i>Show all Results</i>	70
Figura 34 – Seção A do coral <i>Aus Meines Herzens Grunde</i> de Johann S. Bach. Fonte: Aragão (2011)....	72
Figura 35 – Lista de notas da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Johann S. Bach, com a remoção dos casos em que não se forma acorde (tríade ou tétrade).....	74
Figura 36 – Lista de Tríades e Tétrade da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	75

Figura 37 – Lista de Tríades Menores da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	76
Figura 38 – Lista de Tríades Maiores da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	76
Figura 39 – Lista de Tríades Aumentadas da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	77
Figura 40 – Lista de Tríades Diminutas da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	77
Figura 41 – Lista de Tríades da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	78
Figura 42 – Lista de Tétrade Maiores da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	78
Figura 43 – Lista de Tétrade Menores da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	79
Figura 44 – Lista de Tétrade da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	79
Figura 45 – Lista de Tríades da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	107
Figura 46 – Lista de Tétrade da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	108
Figura 47 – Lista de Tétrade Menores da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	108
Figura 48 – Lista de Tríades e Tétrade da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	109
Figura 49 – Lista de Tríades Menores da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	110
Figura 50 – Lista de Tríades Aumentadas da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	110
Figura 51 – Lista de Tríades Diminutas da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	111
Figura 52 – Lista de Tétrade Maiores da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	111
Figura 53 – Lista de Tríades Maiores da Seção A do coral <i>Aus Meines Herzens Grunde</i> de Bach.....	112

LISTA DE TABELAS

Tabela 1 – Intervalos musicais compostos.....	18
Tabela 2 – Todas as notas com acidentes e suas freqüências.....	20
Tabela 3 – Acordes mais freqüentes usado como base o acorde de <i>Dó</i> . Fonte: adaptado de VILAR (2005).	21
Tabela 4 – Principais mensagens MIDI.....	28
Tabela 5 – Notas ordenadas por tempo de execução.....	72
Tabela 6 – Notas extraídas, com as linhas em vermelho indicando as que não seriam utilizadas.....	73
Tabela 7 – Notas, ordem de execução e acordes formados.....	73

SUMÁRIO

1 INTRODUÇÃO.....	14
2 TEORIA MUSICAL	18
2.1 Fundamentos da Teoria Musical.....	18
2.1.1 Notas musicais.....	18
2.1.2 Intervalos	19
2.1.3 Acidentes	20
2.1.4 Escalas	21
2.1.5 Cifra ABC.....	22
2.2 Acordes	22
2.2.1 Tríade.....	23
2.2.2 Tétrade	24
3 TECNOLOGIA MIDI	25
3.1 Introdução ao MIDI	25
3.1.1 Protocolo MIDI 1.0	25
3.1.2 Mensagens MIDI.....	27
3.1.3 Os arquivos SMF (<i>Standard MIDI File</i>)	30
4 LINGUAGENS FORMAIS E AUTÔMATOS	32
4.1 Fundamentos de Linguagens Formais	32
4.1.1 Sistema de Estados Finitos	33
4.2 Teoria dos Autômatos	34
4.2.1 Autômatos Finitos	37
4.2.2 Autômato Finito Determinístico	38
4.2.3 Autômato Finito Não Determinístico	41
4.2.4 Autômato Finito com ϵ -transições.....	44
4.2.5 Composição de Autômatos	48
5 AUTÔMATO PROPOSPO E SISTEMA <i>Qmusic</i>	50
5.1 Autômato	50
5.1.1 Reconhecedor de Tríade (TRIADE).....	51
5.1.2 Reconhecedor de Tétrade (TETRADE)	59
5.1.3 Reconhecedor de Tríades e Tétrades (ACORDE)	66
5.2 Sistema <i>Qmusic</i>	68
5.2.1 Visão Geral	68
5.2.2 Carregando, ouvindo e exibindo as notas de um arquivo MIDI.....	69
5.2.3 Reconhecendo Acordes	71
5.2.4 Opções de Reconhecimento	73
6 ESTUDO DE CASO	75
6.1 Música Escolhida.....	75
6.2 Execução do <i>Qmusic</i>	77
7 CONSIDERAÇÕES FINAIS	84

REFERÊNCIAS	85
APÊNDICES	87
7.1 APÊNDICE A – Reconhecedor de Tríade.....	87
7.2 APÊNDICE B – Reconhecedor de Tétrade	100
7.3 APÊNDICE C – Reconhecedor de Tríades e Tétrade.....	109
7.4 APÊNDICE D – Listas de Telas do <i>Qmusic</i> referente ao Estudo de Caso.....	110

1 INTRODUÇÃO

MIDI (*Music Instrument Digital Interface*) é um protocolo que permite a comunicação e sincronização entre instrumentos musicais. Surgiu no início dos anos 80 por fabricantes de sintetizadores¹ japoneses e americanos, tais como *Sequential Circuits*, *Oberheim* e *Roland Corporation*. Tendo sua primeira publicação em Janeiro de 1983 (MIDI, 2011). A Figura 1 retrata o período pré-MIDI.

A motivação para sua criação foi a necessidade de interligação de instrumentos de fabricantes diferentes, especialmente teclado. Inicialmente o protocolo MIDI era utilizado para interligar teclados apenas de uma única empresa, depois expandido para teclados de empresas diferentes, instrumentos diferentes, e por fim a comunicação com computadores.

É comum os músicos utilizarem instrumentos de mais de um fabricante, por isso o grande sucesso do protocolo MIDI entre os músicos.



Figura 1– Os instrumentos de fabricantes diferentes não se comunicavam

Fonte: Machado (2001)

Seu surgimento proporcionou muitos benefícios aos músicos, por exemplo, a capacidade de intercomunicação entre máquinas, que originou um ganho de tempo muito grande. Vantagens também para fabricantes de instrumentos, com o protocolo MIDI os teclados, por exemplo, se popularizou muito. Outros benefícios que podemos elencar são: Interface padrão, a capacidade de comunicação bidirecional entre computador e instrumento, a possibilidade de criação de softwares genéricos (independente de instrumentos) e a facilidade de manipulação e criação das melodias. Com o MIDI, os músicos passaram a montar pequenos estúdios em suas residências (MACHADO, 2001).

¹ Instrumentos capazes de emular e gerar timbres musicais

Porém, um problema surge no momento de recuperar a notação musical² do código MIDI, ou seja, nomear corretamente os acordes. Na música o mesmo som³ pode ter mais de um nome, por exemplo, dó sustenido e ré bemol (Figura 2), embora tenham o mesmo som (frequência), possuem nomes diferentes. A esse evento dá-se o nome ortografia desarmônica, ou enarmônica (*Enharmonic Spelling*) (Figura 2). Encontramos uma definição de *Enharmonic Spelling* em (JONES, 2011, p.2), que afirma, “duas notas são enarmônicas (*enharmonic*) se tem o mesmo som e são differentemente escritas”. Essa diferença de nomes, do ponto de vista de teoria musical, mais especificamente formação de acordes, pode definir uma música bem escrita (correta) ou não. Como os arquivos MIDI capturam, dentre outras, as frequências das notas que foram reproduzidas pelo músico (MIDI, 2011), no momento de transcrevê-las para notação musical surge um dilema: usar bemol ou sustenido? Por exemplo, usar dó bemol ou re sustenido?



Figura 2– As notas Dó sustenido e Ré bemol são *Enharmonic*

Existem vários sistemas atualmente que fazem uma conversão de MIDI para partitura, por exemplo: *8notes.com* (8NOTES, 2011) e o *MidiIllustrator Maestro* (MAESTRO, 2011), porém os estudantes de música terão um trabalho dispendioso para corrigirem possíveis equívocos de transcrição, porque além de ser muito trabalhoso nomear corretamente os acordes, requer um conhecimento especializado de teoria musical.

Este ramo da computação que estuda a música é chamado de computação musical. Encontramos uma definição de computação musical em (SBC, 2011), que afirma:

A área de Computação Musical compreende pesquisa científica, tecnológica e artística nas áreas de composição algorítmica, análise/síntese de som, acústica musical, análise musical assistida por computador, composição musical assistida por computador, processamento digital de áudio, multimídia e qualidade de serviço.

Na literatura encontramos alguns trabalhos relacionados. Vilar (2005) em sua dissertação desenvolveu um software, denominado AHO (Analizador Harmônico Otimista),

² “É o nome genérico de qualquer sistema de escrita utilizado para representar graficamente uma peça musical” (MELO, 2010)

³ “Cada nota musical corresponde a uma frequência em particular, que pode ser medida em hertz.” (ARAGÃO, 2011)

que faz análise harmônica de uma música utilizando Gramática Livre de Contexto, trabalho semelhante ao de Högberg (2005), que desenvolveu um sistema denominado Wilow, no qual usa Gramática para gerar música. Cunha (1999) aborda na sua dissertação a previsão de acordes no Jazz, utilizando Redes Neurais. Majchrzak e Gingras (2009) em seu artigo abordam a importância do *Enharmonic Spelling* em modelos computacionais. Aragão (2011) discorre em sua tese⁴ análise harmônica, utilizando Lógica Clássica de Primeira Ordem como ferramenta.

Este trabalho se propõe a nomear os acordes de uma música recebida em modo MIDI. Para isso iremos desenvolver um Autômato que recebe como entrada uma sequência de notas retiradas de um arquivo MIDI, e reconhecerá os acordes (tríades e tétrade), considerando o *Enharmonic Spelling*.

A partir do Autômato iremos desenvolver um sistema *desktop* em Java SE, utilizando as API's (*Application Programming Interface*) Java Sound e *jMusic* para facilitar a manipulação de arquivos MIDI. O Java Sound, possui métodos que permite a execução de arquivos MIDI e o *JMusic* tem métodos que atribui nomes as frequências das notas retiradas de um arquivo MIDI. As notas serão passadas para o autômato, que responderá se estas notas formam, ou não, acordes.

Este sistema irá exibir os acordes corretamente ou uma mensagem de erro (com sugestões de correções) caso a entrada não esteja de acordo com a teoria musical. O fluxo geral está ilustrado na Figura 3.

Com desenvolvimento deste trabalho esperamos um ganho no processo de aprendizagem dos estudantes de música.

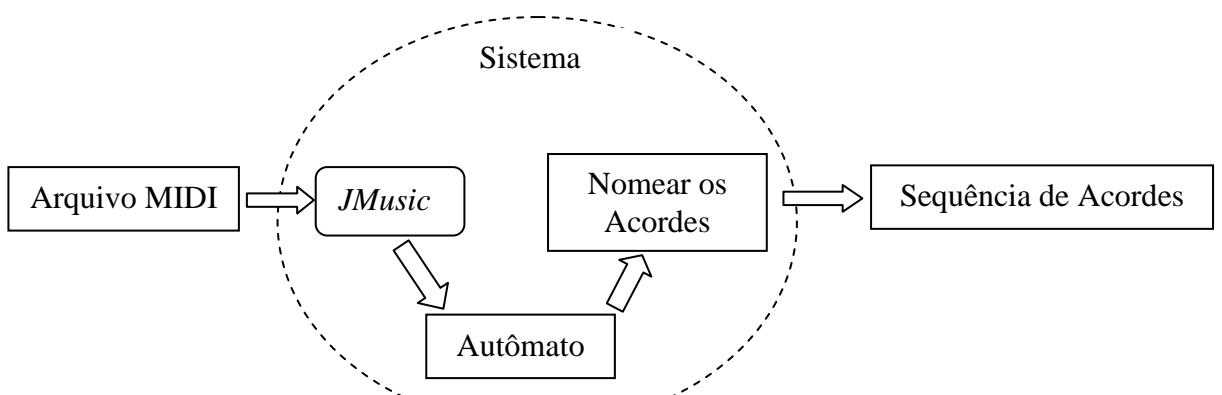


Figura 3– Fluxo geral do trabalho

⁴ Tese não publicada. Passível de modificações.

A seguir abordaremos os conceitos de Teoria de Musical, Tecnologia MIDI 1.0 e Linguagens Formais e Autômatos.

2 TEORIA MUSICAL

Apresentamos, nesta seção, os conceitos de Teoria Musical que serão utilizados neste trabalho. Para detalhes além do que apresentamos a seguir, indicamos: Mello (2010), Cardoso e Mascarenhas (1996) e Schöenberg (1979).

2.1 Fundamentos da Teoria Musical

De acordo com o maestro paulista Júlio Medaglia (2008, p.9): “Desde que se tem notícia da presença do homem na Terra, sabe-se que o som o acompanha das mais diversas maneiras”. Essa relação há muito acontece, existem evidências arqueológicas de escrita musical no Egito e Mesopotâmia por volta do terceiro milênio A.C. (MELLO, 2010).

Como toda linguagem, a música possui sua gramática, chamada notação musical, que é a forma de anotar as sucessões de vibrações sonoras (as frequências). De forma mais técnica, suas regras de produção, ou seja, regras de formação de palavras (partituras ou cifras). Lamentavelmente esta notação é ambígua, pois permite denotar de maneiras diferentes o mesmo objeto (a mesma nota).

A seguir apresentamos definições de Notas musicais, intervalos, acidentes, escalas, cifra ABC e acordes (tríades e tétrades), importantes para a compreensão deste trabalho.

2.1.1 Notas musicais

Notas musicais são nomes atribuídos a cada frequência específica. Neste trabalho escrevemos as notas musicais com letras minúsculas e em itálico, casos especiais serão explicitados para o leitor, por exemplo, 220 Hz é denominado *lá₄*, assim como todos os múltiplos de dois dessa frequência. Se uma nota possui frequência de x Hz e outra $2x$ Hz dizemos que elas formam uma consonância e recebem o mesmo nome, diferenciando a escrita através de índices adicionados no canto direito inferior, por exemplo, *dó₁*, *dó₂*.

Há casos de notas que possuem frequências iguais, porém nomes distintos, esse fenômeno chamamos de enarmonia e as notas de enarmônicas (JONES, 2011), por exemplo, *dó* sustenido e *ré* bemol.

Existem infinitas notas com frequências distintas. São ao total, na tradição européia, sete classes de notas musicais (cada classe possui infinitas notas), a saber, *dó*, *ré*, *mi*, *fa*, *sol*,

lá, si. Também conhecidas em alguns países de língua germânica (MELLO, 2010) como, *a* (*lá*), *b* (*si*), *c* (*dó*), *d* (*ré*), *e* (*mi*), *f* (*fá*), *g* (*sol*). Um detalhe interessante é que a Alemanha conta com um símbolo a mais, o *si bemol* é representado pelo símbolo *h*.

2.1.2 Intervalos

Intervalo é a distância entre duas notas, o menor intervalo possível entre duas notas é dito semitom. Dois semitons equivalem a um tom. Tom é a “medida” dos intervalos. A Tabela 1 exibe os intervalos existentes e seus respectivos valores (em semitons).

Tabela 1. Intervalos musicais compostos

NOME DO INTERVALO	VALOR
Segunda Menor	Um semitom
Segunda Maior	Dois semitons
Terça Menor	Três semitons
Terça Maior	Quatro semitons
Quarta Justa	Cinco semitons
Quarta Aumentada	Seis semitons
Quinta Diminuta	Seis semitons
Quinta Justa	Sete semitons
Quinta Aumentada	Oito semitons
Sexta Menor	Oito semitons
Sexta Maior	Nove semitons
Sétima Menor	Dez semitons
Sétima Maior	Onze semitons
Oitava Justa	Doze semitons

A Figura 4 mostra as sete classes de notas e seus intervalos. Observamos que, entre *mi-fá* e *si-dó*, há apenas meio tom (um semitom) de diferença.

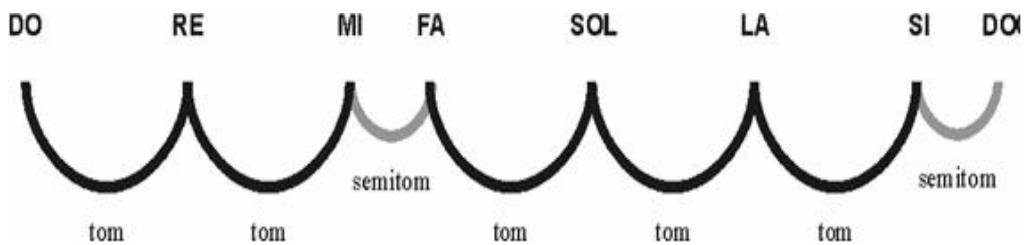


Figura 4 – Diferenças de altura entre as **notas** na escala musical ocidental. Fonte: Mello (2010)

2.1.3 Acidentes

Acidentes são modificadores que ao serem adicionados a uma nota altera sua entonação, acrescentando ou diminuindo semitons. São cinco ao todo (CARDOSO; MASCARENHAS, 1996), a saber:

- **Bemol (b)** - Diminui um semitom à nota;
- **Sustenido (#)** - Acrescenta um semitom à nota;
- **Dobrado bemol (bb)** – Diminui a entonação da nota em um tom;
- **Dobrado sustenido (x ou ##)** – Acrescenta um tom à nota;
- **Bequadro (♩)** – Recupera a entonação anterior da nota;

A livre utilização destes modificadores pode conduzir a dificuldades na correta nomeação das notas. Por exemplo, se aplicarmos o modificador Dobrado sustenido(x) estaremos nos referindo, por definição, a uma nota que se encontra a dois semitons (um tom) da nota que estamos alterando. Por exemplo, na Tabela 2 a nota *dó* Sustenido tem a mesma frequência que a nota *re* Bemol.

Na Tabela 2 encontramos todas as notas do sistema musical ocidental, incluindo os acidentes e suas respectivas frequências.

Tabela 2. Todas as notas com acidentes e suas frequências

NOTA	FREQUÊNCIA EM HERTZ	NOTA	FREQUÊNCIA EM HERTZ
dó	8.175799	fá# ou sol b	11.56233
dó# ou ré b	8.661957	sol	12.24986
ré	9.177024	sol# ou lá b	12.97827
ré# ou mi b	9.722718	lá	13.75
mi	10.30086	lá# ou si b	14.56762
fá	10.91338	si	15.43385

2.1.4 Escalas

Escala, segundo Schöenberg (1979), é a imitação do som posto horizontalmente, de forma sucessiva da esquerda para direita, ou seja, uma sequência de notas sucessivas define uma escala. Existem dois grupos de escalas, a saber: Escalas Maiores e Menores. As Escalas Menores se dividem em três grupos: as Escalas Menores Naturais, Harmônicas e Melódicas (VILAR, 2005).

Todas as escalas seguem um padrão bem definido de espaçamento entre as notas, por exemplo, a Escala Maior segue o padrão tom- tom- semitom; tom; tom- tom- semitom. O tom entre ponto e vírgula, indica a simetria na escala, separando duas partes tom- tom- semitom. A Figura 5 ilustra a Escala de dó Maior.

**Figura 5** – “Escala de *dó* Maior”. Fonte: VILAR (2005)

Cada escala possui uma Tônica associada, também chamada nota Fundamental, que dá nome à escala. “Outras notas que merecem atenção são a terça, que define o modo da escala (menor ou maior), a quinta, bastante utilizada nos acordes e a sensível, que provoca o efeito de conclusão característico da música tonal” (VILAR, 2005, p. 9). As escalas de um mesmo tipo seguem o mesmo espaçamento entre as notas, não importando qual nota seja tomada como tônica.

2.1.5 Cifra ABC

Cifra é a forma como nos referimos aos acordes, ou seja, como escrevemos as sequências de notas que surgem em uma música. A cifra ou notação ABC foi desenvolvida por Chris Walshaw em 1993 para melodias folclóricas da Europa Ocidental (MORAES, 2006), e descreve os acordes utilizando caracteres ASCII.

Neste trabalho usaremos as letras em negrito para nomear os acordes, como segue: **A** (*lá*), **B**(*si*), **C**(*dó*), **D**(*ré*), **E** (*mi*), **F**(*fá*), **G**(*sol*), seu uso irá depender da nota fundamental do acorde. Para indicar acordes maiores, usaremos apenas a letra em negrito, para os menores além da letra que identifica o acorde adicionaremos a letra ‘m’, por exemplo, **Cm** (acorde de Dó menor). Também faremos uso de números para indicar os intervalos que as notas fazem com a tônica, por exemplo, **C7m** (Dó com sétima menor). Os acidentes são adicionados após a letra que nomeia o acorde. Para diminuto usamos o símbolo ‘°’, para os acordes aumentado o símbolo ‘+’, por exemplo, **C#°**(Dó sustenido diminuto) e **C+** (Dó aumentado).

A seguir uma tabela com os acordes mais comuns (VILAR, 2005), utilizando como base o acorde de *dó*.

Tabela 3. Acordes mais freqüentes usado como base o acorde de *Dó*. Fonte: adaptado de VILAR (2005)

CIFRA	NOME DO ACORDE
C	Dó maior
C7	Dó com sétima maior
C7m	Dó com sétima menor
C79	Dó com sétima e nona
C°	Dó diminuto
C+	Dó aumentado

2.2 Acordes

Acorde é um conjunto de três ou mais notas, tocadas de forma simultânea, ou em intervalos (Tabela 1) curtos entre as notas. Quando existem pequenas diferenças de tempo entre as notas do acorde, dizemos que há uma decalagem nas notas.

Abordaremos neste trabalho dois tipos de acordes: Tríades, no qual possuem três notas distintas; e as tétrade que possuem quatro notas distintas e são formadas a partir das tríades. Em ambos os casos (tríades e tétrade), levaremos em consideração notas enarmônicas.

2.2.1 Tríade

Tríade é um acorde contendo um mínimo de três notas distintas. Ela é composta por uma nota fundamental (também chamada tônica que dá nome ao acorde), sua terça (menor ou maior) e sua quinta (justa, diminuta ou aumentada).

Existem quatro tipos de tríades possíveis, são elas:

- **Tríade Maior** - Formada por tônica, intervalo de terça maior com a tônica e intervalo de quinta justa com a tônica, por exemplo: as notas *sol* (Tônica), *si* (terça maior) e *ré* (quinta justa), formam a tríade de Sol maior (Figura 6).

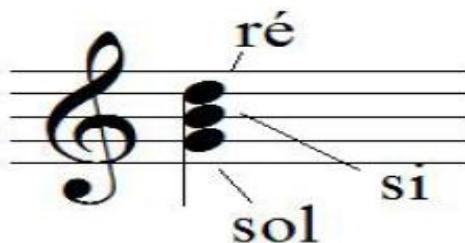


Figura 6– Tríade de Sol maior (G). Fonte: ARAGÃO (2011)

- **Tríade Menor** - Formada por tônica, intervalo de terça menor com a tônica e intervalo de quinta justa com a tônica (Figura 7), por exemplo: *dó* (Tônica), *mi bemol* (terça menor) e *sol* (quinta justa), formam a tríade de Dó menor (Cm).

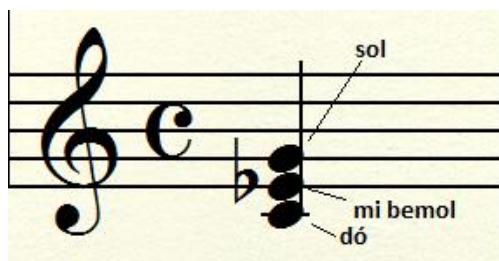


Figura 7– Tríade de Dó menor (Cm)

- **Tríade Diminuta** - Formada por tônica, intervalo de terça menor com a tônica e intervalo de quinta diminuta com a tônica (Figura 8), por exemplo: as notas *lá* (Tônica), *dó* (terça menor) e *mi bemol* (quinta diminuta), Formam a tríade de Lá diminuta.



Figura 8– Tríade de Lá diminuta (A°)

- **Tríade Aumentada** - Formada por tônica, intervalo de terça maior com a tônica e intervalo de quinta aumentada com a tônica (Figura 9), por exemplo: as notas *dó* (Tônica), *mi* (terça menor) e *sol sostenido* (quinta aumentada), Formam a tríade de Dó aumentado.

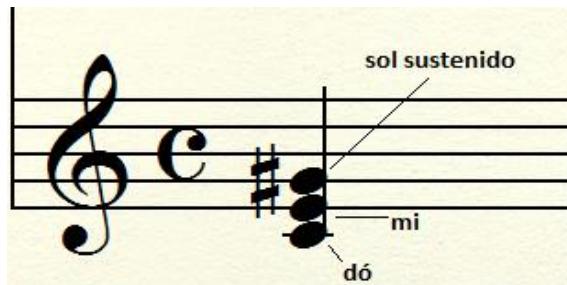


Figura 9– Tríade de Dó aumentado (C^+)

2.2.2 Tétrade

Tétrade é um acorde que possui um mínimo de quatro notas distintas. São formadas a partir das tríades, acrescentando uma nota a mais que forma um intervalo de sétima (maior ou menor) com a tônica do acorde. Possui as mesmas características das Tríades.

Dada uma tríade, para criarmos uma tétrade basta acrescentar uma nota que forma um intervalo de sétima com a tônica do acorde. Por exemplo, dada uma tríade de Dó maior (**C**), formada pelas notas: *dó*, *mi*, *sol*, para formarmos uma tétrade de Dó maior com sétima (**C7**), basta acrescentar um nota que possua intervalo de sétima maior com a tônica, no caso a nota *si*, então formamos **C7**.

3 TECNOLOGIA MIDI

Apresentamos nesta seção os conceitos do Protocolo MIDI 1.0 que serão utilizados neste trabalho. Para detalhes além do que apresentamos a seguir, indicamos: Moraes (2006), Messick (1997), Machado (2001), Ratton (1996) e MIDI (2011).

3.1 Introdução ao MIDI

O protocolo MIDI surgiu da necessidade de interconectar instrumentos musicais (mais especificamente teclados) de fabricantes diferentes, para que os músicos pudessem manipular vários instrumentos a partir de um *sequencer*⁵. Machado (2001, p. 24) afirma:

Esta tarefa não seria em si muito difícil de ser concretizada se o instrumentista utilizasse sintetizadores da mesma marca. Bastaria o fabricante disponibilizar nos sintetizadores um terminal para interconexão com outros sintetizadores.

Esta ideia ganhou adesão por alguns fabricantes, porém os músicos em geral, utilizam instrumentos fabricados por diversos fabricantes (MACHADO, 2001), o que torna esta saída incompleta.

Visando resolver este problema de comunicação, as empresas *Sequential Circuits*, *Oberheim* e *Roland* iniciaram pesquisas no assunto no início da década de 80, mais designadamente no ano de 81. Em 82 outras empresas aderiram à causa, a exemplo, a *Yamaha* (Messick, 1997) e em 83, David Smith da *Sequential Circuits*, em acordo com a *Roland*, divulgou a primeira versão do Protocolo MIDI 1.0.

Atualmente o protocolo MIDI especifica um esquema de conexões entre instrumentos - instrumentos e entre instrumentos - computador. Segundo Ratton (1996, p. 1) “MIDI (*Musical Instrument Digital Interface*) é um padrão de transmissão serial de dados, que permite a troca de informações entre instrumentos e equipamentos de aplicação musical.”

3.1.1 Protocolo MIDI 1.0

É uma linguagem para representação musical em código binário. A cada palavra que descreve uma ação musical é atribuído um código binário. Por ser projetado inicialmente para teclados, muitas das ações da tecnologia MIDI são orientadas a percussão.

⁵ “Equipamento que sincroniza e reproduz vários timbres de instrumentos musicais ao mesmo tempo” (MACHADO, 2001, p. 24)

Para reproduzir uma nota através da especificação MIDI, envia-se uma mensagem “*Note On*” para um canal específico, e atribui-se a essa nota um “*delta-time*” indicando quando a nota tocará, por outro lado para interromper a execução de nota envia-se uma mensagem “*Note off*” especificando qual nota irá parar sua reprodução. Há mensagens MIDI que podem escolher em qual instrumento tocar, misturar e distribuir sons nos canais estéreos, além de controlar vários aspectos de instrumentos musicais eletrônicos.

Canais MIDI são os locais por onde as mensagens MIDI são emitidas. Morais (2006, p.33), afirma:

O protocolo MIDI especifica dezesseis diferentes canais para transmissão de *streams* de mensagens diferentes. Diferentemente dos tocadores de fita, esses canais não são físicos, mas sim endereços eletrônicos diferentes que podem ser assinalados a diferentes dispositivos cujas mensagens trafegam sobre um mesmo barramento MIDI. Em um típico programa MIDI, um harpejo de violão pode ser passado por um determinado canal enquanto que o acompanhamento percussivo pode ser enviado em outro canal. Os dispositivos MIDI que estiverem habilitados em cada um desses canais irão interpretar as mensagens e executá-las de forma que se ouvirá o violão e a percussão ao mesmo tempo, em harmonia. Apesar de possuir um número fixo de canais, um programa MIDI pode utilizar quantos canais forem necessários, contanto que possua um esquema *multiline* de *interface*.

Com o advento do protocolo MIDI, surgiram vários benefícios para os músicos, elencamos os seguintes:

- Separa a unidade controladora da unidade geradora de som, permitindo que se utilize apenas um controlador para vários equipamentos;
- Possibilita a existência de vários tipos de controladores de som;
- Possibilita a comunicação e o controle bidirecional entre equipamentos e computadores através dos mais diversos *softwares*;
- *Softwares* são independentes do dispositivo a ser controlado, mas também é possível a implementação de *softwares* específicos a um dispositivo;
- Pode ser utilizado em equipamentos não-musicais assim como para controle e sincronismo;
- Pode-se trocar dados entre os mais diversos equipamentos.

A tecnologia MIDI se divide em *hardware* e formato de dados (lógico). No que tange *hardware*, ressaltamos os seguintes aspectos:

- a) **Interface MIDI** - Equipamento físico que permite dois sistemas ou equipamentos diferentes se comunicarem através de conectores padrão.

- b) **Dispositivo MIDI** - Dá-se nome de Dispositivo MIDI, qualquer dispositivo capaz de receber, enviar e interpretar o padrão MIDI.
- c) **Cabos MIDI**- São compostos por três fios, dois para envio e recebimento de mensagens, e outro utilizado como referência para interligação dos circuitos eletrônicos entre dois equipamentos MIDI. Os *plugs* devem ser do tipo DIN (macho) de cinco pinos em 180 graus (MIDI, 2011).
- d) **Conectores MIDI**- Os conectores são tipo DIN de 5 pinos, conforme especificação (MIDI, 2011, p.4). Segundo Ratton (1996, p. 1),

Os conectores do equipamento devem ser do tipo DIN de 5 pinos (fêmea) dispostos em 180 graus, e montados em painel. Os conectores devem ser identificados como MIDI IN (entrada), MIDI OUT (saída) e, opcionalmente, MIDI THRU (repetição da entrada). São recomendados conectores *Switchcraft 57 GB5F*.

- e) **Portas MIDI**- local onde são conectados os cabos MIDI. São três no total, utilizadas para envio, recebimentos e retransmissão de mensagens MIDI. As portas são descritas a seguir:
 - **MIDI IN**- Recebe as informações vinda de outro dispositivo MIDI. Apenas recebe, não é capaz de enviá-las.
 - **MIDI OUT**- Envia mensagens de um dispositivo MIDI para outro. Ela só envia mensagens, não é capaz de recebê-las.
 - **MIDI THRU**- Retransmite a mensagem recebida na porta MIDI IN, ou seja, funciona como uma extensão, uma cópia, da entrada MIDI IN. Assim, MIDI THRU é uma porta de saída.

No tocante à parte lógica (formato de dados) do protocolo MIDI, destacamos as mensagens MIDI e Padrão de Arquivos MIDI. A seguir em detalhes:

3.1.2 Mensagens MIDI

As mensagens são compostas por palavras de 10 *bits* e trocadas entre dispositivos MIDI, utilizando as portas adequadas. Elas podem ser mensagens de execução de nota ou comando de configuração de cada dispositivo, por exemplo, a mensagem ‘*Note on*’ especifica uma nota a ser tocada, já a mensagem ‘*Stop*’ interrompe a execução de um *sequencer*. Na Tabela 4 estão indicados os nomes das principais mensagens MIDI e suas respectivas ações.

Messick (1997, p. 14) afirma: “Cada mensagem MIDI, independente de seu tamanho, contém um *byte* de *status* (especificando qual o seu tipo de evento) e zero ou mais

bytes de dados”, por tanto as mensagens MIDI são do tipo *multi-bytes* contendo um *byte* de status (*status byte*) acompanhado por *bytes* de dados (*data bytes*). As exceções a essa regra são as mensagens de Tempo Real e as de Sistema Exclusivo (MIDI, 2011).

As mensagens MIDI são discretizadas em duas categorias principais, são elas: as Mensagens de Canal (*Channel Messages*) e as Mensagens de Sistema (*System Messages*) (RATTON, 1996).

- a) **Mensagens de Canal** – São mensagens endereçadas para equipamentos sintonizados no mesmo canal. São identificadas pelos quatro *bits* mais significativos do *byte* de *status*. Dividem-se em dois tipos, a saber:
 - **Mensagens de Voz (Voice Messages)** - Controlam os geradores de sons dos equipamentos e são transmitidas pelo canal de voz (*Voice Channel*);
 - **Mensagens de Modo (Mode Message)** - Definem como o dispositivo deve reagir as Mensagens de Voz. As Mensagens de Modo são transmitidas pelo Canal Básico (*Basic Channel*).
- b) **Mensagens de Sistemas (System Messages)** - São mensagens que não utilizam canal MIDI. Estas por sua vez, se dividem em mensagens comuns (*System Common Messages*), Mensagens de Tempo Real (*System Real-Time Messages*) e Mensagens Exclusivas (*System Exclusive Messages*).
 - **Mensagens Comuns** – Mensagens de sistemas que são comuns a todos os equipamentos de uma cadeia (MIDI, 2011). Por exemplo, a mensagem ‘*Start*’(ver Tabela 4).
 - **Mensagens de Tempo Real** – Mensagens que podem ser enviadas a qualquer momento, mesmo entre *bytes* de outras mensagens. Segundo Ratton (1996, p. 10), “As Mensagens em Tempo Real tem por finalidade sincronizar todos os equipamentos de um sistema operando em tempo real”. Por exemplo, a mensagem de ‘*reset*’(ver Tabela 4).
 - **Mensagens Exclusivas** – Mensagens que servem para transportar informações dos dispositivos que está sendo utilizado. São definidas pelos fabricantes dos equipamentos (um código na mensagem identifica o fabricante), porém os equipamentos são os responsáveis pelas suas interpretações e execuções. O padrão recomenda que os fabricantes divulguem suas documentações para que todos os

dispositivos interpretem suas mensagens. Caso um equipamento não reconheça um código de um fabricante, os *bytes* de dados das mensagens serão ignorados.

Tabela 4. Principais mensagens MIDI

MENSAGEM MIDI	AÇÃO
Note on	Informações da nota a ser executada pelo sintetizador
Note off	Indica ao sintetizador que a nota não está mais sendo executada ou pressionada (teclado)
Polyphonic key pressure	Indica qual foi a nota pressionada, o canal e a pressão exercida pelo instrumentista
Channel Pressure	Indica a pressão “média” de todas as notas executadas em um determinado canal MIDI
Control Change	Informa a um dispositivo específico a intensidade exercida sobre um pedal ou mesmo se ele está sendo pressionado ou não.
Pitch Bend	Informa a distorção de altura (<i>pitch</i>) sobre uma determinada nota
Program Change	Mensagem que contém um <i>byte</i> para escolha de canal e outro para a escolha de efeito. Indica ao dispositivo que ele deve, por exemplo, mudar de um instrumento para outro durante a execução da melodia. Também indica certos tipos de efeito sobre um canal como reverberação.
Bank Select	Mesma funcionalidade da mensagem ‘ <i>Program change</i> ’. Porém mais utilizada quando o dispositivo possui mais de 128 programas.
Local/ Remote keyboard Control	Desconecta um teclado de seu dispositivo sintetizador. Permite que o sintetizador seja controlado por outro dispositivo externo ao sistema enquanto o teclado continua enviando mensagens de dados.
All Notes off	Mensagem de emergência que para todas as notas que estão sendo executadas.
Reset all controllers	Retorna todos os controladores para o estado inicial.
ModeSelect	Seleciona o modo MIDI a ser utilizado

Start	Gerada quando um sequenciador ou máquina de percussão é iniciado.
Stop	Interrompe a execução do sequenciador ou máquina de percussão.
Continue	Continua a execução do sequenciador ou máquina de percussão.

3.1.3 Os arquivos SMF (*Standard MIDI File*)

Passado a etapa de interconexão entre os instrumentos, surge um problema: como armazenar os eventos realizados pelos músicos, para serem reproduzidos posteriormente em outros instrumentos? Machado (2005, p.35) afirma:

Se cada fabricante utilizasse uma forma particular para armazenar as músicas, apenas eles conseguiriam lê-las. Seria como se um japonês escrevesse uma carta para um brasileiro que não entendesse japonês. Neste caso ele não conseguiria entender o que o primeiro estaria querendo lhe dizer. Assim surgiram os padrões, os formatos **SMF** (*Standard MIDI Files*).

Os SMF são uma padronização para se registrar em arquivos os eventos executados pelos músicos (notas tocadas), para posteriormente serem executadas por um *sequencer*, independente de fabricante. São organizados em blocos, chamados *chunks*. Todo *chunk* possui a mesma estrutura e são identificados por uma *string* de quatro caracteres.

Neste trabalho, estamos interessados em recuperar as notas gravadas nos SMF, por tanto, é necessário saber como os SMF gravam as notas.

Para que uma nota seja executada por um dispositivo MIDI, é necessário que haja uma mensagem de ‘*Note on*’, na qual é informado, dentre outras coisas, o *Delta-time*, ou seja, o instante em que a nota será tocada (caso se queira que a nota seja tocada instantaneamente o *Delta-time* receberá valor zero) e a frequência da nota a ser tocada, por exemplo, se queremos que a nota *dó* seja executada em um SMF instantaneamente, devemos gravar os seguintes valores:

...

00 → *Delta-time* igual a zero;
 3C → Frequência da nota *dó* em hexadecimal.

...

Machado (2005, p.38-39) descreve um algoritmo (Figura 10) para gravar as notas em um SMF. A seguir mostraremos o passo-a-passo para executar este algoritmo:

1. Enviar uma mensagem de um a quatro *bytes*, com um *Delta-Time* informando quando a nota deverá começar a tocar. No caso da nota começar instantaneamente, este *Delta-Time* deverá ser zero;
 - `sendMessage(message.setMessage(Note on, delta time));`
2. Enviar uma mensagem (um *byte*) indicando qual dos 16 canais deverá ativar (tocar) a nota desejada;
 - `Synthesizer.getChannels(número);`
3. Enviar uma mensagem (um *byte*) com o evento que informe qual nota deverá ser ativada, sua intensidade (volume) e duração;
 - `Channel.noteOn(nota, volume, duração);`
4. Enviar uma mensagem (um *byte*) com o evento de desativar nota;
 - `SendMessage(message.setMessage(Note off));`
5. Enviar uma mensagem (um *byte*) contendo o código da nota a ser desativa, com o volume desta nota (pode ser qualquer valor, já que a nota vai ser desativada);
 - `Channel.noteOff(nota, volume);`

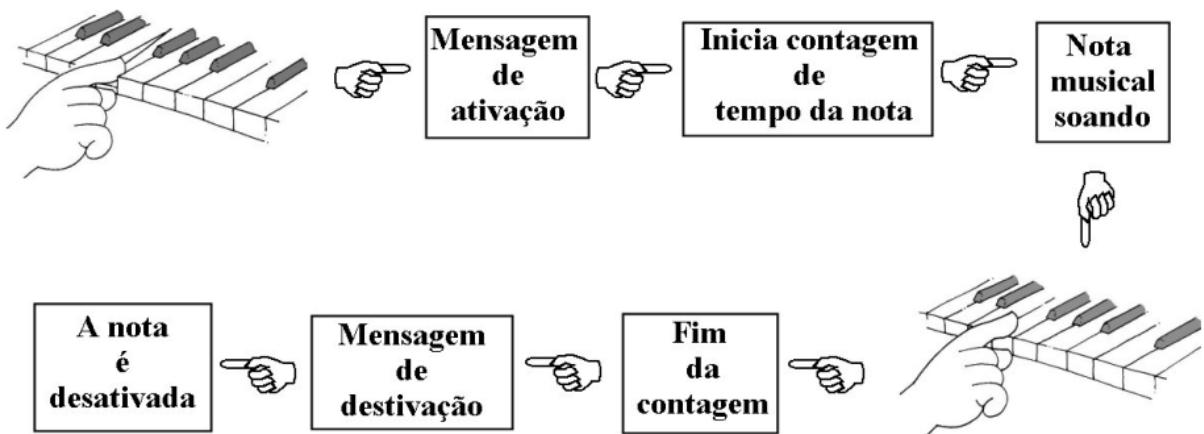


Figura 10– Sequência de passos para gravar as notas em um SMF. Fonte: Machado (2005)

4 LINGUAGENS FORMAIS E AUTÔMATOS

Apresentamos, nesta seção, os conceitos de Linguagens Formais e Autômatos que serão utilizados neste trabalho. Para detalhes além do que apresentamos a seguir, indicamos: Menezes (2008), Rangel (2001) e Hopcroft, Motwani e Ullman (2002).

4.1 Fundamentos de Linguagens Formais

A teoria das Linguagens Formais foi desenvolvida na década de 50, e possuía como foco o estudo das linguagens naturais (MENEZES, 2008). No entanto, percebeu-se que seu estudo nas linguagens ditas artificiais (Linguagem de Programação, por exemplo) era muito importante, principalmente do ponto de vista sintático (MENEZES, 2008).

O uso de Linguagens Formais na computação é vasto, Menezes (2008) apresenta várias aplicações da teoria das Linguagens Formais, dentre as quais encontramos:

- Análise léxica e sintática de linguagens de programação;
- Modelagem de circuitos lógicos;
- Modelagem de sistemas biológicos;
- Aplicação em sistemas de animação;
- Hipertexto e hipermídia;
- Tratamento de linguagens não-lineares;

A utilização das Linguagens Formais está associada a aspectos sintáticos, ou seja, à verificação de se estar ou não de acordo com regras de formação. Teoricamente, não existe ‘palavras com erros (erradas)’, e sim palavras que pertencem ou não à linguagem em questão.

Os Formalismos podem ser classificados, segundo Menezes (2008), em:

I. Operacional

Um Formalismo Operacional, também chamado Reconhecedor, executa a análise de uma entrada dada e verifica se a mesma pertence ou não a linguagem em questão. Podemos citar alguns exemplos deste Formalismo: Autômatos Finitos (determinísticos e não-determinísticos), Autômatos Pilha e Máquina de Turing.

II. Axiomático

Um Formalismo Axiomático, também chamado Gerador, possui regras embutidas que permitem gerar palavras pertencentes à linguagem em questão. Alguns exemplos deste

Formalismo são: Gramática Regular, Gramática Livre de Contexto, Gramática Sensível ao Contexto e Gramática Irrestrita.

III. Denotacional

Um Formalismo Denotacional, também chamado Funcional, define um padrão no qual as palavras podem ser inferidas na linguagem. “Este formalismo é frequentemente denominado, de forma não muito precisa, formalismo gerador” (MENENZES, 2008, p.3). Como exemplo deste Formalismo, podemos citar as Expressões Regulares.

Abordaremos neste trabalho o Formalismo Operacional, mais especificamente a Teoria dos Autômatos. Nas seções seguintes serão abordados conceitos sobre Teoria dos Autômatos.

4.1.1 Sistema de Estados Finitos

Sistemas de Estados Finitos são modelos matemáticos que se baseiam em um conjunto finito de estados, entradas e saídas. Em cada estado a informação guardada serve apenas para decidir para qual estado seguir. Menezes (2008, p.42) afirma, “é um modelo matemático de sistema com entradas e saídas discretas. Pode assumir um número finito e predefinido de estados”.

Várias situações cotidianas podem ser modeladas com Sistemas de Estados Finitos, em Menezes (2008, p.43) encontramos várias aplicações:

- Elevadores - Trata-se de um sistema que não memoriza informação de estados anteriores. Cada estado sumariza as informações ‘andar corrente’ e ‘direção do movimento’;
- Analisadores Léxicos e Processadores de Textos - Trata-se de um sistema de estados finitos. “Cada estado, basicamente, memoriza a estrutura do prefixo da palavra em análise” Menezes (2008, p.43);

Em Hopcroft, Motwani e Ullman (2002, p.29) encontramos mais situações modeláveis com Sistemas de Estados Finitos como, por exemplo, um interruptor (liga/desliga), em que cada estado mantém informação do *status* do sistema (ligado ou desligado). Na Figura 11 vemos os estados e as possíveis mudanças de estado (ações de transição)



Figura 11– Sistema de Estados Finitos que modela um interruptor liga/desliga

Os Sistemas de Estados Finitos se dividem segundo Menezes (2008) em: Composição Sequencial, Concorrente e Não- determinística.

- a) **Sequencial** - Cada transição (mudança de estado) possui no máximo uma possibilidade de estado sucessor. Dado um estado e um símbolo da entrada, só existe um estado futuro ao atual.
- b) **Concorrente** - Os estados são executados de forma independente. A ordem de execução dos estados não é considerada, diferentemente da composição sequencial;
- c) **Não-determinística** - Cada transição (mudança de estados) possui zero ou mais possibilidades de estados sucessores. Dado um estado e um símbolo da entrada, podem existir vários estados ao atual. Esta composição se divide em externa e interna.
 - a. **Interna** - O próximo estado é decidido no próprio sistema, de forma aleatória;
 - b. **Externa** - A escolha do próximo estado se dá externamente ao sistema;

4.2 Teoria dos Autômatos

A teoria dos Autômatos surgiu por volta das décadas de 40 e 50, inicialmente voltada para a tentativa de se modelar as funções do cérebro humano, porém seus estudos se mostraram extremamente úteis na computação (HOPCROFT; MOTWANI; ULLMAN, 2002). São inúmeras as aplicações de Autômatos na computação, podemos citar como exemplos:

- Analisador léxico de um compilador;
- Reconhecedor de palavras em textos;
- Processamentos de textos;

Antes de abordar os conceitos de Autômatos são necessárias definições, como por exemplo, definir Alfabeto, Palavra e Linguagem. As definições a seguir tomam como base a noção de caractere, por tanto não requer definição formal.

Definição 1 - Alfabeto

Alfabeto é um conjunto de caracteres, finito, denominado Σ .

Decorre desta definição, que um conjunto infinito não forma um alfabeto, porém um conjunto vazio forma um alfabeto. Alguns exemplos de alfabetos:

- O conjunto $\Sigma = \{a, e, i, o, u\}$ forma o alfabeto das vogais;
- O conjunto $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ forma o alfabeto dos dígitos;

Definição 2 - Palavra

Palavra w , de um alfabeto Σ é uma sequência de caracteres que pertencem ao alfabeto. Denotamos ' ϵ ' como a palavra vazia. Alguns exemplos de palavras, dado o alfabeto $\Sigma = \{a, b\}$ são:

- $w = ab$;
- $w = aaaa$.

O conjunto de todas as palavras de um alfabeto Σ é denotado Σ^* , com $\Sigma^+ = \Sigma^* - \{\epsilon\}$ significando o conjunto de todas as palavras exceto a palavra vazia.

Subpalavra, prefixo e sufixo são definidos como segue:

- a) **Subpalavra** de uma palavra w é qualquer sequência de caracteres contido em w .
- b) **Prefixo** de uma palavra w é qualquer subpalavra inicial de w .
- c) **Sufixo** de uma palavra w é qualquer subpalavra final de w .

Por exemplo, dado um alfabeto $\Sigma = \{1, 0\}$, e uma palavra $w = 01001$ temos:

- Conjunto de Sufixos = $\{\epsilon, 1, 01, 001, 1001, 01001\}$;
- Conjunto de Prefixos = $\{\epsilon, 0, 01, 010, 0100, 01001\}$;
- Algumas subpalavras = $\{\epsilon, 01001, 00, 10, 100\}$;

Definição 3 - Tamanho de uma palavra

O tamanho de uma palavra w é o número de caracteres que ela possui. Escrevemos $|w|$ para denotar o tamanho de w . Formalmente o tamanho de uma palavra w de um alfabeto Σ , é definido com uma função de assinatura $\|: \Sigma^* \rightarrow \mathbb{N}$. Por exemplo, dado $w = aaab$, temos $|w| = 4$.

Definição 4 - Concatenação

A concatenação de duas palavras w, x de um mesmo alfabeto Σ , com tamanhos m, n respectivamente, é uma função não comutativa \circ , com assinatura $\circ: \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$, que recebe

como argumento duas palavras, por exemplo, w e x e retorna como saída uma palavra c da forma wx , com $|c|=m+n$.

A função \circ adiciona todos os caracteres de uma palavra ao final da outra. Escrevemos $x^{\circ}y$, ou simplesmente xy , para a concatenação das palavras x e y . Ressaltamos algumas propriedades:

1. $x^{\circ}y \neq y^{\circ}x$ (Não comutatividade);
2. $(x^{\circ}y)^{\circ}t = x^{\circ}(y^{\circ}t)$ (Associatividade);
3. $x^{\circ}\epsilon = \epsilon^{\circ}x = x$ (Existência de elemento neutro);

Alguns exemplos dado o alfabeto $\Sigma = \{a, b\}$:

- Se $w=ab$ e $x=abb$ então $w^{\circ}x=ababb$;
- Se $w=bbb$ e $x=\epsilon$ então $w^{\circ}x=bbb$.

A operação $(w^{\circ}w)^{\circ}w$, que nos denota a concatenação sucessiva da palavra w , escreveremos w^n , com $n \in \mathbb{N}$, em que n representa o número de concatenações. Definimos indutivamente a concatenação sucessiva a partir de 1 e 2, como segue:

1. w^0 é uma concatenação e resulta $w^0 = \epsilon$;
2. $w^n = ww^{n-1}$, com $n > 0$.
 - Exemplo de aplicação: $w^3 = ww^2 = www^1 = wwww^0 = www\epsilon$ (usando a propriedade 3), temos $w^3 = www$.

Definição 5 - Linguagem Formal

Linguagem Formal L , ou simplesmente Linguagem L sobre um alfabeto Σ é dito como um subconjunto de Σ^* , ou seja, um conjunto de palavras, tal que $L \subseteq \Sigma^*$. Denotamos o conjunto de todas as linguagens de um alfabeto Σ , como 2^{Σ^*} . Alguns exemplos de Linguagens dado o alfabeto $\Sigma = \{a,b\}$.

- $L = \{w / w$ possui o número pares de ‘a’};
 - Algumas palavras de L , $\{aa, aaaaa, aaaaaaa\}$;
- $L = \{w / w$ possui sempre possuir ‘aa’ como subpalavra};
 - Algumas palavras de L , $\{aa, abaabba, bbaa\}$;
- $L = \{w / w$ possui sempre possuir ‘bb’ como sufixo e ‘a’ como prefixo};
 - Algumas palavras de L , $\{abb, aabbabb, aabbabbaabb\}$;

4.2.1 Autômatos Finitos

Autômatos Finitos são formalismos matemáticos, do tipo Operacional ou Reconhecedor, cujo modelo de composição é Sequencial. É utilizado para decidir se uma dada palavra w pertence ou não a uma Linguagem L. Há vários motivos para seu estudo. Segundo Hopcroft, Motwani e Ullman (2002, p.5) “Os autômatos são essenciais para o estudo dos limites da computação”.

Em relação à complexidade de algoritmos, os Autômatos Finitos estão incluídos nas classes de algoritmos mais eficientes em termos de tempo de processamento. Se um dado problema pode ser solucionado por um Autômato Finito, tal solução será ótima.

Os Autômatos Finitos são um dos formalismos abordados pelo estudo das Linguagens Regulares ou Linguagem Tipo 3, outros são: Expressão Regular e Gramática Regular. Prova-se que estes três formalismos são equivalentes, porém tal prova foge do escopo deste trabalho.

De acordo com a Hierarquia de Chomsky, as Linguagens Regulares são as mais simples e com menos poder de expressão, porém de grande eficiência e de fácil implementação (MENENZES, 2008).

Segundo Rangel (2001, p. 35), os Autômatos Finitos são constituídos por três partes (ver Figura 12), a saber:

- Fita de Entrada** - Local onde a entrada é colocada e processada;
- Unidade de Controle** - Que ler os símbolos de entrada da Fita, de forma sequencial da esquerda para a direita.
- Função de Transição** - Define quando a Unidade de Controle irá para o próximo estado.



Figura 12– Representação Autômato Finito. Fonte: Rangel (2001)

Neste trabalho apresentaremos os Autômatos Finitos Determinísticos, Não-Determinísticos e com Movimentos Vazios, dos quais escolheremos um para desenvolver o reconhecedor dos acordes ortograficamente corretos.

4.2.2 Autômato Finito Determinístico

Autômato Finito Determinístico, ou simplesmente AFD é um Sistema de Estados Finitos do tipo Operacional ou Reconhecedor, com modelo de composição Sequencial. Ele é definido sobre um Alfabeto (Definição 1), e formado por um conjunto de estados, que representam a situação atual do modelo. Dentre esses estados, um em especial chamamos estado inicial, que representa o início da operação de reconhecimento. Além dele, há um subconjunto dos estados, denotado estados de aceitação ou finais. Uma função de transição ou programa que, dado um símbolo do Alfabeto e um estado, define para qual estado seguir de forma determinística, ou seja, para cada aplicação da função programa sobre um estado e um símbolo, só há uma possibilidade de estado futuro.

As linguagens que são reconhecidas por um AFD são ditas regulares, portanto, podemos utilizar AFD's para verificar se uma dada linguagem é Regular, bastando criar um AFD que parte de um estado inicial, e após sucessivas aplicações da função de transição, chegue a um estado de aceitação. Ressaltamos que não é possível, com o uso de Autômatos, demonstrar que uma linguagem não é regular, para isso há outras abordagens, como o lema do Bombeamento (ver Hopcroft, Motwani e Ullman (2002, p.135)). Formalmente um AFD é definido como segue:

Definição 6 - Autômato Finito Determinístico (AFD)

Um Autômato Finito Determinístico M sobre um alfabeto Σ é formalmente definido pela tupla $M = \langle \Sigma, Q, F, \delta, q_0 \rangle$, em que:

- Σ é o alfabeto dos caracteres de entrada ou apenas alfabeto;
- Q é o conjunto finito dos estados do Autômato M;
- F é o conjunto de estados finais ou de aceitação, com $F \subseteq Q$;
- q_0 é um elemento de Q , dito estado inicial ($q_0 \in Q$);
- δ é uma função parcial dita função de transição. δ recebe como argumento um estado pertencente a Q , um caractere pertencente ao alfabeto e retorna um estado de Q .

Formalmente:

$$\delta: Q \times \Sigma \rightarrow Q$$

Portanto, dado os estados $q, p \in Q$ e um caractere $a \in \Sigma$, dizemos que $\delta(q,a)=p$ é uma transição do Autômato M do estado q para o estado p ao ler o caractere a. A computação de um autômato ocorre da esquerda para a direita. Segue um exemplo de um AFD, dado o alfabeto $\Sigma=\{p,r,i,s,c,i,l,a\}$:

Exemplo 1- Um AFD M que reconheça uma linguagem L que possui somente a palavra ‘priscila’, ou seja, $L=\{w / w = ‘priscila’\}$, então temos: $M = <\Sigma, Q, F, \delta, q_0>$, em que:

- Alfabeto é o conjunto $\Sigma=\{p,r,i,s,c,i,l,a\}$;
- Os estados são $Q = \{Q_0, P, PR, PRI, PRIS, PRISC, PRISCI, PRISCIL, PRISCILA\}$;
- Conjunto de estados finais $F=\{PRISCILA\}$;
- Estado inicial $q_0 = Q_0$;
- $\delta(Q_0, p) = P$, $\delta(P, r) = PR$, $\delta(PR, i) = PRI$, $\delta(PRI, s) = PRIS$, $\delta(PRIS, c) = PRISC$, $\delta(PRISC, i) = PRISCI$, $\delta(PRISCI, l) = PRISCIL$, $\delta(PRISCIL, a) = PRISCILA$.

Um AFD é geralmente exibido em modo gráfico através do uso de grafos direcionados. O mapeamento ocorre da seguinte forma: os nós representam o conjunto dos estados (o rótulo do nó representa o nome do estado). As arestas equivalem às transições (para cada aresta um rótulo associado que corresponde à entrada corrente do Autômato). Um triângulo apontando para o estado inicial; Por fim, em todos os estados de aceitação, o nó do grafo terá um contorno. A Figura 13 apresenta o autômato M do Exemplo 1 em modo gráfico.

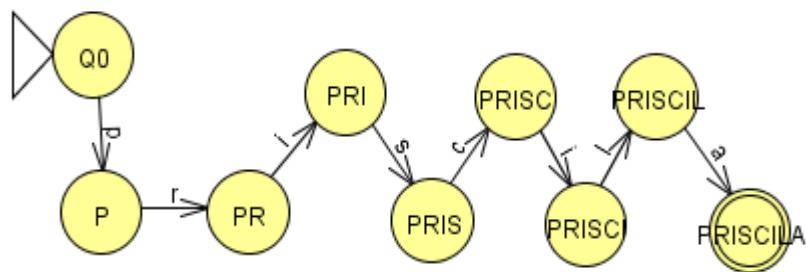


Figura 13– AFD que reconhece uma linguagem que possui somente a palavra ‘priscila’

Enfatizamos que a função δ recebe como argumento um estado e um caractere, portanto para que o Autômato trate de palavras é necessário estender a função de transição δ .

Definição 6.1 – Função programa estendida ou função de transição estendida

Dado um Autômato $M = \langle \Sigma, Q, F, \delta, q_0 \rangle$, Denotamos como função programa estendida de δ a função δ^* . Formalmente:

$$\delta^*: Q \times \Sigma^* \rightarrow Q$$

A função programa estendida δ^* consiste na sucessiva aplicação de δ a partir do estado inicial. Se após essas aplicações chega-se a um estado final, dizemos que a palavra lida pertence à linguagem em questão. Definimos δ^* indutivamente como:

- Caso base ($|w|=0$ ou $w=\varepsilon$)
 - $\delta^*(p, \varepsilon) = p;$
- Passo indutivo ($|w|=k+1$ ou $w=xa$)
 - $\delta^*(q, xa) = \delta(\delta^*(q, x), a)$, usando $\delta^*(q, x)$ como hipótese de indução ($|x|=k$ ou $w=x$).

Aplicando a função estendida ao Autômato M do Exemplo 1, temos:

$$\delta^*(Q_0, priscila) = \delta(\delta^*(Q_0, priscil), a)$$

$$\delta^*(Q_0, priscil) = \delta(\delta^*(Q_0, prisci), l)$$

$$\delta^*(Q_0, prisci) = \delta(\delta^*(Q_0, prisc), i)$$

$$\delta^*(Q_0, prisc) = \delta(\delta^*(Q_0, pris), c)$$

$$\delta^*(Q_0, pris) = \delta(\delta^*(Q_0, pri), s)$$

$$\delta^*(Q_0, pri) = \delta(\delta^*(Q_0, pr), i)$$

$$\delta^*(Q_0, pr) = \delta(\delta^*(Q_0, p), r)$$

$$\delta^*(Q_0, p) = \delta(\delta^*(Q_0, \varepsilon), p)$$

$$\delta^*(Q_0, \varepsilon) = Q_0, \text{ resolvendo todas recursões:}$$

$$\delta^*(Q_0, p) = \delta(Q_0, p) = P$$

$$\delta^*(Q_0, pr) = \delta(P, r) = PR$$

$$\delta^*(Q_0, pri) = \delta(PR, i) = PRI$$

$$\delta^*(Q_0, pris) = \delta(PRI, s) = PRIS$$

$$\delta^*(Q_0, prisc) = \delta(PRIS, c) = PRISC$$

$$\delta^*(Q_0, prisci) = \delta(PRISC, i) = PRISCI$$

$$\delta^*(Q_0, priscil) = \delta(PRISCI, l) = PRISCIL, \text{ por fim}$$

$$\delta^*(Q_0, priscila) = \delta(PRISCIL, a) = PRISCILA, \text{ como PRISCILA é um estado final, a palavra}$$

‘priscila’ foi reconhecida pelo Autômato Finito Determinístico M. Segue uma definição de

Linguagem Reconhecida por um Autômato:

Definição 6.2 - Linguagem Reconhecida

Formalmente, denotamos $L(M)$, a linguagem reconhecida por um AFD $M = \langle \Sigma, Q, F, \delta, q_0 \rangle$. O conjunto $L(M)$ é definido como:

$$L(M) = \{ w / \delta^*(q_0, w) \in F \}$$

Por tanto, o conjunto $L(M)$, são todas as palavras que a partir do estado inicial, e mudanças sucessivas de estado de acordo com a função de transição, chega-se a um estado que pertence ao conjunto de estados finais.

4.2.3 Autômato Finito Não Determinístico

Autômato Finito Não Determinístico, ou simplesmente AFN é um Sistema de Estados Finitos do tipo Operacional ou Reconhecedor, com modelo de composição não determinística. Assim como o AFD, Ele é definido sobre um Alfabeto (Definição 1), e formado por um conjunto de estados, que representam a situação atual do modelo. Dentre esses estados, denotamos um em especial como estado inicial, que representa o início da operação de reconhecimento. Além dele, há um subconjunto dos estados, denotado estados de aceitação ou finais. Uma função de transição que, dado um símbolo do Alfabeto e um estado, retorna uma lista de estados futuros, de forma não determinística, para cada aplicação da função programa sobre um estado e um símbolo, há uma lista de possibilidade de estados. Menezes (2008, p. 55) afirma:

Visto como uma máquina composta por fita, unidade de controle e programa, um Autômato Finito Não-Determinístico assume um conjunto de estados alternativos, como se houvesse uma multiplicação da unidade de controle, uma para cada alternativa, processando independentemente, sem compartilhar recursos com as demais. Assim, o processamento de um caminho não influí no estado, símbolo lido e posição da cabeça dos demais caminhos alternativos.

No entanto, o não-determinismo não aumenta a expressividade de reconhecimento do AFN em relação ao AFD. Embora em algumas linguagens construir um AFD seja bastante difícil, toda linguagem que pode ser reconhecida por um AFN é igualmente reconhecida por um AFD, ou seja, constituem formalismos equivalentes.

As linguagens que são reconhecidas tanto por AFN como AFD, são ditas regulares, por tanto, podemos utilizar AFN's para verificar se uma dada linguagem é regular, basta criar um AFN que parte de um estado inicial, e após sucessivas aplicações das funções de transição, chegue a um conjunto que possua algum elemento do conjunto dos estados finais. Formalmente um AFN é definido como segue:

Definição 7- Autômato Finito Não Determinístico (AFN)

Um Autômato Finito Não Determinístico N sobre um alfabeto Σ é formalmente definido pela tupla $N = \langle \Sigma, Q, F, \delta, q_0 \rangle$, em que:

- Σ é o alfabeto dos caracteres de entrada ou apenas alfabeto;
- Q é o conjunto finito dos estados do Autômato N ;
- F é o conjunto de estados finais ou de aceitação, com $F \subseteq Q$;
- q_0 é um elemento de Q , dito estado inicial ($q_0 \in Q$);
- δ é uma função parcial dita função de transição. δ recebe como argumento de entrada um estado pertencente a Q , um caractere pertencente ao alfabeto e retorna conjunto de todas as possíveis combinações de estados de Q , ou seja, o conjunto das partes de Q .

Formalmente:

$$\delta: Q \times \Sigma \rightarrow 2^Q$$

Portanto, dado o estado $q \in Q$ e um caractere $a \in \Sigma$, dizemos que $\delta(q, a) = P$, no qual $P = \{p_1, p_2, \dots, p_n\}$ e $\forall p_i \in P, p_i \in Q$ com $i \in \{1, 2, 3, \dots, n\}$, é uma transição do Autômato M do estado q para o subconjunto P ao ler o caractere a . A computação de um AFN ocorre da esquerda para a direita. Segue um exemplo de um AFN, dado o alfabeto $\Sigma = \{a, m, o, r\}$:

Exemplo 2- Um AFN M que reconheça uma linguagem L que tenha como sufixo a palavra ‘amor’, ou seja, $L = \{w / w = x^0 amor, \text{ com } x \in \Sigma^*\}$ então temos: $M = \langle \Sigma, Q, F, \delta, q_0 \rangle$, no qual:

- Alfabeto é o conjunto $\Sigma = \{a, m, o, r\}$;
- Os estados são $Q = \{Q_0, A, AM, AMO, AMOR\}$;
- Conjunto de estados finais $F = \{AMOR\}$;
- Estado inicial $q_0 = Q_0$;
- $\delta(Q_0, a) = \{Q_0, A\}$, $\delta(Q_0, m) = \{Q_0\}$, $\delta(Q_0, o) = \{Q_0\}$, $\delta(Q_0, r) = \{Q_0\}$, $\delta(A, m) = \{AM\}$, $\delta(AM, o) = \{AMO\}$, $\delta(AMO, r) = \{AMOR\}$;

De forma semelhante ao AFD, um AFN pode ser exibido de forma gráfica através de um grafo direcionado, seguindo o mesmo mapeamento feito para um AFD. A Figura 14 mostra o Autômato do Exemplo 2.

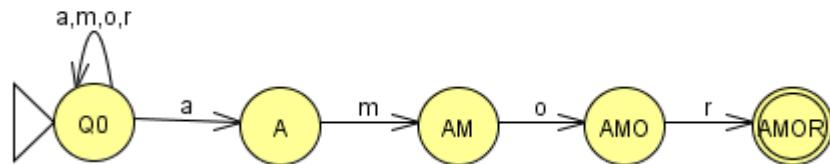


Figura 14– AFN que reconhece uma Linguagem com a palavra ‘*amor*’ como sufixo

Assim como no AFD, a função δ recebe como argumento um estado e um caractere, por tanto para que o Autômato trate de palavras é necessário estender a função de transição δ .

Definição 7.1 – Função programa estendida ou função de transição estendida

Dado um Autômato $N = \langle \Sigma, Q, F, \delta, q_0 \rangle$, Denotamos como função programa estendida de δ a função δ^* . Formalmente:

$$\delta^* : 2^Q \times \Sigma^* \rightarrow 2^Q$$

A função programa estendida δ^* consiste na sucessiva aplicação de δ a partir do estado inicial, se após essas aplicações chega-se a um subconjunto que possua algum elemento do conjunto de estados finais, dizemos que a palavra lida pertence à linguagem em questão. Definimos δ^* indutivamente como:

- Caso base ($|w|=0$ ou $w=\epsilon$)
 - $\delta^*(P, \epsilon) = P;$
- Passo indutivo ($|w|=k+1, k>0$ ou $w=xa$)
 - $\delta^*(P, xa) = \bigcup_{r \in \delta^*(P, x)} \delta(r, a)$, usando $\delta^*(P, x)$ como hipótese de indução ($|x|=k$ ou $w=x$).

A computação de $\delta^*(P, xa)$, se dá computando primeiro $\delta^*(P, x)$, que resulta em um conjunto, por exemplo $T = \{p_1, p_2, \dots, p_n\}$, em seguida computamos todas transições dos elementos de T que possuem transições com rótulo ‘ a ’, ou seja, $\delta(p_1, a) \cup \delta(p_2, a) \cup \dots \cup \delta(p_n, a)$.

Como exemplo, vamos verificar se a palavra ‘*amor*’ é reconhecida pelo Autômato N do Exemplo 2:

Inicialmente temos $\delta^*(\{Q_0\}, amor)$, então:

$$\delta^*(\{Q_0\}, amor) = \bigcup_{p \in \delta^*(\{Q_0\}, amo)} \delta(p, r)$$

$$\delta^*(\{Q_0\}, amo) = \bigcup_{p \in \delta^*(\{Q_0\}, am)} \delta(p, o)$$

$$\delta^*(\{Q_0\}, am) = \bigcup_{p \in \delta^*(\{Q_0\}, a)} \delta(p, m)$$

$$\delta^*(\{Q_0\}, a) = \bigcup_{p \in \delta^*(\{Q_0\}, \epsilon)} \delta(p, a)$$

$\delta^*(\{Q_0\}, \varepsilon) = \{Q_0\}$, aplicando em todas as recursões:

$$\delta^*(\{Q_0\}, a) = U_{p \in \{Q_0\}} \delta(p, a) = \delta(Q_0, a) = \{Q_0, A\}$$

$$\delta^*(\{Q_0\}, am) = U_{p \in \{Q_0, A\}} \delta(p, m) = \delta(Q_0, m) \cup \delta(A, m) = \{Q_0, AM\}$$

$$\delta^*(\{Q_0\}, amo) = U_{p \in \{Q_0, AM\}} \delta(p, o) = \delta(Q_0, o) \cup \delta(AM, o) = \{Q_0, AMO\}$$

$\delta^*(\{Q_0\}, amor) = U_{p \in \{Q_0, AMO\}} \delta(p, r) = \delta(Q_0, r) \cup \delta(AMO, r) = \{Q_0, AMOR\}$, como no conjunto gerado por $\delta^*(\{Q_0\}, amor)$ existe um elemento que pertence ao conjunto dos estados finais, a palavra ‘amor’ é reconhecida pelo Autômato N. Segue uma definição de Linguagem Reconhecida por um Autômato Finito Não Determinístico:

Definição 7.2 - Linguagem Reconhecida

Formalmente, denotamos $L(N)$, a linguagem reconhecida pelo AFN $N = <\sum, Q, F, \delta, q_0>$. O conjunto $L(N)$ é definido como:

$$L(N) = \{w / \delta^*(q_0, w) \cap F \neq \emptyset\}$$

Por tanto, o conjunto $L(N)$ são todas as palavras que a partir do estado inicial, e mudanças sucessivas de estado de acordo com a função de transição, chega-se a um conjunto cuja interseção com o conjunto dos estados finais é não nula.

4.2.4 Autômato Finito com ε -transições

A fim de se facilitar as construções e demonstrações utilizando Autômatos Finitos, criaram-se transições sem necessariamente um símbolo ser lido. Tais movimentos podem ser entendidos como um não-determinismo interno ao Autômato. Se algum Autômato permite tais movimentos, dizemos que este Autômato possui movimentos vazios, ou transições sem leitura de símbolo na fita (MENEZES, 2008). Hopcroft, Motwani e Ullman (2002, p.78) afirma: “[...] este novo recurso não expande a classe de linguagens que podem ser reconhecida por Autômatos Finitos, mas nos dá uma certa “conveniência de programação” adicional”. Por tanto o poder de reconhecimento permanece o mesmo, e ambos os formalismos são equivalentes. Prova-se que qualquer Autômato Finito com movimento vazio pode ser simulado por um AFD ou AFN, porém tal prova foge do escopo deste trabalho.

Um Autômato Finito com movimentos vazios ou AFN ε possui os mesmos elementos de um AFN, porém requer mudanças na função de transição. Segue uma definição formal de um AFN ε .

Definição 8- Autômato Finito com Movimentos vazios

Um Autômato Finito com Movimentos Vazios ou AFN ε M sobre um alfabeto Σ é formalmente definido pela tupla $M = \langle \Sigma, Q, F, \delta, q_0 \rangle$, em que:

- Σ é o alfabeto dos caracteres de entrada ou apenas alfabeto;
- Q é o conjunto finito dos estados do Autômato M;
- F é o conjunto de estados finais ou de aceitação, com $F \subseteq Q$;
- q_0 é um elemento de Q , dito estado inicial ($q_0 \in Q$);
- δ é uma função parcial dita função de transição. δ recebe como argumento de entrada um estado pertencente a Q , um caractere pertencente ao alfabeto ou a palavra vazia ε e retorna um conjunto de todas as possíveis combinações de estados de Q , ou seja, o conjunto das partes de Q . Formalmente:

$$\delta: Q \times \Sigma \cup \{\varepsilon\} \rightarrow 2^Q$$

Portanto, dado o estado $q \in Q$ e um caractere a ou $\varepsilon \in \Sigma$, dizemos que $\delta(q, a) = P$ ou $\delta(q, \varepsilon) = P$, no qual $P = \{p_1, p_2, \dots, p_n\}$ e $\forall p_i \in P, p_i \in Q$ é uma transição do Autômato M do estado q para o subconjunto P ao ler o caractere a ou não ler nenhum símbolo. A computação de um AFN ε ocorre da esquerda para a direita. Segue um exemplo de um AFN ε , dado o alfabeto $\Sigma = \{a, b\}$:

Exemplo 3- Um AFN ε M que reconheça uma linguagem L que possua b 's seguido de a 's, ou seja, $L = \{w / w = b^k a^n, \text{ com } k, n \geq 0 \text{ e } k \neq n\}$ então temos: $M = \langle \Sigma, Q, F, \delta, q_0 \rangle$, em que:

- Alfabeto é o conjunto $\Sigma = \{a, b\}$;
- Os estados são $Q = \{B, A\}$;
- Conjunto de estados finais $F = \{A\}$;
- Estado inicial $q_0 = B$;
- $\delta(B, b) = \{B\}$, $\delta(B, \varepsilon) = \{A\}$, $\delta(A, a) = \{A\}$.

De forma semelhante aos anteriores, um AFN ε pode ser exibido de forma gráfica através de um grafo direcionado, seguindo o mesmo mapeamento dos anteriores. A Figura 15 mostra o Autômato do Exemplo 3.

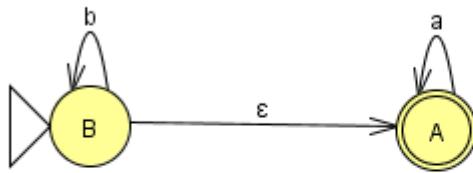


Figura 15– AFN ϵ que reconhece uma Linguagem com b 's seguido por a 's

Assim como os Autômatos Finitos Determinísticos e Não-Determinísticos, a função δ recebe como argumento um estado e um caractere, por tanto, para que o Autômato em questão trate de palavras é necessário estender a função de transição δ , porém nos AFN ϵ é necessário definir inicialmente computação vazia, ou seja, as transições em que temos como rótulo ϵ .

Definição 8.1- Computação vazia

Dado um AFN ϵ $M = <\Sigma, Q, F, \delta, q_0>$, Denotamos a função δ_ϵ computação vazia de M . δ_ϵ possui a seguinte assinatura:

$$\delta_\epsilon: Q \rightarrow 2^Q$$

É definida indutivamente como segue:

$\delta_\epsilon(p) = \{p\}$, se $\delta(p, \epsilon)$ é indefinida;

$\delta_\epsilon(p) = \{p\} \cup \delta(p, \epsilon) \cup (\bigcup_{s \in \delta(p, \epsilon)} \delta_\epsilon(s))$, caso contrário;

Portanto, dado um estado $p \in Q$ a computação vazia de p , consiste buscar todos os estados que são acessíveis por p através de transições vazias. Assim como na função de transição, a função δ_ϵ pode ser estendida para receber um conjunto de estados. A seguir definimos sua extensão.

Definição 8.2- Computação vazia estendida

A função computação estendida de δ_ϵ , denotamos por δ_ϵ^* , consiste em sucessivas aplicações de δ_ϵ . δ_ϵ^* possui a seguinte assinatura:

$$\delta_\epsilon^*: 2^Q \rightarrow 2^Q$$

É formalmente definida como segue:

$$\delta_\epsilon^*(P) = \bigcup_{q \in P} \delta_\epsilon(q).$$

Definido computação vazia, podemos estender a função transição δ de um AFN ϵ .

Sua definição é como segue:

Definição 8.3 – Função programa estendida ou função de transição estendida

Dado um AFN ϵ M = $\langle \sum, Q, F, \delta, q_0 \rangle$, Denotamos como função programa estendida de δ a função δ^* . Formalmente:

$$\delta^*: 2^Q \times \sum^* \rightarrow 2^Q$$

A função programa estendida δ^* consiste na sucessiva aplicação de δ a partir do estado inicial. Se após essas aplicações chega-se a um subconjunto que possua algum elemento do conjunto de estados finais, dizemos que a palavra lida pertence à linguagem em questão. Definimos δ^* indutivamente como:

- Caso base ($|w|=0$ ou $w=\epsilon$)
 - $\delta^*(P, \epsilon) = \delta_\epsilon^*(P);$
- Passo indutivo ($|w|=k+1, k>0$ ou $w=xa$)
 - $\delta^*(P, xa) = \delta_\epsilon^*(R)$, em que $R = \bigcup_{p \in \delta^*(P, x)} \delta(p, a)$, usando $\delta^*(q, x)$ como hipótese de indução.

A computação de $\delta^*(P, xa)$, se dá computando primeiro $\delta^*(P, x)$, que resulta em um conjunto, por exemplo $T = \{p_1, p_2, \dots, p_n\}$, em seguida computamos todas transições dos elementos de T que possuem transições com rótulo ‘a’, ou seja, $R = \delta(p_1, a) \cup \delta(p_2, a) \cup \dots \cup \delta(p_n, a)$ por fim computamos as transições vazias dos elementos de R.

Como exemplo, vamos verificar se a palavra ‘ba’ é reconhecida pelo Autômato N do Exemplo 3:

Inicialmente temos $\delta^*(\{B\}, ba)$, então:

$$\delta^*(\{B\}, ba) = \delta_\epsilon^*(R_1), \text{ no qual } R_1 = \bigcup_{p \in \delta^*(\{B\}, b)} \delta(p, a),$$

$$\delta^*(\{B\}, b) = \delta_\epsilon^*(R_2), \text{ e } R_2 = \bigcup_{p \in \delta^*(\{B\}, \epsilon)} \delta(p, b),$$

$$\delta^*(\{B\}, \epsilon) = \delta_\epsilon^*(\{B\}) = \delta_\epsilon(B),$$

$$\delta_\epsilon(B) = \{B\} \cup \delta(B, \epsilon) \cup (\bigcup_{s \in \delta(B, \epsilon)} \delta_\epsilon(s)), \text{ como } \delta(B, \epsilon) = \{A\}, \text{ desenvolveremos } \bigcup_{s \in \delta(B, \epsilon)} \delta_\epsilon(s):$$

$$U_{s \in \{A\}} \delta_\epsilon(s) = \delta_\epsilon(A) = \emptyset, \text{ então,}$$

$$\delta_\epsilon(B) = \{B, A\}, \text{ logo,}$$

$$\delta^*(\{B\}, \epsilon) = \{B, A\}, \text{ aplicando em } R_2 \text{ temos,}$$

$$R_2 = \bigcup_{p \in \{B, A\}} \delta(p, b) = \delta(B, b) \cup \delta(A, b) = \{A\}, \text{ por tanto,}$$

$$\delta^*(\{B\}, b) = \delta_\epsilon^*(R_2) = \delta_\epsilon^*(\{A\}) = \delta_\epsilon(A) = \{A\}, \text{ pois } \delta(A, \epsilon) \text{ é indefinido. Então temos,}$$

$$\delta^*(\{B\}, b) = \{A\}, \text{ aplicando ao } R_1, \text{ temos,}$$

$$R_1 = \bigcup_{p \in \{A\}} \delta(p, a) = \delta(A, a) = \{A\}, \text{ aplicando em } \delta_\epsilon^*(R_1), \text{ temos,}$$

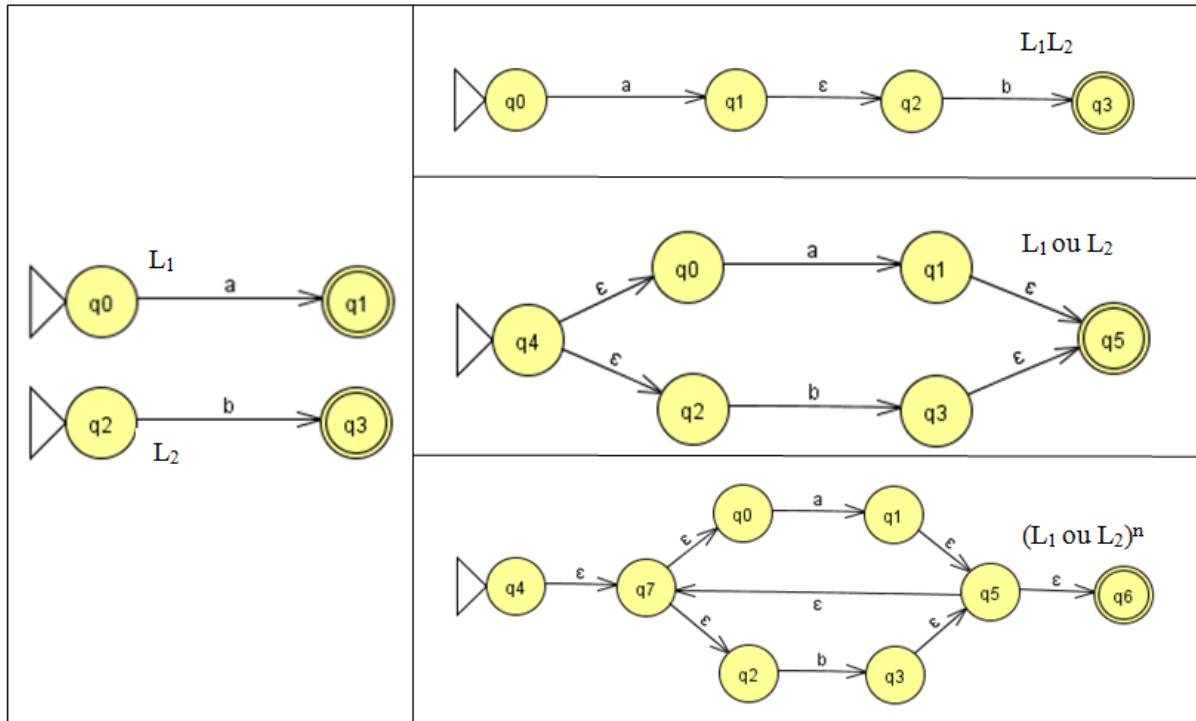
$$\delta_\epsilon^*(\{A\}) = \delta_\epsilon(A) = \{A\}, \text{ como } \delta(A, \epsilon) \text{ é indefinido. Concluindo,}$$

$\delta^*(\{B\}, ba) = \{A\}$, como $\{A\}$ tem interseção não nula com o conjunto de estados finais, dizemos que a palavra ‘ba’ é reconhecida pelo AFN ϵ .

A definição de linguagem reconhecida por um Autômato com ϵ -Transições é idêntica a Definição 7.2.

4.2.5 Composição de Autômatos

O método de composição de autômatos consiste em compor dois autômatos para construir outro. Para compor autômatos utilizamos regras bem definidas e comportadas (MENEZES, 2008). A composição se dá, da seguinte forma: dadas as Linguagens L_1 e L_2 reconhecidas pelos autômatos M_1 e M_2 , respectivamente. Para compor o Autômato que reconheça a Linguagem L_1L_2 (a linguagem L_2 concatenada no final de L_1) ou linguagem a que reconhece L_1 ou L_2 e L_1 ou L_2 repetidas vezes (reconhecer L_1 ou L_2 várias vezes, podendo alternar entre L_1 e L_2), se dá de acordo com a ilustração a seguir:



Como podemos visualizar, para reconhecer a linguagem L_1L_2 , conectamos o estado final do Autômato M_1 ao estado inicial de M_2 e mantemos como estado final o estado de M_2 . Para reconhecer L_1 ou L_2 , criamos dois estados a mais, um para ser o inicial (conecta aos antigos estados iniciais de M_1 e M_2) e outro para ser o final (é conectado pelos antigos estados finais de M_1 e M_2). Para reconhecer $(L_1 \text{ ou } L_2)^n$, ou seja, a possibilidade de reconhecimento

sucessivo, criamos mais dois estados, onde conectamos os estados finais de M_1 e M_2 a um destes novos estados, ele por sua vez, reconecta ao estado que inicia o processo de escolha entre M_1 e M_2 . Ressaltamos que todos os estados adicionados possuem apenas transições vazias, ou seja, o símbolo ϵ na fita de leitura (entrada).

5 AUTÔMATO PROPOSPO E SISTEMA *Qmusic*

Nas subseções seguintes apresentamos o AFN ϵ que reconhece os acordes de uma música (Tríades e Tétrade), assim como o sistema que o implementa em Java.

5.1 Autômatos

Escolhemos os Autômatos Finitos com transições vazias para compor o Autômato deste trabalho. A escolha deste formalismo se deu por, geralmente, facilitar as construções de autômatos (HOPCROFT; MOTWANI; ULLMAN, 2002). O AFN ϵ foi construído em módulos, a partir de autômatos menores. Criamos um AFN ϵ que reconhece as Tríades (TRIADE) e outro que reconhece as Tétrade (TETRADE), em seguida utilizamos composição de autômatos para formar o AFN ϵ ACORDE que reconhece Tríades e Tétrade (ver Figura 16).

Conforme apresentamos na seção 4.2.1, um autômato é formado por um alfabeto, um conjunto de estados, no qual destacamos um em especial como estado inicial, e um subconjunto deste, em que seus elementos são os estados de aceitação. Há também uma função de transição, que define como se dará as mudanças de estados de acordo com os símbolos lidos da entrada.

O mapeamento da Estrutura de Formação de Acorde para um Autômato ocorre da seguinte forma:

- O alfabeto é o conjunto de notas musicais e seus acidentes;
- Os estados dos autômatos representam os valores que estão sendo consumidos da entrada, por exemplo, uma entrada *mi*, *sol*, *si*, se o autômato consumir a entrada *mi*, ele irá para o estado MI, caso consuma a entrada *sol* irá para o estado SOL. Perceba que poderá existir o consumo da mesma entrada em ocasiões diferentes, por exemplo, a entrada *mi*, *sol*, *si*, *mi*, neste caso diferenciaremos os estados por índices acrescentados aos mesmos, no caso MI, MI₂. Casos especiais serão explicitados aos leitores;
- Os estados de aceitação indicam que já foram lidos acordes (Tríades ou Tétrade);
- O estado inicial Q₀, para indicar o início da operação de reconhecimento (nenhuma nota lida); e
- A função de transição representa a teoria de formação de acorde, ou seja, ela define para qual estado seguir de acordo com a teoria musical. É neste ponto que tratamos as notas enarmônicas. Cada transição representa uma nota lida;

A Figura abaixo ilustra como o autômato foi desenvolvido, ressaltamos que se trata de uma representação informal.

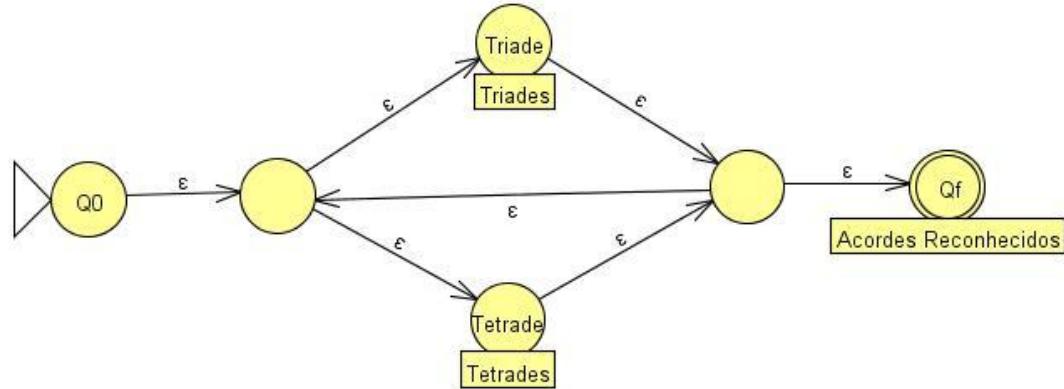


Figura 16– Ilustração do AFN ϵ ACORDE que Reconhece Triades e Tétrade

Apresentaremos a seguir, os Autômatos TRIADE, TETRADE, logo após o Autômato ACORDE.

5.1.1 Reconhecedor de Tríade (TRIADE)

O autômato que reconhece uma Tríade, chamaremos TRIADE, também foi construído em módulos, a partir dos autômatos que reconhece a Tríade Aumentada, Maior, Menor e Diminuta, ambos considerando notas enarmônicas. Conforme ilustrado na Figura abaixo.

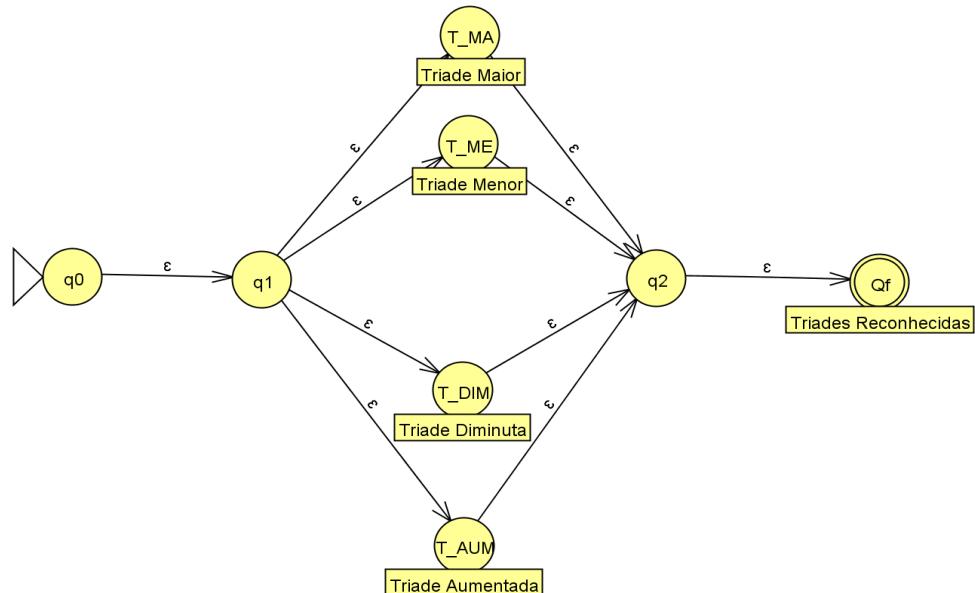


Figura 17– AFN ϵ TRIADE que Reconhece Tríade

Vamos definir formalmente os módulos (autômatos) da Figura 17 em separado, para depois compor o autômato TRIADE.

Ressaltamos que todos os Autômatos utilizados para compor o Autômato TRIADE, podem ser minimizados, ou seja, eles possuem redundância. Optamos por deixá-las para facilitar na compreensão dos mesmos.

I. Autômato que Reconhece Tríade Maior

O AFN ϵ que reconhece Tríade Maior, denotamos como T_MA, é definido de forma gráfica como segue na figura. Optamos por exibir todos os Autômatos nesta seção de forma gráfica, por julgarmos ser mais intuitivo, porém se o leitor desejar ver sua definição formal, a mesma encontra-se no Apêndice A, item a.

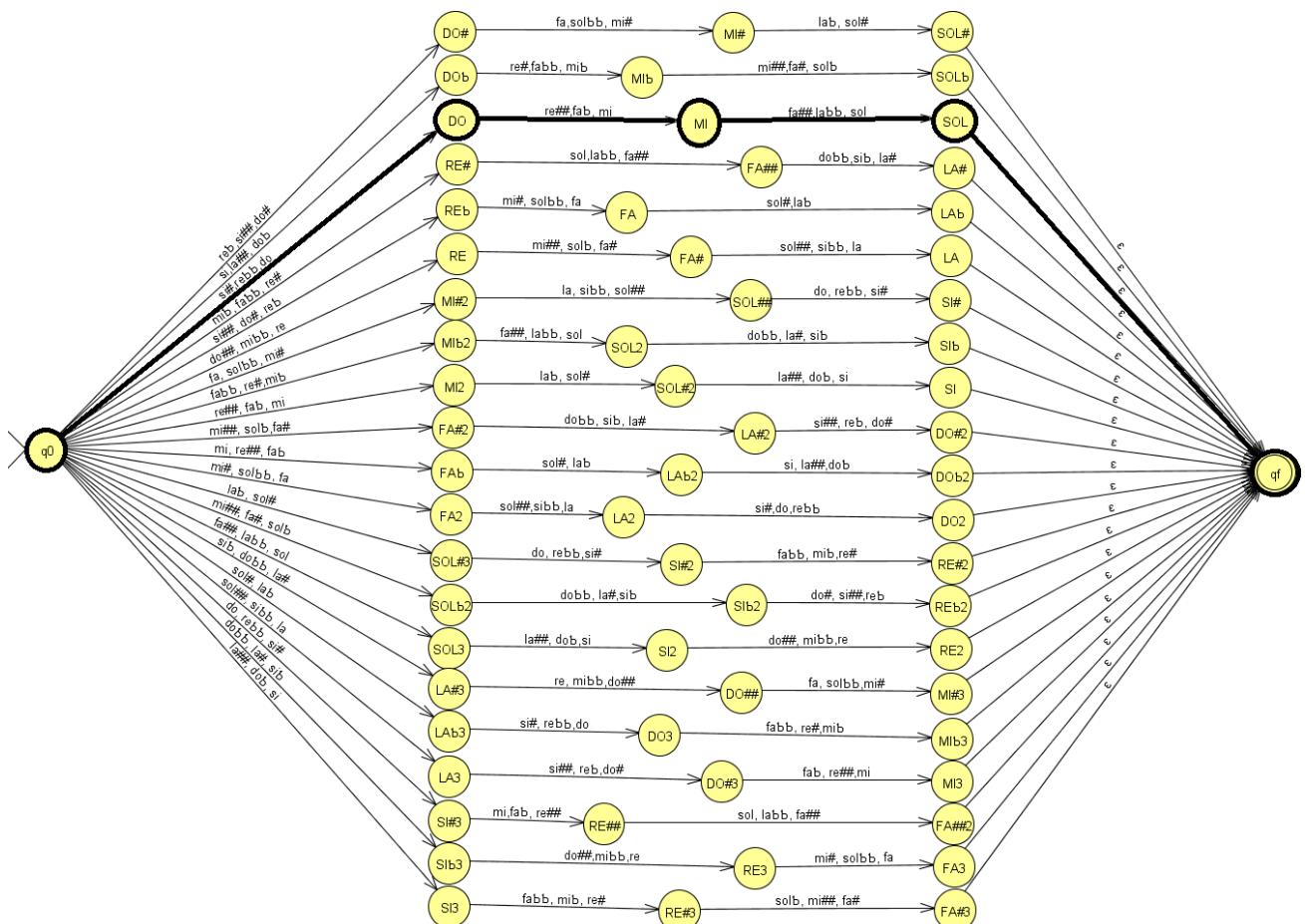


Figura 18– AFN ϵ T_MA que Reconhece Tríade Maior

A seguir mostraremos uma simples execução do Autômato definido acima.

Exemplo 4- Verificar se o Autômato T_MA aceita a entrada *do, mi, sol* (Tríade de dó maior ou C) como um acorde.

Para melhor visualizar o exemplo acima e facilitar a explicação de nossa abordagem, realçamos o pedaço do autômato que indica os estados que representam a Tríade de dó maior, composto pelas notas *do, mi, sol*, conforme indicado na Figura 18. Para verificar se as notas do exemplo 4 são reconhecidas, mostraremos de forma intuitiva o processo de reconhecimento: vemos que ao ler a nota *do* no estado Q_0 , o autômato avança para o estado DO, no estado DO ao ler a nota *mi*, o estado que surge é o MI, em seguida, ao processar a nota *sol* no estado MI, o autômato avança para o estado SOL, do estado SOL podemos avançar para o estado final Q_f , pois do estado SOL para Q_f há transição vazia. Concluímos assim, que as notas *do, mi, sol* são reconhecidas pelo T_MA.

Contudo, se ao invés de *do* tivéssemos a nota *si#* (enarmônica, ver seção 2.1.1), teríamos uma Tríade do ponto de vista sonoro, portanto deve ser reconhecida pelo Autômato, porém esta tríade está em desacordo com a teoria musical, que indica: tônica, terça e quinta (ver seção 2.2.1). Note que no trecho destacado na Figura 18, a mudança do estado Q_0 para DO se dá com as *do* e suas enarmônicas (*si#,reb#*), de forma semelhante para todos os outros estados do autômato, no caso em especial, os estados MI e SOL. Portanto, além de reconhecer a Tríade em sua forma sintaticamente correta o autômato reconhece também a tríade sonoramente correta (com notas enarmônicas). Ressaltamos que neste ponto, onde ocorrer enarmonia o sistema que implementa o autômato irá sugerir uma simples troca das notas enarmônicas pelas notas que deixam a tríade de acordo com a teoria musical. Por exemplo, *si#, mi, sol*, o autômato reconhecia como tríade, porém com uma sugestão: troca da nota *si#* por *do*.

Nos exemplos seguintes a ilustração das execuções dos autômatos será mais direta, por ser semelhante a atual.

II. Autômato que Reconhece Tríade Menor

O AFN ϵ que reconhece Tríade Menor, denotamos como T_ME, é definido de forma gráfica conforme a Figura 19. De forma semelhante ao T_MA, caso o leitor deseje ver sua definição formal, a mesma encontra-se no Apêndice A, item b.

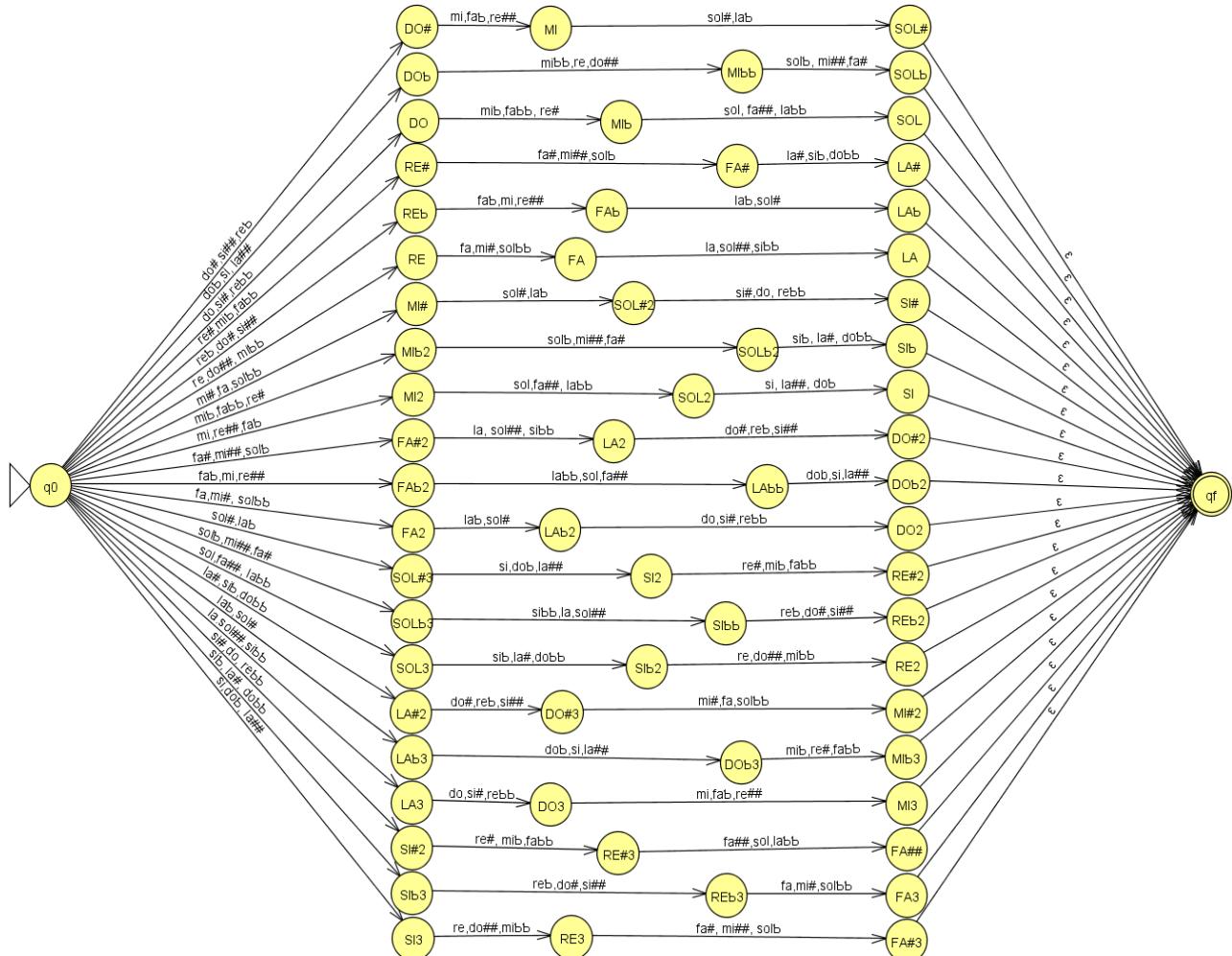


Figura 19– AFN ϵ T_ME que Reconhece Tríade Menor

A seguir mostraremos uma simples execução do Autômato T_ME definido acima.

Exemplo 5- Verificar se o Autômato T_ME aceita a entrada $do\#, mi, sol\#$ (Tríade de dó sustenido menor ou C#m) como um acorde.

Vamos executar de forma intuitiva o Autômato, por tanto ao ler a nota $do\#$ no estado Q_0 , o autômato avança para o estado $DO\#$, no estado $DO\#$ ao ler a nota mi , o estado que surge é o MI . Em seguida, ao processar a nota $sol\#$ no estado MI , o autômato avança para o estado $SOL\#$, do estado $SOL\#$ podemos avançar para o estado final Q_f , pois do estado $SOL\#$ para Q_f há transição vazia. Concluímos dessa forma que as notas $do\#, mi, sol\#$ formam uma tríade menor e são reconhecidas pelo T_ME. Ressaltamos que o AFN ϵ T_ME também considera notas enarmônicas, note que ao processarmos $si\#$ ao invés de $do\#$, o resultado será o mesmo,

neste ponto o sistema que implementa o autômato irá sugerir uma simples troca da nota *si##* por *do#*.

III. Autômato que Reconhece Tríade Aumentada

O AFN ϵ que reconhece Tríade Aumentada, denotamos como T_AUM, é definido de forma gráfica como exibido na Figura 20. De forma semelhante aos anteriores, caso o leitor deseje ver sua definição formal, a mesma encontra-se no Apêndice A, item c. Segue um exemplo que mostra uma simples execução do AFN ϵ T_AUM.

Exemplo 6- Verificar se o Autômato T_AUM aceita a entrada *do, mi, sol#* (Tríade de dó aumentado ou C+) como um acorde.

Assim como procedemos nos Exemplo 5, executaremos de forma intuitiva o Autômato, por tanto ao ler a nota *do* no estado Q_0 , o autômato avança para o estado DO, no estado DO ao ler a nota *mi*, o estado que surge é o MI. Em seguida, ao processar a nota *sol#* no estado MI, o autômato avança para o estado SOL#, do estado SOL# podemos avançar para o estado final Q_f , pois do estado SOL# para Q_f há transição vazia. Dessa forma, as notas *do, mi, sol#* formam uma tríade aumentada e são reconhecidas pelo T_AUM. Reavivamos que o AFN ϵ T_AUM, assim como os demais, considera notas enarmônicas, note que ao processarmos *si#* ao invés de *do*, o resultado será o mesmo, neste ponto o sistema que implementa o autômato irá sugerir uma simples troca da nota *si#* por *do#*.

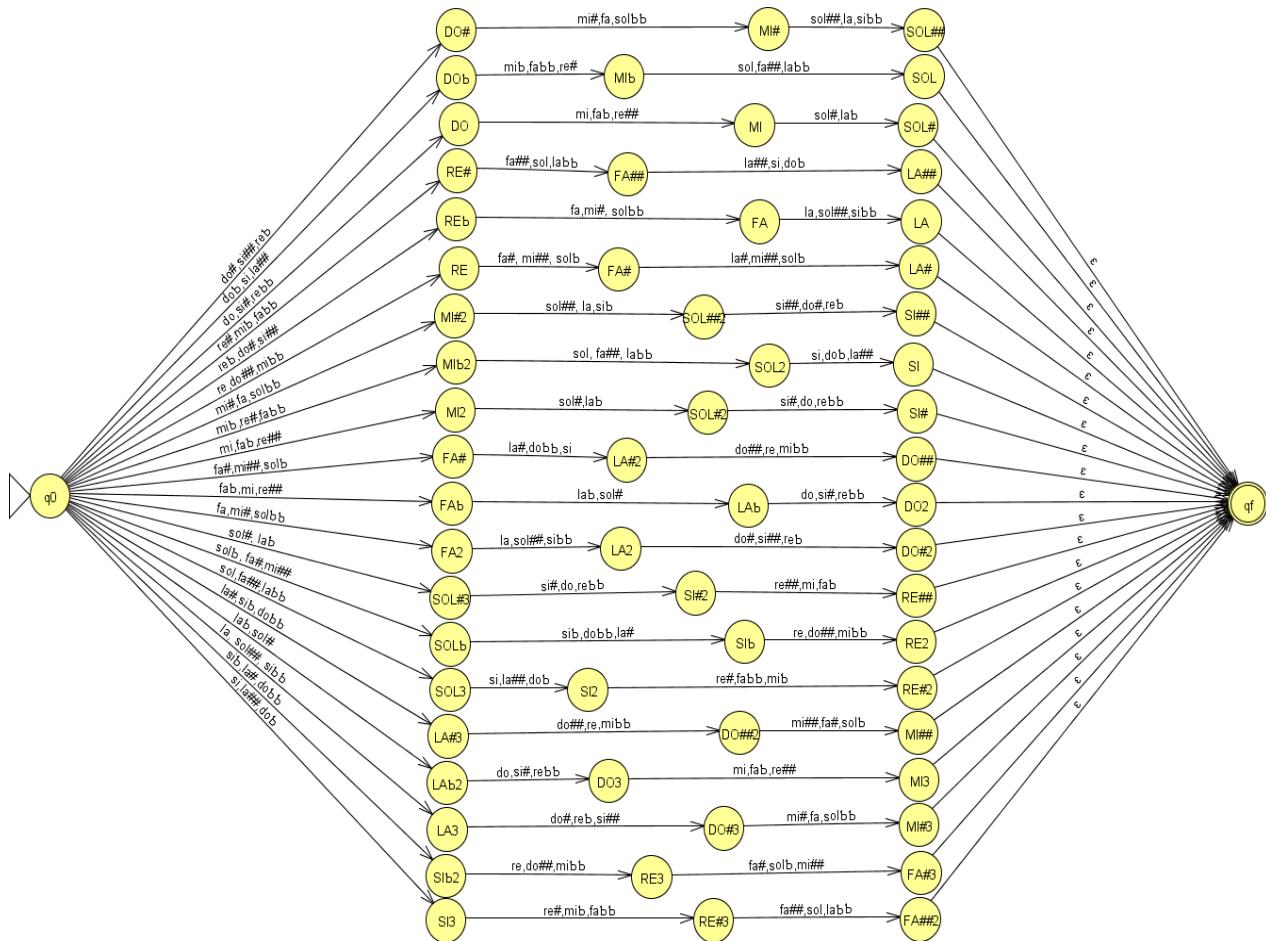


Figura 20– AFN ϵ T_AUM que Reconhece Tríade Aumentada

IV. Autômato que Reconhece Tríade Diminuta

O AFN ϵ que reconhece Tríade Diminuta, denotamos como T_DIM, é definido de forma gráfica como mostrado na Figura 21. De forma semelhante aos demais, caso o leitor deseje ver sua definição formal, a mesma encontra-se no Apêndice A, item d.

A seguir um exemplo que mostra uma simples execução do AFN ϵ T_DIM.

Exemplo 7- Verificar se o Autômato T_DIM aceita a entrada *mi, sol, si♭* (Tríade diminuta de *mi* ou E°) como um acorde.

Assim como procedemos nos exemplos anteriores, executaremos de forma intuitiva o Autômato, então ao ler a nota *mi* no estado Q_0 , o autômato avança para o estado MI_2 , no estado MI_2 ao ler a nota *sol*, o estado que surge é o SOL_2 . Em seguida, ao processar a nota *si♭* no estado SOL_2 , o autômato avança para o estado SIb , do estado SIb podemos avançar para o

estado final Q_f , pois do estado SIB para Q_f há transição vazia. Concluímos dessa forma que as notas mi , sol , sib formam uma tríade diminuta. Ressaltamos que o AFN ϵ T_DIM também considera notas enarmônicas, note que ao processarmos $re\#\#$ ao invés de mi , o resultado será o mesmo, neste ponto o sistema que implementa o autômato irá sugerir uma simples troca da nota $re\#\#$ por mi .

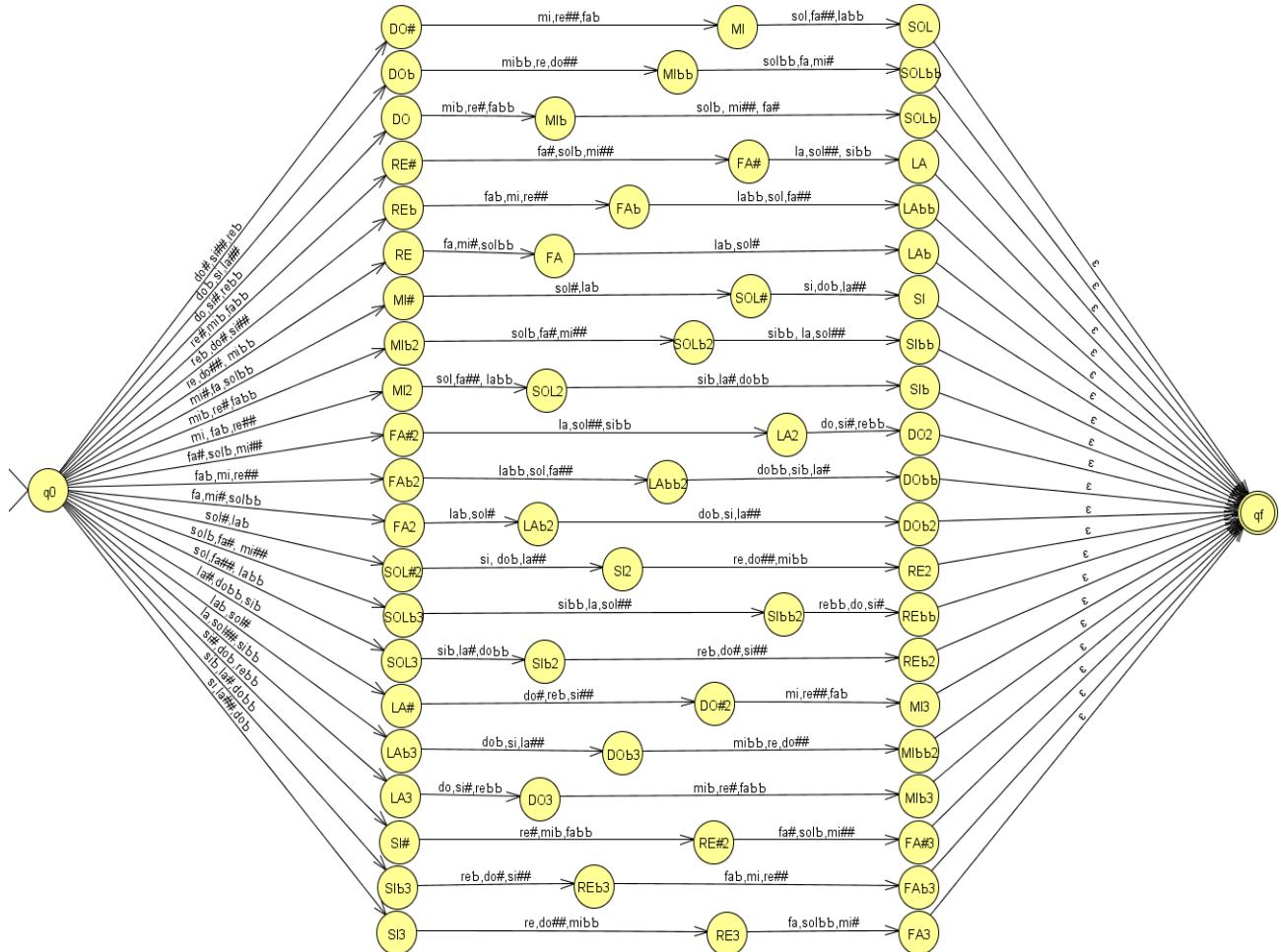


Figura 21– AFN ϵ T_DIM que Reconhece Tríade Diminuta

V. Autômato TRIADE

O AFN ϵ que reconhece uma Tríade, denominado como TRIADE, é definido a partir dos Autômatos anteriormente definidos, a saber: T_MA (reconhece Tríade Maior), T_ME (reconhece Tríade Menor), T_DIM (reconhece Tríade Diminuta) e T_AUM (reconhece Tríade Aumentada). Assim como fizemos aos demais, iremos exibi-lo em modo gráfico, contudo

caso o leitor deseje visualizar sua definição formal, a mesma encontra-se no Apêndice A, item e. Segue abaixo o Autômato TRIADE.

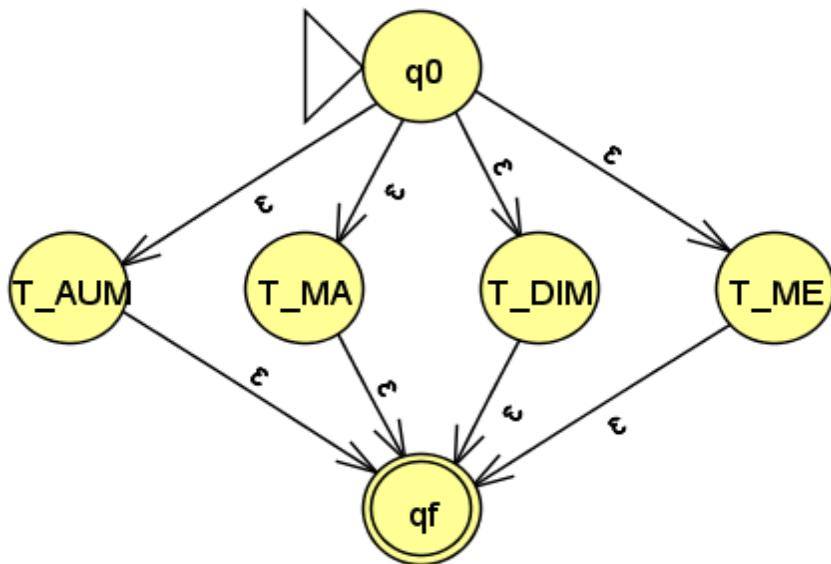


Figura 22– AFN ϵ TRIADE que Reconhece uma Tríade

Assim como fizemos aos autômatos anteriores, mostraremos em seguida uma simples execução do AFN ϵ TRIADE.

Exemplo 8- Verificar se o Autômato TRIADE aceita a entrada *mi, sol, sib* (Tríade diminuta de **mi ou E°**) como um acorde.

Conforme procedemos nos exemplos anteriores, executaremos de forma intuitiva o Autômato, vimos que a entrada *mi, sol, sib* é reconhecida pelo AFN ϵ T_DIM, por tanto igualmente reconhecida pelo AFN ϵ TRIADE, pois a partir do estado inicial Q₀ com uma transição vazia (símbolo ‘ ϵ ’ na entrada) o Autômato passa a executar os mesmos passos de T_DIM, do estado final de T_DIM com uma transição vazia, chegamos ao estado final de TRIADE. Desse modo concluímos que a entrada dada no Exemplo 8 é reconhecida pelo Autômato TRIADE. Ressaltamos que o AFN ϵ TRIADE considera notas enarmônicas, pois todos os Autômatos utilizados para lhe compor consideram.

5.1.2 Reconhecedor de Tétrade (TETRADE)

O Autômato que reconhece uma Tétrade será formado a partir dos autômatos que reconheçam as Tríades (Maior, Menor, Aumentada, Diminuta) com um acrescimento de um estado que representa o intervalo de sétima (maior ou menor). Por tanto, temos os autômatos:

- TE_MA – Autômato que reconhece uma Tétrade, formado a partir do que reconhece uma Tríade Maior (T_MA);
- TE_ME – Autômato que reconhece uma Tétrade, formado a partir do que reconhece uma Tríade Menor (T_ME);
- TE_AUM – Autômato que reconhece uma Tétrade, formado a partir do que reconhece uma Tríade Aumentada (T_AUM);
- TE_DIM – Autômato que reconhece uma Tétrade, formado a partir do que reconhece uma Tríade Diminuta (T_DIM);

A seguir as definições de todos em separado. De forma análoga ao Autômato TRIADE, todos os Autômatos utilizados para como por o Autômato TETRADE, podem ser minimizados, ou seja, eles possuem redundância. Optamos por deixá-las para facilitar na compreensão dos mesmos.

I. Autômato que Reconhece uma Tétrade formado a partir do AFN ϵ T_MA

O AFN ϵ que reconhece Tétrade maior ou menor formado a partir do que reconhece uma Tríade Maior, denotamos como TE_MA, é definido de forma gráfica conforme apresentado na Figura 23. Caso o leitor deseje ver sua definição formal, a mesma encontra-se no Apêndice B, item a. Segue um exemplo que mostra uma simples execução do AFN ϵ TE_MA.

Exemplo 9- Verificar se o Autômato TE_MA aceita a entrada *dó, mi, sol, si* (Tétrade de dó maior formada a partir da Tríade dó maior ou C(7M)) como um acorde.

De forma análoga as demais, executaremos de forma intuitiva o Autômato, assim, ao ler a nota *dó* no estado Q₀, o autômato avança para o estado DO, no estado DO ao ler a nota *mi*, o estado que surge é o MI. Em seguida, ao processar a nota *sol* no estado MI, o autômato avança para o estado SOL, no estado SOL lendo *si*, o estado que surgir é o SI(b), do estado SI(b) podemos avançar para o estado final Q_f, pois de SI(b) para Q_f há transição vazia.

Concluímos dessa forma que as notas *do*, *mi*, *sol*, *si* formam uma tétrade maior. Ressaltamos que o AFN ε TE_MA também considera notas enarmônicas, note que ao processarmos *si* \sharp ao invés de *do*, o resultado será o mesmo, neste ponto o sistema que implementa o autômato irá sugerir uma simples troca da nota *si* \sharp por *do*.

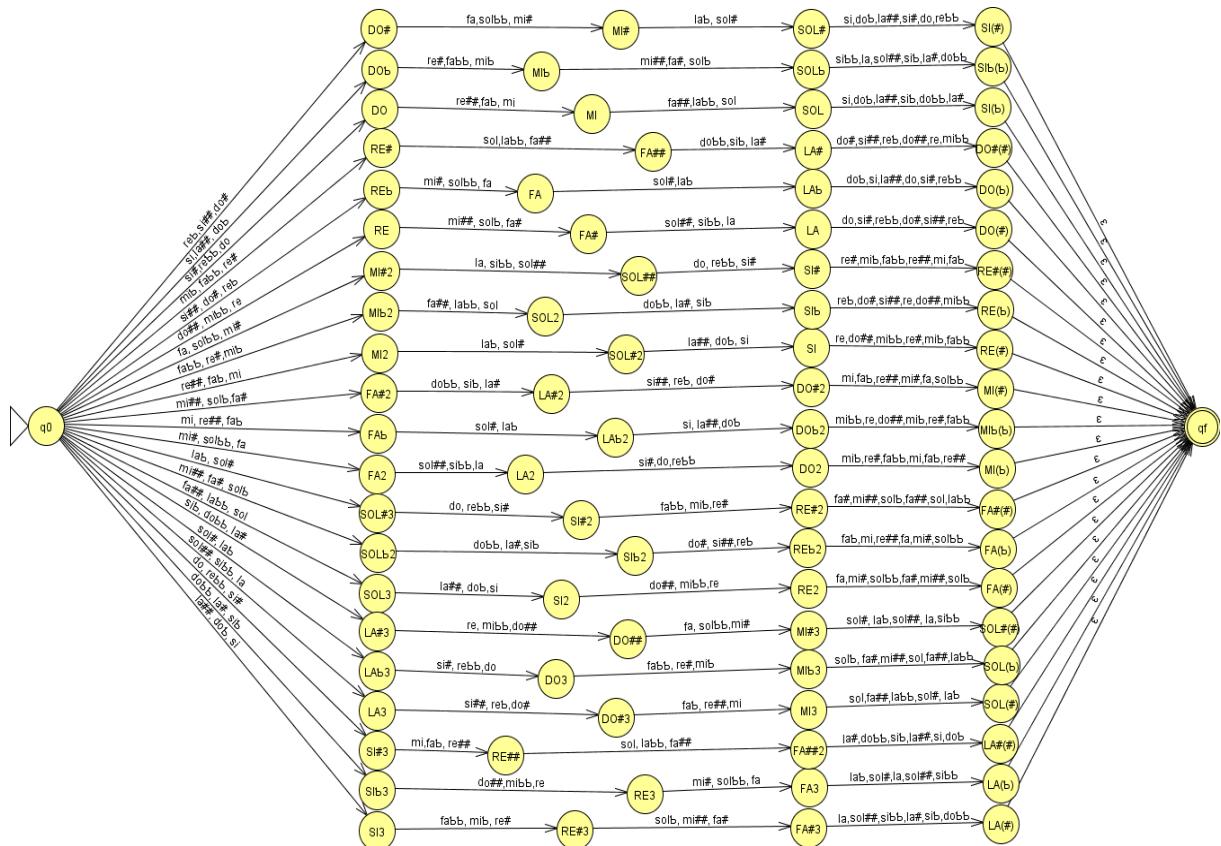


Figura 23– AFN ϵ TE_MA que Reconhece uma Tétrade a partir de uma Tríade Maior

II. Autômato que Reconhece Tétrade formado a partir do AFN ϵ T_ME

O AFN ε que reconhece Tétrade maior ou menor formado a partir do que reconhece uma Tríade Menor, denotamos como TE_ME, é definido de forma gráfica conforme apresentado na Figura 24. Tal qual o anterior, sua definição formal encontra-se no Apêndice B, item b. Segue um exemplo que mostra uma simples execução do AFN ε TE_ME.

Exemplo 10- Verificar se o Autômato TE_ME aceita a entrada *do, mi, sol, si* (Tétrade de dó maior formada a partir da Tríade dó menor ou **Cm(7M)**) como um acorde.

De maneira análoga as demais, executaremos intuitivamente o Autômato TE_ME, assim, ao ler a nota *do* no estado Q_0 , o autômato avança para o estado DO, no estado DO ao ler a nota *mib*, o estado que surge é o MI b . Em seguida, ao processar a nota *sol* no estado MI b , o autômato avança para o estado SOL, no estado SOL lendo *si*, o estado que surgir é o SI(b), do estado SI(b) podemos avançar para o estado final Q_f , pois de SI(b) para Q_f há transição vazia. Portanto, as notas *do*, *mib*, *sol*, *si* formam uma tétrade maior. Ressaltamos que o AFN ϵ TE_ME, assim como os demais, considera notas enarmônicas, note que ao processarmos *re* $\#$ ao invés de *mib*, o resultado será o mesmo, neste ponto o sistema que implementa o autômato irá sugerir uma simples troca da nota *re* $\#$ por *mib*.

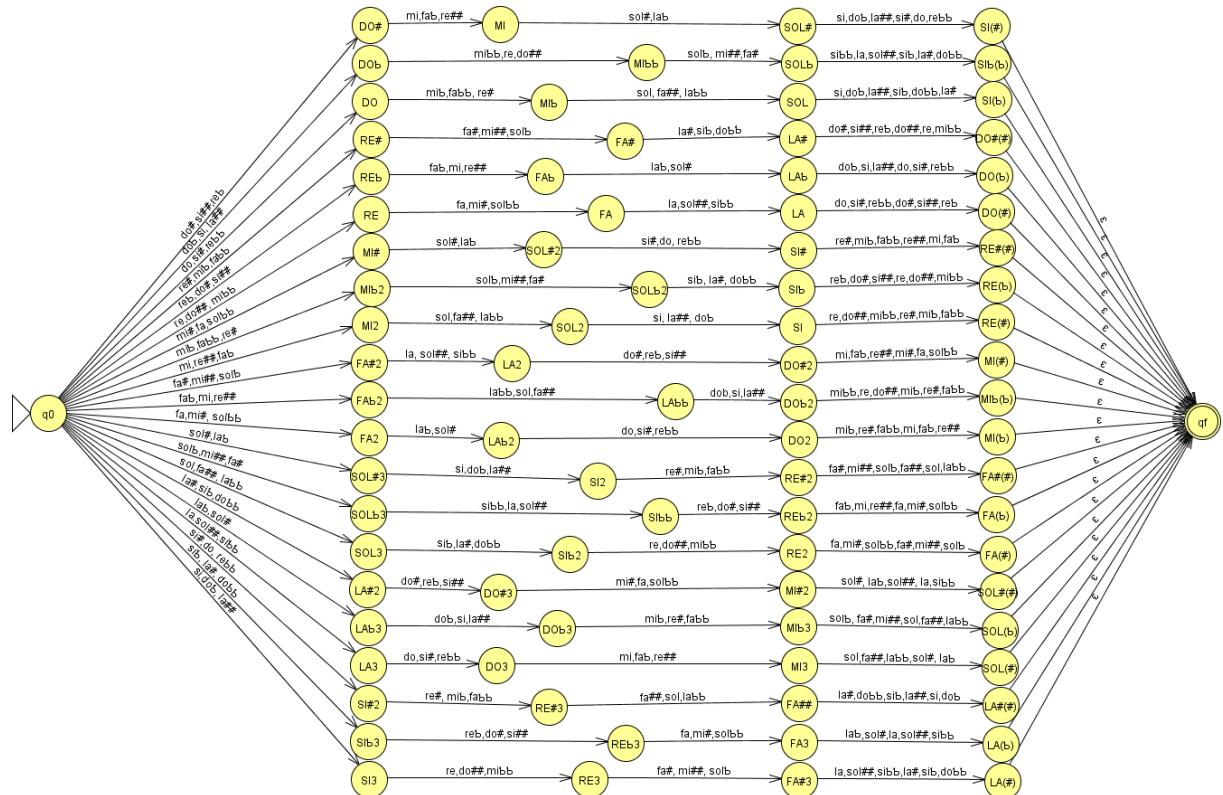


Figura 24– AFN ϵ TE_ME que Reconhece uma Tétrade a partir de uma Tríade Menor

III. Autômato que Reconhece Tétrade formado a partir do AFN ϵ T_DIM

O AFN ϵ que reconhece Tétrade maior ou menor formado a partir do que reconhece uma Tríade Diminuta, denotamos como TE_DIM, é definido de forma gráfica conforme apresentado na Figura 25. Sua definição formal encontra-se no Apêndice B, item c. A Seguir um exemplo que mostra uma simples execução do AFN ϵ TE_DIM.

Exemplo 10- Verificar se o Autômato TE_DIM aceita a entrada $mi, sol, si\flat, re\#$ (Tétrade maior de mi formada a partir da Tríade diminuta de mi ou $E^o(7M)$) como um acorde.

Como os demais, executaremos de forma intuitiva o Autômato, então, ao ler a nota mi no estado Q_0 , o autômato avança para o estado MI_2 , no estado MI_2 ao ler a nota sol , o estado que surge é o SOL_2 . Em seguida, ao processar a nota $si\flat$ no estado SOL_2 , o autômato avança para o estado $SI\flat$, no estado $SI\flat$ lendo $re\#$, o estado que surgir é o $RE(\#)$, do estado $RE(\#)$ podemos avançar para o estado final Q_f , pois de $RE(\#)$ para Q_f há transição vazia. Portanto, as notas $mi, sol, si\flat, re\#$ formam uma tétrade maior. Ressaltamos que o AFN ϵ TE_DIM, assim como os demais, considera notas enarmônicas, note que ao processarmos $fa\flat$ ao invés de mi , o resultado será o mesmo, neste ponto o sistema que implementa o autômato irá sugerir uma simples troca da nota $fa\flat$ por mi .

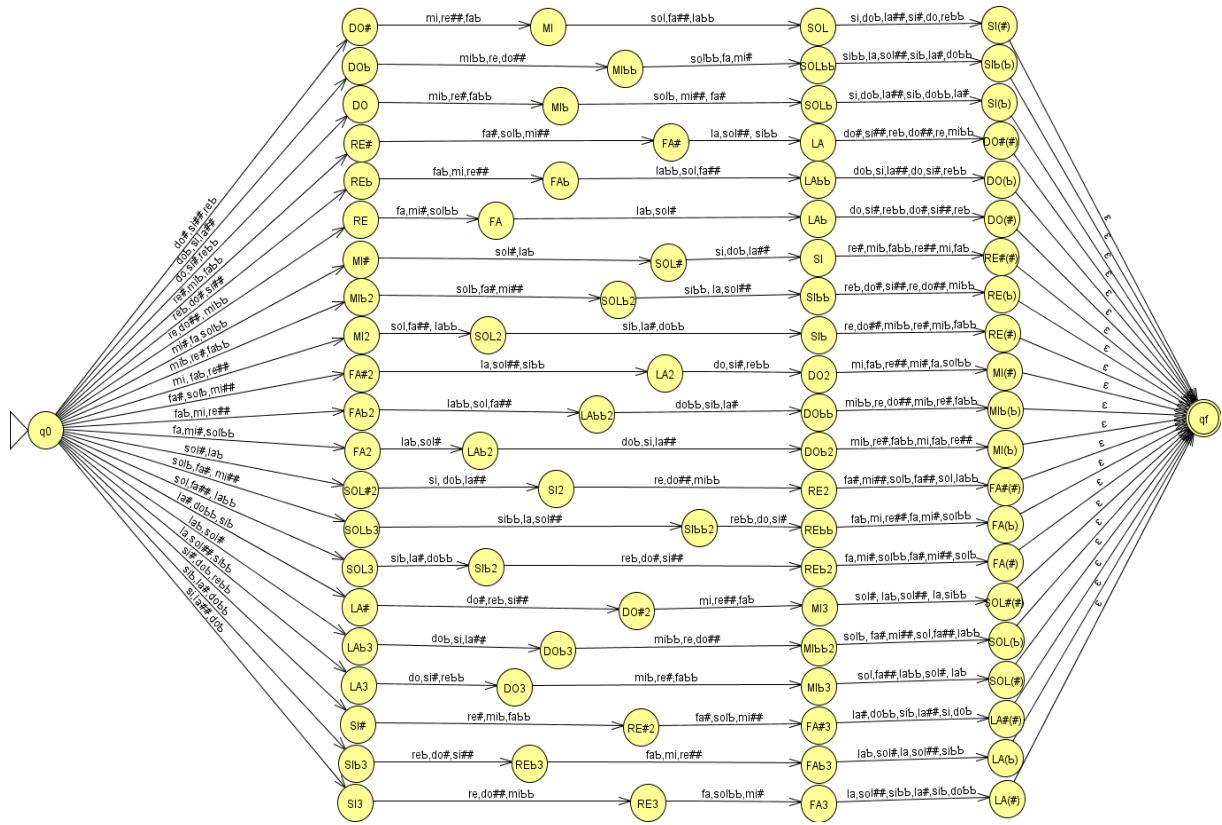


Figura 25– AFN ϵ TE_DIM que Reconhece uma Tétrade a partir de uma Tríade Diminuta

IV. Autômato que Reconhece Tétrade formado a partir do AFN ϵ T_AUM

O AFN ϵ que reconhece Tétrade maior ou menor formado a partir de que reconhece uma Tríade Aumentada, denotamos como TE_AUM, é definido de forma gráfica conforme apresentado na Figura 26. Sua definição formal encontra-se no Apêndice B, item d. Segue um exemplo que mostra uma simples execução do AFN ϵ TE_AUM.

Exemplo 11- Verificar se o Autômato TE_AUM aceita a entrada $do, mi, sol\#, si\flat$ (Tétrade de dó menor formada a partir da Tríade de dó aumentado ou C+(7M)) como um acorde.

De forma análoga as demais, executaremos o AFN ϵ TE_AUM de forma intuitiva, assim, ao ler a nota do no estado Q_0 , o autômato avança para o estado DO , no estado DO ao ler a nota mi , o estado que surge é o MI . Em seguida, ao processar a nota $sol\#$ no estado MI , o autômato avança para o estado $SOL\#$, no estado $SOL\#$ lendo $si\flat$, o estado que surgir é o $SI(b)$, do estado $SI(b)$ podemos avançar para o estado final Q_f , pois de $SI(b)$ para Q_f há transição vazia. Concluímos dessa forma que as notas $do, mi, sol\#, si\flat$ formam uma tétrade menor. Ressaltamos que o AFN ϵ TE_MA também considera notas enarmônicas, note que ao processarmos $si\#$ ao invés de do , o resultado será o mesmo, neste ponto o sistema que implementa o autômato irá sugerir uma simples troca da nota $si\#$ por do .

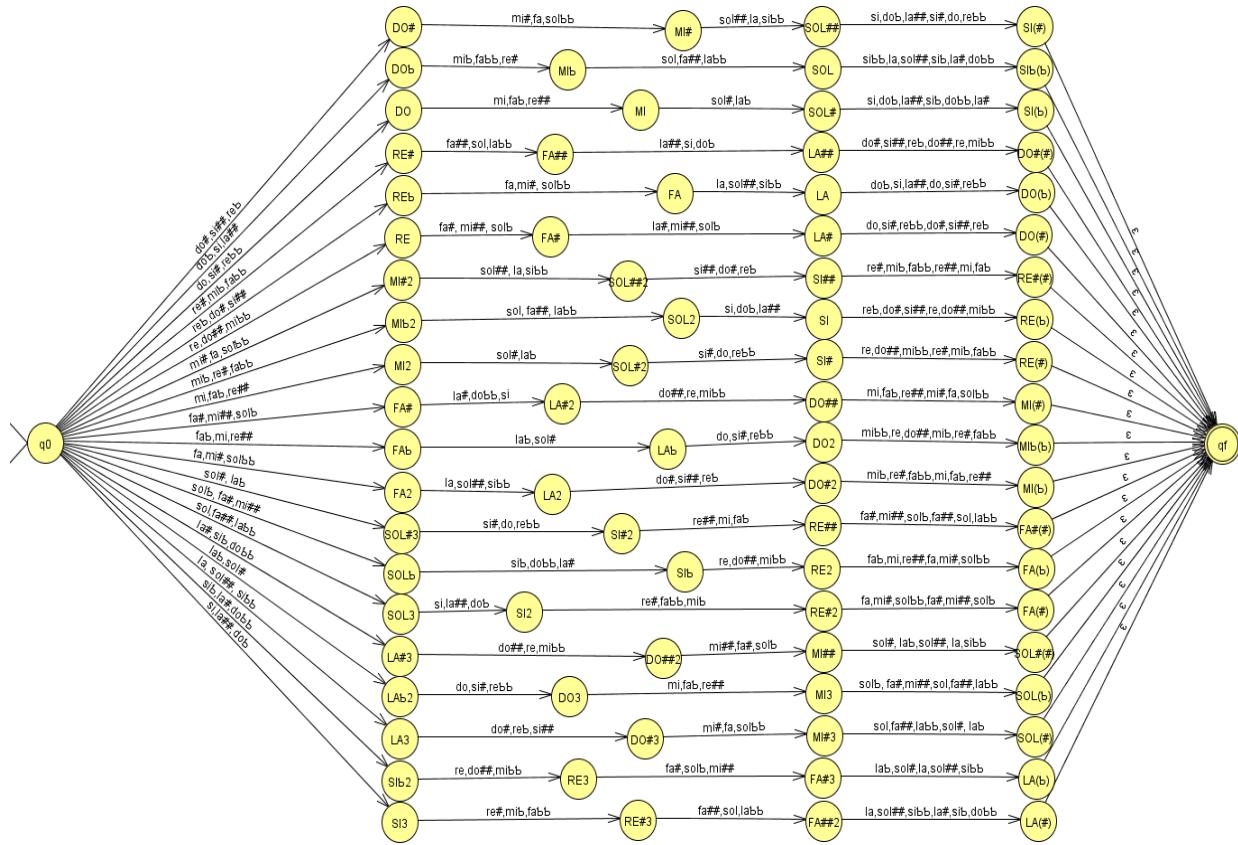


Figura 26— AFN_E TE_AUM que Reconhece uma Tétrade a partir de uma Tríade Aumentada

V. Autômato que Reconhece Tétrade (TETRADE)

O AFN ϵ que reconhece uma Tétrade, denotamos como TETRADE, é definido a partir dos Autômatos anteriormente definidos, a saber: TE_MA (reconhece Tétrade a partir do que reconhece Tríade Maior), TE_ME (reconhece Tétrade a partir do que reconhece Tríade Menor), TE_DIM (reconhece Tétrade a partir do que reconhece Tríade Diminuta) e TE_AUM (reconhece Tétrade a partir do que reconhece Tríade Aumentada). Assim como fizemos aos demais, iremos exibi-lo em modo gráfico, contudo caso o leitor deseje visualizar sua definição formal, a mesma encontra-se no Apêndice B, item e. Abaixo segue o Autômato TETRADE.

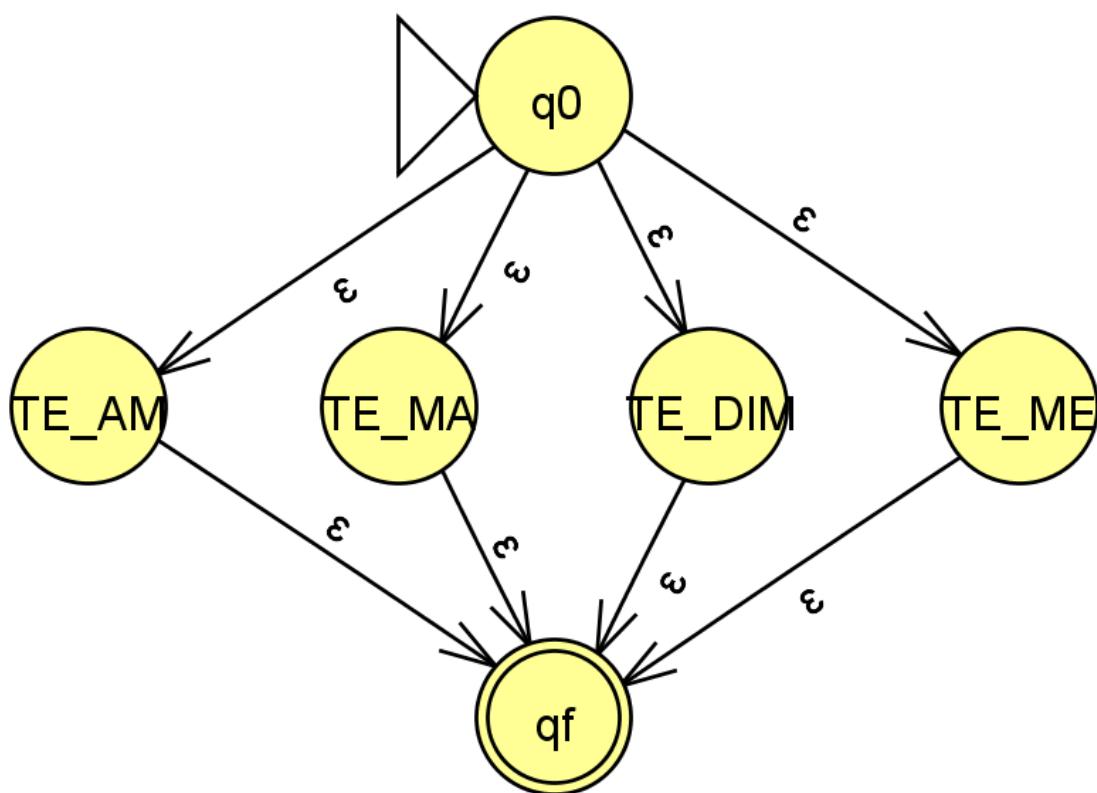


Figura 27– AFN ϵ TETRADE que Reconhece uma Tétrade

Conforme fizemos aos autômatos anteriores, mostraremos em seguida uma simples execução do AFN ϵ TETRADE.

Exemplo 12- Verificar se o Autômato TETRADE aceita a entrada *mi, sol, sib, re#* (Tétrade maior de *mi* formada a partir da Tríade diminuta de *mi* ou E°(7M)) como um acorde.

De maneira semelhante como procedemos nos exemplos anteriores, executaremos intuitivamente o Autômato TETRADE. Vimos que a entrada *mi, sol, si♭, re#* é reconhecida pelo AFN ϵ TE_DIM, por tanto igualmente reconhecida pelo AFN ϵ TETRADE, pois a partir do estado inicial Q_0 com uma transição vazia (símbolo ‘ ϵ ’ na entrada) o Autômato passa a executar os mesmos passos de TE_DIM, e do estado final de TE_DIM com uma transição vazia chegamos ao estado final de TETRADE. Assim concluímos que a entrada dada no Exemplo 12 é reconhecida pelo Autômato TETRADE. Como todos os Autômatos que compõe TETRADE consideram notas enarmônicas, logo, TETRADE também o faz.

5.1.3 Reconhecedor de Tríades e Tétrade (ACORDE)

Até o momento, os autômatos construídos reconhecem apenas um acorde (tríade ou tétrade), O AFN ϵ ACORDE que definiremos a seguir reconhece uma sequência deles. Ele foi construído a partir dos Autômatos anteriormente definidos, a saber: TRIADE (reconhece uma Tríade), TETRADE (reconhece uma Tétrade). Fizemos uma composição desses dois autômatos, de modo que o AFN ϵ gerado reconheça uma sequência de tríades e tétrade. Intuitivamente a composição dos AFN ϵ TRIADE e TETRADE para gerar o AFN ϵ ACORDE é da seguinte maneira:

1. Um novo estado, chamado S_OR com transições vazias para os estados iniciais dos AFN ϵ TRIADE e TETRADE, ou seja, este estado representa a possibilidade de reconhecimento de ou uma tríade ou uma tétrade;
2. Os estados finais dos AFN ϵ TRIADE e TETRADE terão transição para um novo estado chamado S_*, e S_* com transições vazias para o estado final e para o estado S_OR, ou seja, o estado S_* representa a possibilidade de reconhecimento de várias tríades e tétrade;

A Figura 28 mostra, em modo gráfico, o Autômato ACORDE definido informalmente acima. Contudo, caso o leitor deseje visualizar sua definição formal, a mesma encontra-se no Apêndice C. A seguir uma simples execução do Autômato ACORDE.

Exemplo 13- Verificar se o Autômato ACORDE aceita a entrada *do, mi, sol#, do, mi, sol, si* (Tríade de dó aumentado ou C+ e Tétrade de dó maior formada a partir da Tríade dó maior ou C(7M)) como uma sequência de acordes.

Conforme procedemos nos exemplos anteriores, executaremos de forma intuitiva o Autômato ACORDE. Observamos que a entrada *do, mi, sol#* é reconhecida pelo AFN ϵ TRIADE, por tanto igualmente reconhecida pelo AFN ϵ ACORDE, pois a partir do estado inicial Q_0 com uma transição vazia (símbolo ‘ ϵ ’ na entrada) o Autômato passa para o estado $S_{_OR}$, dele com uma transição vazia podemos executar os mesmos passos do AFN ϵ TRIADE, de TRIADE podemos ir para o estado $S_{_*}$, e deste retornar para $S_{_OR}$, de $S_{_OR}$ com uma transição vazia, podemos executar os mesmos passos do AFN ϵ TETRADE, por outro lado, mostramos que as notas *do, mi sol, si* são reconhecidas pelo AFN ϵ TETRADE, por tanto igualmente reconhecidas pelo AFN ϵ ACORDE. Desse modo concluímos que a entrada dada no Exemplo 12 é reconhecida pelo Autômato ACORDE. Como ambos os Autômatos que compõem o AFN ϵ ACORDE tratam notas enarmônicas, ele também o faz.

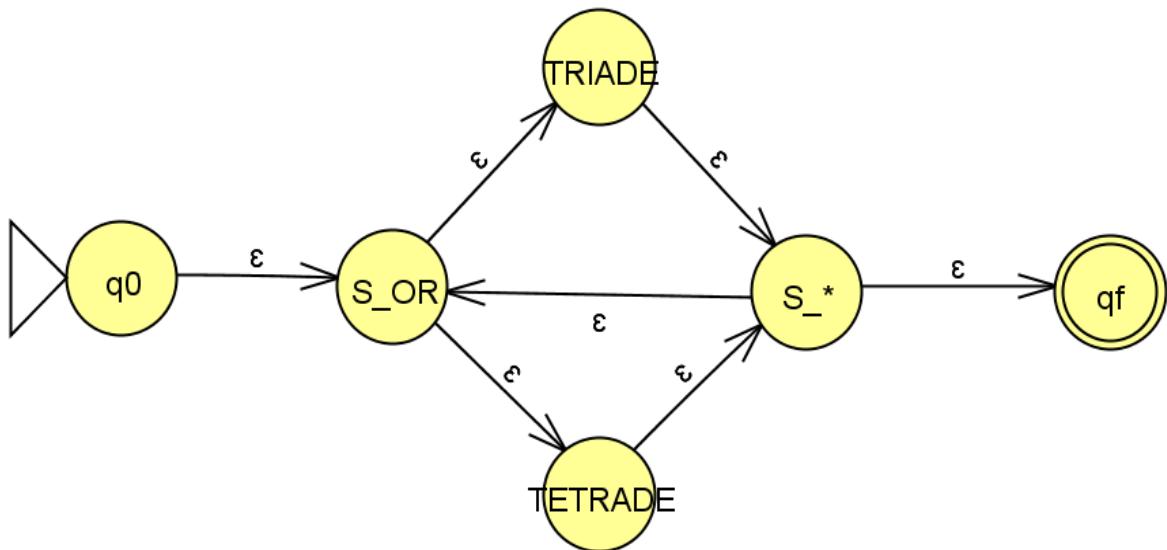


Figura 28– AFN ϵ ACORDE que Reconhece Tríades e Tétrade

5.2 Sistema *Qmusic*

Nesta seção abordaremos o Sistema *QMusic*, discorrendo sobre suas funcionalidades, modo de exibição e como utilizá-lo.

5.2.1 Visão Geral

O sistema *Qmusic* foi desenvolvido em JAVA SE, utilizando as APIs Java Sound e *jMusic*. Ele implementa o Autômato ACORDE e todos os módulos que o compõe, a saber:

- AFN_E que reconhece uma Tríade e seus Módulos:
 - AFN_E que Reconhece Tríade Maior, Menor, Aumentada e Diminuta;
- AFN_E que reconhece uma Tétrade e seus Módulos
 - Maior e menor formadas a partir das Tríades;

Após a leitura do arquivo MIDI, utilizamos o *Jmusic* para nomear as notas, em seguida utilizamos o autômato criado para reconher os acordes. Logo após, nomeamos os acordes que foram reconhecidos pelo autômato, seguindo a teoria musical, que indica: o nome do acorde é dado pelo nome da nota tônica (ver seção 2.2). Caso ocorra problema de enarmonia no momento do processo de nomear as notas pelo *Jmusic*, iniciamos um processo de troca das notas por suas notas enarmônicas até o acorde fique em acordo com a teoria musical.

Para utilizar o sistema, o processo não é complexo. De modo geral, o usuário seleciona uma música no formato MIDI e escolhe quais as opções de reconhecimento que deseja executar, por exemplo, depois de ter selecionado uma música, o sistema carrega todos os instrumentos desta música e habilita as opções de reconhecimento, o usuário pode optar por reconhecer somente Tríades Aumentadas, somente Tríades, somente Tétrades dentre outras opções. Pode escolher também o modo como o resultado será exibido, optando pelos seguintes modos:

- *Standard* (Padrão) – Exibe os acordes reconhecidos em notação ABC (ver seção 2.1.1, item 2.1.5);
 - Exemplo: Cm G C;
- *Advanced* (Avançado) – Exibe os acordes no padrão [nome-composição], mostrando seu nome, exemplo, Tríade Maior de Dó, exibindo sua nota tônica, terça e quinta e ainda sétima se for tétrade;
 - Exemplo: Tríade de Dó menor [tônica = *dó*, Terça = *mi bemol*, Quinta = *sol*];

- *Show all Results* (Mostrar todo o Resultado) – Exibe o resultado tal qual o avançado, porém exibindo, inclusive, os casos de falha (casos onde não foi possível formar Acordes).
 - Exemplo: Caso de sucesso → Tríade de Dó menor [tônica = *dó*, Terça = *mí bemol*, Quinta = *sol*]; Caso de falha → Não podemos construir um acorde a partir da nota *do*;

O sistema possibilita ao usuário a opção de ouvir a música selecionada, assim como a de mostrar as notas dessa música.

A seguir, a descrição detalhada de como utilizar o sistema.

5.2.2 Carregando, ouvindo e exibindo as notas de um arquivo MIDI

Ao iniciar o sistema, o usuário irá se deparar com a imagem abaixo.

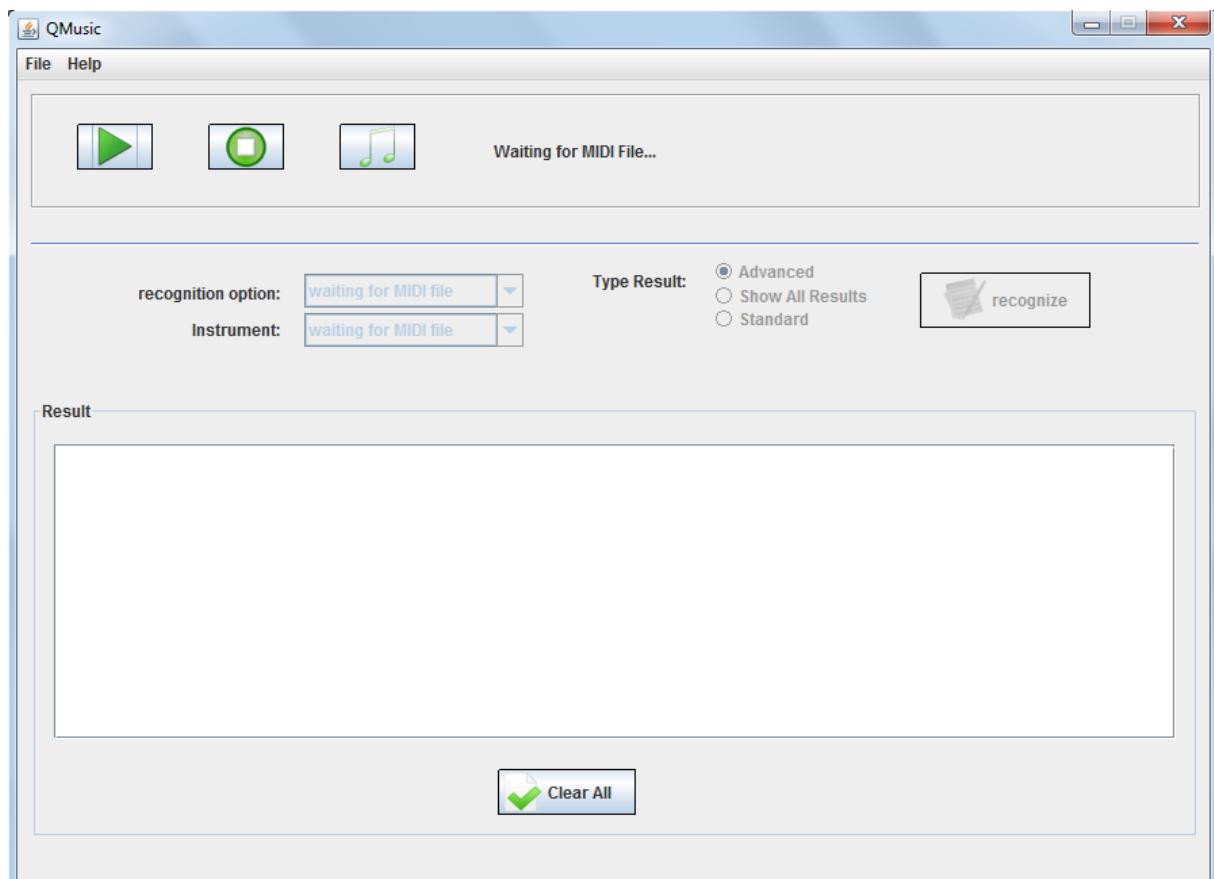


Figura 29– Tela inicial do *Qmusic*

Para carregar e ouvir um arquivo MIDI, o usuário deverá proceder da seguinte maneira: *File* → *Open MIDI file*. Caso o arquivo realmente seja do formato MIDI, o sistema irá exibir na caixa *result* a mensagem *file uploaded successfully*, exibirá na tela algumas propriedades do arquivo, habilitará as opções de reconhecimento e carregará os instrumentos que a música ostenta. Tal qual a figura abaixo.

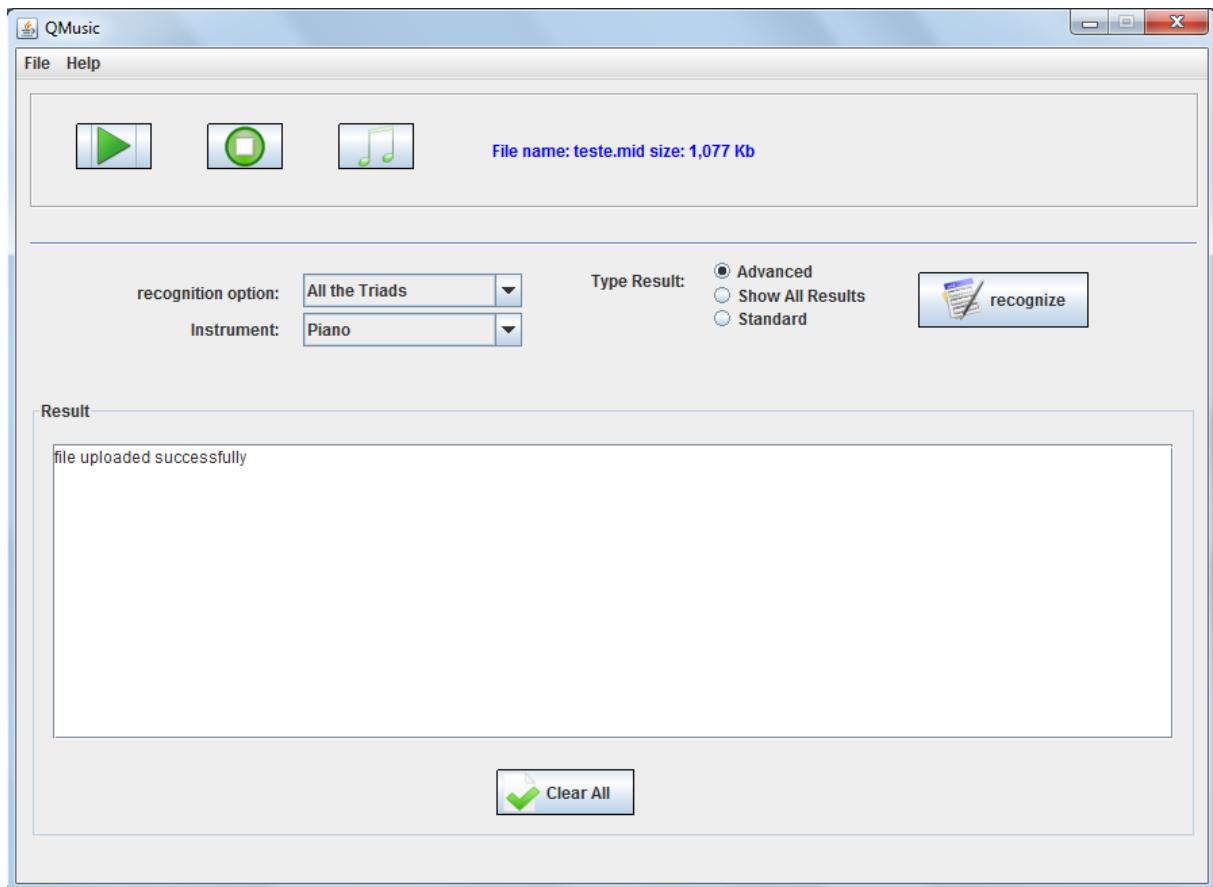


Figura 30 – Tela depois do carregamento do arquivo teste.mid

Caso o sistema esteja em acordo com a Figura 34, o usuário poderá executar:

- Para tocar a música selecionada;
- Para interromper a música em andamento;
- Para listar as notas da música selecionada;

5.2.3 Reconhecendo Acordes

Para utilizar a funcionalidade de reconhecer acordes, o usuário deverá carregar um arquivo MIDI, escolher algum instrumento da música, escolher a opção de reconhecimento, o método de visualização e por fim, clicar em *recognize*. O resultado será exibido dentro do painel *result*. A qualquer momento o usuário poderá clicar em *clear all*, para limpar a tela. A seguir uma ilustração dessa operação.

- Exibição em modo *standard* (notação ABC):

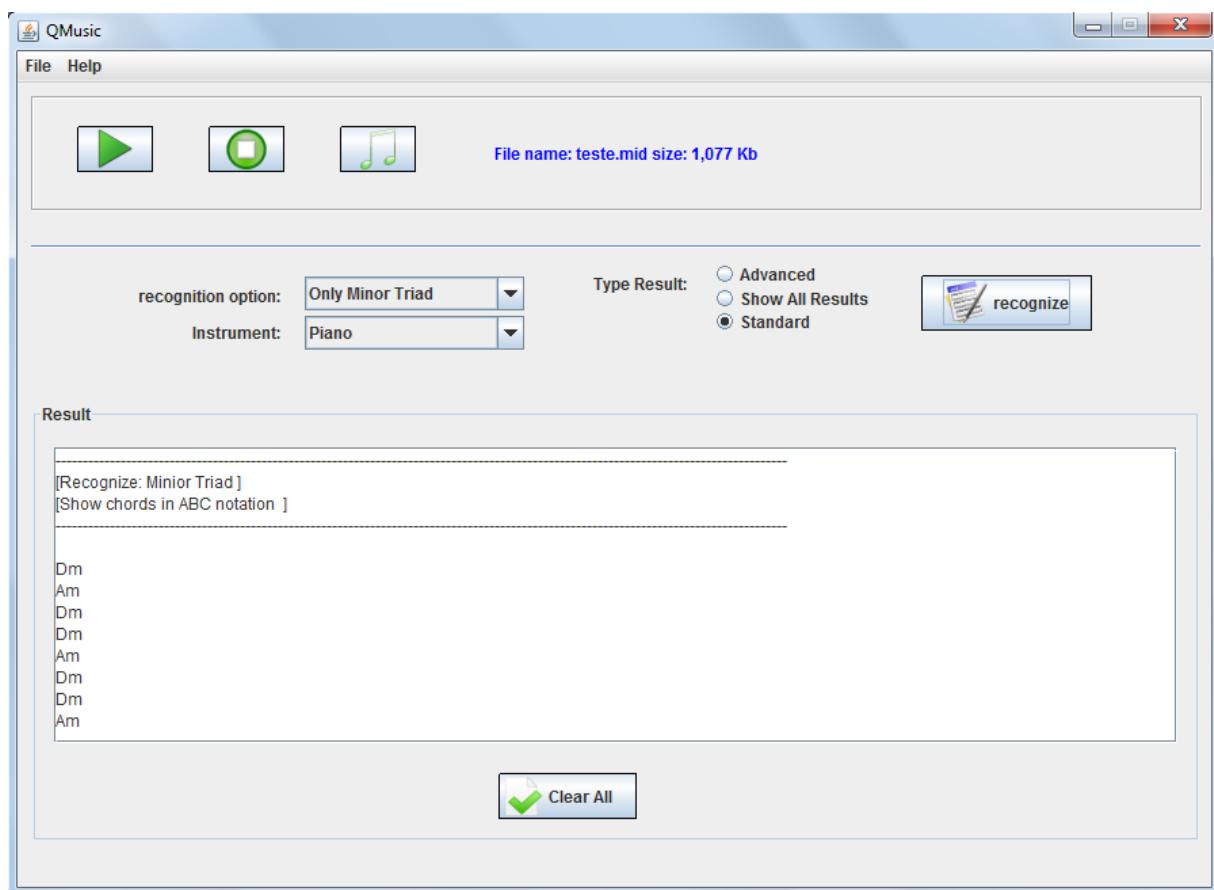


Figura 31 – Tela em modo *Standard*

- b) Exibição em modo *Advanced* (Exibição de nome e notas do acorde);

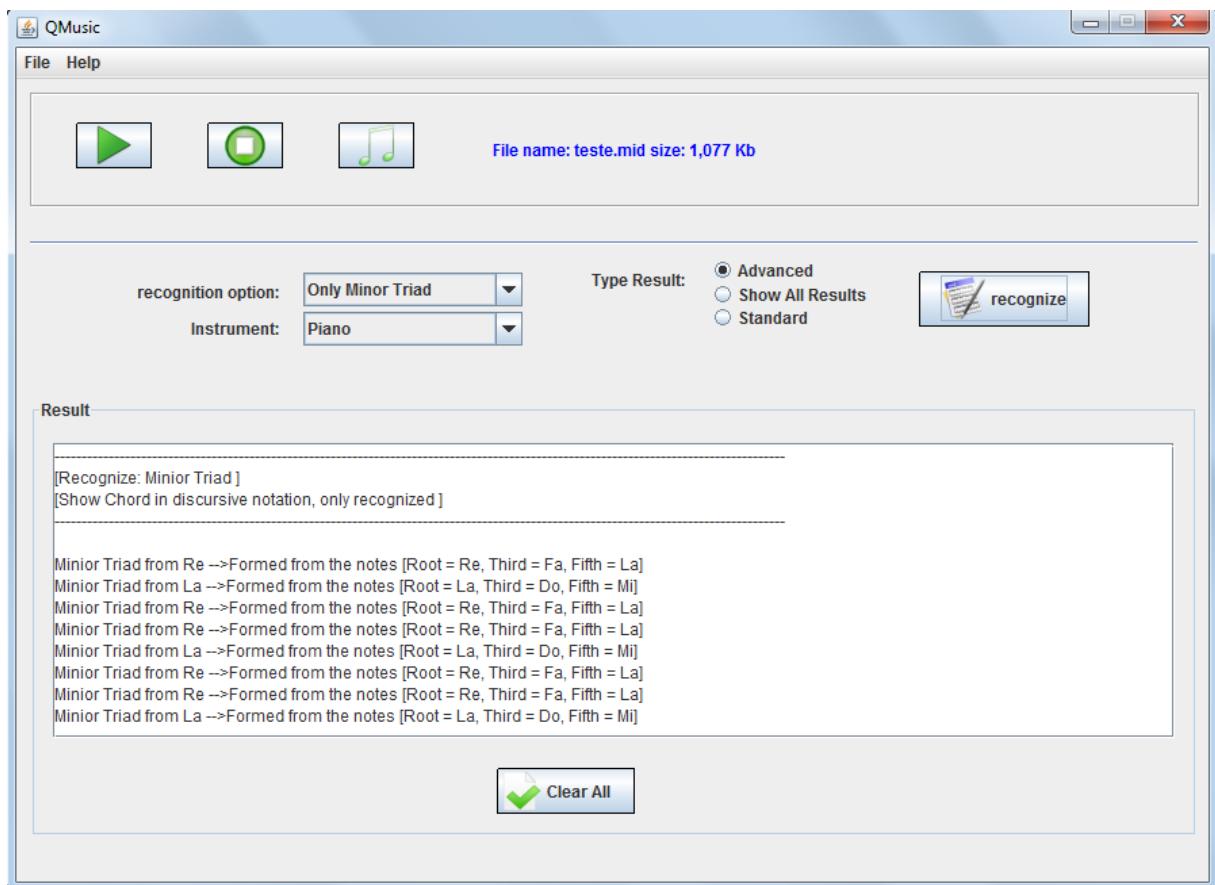


Figura 32 – Tela em modo *Advanced*

- c) Exibição em modo *show all results* (Exibição de nome e notas do acorde, inclusive falhas);

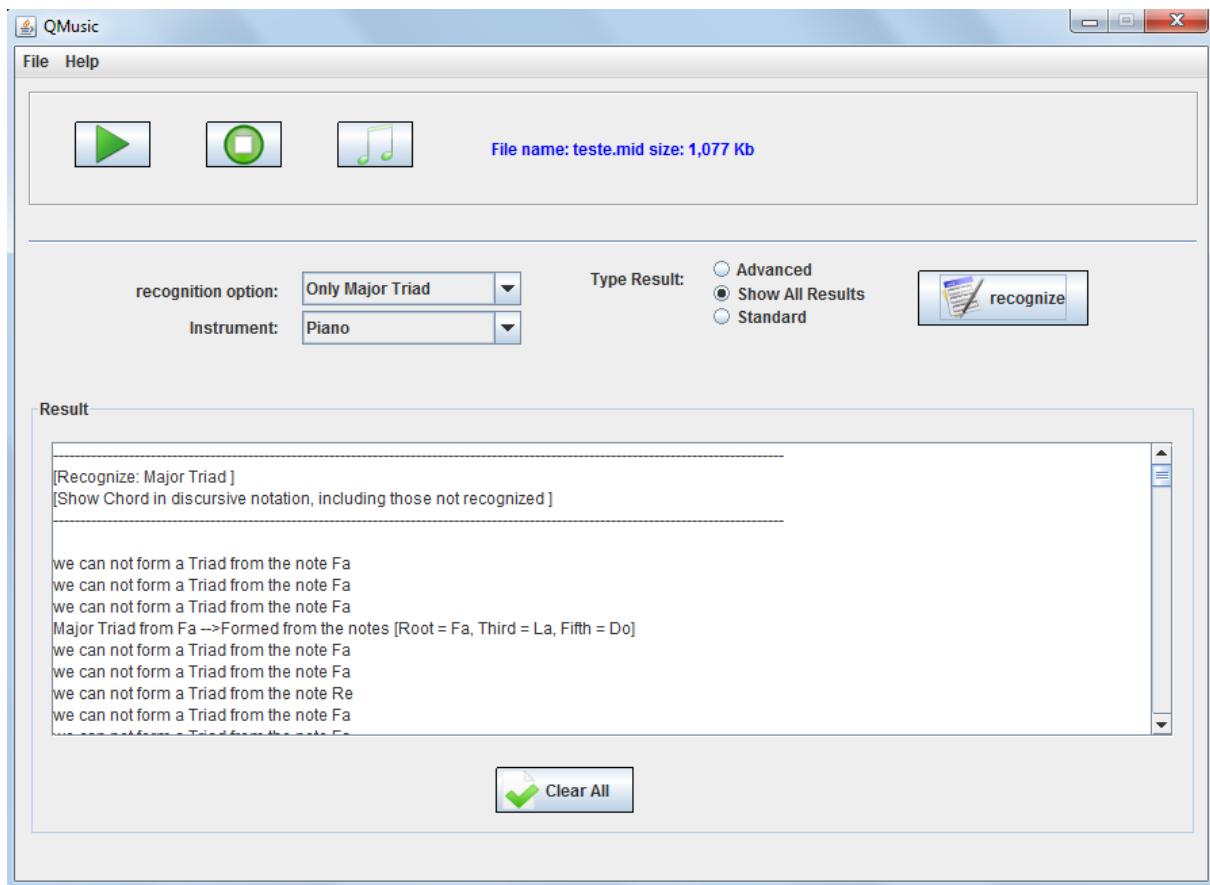


Figura 33 – Tela em modo *Show all Results*

5.2.4 Opções de Reconhecimento

O sistema permite várias opções de reconhecimento, a saber:

- *All the Triads* (Todas as Tríades) – busca por todas as tríades no arquivo MIDI;
- *All the Sevenths* (Todas as Tétrade) – busca por todas as tétrade no arquivo MIDI;
- *Triads na Sevenths* (Tríades e Tétrade) – busca por tríades e tétrade no arquivo MIDI;
- *Only Minor Triad* (Apenas Tríades Menores) – busca somente por Tríades menores no arquivo MIDI;

- *Only Major Triad* (Apenas Tríades Maiores) – busca somente por Tríades maiores no arquivo MIDI;
- *Only Augmented Triad* (Apenas Tríades Aumentadas) – busca somente por Tríades aumentadas no arquivo MIDI;
- *Only Diminished Triad* (Apenas Tríades Diminutas) – busca somente por Tríades diminutas no arquivo MIDI;
- *Only Minor Sevenths* (Apenas Tétrade Menores) – busca somente por Tétrade menores no arquivo MIDI;
- *Only Major Sevenths* (Apenas Tétrade Maiores) – busca somente por Tétrade maiores no arquivo MIDI;

Esta gama de possibilidades de reconhecimentos pode servir, por exemplo, para um estudante de música interessado em estudar apenas alguns aspectos da teoria de formação de acordes.

A seguir, um capítulo voltado a um estudo de caso.

6 ESTUDO DE CASO

Apresentamos nesta seção um estudo de caso realizado. Executamos o sistema *Qmusic*, dando-lhe como entrada um trecho de uma das músicas do compositor Barroco Alemão Johann Sebastian Bach. Todas as telas que iremos exibir serão em modo *Standard*, caso o leitor deseje visualizar as telas da execução desde estudo de caso em outros modos, veja o apêndice D.

6.1 Música Escolhida

Escolhemos um trecho da Seção A do coral *Aus Meines Herzens Grunde*, por se tratar de um clássico, e também por ser construída por um renomado compositor. A seguir, uma ilustração do trecho escolhido.



Figura 34 – Seção A do coral *Aus Meines Herzens Grunde* de Johann S. Bach. Fonte: Oliveira e Aragão (2010)

Fizemos uma análise do trecho da música ilustrado acima, começamos extraíndo as notas agrupadas por tempo de execução, conforme a Tabela 5.

Tabela 5. Notas ordenadas por tempo de execução

Ordem	Notas	Ordem	Notas
1	<i>sol, si, re, sol</i>	13	<i>lá, fa#, lá</i>
2	<i>sol, si, re, sol</i>	14	<i>sol, re, sol, si</i>
3	<i>mi, do, mi</i>	15	<i>re, re, fa#, lá</i>
4	<i>si</i>	16	<i>sol, re, sol, si</i>
5	<i>fá#, lá, re, re</i>	17	<i>fa#, lá, re, re</i>
6	<i>sol, re, si</i>	18	<i>sol, si, mi</i>
7	<i>re, fá#, lá</i>	19	<i>lá, do, fa#, do</i>
8	<i>mi, sol, si, sol</i>	20	<i>si, ré, sol, si</i>
9	<i>do, do, mi, sol</i>	21	<i>do, mi, lá</i>
10	<i>si, re</i>	22	<i>re, re, fa#</i>
11	<i>si</i>	23	<i>do</i>
12	<i>lá, do, mi</i>	24	<i>sol, si, re, sol</i>

Por não formar tríades ou tétrade, de acordo com a teoria musical abordada neste trabalho (ver seção 2.1), da Tabela 5 descartamos as linhas que estão com asterisco, tal qual ilustrado na Tabela 6.

Tabela 6. Notas extraídas, com as linhas em vermelho indicando as que não seram

Ordem	Notas	Ordem	Notas
1	sol,si,re,sol	*13	lá,fa#,lá
2	sol,si,re,sol	14	sol,re,sol,si
*3	mi,do,mi	15	re,re,fa#,lá
*4	si	16	sol,re,sol,si
5	fá#,lá,re,re	17	fa#,lá, re,re
6	sol,re,si	18	sol,si,mi
7	re, fá#,lá	19	lá,do,fa#,do
8	mi,sol,si,sol	20	si,ré,sol,si
9	do,do, mi, sol	21	do,mi,lá
*10	si,re	*22	re,re,fa#
*11	si	*23	do
12	lá,do,mi	24	sol,si,re,sol

Em seguida, para cada linha sem o asterisco na Tabela 6, formarmos os acordes como mostra a tabela abaixo.

Tabela 7. Notas, ordem de execução e acordes formados

Ordem	Notas	Acorde	Ordem	Notas	Acorde
1	sol,si,re,sol	GM	10	re,re,fa#,lá	DM
2	sol,si,re,sol	GM	11	sol,re,sol,si	GM
3	fá#,lá,re,re	DM	12	fa#,lá, re,re	DM
4	sol,re,si	GM	13	sol,si,mi	Em
5	re, fá#,lá	DM	14	lá,do,fa#,do	F#°
6	mi,sol,si,sol	Em	15	si,ré,sol,si	GM
7	do,do, mi, sol	CM	16	do,mi,lá	Am
8	lá,do,mi	Am	17	sol,si,re,sol	GM
9	sol,re,sol,si	GM			

Os acordes que resultaram de nossa análise, como exibidos na Tabela 7 são os mesmos obtidos por Oliveira e Aragão (2010) em seu artigo. A seguir, mostraremos o resultado obtido pelo *Qmusic*.

6.2 Execução do *Qmusic*

Como o sistema *Qmusic* recebe como entrada um arquivo no formato MIDI, utilizamos o sistema MIDILLustrator Maestro (MAESTRO, 2012) que recebe uma partitura e permite a possibilidade de exportação para o formato *.mid*. Desse modo, geramos um arquivo MIDI correspondente a Figura 34.

Após o carregamento do arquivo com extensão *.mid*, clicamos em exibir as notas. O resultado deste processo segue na figura abaixo.

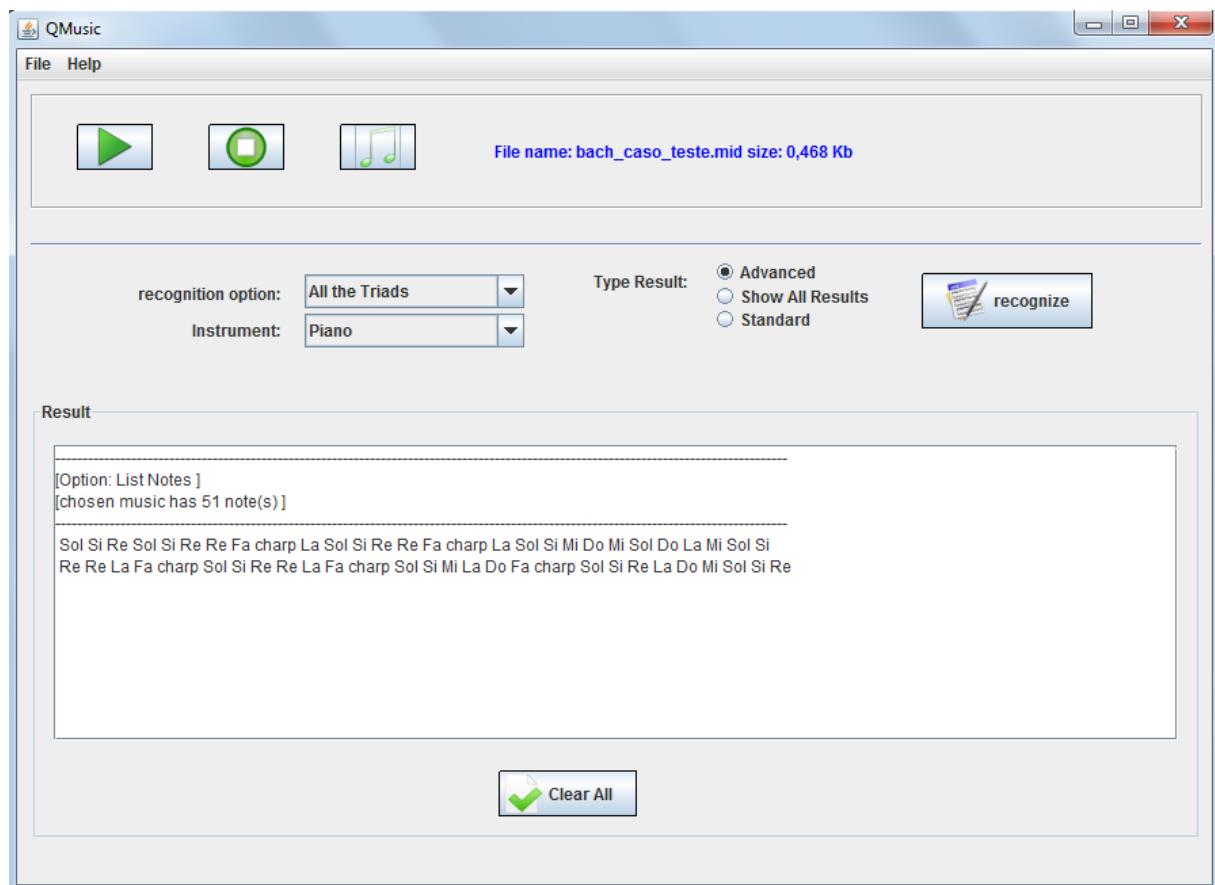


Figura 35 – Lista de notas da Seção A do coral *Aus Meines Herzens Grunde* de Johann S. Bach, com a remoção dos casos em que não se forma acorde (tríade ou tétrade)

Em seguida, realizamos a operação de nomear os acordes, assim como fizemos para obter a Tabela 7, no entanto, neste caso, o trabalho de nomear os acordes foi realizado pelo *Qmusic*. O resultado da operação está logo abaixo.

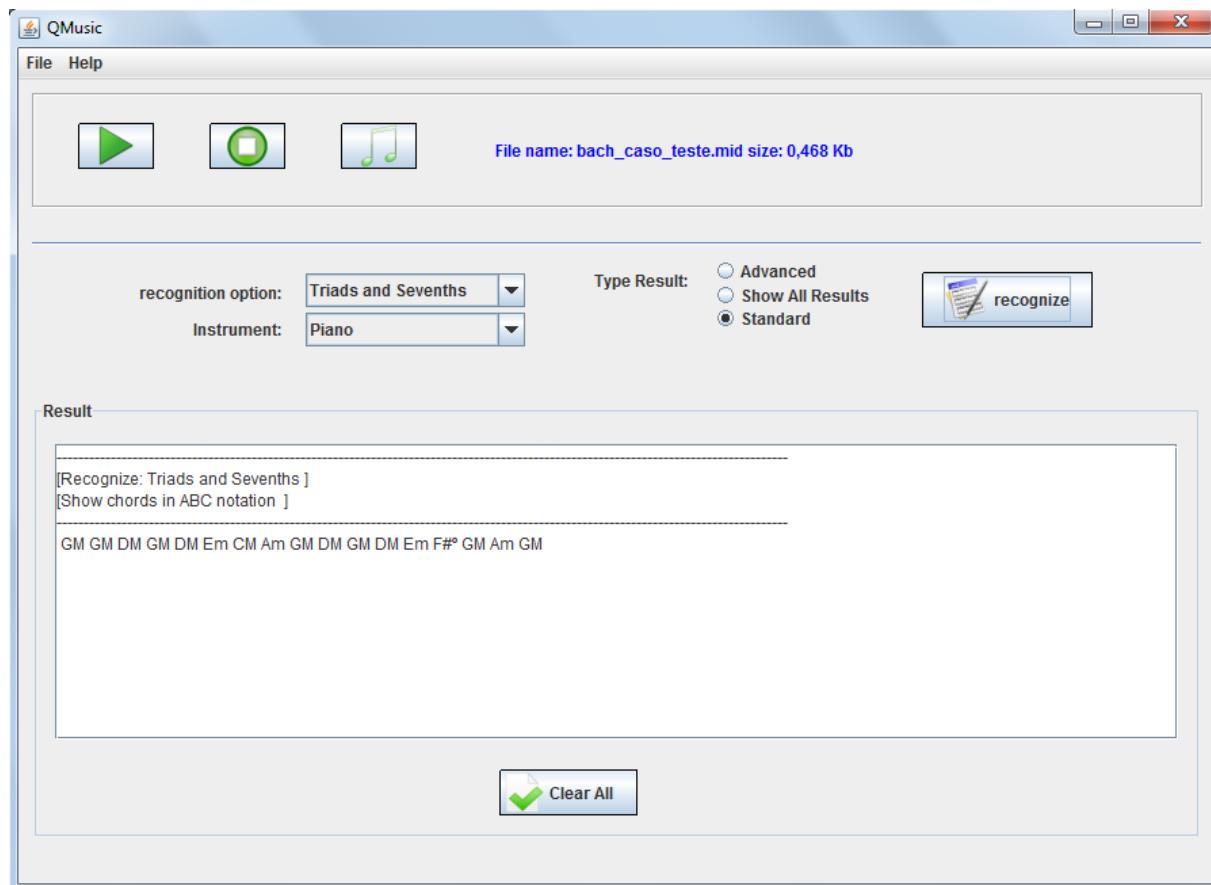


Figura 36 – Lista de Tríades e Tétrade da Seção A do coral *Aus Meines Herzens Grunde* de Bach

Podemos constatar que a sequência de acordes gerada pelo sistema é idêntica a mostrada na Tabela 7.

O *Qmusic* permite ainda a possibilidade de selecionar os tipos de acordes para serem exibidos, conforme explanado na seção 5.2.4. A seguir mostraremos a execução de todas as opções de reconhecimento do *Qmusic*.

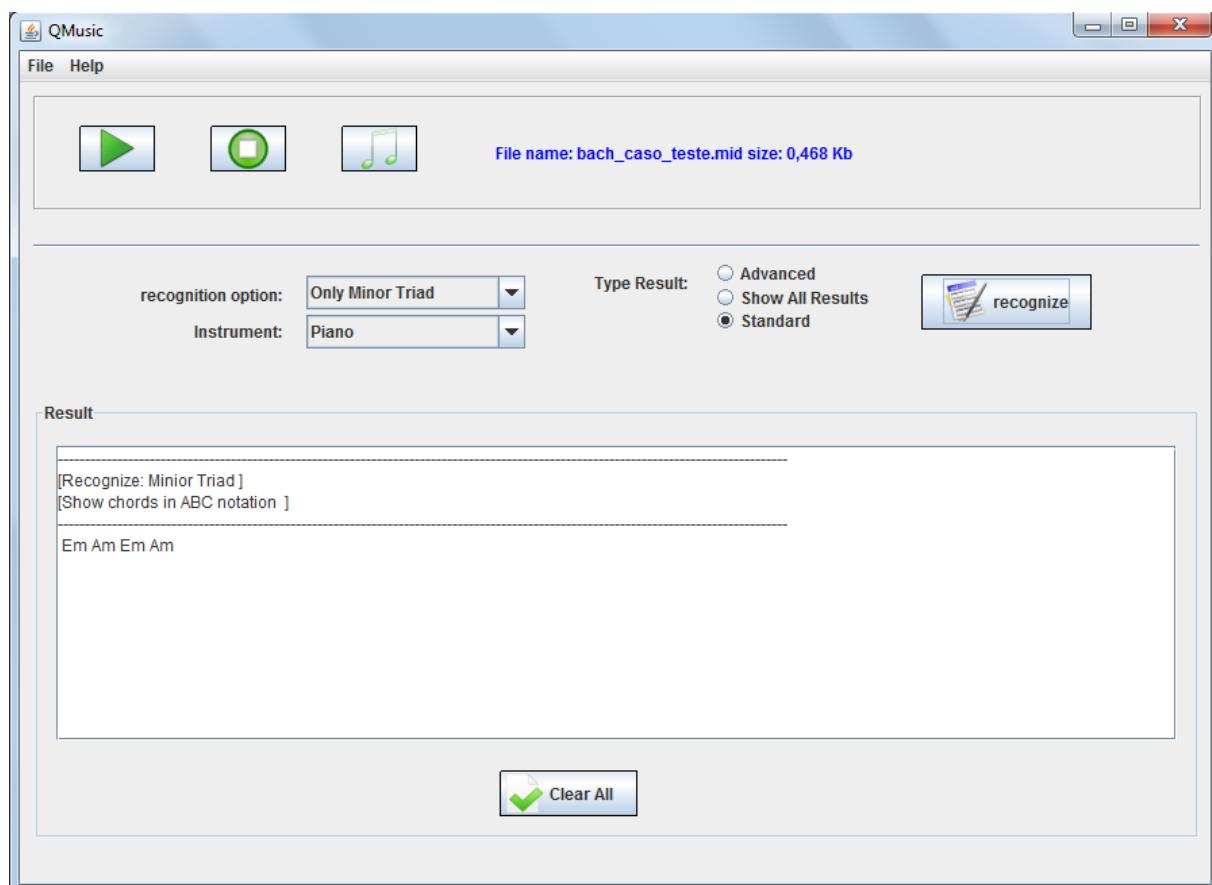


Figura 37 – Lista de Tríades Menores da Seção A do coral *Aus Meines Herzens Grunde* de Bach

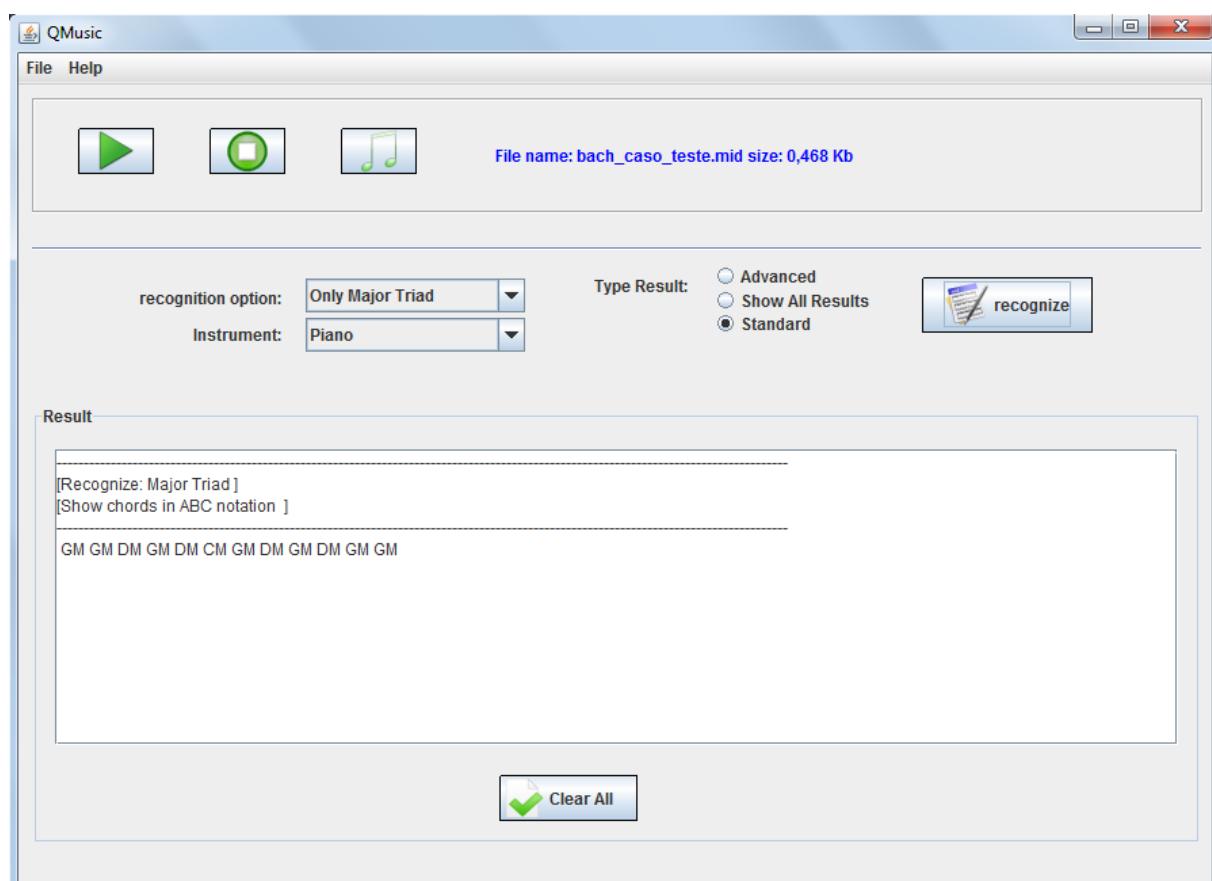


Figura 38 – Lista de Tríades Maiores da Seção A do coral *Aus Meines Herzens Grunde* de Bach

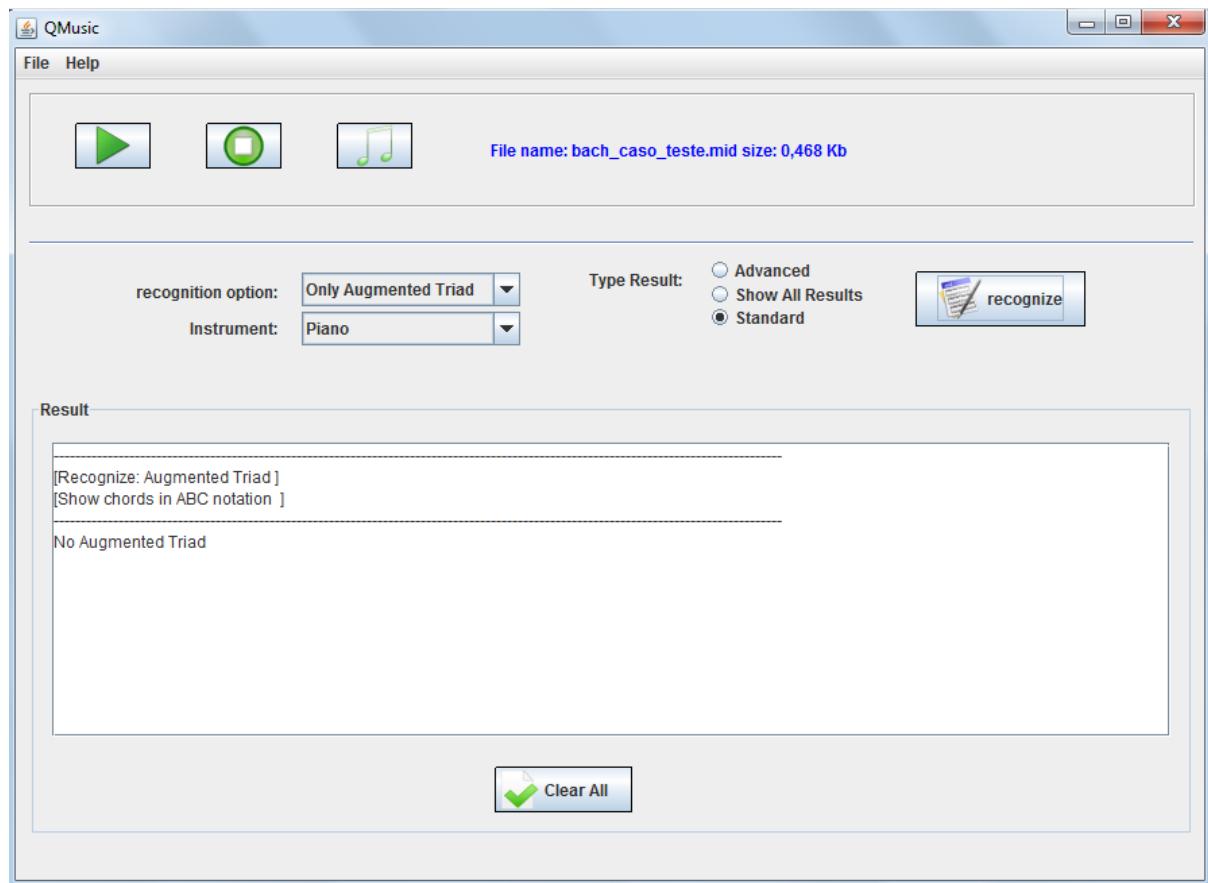


Figura 39 – Lista de Tríades Aumentadas da Seção A do coral *Aus Meines Herzens Grunde* de Bach

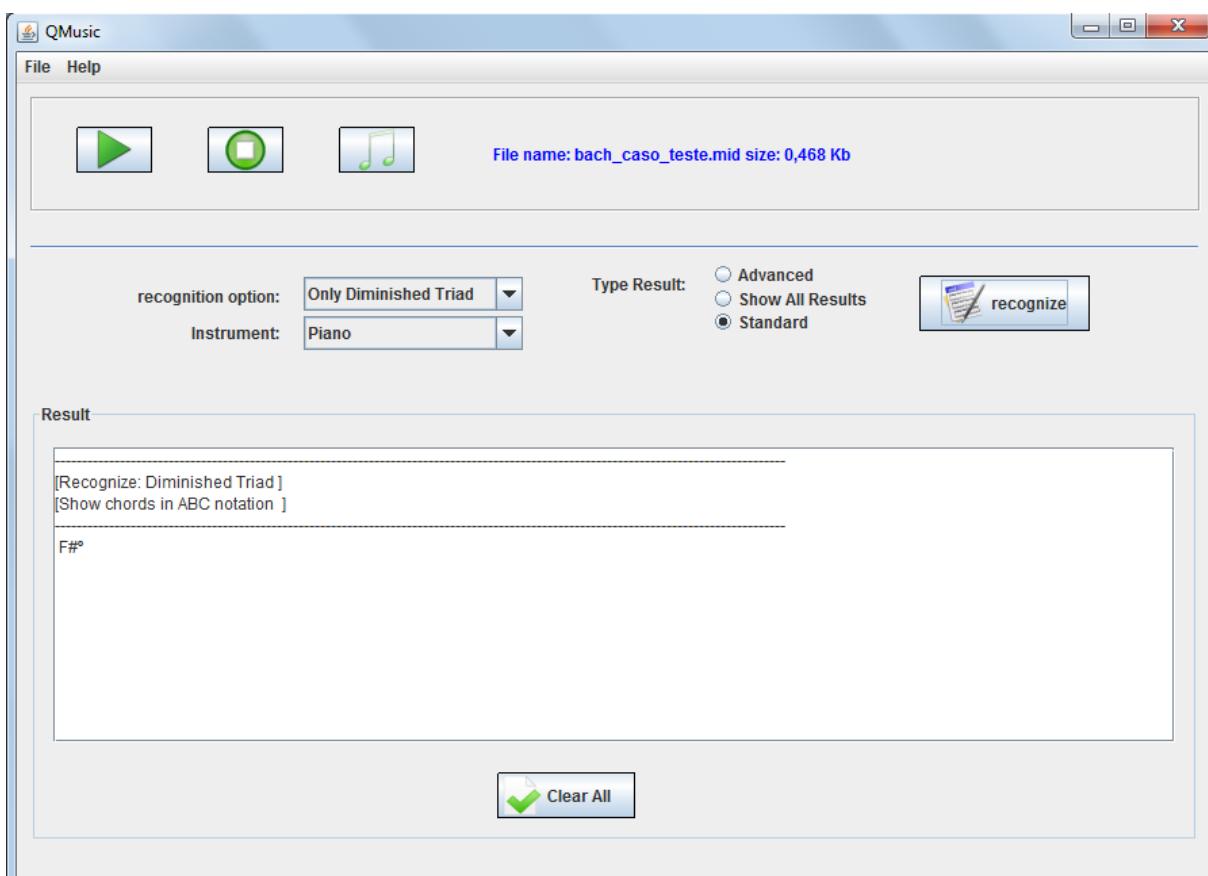


Figura 40 – Lista de Tríades Diminutas da Seção A do coral *Aus Meines Herzens Grunde* de Bach

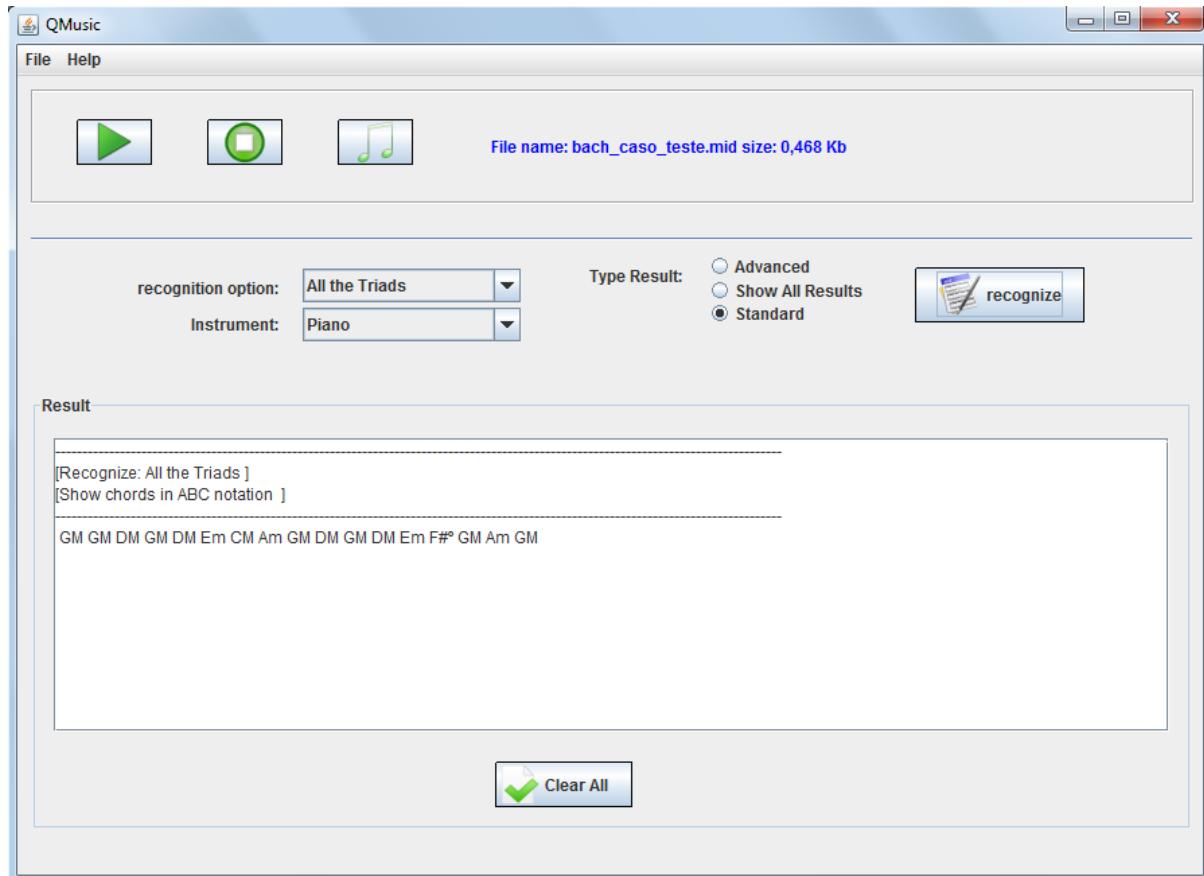


Figura 41 – Lista de Tríades da Seção A do coral *Aus Meines Herzens Grunde* de Bach

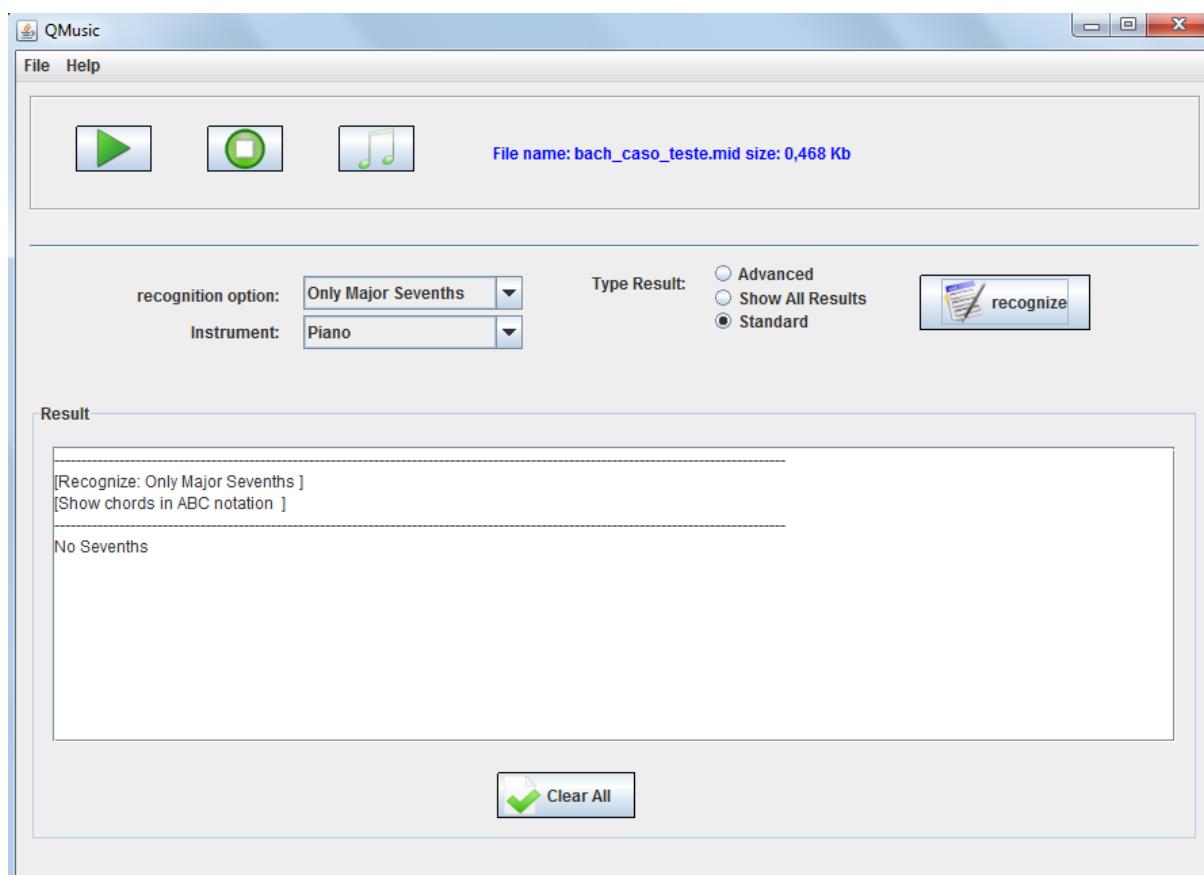


Figura 42 – Lista de Tétrades Maiores da Seção A do coral *Aus Meines Herzens Grunde* de Bach

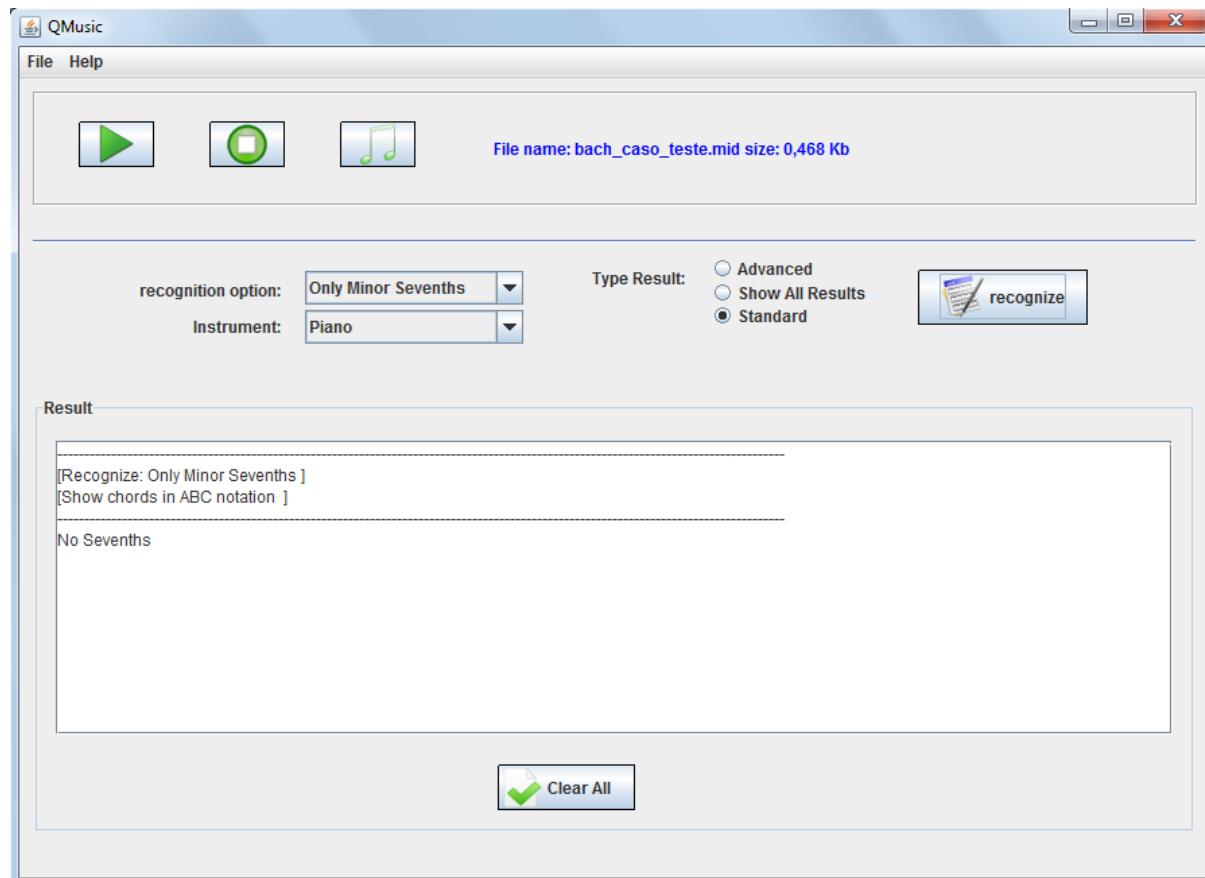


Figura 43 – Lista de Tétrades Menores da Seção A do coral *Aus Meines Herzens Grunde* de Bach

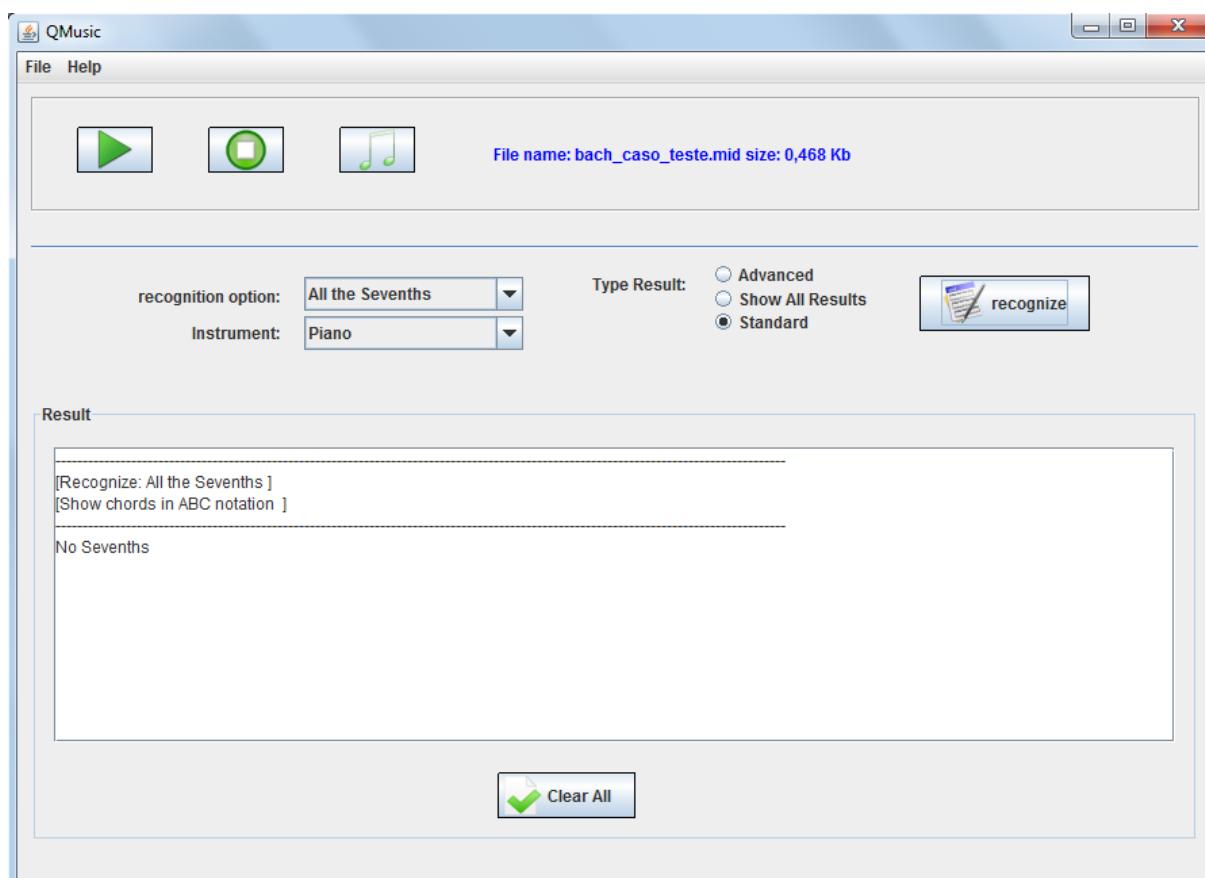


Figura 44 – Lista de Tétrades da Seção A do coral *Aus Meines Herzens Grunde* de Bach

Pela análise das Figuras acima, podemos inferir que o sistema *Qmusic* cumpriu seu papel, ou seja, reconheceu corretamente os acordes (Tríades e Tétrade) dada uma sequência de notas, e permitiu também, várias opções de reconhecimento.

Ao utilizar o sistema, o usuário poderá inclusive filtrar apenas Tríades Diminutas ou Tétrade, conforme seu interesse na música em questão.

7 CONSIDERAÇÕES FINAIS

Ao início deste trabalho, traçamos como propósito desenvolver um Autômato que reconhecesse os acordes (tríades e tétrade) de uma sequência de notas retiradas de um arquivo MIDI, considerando notas enarmônicas. A partir desse Autômato, construir um sistema *desktop* Java SE que o utiliza como rotina para exibir ao usuário a sequência de acordes.

Para sua realização, inicialmente modelamos o problema de nomeação de acordes com Teoria dos Autômatos, mais especificamente, Autômatos Finitos com Transições Vazias. Definimos, então, quem eram seus elementos: alfabeto, estados, estado inicial, estado final e função de transição. Em seguida, desenvolvemos um sistema que utiliza o autômato criado como rotina. O sistema recebe como entrada uma música no formato MIDI e retorna ao usuário a sequência de acordes desta música.

Como podemos observar no capítulo voltado ao estudo de caso, a ferramenta se portou de forma coerente em uma instância do problema, encontrando corretamente os acordes de um trecho de uma das músicas do famoso compositor Bach. Contudo, melhorias podem ser realizadas, por exemplo, a possibilidade de imprimir o resultado. Destacamos também como trabalhos futuros:

- Fazer melhorias na interface e usabilidade do sistema, para que ele possa ser disponibilizado gratuitamente a quem tenha interesse em utilizá-lo;
- Utilizar outros tipos de formatos de entrada para o sistema, além da Tecnologia MIDI, usar as Tecnologias WAV e MP3;
- Empregar outras formas de geração de acordes além de tríades e tétrade, como por exemplo, utilizar intervalos de nona e as inversões;
- Desenvolver um manual para o sistema, objetivando a melhoria em seu uso pelos usuários;

REFERÊNCIAS

- ARAGÃO, F.E.F. [RASCUNHO] **A Fundamentação Lógica do Tonalismo**, 2011. Tese (Doutorado em Computação) – Departamento de Computação da UFC, Fortaleza, 2011.
- CARDOSO, B.; MASCARENHAS, M. **Curso completo de teoria musical e solfejo**. São Paulo – Rio de Janeiro: Irmãos Vitalle, 1996.
- CUNHA, U. S. G. C. **Um Ambiente Híbrido Inteligente para Previsão de Acordes Musicais em Tempo Real**, 1999. Recife: UFPE. Disponível em : <<http://www.cin.ufpe.br/~posgraduacao>>. Acesso em: 29 de agosto de 2011.
- HÖGBERG, J. **Wind in the Willows**, 2005. Sweden: Department of Computing Science, Umea University. Disponível em: <http://www8.cs.umu.se/index_eng.html>. Acesso em: 01 de setembro de 2011.
- HOPCROFT, E. J.; MOTWANI, R.; ULLMAN; D. J. **INTRODUÇÃO A TEORIA DE AUTÔMATOS, LINGUAGENS E COMPUTAÇÃO**, 2^a Ed. Rio de Janeiro: Elsevier, 2002.
- JONES, C. S. **ENAHARMONIC SPELLING**, 2011. Disponível em: <http://cnx.org/content/expanded Browse_authors?b_start:int=50&letter=S&author=Catherine>. Acesso em: 27 de Agosto de 2011.
- MACHADO, A. C. **Tradutor de Arquivos MIDI para Texto Utilizando Linguagem Funcional CLEAN**, 2001. Uberlândia: UFU. Disponível em: <<http://www.demac.ufu.br/>>. Acesso em: 28 de Agosto de 2011. Acesso em 10 de setembro de 2011.
- MAJCHRZAK, M.; GINGRAS, B. **Importance of Enharmonic Tone Spelling in Computational Analysis of Tonal Structure**, 2009. 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009). Disponível em: <https://jyx.jyu.fi/dspace/bitstream/handle/123456789/20894/urn_nbn_fi_jyu-2009411276.pdf?sequence=1>. Acesso em: 01 de setembro de 2011.
- MEDAGLIA, J. **Música, Maestro! Do Canto Gregoriano ao Sintetizador**. Globo, 2008.
- MELLO, M. S. F. **APOSTILA DE TEORIA MUSICAL**, 2010. Disponível em: <<http://www.marcelomelloweb.inghost.net/>>. Acesso em: 29 de Agosto de 2011.
- MENEZES, P. B. **Linguagens Formais e Autômatos**, 5. Ed. Porto Alegre, Bookman: Instituto de Informática da UFRGS, 2008.
- MESSICK, P. **MAXIMUM MIDI: music applications in C++**. USA: MANNING, 1997.
- MIDIIllustrator Maestro(MAESTRO). Executável disponível em: <<http://www.midiillustrator.com/download.htm>>. Acesso em 12 de setembro de 2011.
- MIDI MANUFACTURES ASSOCIATE (MIDI). **The MIDI Specification**. Disponível em: <<http://www.midi.org/>>. Acesso em: 27 de Agosto de 2011.

MORAES, L. M. A. **Solfeggiare: Um compilador para Partituras Musicais, 2006.** Escola Politécnica de Pernambuco. Disponível em <<http://dsc.upc.br/~tcc/>>. Acesso em 28 de Agosto de 2011.

OLIVEIRA, L. J. P; ARAGÃO, F.E.F. **Conversão diatônica entre sistemas riemannianos não-redundantes.** In: XX CONGRESSO DA ASSOCIAÇÃO NACIONAL DE PESQUISA E PÓS-GRADUAÇÃO EM MÚSICA, 2010, Florianópolis. **Anais da ANPPOM:** Editora UDESC, 2010. p.1558-1564.

RATTON, M. **MIDI - Especificação 1.0, 1996.** Disponível em: <<http://www.music-center.com.br/index.php?view=article>>. Acesso em: 01 de setembro de 2011.

RANGEL, J. L. **Apostila de Linguagens Formais e Autômatos, 2001.** Disponível em: <<http://www-di.inf.puc-rio.br/~rangel/>>. Acesso em: 27 de setembro de 2011.

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO (SBC). Disponível em: <<http://www.sbc.org.br/>>. Acesso em: 01 de setembro de 2011.

SCHÖENBERG, A. **Armonia.** Real Musical Editores, Madrid: 1979.

Sistema online 8Notes (8NOTES). Disponível em <http://www.8notes.com/articles/major_chords.asp>. Acesso em 29 de agosto de 2011.

VILAR, P. L. **AHO - Analisador Harmônico Otimista.** UFC, Fortaleza, 2005.

APÊNDICES

7.1 APÊNDICE A – Reconcedor de Tríade

a. Autômato que Reconhece Tríade Maior

O AFN ϵ que reconhece Tríade Maior, denotamos como T_{MA} , é definido formalmente como segue: $T_{MA} = \langle \sum_{t_ma}, Q_{t_ma}, F, \delta_{t_ma}, Q_0 \rangle$, em que:

- $\sum_{t_ma} = \{do, re, mi, fa, sol, la, si, do\#, re\#, mi\#, fa\#, sol\#, la\#, si\#, do\flat, re\flat, mi\flat, fa\flat, sol\flat, la\flat, si\flat, do\#\#, re\#\#, fa\#\#, sol\#\#\};$
- $Q_{t_ma} = \{q_0, DO, DO_2, DO_3, RE, RE_2, RE_3, MI, MI_2, MI_3, FA, FA_2, FA_3, SOL, SOL_2, SOL_3, LA, LA_2, LA_3, SI, SI_2, SI_3, DO\#, DO\#_2, DO\#_3, RE\#, RE\#_2, RE\#_3, MI\#, MI\#_2, MI\#_3, FA\#, FA\#_2, FA\#_3, SOL\#, SOL\#_2, SOL\#_3, LA\#, LA\#_2, LA\#_3, SI\#, SI\#_2, SI\#_3, DO\flat, DO\flat_2, RE\flat, RE\flat_2, MI\flat, MI\flat_2, MI\flat_3, FA\flat, SOL\flat, SOL\flat_2, LA\flat, LA\flat_2, LA\flat_3, SI\flat, SI\flat_2, SI\flat_3, DO\#\#, RE\#\#, FA\#\#, FA\#\#_2, SOL\#\#\}, q_f\};$
- $Q_0 = q_0;$
- $F = \{q_f\};$
- δ_{t_ma} é como segue:

<ul style="list-style-type: none"> ○ $\delta_{t_ma}(q_0, do\#) = \{DO\#}\}$ ○ $\delta_{t_ma}(q_0, si\#\#\}) = \{DO\#\}$ ○ $\delta_{t_ma}(q_0, re\flat) = \{DO\#}\}$ ○ $\delta_{t_ma}(DO\#, mi\#) = \{MI\#}\}$ ○ $\delta_{t_ma}(DO\#, fa) = \{MI\#}\}$ ○ $\delta_{t_ma}(DO\#, sol\flat) = \{MI\#}\}$ ○ $\delta_{t_ma}(MI\#, sol\#) = \{SOL\#}\}$ ○ $\delta_{t_ma}(MI\#, la\flat) = \{SOL\#}\}$ ○ $\delta_{t_ma}(SOL\#, \varepsilon) = \{q_f\}$ ○ $\delta_{t_ma}(q_0, do\flat) = \{DO\flat}\}$ ○ $\delta_{t_ma}(q_0, si) = \{DO\flat}\}$ ○ $\delta_{t_ma}(q_0, la\#\#\}) = \{DO\flat}\}$ ○ $\delta_{t_ma}(DO\flat, mi\flat) = \{MI\flat}\}$ ○ $\delta_{t_ma}(DO\flat, re\#) = \{MI\flat}\}$ ○ $\delta_{t_ma}(DO\flat, fa\flat) = \{MI\flat}\}$ ○ $\delta_{t_ma}(MI\flat, sol\flat) = \{SOL\flat}\}$ ○ $\delta_{t_ma}(MI\flat, mi\#\#\}) = \{SOL\flat}\}$ ○ $\delta_{t_ma}(MI\flat, fa\#) = \{SOL\flat}\}$ ○ $\delta_{t_ma}(RE, fa\#\#\}) = \{SOL\flat}\}$ ○ $\delta_{t_ma}(RE, mi\#\#\}) = \{SOL\flat}\}$ ○ $\delta_{t_ma}(RE, sol\flat) = \{SOL\flat}\}$ ○ $\delta_{t_ma}(FA\#, la) = \{LA\}$ ○ $\delta_{t_ma}(FA\#, la) = \{LA\}$ ○ $\delta_{t_ma}(FA\#, la) = \{LA\}$ 	<ul style="list-style-type: none"> ○ $\delta_{t_ma}(q_0, do) = \{DO\}$ ○ $\delta_{t_ma}(q_0, si\#) = \{DO\}$ ○ $\delta_{t_ma}(q_0, re\flat\flat) = \{DO\}$ ○ $\delta_{t_ma}(DO, mi) = \{MI\}$ ○ $\delta_{t_ma}(DO, re\#\#\}) = \{MI\}$ ○ $\delta_{t_ma}(DO, fa\flat) = \{MI\}$ ○ $\delta_{t_ma}(MI, sol) = \{SOL\}$ ○ $\delta_{t_ma}(MI, fa\#\#\}) = \{SOL\}$ ○ $\delta_{t_ma}(MI, la\flat\flat) = \{SOL\}$ ○ $\delta_{t_ma}(SOL, \varepsilon) = \{q_f\}$ ○ $\delta_{t_ma}(q_0, re) = \{RE\}$ ○ $\delta_{t_ma}(q_0, do\#\#\}) = \{RE\}$ ○ $\delta_{t_ma}(q_0, mi\flat\flat) = \{RE\}$ ○ $\delta_{t_ma}(RE, fa\#\#\}) = \{FA\#}\}$ ○ $\delta_{t_ma}(RE, mi\#\#\}) = \{FA\#}\}$ ○ $\delta_{t_ma}(RE, sol\flat) = \{FA\#}\}$ ○ $\delta_{t_ma}(FA\#, la) = \{LA\}$ ○ $\delta_{t_ma}(FA\#, la) = \{LA\}$ ○ $\delta_{t_ma}(FA\#, la) = \{LA\}$
--	---

- $\delta_{t_ma}(LA, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, re\#) = \{RE\#\}$
- $\delta_{t_ma}(q_0, fab\mathbf{b}) = \{RE\#\}$
- $\delta_{t_ma}(q_0, mi\mathbf{b}) = \{RE\#\}$
- $\delta_{t_ma}(RE\#, fa\#\#\#) = \{FA\#\#\#}$
- $\delta_{t_ma}(RE\#, sol) = \{FA\#\#\#}$
- $\delta_{t_ma}(RE\#, lab\mathbf{b}) = \{FA\#\#\#}$
- $\delta_{t_ma}(FA\#\#\#, la\#) = \{LA\#\}$
- $\delta_{t_ma}(FA\#\#\#, la\#) = \{LA\#\}$
- $\delta_{t_ma}(FA\#\#\#, la\#) = \{LA\#\}$
- $\delta_{t_ma}(LA\#, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, reb) = \{RE\mathbf{b}\}$
- $\delta_{t_ma}(q_0, do\#\#) = \{RE\mathbf{b}\}$
- $\delta_{t_ma}(q_0, si\#\#\#) = \{RE\mathbf{b}\}$
- $\delta_{t_ma}(RE\mathbf{b}, fa) = \{FA\}$
- $\delta_{t_ma}(RE\mathbf{b}, mi\#\#) = \{FA\}$
- $\delta_{t_ma}(RE\mathbf{b}, sol\mathbf{b}\mathbf{b}) = \{FA\}$
- $\delta_{t_ma}(FA, lab) = \{LA\mathbf{b}\}$
- $\delta_{t_ma}(FA, sol\#\#) = \{LA\mathbf{b}\}$
- $\delta_{t_ma}(LA\mathbf{b}, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, mi) = \{MI_2\}$
- $\delta_{t_ma}(q_0, fab) = \{MI_2\}$
- $\delta_{t_ma}(q_0, re\#\#\#) = \{MI_2\}$
- $\delta_{t_ma}(MI_2, sol\#\#) = \{SOL\#\#_2\}$
- $\delta_{t_ma}(MI_2, lab) = \{SOL\#\#_2\}$
- $\delta_{t_ma}(SOL\#\#_2, si) = \{SI\}$
- $\delta_{t_ma}(SOL\#\#_2, do\mathbf{b}) = \{SI\}$
- $\delta_{t_ma}(SOL\#\#_2, la\#\#\#) = \{SI\}$
- $\delta_{t_ma}(SI, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, mi\#\#) = \{MI\#\#_2\}$
- $\delta_{t_ma}(q_0, fa) = \{MI\#\#_2\}$
- $\delta_{t_ma}(q_0, sol\mathbf{b}\mathbf{b}) = \{MI\#\#_2\}$
- $\delta_{t_ma}(MI\#\#_2, sol\#\#\#) = \{SOL\#\#\#_2\}$
- $\delta_{t_ma}(MI\#\#_2, la) = \{SOL\#\#\#_2\}$
- $\delta_{t_ma}(MI\#\#_2, si\mathbf{b}\mathbf{b}) = \{SOL\#\#\#_2\}$
- $\delta_{t_ma}(SOL\#\#\#_2, si\#\#) = \{SI\#\#_2\}$
- $\delta_{t_ma}(SOL\#\#\#_2, do) = \{SI\#\#_2\}$
- $\delta_{t_ma}(SOL\#\#\#_2, reb\mathbf{b}) = \{SI\#\#_2\}$
- $\delta_{t_ma}(SI\#\#, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, mi\mathbf{b}) = \{MI\mathbf{b}_2\}$
- $\delta_{t_ma}(q_0, re\#\#) = \{MI\mathbf{b}_2\}$
- $\delta_{t_ma}(q_0, fab\mathbf{b}\mathbf{b}) = \{MI\mathbf{b}_2\}$
- $\delta_{t_ma}(MI\mathbf{b}_2, sol) = \{SOL_2\}$
- $\delta_{t_ma}(MI\mathbf{b}_2, fa\#\#\#) = \{SOL_2\}$
- $\delta_{t_ma}(MI\mathbf{b}_2, lab\mathbf{b}) = \{SOL_2\}$
- $\delta_{t_ma}(SOL_2, si\mathbf{b}) = \{SI\mathbf{b}\}$
- $\delta_{t_ma}(SOL_2, do\mathbf{b}\mathbf{b}) = \{SI\mathbf{b}\}$
- $\delta_{t_ma}(SOL_2, la\#\#) = \{SI\mathbf{b}\}$
- $\delta_{t_ma}(SI\mathbf{b}, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, fa) = \{FA_2\}$
- $\delta_{t_ma}(q_0, mi\#\#) = \{FA_2\}$
- $\delta_{t_ma}(q_0, sol\mathbf{b}\mathbf{b}) = \{FA_2\}$
- $\delta_{t_ma}(FA_2, la) = \{LA_2\}$
- $\delta_{t_ma}(FA_2, sol\#\#\#) = \{LA_2\}$
- $\delta_{t_ma}(FA_2, si\mathbf{b}\mathbf{b}) = \{LA_2\}$
- $\delta_{t_ma}(LA_2, do) = \{DO_2\}$
- $\delta_{t_ma}(LA_2, reb\mathbf{b}) = \{DO_2\}$
- $\delta_{t_ma}(LA_2, si\#\#) = \{DO_2\}$
- $\delta_{t_ma}(DO_2, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, fa\#\#) = \{FA\#\#_2\}$
- $\delta_{t_ma}(q_0, mi\#\#\#) = \{FA\#\#_2\}$
- $\delta_{t_ma}(q_0, sol\mathbf{b}) = \{FA\#\#_2\}$
- $\delta_{t_ma}(FA\#\#_2, la\#\#) = \{LA\#\#_2\}$
- $\delta_{t_ma}(FA\#\#_2, si\mathbf{b}) = \{LA\#\#_2\}$
- $\delta_{t_ma}(FA\#\#_2, do\mathbf{b}\mathbf{b}) = \{LA\#\#_2\}$
- $\delta_{t_ma}(LA\#\#_2, do\#\#) = \{DO\#\#_2\}$
- $\delta_{t_ma}(LA\#\#_2, reb) = \{DO\#\#_2\}$
- $\delta_{t_ma}(LA\#\#_2, do\#\#) = \{DO\#\#_2\}$
- $\delta_{t_ma}(LA\#\#_2, si\#\#\#) = \{DO\#\#_2\}$
- $\delta_{t_ma}(DO\#\#_2, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, fab) = \{FAB\}$
- $\delta_{t_ma}(q_0, mi) = \{FAB\}$
- $\delta_{t_ma}(q_0, re\#\#\#) = \{FAB\}$
- $\delta_{t_ma}(FAB, la) = \{LA\mathbf{b}_2\}$
- $\delta_{t_ma}(FAB, sol\#\#) = \{LA\mathbf{b}_2\}$
- $\delta_{t_ma}(LA\mathbf{b}_2, do\mathbf{b}) = \{DO\mathbf{b}_2\}$
- $\delta_{t_ma}(LA\mathbf{b}_2, si) = \{DO\mathbf{b}_2\}$
- $\delta_{t_ma}(LA\mathbf{b}_2, la\#\#\#) = \{DO\mathbf{b}_2\}$
- $\delta_{t_ma}(DO\mathbf{b}_2, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, sol) = \{SOL_3\}$
- $\delta_{t_ma}(q_0, fa\#\#\#) = \{SOL_3\}$
- $\delta_{t_ma}(q_0, lab\mathbf{b}) = \{SOL_3\}$
- $\delta_{t_ma}(SOL_3, si) = \{SI_2\}$

- $\delta_{t_ma}(\text{SOL}_3, la\#\#) = \{\text{SI}_2\}$
- $\delta_{t_ma}(\text{SOL}_3, do\mathbf{b}) = \{\text{SI}_2\}$
- $\delta_{t_ma}(\text{SI}_2, re) = \{\text{RE}_2\}$
- $\delta_{t_ma}(\text{SI}_2, do\#\#) = \{\text{RE}_2\}$
- $\delta_{t_ma}(\text{SI}_2, mi\mathbf{b}\mathbf{b}) = \{\text{RE}_2\}$
- $\delta_{t_ma}(\text{RE}_2, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, sol\#) = \{\text{SOL}\#_3\}$
- $\delta_{t_ma}(q_0, lab) = \{\text{SOL}\#_3\}$
- $\delta_{t_ma}(\text{SOL}\#_3, si\#) = \{\text{SI}\#_2\}$
- $\delta_{t_ma}(\text{SOL}\#_3, do) = \{\text{SI}\#_2\}$
- $\delta_{t_ma}(\text{SOL}\#_3, reb\mathbf{b}) = \{\text{SI}\#_2\}$
- $\delta_{t_ma}(\text{SI}\#_2, re\#) = \{\text{RE}\#_2\}$
- $\delta_{t_ma}(\text{SI}\#_2, mi\mathbf{b}) = \{\text{RE}\#_2\}$
- $\delta_{t_ma}(\text{SI}\#_2, fab\mathbf{b}) = \{\text{RE}\#_2\}$
- $\delta_{t_ma}(\text{RE}\#_2, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, sol\mathbf{b}) = \{\text{SOL}\mathbf{b}_2\}$
- $\delta_{t_ma}(q_0, fa\#) = \{\text{SOL}\mathbf{b}_2\}$
- $\delta_{t_ma}(q_0, mi\#\#) = \{\text{SOL}\mathbf{b}_2\}$
- $\delta_{t_ma}(\text{SOL}\mathbf{b}_2, si) = \{\text{SI}\mathbf{b}_2\}$
- $\delta_{t_ma}(\text{SOL}\mathbf{b}_2, do\mathbf{b}) = \{\text{SI}\mathbf{b}_2\}$
- $\delta_{t_ma}(\text{SOL}\mathbf{b}_2, la\#\#) = \{\text{SI}\mathbf{b}_2\}$
- $\delta_{t_ma}(\text{SI}\mathbf{b}_2, reb) = \{\text{RE}\mathbf{b}_2\}$
- $\delta_{t_ma}(\text{SI}\mathbf{b}_2, do\#) = \{\text{RE}\mathbf{b}_2\}$
- $\delta_{t_ma}(\text{SI}\mathbf{b}_2, si\#\#) = \{\text{RE}\mathbf{b}_2\}$
- $\delta_{t_ma}(\text{RE}\mathbf{b}_2, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, la) = \{\text{LA}_3\}$
- $\delta_{t_ma}(q_0, sol\#\#) = \{\text{LA}_3\}$
- $\delta_{t_ma}(q_0, si\mathbf{b}\mathbf{b}) = \{\text{LA}_3\}$
- $\delta_{t_ma}(\text{LA}_3, do\#) = \{\text{DO}\#_3\}$
- $\delta_{t_ma}(\text{LA}_3, reb) = \{\text{DO}\#_3\}$
- $\delta_{t_ma}(\text{LA}_3, si\#\#) = \{\text{DO}\#_3\}$
- $\delta_{t_ma}(\text{DO}\#_3, mi) = \{\text{MI}_3\}$
- $\delta_{t_ma}(\text{DO}\#_3, re\#\#) = \{\text{MI}_3\}$
- $\delta_{t_ma}(\text{DO}\#_3, fab) = \{\text{MI}_3\}$
- $\delta_{t_ma}(\text{MI}_3, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, la\#) = \{\text{LA}\#_3\}$
- $\delta_{t_ma}(q_0, si\mathbf{b}) = \{\text{LA}\#_3\}$
- $\delta_{t_ma}(q_0, do\mathbf{b}\mathbf{b}) = \{\text{LA}\#_3\}$
- $\delta_{t_ma}(\text{LA}_3, do\#\#) = \{\text{DO}\#\#\}$
- $\delta_{t_ma}(\text{LA}_3, re) = \{\text{DO}\#\#\}$
- $\delta_{t_ma}(\text{LA}_3, mi\mathbf{b}\mathbf{b}) = \{\text{DO}\#\#\}$
- $\delta_{t_ma}(\text{DO}\#\#, mi\#) = \{\text{MI}\#_3\}$
- $\delta_{t_ma}(\text{DO}\#\#, fa) = \{\text{MI}\#_3\}$
- $\delta_{t_ma}(\text{DO}\#\#, sol\mathbf{b}\mathbf{b}) = \{\text{MI}\#_3\}$
- $\delta_{t_ma}(\text{MI}\#_3, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, lab) = \{\text{LAB}_3\}$
- $\delta_{t_ma}(q_0, sol\#) = \{\text{LAB}_3\}$
- $\delta_{t_ma}(\text{LA}_3, do) = \{\text{DO}_3\}$
- $\delta_{t_ma}(\text{LA}_3, si\#) = \{\text{DO}_3\}$
- $\delta_{t_ma}(\text{LA}_3, reb\mathbf{b}) = \{\text{DO}_3\}$
- $\delta_{t_ma}(\text{DO}_3, mi\mathbf{b}) = \{\text{MIb}_3\}$
- $\delta_{t_ma}(\text{DO}_3, re\#) = \{\text{MIb}_3\}$
- $\delta_{t_ma}(\text{DO}_3, fab\mathbf{b}) = \{\text{MIb}_3\}$
- $\delta_{t_ma}(\text{MIb}_3, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, si) = \{\text{SI}_3\}$
- $\delta_{t_ma}(q_0, do\mathbf{b}) = \{\text{SI}_3\}$
- $\delta_{t_ma}(q_0, la\#\#) = \{\text{SI}_3\}$
- $\delta_{t_ma}(\text{SI}_3, re\#) = \{\text{RE}\#_3\}$
- $\delta_{t_ma}(\text{SI}_3, mi\mathbf{b}) = \{\text{RE}\#_3\}$
- $\delta_{t_ma}(\text{SI}_3, fab\mathbf{b}) = \{\text{RE}\#_3\}$
- $\delta_{t_ma}(\text{RE}\#_3, fa\#) = \{\text{FA}\#_3\}$
- $\delta_{t_ma}(\text{RE}\#_3, sol\mathbf{b}) = \{\text{FA}\#_3\}$
- $\delta_{t_ma}(\text{RE}\#_3, mi\#\#) = \{\text{FA}\#_3\}$
- $\delta_{t_ma}(\text{FA}\#_3, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, si\#) = \{\text{SI}\#_3\}$
- $\delta_{t_ma}(q_0, do) = \{\text{SI}\#_3\}$
- $\delta_{t_ma}(q_0, reb\mathbf{b}) = \{\text{SI}\#_3\}$
- $\delta_{t_ma}(\text{SI}\#_3, re\#\#) = \{\text{RE}\#\#\}$
- $\delta_{t_ma}(\text{SI}\#_3, mi) = \{\text{RE}\#\#\}$
- $\delta_{t_ma}(\text{SI}\#_3, fab) = \{\text{RE}\#\#\}$
- $\delta_{t_ma}(\text{RE}\#\#, fa\#\#) = \{\text{FA}\#\#_2\}$
- $\delta_{t_ma}(\text{RE}\#\#, sol) = \{\text{FA}\#\#_2\}$
- $\delta_{t_ma}(\text{RE}\#\#, lab\mathbf{b}) = \{\text{FA}\#\#_2\}$
- $\delta_{t_ma}(\text{FA}\#\#_2, \varepsilon) = \{q_f\}$
- $\delta_{t_ma}(q_0, si\mathbf{b}) = \{\text{SI}\mathbf{b}_3\}$
- $\delta_{t_ma}(q_0, la\#) = \{\text{SI}\mathbf{b}_3\}$
- $\delta_{t_ma}(q_0, do\mathbf{b}\mathbf{b}) = \{\text{SI}\mathbf{b}_3\}$
- $\delta_{t_ma}(\text{SI}_3, re) = \{\text{RE}_3\}$
- $\delta_{t_ma}(\text{SI}_3, mi\mathbf{b}\mathbf{b}) = \{\text{RE}_3\}$
- $\delta_{t_ma}(\text{SI}_3, do\#\#) = \{\text{RE}_3\}$
- $\delta_{t_ma}(\text{RE}_3, fa) = \{\text{FA}_3\}$
- $\delta_{t_ma}(\text{RE}_3, sol\mathbf{b}\mathbf{b}) = \{\text{FA}_3\}$
- $\delta_{t_ma}(\text{RE}_3, mi\#) = \{\text{FA}_3\}$
- $\delta_{t_ma}(\text{FA}_3, \varepsilon) = \{q_f\}$

b. Autômato que Reconhece Tríade Menor

O AFN ϵ que reconhece Tríade Menor, denotamos como T_ME, é definido formalmente como segue: $T_ME = \langle \Sigma, Q_{t_me}, F, \delta_{t_me}, Q_0 \rangle$, em que:

- $\Sigma = \{do, re, mi, fa, sol, la, si, do\#, re\#, mi\#, fa\#, sol\#, la\#, si\#, do\flat, re\flat, mi\flat, fa\flat, sol\flat, la\flat, si\flat, fa\#\#, mi\flat\flat, la\flat\flat, si\flat\flat\}$;
- $Q_{t_me} = \{q_0, DO, DO_2, DO_3, RE, RE_2, RE_3, MI, MI_2, MI_3, FA, FA_2, FA_3, SOL, SOL_2, SOL_3, LA, LA_2, LA_3, SI, SI_2, SI_3, DO\#, DO\#_2, DO\#_3, RE\#, RE\#_2, RE\#_3, MI\#, MI\#_2, FA\#, FA\#_2, FA\#_3, SOL\#, SOL\#_2, SOL\#_3, LA\#, LA\#_2, SI\#, SI\#_2, DO\flat, DO\flat_2, DO\flat_3, RE\flat, RE\flat_2, RE\flat_3, MI\flat, MI\flat_2, MI\flat_3, FA\flat, FA\flat_2, SOL\flat, SOL\flat_2, SOL\flat_3, LA\flat, LA\flat_2, LA\flat_3, SI\flat, SI\flat_2, SI\flat_3, FA\#\#, MI\flat\flat, LA\flat\flat, SI\flat\flat, q_f\}$;
- $Q_0 = q_0$;
- $F = \{q_f\}$;
- δ_{t_me} é como segue:

<ul style="list-style-type: none"> ○ $\delta_{t_me}(q_0, do) = \{DO\}$ ○ $\delta_{t_me}(q_0, re\flat\flat) = \{DO\}$ ○ $\delta_{t_me}(q_0, si\#\#) = \{DO\}$ ○ $\delta_{t_me}(DO, mi\flat) = \{MI\flat\}$ ○ $\delta_{t_me}(DO, fa\flat\flat) = \{MI\flat\}$ ○ $\delta_{t_me}(DO, re\#) = \{MI\flat\}$ ○ $\delta_{t_me}(MI\flat, sol) = \{SOL\}$ ○ $\delta_{t_me}(MI\flat, fa\#\#\#) = \{SOL\}$ ○ $\delta_{t_me}(MI\flat, lab\flat) = \{SOL\}$ ○ $\delta_{t_me}(SOL, \epsilon) = \{q_f\}$ ○ $\delta_{t_me}(q_0, do\#) = \{DO\#\}$ ○ $\delta_{t_me}(q_0, re\flat) = \{DO\#\}$ ○ $\delta_{t_me}(q_0, si\#\#\#) = \{DO\#\}$ ○ $\delta_{t_me}(DO\#, mi) = \{MI\}$ ○ $\delta_{t_me}(DO\#, re\#\#\#) = \{MI\}$ ○ $\delta_{t_me}(DO\#, fa\flat) = \{MI\}$ ○ $\delta_{t_me}(MI, sol\#) = \{SOL\#\}$ ○ $\delta_{t_me}(MI, lab\flat) = \{SOL\#\}$ ○ $\delta_{t_me}(SOL\#, \epsilon) = \{q_f\}$ ○ $\delta_{t_me}(q_0, do\flat) = \{DO\flat\}$ ○ $\delta_{t_me}(q_0, si) = \{DO\flat\}$ ○ $\delta_{t_me}(q_0, la\#\#\#) = \{DO\flat\}$ 	<ul style="list-style-type: none"> ○ $\delta_{t_me}(DO\flat, mi\flat\flat) = \{MI\flat\flat\}$ ○ $\delta_{t_me}(DO\flat, re) = \{MI\flat\flat\}$ ○ $\delta_{t_me}(DO\flat, do\#\#\#) = \{MI\flat\flat\}$ ○ $\delta_{t_me}(MI\flat\flat, sol\flat) = \{SOL\flat\}$ ○ $\delta_{t_me}(MI\flat\flat, fa\#) = \{SOL\flat\}$ ○ $\delta_{t_me}(MI\flat\flat, mi\#\#\#) = \{SOL\flat\}$ ○ $\delta_{t_me}(SOL\flat, \epsilon) = \{q_f\}$ ○ $\delta_{t_me}(q_0, re) = \{RE\}$ ○ $\delta_{t_me}(q_0, mi\flat\flat) = \{RE\}$ ○ $\delta_{t_me}(q_0, do\#\#\#) = \{RE\}$ ○ $\delta_{t_me}(RE, mi\#) = \{FA\}$ ○ $\delta_{t_me}(RE, fa) = \{FA\}$ ○ $\delta_{t_me}(RE, sol\flat\flat) = \{FA\}$ ○ $\delta_{t_me}(FA, la) = \{LA\}$ ○ $\delta_{t_me}(FA, si\flat\flat) = \{LA\}$ ○ $\delta_{t_me}(FA, sol\#\#\#) = \{LA\}$ ○ $\delta_{t_me}(LA, \epsilon) = \{q_f\}$ ○ $\delta_{t_me}(q_0, re\#) = \{RE\#}\}$ ○ $\delta_{t_me}(q_0, mi\flat) = \{RE\#}\}$ ○ $\delta_{t_me}(q_0, fa\flat\flat) = \{RE\#}\}$ ○ $\delta_{t_me}(RE, fa\#) = \{FA\#}\}$ ○ $\delta_{t_me}(RE, mi\#\#\#) = \{FA\#}\}$
---	---

- $\delta_{t_me}(RE, solb) = \{FA\#}\}$
- $\delta_{t_me}(FA\#, la\#} = \{LA\#\}$
- $\delta_{t_me}(FA\#, sib) = \{LA\#\}$
- $\delta_{t_me}(FA\#, dobb) = \{LA\#\}$
- $\delta_{t_me}(LA\#, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, reb) = \{REb\}$
- $\delta_{t_me}(q_0, do\#} = \{REb\}$
- $\delta_{t_me}(q_0, si\#\#} = \{REb\}$
- $\delta_{t_me}(REb, fab) = \{FAb\}$
- $\delta_{t_me}(REb, mi) = \{FAb\}$
- $\delta_{t_me}(REb, re\#\#\#} = \{FAb\}$
- $\delta_{t_me}(FAb, lab) = \{LAb\}$
- $\delta_{t_me}(FAb, sol\#} = \{LAb\}$
- $\delta_{t_me}(LAb, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, mi) = \{MI_2\}$
- $\delta_{t_me}(q_0, fab) = \{MI_2\}$
- $\delta_{t_me}(q_0, re\#\#\#} = \{MI_2\}$
- $\delta_{t_me}(MI_2, sol) = \{SOL_2\}$
- $\delta_{t_me}(MI_2, fa\#\#\#} = \{SOL_2\}$
- $\delta_{t_me}(MI_2, labb) = \{SOL_2\}$
- $\delta_{t_me}(SOL_2, si) = \{SI\}$
- $\delta_{t_me}(SOL_2, do\#} = \{SI\}$
- $\delta_{t_me}(SOL_2, la\#\#\#} = \{SI\}$
- $\delta_{t_me}(SI, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, mi\#} = \{MI\#\}$
- $\delta_{t_me}(q_0, fa) = \{MI\#\}$
- $\delta_{t_me}(q_0, solbb) = \{MI\#\}$
- $\delta_{t_me}(MI\#, sol\#} = \{SOL\#_2\}$
- $\delta_{t_me}(MI\#, lab) = \{SOL\#_2\}$
- $\delta_{t_me}(SOL\#_2, si\#} = \{SI\#\}$
- $\delta_{t_me}(SOL\#_2, do) = \{SI\#\}$
- $\delta_{t_me}(SOL\#_2, reb) = \{SI\#\}$
- $\delta_{t_me}(SI\#, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, mib) = \{Mib_2\}$
- $\delta_{t_me}(q_0, re\#} = \{Mib_2\}$
- $\delta_{t_me}(q_0, fabb) = \{Mib_2\}$
- $\delta_{t_me}(Mib_2, solb) = \{SOLb_2\}$
- $\delta_{t_me}(Mib_2, fa\#} = \{SOLb_2\}$
- $\delta_{t_me}(Mib_2, mi\#\#\#} = \{SOLb_2\}$
- $\delta_{t_me}(SOLb_2, sib) = \{SIB\}$
- $\delta_{t_me}(SOLb_2, dobb) = \{SIB\}$
- $\delta_{t_me}(SOLb_2, la\#} = \{SIB\}$
- $\delta_{t_me}(SIB, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, fa) = \{FA_2\}$
- $\delta_{t_me}(q_0, mi\#\#} = \{FA_2\}$
- $\delta_{t_me}(q_0, solb) = \{FA_2\}$
- $\delta_{t_me}(FA_2, lab) = \{LAb_2\}$
- $\delta_{t_me}(FA_2, sol\#} = \{LAb_2\}$
- $\delta_{t_me}(LAb_2, do) = \{DO_3\}$
- $\delta_{t_me}(LAb_2, si\#\#} = \{DO_3\}$
- $\delta_{t_me}(LAb_2, reb) = \{DO_3\}$
- $\delta_{t_me}(DO_3, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, fa\#\#} = \{FA\#_2\}$
- $\delta_{t_me}(q_0, mi\#\#\#} = \{FA\#_2\}$
- $\delta_{t_me}(q_0, solb) = \{FA\#_2\}$
- $\delta_{t_me}(FA\#_2, la) = \{LA_2\}$
- $\delta_{t_me}(FA\#_2, sol\#\#\#} = \{LA_2\}$
- $\delta_{t_me}(FA\#_2, sib) = \{LA_2\}$
- $\delta_{t_me}(LA_2, do\#} = \{DO\#_2\}$
- $\delta_{t_me}(LA_2, reb) = \{DO\#_2\}$
- $\delta_{t_me}(LA_2, si\#\#\#} = \{DO\#_2\}$
- $\delta_{t_me}(DO\#_2, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, fab) = \{FAb_2\}$
- $\delta_{t_me}(q_0, mi) = \{FAb_2\}$
- $\delta_{t_me}(q_0, re\#\#\#} = \{FAb_2\}$
- $\delta_{t_me}(FAb_2, labb) = \{LAbb\}$
- $\delta_{t_me}(FAb_2, sol) = \{LAbb\}$
- $\delta_{t_me}(FAb_2, fab\#\#} = \{LAbb\}$
- $\delta_{t_me}(LAbb, do\#} = \{DOB_2\}$
- $\delta_{t_me}(LAbb, si) = \{DOB_2\}$
- $\delta_{t_me}(LAbb, la\#\#\#} = \{DOB_2\}$
- $\delta_{t_me}(DOB_2, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, sol) = \{SOL_3\}$
- $\delta_{t_me}(q_0, fa\#\#\#} = \{SOL_3\}$
- $\delta_{t_me}(q_0, labb) = \{SOL_3\}$
- $\delta_{t_me}(SOL_3, sib) = \{SIB_2\}$
- $\delta_{t_me}(SOL_3, dobb) = \{SIB_2\}$
- $\delta_{t_me}(SOL_3, la\#} = \{SIB_2\}$

- $\delta_{t_me}(SIB_2, re) = \{RE_2\}$
- $\delta_{t_me}(SIB_2, do\#\#) = \{RE_2\}$
- $\delta_{t_me}(SIB_2, mi\#\#\b) = \{RE_2\}$
- $\delta_{t_me}(RE_2, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, sol\#\#) = \{SOL\#_3\}$
- $\delta_{t_me}(q_0, lab) = \{SOL\#_3\}$
- $\delta_{t_me}(SOL\#_3, si) = \{SI_2\}$
- $\delta_{t_me}(SOL\#_3, dob) = \{SI_2\}$
- $\delta_{t_me}(SOL\#_3, la\#\#) = \{SI_2\}$
- $\delta_{t_me}(SI_2, mi\#\b) = \{RE\#_2\}$
- $\delta_{t_me}(SI_2, fab\#\b) = \{RE\#_2\}$
- $\delta_{t_me}(SI_2, re\#\#) = \{RE\#_2\}$
- $\delta_{t_me}(RE\#_2, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, sol\#\b) = \{SOL\#_3\}$
- $\delta_{t_me}(q_0, fa\#\#) = \{SOL\#_3\}$
- $\delta_{t_me}(q_0, mi\#\#\#) = \{SOL\#_3\}$
- $\delta_{t_me}(SOL\#_3, si\#\#\b) = \{SIB\#_3\}$
- $\delta_{t_me}(SOL\#_3, la) = \{SIB\#_3\}$
- $\delta_{t_me}(SOL\#_3, sol\#\#\#) = \{SIB\#_3\}$
- $\delta_{t_me}(SIB\#_3, re\#\b) = \{REB_2\}$
- $\delta_{t_me}(SIB\#_3, do\#\#) = \{REB_2\}$
- $\delta_{t_me}(SIB\#_3, si\#\#\#) = \{REB_2\}$
- $\delta_{t_me}(REB_2, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, la) = \{LA_3\}$
- $\delta_{t_me}(q_0, sol\#\#\#) = \{LA_3\}$
- $\delta_{t_me}(q_0, si\#\#\b) = \{LA_3\}$
- $\delta_{t_me}(LA_3, do) = \{DO_3\}$
- $\delta_{t_me}(LA_3, si\#\#) = \{DO_3\}$
- $\delta_{t_me}(LA_3, reb\#\b) = \{DO_3\}$
- $\delta_{t_me}(DO_3, mi) = \{MI_3\}$
- $\delta_{t_me}(DO_3, re\#\#\#) = \{MI_3\}$
- $\delta_{t_me}(DO_3, fab\#\b) = \{MI_3\}$
- $\delta_{t_me}(MI_3, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, la\#\#) = \{LA\#_2\}$
- $\delta_{t_me}(q_0, si\#\b) = \{LA\#_2\}$
- $\delta_{t_me}(q_0, dob\#\b) = \{LA\#_2\}$
- $\delta_{t_me}(LA\#_2, do\#\#) = \{DO\#_3\}$
- $\delta_{t_me}(LA\#_2, reb\#\b) = \{DO\#_3\}$
- $\delta_{t_me}(LA\#_2, si\#\#\#) = \{DO\#_3\}$
- $\delta_{t_me}(DO\#_3, mi\#\#) = \{MI\#_2\}$
- $\delta_{t_me}(DO\#_3, fa) = \{MI\#_2\}$
- $\delta_{t_me}(DO\#_3, sol\#\b) = \{MI\#_2\}$
- $\delta_{t_me}(MI\#_2, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, lab) = \{LAB_3\}$
- $\delta_{t_me}(q_0, sol\#\#) = \{LAB_3\}$
- $\delta_{t_me}(LAB_3, dob) = \{DOB_3\}$
- $\delta_{t_me}(LAB_3, si) = \{DOB_3\}$
- $\delta_{t_me}(LAB_3, la\#\#\#) = \{DOB_3\}$
- $\delta_{t_me}(DOB_3, mi\#\b) = \{MIb_3\}$
- $\delta_{t_me}(DOB_3, re\#\#) = \{MIb_3\}$
- $\delta_{t_me}(DOB_3, fab\#\b) = \{MIb_3\}$
- $\delta_{t_me}(MIb_3, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, si) = \{SI_3\}$
- $\delta_{t_me}(q_0, dob) = \{SI_3\}$
- $\delta_{t_me}(q_0, la\#\#\#) = \{SI_3\}$
- $\delta_{t_me}(SI_3, re) = \{RE_3\}$
- $\delta_{t_me}(SI_3, do\#\#) = \{RE_3\}$
- $\delta_{t_me}(SI_3, mi\#\#\b) = \{RE_3\}$
- $\delta_{t_me}(RE_3, fa\#\#) = \{FA\#_3\}$
- $\delta_{t_me}(RE_3, sol\#\b) = \{FA\#_3\}$
- $\delta_{t_me}(RE_3, mi\#\#\#) = \{FA\#_3\}$
- $\delta_{t_me}(q_0, si\#\#) = \{SI\#_2\}$
- $\delta_{t_me}(q_0, do) = \{SI\#_2\}$
- $\delta_{t_me}(q_0, reb\#\b) = \{SI\#_2\}$
- $\delta_{t_me}(SI\#_2, re\#\#) = \{RE\#_3\}$
- $\delta_{t_me}(SI\#_2, mi\#\b) = \{RE\#_3\}$
- $\delta_{t_me}(SI\#_2, fab\#\b) = \{RE\#_3\}$
- $\delta_{t_me}(RE\#_3, fa\#\#\#) = \{FA\#\#\#}$
- $\delta_{t_me}(RE\#_3, sol) = \{FA\#\#\#}$
- $\delta_{t_me}(RE\#_3, lab\#\b) = \{FA\#\#\#}$
- $\delta_{t_me}(FA\#\#\#, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(q_0, si\#\b) = \{SIB_3\}$
- $\delta_{t_me}(q_0, dob\#\b) = \{SIB_3\}$
- $\delta_{t_me}(q_0, la\#\#) = \{SIB_3\}$
- $\delta_{t_me}(SIB_3, reb) = \{REB_3\}$
- $\delta_{t_me}(SIB_3, do\#\#) = \{REB_3\}$
- $\delta_{t_me}(SIB_3, si\#\#\#) = \{REB_3\}$
- $\delta_{t_me}(REB_3, fa) = \{FA_3\}$
- $\delta_{t_me}(REB_3, mi\#\#) = \{FA_3\}$
- $\delta_{t_me}(REB_3, sol\#\b) = \{FA_3\}$
- $\delta_{t_me}(FA_3, \varepsilon) = \{q_f\}$
- $\delta_{t_me}(FA\#_3, \varepsilon) = \{q_f\}$

c. Autômato que Reconhece Tríade Aumentada

O AFN ϵ que reconhece Tríade Aumentada, denotamos como T_AUM, é definido formalmente como segue: $T_{AUM} = \langle \sum, Q_{t_aum}, F, \delta_{t_aum}, Q_0 \rangle$, em que:

- $\Sigma = \{do, re, mi, fa, sol, la, si, do\#, re\#, mi\#, fa\#, sol\#, la\#, si\#, do\flat, re\flat, mi\flat, fa\flat, sol\flat, la\flat, si\flat, do\#\#, re\#\#, mi\#\#, fa\#\#, sol\#\#, la\#\#, si\#\#\};$
- $Q_{t_aum} = \{q_0, DO, DO_2, DO_3, RE, RE_2, RE_3, MI, MI_2, MI_3, FA, FA_2, SOL, SOL_2, SOL_3, LA, LA_2, LA_3, SI, SI_2, SI_3, DO\#, DO\#_2, DO\#_3, RE\#, RE\#_2, RE\#_3, MI\#, MI\#_2, MI\#_3, FA\#, FA\#_2, FA\#_3, SOL\#, SOL\#_2, SOL\#_3, LA\#, LA\#_2, LA\#_3, SI\#, SI\#_2, DO\flat, RE\flat, MI\flat, MI\#_2, FA\#\#, FA\#\#_2, SOL\#\#\#, SOL\#\#_2, LA\#\#\#, SI\#\#\#, q_f\};$
- $Q_0 = q_0;$
- $F = \{q_f\};$
- δ_{t_aum} é como segue:

<ul style="list-style-type: none"> ○ $\delta_{t_aum}(q_0, do) = \{DO\}$ ○ $\delta_{t_aum}(q_0, si\#) = \{DO\}$ ○ $\delta_{t_aum}(q_0, re\flat\flat) = \{DO\}$ ○ $\delta_{t_aum}(DO, mi) = \{MI\}$ ○ $\delta_{t_aum}(DO, fab) = \{MI\}$ ○ $\delta_{t_aum}(DO, re\#\#\#) = \{MI\}$ ○ $\delta_{t_aum}(MI, sol\#) = \{SOL\#\}$ ○ $\delta_{t_aum}(MI, lab) = \{SOL\#\}$ ○ $\delta_{t_aum}(SOL\#, \epsilon) = \{q_f\}$ ○ $\delta_{t_aum}(q_0, do\#) = \{DO\#\}$ ○ $\delta_{t_aum}(q_0, re\flat) = \{DO\#\}$ ○ $\delta_{t_aum}(q_0, si\#\#\#) = \{DO\#\}$ ○ $\delta_{t_aum}(DO\#, mi\#) = \{MI\#\}$ ○ $\delta_{t_aum}(DO\#, fa) = \{MI\#\}$ ○ $\delta_{t_aum}(DO\#, sol\flat\flat) = \{MI\#\}$ ○ $\delta_{t_aum}(MI\#, sol\#\#\#) = \{SOL\#\#\#}\}$ ○ $\delta_{t_aum}(MI\#, la) = \{SOL\#\#\#}\}$ 	<ul style="list-style-type: none"> ○ $\delta_{t_aum}(MI\#, si\flat\flat\flat) = \{SOL\#\#\#}\}$ ○ $\delta_{t_aum}(SOL\#\#\#, \epsilon) = \{q_f\}$ ○ $\delta_{t_aum}(q_0, do\flat) = \{DO\flat\}$ ○ $\delta_{t_aum}(q_0, si) = \{DO\flat\}$ ○ $\delta_{t_aum}(q_0, la\#\#\#) = \{DO\flat\}$ ○ $\delta_{t_aum}(DO\flat, mi\flat) = \{MI\flat\}$ ○ $\delta_{t_aum}(DO\flat, re\#) = \{MI\flat\}$ ○ $\delta_{t_aum}(DO\flat, fab\flat) = \{MI\flat\}$ ○ $\delta_{t_aum}(MI\flat, sol) = \{SOL\}$ ○ $\delta_{t_aum}(MI\flat, fa\#\#\#) = \{SOL\}$ ○ $\delta_{t_aum}(MI\flat, lab\flat) = \{SOL\}$ ○ $\delta_{t_aum}(SOL, \epsilon) = \{q_f\}$ ○ $\delta_{t_aum}(q_0, re) = \{RE\}$ ○ $\delta_{t_aum}(q_0, do\#\#\#) = \{RE\}$ ○ $\delta_{t_aum}(q_0, mi\flat\flat\flat) = \{RE\}$ ○ $\delta_{t_aum}(RE, fa\#) = \{FA\#\}$
---	--

- $\delta_{t_aum}(\text{RE}, mi\#\#) = \{\text{FA}\# \}$
- $\delta_{t_aum}(\text{RE}, sol\#\b) = \{\text{FA}\# \}$
- $\delta_{t_aum}(\text{FA}\#, la\#\b) = \{\text{LA}\# \}$
- $\delta_{t_aum}(\text{FA}\#, si\#\b) = \{\text{LA}\# \}$
- $\delta_{t_aum}(\text{FA}\#, dob\#\b) = \{\text{LA}\# \}$
- $\delta_{t_aum}(\text{LA}\#, \varepsilon) = \{q_f \}$
- $\delta_{t_aum}(q_0, re\#\b) = \{\text{RE}\# \}$
- $\delta_{t_aum}(q_0, mi\#\b) = \{\text{RE}\# \}$
- $\delta_{t_aum}(q_0, fab\#\b) = \{\text{RE}\# \}$
- $\delta_{t_aum}(\text{RE}\#, fa\#\#\b) = \{\text{FA}\#\#\b \}$
- $\delta_{t_aum}(\text{RE}\#, sol) = \{\text{FA}\#\#\b \}$
- $\delta_{t_aum}(\text{RE}\#, lab\#\b) = \{\text{FA}\#\#\b \}$
- $\delta_{t_aum}(\text{FA}\#\#\b, la\#\#\b) = \{\text{LA}\#\#\b \}$
- $\delta_{t_aum}(\text{FA}\#\#\b, si) = \{\text{LA}\#\#\b \}$
- $\delta_{t_aum}(\text{FA}\#\#\b, dob) = \{\text{LA}\#\#\b \}$
- $\delta_{t_aum}(\text{LA}\#\#\b, \varepsilon) = \{q_f \}$
- $\delta_{t_aum}(q_0, reb) = \{\text{REb} \}$
- $\delta_{t_aum}(q_0, do\#\b) = \{\text{REb} \}$
- $\delta_{t_aum}(q_0, si\#\#\b) = \{\text{REb} \}$
- $\delta_{t_aum}(\text{REb}, fa) = \{\text{FA} \}$
- $\delta_{t_aum}(\text{REb}, mi\#\b) = \{\text{FA} \}$
- $\delta_{t_aum}(\text{REb}, sol\#\b) = \{\text{FA} \}$
- $\delta_{t_aum}(\text{FA}, la) = \{\text{LA} \}$
- $\delta_{t_aum}(\text{FA}, sol\#\#\b) = \{\text{LA} \}$
- $\delta_{t_aum}(\text{FA}, si\#\b) = \{\text{LA} \}$
- $\delta_{t_aum}(\text{LA}, \varepsilon) = \{q_f \}$
- $\delta_{t_aum}(q_0, mi) = \{\text{MI}_2 \}$
- $\delta_{t_aum}(q_0, re\#\#\b) = \{\text{MI}_2 \}$
- $\delta_{t_aum}(q_0, fab) = \{\text{MI}_2 \}$
- $\delta_{t_aum}(\text{MI}_2, sol\#\b) = \{\text{SOL}\#\#_2 \}$
- $\delta_{t_aum}(\text{MI}_2, lab\#\b) = \{\text{SOL}\#\#_2 \}$
- $\delta_{t_aum}(\text{SOL}\#\#_2, si\#\b) = \{\text{SI}\#\b \}$
- $\delta_{t_aum}(\text{SOL}\#\#_2, do) = \{\text{SI}\#\b \}$
- $\delta_{t_aum}(\text{SOL}\#\#_2, reb\#\b) = \{\text{SI}\#\b \}$
- $\delta_{t_aum}(\text{SI}\#\b, \varepsilon) = \{q_f \}$
- $\delta_{t_aum}(q_0, mi\#\b) = \{\text{MI}\#\#_2 \}$
- $\delta_{t_aum}(q_0, re\#\b) = \{\text{MI}\#\#_2 \}$
- $\delta_{t_aum}(q_0, fab\#\b) = \{\text{MI}\#\#_2 \}$
- $\delta_{t_aum}(\text{MI}\#\#_2, sol\#\#\b) = \{\text{SOL}\#\#\#_2 \}$
- $\delta_{t_aum}(\text{MI}\#\#_2, la) = \{\text{SOL}\#\#\#_2 \}$
- $\delta_{t_aum}(\text{MI}\#\#_2, si\#\b) = \{\text{SOL}\#\#\#_2 \}$
- $\delta_{t_aum}(\text{SOL}\#\#\#_2, do\#\b) = \{\text{SI}\#\#\b \}$
- $\delta_{t_aum}(\text{SOL}\#\#\#_2, reb) = \{\text{SI}\#\#\b \}$
- $\delta_{t_aum}(\text{SI}, \varepsilon) = \{q_f \}$
- $\delta_{t_aum}(q_0, fa) = \{\text{FA}_2 \}$
- $\delta_{t_aum}(q_0, mi\#\b) = \{\text{FA}_2 \}$
- $\delta_{t_aum}(q_0, sol\#\b) = \{\text{FA}_2 \}$
- $\delta_{t_aum}(\text{FA}_2, la) = \{\text{LA}_2 \}$
- $\delta_{t_aum}(\text{FA}_2, sol\#\#\b) = \{\text{LA}_2 \}$
- $\delta_{t_aum}(\text{FA}_2, si\#\b) = \{\text{LA}_2 \}$
- $\delta_{t_aum}(\text{LA}_2, do\#\b) = \{\text{DO}\#\#_2 \}$
- $\delta_{t_aum}(\text{LA}_2, reb) = \{\text{DO}\#\#_2 \}$
- $\delta_{t_aum}(\text{LA}_2, si\#\#\b) = \{\text{DO}\#\#_2 \}$
- $\delta_{t_aum}(\text{DO}\#\#_2, \varepsilon) = \{q_f \}$
- $\delta_{t_aum}(q_0, fa\#\b) = \{\text{FA}\# \}$
- $\delta_{t_aum}(q_0, mi\#\#\b) = \{\text{FA}\# \}$
- $\delta_{t_aum}(q_0, sol\#\b) = \{\text{FA}\# \}$
- $\delta_{t_aum}(\text{FA}\#\b, la\#\b) = \{\text{LA}\#\#_2 \}$
- $\delta_{t_aum}(\text{FA}\#\b, si\#\b) = \{\text{LA}\#\#_2 \}$
- $\delta_{t_aum}(\text{FA}\#\b, dob\#\b) = \{\text{LA}\#\#_2 \}$
- $\delta_{t_aum}(\text{LA}\#\#_2, do\#\#\b) = \{\text{DO}\#\#\b \}$
- $\delta_{t_aum}(\text{LA}\#\#_2, re) = \{\text{DO}\#\#\b \}$
- $\delta_{t_aum}(\text{LA}\#\#_2, mi\#\b) = \{\text{DO}\#\#\b \}$
- $\delta_{t_aum}(\text{DO}\#\#\b, \varepsilon) = \{q_f \}$
- $\delta_{t_aum}(q_0, fab\#\b) = \{\text{FAb} \}$
- $\delta_{t_aum}(q_0, mi) = \{\text{FAb} \}$
- $\delta_{t_aum}(q_0, re\#\#\b) = \{\text{FAb} \}$
- $\delta_{t_aum}(\text{FAb}, lab\#\b) = \{\text{LA}\#\b \}$
- $\delta_{t_aum}(\text{FAb}, sol\#\b) = \{\text{LA}\#\b \}$
- $\delta_{t_aum}(\text{LA}\#\b, do) = \{\text{DO}_2 \}$
- $\delta_{t_aum}(\text{LA}\#\b, si\#\b) = \{\text{DO}_2 \}$
- $\delta_{t_aum}(\text{LA}\#\b, reb\#\b) = \{\text{DO}_2 \}$
- $\delta_{t_aum}(\text{DO}_2, \varepsilon) = \{q_f \}$
- $\delta_{t_aum}(q_0, sol) = \{\text{SOL}_3 \}$
- $\delta_{t_aum}(q_0, fa\#\#\b) = \{\text{SOL}_3 \}$

- $\delta_{t_aum}(q_0, lab) = \{SOL_3\}$
- $\delta_{t_aum}(SOL_3, si) = \{SI_2\}$
- $\delta_{t_aum}(SOL_3, do) = \{SI_2\}$
- $\delta_{t_aum}(SOL_3, la##) = \{SI_2\}$
- $\delta_{t_aum}(SI_2, re##) = \{RE\#_2\}$
- $\delta_{t_aum}(SI_2, mi) = \{RE\#_2\}$
- $\delta_{t_aum}(SI_2, fab) = \{RE\#_2\}$
- $\delta_{t_aum}(RE\#_2, \varepsilon) = \{q_f\}$
- $\delta_{t_aum}(q_0, sol##) = \{SOL\#_3\}$
- $\delta_{t_aum}(q_0, lab) = \{SOL\#_3\}$
- $\delta_{t_aum}(SOL\#_3, si##) = \{SI\#_2\}$
- $\delta_{t_aum}(SOL\#_3, do) = \{SI\#_2\}$
- $\delta_{t_aum}(SOL\#_3, reb) = \{SI\#_2\}$
- $\delta_{t_aum}(SI\#_2, re##) = \{RE\#\#\}$
- $\delta_{t_aum}(SI\#_2, mi) = \{RE\#\#\}$
- $\delta_{t_aum}(SI\#_2, fab) = \{RE\#\#\}$
- $\delta_{t_aum}(RE\#\#\varepsilon) = \{q_f\}$
- $\delta_{t_aum}(q_0, solb) = \{SOLb\}$
- $\delta_{t_aum}(q_0, fab) = \{SOLb\}$
- $\delta_{t_aum}(q_0, mi##) = \{SOLb\}$
- $\delta_{t_aum}(SOLb, sib) = \{SIB\}$
- $\delta_{t_aum}(SOLb, dob) = \{SIB\}$
- $\delta_{t_aum}(SOLb, la##) = \{SIB\}$
- $\delta_{t_aum}(SIB, re) = \{RE_2\}$
- $\delta_{t_aum}(SIB, do##) = \{RE_2\}$
- $\delta_{t_aum}(SIB, mi) = \{RE_2\}$
- $\delta_{t_aum}(RE_2, \varepsilon) = \{q_f\}$
- $\delta_{t_aum}(q_0, la) = \{LA_3\}$
- $\delta_{t_aum}(q_0, sol##) = \{LA_3\}$
- $\delta_{t_aum}(q_0, sib) = \{LA_3\}$
- $\delta_{t_aum}(LA_3, do##) = \{DO\#_3\}$
- $\delta_{t_aum}(LA_3, si##) = \{DO\#_3\}$
- $\delta_{t_aum}(LA_3, reb) = \{DO\#_3\}$
- $\delta_{t_aum}(DO\#_3, mi) = \{MI\#_3\}$
- $\delta_{t_aum}(DO\#_3, fa) = \{MI\#_3\}$
- $\delta_{t_aum}(DO\#_3, solb) = \{MI\#_3\}$
- $\delta_{t_aum}(MI\#_3, \varepsilon) = \{q_f\}$
- $\delta_{t_aum}(q_0, sib) = \{LA\#_3\}$
- $\delta_{t_aum}(q_0, la##) = \{LA\#_3\}$
- $\delta_{t_aum}(q_0, dob) = \{LA\#_3\}$
- $\delta_{t_aum}(LA\#_3, do##) = \{DO\#\#\#_2\}$
- $\delta_{t_aum}(LA\#_3, re) = \{DO\#\#\#_2\}$
- $\delta_{t_aum}(LA\#_3, mi) = \{DO\#\#\#_2\}$
- $\delta_{t_aum}(DO\#\#\#_2, mi##) = \{MI\#\#\#_2\}$
- $\delta_{t_aum}(DO\#\#\#_2, fa) = \{MI\#\#\#_2\}$
- $\delta_{t_aum}(DO\#\#\#_2, sol) = \{MI\#\#\#_2\}$
- $\delta_{t_aum}(MI\#\#\#_2, \varepsilon) = \{q_f\}$
- $\delta_{t_aum}(q_0, lab) = \{LAB_2\}$
- $\delta_{t_aum}(q_0, sol##) = \{LAB_2\}$
- $\delta_{t_aum}(LA_2, do) = \{DO_3\}$
- $\delta_{t_aum}(LA_2, si##) = \{DO_3\}$
- $\delta_{t_aum}(LA_2, reb) = \{DO_3\}$
- $\delta_{t_aum}(DO_3, mi) = \{MI_3\}$
- $\delta_{t_aum}(DO_3, re##) = \{MI_3\}$
- $\delta_{t_aum}(DO_3, fab) = \{MI_3\}$
- $\delta_{t_aum}(MI_3, \varepsilon) = \{q_f\}$
- $\delta_{t_aum}(q_0, si) = \{SI_3\}$
- $\delta_{t_aum}(q_0, dob) = \{SI_3\}$
- $\delta_{t_aum}(q_0, la##) = \{SI_3\}$
- $\delta_{t_aum}(SI_3, re##) = \{RE\#_3\}$
- $\delta_{t_aum}(SI_3, mi) = \{RE\#_3\}$
- $\delta_{t_aum}(SI_3, fab) = \{RE\#_3\}$
- $\delta_{t_aum}(RE\#_3, fa##) = \{FA\#\#\#_2\}$
- $\delta_{t_aum}(RE\#_3, sol) = \{FA\#\#\#_2\}$
- $\delta_{t_aum}(RE\#_3, lab) = \{FA\#\#\#_2\}$
- $\delta_{t_aum}(FA\#\#\#_2, \varepsilon) = \{q_f\}$
- $\delta_{t_aum}(q_0, sib) = \{SIB_2\}$
- $\delta_{t_aum}(q_0, la##) = \{SIB_2\}$
- $\delta_{t_aum}(q_0, dob) = \{SIB_2\}$
- $\delta_{t_aum}(SI_2, re) = \{RE_3\}$
- $\delta_{t_aum}(SI_2, do##) = \{RE_3\}$
- $\delta_{t_aum}(SI_2, mi) = \{RE_3\}$
- $\delta_{t_aum}(RE_3, fa) = \{FA\#_3\}$
- $\delta_{t_aum}(RE_3, mi##) = \{FA\#_3\}$
- $\delta_{t_aum}(RE_3, sol) = \{FA\#_3\}$
- $\delta_{t_aum}(FA\#_3, \varepsilon) = \{q_f\}$

d. Autômato que Reconhece Tríade Diminuta

O AFN ε que reconhece Tríade Diminuta, denotamos como T_DIM, é definido formalmente como segue: $T_{AUM} = \langle \Sigma_{t_dim}, Q_{t_dim}, F, \delta_{t_dim}, Q_0 \rangle$, em que:

- $\Sigma_{t_dim} = \{do, re, mi, fa, sol, la, si, do\#, re\#, mi\#, fa\#, sol\#, la\#, si\#, dob, reb, mi\#b, fab, sol\#b, lab, si\#b, dob\#b, reb\#b, mi\#bb, sol\#bb, lab\#b, si\#bb\};$
- $Q_{t_dim} = \{q_0, DO, DO_2, DO_3, RE, RE_2, RE_3, MI, MI_2, MI_3, FA, FA_2, FA_3, SOL, SOL_2, SOL_3, LA, LA_2, LA_3, SI, SI_2, SI_3, DO\#, DO\#_2, RE\#, RE\#_2, MI\#, MI\#_2, FA\#, FA\#_2, FA\#_3, SOL\#, SOL\#_2, LA\#, SI\#, DO\#b, DO\#b_2, DO\#b_3, RE\#b, RE\#b_2, RE\#b_3, MI\#b, MI\#b_2, MI\#b_3, FA\#b, FA\#b_2, FA\#b_3, SOL\#b, SOL\#b_2, LAB, LAB_2, LAB_3, SIB, SIB_2, SIB_3, DO\#bb, RE\#bb, MI\#bb, MI\#bb_2, SOL\#bb, SOL\#bb_2, LAB\#b, LAB\#bb_2, SIB\#b, SIB\#bb_2, q_f\};$
- $Q_0 = q_0;$
- $F = \{q_f\};$
- δ_{t_dim} é como segue:

<ul style="list-style-type: none"> ○ $\delta_{t_dim}(q_0, do) = \{DO\}$ ○ $\delta_{t_dim}(q_0, si\#) = \{DO\}$ ○ $\delta_{t_dim}(q_0, reb\#b) = \{DO\}$ ○ $\delta_{t_dim}(DO, mi\#b) = \{MI\#b\}$ ○ $\delta_{t_dim}(DO, re\#\#) = \{MI\#b\}$ ○ $\delta_{t_dim}(DO, fab\#b) = \{MI\#b\}$ ○ $\delta_{t_dim}(MI\#b, sol\#b) = \{SOL\#b\}$ ○ $\delta_{t_dim}(MI\#b, fa\#\#) = \{SOL\#b\}$ ○ $\delta_{t_dim}(MI\#b, mi\#\#\#) = \{SOL\#b\}$ ○ $\delta_{t_dim}(SOL\#b, \varepsilon) = \{q_f\}$ ○ $\delta_{t_dim}(q_0, do\#\#) = \{DO\#\#$ ○ $\delta_{t_dim}(q_0, reb) = \{DO\#\#$ ○ $\delta_{t_dim}(q_0, si\#\#\#) = \{DO\#\#$ ○ $\delta_{t_dim}(DO, mi) = \{MI\}$ ○ $\delta_{t_dim}(DO, re\#\#\#) = \{MI\}$ ○ $\delta_{t_dim}(DO, fab) = \{MI\}$ ○ $\delta_{t_dim}(MI, sol) = \{SOL\}$ ○ $\delta_{t_dim}(MI, fa\#\#\#) = \{SOL\}$ ○ $\delta_{t_dim}(MI, lab\#b) = \{SOL\}$ ○ $\delta_{t_dim}(SOL, \varepsilon) = \{q_f\}$ ○ $\delta_{t_dim}(q_0, dob) = \{DO\#b\}$ ○ $\delta_{t_dim}(q_0, si) = \{DO\#b\}$ ○ $\delta_{t_dim}(q_0, la\#\#\#) = \{DO\#b\}$ ○ $\delta_{t_dim}(DO\#b, mi\#bb) = \{MI\#bb\}$ 	<ul style="list-style-type: none"> ○ $\delta_{t_dim}(DO\#b, re) = \{MI\#bb\}$ ○ $\delta_{t_dim}(DO\#b, do\#\#\#) = \{MI\#bb\}$ ○ $\delta_{t_dim}(MI\#bb, sol\#bb) = \{SOL\#bb\}$ ○ $\delta_{t_dim}(MI\#bb, fa) = \{SOL\#bb\}$ ○ $\delta_{t_dim}(MI\#bb, mi\#\#) = \{SOL\#bb\}$ ○ $\delta_{t_dim}(SOL\#bb, \varepsilon) = \{q_f\}$ ○ $\delta_{t_dim}(q_0, re) = \{RE\}$ ○ $\delta_{t_dim}(q_0, do\#\#\#) = \{RE\}$ ○ $\delta_{t_dim}(q_0, mi\#bb) = \{RE\}$ ○ $\delta_{t_dim}(RE, fa) = \{FA\}$ ○ $\delta_{t_dim}(RE, mi\#\#) = \{FA\}$ ○ $\delta_{t_dim}(RE, sol\#bb) = \{FA\}$ ○ $\delta_{t_dim}(FA, lab) = \{LA\#b\}$ ○ $\delta_{t_dim}(FA, sol\#) = \{LA\#b\}$ ○ $\delta_{t_dim}(LA\#b, \varepsilon) = \{q_f\}$ ○ $\delta_{t_dim}(q_0, re\#\#) = \{RE\#\#$ ○ $\delta_{t_dim}(q_0, mi\#b) = \{RE\#\#$ ○ $\delta_{t_dim}(q_0, fab\#b) = \{RE\#\#$ ○ $\delta_{t_dim}(RE\#\#, fa\#\#) = \{FA\#\#$ ○ $\delta_{t_dim}(RE\#\#, mi\#\#\#) = \{FA\#\#$ ○ $\delta_{t_dim}(RE\#\#, sol\#b) = \{FA\#\#$ ○ $\delta_{t_dim}(FA\#\#, la) = \{LA\}$ ○ $\delta_{t_dim}(FA\#\#, sol\#\#\#) = \{LA\}$ ○ $\delta_{t_dim}(FA\#\#, si\#bb) = \{LA\}$
--	---

- $\delta_{t_dim}(LA, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, reb) = \{REb\}$
- $\delta_{t_dim}(q_0, do\#) = \{REb\}$
- $\delta_{t_dim}(q_0, si\#\#\#) = \{REb\}$
- $\delta_{t_dim}(LABb, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, mi) = \{MI_2\}$
- $\delta_{t_dim}(q_0, re\#\#\#) = \{MI_2\}$
- $\delta_{t_dim}(q_0, fab) = \{MI_2\}$
- $\delta_{t_dim}(REb, fab) = \{FAb\}$
- $\delta_{t_dim}(REb, mi) = \{FAb\}$
- $\delta_{t_dim}(REb, re\#\#\#) = \{FAb\}$
- $\delta_{t_dim}(FAb, labb) = \{LABb\}$
- $\delta_{t_dim}(FAb, sol) = \{LABb\}$
- $\delta_{t_dim}(FAb, fa\#\#\#) = \{LABb\}$
- $\delta_{t_dim}(MI_2, sol) = \{SOL_2\}$
- $\delta_{t_dim}(MI_2, fa\#\#\#) = \{SOL_2\}$
- $\delta_{t_dim}(MI_2, labb) = \{SOL_2\}$
- $\delta_{t_dim}(SOL_2, sib) = \{SIB\}$
- $\delta_{t_dim}(SOL_2, la\#) = \{SIB\}$
- $\delta_{t_dim}(SOL_2, dob\#) = \{SIB\}$
- $\delta_{t_dim}(SIB, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, mi\#) = \{MI\#\}$
- $\delta_{t_dim}(q_0, fa) = \{MI\#\}$
- $\delta_{t_dim}(q_0, solbb) = \{MI\#\}$
- $\delta_{t_dim}(MI\#, sol\#) = \{SOL\#\}$
- $\delta_{t_dim}(MI\#, lab) = \{SOL\#\}$
- $\delta_{t_dim}(SOL, si) = \{SI\}$
- $\delta_{t_dim}(SOL, la\#\#\#) = \{SI\}$
- $\delta_{t_dim}(SOL, dob) = \{SI\}$
- $\delta_{t_dim}(SI, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, mi\#b) = \{MIB_2\}$
- $\delta_{t_dim}(q_0, re\#) = \{MIB_2\}$
- $\delta_{t_dim}(q_0, fabbb) = \{MIB_2\}$
- $\delta_{t_dim}(MIB_2, solb) = \{SOLb_2\}$
- $\delta_{t_dim}(MIB_2, fa\#) = \{SOLb_2\}$
- $\delta_{t_dim}(MIB_2, mi\#\#\#) = \{SOLb_2\}$
- $\delta_{t_dim}(SOLb_2, sibb) = \{SIBb\}$
- $\delta_{t_dim}(SOLb_2, la) = \{SIBb\}$
- $\delta_{t_dim}(SOLb_2, sol\#\#\#) = \{SIBb\}$
- $\delta_{t_dim}(SIBb, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, fa) = \{FA_2\}$
- $\delta_{t_dim}(q_0, mi\#) = \{FA_2\}$
- $\delta_{t_dim}(q_0, solbb) = \{FA_2\}$
- $\delta_{t_dim}(FA_2, lab) = \{LAB_2\}$
- $\delta_{t_dim}(FA_2, sol\#) = \{LAB_2\}$
- $\delta_{t_dim}(LAB_2, dob) = \{DOB_2\}$
- $\delta_{t_dim}(LAB_2, si) = \{DOB_2\}$
- $\delta_{t_dim}(LAB_2, la\#\#\#) = \{DOB_2\}$
- $\delta_{t_dim}(DOB_2, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, fa\#) = \{FA\#_2\}$
- $\delta_{t_dim}(q_0, solb) = \{FA\#_2\}$
- $\delta_{t_dim}(q_0, mi\#\#\#) = \{FA\#_2\}$
- $\delta_{t_dim}(FA\#_2, la) = \{LA_2\}$
- $\delta_{t_dim}(FA\#_2, sol\#\#\#) = \{LA_2\}$
- $\delta_{t_dim}(FA\#_2, sibb) = \{LA_2\}$
- $\delta_{t_dim}(LA_2, do) = \{DO_2\}$
- $\delta_{t_dim}(LA_2, si\#) = \{DO_2\}$
- $\delta_{t_dim}(LA_2, reb\#) = \{DO_2\}$
- $\delta_{t_dim}(DO_2, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, fab) = \{FAb_2\}$
- $\delta_{t_dim}(q_0, mi) = \{FAB_2\}$
- $\delta_{t_dim}(q_0, re\#\#\#) = \{FAB_2\}$
- $\delta_{t_dim}(FAB_2, labb) = \{LABb_2\}$
- $\delta_{t_dim}(FAB_2, sol) = \{LABb_2\}$
- $\delta_{t_dim}(FAB_2, fa\#\#\#) = \{LABb_2\}$
- $\delta_{t_dim}(LABb_2, dob\#) = \{DOBb\}$
- $\delta_{t_dim}(LABb_2, sib) = \{DOBb\}$
- $\delta_{t_dim}(LABb_2, la\#) = \{DOBb\}$
- $\delta_{t_dim}(DOBb, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, sol) = \{SOL_3\}$
- $\delta_{t_dim}(q_0, fa\#\#\#) = \{SOL_3\}$
- $\delta_{t_dim}(q_0, labb) = \{SOL_3\}$
- $\delta_{t_dim}(SOL_3, sib) = \{SIB_2\}$
- $\delta_{t_dim}(SOL_3, la\#) = \{SIB_2\}$
- $\delta_{t_dim}(SOL_3, dob\#) = \{SIB_2\}$
- $\delta_{t_dim}(SIB_2, reb) = \{REb_2\}$
- $\delta_{t_dim}(SIB_2, do\#) = \{REb_2\}$
- $\delta_{t_dim}(SIB_2, si\#\#\#) = \{REb_2\}$
- $\delta_{t_dim}(REb_2, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, sol\#) = \{SOL\#_3\}$
- $\delta_{t_dim}(q_0, lab) = \{SOL\#_3\}$
- $\delta_{t_dim}(SOL_3, si) = \{SI_2\}$
- $\delta_{t_dim}(SOL_3, la\#\#\#) = \{SI_2\}$
- $\delta_{t_dim}(SOL_3, dob) = \{SI_2\}$

- $\delta_{t_dim}(SI_2, re) = \{RE_2\}$
- $\delta_{t_dim}(SI_2, do##) = \{RE_2\}$
- $\delta_{t_dim}(SI_2, mibb) = \{RE_2\}$
- $\delta_{t_dim}(RE_2, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, solb) = \{SOLb_3\}$
- $\delta_{t_dim}(q_0, fa##) = \{SOLb_3\}$
- $\delta_{t_dim}(q_0, mi##) = \{SOLb_3\}$
- $\delta_{t_dim}(SOLb_3, sibb) = \{SIBb_2\}$
- $\delta_{t_dim}(SOLb_3, la) = \{SIBb_2\}$
- $\delta_{t_dim}(SOLb_3, sol##) = \{SIBb_2\}$
- $\delta_{t_dim}(SIBb_2, rebb) = \{REbb\}$
- $\delta_{t_dim}(SIBb_2, do) = \{REbb\}$
- $\delta_{t_dim}(SIBb_2, si##) = \{REbb\}$
- $\delta_{t_dim}(REbb, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, la) = \{LA_3\}$
- $\delta_{t_dim}(q_0, sol##) = \{LA_3\}$
- $\delta_{t_dim}(q_0, sibb) = \{LA_3\}$
- $\delta_{t_dim}(LA_3, do) = \{DO_3\}$
- $\delta_{t_dim}(LA_3, si##) = \{DO_3\}$
- $\delta_{t_dim}(LA_3, rebb) = \{DO_3\}$
- $\delta_{t_dim}(DO_3, mib) = \{Mib_3\}$
- $\delta_{t_dim}(DO_3, re##) = \{Mib_3\}$
- $\delta_{t_dim}(DO_3, fabb) = \{Mib_3\}$
- $\delta_{t_dim}(Mib_3, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, la##) = \{LA#\}$
- $\delta_{t_dim}(q_0, sib) = \{LA#\}$
- $\delta_{t_dim}(q_0, dob) = \{LA#\}$
- $\delta_{t_dim}(LA#, do##) = \{DO#_2\}$
- $\delta_{t_dim}(LA#, reb) = \{DO#_2\}$
- $\delta_{t_dim}(LA#, si##) = \{DO#_2\}$
- $\delta_{t_dim}(DO#_2, mi) = \{MI_3\}$
- $\delta_{t_dim}(DO#_2, fab) = \{MI_3\}$
- $\delta_{t_dim}(DO#_2, re##) = \{MI_3\}$
- $\delta_{t_dim}(MI_3, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, lab) = \{LAB_3\}$
- $\delta_{t_dim}(q_0, sol##) = \{LAB_3\}$
- $\delta_{t_dim}(LAB_3, dob) = \{DOB_3\}$
- $\delta_{t_dim}(LAB_3, si) = \{DOB_3\}$
- $\delta_{t_dim}(LAB_3, la##) = \{DOB_3\}$
- $\delta_{t_dim}(DOB_3, mibb) = \{Mib_2\}$
- $\delta_{t_dim}(DOB_3, re) = \{Mib_2\}$
- $\delta_{t_dim}(DOB_3, do##) = \{Mib_2\}$
- $\delta_{t_dim}(Mib_2, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, si) = \{SI_3\}$
- $\delta_{t_dim}(q_0, dob) = \{SI_3\}$
- $\delta_{t_dim}(q_0, la##) = \{SI_3\}$
- $\delta_{t_dim}(SI_3, re) = \{RE_3\}$
- $\delta_{t_dim}(SI_3, do##) = \{RE_3\}$
- $\delta_{t_dim}(SI_3, mibb) = \{RE_3\}$
- $\delta_{t_dim}(RE_3, fa) = \{FA_3\}$
- $\delta_{t_dim}(RE_3, mi##) = \{FA_3\}$
- $\delta_{t_dim}(RE_3, solb) = \{FA_3\}$
- $\delta_{t_dim}(FA_3, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, si##) = \{SI#\}$
- $\delta_{t_dim}(q_0, do) = \{SI#\}$
- $\delta_{t_dim}(q_0, reb) = \{SI#\}$
- $\delta_{t_dim}(SI#, re##) = \{RE#_2\}$
- $\delta_{t_dim}(SI#, mib) = \{RE#_2\}$
- $\delta_{t_dim}(SI#, fabb) = \{RE#_2\}$
- $\delta_{t_dim}(RE#_2, fa##) = \{FA#_3\}$
- $\delta_{t_dim}(RE#_2, solb) = \{FA#_3\}$
- $\delta_{t_dim}(RE#_2, mi##) = \{FA#_3\}$
- $\delta_{t_dim}(FA#_3, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, sib) = \{SIB_3\}$
- $\delta_{t_dim}(q_0, dob) = \{SIB_3\}$
- $\delta_{t_dim}(q_0, la##) = \{SIB_3\}$
- $\delta_{t_dim}(SIB_3, reb) = \{REb_3\}$
- $\delta_{t_dim}(SIB_3, do##) = \{REb_3\}$
- $\delta_{t_dim}(SIB_3, si##) = \{REb_3\}$
- $\delta_{t_dim}(REb_3, fab) = \{FAb_3\}$
- $\delta_{t_dim}(REb_3, mi) = \{FAb_3\}$
- $\delta_{t_dim}(REb_3, re##) = \{FAb_3\}$
- $\delta_{t_dim}(FAb_3, \varepsilon) = \{q_f\}$
- $\delta_{t_dim}(q_0, sib) = \{SIB_3\}$
- $\delta_{t_dim}(q_0, la##) = \{SIB_3\}$
- $\delta_{t_dim}(q_0, dob) = \{SIB_3\}$
- $\delta_{t_dim}(SIB_3, reb) = \{REb_3\}$
- $\delta_{t_dim}(SIB_3, do##) = \{REb_3\}$
- $\delta_{t_dim}(SIB_3, si##) = \{REb_3\}$
- $\delta_{t_dim}(REb_3, fab) = \{FAb_3\}$
- $\delta_{t_dim}(REb_3, mi) = \{FAb_3\}$
- $\delta_{t_dim}(REb_3, re##) = \{FAb_3\}$
- $\delta_{t_dim}(FAb_3, \varepsilon) = \{q_f\}$

e. Autômato TRIADE

O AFN ε que reconhece uma Tríade, denotamos como TRIADE, é definido formalmente a partir dos Autômatos anteriormente definidos, a saber: T_MA (reconhece Tríade Maior), T_ME (reconhece Tríade Menor), T_DIM (reconhece Tríade Diminuta) e T_AUM (reconhece Tríade Aumentada). Definimos o Autômato TRIADE como segue: TRIADE = $\langle \sum_{tri}, Q_{tri}, F_{tri}, \delta_{tri}, Q_0 \rangle$, em que:

- $\sum_{tri} = \sum_{t_ma} U \sum_{t_me} U \sum_{t_dim} U \sum_{t_aum}$;
- $Q_{tri} = \{q_0, q_f\} U Q_{t_ma} U Q_{t_me} U Q_{t_dim} U Q_{t_aum}$, para que não se perda nenhum estado ao fazer o processo de união, fizemos as seguintes mudanças:
 - Para todo $p \in Q_{t_ma}$, renomeamos p para p_{t_ma} ;
 - Ex.: O estado DO₁ de Q_{t_ma} , se torna DO_{1t_ma};
 - Para todo $p \in Q_{t_me}$, renomeamos p para p_{t_me} ;
 - Ex.: O estado DO₁ de Q_{t_me} , se torna DO_{1t_me};
 - Para todo $p \in Q_{t_dim}$, renomeamos p para p_{t_dim} ;
 - Ex.: O estado DO₁ de Q_{t_dim} , se torna DO_{1t_dim};
 - Para todo $p \in Q_{t_aum}$, renomeamos p para p_{t_aum} ;
 - Ex.: O estado DO₁ de Q_{t_aum} , se torna DO_{1t_aum};
- $F_{tri} = \{q_f\}$;
- $Q_0 = q_0$;
- δ_{tri} possui todas as transições de δ_{t_ma} , δ_{t_me} , δ_{t_dim} , δ_{t_aum} , mais as seguintes:
 - $\delta_{tri}(q_0, \varepsilon) = \{q_{0t_ma}, q_{0t_me}, q_{0t_dim}, q_{0t_aum}\}$
 - $\delta_{tri}(q_{ft_ma}, \varepsilon) = \{q_f\}$
 - $\delta_{tri}(q_{ft_me}, \varepsilon) = \{q_f\}$
 - $\delta_{tri}(q_{ft_dim}, \varepsilon) = \{q_f\}$
 - $\delta_{tri}(q_{ft_aum}, \varepsilon) = \{q_f\}$

7.2 APÊNDICE B – Reconhecedor de Tétrade

a. Autômato que Reconhece uma Tétrade formado a partir do AFN ε T_MA

O AFN ε que reconhece Tétrade maior ou menor formado a partir do que reconhece uma Tríade Maior, denotamos como TE_MA, é definido formalmente como segue:

TE_MA = $\langle \sum_{te_ma}, Q_{te_ma}, F, \delta_{te_ma}, Q_0 \rangle$, em que:

- $\sum_{te_ma} = \sum_{t_ma}$;
- $Q_{te_ma} = \{SI(\#), SIb(b), SI(b), DO\#(\#), DO(b), DO(\#), RE\#(\#), RE(b), RE(\#), MI(\#), MIb(b), MI(b), FA\#(\#), FA(b), FA(\#), SOL\#(\#), SOL(b), SOL(\#), LA\#(\#), LA(b), LA(\#)\} \cup Q_{t_ma}$;
- $Q_0 = q_0$;
- $F = \{q_f\}$;
- δ_{te_ma} tem todas as transições da função δ_{t_ma} , exceto as que levam ao estado final, mais as seguintes:

- | | |
|--|---|
| <ul style="list-style-type: none"> ○ $\delta_{te_ma}(SOL\#, si) = \{SI(\#)\}$ ○ $\delta_{te_ma}(SOL\#, la\#\#) = \{SI(\#)\}$ ○ $\delta_{te_ma}(SOL\#, dob) = \{SI(\#)\}$ ○ $\delta_{te_ma}(SOL\#, si\#) = \{SI(\#)\}$ ○ $\delta_{te_ma}(SOL\#, do) = \{SI(\#)\}$ ○ $\delta_{te_ma}(SOL\#, reb\#b) = \{SI(\#)\}$ ○ $\delta_{te_ma}(SI(\#), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(SOLb, si\#b) = \{SIb(b)\}$ ○ $\delta_{te_ma}(SOLb, dob\#b) = \{SIb(b)\}$ ○ $\delta_{te_ma}(SOLb, la\#) = \{SIb(b)\}$ ○ $\delta_{te_ma}(SOLb, si\#b\#) = \{SIb(b)\}$ ○ $\delta_{te_ma}(SOLb, la) = \{SIb(b)\}$ ○ $\delta_{te_ma}(SOLb, sol\#\#) = \{SIb(b)\}$ ○ $\delta_{te_ma}(SIb(b), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(SOL, si) = \{SI(b)\}$ ○ $\delta_{te_ma}(SOL, dob) = \{SI(b)\}$ ○ $\delta_{te_ma}(SOL, la\#\#) = \{SI(b)\}$ ○ $\delta_{te_ma}(SOL, si\#b) = \{SI(b)\}$ ○ $\delta_{te_ma}(SOL, la\#) = \{SI(b)\}$ ○ $\delta_{te_ma}(SOL, dob\#b) = \{SI(b)\}$ ○ $\delta_{te_ma}(SI(b), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(LA\#, do\#) = \{DO\#(\#)\}$ ○ $\delta_{te_ma}(LA\#, reb) = \{DO\#(\#)\}$ ○ $\delta_{te_ma}(LA\#, si\#\#) = \{DO\#(\#)\}$ ○ $\delta_{te_ma}(LA\#, mi\#b) = \{DO\#(\#)\}$ ○ $\delta_{te_ma}(DO\#(\#), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(LA\#b, do) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, reb\#b) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, si\#) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, dob) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, si) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, la\#\#) = \{DO(b)\}$ ○ $\delta_{te_ma}(DO\#(\#), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(LA\#, do) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, reb\#) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, si\#) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, do\#) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, reb) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, si\#\#) = \{DO(\#)\}$ ○ $\delta_{te_ma}(DO\#(\#), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(SI\#, re\#) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, mi\#b) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, fab\#b) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, re\#\#) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, mi) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, fab) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(RE\#(\#), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(SIb, re) = \{RE(b)\}$ | <ul style="list-style-type: none"> ○ $\delta_{te_ma}(LA\#, re) = \{DO\#(\#)\}$ ○ $\delta_{te_ma}(LA\#, mi\#b) = \{DO\#(\#)\}$ ○ $\delta_{te_ma}(DO\#(\#), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(LA\#b, do) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, reb\#b) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, si\#) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, dob) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, si) = \{DO(b)\}$ ○ $\delta_{te_ma}(LA\#b, la\#\#) = \{DO(b)\}$ ○ $\delta_{te_ma}(DO\#(\#), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(LA\#, do) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, reb\#) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, si\#) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, do\#) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, reb) = \{DO(\#)\}$ ○ $\delta_{te_ma}(LA\#, si\#\#) = \{DO(\#)\}$ ○ $\delta_{te_ma}(DO\#(\#), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(SI\#, re\#) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, mi\#b) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, fab\#b) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, re\#\#) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, mi) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(SI\#, fab) = \{RE\#(\#)\}$ ○ $\delta_{te_ma}(RE\#(\#), \varepsilon) = \{q_f\}$ ○ $\delta_{te_ma}(SIb, re) = \{RE(b)\}$ |
|--|---|

- $\delta_{te_ma} (SIb, do\#\#) = \{RE(b)\}$
- $\delta_{te_ma} (SIb, mi\#\#\bar{b}) = \{RE(b)\}$
- $\delta_{te_ma} (SIb, reb) = \{RE(b)\}$
- $\delta_{te_ma} (SIb, do\#) = \{RE(b)\}$
- $\delta_{te_ma} (SIb, si\#\#) = \{RE(b)\}$
- $\delta_{te_ma} (RE(b), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (SI, re) = \{RE(\#)\}$
- $\delta_{te_ma} (SI, do\#\#) = \{RE(\#)\}$
- $\delta_{te_ma} (SI, mi\#\#\bar{b}) = \{RE(\#)\}$
- $\delta_{te_ma} (SI, re\#) = \{RE(\#)\}$
- $\delta_{te_ma} (SI, mi\bar{b}) = \{RE(\#)\}$
- $\delta_{te_ma} (SI, fab\#\bar{b}) = \{RE(\#)\}$
- $\delta_{te_ma} (RE(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (DO\#_2, mi) = \{MI(\#)\}$
- $\delta_{te_ma} (DO\#_2, fab) = \{MI(\#)\}$
- $\delta_{te_ma} (DO\#_2, re\#\#) = \{MI(\#)\}$
- $\delta_{te_ma} (DO\#_2, mi\#) = \{MI(\#)\}$
- $\delta_{te_ma} (DO\#_2, fa) = \{MI(\#)\}$
- $\delta_{te_ma} (DO\#_2, sol\#\bar{b}) = \{MI(\#)\}$
- $\delta_{te_ma} (MI(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (DO\bar{b}_2, mi\bar{b}) = \{MI\bar{b}(b)\}$
- $\delta_{te_ma} (DO\bar{b}_2, re\#) = \{MI\bar{b}(b)\}$
- $\delta_{te_ma} (DO\bar{b}_2, fab\#\bar{b}) = \{MI\bar{b}(b)\}$
- $\delta_{te_ma} (DO\bar{b}_2, mi\#\bar{b}) = \{MI\bar{b}(b)\}$
- $\delta_{te_ma} (DO\bar{b}_2, re) = \{MI\bar{b}(b)\}$
- $\delta_{te_ma} (DO\bar{b}_2, do\#\#) = \{MI\bar{b}(b)\}$
- $\delta_{te_ma} (MI\bar{b}(b), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (DO_2, mi) = \{MI(b)\}$
- $\delta_{te_ma} (DO_2, fab) = \{MI(b)\}$
- $\delta_{te_ma} (DO_2, re\#\#) = \{MI(b)\}$
- $\delta_{te_ma} (DO_2, mi\bar{b}) = \{MI(b)\}$
- $\delta_{te_ma} (DO_2, re\#) = \{MI(b)\}$
- $\delta_{te_ma} (DO_2, fab\#\bar{b}) = \{MI(b)\}$
- $\delta_{te_ma} (MI(b), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (RE\#_2, fa\#) = \{FA\#(\#)\}$
- $\delta_{te_ma} (RE\#_2, mi\#\#) = \{FA\#(\#)\}$
- $\delta_{te_ma} (RE\#_2, sol\#) = \{FA\#(\#)\}$
- $\delta_{te_ma} (RE\#_2, fa\#\#\bar{b}) = \{FA\#(\#)\}$
- $\delta_{te_ma} (RE\#_2, sol\#\bar{b}) = \{FA\#(\#)\}$
- $\delta_{te_ma} (RE\#_2, la\#\#\bar{b}) = \{FA\#(\#)\}$
- $\delta_{te_ma} (FA\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (REb_2, fa) = \{FA(b)\}$
- $\delta_{te_ma} (REb_2, mi\#) = \{FA(b)\}$
- $\delta_{te_ma} (REb_2, sol\#\bar{b}) = \{FA(b)\}$
- $\delta_{te_ma} (REb_2, fab\#\bar{b}) = \{FA(b)\}$
- $\delta_{te_ma} (REb_2, mi\#\#\bar{b}) = \{FA(b)\}$
- $\delta_{te_ma} (REb_2, re\#\#) = \{FA(b)\}$
- $\delta_{te_ma} (FA(b), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (RE_2, fa) = \{FA(\#)\}$
- $\delta_{te_ma} (RE_2, mi\#) = \{FA(\#)\}$
- $\delta_{te_ma} (RE_2, sol\#\bar{b}) = \{FA(\#)\}$
- $\delta_{te_ma} (RE_2, fa\#) = \{FA(\#)\}$
- $\delta_{te_ma} (RE_2, sol\#) = \{FA(\#)\}$
- $\delta_{te_ma} (RE_2, mi\#\#) = \{FA(\#)\}$
- $\delta_{te_ma} (FA(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (MI\#_3, sol\#) = \{SOL\#(\#)\}$
- $\delta_{te_ma} (MI\#_3, lab) = \{SOL\#(\#)\}$
- $\delta_{te_ma} (MI\#_3, sol\#\#) = \{SOL\#(\#)\}$
- $\delta_{te_ma} (MI\#_3, la) = \{SOL\#(\#)\}$
- $\delta_{te_ma} (MI\#_3, si\#\bar{b}) = \{SOL\#(\#)\}$
- $\delta_{te_ma} (SOL\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (MI\bar{b}_3, sol) = \{SOL(b)\}$
- $\delta_{te_ma} (MI\bar{b}_3, fa\#\#) = \{SOL(b)\}$
- $\delta_{te_ma} (MI\bar{b}_3, lab\#\bar{b}) = \{SOL(b)\}$
- $\delta_{te_ma} (MI\bar{b}_3, sol\#\bar{b}) = \{SOL(b)\}$
- $\delta_{te_ma} (MI\bar{b}_3, fa\#) = \{SOL(b)\}$
- $\delta_{te_ma} (MI\bar{b}_3, mi\#\#) = \{SOL(b)\}$
- $\delta_{te_ma} (SOL(b), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (MI_3, sol) = \{SOL(\#)\}$
- $\delta_{te_ma} (MI_3, fa\#\#) = \{SOL(\#)\}$
- $\delta_{te_ma} (MI_3, lab\#\bar{b}) = \{SOL(\#)\}$
- $\delta_{te_ma} (MI_3, sol\#) = \{SOL(\#)\}$
- $\delta_{te_ma} (MI_3, la\#\bar{b}) = \{SOL(\#)\}$
- $\delta_{te_ma} (SOL(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (FA\#\#_2, la\#) = \{LA\#(\#)\}$
- $\delta_{te_ma} (FA\#\#_2, si\#\bar{b}) = \{LA\#(\#)\}$
- $\delta_{te_ma} (FA\#\#_2, dob\#\bar{b}) = \{LA\#(\#)\}$
- $\delta_{te_ma} (FA\#\#_2, la\#\#\bar{b}) = \{LA\#(\#)\}$
- $\delta_{te_ma} (FA\#\#_2, si) = \{LA\#(\#)\}$
- $\delta_{te_ma} (FA\#\#_2, do\bar{b}) = \{LA\#(\#)\}$
- $\delta_{te_ma} (LA\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (FA_3, la) = \{LA(b)\}$
- $\delta_{te_ma} (FA_3, si\#\bar{b}) = \{LA(b)\}$
- $\delta_{te_ma} (FA_3, sol\#\#) = \{LA(b)\}$
- $\delta_{te_ma} (FA_3, lab) = \{LA(b)\}$
- $\delta_{te_ma} (FA_3, sol\#) = \{LA(b)\}$

- $\delta_{te_ma} (LA(b), \varepsilon) = \{q_f\}$
- $\delta_{te_ma} (FA\#_3, la) = \{LA(\#)\}$
- $\delta_{te_ma} (FA\#_3, sol\#\#\#) = \{LA(\#)\}$
- $\delta_{te_ma} (FA\#_3, si\#\#\#) = \{LA(\#)\}$
- $\delta_{te_ma} (FA\#_3, do\#\#\#) = \{LA(\#)\}$
- $\delta_{te_ma} (LA(\#), \varepsilon) = \{q_f\}$

b. Autômato que Reconhece Tétrade formado a partir do AFN ε T_ME

O AFN ε que reconhece Tétrade maior ou menor formado a partir do que reconhece uma Tríade Menor, denotamos como TE_ME, é definido formalmente como segue:

TE_ME = $\langle \sum_{te_me}, Q_{te_me}, F, \delta_{te_me}, Q_0 \rangle$, em que:

- $\sum_{te_me} = \sum_{t_me}$;
- $Q_{te_me} = \{SI(\#), SIb(b), SI(b), DO\#(\#), DO(b), DO(\#), RE\#(\#), RE(b), RE(\#), MI(\#), MIb(b), MI(b), FA\#(\#), FA(b), FA(\#), SOL\#(\#), SOL(b), SOL(\#), LA\#(\#), LA(b), LA(\#)\} \cup Q_{t_me}$;
- $Q_0 = q_0$;
- $F = \{q_f\}$;
- δ_{te_me} tem todas as transições da função δ_{t_me} , exceto as que levam ao estado final, mais as seguintes:

- $\delta_{te_me} (SOL\#, si) = \{SI(\#)\}$
- $\delta_{te_me} (SOL\#, la\#\#\#) = \{SI(\#)\}$
- $\delta_{te_me} (SOL\#, do\#\#\#) = \{SI(\#)\}$
- $\delta_{te_me} (SOL\#, si\#\#\#) = \{SI(\#)\}$
- $\delta_{te_me} (SOL\#, do\#\#\#) = \{SI(\#)\}$
- $\delta_{te_me} (SOL\#, re\#\#\#) = \{SI(\#)\}$
- $\delta_{te_me} (SI(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (SOLb, si\#\#\#) = \{SIb(b)\}$
- $\delta_{te_me} (SOLb, do\#\#\#) = \{SIb(b)\}$
- $\delta_{te_me} (SOLb, la\#\#\#) = \{SIb(b)\}$
- $\delta_{te_me} (SOLb, si\#\#\#) = \{SIb(b)\}$
- $\delta_{te_me} (SOLb, la\#\#\#) = \{SIb(b)\}$
- $\delta_{te_me} (SOLb, sol\#\#\#) = \{SIb(b)\}$
- $\delta_{te_me} (SIb(b), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (SOL, si) = \{SI(b)\}$
- $\delta_{te_me} (SOL, do\#\#\#) = \{SI(b)\}$
- $\delta_{te_me} (SOL, la\#\#\#) = \{SI(b)\}$
- $\delta_{te_me} (SOL, si\#\#\#) = \{SI(b)\}$
- $\delta_{te_me} (SOL, do\#\#\#) = \{SI(b)\}$
- $\delta_{te_me} (SI(b), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (LA\#, do\#\#\#) = \{DO\#(\#)\}$
- $\delta_{te_me} (LA\#, si\#\#\#) = \{DO\#(\#)\}$
- $\delta_{te_me} (LA\#, do\#\#\#) = \{DO\#(\#)\}$
- $\delta_{te_me} (LA\#, re) = \{DO\#(\#)\}$
- $\delta_{te_me} (LA\#, mi\#\#\#) = \{DO\#(\#)\}$
- $\delta_{te_me} (DO\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (LA\#, do) = \{DO(b)\}$
- $\delta_{te_me} (LA\#, re\#\#\#) = \{DO(b)\}$
- $\delta_{te_me} (LA\#, si\#\#\#) = \{DO(b)\}$
- $\delta_{te_me} (LA\#, do\#\#\#) = \{DO(b)\}$
- $\delta_{te_me} (LA\#, si) = \{DO(b)\}$
- $\delta_{te_me} (LA\#, la\#\#\#) = \{DO(b)\}$
- $\delta_{te_me} (DO\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (LA, do) = \{DO(\#)\}$
- $\delta_{te_me} (LA, re\#\#\#) = \{DO(\#)\}$
- $\delta_{te_me} (LA, si\#\#\#) = \{DO(\#)\}$
- $\delta_{te_me} (LA, do\#\#\#) = \{DO(\#)\}$
- $\delta_{te_me} (LA, re\#\#\#) = \{DO(\#)\}$
- $\delta_{te_me} (LA, si\#\#\#) = \{DO(\#)\}$
- $\delta_{te_me} (DO\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (SI\#, re\#\#\#) = \{RE\#(\#)\}$
- $\delta_{te_me} (SI\#, mi\#\#\#) = \{RE\#(\#)\}$

- $\delta_{te_me} (SI\#, fa\mathbf{b}\mathbf{b}) = \{RE\#(\#)\}$
- $\delta_{te_me} (SI\#, re\#\#) = \{RE\#(\#)\}$
- $\delta_{te_me} (SI\#, mi) = \{RE\#(\#)\}$
- $\delta_{te_me} (SI\#, fab) = \{RE\#(\#)\}$
- $\delta_{te_me} (RE\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (SI\mathbf{b}, re) = \{RE(\mathbf{b})\}$
- $\delta_{te_me} (SI\mathbf{b}, do\#\#) = \{RE(\mathbf{b})\}$
- $\delta_{te_me} (SI\mathbf{b}, mi\mathbf{b}\mathbf{b}) = \{RE(\mathbf{b})\}$
- $\delta_{te_me} (SI\mathbf{b}, reb) = \{RE(\mathbf{b})\}$
- $\delta_{te_me} (SI\mathbf{b}, do\#) = \{RE(\mathbf{b})\}$
- $\delta_{te_me} (SI\mathbf{b}, si\#\#) = \{RE(\mathbf{b})\}$
- $\delta_{te_me} (RE(\mathbf{b}), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (SI, re) = \{RE(\#)\}$
- $\delta_{te_me} (SI, do\#\#) = \{RE(\#)\}$
- $\delta_{te_me} (SI, mi\mathbf{b}\mathbf{b}) = \{RE(\#)\}$
- $\delta_{te_me} (SI, re\#) = \{RE(\#)\}$
- $\delta_{te_me} (SI, mi\mathbf{b}) = \{RE(\#)\}$
- $\delta_{te_me} (SI, fab\mathbf{b}) = \{RE(\#)\}$
- $\delta_{te_me} (RE(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (DO\#_2, mi) = \{MI(\#)\}$
- $\delta_{te_me} (DO\#_2, fab) = \{MI(\#)\}$
- $\delta_{te_me} (DO\#_2, re\#\#) = \{MI(\#)\}$
- $\delta_{te_me} (DO\#_2, mi\#) = \{MI(\#)\}$
- $\delta_{te_me} (DO\#_2, fa) = \{MI(\#)\}$
- $\delta_{te_me} (DO\#_2, sol\mathbf{b}\mathbf{b}) = \{MI(\#)\}$
- $\delta_{te_me} (MI(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (DO\mathbf{b}_2, mi\mathbf{b}) = \{MI\mathbf{b}(\mathbf{b})\}$
- $\delta_{te_me} (DO\mathbf{b}_2, reb) = \{MI\mathbf{b}(\mathbf{b})\}$
- $\delta_{te_me} (DO\mathbf{b}_2, fab\mathbf{b}) = \{MI\mathbf{b}(\mathbf{b})\}$
- $\delta_{te_me} (DO\mathbf{b}_2, mi\mathbf{b}\mathbf{b}) = \{MI\mathbf{b}(\mathbf{b})\}$
- $\delta_{te_me} (DO\mathbf{b}_2, re) = \{MI\mathbf{b}(\mathbf{b})\}$
- $\delta_{te_me} (DO\mathbf{b}_2, do\#\#) = \{MI\mathbf{b}(\mathbf{b})\}$
- $\delta_{te_me} (MI\mathbf{b}(\mathbf{b}), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (DO_2, mi) = \{MI(\mathbf{b})\}$
- $\delta_{te_me} (DO_2, fab) = \{MI(\mathbf{b})\}$
- $\delta_{te_me} (DO_2, re\#\#) = \{MI(\mathbf{b})\}$
- $\delta_{te_me} (DO_2, mi\mathbf{b}) = \{MI(\mathbf{b})\}$
- $\delta_{te_me} (DO_2, reb) = \{MI(\mathbf{b})\}$
- $\delta_{te_me} (DO_2, fab\mathbf{b}) = \{MI(\mathbf{b})\}$
- $\delta_{te_me} (MI(\mathbf{b}), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (RE\#_2, fa\#) = \{FA\#(\#)\}$
- $\delta_{te_me} (RE\#_2, mi\#\#) = \{FA\#(\#)\}$
- $\delta_{te_me} (RE\#_2, sol\mathbf{b}) = \{FA\#(\#)\}$
- $\delta_{te_me} (RE\#_2, fa\#\#) = \{FA\#(\#)\}$
- $\delta_{te_me} (RE\#_2, sol) = \{FA\#(\#)\}$
- $\delta_{te_me} (FA\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (RE\mathbf{b}_2, fa) = \{FA(\mathbf{b})\}$
- $\delta_{te_me} (RE\mathbf{b}_2, mi\#) = \{FA(\mathbf{b})\}$
- $\delta_{te_me} (RE\mathbf{b}_2, sol\mathbf{b}\mathbf{b}) = \{FA(\mathbf{b})\}$
- $\delta_{te_me} (RE\mathbf{b}_2, fab) = \{FA(\mathbf{b})\}$
- $\delta_{te_me} (RE\mathbf{b}_2, mi) = \{FA(\mathbf{b})\}$
- $\delta_{te_me} (RE\mathbf{b}_2, re\#\#) = \{FA(\mathbf{b})\}$
- $\delta_{te_me} (FA(\mathbf{b}), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (RE_2, fa) = \{FA(\#)\}$
- $\delta_{te_me} (RE_2, mi\#) = \{FA(\#)\}$
- $\delta_{te_me} (RE_2, sol\mathbf{b}\mathbf{b}) = \{FA(\#)\}$
- $\delta_{te_me} (RE_2, fa\#) = \{FA(\#)\}$
- $\delta_{te_me} (RE_2, sol\mathbf{b}) = \{FA(\#)\}$
- $\delta_{te_me} (RE_2, mi\#\#) = \{FA(\#)\}$
- $\delta_{te_me} (FA(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (MI\#_2, sol\#) = \{SOL\#(\#)\}$
- $\delta_{te_me} (MI\#_2, lab) = \{SOL\#(\#)\}$
- $\delta_{te_me} (MI\#_2, sol\#\#) = \{SOL\#(\#)\}$
- $\delta_{te_me} (MI\#_2, la) = \{SOL\#(\#)\}$
- $\delta_{te_me} (MI\#_2, si\mathbf{b}\mathbf{b}) = \{SOL\#(\#)\}$
- $\delta_{te_me} (SOL\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (MI\mathbf{b}_3, sol) = \{SOL(\mathbf{b})\}$
- $\delta_{te_me} (MI\mathbf{b}_3, fa\#\#) = \{SOL(\mathbf{b})\}$
- $\delta_{te_me} (MI\mathbf{b}_3, lab\mathbf{b}) = \{SOL(\mathbf{b})\}$
- $\delta_{te_me} (MI\mathbf{b}_3, sol\mathbf{b}) = \{SOL(\mathbf{b})\}$
- $\delta_{te_me} (MI\mathbf{b}_3, fa\#) = \{SOL(\mathbf{b})\}$
- $\delta_{te_me} (MI\mathbf{b}_3, mi\#\#) = \{SOL(\mathbf{b})\}$
- $\delta_{te_me} (SOL(\mathbf{b}), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (MI_3, sol) = \{SOL(\#)\}$
- $\delta_{te_me} (MI_3, fa\#\#) = \{SOL(\#)\}$
- $\delta_{te_me} (MI_3, lab) = \{SOL(\#)\}$
- $\delta_{te_me} (MI_3, sol\#) = \{SOL(\#)\}$
- $\delta_{te_me} (MI_3, la) = \{SOL(\#)\}$
- $\delta_{te_me} (SOL(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (FA\#\#, la\#) = \{LA\#(\#)\}$
- $\delta_{te_me} (FA\#\#, si\mathbf{b}) = \{LA\#(\#)\}$
- $\delta_{te_me} (FA\#\#, dob\mathbf{b}) = \{LA\#(\#)\}$
- $\delta_{te_me} (FA\#\#, la\#\#) = \{LA\#(\#)\}$
- $\delta_{te_me} (FA\#\#, si) = \{LA\#(\#)\}$
- $\delta_{te_me} (FA\#\#, dob) = \{LA\#(\#)\}$
- $\delta_{te_me} (LA\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_me} (FA_3, la) = \{LA(\mathbf{b})\}$
- $\delta_{te_me} (FA_3, si\mathbf{b}\mathbf{b}) = \{LA(\mathbf{b})\}$
- $\delta_{te_me} (FA_3, sol\#\#) = \{LA(\mathbf{b})\}$

- $\delta_{te_me}(FA_3, lab) = \{LA(b)\}$
- $\delta_{te_me}(FA_3, sol\#) = \{LA(b)\}$
- $\delta_{te_me}(LA(b), \varepsilon) = \{q_f\}$
- $\delta_{te_me}(FA\#, la) = \{LA(\#\})$
- $\delta_{te_me}(FA\#, sol\#\#) = \{LA(\#\})$
- $\delta_{te_me}(FA\#, sib\#b) = \{LA(\#\})$
- $\delta_{te_me}(FA\#, la\#) = \{LA(\#\})$
- $\delta_{te_me}(FA\#, sib) = \{LA(\#\})$
- $\delta_{te_me}(FA\#, dob\#b) = \{LA(\#\})$
- $\delta_{te_me}(LA(\#), \varepsilon) = \{q_f\}$

c. Autômato que Reconhece Tétrade formado a partir do AFN ε T_DIM

O AFN ε que reconhece Tétrade maior ou menor formado a partir do que reconhece uma Tríade Diminuta, denotamos como TE_DIM, é definido formalmente como segue:

TE_DIM = $\langle \Sigma_{te_dim}, Q_{te_dim}, F, \delta_{te_dim}, Q_0 \rangle$, em que:

- $\Sigma_{te_dim} = \Sigma_{t_dim}$;
- $Q_{te_dim} = \{SI(\#), SIb(b), SI(b), DO\#(\#), DO(b), DO(\#), RE\#(\#), RE(b), RE(\#), MI(\#), MIb(b), MI(b), FA\#(\#), FA(b), FA(\#), SOL\#(\#), SOL(b), SOL(\#), LA\#(\#), LA(b), LA(\#)\} \cup Q_{t_dim}$;
- $Q_0 = q_0$;
- $F = \{q_f\}$;
- δ_{te_dim} tem todas as transições da função δ_{t_dim} , exceto as que levam ao estado final, mais as seguintes:

- $\delta_{te_dim}(SOL, si) = \{SI(\#)\}$
- $\delta_{te_dim}(SOL, la\#\#) = \{SI(\#)\}$
- $\delta_{te_dim}(SOL, dob) = \{SI(\#)\}$
- $\delta_{te_dim}(SOL, si\#) = \{SI(\#)\}$
- $\delta_{te_dim}(SOL, do) = \{SI(\#)\}$
- $\delta_{te_dim}(SOL, reb\#b) = \{SI(\#)\}$
- $\delta_{te_dim}(SI(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(SOLb\#b, sib) = \{SIb(b)\}$
- $\delta_{te_dim}(SOLb\#b, dob\#b) = \{SIb(b)\}$
- $\delta_{te_dim}(SOLb\#b, la\#) = \{SIb(b)\}$
- $\delta_{te_dim}(SOLb\#b, sib\#b) = \{SIb(b)\}$
- $\delta_{te_dim}(SOLb\#b, la) = \{SIb(b)\}$
- $\delta_{te_dim}(SOLb\#b, sol\#\#) = \{SIb(b)\}$
- $\delta_{te_dim}(SIb(b), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(SOLb, si) = \{SI(b)\}$
- $\delta_{te_dim}(SOLb, dob) = \{SI(b)\}$
- $\delta_{te_dim}(SOLb, la\#\#) = \{SI(b)\}$
- $\delta_{te_dim}(SOLb, si\#) = \{SI(b)\}$
- $\delta_{te_dim}(SOLb, la\#) = \{SI(b)\}$
- $\delta_{te_dim}(SOLb, dob\#b) = \{SI(b)\}$
- $\delta_{te_dim}(LA\#b, si) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, la\#\#) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, re\#b) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, si\#) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, do\#b) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, reb\#b) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, mi\#b) = \{DO(b)\}$
- $\delta_{te_dim}(DO\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(LA\#b, do) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, reb\#b) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, si\#) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, dob) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, si) = \{DO(b)\}$
- $\delta_{te_dim}(LA\#b, la\#\#) = \{DO(b)\}$
- $\delta_{te_dim}(DO\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(LA\#b, do) = \{DO(\#)\}$
- $\delta_{te_dim}(LA\#b, reb\#b) = \{DO(\#)\}$
- $\delta_{te_dim}(LA\#b, si\#) = \{DO(\#)\}$
- $\delta_{te_dim}(LA\#b, do\#) = \{DO(\#)\}$
- $\delta_{te_dim}(LA\#b, reb) = \{DO(\#)\}$

- $\delta_{te_dim}(LAB, si\#) = \{DO(\#)\}$
- $\delta_{te_dim}(DO\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(SI, re\#) = \{RE\#(\#)\}$
- $\delta_{te_dim}(SI, mi\textbf{b}) = \{RE\#(\#)\}$
- $\delta_{te_dim}(SI, fab\textbf{b}) = \{RE\#(\#)\}$
- $\delta_{te_dim}(SI, re\#\#\#) = \{RE\#(\#)\}$
- $\delta_{te_dim}(SI, mi) = \{RE\#(\#)\}$
- $\delta_{te_dim}(SI, fab) = \{RE\#(\#)\}$
- $\delta_{te_dim}(RE\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(SIB\textbf{b}, re) = \{RE(b)\}$
- $\delta_{te_dim}(SIB\textbf{b}, do\#\#\#) = \{RE(b)\}$
- $\delta_{te_dim}(SIB\textbf{b}, mi\textbf{b}\textbf{b}) = \{RE(b)\}$
- $\delta_{te_dim}(SIB\textbf{b}, reb) = \{RE(b)\}$
- $\delta_{te_dim}(SIB\textbf{b}, do\#) = \{RE(b)\}$
- $\delta_{te_dim}(SIB\textbf{b}, si\#\#\#) = \{RE(b)\}$
- $\delta_{te_dim}(RE(b), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(SIB, re) = \{RE(\#)\}$
- $\delta_{te_dim}(SIB, do\#\#\#) = \{RE(\#)\}$
- $\delta_{te_dim}(SIB, mi\textbf{b}\textbf{b}) = \{RE(\#)\}$
- $\delta_{te_dim}(SIB, re\#) = \{RE(\#)\}$
- $\delta_{te_dim}(SIB, mi\textbf{b}) = \{RE(\#)\}$
- $\delta_{te_dim}(SIB, fab\textbf{b}) = \{RE(\#)\}$
- $\delta_{te_dim}(RE(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(DO_2, mi) = \{MI(\#)\}$
- $\delta_{te_dim}(DO_2, fab) = \{MI(\#)\}$
- $\delta_{te_dim}(DO_2, re\#\#\#) = \{MI(\#)\}$
- $\delta_{te_dim}(DO_2, mi\#) = \{MI(\#)\}$
- $\delta_{te_dim}(DO_2, fa) = \{MI(\#)\}$
- $\delta_{te_dim}(DO_2, sol\textbf{b}\textbf{b}) = \{MI(\#)\}$
- $\delta_{te_dim}(MI(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(DOB\textbf{b}, mi\textbf{b}) = \{MIb(b)\}$
- $\delta_{te_dim}(DOB\textbf{b}, re\#) = \{MIb(b)\}$
- $\delta_{te_dim}(DOB\textbf{b}, fab\textbf{b}) = \{MIb(b)\}$
- $\delta_{te_dim}(DOB\textbf{b}, mi\textbf{b}\textbf{b}) = \{MIb(b)\}$
- $\delta_{te_dim}(DOB\textbf{b}, re) = \{MIb(b)\}$
- $\delta_{te_dim}(DOB\textbf{b}, do\#\#\#) = \{MIb(b)\}$
- $\delta_{te_dim}(MIb(b), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(DOB_2, mi) = \{MI(b)\}$
- $\delta_{te_dim}(DOB_2, fab) = \{MI(b)\}$
- $\delta_{te_dim}(DOB_2, re\#\#\#) = \{MI(b)\}$
- $\delta_{te_dim}(DOB_2, mi\textbf{b}) = \{MI(b)\}$
- $\delta_{te_dim}(DOB_2, re\#) = \{MI(b)\}$
- $\delta_{te_dim}(DOB_2, fab\textbf{b}) = \{MI(b)\}$
- $\delta_{te_dim}(MI(b), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(RE_2, fa\#) = \{FA\#(\#)\}$
- $\delta_{te_dim}(RE_2, mi\#\#\#) = \{FA\#(\#)\}$
- $\delta_{te_dim}(RE_2, sol\textbf{b}) = \{FA\#(\#)\}$
- $\delta_{te_dim}(RE_2, fa\#\#\#) = \{FA\#(\#)\}$
- $\delta_{te_dim}(RE_2, sol) = \{FA\#(\#)\}$
- $\delta_{te_dim}(RE_2, lab\textbf{b}) = \{FA\#(\#)\}$
- $\delta_{te_dim}(FA\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(REb\textbf{b}, fa) = \{FA(b)\}$
- $\delta_{te_dim}(REb\textbf{b}, mi\#) = \{FA(b)\}$
- $\delta_{te_dim}(REb\textbf{b}, sol\textbf{b}\textbf{b}) = \{FA(b)\}$
- $\delta_{te_dim}(REb\textbf{b}, fab) = \{FA(b)\}$
- $\delta_{te_dim}(REb\textbf{b}, mi) = \{FA(b)\}$
- $\delta_{te_dim}(REb\textbf{b}, re\#\#\#) = \{FA(b)\}$
- $\delta_{te_dim}(FA(b), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(REb_2, fa) = \{FA(\#)\}$
- $\delta_{te_dim}(REb_2, mi\#) = \{FA(\#)\}$
- $\delta_{te_dim}(REb_2, sol\textbf{b}\textbf{b}) = \{FA(\#)\}$
- $\delta_{te_dim}(REb_2, fa\#) = \{FA(\#)\}$
- $\delta_{te_dim}(REb_2, sol\textbf{b}) = \{FA(\#)\}$
- $\delta_{te_dim}(REb_2, mi\#\#\#) = \{FA(\#)\}$
- $\delta_{te_dim}(FA(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(MI_3, sol\#) = \{SOL\#(\#)\}$
- $\delta_{te_dim}(MI_3, lab) = \{SOL\#(\#)\}$
- $\delta_{te_dim}(MI_3, sol\#\#\#) = \{SOL\#(\#)\}$
- $\delta_{te_dim}(MI_3, la) = \{SOL\#(\#)\}$
- $\delta_{te_dim}(MI_3, si\textbf{b}\textbf{b}) = \{SOL\#(\#)\}$
- $\delta_{te_dim}(SOL\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(MIb\textbf{b}_2, sol) = \{SOL(b)\}$
- $\delta_{te_dim}(MIb\textbf{b}_2, fa\#\#\#) = \{SOL(b)\}$
- $\delta_{te_dim}(MIb\textbf{b}_2, lab\textbf{b}) = \{SOL(b)\}$
- $\delta_{te_dim}(MIb\textbf{b}_2, sol\textbf{b}) = \{SOL(b)\}$
- $\delta_{te_dim}(MIb\textbf{b}_2, fa\#) = \{SOL(b)\}$
- $\delta_{te_dim}(MIb\textbf{b}_2, mi\#\#\#) = \{SOL(b)\}$
- $\delta_{te_dim}(SOL(b), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(MIb_3, sol) = \{SOL(\#)\}$
- $\delta_{te_dim}(MIb_3, fa\#\#\#) = \{SOL(\#)\}$
- $\delta_{te_dim}(MIb_3, lab\textbf{b}) = \{SOL(\#)\}$
- $\delta_{te_dim}(MIb_3, sol\#) = \{SOL(\#)\}$
- $\delta_{te_dim}(MIb_3, lab) = \{SOL(\#)\}$
- $\delta_{te_dim}(SOL(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(FA\#_3, la\#) = \{LA\#(\#)\}$
- $\delta_{te_dim}(FA\#_3, si\textbf{b}) = \{LA\#(\#)\}$
- $\delta_{te_dim}(FA\#_3, do\textbf{b}\textbf{b}) = \{LA\#(\#)\}$
- $\delta_{te_dim}(FA\#_3, la\#\#\#) = \{LA\#(\#)\}$

- $\delta_{te_dim}(FA\#_3, si) = \{LA\#(\#)\}$
- $\delta_{te_dim}(FA\#_3, do\b{b}) = \{LA\#(\#)\}$
- $\delta_{te_dim}(LA\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(FA\b{b}_3, la) = \{LA(\b{b})\}$
- $\delta_{te_dim}(FA\b{b}_3, si\b{b}\b{b}) = \{LA(\b{b})\}$
- $\delta_{te_dim}(FA\b{b}_3, sol\#\#\b{b}) = \{LA(\b{b})\}$
- $\delta_{te_dim}(FA\b{b}_3, lab) = \{LA(\b{b})\}$
- $\delta_{te_dim}(FA\b{b}_3, sol\#) = \{LA(\b{b})\}$
- $\delta_{te_dim}(LA(\b{b}), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(LA(\b{b}), \varepsilon) = \{q_f\}$
- $\delta_{te_dim}(FA_3, la) = \{LA(\#)\}$
- $\delta_{te_dim}(FA_3, sol\#\#\b{b}) = \{LA(\#)\}$
- $\delta_{te_dim}(FA_3, si\b{b}\b{b}) = \{LA(\#)\}$
- $\delta_{te_dim}(FA_3, la\#) = \{LA(\#)\}$
- $\delta_{te_dim}(FA_3, si\b{b}) = \{LA(\#)\}$
- $\delta_{te_dim}(FA_3, do\b{b}\b{b}) = \{LA(\#)\}$
- $\delta_{te_dim}(LA(\#), \varepsilon) = \{q_f\}$

d. Autômato que Reconhece Tétrade formado a partir do AFNε T_AUM

O AFNε que reconhece Tétrade maior ou menor formado a partir do que reconhece uma Tríade Aumentada, denotamos como TE_AUM, é definido formalmente como segue:

$TE_AUM = <\sum_{te_aum}, Q_{te_aum}, F, \delta_{te_aum}, Q_0>$, em que:

- $\sum_{te_aum} = \sum_{t_aum};$
- $Q_{te_aum} = \{SI(\#), SI\b{b}(\b{b}), SI(\b{b}), DO\#(\#), DO(\b{b}), DO(\#), RE\#(\#), RE(\b{b}), RE(\#), MI(\#), MI\b{b}(\b{b}), MI(\b{b}), FA\#(\#), FA(\b{b}), FA(\#), SOL\#(\#), SOL(\b{b}), SOL(\#), LA\#(\#), LA(\b{b}), LA(\#)\} \cup Q_{t_aum};$
- $Q_0 = q_0;$
- $F = \{q_f\};$
- δ_{te_aum} tem todas as transições da função δ_{t_aum} , exceto as que levam ao estado final, mais as seguintes:

- $\delta_{te_aum}(SOL\#\#\b{b}, si) = \{SI(\#)\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, la\#\#\b{b}) = \{SI(\#)\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, do\b{b}) = \{SI(\#)\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, si\#) = \{SI(\#)\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, do) = \{SI(\#)\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, reb\b{b}) = \{SI(\#)\}$
- $\delta_{te_aum}(SI(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum}(SOL, si\b{b}) = \{SI\b{b}(\b{b})\}$
- $\delta_{te_aum}(SOL, do\b{b}\b{b}) = \{SI\b{b}(\b{b})\}$
- $\delta_{te_aum}(SOL, la\#) = \{SI\b{b}(\b{b})\}$
- $\delta_{te_aum}(SOL, si\b{b}\b{b}) = \{SI\b{b}(\b{b})\}$
- $\delta_{te_aum}(SOL, la) = \{SI\b{b}(\b{b})\}$
- $\delta_{te_aum}(SOL, sol\#\#\b{b}) = \{SI\b{b}(\b{b})\}$
- $\delta_{te_aum}(SI\b{b}(\b{b}), \varepsilon) = \{q_f\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, si) = \{SI(\#)\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, do\b{b}) = \{SI(\b{b})\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, la\#\#\b{b}) = \{SI(\b{b})\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, si\#) = \{SI(\b{b})\}$
- $\delta_{te_aum}(SOL\#\#\b{b}, do\b{b}\b{b}) = \{SI(\b{b})\}$
- $\delta_{te_aum}(SI(\b{b}), \varepsilon) = \{q_f\}$
- $\delta_{te_aum}(LA\#\#\b{b}, do\#) = \{DO\#(\#)\}$
- $\delta_{te_aum}(LA\#\#\b{b}, reb\b{b}) = \{DO\#(\#)\}$
- $\delta_{te_aum}(LA\#\#\b{b}, si\#\#\b{b}) = \{DO\#(\#)\}$
- $\delta_{te_aum}(LA\#\#\b{b}, do\#\#\b{b}) = \{DO\#(\#)\}$
- $\delta_{te_aum}(LA\#\#\b{b}, re) = \{DO\#(\#)\}$
- $\delta_{te_aum}(LA\#\#\b{b}, mi\b{b}\b{b}) = \{DO\#(\#)\}$
- $\delta_{te_aum}(DO\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum}(LA, do) = \{DO(\b{b})\}$
- $\delta_{te_aum}(LA, reb\b{b}) = \{DO(\b{b})\}$

- $\delta_{te_aum} (LA, si\#) = \{DO(b)\}$
- $\delta_{te_aum} (LA, do\#b) = \{DO(b)\}$
- $\delta_{te_aum} (LA, si) = \{DO(b)\}$
- $\delta_{te_aum} (LA, la\#\#) = \{DO(b)\}$
- $\delta_{te_aum} (DO\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (LA\#, do) = \{DO(\#)\}$
- $\delta_{te_aum} (LA\#, reb\#b) = \{DO(\#)\}$
- $\delta_{te_aum} (LA\#, si\#) = \{DO(\#)\}$
- $\delta_{te_aum} (LA\#, do\#) = \{DO(\#)\}$
- $\delta_{te_aum} (LA\#, reb) = \{DO(\#)\}$
- $\delta_{te_aum} (LA\#, si\#\#) = \{DO(\#)\}$
- $\delta_{te_aum} (DO\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (SI\#\#, re\#) = \{RE\#(\#)\}$
- $\delta_{te_aum} (SI\#\#, mi\#b) = \{RE\#(\#)\}$
- $\delta_{te_aum} (SI\#\#, fab\#b) = \{RE\#(\#)\}$
- $\delta_{te_aum} (SI\#\#, re\#\#) = \{RE\#(\#)\}$
- $\delta_{te_aum} (SI\#\#, mi) = \{RE\#(\#)\}$
- $\delta_{te_aum} (SI\#\#, fab) = \{RE\#(\#)\}$
- $\delta_{te_aum} (RE\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (SI, re) = \{RE(b)\}$
- $\delta_{te_aum} (SI, do\#\#) = \{RE(b)\}$
- $\delta_{te_aum} (SI, mi\#bb) = \{RE(b)\}$
- $\delta_{te_aum} (SI, reb) = \{RE(b)\}$
- $\delta_{te_aum} (SI, do\#) = \{RE(b)\}$
- $\delta_{te_aum} (SI, si\#\#) = \{RE(b)\}$
- $\delta_{te_aum} (RE(b), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (SI\#, re) = \{RE(\#)\}$
- $\delta_{te_aum} (SI\#, do\#\#) = \{RE(\#)\}$
- $\delta_{te_aum} (SI\#, mi\#bb) = \{RE(\#)\}$
- $\delta_{te_aum} (SI\#, reb\#) = \{RE(\#)\}$
- $\delta_{te_aum} (SI\#, mi\#b) = \{RE(\#)\}$
- $\delta_{te_aum} (SI\#, fab\#b) = \{RE(\#)\}$
- $\delta_{te_aum} (RE(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (DO\#\#, mi) = \{MI(\#)\}$
- $\delta_{te_aum} (DO\#\#, fab) = \{MI(\#)\}$
- $\delta_{te_aum} (DO\#\#, re\#\#) = \{MI(\#)\}$
- $\delta_{te_aum} (DO\#\#, mi\#) = \{MI(\#)\}$
- $\delta_{te_aum} (DO\#\#, fa) = \{MI(\#)\}$
- $\delta_{te_aum} (DO\#\#, sol\#bb) = \{MI(\#)\}$
- $\delta_{te_aum} (MI(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (DO_2, mi\#b) = \{MI\#(b)\}$
- $\delta_{te_aum} (DO_2, re\#) = \{MI\#(b)\}$
- $\delta_{te_aum} (DO_2, fab\#b) = \{MI\#(b)\}$
- $\delta_{te_aum} (DO_2, mi\#bb) = \{MI\#(b)\}$
- $\delta_{te_aum} (DO_2, reb\#b) = \{MI\#(b)\}$
- $\delta_{te_aum} (DO_2, re\#\#) = \{MI(b)\}$
- $\delta_{te_aum} (DO\#_2, mi\#b) = \{MI(b)\}$
- $\delta_{te_aum} (DO\#_2, re\#) = \{MI(b)\}$
- $\delta_{te_aum} (DO\#_2, fab\#b) = \{MI(b)\}$
- $\delta_{te_aum} (DO\#_2, fa\#) = \{MI(b)\}$
- $\delta_{te_aum} (RE\#\#, fa\#) = \{FA\#(\#)\}$
- $\delta_{te_aum} (RE\#\#, mi\#\#) = \{FA\#(\#)\}$
- $\delta_{te_aum} (RE\#\#, sol\#b) = \{FA\#(\#)\}$
- $\delta_{te_aum} (RE\#\#, fa\#\#) = \{FA\#(\#)\}$
- $\delta_{te_aum} (RE\#\#, sol) = \{FA\#(\#)\}$
- $\delta_{te_aum} (RE\#\#, lab\#b) = \{FA\#(\#)\}$
- $\delta_{te_aum} (FA\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (RE_2, fa) = \{FA(b)\}$
- $\delta_{te_aum} (RE_2, mi\#) = \{FA(b)\}$
- $\delta_{te_aum} (RE_2, sol\#bb) = \{FA(b)\}$
- $\delta_{te_aum} (RE_2, fab) = \{FA(b)\}$
- $\delta_{te_aum} (RE_2, mi) = \{FA(b)\}$
- $\delta_{te_aum} (RE_2, re\#\#) = \{FA(b)\}$
- $\delta_{te_aum} (FA(b), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (RE\#_2, fa) = \{FA(\#)\}$
- $\delta_{te_aum} (RE\#_2, mi\#) = \{FA(\#)\}$
- $\delta_{te_aum} (RE\#_2, sol\#bb) = \{FA(\#)\}$
- $\delta_{te_aum} (RE\#_2, fa\#) = \{FA(\#)\}$
- $\delta_{te_aum} (RE\#_2, sol\#b) = \{FA(\#)\}$
- $\delta_{te_aum} (RE\#_2, mi\#\#) = \{FA(\#)\}$
- $\delta_{te_aum} (FA(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (MI\#\#, sol\#) = \{SOL\#(\#)\}$
- $\delta_{te_aum} (MI\#\#, lab) = \{SOL\#(\#)\}$
- $\delta_{te_aum} (MI\#\#, sol\#\#) = \{SOL\#(\#)\}$
- $\delta_{te_aum} (MI\#\#, la) = \{SOL\#(\#)\}$
- $\delta_{te_aum} (MI\#\#, si\#bb) = \{SOL\#(\#)\}$
- $\delta_{te_aum} (SOL\#(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum} (MI_3, sol) = \{SOL(b)\}$
- $\delta_{te_aum} (MI_3, fa\#\#) = \{SOL(b)\}$
- $\delta_{te_aum} (MI_3, lab\#b) = \{SOL(b)\}$
- $\delta_{te_aum} (MI_3, sol\#b) = \{SOL(b)\}$
- $\delta_{te_aum} (MI_3, fa\#) = \{SOL(b)\}$
- $\delta_{te_aum} (MI_3, mi\#\#) = \{SOL(b)\}$
- $\delta_{te_aum} (SOL(b), \varepsilon) = \{q_f\}$

- $\delta_{te_aum}(MI\#_3, sol) = \{SOL(\#)\}$
- $\delta_{te_aum}(MI\#_3, fa\#\#\#) = \{SOL(\#)\}$
- $\delta_{te_aum}(MI\#_3, lab\mathbf{b}) = \{SOL(\#)\}$
- $\delta_{te_aum}(MI\#_3, sol\#\#) = \{SOL(\#)\}$
- $\delta_{te_aum}(MI\#_3, lab\mathbf{b}) = \{SOL(\#)\}$
- $\delta_{te_aum}(SOL(\#), \varepsilon) = \{q_f\}$
- $\delta_{te_aum}(FA\#_3, la) = \{LA(\mathbf{b})\}$
- $\delta_{te_aum}(FA\#_3, sib\mathbf{b}) = \{LA(\mathbf{b})\}$
- $\delta_{te_aum}(FA\#_3, sol\#\#\#) = \{LA(\mathbf{b})\}$
- $\delta_{te_aum}(FA\#_3, lab\mathbf{b}) = \{LA(\mathbf{b})\}$
- $\delta_{te_aum}(FA\#_3, sol\#\#) = \{LA(\mathbf{b})\}$
- $\delta_{te_aum}(FA\#_3, la) = \{LA(\mathbf{b})\}$
- $\delta_{te_aum}(FA\#_2, la\#\#) = \{LA(\#)\}$
- $\delta_{te_aum}(FA\#_2, sib\mathbf{b}) = \{LA(\#)\}$
- $\delta_{te_aum}(FA\#_2, la\#) = \{LA(\#)\}$
- $\delta_{te_aum}(FA\#_2, sib) = \{LA(\#)\}$
- $\delta_{te_aum}(FA\#_2, dob\mathbf{b}) = \{LA(\#)\}$
- $\delta_{te_aum}(LA(\#), \varepsilon) = \{q_f\}$

e. Autômato que Reconhece Tétrade (TETRADE)

O AFN ε que reconhece uma Tétrade, denotamos como TETRADE, é definido formalmente a partir dos Autômatos anteriormente definidos, a saber: TE_MA (reconhece Tétrade a partir da Tríade Maior), TE_ME (reconhece Tétrade a partir da Tríade Menor), TE_DIM (reconhece Tétrade a partir da Tríade Diminuta) e TE_AUM (reconhece Tétrade a partir Tríade da Aumentada). Definimos o Autômato TETRADE como segue:

TETRADE = $\langle \sum_{te}, Q_{te}, F_{te}, \delta_{te}, Q_0 \rangle$, em que:

- $\sum_{te} = \sum_{te_ma} \cup \sum_{te_me} \cup \sum_{te_dim} \cup \sum_{te_aum}$;
- $Q_{te} = \{q_0, q_f\} \cup Q_{te_ma} \cup Q_{te_me} \cup Q_{te_dim} \cup Q_{te_aum}$, para que não se perda nenhum estado ao fazer o processo de união, fizemos as seguintes mudanças:
 - Para todo $p \in Q_{te_ma}$, renomeamos p para p_{te_ma} ;
 - Ex.: O estado DO₁ de Q_{te_ma} , se torna DO_{1te_ma};
 - Para todo $p \in Q_{te_me}$, renomeamos p para p_{te_me} ;
 - Ex.: O estado DO₁ de Q_{te_ma} , se torna DO_{1te_me};
 - Para todo $p \in Q_{te_dim}$, renomeamos p para p_{te_dim} ;
 - Ex.: O estado DO₁ de Q_{te_dim} , se torna DO_{1te_dim};
 - Para todo $p \in Q_{te_aum}$, renomeamos p para p_{te_aum} ;
 - Ex.: O estado DO₁ de Q_{te_aum} , se torna DO_{1te_aum};
- $F_{te} = \{q_f\}$;
- $Q_0 = q_0$;
- δ_{te} possui todas as transições de δ_{te_ma} , δ_{te_me} , δ_{te_dim} e δ_{te_aum} , mais as seguintes:
 - $\delta_{te}(q_0, \varepsilon) = \{q_{0te_ma}, q_{0te_me}, q_{0te_dim}, q_{0te_aum}\}$
 - $\delta_{te}(q_{ft_ma}, \varepsilon) = \{q_f\}$
 - $\delta_{te}(q_{ft_me}, \varepsilon) = \{q_f\}$
 - $\delta_{te}(q_{ft_dim}, \varepsilon) = \{q_f\}$
 - $\delta_{te}(q_{ft_aum}, \varepsilon) = \{q_f\}$

7.3 APÊNDICE C – Reconhecedor de Tríades e Tétrade

O AFN ε que reconhece Acordes (Tríades e Tétrade), denotamos como ACORDE, é definido formalmente a partir dos Autômatos TRIADE e TETRADE, da seguinte maneira:

ACORDE = $\langle \sum_{ac}, Q_{ac}, F_{ac}, \delta_{ac}, Q_0 \rangle$, em que:

- $\sum_{ac} = \sum_{te} \cup \sum_{tri}$;
- $Q_{ac} = \{q_0, q_f, S_OR, S_*\} \cup Q_{te} \cup Q_{tri}$, para que não se perda nenhum estado ao fazer o processo de união, fizemos as seguintes mudanças:
 - Os estados inicial e final de Q_{te} renomeamos para q_{0_te} e q_{f_te} , respectivamente;
 - Os estados inicial e final de Q_{tri} renomeamos para q_{0_tri} e q_{f_tri} , respectivamente;
- $F_{ac} = \{q_f\}$;
- $Q_0 = q_0$;
- δ_{ac} possui todas as transições de δ_{te} e δ_{tri} , mais as seguintes:
 - $\delta_{ac}(q_0, \varepsilon) = \{S_OR\}$
 - $\delta_{ac}(S_OR, \varepsilon) = \{q_{0_te}, q_{0_tri}\}$
 - $\delta_{ac}(q_{f_tri}, \varepsilon) = \{S_*\}$
 - $\delta_{ac}(q_{f_te}, \varepsilon) = \{S_*\}$
 - $\delta_{ac}(S_*, \varepsilon) = \{q_f, S_OR\}$

7.4 APÊNDICE D – Listas de Telas do *Qmusic* referente ao Estudo de Caso

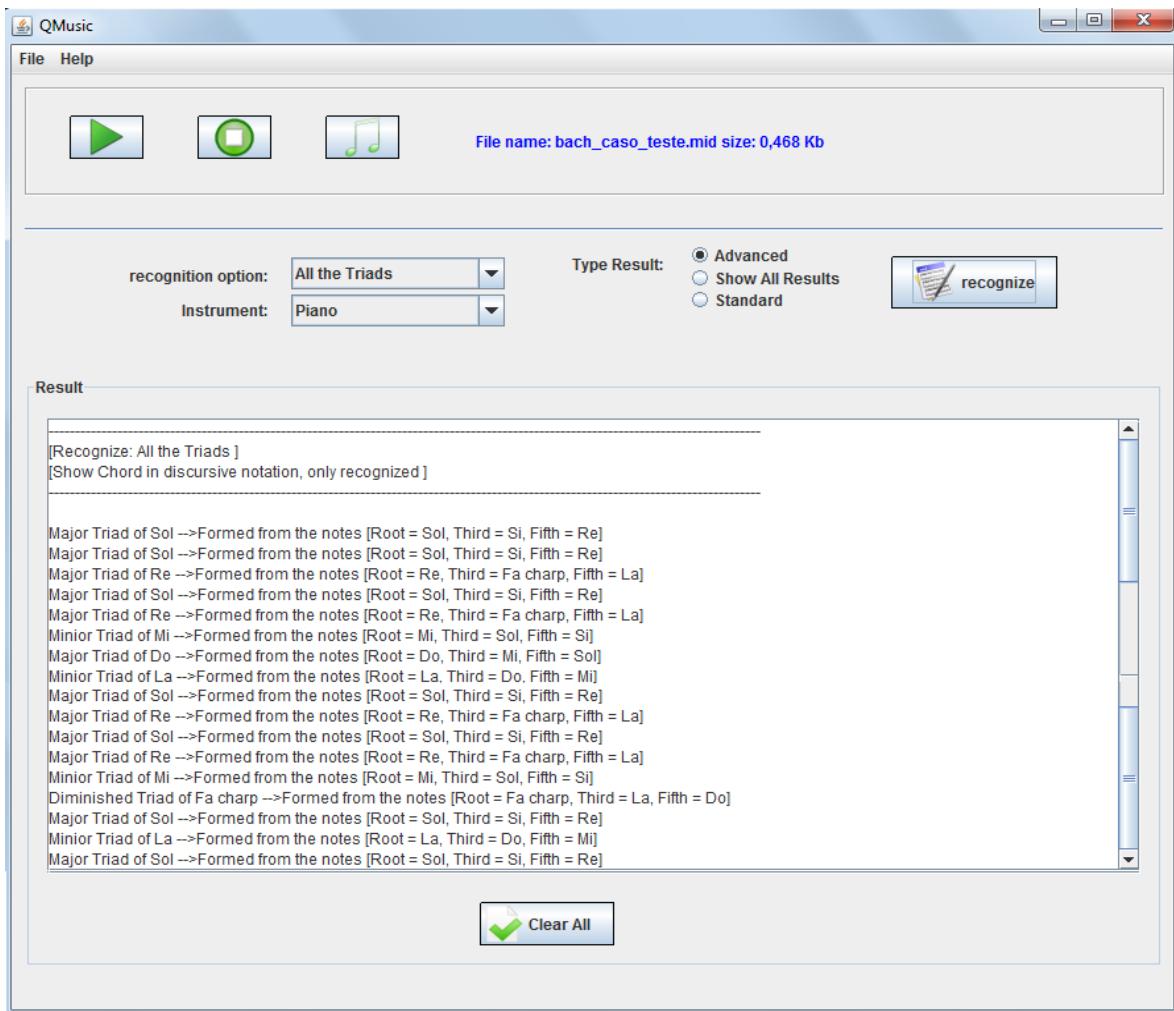


Figura 45 – Lista de Tríades da Seção A do coral *Aus Meines Herzens Grunde* de Bach

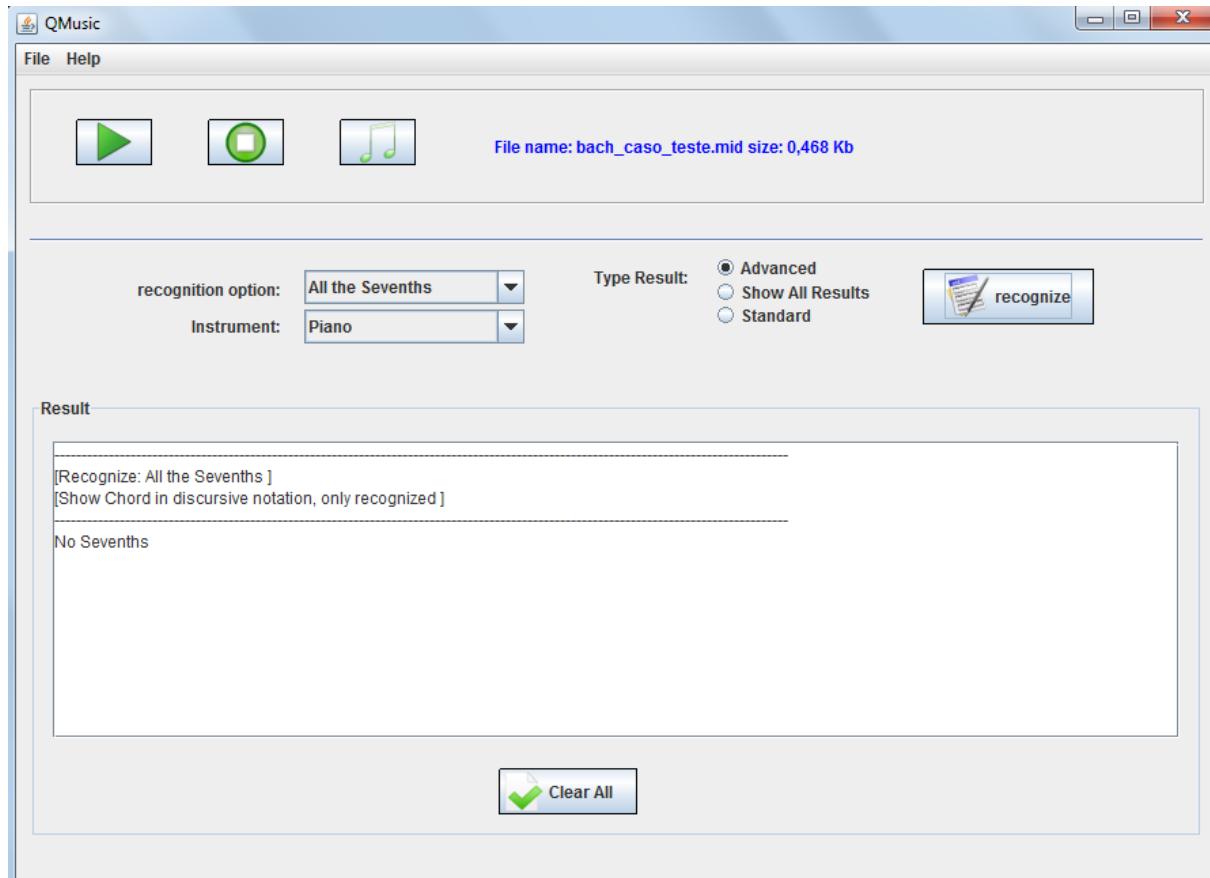


Figura 46 – Lista de Tétrades da Seção A do coral *Aus Meines Herzens Grunde* de Bach

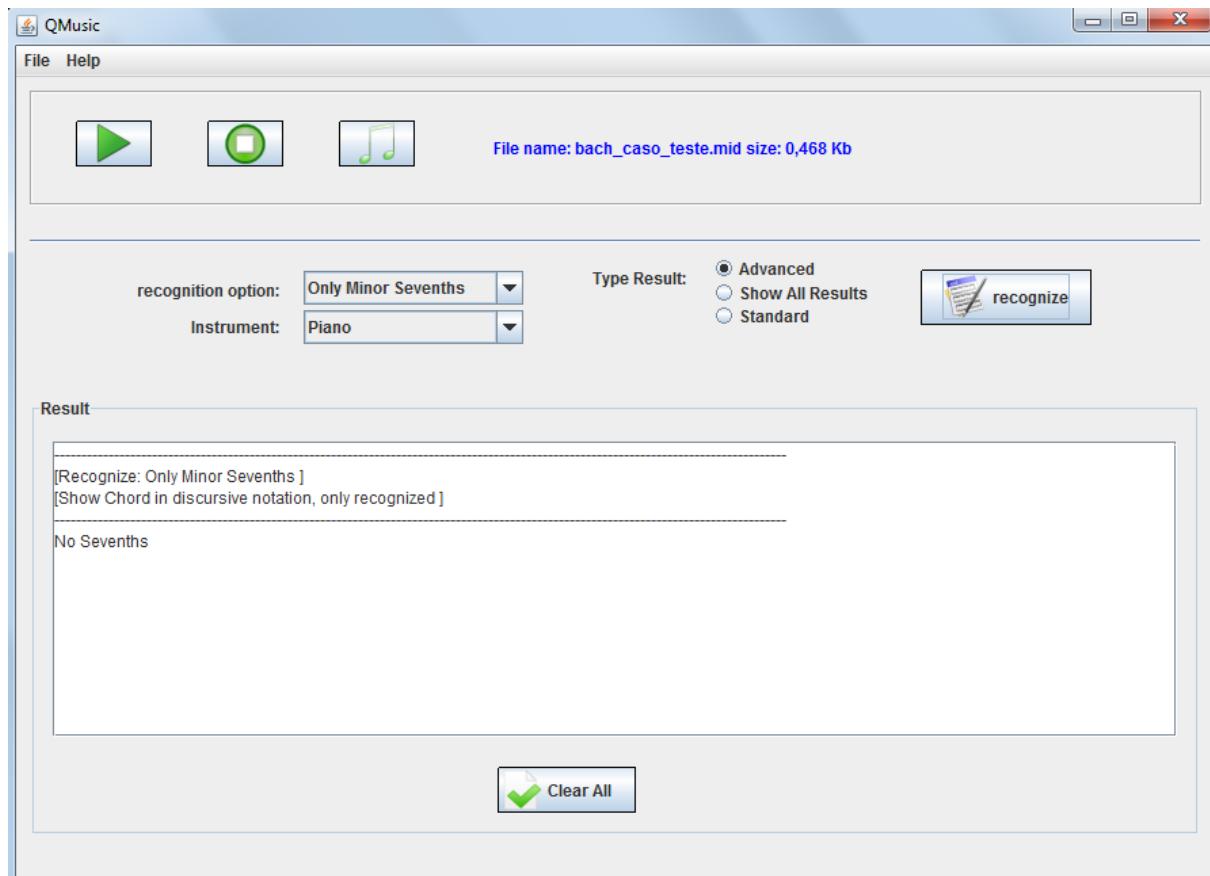


Figura 47 – Lista de Tétrades Menores da Seção A do coral *Aus Meines Herzens Grunde* de Bach

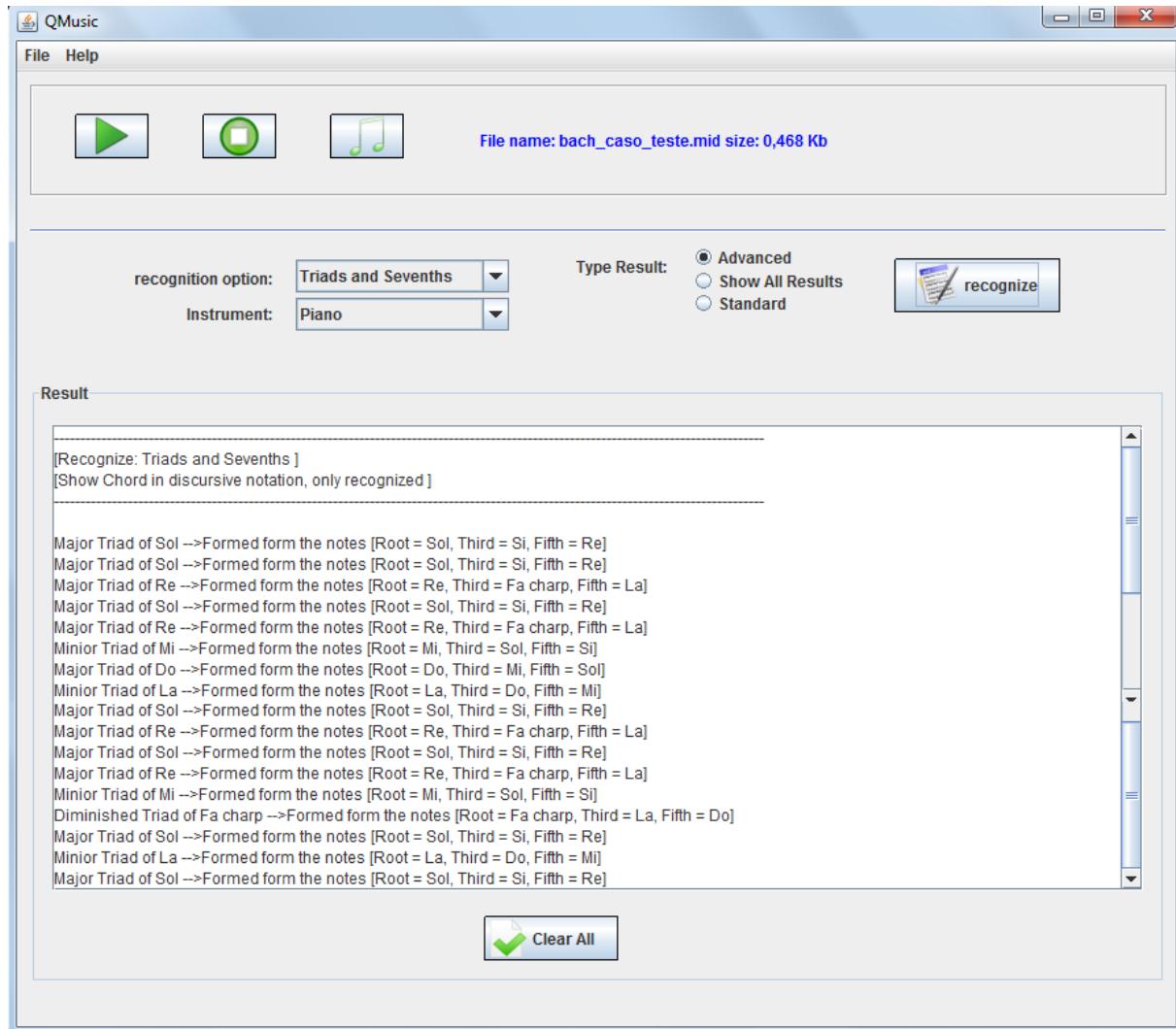


Figura 48 – Lista de Tríades e Tétrade da Seção A do coral *Aus Meines Herzens Grunde* de Bach

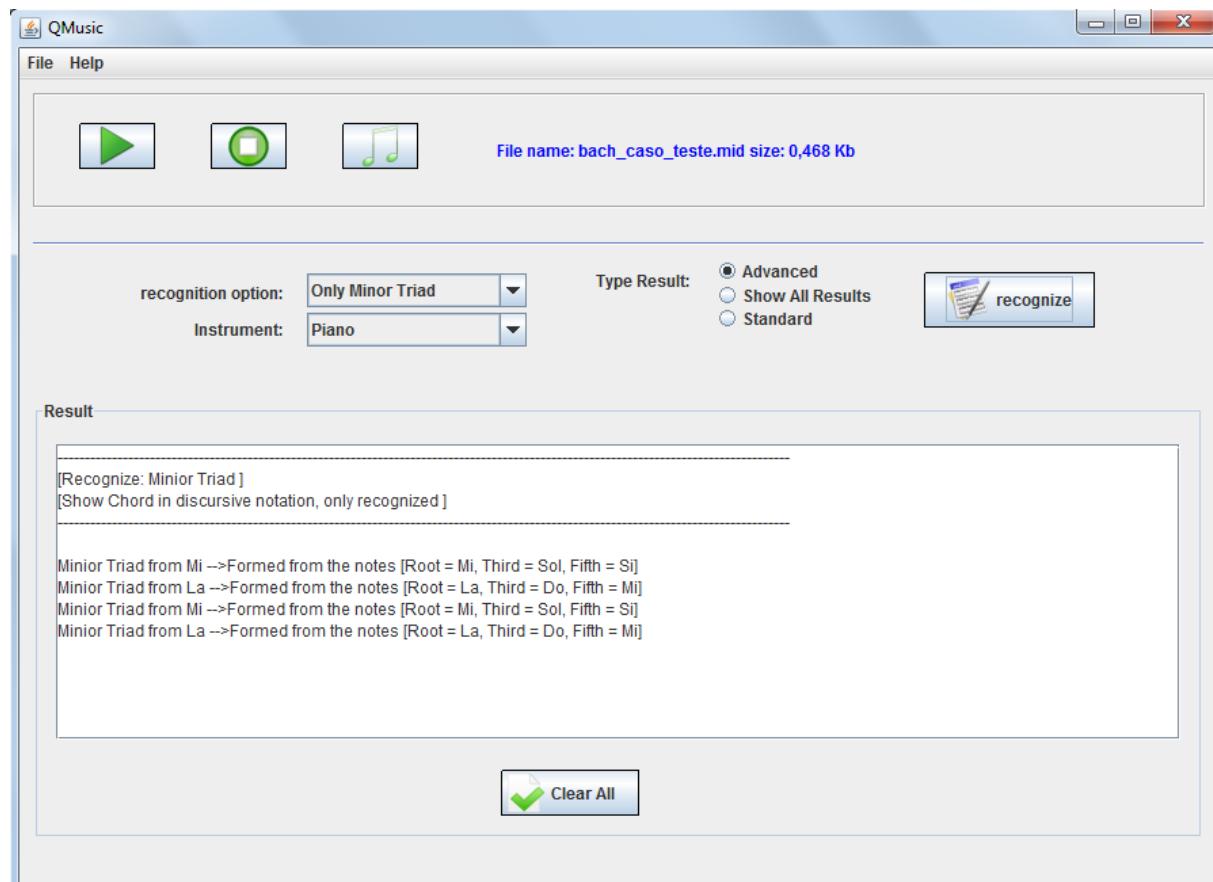


Figura 49 – Lista de Tríades Menores da Seção A do coral *Aus Meines Herzens Grunde* de Bach

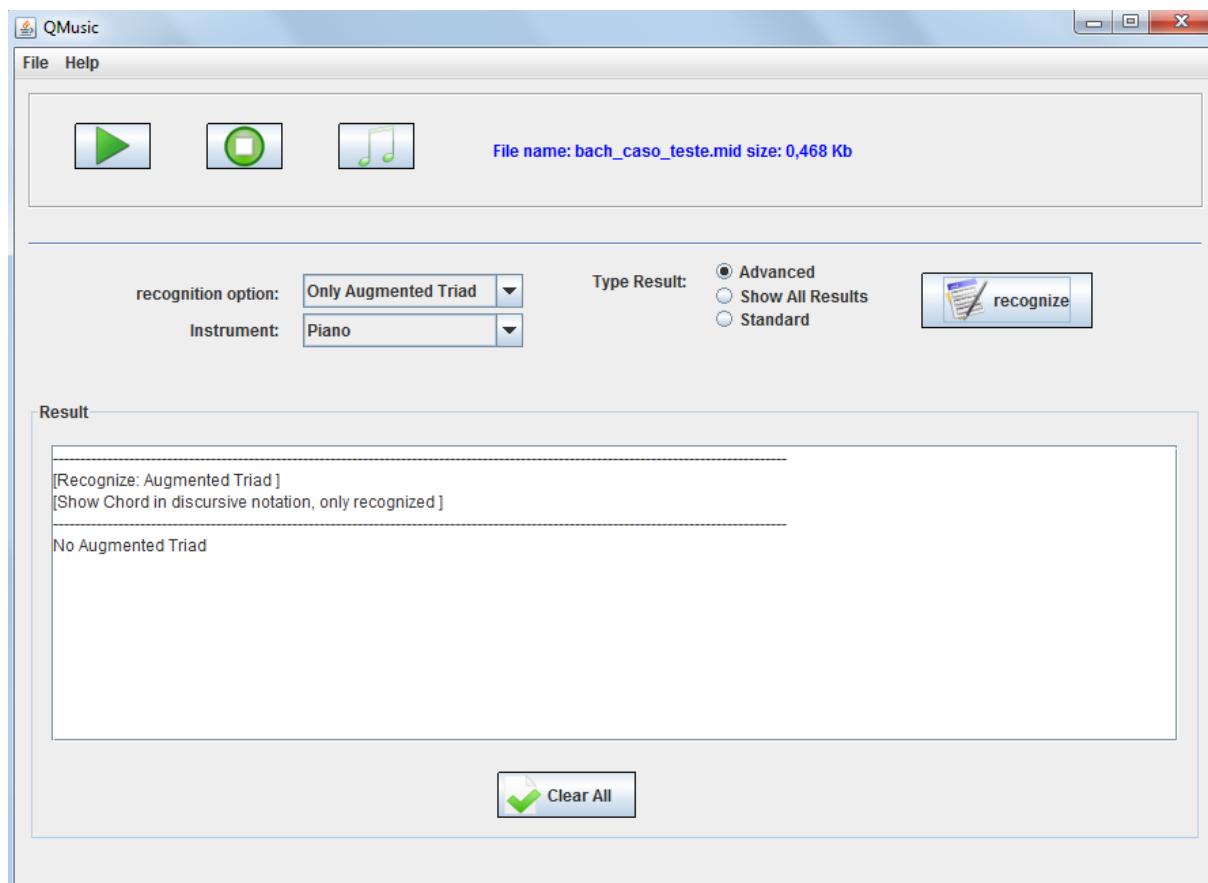


Figura 50 – Lista de Tríades Aumentadas da Seção A do coral *Aus Meines Herzens Grunde* de Bach

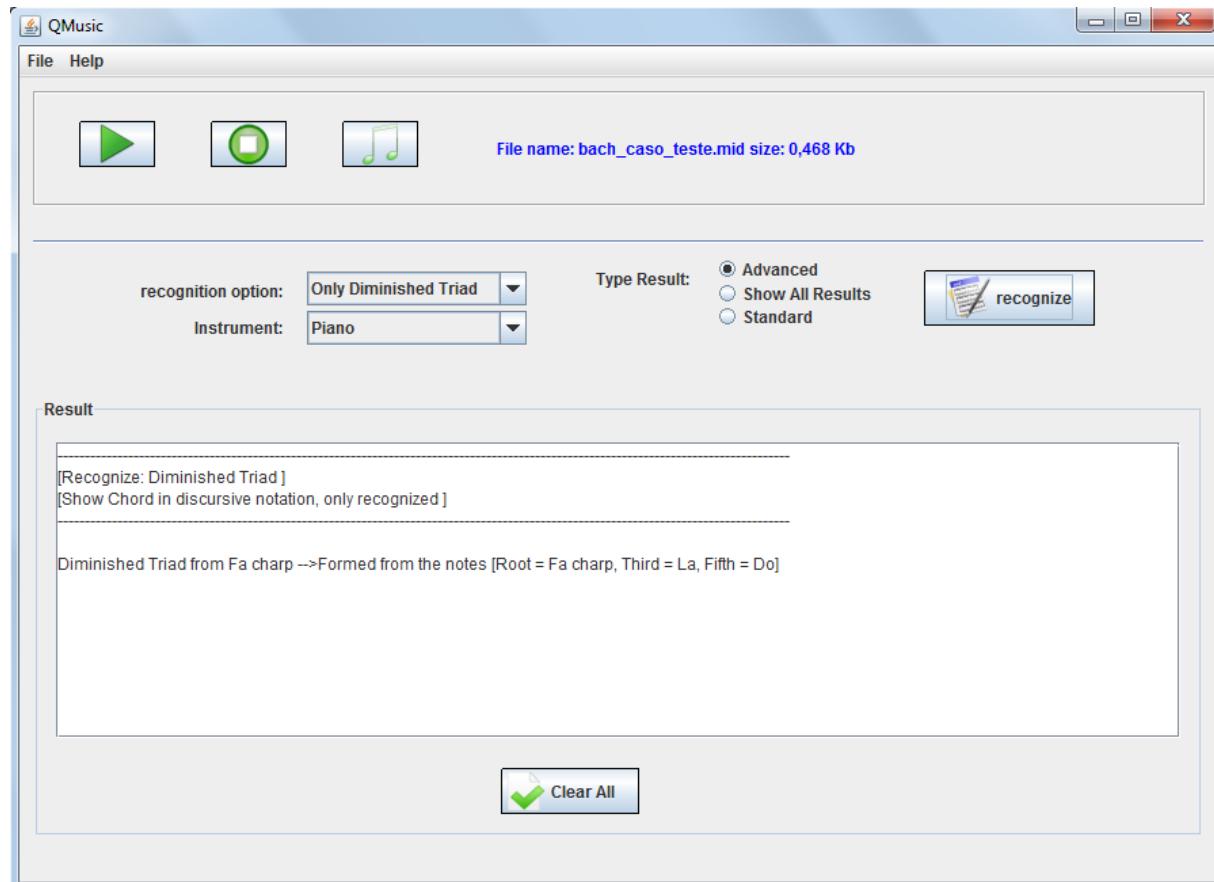


Figura 51 – Lista de Tríades Diminutas da Seção A do coral *Aus Meines Herzens Grunde* de Bach

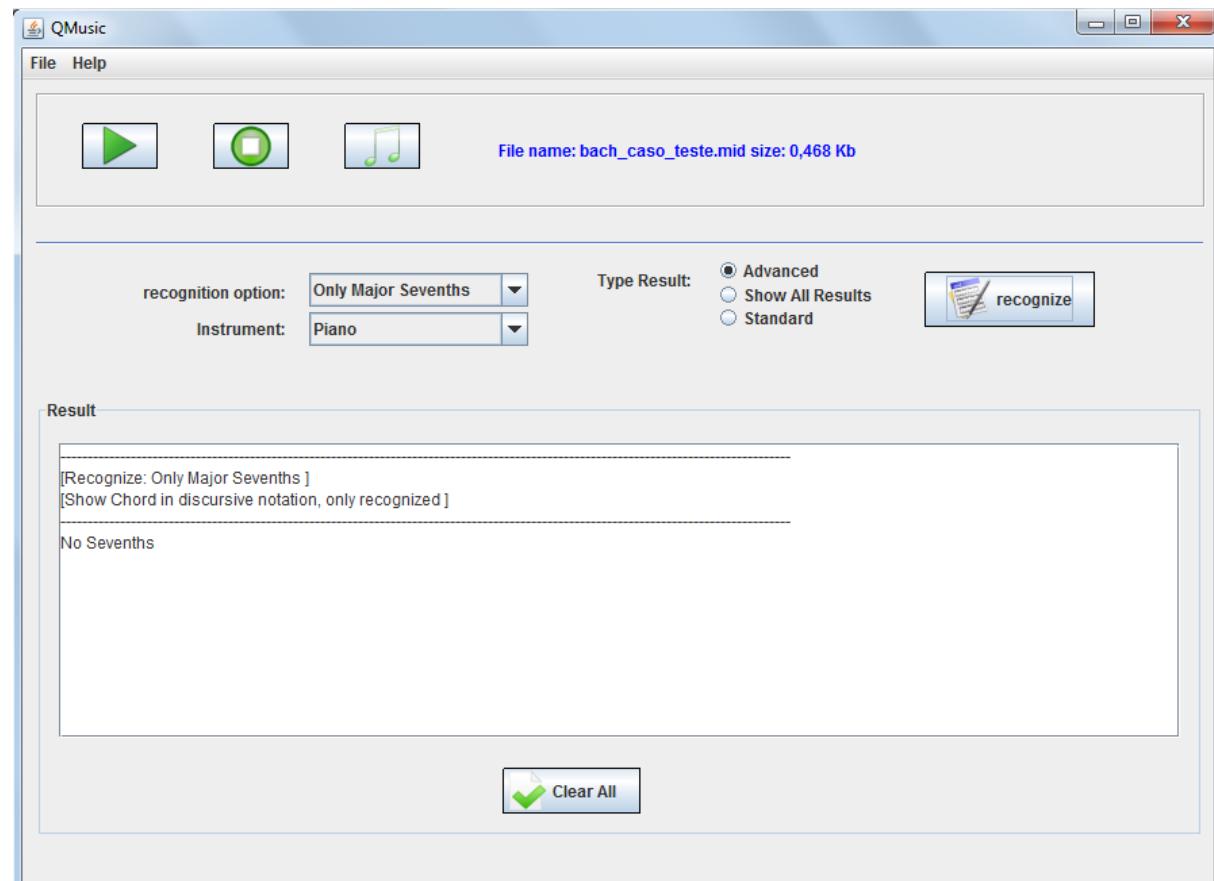


Figura 52 – Lista de Tétrade Maiores da Seção A do coral *Aus Meines Herzens Grunde* de Bach

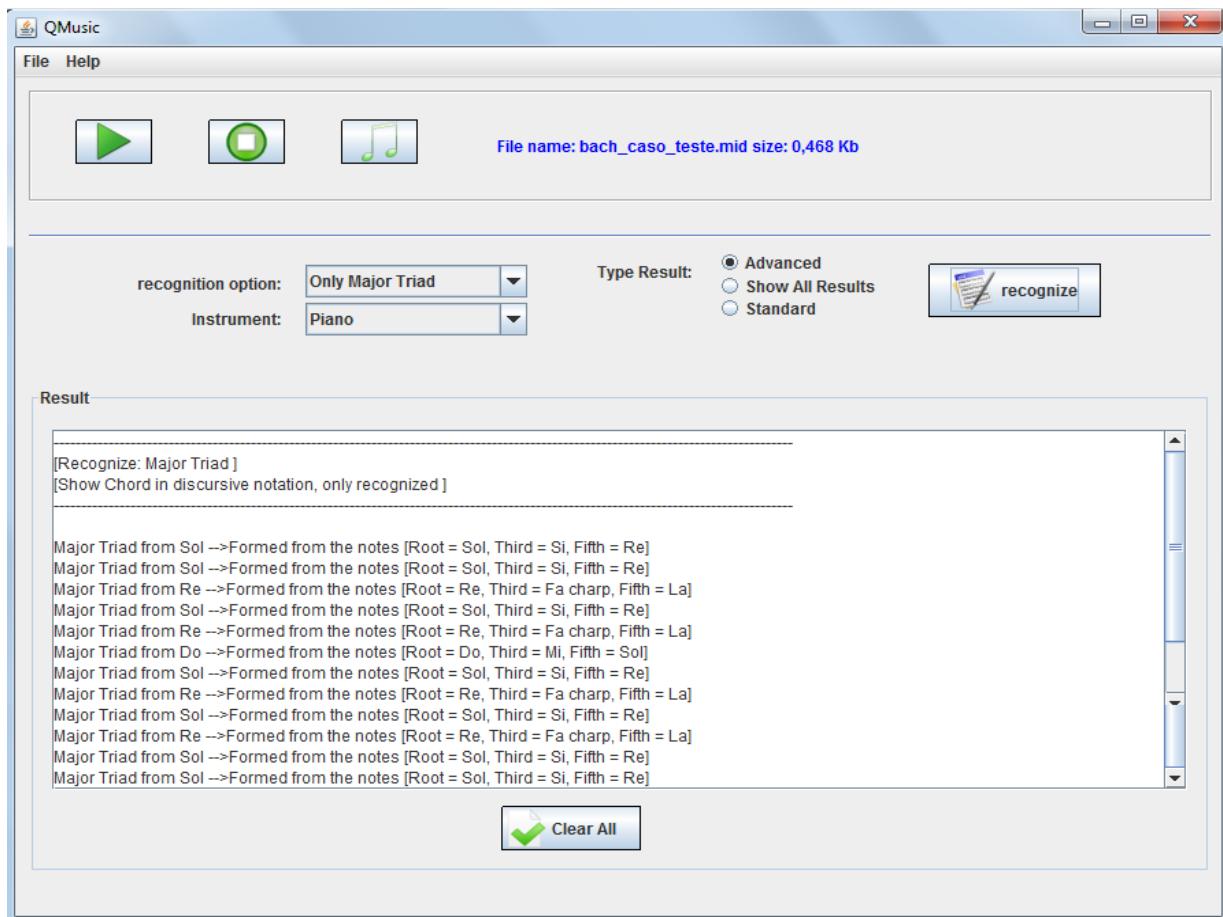


Figura 53 – Lista de Tríades Maiores da Seção A do coral *Aus Meines Herzens Grunde* de Bach

Como não houve nenhum problema durante o processo de reconhecimento, as telas do modo *show all results*, são as mesmas do modo *advanced*. Portanto, não há razão para listá-las.