
Luciênio de Macêdo Teixeira

**Da representação do conhecimento
musical ao esboço conceitual de uma
sociedade de agentes em Harmonia**

Campina Grande - PB
setembro de 1997

Ficha catalográfica

Teixeira, L de M. - Da representação do conhecimento musical ao esboço conceitual de uma sociedade de agentes em Harmonia. Campina Grande: UFPB/CCT, 1997.
118p.

1. Computação e Música. 2. Informática na Educação. 3. Representação de Conhecimento. 4. Teoria Musical. I Título.

CDD 001.642

Luciênio de Macêdo Teixeira

Da representação do conhecimento musical ao esboço conceitual de uma sociedade de agentes em Harmonia

Dissertação apresentada ao Curso de Mestrado em Informática da Universidade Federal da Paraíba, como exigência parcial para a obtenção de Grau de Mestre.

Área de Concentração: Ciência da Computação

Orientadores: Prof. Edilson Fernalda e
Prof. Evandro de Barros Costa

Campina Grande - PB
1997

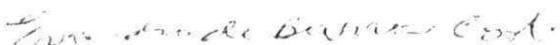
**DA REPRESENTAÇÃO DO CONHECIMENTO MUSICAL AO ESBOÇO
CONCEITUAL DE UMA SOCIEDADE DE AGENTES EM HARMONIA**

LUCIÊNIO DE MACÊDO TEIXEIRA

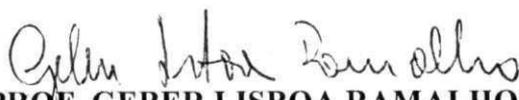
DISSERTAÇÃO APROVADA EM 19.09.1997



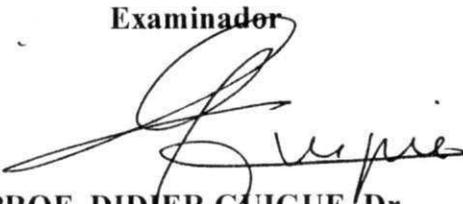
PROF. EDILSON FERNEDA, Dr.
Presidente



PROF. EVANDRO DE BARROS COSTA, D.Sc
Examinador



PROF. GEBER LISBOA RAMALHO, Dr.
Examinador



PROF. DIDIER GUIGUE, Dr.
Examinador



PROF. BERNARDO LULA JUNIOR, Dr.
Examinador

CAMPINA GRANDE - PB

A minha amiga, companheira e esposa:

Mônica

Agradecimentos

A todos os que contribuíram para meu crescimento nestes anos de aprendizado; aos que confiaram em mim e aos que duvidaram também ...

Em especial aos meus orientadores, Edilson e Evandro, pela paciência e boa vontade; ao Starch pelas horas de aprimoramento na programação; à Mônica pelos incentivos e aos professores pela confiança.

Por fim, aos que indiretamente se envolveram, de todas as formas possíveis.

A todos vocês minha gratidão!

Sumário

1	Introdução	1
1.1	Motivações	1
1.2	Objetivos	2
1.3	Organização do Trabalho	3
2	Elementos de Música	5
2.1	Introdução	5
2.2	Conceitos Musicais	6
2.2.1	Nota Musical	7
2.2.2	Semitom e Tom	10
2.2.3	Intervalos	11
2.2.4	Escalas, Acordes e Cifras	13
2.2.5	Tonalidades	16
2.3	Harmonia	16
2.3.1	Pequeno Histórico	18
2.3.2	Harmonia Funcional	20
2.3.3	Harmonia Normativa (Tradicional)	21
2.3.4	Aprendizagem e Didática em Harmonia	22
2.3.5	Princípio de Condução de Vozes (Regras)	24
3	Computação e Música	28
3.1	Introdução	28
3.2	Histórico	30
3.3	Música e Inteligência Artificial	38
3.3.1	Aspectos de Representação em Música	39
3.3.2	Atividades em IA e Música	40
3.3.3	Problemáticas em IA e Música	41
3.4	Sistemas Baseados em Conhecimento dedicados à Música	44
4	Trabalhos Correlatos	46
4.1	Introdução	46
4.2	Expert Piano	47
4.3	Classificação Harmônica Tonal	48
4.4	SETMUS	50

4.5 MusES	51
4.6 Considerações Finais	52
5 Representação do Conhecimento Musical	53
5.1 Introdução	53
5.2 Aspectos sobre a Representação de Objetos Musicais	53
5.3 Estrutura da Representação	56
5.4 Representação dos Conceitos Fundamentais	61
5.4.1 Notas e Pausas	63
5.4.2 Tom e Semitom	68
5.4.3 Intervalo	70
5.4.4 Escala	74
5.4.5 Acorde	77
5.4.6 Tonalidade	78
5.5 Representação de Vozes	83
6 O Ambiente MATHEMA	87
6.1 Introdução	87
6.2 O Ambiente de Trabalho	88
6.3 Dimensões do Conhecimento	89
6.4 Dimensões do Conhecimento em Harmonia	91
6.5 Interação Aprendiz × Tutor	94
6.6 Ilustração da interação entre o Aprendiz e o Tutor	95
7 Conclusão e Trabalhos Futuros	98
Referências Bibliográficas	100
Anexo 1 - Taxionomia em Computação e Música	106
Anexo 2 - Código em LPA-Prolog++	109

Lista de Figuras

Figura 2.1: objetos musicais em pentagrama	06
Figura 2.2: teclado e oitavas	08
Figura 2.3: notas naturais e alteradas	09
Figura 2.4: representação gráfica e numérica de figuras de tempo	10
Figura 2.5: graus de intervalo	12
Figura 2.6: tratamento polifônico	18
Figura 2.7: tratamento harmônico	18
Figura 2.8: exemplo de cifra (harmonia funcional)	21
Figura 2.9: exemplo de cifra (harmonia tradicional)	22
Figura 3.1: gráfico de um envelope de som	33
Figura 5.1: atributos de uma nota	57
Figura 5.2: estruturação dos objetos musicais relevantes à Harmonia	58
Figura 5.3: relacionamento entre os objetos Tonalidade, Acordes e Escalas	60
Figura 5.4: objeto encadeamento	61
Figura 5.5: diagrama de comunicação entre métodos de classes diferentes	62
Figura 5.6: valor numérico para nome de notas	64
Figura 5.7: trecho musical, sua representação e valores numéricos	68
Figura 5.8: valor numérico para notas e suas alterações	70
Figura 6.1: arquitetura simplificada do MATHEMA	88
Figura 6.2: visão interna de um agente d_i	90
Figura 6.3: dimensões do conhecimento	91
Figura 6.4: exemplos de acordes em dó maior	93
Figura 6.5: contra-exemplos de tríades que não formam acordes	93
Figura 6.6: interação entre Aprendiz e Tutor	94
Figura 6.5: problema a ser resolvido	96

Lista de Tabelas

Tabela 2.1: nomes dos intervalos e seus graus	12
Tabela 2.2: inversão de intervalos	13
Tabela 2.3: distância entre os graus de uma escala diatônica	13
Tabela 2.4: distância entre os graus das escala diatônica menores	14
Tabela 2.5: relação entre terças para formação de acordes	15

Resumo

O objetivo deste trabalho é propor um modelo de representação e manipulação do conhecimento musical, que permita a sua utilização de forma coerente em um ambiente computacional, voltado ao ensino de Harmonia. A base para definição de um tal modelo é, por um lado, uma representação proposta segundo a idéia de orientação a objetos e programação em lógica, e por outro lado, é um modelo de organização e representação do conhecimento de forma multidimensional proposto no ambiente MATHEMA.

A motivação inicial para tal investimento foi a observada carência de representações do conhecimento musical que contemplem aspectos didáticos no ensino de Harmonia.

De posse do referido modelo de representação, partiu-se para sua adequação segundo uma abordagem de agentes, visando a construção futura de um sistema tutor inteligente baseado em múltiplos agentes. Essa pretensão é apoiada pelo modelo de ambiente interativo de aprendizagem definido no MATHEMA.

Abstract

The objective of this work is to propose a musical knowledge representation and manipulation model that can be used in a coherent way in a computational environment to support Harmony teaching. The basis for the definition of such model is an object oriented and logic programming representation. It also bases its knowledge organisation and representation in a multidimensional form proposed in the MATHEMA environment.

The initial motivation of this work was the need for a musical knowledge representation that deals with didactic aspects in Harmony teaching.

This representation model uses an agent approach for the construction of an intelligent tutoring system based on multiple agents. This statement is supported by the interactive learning environment model defined in the MATHEMA.

1 Introdução

1.1 Motivações

Desde a década de 50, Música e Computação formam um “dueto”. O resultado desse encontro é o aumento nas pesquisas e no desenvolvimento de produtos voltados à tecnologia musical. Uma das razões para isto é o fascínio que a Música exerce, não apenas pelo seu lado artístico (estética, reprodução de fatos sociais), mas também por seu aspecto científico (formas, frequências, harmonias, concatenação de idéias, etc).

Muitas das áreas ligadas à Música têm despertado o interesse de pesquisadores de diversas áreas de conhecimento. Na Computação isso não poderia ser diferente. Em sua marcha inexorável rumo a uma integração cada vez maior com outras ciências, a Computação, no Brasil em particular, viu surgir, ainda com uma certa timidez, seus primeiros grupos de pesquisadores em Computação e Música.

E não faz tanto tempo assim. Um ano após a criação, em 1993, do NUCOM - Núcleo Brasileiro de Computação e Música -, realizou-se o *I Simpósio Brasileiro de Computação em Música (SBCM)*, dentro do *XIV Congresso da Sociedade Brasileira de Computação (SBC)*. Desde então, o SBCM passou a ser um dos eventos anuais promovidos pela SBC. Hoje, o NUCOM está constituído como Comissão Especial da Sociedade Brasileira de Computação.

Atualmente, diversas atividades acontecem em Computação e Música (não apenas no Brasil, mas em termos mundiais), seja com o surgimento de laboratórios dedicados, cursos de pós-graduação, pesquisas financiadas, simpósios, *workshops*, etc. Uma das consequências imediatas de algumas dessas ações e também do aumento da massa crítica no tema, foi a ampliação da visão musical tendo como ponto de referência o seu formalismo intrínseco. Com isto surgiram, por exemplo, vários sistemas informatizados dedicados à *composição musical* (arte “formal” por meios eletrônicos) e, posteriormente, o próprio meio de expressão da Música foi sendo alvo de diversos tipos de estudos (síntese sonora), gerando resultados cada vez mais satisfatórios.

E não parou por aí. O avanço tecnológico não trouxe apenas alternativas na maneira de manifestação da arte musical. Outro ponto de importância foi o surgimento de ferramentas que promovem cada vez mais a aproximação do ser humano com a Música.

Ferramentas voltadas ao armazenamento, à reprodução sonora, à composição, à síntese, etc, trouxeram facilidades antes não encontradas. Pode-se destacar, entre as pesquisas em Computação e Música, duas áreas: *análise* e *ensino musical*. Em *análise musical*, o motivador é o fato de ser este um campo em constante evolução e que envolve aspectos não apenas formais mas também perceptivos. No *ensino*, tenta-se prover os interessados na área com diversas ferramentas instigadoras à aprendizagem, contando com contribuições de ciências tais como a Psicologia, a Didática, a Ciência da Computação, etc.

É no ensino de Música, mais particularmente no ensino de Harmonia, que este trabalho está inserido. A carência de representações do conhecimento musical que contemplem aspectos didáticos no ensino de Harmonia foi a força motivadora. Embora alguns trabalhos tenham surgido com o intuito de ocupar-se com a Harmonia - em sua grande maioria tendo como primeiro plano as áreas de *Análise Musical* ou *Composição* - todos acabaram por não explorar adequadamente alguns aspectos teóricos relevantes ao ensino propriamente dito, por exemplo a conceituação de acordes e intervalos em todos seus aspectos.

Neste ponto, o trabalho aqui proposto vem preencher uma lacuna: nos sistemas existentes aqui avaliados, a maioria ignora, por exemplo, o conceito de enarmonia e de intervalos enarmônicos. Sem definição de enarmonia não se pode transmitir, em sua totalidade, os conceitos formais da Harmonia.

1.2 Objetivos

O objetivo deste trabalho é propor um modelo de representação e manipulação do conhecimento musical, adequado para ser utilizado de forma coerente em um ambiente computacional, voltado ao ensino de Harmonia. Para isto, deve-se representar os objetos musicais necessários ao domínio, as relações relevantes entre estes objetos e também a maneira de, respeitando os diversos contextos que são encontrados na Música, estabelecer limites às opções as quais os alunos, de um modo geral, se defrontam durante o processo de aprendizagem.

Nesta perspectiva, é definido um modelo de representação do conhecimento musical baseado na união dos paradigmas de Programação em Lógica e Orientação a Objetos. Este modelo possui como características:

- a definição dos objetos musicais e dos métodos responsáveis pela manipulação e formação destes,
- a partir dos objetos musicais definidos, estruturá-los dentro dos conceitos de classe, herança, atributo e mensagem entre instâncias,
- encontrar respostas completas a partir de objetos musicais incompletos,
- possibilitar o tratamento do conhecimento em profundidade, onde as estruturas simples formam estruturas mais complexas,
- tratar adequadamente a questão das enarmonias,
- permitir a manipulação do conhecimento de acordo com a necessidade imposta pela relação tutor-aprendiz.

Partindo da proposta de representação do conhecimento musical, será apresentada uma visão de organização deste conhecimento, utilizando como perspectiva a sua futura disponibilização em um Sistema Tutorial Inteligente a ser desenvolvido para o domínio em questão. Dessa forma, o modelo de ambiente MATHEMA, proposto por [COSTA, 1995, 1996] foi utilizado como alternativa para a modelagem do domínio de conhecimento, tendo como metas os seguintes pontos:

- (i) a organização do conhecimento utilizando como paradigma a sua visão multidimensional e
- (ii) a posterior utilização de agentes tutores, responsáveis por ensinar conhecimentos específicos do domínio.

1.3 Organização do Trabalho

Basicamente, pode-se dividir esta dissertação em três partes:

- (i) a que trata das informações preliminares ligadas aos conceitos musicais e à ligação entre estes e a Computação;
- (ii) a que trata especificamente da representação do conhecimento musical aqui proposta e sua relação com outras representações,
- (iii) e por fim a exposição conceitual de uma proposta de direção para alcançar a construção de um Sistema Tutorial Inteligente aplicado ao ensino-aprendizagem de Harmonia Musical.

A primeira parte compreende os capítulos 2 e 3, sendo a segunda parte composta pelos capítulos 4 e 5, finalizando a terceira parte com o capítulo 6.

Inicialmente, no capítulo 2, são descritos os conceitos musicais envolvidos no domínio de Harmonia. Estes conceitos contemplam algumas das primitivas (frequência e duração) do universo musical e incluem também os elementos mais complexos que são formados a partir dessas mesmas primitivas.

Ainda neste capítulo, é feita uma breve descrição histórica da Harmonia, com suas duas principais correntes de ensino (Normativa e Funcional), suas questões didáticas e suas estratégias para a resolução de problemas harmônicos.

No capítulo 3, um pequeno relato histórico é feito, compreendendo os diversos períodos pelos quais passaram as pesquisas em Computação e Música, além de alguns aspectos do envolvimento mais específico da Música com pesquisas na área de Inteligência Artificial.

O capítulo 4 trata de alguns sistemas que possuem relação mais próxima com o presente trabalho. Aqui são analisados quatro sistemas que possuem ligação com o ensino em Música. Dos sistemas escolhidos, um é dedicado ao ensino de performance (piano), outro ao ensino de teoria, e os três últimos estão relacionados à Harmonia, tratando de análise harmônica, treinamento de intervalos e representação voltada à Harmonia Tonal.

Em seguida, no capítulo 5, o enfoque é dado sobre a *Representação do Conhecimento Musical*, constituindo o capítulo central. Como estratégia para este capítulo, são feitas algumas considerações sobre a representação do conhecimento musical formal. Em seguida, é apresentada a estrutura geral dos conceitos musicais (objetos) e seus relacionamentos. O próximo passo é a definição formal dessas estruturas e objetos, constituindo assim a representação dos conceitos fundamentais. Finalizando o capítulo, faz-se uma adequação/integração dessa representação ao modelo filosófico-computacional do conhecimento, definido no âmbito do MATHEMA para dar suporte a uma visão multidimensional de um domínio de aplicação.

O capítulo 6 é a descrição sucinta do MATHEMA como um ambiente interativo de apoio ao ensino e aprendizagem sendo, neste contexto, orientado à construção de um Sistema Tutorial Inteligente (STI) no domínio da Harmonia. Este modelo, juntamente com a representação apresentada no capítulo 5, será utilizado para dar suporte ao processo de interação didática entre um Tutor (máquina) e um Aprendiz (humano).

2 Elementos de Música

2.1 Introdução

A Música pode ser definida de várias maneiras. Evidentemente que, de acordo com o objetivo que se deseja, algumas definições se adaptam melhor a determinadas situações. Para o objetivo deste trabalho, a Música pode ser definida da seguinte maneira:

A Arte de criação sonora, com intuito de demonstrar os sentimentos de um artista e de acordo com certos preceitos formais (Estética).

Existe uma linha separadora, muito tênue, entre ciência e arte em Música. A maneira como um músico interpreta uma obra musical em seu instrumento é complementada por uma bagagem histórica e técnica que traz à tona o *porquê* (motivação criadora), o *como* (técnicas de interpretação) e o *quando* (contextualização histórico-social) desta obra.

Os conceitos dos objetos musicais descritos neste capítulo formarão o conhecimento musical a ser representado no capítulo 5. Serão apresentadas também as regras de condução de vozes que, muito embora não estejam no escopo da representação do conhecimento aqui considerado, algumas delas serão utilizadas no capítulo 6, quando serão apresentados exemplos de interações didáticas entre um Aprendiz e um Tutor.

Ao examinar uma partitura¹, pode-se catalogar todos os seus elementos formais. Estes elementos são a matéria-prima com a qual a maioria dos teóricos trabalham. Neste capítulo são apresentados os conceitos musicais relevantes para este trabalho. Aqui, cabe ainda ressaltar as estruturas musicais consideradas complexas (como acordes, escalas e tonalidades), não só por suas constituições (formada por estruturas mais simples) mas também pelo inter-relacionamento destas estruturas dentro de determinados contextos. Por exemplo, as diversas funções (tônica, dominante, relativa da dominante, etc) que um mesmo acorde pode assumir em tonalidades diferentes.

Deve-se buscar a abstração dos conceitos de forma a, quando necessário, determinar os aspectos relevantes de uma estrutura musical, de acordo com um contexto definido e, a

¹ Uma partitura é o conjunto de todas as partes de uma obra musical, dispostas verticalmente, de forma a permitir a visualização das vozes e da harmonia resultante [PÉREZ, 1985].

partir disto, resolver questões de aparente conflito. Por exemplo: pode-se dizer que, em um certo período da História da Música, eram realizadas determinadas práticas que, com o passar do tempo, foram dando lugar a outras. A consequência prática disto é que uma assertiva só pode ser considerada correta ou errada se devidamente enquadrada.

2.2 Conceitos Musicais

Quem observar uma partitura irá encontrar muitos tipos de símbolos e, similarmente a uma língua viva qualquer, sempre estará a encontrar novos símbolos. Para os que não estão familiarizados com toda esta gama de sinais, existem vários livros que se dedicam ao ensino de teoria musical, como [PRIOLLI, 1985; CHEDIK, 1986; HINDEMITH, 1988], entre outros.

A área de trabalho para quem se ocupa da Música é o pentagrama²; é sobre o pentagrama que se escreve uma partitura. Como o nome diz, este é um conjunto de cinco linhas onde são postas as notas e pausas. Um exemplo de uma partitura pode ser visto na Figura 2.1, onde são indicados os elementos de interesse para este trabalho.



Figura 2.1: objetos musicais em pentagrama

Apesar do número de símbolos musicais estar em constante processo de expansão, pode-se classificar os atributos básicos da Música em dois: *silêncio* e *som*. Ambos, em uma partitura, são representados graficamente. Os símbolos utilizados nesta representação sugerem não apenas onde existe silêncio ou som, tendo como base uma “linha” temporal imaginária, mas inclui também informações sobre ritmo e natureza do som.

O símbolo de som é denominado *nota* e o símbolo do silêncio é denominado *pausa*. Intrinsecamente, cada um destes símbolos traz uma série de atributos necessários ao som.

² Conjunto de cinco linhas e quatro espaços (também conhecida como pauta) sobre os quais se escrevem as figuras musicais (notas, pausas, símbolos, etc). Um pentagrama possui no seu início, conforme a Figura 2.1, uma clave (que determina o nome das notas), uma armadura (indica a tonalidade) e um compasso (métrica) definidos respectivamente pelos três primeiros símbolos [PÉREZ, 1985].

Portanto, pode-se dizer que notas e pausas, em analogia à Física, constituem os “átomos” da Música. Com um conjunto ordenado destes “átomos”, podem-se formar diversas estruturas musicais. Abaixo seguem as que são relevantes a este trabalho:

- *intervalos* são formados quando da existência de duas notas. A particularidade dos intervalos encontra-se em sua natureza: duas notas subseqüentes determinam intervalos melódicos (horizontais), enquanto notas concomitantes determinam intervalos harmônicos (verticais).
- *escalas* são conjuntos de intervalos, dispostos melodicamente (sons consecutivos) dentro de determinados padrões intervalares (distâncias). Dentre as diversas classes de escalas (pentatônicas, modais, menores melódicas, etc), apenas duas serão consideradas neste trabalho: escalas *maiores* e escalas *menores*, sendo esta última apenas em sua forma *harmônica*. Estas diferenças serão explicitadas no item 2.2.4.
- *acordes* são conjuntos de intervalos verticais (três ou mais notas), dispostos segundo as leis da Harmonia. Os acordes podem ser consonantes ou dissonantes, maiores ou menores, tríades, tétrades, de sétima, etc. Aqui, serão considerados apenas os acordes *maiores*, *menores*, *diminutos*, *aumentados* (estes constituídos por três notas) e *de sétima* (constituído por quatro notas). Tais conceitos serão apresentados no item 2.2.4.

Continuando a comparação entre Música e Física, assim como os átomos possuem partículas menores, as notas reúnem em si vários outros conceitos musicais. A seguir serão descritos, com mais detalhes, estes conceitos.

2.2.1 Nota Musical

Para a definição de uma nota, faz-se necessário a noção de quatro atributos musicais básicos. São eles: *nome*, *oitava*, *alteração* e *duração* (sendo que os três primeiros definem a sua frequência propriamente dita).

Nome

Os nomes das sete notas musicais naturais são: *dó, ré, mi, fá, sol, lá e si*. Um outro tipo de convenção se utiliza de letras para representar as notas musicais. Este tipo de notação é chamada de notação alfabética e remonta aos gregos³. Esta é sua equivalência:

dó (c), ré (d), mi (e), fá (f), sol (g), lá (a), si (b)

Oitava

Oitava é um símbolo de execução que resulta em dobrar (oitava alta) ou dividir (oitava baixa) a frequência de uma determinada nota. Na Figura 2.2, pode-se ver a disposição das oitavas em um teclado de piano. O asterisco indica, no teclado, o início de cada oitava (nota dó).



Figura 2.2: teclado e oitavas

Alteração

Alterações são símbolos utilizados para elevar ou abaixar uma nota, um *semitom* ou um *tom*⁴ a partir de sua afinação original. O sustenido (#) eleva um semitom, o dobrado-sustenido (x) que eleva um tom, o bemol (b) abaixa um semitom e o dobrado-bemol que abaixa um tom (bb). A Figura 2.3 mostra a disposição das notas originais e de suas alterações (em geral nas teclas pretas).

Em determinados casos as notas podem vir acompanhadas de uma dessas alterações, se o desejo for o de retornar a nota ao seu estado original (teclas brancas), utiliza-se o símbolo do bequadro (h).

Como consequência imediata da colocação das alterações, o número possível de notas passa de sete para trinta e cinco, sendo sete naturais, sete sustenizadas, sete bemolizadas, sete com dobrado-sustenido e sete com dobrado-bemol. Na Figura 2.3 é visto que algumas teclas são referidas com mais de um nome (são doze teclas para trinta e cinco no-

³ Atualmente este tipo de notação é utilizado nos países anglosaxões (seu outro nome é sistema germânico de notação), em tratados de física. Sua difusão foi bastante acentuada por sua utilização na música popular (cifras), como forma de abreviar e economizar espaço [PÉREZ, 1985].

⁴ Estes conceitos serão explicados no item 2.2.2.

mes), isto se deve ao que se chama de temperamento da escala musical⁵. Este temperamento gerou o conceito de *enarmonia*.

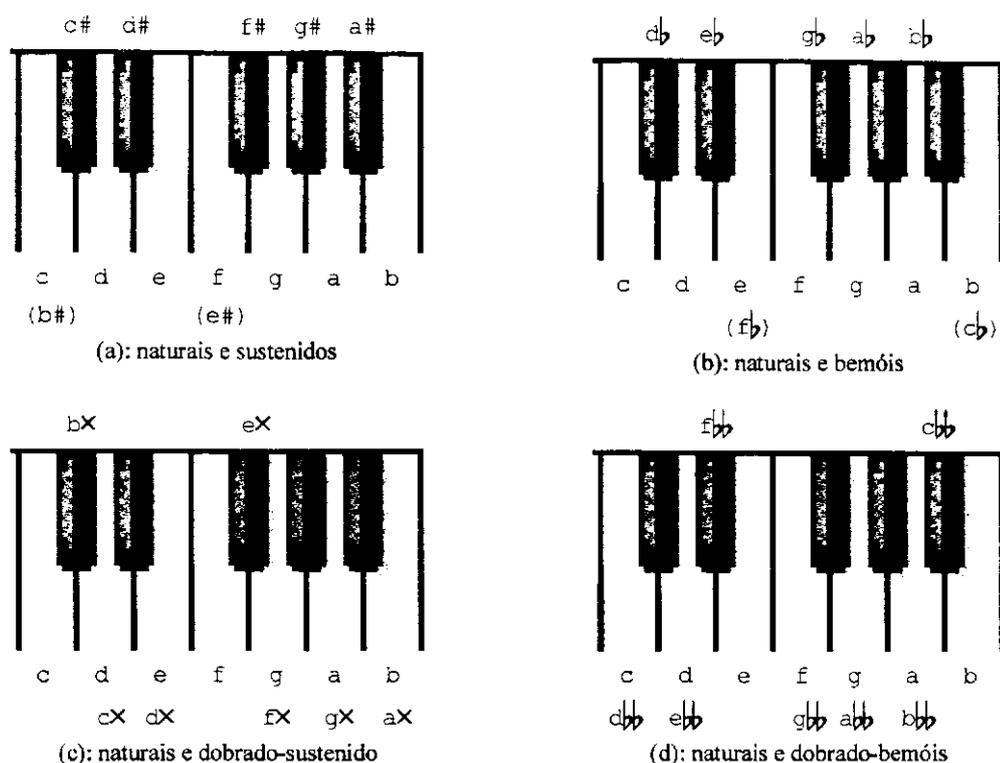


Figura 2.3: notas naturais e alteradas

Enarmonia

Atualmente, os sons ou intervalos enarmônicos são os que, embora tenham nomes diferentes, são iguais em altura (frequência) dentro do sistema temperado. Em alguns instrumentos desprovidos de teclado, o sustenido é mais elevado (frequência um pouco mais alta) que o bemol [PÉREZ, 1985].

Duração

Em Música, a duração das notas obedece uma proporcionalidade e uma relatividade entre os símbolos que representam o tempo. Não se pode afirmar categoricamente que uma determinada figura musical possua um valor de tempo que seja imutável. O parâmetro para este tipo de determinação é a fração colocada no início de um pentagrama⁶ (Figura

⁵ Maiores detalhes poderam ser vistos no subitem 2.2.4.

⁶ O conceito de *métrica* está intimamente relacionado com o conceito de compasso que é uma divisão rítmica regular, representada graficamente por barras verticais (Figura 2.1) e esta vinculado com a fração do início do pentagrama.

2.1). O numerador diz quantos tempos possui a métrica da música (por exemplo: 1, 2, 3, 1, 2, 3, ... ou 1, 2, 3, 4, 1, 2, 2, 3, 4, ..., etc). Já o denominador especifica qual a figura musical que equivale à pulsação da música. A Figura 2.4 mostra a numeração de cada figura. Uma figura logo abaixo da utilizada como referencial terá a metade da duração da anterior e a que lhe antecede, o dobro.

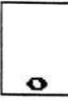
	semibreve	=	1
	mínima	=	2
	semínima	=	4
	colcheia	=	8
	semicolcheia	=	16
	fusa	=	32
	semifusa	=	64

Figura 2.4: representação gráfica e numérica de figuras de tempo

Todas estas informações estão contidas em qualquer partitura musical. A *duração* e o *ritmo* dependem da figura de tempo utilizada. A distribuição espacial (sentido horizontal) destas figuras, sobre um pentagrama, indica o ordem cronológica dos acontecimentos. Já a altura das notas (frequências) está relacionada com a disposição destas no sentido vertical. Quanto mais em cima, em relação ao pentagrama, mais aumenta a frequência, o contrário também é verdadeiro: se a nota é colocada mais para baixo, diminui a frequência.

2.2.2 Semitom e Tom

Estes dois conceitos e os próximos a serem apresentados, passam a ser obtidos a partir da consideração de duas notas subseqüentes. Estes conceitos trabalham com duas ou mais notas (intervalo) e levam em consideração a distância entre elas (distância entre frequências).

Semitom

Um semitom, tendo como base um teclado de piano, é a distância entre teclas vizinhas, sejam estas brancas ou pretas. Na Figuras 2.3, pode-se ver alguns exemplos de semitom: entre $c-c\#$, entre $e-f$, etc. No caso do intervalo ($c-c\#$), este tipo de semitom é chamado de *semitom cromático*, por ocorrer entre notas com o mesmo nome (c , no caso) sendo que uma ou ambas devem possuir uma alteração. Se ocorrer entre notas de nomes diferentes ($e-f$) é chamado *semitom diatônico*.

Tom

Para o conceito de tom, pode-se encontrar diversas acepções. De uma maneira prática, pode-se afirmar que tom é a distância entre teclas, formada por dois semitons. Por exemplo: $c-d$, $e-f\#$, etc.

2.2.3 Intervalos

Intervalo

Intervalo é a distância entre duas notas. Quando as notas que compõem o intervalo se encontram no perímetro de uma oitava, tem-se um *intervalo simples* e, quando ultrapassa uma oitava, tem-se um *intervalo composto*.

Um intervalo possui como características:

- **natureza (vertical - horizontal)**: se o intervalo ocorrer em uma melodia, estará caracterizada a natureza horizontal do intervalo (melódico). Caso ocorra entre notas de melodias diferentes e simultâneas, estará caracterizado o intervalo vertical (harmônico).
- **direção**: de acordo com a posição da segunda nota, o intervalo pode ser ascendente (quando a segunda nota é mais aguda que a primeira), ou descendente (quando a segunda nota é mais grave que a primeira).
- **grau**: o grau do intervalo é medido pela distância que separa dois nomes de notas dentro da lista (c, d, e, f, g, a, b^7). A Figura 2.5 exemplifica todos os graus de um intervalo simples tendo como base a nota c^8 .

⁷ Não esquecer que, por existirem várias oitavas, esta lista se repete em quantas forem as oitavas consideradas.

⁸ Os intervalos acontecem entre quaisquer nomes, a sua contagem deve incluir o primeiro nome, os nomes compreendidos entre este e o segundo nome. O resultado é o nome do grau.

- **Número de Semitons:** dependendo do grau do intervalo, o número de semitons irá estabelecer o restante do nome do intervalo (diminutos, menores, maiores, aumentados e justos). A Tabela 2.1 dá a nomeação dos graus.

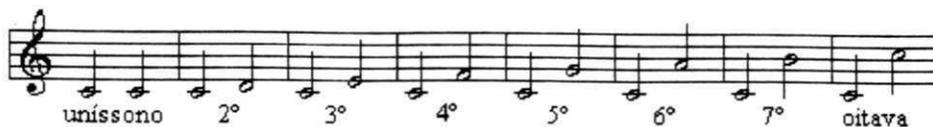


Figura 2.5: graus de intervalo

Tabela 2.1: nomes dos intervalos e seus graus

grau	nº de semitons				
	DIMINUTO	MENOR	JUSTO	MAIOR	AUMENTADO
2 ^a	0	1		2	3
3 ^a	2	3		4	5
4 ^a	4		5		6
5 ^a	6		7		8
6 ^a	7	8		9	10
7 ^a	9	10		11	12
8 ^a	11		12		13

Para a definição completa do nome de um intervalo, são necessárias três partes distintas:

- (i) a que diz qual a direção do intervalo, se ascendente ou descendente;
- (ii) a parte composta pelos nomes das notas que compõem o intervalo, o que determina o seu grau; e
- (iii) a que contém o número de semitons entre as notas. Estas partes, unidas, resultam no nome completo do intervalo, que será, por exemplo: intervalo de 2º menor ascendente e assim por diante.

Inversão de intervalos: uma das propriedades de um intervalo diz respeito à sua *inversão*. Inverter um intervalo é o mesmo que deslocar a nota mais grave uma oitava acima. Desta maneira, para um determinado intervalo simples existirá um determinado intervalo invertido “contra-sujeito” (Tabela 2.2).

Da mesma maneira, pode-se dizer, com respeito ao número de semitons entre os intervalos, que: intervalos maiores, quando invertidos, transformam-se em menores e vice-versa, intervalos diminutos transformam-se em aumentados e vice-versa e intervalos justos continuam justos.

Tabela 2.2: inversão de intervalos

Intervalo	Invertido
uníssono	oitava
2 ^a	7 ^a
3 ^a	6 ^a
4 ^a	5 ^a
5 ^a	4 ^a
6 ^a	3 ^a
7 ^a	2 ^a
oitava	uníssono

2.2.4 Escalas, Acordes e Cifras

Escalas

Escala é uma “sucessão ascendente e descendente de vários sons, com intervalos que guardam entre si uma certa relação de tons e semitons” [PÉREZ, 1985]. Existem vários tipos de escalas⁹. Neste trabalho serão consideradas as escalas diatônicas, que correspondem à seqüência natural de tons e semitons. As escalas diatônicas equivalem à sucessão de oito notas obedecendo ao esquema da Tabela 2.3.

Tabela 2.3: distância entre os graus de uma escala diatônica.

Graus	Distância
entre o 1 ^o e 2 ^a graus	distância de um tom
entre o 2 ^a e 3 ^a graus	distância de um tom
entre o 3 ^a e 4 ^a graus	distância de um semitom
entre o 4 ^a e 5 ^a graus	distância de um tom
entre o 5 ^a e 6 ^a graus	distância de um tom
entre o 6 ^a e 7 ^a graus	distância de um tom
entre o 7 ^a e 8 ^a graus	distância de um semitom

Como variações da escala diatônica, têm-se duas seqüências de tons e semitons: modo *maior* e modo *menor*. O modo maior corresponde à Tabela 2.3. O modo menor possui algumas variações possíveis. Na Tabela 2.4 são mostradas as escalas menores em suas for-

⁹ Cabe aqui uma observação: a escala utilizada atualmente na música ocidental é a chamada escala temperada. Esta se divide em 12 semitons iguais, eliminando dessa forma a necessidade de se ter um instrumento afinado com cada uma das escalas possíveis. No temperamento, as 35 notas vistas no subitem 2.2.1, são confinadas em doze dando uma frequência fixa (gerando enarmonias). Na escala acústica natural há diferenças físicas (frequência) entre duas notas com nomes iguais. Por exemplo: um dó sustenido soará uma frequência um pouco mais alta que o ré bemol.

mas natural, melódica e harmônica sendo esta última a mais utilizada no ensino de Harmonia.

Tabela 2.4: distância entre os graus das escala diatônica menores.

Grau	Natural	Harmônica	Melódica
1 ^o e 2 ^a graus	tom	tom	tom
2 ^a e 3 ^a graus	semitom	semitom	semitom
3 ^a e 4 ^a graus	tom	tom	tom
4 ^a e 5 ^a graus	tom	tom	tom
5 ^a e 6 ^a graus	semitom	tom	semitom
6 ^a e 7 ^a graus	tom	tom + semitom	tom
7 ^a e 8 ^a graus	tom	semitom	semitom

Acordes

Um acorde é a emissão simultânea (intervalos verticais) de três ou mais notas, dispostas segundo as leis da Harmonia¹⁰. Os acordes podem ser *consonantes* ou *dissonantes*, *maiores* ou *menores*, *triades*, *tétrades*, *de sétima*, ...

Após toda a revolução provocada pela Música Contemporânea, tem-se hoje um universo bastante amplo de acordes possíveis. Ao se limitar, por exemplo, às primeiras lições da Harmonia, reduz-se o número de acordes de interesse. Em um estágio inicial, apenas os acordes maiores e menores são trabalhados. Posteriormente, passa-se a trabalhar com os acordes de sétima, diminutos, e aumentados.

Em relação à escala, os acordes utilizam cada uma das notas da escala para a formação de um conjunto de acordes da seguinte maneira: cada nota da escala dará nome a um acorde específico (e suas variações), segundo descrição abaixo, formando assim um conjunto de sete nomes possíveis para os acordes.

Similarmente às escalas, os acordes são “montados” segundo determinadas relações intervalares. Para resumir a construção de um acorde pode-se afirmar:

- acordes maiores, menores, diminutos e aumentados são formados com três notas (triades);
- acordes de sétima são formados com quatro notas (tétrade);

¹⁰ Vide item 2.3.

- todo acorde possui uma nota que lhe dá nome (inicialmente trabalha-se com acordes em posição de fundamental, ou seja, a nota que dá nome ao acorde é sempre a mais grave);
- partindo da fundamental (inferior), tem-se uma distância de 3º grau até a segunda nota (chamada de terça do acorde) e de 5º grau até a terceira nota (chamada de quinta do acorde). Por exemplo: partindo da nota *c* tem-se como 3º grau a nota *e* e como 5º grau a nota *g* (terças superpostas), formando o acorde de dó. Tratando-se de acordes de sétima, tem-se uma quarta nota posta a distância de 3º grau da quinta do acorde. Por exemplo, o acorde de dó com sétima terá a nota *b* acrescentada às anteriores.

Uma vez definidas as notas que compõem o acorde, resta definir o tipo do acorde. A Tabela 2.5 fornece os intervalos entre as notas do acorde e a denominação do mesmo.

Tabela 2.5: relação entre terças para formação de acordes.

Tipo	3ª inferior	3ª superior	sétima
maior	maior	menor	-
menor	menor	maior	-
diminutô	menor	menor	-
aumentado	maior	maior	-
com sétima	maior	menor	menor

Um acorde, em determinados momentos, pode aparecer desfigurado: vir desfalcado de uma de suas notas (acorde incompleto), ser acrescido de outras notas que não façam parte do mesmo (notas de passagem), etc.

Cifras

Cifras são símbolos que identificam relações entre notas dentro de um acorde. As cifras podem ser empregadas para diversos fins, aqui ela é utilizada para a definição dos graus de uma escala e, posteriormente, na disposição dos acordes. Por exemplo, na escala de dó cada nota é identificada por um algarismo romano, nomeando dessa forma os graus da escala, como se vê abaixo.

I (c), II (d), III (e), IV (f), V (g), VI (a), VII (b)

2.2.5 Tonalidades

Tonalidade é o denominador do caráter de uma obra musical. Quando se sabe a tonalidade de uma obra, sabe-se também as principais relações entre as notas que a compõem. A tonalidade indica quais os principais acordes e qual a escala utilizada.

Atualmente, o conceito de tonalidade engloba também o conceito de *modo*. O modo, em Música, acaba se misturando com o conceito de escala. Existem escalas maiores e menores, assim como tonalidades no modo maior e menor. Isto já é bastante corriqueiro, embora alguns teóricos façam distinção entre as duas coisas: tonalidade e modo [CARVALHO, 1994]. O que deve ficar claro é que: (i) tonalidades maiores se apoiam em escalas maiores e (ii) tonalidades menores se apoia em escalas menores (sempre escalas menores harmônicas, para efeitos didáticos).

Todas as tonalidades maiores possuem uma tonalidade relativa no conjunto de tonalidades menores. Por exemplo: a tonalidade de *lá menor* é a relativa da tonalidade de *dó maior*. Uma maneira de se determinar uma tonalidade, ou sua relativa, é através das alterações postas no início de um pentagrama (Figura 2.1), como também através da audição e/ou de determinadas características que facilitam a definição dos modos¹¹.

No ítem seguinte, serão vistas algumas questões relevantes à Harmonia: a sua evolução, principais correntes e aspectos sobre aprendizagem e didática em Harmonia. Finalizando o item e o capítulo, são relacionadas as principais regras harmônicas com as quais se deparam os alunos iniciantes neste domínio.

2.3 Harmonia

Em uma visão simplista, o dicionário Aurélio define *harmonia* como uma “agradável sucessão de sons”. Evidentemente que tal conceito não é de todo incompleto. No entanto, ao se aprofundar no universo musical, descobre-se que a compreensão dessa “agradável sucessão de sons” acaba se tornando uma ciência com grande lastro de estudo. Isto se dá pelas inúmeras possibilidades e maneiras de enfocar este assunto que, por sua vez, se reflete em muitas teorias que culminam na criação musical propriamente dita. E é a isto que a Música, e em particular a Harmonia, vem se sujeitando nos últimos séculos.

¹¹ Em [CHEDIAK, 1985] são dados maiores detalhes sobre este assunto.

Outro fator que contribuiu para torná-la mais atraente do ponto de vista computacional, é a sua ligação com a Matemática [GUIMARÃES, 1990], mas isto será visto com mais atenção no capítulo 5. Este fator “matemático” produziu divisões históricas, onde, por um lado, situam-se os harmonistas pelo cálculo e, pelo outro, os harmonistas pela intuição [ASSAYAG, 1996].

Embora não se esteja aqui para julgar o mérito de qualquer um desses dois grupos, argumenta-se o fato de que o estudo metódico da Harmonia criará em quem a pratica, no mínimo uma consciência de conjunto quando envolve, além do formalismo inerente à Harmonia, as várias ciências que devem caminhar ao seu lado (ingredientes conjunturais, tais como a Estética e seus conceitos de Física, Psicologia, Sociologia e Estatística, juntamente com a História das Artes [FERNEDA, 1994]).

Procedendo dessa maneira, o praticante irá tornar-se um estudioso consciente e não apenas um mero arranjador de notas [SCHONBERG, 1979]. É esta visão, a de uma ciência Harmônica, que se persegue como modelo. O ensino de Harmonia, como será visto mais adiante, deve “conscientizar” o aluno sem a cobrança de torná-lo um compositor. Já para um compositor, ela deverá servir de suporte à compreensão, criação e execução.

A Harmonia como se conhece hoje, surgiu diretamente do *Contraponto*¹². Embora nunca tenha desaparecido, nota-se claramente a grande diferença entre este e a Harmonia. O Contraponto dá ênfase à condução das vozes no seu sentido horizontal, melódico, estudando maneiras de construir melodias de forma natural, combinando-as entre si [PISTON, 1982].

A simultaneidade de várias melodias, acabou por despertar a atenção para os eventos sonoros que eram obtidos. Evidentemente que, com a junção de três ou mais melodias acabava por transparecer blocos sonoros que, por questões de consonância, terminavam constituindo acordes.

Nas Figuras 2.6 e 2.7 mostra-se a diferença de tratamento de um exercício. Na Figura 2.6 pode-se notar a existência de melodias (vozes inferiores) que acompanham a melodia superior. Na figura 2.7, os blocos de notas (acordes) caracterizam o sentido vertical e harmônico deste tipo de tratamento.

¹² A palavra *Contraponto* é derivada da expressão em latim *punctum contra punctum* que quer dizer nota contra nota (quando as mesmas eram representadas por pontos) [PEREZ, 1985]. Isto equivale a dizer que a uma linha melódica superior qualquer se sobrepõe uma linha inferior ou mais linhas. Este conceito da Idade Média posteriormente se assemelhou freqüentemente ao de Polifonia nos séculos de XVI a XVIII.



Figura 2.6: tratamento polifônico



Figura 2.7: tratamento harmônico

Embora se tenha uma diferença visual (gráfica), na realidade é bastante difícil estabelecer uma fronteira formal entre Contraponto e Harmonia [KAPLAN, 1991]. No resultado sonoro, acaba-se por haver uma fusão: as várias melodias terminam por formar acordes e estes, por fim, produzem melodias. Evidentemente que, em termos históricos, tem-se, justamente, a formação de acordes como resultado da prática e da concepção contrapontística [KAPLAN, 1991].

2.3.1 Pequeno Histórico

Foi no século XVII que iniciou-se o interesse no estudo desses resultados sonoros produzidos verticalmente. Buscou-se, então, descobrir regras que pudessem nortear um tratamento coerente, possibilitando uma nova maneira de trabalhar, sistematicamente, a condução das vozes.

Com tal pensamento, a Arte da Música viu uma migração paulatina que principiou com os *motetes*¹³ do século XVII, chegando à monodia e a aparição do baixo cifrado¹⁴, este último coroando uma nova maneira de analisar e fazer música. Isto tudo teve o seu ápice com J. S. Bach (1685-1750) que com maestria “representou o equilíbrio de ambos os elementos, harmônicos e contrapontísticos” [PÉREZ, 1985].

¹³ O motete foi uma das formas populares de polifonia características da Idade Média [PÉREZ, 1985].

¹⁴ Monodia e baixo cifrado, são, respectivamente, uma melodia acompanhada e cifras para a codificação da harmonia tendo como parâmetro um baixo dado.

Com o passar das décadas, a Harmonia avançou mais ainda. Mesmo após um crescente até a etapa clássica e romântica, o século XIX presenciou a quebra de muitos paradigmas e o surgimento de outros. Em Schönberg e seu *dodecafonismo*¹⁵ se deu a culminação disso tudo.

Com isto em disposição, não é justo dizer pura e simplesmente que existe uma Harmonia Musical. Na realidade, o que se tem são “práticas harmônicas”, cada uma delas apropriada à cultura predominante, levando em seu âmago as suas nuances, regras, cuidados, etc [KAPLAN, 1991].

Hoje a Harmonia, tal qual se vê nas escolas e universidades, contempla os seguintes estudos (para maiores detalhes, ver [PÉREZ, 1985]):

- ordenação, combinação e formação de acordes,
- estudo de *encadeamento*¹⁶,
- estudo de modulações (mudanças de modo e de tonalidades),
- ornamentos (tais como notas de passagem, retardos, apojeturas, entre outros),
- cadências,
- resolução de progressões,
- pedal,
- visão histórica.

A partir dessa organização da Harmonia, vários tratados e estudos surgiram com o objetivo de sistematizar a sua aprendizagem. Como exemplo pode-se citar Jean-Philippe Rameau (1683-1764), considerado o organizador dos princípios harmônicos quando da publicação de seu “*Traité de l’Harmonie*” em 1722. Em 1880 um outro grande teórico, Hugo Riemann (1849-1919) publica seu “*Handbuch der Harmonielehre*” assentando os fundamentos da teoria harmônica contemporânea [PÉREZ, 1985].

Com estas condições, acabou-se por se fazer uma divisão bastante importante entre duas correntes no trato e ensino da Harmonia:

- Harmonia Tradicional ou Normativa (com Rameau) e

¹⁵ Sistema de composição atonal, criado pelo compositor austríaco Arnold Schönberg (1874-1951) e baseado no livre emprego dos 12 semitons da escala temperada. De certa forma, este sistema negava todas as regras que até em tão regiam a Harmonia e Tonalidade [PISTON, 1982].

¹⁶ União de dois ou mais acordes segundo as regras da harmonia [PÉREZ, 1985].

- Harmonia Funcional (com Riemann).

A seguir serão vistas as principais características destas duas escolas.

2.3.2 Harmonia Funcional

A principal característica da harmonia funcional é o tratamento harmônico baseado na *Teoria das Funções Harmônicas*. Sobre isto Brisolla fala o seguinte [BRISOLLA, 1979]:

“O conceito de função surgido evidentemente muito mais tarde, na história da música, decorre da idéia de estudar esses acordes quando em movimento, isto é, na concatenação harmônica, e as relações que mantém entre si e com o todo, nesse movimento”.

Aqui, cada acorde assume um “função” dentro do universo harmônico. Estas funções são determinadas pela “atração” que cada acorde exerce uns em relação aos outros. Assim sendo, os acordes de primeiro, quarto e quinto graus recebem as funções de *tônica*, *subdominante* e *dominante*, respectivamente. Todo um estudo é feito para corroborar estas funções. Em torno delas irá girar todos os outros acordes.

Como o interessé aqui não é o de escrever um tratado de Harmonia Funcional, são citadas apenas as chamadas *leis tonais*. Estas leis são em número de cinco e são responsáveis por todas as diretrizes estruturais da harmonia funcional. As cinco leis, segundo [BRISOLLA, 1979], rezam o seguinte:

1ª Lei Tonal: Todos os acordes da estrutura harmônica são relacionados com as três funções principais: tônica(T), subdominante(S) e dominante(D).

2ª Lei Tonal: Todos os acordes da estrutura harmônica são relacionados àquela tônica, subdominante ou dominante da qual são vizinhos de terça.

3ª Lei Tonal: Todos os acordes da estrutura harmônica possuem uma dominante (D) própria.

4ª Lei Tonal: Qualquer acorde pode ser seguido por qualquer outro. Acordes de tons afastados exigem explicação por harmonias intermediárias.

5ª Lei Tonal: Mudança de função significa mudança de tonalidade.

A cada nova lei, vários conhecimentos e regras sobrecondução de vozes (encadeamento) são acrescentados. Para isso existe um conjunto de *fórmulas harmônicas* que, segundo Brisolla, devem ser exercitadas, através de exercícios e da audição das mesmas [BRISOLLA, 1979].

Outra característica marcante da Harmonia Funcional é o tipo de cifra. Os acordes são identificados através das iniciais de tônica, subdominante e dominante: “T”, “S” e “D”, e como todo o restante dos acordes são relacionados com estes três, para as tonalidades relativas e anti-relativas (menores e maiores) acrescentasse a letra “r” ou “a” respectivamente: Tr, Sr, Dr, Ta, Sa e Da. Isto implica em se ter apenas três símbolos (e suas variações) para todos os acordes encontrados em uma peça musical.

Na Figura 2.8 tem-se um exemplo de cifras funcionais.

T Tr S D⁷ T S Tr D D⁷ T

Figura 2.8: exemplo de cifra (harmonia funcional)

No caso da condução das vozes, utiliza-se regras similares (as quais serão vistas mais adiante) às encontradas na harmonia tradicional.

2.3.3 Harmonia Normativa (Tradicional)

Como o próprio nome faz presumir, a Harmonia Tradicional¹⁷ ou Normativa antecede a Harmonia Funcional. Em termos técnicos, pode-se dizer que a Harmonia Normativa possui uma preocupação bastante “exagerada” no que diz respeito ao encadeamento de acordes. Este fato é o responsável por se ter escolhido tal abordagem para esta pesquisa.

Pode-se dizer que a Harmonia Funcional é, antes de tudo, uma teoria onde a aplicação das regras de condução das vozes é testada. Estas regras, entretanto, foram desenvolvidas e praticadas até a exaustão na Harmonia Normativa. Na Figura 2.9 tem-se o exemplo anterior (Figura 2.8) representado agora em cifra tradicional. Esta utiliza os algarismos romanos como escrita para a representação dos graus da escala, como visto no item 2.2.4.

¹⁷ O termo Harmonia Normativa será utilizado deste ponto em diante por este vir desprovido de certos conceitos que o termo Tradicional naturalmente invoca. Aqui fica bastante claro que esta corrente teve como preocupação a compilação de diversas práticas harmônicas e que, posteriormente, foram ordenadas de maneira a facilitar a sua absorção.

Figura 2.9: exemplo de cifra (harmonia tradicional).

Diferente da Harmonia Funcional, que se utiliza de leis para orientar a aprendizagem, a Harmonia Normativa trabalha com regras de condução das vozes. Estas regras por sua vez, possuem determinados graus de importância, segundo o estágio em que se encontra o aluno. No início, este se vê envolto em algumas poucas regras, destas ele não deve “fugir”. Como consequência do acúmulo de conhecimento, novas regras são, aos poucos, dispensadas enquanto outras vão surgindo.

Dessa forma, o ensino da harmonia, através da escola Normativa, possui como um de seus fundamentos, o acréscimo, paulatino, de regras rígidas de encadeamento.

Estas regras são responsáveis por “prender” o aluno ao que estiver sendo prioritariamente ensinado. A lógica destas didáticas, funcional e normativa, serão melhor analisadas no item seguinte. Por enquanto, vale apenas saber que, independentemente da abordagem pedagógica adotada, a representação aqui proposta pode ser utilizada em ambas.

Uma única observação deve ser feita: como dito anteriormente, não se deve negligenciar os aspectos histórico-culturais presentes em todo o caminho da Harmonia Musical. É necessário contemplar tais domínios, tornando possível o reconhecimento de estilos, técnicas, etc.

2.3.4 Aprendizagem e Didática em Harmonia

Para se definir as obrigações pedagógicas do ensino de harmonia, deve-se partir do princípio de que este estudo objetiva tornar o aluno capaz de:

- analisar uma obra musical estrutural e historicamente. Esta análise permitirá ao mesmo não só a identificação formal (tipos de acordes, inversões, retardos, ...), mas também identificar a qual período histórico aquela obra se enquadra;
- resolver exercícios harmônicos no que diz respeito à condução das vozes;
- fazer arranjos corais a quatro vozes, orquestrações e;

- em relação ao compositor, torná-lo consciente de sua criação munindo o mesmo das técnicas utilizadas nos vários períodos históricos.

Estes objetivos não devem limitar-se única e estritamente a conceitos teóricos. Estes devem vir acompanhados de intensa prática, seja através de exercícios e de estudo de obras de compositores consagrados. Tudo isto, evidentemente, seguido de material sonoro (um instrumento harmônico, gravações, etc) [KAPLAN, 1991; BRISOLLA, 1979; SCHONBERG, 1979]. O convívio com os grandes compositores e suas obras, tornará mais concreta boa parte dos problemas que o aluno irá se deparar e também as possibilidades de soluções dos mesmos.

Outro fator, diz respeito ao relacionamento professor/aluno. Sobre isto, Schonberg, nas palavras de Ramon Barse¹⁸, relata o seguinte [SCHONBERG, 1979]:

“Nenhuma outra arte se acha tão obstaculizada por seus professores quanto a música. O professor ‘quer ser infalível’ e isto, além de ridículo, é bastante prejudicial ao aluno. A relação professor-aluno deve ser ... uma busca em comum em que o professor, mais experimentado, pode servir de guia até certo ponto: justamente ao ponto em que o aluno, agitado espiritualmente pelo mestre, se separa naturalmente para criar por si mesmo.”

Isto posto neste caso, deve ficar bastante claro que o objetivo pretendido por um sistema inteligente voltado ao ensino de Harmonia é que este deve ser capaz de permitir este aparte. Entrever semelhante momento é algo inerente ao mestre. Quanto ao aluno, cabe respeitar e se conscientizar de tal decisão.

Quando do ensino, a utilização de leis e regras de forma autoritária pode acarretar frases emitidas por professores tais como: “*Você errou aqui, ali... Mas está bonito!!!*” ou “*Está tudo errado, não era nada disso que eu queria!!!*”. Estes conceitos de certo e errado, em arte, acabam possuindo um limite bastante sutil. Não cabe ao professor penetrar neste campo. Bastaria avaliar o aluno pelo o que foi exposto em aula, sem a necessidade de desprezar o trabalho do mesmo (na maioria das vezes correto se em outro contexto) [TEIXEIRA, 1996].

Quando se objetiva uma representação do conhecimento musical, estas observações são de extrema importância. É preciso levar em consideração a possibilidade de determinadas “escapulidas” por parte do aluno e deve-se representar o conhecimento de maneira a permitir que o sistema inquiria por outras alternativas (de modo a não descartar nenhuma res-

¹⁸ Tradutor da obra “*Harmonielehre*”, de Schonberg, para o espanhol.

posta dada). Esta representação deve ser capaz de fornecer respostas divergentes a um mesmo problema, respostas estas que devem ser plausíveis de acordo com determinados contextos.

O perigo dessa “ditadura” da regra, onde o aluno termina por resignar-se a determinados modelos é apontado por Schonberg [SCHONBERG, 1979]:

“...as leis naturais não conhecem exceções; as leis artísticas se compõem, antes de tudo, de exceções.”

Com este pensamento, pode-se chegar à conclusão que no ensino de Harmonia deve vir acompanhado de preceitos de contraponto, de forma e de estrutura musical, além de outras ciências correlatas.

No modelo de representação proposto no capítulo 5, estas possibilidades não serão negligenciadas. Para cada contexto pretendido durante uma sessão de aprendizagem, os vários objetos necessários para tal propósito devem sobressair dos demais. Além disso, determinadas características e/ou regras podem ser menos ou mais relevantes à determinados problemas.

2.3.5 Princípio de Condução de Vozes (Regras)

Neste ítem, é apresentado uma compilação das regras harmônicas básicas. Estas regras foram compiladas dos mais importantes livros didáticos sobre Harmonia utilizados nas escolas brasileiras [KOSTKA, 1984; HINDEMITH, 1986].

Este apanhado foi feito tendo como base a fase inicial do ensino de Harmonia, ou seja, aquela que trata dos encadeamentos das tríades principais (tônica, subdominante e dominante, ou I, IV e V graus). Este limite foi imposto com o objetivo de esmiuçar as possibilidades encontradas por um iniciante na aprendizagem da Harmonia. Os encadeamentos destas tríades, em suas *posições fundamentais*, são a base para um entendimento posterior dos demais encadeamentos, que acabam sendo extensões destes.

Neste trabalho, conseguiu-se identificar as seguintes regras que tratam da condução das vozes do Soprano (S), Contralto (C), Tenor (T) e Baixo (B)¹⁹.

¹⁹ Na Figura 2.9 as vozes podem ser vistas e diferenciadas: no primeiro pentagrama (clave de sol) têm-se as duas vozes femininas (soprano e contralto), respectivamente de cima para baixo. No pentagrama inferior encontram-se dispostas as vozes masculinas tenor e baixo), também na mesma ordem.

A partir do instante em que for ampliado o material de trabalho (acordes em suas diversas posições, acordes de sétima, etc) algumas regras serão banidas e darão lugar a outras. É com este conceito que se pode, partindo de uma conjunto coeso de regras, analisar e localizar uma determinada obra musical em um contexto histórico.

Regras Básicas²⁰

1. O baixo deve tocar a fundamental do acorde; (*)
2. Deve-se proceder com acordes em posição aberta ou cerrada; (*)
3. Evitar “sonoridades turvas”, que são aquelas onde a distância entre S e C ou entre C e T, ultrapassa uma oitava [KOSTKA, 1984]; (*)
4. A nota da tríade a ser duplicada nas quatro vozes deve ser, de preferência, a fundamental, tendo como segunda opção a quinta; (*)
5. Não duplicar a terça; (*)
6. Se existir uma nota comum aos dois acordes, deve-se mantê-la;
7. O movimento de uma nota a outra em uma voz deve ser feito pelo caminho mais curto possível, ou seja, pela menor distância;
8. Caso exista duas notas iguais e comuns, deve-se manter uma e mover a outra;
9. Quando não existir uma nota comum entre os acordes, os movimentos das vozes superiores (S, C e T) devem ser contrários ao baixo. Dessa forma compensa-se as energias melódicas e harmônicas;
10. Evitar 5ª, 8ª paralelas ou uníssono paralelo “censuráveis”, principalmente se acontecerem entre S e B [KOSTKA, 1984];
11. Não movimentar as quatro vozes em uma mesma direção por movimento direto;
12. Evitar 5ª e 8ª paralelas ocultas quando de um intervalo qualquer, principalmente intervalos menores, as vozes do S e B resolvem a uma distância de 5ª ou 8ª no segundo acorde;
13. Movimentos de três vozes em uma mesma direção devem ser compensados por uma nota sustentada ou em movimento contrário (vide regra 9);
14. Evitar saltos maiores que uma quinta nas vozes superiores (S, C e T);

²⁰ As regras destacadas com um asterisco dizem respeito à distribuição das quatro vozes, independentemente de encadeamentos.

15. Os intervalos aumentados e diminutos (difíceis de serem cantados) devem ser evitados;
16. O sensível deve sempre “resolver” em direção à tônica da tonalidade;
17. Não é permitido saltos melódicos maiores que uma oitava;
18. As vozes nunca devem cruzar umas às outras; (*)
19. Em alguns casos, pode-se suprimir a quinta e triplicar a fundamental. (*)

Esta relação de regras estão apresentadas acima de modo genérico. Quando do ensino dos encadeamentos, cada caso possui um procedimento próprio que garante a sua resolução de maneira correta.

Procedimentos Específicos de Encadeamento

Encadeamentos de I - V, V - I, I - IV e IV - I (fundamental duplicada):

1. Escrever a progressão do baixo do primeiro acorde ao segundo (regra 1),
2. Completar o primeiro acorde (regras 2, 3, 4 e 5),
3. Manter, no segundo acorde, a nota comum a ambos (regra 6),
4. Conduzir as duas notas restantes do primeiro acorde por grau conjunto, às notas mais próximas do segundo (regras 7 e 16).

Encadeamentos de I - V e I - IV (quinta duplicada no 1º acorde):

1. Escrever a progressão do baixo do primeiro acorde ao segundo (regra 1),
2. Completar o primeiro acorde (regras 2, 3, 4 e 5),
3. Se houver duas notas comuns, manter uma e movimentar a segunda pelo caminho mais curto (salto de terça maior ou menor) à nota mais próxima do segundo acorde (regras 7 e 8),
4. Se não se puder manter nenhuma nota, conduzir cada uma das três vozes superiores pelo caminho mais curto (salto não maior que uma quarta) à nota mais próxima do segundo acorde (regras 7, 9, 10, 11 e 13).

Encadeamentos de IV - V e V - IV (fundamental duplicada no 1º acorde):

1. Escrever a progressão do baixo do primeiro acorde ao segundo (regra 1),
2. Completar o primeiro acorde (regras 2, 3, 4 e 5),

3. Conduzir as três vozes superiores em movimento contrário ao baixo e pelo caminho mais curto possível (regras 6, 9, 12 e 13),

Encadeamentos de IV - V e V - IV (quinta duplicada no 1º acorde):

1. Escrever a progressão do baixo do primeiro acorde ao segundo (regra 1),
2. Completar o primeiro acorde (regras 2, 3, 4 e 5),
3. Conduzir as três vozes superiores em qualquer direção e pelo caminho mais curto, às notas do segundo acorde (regras 6, 14, 15 e, em alguns casos, 20).

Como pode ser observado, quase todas as regras citadas anteriormente foram utilizadas nos procedimentos de encadeamento. As regras 17 e 18 devem ser observadas após a resposta dada pelo aluno ou pelo sistema que esteja utilizando este esquema.

3 Computação e Música

3.1 Introdução

O ser humano, desde o momento em que deixou a sua passividade diante dos fenômenos da natureza, vem aos poucos ao encontro de seus sonhos. Pode-se dizer que toda “revolução” promovida pelo homem possui como ponto de partida a sua curiosidade diante das forças naturais.

A palavra “revolução” surgiu com Nicolau Copérnico (1473 - 1543) em 1514, utilizando-a para descrever o movimento do sistema solar. Por fim, este termo passou a significar tudo aquilo que vai de encontro às estruturas sociais, políticas e econômicas.

Revoluções acontecem tanto nas ciências, como nas artes. Como este trabalho trata de Música (como arte e ciência!), deve-se remontar ao ano de 1626, ano em que foi publicada “*A Nova Atlântida*”, do filósofo inglês Francis Bacon (1561 - 1627). Esta obra já previa muitas das tecnologias que vieram a ser aplicadas à Música [MOORE, 1996]:

“Temos ainda casas de som, onde são experimentadas todas as espécies de som e seus derivados. Temos harmonias que não tendes, de quarto de tom, e de notas a intervalos ainda menores; temos alguns instrumentos musicais desconhecidos de vós, alguns deles soam mais docemente que os vossos, e também sinos e timbres harmoniosos e delicados. Transformamos os sons fracos em fortes e profundos e, inversamente, em fracos e tênues os fortes; sabemos produzir tremulações e vibrações diversas de um som originariamente contínuo. Reproduzimos e imitamos todos os sons e letras articulados, as vozes e as notas dos animais e dos pássaros. Temos certos aparelhos que, aplicados ao ouvido, aumentam a audição, e ainda diversos ecos estranhos e artificiais que repetem as várias vozes diversas vezes, como que as repercutindo. Alguns outros tornam as vozes mais agudas do que eram originariamente, outros mais profundas, enquanto alguns outros devolvem as vozes com letras e articulações diferentes da forma original. Temos ainda instrumentos para conduzir os sons em tubos e condutos a uma grande distância, mesmo em curvas. ...”

Bem antes do século XVII, porém, o homem já procurava meios mecânicos de armazenar música. Estas tentativas remontam ao século XIV e se estende até nossos dias. Exemplos recentes disto são os carrilhões mecânicos (capazes de armazenar cerca de 9000 notas), as baterias automáticas (*Polyphon* - 1904) e, talvez o mais interessante deles, o *Componium* (concluído em 1821). Este último aparelho, desenvolvido por D. Winkel, propunha-se a

produzir variações sobre um tema dado [MOORE, 1996].

Em meados do século XIX, diversos aparelhos reprodutores de música eram comercializados (caixinhas-de-música, pianolas, etc). Em 1906, Thaddeus Cahill inventava o *Dynamophone*. Este aparelho, composto de indutores eletromagnéticos, causava espanto nas platéias, pela sua capacidade de criar sons. Outro nome que deve ser lembrado é o do russo Leon Theremin que, em 1924, inventou um instrumento batizado com o seu nome que até hoje é montado por estudantes de eletrônica. Depois, com o aparecimento do fonógrafo, ficou mais fácil registrar eventos sonoros.

Surge então, na França da década de 40, um movimento chamado *Musique Concrète* (Música Concreta). Seu idealizador foi Pierre Schaeffer (1910 - 1995) [LOVELOCK, 1987]. A *Musique Concrète* fazia uso de fitas magnéticas de alta fidelidade (*tapes*: inventadas durante a Segunda Guerra Mundial), nas quais se gravavam sons naturais (sons de sino, gongo, chuva etc) [COPLAND, 1969]. Estes sons podiam ser revisados e transformados posteriormente, de maneira a torná-los irreconhecíveis. Esta manipulação de fitas foi apresentada pela primeira vez em 1948.

Na metade do século XX, estava havendo, segundo Carpeaux [CARPEAUX, 1958], algo como um certo “enjôo dos mundos sonoros antigos”, visto o florescimento de novos instrumentos e o aparecimento avassalador de novas informações musicais. Estes fatores, juntamente com os grandes acontecimentos sociais contemporâneos, requeriam uma máquina capaz de “absorver” tais inovações na rapidez de suas ocorrências.

Na década de 50 surge, na Alemanha, um outro grupo preocupado em diversificar esta nova música. Este grupo era liderado por um discípulo de Schaeffer, Karlheinz Stockhausen (1928 -). Ele fundou o Laboratório de Música Eletrônica (*elektronische Musik*), que tinha como principal apoio científico para as suas composições, as teorias do famoso matemático Jean-Baptiste Joseph Fourier (1768 - 1830). A música eletrônica de Stockhausen negava os sons reais da *musique concrète*, concentrando-se em sons sintéticos produzidos por geradores analógicos de tonalidades.

Foi porém o surgimento dos computadores e a sua disseminação após a Segunda Guerra Mundial que possibilitou, posteriormente, sua utilização nas atividades de codificação, geração, registro e análise musical [ROADS, 1985].

A seguir, a ênfase é dada ao desenvolvimento da Música aliada à Computação. Para entender melhor todo o processo de assimilação desta nova tecnologia por parte de músicos,

engenheiros, matemáticos e informatas, é utilizada a divisão em quatro períodos de 10 anos, proposta por Moore [MOORE, 1996].

3.2 Histórico

A primeira década inicia-se em 1955. O expoente deste primeiro período é o americano Lejaren A. Hiller (1924 -).

Hiller era químico e compositor amador e, através do uso da matemática na Química, acabou vislumbrando a possibilidade de transportar as idéias dos *processos estocásticos*²¹ para a Música. Em 1959, Hiller, e seu colega da Universidade de Illinois, o matemático Leonard M. Issacson, publicam o livro "*Experimental Music - Composition with an Electronic Music*", cujo subtítulo era bastante sugestivo para a época. Este livro narra a experiência da utilização do, na época, "supercomputador" ILLIAC, na geração de melodias. Na realidade, o computador gerava uma série de números aleatórios que eram depois associados a parâmetros de ritmo, tonalidade, dinâmica, etc.

Estes números eram gerados segundo o Método de Monte Carlo²². Posteriormente, Hiller buscou ajuda nas idéias de outro matemático, Andrei Andreevich Markov, para analisar obras de Mozart. A idéia de Hiller era fazer uma análise probabilística das obras de um determinado compositor. Por exemplo, qual a probabilidade de, após um dó natural, aparecer um ré, ou um mi, ou um mi bemol? Desta maneira, Hiller julgava que um sistema computacional poderia produzir música tendo como dados de entrada estas informações, e que pudesse não só criar músicas no estilo do compositor, mas também analisar a qualidade de uma obra! Seu sonho era gerar automaticamente "obras póstumas" dos grandes compositores clássicos.

Como resultado prático de suas experiências, as seguintes obras foram publicadas com nomes bastante sugestivos: *Illiad Suite for String Quartet* (a mais famosa), *Computer Sonata* e *Algorithmus*, entre outras [PÉREZ, 1985].

É bom lembrar que, até este momento, o computador não produzia e nem controlava nenhuma espécie de som; ele apenas a gerava dados para a confecção de uma partitura, ficando a execução a cargo de músicos.

²¹ Já em 1922, o compositor grego Yannis Xénakis se utilizava de determinados conhecimentos e técnicas matemáticas para modelar obras musicais. A utilização de probabilidades matemáticas na música ficou conhecida como música estocástica.

²² Este método utiliza a estratégia de tentativa e erro aleatoriamente e a melhor solução é conservada como resultado de sua aprendizagem.

Nesta mesma época, em 1957, o computador passou a controlar e a gerar sons. O responsável por isso foi o engenheiro da AT&T Bell Telephone Laboratories e violinista amador norte-americano Max Mathews. A idéia de Mathews era, utilizando os mesmos princípios da telefonia, converter os sinais digitais fornecidos pelo computador em sinais analógicos, controlando, assim, um alto-falante. Para isto, ele utilizou um aparelho chamado *conversor digital-analógico*.

Em princípio, o sistema de Mathews, chamado de *Music I*, era capaz de sintetizar qualquer espécie de som. No entanto, na prática, a coisa era bastante diferente. Primeiro pelas pressões vindas de críticas e preconceitos inerentes a qualquer trabalho pioneiro, segundo pelas dificuldades. Por exemplo:

- demora na programação advinda das dificuldades de alimentação e recuperação de dados, comuns aos recursos disponíveis na época (codificação em *Assembly* e armazenamento em cartões e fitas): costumava-se gastar um dia inteiro na programação de uma única tentativa de síntese sonora;
- além do resultado sonoro, o computador dava como saída material, um grande conjunto de informação, o que tornava bastante lenta a leitura dos dados, bem como a sua posterior alteração;
- após o trabalho de sintetização pronto, havia também o esforço, nem um pouco trivial, para a conversão analógica em fita magnética;

Superada estas dificuldades, tinha-se como resposta as seguintes possibilidades:

1. Silêncio: o que era o mais comum!
2. Um longo e alto ruído: o segundo resultado mais freqüente.
3. Algo inesperado e diferente das opções acima: geralmente provocado por erros de programação.
4. Ou, por fim, o som esperado: que vinha “fino, débil e pálido e sem muita definição...” [MOORE, 1996].

Apesar das frustrações, mais um passo tinha sido dado para a união mais “íntima” entre Computação e Música, em seus domínios de Processamento Digital, Performance, Composição, Educação e Teoria Musical [ROADS, 1985].

* * *

Pode-se dizer que o segundo período (1965 - 1975) ficou caracterizado pelas pesquisas

em processamento digital do som. Os engenheiros e músicos tinham pela frente duas opções de trabalho:

- (i) tentar “imitar” o som de instrumentos existentes e conhecidos, ou
- (ii) produzir novos sons.

Em ambas as abordagens, as dificuldades são evidentes. Mesmo que um resultado aproximado deixasse satisfeitos os engenheiros, ficava evidente para os ouvidos mais exigentes o longo caminho a ser vencido para se chegar a um resultado aceitável. A segunda abordagem tinha como contrapartida as questões de Estética e de Percepção. Como vencer hábitos sonoros dominantes, já bastante arraigados?

Um dos expoentes dessa época foi o físico e compositor francês Jean-Claude Risset, dos Laboratórios Bell, que, juntamente com Max Mathews, trabalhou com a sintetização do som do Trompete. O trabalho anterior de Mathews com o *Music I* tinha servido para mostrar aos músicos as propriedades e virtudes do som musical, bem como as suas alterações físicas e psicológicas (*Psicoacústica*²³).

Como resultado das pesquisas de Risset, chegou-se à conclusão que o que determina a característica do timbre²⁴ de um instrumento é justamente o número de harmônicos contidos no som. A emissão de um som por um instrumento de metal e também pela maioria dos instrumentos está representada na Figura 3.1. Pode-se detectar quatro fases distintas: a primeira, que corresponde à linha ascendente vertical, é o que se chama de *ataque*²⁵ da nota; a segunda (linha horizontal) corresponde à *sustentação* do som; a isto, segue-se a fase cujo o som decresce a um nível mínimo (*declínio*); logo depois vem a fase de *silêncio*: a nota deixou de ser emitida.

Com estes dados, Risset construiu seu almejado som sintetizado de Trompete e, a partir deste som, fez com que seu sistema gerasse pequenas melodias. Estas melodias foram inseridas (melodias incidentais) em sua composição *Little Boy* (em alusão à bomba atômica de Hiroshima). Nesta obra, por meio de seu computador-sintetizador e a partir de suas pesquisas anteriores, Risset criou uma ilusão acústica bastante interessante que chamou de “*sons paradoxaes*”. Estes sons aparentam “descer” infinitamente, quando na realidade

²³ *Psicoacústica* é a disciplina que compreende o estudo das relações entre o sons, suas propriedades e efeitos, e a mente humana [PÉREZ, 1985].

²⁴ *Timbre* é a qualidade que diferencia um som de outro, mesmo que estes tenham frequência e intensidade iguais. Até mesmo dois instrumentos de uma mesma espécie (duas flautas por exemplo) possuem diferenças de timbre [PÉREZ, 1985].

²⁵ *Ataque* é a maneira de emitir um som vocálico; de fazer vibrar uma coluna de ar de um tubo, as cordas ou a forma

eles estão “subindo”, efeito mais do que oportuno para expressar a relação bomba *versus* Hiroshima.

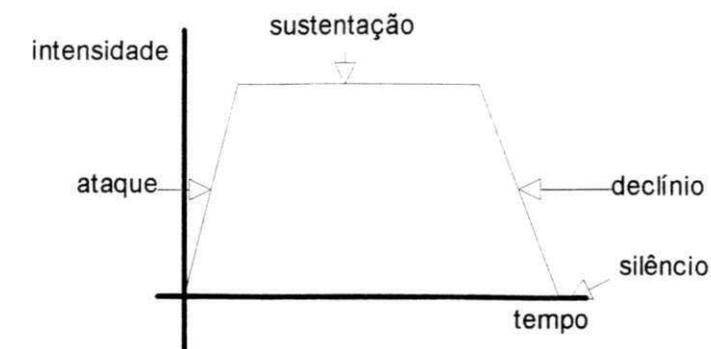


Figura 3.1: gráfico de um envelope de som

Paralelamente a este trabalho, um outro pesquisador também desenvolve um sistema semelhante, embora por outro caminho. Seu nome é Pierre Ruiz, aluno de Hiller. Ruiz era físico e propôs, através de equações diferenciais, descrever o comportamento de sistemas físicos, incluindo também os instrumentos musicais [MOORE, 1996].

Como as equações diferenciais podem ser resolvidas por diversos caminhos matemáticos, Ruiz utilizou o computador para solucioná-las, ao mesmo tempo em que podia ter visões instantâneas de cada passo da resolução das mesmas. Dessa maneira, Ruiz chegou a sintetizar o som do violino, considerado, na época, o som sintetizado mais aproximado do instrumento verdadeiro.

Em 1967, na Universidade Carnegie-Mellom e na Universidade de Illinois, Richard Moore e Max Mathews trabalharam na tarefa de desenvolver um programa capaz de economizar tempo no trabalho de síntese de som. A este projeto foi dado o nome *Music V* (uma quinta versão do *Music I*²⁶).

Diferentemente de sua primeira versão, o *Music V* era programado em uma linguagem de alto nível (FORTRAN IV). Os sistemas computacionais utilizados foram um IBM 7064, um Honeywell/GE 645 e um Control Data 6600.

O *Music V* era considerado uma ferramenta muito poderosa que, além de sintetizar, proporcionava estudos sobre a relação entre percepção e física do som. O *Music V* trabalhava

de apoiar-se sobre as teclas [PÉREZ, 1985].

²⁶ Ao *Music I* seguiu-se uma série de programas voltados ao objetivo de sintetizar sons. Já em 1958, Mathews lança o *Music II* desenvolvido no IBM 704. O *Music III* veio em 1959, utilizando o IBM 7094. Ainda neste período, surge, em 1964, o *Music IV*, escrito em FORTRAN. Entretanto, esta versão IV, em relação à versão anterior, não apresentava uma performance tão boa quanto [MIRANDA, 1990].

da seguinte maneira: alimentava-se o sistema com a descrição matemática de um ou mais instrumentos virtuais previamente produzidos e, depois, uma lista com as notas a serem executadas. É evidente que ainda havia uma distância considerável entre o resultado alcançado e a performance de um músico humano. As razões são as seguintes:

- a máquina executava a música de maneira inexorável, diferente de um músico que a cada execução a interpreta com diferentes nuanças;
- a máquina, neste estágio, ainda omitia certas informações essenciais a uma boa interpretação de uma música (dinâmica, alteração de andamento, etc);
- um ponto importantíssimo é justamente o que diz respeito às características físicas de um instrumento e a sua relação com o ambiente (umidade e acústica, por exemplo) e
- o aspecto estático de uma máquina ao executar uma música.

Embora, aos olhos de um técnico, estes pontos sejam pequenas “imperfeições” e aparentemente supérfluos, para a Arte tornam-se fundamentais.

Tem-se agora uma máquina capaz de sintetizar um instrumento e executar melodias. Porém, até então nenhuma interface musical (um teclado de piano, por exemplo) fora colocada entre a máquina e o homem. Todo o trabalho ainda era feito através de botões e potenciômetros e sem ações em tempo real. Pensando nisso, Moore e Mathews trabalharam em um outro sistema o qual batizaram GROOVE (*Generated Real-time Operations On Voltage-controlled Equipament*). Este sistema era implantado em um “pequeno” computador (ocupando algo em torno de uma sala e um quarto de uma casa), acoplado a equipamentos de gravação em fita e uma coleção de módulos de sintetizadores Moog analógicos²⁷.

Embora ainda não utilizando uma interface musical, o GROOVE permitia a ação do músico através de chaves e potenciômetros em tempo real. Toda a síntese do som era feita através de variação de voltagem (alturas e amplitudes). Posteriormente, foi acoplado um teclado ao GROOVE, juntamente com pedais e *joysticks*, permitindo que usuários não técnicos pudessem utilizar o sistema.

Desta maneira, a área de Performance em Computação e Música ganhava terreno entre os

²⁷ Robert A. Moog (1934 -), engenheiro norte-americano que construiu sintetizadores analógicos até hoje admirados pela sua qualidade sonora. Moog utilizava o princípio da “síntese por subtração”, onde um som básico é criado eletronicamente; este efeito sonoro passa depois por uma série de processos eletrônicos. Cada processo altera e divide o som, de maneira que adquira formas utilizáveis pelos músicos. Seus sintetizadores passaram a ser co-

cientistas e pesquisadores. Buscava-se saber até que ponto um sistema digital conseguiria responder às intenções de um músico e, se não conseguisse, como minimizar tal distância.

Ainda nesta década, mais precisamente no final dos anos 60, John Chowning, utilizando o *Music V* na Universidade de Stanford, ao investigar o efeito de vibrato²⁸ de uma nota, acidentalmente alterou um dos parâmetros responsáveis por este efeito (os valores normais ficavam entre 5 e 6 Hz) [MOORE, 1996]. A sua alteração gerou como saída um timbre totalmente diferente do esperado. Desta maneira, Chowning acabava de descobrir a síntese por modulação de frequência (FM - *Frequency Modulation*).

Este tipo de modulação não era nenhuma novidade; desde a década de 30 já se usava a FM como modo de transmissão por ondas de rádio. No entanto, a descoberta de seu uso em síntese veio como uma ótima possibilidade de sintetizar sons e também como forma de diminuir custos. Esta possibilidade se apoiava no fato de não ser mais necessário alterar o som em sua origem; incluindo-se mais um oscilador e, neste ponto, alterando-se seus valores de modulação de frequência, criavam-se novos sons.

Após esta descoberta e posterior registro, a Universidade de Stanford vendeu a idéia à multinacional japonesa Yamaha que, por sua vez, foi responsável por tornar acessível a um número maior de pessoas os seus sintetizadores baseados em FM. Isto se deu no início da década de 80 com a famosa série de teclados/sintetizadores batizada de DX-7.

* * *

Enquanto os anos entre 1965 e 1975 foram dedicados às descobertas em síntese, o período seguinte (1975 - 1985) irá presenciar uma “revolução digital” com a disseminação dos microprocessadores. Como consequência imediata, veio a queda dos preços e o aumento do número de pessoas que tinham acesso a esta tecnologia, em particular sua aplicação em Música. Aos poucos, foram surgindo os laboratórios e estúdios domésticos.

Nos meios acadêmicos, esta década viu surgir grandes centros de pesquisa que se dedicavam ao estudo em Música & Computação. Entre eles pode-se citar como referência mundial os seguintes:

- CCRMA: *Center for Computer Research in Music and Acustics* (Universidade de Stanford), criado em 1975.
- IRCAM: *Institute de Recherche et Coordination, Acoustique/Musique* (Paris), cria-

mercializados em 1963.

²⁸ *Vibrato* é um efeito especial ondulatório do som, produzido por leves oscilações do mesmo durante a execução, produzindo um efeito bastante interessante se usado com bom gosto [PÉREZ, 1985].

do em 1978.

- CARL: *Computer Audio Research Laboratory* (Universidade da Califórnia - San Diego), criado em 1979.

A partir dos DX7 da Yamaha, os sintetizadores de qualidade ficaram acessíveis se comparados aos sintetizadores anteriores à época da FM²⁹.

Outros desenvolvimentos tecnológicos não paravam de acontecer:

- instrumentos com sensibilidade ao toque³⁰;
- em 1978, o primeiro sintetizador polifônico (Prophet 5), capaz de tocar mais de uma nota ao mesmo tempo;
- capacidade de memória ampliada;
- surgimento de outros processos de síntese (síntese digital, síntese aditiva, síntese por algoritmos, etc);
- a tecnologia de *sampler*³¹, possibilitando digitalizar um som analógico e sua reprodução com ou sem alterações do som original e
- o padrão *MIDI specification 1.0* (Musical Instrument Digital Interface) em 1982.

Este último, o MIDI, veio como forma de unificar um protocolo comum de comunicação entre sintetizadores, sequenciadores, computadores e qualquer outro tipo de aparelho que suporte informações digitais. Até a década de 70, os sintetizadores analógicos tinham como modo de comunicação a variação de tensão (por exemplo: o aumento de 1 volt significava uma oitava acima, etc). O MIDI, através de acopladores óticos, passou a comandar estas informações de maneira digital e com mais segurança [RATTON, 1992].

* * *

Finalmente, chega-se ao período de 1985 - 1995, época em que ocorre a “revolução virtual”. Com a palavra, Miguel Ratton:

Computadores, processamento digital e instrumentos repletos de recursos, tudo isso a um

²⁹ Os DX-7 foram vendidos ao preço de US\$2,000. E em 1983, venderam-se em torno de 500.000 unidades em todo o mundo [MOORE, 1996].

³⁰ De acordo com a velocidade com que uma tecla é pressionada, o sintetizador responderá com maior ou menor intensidade (similarmente a um teclado de piano).

³¹ Um *sampler* é uma amostra digitalizada de um som real. O *sampler* serve para reproduzir os timbres complexos (harmônicos) dos sons naturais [RUBIN, 1996]. Isto foi possibilitado pela redução dos custos das memórias digitais [RATTON, 1992].

custo acessível para muitas pessoas, têm levado profissionais da área de música a optar pela realização individual de todo um trabalho musical, em casa ... A facilidade de veiculação (mídia) do trabalho musical, aliada às perspectivas de lucro (financeiro) devido a sua divulgação e venda em massa, tem determinado rumos, às vezes questionáveis, para a expressão musical". [RATTON, 1992]

Nas últimas três décadas, em termos industriais, nota-se o surgimento de três correntes de investimento. Estas correntes possuem como base para o seu progresso, o mercado consumidor. Este mercado, sempre crescente, é com certeza o grande incentivador do surgimento de novos produtos na indústria de computadores pessoais e, mais particularmente, na indústria de teclados e sintetizadores e no desenvolvimento de *softwares* voltados à Música [SPIEGEL, 1996].

* * *

Muito embora muitos problemas em Computação e Música ainda persistam, o progresso conseguido nos últimos 20 anos foi surpreendente. Assim, cada vez mais surgem trabalhos em áreas mais especializadas e bem definidas. Partindo das quatro áreas principais detectadas por [ROADS, 1985], chega-se hoje a algumas especializações³²:

(i) em Processamento Digital:

- sistemas e linguagens para síntese;
- processamento de sinais e síntese sonora;
- transformação do som.

(ii) em Performance:

- desempenho,
- interface com o usuário;
- projeto de instrumentos;
- música eletroacústica.

(iii) em Composição:

- sistemas e linguagens de composição.

(iv) em Teoria Musical:

- sistemas de notação musical;

³² Para maiores informações sobre taxionomias em Computação e Música, vide Anexo 1.

- reconhecimento automático de partituras;
- sistemas de análise musical;
- sistemas tutoriais em educação musical.

Segundo E. Miranda [MIRANDA, 1990], o domínio da Composição é o que apresenta maiores desafios futuros (sistemas inteligentes de composição). Para a realização de tal intento, os cientistas trabalham sob dois paradigmas que, embora correlatos, acabam por criar uma lacuna considerável entre si. São eles:

- Composição sem síntese sonora simultânea e
- Composição e síntese sonora.

O primeiro paradigma trabalha com a geração de música, tendo dois sistemas distintos: o sistema de geração de notas (onde o padrão MIDI é geralmente utilizado na execução), e o bloco de sintetização com timbres previamente definidos. No caso do segundo paradigma, um mesmo sistema é responsável pela manipulação tanto da forma musical quanto do som. Tanto um quanto o outro paradigma possuem vantagens e desvantagens e, ainda segundo E. Miranda, a segunda opção é bem mais significativa pelo fato de influenciar o avanço da linguagem musical.

Entremeada nestas áreas, a Inteligência Artificial vem contribuindo de forma bastante significativa. Na seção a seguir serão apresentadas algumas relações entre Inteligência Artificial e Música, seus progressos e suas limitações.

3.3 Música e Inteligência Artificial

Como foi visto anteriormente, desde as pesquisas de L. Hiller e de seu desejo de possuir uma máquina capaz de compor e fazer críticas, uma porta foi aberta para as pesquisas em Computação e Música. Apesar da pretensão, desde os anos 50, de se obter uma máquina voltada à Música, foi somente em 1980, em um artigo publicado por Curtis Roads [ROADS, 1985], que as potencialidades da aplicação da Inteligência Artificial foram expostas.

Mas o que é um programa inteligente em música? Para E. Miranda, um sistema capaz de realizar tarefas dentro do domínio da música (seja de composição, análise, arranjo, educação musical, etc) pode ser chamado de inteligente se for capaz de:

- resolver novos problemas;

- absorver novos conhecimentos (fornecidos por um especialista);
- explicar, justificar e, se for o caso, sugerir possibilidades de resposta a um usuário aprendiz [LASKE, 1989; FRITSCH, 1994];
- se adaptar a situações novas e inesperadas, passíveis de resolução [FRITSCH, 1994];

3.3.1 Aspectos de Representação em Música

Evidentemente, no domínio da Música existem muitas coisas que não são formalizáveis, no sentido de, por exemplo, se quantificar e definir alguns de seus elementos (tempo, duração, intensidade, etc). Nestes casos, conceitos como tensão, expectativa, beleza, emoção, etc, acabam por comprometer um sistema dedicado à resolução de problemas em Música. Esta falta de formalismo se dá na medida em que a simples referência a um componente musical, como por exemplo:

“resolva a tensão harmônica no segundo compasso”,

implica em questões formais (neste caso, pode-se definir com precisão onde inicia e termina o segundo compasso, as notas e pausas que se encontram neste âmbito, os acidentes, acordes, etc). Já a primeira parte da frase acima, pode gerar um número maior de interpretações. Isto se deve ao fato de tais termos dependerem em grande parte do contexto. A palavra “tensão”, dependendo do estilo e época de uma obra, pode sofrer pequenas variações (importantes para quem interpreta), que não devem ser negligenciadas.

Como conclusão, pode-se dizer que em Música existem duas classes de conhecimento [FRITSCH, 1994]: *discursivo* (formalizável) e *auditivo* (não formalizável).

Quando se trata do conhecimento discursivo, diversos sistemas em funcionamento, que se utilizam de técnicas de IA, apresentam bons resultados (por exemplo, [BRANDÃO, 1995; FRITSCH, 1995; MARTÍN, 1995]). Já o conhecimento auditivo ainda se encontra fora do ambiente de compreensão da máquina [FRITSCH, 1994].

Pode-se dizer que, a partir da Música Contemporânea, o homem passou a dar mais valor ao resultado sonoro no sentido de abandonar regras e fórmulas passadas. Como resultado desse modo de pensar, hoje em dia existem compositores que, ao invés de falar em termos tais como tema, frase, motivo, forma, etc, preferem utilizar termos como objetos sonoros, sons ásperos, rugosos, lisos e assim por diante [GUIGUE, 1995].

3.3.2 Atividades em IA e Música

Segundo E. Miranda [MIRANDA, 1990], pode-se dividir a atuação da Inteligência Artificial em Música em seis campos distintos. São eles:

- a) Sistemas Especialistas (SE): constituídos de um banco de conhecimento sobre um domínio específico da Música. Esta especificidade se dá por existirem diferentes abordagens para um determinado objeto musical, seja quanto à análise, à construção harmônica e/ou melódica, à interpretação, etc³³. Espera-se que estes tipos de sistemas possuam o máximo de conhecimento que um especialista possa fornecer. Uma das grandes questões dos SE, relacionada com aquisição do conhecimento, está ligada ao “ruído” possível na interação especialista - engenheiro de conhecimento - sistema.
- b) Sistemas Tutoriais (ST): os ST são programas que têm por finalidade o ensino de determinado conhecimento, tendo como parâmetro um conjunto de regras e restrições. Atualmente, estão disponíveis no mercado diversos *softwares* destinados ao ensino de música. Além dos ST, têm-se os STI (Sistemas Tutoriais Inteligentes). O que caracteriza um STI é a sua capacidade de se adaptar, durante as sessões de ensino, ao comportamento individual de cada aluno³⁴.
- c) Linguagens: as linguagens em IA e Música podem ser vistas de duas maneiras: (i) através do conceito tradicional de linguagem de programação, sendo, neste caso, voltada a programas de análise ou composição. Neste sentido, o termo linguagem passa a ser interpretado como um sistema de representação para os objetos mais complexos da Música (frases, motivos, etc). (ii) através do conceito de gramáticas gerativas, onde se busca a vinculação a estilos musicais compostos³⁵.
- d) Modelagem e Representação do Conhecimento: as dificuldades em modelar e representar em Música serão apresentadas posteriormente com mais detalhes. Por enquanto, deve-se entender que, em geral, as representações propostas limitam-se a determinados domínios e/ou pontos de vista [TEIXEIRA, 1996]. Quando se trata de aspectos estruturais, por exemplo, uma representação voltada ao ritmo acaba

³³ Como exemplos de Sistemas Especialistas recentes, aqui no Brasil pode-se citar: SETMUS [FRITSCH, 1995], *Expert Piano* [GLANZMANN, 1995].

³⁴ Em [COSTA, 1996] encontra-se a descrição de um STI voltado ao ensino da Harmonia. Neste sistema é levado em consideração não apenas os aspectos individuais de cada aluno; um outro fator de grande importância a sistemas dessa natureza é a capacidade de interpretar determinados objetos, de acordo com o contexto em que se encontra o aluno.

³⁵ Um trabalho interessante nesta área é o de David Cope (citado em [MIRANDA, 1990]). Neste trabalho, Cope utiliza a aplicação de diferentes bancos de regras (chamados de “dicionários”) para a criação de novos estilos de

por desconsiderar as informações que dizem respeito à harmonia; quando se trata de representação das interações com o usuário (MIDI, *Piano Roll*, etc), determinados fatores não podem ser desprezados, tais como tonalidade, acordes, escalas, etc.

- e) Redes Neurais e Processamento Paralelo Distribuído: enquanto, por um lado, existe uma IA baseada na resolução de problemas através de processamentos simbólicos (IA simbólica) [LASKE, 1989], por outro existe uma IA que tem como paradigma, a simulação de um cérebro humano por uma máquina. É neste segundo caminho que se enquadra as Redes Neurais e o Processamento Paralelo Distribuído (PPD). Dentro desta maneira de pensar a IA, alguns trabalhos estão surgindo nas áreas de análise harmônica, reconhecimento de timbres, reconhecimento de padrões rítmicos, síntese musical, etc. Pode-se destacar, dentro da análise harmônica, o trabalho de F. Beckenkamp [BECKENKAMP, 1995]. Pode-se citar os trabalhos de Pérez [PÉREZ, 1995] sobre reconhecimento de timbres musicais. Em análise e segmentação de obras musicais, tem-se o trabalho de O. Carpinteiro [CARPINTEIRO, 1995].
- f) Dispositivos para “performance” (interfaces): em geral, quando se fala em interfaces para músicos, pensa-se logo em um teclado. No entanto, hoje em dia existe uma variedade enorme de dispositivos destinados a diminuir ao máximo a distância entre uma peça musical e a máquina responsável pela sua execução. Atualmente, existem dispositivos que permitem a violonistas, instrumentistas de sopro, etc, se utilizarem de tecnologia sem perda de espontaneidade [RUBIN, 1996]. A IA entra aqui com o intuito de integrar, literalmente, o músico e a máquina. Neste caso, sensores são desenvolvidos com a capacidade de poder gerar música, seja através de instrumentos parecidos com os tradicionais, seja pelo uso de propostas inovadoras (sensores bioeletromagnéticos, por exemplo).

3.3.3 Problemáticas em IA e Música

Tendo como base as cinco áreas principais da música (Composição, Performance, Teoria, Processamento de Som e Educação Musical) serão vistas abaixo, algumas dificuldades existentes quando da ligação entre Música e IA.

Composição

Talvez um dos maiores problemas relativo à arte da composição utilizando-se meios eletrônicos, seja a distância existente entre compositores e computadores. Isso se deve ao fato de que músicos, além do estudo tradicional de Música, tenham que aprender a manipular outros conjuntos de símbolos quando tratando de síntese, algoritmos, linguagens, etc [ROADS, 1985]. Tal barreira começa agora a dar sinais de fraqueza e hoje depara-se com sistemas poderosos e de interfaces bastante amigáveis. Em composição, pode-se destacar duas maneiras de se integrar compositor e máquina:

- Composição Assistida por Computador (CAC). Como referência, têm-se os trabalhos de M. Malt [MALT, 1994] que trata de modelos matemáticos em sistemas estocásticos e caóticos. Outro trabalho bastante importante é o de D. Little [LITTLE, 1995], onde é proposto um sistema composicional, tendo como base algumas fórmulas matemáticas (fractais) da teoria do caos [GLEICK, 1991]. Já N. Almeida [ALMEIDA, 1995] utiliza o padrão MIDI (e sua representação numérica) no seu sistema de composição.
- Composição Algorítmica (CA). O trabalho de G. Monro [MONRO, 1996] é bastante ilustrativo para um sistema de CA. Ele descreve um ambiente, utilizando o MIDI, que possibilita compor algorítmicamente e com total controle sobre o resultado. Já para J. Manzolli [MANZOLLI, 1996], todo um ambiente gráfico (interface) já está em pleno funcionamento.

No primeiro caso, encontra-se um processo onde o compositor possui liberdade de, a qualquer momento, interromper, fazer escolhas e modificar parâmetros. Outro nome para a CAC é o de *composição interativa*. É bem visível, neste caso, a visão do computador como uma ferramenta, seja para processar uma quantidade muito grande de informação, seja criar sons em tempo real.

Já no caso da Composição Algorítmica, a máquina, além do papel de otimizadora do processo composicional, assume também o papel de compositor. Em geral, sistemas que trabalham com composição algorítmica são sistemas prontos onde, a partir de uma entrada de dados qualquer, é feito o processamento e a composição musical.

Na maioria dos sistemas de CAC ou CA, a grande barreira é a limitação do material sonoro produzido. Enquanto no ser humano a capacidade de criação é proporcional aos estímulos externos e internos que o mesmo recebe [MANZOLLI, 1996], a máquina ainda se encontra limitada aos dados iniciais.

Performance

Esta é uma das áreas da Música que mais evolui, principalmente pelos ditames do mercado. Desde o *Theremin*, de 1919, busca-se novas interfaces que possibilitem a execução de música eletroacústica em tempo real [MANZOLLI, 1995]. No início, as composições eram feitas em sintetizadores e computadores, gravadas em fitas magnéticas e executadas com a presença de um ou mais músicos humanos. Para ser executada, o grande inconveniente era o ativar do gravador e a sua sincronia com o músico humano.

Hoje a IA busca aumentar a flexibilidade e a espontaneidade do executante [ROADS, 1985]. O desafio maior em performance continua sendo a necessidade e o desejo de se fazer síntese de som em tempo real.

Teoria Musical

Quando se fala de música teórica, fala-se não só de escrita musical, mas também de análise e construção de modelos (estruturas) musicais [FRITSCH, 1994]. É evidente que nesta análise e construção de modelos não existe uma formalização matemática que permita resultados com níveis zero de erro. Não é isto que acontece com especialistas humanos e, portanto, não é este o resultado que se pode esperar. Isto se deve ao caráter semi-formal da música que permite a codificação de modelos³⁶ em análise, mas quase sempre não produz um bom modelo ao se basear apenas em análises. O fato mais marcante é o inter-relacionamento da Música com outras ciências, tais como, Estética (juntando nesta a Física, Psicologia, Estatística e a Sociologia) e a História, contando as peculiaridades de cada época [FERNEDA, 1994; DANNENBERG, 1994; TEIXEIRA, 1996]. Somente com tais subsídios pode-se fundamentar análises e codificações de modelos.

Processamento Digital de Som

Desde os primórdios das pesquisas dedicadas à síntese sonora, os músicos acabam se deparando com um trabalho, normalmente, bastante cansativo e frustrante [FRITSCH 94]. Em detrimento ao grande avanço tecnológico atual (com máquinas cada vez mais poderosas, acessíveis e com grande capacidade de armazenamento de informação), ainda não existem interfaces tão amigáveis quanto desejariam os músicos. Neste caso, a IA pode sugerir técnicas de representação do conhecimento voltadas à síntese sonora. Tais representações, além de diminuir a distância entre músicos e máquinas, devem ser capazes de

³⁶ Esta codificação se dá no sentido de se determinar as principais características de uma determinada época, através da utilização de efeitos musicais. Uma coisa que não se deve deixar de lado, é justamente a característica básica de todas as artes: inovação por quebra de regras [FERNEDA, 1995].

criar sínteses interativas em tempo real [FRITSCH, 1994; MIRANDA, 1994].

Educação Musical

Hoje em dia, está cada vez mais se popularizando uma gama muito grande de *softwares* auto-instrucionais (também conhecidos como *courseware*). Estes cursos têm como objetivo possibilitar que um aluno aprenda de forma individualizada [BRANDÃO, 1995].

O desafio proposto à IA é o de aperfeiçoar estes tipos de programas. Uma das maneiras de se conseguir isto é tornar a máquina sensível às resposta do aprendiz. Através de diversas sessões entre a máquina e o aprendiz, pode-se detectar estratégias de comportamento do sistema, de maneira a descrever determinados caminhos didático-pedagógicos previsíveis entre estes contatos [COSTA, 1996].

Em resumo: enquanto estiver sendo utilizado, o sistema estará ensinando um determinado domínio, ao mesmo tempo em que estará se adaptando para oferecer a melhor técnica de ensino para uma determinada situação. A seguir, este tema é explorado mais detalhadamente.

3.4 Sistemas Baseados em Conhecimento dedicados à Música

Neste tópico, são feitas algumas considerações sobre sistemas dedicados à Música, e que utilizam o Conhecimento como “ingrediente” essencial para que sejam considerados sistemas inteligentes. São eles:

- (i) sistemas de análise musical e
- (ii) sistemas tutoriais para educação musical.

Sistemas de Análise Musical

Estes tipos de sistemas possuem como característica principal a análise de um ou mais aspectos musicais. O caráter multifacetado da Música dificulta bastante as tentativas de englobar diversas abordagens ao mesmo tempo. Aqui, mais uma vez, o grande limitador é ter ou não uma representação poderosa que consiga, não apenas tratar dos aspectos que fazem referências gerais mas, também, tratar das exceções.

Uma maneira de garantir que um determinado sistema tenha domínio sobre mais de um aspecto musical é limitá-lo a trabalhar dentro do universo de obras de um determinado compositor e/ou estilo musical.

Dentro desta ótica, pode-se destacar o trabalho produzido por F. Pachet e R. Mouton [PACHET, 1991(a), 1991(b); MOUTON, 1995]. No trabalho de Pachet, o enfoque representacional é dado, tendo como ponto de convergência um estilo bem definido, no caso o *Jazz*. Além disso, apenas o aspecto de análise harmônica é contemplado. Mais especificamente ainda: dentro da análise harmônica, o que é relevante são as seqüências de acordes utilizados por músicos de *Jazz*³⁷.

Sistemas Tutoriais para Educação Musical

Ao tratar de questões relativas ao ensino, seja de qualquer área de conhecimento humano, a primeira preocupação deve ser a de determinar quais estratégias didáticas devem ser utilizadas. Quando se passa da relação aluno-professor para uma relação máquina-aluno, não se deve ter em mente apenas tais preocupações mas, também, tornar o sistema o mais interativo possível.

Esta forma de interação pode vir através da utilização, em se tratando de Música, de recursos multimídia [MARTÍN, 1995]. O trabalho de M. Brandão e M. Sambuichi [BRANDÃO, 1995] é um exemplo de software auto-instrucional que ensina teoria musical. Apesar de utilizar de muitos recursos multimídia, este programa não permite uma maior interação com o usuário.

Uma proposta um pouco mais ambiciosa é a de E. Costa [COSTA, 1996]. A idéia de Costa é a de construir um sistema capaz de tratar os erros comumente cometidos pelos alunos e tratá-los de maneira independente. Outra proposta interessante do mesmo autor é fazer com que o sistema armazene novos erros e/ou vícios durante a aprendizagem quando da tentativa de resolvê-los³⁸.

³⁷ Vide item 4.4.

³⁸ Maiores detalhes, vide o capítulo 6.

4 Trabalhos Correlatos

4.1 Introdução

Com o aumento do volume de pesquisas nos últimos 20 anos, encontra-se hoje um número bastante razoável de sistemas prontos e/ou em constante aprimoramento. Evidentemente que algumas áreas foram mais exploradas do que outras (pode-se citar as áreas de Composição, Musicologia e Análise como bastante promissoras em termos de pesquisas em Música e IA) [MIRANDA, 1990], não apenas por possuir atrativos em termos de desafio, mas por possuir um forte apelo mercadológico e aqui, em particular, as dezenas de sistemas comerciais para edição de partituras, seqüenciamento, síntese, etc [RATTON, 1992].

Hoje, encontram-se no mercado vários *softwares* com diversas aplicações em Música. Desde os que se dedicam a executar pequenas seqüências musicais, passando pelos editores de partituras até chegar a sistemas complexos que permitem ao usuário possuir em seu microcomputador um estúdio de gravação totalmente digital [AIKIN, 1997]. A maioria destes produtos foram provenientes de trabalhos realizados em universidades e laboratórios de pesquisa.

Como a área de interesse em Informática aqui abordada é a Inteligência Artificial, são relacionados abaixo alguns sistemas e/ou propostas que utilizam técnicas de IA em alguma das diversas áreas musicais (vide Anexo 1).

IA e Musicologia

- “*A Symbolic-numeric Approach to Quantization in Music*”: este trabalho, proposto em [BEL, 1996], é voltado à *quantização*³⁹ onde a mesma é orientada através de representações textuais em seus diversos níveis de abstração, na tentativa de melhorar a correção da duração e ataque das notas.
- “*Machine Learning of Sound Attributes: Computer-Assistance in Concept*”: neste sistema o interesse é a composição, conjuntamente com um sistema de síntese [MIRANDA, 1996].

³⁹ Segundo [RUBIN, 1996], quantização é “o processo pelo qual o tempo de uma nota é alterado para alinhar-se à subdivisão métrica mais próxima que foi escolhida, tal como uma semínima ou colcheia”.

Psicoacústica

- “*Representing Musician’s Actions for Simulating Improvisation in Jazz*” [RAMALHO, 1996]: este trabalho é baseado no conceito de memória musical e está fundamentado no princípio do Raciocínio Baseado em Casos [SCHANK, 1989].

Análise Musical

- “*Aquisição do Conhecimento em Harmonia: Um Ambiente de Aprendizagem*”: a proposta deste trabalho [TEIXEIRA, 1995] é a de um ambiente voltado à análise das etapas de aprendizagem de um aluno artificial (máquina), quando interagindo com um especialista humano. Aqui, a aquisição de novos conhecimentos é incremental e é baseado no protocolo de aprendizagem MOSCA [REITZ, 1992].

Como a especificidade deste trabalho é em ensino de Música, são descritos a seguir alguns sistemas relativos a esta área. Dos sistemas escolhidos, um é dedicado ao ensino de performance (piano), outro ao ensino de teoria, e os três últimos estão relacionados à Harmonia (análise harmônica, treinamento de intervalos e representação voltada à Harmonia Tonal).

4.2 Expert Piano

Expert Piano é um ambiente computacional e educacional para auxiliar o estudo de piano e Música. Proposto por Glanzmann [GLANZMANN, 1995], *Expert Piano* tem como objetivo o treinamento e aperfeiçoamento da técnica pianística, possibilitando a análise de performance em busca de detectar e corrigir possíveis erros de execução por parte de um aluno. Além disso, este sistema possui informações complementares sobre as obras musicais e seus autores, e propicia o acompanhamento de aproveitamento do aluno, através do armazenamento de dados.

O sistema é constituído fisicamente por um teclado, uma interface MIDI, um microcomputador, caixas de som e uma impressora. A sua utilização deve ser feita somente com a supervisão de um professor especialista.

Expert Piano oferece as seguintes funcionalidades [GLANZMANN, 1995]:

- *escolha das opções de estudo*: podendo, o usuário, tocar ou ouvir uma determinada obra musical;
- *diagnóstico e análise de erros*: após a execução por parte do aluno, o sistema está

apto a apontar os erros cometidos durante a interpretação (comparação entre os dados fornecidos pelo aluno e os armazenados na base de conhecimento)

- *apresentação de remediações e sugestões*: de acordo com as suas performances anteriores e às características da obra, *Expert Piano* apresenta informações apropriadas para a solução dos erros (recursos visuais e auditivos);
- *relatórios de desempenho*: apresenta os compassos tocados com erro (em notação musical tradicional) e os compara com os compassos armazenados;
- *acesso à informações complementares*: em formato de hipertexto, aqui o aluno encontra informações que vão desde dados sobre vida e obra dos compositores até as especificidades da obra em questão e, por último,
- *conclusão do processo*: aqui o aluno pode se dar por satisfeito ou reiniciar o processo e se reincidente nos erros, o sistema apresentará novas remediações e sugestões.

As técnicas de IA são utilizadas na fase de diagnóstico e análise de erros, incorporando as principais características dos STI's, juntamente com os subsistemas de aquisição do conhecimento, capazes de realizar a manutenção da base de conhecimento.

Um fator que pode prejudicar o desempenho deste sistema é a utilização de teclados musicais no ensino de piano (excetuando-se os teclados com teclas balanceadas), que acabam por diferenciar dos pianos em termos de técnicas (peso das teclas, tempo de resposta, ressonância, etc).

Por se tratar de um sistema voltado à performance musical, a sua representação musical é baseada em MIDI, sendo esta utilizada para a captação e comparação de informações, o que nem sempre condiz com a realidade em termos de teoria musical. No item seguinte, são vistas com mais detalhes as questões que envolvem a utilização do MIDI como representação e as suas implicações no ensino de teoria.

4.3 Classificação Harmônica Tonal

Neste trabalho, proposto por Beckenkamp [BECKENKAMP, 1994, 1995], utiliza-se redes neurais para a classificação harmônica tonal. Afirma-se que o sistema é capaz de:

- (i) classificar acordes a partir de um trecho musical dado (percepção cognitiva) e
- (ii) validar a aplicação de redes neurais na domínio de cognição em música.

O algoritmo utilizado é o de *Back-propagation* e está baseado em D. Rumelhart e J. McClelland [RUMELHART, 1986]. Este sistema é composto por três níveis de neurônios (camadas), são eles:

- *entrada*: composta por 12 neurônios, sendo estes associados a cada uma das notas da escala cromática. Quando uma determinada nota é tocada, seu respectivo neurônio é ativado.
- *interno* (escondido): nesta camada tem-se um total de 24 neurônios correspondendo, segundo Beckenkamp, às 24 tríades (maiores e menores) da escala tonal ocidental.
- *saída*: composto por três neurônios que, de acordo com seus valores, indicam o tipo de acorde em relação às notas de entrada (acordes maiores, menores, diminutos ou aumentados). Esta saída, posteriormente, foi aumentada para cinco neurônios permitindo o tratamento de acordes de sétima e de nona.

Ainda segundo Beckenkamp, a entrada das notas na primeira camada pode ser feita em tempo real ou através de um seqüenciador. Em ambas as opções os dados são armazenado a partir de informações MIDI.

Finalizando, o autor afirma que após o treinamento da rede, o percentual de acerto foi da ordem de 100%.

As críticas cabíveis aqui são devidas a alguns aspectos musicais e representacionais. A primeira tem como base, como visto no capítulo 2, a não existência de apenas doze notas musicais relevantes. Este total é de trinta e cinco notas, respeitando-se o conceito de enarmonia inerente à escala cromática temperada.

Por este motivo, fica comprometido o uso de apenas doze neurônios de entrada (notas). Quando da introdução dos dados, mesmo que na representação gráfica (partitura) exista notas diferentes (enarmônicas). Na prática existirá um único neurônio respondendo por até quatro representações distintas. Evidentemente que se os exemplos forem restritos a peças simples (sem notas alteradas, etc), poderá ter um resultado satisfatório. Se no entanto, diante da expansão das possibilidades (notas estranhas, modulações, etc) há uma grande risco das respostas não corresponderem à análise feita por um especialista humano.

Uma solução para tal problema é relativamente simples e em dois caminhos:

- (i) acrescentar as notas enarmônicas, aumentando de doze para trinta e cinco o número de neurônios de entrada e

- (ii) não utilizar com entrada apenas as informações MIDI (representação pobre). O MIDI, neste caso, possui um componente forte que é o *feedback* sonoro, mas ele, por si só, não representa determinados parâmetros importantes (um deles é a enarmonia), ficando restrito à reprodução sonora.

Como se trata de um sistema voltado à análise harmônica, conjuntamente com um paradigma conexionista, os seus mecanismos de resposta ficam ofuscados e que, por se tratar de análise, deveria ser mais transparente ao usuário.

Para este trabalho, o sistema de Beckenkamp deixa de lado aspectos musicais importantíssimos. Além desse aspectos, o próprio tratamento através de redes neurais dificulta o entendimento do processo de aprendizagem e, conseqüentemente, a sua utilização em um sistema preocupado com questões didáticas [ANDRADE, 1995].

4.4 SETMUS

SETMUS (Sistema Especialista para Teoria Musical), é um sistema totalmente voltado ao ensino de teoria musical. Este trabalho, proposto por Fritsch [FRITSCH, 1995] está implementado em *Hypercard* [GOODMAN, 1990] (para Macintosh) e utiliza o paradigma de orientação a objetos.

A proposta básica deste sistema é o ensino de elementos de teoria musical, tais como escalas e arpejos⁴⁰. Aqui o aluno, segundo Fritsch, pode provar seus conhecimentos teóricos e treinar a sua percepção, substituindo o conhecimento do professor (nos exercícios) e acumulando a função de livro didático.

Pela característica do ambiente *Hypercard*, cada escala e cada arpejo é um cartão (objeto) independente. Para cada objeto, os quais o aluno se propõe a aprender, abre-se uma janela onde aparece a sua representação gráfica (partitura) e que também possibilita a sua execução sonora.

Diante do computador, o aluno pode selecionar, por exemplo, uma escala e, a partir daí, percorrer um caminho didático que pode levá-lo a diversas possibilidades, entre elas:

- “tocar”, através do *mouse*, uma seqüência de notas (escalas ou arpejos) e pedir o reconhecimento desta por parte do sistema;
- se o que foi “tocado” fizer sentido, o SETMUS pode dar mais detalhes sobre a es-

⁴⁰ *Arpejo* são as notas de um acorde quando tocadas sucessivamente (nota após nota) ao invés de simultaneamente.

estrutura reconhecida. Caso contrário, o sistema instiga o aluno a questionar o porquê da não existência do objeto;

- através de perguntas e respostas (existentes na base de conhecimento) o sistema pode dirimir as dúvidas do aluno;
- a aluno pode pedir que seja executada a estrutura analisada (função *playback*). Isto permite, no caso de erros, não apenas a comparação gráfica dos itens anteriores, mas também a comparação auditiva;
- por fim, a aluno tem à sua disposição uma “Calculadora Musical” que lhe permite formar as escalas tendo como base o nome das notas, suas alterações (sustenidos e bemóis), seu modo e se será uma escala ou um arpejo.

Apesar de sua estruturação bastante coerente, em termos de representação do conhecimento utilizada, SETMUS não contempla a utilização das duas alterações simultaneamente, nem o uso dos dobrado-sustenidos e dos dobrado-bemóis. É evidente que aqui está considerado o ensino de conceitos elementares e que a falta dos objetos acima não implica em depreciação do sistema. No entanto, esta representação fica um pouco distante do universo mínimo de conhecimento musical para o trabalho em Harmonia, além de sua atuação sonora ser restrita à monofonia⁴¹.

4.5 MusES

Um outro trabalho bastante relevante é o proposto por F. Pachet [PACHET, 1994(b)]. Seu sistema, *MusES System*, é um sistema especialista totalmente dedicado às técnicas de representação do conhecimento em Harmonia Tonal. O MusES foi desenvolvido em Smalltalk (utilizando o conceito de programação orientada a objetos) e, similarmente a este trabalho, trata especificamente de um sistema de representação do conhecimento musical relevante ao domínio da Harmonia.

A maior contribuição do MusES, no que diz respeito à representação do conhecimento musical, é o estudo das variadas técnicas para capturar o essencial de cada estrutura musical [PACHET, 1994(b)]. Desta forma, três projetos paralelos baseados no MusES foram desenvolvidos. São eles:

- *Análise de Sequência de Acordes*: aqui a representação volta-se à análise de acordes utilizados em música contemporânea (Jazz).

⁴¹ Música constituída por uma única linha melódica, sem acompanhamento.

- *Harmonização Automática e Satisfação de Restrições*: aqui, tenta-se capturar regras musicais dos tratados de Harmonia e Contraponto. este trabalho vem ao encontro do que é proposto em [TEIXEIRA, 1995; COSTA, 1996], onde mecanismos inteligentes de busca são contrastados com os mecanismos de satisfação e restrição [PACHET, 1994(a)].
- *Simulação de Improvisação Jazzística em Tempo Real*: este basea-se na idéia de memória musical [RAMALHO, 1994].

Dois fatores contribuíram para que MusES seja considerado um poderoso sistema de representação:

- (i) a consciência musical por parte do autor e
- (ii) a necessidade de uma representação que contemple os principais aspectos envolvidos na resolução de determinados problemas harmônicos.

Como resultado, tem-se a representação de enarmonias (com suas trinta e cinco notas possíveis), intervalos, escalas e acordes, onde a temática principal é a análise e a geração automática de harmonizações e improvisações.

4.6 Considerações Finais

O que é proposto no capítulo seguinte é uma representação que possui características muito próximas ao MusEs (no que diz respeito às preocupações de representação em nível teórico). O diferencial aqui é a abordagem declarativa, sendo utilizada a linguagem Prolog++ (LPA-Prolog) como ferramenta de especificação e implementação.

Aqui, a preocupação está voltada ao ensino de Harmonia e a capacidade de permitir a manipulação do conhecimento de acordo com as necessidades existentes entre o mestre e o aprendiz. Ao se trabalhar com didática e aprendizagem, um outro caminho deve ser trilhado: o da representação estruturada em níveis técnicos necessários ao ensino. Neste caso, a representação proposta, dentro da ambiente em relevância no capítulo 6, permite a manipulação do conhecimento em consonância a estes níveis técnicos.

Em resumo: as sessões de ensino devem refletir, a cada etapa, um conhecimento próprio e completo, sendo que a representação inserida neste esquema deve assimilar os detalhes desta forma de trabalho. Para tanto, no capítulo seguinte são descritas as estruturas de representação envolvidas nestes aspectos didáticos-musicais e, em seguida, no capítulo 6, é exposto o modelo de ambiente onde a representação será utilizada.

5 Representação do Conhecimento Musical

5.1 Introdução

Conforme foi argumentado no capítulo 4, a ausência de representações voltadas ao ensino de Harmonia foi a mola impulsionadora deste trabalho. Esta motivação, somada com a experiência de seu autor, no que se refere diretamente ao ensino de teoria harmônica, e conjuntamente com as técnicas oferecidas pela IA, possibilitaram a construção da representação presente neste capítulo.

Como estratégia de trabalho, o material tradicionalmente utilizado no ensino de Harmonia (exercícios específicos, conceitos formais, etc) foi avaliado com o intuito de dar uma maior abrangência no que se refere à representação dos conceitos, objetos e estruturas encontradas neste domínio. Como comprovação deste caminho, logo após a implementação (código) das estruturas relevantes em linguagem LPA-Prolog++ [VASEY, 1995], o programa gerado “retornou” à sala de aula para a correção de exercícios aplicados, demonstrando positivamente a sua eficácia.

Neste capítulo, em uma visão mais geral, são feitas algumas considerações sobre a representação do conhecimento musical formal. Em seguida, é apresentada a estrutura geral dos conceitos musicais (objetos) e seus relacionamentos. O próximo passo é a definição formal dessas estruturas e objetos, constituindo assim a representação dos conceitos fundamentais. Na seqüência, são apresentadas as diferentes dimensões do conhecimento conforme o que está previsto na especificação do ambiente MATHEMA [COSTA, 1996], tendo como parâmetros o conhecimento musical antes representado e a sua adequação estrutural, dentro deste modelo filosófico-computacional de conhecimento.

5.2 Aspectos sobre a Representação de Objetos Musicais

Para demonstrar a complexidade e a estruturação da Música, mais particularmente da Harmonia, basta analisar seus componentes (vide capítulo 2). Percebe-se logo de início que os objetos musicais estão posto em dois grupos distintos, *formal* e *cognitivo* [DANNENBERG, 1993], que respectivamente são:

- *os objetos que podem ser tratados matematicamente*: que em geral são os que po-

dem ser dispostos em uma partitura, por exemplo: notas, tempo, harmonia, alterações, tonalidades, frequências, etc e

- *os objetos que se encontram no domínio do perceptivo*: aqui cabem todas as informações que, por sua natureza, não podem ser organizadas em uma folha de papel. Pode-se citar como exemplos as questões de tensão e relaxamento, expectativa, emoção, contexto histórico e performance.

Estes dois grupos interatuam de forma a se complementarem [FERNEDA, 1994]. Mesmo quando das tentativas de representação da parte formal, os pesquisadores sempre buscam, na medida do possível, levar em consideração os aspectos, em geral estudados em Musicologia, Análise e Composição, que não “aparecem” explicitamente na partitura. Separar uma coisa da outra, genericamente, pode terminar por resultar em uma “caricatura” do domínio alvo.

Como consequência deste relacionamento pode-se presumir os mais variados níveis de representação que podem ser encontrados em uma obra musical. Isso se dá pela forma como necessita ser direcionada uma representação, a depender do objetivo (relevância), onde alguns aspectos devem ser colocados em segundo plano e/ou ignorados (abstração) [WIGGINS, 1993].

Embora este trabalho contemple os objetos do primeiro grupo, existe a preocupação em tornar disponível esta representação de forma a permitir seu uso em estágios mais avançados⁴² do ensino de Harmonia. Aqui é cabido dotar a representação de mecanismos que façam referência a diversos contextos [TEIXEIRA, 1995].

A conclusão disso tudo é que, embora se trabalhe com determinadas partes formais do universo musical, há a necessidade de não negligenciar a existência de todo um domínio informal (cognitivo) que é de suma importância na resolução de determinados problemas [WIDMER, 1992].

Representação de estruturas musicais

Após a separação dos elementos desses dois grupos (formal e cognitivo), o passo seguinte é definir a *estruturação e hierarquização* da representação.

⁴² Aqui a palavra estágios pode ser associada às etapas pelos quais o aluno de Harmonia deve trilhar. Da mesma maneira que um professor demarca fronteiras, a representação deve limitar determinados objetos ou atributos de acordo com a necessidade da etapa estabelecida.

Por *estruturação* entende-se a organização do material sonoro, de maneira a compor estruturas que encerrem em si mesmas um significado completo [DANNENBERG, 1993]. Estruturas menores podem continuar gerando outras estruturas, sempre respeitando um sentido estético e formal. Já a *hierarquização* é a disposição lógica dessas estruturas onde pode haver dependências, abstrações ou heranças entre elas.

Um outro cuidado a ser tomado é resultado da aparente simplicidade da informação musical. Esta pode levar à criação de estruturas aparentemente não tão complexas e por isso deve-se esquematizá-las de maneira a permitir que as mesmas possam ser estendidas posteriormente. O conceito de tonalidade, por exemplo, pode ser visto de forma equivocadamente simples (como a definição de uma escala por exemplo). Por outro lado, a definição de tonalidade implica na definição de uma nota inicial, um conjunto de intervalos predefinidos, um coleção fechada de acordes e uma escala também formada por intervalos predefinidos. Dependendo do ponto de vista, pode-se restringir inconscientemente uma estrutura, deixando sua representação aquém das necessidades reais.

Como passo inicial, deve-se definir as primitivas e como estas devem ser estruturadas. Já aqui, um descrição puramente teórica pode produzir as “caricaturas” anteriormente referidas. Portanto, o conhecimento cognitivo deve servir de *interprete* do formalismo utilizado, na intenção de definir o que é adequado ou não, ou se o potencial teórico está bem aplicado.

É justamente neste momento que se faz a separação nos dois grupos descritos no tópico anterior (representação de natureza teórica e representação de natureza cognitiva). Estas representações irão assumir diferentes perspectivas, de acordo com um domínio musical específico [HONING, 1993]. Neste ponto, deve-se compreender que dificilmente existirá uma representação ou conjunto de representações que consiga envolver todas as esferas de ação da Música. Para cada domínio devem ser considerados diferentes aspectos e, a depender das ações, a parte formal pode ser preponderante em relação à parte cognitiva e vice-versa.

Abaixo são analisadas algumas das áreas do conhecimento musical (mais especificamente em Análise e Produção Musical) as quais a representação formal (teoria), segundo Honing [HONING, 1993], prepondera em relação à representação cognitiva:

Musicologia

O desenvolvimento e as adaptações de maneiras diferentes de representação da notação

musical, surge com a especificação de determinadas teorias de análise musical. Como ponto comum, a maioria dessas teorias concordam com o pressuposto de que a música encontra-se na partitura. Sobre isto, pode-se dizer que, no geral, são poucas as obras musicais que podem ser caracterizadas única e exclusivamente de acordo com o que está em consonância com a partitura [GOODMAN, 1968].

Com relação a definir onde “está” a Música, se no papel, no ar, em quem a ouve ou apenas na mente das pessoas é uma questão de considerá-la ou não como cognitiva.

Música Computadorizada

O interesse nesse campo é o de criar estruturas de dados, capazes de fornecer uma base a sistemas de composição, sejam estes interativos ou não. Neste caso, é de suma importância a distinção entre sistemas em tempo real ou não. Como exemplo de representação com tais propósitos, tem-se o MIDI [IMA, 1983]. Além de ser um protocolo de comunicação entre instrumentos musicais digitais e computadores, o MIDI pode ser considerado como uma representação musical (ainda que em um nível muito baixo). Em geral, muitos dos sistemas dedicados à composição se utilizam do MIDI no que diz respeito à execução sonora, através de representações em níveis mais altos (interfaces gráficas, etc).

Edição Musical e Sistemas de Recuperação

Aqui diversas propostas de padronização são encontradas, frutos de pesquisas, no que se refere à edição musical. Como resultado dessas tentativas, o ANSI (*American National Standards Institute*) publicou a mais recente tentativa de criação de uma especificação técnica e metodológica para a uniformização de uma linguagem de descrição [ANSI, 1989].

Da mesma forma que outras tentativas de padronização, o resultado está mais para pragmatismos (eficiência em termos de tamanho e velocidade) do que em relação a consistências e generalidades [HONING, 1993].

Ao se considerar a área de *Educação Musical* vê-se que o conhecimento formal está mais intimamente atrelado aos primeiros contatos de um aprendiz com a Música, e que o acúmulo destes será o responsável pela geração do conhecimento considerado cognitivo.

5.3 Estrutura da Representação

O que se deseja com a representação aqui considerada, é a sua utilização voltada ao ensi-

no de Harmonia. A principal preocupação desta representação é a de prover mecanismos que possam suprir as necessidades de ensino e aprendizagem em Harmonia.

A Harmonia é um exemplo típico de domínio, onde se constata a presença de objetos estruturados. Por este motivo, a primeira providência foi identificar quais elementos musicais são preponderantes na resolução de problemas harmônicos. Nesta primeira fase, a preocupação foi a de isolar as menores partículas musicais e, partindo delas, ir construindo os objetos subseqüentes por ordem de complexidade. O conhecimento de apoio ao entendimento de cada objeto representado a seguir encontra-se descrito no capítulo 2.

De início, pode-se perceber que as notas musicais, conforme definição do item 2.2.1, são compostas por freqüências fixas e definidas. Para que estas freqüências sejam utilizadas musicalmente, são necessários alguns conceitos musicais complementares em sua significação. Na Figura 5.1 são mostrados os parâmetros necessários à definição de uma nota, onde os elementos musicais relativos à freqüência estão contidos no retângulo (composto por nome, oitava e alteração).

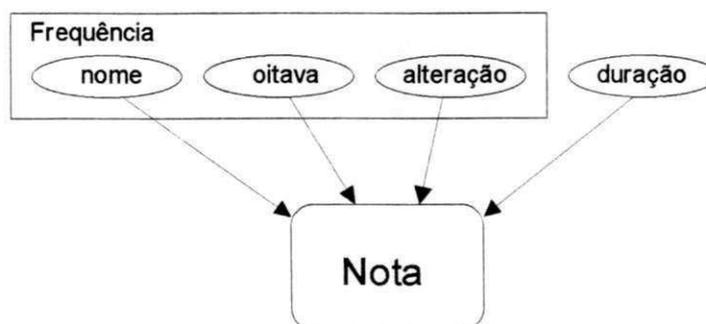


Figura 5.1: atributos de uma nota

Por sua composição a *nota* possui, em si mesma, uma certa complexidade (encapsulamento), pois a ausência de alguns de seus elementos pode, por exemplo, não ser relevante em um dado momento e essencial em outro. A parte temporal, como se vê na figura acima, fica separada dos atributos de freqüência.

No caso da pausa, a duração é o seu único atributo que, no momento, ficará em segundo plano nesta representação. Portanto, no decorrer da ampliação de um sistema dedicado ao ensino de Harmonia, tal ingrediente torna-se importante e deve ser levado em consideração. Por ora, deve-se saber que com estes componentes é possível situar uma nota qualquer dentro de todo o espectro audível (vide Figura 2.2).

Colocando o objeto nota como base das estruturas musicais, tem-se como resultado o fluxograma apresentado na Figura 5.2, com a dimensão das relações entre os vários objetos

desse domínio. Aqui a hierarquia intrínseca vai do objeto mais simples (estruturalmente) até o mais complexo.

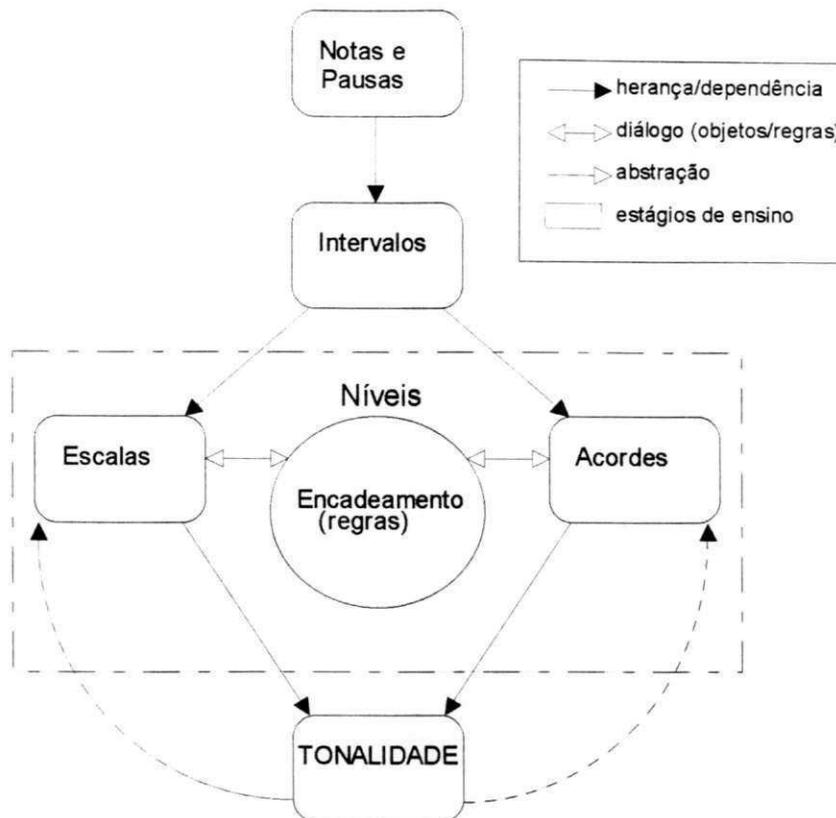


Figura 5.2: estruturação dos objetos musicais relevantes à Harmonia

Seguindo esta ordem encontra-se:

- que os *Intervalos* estão em relação de dependência e de herança horizontal⁴³ com o objeto *Notas*;
- que as *Escalas* e *Acordes* estão em relação de dependência e herança horizontal com o objeto *Intervalo* e que de acordo com sua disposição (vertical ou horizontal) gera um ou outro,
- que alguns atributos e métodos de *Escalas* e *Acordes* são definidos pelo objeto *Níveis*, que representa os vários estágios de aprendizagem,
- que a relação entre *Escalas* e *Acordes* com o objeto *Encadeamento* se dá através de métodos (regras) de manipulação e tratamento;

⁴³ Entende-se por *herança horizontal* a relação onde o valor de um atributo de um objeto é definido pelo valor deste mesmo atributo em um dos componentes do objeto. Por exemplo, a definição de um *Intervalo* pressupõe a definição de duas notas e seus atributos intrínsecos e, conseqüentemente, o objeto *Intervalo* herda horizontalmente os atributos de *Notas*.

- que *Escala* e *Acordes* definem o objeto *Tonalidade* ao mesmo tempo que são uma abstração desta última e
- que *Tonalidade* define um conjunto singular de *Acordes* e uma única *Escala*.

De maneira similar, e principalmente em relação aos acordes, talvez o objeto de maior interesse em Harmonia, a sua estrutura deve permitir certas possibilidades de inferência, tais como:

- a partir de um acorde incompleto chegar às notas faltantes,
- de um acorde complexo deduzir sua construção básica (sem dissonâncias, etc),
- construir todas as possibilidades de inversões de um acorde,
- localizar, dentro de uma tonalidade, qual a função de determinado acorde.

Torna-se evidente que para cada um dos exemplos acima, quando tratados por uma representação orientada a objetos, cada “metamorfose”, ou variação, deve ser vista como objetos independentes. Esta preocupação é necessária pelo fato de que tal complexidade deve ser encarada com um certo desvelo, não só pelo motivo da representação em si, mas, principalmente, pelas necessidades inerentes ao ensino e aprendizagem de tal matéria, como dito anteriormente.

Ainda dentro do objeto *Níveis*, tem-se o objeto *Encadeamento*. Este é responsável pelas regras que irão guiar as diversas possibilidades de resolução de um encadeamento (vide item 2.3.5). Mais uma vez os mesmos níveis de aprendizado impostos a um aprendiz humano devem ser ressaltados: neste caso, determinadas regras podem ser relevantes em um instante e, em outro, podem ser relegadas a um segundo plano.

Algumas considerações sobre o objeto *Tonalidade*: a *Tonalidade*, em um determinado instante do ensino de Harmonia, possui uma função básica: definir quais acordes e quais escalas farão parte de um problema. De posse dessa informação, pode-se de cara, eliminar uma grande parte do conjunto de acordes e escalas quando da resolução de um exercício.

Outrossim, a *Tonalidade* serve como limitador de possibilidades em nível didático. Dessa forma, pode-se visualizar as relações entre *Tonalidade*, *Escala* e *Acordes* da maneira vista na Figura 5.3, onde *E* é uma escala específica, *A*'s os acordes da tonalidade e *C*'s as cifras implícitas a estes acordes. Aqui, como dito anteriormente, a determinação de uma tonalidade fornece um subconjunto de acordes (e suas cifras respectivas) e uma escala; já a existência de uma determinada escala e/ou acordes nos remete a uma tonalidade

específica.

As *cifras* estão associados aos *acordes* e são responsáveis por determinar as funções destes. Dessa maneira, um *cifra* pode estar associada a diversos *acordes*, sendo que a definição biunívoca entre *acordes* e *cifras* é especificada pela *tonalidade*.

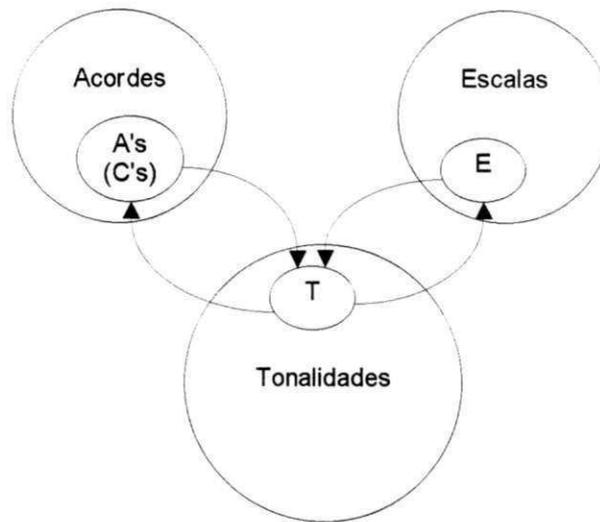


Figura 5.3: relacionamento entre os objetos Tonalidade, Acordes e Escalas

Algumas considerações sobre o objeto *encadeamento*: a Figura 5.4 é a ampliação de *encadeamento*. Pode-se observar a existência de dois conjuntos distintos de regras que devem ser observadas na resolução de problemas (vide 2.3.5):

- as *regras de encadeamento* propriamente ditas: são aquelas que regem a maneira como devem se relacionar as seqüências de acordes,
- as *regras de distribuição* de vozes: responsáveis por manter cada melodia gerada com as regras de encadeamento dentro de parâmetros estéticos aceitáveis (melodias coerentes, dificuldades de execução por um cantor, extensão e respeitar certos relacionamentos entre as demais vozes).

Como mostra a figura, estes dois conjuntos se inter-relacionam e, com relação à proporção, pode-se dizer que as regras de distribuição são em número menor que as regras de encadeamento. No escopo deste trabalho, não se tem a intenção de representar o universo das regras de encadeamento. O motivo é que, como antes dito, de acordo com o encaminhamento dado pelo professor, pode-se acrescentar ou suprimir um conjunto determinado de regras.



Figura 5.4: objeto encadeamento

5.4 Representação dos Conceitos Fundamentais

Na Figura 5.5 é apresentado o diagrama de comunicação entre as diversas classes e seus métodos. A intenção deste diagrama é a de mostrar um resumo das comunicações entre as diversas classes, cada seta indica a chamada de um método por outro, estando ambos em classes diferentes.

Neste tópico são conceituados os objetos, dentro da escrita musical tradicional, que são relevantes ao estudo da Harmonia. A seqüência de apresentação dos mesmos obedece à mesma seqüência encontrada no capítulo 2.

O ambiente de especificação coincide com o de implementação, tendo sido escolhido o LPA-Prolog ++ [MOSS, 1995; VASE, 1995], linguagem híbrida que une recursos dos paradigmas de orientação a objeto e programação em lógica⁴⁴.

⁴⁴ Uma pequena explanação sobre as linhas código a seguir:

- os componentes de um atributo de uma classe são declarados separadamente (atomização),
- antes da descrição de um método é necessária a definição de seu tipo (se privado ou público) e da quantidade de parâmetros necessários,
- a chamada de um método a uma classe específica é feito através do uso de "<-", por exemplo: `classe<-método`, onde o método que chama deve estar localizado na classe chamada,
- o termo `copy_term` é usado para copiar um termo específico para um determinado valor de objeto, por exemplo: `copy_term(bossa-nova, estilo)`, onde o termo bossa-nova é atribuído ao objeto estilo,
- utiliza-se a exclamação "!", dentro de um método, como símbolo de parada obrigatória,
- a expressão `X = . . [Título|[Texto]]`, é utilizada para se fazer a separação entre os elementos de uma lista (texto) e seu cabeçalho,
- a primeira linha de cada método será considerada após o sinal ":-".

Vale lembrar que, em todas as entradas dos métodos que se seguirão é possível deduzir qualquer um de seus elementos, sejam estes considerados dados de entrada ou de saída.

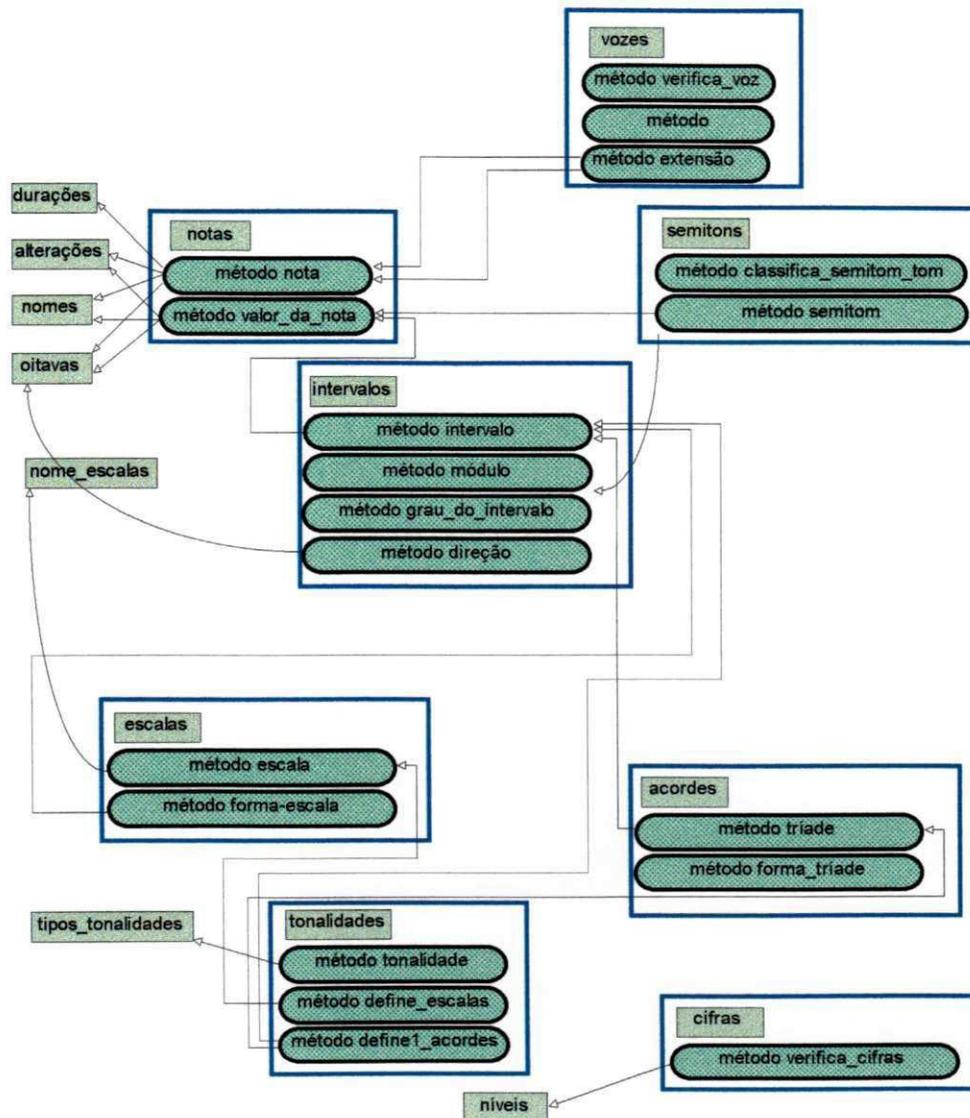


Figura 5.5: diagrama de comunicação entre métodos de classes diferentes.

Esta escolha se justifica, por um lado, pelo caráter declarativo desta linguagem e, por outro lado, pela necessidade de uma representação hierárquica do conhecimento e de mecanismos capazes de inferir conhecimento a partir de dados incompletos. A opção feita pela Programação em Lógica é devida a sua capacidade de inferir conhecimento a partir de dados incompletos. Esta capacidade será bastante utilizada em todos os métodos descritos nas classes a seguir.

Motivado pela maneira encontrada na organização dos elementos musicais, juntamente com a necessidade de garantir que a manipulação de alguns atributos preponderassem em relação aos demais; optou-se pela utilização do paradigma de *Orientação a Objetos* (OO). A consistência do paradigma de OO aplicado à Música pode ser avaliada em diversos trabalhos [POPE, 1991]. Aqui, os benefícios de modularização, encapsulamento e especifica-

ção hierárquica de componentes foram preponderantes na escolha da OO. A consequência dessa escolha e de seus benefícios assemelhasse muito a um sistema que tenha como paradigma a utilização de *frames*.

Com respeito às demais linhas de código, estas são comentadas nos textos que seguem abaixo. O Anexo 2 contém o código da representação em sua versão integral.

5.4.1 Notas e Pausas

Na Figura 5.1 está explícito que toda nota musical é composta por uma lista de parâmetros que necessitam ser nomeados e representados. Esta lista é composta pelos seguintes atributos:

- *nome*: baseado na nomeação germânica (a, b, c, d, e, f, g);
- *oitava*: necessário para determinar e situar um nome específico e, conseqüentemente a sua freqüência;
- *duração*: o espaço de tempo em que a nota é executada;
- *alteração*: necessária para a representação do universo dos trinta e cinco possíveis nomes de notas.

Dessa forma, representa-se uma nota como um lista composta por quatro elementos, onde cada um é referência à sua classe de elementos. Dessa forma, cada um deles é definido como classes distintas, com os atributos descritos abaixo.

Classe Nomes

A *classe nomes* possui como declarações a designação dos nomes das notas juntamente com a definição de um valor numérico. Em código, tem-se o seguinte resultado:

```
class nomes.  
    nome (p) .  
    nome (a) .  
    nome (b) .  
    nome (c) .  
    nome (d) .  
    nome (e) .  
    nome (f) .  
    nome (g) .  
  
    valor_posicao(c, 1) .  
    valor_posicao(d, 3) .
```

```

valor_posicao(e, 5).
valor_posicao(f, 6).
valor_posicao(g, 8).
valor_posicao(a,10).
valor_posicao(b,12).

```

end nomes.

Onde o nome (*p*) corresponde às pausas, e os nomes restantes às notas conhecidas, conforme visto no capítulo 2:

dó (c), ré (d), mi (e), fá (f), sol (g), lá (a), si (b)

A cada um desses nome é associado um número que identifica o nome correspondente, tendo como ponto de partida a nota dó (c). Em termos ilustrativos, isto equivale a numerar as teclas brancas do piano, conforme a Figura 5.6.

Estes valores (1, 3, 5, 6, 8, 10 e 12) serão utilizados posteriormente no método que define o valor final de uma nota. Esta quantificação é feita baseada na disposição espacial das teclas brancas como pode ser denotado na Figura 5.6.

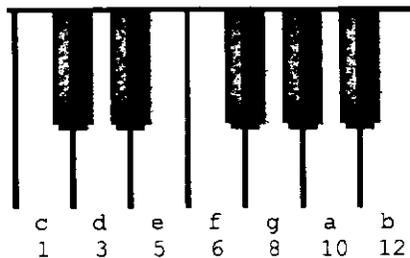


Figura 5.6: valor numérico para nome de notas

Classe Oitavas

Conforme o capítulo 2, as oitavas são em número de sete, no entanto, no universo voltado à música coral, apenas quatro oitavas são utilizadas, correspondendo à extensão da voz humana. Dessa forma, na classe oitavas encontra-se, similarmente à classe anterior, a definição de seus nomes juntamente com a definição de um valor numérico (este valor também é utilizado no método de definição do valor final de uma nota):

```

class oitavas.
    oitava(o1).
    oitava(o2).
    oitava(o3).
    oitava(o4).

    valor_oitava(o1, 1).
    valor_oitava(o2, 2).
    valor_oitava(o3, 3).

```

```

    valor_oitava(o4, 4).
end oitavas.

```

Classe Durações

A duração de uma nota é definida por um número inteiro qualquer, no entanto, como visto na Figura 2.4, os símbolos musicais para duração já possuem valores numéricos historicamente definidos.

Inicialmente, será utilizado um valor *default* (1) para as durações⁴⁵, portanto a *classe durações* fica definida da seguinte maneira:

```

class duracoes.
    duracao(1).
end duracoes.

```

Classe Alterações

A *classe alterações* é composta pelo símbolo atribuído a cada uma das alterações e pela atribuição de um valor numérico a cada destes símbolo:

```

class alteracoes.
    alteracao(n).
    alteracao(-).
    alteracao(=).
    alteracao(#).
    alteracao(+).

    valor_alteracao(n, 0).
    valor_alteracao(#, 1).
    valor_alteracao(-, -1).
    valor_alteracao(+, 2).
    valor_alteracao(=, -2).

end alteracoes.

```

Para entender as alterações musicais (sustenido, dobrado-sustenido, bemol e dobrado-bemol) basta associá-las a simples operações matemáticas. As alterações servem para baixar ou elevar semitons e isto é relativo a subtrações e adições feitas nos valores dos nomes das notas, ficando da seguinte maneira:

uma nota "xn" equivale a $x + 0$

uma nota "x-" equivale a $x - 1$

⁴⁵ Os exercícios iniciais de Harmonia em geral utilizam uma figura musical cujo valor é utilizado em todas as vozes. Deve-se lembrar que a atenção deve estar voltada às notas, suas alturas e nomes, ficando a duração em segundo plano neste ponto.

uma nota "x#" equivale a $x + 1$

uma nota "x=" equivale a $x - 2$

uma nota "x+" equivale a $x + 2$

Em relação ao teclado de piano significa teclas acima ou abaixo de "x".

Classe Notas

De acordo com a Figura 5.1, a *classe notas* define uma nota como sendo uma lista quádrupla, composta pelas declarações encontrados nas classes vistas anteriormente. Suas relações de herança podem ser visualizadas melhor no código abaixo⁴⁶:

```
class notas.
  inherits oitavas.
  inherits nomes.
  inherits alteracoes.
  inherits duracoes.

  private methods
    nota /1.

  public methods
    valor_da_notas /2.

. . .
```

As quatro primeiras linhas internas são responsáveis pela ligação entre a classe *notas* e suas classes formadoras. Dessa maneira, todas as declarações contidas nas classes *oitavas*, *notas*, *alteracoes* e *duracoes* estão agora fazendo parte da classe *notas*. As linhas seguintes informam que nesta classe existem métodos privados (*nota*), sendo que este possui como atributo apenas um elemento, e um método privado (*valor_da_notas*) com dois elementos em seu escopo.

```
. . .

/* Metodo nota */
nota ([Oitava, Nome, Alteracao, Duracao]) :-
  oitavas <- oitava(Oitava),
  nomes <- nome(Nome),
  alteracoes <- alteracao(Alteracao),
  duracoes <- duracao(Duracao).

. . .
```

O método acima é responsável pela verificação de todos os elementos que compõem uma nota, quando da entrada dos dados de oitava, nome, alteração e duração. Nas linhas se-

⁴⁶ No Anexo 2 o código pode ser visualizado na íntegra.

guintes são feitas chamadas às respectivas declarações de cada uma das classes. A veracidade destas declarações é a garantia de que os dados entrados constituem uma nota.

Observação: Um detalhe a ser observado é que, a partir do método acima, já é tratado pela representação a questão da busca de respostas tendo como ponto de partida um conhecimento incompleto. Por exemplo, a ausência de um dos dados de nota é prontamente preenchida pelo método acima, ocorrendo isto com todos os métodos aqui expostos.

Para a definição do valor de uma nota é necessário um método (`processo`) que produza este valor. Este método é responsável por calcular um valor único a cada nota, de forma que este valor possa representá-la posteriormente:

```
. . .

/* Metodo valor da nota */
valor_da_nota(nota([Oitava, Nome, Alteracao, Duracao]),
              Valor_da_Nota) :-
    oitavas <- valor_oitava(Oitava, Numero_Oitava),
    Valor_Oitava is ( Numero_Oitava - 1 ) * 12,
    nomes <- valor_posicao(Nome, Valor_Posicao),
    alteracoes <- valor_alteracao(Alteracao,
                                  Valor_Alteracao),
    Valor_da_Nota is Valor_Posicao + Valor_Alteracao
                   + Valor_Oitava + 35.

end notas.
```

Este método tem como dados de entrada os atributos de uma nota e retorna um valor numérico para cada uma delas. Este valor é uma representação interna utilizada para a manipulação das estruturas decorrentes do objeto nota. Para cada um dos atributos é feita uma chamada à classe equivalente e é retornado um número. Este número é utilizado para a definição de valores que são computados para a definição exata de valores de uma nota.

Aqui será feito um pequeno parêntese para um comentário sobre enarmonia.

Observação: Ao representar uma nota somente através de um valor numérico pode-se cair em uma armadilha, conforme visto no item 4.3. No entanto, a utilização separada de dados de nome, alteração e oitava, garante o tratamento de enarmonia, pois a identidade da nota só será possível mediante a presença de todos estes dados. Assim, a geração de um número segundo este método, garante um mesmo número para as notas enarmônicas sem, no entanto, privá-las de suas identidades e de seus dados diferenciais (nome e alteração). Dessa maneira, a questão das notas enarmonia fica totalmente resol-

vida, o mesmo ocorrendo com o conceito de intervalos enarmônicos que serão vistos no item 5.4.3.

Os resultados numéricos de *Valor_da_Nota*, *Valor_Posicao*, *Valor_Alteracao* e *Valor_Oitava* são somados, juntamente com o número 35 colocado na última linha do método acima para que o retorno do valor numérico seja equivalente ao número MIDI⁴⁷ da nota.

Na Figura 5.7 está representado um trecho musical, onde para cada nota é mostrada sua composição e logo depois o seu valor numérico⁴⁸:



```

nota01(o3, g, n, 1),
nota02(o3, b, -, 2),
nota03(o4, d, n, 4),
nota04(o3, f, #, 4),
nota05(o3, g, n, 8),
nota06(o3, a, n, 16),
nota07(o3, b, -, 16),
nota08(o4, d, n, 16),
nota09(o4, c, n, 16),
nota10(o3, b, n, 16),
nota11(o3, a, n, 16),
nota12(o3, c, +, 6),
nota13(o3, e, #, 6),
nota14(o3, f, #, 6),
nota15(o3, g, n, 4).

```

67, 70, 74, 66, 67, 69, 70, 74, 72, 71, 69, 62, 65, 66, 67

Figura 5.7: trecho musical, sua representação e valores numéricos

Estes números são utilizados nos métodos subseqüentes para a definição de outros objetos.

5.4.2 Tom e Semitom

Embora estes dois objetos não apareçam na estrutura apresentada na Figura 5.2, a sua definição em código é relevante pelo fato de seus conceitos estarem intimamente ligados às definições didático-musicais para acordes e escalas. Por esta razão a *classe tom* e *semitom* é aqui apresentada em forma de código.

⁴⁷ Isto facilitará o trabalho posterior quando do mapeamento das notas e seu valor MIDI para reprodução sonora. Dessa forma a primeira nota dó da primeira escala terá o valor numérico 36, enquanto a nota si da quarta oitava terá o valor 83.

Conforme visto no capítulo 2, pode-se representar o objeto musical *tom* e *semitom* através do cálculo da distância entre duas notas.

Classe Tom/Semitom

```
class semitons.
  inherits notas.
  inherits intervalos.

  private methods
    classifica_semitom_tom /4.
    semitom                /3.
  . . .
```

A classe *semitons* herda horizontalmente as declarações da classe *notas* e da classe *intervalos*, sendo que desta última a ligação acontece através do método *modulo*, visto mais adiante. Dois métodos privados são responsáveis pela definição da distância entre duas notas em tom ou semitom; o que for diferente disso é considerado erro.

Para a classificação de um intervalo entre notas, na qualidade de ser um intervalo de tom ou semitom, são necessárias:

- duas notas como entrada (com seus atributos de frequência) e
- um método para a definição desses conceitos. Isto é feito através do cálculo do valor da nota e de sua comparação dentro do método *classifica_tom/semitom*.

```
. . .
/* Metodo semitom */
  semitom(nota([O1,N1,A1,D1]),nota([O2,N2,A2,D2]),Tipo) :-
    notas <- (valor_da_nota(nota([O1,N1,A1,D1]),
                          Valor_da_Nota1),
             valor_da_nota(nota([O2,N2,A2,D2]),
                          Valor_da_Nota2)),
    Diferenca is Valor_da_Nota1 - Valor_da_Nota2,
    intervalos <- modulo(Diferenca,ValorModulo),
    classifica_semitom_tom(ValorModulo,Nome1,Nome2,Tipo).
. . .
```

A primeira chamada à classe *notas* faz a conversão das duas notas em valores numéricos, em seguida é feita a diferença entre estes dois valores. A existência de valores negativos é descartada com a chamada ao método *modulo* na classe *intervalos* (item 5.4.3). Na última linha o método *classifica_semitom_tom* faz a verificação.

⁴⁸ Equivalente ao valor MIDI.

```

...

/* Metodo classifica_semitom_tom */
classifica_semitom_tom(1, Nome1, Nome2, Tipo) :-
    (Nome1 = Nome2, copy_term(semitom_cromatico, Tipo), !);
    (copy_term(semitom_diatonico, Tipo)).
classifica_semitom_tom(2, _, _, tom).

end semitons.

```

De posse dos dados *ValorModulo*, *Nome1* e *Nome2*, basta agora encontrar o *Tipo*. Na primeira linha, se o *ValorModulo* for igual a 1, é feita a comparação entre nomes: se forem iguais, estará caracterizado o *semitom cromático*, caso contrário será *diatônico*. No caso de *ValorModulo* ser 2 a resposta será *tom*.

Um exemplo de associação entre teclas e números pode ser visto na Figura 5.8. De acordo com o método, se a diferença entre dois valores de nota for 1, então a distância é de um semitom; se for 2, então a distância é de um tom. Em resultados diferentes a resposta será nula (\emptyset).

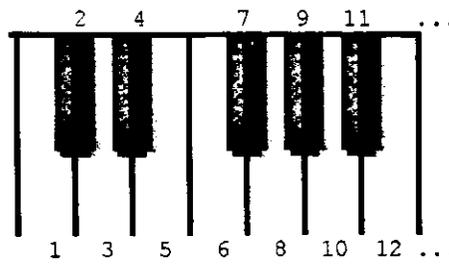


Figura 5.8: valor numérico para notas e suas alterações

5.4.3 Intervalo

Apesar da classe *tom/semitom* estar sendo apresentada em relevo, na realidade ela mesma encontra-se embutida dentro da *classe intervalos* que é mais abrangente. Só que neste caso, os conceitos de *tom* e *semitom* assumem denominações diferentes. Já a terminologia utilizada para intervalos em um exercício harmônico é a que aparece abaixo.

Classe Intervalos

De acordo com o que foi considerado no item 2.2.3, os intervalos são compostos por duas notas e pode-se representá-lo, em código, da seguinte maneira:

```

class intervalos.
    inherits notas.
    public methods
        intervalo /3,
        modulo /2.
    private methods
        grau_do_intervalo /4,
        direcao /9.
    classifica_intervalo_nota(1, unissono, 0).
    classifica_intervalo_nota(2, diminuta, 0).
    classifica_intervalo_nota(2, menor, 1).
    classifica_intervalo_nota(2, maior, 2).
    classifica_intervalo_nota(2, aumentada, 3).
    classifica_intervalo_nota(3, diminuta, 2).
    classifica_intervalo_nota(3, menor, 3).
    classifica_intervalo_nota(3, maior, 4).
    classifica_intervalo_nota(3, aumentada, 5).
    classifica_intervalo_nota(4, diminuta, 4).
    classifica_intervalo_nota(4, justa, 5).
    classifica_intervalo_nota(4, aumentada, 6).
    classifica_intervalo_nota(5, diminuta, 6).
    classifica_intervalo_nota(5, justa, 7).
    classifica_intervalo_nota(5, aumentada, 8).
    classifica_intervalo_nota(6, diminuta, 7).
    classifica_intervalo_nota(6, menor, 8).
    classifica_intervalo_nota(6, maior, 9).
    classifica_intervalo_nota(6, aumentada, 10).
    classifica_intervalo_nota(7, diminuta, 9).
    classifica_intervalo_nota(7, menor, 10).
    classifica_intervalo_nota(7, maior, 11).
    classifica_intervalo_nota(7, aumentada, 12).
    classifica_intervalo_nota(8, diminuta, 11).
    classifica_intervalo_nota(8, justa, 12).
    classifica_intervalo_nota(8, aumentada, 13).
    classifica_intervalo_nota(9, diminuta, 12).
    classifica_intervalo_nota(9, menor, 13).
    classifica_intervalo_nota(9, maior, 14).
    classifica_intervalo_nota(9, aumentada, 15).
    grau(c, 1).
    grau(d, 2).
    grau(e, 3).
    grau(f, 4).
    grau(g, 5).
    grau(a, 6).
    grau(b, 7).
    . . .

```

Como herança, a classe *intervalo* herda as declarações da classe *notas*. Possui um total de quatro métodos, sendo dois privados e dois públicos. Os métodos públicos são responsáveis pela formação do intervalo, como o próprio nome do método sugere, e pela transformação de números negativos em positivos (módulo). Dos métodos privados, um é dedicado ao cálculo do grau do intervalo e o outro a calcular a direção do mesmo.

Na continuação do código acima aparece a descrição dos intervalos compreendidos entre unísono e nona⁴⁹, representado pelo primeiro número na lista de `classifica_intervalo_nota`, seguido da definição nominal, baseado na quantidade de semitons entre as duas notas (último elemento da lista). Por último, vem a lista de grau que é responsável pela atribuição de um valor numérico (note que diferente do que foi atribuído na classe nota) que será utilizado posteriormente nos métodos atrelados à classe intervalos.

. . .

/* Metodo intervalo */

```
intervalo(nota([O1,N1,A1,D]), nota([O2,N2,A2,D]),
          intervalo_nota(Grau, Nome, Direcao)) :-
  grau_do_intervalo(nota([O1,N1,_,_]),
                  nota([O2,N2,_,_]), Grau, Direcao),
  notas <- (valor_da_nota(nota([O1,N1,A1,D]),
                        Valor_da_Nota1),
           valor_da_nota(nota([O2,N2,A2,D]),
                        Valor_da_Nota2)),
  Diferenca is Valor_da_Nota1 - Valor_da_Nota2,
  modulo(Diferenca, Modulo_Diferenca),
  classifica_intervalo_nota(Grau, Nome,
                           Modulo_Diferenca).
```

. . .

Como dados de entrada tem-se duas notas, obtendo-se a descrição completa do intervalo (o grau, o nome e a direção). O primeiro passo para se conseguir isto é calcular inicialmente o grau do intervalo (na chamada ao método `grau_do_intervalo`). Em seguida, calcula-se o valor das notas envolvidas e, após o cálculo da diferença entre as notas e a transformação deste valor em seu valor absoluto, é feita a comparação do resultado em `classifica_intervalo_nota` e os dados armazenados anteriormente.

. . .

/* Metodo grau_do_intervalo */

```
grau_do_intervalo(nota([O1,N1,_,_]), nota([O2,N2,_,_]),
                 Grau, Direcao) :-
  direcao(O1,N1,Numero_Oitava1,Valor_do_Grau1,
          O2,N2,Numero_Oitava2,Valor_do_Grau2,
          Diferenca_Grau),
  ((Numero_Oitava1 < Numero_Oitava2,
   copy_term(ascendente, Direcao),
   Grau is 8 - Diferenca_Grau);
   (Numero_Oitava1 > Numero_Oitava2,
   modulo(Diferenca_Grau, Modulo_Diferenca_Grau),
   copy_term(descendente, Direcao),
   Grau is 8 - Modulo_Diferenca_Grau);
   (Numero_Oitava1 = Numero_Oitava2,
   Valor_do_Grau1 < Valor_do_Grau2,
```

⁴⁹ Caso haja necessidade de incluir os intervalos subsequentes, basta aumentar esta relação.

```

copy_term(ascendente, Direcao),
modulo(Diferenca_Grau, Modulo_Diferenca_Grau),
Grau is Modulo_Diferenca_Grau + 1);
(Numero_Oitava1 = Numero_Oitava2,
Valor_do_Grau1 > Valor_do_Grau2,
copy_term(descendente, Direcao),
Grau is Diferenca_Grau + 1);
(Numero_Oitava1 = Numero_Oitava2,
Valor_do_Grau1 = Valor_do_Grau2,
copy_term(repetido, Direcao),
Grau is Diferenca_Grau + 1)).

```

. . .

O método acima utiliza como dados de entrada os valores de oitava e nome das duas notas respectivamente, já a sua saída é composta pela definição do grau e da direção do movimento (se descendente, ascendente ou uma repetição da voz anterior). Antes do cálculo do grau do intervalo, é feito inicialmente a chamada ao método `direcao`. De posse do resultado encontrado em método `direcao`, faz-se necessário comparar os valores de oitava e de grau, segundo a relação abaixo:

1. se $oitava1 < oitava2 \Rightarrow$ intervalo ascendente: esta diferença de oitavas deve ser computada, no final, para se chegar ao nome e grau do intervalo.
2. se $oitava1 > oitava2 \Rightarrow$ intervalo descendente: com a mesma implicação anterior.
3. se $(oitava1 = oitava2) \wedge (nota1 < nota2) \Rightarrow$ intervalo ascendente: onde as notas encontram-se em uma mesma oitava.
4. se $(oitava1 = oitava2) \wedge (nota1 > nota2) \Rightarrow$ intervalo descendente: similar à implicação anterior.

. . .

/* Metodo direcao */

```

direcao(O1,N1,Numero_Oitava1,Valor_do_Grau1,
        O2,N2,Numero_Oitava2,Valor_do_Grau2,
        Diferenca_Grau ) :-
oitavas <- (valor_oitava(O1,Numero_Oitava1),
            valor_oitava(O2,Numero_Oitava2)),
grau(N1,Valor_do_Grau1),
grau(N2,Valor_do_Grau2),
Diferenca_Grau is Valor_do_Grau1 - Valor_do_Grau2.

```

. . .

Apesar de conter nove parâmetros, o método `direcao` tem como tarefa fazer o cálculo da diferença entre os valores atribuídos a grau. É este valor que irá determinar no método `grau_do_intervalo`, juntamente com `Numero_Oitava` de cada nota, a definição

completa do intervalo em questão: seu grau (do 1º grau até o 9º grau), seu nome (se uníssono, diminuto, menor, maior ou aumentado) e sua direção (se ascendente ou descendente)⁵⁰.

Finalizando a classe `intervalos` existe o método `modulo`, que é responsável por achar o valor absoluto de um número. Entra-se com um valor qualquer e este valor sendo menor que zero basta multiplicá-lo por -1.

```

. . .

/* Metodo modulo */
    modulo(Valor,Modulo_do_Valor) :-
        (Valor < 0, Modulo_do_Valor is Valor * -1, !);
        Modulo_do_Valor is Valor.

end intervalos.

```

5.4.4 Escala

A classe abaixo é responsável pela relação de escalas possíveis. A necessidade de uma classe que defina as possibilidades de escalas, justifica-se pela não existência de 35 escalas diferentes (uma relativa a cada nota com e sem alterações). Na prática, o seu número é reduzido a 30, levando-se em consideração tanto as escalas maiores, quanto as suas relativas menores.

Classe Nome_escalas

A classe `nome_escalas` é formada por uma lista tríplice, onde o primeiro item é o *nome da nota* que nomeia a *escala*, o segundo item é a *alteração* e, por fim, o *modo*. No caso do *modo* não estar diretamente mencionado, subentende-se a existência de ambos.

```

class nome_escalas.
    nome_escalas(c, n, _).
    nome_escalas(g, n, _).
    nome_escalas(d, n, _).
    nome_escalas(a, n, _).
    nome_escalas(e, n, _).
    nome_escalas(b, n, _).
    nome_escalas(f, #, _).
    nome_escalas(c, #, _).
    nome_escalas(f, n, _).

```

⁵⁰ Neste código estão representados apenas os intervalos melódicos. No caso de intervalos harmônicos é necessário que não se tenha apenas as notas como entrada, mas também as suas respectivas vozes, por exemplo:

```
intervalo_nota(Entrada:Soprano(Nota1), Contralto(Nota1); Saída:Intervalo)
```

O condicional aqui é a presença ou não de vozes, bastando apenas acrescentar poucas linhas de código.

```

nome_escala(b, -, _).
nome_escala(e, -, _).
nome_escala(a, -, _).

nome_escala(d, -, maior).
nome_escala(g, -, maior).
nome_escala(c, -, maior).
nome_escala(g, #, menor).
nome_escala(d, #, menor).
nome_escala(a, #, menor).

end nome_escalas.

```

Classe Escalas

Como visto no item 2.2.4, o objeto escala pode ser definido como uma sucessão de intervalos predeterminados, formando em si um estrutura coerente. Ainda com referência a este item, a escala utilizada nos exercícios harmônicos são as escalas diatônicas maiores ou menores (esta última apenas no seu formato harmônico). Por ser composta por oito notas musicais (incluindo a repetição do 1º grau), as escalas diatônicas possuem sete intervalos (vide tabelas 2.4 e 2.5).

```

class escalas.
    inherits nome_escalas.
    inherits intervalos.

    public methods
        escala /3.

    private methods
        forma_escala /5.

    classifica_escala(maior,2,
        [maior,maior,menor,maior,maior,maior,menor]).
    classifica_escala(menor,2,
        [maior,menor,maior,maior,menor,maior,maior]).
    classifica_escala(harmonica,2,
        [maior,menor,maior,maior,menor,aumentada,menor]).

    . . .

```

No código acima, além da definição de heranças e da apresentação dos métodos, existe a declaração de `classifica_escala`. A lista contida em `classifica_escala` é responsável pela construção propriamente dita do objeto escala. O primeiro item indica o modo (neste caso o modo maior e modo menor natural e harmônico); o segundo item é a definição de escala diatônica (onde cada nota está na distância de um intervalo de 2º grau); por fim tem-se a lista onde para cada par de notas é definido um intervalo sempre de 2ª, sendo maior, menor ou aumentado, segundo a sua posição na lista.

. . .

```

/* Metodo escala */
escala(nota([O,N,A,D]),Modo,[Escala]) :-
    nome_escalas <- nome_escala(N,A,Modo),
    escalas <- classifica_escala(Modo,Grau,Modo2),
    forma_escala(nota([O,N,A,D]),Grau,Modo2,Escala,7),
    !.

```

Este método é voltado à verificação da existência ou não da escala pretendida e da preparação de sua construção. É dado como entrada os valores de uma nota inicial e o modo da escala, sendo o retorno a escala propriamente dita. Inicialmente, é feita, utilizando-se os dados de entrada, a verificação da existência ou não da escala através da chamada à classe *nomes_escalas*. Em seguida esta é classificada e, por fim, é feita a chamada ao método *forma_escala*.

```

/* Metodo forma_escala */
forma_escala(Nota1,Grau,Modo,[Nota1],0) :-
    intervalos <- intervalo(Nota1,Nota2,
                           intervalo_nota(Grau,Mod,ascendente)),
    !.
forma_escala(Nota1,Grau,[Mod|Modo],[Nota1|Escala],N) :-
    intervalos <- intervalo(Nota1,Nota2,
                           intervalo_nota(Grau,Mod,ascendente)),
    M is N - 1,
    forma_escala(Nota2,Grau,Modo,Escala,M).

end escalas.

```

Aqui, pela primeira vez, surge um método que se utiliza de recursão em sua tarefa. O método *forma_escala* é composto dos seguintes dados: uma nota inicial, um grau (2º grau⁵¹), um modo, uma lista (onde será armazenado as notas da escala) e de um quantificador que servirá de guia para ao número de chamadas ao método (mecanismo de parada). Desta forma, o método será ativado a partir da segunda declaração do mesmo⁵².

A chamada à classe *intervalos* é a responsável pela definição da próxima nota, e ocorre da seguinte maneira:

- os parâmetros de entrada são compostos por uma nota inicial (Nota1) e pelo tipo de intervalo que se deseja,

⁵¹ Neste caso específico o grau é constante (sempre de 2º grau), se a necessidade for a de trabalhar com escalas diferentes, pode-se alterar este valor ou mesmo deixá-lo variável utilizando uma lista como parâmetros.

⁵² Devido ao valor do quantificador guia, que inicialmente é sete. Onde se lê *M is N - 1*, este quantificar é decrementado. Somente ao atingir o valor zero é que é feita a chamada à primeira declaração de *forma_escala*, finalizando a recursão.

- na primeira chamada do método `forma_escala` tem-se a chamada da classe `intervalos` onde os dados de entrada são computados (Nota1 e intervalo de 2ª ascendente) e é dada uma segunda nota como resposta,
- na segunda chamada, a lista de modos (definição das segundas menores, maiores ou aumentadas) é incrementada e a primeira nota é colocada na lista escala,
- é feito mais uma vez todo o processo decorrente da chamada à classe `intervalos`,
- na terceira chamada ao método `forma_escala` o parâmetro de entrada é alterado e Nota2 passa a ser a nota inicial reiniciando o processo.

5.4.5 Acorde

A construção do objeto `acordes` é bastante similar à construção de escalas. Neste caso, tem-se uma relação de dados voltados à criação de acordes com três e quatro notas (tríades e tétrades).

Classe Acordes (tríades e tétrades)

```
class acordes.
    inherits intervalos.
    public methods
        triade          /3.
    private methods
        forma_triade /5.
        classifica_triade(maior,3,[maior,menor],2).
        classifica_triade(menor,3,[menor,maior],2).
        classifica_triade(diminuto,3,[menor,menor],2).
        classifica_triade(aumentado,3,[maior,maior],2).
        classifica_triade(setima_menor,3,[maior,menor,menor],3).
        classifica_triade(setima_maior,3,[maior,menor,maior],3).
    . . .
```

Após as declarações de herança e relação de métodos, aparece uma lista em `classifica_triade` nos mesmos moldes da encontrada em `classifica_escala`. O primeiro item diz respeito ao tipo da tríade/tétrade, o segundo ao grau dos intervalos (3º grau, ou seja, terças), a relação de intervalos entre notas e o quantificador (2 no caso das tríades e 3 no caso das tétrades).

```
. . .
/* Metodo triade */
```

```

triade(nota([O,N,A,D]),Tipo,[Triade]) :-
    acordes <- classifica_triade(Tipo,Grau,Tipo2,Numero),
    forma_triade(nota([O,N,A,D]),Grau,Tipo2,
        Triade,Numero),
    !.
. . .

```

O método acima é responsável pela classificação do acorde. Este é um método típico que pode ser aplicado ao ensino, pois de acordo com o nível que se deseja trabalhar pode-se ampliar a base de dados contida em `classifica_triade`, incluindo outros tipos de tríades/tétrades. A chamada ao método `forma_triade` é responsável pela construção do acorde.

```

. . .

/* Metodo forma_triade */

forma_triade(Nota1,Grau,Tipo,[Nota1],0) :-
    intervalos <- intervalo(Nota1,Nota2,
        intervalo_nota(Grau,Tipo,ascendente)),
    !.
forma_triade(Nota1,Grau,[Tip|Tipo],[Nota1|Triade],N) :-
    intervalos <- intervalo(Nota1,Nota2,
        intervalo_nota(Grau,Tipo,ascendente)),
    M is N - 1,
    forma_triade(Nota2,Grau,Tipo,Triade,M).

end acordes.

```

O processo deste método é similar ao processo de `forma_escala`. Após a classificação do acorde, passa-se à fase de construção do mesmo. A base é uma nota inicial e a descrição do mesmo (se diminuto, menor, maior, aumentado, sétima menor ou sétima maior), com estes elementos torna-se fácil a construção do restante das notas.

Em ambos os métodos para a formação de escalas e acordes, a inferência pode ser feita não apenas partindo de uma nota e uma descrição; mas a partir de um acorde, completo ou incompleto, tem-se condições de definir o acorde e de encontrar as notas faltantes.

5.4.6 Tonalidade

Segundo o que foi dito no item 2.2.5, a Tonalidade é responsável pela definição de determinadas relações entre as notas. Estas relações são de grande importância no ensino de Harmonia. Na prática, conforme a Figura 5.3, pode-se ver a tonalidade como a definidora de quais acordes e escala serão utilizados em um determinado instante. Portanto, ao objeto musical Tonalidade, têm-se duas classes distintas: a que define os tipos de acordes possíveis em um determinado nível; e a que é responsável pela relação *tonalidade-escala-*

acorde.

Além da capacidade de abstrair certos acordes de seu conjunto total, a *classe tonalidades* é que define a função (vide item 4.6) destes acordes. Como exemplo têm-se os principais acordes que são montados sobre o I, IV e V graus da escala e esta, com dito acima, é definida pela Tonalidade.

Por este motivo, considera-se a Tonalidade como um objeto distinto, pois a sua presença em uma sentença lógica será responsável pela limitação de acordes e escala. Portanto, o código de Tonalidade pode ser visto de duas maneiras:

- *definição de escala e acordes*: entra-se com os dados sobre uma determinada tonalidade (nome, alteração e modo) e a saída complementa os dados definindo a escala e os acordes⁵³.
- *definição de relações entre acordes*: onde é feita a distribuição em graus (associação de cada acorde gerado com os graus da escala). A importância de um método responsável por esta ação é bem visível quando da análise e correção de exercícios, onde uma visão geral pode ser feita antes das considerações sobre o encadeamento

Classe Tipos_Tonalidades

A *classe tipos_tonalidades* é responsável pela nomeação dos tipos de tonalidades necessárias. Caso se deseje um universo menor de tonalidades basta restringir o número de declarações desta classe. No caso abaixo, é descrito em *tipo_tonalidade*, todas as tonalidades possíveis (15 tonalidades maiores e 15 tonalidade menores).

```
class tipos_tonalidades.
    tipo_tonalidade(c, n, _).
    tipo_tonalidade(g, n, _).
    tipo_tonalidade(d, n, _).
    tipo_tonalidade(a, n, _).
    tipo_tonalidade(e, n, _).
    tipo_tonalidade(b, n, _).
    tipo_tonalidade(f, #, _).
    tipo_tonalidade(c, #, _).
    tipo_tonalidade(f, n, _).
    tipo_tonalidade(b, -, _).
    tipo_tonalidade(e, -, _).
    tipo_tonalidade(a, -, _).
    tipo_tonalidade(d, -, maior).
```

⁵³ O método para a definição de acordes pode estar totalmente vinculado ao nível de aprendizagem do aprendiz. Se o mesmo se encontrar no nível um, por exemplo, pode-se definir que o conjunto de acordes estará restrito aos de tônica, subdominante e dominante (respectivamente I, IV e V graus). Em níveis posteriores pode-se acrescentar as inversões e os acordes relativos, se for o caso.

```

tipo_tonalidade(g, -, maior).
tipo_tonalidade(c, -, maior).
tipo_tonalidade(g, #, menor).
tipo_tonalidade(d, #, menor).
tipo_tonalidade(a, #, menor).

relacao_acordes(maior, [maior, menor, menor, maior, maior,
                        menor, diminuto]).
relacao_acordes(menor, [menor, diminuto, aumentado, menor,
                        maior, menor, diminuto]).

end tipos_tonalidades.

```

Na declaração *relacao_acordes* é descrita a construção dos acordes, tendo como base um escala predefinida (maior ou menor), ou seja: em uma tonalidade maior a seqüência de acordes sobre a escala é composta pelo primeiro acorde maior, o segundo menor e assim sucessivamente (lista). O mesmo acontecendo com as escalas menores.

Classe Tonalidades

Esta classe é responsável pela construção das tonalidades, seja através da entrada de todos os seus dados para validação (incluindo os acordes e escala), seja através da saída de alguns de seus objetos componentes.

```

class tonalidades.
    inherits escalas.
    inherits acordes.

    public methods
        tonalidade           /3.
        define_escala        /4.
        define1_acordes     /7.

/* Metodo tonalidade */
    tonalidade(Nome,Alteracao,Modo) :-
        tipos_tonalidades <- tipo_tonalidade(Nome,
                                                Alteracao,Modo),
        tipos_tonalidades <- relacao_acordes(Modo,Lista).

    . . .

```

O método *tonalidade* é responsável pela verificação dos componentes da classe *tonalidades*. Os dois métodos seguintes (*define_escala* e *define1_Acordes*) respondem pela formação de escalas e acordes tendo como referência a tonalidade específica.

```

    . . .

/* Metodo define_escala */
    define_escala(Nome,Alteracao,maior,[Escala]) :-
        escalas <- escala(nota([_,Nome,Alteracao,1]),
                          Modo,Escala),

```

```

!.
define_escala(Nome,Alteracao,menor,[Escala]) :-
    escalas <- escala(nota([_,Nome,Alteracao,1]),
                    harmonica,Escala),
!.

```

. . .

Para a chamada ao método `escala_nota` é enviado o nome e alteração que irá compor os atributos de nota encontrados no método. Deve-se observar que em `define_escala` é feita uma divisão utilizando o parâmetro `modo` (maior no primeiro caso e menor no segundo). Esta estratégia melhora o código, já que não é necessário um método que faça esta verificação.

Com a definição da escala, o passo seguinte é o da definição dos acordes. Neste caso o procedimento é similar ao método `define_escala` onde são utilizados os atributos de nome e alteração do objeto notas para a construção da escala.

. . .

/* Metodo define1_acordes */

```

define1_acordes(n1,Nome,Alteracao,maior,I,IV,V) :-
    acordes <- triade(nota([o2,Nome,Alteracao,1]),
                    maior,I),
    intervalos <- intervalo(nota([o2,Nome,Alteracao,1]),
                          Nota2,intervalo_nota(4,justa,ascendente)),
    acordes <- triade(Nota2,maior,IV),
    intervalos <- intervalo(nota([o2,Nome,Alteracao,1]),
                          Nota3,intervalo_nota(5,justa,ascendente)),
    acordes <- triade(Nota3,maior,V),
!.
define1_acordes(n1,Nome,Alteracao,menor,I,IV,V) :-
    acordes <- triade(nota([o2,Nome,Alteracao,1]),
                    menor,I),
    intervalos <- intervalo(nota([o2,Nome,Alteracao,1]),
                          Nota2,intervalo_nota(4,justa,ascendente)),
    acordes <- triade(Nota2,menor,IV),
    intervalos <- intervalo(nota([o2,Nome,Alteracao,1]),
                          Nota3,intervalo_nota(5,justa,ascendente)),
    acordes <- triade(Nota3,maior,V),
!.

```

end tonalidades.

Aqui mais uma vez foi utilizada a divisão por modos (modo maior separado do modo menor) já feita no método anterior. O método `define_acordes` possui a mesma estrutura independentemente do tipo de acordes que se deseja. O que irá definir a quantidade e os tipos de acordes é o primeiro item de sua lista (representando o nível). No caso acima, é feita uma representação baseada em acordes de I, IV e V graus, sem alteração nenhuma

o que corresponde a um possível nível inicial⁵⁴.

Os passos para se encontrar os acordes são os seguintes:

1. a partir dos dados iniciais (nome, alteração e modo), forma-se a tríade que corresponde ao primeiro grau da escala (chamada à *classe acordes*);
2. utiliza-se a *classe intervalos* para que, partindo da nota inicial, se encontre a nota do grau desejado (no caso o IV, conforme a declaração `intervalo_nota(4, justa, ascendente)`);
3. com o dado anterior, repete-se a ação de formação de tríade;
4. este mesmo processo (itens 2 e 3) é repetido para se encontrar o graus seguintes.

O mesmo processo é feito nas linhas seguintes, tendo como base o modo menor.

Como coadjuvante da *classe tonalidades* temos a *classe níveis* e a *classe cifras* que serão vistas a seguir.

Classe Cifras

A classe *cifras* é composta apenas pelo método que faz a verificação se as cifras dadas como resposta pelo aluno estão contidas dentro da tonalidade e do nível pretendidos.

```
class cifras.
  public methods
    verifica_cifras /2.
  /* Metodo verifica_cifras */
    verifica_cifras(Nivel, []).
    verifica_cifras(Nivel, [Cifra1|Resto]) :-
      niveis <- Nivel(Cifra1),
      verifica_cifras(Nivel, Resto).
end cifras.
```

O método para verificação de cifras faz a chamada à classe *niveis*, verificando a existência do acorde em um nível específico.

Classe Níveis

Aqui, de acordo com os objetivos do professor-especialista, pode-se acrescentar quantos níveis forem desejados. A representação abaixo retrata apenas os três níveis iniciais.

⁵⁴ Cabe aqui ao professor-especialista, de acordo com o propósito, criar um nível particular a cada objetivo preten-

```

class niveis.
    n1(i).
    n1(iv).
    n1(v).
    n2(i).
    n2(iv).
    n2(v).
    n2(v7).
end niveis.

```

5.5 Representação de Vozes

Conforme foi dito no capítulo 2, a disposição das quatro vozes coral obedecem a certas regras. Em geral, quando da correção de um exercício, os professores costumam ter uma visão preliminar da resposta do aluno através da verificação das regras de distribuição. Sendo assim, é necessário que para cada voz de um encadeamento seja verificado a quebra ou não destas regras antes da correção propriamente dita (regras de encadeamento).

```

class vozes.
    inherits notas.
    public methods
        verifica_voz    /1.
    private methods
        nota_correta    /2.
        extensao        /3.

    tessitura(soprano, nota([o3,c,n,_]), nota([o4,g,n,_])).
    tessitura(alto,    nota([o2,g,n,_]), nota([o3,d,n,_])).
    tessitura(tenor,  nota([o2,c,n,_]), nota([o3,g,n,_])).
    tessitura(baixo, nota([o1,e,n,_]), nota([o3,c,n,_])).
    . . .

```

As declarações contidas em *tessitura*⁵⁵ define os limites de cada voz, de sua nota mais grave até a mais aguda.

```

. . .

/* Metodo verifica_voz */
verifica_voz([]).
verifica_voz([V|R]) :-
    V =.. [Voz|[Notas]],
    nota_correta(Voz,Notas),

```

dido.

⁵⁵ Tessitura é o âmbito de notas útil à voz e/ou instrumento musical. Em geral, a tessitura é colocada como a região sonora na qual a voz/instrumento tem total domínio.

```
verifica_voz(R).
```

. . .

Por se estar trabalhando com uma lista de listas, onde as notas de cada uma das vozes aparecem, o método `verifica_voz` é necessário por fazer separação de cada uma das vozes e entre estas e suas listas de notas. Esta decomposição é feita na linha `v = . . [Voz | [Notas]]`. Em seguida é feita a chamada ao método `nota_correta`, responsável por obter os dados contidos na declaração `tessitura`. Mais uma vez é feita uma recursão, onde o mecanismo de parada é a total ausência de elementos na lista.

. . .

```
/* Metodo nota_correta */
```

```
nota_correta(Voz, []).
nota_correta(Voz, [Nota|R]) :-
    tessitura(Voz, Inicial, Final),
    extensao(Nota, Inicial, Final),
    nota_correta(Voz, R).
```

. . .

Com a voz definida e a inclusão dos dados das notas iniciais e finais de sua tessitura, o passo seguinte é a chamada ao método que verifica se as notas se encontram na extensão delimitada em `tessitura`.

. . .

```
/* Metodo extensao */
```

```
extensao(nota([_,p,_,_]),_,_).
extensao(nota([Oitava, Nome, Alteracao, _]),
    nota([Oitava_i, Nome_i, Alteracao_i, _]),
    nota([Oitava_f, Nome_f, Alteracao_f, _])) :-
    valor_da_nota(nota([Oitava, Nome, Alteracao, _]),
        Valor_nota),
    valor_da_nota(nota([Oitava_i, Nome_i, Alteracao_i, _]),
        Valor_nota_i),
    valor_da_nota(nota([Oitava_f, Nome_f, Alteracao_f, _]),
        Valor_nota_f),
    Valor_nota >= Valor_nota_i,
    Valor_nota =< Valor_nota_f.
```

```
end vozes.
```

No método `extensao`, é feita a comparação, utilizando para isto o valores das notas (`nota`, `nota_inicial` e `nota_final`). Se as duas últimas declarações do método `extensao` forem confirmadas, a nota estará em sua tessitura.

Ainda em relação às vozes de um encadeamento, regras adicionais devem existir para que se faça a verificação óbvia da existência das quatro vozes. No entanto, como dito anteri-

ormente, o trabalho com lista de listas exige que se tenha um mecanismo eficiente para a manipulação destas listas. Abaixo segue a descrição em código desta classe *distribuicao*.

```
class distribuicao.
    private methods
        verifica_vozes    /1.
        tipo_voz          /2.
        difere_voz        /1.
        difere_notas      /1.
    . . .
```

A classe *distribuicao* é composta por quatro métodos, responsáveis por verificar a existência das quatro vozes. Basicamente eles operam da seguinte maneira:

- verificam a existência das quatro vozes distintas,
- faz a separação em unidades e
- compara se nenhuma é repetição da outra.

```
. . .
/* Metodo verifica_vozes */
verifica_vozes([Soprano,Alto,Tenor,Baixo]) :-
    tipo_voz(Soprano,Tipo_Voz1),
    tipo_voz(Alto, Tipo_Voz2),
    tipo_voz(Tenor, Tipo_Voz3),
    tipo_voz(Baixo, Tipo_Voz4),
    difere_voz([Tipo_Voz1,Tipo_Voz2,Tipo_Voz3,
               Tipo_Voz4]).
. . .
```

Na chamada ao método *tipo_voz* é feita a separação entre nome da voz e suas notas. Em seguida, os nomes são comparados no método *difere_voz*.

```
. . .
/* Metodo tipo_voz */
tipo_voz(Voz,Tipo) :-
    Voz =.. [Tipo|_].
. . .
```

Em *difere_voz*, a linha `not member(Cabeca,Cauda)` é responsável por comparar cada uma das vozes com o restante da lista.

```
. . .
/* Metodo: difere_voz */
difere_voz([]).
```

```
difere_voz([Cabeca]).  
difere_voz([Cabeca|Cauda]) :-  
    not member(Cabeca, Cauda),  
    difere_voz(Cauda).
```

end distribuicao.

Como conclusão da representação do conhecimento musical acima, convêm ressaltar que muito embora não se tenha explorado a tarefa de representar as regras Harmônicas (objetivo previsto em trabalhos futuros), pode-se entrever as contribuições da representação. A capacidade de resolver algumas das questões teóricas de Música, tais como:

- a enarmonia em particular,
- encontrar respostas completas a partir de objetos musicais incompletos,
- capacidade de construir os objetos de escalas, acordes e tonalidades, relevantes ao domínio de Harmonia.

Estas questões podem ser consideradas como incentivo à ampliação deste modelo de representação, com o objetivo de abranger o universo de regras (sejam em contexto Normativo ou Funcional).

6 O Ambiente MATHEMA

6.1 Introdução

O encaminhamento dado à representação do conhecimento musical e sua adequação ao modelo de dimensões, visto no capítulo anterior, dispõe de uma proposta apropriada para ser trabalhada num ambiente tutorial, no qual a ênfase dada ao ensino trilha o mesmo caminho discutido no capítulo 2. A preocupação didática em definir limites ao aprendiz, em oposição à necessidade de não deixar que o mesmo seja escravo destes limites, irá exigir de um sistema computacional dessa natureza, a mesma maleabilidade de um especialista e sua capacidade de adaptar suas “lições” a cada momento do aprendizado.

Por motivos similares ao exposto acima, a área de pesquisa envolvendo Inteligência Artificial e Educação, através dos Sistemas Tutores Inteligentes (STI) [WENGER, 1987; PONTES, 1992], investe na possibilidade de disponibilização de sistemas com propósitos de promover um ensino adaptativo ao estudante, considerando suas características individuais.

Neste capítulo, será descrito sucintamente o ambiente interativo de apoio ao ensino e aprendizagem. Trata-se do ambiente denominado MATHEMA, inicialmente descrito em [COSTA, 1995] que, neste contexto, estará utilizando a Harmonia como domínio de aplicação. A idéia aqui, é partir para a construção de um esboço de STI no domínio da Harmonia, tomando o ambiente MATHEMA como modelo [COSTA, 1997a].

Em MATHEMA, adota-se uma arquitetura baseada em múltiplos agentes, onde cada um deles é especializado em partes do conhecimento associado ao domínio que está sendo considerado. O objetivo mais específico deste capítulo é o de apresentar a modelagem do domínio da Harmonia segundo o esquema de dimensões do conhecimento [TEIXEIRA, 1997; COSTA, 1997a]. Com base neste esquema, define-se uma sociedade de agentes tutores que irão interagir com um aprendiz humano. Em seguida será descrito um esboço de um modelo das interações didáticas entre o aprendiz e um agente tutor artificial.

A idéia básica a ser trabalhada num tal sistema tutor é colocar o aprendiz numa situação de resolução de problemas e, sempre que necessário, o sistema oferece-lhe um *feedback* apropriado. Em alguns casos mais extremados, o sistema tutor multi-agente recorrerá a

um suporte externo (Sociedade de Especialistas Humanos).

Nesta perspectiva, inicialmente será descrito sucintamente o ambiente propriamente dito, destacando sua arquitetura. Em seguida, serão apresentadas as diferentes dimensões de conhecimentos nas quais a Harmonia pode ser vista. Por último, serão apresentados alguns cenários ilustrando interações didáticas entre o aprendiz e o sistema.

6.2 O Ambiente de Trabalho

MATHEMA é um ambiente de aprendizagem interativo, sendo concebido para possibilitar um ensino adaptativo e seus desdobramentos no processo de aprendizagem. O ensino adaptativo é aqui visto como consequência do processo de interações cooperativas envolvendo os seus componentes (*Aprendiz, Tutores*) em atividades baseadas em resolução de problemas. A aprendizagem, por sua vez, é assumida neste trabalho como sendo favorecida e decorrente de atividades provenientes do ensino adaptativo, significando aquisição de conhecimento. Nesta perspectiva, o MATHEMA propõe-se a prover princípios e uma arquitetura alternativa, necessários para orientar o desenvolvimento de STI particulares [COSTA, 1996a].

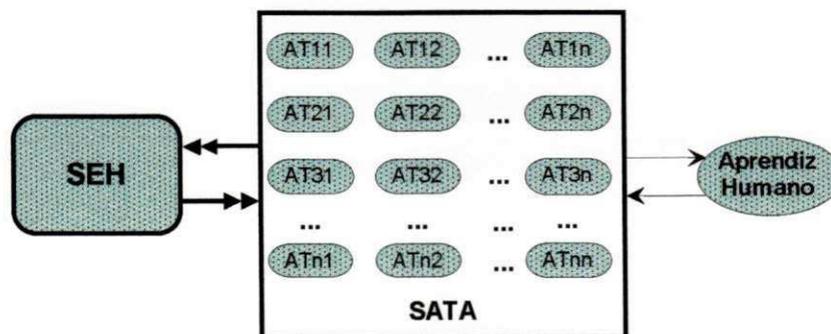


Figura 6.1: arquitetura simplificada do MATHEMA

A arquitetura simplificada do MATHEMA é mostrada na Figura 6.1. Ela consiste essencialmente em três componentes:

1. *Aprendiz Humano*: agente interessado em aprender algo sobre um dado domínio, trabalhado no ambiente MATHEMA.
2. *Sociedade de Agentes Tutores Artificiais (SATA)*: conjunto de agentes que algumas vezes cooperam entre si a fim de promover a aprendizagem de um dado aprendiz em atividade de ensino. SATA é definida, conforme mencionado anteriormente, em função de uma visão sobre o conhecimento do domínio, resultando em uma série de subdomínios, obedecendo a um certo critério. Essa idéia foi ini-

cialmente inspirada nas reflexões contidas em “Sociedade da Mente”, de Marvin Minsky [MINSKY, 1985]. Segundo Minsky, a inteligência emerge da combinação de agentes mentais, cada um responsável por um pequeno processo.

3. *Sociedade de Especialistas Humanos* (SEH): fonte de conhecimento externa ao sistema computacional (algo como um agente oracular) para a SATA. Dessa Sociedade é requerida a manutenção da SATA e eventual assistência aos aprendizes.

6.3 Dimensões do Conhecimento

Numa perspectiva de ensino-aprendizagem, o conhecimento do domínio está reunido nos objetos constantes na Figura 5.2, juntamente com o relacionamento entre as regras de distribuição de vozes e as regras de encadeamento mostradas na Figura 5.4. Por toda complexidade do conhecimento e de sua representação exposta no decorrer deste capítulo, será apresentada neste tópico uma visão de organização deste conhecimento, utilizando como perspectiva a sua disponibilização em um Sistema Tutorial Inteligente. Dessa forma, o modelo de ambiente MATHEMA foi utilizado como alternativa, tendo como metas os seguintes pontos:

- (i) a organização do conhecimento utilizando como paradigma a sua visão em dimensões,
- (ii) a utilização de agentes tutores, responsáveis por ensinar conhecimentos específicos.

A opção pelo MATHEMA implicou na adequação e na distribuição dos conceitos musicais vistos no item anterior através do mapeamento do conhecimento em diferentes visões (dimensões). Esta abordagem constitui uma das características do MATHEMA: a preocupação com qualidade da organização do conhecimento e com os mecanismos de interação entre as entidades no processo de ensino. Portanto, o objetivo principal é um compromisso entre riqueza e estruturação do conhecimento [COSTA, 1996a, 1997b].

A visão multidimensional mencionada considera as três dimensões seguintes:

- (i) o *contexto*, que define múltiplos pontos de vista do domínio,
- (ii) a *profundidade*, que diz respeito a diferentes níveis de abordagem do domínio e
- (iii) a *lateralidade*, que está vinculada aos conhecimentos fortemente relacionados ao domínio, tais como os conhecidos pré-requisitos.

Nesta perspectiva, um domínio D qualquer no MATHEMA pode ser definido como o conjunto de vários subdomínios, da seguinte maneira:

$D = \{d_{11}, d_{12}, \dots, d_{nm}\}$, onde d_{ij} é definido a partir de um par $\langle C_i, P_{ij} \rangle$, sendo C_i um contexto particular e P_{ij} , uma profundidade associada a este contexto.

Junto ao contexto e à profundidade atribui-se o conceito de lateralidade, onde para cada subdomínio d_{ij} associa-se uma lateralidade l_{ijk} .

A cada d_{ij} é associado um agente tutor [COSTA, 1996b], o mesmo acontecendo com cada l_{ijk} . Em resumo, a visão de um determinado conhecimento sobre um domínio é a união de domínios específicos, mais o conhecimento essencial adjacente:

$$d_{11} \cup d_{12} \cup d_{21} \cup d_{22} \cup \dots \cup \text{lateralidades correspondentes}$$

Numa visão interna de cada d_{ij} , identificam-se os recursos de ensino: conceitos, resultados, exemplos, contra-exemplos, exercícios resolvidos, problemas, dicas, etc envolvidos em cada unidade de conhecimento do *curriculum* em d_{ij} e também em dl_{ij} , onde o l_{ijk} é a lateralidade correlata. Além disso, cada d_{ij} pode ser visto através do desenvolver de diversas situações (*curriculum* ou módulos), juntamente com o referido conjunto de recursos de ensino (Figura 6.2).

No plano superior da Figura 6.2 encontram-se as situações e, logo abaixo, os problemas (com seus relacionamentos intrínsecos) e o material didático necessários às sessões de aprendizagem.

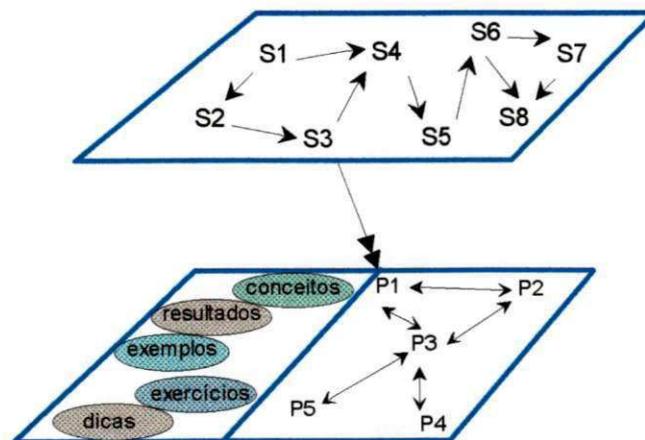


Figura 6.2: visão interna de um agente d_{ij} .

Esta visão do conhecimento guarda uma certa semelhança com a estrutura da representação aqui apresentada. Esta semelhança se dá na medida em que, dependendo da profundidade adotada, os conceitos musicais podem “escolher” determinados conceitos e ou lateralidades específicas. A Figura 6.3 mostra estas relações, onde um determinado conhecimento pode ser localizado tridimensionalmente de acordo com seu contexto, sua profun-

didática e sua lateralidade.

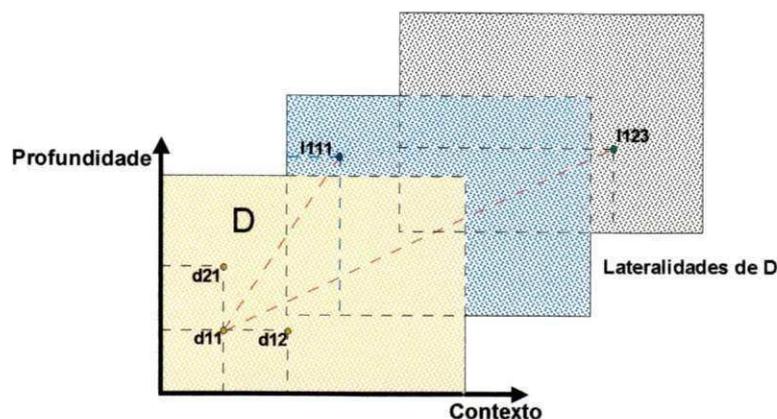


Figura 6.3: dimensões do conhecimento.

* * *

A união da representação aqui proposta, juntamente com o modelo de ambiente proposto no MATHEMA, constitui um esquema bastante promissor em matéria de ensino em Música. No próximo capítulo será demonstrado um dos domínios específicos do ensino de Harmonia e suas interações didáticas.

6.4 Dimensões do Conhecimento em Harmonia

Dentro do domínio da Harmonia, pode-se definir estas três dimensões como um conjunto de elementos, obedecendo a uma organização onde se encontra a definição das diversas fases do ensino (domínios relativos). Entendendo-se o domínio D como sendo o de Harmonia, pode-se considerar, entre outros, os dois contextos seguintes:

C_1 = contexto Harmonia Normativa e

C_2 = contexto Harmonia Funcional.

Cada contexto acima possui níveis de conhecimento (profundidades), desde o mais elementar até o mais complexo. A definição destes níveis garante que o conhecimento seja tratado de maneira gradativa. Por exemplo, ao contexto C_1 podem ser associadas as seguintes profundidades relativas ao conceito de acordes:

P_{11} = acordes maiores, menores, aumentados e diminutos compostos por três notas (terças superpostas) e em posição fundamental,

P_{12} = acordes compostos por quatro notas em posição fundamental,

P_{13} = acordes em posições invertidas (terças no baixo),

P_{14} = acordes com omissão de uma das notas (terça ou quinta),

P_{15} = acorde com omissão de fundamental,

P_{16} = acordes compostos por cinco ou mais notas (9^a , 11^a , 13^a , etc)

P_{17} = acordes com notas estranhas,

P_{18} = acordes dependentes do contexto,

Estes oito exemplos de profundidade podem ser ampliados ou reorganizados de acordo com o objetivo didático almejado pelo especialista.

Uma vez definidos os pares $\langle C_i, P_{ij} \rangle$, associa-se cada um deles a um subdomínio d_{ij} . Daí, para cada par, deve-se identificar suas lateralidades. A lateralidade pode ser definida também, como os “pré-requisitos” externos que cada d_{ij} necessita para seu entendimento.

Seguindo ainda o exemplo de profundidades do conceito de acordes, têm-se as lateralidades, onde para cada subdomínio d_{ij} associa-se uma lateralidade l_{ijk} . Dessa forma, tem-se a seguinte relação:

l_{111} = conceitos de notas (*classe notas*),

l_{112} = conceitos de intervalos (*classe intervalos*),

l_{113} = conceitos de escalas (*classe escalas*),

No exemplo do conceito de acorde (vide item 2.2.4) existe um:

$$\langle C_1, P_{11} \rangle = d_{11}, \text{ onde}$$

d_{11} = {conceito de acordes (tríades), exemplos, contra-exemplos, exercícios resolvidos, problemas, dicas} e suas lateralidades

l_{111} = {conceito de nota},

l_{112} = {conceito de intervalo} e

l_{113} = {conceito de escala}.

Os elementos que compõem d_{11} são os seguintes:

- *conceito de acorde*: composto pela *classe acordes* (vide item 5.4.5) e suas variantes,
- *exemplos*: tríades dos acordes formados na escala de dó maior (figura 6.4). Cabe ao

especialista definir se deseja ou não adicionar mais exemplos e quais.

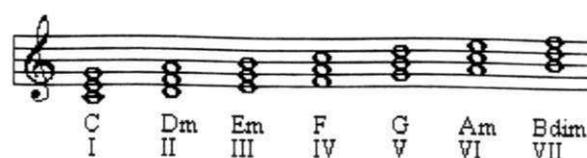


Figura 6.4: exemplos de acordes em dó maior

- *contra-exemplos*: tríades sem terças superpostas, neste caso inúmeros contra-exemplos podem ser adicionados (figura 6.5),



Figura 6.5: contra-exemplos de tríades que não formam acordes

- *exercícios resolvidos*: aqui pode-se fazer a associação entre o símbolo do acorde (C, Am, F, etc) e a sua representação gráfica e vice-versa,
- *problemas*: similar ao item acima, sendo que neste caso expõe-se a representação gráfica e pede-se o símbolo ou faz-se o mesmo em sentido inverso, ou apresenta-se acordes incompletos e pede-se o complemento, etc.
- *dicas*: tais como: acordes maiores são formados por uma terça maior e outra menor; acordes menores são formados por uma terça menor e outra maior, etc.

As lateralidades apresentadas acima estão respectivamente ligadas às *classes nota, intervalo* e *escalas*. Vale lembrar que um determinado conceito, dependendo o objetivo, pode ser uma lateralidade ou fazer parte de um subdomínio. No caso da lateralidade l_{111} (conceito de nota), por exemplo, quando este mesmo conceito for tratado como subdomínio, suas lateralidades serão os conceitos de *nome, oitava, duração* e *alteração*.

O próximo par:

$$\langle C_1, P_{12} \rangle = d_{12}, \text{ onde}$$

$d_{12} = \{\text{conceito de acordes (tétrades), exemplos, contra-exemplos, exercícios resolvidos, problemas, dicas}\}$ e suas lateralidades

$l_{121} = \{\text{conceito de nota}\},$

$l_{122} = \{\text{conceito de intervalo}\},$

$l_{123} = \{\text{conceito de escala}\},$

$$l_{124} = \{\text{conceito de sétima}\}.$$

Os componentes de d_{12} são similares ao d_{11} e à lateralidade pode ser acrescida o conceito de sétima (sensível). Este mesmo esquema deve ocorrer com os outros pares e o conjunto destes que, juntamente com as lateralidades, formam a visão completa do contexto abordado.

6.5 Interação Aprendiz × Tutor

Conforme a Figura 6.6, a interação entre o Aprendiz e o Tutor acontece na forma de diálogo. O Aprendiz envia uma mensagem com conteúdo X ao Tutor e este, por sua vez, envia uma resposta de retorno com conteúdo Y. Para maiores detalhes sobre o modelo de interação, ver [COSTA, 1997b].

No caso do Aprendiz, pode-se ter como mensagem as seguintes opções: um pedido de consulta, de diagnóstico, a solução de um problema,...

Já o Agente Tutor pode responder de diversas maneiras: através de um esclarecimento, uma atividade de ensino (exercícios, dicas, instruções, exemplos), um recurso sonoro ou de hipertexto, uma solução, um diagnóstico, etc.

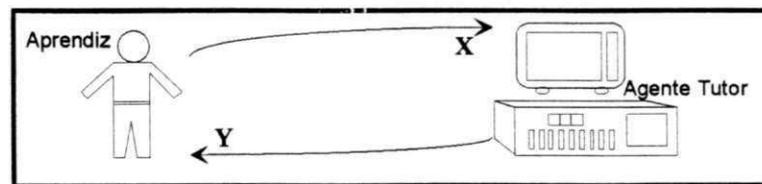


Figura 6.6: interação entre Aprendiz e Tutor.

Para dar uma idéia das funcionalidades do ambiente no domínio da Harmonia, será descrita uma situação particular, onde o aprendiz interage com o agente tutor, visando resolver o encadeamento de um “baixo dado”⁵⁶, apresentado na Figura 6.6. Seu objetivo é, além de harmonizar (cifras e baixo dados), escrever as vozes restantes (soprano, contralto e tenor). Antes porém, serão destacados, como exemplo, alguns dos agentes em SATA, são eles:

- (i) AT_{11} = Agente de Acordes (tríades),
- (ii) AT_{12} = Agente de Acordes (tétrades),

⁵⁶ *Baixo dado* é a expressão utilizada para designar um tipo de exercício de harmonia composto apenas pelos dados da voz mais grave (baixo). De posse dessa informação, o aluno deve escrever as vozes restantes, tendo que obe-

- (iii) AT_{13} = Agente de Acordes (inversões),
- (iv) AT_{14} = Agente de Acordes (omissão de uma das notas).

Os agentes AT_{11} , AT_{12} e AT_{14} estão relacionados ao conceito de acorde representado no capítulo 5⁵⁷, conforme visto no exemplo do item 6.3. Já os agentes enumerados a seguir são os responsáveis pelas lateralidades e estão associados a um par $\langle C_1, P_{12} \rangle$. estão associados a outros conceitos relativos à resolução do problema exposto abaixo:

- (v) AT_{15} = Agente de Cifras,
- (vi) AT_{16} = Agente de Distribuição de Vozes,
- (vii) AT_{17} = Agente de encadeamento ($\langle I-V, I-IV, V-I, IV-I \rangle$),
- (viii) AT_{18} = Agente de encadeamento ($\langle I-V, I-IV \rangle$) com duplicação da quinta no primeiro acorde,
- (ix) AT_{19} = Agente de encadeamento ($\langle IV-V, V-IV \rangle$) com duplicação da fundamental no primeiro acorde,
- (x) AT_{21} = Agente de encadeamento ($\langle IV-V, V-IV \rangle$), com duplicação da quinta no primeiro acorde, etc [HINDEMITH, 1949; KOSTKA, 1984], para corresponder à evolução da lateralidade do material harmônico no decorrer das sessões de ensino/aprendizagem

...

Nesta perspectiva, chega-se ao esboço inicial de uma sociedade de agentes tutores, considerando-se os agentes enunciados anteriormente.

6.6. Ilustração da interação entre o Aprendiz e o Agente Tutor

O modelo de interação aqui adotado é baseado em diálogos cooperativos entre o Aprendiz e o Agente Tutor, onde o diálogo acontece através de um esquema de troca de mensagens. Os tipos de mensagens utilizados incluem: resolução de problemas, diagnóstico, consultas, dicas, entre outros. Para ilustrar as funcionalidades deste modelo, estão descritas a seguir algumas situações particulares de diálogos, onde o Aprendiz interage com o Agente Tutor, visando resolver um problema relacionado ao encadeamento de um “baixo dado”, apresentado na Figura 6.7.

decer a determinados parâmetros harmônicos para esta resolução.

⁵⁷ O métodos apresentados na classe *notas* são responsáveis pela formação das tríades, das tétrades e, no caso de

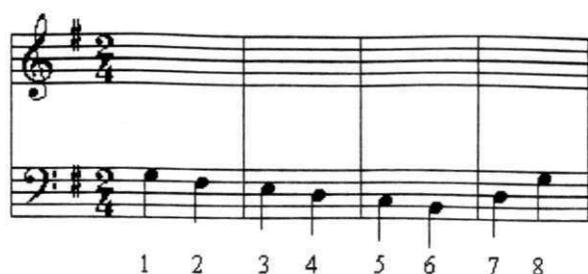


Figura 6.7: problema a ser resolvido

Seu objetivo é, além de harmonizar (cifras), escrever as vozes restantes (soprano, contralto e tenor). Como cada nota possui um acorde correspondente e cada passagem entre acordes situações diferentes, serão destacados, como exemplo, alguns dos agentes em SATA de acordo com estas situações:

Situação 1

AT₁₁ *Identifique o primeiro acorde.* Esta identificação deve estar de acordo com a voz do baixo já escrita;

Aprendiz Pede uma consulta (dica) para verificar a tonalidade da peça;

AT₁₁ Como a primeira nota pode ser interpretada de diferentes maneiras, de acordo com uma tonalidade escolhida, é razoável o pedido desta dica. Dessa forma, o agente AT₁₁ deverá recorrer ao agente AT₁₁₁, responsável pela explicação de tonalidade. Como resultado dessa cooperação, é dada a seguinte dica:

Lembre-se que a armadura possui apenas um sustenido e que a nota do baixo é um sol (g)

Com esta informação, o agente AT₁₁ exhibe a dica solicitada ao aprendiz.

Este trabalho de escolha de cifras será feito para cada uma das notas apresentadas na Figura 6.6. Munido da informação de tonalidade, o Aprendiz pode expor a solução para cada um dos acordes.

Situação 2

Aprendiz Indica que não conseguiu entender a dica apresentada.

AT₁₁ Pergunta ao Aprendiz:

Você pode escolher entre: (i) Uma explicação sobre a dica dada e (ii) Uma nova instrução sobre o assunto

Aprendiz Opta por uma nova instrução sobre o assunto.

AT₁₁ Faz a chamada ao agente responsável pelo ensino de Tonalidade.

Situação 3

Aprendiz Devolve como resposta para a quarta nota a seguinte cifra: III6

AT₁₁ Por se tratar de uma cifra que está além do âmbito pretendido por este agente, o mesmo devolve a resposta:

Você poderia indicar outra cifra.

Aprendiz Neste momento, o Aprendiz resolve manter sua resposta: III6

AT₁₁ A partir desse ponto o agente busca a cooperação do agente responsável pelas cifras com inversão (III6), passando a resposta ao agente AT₁₃.

AT₁₃ Com este dado, o agente AT₁₃ devolve como resposta para AT₁₁:

Resposta correta.

AT₁₁ Com a resposta do AT₁₃, o aprendiz recebe a seguinte mensagem:

Resposta correta, muito embora a mesma faça parte de uma outra unidade de conhecimento.

Situação 4

AT₁₁ Identifique o primeiro acorde.

Aprendiz Devolve como resposta o seguinte cifra: I (pertencente ao contexto da Harmonia Funcional;

AT₁₁ É devolvida ao Aprendiz a seguinte resposta:

Esta cifra está relacionada à Harmonia Funcional, você deseja:

(i) fornecer outra resposta;

(ii) converter a cifra para a Harmonia Normativa ou

(iii) mudar de contexto?

Dependendo da intenção do Aprendiz o diálogo irá se desenvolver, seja no contexto da Harmonia Funcional ou Normativa.

7 Conclusão

A representação do conhecimento musical apresentada é uma tentativa de refletir os aspectos necessários e relevantes para o ensino por computador da Harmonia Musical, conforme os principais métodos e material didático adotados na maioria das escolas [PRIOLLI, 1985; CHEDIAK, 1986; HINDEMITH, 1988].

Partindo de objetivos didáticos predefinidos, pode-se escalonar o material de trabalho dado ao aluno, assim como priorizar determinados objetos e regras. Mesmo a ampliação do material de trabalho (vozes) ou do conhecimento como um todo (regras) é prevista, sem a necessidade de reprogramação. Tal ampliação pode ser feita alterando-se as bases de conhecimento contidas nas classes envolvidas, ou acrescentando classes específicas.

Outro fator de interesse da representação proposta neste trabalho é o fato da mesma poder ser direcionada tanto para a escola Normativa, quanto Funcional. Uma outra consequência é tornar possível o seu uso na área de Análise, em concordância com os trabalhos propostos em [FERNEDA, 1994; GUIGUE, 1995], fazendo parte de um Sistema de Apoio à Descoberta em Análise Musical [FERNEDA, 1994].

Com relação à visão do conhecimento em dimensões, proposta no modelo MATHEMA, e a representação apresentada, pode-se notar uma certa naturalidade no mapeamento da representação para o modelo. Esta naturalidade encontra-se na proposta de fracionamento do conhecimento (a primeira utilizando uma abordagem de *agentes* e a segunda a de *classes*), que dessa forma permite a manipulação didático-pedagógica do conhecimento. Esta união promove uma concretização de um compromisso de qualidade no que diz respeito à organização e representação do conhecimento, e aos mecanismos de interação entre as entidades no processo de ensino.

Por outro lado, com respeito à escolha da Orientação a Objetos juntamente com a Programação em Lógica como formalismo para representação do conhecimento⁵⁸, foi consequência direta do caminho adotado durante a estruturação do conhecimento musical indispensável ao estudo da Harmonia. Dessa forma, ficaram patentes as qualidades de tais paradigmas no que diz respeito à representação do conhecimento musical e à resolução de

⁵⁸ A utilização de *frames* também é adequada para representar um domínio de conhecimento como a Harmonia.

alguns dos principais problemas relativos ao ensino de música como um todo, quais sejam:

- resolver exercícios harmônicos no que diz respeito à condução das vozes;
- fazer arranjos corais a quatro vozes,
- encontrar respostas completas a partir de objetos musicais incompletos,
- tratar adequadamente a questão das enarmonias e
- permitir a manipulação do conhecimento de acordo com a necessidade imposta pela relação tutor-aprendiz.

Já a linguagem LPA-Prolog++ mostrou-se interessante no que diz respeito à especificação, muito embora não seja este o seu intuito inicial. Suas ferramentas de controle, como por exemplo o *Browse* (que permite visualizar as relações entre as classes) são de grande utilidade quando do seu uso em especificação. O LPA-Prolog possibilita ainda a construção, em uma fase posterior, da interface, já existindo toda uma biblioteca criada para facilitar esta tarefa.

Partindo do que foi aqui exposto, alguns trabalhos estão previstos como continuidade a esta dissertação:

- (i) compilação e representação das regras gerais da Harmonia Normativa,
- (ii) compilação e representação das regras gerais da Harmonia Funcional e
- (iii) a construção de um sistema tutor em Harmonia.

O primeiro trabalho já se encontra iniciado e que, por sua vez, servirá de base as regras funcionais. Já a construção de um sistema tutor como um todo deverá envolver um grupo maior de pesquisadores.

Por fim, vale a pena lembrar que muito dos sistemas atuais que trabalham IA e Música se utilizam da Orientação a Objetos [BALABAN, 1992; POPE, 1991].

Referências Bibliográficas

- AIKIN, J., "Gravadores Multitrack Windows: áudio profissional chega em casa", Revista Keyboard, Ano 1, nº 3, pp. 43-52, São Paulo, SP, 1997.
- ALMEIDA, N. C., "CAC - MIDI - The Computer without algorithm", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 234-239, Canela, RS, 1995.
- ANDRADE, P., TEIXEIRA, L. de M., "Computabilidade das Redes Neurais", Monografia da disciplina de Teoria da Computação, COPIN-DSC, UFPB, 1995.
- ANSI (American National Standards Institute), "X3V1.8M/SD-6 Journal of Development Standard Music Description language - SMDL", Computer Music Association, San Francisco, EUA, 1989.
- ASSAYAG, G., CHOELLETON J-P, "Música, Números e Computadores", Revista Ciência Hoje, Vol. 20, nº 116, pp. 38-46, São Paulo, SP, 1996.
- BACON, F., "Nova Atlântida", Coleção os Pensadores, Abril Cultural, Rio de Janeiro, RJ, 1984 (3ª edição).
- BALABAN, M., EBCIOGLU, K., LASKE, O., (Eds) "Understanding Music with IA", AAAI Press/MIT Press, Menlo Park, Califórnia, EUA, 1992.
- BECKENKAMP, F. G., Engel, P. M., "A Connectionist Model for Chord Classification", Anais do I Simpósio Brasileiro de Computação e Música, pp. 189-193, Caxambu, MG, 1994.
- BECKENKAMP, F. G., Engel, P. M., "Um Modelo Inteligente para Classificação Harmônica Tonal", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 121-131, Canela, RS, 1995.
- BEL, B., "A Symbolic-numeric Approach to Quantization in Music", Anais do III Simpósio Brasileiro de Computação e Música, pp. 2-7, Recife, 1996.
- BRANDÃO, M. da C. P., Sambuichi M. R., "Software Auto-Instrucional em Teoria Musical", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 263-266, Canela, RS, 1995.
- BRISOLLA, C. M., "Princípios de Harmonia Funcional", Editora Novas Metas, São Paulo, SP, 1979.
- CARPEAUX, O. M., "Uma Nova História da Música", Ediouro, Rio de Janeiro, RJ, 1958.
- CARPINTEIRO, O. A. S., "A Neural Model to Segment Musical Pieces", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 114-120, Canela, RS, 1995.
- CARVALHO, R., "Teoria Musical III: Parâmetro Altura", Fundação Cultural Monsenhor Chaves, Teresina, PI, 1994.
- CHEDIAK, A., "Harmonia & Improvisação", Lumiar Editora, Rio de Janeiro, RJ, 1985.

- COPLAND, A., "*A nova Música*", Editora Record, Rio de Janeiro, RJ, 1969.
- COSTA, A. C. R., "*Sobre os Fundamentos da Inteligência Artificial*", Relatório de Pesquisa - CPGCC-UFRGS, Porto Alegre, RS, 1986.
- COSTA, E. de B., LOPES, M. A., FERNEDA, E., "*MATHEMA: A Learning Environment Based on a Multi-Agent Architecture*", Proceedings of the 12nd Brazilian Symposium on Artificial Intelligence, pp. 141-150, Campinas, SP, 1995.
- COSTA, E. de B., PERKUSICH, A., "*Modelling the Cooperative Interaction in a Teaching/Learning Situation*", Proceedings of the Third International Conference on Intelligent Tutoring Systems - ITS'96, Montreal, Canadá, 1996a.
- COSTA, E. de B., TEIXEIRA, L. de M., SANTOS, V. L. P., FERNEDA, E., "*Ensino de Harmonia Musical em um Ambiente Interativo de Aprendizagem*", Anais do III Simpósio Brasileiro de Computação e Música, pp. 62-67, Recife, 1996b.
- COSTA, E. de B., TEIXEIRA, L. de M., FERNEDA, E., "*Um Sistema Tutor Multi-Agente em Harmonia Musical*", Anais do Simpósio Brasileiro de Informática na Educação, São José dos Campos, SP, 1997a. (artigo submetido)
- COSTA, E. de B., PERKUSICH, A., "*A Multi-Agent Interactive Warning Environment Model*", *Artificial Intelligence in Education (AI-ED97): Workshop on Pedagogical Agents*, Kobe, Japão, august, 1997b.
- DANNENBERG, R. B., "*Music Representation Issues, Techniques, and Systems*", *Computer Music Journal*, Vol. 17, nº 3, pp. 20-30, MIT Press, EUA, 1994.
- DANNENBERG, R. B., "*A Perspective on Computer Music*", *Computer Music Journal*, Vol. 20, nº 1, pp. 52-56, MIT Press, EUA, 1996.
- FERNEDA, E., SILVA, C. A. P. da, TEIXEIRA, L. de M., SILVA, H. de M., "*A System for Aiding Discovery in Musical Analysis*", Anais do I Simpósio Brasileiro de Computação e Música, pp. 177-184, Caxambu, MG, 1994.
- FIREBAUGH, M. W., "*Artificial Intelligence: A Knowledge-Based Approach*", PWS-Kent Publishing Company, Boston, EUA, 1988.
- FRITSCH, E. F., "*Compositor Automático de Melodias Musicais - CAMM*", Dissertação de Mestrado - CPGCC-UFRGS, Porto Alegre, RS, 1994.
- FRITSCH, E. F., VICCARI, R. M., "*SETMUS: Uma Ferramenta Computacional para o Ensino da Música*", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 267-273, Canela, RS, 1995.
- GLANZMANN, J. H., SANTOS, N., ROCHA, A. R., "*Expert Piano: Um Ambiente para Auxiliar o Estudo de Piano e Música*", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 281-299, Canela, RS, 1995.
- GLEICK, J., "*Caos - A Criação de Uma Nova Ciência*", Editora Campus, Rio de Janeiro, RJ, 1991.
- GOEBEL, J., "*Freedom and Precision of Control*", *Computer Music Journal*, Vol. 20, nº 1, pp. 46-48, MIT Press, EUA, 1996.

- GOODMAN, D., "*The Complete Hypercard 2.0 Handbook*", Bantam Doubleday Dell Publishing Group, New York, EUA, 1990. (citado em [Fritsch 95])
- GOODMAN, N., "*The Languages of Art: An Approach to a Theory of Symbols*", Bobbs-Merrill Co., Indianapolis, EUA, 1968.
- GUIGUE, D., "*A Computer-aided Object-oriented Analysis*", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 161-168, Canela, RS, 1995.
- GUIMARÃES, M. C., "*Harmonia Teórica e Prática pela Teoria dos Conjuntos*", Oficina de Livros, Belo Horizonte, MG, 1990.
- HINDEMITH, P., "*Harmonia Tradicional*", Irmãos Vitale, São Paulo, 1986.
- HINDEMITH, P., "*Treinamento Elementar para Músicos*", Ricordi Brasileira S/A, São Paulo, SP, 1988 (4ª edição).
- HOARE, C. A. R., "*Studies in Data Processing. Structured Programming*", em *Notes on Data Structuring*, editado por O. J. Dahl, E. W. Dijkstra, e C. A. R. Hoare, Academic Press, New York, EUA, 1972.
- HONING, H., "*Issues in the representation of time and structure in music*", Contemporary Music Review, nº 9, 1993. (<ftp://mars.let.uva.nl/honing/PAPERS/H-93-B>. HTML).
- IMA, "*Midi Specification 1.0*", IMA, 1983.
- JACKSON, P., "*Introduction to Expert Systems*", Addison-Wesley, Londres, RU, 1986.
- KAPLAN, J. A., "*Harmonia*", *Notas de Aula*, Departamento de Música, Universidade Federal da Paraíba, João Pessoa, PB, 1991.
- KOSTKA, S., PAYNE, D., "*Tonal Harmony: with an Introduction to Twentieth-Century Music*", Alfred A. Knopf, New York, EUA, 1984.
- LASKE, O., "*Comment on the First Workshop on IA and Music in Minnesota*", In Perspective of New Music, Nº 2, Vol. 27, Washington, EUA, 1989.
- LITTLE, D. C., "*Composing with Chaos: applications of a new science for music*", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 44-61, Canela, RS, 1995.
- LOVELOCK, W., "*História Concisa da Música*", Martins Fontes, São Paulo, SP, 1987.
- MALT, M., "*Modelos Matemáticos e Composição Assistida por Computador, Sistemas Estocásticos e Sistemas Caóticos*", Anais do I Simpósio Brasileiro de Computação e Música, pp. 125-132, Caxambu, MG, 1994.
- MANZOLLI, J., "*The Development of a Gesture Interfaces Laboratory*", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 81-84, Canela, RS, 1995.
- MANZOLLI, J., OTSHUKI, W., "*INTERASOM: A Desktop for Algorithmic Composition*", Anais do III Simpósio Brasileiro de Computação e Música, pp. 84-88, Recife, PE, 1996.
- MARTÍN, P. S., "*Una Experiência de Diseño y Programación de Software para Educación Musical*", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 274-280, Canela, RS, 1995.

- MARTINS, A. J. B., BALLISTA, A. L. C., PIMENTA, M. S., "Um ambiente de Auxílio à Composição Musical", Anais do I Simpósio Brasileiro de Computação e Música, pp. 139-146, Caxambu, MG, 1994.
- MCDONALD, F. J., "Educational Psychology", Wadsworth Publishing Company, California, 1965.
- MINSKY, M., "A Framework for Representing Knowledge", In: *The Psychology of Computer Vision*, editado por P. Winston, McGraw-Hill, New York, EUA, 1975.
- MINSKY, M., "The Society of Mind", Simon and Schuster, New York, EUA, 1985.
- MINSKY, M., LASKE, O., "A Conversation with Marvin Minsky", AI Magazine, fall, pp. 31-45, 1992.
- MIRANDA, E. R., "Música e Inteligência Artificial (Paradigmas e Aplicações)", Trabalho Individual, CPGCC-UFRGS, Porto Alegre, 1990.
- MIRANDA, E. R., "ARTIST: an AI-based Tool for the Design of Intelligent Assistants for Sound Synthesis", Anais do I Simpósio Brasileiro de Computação e Música, Caxambu, MG, 1994.
- MIRANDA, E. R., "Machine Learning of Sound Attributes: Computer-Assistance in Concept Formation and Musical Invention", Anais do III Simpósio Brasileiro de Computação e Música, pp. 8-11, Recife, PE, 1996.
- MONRO, G., "The Algorithmic Composition of "Dry Rivers"", Anais do III Simpósio Brasileiro de Computação e Música, pp. 89-94, Recife, PE, 1996.
- MOORE, F. R., "Dreams of Computer Music - Then and Now", Computer Music Journal, Vol. 20, nº 1, pp. 25-41, MIT Press, EUA, 1996.
- MOSS, C., "Prolog++: The Power of Object-Oriented and Logic Programming", Addison-Wesley Publishing Company, Cambridge, Inglaterra, 1995.
- MOUTON, R., PACHET, F., "Numeric vs Symbolic Controversy in Automatic Analysis of Tonal Music", Proceedings of the IJCAI'95 - Workshop on Artificial intelligence and Music, 1995. (<http://www.laforia.ibp.fr/~fdp/MyPapers/MuES/IJCAI-mouton-final.ps.Z>).
- PACHET, F., "A Meta-level Architecture Applied to the Analysis of Jazz Chord Sequences", Proceedings of the 10th International Conference on Computer Music, Montreal, Canadá, 1991a. (<http://www.laforia.ibp.fr/~fdp/MyPapers/MuES/ICMC.RL.ps.Z>).
- PACHET, F., "Representing Knowledge Used by jazz Musicians", Proceedings of the International Conference on Computer Music, Montreal, Canadá, 1991b. (<http://www.laforia.ibp.fr/~fdp/MyPapers/>).
- PACHET, F., "An Object-Oriented Representation of Pitch-Classes, intervals, scales and chords", Journées d'informatique Musicale, Bordeaux, França, 1994a.
- PACHET, F., "The MusEs System: an Environment for Experimenting with Knowledge Representation Techniques in Tonal Harmony", Anais do I Simpósio Brasileiro de Computação e Música, pp. 195-201, Caxambu, MG, 1994b.

- PÉREZ, M. J., FORNARI, J. E., DAMIANI, F., "Reconhecimento de Timbres Musicais através de rede neural auto-organizável de Kohonen", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. , 107-113, Canela, RS, 1995.
- PÉREZ, M., "Diccionario de la Música y los Músicos", Ediciones ISTMO, Madrid, Espanha, 1985.
- PISTON, W., "Counterpoint", Victor Gollancz LTD., Londres, RU, 1982.
- PONTES, E. V., COSTA, E. B., & all, "HIPERPLAN: Um Ambiente de Aprendizagem Baseado em Hipertextos e Planos", Proceedings of the Iberoamerican Symposium on Informatics and Education, Santo Domingo, República Dominicana, 1992.
- POPE, S. T., (Ed) "The Well-Tempered Object: Musical Applications of Object-Oriented Software Technology", MIT Press, EUA, 1991.
- POPE, S. T., "A Taxonomy of Computer Music", Computer Music Journal, Vol. 18, nº 1, pp. 5-7, MIT Press, EUA, 1994.
- PRIOLLI, M. L. de M., "Princípios Básicos da Música para a Juventude", Editora Casa Oliveira de Músicas LTDA., Rio de Janeiro, RJ, 1985.
- RAMALHO, G., "Representing Musicians' Actions for Simulating Improvisation in Jazz", Anais do I Simpósio Brasileiro de Computação e Música, pp. 217-222, Caxambu, MG, 1994.
- RATTON, M. B., "MIDI - Guia Básico de Referência", Editora Campus, Rio de Janeiro, RJ, 1992.
- REITZ, Ph., "Contribution à l'étude des environnements d'apprendissage, Conceptualisation, Spécification et Prototypage", Tese de Doutorado, Université Montpellier II, Montpellier, França, 1992.
- RICH, E., KNIGHT, K., "Inteligência Artificial", Makron Books, São Paulo, SP, 1993.
- RINGLAND G., DUCE, D., "Approaches to Knowledge Representation - An Introduction", Research Studies Press LTD., Tauton, EUA, 1988.
- ROADS, C., "Research in Music and Artificial Intelligence", Computing Surveys, Vol. 17, nº 2, pp. 163-190, MIT Press, EUA, 1985.
- RUBIN, D. M., "O Músico Desktop: criando música com seu computador", Makron Books, São Paulo, SP, 1996.
- RUMELHART, D., MCCLELLAND, J., "Parallel Distributed Processing", pp. 318-334, MIT Press, EUA, 1986. (citado em [BECKENKAMP, 1994])
- SCHANK, R., RIESBECK, C., "Inside Case-base Reasoning", Lawrence Erlbaum Association Pub., New Jersey, EUA, 1989. (citado em [RAMALHO, 1994])
- SCHONBERG, A., "Tratado de Armonía", Real Musical, Madri, Espanha, 1979.
- SLOMAN, A., "Epistemology and Artificial Intelligence", em *Expert Systems in the Micro-electronic Age*, editado por D. Michie, Edinburgh University Press, Edinburgh, RU, 1979.

- SMOLIAR, S. W., "Representing Listening Behavior: Problems and Prospects", In: [BALABAN, 1992 - pp. 53-63].
- SPIEGEL, L., "That was Then - This is Now", Computer Music Journal, Vol. 20, n° 1, pp. 42-45, MIT Press, EUA, 1996.
- TEIXEIRA, L. de M., COSTA, E. de B., SILVA, C. A. P. da, FERNEDA, E., "Aquisição do Conhecimento em Harmonia: Um ambiente de Aprendizagem", Proceedings of the 2nd Brazilian Symposium on Computer Music, pp. 290-296, Canela, RS, 1995.
- TEIXEIRA, L. de M., ALEIXO, F. A., CAMINHA, A. de O., COSTA, E. de B., FERNEDA, E., "A Noção de Ponto de Vista Aplicada à Representação de Eventos Musicais", Anais da VI Semana de Informática da UFBA - SEMINFO'96, pp. 67-78, Salvador, BA, 1996.
- TEIXEIRA, L. de M., COSTA, E. de B., FERNEDA, E., "Representação de Conhecimento Musical aplicada à Harmonia", Anais do I Encontro Nacional de Inteligência Artificial, SBC, Brasília, DF, 1997.
- VASEY P., "LPA-Prolog++ 2.0, Programmer's Reference", Logic Programming Associates Ltd., Londres, Inglaterra, 1995.
- WENGER E., "Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge", Morgan Kaufman, Los Altos, EUA, 1987.
- WIDMER, G., "Qualitative Perception Modeling and Intelligent Musical Learning", Computer Music Journal, Vol. 16, n° 2, pp. 51-83, MIT Press, EUA, 1992.
- WIGGINS, G., MIRANDA, E., SMAILL, A., HARRIS, M., "A Framework for the Evaluation of Music Representation Systems", Computer Music Journal, Vol. 17, n° 3, pp. 31-42, MIT Press, EUA, 1993.

Anexo 1 - Taxionomia em Computação e Música

Neste anexo é descrita uma taxionomia em Computação e Música, compilada por Pope [POPE, 1994]. Esta taxionomia abrange os mais importantes temas em Computação e Música.

1. Teoria Musical, Composição e Performance

1.1. Teoria Musical, Sociologia e Estética

- 1.1.1. Teoria Musical e Análise
- 1.1.2. Sistemas de Temperamento e Ajustagem
- 1.1.3. Novas Estéticas Musicais e Sociologia

1.2. Composição e Música Eletroacústica

- 1.2.1. Sons, Modelos de Composição e Notação
- 1.2.2. Modelos de Composição e Processos de Performance
- 1.2.3. *Design* de Som e Processamento
- 1.2.4. Realização e Produção de Técnicas
- 1.2.5. Essência da Interpretação (sonificação)

1.3. Composição Algorítmica e Composição Assistida por Computador

- 1.3.1. Composição Algorítmica e Linguagens
- 1.3.2. Sistemas de Composição para Partituras ou Síntese Sonora
- 1.3.3. Inteligência Artificial e Composição

1.4. Performance e Interfaces

- 1.4.1. Interpretação e Condução (Regência)
- 1.4.2. Reconhecimento Gestual e Interfaces
- 1.4.3. Performance e Leitura de Partituras
- 1.4.4. Representação de Expressões e Análise

2. Acústica, Psicoacústica, Percepção e Cognição

2.1. Acústica Musical e Psicoacústica

- 2.1.1. Acústica de Instrumentos Musicais e Vozes
- 2.1.2. Psicoacústica
- 2.1.3. Ambientes e Acústica Espacial

2.2. Percepção Musical e Psicologia

- 2.2.1. Fisiologia do Ouvido

- 2.2.2. Percepção Auditiva (Altura)
- 2.2.3. Percepção Rítmica
- 2.2.4. Percepção Auditiva (Timbre)

2.3. Compreensão Musical e Cognição

- 2.3.1. Entendimento Rítmico
- 2.3.2. Reconhecimento de Escalas e Tonalidades
- 2.3.3. Compreensão das estruturas Musicais (Alto-nível)

3. Sinais e Eventos Musicais, Representação e Notação

3.1. Modelos de Sinais e Eventos

- 3.1.1. Sistemas de Linguagem
- 3.1.2. Decodificação e Formato de Arquivos
- 3.1.3. Sistemas de Notação Gráfica

3.2. Linguagens de Descrição de Eventos Musicais

- 3.2.1. Formatos de Listas de Notas
- 3.2.2. Linguagens de Entrada para Música
- 3.2.3. Linguagens de Programação para Música

3.3. Linguagens de Descrição de Símbolos Musicais

- 3.3.1. Modelos Simbólicos e Descrição
- 3.3.2. Linguagens de Síntese (*Software*)

3.4. Notação Musical e Ferramentas de Impressão

- 3.4.1. Transcrição de Performances
- 3.4.2. Reconhecimento Ótico de Partituras

4. Controle Digital, Síntese Sonora e Processamento de Som

4.1. Métodos de Síntese Sonora

- 4.1.1. Métodos de Síntese Sonora Aditiva
- 4.1.2. Métodos de Síntese Sonora Subtrativa
- 4.1.3. Métodos de Síntese Sonora Não-Lineares
- 4.1.4. Modelos Físicos de Sistemas Acústicos
 - 4.1.4.1. *Vários Tipos de Modelos Físicos*
- 4.1.5. Outros Métodos de Síntese
- 4.1.6. Sistemas de Re-síntese e Análise

4.2. Processamento de Sinais (Tempo e Frequência)

- 4.2.1. Arquitetura de *Software*

- 4.2.2. Modelo de Síntese por Domínio do Tempo
- 4.2.3. Modelo de Síntese por Domínio de Frequência
- 4.2.4. Técnicas de Síntese *ad hoc*
- 4.2.5. Efeitos e Filtros

- 4.3. *Localização e Espacialização Sonora*

- 4.4. *Máquina para o Reconhecimento de Símbolos e Eventos*

- 4.5. *Processamento em Tempo real e Programado*
 - 4.5.1. Programação em Tempo Real
 - 4.5.2. Linguagens em Tempo Real
 - 4.5.3. Arquitetura de *Hardware*

- 4.6. *MIDI e Controle de Processamento*

- 5. *Hardware para Instrumentos Musicais e Ferramentas*
 - 5.1. *Hardware para Processamento Digital de Sinais e Áudio*
 - 5.2. *Estações de Trabalho dedicadas à Música Computadorizada*
 - 5.3. *Dispositivos de Entrada/Saída (I/O) para Música*

- 6. *Computação na Educação Musical e Educação Musical Computadorizada*
 - 6.1. *Computadores na Educação Musical*
 - 6.2. *Educação Musical Computadorizada*

- 7. *Literatura Musical Computadorizada, História e Recursos*
 - 7.1. *Bibliografia/Discografia*
 - 7.2. *Boletins de Estúdio*
 - 7.3. *Descrição de Composições*
 - 7.4. *História da Música Eletroacústica*

Anexo 2 - Código em LPA-Prolog++

Para uma melhor visualização da representação e, conseqüentemente, do programa em LPA-Prolog++, segue abaixo o código gerado e comentários.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% classe notas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

class notas.

    inherits oitavas.
    inherits nomes.
    inherits alteracoes.
    inherits duracoes.

    private methods

        nota          /1.

    public methods

        valor_da_nota /2.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: nota/1
% Comentario: metodo privado para verificar se o objeto é uma nota
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nota([Oitava, Nome, Alteracao, Duracao]) :-
    oitavas    <- oitava(Oitava),
    nomes      <- nome(Nome),
    alteracoes <- alteracao(Alteracao),
    duracoes   <- duracao(Duracao).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: valor_da_nota/2
% Comentario: metodo publico para calcular o valor da nota
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

valor_da_nota(nota([Oitava, Nome, Alteracao, Duracao]), Valor_da_Nota) :-
    oitavas    <- valor_oitava(Oitava, Numero_Oitava),
    Valor_Oitava is ( Numero_Oitava - 1 ) * 12,
    nomes      <- valor_posicao(Nome, Valor_Posicao),
    alteracoes <- valor_alteracao(Alteracao, Valor_Alteracao),
    Valor_da_Nota is Valor_Posicao + Valor_Alteracao + Valor_Oitava + 35.

end notas.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% classe oitavas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

class oitavas.

    oitava(o1).
    oitava(o2).
    oitava(o3).
    oitava(o4).
    valor_oitava(o1, 1).
    valor_oitava(o2, 2).
```

```
        valor_oitava(o3, 3).
        valor_oitava(o4, 4).

end oitavas.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% classe nomes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

class nomes.

    nome(p).
    nome(a).
    nome(b).
    nome(c).
    nome(d).
    nome(e).
    nome(f).
    nome(g).

    valor_posicao(c, 1).
    valor_posicao(d, 3).
    valor_posicao(e, 5).
    valor_posicao(f, 6).
    valor_posicao(g, 8).
    valor_posicao(a,10).
    valor_posicao(b,12).

end nomes.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% classe nome_escalas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

class nome_escalas.

    nome_escala(c, n,    _).
    nome_escala(g, n,    _).
    nome_escala(d, n,    _).
    nome_escala(a, n,    _).
    nome_escala(e, n,    _).
    nome_escala(b, n,    _).
    nome_escala(f, #,    _).
    nome_escala(c, #,    _).
    nome_escala(f, n,    _).
    nome_escala(b, -,    _).
    nome_escala(e, -,    _).
    nome_escala(a, -,    _).
    nome_escala(d, -, maior).
    nome_escala(g, -, maior).
    nome_escala(c, -, maior).
    nome_escala(g, #, menor).
    nome_escala(d, #, menor).
    nome_escala(a, #, menor).

end nome_escalas.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% classe alteracoes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

class alteracoes.

    alteracao(n).
    alteracao(-).
    alteracao(=).
    alteracao(#).
    alteracao(+).
```

```

        valor_alteracao(n, 0).
        valor_alteracao(#, 1).
        valor_alteracao(-, -1).
        valor_alteracao(+, 2).
        valor_alteracao(=, -2).

end alteracoes.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% classe duracoes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

class duracoes.

        duracao(1).

end duracoes.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classe de semitom e tom
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

class semitons.

        inherits notas.
        inherits intervalos.

        private methods

                classifica_semitom_tom    /4.
                semitom                    /3.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: classifica_semitom_tom/4
% Comentario: metodo privado para definir se semitom cromatico ou
%             diatonico
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

classifica_semitom_tom(1, Nome1, Nome2, Tipo) :-
        ( Nome1 = Nome2, copy_term(semitom_cromatico, Tipo), ! ) ;
        ( copy_term(semitom_diatonico, Tipo) ).
classifica_semitom_tom( 2, _, _, tom).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: semitom/3
% Comentario: metodo privado para verificar se tom ou semitom
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

semitom(nota([O1,N1,A1,D1]), nota([O2,N2,A2,D2]), Tipo) :-
        notas <- (valor_da_nota(nota([O1,N1,A1,D1]), Valor_da_Nota1),
                valor_da_nota(nota([O2,N2,A2,D2]), Valor_da_Nota2)),
        Diferenca is Valor_da_Nota1 - Valor_da_Nota2,
        intervalos <- modulo(Diferenca, ValorModulo),
        classifica_semitom_tom(ValorModulo, Nome1, Nome2, Tipo).

end semitons.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classe de intervalo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

class intervalos.

        inherits notas.

        public methods

                intervalo    /3,
                modulo        /2.

```

```
private methods
```

```
    grau_do_intervalo /4,
    direcao /9.
```

```
classifica_intervalo_nota(1, unissono, 0).
classifica_intervalo_nota(2, diminuta, 0).
classifica_intervalo_nota(2, menor, 1).
classifica_intervalo_nota(2, maior, 2).
classifica_intervalo_nota(2, aumentada, 3).
classifica_intervalo_nota(3, diminuta, 2).
classifica_intervalo_nota(3, menor, 3).
classifica_intervalo_nota(3, maior, 4).
classifica_intervalo_nota(3, aumentada, 5).
classifica_intervalo_nota(4, diminuta, 4).
classifica_intervalo_nota(4, justa, 5).
classifica_intervalo_nota(4, aumentada, 6).
classifica_intervalo_nota(5, diminuta, 6).
classifica_intervalo_nota(5, justa, 7).
classifica_intervalo_nota(5, aumentada, 8).
classifica_intervalo_nota(6, diminuta, 7).
classifica_intervalo_nota(6, menor, 8).
classifica_intervalo_nota(6, maior, 9).
classifica_intervalo_nota(6, aumentada, 10).
classifica_intervalo_nota(7, diminuta, 9).
classifica_intervalo_nota(7, menor, 10).
classifica_intervalo_nota(7, maior, 11).
classifica_intervalo_nota(7, aumentada, 12).
classifica_intervalo_nota(8, diminuta, 11).
classifica_intervalo_nota(8, justa, 12).
classifica_intervalo_nota(8, aumentada, 13).
classifica_intervalo_nota(9, diminuta, 12).
classifica_intervalo_nota(9, menor, 13).
classifica_intervalo_nota(9, maior, 14).
classifica_intervalo_nota(9, aumentada, 15).
```

```
grau(c, 1).
grau(d, 2).
grau(e, 3).
grau(f, 4).
grau(g, 5).
grau(a, 6).
grau(b, 7).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: intervalo/3
% Comentario: metodo privado para calcular o intervalo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
intervalo(nota([O1,N1,A1,D]),nota([O2,N2,A2,D]),
    intervalo_nota(Grau,Nome,Direcao)) :-
    grau_do_intervalo(nota([O1,N1,_,_]),nota([O2,N2,_,_]),Grau,Direcao),
    notas <- ( valor_da_nota(nota([O1,N1,A1,D]),Valor_da_Nota1),
        valor_da_nota(nota([O2,N2,A2,D]),Valor_da_Nota2) ),
    Diferenca is Valor_da_Nota1 - Valor_da_Nota2,
    modulo(Diferenca,Modulo_Diferenca),
    classifica_intervalo_nota(Grau,Nome,Modulo_Diferenca).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: grau_do_intervalo/4
% Comentario: metodo privado para calcular o grau do intervalo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
grau_do_intervalo(nota([O1,N1,_,_]),nota([O2,N2,_,_]),Grau,Direcao) :-
    direcao(O1,N1,Numero_Oitava1,Valor_do_Grau1,
        O2,N2,Numero_Oitava2,Valor_do_Grau2,Diferenca_Grau),
    ( ( Numero_Oitava1 < Numero_Oitava2,
        copy_term(ascendente,Direcao),
        Grau is 8 - Diferenca_Grau);
      ( Numero_Oitava1 > Numero_Oitava2,
        modulo(Diferenca_Grau,Modulo_Diferenca_Grau),
```

```

        copy_term(descendente,Direcao),
        Grau is 8 - Modulo_Diferenca_Grau);
    ( Numero_Oitava1 = Numero_Oitava2,
      Valor_do_Grau1 < Valor_do_Grau2,
      copy_term(ascendente,Direcao),
      modulo(Diferenca_Grau,Modulo_Diferenca_Grau),
      Grau is Modulo_Diferenca_Grau + 1);
    ( Numero_Oitava1 = Numero_Oitava2,
      Valor_do_Grau1 > Valor_do_Grau2,
      copy_term(descendente,Direcao),
      Grau is Diferenca_Grau + 1);
    ( Numero_Oitava1 = Numero_Oitava2,
      Valor_do_Grau1 = Valor_do_Grau2,
      copy_term(repetido,Direcao),
      Grau is Diferenca_Grau + 1) ).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: direcao/9
% Comentario: metodo privado para calcular a direcao do intervalo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

direcao(O1,N1,Numero_Oitava1,Valor_do_Grau1,
        O2,N2,Numero_Oitava2,Valor_do_Grau2,Diferenca_Grau) :-
    oitavas <- (valor_oitava(O1,Numero_Oitava1),
               valor_oitava(O2,Numero_Oitava2)),
    grau(N1,Valor_do_Grau1),
    grau(N2,Valor_do_Grau2),
    Diferenca_Grau is Valor_do_Grau1 - Valor_do_Grau2.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: modulo/2
% Comentario: metodo publico para calcular o modulo de um valor
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

modulo(Valor,Modulo_do_Valor) :-
    (Valor < 0 , Modulo_do_Valor is Valor * -1, !);
    Modulo_do_Valor is Valor.

end intervalos.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Classe de escala
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

class escalas.

    inherits nome_escalas.
    inherits intervalos.

    public methods

        escala /3.

    private methods

        forma_escala /5.

    classifica_escala(maior,2,[maior,maior,menor,maior,maior,maior,menor]).
    classifica_escala(menor,2,[maior,menor,maior,maior,menor,maior,maior]).
    classifica_escala(harmonica,2,[maior,menor,maior,maior,menor,aumentada,
                                   menor]).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: escala/3
% Comentario: metodo privado para classificar uma escala
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

escala(nota([O,N,A,D]),Modo,[Escala]) :-
    nome_escalas <- nome_escala(N,A,Modo),
    escalas <- classifica_escala(Modo,Grau,Modo2),
    forma_escala(nota([O,N,A,D]),Grau,Modo2,Escala,7),
    !.

```



```

% Comentario: verifica se todas as vozes estao presentes no
% encadeamento e se o numero de notas e' o mesmo
% para todas as vozes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

verifica_vozes([Soprano,Alto,Tenor,Baixo]) :-
    tipo_voz(Soprano, Tipo_Voz1),
    tipo_voz(Alto, Tipo_Voz2),
    tipo_voz(Tenor, Tipo_Voz3),
    tipo_voz(Baixo, Tipo_Voz4),
    difere_voz([Tipo_Voz1,Tipo_Voz2,Tipo_Voz3,Tipo_Voz4]).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: tipo_voz/2
% Comentario: separa voz das notas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tipo_voz(Voz,Tipo) :-
    Voz =.. [Tipo|_].

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Metodo: difere_voz/1
% Comentario: compara se nao existe voz duplicada
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

difere_voz([]).
difere_voz([Cabeca]).
difere_voz([Cabeca|Cauda]) :-
    not member(Cabeca,Cauda),
    difere_voz(Cauda).

end distribuicao.

```